



```

SSSSSSSS  YY  YY  SSSSSSSS  AAAAAA  SSSSSSSS  CCCCCCCC  EEEEEEEEE  FFFFFFFF  CCCCCCCC
SSSSSSSS  YY  YY  SSSSSSSS  AAAAAA  SSSSSSSS  CCCCCCCC  EEEEEEEEE  FFFFFFFF  CCCCCCCC
SS  YY  YY  SS  AA  AA  SS  CC  CC  EE  FF  CC
SS  YY  YY  SS  AA  AA  SS  CC  CC  EE  FF  CC
SS  YY  YY  SS  AA  AA  SS  CC  CC  EE  FF  CC
SSSSSS  YY  YY  SSSSSS  AA  AA  SSSSSS  CC  CC  EEEEEEE  FFFFFFFF  CC
SSSSSS  YY  YY  SSSSSS  AA  AA  SSSSSS  CC  CC  EEEEEEE  FFFFFFFF  CC
SS  YY  YY  SS  AAAAAAAAAA  SS  CC  CC  EE  FF  CC
SS  YY  YY  SS  AAAAAAAAAA  SS  CC  CC  EE  FF  CC
SS  YY  YY  SS  AA  AA  SS  CC  CC  EE  FF  CC
SS  YY  YY  SS  AA  AA  SS  CC  CC  EE  FF  CC
SSSSSSSS  YY  SSSSSSSS  AA  AA  SSSSSSSS  CCCCCCCC  EEEEEEEEE  FF  CCCCCCCC
SSSSSSSS  YY  SSSSSSSS  AA  AA  SSSSSSSS  CCCCCCCC  EEEEEEEEE  FF  CCCCCCCC

```

```

LL  IIIIII  SSSSSSSS
LL  IIIIII  SSSSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SSSSSS
LL  II  SSSSSS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```

(1)	39	HISTORY	; DETAILED
(1)	57	DECLARATIONS	
(1)	90	RETQUOTA	- RETURN CEF QUOTA
(1)	119	EXESDACEFC	- DISASSOCIATE COMMON EVENT CLUSTER
(1)	161	GETCLUSTER	- GET AND VERIFY CLUSTER NUMBER
(1)	196	EXESASCEFC	- ASSOCIATE AND CREATE COMMON EVENT CLUSTER
(1)	440	EXESDLCEFC	- DELETE COMMON EVENT CLUSTER
(1)	505	CEBLOCK	- LOCK COMMON EVEN BLOCK MUTEX FOR WRITE
(1)	526	LOCLUST	- LOCATE CLUSTER GIVEN NAME
(1)	682	ALOSHMCB	- ALLOCATE SHARED MEMORY MASTER COMMON EVENT BLOCK
(1)	769	INC SHMCB REF/DEC SHMCB REF	- ALTER REF CNT ON SHMEM MASTER CEB
(1)	847	EXESSHMCBDEL	- TRY TO DEETE (RELEASE) MASTER CEB IN SH MEM
(1)	915	UNIQUE CEB	- CHECK THAT SH MEM CEB IS UNIQUE
(1)	1007	EXESCEBREFLCK	- ACQUIRE SHMCB REFERENCE COUNT LOCK
(1)	1062	SHMCFTBLK	- ACQUIRE EXCLUSIVE USE OF SHARED MEMORY TABLE
(1)	1115	SHD_FIND_SHB	- FIND SH MEM CONTROL BLOCK FOR SPECIFIC SHD

```
0000 1
0000 2 .TITLE SYSASCEFC - ASSOCIATE, DISASSOCIATE AND DELETE COMMON EVENT FLAG CLU
0000 3 .IDENT 'V04-000'
0000 4
0000 5
0000 6
0000 7
0000 8 *
0000 9 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 10 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 11 * ALL RIGHTS RESERVED.
0000 12 *
0000 13 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 14 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 15 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 16 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 17 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 18 * TRANSFERRED.
0000 19 *
0000 20 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 21 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 22 * CORPORATION.
0000 23 *
0000 24 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 25 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 26 *
0000 27 *
0000 28
0000 29 ++
0000 30 FACILITY: EXECUTIVE, SYSTEM SERVICES
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 ENVIRONMENT:
0000 35
0000 36 --
0000 37
0000 38 .PAGE
0000 39 .SBTTL HISTORY ; DETAILED
0000 40
0000 41 AUTHOR: R. MUSTVEDT CREATION DATE: 3-OCT-76
0000 42
0000 43 MODIFIED BY:
0000 44
0000 45 V03-001 KDM42758 Kathleen D. Morse 05-Jan-1983
0000 46 Add logic to implement the port quota for shared memory
0000 47 common event flag clusters. Change GET REFCNT LCK to the
0000 48 global, EXESCEBREFLCK, for use by the SYSGEN SHARE command.
0000 49 Change EXES$HM_DELETE to EXES$HMCEBDEL for use by SYSGEN, also.
0000 50
0000 51
0000 52
0000 53 . : VERSION
0000 54 01
0000 55
```

```
0000 57 .SBTTL DECLARATIONS
0000 58
0000 59 :
0000 60 : INCLUDE FILES:
0000 61 :
0000 62 :
0000 63 $CEBDEF ; DEFINE COMMON EVENT CONTROL BLOCK
0000 64 $DYNDEF ; DEFINE DATA STRUCTURE TYPES
0000 65 $IPLDEF ; DEFINE IPL VALUES
0000 66 $JIBDEF ; DEFINE JIB OFFSETS
0000 67 $PCBDEF ; DEFINE PCB OFFSETS
0000 68 $PRDEF ; DEFINE PROCESSOR REGISTERS
0000 69 $PRQDEF ; DEFINE INTER-PROCESSOR REQUEST
0000 70 $PRVDEF ; DEFINE PRIVILEGE BITS
0000 71 $$SHBDEF ; DEFINE SHARED MEMORY CONTROL BLOCK
0000 72 $$SHDDEF ; DEFINE SHARED MEMORY COMMON DATA PAGE
0000 73 $$SDEF ; DEFINE STATUS CODE VALUES
0000 74 $STATEDEF ; DEFINE SCHEDULER STATES
0000 75
0000 76 :
0000 77 : EQUATED SYMBOLS:
0000 78 :
0000 79 :
00000004 0000 80 EFN=4 ; EFN ARGUMENT DISPLACEMENT
00000008 0000 81 NAME=8 ; NAME ARGUMENT DISPLACEMENT
0000000C 0000 82 PROT=12 ; PROTECTION MASK DISPLACEMENT
00000010 0000 83 PERM=16 ; PERMANENT INDICATOR DISPLACEMENT
0000 84
0000 85 :
0000 86 : OWN STORAGE:
0000 87 :
0000 88
```

```

0000 90      .SBTTL  RETQUOTA - RETURN CEF QUOTA
0000 91      :
0000 92      :+
0000 93      : FUNCTIONAL DESCRIPTION:
0000 94      : RETQUOTA RETURNS THE QUOTA TO THE ORIGINAL CREATOR OF THE
0000 95      : COMMON EVENT CLUSTER.  IF THE PROCESS IDENTIFIED AS THE CLUSTER
0000 96      : CREATOR HAS BEEN DELETED, THEN NO OPERATION IS PERFORMED.
0000 97      :
0000 98      : CALLING SEQUENCE:
0000 99      : BSB/JSB RETQUOTA
0000 100     :
0000 101     : INPUT PARAMETERS:
0000 102     : R0 - ADDRESS OF COMMON EVENT BLOCK
0000 103     : R2 - PROCESS INDEX OF CREATOR
0000 104     :
0000 105     : OUTPUT PARAMETERS:
0000 106     : R0 - R0 (PRESERVED)
0000 107     :
0000 108     :
0000 109     : RETQUOTA:
0000 110     : DSBINT #IPL$ SYNCH ; RETURN QUOTA
51 0000'DF42 D0 0006 111     : MOVL @W^SCR$GL PCBVEC[R2],R1 ; BLOCK SYSTEM EVENTS
OC A0 60 A1 D1 000C 112     : CML PCBSL_PID(R1),CEBSL_PID(R0) ; CONVERT TO PCB ADDRESS
      08 12 0011 113     : BNEQ 10$ ; CHECK FOR EXISTENCE
51 0080 C1 D0 0013 114     : MOVL PCBSL_JIB(R1),R1 ; NO, IGNORE QUOTA
      34 A1 B6 0018 115     : INCW JIB$W_TQCNT(R1) ; GET JIB ADDRESS
      001B 116 10$: ENBINT ; GIVE BACK QUOTA
05 001E 117     : RSB ; ENABLE SYSTEM EVENTS
      : ; RETURN

```

```

001F 119          .SBTTL  EXESDACEFC - DISASSOCIATE COMMON EVENT CLUSTER
001F 120
001F 121      :++
001F 122      : FUNCTIONAL DESCRIPTION:
001F 123      : EXESDACEFC IMPLEMENTS THE DISASSOCIATE COMMON EVENT CLUSTER
001F 124      : SYSTEM SERVICE. THE ISSUING PROCESS IS DISASSOCIATED WITH
001F 125      : THE CLUSTER SELECTED BY THE SPECIFIED EVENT FLAG NUMBER AND
001F 126      : THE CLUSTER CONTROL BLOCK (CEB) WILL BE RELEASED IF THE REFERENCE
001F 127      : COUNT BECOMES ZERO AND IT IS NOT MARKED PERMANENT.
001F 128
001F 129      : CALLING SEQUENCE:
001F 130      : CALLG  ARGLIST,EXESDACEFC
001F 131
001F 132      : INPUT PARAMETERS:
001F 133      : EFN(AP) - EVENT FLAG NUMBER TO SELECT CLUSTER TO DISASSOCIATE.
001F 134      : R4 - PCB ADDRESS OF CURRENT PROCESS
001F 135
001F 136      : OUTPUT PARAMETERS:
001F 137      : R0 - COMPLETION STATUS
001F 138
001F 139      : COMPLETION CODES:
001F 140      : SSS_NORMAL - SUCCESSFUL COMPLETION
001F 141      : SSS_ILLEFC - ILLEGAL EVENT FLAG NUMBER FOR COMMON EVENT CLUSTER
001F 142      : MUST BE 64-127
001F 143
001F 144      : SIDE EFFECTS:
001F 145      : IF THE REFERENCE COUNT FOR THE SELECTED COMMON EVENT CLUSTER
001F 146      : BECOMES ZERO AND IT IS NOT MARKED AS PERMANENT, IT WILL BE
001F 147      : DELETED.
001F 148
001F 149      :--
001F 150
00000000 151      .PSECT  YSEXEPAGED, BYTE
OFFC 0000 152      EXESDACEFC::
58 11 10 0002 153      .WORD  ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; SAVE REGISTERS R2-R11
58 A448 DD 0004 154      BSBB   GETCLUSTER ; GET CLUSTER INDEX AND VALIDATE
58 A448 D4 0008 155      PUSHL  PCB$EFC2P(R4)(R8) ; PUSH ADDRESS OF CEB FOR THIS CLUSTER
57 026D 30 000C 156      CLRL  PCB$EFC2P(R4)(R8) ; CLEAR POINTER
017F 54 D0 000F 157      BSBW  CEBLOCK ; LOCK MUTEX FOR WRITE
017F 31 0012 158      MOVL  R4,R7 ; REMEMBER PCB ADDRESS
017F 31 0012 159      BRW   DISASOC ; DISASSOCIATE CLUSTER

```

```

0015 161          .SBTTL  GETCLUSTER - GET AND VERIFY CLUSTER NUMBER
0015 162
0015 163 :++
0015 164 : FUNCTIONAL DESCRIPTION:
0015 165 : GETCLUSTER EXTRACTS THE CLUSTER NUMBER FROM THE EFN ARGUMENT,
0015 166 : VALIDATES IT AND SUBTRACTS THE LOCAL EVENT CLUSTER BIAS TO
0015 167 : FORM A COMMON EVENT CLUSTER INDEX.  IF THE CLUSTER NUMBER
0015 168 : IS NOT A VALID COMMON EVENT CLUSTER, AN EXIT IS MADE FROM
0015 169 : THE SERVICE BY ISSUING A RET.
0015 170
0015 171 : CALLING SEQUENCE:
0015 172 : BSB/JSB GETCLUSTER
0015 173
0015 174 : INPUT PARAMETERS:
0015 175 : EFN(AP) - EVENT FLAG NUMBER
0015 176
0015 177 : OUTPUT PARAMETERS:
0015 178 : R8 - COMMON EVENT CLUSTER INDEX
0015 179
0015 180 : COMPLETION CODES:      (ON ERROR RETURN ONLY)
0015 181 : $$$_ILLEFC - ILLEGAL COMMON EVENT CLUSTER
0015 182
0015 183 :--
0015 184
0015 185 GETCLUSTER:
58  04 AC  03  05  EE 0015 186      EXTV    #5,#3,EFN(AP),R8      : GET COMMON EVENT CLUSTER INDEX
      58  02  C2 001B 187      SUBL    #2,R8          : EXTRACT CLUSTER FROM EFN
      01  19  01 001E 188      BLSS    10$          : SUBTRACT BIAS
05  0020 189      RSB          : ILLEGAL CLUSTER NUMBER
      0021 190          : SUCCESSFUL RETURN
      0021 191 10$:  MOVZWL  #$$$_ILLEFC,R0      : SET ERROR CODE
04  0026 192      RET          : AND EXIT SERVICE CALL
      0027 193
      0027 194

```



