

(3) 85

STRSMATCH_WILD, general wild card matching

```

0000 1      .TITLE STRSMATCH_WILD      Match General Wild Card Specification
0000 2      .IDENT 'V03-002'          ; File: STRSMATCH.MAR Edit:LEB3002
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
0000 30 : FACILITY: General Utility Library
0000 31
0000 32 : ABSTRACT:
0000 33
0000 34 :   This routine performs the general embedded wild card matching
0000 35 :   algorithm.
0000 36
0000 37 : ENVIRONMENT:
0000 38
0000 39 :   Runs at any access mode, AST Reentrant
0000 40
0000 41 : AUTHOR: Andrew C. Goldstein, CREATION DATE: 10-Aug-1979 11:36
0000 42
0000 43 : MODIFIED BY:
0000 44
0000 45 :   V03-002 LEB          Linda Benson 15-Dec-1983
0000 46 :   Change name from STRSMATCH_NAME to STRSMATCH_WILD
0000 47 :   to more correctly match intent of this routine.
0000 48 :   This marks version that has been incorporated into
0000 49 :   the RTL. Add EDIT field to module.
0000 50
0000 51 :   V03-001 BLS0178     Benn Schreiber 13-Mar-1982
0000 52 :   Add interface to call as str$match_name
0000 53
0000 54 :   V02-001 MLJ0031     Martin L. Jack, 4-Aug-1981 6:32
0000 55 :   Reorganize for simplicity and speed.
0000 56
0000 57 :**

```



```
0000 59 :  
0000 60 : EXTERNAL DECLARATIONS:  
0000 61 :  
0000 62 : Prevent undeclared symbols from being automatically global.  
0000 63 :  
0000 64 :     .DISABLE GLOBAL  
0000 65 :     .EXTERNAL STR$ANALYZE_SDESC R1  
0000 66 :     .EXTERNAL STR$_MATCH,STR$_NOMATCH  
0000 67 :  
0000 68 : MACROS:  
0000 69 :  
0000 70 :     NONE  
0000 71 :  
0000 72 : EQUATED SYMBOLS:  
0000 73 :  
0000 74 :     NONE  
0000 75 :  
0000 76 : OWN STORAGE:  
0000 77 :  
0000 78 :     NONE  
0000 79 :  
0000 80 : PSECT DECLARATIONS:  
0000 81 :  
00000000 82 :     .PSECT _STR$CODE PIC,USR,CON,REL,LCL,SHR,-  
0000 83 :     EXE, RD, NOWRT, LONG
```

```

0000 85 .SBTTL STR$MATCH_WILD, general wild card matching
0000 86 :++
0000 87 : Functional Description:
0000 88 : This routine performs the general embedded wild card matching
0000 89 : algorithm.
0000 90 :
0000 91 : Calling Sequence:
0000 92 : ret_status.wlc.v = STR$MATCH_WILD (CAND.rt.dx,PATRN.rt.dx)
0000 93 :
0000 94 : Formal Parameters:
0000 95 : CAND.rt.dx Address of string descriptor for candidate string
0000 96 : (The current item being looked at)
0000 97 : PATRN.rt.dx Address of string descriptor for pattern string
0000 98 : (The item looking for)
0000 99 :
0000 100 : Implicit Inputs:
0000 101 : none
0000 102 :
0000 103 : Output Parameters:
0000 104 : none
0000 105 :
0000 106 : Implicit Outputs:
0000 107 : none
0000 108 :
0000 109 : Routines Called:
0000 110 : STR$ANALYZE_SDESC_R1
0000 111 :
0000 112 : Routine Value:
0000 113 : STR$_MATCH if the strings match.
0000 114 : STR$_NOMATCH if the strings don't match
0000 115 :
0000 116 : Signals:
0000 117 : Errors from STR$ANALYZE_SDESC
0000 118 :
0000 119 : Side Effects:
0000 120 : none
0000 121 :
0000 122 :--
0000 123 :
03FC 0000 124 .ENTRY str$match_wild,^M<R2,R3,R4,R5,R6,R7,R8,R9>
0002 125 :
50 04 AC D0 0002 126 MOVL 4(AP),R0 ; get first descriptor address
00000000'GF 16 0006 127 JSB G^STR$ANALYZE_SDESC_R1 ; extract string length and address
7E 50 7D 000C 128 MOVQ R0,-(SP) ; save descriptor
50 08 AC D0 000F 129 MOVL 8(AP),R0 ; get second descriptor address
00000000'GF 16 0013 130 JSB G^STR$ANALYZE_SDESC_R1 ; analyze second descriptor
54 50 7D 0019 131 MOVQ R0,R4 ; set up for match algorithm
52 8E 7D 001C 132 MOVQ (SP)+,R2 ; retrieve first descriptor
50 00000000'8F D0 001F 133 MOVL #STR$_NOMATCH,R0 ; Assume failure
2A 51 91 0026 134 CLRL R6 ; Clear saved candidate count
0028 135 :
0028 136 : Main scanning loop.
0028 137 :
54 D7 0028 138 10$: DECL R4 ; Pattern exhausted?
24 19 002A 139 BLSS 30$ ; Branch if yes
51 85 9A 002C 140 MOVZBL (R5)+,R1 ; Get next character in pattern
2A 51 91 002F 141 CMPB R1,#^A'^* ; Pattern specifies wild string?

