

BBBBBBBBBBBBBB		AAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAAAAAAAAA	CCCCCCCCCCCC	KKK	KKK	UUU	UUU	PPPPPPPPPPPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKKK	UUU	UUU	PPPPPPPPPPPP
BBBBBBBBBBBBBB		AAA	AAA	CCC	KKKKKKKKKK	UUU	UUU	PPPPPPPPPPPP
BBB	BBB	AAAAAAAAAAAAAAAA	CCC	KKK	KKK	UUU	UUU	PPP
BBB	BBB	AAAAAAAAAAAAAAAA	CCC	KKK	KKK	UUU	UUU	PPP
BBB	BBB	AAAAAAAAAAAAAAAA	CCC	KKK	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBB	BBB	AAA	AAA	CCC	KKK	UUU	UUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUUUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUUUU	PPP
BBBBBBBBBBBBBB		AAA	AAA	CCCCCCCCCCCC	KKK	KKK	UUUUUUUUUUUUUUUU	PPP

```
CCCCCCCC 000000 MM MM MM MM 000000 NN NN
CCCCCCCC 000000 MM MM MM MM 000000 NN NN
CC 00 00 MMMM MMMM MMMM MMMM 00 00 NN NN
CC 00 00 MMMM MMMM MMMM MMMM 00 00 NN NN
CC 00 00 MM MM MM MM MM MM 00 00 NNNN NN
CC 00 00 MM MM MM MM MM MM 00 00 NNNN NN
CC 00 00 MM MM MM MM MM MM 00 00 NN NN
CC 00 00 MM MM MM MM MM MM 00 00 NN NN
CC 00 00 MM MM MM MM MM MM 00 00 NN NN
CC 00 00 MM MM MM MM MM MM 00 00 NN NN
CC 00 00 MM MM MM MM MM MM 00 00 NN NN
CCCCCCCC 000000 MM MM MM MM 000000 NN NN
CCCCCCCC 000000 MM MM MM MM 000000 NN NN
NNNN
```

```
RRRRRRRR EEEEEEEEE QQQQQQ
RRRRRRRR EEEEEEEEE QQQQQQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RRRRRRRR EEEEEEEEE QQ QQ
RRRRRRRR 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 EE QQ QQ
RR RR EEEEEEEEE QQQQ QQ
RR RR EEEEEEEEE QQQQ QQ
```

COMMON.REQ - BACKUP Common Definitions

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:

Backup/Restore

ABSTRACT:

This file contains the common definitions for BACKUP.

ENVIRONMENT:

VAX/VMS user mode.
--

AUTHOR: M. Jack, CREATION DATE: 27-Aug-1980

MODIFIED BY:

V03-014 LY0521 Larry Yetto 14-AUG-1984 11:01
Change version to V4.0

V03-013 LY0510 Larry Yetto 19-JUL-1984 08:40
MAXRSS was increased from 252 to 255 so some structures have
to be adjusted so that they remain longword aligned. Increase DEVTYP
in the physical volume attributes from a byte to a longword.
Add a new flag to output flags to indicate that the input
save set is nonsequential.

V03-012 LY0500 Larry Yetto 25-JUN-1984 12:38

Add a new flags for the [000000] directory parsing and
a variable to hold the journal file extend size.

V03-011 LY0487 Larry Yetto 21-MAY-1984 16:06
Increase size of list buffer so that
it will be large enough for long directory
names (ie. 8 levels of 32 char names)

V03-010 LY0473 Larry Yetto 9-APR-1984 08:25
Add OUTPUT_DIR_OPEN flag to output flags.

V03-009 LY0459 Larry Yetto 1-FEB-1984 10:22
Add support for new journal file structure level.

V03-008 CWH3008 CW Hobbs 29-Oct-1983
Update version to 'V4FT'

V03-007 JEP0004 J. Eric Pollack 7-Aug-1983 10:00
Extend syntax of /encrypt qualifier

V03-006 ACG0342 Andrew C. Goldstein, 29-Jun-1983 13:15
Use general for all externals in BACKUP

V03-005 JEP0003 J. Eric Pollack 23-Apr-1983 10:21
Add definitions for encrypted savesets.

V03-004 ACG0332 Andrew C. Goldstein, 19-Apr-1983 17:13
Add support for file highwater mark and RMS journal flags;
extend directory string buffer for long directory names.

V03-003 MLJ0104 Martin L. Jack, 24-Jan-1983 23:33
Update version number to V3.3. Add D_WILD_TERM for execute-only
directory scanning. Add D_NON_TERM for stricter directory
saving. Add INPUT_SAVE_OK for /RECORD/DELETE with errors.
Add bitmap size field to VCB.

V03-002 LMP0044 L. Mark Pilant, 21-Oct-1982 15:40
Add support for saving and restoring ACL's.

V03-001 LMP0032 L. Mark Pilant 21-Jun-1982 16:00
Add wildcard support for listings and restore operations.

V02-019 ACG0281 Andrew C. Goldstein, 8-Apr-1982 18:35
Add ODS-1 multi-header index file support

V02-018 ACG0279 Andrew C. Goldstein, 1-Apr-1982 22:00
Remove MTL_ALLOC_xxx fields

V02-017 ACG0277 Andrew C. Goldstein, 30-Mar-1982 15:07
Add ALT_SSNAME cell to hold alternate save set name

V02-016 MLJ0081 Martin L. Jack, 26-Feb-1982 15:25
Add VSR_RETAINMIN and VSR_RETAINMAX to support new home block
fields.

V02-015 MLJ0077 Martin L. Jack, 8-Feb-1982 15:11

Implement negative version numbers.

- V02-014 MLJ0075 Martin L. Jack, 28-Jan-1982 20:05
Add DIR_VERLIM and VERLIMIT attributes and field D_VERLIM to support version limit handling.
- V02-013 MLJ0073 Martin L. Jack, 19-Jan-1982 18:19
Implement /PROTECTION qualifier for save set protection and extend /LABEL qualifier to be a list.
- V02-012 MLJ0063 Martin L. Jack, 31-Dec-1981 7:30
Update BACKUP version number.
- V02-011 MLJ0062 Martin L. Jack, 2-Dec-1981 13:29
Add new structures to support /INCREMENTAL.
Add OUTPUT_IMPLICIT flag.
- V02-010 MLJ0054 Martin L. Jack, 18-Oct-1981 21:46
Remove COM_IMP_NOBACK. Add INPUT_PLACEMENT, INPUT_PLACE_LEN.
Add definitions of data structure for INPUT_PLACEMENT. Add INPUT_ON_RVN bit. Add /DELETE qualifier. Add disk quota table for quota file reconstruction. Move STAACP globals to common. Delete JPI_STS, add COM_INTERACT flag.
- V02-009 MLJ0052 Martin L. Jack, 1-Oct-1981 13:15
Implement /INTERCHANGE qualifier.
- V02-008 MLJ0043 Martin L. Jack, 8-Sep-1981 16:54
Account for RMS logical device names. Install \$GETSYI.
- V02-007 MLJ0036 Martin L. Jack, 29-Aug-1981 16:02
Extensive rewriting to complete implementation.
- V02-006 ACG0211 Andrew C. Goldstein, 16-Jul-1981 10:23
Add structures for sequential disk support
- V02-005 MLJ0025 Martin L. Jack, 8-May-1981 11:06
Reorganize qualifier database. Additions to globals area.
- V02-004 MLJ0023 Martin L. Jack, 23-Apr-1981 11:23
Implement placement attribute.
- V02-003 MLJ0021 Martin L. Jack, 20-Apr-1981 21:53
Implement /TRUNCATE qualifier.
- V02-002 MLJ0017 Martin L. Jack, 7-Apr-1981 0:48
Correct inadvertent transposition of subfields of UIC
- V02-001 MLJ0010 Martin L. Jack, 25-Mar-1981 17:09
Reorganize global storage. Prune unnecessary parts of qualifier database. Add structures for standalone ACP. Add attribute-buffer structure for file attributes.

```

!
!
! LITERAL
!
!     TRUE=      1,
!     FALSE=     0;
!
!
!

