

CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCCCCCCCCCCC	LLL	IIIIIIII	UUU	UUU	TTTTTTTTTTTTTTTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCC	LLL	III	UUU	UUU	TTT	LLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL
CCCCCCCCCCCC	LLLLLLLLLLLLLLLL	IIIIIIII	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	TTTT	LLLLLLLLLLLLLLLL

```

CCCCCCCC LL      IIIIII UU      UU      TTTTTTTTTT LL      MM      MM      AAAAAA CCCCCCCC
CCCCCCCC LL      IIIIII UU      UU      TTTTTTTTTT LL      MM      MM      AAAAAA CCCCCCCC
CC        LL      II      UU      UU      TT      LL      MMMM  MMMM  AA      AA  CC
CC        LL      II      UU      UU      TT      LL      MMMM  MMMM  AA      AA  CC
CC        LL      II      UU      UU      TT      LL      MM   MM   MM   AA      AA  CC
CC        LL      II      UU      UU      TT      LL      MM   MM   MM   AA      AA  CC
CC        LL      II      UU      UU      TT      LL      MM   MM   MM   AA      AA  CC
CC        LL      II      UU      UU      TT      LL      MM   MM   MM   AA      AA  CC
CC        LL      II      UU      UU      TT      LL      MM   MM   MM   AA      AA  CC
CC        LL      II      UU      UU      TT      LL      MM   MM   MM   AA      AA  CC
CCCCCCCC LLLLLLLLLL IIIIII UUUUUUUUUU TT      LLLLLLLLLL MM   MM   AA      AA  CC
CCCCCCCC LLLLLLLLLL IIIIII UUUUUUUUUU TT      LLLLLLLLLL MM   MM   AA      AA  CC

```

```

MM      MM      AAAAAA RRRRRRRR
MM      MM      AAAAAA RRRRRRRR
MMMM  MMMM  AA      AA  RR      RR
MMMM  MMMM  AA      AA  RR      RR
MM   MM   MM   AA      AA  RR      RR
MM   MM   MM   AA      AA  RRRRRRRR
MM   MM   MM   AA      AA  RRRRRRRR
MM   MM   MM   AAAAAAAAAA RR  RR
MM   MM   MM   AAAAAAAAAA RR  RR
MM   MM   MM   AA      AA  RR      RR
MM   MM   MM   AA      AA  RR      RR
MM   MM   MM   AA      AA  RR      RR
MM   MM   MM   AA      AA  RR      RR

```

```

....
....
....
....

```

.TITLE CLIUTLMAC - CLI UTILITY COMMAND MACRO FILE
.IDENT '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: STARLET CLI
* ABSTRACT: MACRO LIBRARY
* ENVIRONMENT:
* AUTHOR: W.H.BROWN, CREATION DATE: 7-JUN-1977
* MODIFIED BY:
*
* V02-003 MLJ0070 Martin L. Jack, 24-Jan-1982 3:13
* Remove unused variables from SUBMIT context block, add
* new ones to support queue name translation and /NAME vs. /LOG
* semantics cleanup.
*
* 101 T.HALVORSEN 08-NOV-1978
* REMOVED UNUSED 2 ARGS FROM DEF_TAG CALLS IN TYPCTXBLK
*
* 102 G.FOWLER 04-SEP-1979
* MODIFIED DEF OF QMNBITS TO ALLOW FOR TEST OF CONFLICTING
* QUALIFIERS ON STOP COMMAND.
*
* --

```

.SBTTL DECLARATIONS

INCLUDE FILES:
NONE

MACROS:

BUG_CHK - MACRO TO HANDLE AN UN-EXPECTED ERROR