```

0027 196          .SBTTL EXESASCEFC - ASSOCIATE AND CREATE COMMON EVENT CLUSTER
0027 197
0027 198 :++
0027 199 : FUNCTIONAL DESCRIPTION:
0027 200 : EXESASCEFC IMPLEMENTS THE ASSOCIATE COMMON EVENT CLUSTER SYSTEM
0027 201 : SERVICE. IF THE NAMED COMMON EVENT CLUSTER DOES NOT ALREADY
0027 202 : EXIST, IT WILL BE CREATED. IF IT ALREADY EXISTS, THE PROCESS
0027 203 : WILL BE ASSOCIATED WITH IT, PROTECTION PERMITTING.
0027 204
0027 205 : CALLING SEQUENCE:
0027 206 : CALLG  ARGLIST,EXESASCEFC
0027 207
0027 208 : INPUT PARAMETERS:
0027 209 : EFN(AP) - EVENT FLAG NUMBER TO SPECIFY CLUSTER NUMBER
0027 210 : NAME(AP) - ADDRESS OF STRING DESCRIPTOR FOR NAME OF CLUSTER
0027 211 : PROT(AP) - PROTECTION MASK FOR CLUSTER IF CREATING
0027 212 : PERM(AP) - INDICATOR TO SPECIFY CREATION OF PERMANENT CLUSTER
0027 213 : R4 - PCB ADDRESS
0027 214
0027 215 : IMPLICIT INPUTS:
0027 216 : PCB OF CURRENT PROCESS
0027 217
0027 218 : OUTPUT PARAMETERS:
0027 219 : R0 - COMPLETION STATUS CODE
0027 220
0027 221 : IMPLICIT OUTPUTS:
0027 222 : PCB$EFC2P - POINTER TO COMMON EVENT CONTROL BLOCK IF CLUSTER 2
0027 223 : PCB$EFC3P - POINTER TO COMMON EVENT CONTROL BLOCK IF CLUSTER 3
0027 224
0027 225 : COMPLETION CODES:
0027 226 : $$$_NORMAL          - SUCCESSFUL COMPLETION
0027 227 : $$$_ACCVIO         - LOGICAL NAME INACCESSIBLE
0027 228 : $$$_EXPORTQUOTA   - NO PORT QUOTA
0027 229 : $$$_EXQUOTA       - QUOTA FOR COMMON EVENT BLOCK CREATION EXHAUSTED
0027 230 : $$$_ILLEFC        - ILLEGAL CLUSTER NUMBER (NOT 0,1,2,3)
0027 231 : $$$_IVLOGNAM      - COUNT FOR LOGICAL NAME NOT 1-15
0027 232 : $$$_NOPRIV        - PRIVILEGE VIOLATION
0027 233 : $$$_SHMNOTCNCT    - SHARED MEMORY NOT CONNECTED
0027 234 : $$$_TOOMANYLNAM   - TOO MANY LOGICAL NAME TRANSLATIONS
0027 235
0027 236 : SIDE EFFECTS:
0027 237 : AN IMPLIED DISASSOCIATE IS PERFORMED ON THE SPECIFIED CLUSTER
0027 238 : AND THE PREVIOUSLY ASSOCIATED COMMON EVENT CLUSTER MAY BE
0027 239 : DELETED IF IT IS NOT MARKED PERMANENT AND ITS REFERENCE COUNT
0027 240 : BECOMES ZERO.
0027 241
0027 242 :--
0027 243
0027 244          .ENABL  LSB
0027 245 EXESASCEFC: : ASSOCIATE COMMON EVENT CLUSTER
0027 246          .WORD  ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> : REGISTER SAVE MASK
0027 247          BSBB   GETCLUSTER : GET CLUSTER INDEX AND VALIDATE
0027 248          MOVL  NAME(AP),R9 : GET POINTER TO CLUSTER NAME
0027 249          PUSHL PERM(AP) : SAVE PERMANENT INDICATOR
0027 250          SUBL  #<4*8>,SP : ALLOCATE SPACE FOR NAMES ON STACK
0027 251          BSBW  LOCLUST : LOCATE CLUSTER
0027 252          MOVL  PCB$_JIB(R7),R10 : GET ADDRESS OF JIB
0027 253
0027 254
0027 255
0027 256
0027 257
0027 258
0027 259
0027 260
0027 261
0027 262
0027 263
0027 264
0027 265
0027 266
0027 267
0027 268
0027 269
0027 270
0027 271
0027 272
0027 273
0027 274
0027 275
0027 276
0027 277
0027 278
0027 279
0027 280
0027 281
0027 282
0027 283
0027 284
0027 285
0027 286
0027 287
0027 288
0027 289
0027 290
0027 291
0027 292
0027 293
0027 294
0027 295
0027 296
0027 297
0027 298
0027 299
0027 300
0027 301
0027 302
0027 303
0027 304
0027 305
0027 306
0027 307
0027 308
0027 309
0027 310
0027 311
0027 312
0027 313
0027 314
0027 315
0027 316
0027 317
0027 318
0027 319
0027 320
0027 321
0027 322
0027 323
0027 324
0027 325
0027 326
0027 327
0027 328
0027 329
0027 330
0027 331
0027 332
0027 333
0027 334
0027 335
0027 336
0027 337
0027 338
0027 339
0027 340
0027 341
0027 342
0027 343
0027 344
0027 345
0027 346
0027 347
0027 348
0027 349
0027 350
0027 351
0027 352
0027 353
0027 354
0027 355
0027 356
0027 357
0027 358
0027 359
0027 360
0027 361
0027 362
0027 363
0027 364
0027 365
0027 366
0027 367
0027 368
0027 369
0027 370
0027 371
0027 372
0027 373
0027 374
0027 375
0027 376
0027 377
0027 378
0027 379
0027 380
0027 381
0027 382
0027 383
0027 384
0027 385
0027 386
0027 387
0027 388
0027 389
0027 390
0027 391
0027 392
0027 393
0027 394
0027 395
0027 396
0027 397
0027 398
0027 399
0027 400
0027 401
0027 402
0027 403
0027 404
0027 405
0027 406
0027 407
0027 408
0027 409
0027 410
0027 411
0027 412
0027 413
0027 414
0027 415
0027 416
0027 417
0027 418
0027 419
0027 420
0027 421
0027 422
0027 423
0027 424
0027 425
0027 426
0027 427
0027 428
0027 429
0027 430
0027 431
0027 432
0027 433
0027 434
0027 435
0027 436
0027 437
0027 438
0027 439
0027 440
0027 441
0027 442
0027 443
0027 444
0027 445
0027 446
0027 447
0027 448
0027 449
0027 450
0027 451
0027 452
0027 453
0027 454
0027 455
0027 456
0027 457
0027 458
0027 459
0027 460
0027 461
0027 462
0027 463
0027 464
0027 465
0027 466
0027 467
0027 468
0027 469
0027 470
0027 471
0027 472
0027 473
0027 474
0027 475
0027 476
0027 477
0027 478
0027 479
0027 480
0027 481
0027 482
0027 483
0027 484
0027 485
0027 486
0027 487
0027 488
0027 489
0027 490
0027 491
0027 492
0027 493
0027 494
0027 495
0027 496
0027 497
0027 498
0027 499
0027 500
0027 501
0027 502
0027 503
0027 504
0027 505
0027 506
0027 507
0027 508
0027 509
0027 510
0027 511
0027 512
0027 513
0027 514
0027 515
0027 516
0027 517
0027 518
0027 519
0027 520
0027 521
0027 522
0027 523
0027 524
0027 525
0027 526
0027 527
0027 528
0027 529
0027 530
0027 531
0027 532
0027 533
0027 534
0027 535
0027 536
0027 537
0027 538
0027 539
0027 540
0027 541
0027 542
0027 543
0027 544
0027 545
0027 546
0027 547
0027 548
0027 549
0027 550
0027 551
0027 552
0027 553
0027 554
0027 555
0027 556
0027 557
0027 558
0027 559
0027 560
0027 561
0027 562
0027 563
0027 564
0027 565
0027 566
0027 567
0027 568
0027 569
0027 570
0027 571
0027 572
0027 573
0027 574
0027 575
0027 576
0027 577
0027 578
0027 579
0027 580
0027 581
0027 582
0027 583
0027 584
0027 585
0027 586
0027 587
0027 588
0027 589
0027 590
0027 591
0027 592
0027 593
0027 594
0027 595
0027 596
0027 597
0027 598
0027 599
0027 600
0027 601
0027 602
0027 603
0027 604
0027 605
0027 606
0027 607
0027 608
0027 609
0027 610
0027 611
0027 612
0027 613
0027 614
0027 615
0027 616
0027 617
0027 618
0027 619
0027 620
0027 621
0027 622
0027 623
0027 624
0027 625
0027 626
0027 627
0027 628
0027 629
0027 630
0027 631
0027 632
0027 633
0027 634
0027 635
0027 636
0027 637
0027 638
0027 639
0027 640
0027 641
0027 642
0027 643
0027 644
0027 645
0027 646
0027 647
0027 648
0027 649
0027 650
0027 651
0027 652
0027 653
0027 654
0027 655
0027 656
0027 657
0027 658
0027 659
0027 660
0027 661
0027 662
0027 663
0027 664
0027 665
0027 666
0027 667
0027 668
0027 669
0027 670
0027 671
0027 672
0027 673
0027 674
0027 675
0027 676
0027 677
0027 678
0027 679
0027 680
0027 681
0027 682
0027 683
0027 684
0027 685
0027 686
0027 687
0027 688
0027 689
0027 690
0027 691
0027 692
0027 693
0027 694
0027 695
0027 696
0027 697
0027 698
0027 699
0027 700
0027 701
0027 702
0027 703
0027 704
0027 705
0027 706
0027 707
0027 708
0027 709
0027 710
0027 711
0027 712
0027 713
0027 714
0027 715
0027 716
0027 717
0027 718
0027 719
0027 720
0027 721
0027 722
0027 723
0027 724
0027 725
0027 726
0027 727
0027 728
0027 729
0027 730
0027 731
0027 732
0027 733
0027 734
0027 735
0027 736
0027 737
0027 738
0027 739
0027 740
0027 741
0027 742
0027 743
0027 744
0027 745
0027 746
0027 747
0027 748
0027 749
0027 750
0027 751
0027 752
0027 753
0027 754
0027 755
0027 756
0027 757
0027 758
0027 759
0027 760
0027 761
0027 762
0027 763
0027 764
0027 765
0027 766
0027 767
0027 768
0027 769
0027 770
0027 771
0027 772
0027 773
0027 774
0027 775
0027 776
0027 777
0027 778
0027 779
0027 780
0027 781
0027 782
0027 783
0027 784
0027 785
0027 786
0027 787
0027 788
0027 789
0027 790
0027 791
0027 792
0027 793
0027 794
0027 795
0027 796
0027 797
0027 798
0027 799
0027 800
0027 801
0027 802
0027 803
0027 804
0027 805
0027 806
0027 807
0027 808
0027 809
0027 810
0027 811
0027 812
0027 813
0027 814
0027 815
0027 816
0027 817
0027 818
0027 819
0027 820
0027 821
0027 822
0027 823
0027 824
0027 825
0027 826
0027 827
0027 828
0027 829
0027 830
0027 831
0027 832
0027 833
0027 834
0027 835
0027 836
0027 837
0027 838
0027 839
0027 840
0027 841
0027 842
0027 843
0027 844
0027 845
0027 846
0027 847
0027 848
0027 849
0027 850
0027 851
0027 852
0027 853
0027 854
0027 855
0027 856
0027 857
0027 858
0027 859
0027 860
0027 861
0027 862
0027 863
0027 864
0027 865
0027 866
0027 867
0027 868
0027 869
0027 870
0027 871
0027 872
0027 873
0027 874
0027 875
0027 876
0027 877
0027 878
0027 879
0027 880
0027 881
0027 882
0027 883
0027 884
0027 885
0027 886
0027 887
0027 888
0027 889
0027 890
0027 891
0027 892
0027 893
0027 894
0027 895
0027 896
0027 897
0027 898
0027 899
0027 900
0027 901
0027 902
0027 903
0027 904
0027 905
0027 906
0027 907
0027 908
0027 909
0027 910
0027 911
0027 912
0027 913
0027 914
0027 915
0027 916
0027 917
0027 918
0027 919
0027 920
0027 921
0027 922
0027 923
0027 924
0027 925
0027 926
0027 927
0027 928
0027 929
0027 930
0027 931
0027 932
0027 933
0027 934
0027 935
0027 936
0027 937
0027 938
0027 939
0027 940
0027 941
0027 942
0027 943
0027 944
0027 945
0027 946
0027 947
0027 948
0027 949
0027 950
0027 951
0027 952
0027 953
0027 954
0027 955
0027 956
0027 957
0027 958
0027 959
0027 960
0027 961
0027 962
0027 963
0027 964
0027 965
0027 966
0027 967
0027 968
0027 969
0027 970
0027 971
0027 972
0027 973
0027 974
0027 975
0027 976
0027 977
0027 978
0027 979
0027 980
0027 981
0027 982
0027 983
0027 984
0027 985
0027 986
0027 987
0027 988
0027 989
0027 990
0027 991
0027 992
0027 993
0027 994
0027 995
0027 996
0027 997
0027 998
0027 999
0027 1000
0027 1001
0027 1002
0027 1003
0027 1004
0027 1005
0027 1006
0027 1007
0027 1008
0027 1009
0027 1010
0027 1011
0027 1012
0027 1013
0027 1014
0027 1015
0027 1016
0027 1017
0027 1018
0027 1019
0027 1020
0027 1021
0027 1022
0027 1023
0027 1024
0027 1025
0027 1026
0027 1027
0027 1028
0027 1029
0027 1030
0027 1031
0027 1032
0027 1033
0027 1034
0027 1035
0027 1036
0027 1037
0027 1038
0027 1039
0027 1040
0027 1041
0027 1042
0027 1043
0027 1044
0027 1045
0027 1046
0027 1047
0027 1048
0027 1049
0027 1050
0027 1051
0027 1052
0027 1053
0027 1054
0027 1055
0027 1056
0027 1057
0027 1058
0027 1059
0027 1060
0027 1061
0027 1062
0027 1063
0027 1064
0027 1065
0027 1066
0027 1067
0027 1068
0027 1069
0027 1070
0027 1071
0027 1072
0027 1073
0027 1074
0027 1075
0027 1076
0027 1077
0027 1078
0027 1079
0027 1080
0027 1081
0027 1082
0027 1083
0027 1084
0027 1085
0027 1086
0027 1087
0027 1088
0027 1089
0027 1090
0027 1091
0027 1092
0027 1093
0027 1094
0027 1095
0027 1096
0027 1097
0027 1098
0027 1099
0027 1100
0027 1101
0027 1102
0027 1103
0027 1104
0027 1105
0027 1106
0027 1107
0027 1108
0027 1109
0027 1110
0027 1111
0027 1112
0027 1113
0027 1114
0027 1115
0027 1116
0027 1117
0027 1118
0027 1119
0027 1120
0027 1121
0027 1122
0027 1123
0027 1124
0027 1125
0027 1126
0027 1127
0027 1128
0027 1129
0027 1130
0027 1131
0027 1132
0027 1133
0027 1134
0027 1135
0027 1136
0027 1137
0027 1138
0027 1139
0027 1140
0027 1141
0027 1142
0027 1143
0027 1144
0027 1145
0027 1146
0027 1147
0027 1148
0027 1149
0027 1150
0027 1151
0027 1152
0027 1153
0027 1154
0027 1155
0027 1156
0027 1157
0027 1158
0027 1159
0027 1160
0027 1161
0027 1162
0027 1163
0027 1164
0027 1165
0027 1166
0027 1167
0027 1168
0027 1169
0027 1170
0027 1171
0027 1172
0027 1173
0027 1174
0027 1175
0027 1176
0027 1177
0027 1178
0027 1179
0027 1180
0027 1181
0027 1182
0027 1183
0027 1184
0027 1185
0027 1186
0027 1187
0027 1188
0027 1189
0027 1190
0027 1191
0027 1192
0027 1193
0027 1194
0027 1195
0027 1196
0027 1197
0027 1198
0027 1199
0027 1200
0027 1201
0027 1202
0027 1203
0027 1204
0027 1205
0027 1206
0027 1207
0027 1208
0027 1209
0027 1210
0027 1211
0027 1212
0027 1213
0027 1214
0027 1215
0027 1216
0027 1217
0027 1218
0027 1219
0027 1220
0027 1221
0027 1222
0027 1223
0027 1224
0027 1225
0027 1226
0027 1227
0027 1228
0027 1229
0027 1230
0027 1231
0027 1232
0027 1233
0027 1234
0027 1235
0027 1236
0027 1237
0027 1238
0027 1239
0027 1240
0027 1241
0027 1242
0027 1243
0027 1244
0027 1245
0027 1246
0027 1247
0027 1248
0027 1249
0027 1250
0027 1251
0027 1252
0027 1253
0027 1254
0027 1255
0027 1256
0027 1257
0027 1258
0027 1259
0027 1260
0027 1261
0027 1262
0027 1263
0027 1264
0027 1265
0027 1266
0027 1267
0027 1268
0027 1269
0027 1270
0027 1271
0027 1272
0027 1273
0027 1274
0027 1275
0027 1276
0027 1277
0027 1278
0027 1279
0027 1280
0027 1281
0027 1282
0027 1283
0027 1284
0027 1285
0027 1286
0027 1287
0027 1288
0027 1289
0027 1290
0027 1291
0027 1292
0027 1293
0027 1294
0027 1295
0027 1296
0027 1297
0027 1298
0027 1299
0027 1300
0027 1301
0027 1302
0027 1303
0027 1304
0027 1305
0027 1306
0027 1307
0027 1308
0027 1309
0027 1310
0027 1311
0027 1312
0027 1313
0027 1314
0027 1315
0027 1316
0027 1317
0027 1318
0027 1319
0027 1320
0027 1321
0027 1322
0027 1323
0027 1324
0027 1325
0027 1326
0027 1327
0027 1328
0027 1329
0027 1330
0027 1331
0027 1332
0027 1333
0027 1334
0027 1335
0027 1336
0027 1337
0027 1338
0027 1339
0027 1340
0027 1341
0027 1342
0027 1343
0027 1344
0027 1345
0027 1346
0027 1347
0027 1348
0027 1349
0027 1350
0027 1351
0027 1352
0027 1353
0027 1354
0027 1355
0027 1356
0027 1357
0027 1358
0027 1359
0027 1360
0027 1361
0027 1362
0027 1363
0027 1364
0027 1365
0027 1366
0027 1367
0027 1368
0027 1369
0027 1370
0027 1371
0027 1372
0027 1373
0027 1374
0027 1375
0027 1376
0027 1377
0027 1378
0027 1379
0027 1380
0027 1381
0027 1382
0027 1383
0027 1384
0027 1385
0027 1386
0027 1387
0027 1388
0027 1389
0027 1390
0027 1391
0027 1392
0027 1393
0027 1394
0027 1395
0027 1396
0027 1397
0027 1398
0027 1399
0027 1400
0027 1401
0027 1402
0027 1403
0027 1404
0027 1405
0027 1406
0027 1407
0027 1408
0027 1409
0027 1410
0027 1411
0027 1412
0027 1413
0027 1414
0027 1415
0027 1416
0027 1417
0027 1418
0027 1419
0027 1420
0
```