```

```

STRUCTURE
    BBLOCK[O,P,S,E;N]=
        [N]
        (BBLOCK + 0)<P,S,E>;

```

```

MACRO
    BUG_CHECK(CODE,MESSAGE)=
        BEGIN
            EXTERNAL LITERAL %NAME (BACKUP$, CODE);
            SIGNAL_STOP(%NAME(BACKUP$, CODE));
            END %;

```

```

LITERAL
    COPY_BUFF_COUNT=2,           ! Default buffer count for disk-to-disk
    COPY_BUFF_SIZE= 32768+256+16, ! Default buffer size for disk-to-disk
    SMALL_DISK= 4096,           ! Largest disk with no bad block data
    LIST_SIZE= 256,             ! Size of listing line
    MAX_RECORD= 2048,           ! Maximum length attribute record
    MAX_ATTRS= 14,              ! Maximum number of attributes in list
    STA_IN_CHAN= %X'1FFFFF',    ! Pseudo-channel for input
    STA_OUT_CHAN= %X'2FFFFF',   ! Pseudo-channel for output
    ATR$C_EXTFID= 39,           ! Internal attribute code for
                                ! extension file ID
    ATR$S_EXTFID= 6,            !
    ATR$C_SEGNUM= 40,           ! Internal attribute code for
                                ! extension section number
    ATR$S_SEGNUM= 2;            !

```

```

SWITCHES
    ADDRESSING_MODE(
        EXTERNAL=GENERAL,
        NONEXTERNAL=WORD_RELATIVE);

```

```

PSECT
    CODE=      CODE,
    PLIT=      CODE,
    OWN=      DATA(ADDRESSING_MODE(LONG_RELATIVE)),
    GLOBAL=    DATA;

```

! Macros to direct QIO and QIOW services to the standalone ACP or the real ACP depending on the operation in progress. The C\$ forms are conditional on whether the output volume is in image mode. The S\$ form invokes the standalone ACP and is used when we already know the output volume is in image mode.

```
KEYWORDMACRO C$QIO(EFN=0,CHAN=0,FUNC=0,IOSB=0,ASTADR=0,ASTPRM=0,P1=0,P2=0,P3=0,P4=0,P5=0,P6=0)=
(
  IF (CHAN) GEQU STA_IN_CHAN
  THEN
    BEGIN
      EXTERNAL ROUTINE STA_QIO;
      STA_QIO(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
    END
  ELSE
    BEGIN
      EXTERNAL ROUTINE SY$QIO: ADDRESSING MODE(ABSOLUTE);
      SY$QIO(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
    END
  ) %;
```

```
KEYWORDMACRO C$QIOW(EFN=0,CHAN=0,FUNC=0,IOSB=0,ASTADR=0,ASTPRM=0,P1=0,P2=0,P3=0,P4=0,P5=0,P6=0)=
(
  IF (CHAN) GEQU STA_IN_CHAN
  THEN
    BEGIN
      EXTERNAL ROUTINE STA_QIOW;
      STA_QIOW(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
    END
  ELSE
    BEGIN
      EXTERNAL ROUTINE SY$QIOW: ADDRESSING MODE(ABSOLUTE);
      SY$QIOW(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
    END
  ) %;
```

```
KEYWORDMACRO S$QIO(EFN=0,CHAN=0,FUNC=0,IOSB=0,ASTADR=0,ASTPRM=0,P1=0,P2=0,P3=0,P4=0,P5=0,P6=0)=
(
  EXTERNAL ROUTINE STA_QIO;
  STA_QIO(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
  ) %;
```

```
KEYWORDMACRO S$QIOW(EFN=0,CHAN=0,FUNC=0,IOSB=0,ASTADR=0,ASTPRM=0,P1=0,P2=0,P3=0,P4=0,P5=0,P6=0)=
(
  EXTERNAL ROUTINE STA_QIOW;
  STA_QIOW(EFN,CHAN,FUNC,IOSB,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
  ) %;
```

MACRO

LITERAL

MACRO

```
FC_FAB= 0,0,0,0 %,
FC_RAB= FAB$C_BLN,0,0,0 %,
FC_NAM= FAB$C_BLN+RAB$C_BLN,0,0,0 %,
FC_RLF= FAB$C_BLN+RAB$C_BLN+NAM$C_BLN,0,0,0 %,
FC_ESA= FAB$C_BLN+RAB$C_BLN+2*NAM$C_BLN,0,0,0 %,
FC_RSA= FAB$C_BLN+RAB$C_BLN+2*NAM$C_BLN+(NAM$C_MAXRSS+1),0,0,0 %,
FC_S_AREA= FAB$C_BLN+RAB$C_BLN+2*NAM$C_BLN+2*(NAM$C_MAXRSS+1) %;
```

MACRO

QUAL_INPU_LIST=	0.0.32.0 %	:	Input parameters
QUAL_OUTP_LIST=	4.0.32.0 %	:	Output parameters
QUAL_ANAL=	8.0.1.0 %	:	/ANALYZE
QUAL_BACK=	8.1.1.0 %	:	/BACKUP
QUAL_BEFO=	8.2.1.0 %	:	/BEFORE
QUAL_BEFO BACK=	8.3.1.0 %	:	/BEFORE=BACKUP
QUAL_BLOC=	8.4.1.0 %	:	/BLOCK SIZE
QUAL_BUFF=	8.5.1.0 %	:	/BUFFER COUNT
QUAL_COMM=	8.6.1.0 %	:	/COMMENT
QUAL_COMP=	8.7.1.0 %	:	/COMPARE
QUAL_CONF=	8.8.1.0 %	:	/CONFIRM
QUAL_CRC=	8.9.1.0 %	:	/CRC
QUAL_CREA=	8.10.1.0 %	:	/CREATED
QUAL_DENS=	8.11.1.0 %	:	/DENSITY
QUAL_EXCL=	8.12.1.0 %	:	/EXCLUDE
QUAL_EXPI=	8.13.1.0 %	:	/EXPIRED
QUAL_FAST=	8.14.1.0 %	:	/FAST
QUAL_FULL=	8.15.1.0 %	:	/FULL
QUAL_GROU=	8.16.1.0 %	:	/GROUP SIZE
QUAL_IGNO_NOBA=	8.17.1.0 %	:	/IGNORE=NOBACKUP
QUAL_IGNO_INTE=	8.18.1.0 %	:	/IGNORE=INTERLOCK
QUAL_IMAGE=	8.19.1.0 %	:	/IMAGE
QUAL_INCR=	8.20.1.0 %	:	/INCREMENTAL