```
.MACRO BUG_CHK ?L1
BLBS RO,L1
$EXIT_S RO
```

L1:

.ENDM

MACRO TO GENERATE A OFFSET LIST FOR A DATA STRUCTURE

IT IS USEFUL FOR INPUT ARGUMENT LISTS POSITIVELY INDEXED FROM AP, AND
WORK AREAS ALLOCATED IN CALL STACK AND NEGATIVELY INDEXED FROM FP.

CALL: \$OFFSET INITIAL,DIRECTION,<<LAB1,[SIZE]>>,...,<LABN,[SIZE]>>

WHERE: INITIAL IS A REQUIRED VALUE FOR THE INTIAL INDEX WHEN
ORIGINATING A DATA STRUCTURE DEFINITION. IT IS NORMALLY
(+) 4 FOR ARGUMENT LISTS AND 0 FOR WORK AREAS.

DIRECTION IS A KEYWORD THAT MUST BE:
POSITIVE - FOR STRUCTURES GROWING UP IN MEMEORY
NEGATIVE - FOR STRUCTURES GROWING DOWN IN MEMEORY
OR BLANK, IN WHICH CASE "POSITIVE" IS ASSUMED.

THE LABEL, SIZE LIST IS THE SYMBOLIC NAME FOR THE LOCATION
AND THE OPTIONAL SIZE OF THE ELEMENT. IF BLANK, SIZE IS
ASSUMED TO BE 4 (ONE LONGWORD).

TO PERMIT THE DEFINITION OF AN INDEFINITLY LARGE NUMBER OF LABELS,
THE MACRO MAY BE CONTINUED. IN THIS CASE THE "INITIAL" AND
'DIRECTION' ARGUMENTS MUST BE BLANK.

```
.MACRO $OFFSET INITVALUE,DIRECTION,SYMLST
.SAVE
.PSECT $ABSS ABS
.IF B,INITVALUE
.IF NB,DIRECTION
.ERROR : DIRECTION MUST BE BLANK WHEN CONTIUING;
.MEXIT
.ENDC
.IFF
DIR...=1
.=INITVALUE
.IF NB,DIRECTION
.IF IDN <DIRECTION>,<POSITIVE>
.IFF
.IF IDN <DIRECTION>,<NEGATIVE>
```

```

DIR...=-1
.IFF
.ERROR ; 'DIRECTION' MUST BE 'POSITIVE','NEGATIVE', OF BLANK;
.ENDC
.ENDC
.ENDC
.ENDC
.IRP SYM,<SYMLST>
$OFFST1 SYM
.ENDR
.RESTORE
.ENDM $OFFSET
    
```

```

.MACRO $OFFST1 SYM,SIZ=4
.IF LT,SIZ
.ERROR ;***** SIZ PARAMETER NEGATIVE *****;
.ENDC
.IF LT,DIR...
.BLKB -SIZ
.ENDC
.IF NB,SYM
.LIST MEB
SYM:
.NLIST MEB
.ENDC
.IF GT,DIR...
.BLKB SIZ
.ENDC
.ENDM $OFFST1
    
```

```

:
: *
: INTERNAL MACRO _SETATA
: THIS MACRO IS USED TO INITIALIZE A DATA AREA
: CALL: _SETATA TYPE,LIST
: WHERE: TYPE IS 'B','W','L','A','Q','CS','T'
: INDICATES THE TYPE OF DATA FOR INITIALIZATION
: LIST IS THE VALUES TO BE INITIALIZED
:
: -
    
```

```

.MACRO _SETATA TYPE=L,LIST
.IRP -ZZ-,<LIST>
.IIF IDN <TYPE><B>, .BYTE -ZZ-
.IIF IDN <TYPE><W>, .WORD -ZZ-
.IIF IDN <TYPE><L>, .LONG -ZZ-
.IIF IDN <TYPE><A>, .LONG -ZZ-
.IIF IDN <TYPE><Q>, .QUAD -ZZ-
.IIF IDN <TYPE><CS>, .ASCII ?-ZZ-!
.IIF IDN <TYPE><T>, .ASCII !-ZZ-!
.ENDR
.ENDM
    