```

5A 34 AA 9C 003D 253      MOVAB  JIBSW_TQCNT(R10),R10 ; POINT TO TQCNT FIELD
      0041 254      :
      0041 255      : R2 = ADR OF LOCAL COMMON EVENT BLOCK (PERHAPS SLAVE)
      0041 256      : (0 IF NOT YET CREATED)
      0041 257      : R4 = ADR OF SHARED MEMORY CONTROL BLOCK (IF IN SHARED MEMORY)
      0041 258      : R5 = ADR OF SHARED MEMORY COMMON DATA PAGE (IF IN SHARED MEMORY)
      0041 259      : R6 = ADR OF SHARED MEMORY MASTER COMMON EVENT BLOCK (IF IN SHARED MEMORY)
      0041 260      : (0 IF NOT YET CREATED)
      0041 261      : R7 = ADR OF PROCESS CONTROL BLOCK
      0041 262      : R8 = INDEX TO APPROPRIATE CLUSTER IN PCB
      0041 263      : R10 = ADDRESS OF JIBSW_TQCNT, QUOTA FOR COMMON EVENT BLOCKS
      0041 264      : 0(SP) = ASCII COMMON EVENT FLAG CLUSTER NAME
      0041 265      : 16(SP) = ASCII SHAR D MEMORY NAME
      0041 266      : 32(SP) = PERMANENT INDICATOR
      0041 267      :
      52  D5 0041 268      TSTL  R2 ; WAS NAME FOUND
      03  13 0043 269      BEQL  10$ ; BR IF NAME NOT FOUND, CREATE CEB
      0140 31 0045 270      BRW   ASSOCIATE ; BR AS NAME FOUND, ASSOCIATE IT
      56  D5 0048 271 10$: TSTL  R6 ; IS THERE A MASTER SHMCEB?
      05  13 004A 272      BEQL  11$ ; BR IF NO MASTER EXISTS
20 AE 0B A6 90 004C 273      MOVVB (CEB$B_STS(R ),32(SP) ; FORCE SLAVES TO BE SAME AS MASTER
      0A 20 AE E9 0051 274 11$: BLBC  32(SPT,15$ ; SKIP PRMCEB PRIV CHK IF CREATING TEMP
05 6C B7 0A E0 0055 275      BBS  #PRV$V_PRM_LB,@PCBSL_PHD(R7),15$ ; BR IF HAVE PRIV TO CREATE
      50  24 9A 005A 276 12$: MOVZBL #SS$_NOPRIV,R0 ; SET ERROR CODE
      38  11 005D 277      BRB   36$ ; AND RETURN FROM SYSTEM SERVICE
      10 AE 95 005F 278 15$: TSTB  16(SP) ; IS THIS A SHARED MEMORY CLUSTER?
      11  12 0062 279      BNEQ  30$ ; BR IF YES, DONT CHARGE QUOTA
      1C 20 AE E8 0064 280      BLBS  32(SP),32$ ; DONT CHARGE QUOTA IF PERM CLUSTER
      6A  B5 0068 281      TSTW  (R10) ; CHECK IF QUOTA REMAINING
      05  1A 006A 282      BGTRU 20$ ; CONTINUE IF QUOTA REMAINING
      50  1C 9A 006C 283      MOVZBL #SS$_EXQUOTA,R0 ; SET STATUS CODE FOR QUOTA EXHAUSTED
      3A  11 006F 284      BRB   37$ ; AND RETURN FROM SYSTEM SERVICE
      6A  B7 0071 285 20$: DECW  (R10) ; ACCOUNT FOR COMMON EVENT BLOCK
      0F  11 0073 286      BRB   32$ ; GO ALLOCATE A LOCAL MEMORY CEB
      56  D5 0075 287 30$: TSTL  R6 ; IS MASTER SHMCEB ALREADY CREATED?
      0B  12 0077 288      BNEQ  32$ ; BR IF MASTER SHMCEB EXISTS
DC 6C B7 0B E1 0079 289      BBC  #PRV$V_SHMEM,@PCBSL_PHD(R7),12$ ; EXIT SERVICE IF NOT PRIVILEGED
      0317 30 007E 290      BSBW  ALOSHMCEB ; GO ALLOCATE A MASTER SHMCEB
      0081 291      :
      27 50 E9 0081 292      BLBC  R0,37$ ; R6 = ADR OF MASTER SHMCEB
      FF79 30 0084 293 32$: BSBW  EXES$ALLOCCEB ; BR IF NO FREE CEB
      0087 294      :
      24 50 E8 0087 295      BLBS  R0,40$ ; ALLOCATE SPACE FOR CEB/SLAVCEB
      56  D5 008A 296      TSTL  R6 ; R2 = ADR OF SLAVE CEB
      09  12 008C 297      BNEQ  36$ ; CONTINUE IF SUCCESSFUL
      19 20 AE E8 008E 298      BLBS  32(SP),37$ ; IS THIS A SHARED MEMORY CLUSTER?
      6A  B6 0092 299 35$: INCW  (R10) ; BR IF YES, DONT RETURN QUOTA
      0166 31 0094 300      BRW   EXIT ; SKIP QUOTA IF CREATING PERM
      56  D5 0097 301 36$: TSTL  R6 ; RETURN QUOTA IF ERROR
      10  13 0099 302      BEQL  37$ ; RETURN ERROR TO SYSTEM SERVICE
      50  DD 009B 303      PUSHL R0 ; IS THERE A MASTER CEB?
      03A5 30 009D 304      BSBW  DEC_SHMCEB_REF ; BR ON NO MASTER CEB
      50  BEO0 00A0 305      POPL  R0 ; REMEMBER FIRST ERROR CODE
      04 66 00 E0 00A3 306      BBS  #CEB$V_VALID,CEB$L_CEBFL(R6),37$ ; RELEASE LOCK PREVENTING DELETION
      00 66 01 E7 00A7 307      BBCCI #CEB$V_LOCKED,CEB$L_CEBFL(R6),37$ ; IGNORE POTENTIAL INTERLOCKING PROBLEM
      014F 31 00AB 308 37$: BRW   EXIT ; BR IF MASTER NOT JUST CREATED
      00AE 309 40$: ; #CEB$V_VALID,CEB$L_CEBFL(R6),37$ ; BR IF MASTER NOT JUST CREATED
      : ; #CEB$V_LOCKED,CEB$L_CEBFL(R6),37$ ; RELEASE MASTER FOR OTHER USE
      : ; EXIT IF NO SPACE FOR CEB

```

