



SSSSSSSS	TTTTTTTTT	AAAAAA	RRRRRRRR	DDDDDDDD	EEEEEEEEEE	FFFFFFFFFF	QQQQQQ	ZZZZZZZZZ	
SSSSSSSS	TTTTTTTTT	AAAAAA	RRRRRRRR	DDDDDDDD	EEEEEEEEEE	FFFFFFFFFF	QQQQQQ	ZZZZZZZZZ	
SS	TT	AA	RR	DD	EE	FF	QQ	ZZ	
SS	TT	AA	RR	DD	EE	FF	QQ	ZZ	
SS	TT	AA	RR	DD	EE	FF	QQ	ZZ	
SS	TT	AA	RR	DD	EE	FF	QQ	ZZ	
SSSSSS	TT	AA	RRRRRRRR	DD	EEEEEEEE	FFFFFFFF	QQ	ZZ	
SSSSSS	TT	AA	RRRRRRRR	DD	EEEEEEEE	FFFFFFFF	QQ	ZZ	
	TT	AAAAAAAAA	RR	DD	EE	FF	QQ	ZZ	
	TT	AAAAAAAAA	RR	DD	EE	FF	QQ	ZZ	
	TT	AA	RR	DD	EE	FF	QQ	ZZ	
	TT	AA	RR	DD	EE	FF	QQ	ZZ	
	TT	AA	RR	DD	EE	FF	QQ	ZZ	
	TT	AA	RR	DD	EE	FF	QQ	ZZ	
SSSSSSSS	TT	AA	RR	DD	EE	FF	QQ	ZZ	
SSSSSSSS	TT	AA	RR	DD	EE	FF	QQ	ZZ	
				DDDDDDDD	EEEEEEEEEE	FFFFFFFF	QQQQ	ZZZZZZZZZ	....
				DDDDDDDD	EEEEEEEEEE	FFFFFFFF	QQQQ	ZZZZZZZZZ	....

SSSSSSSS	DDDDDDDD	LL
SSSSSSSS	DDDDDDDD	LL
SS	DD	DD
SS	DD	DD
SS	DD	DD
SS	DD	DD
SSSSSS	DD	DD
SSSSSS	DD	DD
	DD	DD
	DD	DD
	DD	DD
	DD	DD
SSSSSSSS	DDDDDDDD	LLLLLLLLLL
SSSSSSSS	DDDDDDDD	LLLLLLLLLL

{ STARDEFQZ.SDL - system user interface 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: VAX/VMS System Macro Libraries

{ ABSTRACT:  
{ This file contains the SDL source for all user visible operating  
{ system interfaces from Q to Z.

{ ENVIRONMENT:  
{ n/a

```

{--
{ AUTHOR: The VMS Group          CREATION DATE: 1-Aug-1976
{ MODIFIED BY:
{ V03-072 CWH3072      CW Hobbs      24-Jul-1984
{           Add WS_OPA0 parameter, workstation using QVSS for console.
{ V03-071 CDS0001      Christian D. Saether  20-Jul-1984
{           Add ACP_DINDXCACHE and ACP_REBLDSYSD parameters.
{ V03-070 ACG0436      Andrew C. Goldstein, 12-Jul-1984 12:02
{           Add LGI_BRK_TERM and LGI_BRK_DISUSER parameters

```

V03-069 JAK0218 J A Krycka 10-Jul-1984  
Add BUFFER\_COUNT and EXTEND\_QUANTITY item codes to \$\$JCDEF.  
Add RESERVED\_xxx item and function codes to \$\$JCDEF.  
Add RESERVED\_xxx item and function codes to \$QUIDEF.

V03-068 WHM0001 Bill Matthews 04-May-1984  
Remove USESYSPARAMS getsyi item.

V03-067 JAK0204 J A Krycka 20-Apr-1984  
Add the following \$GETQUI flags: QUI\$\_FILE\_PAGINATE\_EXP,  
QUI\$\_JOB\_FILE\_PAGINATE, QUI\$\_JOB\_FILE\_PAGINATE\_EXP, and  
QUI\$\_QUEUE\_FILE\_PAGINATE.

V03-066 JAK0203 J A Krycka 17-Apr-1984  
Add ACCOUNT\_NAME, UIC, and USERNAME input item codes to \$\$JCDEF  
which correspond to existing output item codes in \$QUIDEF. Also  
declare USER\_IDENTIFICATION to be a code reserved for internal  
use by Digital so that it can be retired after all utilities  
have been converted to use the three new item codes.

V03-065 MIR0420 Michael I. Rosenblum 12-Apr-1984  
Add sysgen parameter tty\_defport to sytables.  
Add readverify picture string definitions to \$TTDEF

V03-064 WHM0001 Bill Matthews 08-Apr-1984  
Replaced SCSNODEL and SCSNODEH with SCSNODE.  
Replaced DISK\_QUORUM1-4 with DISK\_QUORUM.  
Added SYSBOOT parameters USESYSPARAMS and WRITESYSPARAMS

V03-063 MED0069 Ellen M. Dusseault 06-Apr-1984  
Add new terminal device's name, LA84, to TTDEF.

V03-062 RAS0281 Ron Schaefer 28-Mar-1984  
Add RMS\_DFNBC (network block count) GETSYI item.

V03-061 JEJ0014 J E Johnson 25-Mar-1984  
Add RMS\_GBLBUFQUO (global buffer quota) GETSYI item.

V03-060 WMC0060 Wayne Cardoza 24-Mar-1984  
Add ACP\_XQP\_RES GETSYI item.

V03-059 WMC0059 Wayne Cardoza 7-Mar-1984  
Add SEC\$x\_RESIDENT.

V03-058 EMD0050 Ellen M. Dusseault 28-Feb-1984  
Add new terminal name, PRO\_SERIES, to TTDEF and a  
new terminal characteristic, DEC\_CRT2 to TT2DEF.

V03-057 MMD0240 Meg Dumont, 24-Feb-1984 10:55  
GETSYI item for LOADMTACCESS. Installation loadable  
mag tape access routine.

V03-056 TMK0004 Todd M. Katz 02-Feb-1984  
Change the name of SYI\$\_PAPORTPOLL to SYI\$\_PANOPOLL.

V03-055 TMK0003 Todd M. Katz 01-Feb-1984

Change the name of SYIS\_PAPOLL to SYIS\_PAPORTPOLL.

V03-054 WMC0054 Wayne Cardoza 31-Jan-1984  
GETSYI items for emulated instruction classes.

V03-053 TMK0002 Todd M. Katz 31-Jan-1984  
Add SYIS\_PE1, SYIS\_PE2, SYIS\_PE3, SYIS\_PE4, SYIS\_PE5, SYIS\_PE6, and  
SYIS\_PAPOLL to \$\$YIDEF.

V03-052 ACG0392 Andrew C. Goldstein, 30-Jan-1984 12:19  
Add UIC\$K\_MATCH\_ALL identifier

V03-051 JLV0326 Jake VanNoy 11-JAN-1984  
Add TTY\_TIMEOUT and TTY\_AUTOCHAR to SYI definitions.

V03-050 MIRO300 Michael I. Rosenblum 10-Jan-1984  
add definition for BTS terminals and make a note  
that the negitive terminal device types are reserved  
for the RTL foreign terminal support.  
Move the definition for VT200 out of the negitive range

V03-049 WMC0048 Wayne Cardoza 01-Jan-1984  
SYIDEF codes for page and swap file data.

V03-048 LMP0177 L. Mark Pilant, 7-Dec-1983 12:29  
Add CLASS\_PROT to \$\$YIDEF to support non-discretionary  
classification checking.

V03-047 SSA0003 Stan Amway 5-Dec-1983  
Added DORMANTWAIT to \$\$YIDEF to support outswap scheduling  
changes.

V03-046 DWT0151 David W. Thiel 15-Nov-1983  
Add SYSGEN parameters LOCKDIRWT and QDSKVOTES.  
Undefine VAXCLUSTER bit-field and define VAXCLUSTER  
as a byte.

V03-045 TMK0001 Todd M. Katz 26-Oct-1983  
Add SYIS\_PQL\_DJTQUOTA and SYIS\_PQL\_MJTQUOTA to \$\$YIDEF.  
These item list codes represent the sysgen parameters  
PQL\_DJTQUOTA and PQL\_MJTQUOTA.

V03-044 KFH0006 Ken Henderson 19 Sep 1983  
Re-add ALLOCLASS and LGI\_security param  
into \$\$YIDEF. They were SYSGEN parameters.

V03-043 KFH0005 Ken Henderson 9 Sep 1983  
Remove SYIS\_SERIAL, MFGPLANT, HWREVISION,  
ALLOCLASS and LGI\_security params from  
\$\$YIDEF.

V03-042 MIRO085 Michael I. Rosenblum 26-Aug-1983  
Re-insert definitions for DCL\_OUTBND and DCL\_CTRLC  
in TT2DEF.

V03-041 MLJ0118 Martin L. Jack 25-Aug-1983

Update spellings in \$\$JCDEF and \$QUIDEF per documentation,  
add PAGE\_SETUP.

V03-040 MIR0082 Michael I. Rosenblum 19-Aug-1983  
Add TT2\$V\_SYSPWD to TT2DEF.

V03-039 MLJ0116 Martin L. Jack 17-Aug-1983  
Update \$USGDEF for long filenames.

V03-038 MLJ0115 Martin L. Jack 30-Jul-1983  
Update \$\$JCDEF and \$QUIDEF.

V03-037 GAS0162 Gerry Smith 30-Jul-1983  
Add LGI\_PWD\_TMO to \$\$SYIDEF.

V03-036 RAS0179 Ron Schaefer 29-Jul-1983  
Remove LOGxHASHTBL sysgen params from \$\$SYIDEF.

V03-035 JLV0282 Jake VanNoy 28-JUL-1983  
Remove TRMS\_VALUESTR.

V03-034 KFH0004 Ken Henderson 28 Jul 1983  
Add MAXQUEPRI, DEFQUEPRI, QDSKINTERVAL in \$\$SYIDEF.  
Remove JOBQUEUES, REINITQUE, MAXPRINTSYMB from \$\$SYIDEF.  
Change VMS CLUSTER to VAXCLUSTER in \$\$SYIDEF.  
Remove <RSD> section from \$\$SYIDEF.

V03-033 NPK3029 N. Kronenberg 27-Jul-1983  
Remove PASTRETRY, add PAMAXPORT, PASANITY in \$\$SYIDEF.

V03-032 MSH0001 Maryann Hinden 11-Jul-1983  
Add SYIS\_DISK\_QUORUMx and SYIS\_SCS\_EXISTS.

V03-031 LMPBUILD L. Mark Pilant, 24-Jun-1983 14:48  
Change comment characters '/' to 'C' as SDL doesn't like the  
former type outside a module definition.

V03-030 GAS0142 Gerry Smith 23-Jun-1983  
Add login security parameters, and ALLOCLASS, to SYIDEF.

V03-029 MLJ0114 Martin L. Jack 22-Jun-1983  
Add \$QUIDEF and a few \$\$JCDEF items.

V03-028 MLJ0113 Martin L. Jack 26-May-1983  
Change spelling of a few \$\$JCDEF items.

V03-027 WMC0027 Wayne Cardoza 24-May-1983  
Add execute bit to SECDEF

V03-026 MIR0051 Michael I. Rosenblum 24-May-1983  
Add Psthru bit to TT2DEF.

V03-025 MLJ0113 Martin L. Jack 21-May-1983  
Removed definition of on-disk queue file format. This is not  
a public interface.

V03-024 KFH0003 Ken Henderson 21 May 1983  
Added cluster, uVAX and JNL items to \$\$SYIDEF

V03-023 MIR0050 Michael I. Rosenblum 07-May-1983  
Add VT200 support and Altdispar bit.

V03-022 EAD0022 Elliott A. Drayton 1-May-1983  
Removed one of two definitions in \$\$SYIDEF for STARTUP\_P3.

V03-021 RKS0021 RICK SPITZ 29-APR-1983  
ADD NEW TERMINAL CHARACTERISTICS

V03-020 MLJ0112 Martin L. Jack 27-Apr-1983  
Update \$\$JCDEF for job controller baselevel.

V03-019 PCA1015 Paul C. Anagnostopoulos 27-Apr-1983  
Add sysgen parameters TAILORED and STARTUP\_Pn to \$\$SYIDEF.  
Add sysgen parameter LOCKIDTBL\_MAX for Steve Beckhardt.

V03-018 MLJ0109 Martin L. Jack, 14-Apr-1983 14:14  
Update \$\$JCDEF for job controller baselevel.

V03-017 MIR0031 Michael I. Rosenblum 1-Apr-1983  
Add bits to TTDEF to include user specified frame  
size, number or stop bits, and parity handling

V03-016 MIR0030 Michael I. Rosenblum 29-Mar-1983  
Add modifiers for Read with verification.

V03-015 MIR0029 Michael I. Rosenblum 25-Mar-1983  
Add Applications mode keypad bit. Remove unused  
EXTEDIT bit from \$TRMDEF

V03-014 KFH0002 Ken Henderson 8 Mar 1983  
Added BOOTTIME to \$\$SYIDEF.

V03-013 ROW0167 Ralph O. Weber 2-MAR-1983  
Add connection manager parameters QUORUM, VOTES, and  
RECNXINTERVAL.

V03-012 MIR1026 Michael I. Rosenblum 01-Mar-1983  
Fix typo in edit MIR0026.

V03-011 MIR0026 Michael I. Rosenblum 01-Mar-1983  
Add bit definitions for TRMSV\_TM\_NOEDIT and TRMSM\_TM\_NORECALL  
to allow the user to specify explicitly inhibit editing  
and command recall. Also add bit to enable advanced editing.

V03-010 MLJ0106 Martin L. Jack, 01-Mar-1983  
Add \$\$JCDEF items corresponding to job controller baselevel.

V03-009 KFH0001 Ken Henderson 15 Feb 1983  
Added all SYSBOOT parameters to \$\$SYIDEF and also  
defined OLDVERSION, OLDSID, OLDCPU for compatibility.

V03-008 MIR0025 Michael I. Rosenblum 03-Feb-1983

Add \$TRMDEF to be the new bit definitions for the item list QIO's.

V03-007 DMW SYSTEMBUILD DMWalp 25-Jan-1982  
Hardcoded SYISC SFWTYPE@8 as 256 in SYIDEF to get around SDL problem. This packet will be removed when the SDL problem is fixed!!!!!!!!!!!!!!!!!!!!

V03-006 MLJ0103 Martin L. Jack, 7-Jan-1983 15:50  
Add \$\$JCDEF items corresponding to job controller baselevel.