```

```

:
: LOCAL MACROS TO GENERATE COUNTED STRING OR TEXT BLOCK
:
: .MACRO .BLKCS VAL
: .BLKB VAL
    
```

```
.ENDM
.MACRO .BLKT VAL
.BLKB VAL
.ENDM
```

```
.*
: MACRO DEF_CTXBLK
: THIS MACRO IS USED TO SET THE BASE OF THE CONTEXT BLOCK
: CALL: DEF_CTXBLK NAME [,OFFSET] [,ALLOCATE] [,GBL]
: WHERE: NAME IS THE POINT OF REFERENCE IN THE BLOCK
: OFFSET IS AN OPTION VALUE INDICATING THE OFFSET FORM THE
: LOWEST ADDRESS ASSIGNED TO THE BLOCK TO THE ZEROth INDEX
: ALLOCATE IS A KEYWORD, THAT IF EQUAL TO YES ALLOCATES THE BLOCK
: GBL IS A FLAG THAT IF IDENTICAL TO "YES" WILL CAUSE THE
: BLOCK REFERENCE LABEL TO BE DEFINED GLOBALLY.
:-
```

```
.MACRO DEF_CTXBLK NAME,OFFSET=0,ALLOCATE=NO,GBL=NO
_GBL_ = 0
_ALL_ = 0
: IF IDN <GBL><YES>
_GBL_ = 1
.GLOBL NAME
.ENDC
: IF IDN <ALLOCATE><YES>
_ALL_ = 1
NAME = . + OFFSET
_BASE_ = NAME
: IFF
: SAVE
.PSECT $ABS$ ABS
: = 0
_BASE_ = OFFSET
.ENDC
.ENDM
```

```
.*
: MACRO DEF_CTXEND - END OF CONTEXT BLOCK
:-
```

```
.MACRO DEF_CTXEND
.IF EQ <_ALL_>, .RESTORE
.ENDM
```

```
.*
: MACRO DEF_ELE
: THIS MACRO IS USED TO DEFINE A ELEMENT IN THE CONTEXT BLOCK
: CALL: DEF_ELE NAME,TYPE,SIZE,VALIST
:
```

```

: WHERE:      NAME IS THE SYMBOLIC INDEX INTO THE BLOCK
:             TYPE IS 'B','W','L','A','Q','CS','T'
:             SIZE IS THE NUMBER OF DATA CELLS IN THE ELEMENT
:             VALIST IS THE LIST OF INITIAL VALUES FOR THE CELLS
:-

```

```

: .MACRO DEF_ELE NAME,TYPE = L,SIZE=1,VALIST
: .IF EQ GBL_
NAME = . - BASE_
: .IFF
NAME == . - BASE_
: .ENDC
:_TOP_ = .BLK'TYPE      SIZE
:_END_ = .
: .IF EQ <ALL -1>
: .IF NB <ZVALIST>
. = _TOP_
:_SETATA TYPE,<VALIST>
. = _END_
: .ENDC
: .ENDC
: .ENDM

```

```

: +
: .MACRO DEF_TAG

```

```

: THIS MACRO IS USED TO DEFINE AN OFFSET INTO THE BLOCK
: WITHOUT ALLOCATING ANY SPACE. THIS IS USED TO THE SAME AREA
: ADDRESS IN DIFFERENT WAYS, SUCH AS, EITHER A QUADWORD, OR
: AS WORD,WORD,LONGWORD IN THE CASE OF A STRING DESCRIPTOR.
:-

```

```

: .MACRO DEF_TAG NAME
: .IF EQ GBL_
NAME = . - BASE_
: .IFF
NAME == . - BASE_
: .ENDC
: .ENDM

```

```

: +
: .MACRO $KEYWORD

```

```

: THIS MACRO IS USED TO GENERATE A STRING FOR THE MATCHKEY SUBROUTINE.
: THE STRING IS A COUNTED STRING, BUT MACRO NAME IS UNIQUE TO PREVENT
: CONFLICTS IN THE FUTURE.
:-

```

