


```

SSSSSSSS YY YY SSSSSSS AAAAAA DDDDDDD JJ WW WW SSSSSSS LL
SSSSSSSS YY YY SSSSSSS AAAAAA DDDDDDD JJ WW WW SSSSSSS LL
SS YY YY SS SSSSSS AA AA DD DD JJ JJ WW WW SS SSSSSSS LL
SS YY YY SS SSSSSS AA AA DD DD JJ JJ WW WW SS SSSSSSS LL
SS YY YY SS SSSSSS AA AA DD DD JJ JJ WW WW SS SSSSSSS LL
SSSSSSS YY YY SSSSSS AA AA DD DD JJ JJ WW WW SS SSSSSSS LL
SSSSSSS YY YY SSSSSS AA AA DD DD JJ JJ WW WW SS SSSSSSS LL
SS YY YY SS SSSSSS AA AA DD DD JJ JJ WW WW SS SSSSSSS LL
SS YY YY SS SSSSSS AA AA DD DD JJ JJ WW WW SS SSSSSSS LL
SSSSSSSS YY YY SSSSSSS AA AA DDDDDDD JJJJJJ WW WW SSSSSSS LL
SSSSSSSS YY YY SSSSSSS AA AA DDDDDDD JJJJJJ WW WW SSSSSSS LL

```

```

LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL IIIIII SSSSSSS
LLLLLLLLLL IIIIII SSSSSSS
LLLLLLLLLL IIIIII SSSSSSS

```



```

0000 1      .TITLE SYSADJWSL - SYSTEM SERVICE ADJUST WORKING SET LIMIT
0000 2      .IDENT 'V04-000'
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27
0000 28 :++
0000 29 : FACILITY: EXECUTIVE, MEMORY MANAGEMENT SERVICE
0000 30
0000 31 : ABSTRACT: SYSADJWSL IMPLEMENTS THE ADJUST WORKING SET LIMIT
0000 32 : SYSTEM SERVICE.
0000 33
0000 34 : ENVIRONMENT:
0000 35
0000 36 : AUTHOR: PETER H. LIPMAN , CREATION DATE: 10-DEC-76
0000 37
0000 38 : MODIFIED BY:
0000 39
0000 40 : V03-005 WMC0003 Wayne Cardoza 05-MAY-1983
0000 41 : Change ESTRADYNWS calculation to account for locked page
0000 42 : tables in the dynamic portion of the working set.
0000 43
0000 44 : V03-004 TCM0001 Trudy C. Matthews 1-Apr-1983
0000 45 : Change references to working set fields in PHD so that
0000 46 : they are used as unsigned words.
0000 47
0000 48 : V03-003 WMC0002 Wayne Cardoza 22-Dec-1982
0000 49 : Expansion of process header for new working set entries must
0000 50 : be iterative process. Don't request more than can currently
0000 51 : be locked in working set.
0000 52
0000 53 : V03-002 WMC0001 Wayne Cardoza 04-Oct-1982
0000 54 : EXTRADYNS changed to use minimum of quota and size.
0000 55
0000 56 : V03-001 HRJ0062 Herb Jacobs 05-Apr-1982
0000 57 : Fix EXTRADYNWS calculation. This could cause IPL 2 loops