V03-005 ACG0307 Andrew C. Goldstein, 30-Dec-1982 16:42  
Add UIC format definition

V03-004 ACG0303 Andrew C. Goldstein, 9-Dec-1982 16:06  
Add FILL attribute to extraneous field names

V03-003 MLJ0097 Martin L. Jack, 9-Sep-1982 16:52  
Add \$\$JCDEF.

V03-002 MMD0001 Meg Dumont, 5-Apr-1982 14:49  
Add \$XWDEF for XWDIVER

V03-001 RKS0001 RICK SPITZ 29-MAR-1982  
ADD SPECIAL TT2 CHARACTERISTICS FOR DCL SPAWN



```
module $QUIDEF;
```

```
/* Get Queue Information Service
```

```
/*
/* NOTE: This service and corresponding symbol definitions is intentionally NOT
/* being documented for VMS V4.0 and is supported only for internal Digital
/* component use. Symbol definitions and the size of strings returned, for
/* example, may change in a subsequent release or update of VMS. Beware!
```

```
/* NOTE: New items must always be added at the end so users will not have to
/* relink.
```

```
/* Function codes
```

```
constant (
    CANCEL_OPERATION          /* Cancel a wildcard operation
    DISPLAY_CHARACTERISTIC   /* Return characteristic attributes
    DISPLAY_FILE             /* Return file attributes
    DISPLAY_FORM            /* Return form attributes
    DISPLAY_JOB             /* Return job attributes
    DISPLAY_QUEUE          /* Return queue attributes
    TRANSLATE_QUEUE        /* Validate and translate queue name
    RESERVED_FUNC_1       /* Reserved for Digital use (name may change)
    RESERVED_FUNC_2       /* Reserved for Digital use (name may change)
) equals 1 increment T prefix QUI tag $;
```

```
/* Item codes
```

```
constant (
    ACCOUNT_NAME           /* Job: Submitter's account name
    AFTER_TIME            /* Job: /AFTER=time
    ASSIGNED_QUEUE_NAME   /* Queue: ASSIGN/QUEUE target
    BASE_PRIORITY         /* Queue: /BASE_PRIORITY=n
    CHARACTERISTIC_NAME   /* Characteristic: Name
    CHARACTERISTIC_NUMBER /* Characteristic: Number
    CHARACTERISTICS       /* Job, queue: /CHARACTERISTICS=(c,...)
    CHECKPOINT_DATA      /* Job: Checkpoint data
    CLI                  /* Job: /CLI=filename
    COMPLETED_BLOCKS    /* Job: Completed blocks including checkpoint
    CONDITION_VECTOR     /* Job: Completion status
    CPU_DEFAULT          /* Queue: /CPUDEFAULT=t
    CPU_LIMIT            /* Job, queue: /CPUMAXIMUM=t
    DEVICE_NAME          /* Queue: /ON=device
    ENTRY_NUMBER         /* Job: Entry number
    FILE_COPIES          /* File: /COPIES=n
    FILE_COPIES_CHKPT   /* File: File copies checkpoint
    FILE_COPIES_DONE    /* File: File copies completed
    FILE_FLAGS           /* File: Boolean information
    FILE_SETUP_MODULES  /* File: /SETUP=(module,...)
    FILE_SPECIFICATION  /* File: Full file specification
    FILE_STATUS         /* File: Status information
    FIRST_PAGE          /* File: /PAGES=(n,'')
    FORM_DESCRIPTION     /* Form: /DESCRIPTION=string
    FORM_FLAGS          /* Form: Boolean information
    FORM_LENGTH         /* Form: /LENGTH=n
    FORM_MARGIN_BOTTOM /* Form: /MARGIN=BOTTOM=n
```

```

FORM_MARGIN_LEFT      /* Form: /MARGIN=LEFT=n
FORM_MARGIN_RIGHT     /* Form: /MARGIN=RIGHT=n
FORM_MARGIN_TOP       /* Form: /MARGIN=TOP=n
FORM_NAME             /* Form, job, queue: Form name
FORM_NUMBER           /* Form: Number
FORM_SETUP_MODULES    /* Form: /SETUP=(module,...)
FORM_STOCK            /* Form: /STOCK=stock-name
FORM_WIDTH            /* Form: /WIDTH=n
GENERIC_TARGET        /* Queue: /GENERIC=(queue-name,...)
INTERVENING_BLOCKS   /* Job: Intervening pending blocks
INTERVENING_JOBS     /* Job: Intervening pending jobs
JOB_COPIES            /* Job: /JOB_COUNT=n
JOB_COPIES_CHKPT      /* Job: Job copies checkpoint
JOB_COPIES_DONE       /* Job: Job copies completed
JOB_FLAGS             /* Job: Boolean information
JOB_LIMIT             /* Queue: /JOB_LIMIT=n
JOB_NAME              /* Job: Name
JOB_RESET_MODULES     /* Queue: /SEPARATE=RESET=(module,...)
JOB_SIZE              /* Job: Total blocks in job
JOB_SIZE_MAXIMUM      /* Queue: /BLOCK_LIMIT=n
JOB_SIZE_MINIMUM      /* Queue: /BLOCK_LIMIT=(n,'')
JOB_STATUSES          /* Job: Status information
LAST_PAGE            /* File: /PAGES=n
LIBRARY_SPECIFICATION /* Queue: /LIBRARY=file-specification
LOG_QUEUE            /* Job: /PRINTER=queue-name
LOG_SPECIFICATION     /* Job: /LOG_FILE=file-specification
NOTE                 /* Job: /NOTE=string
OPERATOR_REQUEST      /* Job: /OPERATOR=string
OWNER_UIC            /* Queue: /OWNER=uic
PAGE_SETUP_MODULES    /* Form: /PAGE_SETUP=(module,...)
PARAMETER_1          /* Job: /PARAMETER=string
PARAMETER_2
PARAMETER_3
PARAMETER_4
PARAMETER_5
PARAMETER_6
PARAMETER_7
PARAMETER_8
PRIORITY             /* Job: /PRIORITY=n
PROCESSOR            /* Queue: /PROCESSOR=filename
PROTECTION           /* Queue: /PROTECTION=mask
QUEUE_FLAGS          /* Queue: Boolean information
QUEUE_NAME           /* Job, queue: Queue name
QUEUE_STATUS         /* Queue: Status information
REFUSAL_REASON       /* Job: Reason symbiont refused job
REQUEUE_PRIORITY     /* Job: Priority after requeue
REQUEUE_QUEUE_NAME   /* Job: Queue after requeue
SCSNODE_NAME         /* Queue: /ON=node::
SEARCH_FLAGS         /* Flags to control search
SEARCH_NAME          /* Object name to search for
SEARCH_NUMBER        /* Object number to search for
SUBMISSION_TIME      /* Job: Submission time
UIC                 /* Job: Submitter's UIC
USERNAME             /* Job: Submitter's username
WSDEFAULT            /* Job, queue: /WSDEFAULT=n
WSEXTENT             /* Job, queue: /WSEXTENT=n

```

```

      WSQUOTA                /* Job, queue: /WSQUOTA=n
      RESERVED_BOOLEAN_1    /* Reserved for Digital use (name may change)
      RESERVED_BOOLEAN_2    /* Reserved for Digital use (name may change)
      RESERVED_INPUT_1      /* Reserved for Digital use (name may change)
      RESERVED_INPUT_2      /* Reserved for Digital use (name may change)
      RESERVED_OUTPUT_1     /* Reserved for Digital use (name may change)
      RESERVED_OUTPUT_2     /* Reserved for Digital use (name may change)
      RESERVED_OUTPUT_3     /* Reserved for Digital use (name may change)
      RESERVED_OUTPUT_4     /* Reserved for Digital use (name may change)
      RESERVED_OUTPUT_5     /* Reserved for Digital use (name may change)
      RESERVED_OUTPUT_6     /* Reserved for Digital use (name may change)
) equals 1 increment 1 prefix QUI tag $;
/*
/* Structure of FILE_FLAGS item.
/*
aggregate FILE_FLAGS structure fill prefix QUI$;
      FILE_BURST             bitfield mask; /* /BURST
      FILE_BURST_EXP         bitfield mask; /* /[NO]BURST explicit
      FILE_DELETE           bitfield mask; /* /DELETE
      FILE_DOUBLE_SPACE     bitfield mask; /* /SPACE
      FILE_FLAG              bitfield mask; /* /FLAG
      FILE_FLAG_EXP         bitfield mask; /* /[NO]FLAG explicit
      FILE_TRAILER          bitfield mask; /* /TRAILER
      FILE_TRAILER_EXP      bitfield mask; /* /[NO]TRAILER explicit
      FILE_PAGE_HEADER      bitfield mask; /* /HEADER
      FILE_PAGINATE         bitfield mask; /* /FEED
      FILE_PASSALL          bitfield mask; /* /PASSALL
      FILE_PAGINATE_EXP     bitfield mask; /* /[NO]FEED explicit
      filler                bitfield length 32-^ fill;
end;
/*
/* Structure of FILE_STATUS item.
/*
aggregate FILE_STATUS structure fill prefix QUI$;
      FILE_CHECKPOINTED     bitfield mask; /* File checkpointed
      FILE_EXECUTING        bitfield mask; /* File currently executing
      filler                bitfield length 32-^ fill;
end;
/*
/* Structure of FORM_FLAGS item.
/*
aggregate FORM_FLAGS structure fill prefix QUI$;
      FORM_SHEET_FEED       bitfield mask; /* /SHEET FEED
      FORM_TRUNCATE         bitfield mask; /* /TRUNCATE
      FORM_WRAP             bitfield mask; /* /WRAP
      filler                bitfield length 32-^ fill;
end;
/*
/* Structure of JOB_FLAGS item.
/*
aggregate JOB_FLAGS structure fill prefix QUI$;
      JOB_CPU_LIMIT        bitfield mask; /* /CPU TIME explicit
      JOB_FILE_BURST       bitfield mask; /* /BURST=ALL
      JOB_FILE_BURST_ONE   bitfield mask; /* /BURST=ONE
      JOB_FILE_BURST_EXP   bitfield mask; /* /[NO]BURST explicit
      JOB_FILE_FLAG        bitfield mask; /* /FLAG=ALL

```

```

JOB_FILE_FLAG_ONE      bitfield mask; /* /FLAG=ONE
JOB_FILE_FLAG_EXP      bitfield mask; /* /[NO]FLAG explicit
JOB_FILE_TRAILER       bitfield mask; /* /TRAILER=ALL
JOB_FILE_TRAILER_ONE   bitfield mask; /* /TRAILER=ONE
JOB_FILE_TRAILER_EXP   bitfield mask; /* /[NO]TRAILER explicit
JOB_LOG_DELETE         bitfield mask; /* /NOKEEP
JOB_LOG_NULL           bitfield mask; /* /NOLOG FILE
JOB_LOG_SPOOL          bitfield mask; /* /PRINTER
JOB_LOWERCASE          bitfield mask; /* /LOWERCASE
JOB_NOTIFY             bitfield mask; /* /NOTIFY
JOB_RESTART            bitfield mask; /* /RESTART
JOB_WSDEFAULT          bitfield mask; /* /WSDEFAULT explicit
JOB_WSEXTENT           bitfield mask; /* /WSEXTENT explicit
JOB_WSQUOTA            bitfield mask; /* /WSQUOTA explicit
JOB_FILE_PAGINATE      bitfield mask; /* /FEED
JOB_FILE_PAGINATE_EXP bitfield mask; /* /[NO]FEED explicit
filter                 bitfield length 32-^ fill;

```

end;

/\*

/\* Structure of JOB\_STATUS item.

/\*

aggregate JOB STATUS structure fill prefix QUI\$:

```

JOB_ABORTING          bitfield mask; /* Job is aborting
JOB_EXECUTING         bitfield mask; /* Job is executing
JOB_HOLDING           bitfield mask; /* Job is holding for /HOLD
JOB_INACCESSIBLE      bitfield mask; /* Job is inaccessible
JOB_REFUSED           bitfield mask; /* Job was refused by symbiont
JOB_REQUEUE           bitfield mask; /* Job will requeue after abort
JOB_RESTARTING        bitfield mask; /* Job started execution at least once
JOB_RETAINED          bitfield mask; /* Job was retained by /RETAIN
JOB_STARTING          bitfield mask; /* Job is starting
JOB_TIMED             bitfield mask; /* Job is holding for /AFTER
filter                 bitfield length 32-^ fill;

```

end;

/\*

/\* Structure of QUEUE\_FLAGS item.

/\*

aggregate QUEUE FLAGS structure fill prefix QUI\$:

```

QUEUE_BATCH           bitfield mask; /* /BATCH
QUEUE_CPU_DEFAULT     bitfield mask; /* /CPUDEFAULT specified
QUEUE_CPU_LIMIT       bitfield mask; /* /CPUMAXIMUM specified
QUEUE_FILE_BURST      bitfield mask; /* /DEFAULT=BURST=ALL
QUEUE_FILE_BURST_ONE  bitfield mask; /* /DEFAULT=BURST=ONE
QUEUE_FILE_FLAG       bitfield mask; /* /DEFAULT=FLAG=ALL
QUEUE_FILE_FLAG_ONE   bitfield mask; /* /DEFAULT=FLAG=ONE
QUEUE_FILE_TRAILER    bitfield mask; /* /DEFAULT=TRAILER=ALL
QUEUE_FILE_TRAILER_ONE bitfield mask; /* /DEFAULT=TRAILER=ONE
QUEUE_GENERIC         bitfield mask; /* /GENERIC
QUEUE_GENERIC_SELECTION bitfield mask; /* /ENABLE_GENERIC
QUEUE_JOB_BURST       bitfield mask; /* /SEPARATE=BURST
QUEUE_JOB_FLAG        bitfield mask; /* /SEPARATE=FLAG
QUEUE_JOB_SIZE_SCHED  bitfield mask; /* /SCHEDULE=SIZE
QUEUE_JOB_TRAILER     bitfield mask; /* /SEPARATE=TRAILER
QUEUE_RETAIN_ALL      bitfield mask; /* /RETAIN=ALL
QUEUE_RETAIN_ERROR    bitfield mask; /* /RETAIN=ERROR
QUEUE_SWAP            bitfield mask; /* /NODISABLE_SWAPPING