```

00AE 310 :
00AE 311 : R2 = ADR OF LOCAL COMMON EVENT BLOCK (PERHAPS SLAVE)
00AE 312 : R4 = ADR OF SHARED MEMORY CONTROL BLOCK (IF IN SHARED MEMORY)
00AE 313 : R5 = ADR OF SHARED MEMORY COMMON DATA PAGE (IF IN SHARED MEMORY)
00AE 314 : R6 = ADR OF SHARED MEMORY MASTER COMMON EVENT BLOCK (IF IN SHARED MEMORY)
00AE 315 : R7 = ADR OF PROCESS CONTROL BLOCK
00AE 316 : R10 = ADDRESS OF JIBSW_TQCNT, QUOTA FOR COMMON EVENT BLOCK
00AE 317 :
00AE 318 :
00AE 319 : FIRST INITIALIZE THE CEB FIELDS USED IN THE LOCAL MEMORY CEB AND NOT
00AE 320 : IN THE SLAVE CEB. THIS ALLOWS A BRANCH BACK TO 41$ IF A RACE OCCURS
00AE 321 : DURING CREATION OF THE MASTER CEB.
00AE 322 :
OB A2 01 24 A2 0C AC B0 00AE 323 : MOVW PROT(AP),CEBSW_PROT(R2) ; SET PROTECTION MASK
00B3 324 : INSV 32(SP),#CEBSV_PERM,#1,CEBSB_STS(R2) ; SET PERMANENT FLAG
00BA 325 :
00BA 326 : NOW INITIALIZE THE CEB FIELDS IN COMMON TO A LOCAL MEMORY CEB AND A
00BA 327 : SLAVE CEB.
00BA 328 :
14 A2 14 A2 DE 00BA 329 41$: MOVAL CEBSL_WQFL(R2),CEBSL_WQFL(R2) ; SET UP WAIT QUEUE HEADER
18 A2 14 A2 DE 00BF 330 : MOVAL CEBSL_WQFL(R2),CEBSL_WQBL(R2) ; FORWARDS AND BACKWARDS LINKS
1C A2 B4 00C4 331 : CLRW CEBSW_WQCNT(R2) ; CLEAR QUEUE COUNT
10 A2 D4 00C7 332 : CLRL CEBSL_EFC(R2) ; EVENT FLAG SET
26 A2 B4 00CA 333 : CLRW CEBSW_REFC(R2) ; INIT REFERENCE COUNT TO 1
20 A2 00BC C7 D0 00CD 334 : MOVL PCBSL_UIC(R7),CEBSL_UIC(R2) ; SET UIC OF CLUSTER OWNER
OC A2 20 AA D0 00D3 335 : MOVL <JIBSC_MPID-JIBSW_TQCNT>(R10),CEBSL_PID(R2) ; AND MASTER PID SO
00D8 336 : THAT QUOTA WILL BE RETURNED CORRECTLY
28 A2 10 AE 10 28 00DA 337 : PUSHR #*M<R2,R3,R4,R5> ; SAVE REGISTERS FOR MOVC3
1E A2 03 B0 00E0 338 : MOVC3 #16,16(SP),CEBST_EFCNAM(R2) ; FILL IN CLUSTER NAME
10 AE 95 00E2 339 : POPR #*M<R2,R3,R4,R5> ; RESTORE REGISTERS
10 AE 95 00E6 340 : MOVW #SCHSC_CEF,CEBSW_STATE(R2) ; SET WAIT STATE NUMBER
03 12 00E9 341 : TSTB 16(SP) ; IS THIS IN SHARED MEMORY?
008D 31 00EB 342 : BNEQ 39$ ; BR IF IN SHARED MEMORY
51 15 A4 9A 00EE 343 : BRW 50$ ; BR IF IN LOCAL MEMORY
10 A2 10 A6 D0 00F2 344 39$: MOVZBL SHBSB_PORT(R4),R1 ; GET PROCESSOR PORT INDEX
38 A2 54 D0 00F6 345 : MOVW #DYN$C_SLAVCEB,CEBSB_TYPE(R2) ; INDICATE ENTRY IS SLAVE CEB
38 A641 52 D0 00FB 346 : MOVL CEBSL_EFC(R6),CEBSL_EFC(R2) ; COPY FLAGS FROM MASTER CEB
40 A2 56 D0 00FF 347 : MOVL R4,CEBSL_SHB(R2) ; SET ADR OF SHARED MEMORY CONTROL BLOCK
0315 30 0104 348 : MOVL R2,CEBSL_VASLAVE1(R6)[R1] ; SET VA OF SLAVE CEB
0108 349 : MOVL R6,CEBSL_MASTER(R2) ; SET ADR OF MASTER IN SLAVE CEB
010B 350 : BSBW INC SHMCEB_REF ; ADD REFCNT FOR SLAVCEB TO MASTER CEB
010B 351 : BLBC R0,XXX ; BR IF INTERLOCK ERROR
51 08 A2 01 E5 010B 352 : BBCC #CEBSV_PERM,CEBSB_STS(R2),115$ ; SLAVES ARE ALWAYS TEMP
51 08 A5 55 C1 0110 353 115$: ADDL3 R5,SHD$C_CEFPTR(R5),R1 ; GET ADR OF SHM CEF TABLE
51 56 51 C3 0115 354 : SUBL3 R1,R6,R1 ; GET OFFSET TO CEB ENTRY
50 08 A6 3C 0119 355 : MOVZWL CEBSW_SIZE(R6),R0 ; GET SIZE OF ONE ENTRY
51 50 C6 011D 356 : DIVL2 R0,R1 ; GET # OF ENTRIES INTO TABLE
3C A2 51 B0 0120 357 : MOVW R1,CEBSW_INDX(R2) ; SET INDEX INTO MASTER CEB TABLE
03 66 00 E1 0124 358 : BBC #CEBSV_VALID,CEBSL_CEBFL(R6),112$ ; BR IF MASTER JUST CREATED
0050 31 0128 359 : BRW 50$ ; BR IF MASTER NOT JUST CREATED
012B 360 :
012B 361 : HERE CHECK IF THE NEWLY CREATED MASTER SHMGSD IS UNIQUE OR NOT.
012B 362 :
034C 30 012B 363 112$: BSBW UNIQUE_CEB ; CHECK THAT ANOTHER CEB WAS NOT CREATED
58 D5 012E 364 : TSTL R11 ; WAS THERE A DUPLICATE?
41 13 0130 365 : BEQL 48$ ; BR ON NO DUPLICATE
53 15 A4 9A 0132 366 : MOVZBL SHBSB_PORT(R4),R3 ; GET PORT # INDEX

```

```

00 0B A6 01 E6 0136 367 BBSSI #CEBSV_PERM,CEBSB_STS(R6),113$ ; INHIBIT DELETION CODE EXECUTION
    38 A643 D4 013B 368 113$: CLRL CEBSL_VASLAVE1(R6)[R3] ; INDICATE NO SLAVE ASSOCIATED
    0303 30 013F 369 BSBW DEC_SHMCEB_REF ; RELEASE ASSOC AND SLAVECEB REFCNTS ON
    0300 30 0142 370 BSBW DEC_SHMCEB_REF ; MASTER CEB JUST CREATED
    0145 371 ; IGNORE INTERLOCK ERRORS FOR BOTH
00 66 01 E7 0145 372 BBCCI #CEBSV_LOCKED,CEBSL_CEBFL(R6),42$ ; RELEASE CEB FOR OTHER USE
    38 AB43 D5 0149 373 42$: TSTL CEBSL_VASLAVE1(R11)[R3] ; IS THERE ALREADY A SLAVE CEB?
    14 13 014D 374 BEQL 43$ ; BR TO USE SLAVE CEB JUST CREATED
    50 52 D0 014F 375 MOVL R2,R0 ; SET ADR OF SLAVE CEB JUST CREATED
00000000'EF 16 0152 376 JSB EXESDEANONPAGED ; RELEASE SLAVE CEB
    53 15 A4 9A 0158 377 MOVZBL SHBSB_PORT(R4),R3 ; GET PORT # INDEX (DESTROYED BY JSB)
    52 38 AB43 D0 015C 378 MOVL CEBSL_VASLAVE1(R11)[R3],R2 ; GET ADR OF OLD SLAVE CEB
    25 11 0161 379 BRB ASSOCIATE ; GO ASSOCIATE WITH OLD SLAVE CEB
    56 5B D0 0163 380 43$: MOVL R11,R6 ; SET ADR OF DUPLICATE MASTER CEB
0B A2 0B A6 90 0166 381 MOVB CEBSB_STS(R6),CEBSB_STS(R2) ; SET SLAVE STATUS SAME AS MASTER
24 A2 24 A6 B0 016B 382 MOVW CEBSW_PROT(R6),CEBSW_PROT(R2) ; COPY PROTECTION FROM DUPLICATE
    FF47 31 0170 383 BRW 41$ ; GO INITIALIZE SLAVE CEB AGAIN
    00 66 00 E6 0173 384 48$: BBSSI #CEBSV_VALID,CEBSL_CEBFL(R6),49$ ; INDICATE ENTRY IS VALID
    00 66 01 E7 0177 385 49$: BBCCI #CEBSV_LOCKED,CEBSL_CEBFL(R6),50$ ; CLEAR THE LOCK BIT
    017B 386 ;
    017B 387 ; ADD NEW SLAVE CEB TO LOCAL MEMORY DATA STRUCTURES.
    017B 388 ;
00000000'EF B6 017B 389 50$: INCW SCHSGW_CEBCNT ; INCREMENT COUNT OF COMMON EVENT BLOCKS
00000000'EF 62 OE 0181 390 INSQUE (R2),SCHSGQ_CEBHD ; INSERT CEB IN LIST
    0188 391
    0188 392 .DSABL LSB
    0188 393
    0188 394 ASSOCIATE: ; R6 MAY NOT BE MASTER CEB FOR R2 HERE
    26 A2 B6 0188 395 INCW CEBSW_REFC(R2) ; REFERENCE CLUSTER
    58 A748 DD 018B 396 PUSHL PCB$EFC2P(R7)[R8] ; GET POINTER TO CURRENT CLUSTER
    58 A748 52 D0 018F 397 MOVL R2,PCB$EFC2P(R7)[R8] ; SET CLUSTER POINTER IN PCB
    0194 398 DISASOC:
    0194 399 .ENABL LSB
    50 8E D0 0194 400 MOVL (SP)+,R0 ; GET OLD CLUSTER ADDRESS
    61 13 0197 401 BEQL EXITN ; NONE
    0A A0 2D 91 0199 402 CMPB #DYN$C_SLAVCEB,CEBSB_TYPE(R0) ; IS THIS A SLAVE CEB?
    14 12 019D 403 BNEQ 5$ ; BR IF NORMAL LOCAL MEMORY CEB
    54 38 A0 D0 019F 404 MOVL CEBSL_SHB(R0),R4 ; GET ADR OF SHARED MEMORY CONTROL BLOCK
    55 04 A4 D0 01A3 405 MOVL SHBSL_DATAPAGE(R4),R5 ; GET ADR OF SHARED MEMORY DATA PAGE
    56 40 A0 D0 01A7 406 MOVL CEBSL_MASTER(R0),R6 ; GET ADR OF MASTER CEB
    50 DD 01AB 407 PUSHL R0 ; SAVE ADR OF SLAVE CEB
    0295 30 01AD 408 BSBW DEC_SHMCEB_REF ; DECREMENT ASSOC REFCNT IN MASTER CEB
    50 B E D O 01B0 409 POPL R0 ; IGNORE POTENTIAL INTERLOCK ERROR
    26 A0 B7 01B3 410 5$: DECW CEBSW_REFC(R0) ; DECREMENT REFERENCE COUNT
    42 14 01B6 411 BGTR EXITN ; STILL BUSY, CANT RELEASE YET
    0A A0 2D 91 01B8 412 CMPB #DYN$C_SLAVCEB,CEBSB_TYPE(R0) ; IS THIS A SLAVE CEB?
    14 13 01BC 413 BEQL 10$ ; BR IF YES, DONT RETURN QUOTA
    37 0B A0 01 E0 01BE 414 BBS #CEBSV_PERM,CEBSB_STS(R0),EXITN ; PERMANENT CLUSTER
    0A 0B A0 00 E0 01C3 415 BBS #CEBSV_NOQUOTA,CEBSB_STS(R0),10$ ; CHECK FOR QUOTA
    52 0C A0 3C 01C8 416 MOVZWL CEBSL_PID(R0),R2 ; GET PIX OF CREATOR
    00000000'EF 16 01CC 417 JSB RETQUOTA ; RETURN QUOTA
    00000000'EF B7 01D2 418 10$: DECW SCHSGW_CEBCNT ; DECREASE COUNT OF COMMON EVENT BLOCKS
    51 60 OF 01D8 419 REMQUE (R0),RT ; REMOVE FROM LIST
    0A A0 2D 91 01DB 420 CMPB #DYN$C_SLAVCEB,CEBSB_TYPE(R0) ; IS THIS A SLAVE CEB?
    13 12 01DF 421 BNEQ 20$ ; BR IF NORMAL LOCAL MEMORY CEB
    00000000'EF 16 01E1 422 JSB EXESDEANONPAGED ; RELEASE SLAVE CEB
    51 15 A4 9A 01E7 423 MOVZBL SHBSB_PORT(R4),R1 ; PORT # FOR THIS PROCESSOR

```

38	A641	D4	01EB	424	CLRL	CEB\$L VASLAVE1(R6)[R1]	:	INDICATE NO SLAVE CEB FOR PROCESSOR
	0253	30	01E1	425	BSBW	DEC_SRMCEB_REF	:	RELEASE SLAVCEB REFCNT IN MASTER CEB
	09	11	01F2	426	BRB	EXIT	:	BRANCH TO COMMON CODE
			01F4	427				
00000000	'EF	16	01F4	428	20\$: JSB	EXES\$DEANONPAGED	:	DEALLOCATE SPACE FOR LOCAL CEB
50	01	9A	01FA	429	EXITN: MOVZBL	#SS\$_NORMAL,R0	:	SET NORMAL COMPLETION STATUS
	50	DD	01FD	430	EXIT: PUSHL	R0	:	SAVE COMPLETION STATUS
50	00000000	DE	01FF	431	EXITE: MOVAL	EXES\$GL_CEBMTX,R0	:	GET ADDRESS OF CEB MUTEX
54	57	DD	0206	432	MOVL	R7,R4	:	RESTORE ADDRESS OF PCB
00000000	'EF	16	0209	433	JSB	SCH\$UNLOCK	:	AND FREE IT
	01	BA	020F	434	POPR	#*M<R0>	:	RESTORE COMPLETION CODE
			0211	435	SETIPL	#0	:	ENABLE
		04	0214	436	RET		:	AND RETURN TO CALLER
			0215	437				
			0215	438	.DSABL	LSB		