```

Pha

Ini
Com
Pas
Sym
Sym
Pse
Cro
Ass

The
514
The
444
17

Mac

_S2
_S2
TOT

102

The
MAC

0000 58 :
0000 59 :
0000 60 :--

by being too leniant.

```
0000 62 .SBTTL DECLARATIONS
0000 63 :
0000 64 : INCLUDE FILES:
0000 65 :
0000 66 $ACMDEF ;ACCOUNTING MESSAGE DEFINITIONS
0000 67 $IPLDEF ;PROCESSOR PRIORITY LEVELS
0000 68 $PCBDEF ;PROCESS CONTROL BLOCK DEFINITIONS
0000 69 $PHDDEF ;PROCESS HEADER DEFINITIONS
0000 70 $PRDEF ;PROCESSOR REGISTER DEFINITIONS
0000 71 $PTEDEF ;PAGE TABLE ENTRY DEFINITIONS
0000 72 $SECDEF ;SECTION TABLE OFFSET DEFINITIONS
0000 73 $SSDEF ;SYSTEM SERVICE DEFINITIONS
0000 74 :
0000 75 : MACROS:
0000 76 :
0000 77 :
0000 78 :
0000 79 : EQUATED SYMBOLS:
0000 80 :
0000 81 : OFFSETS FROM AP
0000 82 :
00000004 0000 83 PAGCNT = 4 ;NUMBER OF PAGES TO ADD OR SUBTRACT
00000008 0000 84 WSETLM = 8 ;ADDRESS TO RETURN NEW WORKING SET LIMIT
0000 85 :
0000 86 : OWN STORAGE:
0000 87 :
```

```

0000 89 .SBTTL ADD TO WORKING SET SERVICE
0000 90 :++
0000 91 : FUNCTIONAL DESCRIPTION:
0000 92 :
0000 93 :
0000 94 : CALLING SEQUENCE:
0000 95 :
0000 96 : CALL ARGLIST,SYSADJWSL
0000 97 :
0000 98 : INPUT PARAMETERS:
0000 99 :
0000 100 : PAGCNT(AP) = NUMBER OF PAGES TO ADD IF POSITIVE
0000 101 : = NUMBER OF PAGES TO SUBTRACT IF NEGATIVE
0000 102 : WSETLM(AP) = ADDRESS OF LONG WORD TO RETURN NEW WORKING SET LIMIT
0000 103 : R4 = PROCESS CONTROL BLOCK ADDRESS
0000 104 : SET UP BY THE CHANGE MODE TO KERNEL DISPATCHER
0000 105 :
0000 106 : IMPLICIT INPUTS:
0000 107 :
0000 108 : NONE
0000 109 :
0000 110 : OUTPUT PARAMETERS:
0000 111 :
0000 112 : R0 = SYSTEM STATUS CODE
0000 113 :
0000 114 : IMPLICIT OUTPUTS:
0000 115 :
0000 116 : NONE
0000 117 :
0000 118 : COMPLETION CODES:
0000 119 :
0000 120 : NONE
0000 121 :
0000 122 : SIDE EFFECTS:
0000 123 :
0000 124 : NONE
0000 125 :
0000 126 : --
0000 127 :
0000 128 : *****
0000 129 :
0000 130 : ***** THE FOLLOWING CODE MAY BE PAGED *****
0000 131 :
0000 132 : .PSECT YSEXEPAGED
0000 133 :
0000 134 : *****
0000 135 :
003C 0000 136 EXESADJWSL::
0002 137 .WORD ^M<R2,R3,R4,R5>
0005 138 SETIPL #IPL$ASTDEL ;NO AST'S WHILE MODIFYING PHD
000C 139 MOVL G^CTL$GL PHD,R5 ;PROCESS HEADER ADDRESS (P1 SPACE)
0010 140 MOVL PAGCNT(AP),R1 ;NO. OF PAGES TO ADD TO WORKING SET
0010 141 ;***** REFERENCE COULD PAGE FAULT
0010 142 BGEQ 30$ ;BRANCH IF GROW WORKING SET
0012 143 JSB SHRINK ;SHRINK THE WORKING SET
0018 144 BRW 60$ ;EXIT SYSTEM SERVICE
001B 145 :

