

```

SSSSSSSSSSSS 00000000 RRRRRRRRRR TTTTTTTTTT 33333333 22222222
SSSSSSSSSSSS 00000000 RRRRRRRRRR TTTTTTTTTT 33333333 22222222
SSSSSSSSSSSS 00000000 RRRRRRRRRR TTTTTTTTTT 33333333 22222222
SSS          000      000  RRR      RRR  TTT      333      222
SSS          000      000  RRR      RRR  TTT      333      222
SSS          000      000  RRR      RRR  TTT      333      222
SSS          000      000  RRR      RRR  TTT      333      222
SSS          000      000  RRR      RRR  TTT      333      222
SSS          000      000  RRR      RRR  TTT      333      222
SSSSSSSSSS 000      000  RRRRRRRRRR TTT      333      222
SSSSSSSSSS 000      000  RRRRRRRRRR TTT      333      222
SSSSSSSSSS 000      000  RRRRRRRRRR TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSS          000      000  RRR  RRR  TTT      333      222
SSSSSSSSSS 00000000 RRR      RRR  TTT      33333333 22222222
SSSSSSSSSS 00000000 RRR      RRR  TTT      33333333 22222222
SSSSSSSSSS 00000000 RRR      RRR  TTT      33333333 22222222

```

```

SSSSSSSS 000000 RRRRRRRR CCCCCCCC 000000 MM MM PPPPPPPP AAAAAA RRRRRRRR
SSSSSSSS 000000 RRRRRRRR CCCCCCCC 000000 MM MM PPPPPPPP AAAAAA RRRRRRRR
SS 00 00 RR RR CC 00 00 MMMM MMMM PP PP AA AA RR RR
SS 00 00 RR RR CC 00 00 MMMM MMMM PP PP AA AA RR RR
SS 00 00 RR RR CC 00 00 MM MM MM PP PP AA AA RR RR
SSSSSS 00 00 RRRRRRRR CC 00 00 MM MM PPPPPPPP AA AA RRRRRRRR
SSSSSS 00 00 RRRRRRRR CC 00 00 MM MM PPPPPPPP AA AA RRRRRRRR
SS 00 00 RR RR CC 00 00 MM MM PP AAAAAAAAAA RR RR
SS 00 00 RR RR CC 00 00 MM MM PP AAAAAAAAAA RR RR
SS 00 00 RR RR CC 00 00 MM MM PP AA AA RR RR
SSSSSSSS 000000 RR RR CCCCCCCC 000000 MM MM PP AA AA RR RR
SSSSSSSS 000000 RR RR CCCCCCCC 000000 MM MM PP AA AA RR RR
.....
.....
.....
.....

LL 111111 SSSSSSSS
LL 111111 SSSSSSSS
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 LLLLLLLLLL 111111 SSSSSSSS
LL LLLLLLLLLL 111111 SSSSSSSS