QUAL_INIT=	8,21,1,0 %	/INITIALIZE
QUAL_JOUR=	8,22,1,0 %	/JOURNAL
QUAL_LABE=	8,23,1,0 %	/LABEL
QUAL_LIST=	8,24,1,0 %	/LIST
QUAL_LOG=	8,25,1,0 %	/LOG
QUAL_MODI=	8,26,1,0 %	/MODIFIED
QUAL_NEWV=	8,27,1,0 %	/NEW VERSION
QUAL_OVER=	8,28,1,0 %	/OVERLAY
QUAL_I_OWNE=	8,29,1,0 %	/OWNER UIC on input
QUAL_I_OWN_WGRP=	8,30,1,0 %	Wildcard group
QUAL_I_OWN_WMEM=	8,31,1,0 %	Wildcard member
QUAL_O_OWNE=	12,0,1,0 %	/OWNER UIC on output
QUAL_O_OWN_DEFA=	12,1,1,0 %	=DEFAULT
QUAL_O_OWN_ORIG=	12,2,1,0 %	=ORIGINAL
QUAL_O_OWN_PARE=	12,3,1,0 %	=PARENT
QUAL_O_OWN_UIC=	12,4,1,0 %	=[g,m]
QUAL_PHYS=	12,5,1,0 %	/PHYSICAL
QUAL_RECO=	12,6,1,0 %	/RECORD
QUAL_REPL=	12,7,1,0 %	/REPLACE
QUAL_REWI=	12,8,1,0 %	/REWIND
QUAL_I_SAVE=	12,9,1,0 %	/SAVE_SET on input
QUAL_O_SAVE=	12,10,1,0 %	/SAVE_SET on output
QUAL_SELE=	12,11,1,0 %	/SELECT
QUAL_SINC=	12,12,1,0 %	/SINCE
QUAL_SINC_BACK=	12,13,1,0 %	/SINCE=BACKUP
QUAL_TRUN=	12,14,1,0 %	/TRUNCATE
QUAL_VERI=	12,15,1,0 %	/VERIFY
QUAL_VOLU=	12,16,1,0 %	/VOLUME
QUAL_INTE=	12,17,1,0 %	/INTERCHANGE
QUAL_DELE=	12,18,1,0 %	/DELETE
QUAL_PROT=	12,19,1,0 %	/PROTECTION on output
QUAL_SS_ENCRYP=	12,20,1,0 %	/ENCRYPT save set
QUAL_SS_FILE=	12,27,1,0 %	Save set is a disk file
QUAL_I_FT1=	12,28,1,0 %	Input is Files-11 or physical
QUAL_ISAV=	12,29,1,0 %	Input is save set
QUAL_OF11=	12,30,1,0 %	Output is Files-11 or physical
QUAL_OSAV=	12,31,1,0 %	Output is save set
QUAL_BEFO_VALUE=	16,0,0,0 %	/BEFORE quadword time value
QUAL_COMM_DESC=	24,0,0,0 %	/COMMENT descriptor
QUAL_SINC_VALUE=	32,0,0,0 %	/SINCE quadword time value
QUAL_CMD_DESC=	40,0,0,0 %	Command line descriptor
QUAL_EXCL_LIST=	48,0,32,0 %	/EXCLUDE list head
QUAL_JOUR_FC=	52,0,32,0 %	/JOURNAL file context
QUAL_LIST_FC=	56,0,32,0 %	/LIST file context
QUAL_I_OWN_VALU=	60,0,32,0 %	/OWNER UIC value on input
QUAL_I_OWN_MEM=	60,0,16,0 %	Member portion
QUAL_I_OWN_GRP=	62,0,16,0 %	Group portion
QUAL_O_OWN_VALU=	64,0,32,0 %	/OWNER UIC value on output
QUAL_O_OWN_MEM=	64,0,16,0 %	Member portion
QUAL_O_OWN_GRP=	66,0,16,0 %	Group portion
QUAL_SELE_LIST=	68,0,32,0 %	/SELECT list head
QUAL_BLOC_VALUE=	72,0,16,0 %	/BLOCK SIZE value
QUAL_BUFF_VALUE=	76,0,8,0 %	/BUFFER COUNT value
QUAL_DENS_VALUE=	77,0,8,0 %	/DENSITY code
QUAL_GROU_VALUE=	78,0,8,0 %	/GROUP SIZE value
QUAL_VOLU_VALUE=	79,0,8,0 %	/VOLUME value

```

QUAL_LABE_LIST= 80,0,32,0 %;  ! /LABEL list head
QUAL_PROT_VALUE=84,0,16,0 %;  ! /PROTECTION value
QUAL_Cryp_USERKEY=88,0,0,0 %;  ! Descriptor for user specified key
QUAL_Cryp_USERALG=96,0,0,0 %;  ! Descriptor for user specified algorithm
QUAL_Cryp_VLIST=104,0,32,0 %;  ! Pointer to list of key values

```

```

LITERAL
QUAL_S_AREA= 112 ;           ! Length of qualifiers block

```

```

MACRO
! Parameter value block.
!
QUAL_NEXT= 0,0,32,0 %;       ! Pointer to next block (common to all)
QUAL_PARA_FC= 4,0,32,0 %;    ! Parameter file context
QUAL_EXP_DESC= 8,0,0,0 %;    ! Descriptor for expanded string
QUAL_DEV_DESC= 16,0,0,0 %;   ! Descriptor for device portion only
QUAL_DVI_DESC= 24,0,0,0 %;   ! Descriptor for NAMST_DVI string
QUAL_USE_COUNT= 32,0,32,0 %; ! Count of input files from this spec
QUAL_USE_CHKPT= 36,0,32,0 %; ! Cell to checkpoint QUAL_USE_COUNT

```

```

MACRO
QUAL_EXCL_DESC= 4,0,0,0 %;   ! /EXCLUDE value
QUAL_SELE_DESC= 4,0,0,0 %;   ! /SELECT value
QUAL_LABE_VALUE=4,0,0,0 %;   ! /LABEL value

```

```

!
! Encrypt key value block

```

```

MACRO
QUAL_ENVA_LINK= 0,0,32,0 %;   ! Link word
QUAL_ENVA_DESC= 4,0,0,0 %;   ! Value string descriptor

```

```

LITERAL
QUAL_S_INPU= 40,
QUAL_S_OUTP= 32,
QUAL_S_EXCL= 12,
QUAL_S_SELE= 12,
QUAL_S_LABE= 16,
QUAL_S_ENCV= 12;             ! Encrypt value list item

```

BAD BLOCK DESCRIPTOR

!:
MACRO

! Format of bad block descriptor returned by routine GET_BADBLOCKS.

BAD_NUMDESC=	0,0,32,0 %.	! Number of descriptors
BAD_SERIAL=	4,0,32,0 %.	! Pack serial number
BAD_DESC=	8,0,0,0 %.	! Origin of descriptors
BAD_LBN=	0,0,32,0 %.	! LBN
BAD_COUNT=	4,0,32,0 %:	! Count

LITERAL

BAD_S_HEADER=	8,	! Size of header area
BAD_S_DESC=	8:	! Size of one descriptor

PROCESSED FILE ID LIST

MACRO

```
! Field definitions for structure pointed to by INPUT_PROC_LIST.
!
REC_NEXT=      0,0,32,0 %,      ! Pointer to next block
REC_QUAL=      4,0,32,0 %,      ! Value of INPUT_QUAL for this block
REC_USED=      8,0,16,0 %,      ! Count of used entries in this block
REC_VOLUME=    10,0,16,0 %,     ! Volume number
REC_FID_BASE=  12,0,0,0 %,      ! Base of FID/DID entries

REC_FID=       0,0,0,0 %,       ! File ID
REC_DID=       6,0,0,0 %,       ! Directory ID
```

LITERAL

```
REC_MAX_COUNT= 64,             ! Number of FID/DID slots
REC_S_ENTRY=   12 + REC_MAX_COUNT*12;
```

```

!
!
MACRO

```

BUFFER CONTROL

```

! Buffer control block format.