```

```

001B 146 : MINIMUM OF R1 AND R2 TO R1
001B 147 :
52 51 D1 001B 148 10$: CMPL R1,R2 ;USE THE SMALLER
15 001E 149 BLEQ 20$ ;BRANCH IF R1 IS THE SMALLER
51 52 D0 0020 150 MOVL R2,R1 ;USE R2 INSTEAD
05 0023 151 20$: RSB
0024 152
52 00000000'EF D0 0024 153 30$: MOVL L*SGN$GL MAXWSCNT,R2 ;MAXIMUM SYSTEM WIDE WORKING SET SIZE
50 50 A5 3C 002B 154 MOVZWL PHD$W_WSSIZE(R5),R0 ;CURRENT WORKING SET SIZE
53 08 A5 3C 002F 155 MOVZWL PHD$W_WSLIST(R5),R3 ;GET START OF WSL ENTRIES
53 D7 0033 156 DECL R3 ;ADJUST TO GET COUNT WHEN SUBTRACTED
52 50 C2 0035 157 SUBL R0,R2 ;PAGES THAT POTENTIALLY MAY BE ADDED
E1 10 0038 158 BSBB 10$ ;USE SMALLER OF REQUEST AND MAX
52 16 A5 3C 003A 159 MOVZWL PHD$W_WSEXTENT(R5),R2 ;GET POINTER TO END OF MAX SIZE
52 53 C2 003E 160 SUBL R3,R2 ;GET MAXIMUM ALLOWED SIZE
52 50 C2 0041 161 SUBL R0,R2 ;REDUCE BY WHAT'S IN USE
D5 10 0044 162 BSBB 10$ ;USE SMALLER OF REQUEST AND PROCESS MAX
00000000'EF C3 0046 163 SUBL3 SCH$GL_FREELIM,- ;MAX PHYSICAL NUMBER OF PAGES
52 00000000'EF 004C 164 PINS$GL_PHYPGCNT,R2 ;IS UPPER BOUND ON WORKING SET SIZE
52 50 C2 0052 165 SUBL R0,R2 ;MAXIMUM NUMBER OF ADDITIONAL PAGES
C4 10 0055 166 BSBB 10$ ;USE SMALLER
52 12 A5 3C 0057 167 MOVZWL PHD$W_WSLAST(R5),R2 ;GET POINTER TO END OF CURRENT SIZE
52 53 C2 005B 168 SUBL R3,R2 ;NUMBER OF POTENTIAL PAGES
52 50 C2 005E 169 SUBL R0,R2 ;GET NUMBER OF UNUSED WSLE IN WSL
52 51 D1 0061 170 CMPL R1,R2 ;IS REQUEST FOR LESS THAN UNUSED ONES?
03 18 0064 171 BGEQ 40$ ;BRANCH IF NOT
52 51 D0 0066 172 MOVL R1,R2 ;USE ONLY REQUEST SIZE
50 A5 52 A0 0069 173 40$: ADDW R2,PHD$W_WSSIZE(R5) ;ADD IN THE FREE EXPANSION AMOUNT
51 52 C2 006D 174 SUBL R2,R1 ;NUMBER OF ADDITIONAL PAGES BEYOND WSLAST
4D 15 0070 175 BLEQ 60$ ;BRANCH IF DONE
51 D0 0072 176 42$: PUSHL R1 ;SAVE THE REQUEST SIZE
50 76 A5 3C 0074 177 MOVZWL PHD$W_EXTDYNWS(R5),R0 ;EXTRA WORKING SET ENTRIES
50 50 07 78 0078 178 INCL R0 ;THERE IS ALWAYS AT LEAST ONE MORE THAN EXTR
50 50 51 D1 007E 180 CMPL R1,R0 ;LONGWORDS OF PROCESS HEADER THIS CAN LOCK
03 15 0081 181 BLEQ 45$ ;CAN WE LOCK ENTIRE REQUEST
51 50 D0 0083 182 MOVL R0,R1 ;NO PROBLEM
00000000'EF 16 0086 183 45$: JSB MMG$ALCPHD ;REQUEST LESS
53 8E D0 008C 184 MOVL (SP)+,R3 ;GET ENOUGH SPACE FOR SPECIFIED # OF WSLE'S
008F 185 ;GET BACK THE REQUEST SIZE
008F 186 : R1 IS RETURNED AS THE MINIMUM OF WHAT WAS REQUESTED AND WHAT WAS AVAILABLE
008F 187 : NOTE THAT THE FOLLOWING CODE MUST WORK CORRECTLY IF NO NEW ENTRIES ARE ADDED
008F 188 :
52 51 D0 008F 189 MOVL R1,R2 ;SAVE COUNT FOR AFTER LOOP
2B 13 0092 190 BEQL 60$ ;BRANCH IF HEADER COULDN'T BE EXPANDED
50 12 A5 3C 0094 191 MOVZWL PHD$W_WSLAST(R5),R0 ;GET INDEX TO CURRENT END
10 A5 50 B0 0098 192 MOVW R0,PHD$W_WSNEXT(R5) ;UPDATE NEXT POINTER TO NEW FREE AREA
50 04 A540 DE 009C 193 MOVAL 4(R5)[R0],R0 ;UPDATE NEXT POINTER TO NEW FREE AREA
80 D4 00A1 194 50$: CLRL (R0)+ ;MARK ENTRY FREE
FB 51 F5 00A3 195 SOBGTR R1,50$ ;ONCE FOR EACH NEW WORKING SET ENTRY
12 A5 52 A0 00A6 196 ADDW R2,PHD$W_WSLAST(R5) ;UPDATE TO NEW WSLAST
50 A5 52 A0 00AA 197 ADDW R2,PHD$W_WSSIZE(R5) ;UPDATE TO NEW WORKING SET SIZE
53 52 D1 00AE 198 CMPL R2,R3 ;DO WE HAVE ALL WE NEEDED
OC 18 00B1 199 BGEQ 60$ ;YES
000000C2'EF 16 00B3 200 JSB MM $EXTRADYNWS ;RECOMPUTE EXTRA DYNAMIC WSLE COUNT
51 53 52 C3 00B9 201 SUBL3 R2,R3,R1 ;AMOUNT WE STILL NEED
B3 11 00BD 202 BRB 42$ ;GO TRY AGAIN