```

SOR  
Pse

PSE  
---

Pha  
---  
Ini  
Com  
Pas  
Sym  
Pas  
Sym  
Pse  
Cro  
Ass

The  
116  
The  
387  
1 p

Mac  
---  
\_S2  
0 G  
The  
MAC

SORSCOMPARE  
Table of contents

Compare data fields

J 5

16-SEP-1984 01:19:01 VAX/VMS Macro V04-00

Page 0

\*\*F

(1) 3  
(1) 29

Copyright Notice  
Program description

```
0000 1 .TITLE SOR$COMPARE Compare data fields
0000 2 .IDENT 'V04-000' ; File: SOR$COMPARE.MAR Edit: PDG022
0000 3 .SBTTL Copyright Notice
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 29 .SBTTL Program description
0000 30 :++
0000 31 :
0000 32 : FACILITY:
0000 33 :
0000 34 : Compare data fields
0000 35 :
0000 36 : ABSTRACT:
0000 37 :
0000 38 : NONE
0000 39 :
0000 40 : ENVIRONMENT:
0000 41 :
0000 42 : Native mode, User mode, AST reentrant
0000 43 :
0000 44 : AUTHOR:
0000 45 :
0000 46 : Peter D Gilbert, September 1982
0000 47 :
0000 48 : MODIFIED BY:
0000 49 :
0000 50 : T03-016 PDG016 1-Sep-1982
0000 51 : Original
0000 52 : T03-017 Added SOR$$CONVERT. PDG 21-Dec-1982
0000 53 : T03-018 Handle collating sequences. PDG 22-Dec-1982
0000 54 : T03-019 Fetch (and decrement) the source length after the MOV3 for NR.
0000 55 : PDG 29-Dec-1982
0000 56 : T03-020 New interface for collating sequences. PDG 26-Jan-1983
0000 57 : T03-021 Add COM$B PAD. PDG 8-Feb-1983
0000 58 : T03-022 Modify interface for collating sequences. PDG 7-Mar-1983
0000 59 :--
```

```
0000 61 :+
0000 62 : Note that, due to the way registers R0-R3 are used as parameters in the
0000 63 : decimal instructions, LIB$FIXUP_DEC will be unable to repair the operands
0000 64 : in the case of a reserved operand fault.
0000 65 :
0000 66 : Note that lengths of passed parameters are given as DSC lengths
0000 67 : (i.e., number of bytes, except for packed decimal);
0000 68 : The hardware instructions work in digits.
0000 69 :-
0000 70
0000 71 .DSABL GLOBAL
0000 72
0000 73 .EXTRN COM$L_COLLATE ; Offset from R11 of COM_COLLATE
0000 74 .EXTRN COM$B_PAD ; Offset from R11 of COM_PAD
0000 75
0000 76 .EXTRN SORS$ BAD KEY
0000 77 .EXTRN LIB$AB_CVTTP_0
0000 78 .EXTRN LIB$AB_CVTTP_U
0000 79 .EXTRN LIB$AB_CVTTP_Z
0000 80 .EXTRN LIB$AB_CVTPT_0
0000 81 .EXTRN LIB$AB_CVTPT_U
0000 82 .EXTRN LIB$AB_CVTPT_Z
0000 83 .EXTRN LIB$AB_CVT_U_0
0000 84
00000000 85 .PSECT SORS$RO_CODE NOVEC,NOWRT,RD,EXE,SHR,LCL,REL,CON,PIC,LONG
```