```

00000004 0215 440 .SBTIL EXE$DLCEFC - DELETE COMMON EVENT CLUSTER
          0215 441 NAME=4 ; NAME DISPLACEMENT IN ARGLIST
          0215 442
          0215 443
          0215 444 :++
          0215 445 : FUNCTIONAL DESCRIPTION:
          0215 446 : EXE$DLCEFC IMPLEMENTS THE DELETE COMMON EVENT CLUSTER SYSTEM
          0215 447 : SERVICE WHICH PERMITS A PERMANENT COMMON EVENT CLUSTER TO BE
          0215 448 : MARKED FOR DELETION WHEN ITS REFERENCE COUNT BECOMES ZERO.
          0215 449
          0215 450 : CALLING SEQUENCE:
          0215 451 : CALLG ARGLIST,EXE$DLCEFC
          0215 452
          0215 453 : INPUT PARAMETERS:
          0215 454 : NAME(AP) - ADDRESS OF STRING DESCRIPTOR FOR NAME OF CLUSTER
          0215 455
          0215 456 : OUTPUT PARAMETERS:
          0215 457 : R0 - COMPLETION STATUS
          0215 458
          0215 459 : COMPLETION CODES:
          0215 460 : $$$_NORMAL - SUCCESSFUL COMPLETION OF SERVICE
          0215 461 : $$$_NOPRIV - INSUFFICIENT PRIVILEGE TO DELETE CLUSTER
          0215 462
          0215 463 : SIDE EFFECTS:
          0215 464 : IF THE REFERENCE COUNT FOR THE SPECIFIED CLUSTER IS ZERO,
          0215 465 : THE CLEARING OF THE PERMANENT BIT WILL CAUSE IT TO BE DELETED.
          0215 466
          0215 467 :--
          0215 468
          0215 469 EXE$DLCEFC::
          0215 470 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; REGISTER SAVE MASK
          0217 471 MOVL NAME(AP),R9 ; GET POINTER TO CLUSTER NAME
          0218 472 SUBL #32,SP ; CREATE SPACE FOR NAME STRINGS
          021E 473 BSBB LOCLUST ; LOCATE CLUSTER BY NAME
          0220 474 TSTL R2 ; WAS NAME FOUND
          0222 475 BEQL 40$ ; NO, GO CHECK IF THERE'S A MASTER CEB
          0224 476 CMPL PCBSL_UIC(R7),CEBSL_UIC(R2) ; OWNER OF CLUSTER?
          022A 477 BEQL 10$ ; YES, PERMIT ANYWAY
          022C 478 BBS #PRVSV_PRCMB,@PCBSL_PHD(R7),10$ ; CONTINUE IF PRIVILEGED
          0231 479 5$: MOVZBL #$$$_NOPRIV,R0 ; SET ERROR CODE
          0234 480 BRB EXIT ; EXIT RELEASING MUTEX
          0236 481 10$: BBCC #CEBSV_PERM,CEBSB_STS(R2),20$ ; CLEAR PERMANENT FLAG
          0238 482 BBSS #CEBSV_NOQUOTA,CEBSB_STS(R2),20$ ; SET NO QUOTA
          0240 483 20$: PUSHL R2 ; SAVE CLUSTER ADDRESS
          0242 484 INCW CEBSW_REFC(R2) ; TEMPORARILY REFERENCE IT
          0245 485 TSTL R6 ; IS THERE A MASTER CEB, ALSO?
          0247 486 BEQL 30$ ; BR IF NO MASTER CEB TO DELETE
          0249 487 BBC #PRVSV_SHMEM,@PCBSL_PHD(R7),5$ ; BR IF NOT PRIV TO DELETE
          024E 488 BBCCI #CEBSV_PERM,CEBSB_STS(R6),25$ ; CHANGE STATUS TO TEMP FOR DELET
          0253 489 25$: MOVSB SHBSB_PORT(R4),CEBSB_DELETEPORT(R6) ; INDICATE PORT # OF DELETOR
          0258 490 30$: BRW DISASOC ; DISASSOCIATE
          025B 491 ; RELEASING CEB IF IDLE
          025B 492 40$: TSTL R6 ; IS THERE A MASTER CEB TO DELETE?
          025D 493 BEQL EXITN ; BR IF NONE, EXIT WITH NORMAL STATUS
          025F 494 CMPL PCBSL_UIC(R7),CEBSL_UIC(R6) ; OWNER OF CLUSTER?
          0265 495 BEQL 50$ ; YES, PERMIT DELETION
          0267 496 BBC #PRVSV_PRCMB,@PCBSL_PHD(R7),5$ ; BR IF NOT PRIVILEGED

```

CO 6C B7 1B	E1 026C	497 50\$:	BBC	#PRVSV_SHEM,@PCBSL PHD(R7),5\$ ; BR IF NOT PRIV TO DELETE
00 0B A6 01	E7 0271	498 60\$:	BBCCI	#CEBSV_PERM,(EBSB_STS(R6),70\$ ; CHANGE STATUS TO TEMP FOR DELETE
01CC	30 0276	499 70\$:	BSBW	DEC_SHEB_REF ; RELEASE SEARCH CNT; TRY TO DEL MASTER
FFB1	31 0279	500	BRW	EXIT ; EXIT AT COMPLETION CODE REPORTING
	027C	501		
	027C	502		
	027C	503		

```
027C 505 .SBTTL CEBLOCK - LOCK COMMON EVEN BLOCK MUTEX FOR WRITE
027C 506
027C 507 :++
027C 508 : FUNCTIONAL DESCRIPTION:
027C 509 : CEBLOCK LOCKS THE COMMON EVENT BLOCK MUTEX FOR WRITE AND
027C 510 : RETURNS WITH IPL AT IPL$ASTDEL.
027C 511 :
027C 512 : CALLING SEQUENCE:
027C 513 : BSB/JSB CEBLOCK
027C 514 :
027C 515 : INPUT PARAMETERS:
027C 516 :
027C 517 : R4 - PCB ADDRESS
027C 518 :
027C 519 :--
027C 520
027C 521 CEBLOCK:
50 00000000'EF DE 027C 522 : LOCK CEB MUTEX FOR WRITE
00000000'EF 17 0283 523 : SET ADDRESS OF MUTEX
0289 524 JMP SCH$LOCKW : LOCK IT FOR WRITE
```



```

0289 526 .SBTTL LOCLUST - LOCATE CLUSTER GIVEN NAME
0289 527
00000008 0289 528 NAME=8 ; NAME DISPLACEMENT IN ARGLIST
0289 529 :++
0289 530 : FUNCTIONAL DESCRIPTION:
0289 531 : LOCLUST SEARCHES THE LIST OF COMMON EVENT BLOCKS FOR THE
0289 532 : NAMED CLUSTER AND RETURNS THE ADDRESS OF THE CLUSTER.
0289 533 : IF NO ERROR CONDITIONS ARE ENCOUNTERED, LOCLUST RETURNS WITH
0289 534 : THE CEB MUTEX LOCKED FOR WRITE. IN THE EVENT OF ERROR, THE
0289 535 : MUTEX IS UNLOCKED AND A RET IS ISSUED TO EXIT THE SERVICE.
0289 536
0289 537 : CALLING SEQUENCE:
0289 538 : BSB/JSB LOCLUST
0289 539
0289 540 : INPUT PARAMETERS:
0289 541 : 04(SP) - 16 BYTE AREA INTO WHICH CCF CLUSTER NAME WILL BE MOVED
0289 542 : 14(SP) - 16 BYTE AREA INTO WHICH SHARED MEMORY NAME WILL BE MOVED
0289 543 : R4 - PCB ADDRESS
0289 544 : R9 - ADDRESS OF NAME STRING DESCRIPTOR
0289 545
0289 546 : OUTPUT PARAMETERS:
0289 547 : R0 - COMPLETION STATUS
0289 548 : ALL CASES:
0289 549 : R1,R3 - DESTROYED
0289 550 : R7 - PCB ADDRESS OF CURRENT PROCESS
0289 551 : IF LOCAL MEMORY COMMON EVENT CLUSTER NAME:
0289 552 : R2 - ADDRESS OF LOCAL MEMORY CEB (0 IF NAME NOT FOUND)
0289 553 : R4 - R5 DESTROYED
0289 554 : R6 - 0
0289 555 : IF SHARED MEMORY COMMON EVENT CLUSTER NAME:
0289 556 : R2 - ADDRESS OF SLAVE CEB (0 IF NO SLAVE FOUND)
0289 557 : R4 - SHARED MEMORY CONTROL BLOCK ADDRESS
0289 558 : R5 - SHARED MEMORY COMMON DATA PAGE ADDRESS
0289 559 : R6 - ADDRESS OF SHARED MEMORY MASTER CEB (0 IF NAME NOT FOUND)
0289 560
0289 561 : COMPLETION CODES:
0289 562 :
0289 563 : SSS_ACCVIO - LOGICAL NAME INACCESSIBLE
0289 564 : SSS_IVLOGNAM - COUNT FOR LOGICAL NAME NOT 1-15
0289 565 : SSS_NOPRIV - PRIVILEGE VIOLATION
0289 566 : SSS_SHMNOTCNCT - SHARED MEMORY NOT CONNECTED
0289 567 : SSS_TOOMANYLNAM - TOO MANY LOGICAL NAME TRANSLATIONS
0289 568
0289 569 :--
0289 570
0289 571 LOCLUST: ; LOCATE CLUSTER
0289 572 .ENABL LSB
0289 573
0289 574 PUSHR #^M<R9,R10,R11> ; SAVE REGISTERS
OE00 8F BB 0289 575 MOVL R4,R7 ; SAVE PCB ADDRESS
57 54 D0 0289 576 IFNORD #8,(R9),30$ ; TEST DESCRIPTOR READABILITY
0290 577
0296 578 : CREATE STRING DESCRIPTORS FOR COMMON EVENT CLUSTER AND SHARED MEMORY NAME
0296 579 : BUFFERS. THEN PERFORM LOGICAL TRANSLATION OF USER-PROVIDED NAME TO
0296 580 : DETERMINE ACTUAL CLUSTER NAME AND SHARED MEMORY NAME.
0296 581
51 69 9A 0296 582 MOVZBL (R9),R1 ; GET COUNT FROM DESCRIPTOR (LE 255)

```

```

7A 13 0299 583 BEQL 50$ : ILLEGAL STRING LENGTH
      029B 584 IFNORD R1,24(R9),30$ : CHECK READABILITY OF STRING
11 AE 9F 02A2 585 PUSHAB <<4*4>+1>(SP) : SET UP STRING DESCRIPTOR FOR
      OF DD 02A5 586 PUSHL #15 : CEF CLUSTER NAME
5B SE DO 02A7 587 MOVL SP,R11 : SET ADR OF CEF NAME STR DSC
29 AE 9F 02AA 588 PUSHAB <<4*10>+1>(SP) : SET UP STRING DESCRIPTOR FOR
      OF DD 02AD 589 PUSHL #15 : SHARED MEMORY NAME
5A SE DO 02AF 590 MOVL SP,R10 : SET ADR OF SHM NAME STR DSC
FD4B' 30 02B2 591 BSBW MMG$CEFTRNLOG : DETERMINE CEF AND SHM NAMES
62 50 E9 02B5 592 BLBC R0,60$ : BR IF ERROR IN LOGICAL NAME TRANS
18 AB 6B 90 02B8 593 MOVB (R11),<4*6>(R11) : MAKE ASCII CEF NAME
      57 13 02BC 594 BEQL 50$ : BR IF NO CEF NAME SPECIFIED
      04 AB D7 02BE 595 DECL 4(R11) : CONVERT TO POINT TO ASCII NAME
30 AA 6A 90 02C1 596 MOVB (R10),<4*12>(R10) : MAKE ASCII SHM NAME
      63 12 02C5 597 BNEQ FIND_SHM_CEB : BR IF CEF CLUSTER IS IN SHARED MEMORY
      B3 10 02C7 598 BSBB CEBLOCK : LOCK CEB LIST TO PROHIBIT WRITERS
      02C9 599 :
      02C9 600 : FIND COMMON EVENT BLOCK IN LOCAL MEMORY.
      02C9 601 :
5A 00000000'EF DE 02C9 602 MOVAL SCH$GQ_CEBHD,R10 : GET ADDRESS OF CEB LIST
      56 5A DO 02D0 603 MOVL R10,R6 : AND POINTER TO HEAD OF LIST
      56 66 DO 02D3 604 10$: MOVL (R6),R6 : FLINK ONWARD
      56 5A D1 02D6 605 CMPL R10,R6 : CHECK FOR END OF LIST
      40 13 02D9 606 BEQL 15$ : BR IF AT END OF LIST, ENTRY NOT FOUND
00BE C7 22 A6 B1 02DB 607 40$: CMPW CEB$W_GRP(R6),PCB$W_GRP(R7) : SAME GROUP?
      FO 12 02E1 608 BNEQ 10$ : NO, TRY ANOTHER
28 A6 04 BB 10 29 02E3 609 CMPC3 #16,24(R11),CEB$T_EFCNAM(R6) : COMPARE THE NAMES
      EB 12 02E9 610 BNEQ 10$ : NOT EQUAL, TRY ANOTHER
      0A A6 2D 91 02EB 611 CMPB #DYN$C_SLAVCEB,CEB$B_TYPE(R6) : SLAVE CEB?
      E2 13 02EF 612 BEQL 10$ : BR IF SLAVE, DON'T USE CEB
00BC C7 20 A6 D1 02F1 613 CMPL CEB$S_UIC(R6),PCB$S_UIC(R7) : OWNER UIC?
      24 13 02F7 614 BEQL 20$ : YES CONTINUE
      20 24 A6 E9 02F9 615 BLBC CEB$W_PROT(R6),20$ : CONTINUE IF NO ERROR
      02FD 616 :
      02FD 617 : NO ACCESS ALLOWED TO COMMON EVENT BLOCK.
      02FD 618 :
50 54 57 DO 02FD 619 45$: MOVL R7,R4 : RESET ADR OF PCB
00000000'EF DE 0300 620 MOVAL EXE$GL_CEBMTX,R0 : SET ADDRESS OF MUTEX
00000000'EF 16 0307 621 JSB SCH$UNLOCK : UNLOCK MUTEX
      50 24 9A 030D 622 MOVZBL #SS$NOPRIV,R0 : SET ERROR CODE
      04 04 0310 623 RET : AND RETURN FROM SERVICE
      50 0C 3C 0311 624 30$: MOVZWL #SS$_ACCVIO,R0 : INDICATE ACCESS VIOLATION
      04 0314 625 RET : AND EXIT SERVICE
50 0154 BF 3C 0315 626 50$: MOVZWL #SS$_IVLOGNAM,R0 : INDICATE ILLEGAL STRING
      04 031A 627 60$: RET : EXIT FROM SYSTEM SERVICE
      031B 628 :
      031B 629 : RETURN ADDRESS OF LOCAL COMMON EVENT BLOCK (IF FOUND) ELSE 0.
      031B 630 :
      52 56 D4 031B 631 15$: CLRL R6 : INDICATE NONE FOUND
      56 DO 031D 632 20$: MOVL R6,R2 : RETURN ADR OF CEB
      56 D4 0320 633 CLRL R6 : RETURN NO MASTER SHMGSD
5E 10 CO 0322 634 25$: ADDL #<4*4>,SP : CLEAN STR DSC'S OFF STACK
OE00 BF BA 0325 635 POPR #*M<R9,R10,R11> : RESTORE REGISTERS
      05 0329 636 RSB : AND RETURN
      032A 637 :
      032A 638 :
      032A 639 : FIND COMMON EVENT BLOCK IN SHARED MEMORY.

```