```


000000C2'EF	16	00BF	203	60\$:	JSB	MMG\$EXTRADYNWS	:RECOMPUTE EXTRA DYNAMIC WSLE COUNT	
16	10	00C5	204		BSBB	MMG\$WSPEAKCHK	:ENABLE WORKING SET PEAK CHECK	
		00C7	205				:RETURN WORKING SET SIZE IN R1	
52	08	AC	D0	00C7	206	MOVL	WSETLM(AP),R2	:ADDRESS TO RETURN NEW WORKING SET LIMIT
		00CB	207				:***** COULD FAULT THIS REFERENCE	
	0C	13	00CB	208	BEQL	80\$:BRANCH IF NONE SPECIFIED	
50	0C	3C	00CD	209	MOVZWL	#SS\$ ACCVIO,R0	:RETURN CODE FOR ACCESS VIOLATION	
			00D0	210	IFNOWRT	#4,(R2),90\$:IF NOT WRITABLE	
62	51	3C	00D6	211	MOVZWL	R1,(R2)	:OTHERWISE RETURN THE NEW SIZE	
			00D9	212			:***** COULD FAULT THIS REFERENCE	
50	01	3C	00D9	213	MOVZWL	#SS\$_NORMAL,R0	:INDICATE SUCCESSFUL COMPLETION	
		04	00DC	214	90\$:	RET	:AND RETURN	

```

00DD 216 .SBTTL WSPEAKCHK - ENABLE OR DISABLE WORKING SET PEAK CHECKING
00DD 217
00DD 218
00DD 219 : CALLING SEQUENCE:
00DD 220
00DD 221 BSBW MMGSWSPEAKCHK
00DD 222
00DD 223 : INPUTS:
00DD 224
00DD 225 R5 = PROCESS HEADER ADDRESS (P1 SPACE OK)
00DD 226
00DD 227 : OUTPUTS:
00DD 228
00DD 229 R1 = WORKING SET SIZE
00DD 230 IF THE CURRENT PEAK IS LESS THAN THE CURRENT WORKING SET
00DD 231 SIZE, THE WORKING SET PEAK CHECK IS ENABLED.
00DD 232 IF THE CURRENT PEAK IS GREATER THAN OR EQUAL TO THE
00DD 233 CURRENT WORKING SET SIZE, THE WORKING SET PEAK CHECK IS DISABLED.
00DD 234
00DD 235 *****
00DD 236 ***** THE FOLLOWING CODE MAY BE PAGED *****
00DD 237
00DD 238
000000DD 239 .PSECT YSEXEPAGED
00DD 240
00DD 241 *****
00DD 242
00DD 243 MMGSWSPEAKCHK::
36 A5 14 AA 00DD 244 BICW #<PHDSM_WSPEAKCHK!PHDSM_IWSPEAKCK>, -
00E1 245 PHDSW_FLAGS(R5) :DISABLE WORKING SET PEAK CHECK
51 50 A5 3C 00E1 246 MOVZWL PHDSW_WSSIZE(R5),R1 :RETURN WSSIZE IN R1
00000000'GF 51 B1 00E5 247 CMPW R1,G^CTL$GL_WSPEAK :POSSIBLE TO EXCEED CURRENT PEAK?
04 1F 00EC 248 BLSSU 10$ :BRANCH IF NOT
36 A5 04 A8 00EE 249 BISW #PHDSM_WSPEAKCHK,PHDSW_FLAGS(R5) :YES, ENABLE THE CHECK
OD 00000000'EF 01 E1 00F2 250 10$: BBC #ACMSV_IMAGE,EXE$GL_ACM_FLAGS,20$ : IMAGE ACNT. ENABLED ?
00000000'GF 51 B1 00FA 251 CMPW R1,G^CTL$GL_IWSPEAK :POSSIBLE TO EXCEED CURRENT PEAK?
04 1F 0101 252 BLSSU 20$ :BRANCH IF NOT
36 A5 10 A8 0103 253 BISW #PHDSM_IWSPEAKCK,PHDSW_FLAGS(R5) :YES, ENABLE THE CHECK
05 0107 254 20$: RSB