```

BCB_FLINK	= 00, 0, 32, 0%,	! queue forward link
BCB_BLINK	= 04, 0, 32, 0%,	! queue back link
BCB_SIZE	= 08, 0, 16, 0%,	! buffer size
BCB_STATE	= 10, 0, 08, 0%,	! buffer state
BCB_BUFFER	= 12, 0, 32, 0%,	! buffer address
BCB_RECORD	= 16, 0, 32, 0%,	! current record pointer
BCB_BLOCKNUM	= 20, 0, 32, 0%,	! disk block number
BCB_IOSB	= 24, 0, 00, 0%,	! I/O status block
BCB_STATUS	= 24, 0, 32, 0%,	! I/O status longword
BCB_IO_STATUS	= 24, 0, 16, 0%,	! I/O status word
BCB_IO_BCOUNT	= 26, 0, 16, 0%,	! I/O status byte count
BCB_STATUS2	= 28, 0, 32, 0%,	! second I/O status longword
BCB_SUCC_ACT	= 32, 0, 32, 0%,	! success action routine
BCB_FAIL_ACT	= 36, 0, 32, 0%;	! failure action routine

```

LITERAL
BCB_LENGTH      = 40,
! length of buffer control block

```

```

! Buffer state codes

```

BCB_S_IDLE	= 0,	! idle
BCB_S_READ	= 1,	! read operation pending
BCB_S_WRITE	= 2,	! write operation pending
BCB_S_DATA	= 3,	! holding data, no I/O pending
BCB_S_REREAD	= 4;	! read on alternate channel

ATTRIBUTE INPUT BUFFERS

MACRO

: Backup summary record input buffer.

BSR_SSNAME=	0.0.0.0 %.	: Descriptor for SSNAME
BSR_COMMAND=	8.0.0.0 %.	: Descriptor for COMMAND
BSR_COMMENT=	16.0.0.0 %.	: Descriptor for COMMENT
BSR_USERNAME=	24.0.0.0 %.	: Descriptor for USERNAME
BSR_DATE=	36.0.0.0 %.	: Value of DATE
BSR_SYSVER=	44.0.0.0 %.	: Descriptor for SYSVER
BSR_NODENAME=	52.0.0.0 %.	: Descriptor for NODENAME
BSR_DRIVEID=	60.0.0.0 %.	: Descriptor for DRIVEID
BSR_BACKVER=	68.0.0.0 %.	: Descriptor for BACKVER
BSR_VOLSETNAM=	76.0.0.0 %.	: Descriptor for VOLSETNAM
BSR_BACKSIZE=	84.0.0.0 %.	: Value of BACKSIZE
BSR_USERUIC=	92.0.32.0 %.	: Value of USERUIC
BSR_SIR=	96.0.32.0 %.	: Value of SIR
BSR_BLOCKSIZE=	100.0.32.0 %.	: Value of BLOCKSIZE
BSR_BACKFILES=	104.0.32.0 %.	: Value of BACKFILES
BSR_OPSYS=	108.0.16.0 %.	: Value of OPSYS
BSR_XORSIZE=	110.0.16.0 %.	: Value of XORSIZE
BSR_BUFFERS=	112.0.16.0 %.	: Value of BUFFERS
BSR_NVOLS=	114.0.16.0 %.	: Value of NVOLS
BSR_CRYPTDATA=	116.0.0.0 %.	: Value of CRYPTDATA
:NEXT=	140.0.0.0:	: Next available byte

MACRO

: Volume summary record input buffer.

VSR_VOLNAME=	0.0.0.0 %.	: Descriptor for VOLNAME
VSR_OWNERNAME=	8.0.0.0 %.	: Descriptor for OWNERNAME
VSR_FORMAT=	16.0.0.0 %.	: Descriptor for FORMAT
VSR_VOLDATE=	24.0.0.0 %.	: Value of VOLDATE
VSR_TOTSIZE=	32.0.0.0 %.	: Value of TOTSIZE
VSR_VOLOWNER=	40.0.32.0 %.	: Value of VOLOWNER
VSR_VOLSIZE=	44.0.32.0 %.	: Value of VOLSIZE
VSR_TOTFILES=	48.0.32.0 %.	: Value of TOTFILES
VSR_MAXFILES=	52.0.32.0 %.	: Value of MAXFILES
VSR_MAXFILNUM=	56.0.32.0 %.	: Value of MAXFILNUM
VSR_SERIALNUM=	60.0.32.0 %.	: Value of SERIALNUM
VSR_VOLSTRUCT=	64.0.16.0 %.	: Value of VOLSTRUCT
VSR_STRUCLEV=	65.0.8.0 %.	: Structure level part of VOLSTRUCT
VSR_RVN=	66.0.16.0 %.	: Value of RVN
VSR_PROTECT=	68.0.16.0 %.	: Value of PROTECT
VSR_FILEPROT=	70.0.16.0 %.	: Value of FILEPROT
VSR_RECPROT=	72.0.16.0 %.	: Value of RECPROT
VSR_VOLCHAR=	74.0.16.0 %.	: Value of VOLCHAR
VSR_EXTEND=	76.0.16.0 %.	: Value of EXTEND
VSR_CLUSTER=	78.0.16.0 %.	: Value of CLUSTER
VSR_RESFILES=	80.0.16.0 %.	: Value of RESFILES
VSR_WINDOW=	82.0.8.0 %.	: Value of WINDOW

VSR_LRU_LIM=	83.0.8.0 %	:	Value of LRU_LIM
VSR_INDEXLBN=	84.0.32.0 %	:	Value of INDEXLBN
VSR_BOOTBLOCK=	88.0.0.0 %	:	Descriptor for BOOTBLOCK
VSR_RETAINMIN=	96.0.0.0 %	:	Value of RETAINMIN
VSR_RETAINMAX=	104.0.0.0 %	:	Value of RETAINMAX

MACRO

: Physical volume attribute record input buffer.

PVA_DEVNAM=	0.0.0.0 %	:	Descriptor for DEVNAM
PVA_LABEL=	8.0.0.0 %	:	Descriptor for LABEL
PVA_BADBLOCK=	16.0.0.0 %	:	Descriptor for BADBLOCK
PVA_MAXBLOCK=	24.0.32.0 %	:	Value of MAXBLOCK
PVA_SERIAL=	28.0.32.0 %	:	Value of SERIAL
PVA_CYLINDERS=	32.0.16.0 %	:	Value of CYLINDERS
PVA_SECTORS=	34.0.8.0 %	:	Value of SECTORS
PVA_TRACKS=	35.0.8.0 %	:	Value of TRACKS
PVA_DEVTYP=	36.0.32.0 %	:	Value of DEVTYP

MACRO

: File attribute record input buffer.