```

0398 682      .SBTTL ALOSHMCB - ALLOCATE SHARED MEMORY MASTER COMMON EVENT BLOCK
0398 683
0398 684 :++
0398 685 : FUNCTIONAL DESCRIPTION:
0398 686 : THIS ROUTINE ALLOCATES AND INITIALIZES A SHARED MEMORY MASTER COMMON
0398 687 : EVENT BLOCK. SOME FIELDS HAVE ALREADY BEEN INITIALIZED BY SYSGEN.
0398 688
0398 689 : CALLING SEQUENCE:
0398 690 :   BSBW   ALOSHMCB
0398 691
0398 692 : INPUT PARAMETERS:
0398 693 :   R4 - ADDRESS OF SHARED MEMORY CONTROL BLOCK
0398 694 :   R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
0398 695
0398 696 : IMPLICIT INPUTS:
0398 697 :   COMMON DATA PAGE MUST BE INITIALIZED.
0398 698
0398 699 : OUTPUT PARAMETERS:
0398 700 :   R0 - COMPLETION STATUS CODE
0398 701 :   R6 - ADDRESS OF SHARED MEMORY MASTER CEB, IF SUCCESSFUL
0398 702 :   - 0, IF ERROR
0398 703
0398 704 : IMPLICIT OUTPUTS:
0398 705 :   THE MASTER CEB IS INITIALIZED. ITS REF COUNT IS UP BY ONE TO LOCK IT.
0398 706
0398 707 : COMPLETION CODES:
0398 708 :   SSS_EXPORTQUOTA - NO PORT QUOTA
0398 709 :   SSS_NORMAL      - SUCCESSFUL COMPLETION
0398 710 :   SSS_NOSHMBLOCK - NO SHARED MEMORY MASTER CEB AVAILABLE
0398 711
0398 712 : SIDE EFFECTS:
0398 713 :   THE MASTER CEB IS LOCKED PREVENTING OTHER PROCESSES FROM USING IT.
0398 714 :--
0398 715
0398 716
0398 717      .ENABL  LSB
0398 718 ALOSHMCB:
0398 719      PUSH  #*M<R1,R2>          : SAVE REGISTERS
0398 720      MOVZBL SHB$B PORT(R4),R1  : GET PORT # TO BE USED AS AN INDEX
0398 721      ADAWI #-1,SHD$W_CEFQUOTA(R5)[R1] : SUBTRACT PORT QUOTA FOR CEF
0398 722      BLSS  NO QUOTA          : BR IF NO QUOTA AVAILABLE
0398 723      ADDL3 SHD$W_CEFPTR(R5),R5,R6 : GET ADR OF FIRST CEB IN SHM TABLE
0398 724      MOVZWL CEB$W_SIZE(R6),R2   : GET SIZE OF ONE SHMCEB
0398 725      MOVZWL SHD$W_CEFMAX(R5),R1 : GET # OF ENTRIES IN TABLE
0398 726      BRB   20$              : JOIN COMMON CODE
0398 727 10$: BSBW   DEC_SHMCEB_REF  : UNLOCK THE PREVIOUS ENTRY FOR DELETION
0398 728      : IGNORE POTENTIAL INTERLOCK ERROR
0398 729 15$: ADDL2  R2,R6          : GET ADDRESS OF NEXT CEB IN TABLE
0398 730      SOBGTR R1,20$         : BR IF ANOTHER ENTRY TO CHECK
0398 731      BEQL  NO FREE CEB     : BR IF NO ENTRY AVAILABLE
0398 732 20$: BSBW   INC_SHMCEB_REF  : LOCK THE ENTRY
0398 733      BLBC  R0,T5$         : DONT USE ENTRY IF IT IS REFCNT LOCKED
0398 734      BBS  #CEB$V_LOCKED,CEB$C_CEBFL(R6),10$ : BR IF ENTRY BEING MODIFIED
0398 735      BBS  #CEB$V_VALID,CEB$C_CEBFL(R6),10$ : BR IF ENTRY ALREADY IN USE
0398 736      BBSSI #CEB$V_LOCKED,CEB$C_CEBFL(R6),10$ : BR IF SOMEONE JUST TOOK IT
0398 737 :
0398 738 : FOUND AN UNUSED ENTRY. NOW INITIALIZE IT.

```

```

06 BB 0398 719
15 9A 039A 720
A4 58 039E 721
FFFF 19 03A5 722
8F 56 03A7 723
67 C1 03A7 723
08 A5 C1 03A7 723
08 A6 3C 03AC 724
1C A5 3C 0380 725
0B 11 0384 726
008C 30 0386 727
56 52 C0 0389 729
02 51 F5 038C 730
46 3 038F 731
005C 30 03C1 732
F2 50 E9 03C4 733
EB 66 01 E0 03C7 734
E7 66 00 E0 03CB 735
E3 66 01 E6 03CF 736
03D3 737
03D3 738

```

SYS  
Pse

PSE  
---

. B  
\$AB  
YSE  
AEX

Pha  
---

Ini  
Com  
Pas  
Sym  
Pas  
Sym  
Pse  
Cro  
Ass

The  
711  
The  
117  
25

Mac  
---

\$2  
\$2  
TOT

118

The

MAC

```

OB A6 01 01 2C AE 10 A6 D4 03DF 742
          0C A6 60 A7 DO 03E2 743
          20 A6 00BC C7 DO 03E7 744
          24 A6 0C AC 3C 03ED 745
          28 A6 1C AE 10 3C BB 03F2 746
          3C BA 03FA 748
          1E A6 15 A4 9B 03FC 749
          50 01 9A 0401 751
          06 BA 0404 752 30$:
          05 0406 753
          0407 754
          0407 755 NO_FREE_CEB:
          50 03B4 8F 3C 0407 756 MOVZWL #SS$_NOSHMBLOCK,RO ; REPORT FAILURE
          05 11 040C 757 BRB 40$ ; JOIN COMMON CODE
          040E 758
          040E 759 NO_QUOTA:
          50 03AC 8F 3C 040E 760 MOVZWL #SS$_EXPORTQUOTA,RO ; NO CEF PORT QUOTA
          51 15 A4 9A 0413 761 40$: MOVZBL SHB$B_PORT(R4),R1 ; REPORT FAILURE
          7C A541 01 58 0417 762 ADAMI #1,SHB$B_CEFQUOTA(R5)[R1] ; GET PORT # TO BE USED AS AN INDEX
          56 D4 041C 763 CLRL R6 ; RETURN QUOTA ACQUIRED
          E4 11 041E 764 BRB 30$ ; INDICATE NO MASTER CEB, TO USE COMMON
          0420 765 ; ERROR PATHS IN CALLER CODE
          0420 766 ; RETURN TO CLEAN UP
          0420 767 .DSABL LSB

```

```

0420 769 .SBTTL INC_SHMCEB_REF/DEC_SHMCEB_REF - ALTER REFCNT ON SHMEM MASTER CEB
0420 770
0420 771 :++
0420 772 : FUNCTIONAL DESCRIPTION:
0420 773 : THESE ROUTINES INCREMENT/DECREMENT RESPECTIVELY A PROCESSOR REFERENCE
0420 774 : COUNT IN A SHARED MEMORY COMMON EVENT BLOCK BY ONE. THE PORT NUMBER
0420 775 : IS USED AS AN INDEX IN DETERMINING WHICH PROCESSOR REFERENCE COUNT TO
0420 776 : ALTER. THIS IS DONE TO LOCK AN ENTRY SO THAT IT CANNOT BE RELEASED
0420 777 : WHILE A PROCESSOR IS SCANNING IT. A REFERENCE LOCK BIT IN THE MASTER
0420 778 : MUST BE HELD WHILE UPDATING A REFERENCE COUNT AND WHILE CHECKING THAT
0420 779 : AN ENTRY'S REFERENCE COUNTS FOR DELETION. THIS PREVENTS A RACE
0420 780 : CONDITION BETWEEN ASSOCIATION AND DELETION OF COMMON EVENT FLAGS.
0420 781 : NOTE THAT EXECUTION MUST ALREADY BE SINGLE STREAMED ON THE PROCESSOR
0420 782 : EXECUTING THIS CODE VIA MUTEXES.
0420 783 :
0420 784 : THESE ROUTINES ASSUME THAT THE PROCESSOR REFERENCE COUNTS ARE
0420 785 : IMMEDIATELY PRECEDED BY THE SLAVE CEB VIRTUAL ADDRESSES IN THE
0420 786 : MASTER CEB.
0420 787 :
0420 788 : CALLING SEQUENCE:
0420 789 : BSBW INC_SHMCEB_REF/DEC_SHMCEBREF
0420 790 :
0420 791 : INPUT PARAMETERS:
0420 792 : R4 - ADDRESS OF SHARED MEMORY CONTROL BLOCK
0420 793 : R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE (USED BY EXE$SHMCEBDEL)
0420 794 : R6 - ADDRESS OF SHARED MEMORY MASTER COMMON EVENT BLOCK
0420 795 :
0420 796 : IMPLICIT INPUTS:
0420 797 : THE SHARED MEMORY MUST BE CONNECTED AND THE REFERENCE COUNTS IN THE
0420 798 : MASTER CEB SET UP AT SYSGEN TIME.
0420 799 :
0420 800 : OUTPUT PARAMETERS:
0420 801 : R0 - STATUS CODE
0420 802 :
0420 803 : IMPLICIT OUTPUTS:
0420 804 : THE MASTER CEB'S REF COUNT IS UP/DOWN BY ONE TO LOCK/RELEASE IT.
0420 805 :
0420 806 : COMPLETION CODES:
0420 807 : $$$_NORMAL - SUCCESSFUL COMPLETION
0420 808 : $$$_INTERLOCK - UNABLE TO OBTAIN ENTRY REF COUNT LOCK
0420 809 :
0420 810 : SIDE EFFECTS:
0420 811 : THE MASTER CEB CANNOT DISAPPEAR UNTIL THE REFERENCE COUNT IS RELEASED.
0420 812 :
0420 813 :--
0420 814 :
0420 815 INC_SHMCEB_REF:
51 7E 51 7D 0420 816 MOVQ R1,-(SP) ; SAVE REGISTERS
51 15 A4 9A 0423 817 MOVZBL SHB$B_PORT(R4),R1 ; GET PORT # FOR INDEX TO PROC REFCNT
52 52 1D A6 9A 0427 818 MOVZBL CEB$B_PROCCNT(R6),R2 ; GET # OF PROCESSOR SLAVE VA'S IN CEB
52 52 02 78 042B 819 ASHL #2,R2,R2 ; CONVERT TO BYTE CNT, SLAVEVA=LONGWORD
52 52 56 C0 042F 820 ADDL2 R6,R2 ; OFFSET PAST THE SLAVE VA'S
FC01' 30 0432 821 BSBW EXE$CEBREFLCK ; ACQUIRE REF COUNT LOCK IN SHMCEB
09 50 E9 0435 822 BLBC R0,10$ ; BR IF UNABLE TO ACQUIRE LOCK
38 A241 01 58 0438 823 ADAWI #1,CEB$B_VASLAVE1(R2)[R1] ; INCREMENT PROCESSOR REF COUNT
00 66 02 E7 043D 824 BBCCI #CEB$V_REFCNTLCK,CEB$B_CEB$R1(R6),10$ ; RELEASE REF CNT LOCK
51 8E 7D 0441 825 10$: MOVQ (SP)+,R1 ; RESTORE REGISTER

```

			05	0444	826	RSB	
				0445	827		
				0445	828	DEC_SHMCEB_REF:	
	7E	51	7D	0445	829	MOVQ	R1,-(SP) ; SAVE REGISTERS
51	15	A4	9A	0448	830	MOVZBL	SH\$B_PORT(R4),R1 ; GET PORT # FOR INDEX TO PROC REFCNT
52	1D	A6	9A	044C	831	MOVZBL	CE\$B_PROCCNT(R6),R2 ; GET # OF PROCESSOR SLAVE VA'S IN CEB
52	52	02	78	0450	832	ASHL	#2,R2,R2 ; CONVERT TO BYTE CNT, SLAVEVA=LONGWORD
	52	56	C0	0454	833	ADDL2	R6,R2 ; OFFSET PAST THE SLAVE VA'S
		FBDC'	30	0457	834	BSBW	EX\$CEBREFLCK ; ACQUIRE REF COUNT LOCK IN SHMCEB
		OD 50	E9	045A	835	BLBC	RO,10\$ ; BR IF UNABLE TO ACQUIRE LOCK
38	A241	FFFF 8F	58	045D	836	ADAWI	#-1,CE\$B_VASLAVE1(R2)[R1] ; INCREMENT PROCESSOR REF COUNT
			19	0464	837	BLSS	ERR_BUGCHK ; BUGCHECK IF ILLEGAL REFERENCE COUNT
	00	66 02	E7	0466	838	BBCCI	#CE\$V_REFCNTLCK,CE\$B_CEBFL(R6),10\$ ; RELEASE REF CNT LOCK
		51 8E	7D	046A	839	10\$: MOVQ	(SP)+,R1 ; RESTORE REGISTER
01	0B	A6 01	E1	046D	840	BBC	#CE\$V_PERM,CE\$B_STS(R6),20\$ ; BR IF CLUSTER CAN BE DELETED
			05	0472	841	RSB	
		FB8A'	31	0473	842	20\$: BRW	EX\$SHMCEBDEL ; TRY TO RELEASE CEB
				0476	843		
				0476	844	ERR_BUGCHK:	
				0476	845	BUG_CHECK	CEBREFNEG,FATAL ; FATAL ERROR