```

```

0108 256 .SBTTL SHRINK WORKING SET
0108 257 :
0108 258 *****
0108 259 *****
0108 260 ***** THE FOLLOWING CODE MUST BE RESIDENT *****
0108 261 :
00000000 262 .PSECT $MMGCOD
0000 263 :
0000 264 *****
0000 265 *****
0000 266 ADJUST THE WORKING SET POINTERS TO REFLECT THE NEW SIZE OF THE WORKING SET
0000 267 :
0000 268 CALLING SEQUENCE:
0000 269 :
0000 270 BSB/JSB MMG$SHRINKWS
0000 271 :
0000 272 INPUTS:
0000 273 :
0000 274 R1 = NEGATIVE NUMBER OF PAGES TO DELETE FROM WORKING SET
0000 275 :
0000 276 :
0000 277 REDUCE THE SIZE OF THE WORKING SET
0000 278 :
0000 279 SHRINK:
0000 280 SETIPL #IPL$_SYNCH ;DISABLE SWAPPER
7E 12 A5 3C 0003 281 MOVZWL PHD$W_WSLAST(R5),-(SP) ;SAVE IN CASE FREWSLE SHRINKS THIS
7E 50 A5 3C 0007 282 MOVZWL PHD$W_WSSIZE(R5),-(SP) ;SAVE IN CASE RESOURCEWAIT NEEDED
2C 10 000B 283 BSBW MMG$SHRINKWS ;SHRINK THE WORKING SET
0C BA 000D 284 POPR #*M<R2,R3> ;GET BACK ORIGINAL WSSIZE AND WSLAST
04 12 000F 285 BNEQ 10$ ;BRANCH IF NON SUCCESSFUL
0011 286 SETIPL #IPL$_ASTDEL ;SWAPPABLE AGAIN
05 0014 287 RSB
0015 288 :
0015 289 : MUST WAIT FOR A RESOURCE
0015 290 : IPL = SYNCH, R1 = RESOURCE TO WAIT FOR
0015 291 : R4 = PROCESS CONTROL BLOCK ADDRESS
0015 292 :
50 A5 52 B0 0015 293 10$: MOVW R2,PHD$W_WSSIZE(R5) ;RESET ORIGINAL WS SIZE FOR RETRY
12 A5 53 B0 0019 294 MOVW R3,PHD$W_WSLAST(R5) ;RESET WSLAST, ENTRIES ARE KNOWN ZEROED
02 BB 001D 295 PUSHR #*M<R1> ;SAVE RESOURCE WAIT CODE
00A0 30 001F 296 BSBW MMG$EXTRADYNWS ;RESET EXTRA DYNAMIC WORKING SET COUNT
02 BA 0022 297 POPR #*M<R1> ;RESURE RESOURCE WAIT NUMBER
4C A4 51 D0 0024 298 MOVL R1,PCB$E_FWM(R4) ;SET RESOURCE TO WAIT FOR
00 0000'CF 51 E6 0028 299 BBSSI R1,W^SCH$GL_RESMASK,20$ ;NOTE SOMEONE WAITING
52 0000'CF 7E 002E 300 20$: MOVAQ W^SCH$GQ_MWAIT,R2 ;WAIT ON MUTEX WAIT QUEUE
5E 5D D0 0033 301 MOVL FP,SP ;RESET FP, AP UNTOUCHED
FFC7' 31 0036 302 BRW SCH$WAIT ;WAIT AS CALLER
0039 303 :
0039 304 MMG$SHRINKWS::
00 0039 305 PUSHL #0 ;SET DEFAULT RETURN STATUS
52 52 DD 003B 306 PUSHL R2 ;GET A SCRATCH REGISTER
52 50 A5 3C 003D 307 MOVZWL PHD$W_WSSIZE(R5),R2 ;KEEP WORKING SET SIZE IN R2
53 51 CE 0041 308 MNEGL R1,R3 ;MAKE THE DESIRED PAGE COUNT POSITIVE
0044 309 :
0044 310 : CALCULATE THE MAXIMUM AMOUNT THE WORKING SET CAN BE REDUCED
0044 311 :
50 0E A5 08 A5 A3 0044 312 SUBW3 PHD$W_WSLIST(R5),PHD$W_WSDYN(R5),R0 ;GET SIZE OF LOCKED WS