FAR_FILENAME=	0.0.0.0 %	:	Descriptor for FILENAME
FAR_PLACEMENT=	8.0.0.0 %	:	Descriptor for PLACEMENT
FAR_RECATTR=	16.0.0.0 %	:	Value of RECATTR
FAR_CREDATE=	48.0.0.0 %	:	Value of CREDATE
FAR_REVDATE=	56.0.0.0 %	:	Value of REVDATE
FAR_EXPDATE=	64.0.0.0 %	:	Value of EXPDATE
FAR_BAKDATE=	72.0.0.0 %	:	Value of BAKDATE
FAR_FID=	80.0.0.0 %	:	Value of FID
FAR_FID_NUM=	80.0.16.0 %	:	
FAR_FID_SEQ=	82.0.16.0 %	:	
FAR_FID_RVNW=	84.0.16.0 %	:	
FAR_FID_RVN=	84.0.8.0 %	:	
FAR_FID_NMX=	85.0.8.0 %	:	
FAR_BACKLINK=	86.0.0.0 %	:	Value of BACKLINK
FAR_DID_NUM=	86.0.16.0 %	:	
FAR_DID_SEQ=	88.0.16.0 %	:	
FAR_DID_RVNW=	90.0.16.0 %	:	
FAR_FILESIZE=	92.0.32.0 %	:	Value of FILESIZE
FAR_UIC=	96.0.32.0 %	:	Value of UIC
FAR_UICMEMBER=	96.0.16.0 %	:	Member part of UIC
FAR_UICGROUP=	98.0.16.0 %	:	Group part of UIC
FAR_UCHAR=	100.0.32.0 %	:	Value of UCHAR
FAR_BOOTVBN=	104.0.32.0 %	:	Value of BOOTVBN
FAR_STRUCLEV=	108.0.16.0 %	:	Value of STRUCLEV
FAR_STRUCLEVB=	109.0.8.0 %	:	Structure level part of STRUCLEV
FAR_FPRO=	110.0.16.0 %	:	Value of FPRO
FAR_RPRO=	112.0.16.0 %	:	Value of RPRO
FAR_REVISION=	114.0.16.0 %	:	Value of REVISION
FAR_DIR_UIC=	116.0.32.0 %	:	Value of DIR_UIC
FAR_DIR_FPRO=	120.0.16.0 %	:	Value of DIR_FPRO

FAR_ACLEVEL=	122,0,8,0 %	:	Value of ACLEVEL
FAR_DIR_STATUS=	123,0,8,0 %	:	Value of DIR_STATUS
FAR_DIR_VERLIM=	124,0,16,0 %	:	Value of DIR_VERLIM
FAR_VERLIMIT=	126,0,16,0 %	:	Value of VERLIMIT
FAR_ACLSEGMENT=	128,0,32,0 %	:	ACL segment descriptor
FAR_HIGHWATER=	136,0,32,0 %	:	Value of HIGHWATER
FAR_JNL_FLAGS=	140,0,16,0 %	:	Value of JNL_FLAGS

LITERAL

BSR_LENGTH=	140,	:	Length of BSR area
VSR_LENGTH=	112,	:	Length of VSR area
PVA_LENGTH=	40,	:	Length of PVA area
FAR_LENGTH=	144,	:	Length of FAR area
ATTBUF_LENGTH=	MAX(BSR_LENGTH, VSR_LENGTH, PVA_LENGTH, FAR_LENGTH);		

STANDALONE ACP

MACRO

! Field definitions for RVT/MTL.

MTL_CHAN_1=	0.0,32.0 %	! Pointer to VCB to which each of 2
MTL_CHAN_2=	4.0,32.0 %	channels is assigned
MTL_WINDOW=	8.0,32.0 %	! Pointer to WCB for accessed file
MTL_HEADER=	12.0,32.0 %	! Pointer to header for accessed file
MTL_ACLFL=	16.0,32.0 %	! ACL queue
MTL_ACLBL=	20.0,32.0 %	! head
MTL_FID=	24.0,0.0 %	! File ID of accessed file
MTL_FID_NUM=	24.0,16.0 %	
MTL_FID_SEQ=	26.0,16.0 %	
MTL_FID_RVNW=	28.0,16.0 %	
MTL_FID_RVN=	28.0,8.0 %	
MTL_FID_NMX=	29.0,8.0 %	
MTL_STRUCLEV=	30.0,8.0 %	! Structure level of volume set
MTL_SETCOUNT=	31.0,8.0 %	! Count of volumes in volume set
MTL_FILESIZE=	32.0,32.0 %	! Size of accessed file
MTL_STRUCNAME=	36.0,0.0 %	! Volume set name
MTL_RVN_BASE=	48.0,8.0 %	! RVN mounted on first VCB
MTL_SEQ_DISK=	49.0,1.0 %	! Set if sequential disk volume set
MTL_NEW_ACL=	49.1,1.0 %	! Set if ACL must be written
MTL_NOHWM=	49.2,1.0 %	! Highwater marking disabled for volume set
MTL_VCB(n)=	52+4*(n),0,32.0 %	! Pointer to VCB for RVN n

MACRO

! Field definitions for VCB.

VCB_INDEXF=	0.0,32.0 %	! Pointer to index file window
VCB_CLUSTER=	4.0,16.0 %	! Cluster factor
VCB_RVN=	6.0,8.0 %	! Relative volume number
VCB_OUTPUT=	7.0,1.0 %	! True if output volume
VCB_ODS_2=	7.1,1.0 %	! True if ODS-2 volume
VCB_INIT_DONE=	7.2,1.0 %	! True if initialization has been done
VCB_SAVESET=	7.3,1.0 %	! True if saveset volume
VCB_NOTVOLSET=	7.4,1.0 %	! Sequential disk volume is not in a set
VCB_CHAN=	8.0,16.0 %	! Channel number assigned to this RVN
VCB_IOCOUNTE=	10.0,16.1 %	! Count of pending read/write QIOs
VCB_BITMAP_LBN=	12.0,32.0 %	! LBN of storage bitmap
VCB_IMAP=	16.0,32.0 %	! Pointer to index file bitmap image
VCB_IMAP_LBN=	20.0,32.0 %	! LBN of index file bitmap
VCB_INIT_HDRS=	24.0,16.0 %	! Bit mask of initialized headers
VCB_HDR_OFFSET=	26.0,16.0 %	! VBN offset to file headers
VCB_MAXFILIDX=	28.0,32.0 %	! Number of bits in index file bitmap
VCB_DEVICE=	32.0,0.0 %	! Descriptor for device name
VCB_ACB_FLINK=	40.0,32.0 %	! Queue header for ACB queue
VCB_ACB_BLINK=	44.0,32.0 %	
VCB_FAB=	48.0,32.0 %	! Pointer to FAB with filespec
VCB_BITMAP_SIZE=	52.0,16.0 %	! Storage bitmap size in blocks
VCB_VOLNAME=	56.0,0.0 %	! Volume label

MACRO

! Field definitions for WCB.

WCB_LINK=	0,0,32,0 %	! Link to next block
WCB_VBN=	4,0,32,0 %	! In first block only, starting VBN of window
WCB_SIZE=	8,0,8,0 %	! Number of entries in window block
WCB_FREE=	9,0,8,0 %	! Number of free entries in window
WCB_RVN=	10,0,8,0 %	! RVN
WCB_FLAGS=	11,0,8,0 %	! Flag byte
WCB_BLACKHOLE=	11,0,1,0 %	! Flush write I/Os into this window
WCB_CUR_HWM=	12,0,32,0 %	! Current highwater mark for open file
WCB_SET_HWM=	16,0,32,0 %	! Highwater mark written for open file
WCB_COUNT=	0,0,32,0 %	! Count of blocks
WCB_LBN=	4,0,32,0 %	! LBN of blocks

MACRO

! Field definitions for ACB.