		0000	87		
		0000	88	; Offsets from AP of the parameters	
		0000	89	:	
		00000004	0000	90 DTYPE = 4	; Datatype of keys
		00000008	0000	91 FLD1 = 8	; Address of length/address of first field
		0000000C	0000	92 FLD2 = 12	; Address of length/address of second field
		0000	93		
		003C	0000	94 .ENTRY SOR\$\$COMPARE, ^M<R2,R3,R4,R5>	
			0002	95	
			0002	96	; This routine compares two fields
			0002	97	
			0002	98	; Fetch the descriptions into registers
			0002	99	
50	08 BC	7D	0002	100	MOVQ @FLD1(AP), R0
52	0C BC	7D	0006	101	MOVQ @FLD2(AP), R2
			000A	102	
			000A	103	; Case on the datatype
			000A	104	
25'8F	00	04 AC	8F	000A	105 CASEB DTYPE(AP), #0, #<1\$-0\$-2>/2
		0061'	0010	106 0\$:	.WORD Z - 0\$ 0
		004C'	0012	107	.WORD V - 0\$ 1
		006B'	0014	108	.WORD BU - 0\$ 2
		0070'	0016	109	.WORD WU - 0\$ 3
		00A1'	0018	110	.WORD LU - 0\$ 4
		008C'	001A	111	.WORD QU - 0\$ 5
		00B3'	001C	112	.WORD B - 0\$ 6
		00B8'	001E	113	.WORD W - 0\$ 7
		00BD'	0020	114	.WORD L - 0\$ 8
		009A'	0022	115	.WORD R - 0\$ 9
		00C2'	0024	116	.WORD F - 0\$ 10
		00C7'	0026	117	.WORD D - 0\$ 11
		004C'	0028	118	.WORD FC - 0\$ 12
		004C'	002A	119	.WORD DC - 0\$ 13
		0054'	002C	120	.WORD T - 0\$ 14
		0171'	002E	121	.WORD NU - 0\$ 15
		018F'	0030	122	.WORD NL - 0\$ 16
		00DF'	0032	123	.WORD NLO - 0\$ 17
		0132'	0034	124	.WORD NR - 0\$ 18
		01A7'	0036	125	.WORD NRO - 0\$ 19
		01C5'	0038	126	.WORD NZ - 0\$ 20
		00DB'	003A	127	.WORD P - 0\$ 21
		004C'	003C	128	.WORD ZI - 0\$ 22
		004C'	003E	129	.WORD ZEM - 0\$ 23
		004C'	0040	130	.WORD DSC - 0\$ 24
		0075'	0042	131	.WORD OU - 0\$ 25
		007E'	0044	132	.WORD O - 0\$ 26
		00CC'	0046	133	.WORD G - 0\$ 27
		00D2'	0048	134	.WORD H - 0\$ 28
		004C'	004A	135	.WORD GC - 0\$ 29
		004C'	004C	136	.WORD HC - 0\$ 30
		004C'	004E	137	.WORD CIT - 0\$ 31
		004C'	0050	138	.WORD BPV - 0\$ 32
		004C'	0052	139	.WORD BLV - 0\$ 33
		004C'	0054	140	.WORD 1\$ - 0\$ 34
		009A'	0056	141	.WORD ADT - 0\$ 35
		004C'	0058	142	.WORD 1\$ - 0\$ 36
		004C'	005A	143	.WORD VT - 0\$ 37

```