```

```

    QUEUE_TERMINAL          bitfield mask; /* /TERMINAL or terminal device
    QUEUE_WSDEFAULT         bitfield mask; /* /WSDEFAULT specified
    QUEUE_WSEXTENT          bitfield mask; /* /WSEXTENT specified
    QUEUE_WSQUOTA           bitfield mask; /* /WSQUOTA specified
    QUEUE_FILE_PAGINATE     bitfield mask; /* /DEFAULT=FEED
    filler                  bitfield length 32-^ fill;
end;
/*
/* Structure of QUEUE_STATUS item.
/*
aggregate QUEUE_STATUS structure fill prefix QUI$;
    QUEUE_ALIGNING         bitfield mask; /* Queue is aligning
    QUEUE_IDLE              bitfield mask; /* Queue is idle
    QUEUE_LOWERCASE        bitfield mask; /* Lowercase device
    QUEUE_OPERATOR_REQUEST bitfield mask; /* Queue is doing /OPERATOR
    QUEUE_PAUSED            bitfield mask; /* Queue is paused
    QUEUE_PAUSING           bitfield mask; /* Queue is pausing
    QUEUE_REMOTE            bitfield mask; /* Remote device
    QUEUE_RESETTING         bitfield mask; /* Incomplete remote request to reset
    QUEUE_RESUMING          bitfield mask; /* Queue is resuming
    QUEUE_SERVER            bitfield mask; /* Server symbiont
    QUEUE_STALLED           bitfield mask; /* Device is stalled
    QUEUE_STARTING          bitfield mask; /* Queue is starting
    QUEUE_STOPPED           bitfield mask; /* Queue is stopped
    QUEUE_STOPPING          bitfield mask; /* Queue is stopping
    QUEUE_UNAVAILABLE       bitfield mask; /* Device is unavailable
    filler                  bitfield length 32-^ fill;
end;
/*
/* Definition of SEARCH_FLAGS
/*
aggregate SEARCH_FLAGS structure fill prefix QUI$;
    SEARCH_ALL_JOBS         bitfield mask; /* Select all jobs (else only requesting user's)
    SEARCH_WILDCARD         bitfield mask; /* Force a wildcard operation
    SEARCH_BATCH            bitfield mask; /* Select batch queues
    SEARCH_SYMBIONT         bitfield mask; /* Select symbiont queues
    filler                  bitfield length 32-^ fill;
end;
end_module $QUIDEF;
```

```
module $SBKDEF;
```

```
/*+
/*
/* Statistics block. This block is used to report various data regarding
/* an open file to the requesting program.
/*
/*-
```

```
/*
/* The first two longwords are presented in inverted form for compatibility
/* with the old RSX-11 statistics block.
/*
```

```
aggregate SBKDEF structure prefix SBK$;
```

```
  STLBN_OVERLAY union fill;
    STLBN longword unsigned;          /* starting LBN if contiguous
```

```
    STLBN_FIELDS structure fill;
      STLBNH word unsigned;          /* high order word
      STLBNL word unsigned;          /* low order word
```

```
    end STLBN_FIELDS;
```

```
  end STLBN_OVERLAY;
```

```
  FILESIZE_OVERLAY union fill;
```

```
    FILESIZE longword unsigned;      /* file size
```

```
    FILESIZE_FIELDS structure fill;
      FILESIZEH word unsigned;       /* high order word
      FILESIZE_L word unsigned;      /* low order word
```

```
/*
/* The following two fields are for RSX-11 compatibility
/*
```

```
  end FILESIZE_FIELDS;
```

```
  end FILESIZE_OVERLAY;
```

```
  ACNT byte unsigned;                /* low byte of access count
  LCNT byte unsigned;                /* low byte of lock count
```

```
  FCB longword unsigned;             /* address of file control block
```

```
  FILL_1 word fill prefix SBKDEF tag $$; /* spare
```

```
  ACNT word unsigned;                /* access count
```

```
  LCNT word unsigned;                /* lock count (against writers)
```

```
  WCNT word unsigned;                /* writer count
```

```
  TCNT word unsigned;                /* truncate lock count
```

```
  READS longword unsigned;           /* count of reads executed on channel
```

```
  WRITES longword unsigned;          /* count of writes executed on channel
```

```
  constant 'LENGTH' equals . prefix SBK$ tag K;
```

```
  constant 'LENGTH' equals . prefix SBK$ tag C;
```

```
end SBKDEF;
```

```
end_module $SBKDEF;
```

ST/

MOD

/\*+

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*-

COR

COR

COR

/\*

/\*

/\*

/\*

COR

```
module $SECDEF;
```

```
/*+
/* PROCESS OR GLOBAL SECTION DEFINITIONS
/*-
```

```
/*
/* ***** L VBN, L WINDOW, AND B PFC MUST BE THE SAME OFFSET VALUES AS THE
/* ***** EQUIVALENTLY NAMED OFFSETS IN $PFLDEF
/*
```

```
aggregate SECDEF structure prefix SEC$;
```

```
  GSD_OVERLAY union fill;
    GSD longword unsigned; /*GLOBAL SECTION DESCRIPTOR ADDRESS
    CCB longword unsigned; /*CHANNEL CONTROL BLOCK ADDRESS IF PROCESS SECTION
  end GSD_OVERLAY;
  SECXFL word unsigned; /*SECTION INDEX FORWARD LINK
  SECXBL word unsigned; /*SECTION INDEX BACKWARD LINK
  VPXPFC_OVERLAY union fill;
    VPXPFC longword unsigned; /*START VIRT PAGE INDEX & PFC
    VPXPFC_BITS structure fill;
      VPX bitfield length 22; /* STARTING VIRTUAL PAGE INDEX
      FILL 1 bitfield length 2 fill prefix SECDEF tag $$; /* UNUSED
      PFC bitfield length 8; /* PAGE FAULT CLUSTER
    end VPXPFC_BITS;
    VPXPFC_FIELDS structure fill;
      FILL 5 byte dimension 3 fill prefix SECDEF tag $$;
      PFC byte unsigned; /*PAGE FAULT CLUSTER
    end VPXPFC_FIELDS;
  end VPXPFC_OVERLAY;
  WINDOW longword unsigned; /*WINDOW CONTROL BLOCK
  VBN longword unsigned; /*BASE VIRTUAL BLOCK NUMBER
  FLAGS_OVERLAY union fill;
    FLAGS word unsigned; /*VARIOUS CONTROL FLAGS
    FLAGS_BITS structure fill;
      GBL bitfield mask; /* GLOBAL SECTION
      CRF bitfield mask; /* COPY ON REFERENCE
      DZRO bitfield mask; /* DEMAND ZERO
      WRT bitfield mask; /* WRITABLE
      SHMGS bitfield mask; /* SHARED MEMORY GLOBAL SECTION
      FILL 2 bitfield fill prefix SECDEF tag $$; /* SPARE
      WRTMOD bitfield mask length 2; /* WRITE ACCESS MODE FOR SECTION
      AMOD bitfield mask length 2; /* ACCESS MODE OF SECTION
      FILL 3 bitfield length 3 fill prefix SECDEF tag $$; /* SPARE
      RESIDENT bitfield mask; /* RESIDENT GLOBAL
      PERM bitfield mask; /* PERMANENT SECTION
      SYSGBL bitfield mask; /* 1 = SYSTEM GLOBAL, 0 = GROUP GLOBAL
```

```
/*
/* ***** THE FOLLOWING FLAGS ARE INDICATORS FOR $MGBLSC AND $CRMPSC SYSTEM
/* ***** SERVICE REQUESTS AND ARE NOT STORED IN THE FLAGS WORD FIELD.
/*
```

```
  PFNMAP bitfield mask; /* MAP TO SPECIFIC PFN'S
  EXPREG bitfield mask; /* MAP INTO FIRST FREE ADDRESS SPACE
  PROTECT bitfield mask; /* CHECK WRITE ACCESS MODE (WRTMOD)
  PAGFIL bitfield mask; /* GLOBAL SECTION HAS PAGE FILE BACKING STORE
  EXECUTE bitfield mask; /* CHECK FOR EXECUTE ACCESS
end FLAGS_BITS;
```

```
end FLAGS_OVERLAY;
end SECDEF;

aggregate SECDEF1 structure prefix SECS;
  FILL_6 byte dimension 21 fill prefix SECDEF tag $$;
  AMOD byte unsigned; /*ACCESS MODE OF SECTION
  FILL_4 word fill prefix SECDEF tag $$; /*SPARE
  REFCNT longword unsigned; /*COUNT OF PTE'S REFERENCING THIS SECTION
  PAGCNT longword unsigned; /*NO. OF PAGES IN THE SECTION
/*
/* ***** NOTE THAT NUMBER OF LONG WORDS MUST BE EVEN
/*
  constant 'LENGTH' equals . prefix SECS tag K; /*SIZE OF PSTE/GSTE
  constant 'LENGTH' equals . prefix SECS tag C; /*SIZE OF PSTE/GSTE

/**
/* MATCH CONTROL VIELD VALUES
/*-

  constant(
    MATALL
    , MATEQU
    , MATLEQ
  ) equals 0 increment 1 prefix SEC tag $K; /* BASE OF ZERO , INCR 1
/* MATCH ALWAYS, USE GLOBAL SECTION
/* MATCH IF ISD$L_IDENT EQU GBL ID
/* MATCH IF ISD$L_IDENT LEQ GBL ID

end SECDEF1;
end_module $SECDEF;
```



```
module $SJCDEF;
```

```
/* Send to Job Controller Service
```

```
/*
```

```
/* NOTE: New items must always be added at the end so users will not have to
```

```
/* relink.
```

```
/* Function codes
```

```
constant (
  ABORT_JOB          /* Abort current job in a queue
  ADD_FILE           /* Add a file to an open job
  ALTER_JOB          /* Alter parameters of a job
  ALTER_QUEUE        /* Alter parameters of a queue
  ASSIGN_QUEUE       /* Assign a logical queue to an execution queue
  BATCH_CHECKPOINT   /* Establish a checkpoint in a batch job
  BATCH_SERVICE      /* LOGINOUT communication
  CLOSE_DELETE       /* Close and delete an open job
  CLOSE_JOB          /* Close and enqueue an open job
  CREATE_JOB         /* Create an open job
  CREATE_QUEUE       /* Create a queue
  DEASSIGN_QUEUE     /* Deassign a logical queue
  DEFINE_CHARACTERISTIC /* Create a characteristic definition
  DEFINE_FORM        /* Create a form definition
  DELETE_CHARACTERISTIC /* Delete a characteristic definition
  DELETE_FORM        /* Delete a form definition
  DELETE_JOB         /* Delete a job
  DELETE_QUEUE       /* Delete a queue
  ENTER_FILE         /* Enter a single file job in a queue
  MERGE_QUEUE        /* Requeue all jobs in one queue to another
  PAUSE_QUEUE        /* Suspend processing from a queue
  RESET_QUEUE        /* Reset a queue
  START_ACCOUNTING   /* Start accounting manager
  START_QUEUE        /* Start processing from a queue
  START_QUEUE_MANAGER /* Start system job queue manager
  STOP_ACCOUNTING    /* Stop accounting manager
  STOP_QUEUE         /* Stop processing from a queue
  STOP_QUEUE_MANAGER /* Stop system job queue manager
  SYNCHRONIZE_JOB    /* Synchronize with job completion
  WRITE_ACCOUNTING   /* Write user accounting record
  RESERVED_FUNC_1    /* Reserved for Digital use (name may change)
  RESERVED_FUNC_2    /* Reserved for Digital use (name may change)
) equals 1 increment T prefix SJC tag $;
```

```
/* Item codes
```

```
constant (
  ACCOUNTING_MESSAGE /* User accounting message
  ACCOUNTING_TYPES   /* Mask of accounting record types
  AFTER_TIME         /* Hold until time
  NO_AFTER_TIME      /*
  ALIGNMENT_MASK     /* Mask alignment data
  ALIGNMENT_PAGES    /* Number of alignment pages
  BASE_PRIORITY      /* Batch process base priority
  BATCH              /* Batch queue
```

```
. NO BATCH
. BATCH_INPUT /* LOGINOUT communication
. BATCH_OUTPUT
. CHARACTERISTIC_NAME /* Printer characteristic
. CHARACTERISTIC_NUMBER
. NO_CHARACTERISTICS
. CHECKPOINT_DATA /* Batch checkpoint string
. NO_CHECKPOINT_DATA
. CLI /* Batch process CLI
. NO_CLI
. CPU_DEFAULT /* Default CPU time limit
. NO_CPU_DEFAULT
. CPU_LIMIT /* Maximum CPU time limit
. NO_CPU_LIMIT
. CREATE_START /* Start queue after creation
. DELETE_FILE /* Delete file after processing
. NO_DELETE_FILE
. DESTINATION_QUEUE /* Destination queue name
. DEVICE_NAME /* Output device name
. DOUBLE_SPACE /* Double space output
. NO_DOUBLE_SPACE
. ENTRY_NUMBER /* Job entry number
. ENTRY_NUMBER_OUTPUT
. FILE_BURST /* Print file burst page
. FILE_BURST_ONE
. NO_FILE_BURST
. FILE_COPIES /* Print n copies of file
. FILE_FLAG /* Print file flag page
. FILE_FLAG_ONE
. NO_FILE_FLAG
. FILE_IDENTIFICATION /* DVI, FID, DID of file
. FILE_SETUP_MODULES /* Device control modules for file
. NO_FILE_SETUP_MODULES
. FILE_SPECIFICATION /* File specification of file
. FILE_TRAILER /* Print file trailer page
. FILE_TRAILER_ONE
. NO_FILE_TRAILER
. FIRST_PAGE /* Starting page number
. NO_FIRST_PAGE
. FORM_DESCRIPTION /* Textual description of form
. FORM_LENGTH /* Form length in lines
. FORM_MARGIN_BOTTOM /* Form bottom margin in lines
. FORM_MARGIN_LEFT /* Form left margin in characters
. FORM_MARGIN_RIGHT /* Form right margin in characters
. FORM_MARGIN_TOP /* Form top margin in lines
. FORM_NAME /* Printer form
. FORM_NUMBER
. FORM_SETUP_MODULES /* Device control modules for form
. NO_FORM_SETUP_MODULES
. FORM_SHEET_FEED /* Form is sheet fed
. NO_FORM_SHEET_FEED
. FORM_STOCK /* Stock name for form
. FORM_TRUNCATE /* Truncate long lines
. NO_FORM_TRUNCATE
. FORM_WIDTH /* Form width in characters
. FORM_WRAP /* Wrap long lines
```

```

. NO FORM_WRAP
. GENERIC_QUEUE /* Queue is a generic queue
. NO GENERIC_QUEUE
. GENERIC SELECTION /* Queue can take work from generic queue
. NO GENERIC SELECTION
. GENERIC_TARGET /* Possible execution queue for generic queue
. HOLD /* Place job on hold
. NO HOLD
. JOB BURST /* Print job burst page
. NO JOB BURST
. JOB_COPIES /* Print n copies of entire job
. JOB_FLAG /* Print job flag page
. NO JOB_FLAG
. JOB_LIMIT /* Number of concurrent jobs
. JOB_NAME /* Identifying name of job
. JOB_RESET_MODULES /* Device control modules between jobs
. NO JOB_RESET_MODULES
. JOB_SIZE_MAXIMUM /* Largest job accepted by printer
. NO JOB_SIZE_MAXIMUM
. JOB_SIZE_MINIMUM /* Smallest job accepted by printer
. NO JOB_SIZE_MINIMUM
. JOB_SIZE_SCHEDULING /* Schedule printer queues by size
. NO JOB_SIZE_SCHEDULING
. JOB_STATUS_OUTPUT /* Status message for submitted job
. JOB_TRAILER /* Print job trailer page
. NO JOB_TRAILER
. LAST_PAGE /* Ending page number
. NO LAST_PAGE
. LIBRARY_SPECIFICATION /* Filename of device control library
. NO LIBRARY_SPECIFICATION
. LOG_DELETE /* Delete log file
. NO LOG_DELETE
. LOG_QUEUE /* Log file's print queue
. LOG_SPECIFICATION /* File specification of log file
. NO LOG_SPECIFICATION
. LOG_SPOOL /* Print log file
. NO LOG_SPOOL
. LOWERCASE /* Print on lowercase printer
. NO LOWERCASE
. NEW_VERSION /* Create new version of file
. NEXT_JOB /* Resume at next job
. NOTE /* Flag page note text
. NO NOTE
. NOTIFY /* Notify user on completion
. NO NOTIFY
. OPERATOR_REQUEST /* Operator service text
. NO OPERATOR_REQUEST
. OWNER_UIC /* Queue owner UIC
. PAGE_HEADER /* Print page headers
. NO PAGE_HEADER
. PAGE_SETUP_MODULES /* Device control modules for form page
. NO PAGE_SETUP_MODULES
. PAGINATE /* Paginate output with free form feeds
. NO PAGINATE
. PARAMETER_1 /* Batch job parameters
. PARAMETER_2