```

50	50	50	3C	004A	313	MOVZWL	R0,R0		:GET IT IN A LONGWORD
51	52	50	C3	004D	314	SUBL3	R0,R2,R0		:GET SIZE OF UNLOCKED WS (WSSIZE-LOCKED)
	0000	CF	3C	0051	315	MOVZWL	W*SGN\$GW_MINWSCNT,R1		:GET MINIMUM WS SIZE
	50	51	C2	0056	316	SUBL	R1,R0		:ALLOW CUSHION PAGES
		50	D7	0059	317	DECL	R0		:IN CASE CUSHION IS 0
				005B	318				
				005B	319				
				005B	320				
				005B	321				
				005B	322				
51	18 A5	08 A5	A3	005B	323	SUBW3	PHD\$W_WSLIST(R5),PHD\$W_WSQUOTA(R5),R1		
		51	B6	0061	324	INCW	R1		:QUOTA
	51	51	3C	0063	325	MOVZWL	R1,R1		:GET IT IN A LONGWORD
	51	52	C3	0066	326	SUBL3	R1,R2,R1		:AMOUNT SIZE EXCEEDS QUOTA
		06	15	006A	327	BLEQ	5\$:SIZE SMALLER - USE EXTRADYNS
	51	76 A5	A0	006C	328	ADDW	PHD\$W_EXTDYNWS(R5),R1		:ADD IN THE EXCESS DYNAMIC ENTRIES
		06	11	0070	329	BRB	7\$		
	51	76 A5	3C	0072	330	5\$: MOVZWL	PHD\$W_EXTDYNWS(R5),R1		:EXCESS DYNAMIC WORKING SET LIST ENTRIES
		43	13	0076	331	BEAL	60\$:BRANCH IF NONE LEFT TO TAKE AWAY
				0078	332				
				0078	333				
				0078	334				
				0078	335				
				0078	336				
				0078	337				
				0078	338				
				0078	339				
	51	50	D1	0078	340	7\$: Cmpl	R0,R1		
		03	15	007B	341	BLEQ	10\$		
	50	51	D0	007D	342	MOVL	R1,R0		
				0080	343				
				0080	344				
				0080	345				
				0080	346				
				0080	347				
	50	53	D1	0080	348				
		05	15	0083	349				
	53	50	D0	0085	350				
		31	15	0088	351				
				008A	352				
				008A	353				
				008A	354				
				008A	355				
				008A	356				
				008A	357				
50	34 A4	36 A4	A1	008A	358	20\$: ADDW3	PCB\$W_PPGCNT(R4),PCB\$W_GPGCNT(R4),R0		:PAGE COUNT CURRENTLY IN USE
		50	3C	0090	359	MOVZWL	R0,R0		:GET IT IN A LONGWORD
	51	52	C3	0093	360	SUBL3	R0,R2,R1		:NUMBER OF PAGES IMMEDIATELY RECLAIMABLE
		11	15	0097	361	BLEQ	40\$:BRANCH IF NONE
		53	D1	0099	362	Cmpl	R1,R3		:ARE WE GOING TO GET BACK TOO MANY?
		03	15	009C	363	BLEQ	30\$:BRANCH IF NOT
		51	D0	009E	364	MOVL	R3,R1		:TAKE BACK ONLY WHAT WAS ASKED
	50	A5	A2	00A1	365	30\$: SUBW	R1,PHD\$W_WSSIZE(R5)		:ADJUST WORKING SET DOWN BY EMPTIES
		53	C2	00A5	366	SUBL	R1,R3		:ADJUST COUNT OF PAGES TO STILL FREE
		11	15	00A8	367	BLEQ	60\$:BRANCH IF DONE
				00AA	368				
	04 AE	53	D0	00AA	369	40\$: MOVL	R3,4(SP)		:SAVE COUNT OF PAGES TO FREE

```

      FF4F' 30 00AE 370 50$:
07 50 E9 00AE 371
50 AS B7 00B1 372
F3 04 AE F5 00B4 373
52 8E D0 00B7 374
50 8E D0 00BB 375 60$:
      05 00C1 376
      RSB 377
BSBW MMGSFREWSLE
BLBC R0,60$
DECW PH0$W_WSSIZE(R5)
SOBGTR 4(SP),50$
MOVL (SP)+,R2
MOVL (SP)+,R0
RSB
;GET A FREE WORKING SET LIST ENTRY
;BRANCH TO RETURN WITH NON ZERO COUNT
;ACCOUNT FOR NEWLY FREED PAGE
;REPEAT FOR EACH SLOT TO BE DELETED
;RESTORE R2
;SET RETURN STATUS- NON-ZERO=FAILURE
;RETURN
```

```