ACB_FLINK=	0,0,32,0 %	! Forward link
ACB_BLINK=	4,0,32,0 %	! Backward link
ACB_COUNT=	8,0,32,0 %	! Block count
ACB_LBN=	12,0,32,0 %	! Logical block number

LITERAL

MTL_S_ENTRY=	52,	! Size of entry, less VCB pointers
VCB_S_ENTRY=	68,	! Size of entry
WCB_S_HEADER=	20,	! Size of header
WCB_S_ENTRY=	8,	! Size of each entry
ACB_S_ENTRY=	16,	! Size of entry

DIRECTORY STACK ENTRY

MACRO

D_REC=	0,0,32,0 %	Pointer to current record
D_VER=	4,0,32,0 %	Pointer to current version if ODS-2
D_BUF_LEN=	8,0,32,0 %	Length of buffer in bytes
D_BUF_ADDR=	12,0,32,0 %	Pointer to buffer
D_BUF_LIM=	16,0,32,0 %	Limit of significant part of buffer
D_VBN=	20,0,32,0 %	Next VBN to be read
D_DIR_LEN=	24,0,32,0 %	Length of directory in blocks
D_FID=	28,0,0,0 %	File ID of directory
D_SAV_LEN=	34,0,8,0 %	Saved length of directory string
D_DIR_SCAN=	35,0,1,0 %	True if directory file to be scanned
D_DIR_MATCHES=	35,1,1,0 %	True if current directory matches
D_WILD_TERM=	35,2,1,0 %	True if terminator contains wildcard
D_NON_TERM=	35,3,1,0 %	True if terminator is "*"
D_TERM_DESC=	36,0,0,0 %	Termination string descriptor
D_NAME_DESC=	44,0,0,0 %	Current filename string descriptor
D_TERM_VER=	56,0,16,0 %	Termination version number
D_FPRO=	58,0,16,0 %	Directory file protection
D_UIC=	60,0,32,0 %	Directory file owner
D_VERLIM=	64,0,16,0 %	Directory default version limit
D_VER_COUNT=	66,0,16,1 %	File version counter (0, -1, ...)

LITERAL

```
D_K_NLEVELS=      9,      ! Number of level stack entries
D_S_ENTRY=       68;      ! Size of level stack entry in bytes
```

DISK QUOTA TABLE

MACRO

DQF_LLINK=	0,0,32,0 %	! Pointer to entry with lesser UIC
DQF_RLINK=	4,0,32,0 %	! Pointer to entry with greater UIC
DQF_UIC=	8,0,32,0 %	! UIC
DQF_USAGE=	12,0,32,0 %	! Usage in blocks
DQF_PERMQUOTA=	16,0,32,0 %	! Permanent disk quota
DQF_OVERDRAFT=	20,0,32,0 %	! Overdraft limit

LITERAL

DQF_S_ENTRY=	24;	! Size of entry in bytes
--------------	-----	--------------------------

```
!
! COMMON AREA
```

```
MACRO G$DEF[A,B]=
  OWN      A:      B  ALIGN(0) %;
```

```
! Macro to be called in each module to define the common area.
```

```
MACRO G$DEFINE(A)=
  PSECT OWN=COMMON(OVERLAY,ADDRESSING_MODE(LONG_RELATIVE));
  G$DEF(G$LIST);
  PSECT OWN=DATA;
  %;
```

```
! List of elements of G$AREA. In each entry, the first parameter is the
! name, and the second parameter is null or is a structure attribute.
```

```
MACRO G$LIST =
```

```
! Marker for base of area.
```

```
GLOBAL_BASE, VECTOR[0],
```

```
! Buffer queue headers.
```

```
FREE_LIST, VECTOR[2], ! Free queue header
INPUT_WAIT, VECTOR[2], ! Input pending queue header
REREAD_WAIT, VECTOR[2], ! Re-read pending queue header
OUTPUT_WAIT, VECTOR[2], ! Output pending queue header
```

```
! Environmental values.
```

```
JPI_UIC, LONG, ! JPI$ UIC
JPI_USERNAME, VECTOR[12,BYTE], ! JPI$-USERNAME
JPI_DATE, VECTOR[2], ! $GETTIM
JPI_NODE_DESC, BBLOCK[8], ! Translation of SYSS$NODE
JPI_CURPRIV, BBLOCK[8], ! Current process privileges
SYI_VERSION, LONG, ! SYI$ VERSION
SYI_SID, LONG, ! SYI$-SID
```

```
! Save set handling.
```

```
RWSV_HOLD_LIST, VECTOR[2], ! Queue header for error rewrites
RWSV_CRC16, VECTOR[16], ! CRC-16 polynomial table
RWSV_AUTODIN, VECTOR[16], ! AUTODIN-II polynomial table
RWSV_FILESET_ID, VECTOR[8,BYTE], ! File set identifier
RWSV_VOLUME_ID, VECTOR[12,BYTE], ! Volume identifier
RWSV_VOL_NUMBER, WORD, ! Save set volume number
RWSV_SEG_NUMBER, WORD, ! File section number
RWSV_FILE_NUMBER, LONG, ! File sequence number
RWSV_SAVE_QUAL, REF BBLOCK, ! Pointer to save set qualifiers block
RWSV_SAVE_FAB, REF BBLOCK, ! Pointer to FAB
```

RWSV_CHAN,	LONG,	Channel assigned to save set medium
RWSV_XOR_BCB,	REF BBLOCK,	Pointer to BCB for XOR block
RWSV_IN_SEQ,	LONG,	Input block sequence number
RWSV_IN_SEQ_0,	LONG,	RWSV_IN_SEQ at start of reel
RWSV_IN_XOR_SEQ,	LONG,	Sequence number of last XOR block
RWSV_IN_XOR_RFA,	BBLOCK[6],	RFA of last XOR block
RWSV_LOOKAHEAD,	BYTE,	Buffer count from summary record
RWSV_XOR_SIZE,	BYTE,	XOR group size from summary record
RWSV_IN_GROUP_SIZE,	LONG,	XOR group size of save set
RWSV_IN_ERRORS,	WORD,	Count of input errors
RWSV_IN_XORUSE,	WORD,	Count of XOR recoveries performed
RWSV_IN_ORGERR,	VECTOR[2],	Original STS/STV of a train of loses
RWSV_IN_VBN,	LONG,	Current VBN in save set file
RWSV_IN_VBN_0,	LONG,	Saved input VBN
RWSV_ALLOC,	LONG,	Number of blocks available in save set file
RWSV_EOF,	LONG,	End of file VBN if save set file
RWSV_OUT_SEQ,	LONG,	Output block sequence number
RWSV_OUT_VBN,	LONG,	Output VBN
RWSV_OUT_BLOCK_COUNT,	LONG,	Count of blocks on output tape
RWSV_OUT_ERRORS,	WORD,	Count of output errors
RWSV_SEQ_ERRORS,	WORD,	Count of consecutive input errors
RWSV_OUT_GROUP_COUNT,	BYTE,	Count of blocks in output XOR group
RWSV_PADDING,	BBLOCK[3],	*** Padding ***

! General global variables.