```

```

PARAMETER_3
PARAMETER_4
PARAMETER_5
PARAMETER_6
PARAMETER_7
PARAMETER_8
NO PARAMETERS
PASSALL /* Print file passall
NO PASSALL
PRIORITY /* Job scheduling priority
PROCESSOR /* Filename of symbiont image
NO PROCESSOR
PROTECTION /* Queue protection mask
QUEUE /* Queue on which to operate
QUEUE_FILE_SPECIFICATION /* File specification of queue file
RELATIVE_PAGE /* Resume after forward or back space
REQUEUE /* Requeue job
RESTART /* Job can restart
NO RESTART
RETAIN_ALL_JOBS /* Retain completed jobs
RETAIN_ERROR_JOBS
NO RETAIN_JOBS
SCSNODE_NAME /* Node name of execution node
SEARCH_STRING /* Resume after finding string
SWAP /* Swap batch processes
NO SWAP
TERMINAL /* Queue is a (generic) terminal queue
NO TERMINAL
TOP_OF_FILE /* Resume at top of file
USER_IDENTIFICATION /* Proxy user identification (reserved for Digital use only)
WSDEFAULT /* Working set default
NO WSDEFAULT
WSEXTENT /* Working set extent
NO WSEXTENT
WSQUOTA /* Working set quota
NO WSQUOTA
ACCOUNT_NAME /* Proxy account name (requires CMKRNL privilege)
UIC /* Proxy UIC (requires CMKRNL privilege)
USERNAME /* Proxy username (requires CMKRNL privilege)
BUFFER_COUNT /* Multibuffer count for queue file
EXTEND_QUANTITY /* Allocation and extend quantity for queue file
RESERVED_BOOLEAN_1 /* Reserved for Digital use (name may change)
RESERVED_BOOLEAN_2 /* Reserved for Digital use (name may change)
RESERVED_BOOLEAN_3 /* Reserved for Digital use (name may change)
RESERVED_BOOLEAN_4 /* Reserved for Digital use (name may change)
RESERVED_INPUT_1 /* Reserved for Digital use (name may change)
RESERVED_INPUT_2 /* Reserved for Digital use (name may change)
RESERVED_INPUT_3 /* Reserved for Digital use (name may change)
RESERVED_INPUT_4 /* Reserved for Digital use (name may change)
RESERVED_OUTPUT_1 /* Reserved for Digital use (name may change)
RESERVED_OUTPUT_2 /* Reserved for Digital use (name may change)
) equals 1 increment 1 prefix SJC tag $;

/*
/* Structure of ACCOUNTING_TYPES item. These definitions must track
/* EXESGL_ACMFLAGS.

```

```
/*
aggregate ACCOUNTING_TYPES structure fill prefix SJCS;
  ACCT_PROCESS      bitfield mask; /* Process termination
  ACCT_IMAGE        bitfield mask; /* Image termination
  ACCT_INTERACTIVE  bitfield mask; /* Interactive processes
  ACCT_LOGIN_FAILURE bitfield mask; /* Login failures
  ACCT_SUBPROCESS   bitfield mask; /* Subprocesses
  ACCT_DETACHED     bitfield mask; /* Detached processes
  ACCT_BATCH        bitfield mask; /* Batch processes
  ACCT_NETWORK      bitfield mask; /* Network processes
  ACCT_PRINT        bitfield mask; /* Print job termination
  ACCT_MESSAGE      bitfield mask; /* User messages
  ACCT_UNUSED       bitfield length 32-^;
end;
end_module $SJCDEF;
```

ST

MO

/\*

/\*

/\*

/\*

ag

en

en

module \$SMRDEF;

/\*  
/\* DEFINE SYMBIONT MANAGER REQUEST CODES  
/\*

```
constant(  
  INITIAL /* SYMBIONT MANAGER REQUEST CODES  
  . DELETE /* INITIALIZE A QUEUE  
  . START /* DELETE THE QUEUE  
  . PAUSE /* ACTIVATE OR RESUME PRINTING  
  . MERGE /* SUSPEND PRINTING FOR A WHILE  
  . REDIRECT /* MOVE FILES FORM ONE QUEUE INTO ANOTHER  
  . JUSTIFY /* ASSIGNMENT OF ONE QUEUE TO ANOTHER  
  . STOP /* JUSTIFY PRINT FORMS(IE: ISSUE FF)  
  . ENTER /* DISABLE FURTHER DE-QUEING  
  . CREJOB /* ADD A FILE TO A QUEUE  
  . ADDFIL /* CREATE A PRINT JOB  
  . CLSJOB /* ADD A FILE TO EXISTING JOB  
  . RMVJOB /* CLOSE OPEN JOB  
  . ALTER /* REMOVE JOB FROM QUEUE  
  . ABORT /* CHANGE A FILES CHARACTERISTICS  
  . RELEASE /* ABORT CURRENT JOB, START NEXT  
  . DETINIT /* RELEASE JOB FROM HOLD LIST  
  . SYNCJOB /* DETACHED JOB HAS INITIATED  
  . ASKQNAM /* SYNCRONIZE WITH A BATCH JOB  
  ) equals 0 increment 1 prefix SMR tag $K;
```

/\*  
/\* ASSIGN SYMBIONT MANAGER PRINT JOB/RECORD OPTION CODES  
/\*  
/\*

```
constant(  
  DELETE /* JOB FILE OPTIONS  
  . BRSTPAG /* DELETE THE FILE AFTER PRINTING  
  . NOBRSTPAG /* CREATE A BURST PAGE  
  . FLAGPAG /* INHIBIT BURST PAGE  
  . NOFLAGPAG /* PRINT A FLAG PAGE  
  . PAGHDR /* INHIBIT A FLAG PAGE  
  . DOUBLE /* PRINT FILENAME AS PAGE HEADER  
  . NOFEED /* DOUBLE SPACE THE LISTING  
  . PAGCNT /* INHIBIT FORM FEED GENERATION  
  . COPIES /* SPECIFY PAGE COUNT TO PRINT  
  . FILESIZ /* NUMBER OF COPIES TO PRINT  
  . DESTQUE /* FILE SIZE  
  ) equals 1 increment 1 prefix SMO tag $K;
```

```
constant(  
  RLSTIM /* OPTIONS FOR CREATE JOB  
  . HOLD /* TIME TO HOLD UNTIL QUEUEING  
  . JOBPRI /* HOLD THE JOB  
  . JOBCOPY /* SPECIFY PRINT JOB/DETACHED JOB PRIORITY  
  . FORMTYPE /* NUMBER OF COPIES OF ENTIRE JOB  
  . LOWER /* FORM TYPE TO USED WHEN PRINTED  
  ) equals 1 increment 1 prefix SMO tag $K;
```

ST

MO

/\*

/\*

/\*

/\*

/\*

/\*

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

Co

```

. PARAMS
. JOBNAME
. NOLOWER
. CPULIM
. NOCPULM
. WSQUOTA
. NOWSQUO
. WSDEFLT
. NOWSDFT
. DQCHAR
. LOGFNAM
. LOGQNAM
. KEEPLOG
. NOSPOOL
. NOKEEPLOG
. NOLOG
. NOTIFY
. NONOTFY
. WSEXTNT
. NOWSEXTNT
) equals 32 increment 1 prefix SMO tag $K;

```

```

/* SUPPLY JOB PARAMETERS
/* SUPPLY A NAME TO THE JOB
/* CLEAR LOWER CASE REQUIRED
/* CPU TIME LIMIT
/* NO CPU TIME LIMIT
/* WORKING SET QUOTA
/* NO WORKING SET QUOTA
/* WORKING SET DEFAULT
/* NO WORKING SET DEFAULT
/* CHAR. TO BE USED WHEN PRINTED
/* LOG FILE SPEC.
/* QUEUE NAME FOR LOG FILE
/* KEEP LOG FILE
/* DON'T SPOOL LOG FILE
/* DELETE LOG FILE
/* CREATE LOG ON NULL DEVICE
/* NOTIFY WHEN JOB COMPLETES
/* NO NOTIFY
/* WS EXTENT
/* NO WS EXTENT

```

```

constant(
. CURFORM
. GENPRT
. NOGENPRT
. DETJOB
. GENDEV
. NOGENDEV
. DEFBRST
. NODEFBRST
. DEFFLAG
. NODEFFLAG
. TRMDEV
. NOTRMDEV
. JOBLIM
. INIPRI
. DISWAP
. MCPULM
. NOMCPULM
. DCPULM
. NODCPULM
. WSQUOTA
. NOWSQUOTA
. WSDEFLT
. NOWSDFLT
. CURDQCHAR
. SMBNAME
. WSXTANT
. NOWSXTNT
. NODISWAP
) equals 64 increment 1 prefix SMO tag $K;

```

```

/* OPTIONS FOR INIT OR START A QUEUE
/* CURRENT FORMS MOUNTED ON DEVICE
/* ALLOW GENERIC PRINTING
/* DISALLOW GENERIC PRINTING
/* QUEUE IS FOR DETACHED JOB QUEUING
/* QUEUE IS GENERIC DEVICE QUEUE
/* NOT A GENERIC DEVICE
/* QUEUE HAS BURST PAGE BY DEFAULT
/* NO DEFAULT BURST PAGE
/* QUEUE HAS FLAG PAGE BY DEFAULT
/* NO DEFAULT FLAG
/* DEVICE IS A TERMINAL
/* DEVICE IS NOT A TERMINAL
/* JOB LIMIT FOR BATCH JOBS
/* JOB INITIAL PRIORITY
/* DISABLE SWAPPING
/* MAX CPU TIME LIMIT
/* NO MAX CPU TIME LIMIT
/* DEFAULT CPU TIME LIMIT
/* NO DEFAULT CPU TIME LIMIT
/* WORKING SET QUOTA
/* NO WORKING SET QUOTA
/* WORKING SET DEFAULT
/* NO WORKING SET DEFAULT
/* CURRENT CHAR. OF THE QUEUE
/* SYMBIONT NAME FOR QUEUE
/* WS EXTENT
/* NO WS EXTENT
/* ENABLE SWAPPING

```

```

constant(

```

```

/* OPTIONS FOR RESTART

```





```
module $STSDEF;
```

```
/*
/* MACROS TO DEFINE SYSTEM SERVICE FAILURE AND STATUS CODES.
```

```
/*
/*          $STRUCT          SS
/*
/* THIS STRUCTURE IS NOW CONTAINED IN 'SSMSG.MDL'
```

```
/**
/* Define 32-bit VAX-11 error code longword fields
/* Changed to agree with Handbook - TNH 26-OCT-77 - CHANGE M 1 to P 1
/* Changed to agree with new status value W. H. Brown 18-NOV-1977
/*-
```

```
aggregate STSDEF union prefix STS$;
```

```
  STSDEF_BITS0 structure fill;
    SEVERITY bitfield mask length 3;          /* severity code
    COND_ID bitfield mask length 25;         /* condition id
    CONTROL bitfield mask length 4;          /* control bits
  end STSDEF_BITS0;
```

```
  STSDEF_BITS1 structure fill;
    SUCCESS bitfield mask;                   /* Success
    FILL_1 bitfield length 2 fill prefix STSDEF tag $$; /* skip severity field
    MSG_NO bitfield mask length 13;          /* Message number
  end STSDEF_BITS1;
```

```
  STSDEF_BITS2 structure fill;
    FILL_2 bitfield length 3 fill prefix STSDEF tag $$; /* Skip severity field
    CODE bitfield mask length 12;           /* code
    FAC_SP bitfield mask;                   /* 1 if facility specific
    FILL_3 bitfield length 11 fill prefix STSDEF tag $$;
    CUST_DEF bitfield mask;                 /* 1 if sub-system is customer (or CSS) defined
    INHIB_MSG bitfield mask;               /* 1 if message should not be printed
    FILL_4 bitfield length 3 fill prefix STSDEF tag $$; /* reserved control bits
  end STSDEF_BITS2;
```

```
  STSDEF_BITS3 structure fill;
    FILL_5 bitfield length 16 fill prefix STSDEF tag $$; /* Skip message number field
    FAC_NO bitfield mask length 12;         /* Facility is 12 bits
    FILL_6 bitfield length 4 fill prefix STSDEF tag $$; /* Control bits
  end STSDEF_BITS3;
```

```
/**
/* Define severity codes
/*-
```

```
  constant WARNING equals 0 prefix STS tag $K; /* WARNING
  constant SUCCESS equals 1 prefix STS tag $K; /* SUCCESSFUL COMPLETION
  constant ERROR equals 2 prefix STS tag $K; /* ERROR
  constant INFO equals 3 prefix STS tag $K; /* INFORMATION
  constant SEVERE equals 4 prefix STS tag $K; /* SEVERE ERROR
```

STARDEFQZ.SDL:1

16-SEP-1984 16:47:04.31<sup>15</sup> Page 24

end STSDEF;

end\_module \$STSDEF;

