


```

FFFFFFFFF  SSSSSSSS  PPPPPPPP  AAAAAA  CCCCCCCC  KK      KK
FFFFFFFFF  SSSSSSSS  PPPPPPPP  AAAAAA  CCCCCCCC  KK      KK
FF         SS        PP      PP  AA      AA  CC      KK      KK
FF         SS        PP      PP  AA      AA  CC      KK      KK
FF         SS        PP      PP  AA      AA  CC      KK      KK
FF         SS        PP      PP  AA      AA  CC      KK      KK
FFFFFFFFF  SSSSSS   PPPPPPPP  AA      AA  CC      KKKKKK
FFFFFFFFF  SSSSSS   PPPPPPPP  AA      AA  CC      KKKKKK
FF         SS      PP      AA      AA  CC      KK      KK
FF         SS      PP      AA      AA  CC      KK      KK
FF         SS      PP      AA      AA  CC      KK      KK
FF         SS      PP      AA      AA  CC      KK      KK
FF         SSSSSSS  PP      AA      AA  CCCCCCCC  KK      KK
FF         SSSSSSS  PP      AA      AA  CCCCCCCC  KK      KK

```

```

RRRRRRR  EEEEEEEEE  QQQQQQ
RRRRRRR  EEEEEEEEE  QQQQQQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RRRRRRR  EEEEEEEEE  QQ      QQ
RRRRRRR  EEEEEEEEE  QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EEEEEEEEE  QQQQ  QQ
RR      RR  EEEEEEEEE  QQQQ  QQ

```

V

+

+

+

+

+

+

+

+

+

+

+

+

+

+

F

AE

E

AL

CF

M

LITI

LITI

Version: 'V04-000'

```

*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

```

```

**
FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
ABSTRACT:
  Macros to manipulate FIXED_STRINGS: transportable, fixed-length
  strings.
ENVIRONMENT:   Transportable BLISS
AUTHOR:       Rich Friday
CREATION DATE: 1978
MODIFIED BY:
  002 KAD00002 Keith Dawson 07-Mar-1983
      Global edit of all modules. Updated module names, idents,
      copyright dates. Changed require files to BLISS library.

```

```

*
FIXED STRING is a structure that contains a string of an expected maximum
length, plus additional locations to contain information related to the
status of the string.

```

```

STRUCTURE
  fixed_string[i:n=1] =
    [4*%UPVAL + (HS$ALLOCATION(N*%UPVAL))]

```

```
(fixed_string - %UPVAL*( 1 - I) );
```

```
!+
! The following literals are names for the fields allocated above. These
! names are not to be used by a user.
```

```
! LITERAL
```

```
FS__FCP      = 01,   ! CH$PTR to first character position.
FS__NCP      = 02,   ! CH$PTR to next character position.
FS__MAX      = 03,   ! Capacity of the counted string.
FS__LEN      = 04;   ! Current length.
```

```
!+
! FS_ALLOCATE is used to allocate and initialize a fixed-length string.
```

```
! MACRO
```

```
fs_allocate (fs_name,fs_maxl) =
  fs_name:fixed_string [fs_maxl]
  INITIAL (0,
           0,
           fs_maxl,
           0)
%;
```

```
!+
! Control information is accessed via the following macros.
```

```
! MACRO
```

```
fs_start (fs_name) =
  fs_name [fs__fcp]
%;
fs_next (fs_name) =
  fs_name [fs__ncp]
%;
fs_maxsize (fs_name) =
  fs_name [fs__max]
%;
fs_length (fs_name) =
  fs_name [fs__len]
%;
```

```
!+
! FS_INIT initializes a fixed-length string to the null string.
```

```
! MACRO
```

```
fs_init (fs_name) =
  BEGIN
    fs_length (fs_name)      = 0;
    fs_start (fs_name)      = CH$PTR (fs_name [5]);
    fs_next (fs_name)      = .fs_start (fs_name);
  END
%;
```

```
!+
```

! FS_WCHAR puts a character into the fixed-length string.

```
MACRO
  fs_wchar (fs_name,khar)      =
  BEGIN
    CH$WCHAR_A (khar, fs_next (fs_name)); !Write character and advance pointer.
    fs_length (fs_name) = .fs_length (fs_name) + 1; !Bump total length count.
  END
%;
```

! FS_RCHAR gets a character from the fixed-length string.

```
MACRO
  fs_rchar (fs_name,khar)      =
  BEGIN
    khar = CH$RCHAR_A (fs_next (fs_name)); !Read character and advance pointer.
    fs_length (fs_name) = .fs_length (fs_name) - 1; !Decrement remainder count.
  END
%;
```

! End of FSPACK.REQ