```



```
28 13 0032 142 BEQL 60$ : Branch if yes
52 D7 0034 143 DECL R2 : Candidate exhausted?
23 19 0036 144 BLSS 50$ : Branch if yes
83 51 91 0038 145 CMPB R1,(R3)+ : Compare pattern to candidate
EB 13 003B 146 BEQL 10$ : Branch if pattern equals candidate
25 51 91 003D 147 CMPB R1,#^A'%' : Pattern specifies wild character?
E6 13 0040 148 BEQL 10$ : Branch if yes
0042 149 :
0042 150 : We have detected a mismatch, or we are out of pattern while there is
0042 151 : candidate left. Back up to the last '*', advance a candidate character,
0042 152 : and try again.
0042 153 :
56 D7 0042 154 20$: DECL R6 : Count a saved candidate character
15 19 0044 155 BLSS 50$ : Branch if no saved candidate
57 D6 0046 156 INCL R7 : Set to try next character
52 56 7D 0048 157 MOVQ R6,R2 : Restore descriptors to backup point
54 58 7D 004B 158 MOVQ R8,R4 :
D8 11 004E 159 BRB 10$ : Continue testing
0050 160 :
0050 161 : Here when pattern is exhausted.
0050 162 :
52 D5 0050 163 30$: TSTL R2 : Candidate exhausted?
EE 12 0052 164 BNEQ 20$ : Branch if no
0054 165 :
0054 166 : Here to return.
0054 167 :
50 00000000'8F D0 0054 168 40$: MOVL #STR$_MATCH,R0 : Set success return
04 005B 169 50$: RET : Return
005C 170 :
005C 171 : We have detected a '*' in the pattern. Save the pointers for backtracking.
005C 172 :
54 D5 005C 173 60$: TSTL R4 : Pattern null after '*'?
F4 13 005E 174 BEQL 40$ : Branch if yes
56 52 7D 0060 175 MOVQ R2,R6 : Save descriptors of both strings
58 54 7D 0063 176 MOVQ R4,R8 :
CO 11 0066 177 BRB 10$ : Continue testing
0068 178 :
0068 179 .END
```

STRSMATCH WILD
Symbol table

Match General Wild Card Specification L 13

16-SEP-1984 00:35:08
6-SEP-1984 11:18:02

VAX/VMS Macro V04-00
[LIBRTL.SRC]STRMATCH.MAR;1

Page 5
(3)

STR\$ANALYZE_SDESC_R1
STRSMATCH WILD
STR\$ MATCH
STR\$ NOMATCH

***** X 00
00000000 RG 01
***** X 00
***** X 00

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

PSECT name

Allocation

PSECT No.

Attributes

ABS
STR\$CODE

00000000 (0.)
00000068 (104.)

00 (0.)
01 (1.)

NOPIC USR
PIC USR

CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
CON REL LCL SHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.02	00:00:02.84
Command processing	116	00:00:00.30	00:00:01.88
Pass 1	71	00:00:00.31	00:00:02.72
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	46	00:00:00.22	00:00:02.54
Symbol table output	2	00:00:00.01	00:00:00.01
Psect synopsis output	2	00:00:00.01	00:00:00.50
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	269	00:00:00.88	00:00:10.50

The working set limit was 900 pages.
2203 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 4 non-local and 6 local symbols.
179 source lines were read in Pass 1, producing 11 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

+-----+
! 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/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LISS:STRMATCH/OBJ=OBJ\$:STRMATCH MSRC\$:STRMATCH/UPDATE=(ENH\$:STRMATCH)