ST

en

en

mo

ag

en

en

```
module $$YIDEF;
```

```
/*+
```

```
/* Get System Information Data Identifier Definitions
```

```
/*
```

```
/* **** NOTE ****
```

```
/*
```

```
/* New items must always be added to the END of each item list
/* (preceeding the ENDlistname item) so that users will not have
/* to relink.
```

```
/*-
```

```
/* DEFINE TABLE TYPES
```

```
constant EXETYPE      equals 1  prefix SYI tag $C;  /* Executive cells
constant FLDTYPE      equals 2  prefix SYI tag $C;  /* Sub-field data
constant RSDDTYPE     equals 3  prefix SYI tag $C;  /* Reduced system data
```

```
/* The following item-codes are for executive data cells.
/* Processor registers are a special case of this type.
/* Some EXE-type cells have frequently used sub-fields,
/* which have item-codes in the FLD section. (after this one)
```

```
constant(
  VERSION              /* VMS version number
  . SID                 /* System ID register
  . PFCDEFAULT          /* Default page fault cluster size
  . PAGTBLPFC          /* Default page table page fault cluster
  . SYSPFC              /* Page fault cluster for system paging
  . KFILSTCNT          /* Number of known file lists
  . GBLSECTIONS        /* Global section count
  . GBLPAGES           /* Global page count
  . GBLPAGFIL          /* Global page page file page limit
  . MAXPROCESSCNT      /* Maximum process count
  . PIXSCAN            /* Process scan count
  . PROCSECTCNT        /* Process section count
  . MINWSCNT           /* Minimum working set size
  . PAGFILCNT          /* Number of paging files
  . SWPFILCNT          /* Number of swap files
  . SYSMWCNT           /* System working set count
  . INTSTKPAGES        /* Interrupt stack size
  . DLCKEXTRASTK       /* Extra int stack for deadlock searches
  . BALSETCNT          /* Balance set count
  . IRPCOUNT           /* Count of I/O packets
  . IRPCOUNTV          /* Limit of extension of IRP list
  . WSMAX              /* Max size of process working set
  . NPAGEDYN           /* Non-paged dynamic pool
  . NPAGEVIR           /* Limit of extension of above
  . PAGEDYN            /* Paged dynamic pool
  . VIRTUALPAGECNT    /* Max virtual page count
  . SPTREQ             /* Requested SPT extension
  . EXUSRSTK           /* Extra user stack provided
  . LRPCOUNT           /* Number of large request packets
  . LRPCOUNTV          /* Limit of extension of LRP list
  . LRPSIZE            /* Size of large request packet
  . LRPMIN             /* Min alloc request for LRPs
```

ST

MO

/\*

/\*

/\*

/\*

/\*

/\*

ag

en

en

MO

/\*

/\*

/\*

/\*

ag

```

.* SRPCOUNT /* Number of small request packets
.* SRPCOUNTV /* Limit of extension of SRP list
.* SRPSIZE /* Size of small request packet
.* SRPMIN /* Min alloc request for SRPs
.* CHANNELCNT /* Number of permanent I/O channels
.* PIOPAGES /* Process I/O pages
.* CTLPAGES /* Control region impure pages
.* CTLIMGLIM /* Limit on use of proc alloc region
.* IMGIOCNT /* Default image I/O address space
.* QUANTUM /* Process quantum
.* MPW_WRTCLUSTER /* Page write cluster factor
.* MPW_HILIMIT /* Modified page list high limit
.* MPW_LOLIMIT /* Modified page list low limit
.* MPW_PRIO /* Modified page writer I/O priority
.* SWP_PRIO /* Swapper I/O priority
.* MPW_THRESH /* Modified page writer lower threshold
.* MPW_WAITLIMIT /* Modified page writer busy wait limit
.* TBSKIPWSL /* Max number WSlst entries to skip
.* PHYSICALPAGES /* Max number of physical pages to use
.* PFRATL /* Page fault rate lower threshold
.* PFRATH /* Page fault rate high threshold
.* PFRATS /* Page fault rate system threshold
.* WSINC /* Working set increment
.* WSDEC /* Working set decrement
.* AWSMIN /* Working set minimum
.* AWSTIME /* Working set measurement interval
.* SWPRATE /* Swap rate control (max)
.* SWPOUTPGCNT /* Desired proc page cnt for an outswap
.* SWPALLOCINC /* Swap file allocation increment value
.* IOTA /* I/O time allowance
.* LONGWAIT /* When HIB or LEF proc becomes longwait
.* SWPFAIL /* Swap fail count
.* VMSD1 /* Reserved to VMS development (DYNAMIC)
.* VMSD2 /* ""
.* VMSD3 /* ""
.* VMSD4 /* ""
.* VMS5 /* Reserved to VMS development (STATIC)
.* VMS6 /* ""
.* VMS7 /* ""
.* VMS8 /* ""
.* USERD1 /* Reserved for users (DYNAMIC)
.* USERD2 /* ""
.* USER3 /* Reserved for users (STATIC)
.* USER4 /* ""
.* EXTRACPU /* Extra CPU time allowed on expiration
.* MAXSYSGROUP /* Max group code for system UIC
.* MVTIMEOUT /* Timeout for mount verification
.* MAXBUF /* Maximum allowable buffered I/O size
.* DEFMBXBUFQUO /* Def buffer quota for mailbox creation
.* DEFMBXMXMSG /* Def max msg size for mailbox creation
.* DEFMBXNUMMSG /* Def number of messages for ""
.* FREELIM /* Desired free list length
.* FREGOAL /* Target free list length
.* ROWLIM /* Free > growlim to grow proc > wsquota
.* BORROWLIM /* Same as above???
.* LOCKRETRY /* Multi-CPU lock retry count

```

```

.* XFMXRATE /* Max DR32 data rate
.* LAMAPREGS /* Number of UBA map regs for LPA11
.* REALTIME_SPTS /* SPT entries for CONINTERR processes
.* CLISYMTBL /* Number of pages for CLI symbol table
.* LOCKIDTBL /* Size of lock ID table
.* RESHASHTBL /* Size of resource hash table
.* DEADLOCK_WAIT /* Deadlock detection timeout period
.* SCSBUFFCNT /* SCS buffer descriptor table entries
.* SCSCONNcnt /* SCS connect descriptor table entries
.* SCSRESPCNT /* SCS response descriptor table entries
.* S^SMAXDG /* SCS maximum datagram size
.* S^SMAXMSG /* SCS max sequenced message size
.* SCSFLOWCUSH /* SCS flow control cushion
.* SCSYSTEMID /* SCS system ID (48 bits)
.* SCSYSTEMIDH /* High bits of above
.* SCSNODE /* SCS system node name
.* /* Used to be SCSNODEH. Unused for now.
.* PRCPOLINTERVAL /* SCA process polling interval
.* PASTIMOUT /* CI port START/STACK timeout
.* PASTDGBUF /* CI port DG buffers for START handshake
.* PANUMPOLL /* Number of CI ports to poll
.* PAPOLLINTERVAL /* Time between CI port polls
.* PAPOOLINTERVAL /* Time between SYSAPs pool check
.* TIMEPROMPTWAIT /* Time prompt timeout (when booting)
.* UDABURSTRATE /* Max number of longwords per NPR - 1
.* LNMSHASHTBL /* Size of system space log name hash table
.* LNMPHASHTBL /* Size of process space
.* TTY_SCANDelta /* Delta time for dialup timer scan
.* TTY_DIALTYPE /* Flags for dialup type
.* TTY_SPEED /* Terminal default: speed
.* TTY_RSPEED /* " " receive speed
.* TTY_PARITY /* " " parity
.* TTY_BUF /* " " line width
.* TTY_DEFCHAR /* " " characteristics
.* TTY_DEFCHAR2 /* " " characteristics, 2nd longword
.* TTY_TYPAHDSZ /* Size of typeahead buffer
.* TTY_ALTYPAMD /* Alternate typeahead size
.* TTY_ALTALARM /* Alt typeahead alarm size
.* TTY_DMASIZE /* Dma size
.* TTY_PROT /* Default terminal alloc protection
.* TTY_OWNER /* Default terminal owner
.* TTY_CLASSNAME /* Default terminal class name prefix
.* TTY_SILOTIME /* Default silo timeout for DMF32
.* TTY_DEFPORT /* DEFAULT PORT ACTION PARAMETER
.* RMS_DFMBC /* RMS default: multi-block count
.* RMS_DFMBSDK /* " " multi-buffer count for disk (SEQ)
.* RMS_DFMBSMT /* " " multi-buffer count for tape
.* RMS_DFMBSUR /* " " multi-buffer count for unit record
.* RMS_DFMBSREL /* " " multi-buffer count for REL files
.* RMS_DFMBSIDX /* " " multi-buffer count for ISAM files
.* RMS_DFMBSHSH /* " " multi-buffer count for HASH files
.* RMS_PROLOGUE /* " " prolog
.* RMS_EXTEND_SIZE /* " " file extend quantity
.* RMS_FILEPROT /* " " file protection
.* PQL_DASTLM /* Def AST limit
.* PQL_MASTLM /* Min "

```

```

. PQL_DBIOLM /* Def buffered I/O limit
. PQL_MBIOLM /* Min
. PQL_DBYTLM /* Def buffered I/O byte count limit
. PQL_MBYTLM /* Min
. PQL_DCPULM /* Def CPU time limit
. PQL_MCPULM /* Min
. PQL_DDIOLM /* Def direct I/O limit
. PQL_MDIOLM /* Min
. PQL_DFILLM /* Def open file limit
. PQL_MFILLM /* Min
. PQL_DPGFLQUOTA /* Def paging file quota
. PQL_MPGFLQUOTA /* Min
. PQL_DPRCLM /* Def sub-process limit
. PQL_MPRCLM /* Min
. PQL_DTQELM /* Def timer queue entry limit
. PQL_MTQELM /* Min
. PQL_DWSDEFAULT /* Def working set default size
. PQL_MWSDEFAULT /* Min
. PQL_DWSQUOTA /* Def working set quota
. PQL_MWSQUOTA /* Min
. PQL_DWSEXTENT /* Def working set extent
. PQL_MWSEXTENT /* Min
. PQL_DENQLM /* Def enqueue limit
. PQL_MENQLM /* Min
. ACP_MAPCACHE /* Number of blocks in bitmap cache
. ACP_HDRCACHE /* Number of blocks in file header cache
. ACP_DIRCACHE /* Number of blocks in directory cache
. ACP_WORKSET /* ACP working set
. ACP_FIDCACHE /* Number of cached index file slots
. ACP_EXTCACHE /* Number of cached disk extents
. ACP_EXTLIMIT /* Max fraction of disk to cache
. ACP_QUOCACHE /* Number of file quota entries to cache
. ACP_SYSACC /* Default access for system volumes
. ACP_MAXREAD /* Max number of direct blks to read
. ACP_WINDOW /* Default window size for system volumes
. ACP_WRITEBACK /* Deferred cache writeback enable
. ACP_DATACHECK /* ACP datacheck enable flags
. ACP_BASEPRIO /* ACP base priority
. ACP_SWAPFLGS /* ACP swap flags
. DEFPRI /* Default priority for Job Initiations
. IJOBLIM /* Limit for interactive jobs
. BJOBLIM /* Limit for batch jobs
. NJOBLIM /* Limit for network jobs
. RJOBLIM /* Limit for remote terminal jobs
. QUORUM /* Connection manager cluster QUORUM
. VOTES /* Connection manager VOTES from this system
. RECNXINTERVAL /* Connection manager reconnect interval
. BOOTTIME /* 64-bit abs. system time at system boot
. LOCKIDTBL_MAX /* Maximum size of lock ID table.
. TAILORED /* Is this system tailored?
. STARTUP_P1 /* Parameters for passing information
. STARTUP_P2 /* to system startup procedure.
. STARTUP_P3
. STARTUP_P4
. STARTUP_P5
. STARTUP_P6

```

B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z

```

. STARTUP_P7
. STARTUP_P8
. CLUSTER_NODES /* total number of nodes in cluster
. CLUSTER_VOTES /* " " votes
. CLUSTER_QUORUM /* " " quorum in cluster
. CLUSTER_FSYSID /* founding systemid
. CLUSTER_FTIME /* founding boottime
. CLUSTER_MEMBER /* true if in a live cluster
. NODE_CSID /* cluster system id of target node
. NODE_VOTES /* votes of target node
. NODE_QUORUM /* quorum of " "
. NODE_SYSTEMID /* systemid of " "
. NODE_SWINCARN /* S/W incarnation of " "
. NODE_SWTYPE /* S/W type of " "
. NODE_SWVERS /* S/W version of " "
. NODE_HWTYPE /* H/W type of " "
. NODE_HWVERS /* H/W version of " "
. NODENAME /* SCS nodename of " "
. ARCHFLAG /* Flags used by uVAX code
. SCS_EXISTS /* Flag to indicate presence of SCS
. DISK_QUORUM /* Cluster quorum device name (in ASCII)
. /* Used to be DISK_QUORUM2. Unused for now.
. /* Used to be DISK_QUORUM3. Unused for now.
. /* Used to be DISK_QUORUM4. Unused for now.
. PAMAXPORT /* Max PA port to poll
. PASANITY /* PA sanity timer enable/disable switch
. DEFQUEPRI /* Default Que Priority
. MAXQUEPRI /* Max Que Priority
. QDSKINTERVAL /* Disk Quorum Interval
. ALLOCLASS /* Allocation class
. LGI_RETRY_LIM /* Login retry limit
. LGI_RETRY_TMO /* Login retry timeout
. LGI_BRK_LIM /* Login failure limit
. LGI_BRK_TMO /* Login suspect timeout
. LGI_HID_TIM /* Login evasive action time
. LGI_PWD_TMO /* System password drop dead time
. PQL_DJTQUOTA /* Default Job-wide Logical Name Table Creation Quota
. PQL_MJTQUOTA /* Minimum Job-wide Logical Name Table Creation Quota
. VAXCLUSTER /* VAXcluster participation
. LOCKDIRWT /* Lock manager directory system weight
. QDSKVOTES /* Votes held by quorum disk
. DORMANTWAIT /* When low pri COM proc becomes dormant
. PAGEFILE_PAGE /* Total pagefile pages
. SWAPFILE_PAGE /* Total swapfile pages
. PAGEFILE_FREE /* Free pagefile pages
. SWAPFILE_FREE /* Free swapfile pages
. TTY_TIMEOUT /* Seconds for virtual terminal timeout
. TTY_AUTOCHAR /* autobaud recognition character
. PANOPOLL /* Disable/enable CI port polling
. PE1 /* Reserved TO PEDRIVER (DYNAMIC)
. PE2 /* "
. PE3 /* "
. PE4 /* "
. PE5 /* Reserved TO PEDRIVER (STATIC)
. PE6 /* "
. RMS_GBLBUFQUO /* RMS global buffer quota