00C2 379          .SBTTL EXTRADYNWS - CALCULATE EXTRA DYNAMIC WORKING SET COUNT
00C2 380
00C2 381
00C2 382          : FUNCTIONAL DESCRIPTION:
00C2 383          :
00C2 384          : THIS ROUTINE CALCULATES THE EXTRA NUMBER OF DYNAMIC WORKING SET
00C2 385          : ENTRIES AVAILABLE BEYOND THE NUMBER GUARANTEED BY WSFLUID
00C2 386          :
00C2 387          : CALLING SEQUENCE:
00C2 388          :
00C2 389          :     BSBW   MMGSEXTRADYNWS
00C2 390          :
00C2 391          : INPUTS:
00C2 392          :
00C2 393          :     R5 = PROCESS HEADER ADDRESS
00C2 394          :     IPL = AT LEAST ASTDEL
00C2 395          :
00C2 396          : OUTPUTS:
00C2 397          :
00C2 398          :     R1 = EXTRA DYNAMIC WORKING SET COUNT
00C2 399          :     R0,R2,R3 PRESERVED
00C2 400          :     PHDSW_EXTDYNWS = EXTRA DYNAMIC WORKING SET COUNT
00C2 401          :
00C2 402          :
00C2 403          : *****
00C2 404          : ***** THE FOLLOWING CODE MUST BE RESIDENT *****
00C2 405          : *****
00C2 406          :
000000C2 407          .PSECT $MMGCOD
00C2 408          :
00C2 409          : *****
00C2 410          :
00C2 411          :
00C2 412 MMGSEXTRADYNWS:
51 72 A5 6C A5 52 DD 00C2 413          PUSHL   R2
51 74 A5 51 B1 00C4 414          SUBW3   PHDSW_PTCNTLCK(R5),PHDSW_PTCNTMAX(R5),R1 ;COUNT OF PAGE TABLES
51 74 A5 04 B1 00CA 415          ;WHICH ARE NOT LOCKED DOWN
51 74 A5 3C 00D0 416          CMPW    R1,PHDSW_WSFLUID(R5) ;MINIMIZE WITH FLUID COUNT
51 74 A5 A0 00D4 417          BLEQU  10$ ;BRANCH IF SMALLER
51 6C A5 A0 00D8 418          MOVZWL PHDSW_WSFLUID(R5),R1 ;USE FLUID, IT IS SMALLER
51 51 3C 00DC 419 10$: ADDW   PHDSW_WSFLUID(R5),R1 ;ADD IN FLUID FOR DATA & I STREAM PAGES
51 51 3C 00DF 420          ADDW   PHDSW_PTCNTLCK(R5),R1 ;ADD IN LOCKED PAGE TABLE PAGES
51 51 3C 00DF 421          MOVZWL R1,R1 ;GET IT IN A LONGWORD
00DF 422          :
00DF 423          : We now have count of WSLE's that must be reserved in dynamic portion of WSL
00DF 424          :
52 18 A5 08 A5 A3 00DF 425          SUBW3   PHDSW_WSLIST(R5),PHDSW_WSQUOTA(R5),R2
52 50 A5 52 B6 00E5 426          INCW   R2 ;CALCULATE QUOTA
52 50 A5 52 B1 00E7 427          CMPW   R2,PHDSW_WSSIZE(R5) ;MINIMIZE WITH SIZE
52 50 A5 04 B1 00EB 428          BLEQU  20$
52 52 50 A5 B0 00ED 429          MOVW   PHDSW_WSSIZE(R5),R2
52 51 52 52 3C 00F1 430 20$: MOVZWL R2,R2 ;GET IT IN A LONGWORD
52 51 52 51 C3 00F4 431          SUBL3  R1,R2,R1 ;TAKE AWAY THE RESERVED ONES
00F8 432          :
00F8 433          : Remove non-dynamic portion of WSL from count
00F8 434          :
52 0E A5 08 A5 A3 00F8 435          SUBW3   PHDSW_WSLIST(R5),PHDSW_WSDYN(R5),R2 ;GET COUNT OF LOCKED WS ENTRIES