```

047A 847 .SBTTL EXE$SHMCEBDEL - TRY TO DELETE (RELEASE) MASTER CEB IN SH MEM
047A 848
047A 849 :++
047A 850 : FUNCTIONAL DESCRIPTION:
047A 851 :
047A 852 : THIS ROUTINE PERFORMS TWO FUNCTIONS: (1) IF THE DELETE PENDING FLAG IS
047A 853 : SET, THEN IT INFORMS THE OTHER PROCESSORS ON THE SHARED MEMORY THAT THE
047A 854 : STATUS OF THE CEB IS NOW CHANGED, AND (2) IT CHECKS WHETHER OR NOT THE
047A 855 : SHARED MEMORY CEB CAN BE RELEASED IMMEDIATELY.
047A 856 :
047A 857 : THE SHARED MEMORY CEB CAN ONLY BE RELEASED IF ALL THE REFERENCE COUNTS
047A 858 : ARE ZERO (ONE COUNT PER PROCESSOR). IF A COUNT IS NOT ZERO, THEN ONE OF THE
047A 859 : FOLLOWING IS OUTSTANDING:
047A 860 : 1) A SLAVE CEB STILL EXISTS FOR THE MASTER CEB
047A 861 : 2) AN INTER-PROCESSOR REQUEST MESSAGE EXISTS FOR THE MASTER CEB
047A 862 : 3) A PROCESSOR IS REFERENCING THE MASTER DURING A SHM CEB TBL SEARCH
047A 863 :
047A 864 : CALLING SEQUENCE:
047A 865 : JSB EXE$SHMCEBDEL
047A 866 :
047A 867 : INPUT PARAMETERS:
047A 868 :
047A 869 : R5 - SHARED MEMORY COMMON DATA PAGE ADDRESS
047A 870 : R6 - MASTER CEB ADDRESS
047A 871 :
047A 872 : IMPLICIT INPUTS:
047A 873 :
047A 874 : NONE
047A 875 :
047A 876 : OUTPUT PARAMETERS:
047A 877 :
047A 878 : R0 - STATUS CODE
047A 879 :
047A 880 : IMPLICIT OUTPUTS:
047A 881 :
047A 882 : NONE
047A 883 :
047A 884 : COMPLETION CODES:
047A 885 :
047A 886 : VARIOUS SYSTEM STATUS CODES.
047A 887 :
047A 888 : SIDE EFFECTS:
047A 889 :
047A 890 : THE MASTER CEB MAY BE DELETED. THE OTHER PROCESSORS MAY BE INFORMED
047A 891 : OF A CHANGE IN STATUS OF THE CEB.
047A 892 :
047A 893 :--
047A 894

```

```

00000000 895 .PSECT AEXENONPAGED,BYTE
0000 896 EXE$SHMCEBDEL::
51 7E 51 7D 0000 897 MOVQ R1,-(SP) ; SAVE REGISTER
52 38 A61 9A 0003 898 40$: MOVZBL CEB$B_PROCCNT(R6),R1 ; GET # OF REFERENCE COUNTS TO CHECK
0027 30 000C 899 MOVAL CEB$L_VASLAVE1(R6)[R1],R2 ; GET ADR OF 1ST REF COUNT
20 50 E9 000F 900 BSBW EXE$CEBREFLCK ; ACQUIRE REF COUNT LOCK
82 B5 0012 901 BLBC R0,90$ ; IGNORE POTENTIAL INTERLOCK ERROR
18 12 0014 902 50$: TSTW (R2)+ ; ANY REFERENCES?
903 BNEQ 80$ ; BR IF THERE IS AN OUTSTANDING REF

```



	F9	51	F5	0016	904	SOBGTR	R1,50\$	; REPEAT FOR EACH REF CNT
00	66	01	E6	0019	905	BBSSI	#CEBSV_LOCKED,CEBSL_CEBFL(R6),60\$	; LOCK IT AGAINST SEARCHES
09	66	00	E7	001D	906	BBCCI	#CEBSV_VALID,CEBSL_CEBFL(R6),70\$	; MARK IT NO LONGER VALID
51	1E	A6	9A	0021	907	MOVZBL	CEBSB_CREATPORT(R6),R1	; GET INDEX TO PORT QUOTA
7C	A541	01	58	0025	908	ADAWI	#1,SHDSW_CEFQUOTA(R5)[R1]	; RETURN PORT QUOTA
00	66	01	E7	002A	909	BBCCI	#CEBSV_LOCKED,CEBSL_CEBFL(R6),80\$	; RELEASE GSD FOR OTHER USE
00	66	02	E7	002E	910	BBCCI	#CEBSV_REFCTLCK,CEBSL_CEBFL(R6),90\$	; RELEASE REF CNT LOCK
				0032	911			
	51	8E	7D	0032	912	MOVQ	(SP)+,R1	; RESTORE REGISTERS AND STATUS CODE
			05	0035	913	RSB		; RETURN TO CALLER

```

0036 915          .SBTTL  UNIQUE_CEB - CHECK THAT SH MEM CEB IS UNIQUE
0036 916
0036 917 :++
0036 918 : FUNCTIONAL DESCRIPTION:
0036 919 :
0036 920 : THIS ROUTINE IS CALLED AFTER A SHARED MEMORY MASTER CEB IS INITIALIZED.
0036 921 : A SEARCH OF THE SPECIFIC SHARED MEMORY'S CEB TABLE IS MADE TO ASCERTAIN IF A
0036 922 : COMMON EVENT FLAG CLUSTER OF THE SAME NAME WAS CREATED DURING THE TIME THAT
0036 923 : SASCEFC WAS CREATING THE SECTION.
0036 924 :
0036 925 : TWO LOCKS MUST BE ACQUIRED BEFORE THE SHARED MEMORY CEB TABLE MAY BE SEARCHED
0036 926 : TO VERIFY A SECTION IS UNIQUE.  THE FIRST IS THE SHARED MEMORY CEB MUTEX
0036 927 : WHICH INTERLOCKS PROCESSES ON ONE PROCESSOR.  THE SECOND IS THE SHARED MEMORY
0036 928 : CEB TABLE LOCK CONTAINED IN THE SHARED MEMORY COMMON DATA PAGE, WHICH
0036 929 : INTERLOCKS BETWEEN PROCESSORS.
0036 930 :
0036 931 : THE REGISTER CONVENTION (R6=ADR OF MASTER CEB JUST CREATED AND R11=ADR OF
0036 932 : MASTER CEB THAT IS THE DUPLICATE) IS SWITCHED DURING THIS ROUTINE TO MAKE
0036 933 : THE ROUTINES INC_SHMCEB_REF AND DEC_SHMCEB_REF LOCK THE DUPLICATE ENTRY.
0036 934 :
0036 935 : CALLING SEQUENCE:
0036 936 :
0036 937 :         BSBW  UNIQUE_CEB
0036 938 :
0036 939 : INPUT PARAMETERS:
0036 940 :
0036 941 :         R4 - ADDRESS OF SHARED MEMORY CONTROL BLOCK
0036 942 :         R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
0036 943 :         R6 - ADDRESS OF COMMON EVENT BLOCK TO BE VERIFIED AS UNIQUE
0036 944 :
0036 945 : IMPLICIT INPUTS:
0036 946 :
0036 947 :         NONE
0036 948 :
0036 949 : OUTPUT PARAMETERS:
0036 950 :
0036 951 :         R11 - 0 IF THE CEB IS UNIQUE
0036 952 :              OTHERWISE, ADDRESS OF DUPLICATE CEB
0036 953 :
0036 954 : IMPLICIT OUTPUTS:
0036 955 :
0036 956 :         NONE
0036 957 :
0036 958 : COMPLETION CODES:
0036 959 :
0036 960 :         NONE
0036 961 :
0036 962 : SIDE EFFECTS:
0036 963 :
0036 964 :         NONE
0036 965 :
0036 966 : --
0036 967 :
0000047A 968          .PSECT  Y$EXEPAGED, BYTE
047A 969 UNIQUE_CEB:
047A 970          .ENABL  LSB
034F 8F  BB 047A 971          PUSHR  #*M<R0,R1,R2,R3,R6,R8,R9> ; SAVE REGISTERS
  
```

```

5B 56 D0 047E 972 MOVL R6,R11 ; REMEMBER ADR OF MASTER CEB JUST CREAT
0055 30 0481 973 BSBW SHMCEFTBLK ; GET SHM MUTEX AND BIT LOCK
4E 50 E9 0484 974 BLBC RO,ERROR_EXIT ; REPORT UNABLE TO GET BIT LOCK
00A2 C5 15 A4 90 0487 975 MOVB SHD$B_PORT(R4),SHD$B_CEFLOCK(R5) ; SET OWNER OF CEB TBL LOCK
56 55 08 A5 C1 048D 976 ADDL3 SHD$B_CEFPTR(R5),R5,R6 ; GET ADR OF FIRST CEB IN SH MEM
58 08 A6 3C 0492 977 MOVZWL CEB$W_SIZE(R6),R8 ; GET SIZE OF ONE SHMCEB
59 1C A5 3C 0496 978 MOVZWL SHD$W_CEFMAX(R5),R9 ; GET # OF CEB'S IN SHM TBL
FF83 30 049A 979 30$: BSBW INC_SHMCEB_REF ; LOCK CEB AGAINST DELETION
1F 50 E9 049D 980 BLBC RO,25$ ; DONT USE ENTRY IF IT IS REFCNT LOCKED
18 66 01 E0 04A0 981 BBS #CEB$V_LOCKED,CEB$B_CEBFL(R6),40$ ; BR IF CEB LOCKED FOR READING
14 66 00 E1 04A4 982 BBC #CEB$V_VALID,CEB$B_CEBFL(R6),40$ ; BR IF CEB IS NOT IN USE
22 AB 22 A6 B1 04AB 983 CMPW CEB$W_GRP(R6),CEB$W_GRP(R11) ; ARE CEB'S IN SAME GROUP?
0D 12 04AD 984 BNEQ 40$ ; BR IF NOT, KEEP LOOKING
28 A6 28 AB 10 29 04AF 985 CMPC3 #16,CEBST_EFCNAM(R11),CEBST_EFCNAM(R6) ; CEB'S HAVE SAME NAME?
05 12 04B5 986 BNEQ 40$ ; BR IF DIFFERENT NAMES
5B 56 D1 04B7 987 CMPL R6,R11 ; IS THIS THE CEB JUST MADE?
08 12 04BA 988 BNEQ 50$ ; BR IF DIFFERENT, FOUND DUPLICATE
FF86 30 04BC 989 40$: BSBW DEC_SHMCEB_REF ; RELEASE LOCK AGAINST DELETION
56 58 C0 04BF 990 ; IGNORE POTENTIAL INTERLOCK ERROR
D5 59 F5 04C2 992 45$: SOBGTR R9,30$ ; GET ADR OF NEXT SHMCEB
56 D4 04C5 993 ; LOOK AT EACH ENTRY IN SHM TBL
00 009F C5 04 E7 04C7 994 50$: BBCCI #SHD$V_CEFLOCK,SHD$B_FLAGS(R5),60$ ; RELEASE SHM CEB TBL LOCK
5B 56 D0 04CD 995 60$: MOVL R6,R11 ; RETURN DUPLICATE ENTRY ADR IN R11
034F 8F BA 04D0 996 ; *M<R0,R1,R2,R3,R6,R8,R9> ; RESTORE REGISTERS
05 04D4 997 RSB ; RETURN TO $ASCEFC
04D5 998
04D5 999 ; *****
04D5 1000 ; AT SOME LATER DATE, THIS SHOULD SEND AN ERROR MESSAGE TO THE ERROR LOGGER.
04D5 1001 ; *****
04D5 1002 ERROR_EXIT:
56 D4 04D5 1003 CLRL R6 ; FAILURE TO ACQUIRE BIT LOCK
F4 11 04D7 1004 BRB 60$ ; RETURN ERROR STATUS
04D9 1005 .DSABL LSB
    
```

```

04D9 1007          .SBTTL EXESCEBREFLCK - ACQUIRE SHMCEB REFERENCE COUNT LOCK
04D9 1008
04D9 1009          :++
04D9 1010          : FUNCTIONAL DESCRIPTION:
04D9 1011          :
04D9 1012          : THE ROUTINE IS CALLED TO ACQUIRE EXCLUSIVE USE OF A SHARED MEMORY COMMON
04D9 1013          : EVENT FLAG BLOCK. THIS IS DONE BY LOOPING TO ACQUIRE A BIT LOCK IN THE
04D9 1014          : MASTER CEB. IF THE BIT LOCK COULD NOT BE ACQUIRED, THEN AN ERROR CODE IS
04D9 1015          : RETURNED.
04D9 1016          :
04D9 1017          :
04D9 1018          : CALLING SEQUENCE:
04D9 1019          :
04D9 1020          :     BSBW  EXESCEBREFLCK
04D9 1021          :
04D9 1022          : INPUT PARAMETERS:
04D9 1023          :
04D9 1024          :     R6 - ADDRESS OF SHARED MEMORY COMMON EVENT BLOCK
04D9 1025          :
04D9 1026          : IMPLICIT INPUTS:
04D9 1027          :
04D9 1028          :     NONE
04D9 1029          :
04D9 1030          : OUTPUT PARAMETERS:
04D9 1031          :
04D9 1032          :     R0 - STATUS CODE
04D9 1033          :
04D9 1034          : IMPLICIT OUTPUTS:
04D9 1035          :
04D9 1036          :     NONE
04D9 1037          :
04D9 1038          : COMPLETION CODES:
04D9 1039          :
04D9 1040          :     $$$_NORMAL - SUCCESSFULLY ACQUIRED LOCK
04D9 1041          :     $$$_INTERLOCK - UNABLE TO ACQUIRE LOCK
04D9 1042          :
04D9 1043          : SIDE EFFECTS:
04D9 1044          :
04D9 1045          :     NONE
04D9 1046          :
04D9 1047          :--
04D9 1048
00000036 1049          .PSECT  AEXENONPAGED,BYTE
0036 1050 EXESCEBREFLCK:
51 00000000'GF 51 DD 0036 1051          PUSHL  R1          ; SAVE REGISTER
07 66 02 E6 003F 1052          MOVL   G*EXESGL LOCKRTRY,R1 ; GET LOOP COUNT FOR BIT LOCK
50 01 9A 0043 1053 10$:  BBSSI  #CEBSV REFCNTLCK,CEBSL_CEBFL(R6),20$ ; TRY TO ACQUIRE BIT LOCK
51 8ED0 0046 1054          MOVZBL #$$$_NORMAL,R0      ; REPORT LOCK SUCCESSFULLY ACQUIR
05 0049 1055          POPL  R1          ; RESTORE REGISTER
F2 51 F5 004A 1056          RSB          ; RETURN SUCCESS CODE
51 8ED0 004D 1057 20$:  SOBGR  R1,10$          ; TRY AGAIN TO ACQUIRE BIT LOCK
50 038C BF 3C 0050 1058          POPL  R1          ; RESTORE REGISTER
05 0055 1059          MOVZWL #$$$_INTERLOCK,R0 ; REPORT ERROR STATUS
05 0055 1060          RSB          ; RETURN TO CALLER

```