```

```

. RMS_DFNBC          /* "" network block count
. ACP_DINDXCACHE    /* File system directory index cache blocks

/*   ADD 'EXE' ITEM-CODES ONLY JUST BEFORE THIS COMMENT *****
. LASTEXE          /* LAST INDEX
/* ) equals SYISC_EXETYPE@12 increment 1 prefix SYI tag $;
/* ) equals 4096      increment 1 prefix SYI tag $;

/*   The following item-codes are for 'sub-field data' items.
/*   There usually exists another item-code to retrieve the
/*   whole parent cell (longword) - in the EXE section above.

constant(
. CPU              /* CPU type
. BUGREBOOT        /* Bugcheck reboot enable
. CRDENABE         /* Memory error checking enable
. DUMPBUG          /* Bugcheck dump enable
. BUGCHECKFATAL    /* All bugchecks become fatal
. ACP_MULTIPLE     /* New ACP for each disk class
. NOADTOCONFIG     /* Auto configuration inhibit
. NOCLOCK          /* Interval timer inhibit
. NOCLUSTER        /* Page read clustering inhibit
. POOLPAGING       /* Enable paging of paged pool
. SBIERRENABLE    /* SBI error detection enable
. SETTIME          /* Force entry of time at system boot
. ACP_SHARE        /* Enable sharing of F11ACP
. SYSPAGING        /* Enable paging of system code
. UAFALTERNATE    /* Select alternate authorization file
. WRITABLESYS     /* Leave system writable for debugging
. RESALLOC         /* Enable resource allocation checking
. SSINHIBIT        /* Inhibit system services for processes
. CONCEAL_DEVICES /* Enable concealed devices
. SAVEDUMP         /* Save dump if it's in page file
. MOUNTMSG         /* Enable oper notification of vol mounts
. DISMOUMSG        /* Enable oper notification of vol dism
. LOADERAPT        /* Load alt erase pattern generator
. LOADCHKPRT       /* Load alt protection check routine
. /* This name has been moved VAXCLUSTER /* Load cluster code
. CJFLOAD          /* Load common journaling code
. CJFSYSRUJ        /* Load recovery unit journaling code
. NODE_AREA        /* System ID node area
. NODE_NUMBER      /* System ID node number
. CLASS_PROT       /* Non-discretionary check flag
. CHARACTER_EMULATED /* Character string instructions emulated
. DECIMAL_EMULATED /* Decimal string instructions emulated
. D_FLOAT_EMULATED /* D floating instructions emulated
. F_FLOAT_EMULATED /* F floating instructions emulated
. G_FLOAT_EMULATED /* G floating instructions emulated
. H_FLOAT_EMULATED /* H floating instructions emulated
. LOADMTACCESS     /* Load installation specific access routine
. ACP_XQP_RES      /* Resident XQP
. WRITE_VSPARAMS   /* Write active parameters to the system .PAR file
. LGI_BRK_TERM     /* Use term name in breakin detection
. LGI_BRK_DISUSER  /* Disable user account on breakin
. ACP_REBCDSYSD    /* Rebuild system disk when mounting it
. WS_OPAO          /* Workstation using QVSS for OPAO

```



```
/*      ADD 'FLD' ITEM-CODES ONLY JUST BEFORE THIS COMMENT *****
/*      , LASTFLD                               /* LAST INDEX
/*      ) equals SYISC_FLDTYPE@12 increment 1 prefix SYI tag $;
/*      ) equals 8192                          increment 1 prefix SYI tag $;

/*****
/**
/**      THE FOLLOWING DEFINITIONS EXIST FOR COMPATIBILITY WITH THE 1ST
/**      IMPLEMENTATION OF SYSSGETSYI. DO NOT CHANGE THEM!!!
/**      ADD NEW GETSYI ITEM-CODES ONLY TO THE PREVIOUS LISTS.
/**
/*****

constant SFWTYPE          equals 1 prefix SYI tag $C; /* SYSTEM SOFTWARE ITEM
constant HDWTYPE         equals 2 prefix SYI tag $C; /* HARDWARE ITEM
constant LISTEND         equals 0 prefix SYI tag $C; /* ITMLST TERMINATOR

/* IDENTIFIER FOR SYSTEM SOFTWARE

constant(
  OLDVERSION              /* VMS VERSION NUMBER
  , LASTSFW                /* LAST INDEX
/* *****
/* ***** BEGIN TEMP FIX *****
/* ***** BEGIN TEMP FIX *****
/* *****
/*      ) equals SYISC_SFWTYPE@8 increment 1 prefix SYI tag $;
/*      ) equals 256          increment 1 prefix SYI tag $;
/* *****
/* ***** END TEMP FIX *****
/* ***** END TEMP FIX *****
/* *****
/* IDENTIFIER FOR HARDWARE TYPE

constant(
  OLDCPU                  /* CPU TYPE
  , OLDSID                 /* SYSTEM ID REGISTER
  , LASTHDW               /* LAST INDEX
/*      ) equals SYISC_HDWTYPE@8 increment 1 prefix SYI tag $;
/*      ) equals 512          increment 1 prefix SYI tag $;

end_module $SYIDEF;
```

```
module $TPADEF;
```

```
/*  
/* TPARSE CONTROL BLOCK. THIS BLOCK IS ONE OF THE ARGUMENTS USED TO CALL TPARSE,  
/* AND BECOMES THE ARGUMENT LIST TO ACTION ROUTINES.  
/*
```

```
aggregate TPADEF structure prefix TPAS;  
COUNT longword unsigned; /* ARGUMENT COUNT (NUMBER OF LONGWORDS)  
constant COUNT0 equals 8 prefix TPA tag $K; /* ZERO LEVEL VALUE IS 8  
OPTIONS_OVERLAY union fill;  
  OPTIONS longword unsigned; /* OPTIONS LONGWORD  
  OPTIONS_BITS structure fill;  
    BLANKS bitfield mask; /* PROCESS BLANKS AND TABS EXPLICITLY  
    ABBREV bitfield mask; /* ALLOW MINIMAL ABBREVIATION  
    ABBRFM bitfield mask; /* ALLOW FIRST MATCH ABBREVIATION  
    FILL_1 bitfield length 13 fill prefix TPADEF tag $$;  
    AMBIG bitfield mask; /* AMBIGUOUS KEYWORD AT THIS STATE  
  end OPTIONS_BITS;  
  OPTIONS_FIELDS structure fill;  
    FILL_2 byte dimension 3 fill prefix TPADEF tag $$;  
    MCOUNT byte unsigned; /* MINIMUM ABBREVIATION ON KEYWORDS  
  end OPTIONS_FIELDS;  
end OPTIONS_OVERLAY;  
STRINGCNT longword unsigned; /* BYTE COUNT OF STRING BEING PARSED  
STRINGPTR longword unsigned; /* ADDRESS OF STRING BEING PARSED  
TOKENCNT longword unsigned; /* BYTE COUNT OF CURRENT MATCHING TOKEN  
TOKENPTR longword unsigned; /* ADDRESS OF MATCHING TOKEN  
CHAR_OVERLAY union fill;  
  CHAR longword unsigned; /* ASCII CODE OF SINGLE CHARACTER TOKEN  
  CHAR byte unsigned; /* BYTE FORM OF SINGLE CHAR CELL  
end CHAR_OVERLAY;  
NUMBER longword unsigned; /* NUMERIC VALUE OF NUMERIC TOKEN  
PARAM longword unsigned; /* PARAMETER LONGWORD FROM STATE TABLE  
constant LENGTH0 equals . prefix TPAS tag K; /* LENGTH OF ZERO LEVEL PARAMETER BLOCK  
constant LENGTH0 equals . prefix TPAS tag C; /* LENGTH OF ZERO LEVEL PARAMETER BLOCK  
end TPADEF;  
end_module $TPADEF;
```

```

module $TRMDEF;
/*
/* Define symbols for the item list QIO format
/*
/* Item list type codes
Constant(
    MODIFIERS,
    EDITMODE,
    TIMEOUT,
    TERM,
    PROMPT,
    INISTRNG,
    PICSTRNG,
    FILLCHR,
    INIOFFSET,
    ALTECHSTR,
    LASTITM          /* must remain the last item
) equals 0 increment 1 prefix TRM tag $;
/*
/* Editmode type codes
/*
Constant(
    DEFAULT,
    RDVERIFY
) equals 0 increment 1 prefix TRMS tag K_EM;

Aggregate TRMDEF Union prefix TRMS;
TRMDEF_BITS0 structure fill;
    FICL_1 bitfield length 6 fill prefix TRMDEF tag $$; /*FUNCTION CODE
    TM_NOECHO bitfield mask; /*NOECHO
    TM_TIMED bitfield mask; /*TIMED
    TM_CVTLOW bitfield mask; /*CONVERT LOWER CASE
    TM_NOFILTR bitfield mask; /*NO FILTER
    TM_DSABLMBX bitfield mask; /*DISABLE MAILBOX
    TM_PURGE bitfield mask; /*PURGE TYPEAHEAD
    TM_TRMNOECHO bitfield mask; /*TERMINATORS ARE NOT ECHOED
    TM_REFRESH bitfield mask; /*Control-R interrupted read n
    TM_ESCAPE bitfield mask; /*TERMINATE READ ON ESCAPE SEQUENCE
    TM_NOEDIT bitfield mask; /*DISABLE ADVANCED EDITING
    TM_NORECALL bitfield mask; /*Disable the recall feature of advanced editing
    TM_RJUST bitfield mask; /*Right justification
    TM_AUTO TAB bitfield mask; /*Auto tab field
end TRMDEF_BITS0;

TRMDEF_BITS1 structure fill; /* picture string mask
    CV_UPPER bitfield mask; /* upper case
    CV_LOWER bitfield mask; /* lower case
    CV_NUMERIC bitfield mask; /* numeric (0-9)
    CV_NUMPUNC bitfield mask; /* numeric punctuation (+ - .)
    CV_PRINTABLE bitfield mask; /* printable character
    CV_ANY bitfield mask; /* any character
end TRMDEF_BITS1;

end TRMDEF;

```

STARDEFQZ.SDL:1

16-SEP-1984 16:47:04.31 <sup>G 16</sup> Page 34

end\_module STRMDEF;

```
module $TDEF;
```

```
/*
```

```
/* DEFINE TERMINAL SPECIAL SYMBOLS
```

```
/*
```

```
constant(
```

```
    BAUD_50
    , BAUD_75
    , BAUD_110
    , BAUD_134
    , BAUD_150
    , BAUD_300
    , BAUD_600
    , BAUD_1200
    , BAUD_1800
    , BAUD_2000
    , BAUD_2400
    , BAUD_3600
    , BAUD_4800
    , BAUD_7200
    , BAUD_9600
    , BAUD_19200
```

```
) equals 1 increment 1 prefix TT tag $C;
```

```
constant UNKNOWN equals 0 prefix TT tag $;
```

```
constant VT05 equals 1 prefix TT tag $;
```

```
constant VK100 equals 2 prefix TT tag $;
```

```
constant VT173 equals 3 prefix TT tag $;
```

```
constant TQ BTS equals 4 prefix TT tag $;
```

```
constant TER401X equals 10 prefix TT tag $;
```

```
constant FT1 equals 16 prefix TT tag $;
```

```
constant FT2 equals 17 prefix TT tag $;
```

```
constant FT3 equals 18 prefix TT tag $;
```

```
constant FT4 equals 19 prefix TT tag $;
```

```
constant FT5 equals 20 prefix TT tag $;
```

```
constant FT6 equals 21 prefix TT tag $;
```

```
constant FT7 equals 22 prefix TT tag $;
```

```
constant FT8 equals 23 prefix TT tag $;
```

```
constant LAX equals 32 prefix TT tag $;
```

```
constant LA36 equals 32 prefix TT tag $;
```

```
constant LA120 equals 33 prefix TT tag $;
```

```
constant LA34 equals 34 prefix TT tag $;
```

```
constant LA38 equals 35 prefix TT tag $;
```

```
constant LA12 equals 36 prefix TT tag $;
```

```
constant LA100 equals 37 prefix TT tag $;
```

```
constant LA24 equals 37 prefix TT tag $;
```

```
constant LQP02 equals 38 prefix TT tag $;
```

```
constant LA84 equals 39 prefix TT tag $;
```

```
constant VT5X equals 64 prefix TT tag $;
```

```
constant VT52 equals 64 prefix TT tag $;
```

```
constant VT55 equals 65 prefix TT tag $;
```

```
constant VT100 equals 96 prefix TT tag $;
```

```
constant VT101 equals 97 prefix TT tag $;
```

```
constant VT102 equals 98 prefix TT tag $;
```

```
constant VT105 equals 99 prefix TT tag $;
```

```
constant VT125 equals 100 prefix TT tag $;
```

```
/* *** MATCHES DCDEF MACRO
```

```
/* VT05
```

```
/* strange terminals
```

```
/* FOREIGN TERMINAL TYPES
```

```
/* new support also includes
```

```
/* the negative numbers
```

```
/* END OF FOREIGN TYPES
```

```
/* RESERVE REST UP TO 32 FOR EXTENSIONS
```

```
/* VT100
```

```
constant VT131 equals 101 prefix TT tag $;  
constant VT132 equals 102 prefix TT tag $;  
constant VT200_Series equals 110 prefix TT tag $;  
constant Pro_Series equals 111 prefix TT tag $;  
/* from 128 on is reserved for foreign terminals and must not be  
/* used for terminal definitions
```

```
aggregate TTDEF union prefix TTS;
```

```
TTDEF BITS0 structure fill;
```

```
  PASSALL bitfield mask;  
  NOECHO bitfield mask;  
  NOTYPEAHD bitfield mask;  
  ESCAPE bitfield mask;  
  HOSTSYNC bitfield mask;  
  TTSYNC bitfield mask;  
  SCRIPT bitfield mask;  
  LOWER bitfield mask;  
  MECHTAB bitfield mask;  
  WRAP bitfield mask;  
  CRFILL bitfield mask;  
  LFFILL bitfield mask;  
  SCOPE bitfield mask;  
  REMOTE bitfield mask;  
  HOLDSCREEN bitfield mask;  
  EIGHTBIT bitfield mask;  
  MBXDSABL bitfield mask;  
  NOBRDCST bitfield mask;  
  READSYNC bitfield mask;  
  MECHFORM bitfield mask;  
  HALFDUP bitfield mask;  
  MODEM bitfield mask;  
  OPER bitfield mask;  
  FILL_1 bitfield fill prefix TTDEF tag $$;  
  PAGE bitfield mask length 8;
```

```
end TTDEF BITS0;
```

```
TTDEF BITS1 structure fill;
```

```
  FILL_2 bitfield length 4 fill prefix TTDEF tag $$;  
  ALTFRAME bitfield mask;  
  ALTRPAR bitfield mask;  
  PARITY bitfield mask;  
  ODD bitfield mask;  
  TWOSTOP bitfield mask;  
  DISPARERR bitfield mask;  
  ALTI SPAR bitfield mask;
```

```
end TTDEF BITS1;
```

```
TTDEF BITS2 structure fill;
```

```
  FILL_3 bitfield fill prefix TTDEF tag $$;  
  DS_DTR bitfield mask;  
  FILL_4 bitfield fill prefix TTDEF tag $$;  
  DS_SECTX bitfield mask;  
  DS_RTS bitfield mask;
```

```
end TTDEF BITS2;
```

```
TTDEF BITS3 structure fill;
```

```
  FILL_5 bitfield length 3 fill prefix TTDEF tag $$;  
  DS_SECRC bitfield mask;  
  DS_CTS bitfield mask;
```

```
/*OUTPUT MODEM CONTROL
```

```

    DS_CARRIER bitfield mask;
    DS_RING bitfield mask;
    DS_DSR bitfield mask;
end TTDEF_BITS3;