005C 144 1S: V: FC: DC: ZI: ZEM: DSC: GC: HC: CIT: BPV: BLV: VT:
50 00000000'8F D0 005C 145 MOVL #SORS_BAD_KEY, R0
04 0063 146 RET
55 00000000'EB D0 0064 147 T: MOVL COM$L_COLLATE(R11), R5
04 13 0068 148 BEQL Z
00 B5 16 006D 149 JSB @0(R5)
04 0070 150 RET
63 52 0000'CB 61 50 2D 0071 151 Z: CMPCS R0, (R1), W^COM$B_PAD(R11), R2, (R3)
39 11 0079 152 BRB CMPL
63 61 91 007B 153 BU: CMPB (R1), (R3)
34 11 007E 154 BRB CMPL
63 61 B1 0080 155 WU: CMPW (R1), (R3)
2F 11 0083 156 BRB CMPL
OC A3 OC A1 D1 0085 157 OU: CMPL 12(R1), 12(R3)
28 12 008A 158 BNEQ CMPL
07 11 008C 159 BRB 0 1
OC A3 OC A1 D1 008E 160 O: CMPL 12(R1), 12(R3)
0E 12 0093 161 BNEQ CMPS
08 A3 08 A1 D1 0095 162 O_1: CMPL 8(R1), 8(R3)
18 12 009A 163 BNEQ CMPL
04 A3 04 A1 D1 009C 164 QU: CMPL 4(R1), 4(R3)
0E 13 00A1 165 BEQL LU
16 19 00A3 166 CMPS: BLSS M1
18 14 00A5 167 BGTR P1
50 D4 00A7 168 CLRL R0
04 00A9 169 RET
04 A3 04 A1 D1 00AA 170 ADT: Q: CMPL 4(R1), 4(R3) ; Note that ADT is considered to be signed
F2 12 00AF 171 BNEQ CMPS
63 61 D1 00B1 172 LU: CMPL (R1), (R3)
05 1F 00B4 173 CMPL: BLSSU M1
07 1A 00B6 174 BGTRU P1
50 D4 00B8 175 CLRL R0
04 00BA 176 RET
50 01 CE 00BB 177 M1: MNEGL #1, R0
04 00BE 178 RET
50 01 D0 00BF 179 P1: MOVL #1, R0
04 00C2 180 RET
63 61 91 00C3 181 B: CMPB (R1), (R3)
DB 11 00C6 182 BRB CMPS
63 61 B1 00C8 183 W: CMPW (R1), (R3)
D6 11 00CB 184 BRB CMPS
63 61 D1 00CD 185 L: CMPL (R1), (R3)
D1 11 00D0 186 BRB CMPS
63 61 51 00D2 187 F: CMPF (R1), (R3)
CC 11 00D5 188 BRB CMPS
63 61 71 00D7 189 D: CMPD (R1), (R3)
C7 11 00DA 190 BRB CMPS
63 61 51FD 00DC 191 G: CMPL (R1), (R3)
C1 11 00E0 192 BRB CMPS
63 61 71FD 00E2 193 H: CMPL (R1), (R3)
BB 11 00E6 194 BRB CMPS
63 52 61 50 37 00E8 195 P: CMPP4 R0, (R1), R2, (R3)
B4 11 00ED 196 BRB CMPS
SE CO AE 9E 00EF 197 NLO: MOVAB -64(SP), SP ; Make room on the stack
52 61 9A 00F3 198 MOVZBL (R1), R2 ; Fetch the overpunched byte
7E 00000000'GF42 90 00F6 199 MOVB G^LIB$AB CVTTP_0[R2], -(SP) ; Translate to packed
7E 01 90 00FE 200 MOVB #1, -(SP) ; Make packed number non-zero