```

: $KEYWORD      ST,?L1
: .BYTE      .-L1-1
: .ASCII      \ST\
L1:
: .ENDM

```

```

: +
: LOCAL MACRO LIMITS-DETERMINE THE LIMITS OF THE CASE SELECTION

```

```

:-
.MACRO LIMITS X,Y
.IIF GT <_Z_LO-X>, _Z_LO=X
.IIF LT <_Z_HI-X>, _Z_HI=X
.ENDM
:
: LOCAL MACRO SETCASE-SET A CASE DISPATCH AT THE PROPER INDEX IN THE CASE TABLE
:

```

```

.MACRO _SETCASE X,Y
.= Z ..+<<X-_Z_LO>*2>
.WORD Y-_Z_..
.ENDM

```

```

:
: MACRO SELECT_CASE - MACRO TO GENERATE A CASE TABLE WITH INPUTS
: CONSISTING OF VALUES, ADDRESS PAIRS. THE TABLE IS BUILT FROM THE
: VALUES SUPPLIED, SETTING LIMITS AND UN-USED VALUES CORRECTLY.
: THE VALUE/ADDRESS PAIRS NEED NOT BE SUPPLIED IN ANY SPECIFIC ORDER.
:

```

```

: CALL: SELECT_CASE VAL,<VAL_ADR_LIST>,TYPE
:

```

```

: WHERE: "VAL" IS THE MACRO SOURCE OPERAND USED FOR CASE SELECTION
:
: "VAL ADR_LIST" IS A LIST OF PAIRS OF VALUES AND ASSOCIATED
: ROUTINE ADDRESSES FROM WHICH TO SELECT.
:
: "TYPE" IS AN OPTIONAL KEY VALUE THAT, IF SPECIFIED TO BE
: BYTE CAUSES THE CASE INSTRUCTION TO BE GENERATED WITH
: BYTE CONTEXT, RATHER THAN THE DEFAULT WORD CONTEXT.
:

```

```

.MACRO SELECT_CASE VAL,LIST,TYPE
_Z_LO=1000000
_Z_HI=0
.IRP _Z_,<LIST>
.LIMITS _Z_
.ENDR
_Z_NU=_Z_HI-_Z_LO+1
.IF IDN <TYPE><BYTE>
CASEB VAL,#<_Z_LO>,#<_Z_NU-1>
.IFF
CASEW VAL,#<_Z_LO>,#<_Z_NU-1>
.ENDC
_Z_...-.
.REPT _Z_NU
.WORD Z-_Z_NU*2>
.ENDR
.IRP _Z_,<LIST>
.SETCASE _Z_
.ENDR
.ENDM

```

```

: MACRO TO GENERATE STRING TO BE USED IN A SYSTEM CALL
:

```



```

      .MACRO SYS_STR STRING,?L1,?L2
      .WORD <L2-L1>,0
      .LONG L1
L1:   .ASCII \STRING\
L2:
      .ENDM

```

```

: MACRO TO GENERATE A COUNTED STRING
:

```

```

      .MACRO CNT_STR ST,?L1
      .BYTE L1-1
L1:   .ASCII \ST\
      .ENDM

```

```

: MACRO TO PERFORM A COMAPIR OF 2 QUADWORDS
:

```

```

      NOTE: DISPLACEMENT MODE IS ASSUMED
      .MACRO CMPQUAD ARG1,ARG2,?L1
      CMPL 4+ARG1,4+ARG2
      BNEQ L1
L1:   CMPL ARG1,ARG2
      .ENDM

```

```

: MACRO TO SET OR CLEAR A BIT BY BIT NUMBER
:

```

```

CALL: SETBIT BITNUM,FLAGWORD
OR:   CLRBIT BITNUM,FLAGWORD

```

```

WHERE: BITNUM IS ANY VALID SOURCE OPERAND SECIFYING THE BIT
        OFFSET FROM THE FLAG BASE TO SET/CLEAR

```

```

        FLAGWORD IS ANY VALID DESTINATION OPERAND

```

```

      .MACRO SETBIT VAL,FLAG
      .NTYPE $$ VAL
      .IF EQ Z $$ _ ^XOEF>
      .IF NDF VAL
      BBSS S^#VAL,FLAG,..+1
      .IFF
      .IF LT <VAL-8>
      BISB #<1@VAL>,FLAG
      .IFF
      BBSS #VAL,FLAG,..+1
      .ENDC
      .ENDC
      .IFF
      BBSS VAL,FLAG,..+1
      .ENDC
      .ENDM SETBIT

```