```

```
end TTDEF;
```

```
end_module $TTDEF;
```

```
module $TT2DEF;
```

```
aggregate TT2DEF union prefix TT2S;
```

```

TT2DEF_BITS structure fill;
  LOCALECHO bitfield mask;
  AUTOBALD bitfield mask;
  HANGUP bitfield mask;
  MODHANGUP bitfield mask;
  BRDCSTMBX bitfield mask;
  XON bitfield mask;
  DMA bitfield mask;
  ALTYPEAHD bitfield mask;
  SETSPEED bitfield mask;
  DCL_MAILBX bitfield mask;
  DCL_OUTBND bitfield mask;
  DCL_CTRLC bitfield mask;
  EDITING bitfield mask;
  INSERT BITFIELD MASK;
  FALLBACK BITFIELD MASK;
  DIALUP BITFIELD MASK;
  SECURE BITFIELD MASK;
  DISCONNECT BITFIELD MASK;
  PASTHRU BITFIELD MASK;
  SYSPWD BITFIELD MASK;
  SIXEL BITFIELD MASK;
  DRCS BITFIELD MASK;
  PRINTER BITFIELD MASK;
  APP_KEYPAD bitfield mask;
  ANSICRT bitfield mask;
  REGIS bitfield mask;
  BLOCK bitfield mask;
  AVO bitfield mask;
  EDIT bitfield mask;
  DECCRT bitfield mask;
  DECCRT2 bitfield mask;

```

```
end TT2DEF_BITS;
```

```
end TT2DEF;
```

```
end_module $TT2DEF;
```

```
/* TEMP DEFINITIONS FOR DCL SPAWN
```

```

/* enable advanced editing
/* INPUT EDITING INSTER DEFAULT
/* USE MULTINATIONAL FALLBACK
/* TERMINAL IS CONNECTED TO DIALUP
/* RECOGNIZE SECURE SERVER
/* TERMINAL CAN BE DISCONNECTED
/* passall with flowcontrol
/* System password required
/* SIXEL SUPPORTED
/*
/* PRINTER PORT AVAILABLE
/* Keypad in applicationis mode
/* ANSI, REGIS, BLOCK and AVO
/* must be contiguous for SCRPKG.

```

```
/* DECCRT level 2
```

```
module $UICDEF;
```

```
/*+
/*
/* Format of UIC - user identification code. May be either two part
/* group / member identifier, or a single uniform identifier.
/*
/*-
```

```
aggregate UICDEF union prefix UIC$;
```

```
UIC longword unsigned;          /* Full longword

UIC_FORM structure fill;        /* Group / member format
  constant MAX_UIC equals %X3FFFFFF; /* Highest possible UIC
  MEMBER bitfield length 16;      /* Member number
  constant WILD_MEMBER equals %XFFFF; /* Code for wild card member
  GROUP bitfield length 14;       /* Group number
  constant WILD_GROUP equals %X3FFF; /* Code for wild card group
  FORMAT bitfield length 2;       /* Format code
  constant UIC_FORMAT equals 0;    /* UIC format code
  constant ID_FORMAT equals 2;     /* ID format code
end UIC_FORM;
```

```
ID_FORM structure fill;        /* General identifier format
  ID_CODE bitfield length 28;     /* ID code
  FILL_1 bitfield length 2 fill;  /* Reserved ID bits
  FILL_2 bitfield length 2 fill;  /* Dummy to cover format code
  constant FIRST_ID equals %X80010000; /* First general identifier
  constant LAST_ID equals %X8FFFFFFF; /* Last general identifier
  constant MATCH_ALL equals %XFFFFFFFF; /* Match all identifiers
end ID_FORM;
```

```
end UICDEF;
```

```
end_module $UICDEF;
```

```
module $USGDEF;
```

```
/*+
/* Disk usage accounting file produced by ANALYZE/DISK_STRUCTURE utility.
/*-
```

```
aggregate USGDEF structure prefix USG$;
```

```
TYPE byte unsigned;           /* Record type
  constant "IDENT" equals 1 prefix USG tag $K; /* Identification record
  constant FILE equals 2 prefix USG tag $K; /* File record

SERIALNUM longword unsigned; /* Pack serial number
STRUCNAME character length 12; /* Volume set name
VOLNAME character length 12; /* Volume name
OWNERNAME character length 12; /* Volume owner name
FORMAT character length 12; /* Volume format type
/* *** Preceding same as home block ***

TIME quadword unsigned;      /* Time usage file created
  constant IDENT_LEN equals . prefix USG$ tag K; /* Length of IDENT record
  constant IDENT_LEN equals . prefix USG$ tag C; /* Length of IDENT record
```



end USGDEF;

aggregate USGDEF1 structure prefix USG\$;

FILL\_1 byte fill prefix USGDEF tag \$\$;

FILEOWNER\_OVERLAY union fill;

FILEOWNER longword unsigned;

/\* File owner UIC

FILEOWNER\_FIELDS structure fill;

UICMEMBER word unsigned;

/\* UIC member number

UICGROUP word unsigned;

/\* UIC group number

end FILEOWNER\_FIELDS;

end FILEOWNER\_OVERLAY;

ALLOCATED longword unsigned;

/\* Blocks allocated, including headers

USED longword unsigned;

/\* Blocks used

DIR\_LEN word unsigned;

/\* Length of directory string

SPEC\_LEN word unsigned;

/\* Length of complete file specification

FILESPEC character length 406;

/\* File spec "[dir]nam.typ;ver"

constant FILE\_LEN equals . prefix USG\$ tag K;

/\* Maximum length of FILE record

constant FILE\_LEN equals . prefix USG\$ tag C;

/\* Maximum length of FILE record

end USGDEF1;

end\_module \$USGDEF;

```
module $XADEF;
/*+
/* DR11-W DEFINITIONS FOR DEVICE SPECIFIC CHARACTERISTICS
/*-

aggregate XADEF union prefix XAS;
  XADEF_BITS structure fill;
  DATAPATH bitfield mask;
  LINK bitfield mask;
end XADEF_BITS;
end XADEF;

/* USE BUFFERED DATAPATH
/* INTERPROCESSOR LINK MODE

end_module $XADEF;
```

```
module $XFDEF;
```

```
/*  
/*  
/* DR32 DEFINITIONS FOR COMMAND TABLE, PACKETS, AND CHARACTERISTICS  
/*  
/*-
```

```
/*  
/* QIO COMMAND TABLE OFFSETS  
/*
```

```
aggregate XFDEF union prefix XFS;
```

```
XFDEF_BITS0 structure fill;
```

```
IOS_SUCCES bitfield mask; /* SUCCESS  
IOS_CMDSTD bitfield mask; /* COMMAND STARTED  
IOS_INVPTC bitfield mask; /* INVALID PTE  
IOS_FREQPK bitfield mask; /* FREE QUEUE PACKET  
IOS_DDIDIS bitfield mask; /* DDI DISABLE  
IOS_SLFTST bitfield mask; /* SELF TEST  
IOS_RNGERR bitfield mask; /* RANGE ERROR  
IOS_UNQERR bitfield mask; /* UNALIGNED QUEUE ERROR  
IOS_INVPKT bitfield mask; /* INVALID COMMAND PACKET  
IOS_FREQMT bitfield mask; /* FREE QUEUE EMPTY  
IOS_RNDENB bitfield mask; /* RANDOM ENABLE  
IOS_INVDDI bitfield mask; /* INVALID DDI COMMAND  
IOS_LENERR bitfield mask; /* LENGTH ERROR  
IOS_DRVABT bitfield mask; /* DRIVER ABORT  
IOS_PARERR bitfield mask; /* PARITY ERROR (CI OR DI)  
FILE_7 bitfield fill prefix XFDEF tag $$; /* RESERVED  
IOS_DDISTS bitfield length 8; /* DDI STATUS  
FILE_8 bitfield length 3 fill prefix XFDEF tag $$; /* RESERVED  
IOS_BUSERR bitfield mask; /* BUS ERROR  
IOS_RDSERR bitfield mask; /* READ DATA SUBSTITUTE ERROR  
IOS_WCSPE bitfield mask; /* WCS PARITY ERROR  
IOS_CPIPE bitfield mask; /* CONTROL INTERCONNECT PARITY ERROR  
IOS_DPIPE bitfield mask; /* DATA INTERCONNECT PARITY ERROR
```

```
end XFDEF_BITS0;
```

```
XFDEF_BITS1 structure fill;
```

```
FILL_9 bitfield length 5 fill prefix XFDEF tag $$; /* SKIP OVER 5 BITS  
IOS_NEXREG bitfield mask; /* NON-EXISTENT REGISTER  
IOS_LOG bitfield mask; /* LOG  
IOS_DDIERR bitfield mask; /* ERROR ON FAR-END DEVICE
```

```
end XFDEF_BITS1;
```

```
end XFDEF;
```

```
aggregate XFDEF1 structure prefix XFS;
```

```
CMT_CBLKSZ longword unsigned; /* COMMAND BLOCK SIZE  
CMT_CBLKAD longword unsigned; /* COMMAND BLOCK ADDRESS  
CMT_BBLKSZ longword unsigned; /* BUFFER BLOCK SIZE  
CMT_BBLKAD longword unsigned; /* BUFFER BLOCK ADDRESS  
CMT_PASTAD longword unsigned; /* PACKET AST ADDRESS  
CMT_PASTPM longword unsigned; /* PACKET AST ADDRESS  
CMT_RATE byte unsigned; /* DATA RATE  
CMT_FLAGS_OVERLAY union fill;
```

```

CMT_FLAGS byte unsigned;          /* FLAGS
CMT_FLAGS_BITS structure fill;
  CMT_SETRTE bitfield mask;        /* SET DATA RATE
  CMT_DIPEAB bitfield mask;        /* DISABLE PARITY ERROR ABORT
end CMT_FLAGS_BITS;
end CMT_FLAGS_OVERLAY;
FILL_1 word fill prefix XFDEF tag $$; /* SPARE
CMT_GBITAD longword unsigned;      /* GO BIT ADDRESS
constant CMT_LENGTH equals . prefix XF$ tag K; /* LENGTH OF COMMAND TABLE
constant CMT_LENGTH equals . prefix XF$ tag C; /* LENGTH OF COMMAND TABLE

/*
/* COMMAND PACKET OFFSETS
/*

end XFDEF1;

aggregate XFDEF2 structure prefix XF$:
PKT_FLINK longword unsigned;       /* FORWARD LINK
PKT_BLINK longword unsigned;       /* BACKWARD LINK
PKT_MSGLEN byte unsigned;          /* LENGTH OF DEVICE MESSAGE AREA
PKT_LOGLen byte unsigned;          /* LENGTH OF LOG AREA
PKT_CMDCTL_OVERLAY union fill;
  PKT_CMDCTL byte unsigned;        /* COMMAND CONTROL
  PKT_CMDCTL_BITS structure fill;
    PKT_FUNC bitfield length 4;    /* FUNCTION CODE
    FILL_2 bitfield length 4 fill prefix XFDEF tag $$; /* MUST BE ZERO
  end PKT_CMDCTL_BITS;

/* FUNCTION CODE VALUES
constant PKT_RD equals 0 prefix XF tag $K; /* READ
constant PKT_RDCHN equals 1 prefix XF tag $K; /* READ CHAINED
constant PKT_WRT equals 2 prefix XF tag $K; /* WRITE
constant PKT_WRTCHN equals 3 prefix XF tag $K; /* WRITE CHAINED
constant PKT_WRTCM equals 4 prefix XF tag $K; /* WRITE CONTROL MESSAGE
/* VALUE 5 IS RESERVED
constant PKT_SETTST equals 6 prefix XF tag $K; /* SET SELF TEST
constant PKT_CLRTST equals 7 prefix XF tag $K; /* CLEAR SELF TEST
constant PKT_NOP equals 8 prefix XF tag $K; /* NOP
constant PKT_DIAGRI equals 9 prefix XF tag $K; /* DIAGNOSTIC READ INTERNAL
constant PKT_DIAGWI equals 10 prefix XF tag $K; /* DIAGNOSTIC WRITE INTERNAL
constant PKT_DIAGRD equals 11 prefix XF tag $K; /* DIAGNOSTIC READ DDI
constant PKT_DIAGWC equals 12 prefix XF tag $K; /* WRITE CONTROL MESSAGE
constant PKT_SETRND equals 13 prefix XF tag $K; /* SET RANDOM ENABLE
constant PKT_CLRRND equals 14 prefix XF tag $K; /* CLEAR RANDOM ENABLE
constant PKT_HALT equals 15 prefix XF tag $K; /* HALT

end PKT_CMDCTL_OVERLAY;
PKT_PKTCTL_OVERLAY union fill;
  PKT_PKTCTL byte unsigned;        /* PACKET CONTROL
  PKT_PKTCTL_BITS structure fill;
    FILL_3 bitfield length 3 fill prefix XFDEF tag $$; /* UNUSED
    PKT_CISEL bitfield length 2;    /* CONTROL INTERCONNECT SELECT
    PKT_SLNERR bitfield mask;      /* SUPPRESS LENGTH ERROR
    PKT_INTCTL bitfield length 2;   /* INTERRUPT CONTROL
  end PKT_PKTCTL_BITS;

/* VALUES FOR CONTROL INTERCONNECT SELECT