```



	03 AE	61	50	28	0101	201	MOV C3	RO, (R1), 3(SP)	: Move the string
	02 AE	02	6E	08	0106	202	CVT PS	#2, (SP), #2, 2(SP)	: Convert packed to separate
			5E	C0	010C	203	ADD L2	#2, SP	: Pop the packed number
20	AE	1F	6E	09	010F	204	CVT SP	@FLD1(AP), (SP), #31, 32+00(SP)	: Convert separate to packed
			50	7D	0116	205	MOV Q	@FLD2(AP), RO	
			52	9A	011A	206	MOV ZBL	(R1), R2	: Fetch the overpunched byte
	7E	000000	'GF	90	011D	207	MOV B	G^LIB\$AB CVTTP_0[R2], -(SP)	: Translate to packed
			7E	90	0125	208	MOV B	#1, -(SP)	: Make packed number non-zero
	03 AE	61	50	28	0128	209	MOV C3	RO, (R1), 3(SP)	: Move the string
	02 AE	02	6E	08	012D	210	CVT PS	#2, (SP), #2, 2(SP)	: Convert packed to separate
			5E	C0	0133	211	ADD L2	#2, SP	: Pop the packed number
30	AE	1F	6E	09	0136	212	CVT SP	@FLD2(AP), (SP), #31, 32+16(SP)	: Convert separate to packed
			5E	C0	013D	213	ADD L2	#32, SP	
				11	0140	214	BRB	CMPP31	
			5E	9E	0142	215	MOV AB	-64(SP), SP	
			50	D7	0146	216	DECL	RO	
			6E	90	0148	217	MOV B	(R1)[RO], (SP)	
	01 AE	61	50	28	014C	218	MOV C3	RO, (R1), 1(SP)	
	50	08	BC	A3	0151	219	SUB W3	#1, @FLD1(AP), RO	
20	AE	1F	6E	09	0156	220	CVT SP	RO, (SP), #31, 32+00(SP)	
			50	7D	015C	221	MOV Q	@FLD2(AP), RO	
				D7	0160	222	DECL	RO	
			6E	90	0162	223	MOV B	(R1)[RO], (SP)	
	01 AE	61	50	28	0166	224	MOV C3	RO, (R1), 1(SP)	
	50	08	BC	A3	016B	225	SUB W3	#1, @FLD1(AP), RO	
30	AE	1F	6E	09	0170	226	CVT SP	RO, (SP), #31, 32+16(SP)	
			5E	C0	0176	227	ADD L2	#32, SP	
					0179	228	BRB	CMPP31	
	10	AE	6E	35	0179	229	CMPP31:	CMPP3	#31, (SP), 16(SP)
			FF	31	017E	230	BRW	CMPS	
1F	00000000	'GF	5E	C2	0181	231	NU:	SUB L2	#32, SP
			61	26	0184	232	CVT TP	RO, (R1), G^LIB\$AB_CVTTP_U, #31, 00(SP)	
			6E		018D				
1F	00000000	'GF	50	7D	018E	233	MOV Q	@FLD2(AP), RO	
			61	26	0192	234	CVT TP	RO, (R1), G^LIB\$AB_CVTTP_U, #31, 16(SP)	
			10		019B				
			5E	11	019D	235	BRB	CMPP31	
			50	C2	019F	236	NL:	SUB L2	#32, SP
				D7	01A2	237	DECL	RO	
	6E	1F	61	09	01A4	238	CVT SP	RO, (R1), #31, 00(SP)	
			50	7D	01A9	239	MOV Q	@FLD2(AP), RO	
				D7	01AD	240	DECL	RO	
	10	AE	1F	09	01AF	241	CVT SP	RO, (R1), #31, 16(SP)	
				11	01B5	242	BRB	CMPP31	
1F	00000000	'GF	5E	C2	01B7	243	NRO:	SUB L2	#32, SP
			61	26	01BA	244	CVT TP	RO, (R1), G^LIB\$AB_CVTTP_0, #31, 00(SP)	
			6E		01C3				
1F	00000000	'GF	50	7D	01C4	245	MOV Q	@FLD2(AP), RO	
			61	26	01C8	246	CVT TP	RO, (R1), G^LIB\$AB_CVTTP_0, #31, 16(SP)	
			10		01D1				
			5E	11	01D3	247	BRB	CMPP31	
1F	00000000	'GF	61	C2	01D5	248	NZ:	SUB L2	#32, SP
			6E	26	01D8	249	CVT TP	RO, (R1), G^LIB\$AB_CVTTP_Z, #31, 00(SP)	
					01E1				
1F	00000000	'GF	50	7D	01E2	250	MOV Q	@FLD2(AP), RO	
			61	26	01E6	251	CVT TP	RO, (R1), G^LIB\$AB_CVTTP_Z, #31, 16(SP)	
			10		01EF				