```

0056 1062      .SBTTL SHMCEFTBLK - ACQUIRE EXCLUSIVE USE OF SHARED MEMORY TABLE
0056 1063
0056 1064      :++
0056 1065      : FUNCTIONAL DESCRIPTION:
0056 1066      :
0056 1067      : THE ROUTINE IS CALLED TO ACQUIRE EXCLUSIVE USE OF A SHARED MEMORY COMMON
0056 1068      : EVENT FLAG TABLE. THIS IS DONE BY LOOPING TO ACQUIRE A SHARED MEMORY BIT
0056 1069      : LOCK. IF THE BIT LOCK COULD NOT BE ACQUIRED, THEN AN ERROR CODE IS RETURNED.
0056 1070      :
0056 1071      : CALLING SEQUENCE:
0056 1072      :
0056 1073      :     BSBW  SHMCEF'BLK
0056 1074      :
0056 1075      : INPUT PARAMETERS:
0056 1076      :
0056 1077      :     R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
0056 1078      :
0056 1079      : IMPLICIT INPUTS:
0056 1080      :
0056 1081      :     NONE
0056 1082      :
0056 1083      : OUTPUT PARAMETERS:
0056 1084      :
0056 1085      :     R0 - STATUS CODE
0056 1086      :
0056 1087      : IMPLICIT OUTPUTS:
0056 1088      :
0056 1089      :     NONE
0056 1090      :
0056 1091      : COMPLETION CODES:
0056 1092      :
0056 1093      :     SSS_NORMAL - SUCCESSFULLY ACQUIRED LOCK
0056 1094      :     SSS_INTERLOCK - UNABLE TO ACQUIRE LOCK
0056 1095      :
0056 1096      : SIDE EFFECTS:
0056 1097      :
0056 1098      :     NONE
0056 1099      :
0056 1100      :--
0056 1101
000004D9 1102      .PSECT YSEXEPAGED, BYTE
04D9 1103 SHMCEFTBLK:
51 DD 04D9 1104      PUSHL  R1          ; SAVE REGISTER
07 009F C5 04 E6 04DB 1105      MOVL  G^EXE$GL LOCKRTRY,R1 ; GET LOOP COUNT FOR BIT LOCK
50 01 9A 04E8 1106 10$:  BBSSI  #SHD$V CEFLCK,SHD$B_FLAGS(R5),20$ ; TRY TO ACQUIRE BIT LOCK
51 8ED0 04EB 1107      MOVZBL #SS$_NORMAL,R0 ; REPORT LOCK SUCCESSFULLY ACQUIR
05 04EE 1108      POPL  R1          ; RESTORE REGISTER
F0 51 F5 04EF 1109      RSB ; RETURN SUCCESS CODE
51 8ED0 04F2 1110 20$:  SOBGR  R1,10$ ; TRY AGAIN TO ACQUIRE BIT LOCK
50 038C 8F 3C 04F5 1111      POPL  R1          ; RESTORE REGISTER
05 04FA 1112      MOVZWL #SS$_INTERLOCK,R0 ; REPORT ERROR STATUS
1113      RSB ; RETURN TO CALLER

```

```

04FB 1115      .SBTTL SHD_FIND_SHB - FIND SH MEM CONTROL BLOCK FOR SPECIFIC SHD
04FB 1116
04FB 1117      :++
04FB 1118      : FUNCTIONAL DESCRIPTION:
04FB 1119      :
04FB 1120      : THIS ROUTINE TAKES COMMON DATA PAGE ADDRESS AND FINDS THE SHARED MEMORY
04FB 1121      : CONTROL BLOCK ADDRESS FOR THIS SHARED MEMORY.
04FB 1122      :
04FB 1123      : CALLING SEQUENCE:
04FB 1124      :
04FB 1125      :     BSBW  SHD_FIND_SHB
04FB 1126      :
04FB 1127      : INPUT PARAMETERS:
04FB 1128      :
04FB 1129      :     R5 - ADDRESS OF SHARED MEMORY COMMON DATA PAGE
04FB 1130      :
04FB 1131      : IMPLICIT INPUTS:
04FB 1132      :
04FB 1133      :     NONE
04FB 1134      :
04FB 1135      : OUTPUT PARAMETERS:
04FB 1136      :
04FB 1137      :     R0 - STATUS CODE
04FB 1138      :     R4 - SHB ADR, OR 0
04FB 1139      :
04FB 1140      : IMPLICIT OUTPUTS:
04FB 1141      :
04FB 1142      :     NONE
04FB 1143      :
04FB 1144      : COMPLETION CODES:
04FB 1145      :
04FB 1146      :     SSS_NORMAL - SUCCESSFULLY FOUND SHB ADR
04FB 1147      :     SSS_SHMNOTCNCT - SHARED MEMORY NOT CONNECTED
04FB 1148      :
04FB 1149      : SIDE EFFECTS:
04FB 1150      :
04FB 1151      :     NONE
04FB 1152      :
04FB 1153      :--
04FB 1154
000004FB 1155      .PSECT  YSEXEPAGED,BYTE
04FB 1156      SHD_FIND_SHB:
04FB 1157      .ENABL  LSB
04FB 1158      MOVZBL  #SS$ NORMAL,R0      ; ASSUME SUCCESS
54 00000000'EF  D0 04FE 1159      MOVL    EXE$GL SHBLIST,R4      ; GET ADR OF FIRST SHB
07 0B A4 00  E1 0507 1160 10$:  BEQL    NOT_FOUND          ; BR IF NO SHB ADR
55 04 A4 01  D1 050C 1161      BBC    #SHB$V CONNECT,SHB$B FLAGS(R4),GET_NXT_SHM ; BR IF MEM NOT CNCT
05 0512 1162      CML    SHB$L DATAPAGE(R4),R5 ; DOES THE COM DATA PAGE ADR MATCH?
05 0513 1163      BNEQ   GET_NXT_SHM        ; BR IF NO MATCH, FIND NEXT SHB ADR
05 0513 1164      RSB                    ; SHB FOUND, RETURN ADR
05 0513 1165      GET_NXT_SHM:
05 0513 1166      MOVL   SHB$L_LINK(R4),R4      ; GET NEXT SHB IN LIST
05 0516 1167      BRB    10$                ; CONTINUE
05 0518 1168      NOT_FOUND:
50 037C 8F 3C 0518 1169      MOVZWL #SS$_SHMNOTCNCT,R0 ; NO MORE MEM, REPORT SHB NOT FOUND
05 051D 1170      RSB
05 051E 1171      .DSABL LSB

```

SYSASCEFC  
V04-000

G 14  
- ASSOCIATE, DISASSOCIATE AND DELETE COM 16-SEP-1984 01:39:05 VAX/VMS Macro V04-00  
SHD\_FIND\_SHB - FIND SH MEM CONTROL BLOCK 5-SEP-1984 03:48:41 [SYS.SRC]SYSASCEFC.MAR;1

Page 28  
(1)

051E 1172  
051E 1173  
051E 1174 .END

SYS  
V04

ALOSHMCB	00000398	R	03	JIBSW_TQCNT	= 00000034		
ASSOCIATE	00000188	R	03	LOCLUST	00000289	R	03
BUGS_CEBREFNEG	*****	X	03	MMGSCEFTRNLOG	*****	X	03
CEBSB_CREATPORT	= 0000001E			MMGSFINDSHB	*****	X	03
CEBSB_DELETPORT	= 0000001F			NAME	= 00000008		
CEBSB_LOCK	= 0000001C			NOT_FOUND	00000518	R	03
CEBSB_PROCCNT	= 0000001D			NO_FREE_CEB	00000407	R	03
CEBSB_STS	= 0000000B			NO_QUOTA	0000040E	R	03
CEBSB_TYPE	= 0000000A			PCBSL_EFC2P	= 00000058		
CEBSL_CEBFL	= 00000000			PCBSL_JIB	= 00000080		
CEBSL_EFC	= 00000010			PCBSL_PHD	= 0000006C		
CEBSL_MASTER	= 00000040			PCBSL_PID	= 00000060		
CEBSL_PID	= 0000000C			PCBSL_UIC	= 000000BC		
CEBSL_SHB	= 00000038			PCBSW_GRP	= 000000BE		
CEBSL_UIC	= 00000020			PERM	= 00000010		
CEBSL_VASLAVE1	= 00000038			PRS_IPL	= 00000012		
CEBSL_WQBL	= 00000018			PROT	= 0000000C		
CEBSL_WQFL	= 00000014			PRVSV_PRCB	= 0000000A		
CEBST_EFCNAM	= 00000028			PRVSV_SHMEM	= 0000001B		
CEBSV_LOCKED	= 00000001			RETQUOTA	00000000	R	01
CEBSV_NOQUOTA	= 00000000			SCHSC_CEF	= 00000003		
CEBSV_PERM	= 00000001			SCHSGC_PCBVEC	*****	X	01
CEBSV_REFCNTLCK	= 00000002			SCHSGQ_CEBHD	*****	X	03
CEBSV_VALID	= 00000000			SCHSGW_CEBCNT	*****	X	03
CEBSW_GRP	= 00000022			SCHSLOCKW	*****	X	03
CEBSW_INDX	= 0000003C			SCHSUNLOCK	*****	X	03
CEBSW_PROT	= 00000024			SHBSB_FLAGS	= 0000000B		
CEBSW_REFC	= 00000026			SHBSB_PORT	= 00000015		
CEBSW_SIZE	= 00000008			SHBSL_DATAPAGE	= 00000004		
CEBSW_STATE	= 0000001E			SHBSL_LINK	= 00000000		
CEBSW_TQCNT	= 0000001C			SHBSV_CONNECT	= 00000000		
CEBLOCK	0000027C	R	03	SHDSB_CEFLOCK	= 000000A2		
DEC_SHMCEB_REF	00000445	R	03	SHDSB_FLAGS	= 0000009F		
DISASOC	00000194	R	03	SHDSL_CFPTR	= 00000008		
DYNBC_SLAVCEB	= 0000002D			SHDSV_CEFCK	= 00000004		
EFN	= 00000004			SHDSW_CEFMAX	= 0000001C		
ERROR_EXIT	000004D5	R	03	SHDSW_CEFQUOTA	= 0000007C		
ERR_BUGCHK	00000476	R	03	SHD_FIND_SHB	000004FB	R	03
EXESALLOCCB	*****	X	03	SHMCEFTBCK	000004D9	R	03
EXESASCEFC	00000027	RG	03	SSS_ACCVIO	= 0000000C		
EXESCEBREFLCK	00000036	RG	04	SSS_EXPORTQUOTA	= 000003AC		
EXESDACEFC	00000000	RG	03	SSS_EXQUOTA	= 0000001C		
EXESDEANONPAGED	*****	X	03	SSS_ILLEFC	= 000000EC		
EXESDLCEFC	00000215	RG	03	SSS_INTERLOCK	= 0000038C		
EXESGL_CEBMTX	*****	X	03	SSS_IVLOGNAM	= 00000154		
EXESGL_LOCKRTRY	*****	X	04	SSS_NOPRIV	= 00000024		
EXESGL_SHBLIST	*****	X	03	SSS_NORMAL	= 00000001		
EXESSHMCEBDEL	00000000	RG	04	SSS_NOSHMBLOCK	= 00000384		
EXIT	000001FD	R	03	SSS_SHMNOTCNCT	= 0000037C		
EXITE	000001FF	R	03	UNIQUE_CEB	0000047A	R	03
EXITM	000001FA	R	03				
FIND_SHM_CEB	0000032A	R	03				
GETCLUSTER	00000015	R	03				
GET_NXT_SHM	00000513	R	03				
INC_SHMCEB_REF	00000420	R	03				
IPLS_SYNCH	= 00000008						
JIBSL_MPID	= 00000054						



+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS :	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK :	0000001F ( 31.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$AB\$\$	00000000 ( 0.)	02 ( 2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YSEXEPAGED	0000051E ( 1310.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
AEXENONPAGED	00000056 ( 86.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.05	00:00:01.22
Command processing	125	00:00:00.59	00:00:03.35
Pass 1	356	00:00:11.89	00:00:34.27
Symbol table sort	0	00:00:01.65	00:00:05.37
Pass 2	221	00:00:03.29	00:00:11.87
Symbol table output	14	00:00:00.10	00:00:00.41
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	756	00:00:17.61	00:00:56.53

The working set limit was 1650 pages.  
71103 bytes (139 pages) of virtual memory were used to buffer the intermediate code.  
There were 60 pages of symbol table space allocated to hold 1081 non-local and 74 local symbols.  
1174 source lines were read in Pass 1, producing 19 object records in Pass 2.  
25 pages of virtual memory were used to define 24 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	14
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	21

1184 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

MACRO IS=LIS\$:SYSASCEFC/OBJ=OBJ\$:SYSASCEFC MSRC\$:SYSASCEFC/UPDATE=(ENH\$:SYSASCEFC)+EXECMLS/LIB