```

```

constant PKT_NOTRAN equals 0 prefix XF tag $K; /* NO TRANSMISSION
constant PKT_CB equals 1 prefix XF tag $K; /* COMMAND BYTE ONLY
constant PKT_CBDM equals 2 prefix XF tag $K; /* COMMAND BYTE AND DEVICE MESSAGE
constant PKT_CBDMBC equals 3 prefix XF tag $K; /* CMD. BYTE, DEV. MSG, AND BYTE COUNT
/* VALUES FOR INTERRUPT CONTROL
constant PKT_UNCOND equals 0 prefix XF tag $K; /* UNCONDITIONAL INTERRUPT
constant PKT_TMQMT equals 1 prefix XF tag $K; /* INTERRUPT ON TERM. QUEUE EMPTY
constant PKT_NOINT equals 2 prefix XF tag $K; /* NO INTERRUPT
end PKT_PKTCTL_OVERLAY;
PKT_BFRSIZ longword unsigned; /* BUFFER SIZE
PKT_BFRADR longword unsigned; /* BUFFER ADDRESS
PKT_RMBCNT longword unsigned; /* RESIDUAL MEMORY BYTE COUNT
PKT_RDBCNT longword unsigned; /* RESIDUAL DDI BYTE COUNT
PKT_DSL_OVERLAY union fill;
PKT_DSL longword unsigned; /* DR32 STATUS LONGWORD
PKT_DSL_BITS0 structure fill;
PKT_SUCCES bitfield mask; /* SUCCESS
PKT_CMDSTD bitfield mask; /* COMMAND STARTED
PKT_INVPTC bitfield mask; /* INVALID PTE
PKT_FREQPK bitfield mask; /* FREE QUEUE PACKET
PKT_DDIDIS bitfield mask; /* DDI DISABLE
PKT_SLFTST bitfield mask; /* SELF TEST
PKT_RNGERR bitfield mask; /* RANGE ERROR
PKT_UNQERR bitfield mask; /* UNALIGNED QUEUE ERROR
PKT_INVPKT bitfield mask; /* INVALID COMMAND PACKET
PKT_FREQMT bitfield mask; /* FREE QUEUE EMPTY
PKT_RNDENB bitfield mask; /* RANDOM ENABLE
PKT_INVDDI bitfield mask; /* INVALID DDI COMMAND
PKT_LENERR bitfield mask; /* LENGTH ERROR
PKT_DRVABT bitfield mask; /* DRIVER ABORT
PKT_PARERR bitfield mask; /* PARITY ERROR
FILE 4 bitfield fill prefix XFDEF tag $$; /* RESERVED
PKT_DDISTS bitfield length 8; /* DDI STATUS
FILE 5 bitfield length 8 fill prefix XFDEF tag $$; /* RESERVED
end PKT_DSL_BITS0;
PKT_DSL_BITS1 structure fill;
FILE 6 bitfield length 5 fill prefix XFDEF tag $$; /* SKIP OVER 5 BITS
PKT_NEXREG bitfield mask; /* NON-EXISTENT REGISTER
PKT_LOG bitfield mask; /* LOG
PKT_DDIERR bitfield mask; /* ERROR ON FAR-END DEVICE
end PKT_DSL_BITS1;
end PKT_DSL_OVERLAY;
PKT_DEVMSG Byte unsigned; /* START OF DEVICE MESSAGE

/*
/* SECOND LONGWORD OF I/O STATUS BLOCK DEFINITIONS
/*
end XFDEF2;
end_module $XFDEF;

```

```
module $XKSTSDEF;
```

```
/*
/* DEFINITIONS FOR 3271 LINE STATUS BLOCK (RETURNED BY IO$_RDSTATS)
/*
```

```
aggregate XKSTSDEF structure prefix XKSTSS;
  constant LSTR equals . prefix XKSTSS tag K; /* OFFSET OF BEGINNING OF DATA AREA
  constant LSTR equals . prefix XKSTSS tag C; /* OFFSET OF BEGINNING OF DATA AREA
  CUAD byte unsigned; /* CONTROL UNIT ADDRESS (HEX EBCDIC)
  LSTS_OVERLAY union fill;
    [STS word unsigned; /* LINE STATUS
    LSTS_BITS structure fill;
      ACT bitfield mask; /* IF SET, LINE IS 'ON' (ACTIVE IN 3271 MODE)
      DGN bitfield mask; /* IF SET, LINE IS EXECUTING A DIAGNOSTIC QIO
    end LSTS_BITS;
  end LSTS_OVERLAY;
  PLSZ byte unsigned; /* POOL SIZE
  BSIZ word unsigned; /* MAXIMUM BLOCK TO BE RECEIVED FROM IBM
  ECNT byte unsigned; /* ENQ THRESHOLD
  TNAK word unsigned; /* NUMBER OF NAKS TRANSMITTED
  RNAK word unsigned; /* NUMBER OF NAKS RECEIVED
  TENQ word unsigned; /* NUMBER OF ENQS SENT
  RENQ word unsigned; /* NUMBER OF ENQS RECEIVED
  TRVI word unsigned; /* NUMBER OF RVIS TRANSMITTED
  RRVI word unsigned; /* NUMBER OF RVIS RECEIVED
  RCNV word unsigned; /* NUMBER OF CONVERSATIONAL MODE RESPONSES
  /* RECEIVED
  CHAI word unsigned; /* NUMBER OF CHAINED WRITES RECEIVED
  GPOL word unsigned; /* NUMBER OF GENERAL POLLS RECEIVED
  XACK word unsigned; /* NUMBER OF INVALID ACKS RECEIVED
  HABO word unsigned; /* NUMBER OF HOST ABORTS ENCOUNTERED
  DABO word unsigned; /* NUMBER OF DRIVER ABORTS ENCOUNTERED
  RTIM word unsigned; /* NUMBER OF ACK RESPONSE TIMEOUTS
  DISC word unsigned; /* NUMBER OF DISCONNECTS ENCOUNTERED
  HARD word unsigned; /* NUMBER OF HARDWARE FAILURES (INCLUDING
  /* DISCONNECTS) ENCOUNTERED
  TBLK word unsigned; /* NUMBER OF BLOCKS TRANSMITTED
  RBLK word unsigned; /* NUMBER OF BLOCKS RECEIVED
  HABX word unsigned; /* NUMBER OF DUBIOUS ACK OUTS
  IEOT word unsigned; /* NUMBER OF EOTS IMPLYING ACK
  constant DSTR equals . prefix XKSTSS tag K; /* LENGTH OF THE LINE STATUS BLOCK
  constant DSTR equals . prefix XKSTSS tag C; /* LENGTH OF THE LINE STATUS BLOCK
end XKSTSDEF;
```

```
end_module $XKSTSDEF;
```

```
module $XKDEVDEF;
```

```
/*
/* 3271 DEVICE STATUS BLOCK
/*
/* THE DRIVER RETURNS N COPIES OF THIS BLOCK FOLLOWING THE LINE STATUS BLOCK
/* IN THE USER'S BUFFER, WHERE N IS THE NUMBER OF ACTIVE DEVICES ON THE LINE.
```

/\*

aggregate XKDEVDEF structure prefix XKDEV\$;

DUEB byte unsigned;

DPID longword unsigned;

TBLK word unsigned;

RBLK word unsigned;

SPOL word unsigned;

ASEL word unsigned;

CREJ word unsigned;

constant SIZE equals . prefix XKDEV\$ tag K;

constant SIZE equals . prefix XKDEV\$ tag C;

end XKDEVDEF;

end\_module \$XKDEVDEF;

/\* DEVICE ADDRESS (DU - HEX EBCDIC)  
/\* PROCESS ID OF PROCESS OWNING THIS DU  
/\* NUMBER OF BLOCKS TRANSMITTED  
/\* NUMBER OF BLOCKS RECEIVED  
/\* NUMBER OF SPECIFIC POLLS RECEIVED  
/\* NUMBER OF ADDRESS SELECTS RECEIVED  
/\* NUMBER OF COMMAND REJECTS SENT  
/\* SIZE OF THE DEVICE STATUS BLOCK  
/\* SIZE OF THE DEVICE STATUS BLOCK

STA

E

E

E

```
module $XMDEF;
```

```
/*
/* DEFINITIONS FOR STATUS, CHARACTERISTICS AND ERRORS FOR DMC-11/DMR-11/DMP-11
/*
```

```
aggregate XMDEF union prefix XMS;
```

```
XMDEF_BITS0 structure fill;
```

```
CHR_MOP bitfield mask; /* MOP MODE
CHR_LOOPB bitfield mask; /* LOOP BACK
CHR_HDPLX bitfield mask; /* HALF DUPLEX
CHR_SLAVE bitfield mask; /* HALF D' PLEX SLAVE STATION
CHR_MBX bitfield mask; /* MAILBOX IS ENABLED
CHR_DMC bitfield mask; /* DMC-11 COMPATABILITY MODE
CHR_CTRL bitfield mask; /* CONTROL STATION
CHR_TRIB bitfield mask; /* TRIBUTARY STATION

STS_DCHK bitfield mask; /* DATA CHECK OCCURED
STS_TIMO bitfield mask; /* TIMEOUT OCCURED
STS_ORUN bitfield mask; /* DATA OVER RUN OCCURED
STS_ACTIVE bitfield mask; /* ACTIVE UNIT
STS_BUFFAIL bitfield mask; /* BUFFER ALLOCATION FAILURE OCCURED
STS_RUNNING bitfield mask; /* PROTOCOL RUNNING
STS_DISC bitfield mask; /* LINE DISCONNECT OCCURED
FILC_1 bitfield fill prefix XMDEF tag $$; /* SPARE
```

```
ERR_FATAL bitfield mask; /* HARDWARE ERROR
FILC_2 bitfield length 2 fill prefix XMDEF tag $$;
ERR_MAINT bitfield mask; /* DDCMP MAINT MESSAGE RECEIVED
ERR_LOST bitfield mask; /* DATA LOST BECAUSE OF BUFFER SIZE MISMATCH
ERR_THRESH bitfield mask; /* THRESHOLD ERRORS
ERR_TRIB bitfield mask; /* TRIBUTARY MALFUNCTIONING
ERR_START bitfield mask; /* DDCMP START MESSAGE RECEIVED
```

```
end XMDEF_BITS0;
```

```
XMDEF_BITS1 structure fill;
```

```
MDM_RTSHLD bitfield mask; /* RTS HOLD
MDM_STNDBY bitfield mask; /* SELECT STANDBY
MDM_MAINT2 bitfield mask; /* MAINTENANCE MODE 2
MDM_MAINT1 bitfield mask; /* MAINTENANCE MODE 1
FILC_3 bitfield fill prefix XMDEF tag $$; /* HALF-DUPLEX
MDM_FREQ bitfield mask; /* SELECT FREQUENCY
MDM_RDY bitfield mask; /* DATA TERMINAL READY
MDM_POLL bitfield mask; /* SELECT POLLING MODEM MODE

MDM_SELM bitfield mask; /* MODEM INTERFACE PROGRAM SELECTED
FILC_4 bitfield length 2 fill prefix XMDEF tag $$; /* UNUSED
MDM_INT bitfield mask; /* INTEGRAL MODEM
MDM_V35 bitfield mask; /* V.35
FILC_5 bitfield fill prefix XMDEF tag $$; /* UNUSED
MDM_RS232 bitfield mask; /* RS-232-C OR RS423
MDM_RS422 bitfield mask; /* RS-422
```

```
end XMDEF_BITS1;
```

```
XMDEF_BITS2 structure fill;
```

E

E



```

MDM_CARRDET bitfield mask;      /* CARRIER DETECT
MDM_MSTNDBY bitfield mask;      /* MODEM STANDBY
MDM_CTS bitfield mask;          /* CLEAR TO SEND
MDM_DSR bitfield mask;          /* DATA SET READY
MDM_HDX bitfield mask;          /* HALF-DUPLEX
MDM_RTS bitfield mask;          /* REQUEST TO SEND
MDM_DTR bitfield mask;          /* DATA TERMINAL READY
MDM_RING bitfield mask;         /* RING

MDM_CHRMOD bitfield mask;       /* CHARACTER OR BIT PROTOCOL MODE
MDM_MCLOCK bitfield mask;       /* MAINTENANCE CLOCK
MDM_MODTEST bitfield mask;      /* MODEM TEST MODE
FILC_6 bitfield length 3 fill prefix XMDEF tag $$; /* UNUSED
MDM_SIGQUAL bitfield mask;      /* SIGNAL QUALITY
MDM_SIGRATE bitfield mask;      /* SIGNAL RATE
end XMDEF_BITS2;
end XMDEF;

end_module $XMDEF;

module $XWDEF;

/*++
/*
/* XWDEF - System definitions for software DDCMP
/*
/*--

aggregate XWDEF union prefix XW$;
  XWDEF_BITSO structure fill;
    CRA_FDX bitfield mask;       /* Full duplex line
    CHA_XPR bitfield mask;       /* Transparency
    CHA_DSR bitfield mask;       /* Data Set Ready set
  end XWDEF_BITSO;

/*
/* Device dependent status
/*
  XWDEF_BITS1 structure fill;
    FAT_STAT bitfield mask length 10; /* Beginning of fatal stat flags
    INFO_STAT bitfield mask length 6; /* Start of informational status
  end XWDEF_BITS1;

/*
/* Bit definitions for fatal and informational status fields
/*
  XWDEF_BITS2 structure fill;
    EOT bitfield mask;           /* EOT received
    DATAACK bitfield mask;      /* Retry threshold exceeded
    NODSR bitfield mask;         /* Line not connected
    DISCON bitfield mask;        /* Disconnect (DLE, EOT) rcvd
    TRABINTMO bitfield mask;     /* Binary transmit timeout
    PIPE_MARK bitfield mask;     /* Aborted because of pipe mark
    BADCRAIN bitfield mask;      /* Bad record list
    ILLMOD bitfield mask;        /* Illegal QIO modifier
    FILL_1 bitfield length 2 fill prefix XWDEF tag $$;

```

```
RVI bitfield mask;          /* Buffer ACKed with RVI
CONACK bitfield mask;      /* Buffer ACKed with conver resp
XPR bitfield mask;         /* Buffer rcvd with transparency
ETXEND bitfield mask;      /* Rcvd block ended with ETX
FILL_2 bitfield length 2 fill prefix XWDEF tag $$;
PUNCR bitfield mask;       /* Punch select found
HOR FOR bitfield mask;     /* Horizontal forms record rcvd
end XWDEF_BITS2;

/*
/* Read/Write function modifier definition
/*
XWDEF_BITS3 structure fill;
  FILL_3 bitfield length 13 fill prefix XWDEF tag $$;
  IOMOD bitfield mask length 3; /* I/O modifier field start
end XWDEF_BITS3;
end XWDEF;

end_module $XWDEF;
```



The image displays a grid of 100 small panels, each containing a different type of data visualization or code snippet. The panels are arranged in a 10x10 grid. Some panels are clearly labeled with titles such as 'STARDEFLL SDL', 'OPCDEF SDL', 'SCRDEF SDL', 'SRMDEF SDL', 'STARDEFMP SDL', and 'STARDEFQZ SDL'. The visualizations include bar charts, line graphs, tables, and code blocks. The overall appearance is that of a technical manual or a reference guide for data analysis software.