```

```

52 52 3C 00FE 436      MOVZWL R2,R2      :GET IT IN A LONGWORD
51 52 C2 0101 437      SUBL   R2,R1      :GET COUNT OF UNLOCKED ENTRIES
02 14 0104 438      BGTR   30$        :BRANCH IF POSITIVE
51 51 D4 0106 439      CLRL   R1         :DON'T ALLOW A NEGATIVE EXTRADYNWS COUNT
76 A5 51 B0 0108 440 30$: MOVW   R1,PHD$W_EXTRADYNWS(R5) :SAVE IT IN PROCESS HEADER
52 8E 05 010C 441      MOVL   (SP)+,R2
0110 442      RSB
0110 443
0110 444      .END
```

```

ACMSV_IMAGE = 00000001
CTL$GC_IWSPEAK ***** X 02
CTL$GL_PHD ***** X 02
CTL$GL_WSPEAK ***** X 02
EXESADJWSL 00000000 RG 02
EXESGL_ACMFLAGS ***** X 02
IPLS_ASTDEL = 00000002
IPLS_SYNCH = 00000008
MMGSALCPHD ***** X 02
MMGSEXTRADYNWS 000000C2 RG 03
MMGSFREWSLE ***** X 03
MMGSSHRINKWS 00000039 RG 03
MMGSWSPEAKCHK 000000DD RG 02
PAGCNT = 00000004
PCBSL_EFWM = 0000004C
PCBSW_GPGCNT = 00000034
PCBSW_PPGCNT = 00000036
PFNSGL_PHYPGCNT ***** X 02
PHDSM_IWSPEAKCK = 00000010
PHDSM_WSPEAKCHK = 00000004
PHDSW_EXTDYNWS = 00000076
PHDSW_FLAGS = 00000036
PHDSW_PTCNTLCK = 0000006C
PHDSW_PTCNTMAX = 00000072
PHDSW_WSDYN = 0000000E
PHDSW_WSEXTENT = 00000016
PHDSW_WSFLUID = 00000074
PHDSW_WSLAST = 00000012
PHDSW_WSLIST = 00000008
PHDSW_WSNEXT = 00000010
PHDSW_WSQUOTA = 00000018
PHDSW_WSSIZE = 00000050
PRS_IPL = 00000012
SCH$GL_FREELIM ***** X 02
SCH$GL_RESMASK ***** X 03
SCH$GL_MWAIT ***** X 03
SCH$WAIT ***** X 03
SGNSGL_MAXWSCNT ***** X 02
SGNSGW_MINWSCNT ***** X 03
SHRINK 00000000 R 03
SSS_ACCVIO = 0000000C
SSS_NORMAL = 00000001
WSETLM = 00000008
    
```

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YSEXEPAGED	00000108 (264.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$MMGCOD	00000110 (272.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance Indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.68
Command processing	105	00:00:00.53	00:00:03.35
Pass 1	300	00:00:08.86	00:00:29.99
Symbol table sort	0	00:00:01.45	00:00:03.05
Pass 2	93	00:00:01.89	00:00:05.55
Symbol table output	6	00:00:00.07	00:00:00.61
Psect synopsis output	1	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	536	00:00:12.93	00:00:43.27

The working set limit was 1350 pages.
51408 bytes (101 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 939 non-local and 25 local symbols.
444 source lines were read in Pass 1, producing 15 object records in Pass 2.
17 pages of virtual memory were used to define 16 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	7
-\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	6
TOTALS (all libraries)	13

1020 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

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