```

:
: .MACRO CLRBIT VAL,FLAG
: .NTYPE $$ VAL
: .IF EQ 2 $$_ -^XOEF>
: .IF NDF VAL
BBCC S^#VAL,FLAG,..+1
: .IFF
: .IF LT <VAL-8>
BICB #<1@VAL>,FLAG
: .IFF
BBCC #VAL,FLAG,..+1
: .ENDC
: .ENDC
: .IFF
BBCC VAL,FLAG,..+1
: .ENDC
: .ENDM CLRBIT

```

```

: PROGRAM SECTION DEFINITION MACROS
: ARGUMENTS ARE:

```

- ```

: 1) SECTION NAME (KEY WORD IS NAME)
: 2) ALIGNMENT (KEY WORD IS ALIGN)

```

```

: IN ALL CASE, ARGUMENTS ARE OPTIONAL

```

```

: MACRO TO GENERATE A PURE PROGRAM SECTION

```

```

: .MACRO PURE_SECTION NAME=PURE,ALIGN=BYTE
: .PSECT NAME EXE,RD,NOWRT,ALIGN
: .ENDM PURE_SECTION

```

```

: MACRO TO GENERATE IMPURE DATA SEGMENT

```

```

: .MACRO IMPURE_DATA NAME=RWDATA,ALIGN=LONG
: .PSECT NAME NOEXE,WRT,RD
: .ENDM IMPURE_DATA

```

```

: MACRO TO GENERATE A STRING WITH DESCRIPTOR

```

```

: STRING_DESC <STRING>

```

```

: WHERE:

```

```

: <STRING> IS THE STRING TO BE USED

```

```

: .MACRO STRING_DESC ST
: .NCHR $$.,<ST>
: .LONG $$
: .LONG +4
: .ASCII \ST\

```

.ENDM

MACRO TO DEFINE BITS FOR QUEUE MANAGER

.MACRO QMNBITS

```

_VIELD QMN,1,<- : CLI CONFLICT BITS
 BATCH,- : OPTION APPLIES TO BATCH QUEUE
 DEVICE,- : OPTION APPLIES TO DEVICE QUEUE
 START,- : OPTION IS FOR START
 RESTART,- : OPTION IS FOR RESTART
 QUEUE,- : OPTION IS FOR STOP COMMAND
 REQUE,- : OPTION IS FOR STOP COMMAND
 >

```

```

.MACRO QMNBITS
.ENDM QMNBITS
.ENDM QMNBITS

```

DEFINE CONTEXT BLOCK FOR SUBMIT

```

.MACRO SMTCTXBLK ALLO=NO
.IIF IDN <ALLO><YES>, .GLOBL SMT$CTXBLK
DEF CTXBLK SMT$CTXBLK,ALLO=ALLO
DEF_ELE SMT_W_FLAGS,W : FLAGS
DEF_ELE SMT_W_RSPCHAN,W : MAILBOX CHANNEL FOR RESPONCES FROM JBC
DEF_ELE SMT_L_STATUS : IMAGE STATUS
DEF_ELE SMT_Q_QUEDES,Q : Descriptor for queue name
DEF_ELE SMT_L_PRIPTR : Pointer to ASCIC /PRINT qualifier
DEF_ELE SMT_Q_NAMEDES,Q : Descriptor for /NAME qualifier
DEF_ELE SMT_Q_LOGDES,Q : Descriptor for /LOG qualifier
DEF_ELE SMT_Q_IOSB,Q : IOSB FOR QIO'S
DEF_ELE SMT_L_MSGPTR : POINTER TO NEXT BYTE FREE IN MEASSAGE
DEF_ELE SMT_W_JOBNUM,W : JOB CONTROLLER ASSIGNED NUMBER
DEF_ELE SMT_B_SPARE,B : Spare byte
DEF_ELE SMT_B_BITMAP,B : CLI BITS HERE
DEF_ELE SMT_T_DEVNAM,CS,16 : DEVICE NAME HERE-COUNTED STRING
DEF_CTXEND

```

DEFINE COMMON PARAMETERS