QUAL,	BBLOCK[QUAL_S_AREA],	! Qualifier database
COM_SSNAME,	BBLOCK[8],	Descriptor for save set name
COM_VALID_TYPES,	BITVECTOR[16],	Bit mask of valid input record types
COM_FLAGS,	BBLOCK[2],	Common flags
COM_PADDING,	BBLOCK[1],	*** Padding ***
COM_BUFF_COUNT,	BYTE,	Count of buffers in pool
COM_I_SETCOUNT,	BYTE,	Input volume set count
COM_O_SETCOUNT,	BYTE,	Output volume set count
COM_I_STRUCNAME,	VECTOR[12,BYTE],	Input volume set name
COM_O_STRUCNAME,	VECTOR[12,BYTE],	Output volume set name
COM_O_BSRDATE,	VECTOR[2],	Date of backup from summary record
ALT_SSNAME,	VECTOR[32,BYTE],	! Storage for alternate save set name

INPUT_FUNC,	BYTE,	! IOS READVBLK or IOS READLBLK
INPUT_RTYPE,	BYTE,	! BRH\$K VBN or BRH\$K CBN
OUTPUT_FUNC,	BYTE,	! IOS WRITEVBLK or IOS WRITELBLK
FAST_STRUCLEV,	BYTE,	! Structure level of input volume set

! Input context.

INPUT_BEG,	VECTOR[0],	! Beginning of input context
INPUT_CHAN,	LONG,	! Input channel
INPUT_FLAGS,	BBLOCK[2],	! Input flag bits
INPUT_PADDING,	BBLOCK[2],	! *** Padding ***
INPUT_FAB,	REF BBLOCK,	! Pointer to input FAB
INPUT_NAM,	REF BBLOCK,	! Pointer to input NAM block
INPUT_BCB,	REF BBLOCK,	! Pointer to input BCB

INPUT_QUAL,	REF BBLOCK,	! Pointer to input qualifiers block
INPUT_BAD,	REF BBLOCK,	! Pointer to input bad block data
INPUT_BLOCK,	LONG,	! Current input block number
INPUT_MAXBLOCK,	LONG,	! Maximum input block number
INPUT_MEDIA_ID,	LONG,	! Media id of input device
INPUT_NAMEDESC,	BBLOCK[8],	! Descriptor for input file name
INPUT_STATBLK,	BBLOCK[8],	! Statistics block
INPUT_HDR_BEG,	VECTOR[0],	! Beginning of header data
INPUT_CREDATE,	VECTOR[2],	! Creation date
INPUT_REVDATE,	VECTOR[2],	! Revision date
INPUT_EXPDATE,	VECTOR[2],	! Expiration date
INPUT_BAKDATE,	VECTOR[2],	! Backup date
INPUT_FILEOWNER,	BBLOCK[4],	! File owner UIC
INPUT_FILECHAR,	BBLOCK[4],	! File characteristics
INPUT_RECATTR,	BBLOCK[32],	! Record attributes
INPUT_HDR_END,	VECTOR[0],	! End of header data
INPUT_END,	VECTOR[0],	! End of input context
INPUT_PROC_LIST,	REF BBLOCK,	! List of processed file ID's
INPUT_PLACEMENT,	VECTOR[2],	! List head for placement data
INPUT_VBN_LIST,	VECTOR[2],	! List head for VBN data
INPUT_PLACE_LEN,	WORD,	! Length of placement data as attribute
INPUT_PADDING_2,	BBLOCK[2],	! *** Padding ***

! Output context.

OUTPUT_BEG,	VECTOR[0],	! Beginning of output context
OUTPUT_CHAN,	LONG,	! Output channel
OUTPUT_FLAGS,	BBLOCK[2],	! Output flag bits
OUTPUT_PADDING,	BBLOCK[2],	! *** Padding ***
OUTPUT_FAB,	REF BBLOCK,	! Pointer to output FAB
OUTPUT_NAM,	REF BBLOCK,	! Pointer to output NAM block
OUTPUT_BCB,	REF BBLOCK,	! Pointer to output BCB
OUTPUT_QUAL,	REF BBLOCK,	! Pointer to output qualifiers block
OUTPUT_BAD,	REF BBLOCK,	! Pointer to output bad block data
OUTPUT_BLOCK,	LONG,	! Current output block number
OUTPUT_MAXBLOCK,	LONG,	! Maximum output block number
OUTPUT_DEVGEOM,	BBLOCK[8],	! Output device geometry
OUTPUT_ATTBUF,	BBLOCK[ATTBUF_LENGTH],	! Values from attribute record
OUTPUT_END,	VECTOR[0],	! End of output context

! Listing context.

LIST_TOTFILES,	LONG,	! Listing -- total files
LIST_TOTSIZE,	LONG,	! Listing -- total size

! Verify and compare context.

VERIFY_FAB,	REF BBLOCK,	! Pointer to verification FAB
VERIFY_USE_COUNT,	LONG,	! Use count for current VERIFY_QUAL
VERIFY_QUAL,	REF BBLOCK,	! Corresponding input qualifier block
COMPARE_BCB,	REF BBLOCK,	! BCB for compare buffer

! File scan context.

FAST_BUFFER,	REF BBLOCK,	! Pointer to index file buffer
FAST_BUFFER_SIZE,	LONG,	! Size of index file buffer
FAST_RVN,	BYTE,	! RVN of current MFD
FAST_PADDING,	BBLOCK[1],	! *** Padding ***
DIR_VERLIMIT,	WORD,	! File version limit
FAST_VOL_BEG,	VECTOR[0],	! Beginning of per-volume information
FAST_IMAP_SIZE,	REF VECTOR,	! Number of blocks in index file bitmap
FAST_IMAP,	REF VECTOR,	! Bitmap of valid and selected files
FAST_HDR_OFFSET,	REF VECTOR,	! VBN offset to file header
FAST_BOOT_LBN,	REF VECTOR,	! LBN of boot file
FAST_VOL_END,	VECTOR[0],	! End of per-volume information

! Journalling context.

JOUR_BUFFER,	REF VECTOR[BYTE],	! Journal buffer
JOUR_DIR,	REF VECTOR[BYTE],	! Current directory string
JOUR_HIBLK,	LONG,	! Highest allocated block
JOUR_EFBLK,	LONG,	! End of file block
JOUR_INBLK,	LONG,	! Current input block
JOUR_FFBYTE,	WORD,	! End of file byte
JOUR_INBYTE,	WORD,	! Current input byte
JOUR_STRUCT_LEV,	WORD,	! Current journal structure level
JOUR_COUNT,	BYTE,	! XOR byte count context
JOUR_REVERSE,	BYTE,	! True if reading backward
JOUR_EXSZ,	WORD,	! Default file extension size.
JOUR_PADDING,	BBLOCK[2],	! *** Padding ***

! Checkpointing context.

CHKPT_HIGH_SP,	LONG,	! High value of SP for saved portion
CHKPT_LOW_SP,	LONG,	! Low value of SP for saved portion
CHKPT_STACK,	REF BBLOCK,	! Pointer to save area for stack
CHKPT_VARS,	REF BBLOCK,	! Pointer to save area for variables
CHKPT_STATUS,	LONG,	! Failure status of a re-access

! Directory scan context.