86 11 01F1 252 BRB CMPP31

				01F3	254	
				01F3	255	; Offsets from AP of the parameters
				01F3	256	:
		00000004		01F3	257	DTYPE = 4 ; Datatype of keys
		00000008		01F3	258	FLD1 = 8 ; Address of length/address of constant
		0000000C		01F3	259	FLD2 = 12 ; Address of length/address of field
				01F3	260	
		007C		01F3	261	.ENTRY SOR\$\$CONVERT, ^M<R2,R3,R4,R5,R6>
				01F5	262	
				01F5	263	: This routine converts a decimal string constant to an appropriate
				01F5	264	: datatype.
				01F5	265	:
	50	08 BC	7D	01F5	266	MOVQ @FLD1(AP), R0
		5E 10	C2	01F9	267	SUBL2 #16, SP
				01FC	268	
				01FC	269	: First, convert the number to packed
1F	00000000'GF			01FC	270	
		61	50	01FC	271	CVTTP R0, (R1), G^LIB\$AB_CVTTP_U, #31, (SP)
		6E		0205	272	
				0206	273	:
				0206	274	: Now case on the destination datatype
				0206	275	
	25'8F	52 0C BC	7D	020A	276	MOVQ @FLD2(AP), R2
		00 04 AC	8F	020A	276	CASEB DTYPE(AP), #0, #<1\$-0\$-2>/2
				005A'	277	.WORD vZ - 0\$ : 0
				004C'	278	.WORD vV - 0\$ : 1
				005A'	279	.WORD vBU - 0\$ : 2
				005A'	280	.WORD vWU - 0\$ : 3
				005A'	281	.WORD vLU - 0\$ : 4
				005A'	282	.WORD vQU - 0\$ : 5
				005A'	283	.WORD vB - 0\$ : 6
				005A'	284	.WORD vW - 0\$ : 7
				005A'	285	.WORD vL - 0\$ : 8
				005A'	286	.WORD vQ - 0\$ : 9
				0119'	287	.WORD vF - 0\$ : 10
				014F'	288	.WORD vD - 0\$ : 11
				004C'	289	.WORD vFC - 0\$ : 12
				004C'	290	.WORD vDC - 0\$ : 13
				005A'	291	.WORD vVT - 0\$ : 14
				00AA'	292	.WORD vNU - 0\$ : 15
				0086'	293	.WORD vNL - 0\$ : 16
				00BF'	294	.WORD vNLO - 0\$ : 17
				00DE'	295	.WORD vNR - 0\$ : 18
				00F6'	296	.WORD vNRO - 0\$ : 19
				0102'	297	.WORD vNZ - 0\$ : 20
				010E'	298	.WORD vP - 0\$ : 21
				004C'	299	.WORD vZI - 0\$ : 22
				004C'	300	.WORD vZEM - 0\$ : 23
				004C'	301	.WORD vDSC - 0\$ : 24
				005A'	302	.WORD vOU - 0\$ : 25
				005A'	303	.WORD vO - 0\$ : 26
				0189'	304	.WORD vG - 0\$ : 27
				01C7'	305	.WORD vH - 0\$ : 28
				004C'	306	.WORD vGC - 0\$ : 29
				004C'	307	.WORD vHC - 0\$ : 30
				004C'	308	.WORD vCIT - 0\$ : 31
				004C'	309	.WORD vBPV - 0\$ : 32

: R

					004C'	0252	310		.WORD	vBLV	- 0\$	:	33
					004C'	0254	311		.WORD	1\$	- 0\$	:	34
					005A'	0256	312		.WORD	vADT	- 0\$	:	35
					004C'	0258	313		.WORD	1\$	- 0\$	:	36
					004C'	025A	314		.WORD	vvVT	- 0\$	:	37
						025C	315	1\$: vv:	vFC: vDC: vZ1: vZEM: vDSC: vGC: vHC: vCIT: vBPV: vBLV: vvVT:				
						025C	316		MOVL	#SORS_BAD_KEY, R0			
						0263	317		RET				
						0264	318	TWO_32:	.PACKED	4294967296		:	2**32
						026A	319	vZ: vvT:	vBU: vWU: vLU: vQU: vB: vW: vL: vQ: vOU: vO: vADT:				
						026A	320		SUBL2	#32, SP			
						026D	321		CLRL	R6			
						026F	322		BRB	20\$			
						0271	323	10\$:	MOVL	FLD2(AP), R2			
						0275	324		ADDL3	4(R2), R6, R2			
						027A	325		MOVL	R0, -4(R2)			
						027E	326	20\$:	DIVP	#10, TWO_32, #31, 32(SP), #31, 16(SP)			
						0286							
						0288	327		MULP	#10, TWO_32, #31, 16(SP), #31, 00(SP)			
						0290							
						0291	328		SUBP4	#31, 00(SP), #31, 32(SP)			
						0297	329		CVTPL	#31, 32(SP), R0			
						029C	330		ACBW	@FLD2(AP), #4, R6, 10\$			
						02A3	331		MOVQ	@FLD2(AP), R2			
						02A7	332		SUBW2	R6, R2			
						02AA	333		ADDL2	#4, R2			
						02AD	334		ASHL	#3, R2, R2			
						02B1	335		INSV	R0, #0, R2, -4(R3)[R6]			
						02B8	336		BRB	JOIN			
						02BA	337	vNU:	CVTPT	#31, (SP), G^LIB\$AB_CVTPT_U, R2, (R3)			
						02C3							
						02C4	338		BRB	JOIN			
						02C6	339	vNL:	DECL	R2			
						02C8	340		CVTPT	#31, (SP), R2, (R3)			
						02CD	341		BRB	JOIN			
						02CF	342	vNLO:	MOVAB	G^LIB\$AB_CVT_U_0-^A/0/, R4			
						02D6	343		CVTPT	#31, (SP), G^LIB\$AB_CVTPT_U, R2, (R3)			
						02DF							
						02E0	344		BGEQ	1\$			
						02E2	345		ADDL2	#10, R4			
						02E5	346	1\$:	MOVZBL	(R3), R0			
						02E8	347		MOVB	(R4)[R0], (R3)			
						02EC	348		BRB	JOIN			
						02EE	349	vNR:	DECL	R2			
						02F0	350		CVTPT	#31, (SP), R2, (R3)			
						02F5	351		MOVQ	@FLD2(AP), R2			
						02F9	352		SUBL3	#1, R2, R0			
						02FD	353		MOVCS	R0, 1(R3), (R3), R2, (R3)			
						0304	354		BRB	JOIN			
						0306	355	vNRO:	CVTPT	#31, (SP), G^LIB\$AB_CVTPT_0, R2, (R3)			
						030F							
						0310	356		BRB	JOIN			
						0312	357	vNZ:	CVTPT	#31, (SP), G^LIB\$AB_CVTPT_Z, R2, (R3)			
						031B							
						031C	358		BRB	JOIN			
						031E	359	vP:	ASHP	#0, #31, (SP), #0, R2, (R3)			
						0325	360		BRB	JOIN			

```

50 01 D0 0325 361 JOIN:  MOVL  #1, R0
      04 0328 362          RET
      0329 363          .MACRO CVTPF  X
      0329 364          CLR   -(SP)
      0329 365          CLR'X (R4)
      0329 366          MOVL  #3, R6
      0329 367          MOVB  4(SP), -(SP)
      0329 368 10$:    CVTPL  #9, 1(SP), 1(SP)
      0329 369          MOVB  (SP)+, 4(SP)
      0329 370          CVTL'X (SP)+, R0
      0329 371          MUL'X'2 #100000000, (R4)          ; Multiply by 10**8
      0329 372          ADD'X'2 R0, (R4)
      0329 373          SOBGEQ R6, 10$
      0329 374          EXTZV #0, #8, (SP), R0
      0329 375          BBC   R0, 1^#^B101000, 20$
      0329 376          MNEG'X (R4), (R4)
      0329 377          MOVL  #1, R0
      0329 378 20$:    RET
      0329 379          .ENDM
      0329 380
      0329 381
      0329 382 vF:    CVTPF  F
      035F 383 vD:    CVTPF  D
      0399 384 vG:    CVTPF  G
      03D7 385 vH:    CVTPF  H
      041E 386
      041E 387          .END

```

ADT	000000AA	R	01	VBU	0000026A	R	01
B	000000C3	R	01	VCIT	0000025C	R	01
BLV	0000005C	R	01	VD	0000035F	R	01
BPV	0000005C	R	01	VDC	0000025C	R	01
BU	0000007B	R	01	VDSC	0000025C	R	01
CIT	0000005C	R	01	VF	00000329	R	01
CMPP31	00000179	R	01	VFC	0000025C	R	01
CMPS	000000A3	R	01	VG	00000399	R	01
CMPU	000000B4	R	01	VGC	0000025C	R	01
COM\$B_PAD	*****	X	00	VH	000003D7	R	01
COM\$SL_COLLATE	*****	X	00	VHC	0000025C	R	01
D	000000D7	R	01	VL	0000026A	R	01
DC	0000005C	R	01	VLU	0000026A	R	01
DSC	0000005C	R	01	VNL	000002C6	R	01
DTYPE	= 00000004			VNLO	000002CF	R	01
F	000000D2	R	01	VNR	000002EE	R	01
FC	0000005C	R	01	VNRO	00000306	R	01
FLD1	= 00000008			VNU	000002BA	R	01
FLD2	= 0000000C			VNZ	00000312	R	01
G	000000DC	R	01	VO	0000026A	R	01
GC	0000005C	R	01	VOU	0000026A	R	01
H	000000E2	R	01	VP	0000031E	R	01
HC	0000005C	R	01	VQ	0000026A	R	01
JOIN	00000325	R	01	VQU	0000026A	R	01
L	000000CD	R	01	VT	0000005C	R	01
LIB\$AB_CVTPT_O	*****	X	00	VV	0000025C	R	01
LIB\$AB_CVTPT_U	*****	X	00	VVT	0000026A	R	01
LIB\$AB_CVTPT_Z	*****	X	00	VVVT	0000025C	R	01
LIB\$AB_CVTTP_O	*****	X	00	VW	0000026A	R	01
LIB\$AB_CVTTP_U	*****	X	00	VWU	0000026A	R	01
LIB\$AB_CVTTP_Z	*****	X	00	VZ	0000026A	R	01
LIB\$AB_CVT_U_O	*****	X	00	VZEM	0000025C	R	01
LU	000000B1	R	01	VZI	0000025C	R	01
M1	000000BB	R	01	W	000000C8	R	01
NL	0000019F	R	01	WU	00000080	R	01
NLO	000000EF	R	01	Z	00000071	R	01
NR	00000142	R	01	ZEM	0000005C	R	01
NRO	000001B7	R	01	ZI	0000005C	R	01
NU	00000181	R	01				
NZ	000001D5	R	01				
O	0000008E	R	01				
OU	00000085	R	01				
O_1	00000095	R	01				
P_	000000E8	R	01				
P1	000000BF	R	01				
Q	000000AA	R	01				
QU	0000009C	R	01				
SOR\$\$\$COMPARE	00000000	RG	01				
SOR\$\$\$CONVERT	000001F3	RG	01				
SOR\$_BAD_KEY	*****	X	00				
T	00000064	R	01				
TWO_32	00000264	R	01				
V	0000005C	R	01				
VADT	0000026A	R	01				
VB	0000026A	R	01				
VBLV	0000025C	R	01				
VBPV	0000025C	R	01				

.....

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

PSECT name	Allocation	PSECT No.	Attributes											
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE	
SORSRO_CODE	0000041E ( 1054.)	01 ( 1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG	

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.04	00:00:01.66
Command processing	118	00:00:00.42	00:00:04.29
Pass 1	106	00:00:01.70	00:00:06.60
Symbol table sort	0	00:00:00.09	00:00:00.09
Pass 2	80	00:00:00.82	00:00:03.02
Symbol table output	9	00:00:00.08	00:00:00.08
Psect synopsis output	3	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	349	00:00:03.17	00:00:15.76

The working set limit was 1050 pages.  
11603 bytes (23 pages) of virtual memory were used to buffer the intermediate code.  
There were 10 pages of symbol table space allocated to hold 95 non-local and 15 local symbols.  
387 source lines were read in Pass 1, producing 17 object records in Pass 2.  
1 page of virtual memory was used to define 1 macro.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/DISABLE=TRACE/LIS=LISS:SORSCOMPARE/OBJ=OBJ\$:SORSCOMPARE MSRC\$:SORSCOMPARE/UPDATE=(ENH\$:SORSCOMPARE)