```

_VIELD SMT,1,<- : SUBMIT COMMAND FLAGS
 <NAME,,M>,- : USER SUPPLIED A NAME
 <IDENT,,M>,- : SHOW IDENT AT END OF SUBMIT
 <REMOTE,,M>,- : "/REMOTE" GIVEN
 <PRTCMD,,M>,- : DOING A PRINT COMMAND
 >

```

```

.MACRO SMTCTXBLK
.ENDM SMTCTXBLK
.ENDM SMTCTXBLK

```

```
: GENERATE A CONTEXT BLOCK FOR THE IMAGE "TYPE"
```

```
:
: .MACRO TYPCTXBLK ALLO=NO
.IIF IDN <ALLO><YES>, .GLOBL TYP$CTXBLK
DEF_CTXBLK TYP$CTXBLK,ALLOCATE=ALLO ; DEFINE START OF CONTEXT BLOCK
DEF_ELE TYP_L_STATUS,L,1,1 ; SAVE AREA FOR FINAL STATUS
DEF_TAG TYP_Q_BUFDES ; BUFFER DESCRIPTOR
DEF_ELE TYP_W_BUFcnt,W ; BUFFER COUNT WORD
DEF_ELE TYP_W_BUFSIZ,W,1,LINBUFSIZ ; SIZE KEPT IN SPARE WORD OF DESCRIPTOR
DEF_ELE TYP_A_BUFADR,A,1,LINBUF ; ADDRESS OF BUFFER
DEF_CTXEND

.MACRO TYPCTXBLK
.ENDM TYPCTXBLK
.ENDM TYPCTXBLK
```

```
:
: GENERATE A CONTEXT BLOCK FOR THE IMAGE "XXX"
```

```
:
: .MACRO XXXCTXBLK ALLO=NO
.IIF IDN <ALLO><YES>, .GLOBL XXX$CTXBLK
DEF_CTXBLK XXX$CTXBLK,ALLOCATE=ALLO ; DEFINE START OF CONTEXT BLOCK
DEF_ELE XXX_L_STATUS,L,1,1 ; SAVE AREA FOR FINAL STATUS
DEF_TAG XXX_Q_BUFDES ; BUFFER DESCRIPTOR
DEF_ELE XXX_W_BUFcnt,W ; BUFFER COUNT WORD
DEF_ELE XXX_W_BUFSIZ,W,1,LINBUFSIZ ; SIZE KEPT IN SPARE WORD OF DESCRIPTOR
DEF_ELE XXX_A_BUFADR,A,1,LINBUF ; ADDRESS OF BUFFER
DEF_CTXEND

.MACRO XXXCTXBLK
.ENDM XXXCTXBLK
.ENDM XXXCTXBLK
```

|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|------------------|-----------------|------------------|------------------|--|--|--|--|--|----------------|--|--|--|--|--|--|
| BCPRSDEF<br>REQ  |                 | CNVCLTAB<br>LIS  |                  |  |  |  |  |  | INFO<br>LIS    |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  | TYPE<br>REQ     | CHRSUB<br>LIS    | CNVCLINUM<br>LIS |  |  |  |  |  |                |  |  |  |  |  |  |
| SHODEVDEF<br>REQ |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  | CLIMAC<br>MAR   |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 | CNVCLIFRM<br>LIS |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  | DIGRAMS<br>LIS |  |  |  |  |  |  |
|                  |                 | CALCMAX<br>LIS   |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  | CLITLMAC<br>MAR |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
| SHOWDEF<br>REQ   |                 |                  | CREATE<br>LIS    |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |
|                  |                 |                  |                  |  |  |  |  |  |                |  |  |  |  |  |  |

BCCMPRS  
LIS

CUTTIME  
LIS