DIR_BEG,	VECTOR[0],	! Beginning of context
DIR_CHAN,	LONG,	! Channel assigned to device
DIR_NAM,	REF BBLOCK,	! Pointer to name block
DIR_DEV_DESC,	REF BBLOCK,	! Descriptor for device
DIR_SEL_DIR,	BBLOCK[8],	! Descriptor for selection directory
DIR_SEL_NTV,	BBLOCK[8],	! Descriptor for selection n.t.v
DIR_STROCLEV,	BYTE,	! Structure level of directory
DIR_LEVELS,	BYTE,	! Current number of active levels
DIR_FLAGS,	BBLOCK[1],	! Flag bits
DIR_STATUS,	BBLOCK[1],	! Directory status flags
DIR_STRING,	VECTOR[39*8+7+1, BYTE],	! Current directory path
DIR_STACK,	BBLOCK[D_K_NLEVELS*D_S_ENTRY],	! Directory level stack
DIR_SP,	REF BBLOCK,	! Stack pointer for DIR_STACK
DIR_SEL_LATEST,	LONG,	! Latest version selector

```

DIR_END,      VECTOR[0],      ! End of context
DIR_SCANLIMIT, VECTOR[D_K_NLEVELS], ! ODS-1 directory scan limits

```

```
! Standalone ACP context.
```

```

INPUT_MTL,      REF BBLOCK,      ! Pointer to input MTL entry
OUTPUT_MTL,     REF BBLOCK,      ! Pointer to output MTL entry
CURRENT_MTL,    REF BBLOCK,      ! Pointer to MTL for current operation
CURRENT_VCB,    REF BBLOCK,      ! Pointer to VCB for current operation
CURRENT_WCB,    REF BBLOCK,      ! Pointer to WCB for current operation

```

```
! ACL context for file restoration.
```

```

ACL_FIB_DESCR,  BBLOCK [8],      ! FIB descriptor
ACL_FIB,        BBLOCK [FIB$C_LENGTH], ! FIB storage
ACL_LENGTH,     LONG,            ! Size of the entire ACL
ACL_BUFFER,     REF BBLOCK,      ! Pointer to the ACL to save

```

```
! Encrypted Saveset context and work area
```

```

CRYP_IN_CONTEXT, LONG,          ! Pointer to encrypt context for input ss
CRYP_OU_CONTEXT, LONG,          ! Pointer to output encrypt context
CRYP_DA_CONTEXT, LONG,          ! Pointer to datakey encryption context
CRYP_DATA_ENCIV, BBLOCK [8],    ! Copy of IV for encryption of savesets
! Note: the following values are assumed adjacent
CRYP_DATA_CODE,  BBLOCK [4],    ! Saveset encryption algorithm code
CRYP_DATA_KEY,   BBLOCK [8],    ! Saveset data key
CRYP_DATA_IV,    BBLOCK [8],    ! Saveset encryption IV
CRYP_DATA_CHKSM, LONG,          ! Checksum for code, key, and iv
! Note: the preceding values are assumed adjacent

```

```
%:
```

```
MACRO
```

```
! Field definitions for COM_FLAGS.
```

```

COM_EOV=        0,0,1,0 %,      ! Output save medium is at end
COM_STANDALONE= 0,1,1,0 %,      ! This is the standalone version
COM_FILESEEN=   0,2,1,0 %,      ! At least one file processed
COM_VERIFYING=  0,3,1,0 %,      ! Verify pass in progress
COM_FAIL_RSTRT= 0,4,1,0 %,      ! Reel restart failed to find file
COM_CONTINUE=   0,5,1,0 %,      ! Continue despite high error rate
COM_DSBL_CHKPT= 0,6,1,0 %,      ! Checkpoint can not be requested
                                ! since operation is not restartable
COM_DSBL_RSTRT= 0,7,1,0 %,      ! Checkpoint was requested while
                                ! DSBL_CHKPT was set
COM_INTERACT=   0,8,1,0 %;      ! SYSSCOMMAND is a terminal

```

```
MACRO
```

```
! Field definitions for INPUT_FLAGS.
```

```

!
INPUT_OPEN=      0,0,1,0 %,      ! Input file is open
EOV_IN_PROG=     0,1,1,0 %,      ! EOV processing in progress
EOV_SAVING=      0,2,1,0 %,      ! SAVE_BLOCKS in progress
INPUT_ON_RVN=    0,3,1,0 %,      ! Blocks exist on RVN selected by
                                ! /VOLUME qualifier
INPUT_IGNORE_INTE=0,4,1,0 %,      ! Ignore file access conflict
INPUT_SAVE_OK=   0,5,1,0 %,      ! No errors occurred during save
INPUT_WILDSAVE=  0,6,1,0 %,      ! Wild card save set given
INPUT_REWOUND=   0,7,1,0 %,      ! Input tape rewound for wildcards
INPUT_SSFOUND=   0,8,1,0 %,      ! Input save set found (only for wildcards)

```

MACRO

```

! Field definitions for placement data blocks pointed to
! by INPUT_PLACEMENT and for VBN data blocks pointed to
! by INPUT_VBN_LIST. These share fields for commonality
! in reel restart.

```

```

PLC_FLINK=       0,0,32,0 %,      ! Forward link
PLC_BLINK=      4,0,32,0 %,      ! Backward link
PLC_TYPE=        8,0,8,0 %,      ! Type code
PLC_SIZE=        9,0,8,0 %,      ! Size of block
PLC_DATA=       10,0,0,0 %,      ! Data portion
VBN_FIRST=      10,0,32,0 %,      ! First VBN in range
VBN_LAST=       14,0,32,0 %,      ! Last VBN in range

```

LITERAL

```

PLC_S_HDR=       10,              ! Length of placement block header
VBN_S_ENTRY=     18;              ! Length of VBN data block

```

MACRO

```

! Field definitions for OUTPUT_FLAGS.

```

```

OUTPUT_OPEN=     0,0,1,0 %,      ! Output file is open
OUTPUT_V_FIRST=  0,1,1,0 %,      ! Verifying first block
OUTPUT_V_LAST=   0,2,1,0 %,      ! Verifying last block
OUTPUT_INCR_DIR= 0,3,1,0 %,      ! Special /INCREMENTAL directory scan
OUTPUT_IMPLICIT= 0,4,1,0 %,      ! Output file implicitly opened
ACL_ERROR=       0,5,1,0 %,      ! Error occurred processing ACL
ACL_FIRST_TIME=  0,6,1,0 %,      ! First time through processing ACL
OUTPUT_DIR_OPEN= 0,7,1,0 %,      ! An output directory is open
OUTPUT_NONSEQ_IN=1,0,1,0 %,      ! Input saveset is not sequential

```

MACRO

```

! Field definitions for OUTPUT_DEVGEOM.

```

```

DGM_SECTORS=     0,0,8,0 %,      ! DIB$B_SECTORS
DGM_TRACKS=      1,0,8,0 %,      ! DIB$B_TRACKS
DGM_CYLINDERS=   2,0,16,0 %,     ! DIB$W_CYLINDER

```

DGM_MAXBLOCK= 4,0,32,0 %; ! DIB\$L_MAXBLOCK

MACRO

! Field definitions for DIR_FLAGS.

D_IMAGE_SCAN=	0,0,1,0 %;	! Image mode scan
D_SCAN_FAILED=	0,1,1,0 %;	! Previous scan failed
D_HARD_STOP=	0,2,1,0 %;	! Immediate return on terminator
D_SCANNED_DIRS=	0,3,1,0 %;	! Return scanned directories
D_INITIAL=	0,4,1,0 %;	! Initial call
D_ROOT_MFD=	0,5,1,0 %;	! Scan root MFD only

MACRO

! Field definitions for DIR_STATUS.

D_STAT_VALID=	0,0,1,0 %;	! DIR_STATUS is valid
D_STAT_DIR_SEL=	0,1,1,0 %;	! Directory is selected
D_STAT_SCANNED=	0,2,1,0 %;	! Directory was scanned
D_STAT_FILE_SEL=	0,3,1,0 %;	! Files in directory are selected

0009 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

0010 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY