IDENTIFICATION

Product Id: ZZ-CYDRIVER-6.0

Product Title: CYDRIVER - VAX CI Diagnostic Responder Class Driver

Department: Base Systems Diagnostic Engineering

Product Date: 1-July-1984

Table of Contents

1.0 Loading

To load the VAX diagnostic device driver for the CI780 or CI750 on a VAX/VMS operating system, use the following command sequence:

```
$ MCR SYSGEN
SYSGEN> LOAD SYS$MAINTENANCE:CYDRIVER
SYSGEN> CONNECT CY/NOADAP
SYSGEN> EXIT
```

The state of the device can be checked by doing:

```
$SHO DEV CY
```

If the CI is not online (SHO DEV PA), then the driver will not be loaded.

Object Module Synopsis

Module Name	Ident	Bytes	File	Creation Date	Creator
CYTABLES	V6-000	424	CYTABLES.OBJ;18	7-JUL-1984 15:32	VAX-11 Macro V03-01
CYINIT	V6-000	232	[SHULL.EVXCI.CYDRIVER]CYINIT.OBJ;16	7-JUL-1984 15:30	VAX-11 Macro V03-01
CYINPUT	V6-000	525	[SHULL.EVXCI.CYDRIVER]CYINPUT.OBJ;19	7-JUL-1984 15:30	VAX-11 Macro V03-01
CYCMD	V6-000	1450	[SHULL.EVXCI.CYDRIVER]CYCMD.OBJ;22	7-JUL-1984 15:29	VAX-11 Macro V03-01
CYMAINT	V6-000	423	[SHULL.EVXCI.CYDRIVER]CYMAINT.OBJ;16	7-JUL-1984 15:31	VAX-11 Macro V03-01
CYMISC	V6-000	1704	[SHULL.EVXCI.CYDRIVER]CYMISC.OBJ;16	7-JUL-1984 15:31	VAX-11 Macro V03-01
SYS	V03-047	0	SYS$SYSROOT:[SYSEXE]SYS.STB;1	3-JUN-1984 17:36	VAX-11 Linker V3A-18
GLOBALS	V03-003	0	SYS$SYSROOT:[SYSEXE]SYSDEF.STB;1	3-JUN-1984 15:02	VAX-11 Macro V03-01
SYS$IODEF	V03-001	0	SYS$SYSROOT:[SYSLIB]STARLET.OLB;1	3-JUN-1984 11:29	VAX-11 Macro V03-01

Image Section Synopsis

Cluster	Type	Pages	Base Addr	Disk VBN	PFC	Protection and Paging	Global Sec. Name	Match	Majorid	Minorid
DEFAULT_CLUSTER	2	10	00000000	2	0	READ WRITE COPY ON REF				

Key for special characters above:

R - Relocatable
P - Protected

Program Section Synopsis

Psect Name	Module Name	Base	End	Length		Align	Attributes
$$$105_PROLOGUE		00000000	00000056	00000057 (87.)		BYTE 0	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	CYTABLES	00000000	00000056	00000057 (87.)		BYTE 0	
$$$115_DRIVER		00000058	0000129F	00001248 (4680.)		LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	CYTABLES	00000058	000001A8	00000151 (337.)		LONG 2	
	CYINIT	000001AC	00000293	000000E8 (232.)		LONG 2	
	CYINPUT	00000294	000004A0	0000020D (525.)		LONG 2	
	CYCMD	000004A4	00000A4D	000005AA (1450.)		LONG 2	
	CYMAINT	00000A50	00000BF6	000001A7 (423.)		LONG 2	
	CYMISC	00000BF8	0000129F	000006A8 (1704.)		LONG 2	

Symbols By Name

Symbol	Value	Symbol	Value	Symbol	Value
ACP$ACCESS	80008A8E	BOO$GL_FPEMUL	8002585C	BUG$_BADQHDR	00000478
ACP$ACCESSNET	80008A93	BOO$GL_IRPCNT	800257FC	BUG$_BADRSEIPL	00000098
ACP$DEACCESS	80008ABF	BOO$GL_LRPCNT	8002580C	BUG$_BADRVNWCB	000004C8
ACP$GB_BASEPRIO	80003E86	BOO$GL_LRPMIN	80025804	BUG$_BADSBMBLK	000000A0
ACP$GB_DATACHK	80003E85	BOO$GL_LRPSIZE	80025800	BUG$_BADSWPVBN	000000A8
ACP$GB_MAXREAD	80003E82	BOO$GL_LRPSPLIT	80025808	BUG$_BADWCBPT	000000B0
ACP$GB_SWAPFLGS	80003E87	BOO$GL_MTACCESSLOA	8002584C	BUG$_BDPPURGERR	00000488
ACP$GB_WINDOW	80003E83	BOO$GL_NPAGEDYN	8002_7F4	BUG$_BRDMSGLOST	000004A0
ACP$GB_WRITBACK	80003E84	BOO$GL_PRTDRV	E0025834	BUG$_CEBREFNEG	00000498
ACP$GW_DIRCACHE	80003E74	BOO$GL_SCSLOA	8002583C	BUG$_CHMONIS	000000B8
ACP$GW_EXTCACHE	80003E7A	BOO$GL_SPLITADR	800257F8	BUG$_CHMVEC	000004F0
ACP$GW_EXTLIMIT	80003E7C	BOO$GL_SPTFREH	80003F7C	BUG$_CIPORT	000005E8
ACP$GW_FIDCACHE	80003E78	BOO$GL_SPTFREL	80003F78	BUG$_CJF	00000678
ACP$GW_HDRCACHE	80003E72	BOO$GL_SRPCNT	80025814	BUG$_CJFFAILOVR	00000680
ACP$GW_MAPCACHE	80003E70	BOO$GL_SRPSPLIT	80025810	BUG$_CLUEXIT	00000668
ACP$GW_QUOCACHE	80003E7E	BOO$GL_SYSLOA	800257E4	BUG$_CNXMGRERR	00000638
ACP$GW_SYSACC	80003E80	BOO$GL_TRMDRV	800257E8	BUG$_CONSOLRX50	000006B8
ACP$GW_WORKSET	80003E76	BOO$GL_UCODE	80025838	BUG$_CONTRACT	000000C0
ACP$MODIFY	80008B15	BOO$GL_VAXEMUL	80025858	BUG$_CPUCEASED	00000660
ACP$MOUNT	80008B1D	BOO$GQ_FILCACHE	80025818	BUG$_DBLERR	000000C8
ACP$READBLK	80008B3C	BOO$GQ_INILOA	800257EC	BUG$_DECPTREF	000000D0
ACP$V_READCHK	00000000	BOO$GT_TOPSYS	80025824	BUG$_DELCONPFN	000000D8
ACP$V_SWAPGRP	00000001	BUG$A_PAGEC	80025A00	BUG$_DELGBLSEC	000000E0
ACP$V_SWAPMAG	00000003	BUG$A_PAGEDEND	80028750	BUG$_DELGBLWCB	000000E8
ACP$V_SWAPPRV	00000002	BUG$FATAL	800089E2	BUG$_DELWSLEX	000000F8
ACP$V_SWAPSYS	00000000	BUG$T_MESSAGES	80026004	BUG$_DEQSUBLCKS	00000508
ACP$V_WRITECHK	00000001	BUG$_ACCVIOKSTK	000006A0	BUG$_DIRENTRY	00000100
ACP$WRITEBLK	80008B78	BUG$_ACCVIOMCHK	00000698	BUG$_DISKCLASS	000005E0
ALLOC_BUFFER	00000CC2-R	BUG$_ACPMBFAIL	00000008	BUG$_DOUBLDALOC	00000108
ALLOC_CB	00000BF8-R	BUG$_ACPRECURS	000004B8	BUG$_DOUBLDEALO	00000110
ALLOC_CDRP	00000C35-R	BUG$_ACPUNSTAK	000004C0	BUG$_ERRCACHFUL	000004D0
ALLOC_IRP_CDRP	00000C5B-R	BUG$_ACPVAFAIL	00000010	BUG$_ERRHALT	00000118
ALLOC_MNT_CDRP	00000CA1-R	BUG$_ALCPHD	00000018	BUG$_EXHFUL	00000120
ALLOC_RB	00000C23-R	BUG$_ALCSMBCLR	00000020	BUG$_EXPANDPHD	00000128
AT$_NULL	00000005	BUG$_APTREFHIGH	00000028	BUG$_EXTCACHIV	000004D8
BDL$GL_DISK_LOG	80024400	BUG$_APTWRTERR	00000030	BUG$_FATALEXCPT	00000130
BDL$L_SYSDLOG	00000034	BUG$_ASYNCWRTER	00000038	BUG$_FILCNTNONZ	000004F8
BDL$S_CRELNM_ITMLST	00000074	BUG$_BADALORQSZ	00000040	BUG$_FREEPAGREF	00000138
BOO$A_BOOPARAM	800257E0	BUG$_BADBOOTCB	000000F0	BUG$_FREWSLX	00000140
BOO$C_BOOPARSZ	00000084	BUG$_BADBUFADR	00000048	BUG$_GBLPAGSZRO	00000148
BOO$C_SYSPARSZ	000003E8	BUG$_BADBUFTYP	00000050	BUG$_GBLWSLXERR	00000150
BOO$GB_NODENAME	80025850	BUG$_BADDALRQSZ	00000058	BUG$_CPGNUI.PGFL	00000158
BOO$GB_SYSTEMID	8002582E	BUG$_BADFID	00000060	BUG$_HALT	00000160
BOO$GL_BOOTCB	80025820	BUG$_BADFORKIPL	00000068	BUG$_HDRNOTMAP	00000168
BOO$GL_CHKPRTLOA	80025848	BUG$_BADLCKWSLE	00000070	BUG$_ICONCLUDAT	000006D0
BOO$GL_CLSLOA	80025840	BUG$_BADMCKCOD	00000078	BUG$_ICONPFNDAT	00000170
BOO$GL_DEVNAME	80025860	BUG$_BADPAGFILA	00000080	BUG$_ICPAGELOC	00000178
BOO$GL_DSKDRV	800257EC	BUG$_BADPAGFILD	00000088	BUG$_IFREPAGCNT	00000180
BOO$GL_ERRPATLOA	80025844	BUG$_BADPAGTYPE	00000090	BUG$_ILLEVTNUM	00000188

Symbol	Value	Symbol	Value	Symbol	Value
BUG$_ILLVEC	00000190	BUG$_NETNOBUF	00000240	BUG$_RDSNONRES	00000390
BUG$_INCONSTATE	00000198	BUG$_NETNOSTATE	00000248	BUG$_REFCNTNEG	00000398
BUG$_INCPTREF	000001A0	BUG$_NETRCVPKT	00000250	BUG$_RESEXH	000006B0
BUG$_INSNFREPAG	000001A8	BUG$_NETSYSSRV	00000258	BUG$_RMSBUG	000003A0
BUG$_INSSWPFIL	000001B0	BUG$_NETTRANCNT	00000260	BUG$_RSBREFNEG	00000518
BUG$_INSWAPERR	000001B8	BUG$_NOACPCHAN	00000268	BUG$_RSBREFNZRO	00000520
BUG$_INVCHAN	000001C0	BUG$_NOACPMAIL	00000270	BUG$_RUF	00000620
BUG$_INVCJFIOD	000005F8	BUG$_NOAQBACP	00000278	BUG$_SBIAERROR	00000650
BUG$_INVEXCEPTN	000001C8	BUG$_NOBUFPCKT	00000280	BUG$_SCANDEADPT	000003A8
BUG$_INVLOCKID	00000648	BUG$_NOBVPVCB	00000288	BUG$_SCBRDERR	00000528
BUG$_INVPTEFMT	000001D0	BUG$_NOHDJMT	00000600	BUG$_SECREFNEG	000003B0
BUG$_INVRSPID	000005C0	BUG$_NOMULTBK	00000290	BUG$_SHRCNTNEG	000003B8
BUG$_INVTQEFMT	000001D8	BUG$_NONEXSTACP	00000298	BUG$_SSRVEXCEPT	000003C0
BUG$_IVBAKADIO	000001E0	BUG$_NORCP	00000618	BUG$_STATENTSVD	00000530
BUG$_IVBYTEALGN	00000690	BUG$_NORCVBUF	000002A0	BUG$_STRNOTWCB	000003C8
BUG$_IVGBLTYP	000001E8	BUG$_NOSHMGSD	00000490	BUG$_SWAPWSLE	000003D0
BUG$_IVLISTK	000001F0	BUG$_NOTDDBDDB	000002A8	BUG$_SYSADJWSL	000003D8
BUG$_IVSSRVRQST	000001F8	BUG$_NOTFCBFCB	000002B0	BUG$_SYSTRMERR	000003E0
BUG$_IVWSETLIST	00000200	BUG$_NOTFCBWCB	000002B8	BUG$_TAPECLASS	00000628
BUG$_JNLACP	00000610	BUG$_NOTFCPWCB	000002C0	BUG$_TIPCUFLOW	000003E8
BUG$_JNLDRV	00000608	BUG$_NOTIRPAQB	000002C8	BUG$_UBMAPEXCED	000003F0
BUG$_KRNLSTAKNV	00000208	BUG$_NOTLKB	00000588	BUG$_UDAPORT	000005D8
BUG$_KRPEMPTY	000006C0	BUG$_NOTMTLMTL	000002D0	BUG$_UNABLCREVA	000003F8
BUG$_LKBGRANTED	00000580	BUG$_NOTPCB	000002D8	BUG$_UNEXUBAINT	00000400
BUG$_LKBREFNEG	00000510	BUG$_NOTRULUCB	000005F0	BUG$_UNKNPRQ	00000480
BUG$_LOCKMGRERR	00000630	BUG$_NOTRVTVCB	000002E0	BUG$_UNKRSTRT	00000408
BUG$_MACHINECHK	00000210	BUG$_NOTUCBIRP	000002E8	BUG$_UNSUPRTCPU	00000670
BUG$_MAKEWSLE	00000218	BUG$_NOTUCBRVT	000002F0	BUG$_UNXINTEXC	00000410
BUG$_MAPCNTZER	000004E0	BUG$_NOTUCBUCB	000002F8	BUG$_UNXSIGNAL	00000418
BUG$_MBACBHUNG	000004A8	BUG$_NOTUCBWCB	000004E8	BUG$_VBNMAPFAIL	00000420
BUG$_MFYNULPGFL	00000228	BUG$_NOTVCBUCB	00000300	BUG$_WACKQEMPTY	00000428
BUG$_MODRELNBAK	00000220	BUG$_NOTVVPVCB	00000308	BUG$_WCBFCBMNG	000005C8
BUG$_MPASYNCWRT	00000548	BUG$_NOTWCBIRP	00000310	BUG$_WCSCORR	00000658
BUG$_MPBADMCK	00000538	BUG$_NOTWCBWCB	000005D0	BUG$_WRTINVBUF	00000430
BUG$_MPCHMONIS	00000588	BUG$_NOUSRWCS	00000318	BUG$_WRTINVHDR	00000438
BUG$_MPCHMVEC	00000590	BUG$_OPERATOR	00000470	BUG$_WRTPGSBAK	00000440
BUG$_MPDBLERR	00000560	BUG$_OPERCRASH	000006D8	BUG$_WSLENOVAL	00000448
BUG$_MPERRHALT	00000580	BUG$_OUTOFSYNC	00000688	BUG$_WSLPAGCNT	00000450
BUG$_MPHALT	00000568	BUG$_OUTSWPERR	00000320	BUG$_WSLVANVAL	00000458
BUG$_MPILLVEC	00000570	BUG$_PAGEREDERR	00000328	BUG$_WSLXVANMAT	00000460
BUG$_MPIVLISTK	00000558	BUG$_PAGEWRTERR	00000330	BUG$_WSSIZEERR	00000500
BUG$_MPKNLSTKNV	000005A0	BUG$_PAGNTRNVAL	00000338	BUG$_XQPERR	00000640
BUG$_MPMCHECK	00000540	BUG$_PFNLISTCNT	00000340	BUG$_ZEROPAGE	00000468
BUG$_MPNOUSRWCS	00000578	BUG$_PFNREFNZRO	00000348	BUILD_BUFFER	000...143-R
BUG$_MPSCBRDERR	00000598	BUG$_PGFGBLBAD	00000350	BUILD_DG	000010CF-R
BUG$_MPUNEXPINT	000005A8	BUG$_PGFIPLHI	00000358	BUILD_MSG	000010CF-R
BUG$_MPUNKRSTRT	00000550	BUG$_PGFLOCBAD	00000360	CHECK_CONN_CTLR	00000E1D-R
BUG$_MPWALCIRP	00000230	BUG$_PROCGONE	00000368	CHECK_CONN_RSP	00000DE8-R
BUG$_MSCPCLASS	000006C8	BUG$_PTFLENVIOL	00000370	CHECK_CTP_REQ	00000E69-R
BUG$_MSCPSERV	000006A8	BUG$_PTRCNT	00000378	CIA$GL_MUTEX	80002CA8
BUG$_MTXCNTNONZ	00000238	BUG$_PURGWSSCN	00000380	CIA$GQ_INTRUDER	80002CAC
BUG$_NEGSHBREF	000004B0	BUG$_QUEUEMPTY	00000388	CJF$ASSJNL	7FFEE518

Symbol	Value	Symbol	Value	Symbol	Value
CJF$DEASJNL	7FFEE538	COM$POST	8000DEF7	CTL$GL_CMUSER	7FFEFE2C
CJF$DEL JNL	7FFEE548	COM$SETATTNAST	8000DF4B	CTL$GL_CREPRC_FLAGS	7FFEFFC0
CLEAN_CDRP	00000D79-R	COM$SETCTRLAST	8000E0AB	CTL$GL_CTLBASVA	7FFEFE7C
CLEAN_CONN	00000CE4-R	CON$ABORT	800034E4	CTL$GL_DCLPRSOWN	7FFE2200
CLEAR_BUFFER	00000CD8-R	CON$DISCONNECT	800034EC	CTL$GL_F11BXQP	7FFEFF90
CLS$AL_LOAVEC	800036B8	CON$DS_SET	800034C6	CTL$GL_FINALSTS	7FFEFEB8
CLU$AL_LOAVEC	800036B8	CON$INITIAL	80003502	CTL$GL_FIXUPLNK	7FFE2E08
CLU$GB_QDISK	80003EB0	CON$INITLINE	80003508	CTL$GL_GETMSG	7FFEFE68
CLU$GB_VAXCLUSTER	80003EA8	CON$INTDISI	80001354	CTL$GL_IAFEXE	7FFE2E10
CLU$GL_ALLOCLS	80003EC4	CON$INTDISO	80001378	CTL$GL_IAFLAST	7FFE2E04
CLU$GL_CLUB	800036AC	CON$INTINP	8000350E	CTL$GL_IAFLINK	7FFE2E00
CLU$GL_CLUSVEC	800036B0	CON$INTOUT	80003514	CTL$GL_IAFLNKPTR	7FFEFF8C
CLU$GL_LOA_ADDR	800036B8	CON$NULL	800034F6	CTL$GL_IAFPERM	7FFE2E10
CLU$GW_LCKDIRWT	80003EC8	CON$OWNCTY	80003526	CTL$GL_IBIOCNT	7FFEFEE8
CLU$GW_MAXINDEX	800036B4	CON$RELEASECTY	8000352C	CTL$GL_ICPUTIM	7FFEFED0
CLU$GW_QDSKINTERVAL	80003EC2	CON$RESUME	800034EA	CTL$GL_IDIOCNT	7FFEFEE4
CLU$GW_QDSKVOTES	80003EC0	CON$SENDCONSCMD	8000351A	CTL$GL_IFAULTIO	7FFEFED8
CLU$GW_QUORUM	80003EAA	CON$SET_LINE	800034C0	CTL$GL_IFAULTS	7FFEFED4
CLU$GW_RECNXINT	80003EAE	CON$SET_MODEM	800034F0	CTL$GL_IMGHDRBF	7FFEFE80
CLU$GW_VOTES	80003EAC	CON$STARTIO	800034BA	CTL$GL_IMGLSTPTR	7FFEFE84
CNX$ALLOC_CDRP	8000370A	CON$STOP	800034D8	CTL$GL_IPAGEFL	7FFEFEE0
CNX$ALLOC_CDRP_ONLY	80003710	CON$STOP2	800034DE	CTL$GL_IVOLUMES	7FFEFEEC
CNX$ALLOC_WARMCDRP	80003716	CON$XOFF	800034D2	CTL$GL_IWSPEAK	7FFEFEDC
CNX$ALLOC_WARMCDRP_CSB	8000371C	CON$XON	800034CC	CTL$GL_KNOWNFIL	7FFEFF64
CNX$BLOCK_READ	80003764	CONNECT_DATA	00000169-R	CTL$GL_KRP	7FFE6600
CNX$BLOCK_READ_IRP	8000376A	CONN_LIST	000000A1-R	CTL$GL_KRPBL	7FFEFFBC
CNX$BLOCK_WRITE	80003770	CONTROLLER_NAME	00000159-R	CTL$GL_KRPFL	7FFEFFB8
CNX$BLOCK_WRITE_IRP	80003776	CTL$AG_CLIDATA	7FFE325C	CTL$GL_KSPINI	7FFE7E00
CNX$BLOCK_XFER	80003752	CTL$AG_CLIMAGE	7FFE3008	CTL$GL_KSTKBAS	7FFE7800
CNX$BLOCK_XFER_IRP	80003758	CTL$AG_CLITABLE	7FFE3010	CTL$GL_KSTKBASEXP	7FFE7000
CNX$CHANGE_QUORUM	80003788	CTL$AG_CMEDATA	7FFE1E00	CTL$GL_LNMDIRECT	7FFEFE08
CNX$DEALL_MSG_BUF_CSB	80003722	CTL$AL_CLICALBK	7FFE3000	CTL$GL_LNMHASH	7FFEFE04
CNX$DEALL_WARMCDRP_CSB	80003728	CTL$AL_CMCNTX	7FFE1E00	CTL$GL_P1MERGE	7FFE2E0C
CNX$DISK_CHANGE	8000379A	CTL$AL_FINALEXC	7FFEFF28	CTL$GL_PCB	7FFEFF58
CNX$INIT_CDRP	8000372E	CTL$AL_IPASTVEC	7FFEFF68	CTL$GL_PHD	7FFEFE88
CNX$PARTNER_FINISH	8000377C	CTL$AL_STACK	7FFEFE10	CTL$GL_POWERAST	7FFEFF20
CNX$PARTNER_INIT_CSB	8000375E	CTL$AL_STACKLIM	7FFEFE6C	CTL$GL_PPMSG	7FFEFF48
CNX$PARTNER_RESPOND	80003782	CTL$AQ_EXCVEC	7FFEFE34	CTL$GL_PRCALLCNT	7FFEFF9C
CNX$POWER_FAIL	80003794	CTL$A_COMMON	7FFE1600	CTL$GL_RDIPTR	7FFEFFA0
CNX$SEND_ANY_MSG	80003734	CTL$A_DISPVEC	7FFE6000	CTL$GL_RMSBASE	7FFEFF44
CNX$SEND_MSG	8000373A	CTL$C_CLIDATASZ	000015A4	CTL$GL_RUF	7FFEFF5C
CNX$SEND_MSG_CSB	80003740	CTL$C_KRP_COUNT	00000004	CTL$GL_SITESPEC	7FFEFF60
CNX$SEND_MSG_RESP	80003746	CTL$C_KRP_SIZE	00000200	CTL$GL_THEXEC	7FFEFE54
CNX$SEND_MSG_RSPID	8000374C	CTL$GB_DEFLANG	7FFEFF51	CTL$GL_THSUPR	7FFEFE58
CNX$SHUTDOWN	8000378E	CTL$GB_MSGMASK	7FFEFF50	CTL$GL_UAF_FLAGS	7FFE3018
COM$DELATTNAST	8000DE76	CTL$GB_PWRMODE	7FFEFF24	CTL$GL_USRCHME	7FFEFF1C
COM$DELATTNASTP	8000DE7E	CTL$GB_SSFILTER	7FFEFF25	CTL$GL_USRCHMK	7FFEFF18
COM$DELCTRLAST	8000DFB8	CTL$GL_CCBBASE	7FFEFF38	CTL$GL_USRUNDWN	7FFEFF54
COM$DELCTRLASTP	8000DFC0	CTL$GL_CLINTOWN	7FFE2204	CTL$GL_VECTORS	7FFEFE00
COM$DRVDEALMEM	8000DF05	CTL$GL_CMCNTX	7FFEFF88	CTL$GL_VIRTPEAK	7FFEFEC0
COM$FLUSHATTNS	8000DEC5	CTL$GL_CMHANDLR	7FFEFE30	CTL$GL_VOLUMES	7FFEFEC4
COM$FLUSHCTRLS	8000E05F	CTL$GL_CMSUPR	7FFEFE28	CTL$GL_WSPEAK	7FFEFEBC

Symbol	Value	Symbol	Value	Symbol	Value
CTL$GQ_ALLOCREG	7FFEFE8C	DYN$C_BOOTCB	00000006	DYN$C_JNL_JMT	00000009
CTL$GQ_COMMON	7FFEFE60	DYN$C_BRDCST	0000001A	DYN$C_JNL_MSG	00000012
CTL$GQ_DBGAREA	7FFEFF3C	DYN$C_BUFIO	00000013	DYN$C_JNL_MSGDATA	00000014
CTL$GQ_HELPFLAGS	7FFEFFA8	DYN$C_CDB	00000033	DYN$C_JNL_NDL	00000008
CTL$GQ_ISTART	7FFEFEC8	DYN$C_CDRP	00000039	DYN$C_JNL_RC	00000011
CTL$GQ_LOGIN	7FFEFEB0	DYN$C_CD_BBRPG	00000002	DYN$C_JNL_RCPC	0000000C
CTL$GQ_MOUNTLST	7FFEFE94	DYN$C_CD_CDDB	00000001	DYN$C_JNL_RM	0000000A
CTL$GQ_POALLOC	7FFEFF94	DYN$C_CEB	00000004	DYN$C_JNL_RRP	0000000B
CTL$GQ_PROCPRIV	7FFEFF10	DYN$C_CHIP	00000048	DYN$C_JNL_RUL	0000000D
CTL$GQ_TERMCHAR	7FFEFFB0	DYN$C_CI	00000061	DYN$C_JNL_SFT	00000007
CTL$GT_CLINAME	7FFE301C	DYN$C_CIA	00000045	DYN$C_JNL_VCL	0000000E
CTL$GT_SPAWNCLI	7FFE313C	DYN$C_CIDG	0000003B	DYN$C_JNL_VLE	0000000F
CTL$GT_SPAWNTABLE	7FFE315C	DYN$C_CIMSG	0000003C	DYN$C_JPB	0000001F
CTL$GT_TABLENAME	7FFE303C	DYN$C_CI_BDT	00000001	DYN$C_KFD	00000043
CTL$GW_CHINDX	7FFEFE02	DYN$C_CI_FQDT	00000002	DYN$C_KFE	00000018
CTL$GW_NMIOCH	7FFEFE00	DYN$C_CLASSDRV	00000064	DYN$C_KFPB	00000044
CTL$GW_PPMSGCHN	7FFEFF52	DYN$C_CLU	00000065	DYN$C_KFRH	00000026
CTL$T_ACCOUNT	7FFEFEA8	DYN$C_CLU_BTX	00000004	DYN$C_LC_CHREML	00000006
CTL$T_NODEADDR	7FFEFEF0	DYN$C_CLU_CLUB	00000003	DYN$C_LC_CLS	00000005
CTL$T_NODENAME	7FFEFEF7	DYN$C_CLU_CLUDCB	00000005	DYN$C_LC_FPEMUL	00000007
CTL$T_REMOTEID	7FFEFEFE	DYN$C_CLU_CLUOPT	00000006	DYN$C_LC_MP	00000003
CTL$T_USERNAME	7FFEFE9C	DYN$C_CLU_CLUVEC	00000002	DYN$C_LC_MSCP	00000008
CTP$REVISION	00000000	DYN$C_CLU_CSB	00000001	DYN$C_LC_SCS	00000004
CTP$VERSION	00000003	DYN$C_CLU_LCKDIR	00000007	DYN$C_LC_SYSL	00000009
CY$CONNECT_ERR	0000032D-R	DYN$C_CONF	00000007	DYN$C_LKB	00000035
CY$CONNECT_REQ	00000294-R	DYN$C_CRB	00000005	DYN$C_LKID	00000037
CY$DDT	00000058-R	DYN$C_CST	00000008	DYN$C_LNM	00000040
CY$DG_INPUT	000003BF-R	DYN$C_CXB	0000001B	DYN$C_LOADCODE	00000062
CY$DISCONNECT	0000032D-R	DYN$C_DCCB	00000027	DYN$C_LOG	0000000B
CY$END	000012A0-R	DYN$C_DDB	00000006	DYN$C_LPD	00000034
CY$INIT	000001AD-R	DYN$C_DPT	0000001E	DYN$C_MBX	0000002B
CY$L_SAVED_CDT	000001A1-R	DYN$C_ERP	0000003A	DYN$C_MPWMAP	00000004
CY$L_SAVED_CRB	00000190-R	DYN$C_EXTGSD	00000028	DYN$C_MTL	00000019
CY$L_SAVED_UCB	00000199-R	DYN$C_FCB	00000007	DYN$C_MVL	00000016
CY$L_SPARE_CDRP	000001A5-R	DYN$C_FRK	00000008	DYN$C_NDB	0000001C
CY$MNT_BUFMAP	00000AEF-R	DYN$C_GSD	00000015	DYN$C_NET	00000017
CY$MNT_BUFUNM	00000861-R	DYN$C_IDB	00000009	DYN$C_NON_PAGED	00000001
CY$MNT_MOVBUF	00000893-R	DYN$C_INIT	00000063	DYN$C_ORB	00000049
CY$MNT_STATE	00000A73-R	DYN$C_IRP	0000000A	DYN$C_PAGED	00000002
CY$MONITOR	0000126D-R	DYN$C_IRPE	0000002C	DYN$C_PBH	00000020
CY$MSG_INPUT	000003BF-R	DYN$C_JIB	0000002F	DYN$C_PCB	0000000C
CY$NOACT	00000A50-R	DYN$C_JNL	00000067	DYN$C_PCBVEC	00000001
DEALLOC_MSG_DG	00000DD2-R	DYN$C_JNLWCB	00000024	DYN$C_PDB	00000021
DEALLOC_RB	00000D4F-R	DYN$C_JNL_ABL	00000001	DYN$C_PFB	00000047
DEV$M_AVL	00040000	DYN$C_JNL_ACBM	00000004	DYN$C_PFL	00000023
DEV$M_IDV	04000000	DYN$C_JNL_ADL	00000002	DYN$C_PGD	00000066
DEV$M_ODV	08000000	DYN$C_JNL_BCB	00000003	DYN$C_PGD_F11BC	00000001
DT$_RUJNL	00000001	DYN$C_JNL_BUF	00000005	DYN$C_PHVEC	00000002
DYN$C_ACB	00000002	DYN$C_JNL_BXSTS	00000013	DYN$C_PIB	00000022
DYN$C_ACL	0000003F	DYN$C_JNL_CWQ	00000010	DYN$C_PMB	00000046
DYN$C_ADP	00000001	DYN$C_JNL_DB	00000006	DYN$C_PQB	0000000D
DYN$C_AQB	00000003	DYN$C_JNL_DIOREAD	00000015	DYN$C_PRCMAP	00000005

Symbol	Value	Symbol	Value	Symbol	Value
DYN$C_PTR	00000025	ERL$LOGMESSAGE	8000CA76	ERL$VEC264	800045C4
DYN$C_RBM	00000031	ERL$LOGSTATUS	8000C9FB	ERL$VEC268	800045C8
DYN$C_RIGHTSLIST	00000042	ERL$LOG_DMSCP	8000CAE0	ERL$VEC272	800045CC
DYN$C_RSB	00000036	ERL$LOG_TMSCP	8000CAD5	ERL$VEC276	800045D0
DYN$C_RSHT	00000038	ERL$RELEASEMB	8000CC12	ERL$VEC28	80004674
DYN$C_RVT	0000000E	ERL$UNEXP	80004674	ERL$VEC280	800045D4
DYN$C_SCS	00000060	ERL$VEC0	80004674	ERL$VEC284	800045D8
DYN$C_SCS_CDL	00000001	ERL$VEC100	80004674	ERL$VEC288	800045DC
DYN$C_SCS_CDT	00000002	ERL$VEC104	80004674	ERL$VEC292	800045E0
DYN$C_SCS_DIR	00000003	ERL$VEC108	80004674	ERL$VEC296	800045E4
DYN$C_SCS_HQB	0000000B	ERL$VEC112	80004674	ERL$VEC300	800045E8
DYN$C_SCS_PB	00000004	ERL$VEC116	80004674	ERL$VEC304	800045EC
DYN$C_SCS_PDT	00000005	ERL$VEC12	80004674	ERL$VEC308	800045F0
DYN$C_SCS_RDT	00000006	ERL$VEC120	80004674	ERL$VEC312	800045F4
DYN$C_SCS_SB	00000007	ERL$VEC124	80004674	ERL$VEC316	800045F8
DYN$C_SCS_SPNB	00000009	ERL$VEC128	80004674	ERL$VEC32	80004674
DYN$C_SCS_SPPB	00000008	ERL$VEC132	80004674	ERL$VEC320	800045BC
DYN$C_SCS_UQB	0000000A	ERL$VEC136	80004674	ERL$VEC324	800045C0
DYN$C_SHB	0000002A	ERL$VEC140	80004674	ERL$VEC328	800045C4
DYN$C_SHMCEB	0000002E	ERL$VEC144	80004674	ERL$VEC332	800045C8
DYN$C_SHMGSD	00000029	ERL$VEC148	80004674	ERL$VEC336	800045CC
DYN$C_SHRBUFIO	00000080	ERL$VEC152	80004674	ERL$VEC340	800045D0
DYN$C_SLAVCEB	0000002D	ERL$VEC156	80004674	ERL$VEC344	800045D4
DYN$C_SPECIAL	00000080	ERL$VEC16	80004674	ERL$VEC348	800045D8
DYN$C_SSB	0000001D	ERL$VEC160	80004674	ERL$VEC352	800045DC
DYN$C_SUBTYPE	00000060	ERL$VEC164	80004674	ERL$VEC356	800045E0
DYN$C_SWPMAP	00000003	ERL$VEC168	80004674	ERL$VEC36	80004674
DYN$C_TQE	0000000F	ERL$VEC172	80004674	ERL$VEC360	800045E4
DYN$C_TWP	00000030	ERL$VEC176	80004674	ERL$VEC364	800045E8
DYN$C_TYPAHD	00000014	ERL$VEC180	80004674	ERL$VEC368	800045EC
DYN$C_UCB	00000010	ERL$VEC184	80004674	ERL$VEC372	800045F0
DYN$C_UNUSED_2	00000041	ERL$VEC188	80004674	ERL$VEC376	800045F4
DYN$C_VCA	00000032	ERL$VEC192	80004674	ERL$VEC380	800045F8
DYN$C_VCB	00000011	ERL$VEC196	80004674	ERL$VEC384	800045BC
DYN$C_WCB	00000012	ERL$VEC20	80004674	ERL$VEC388	800045C0
DYN$C_WQE	0000003E	ERL$VEC200	80004674	ERL$VEC392	800045C4
DYN$C_XWB	0000003D	ERL$VEC204	80004674	ERL$VEC396	800045C8
ECC$REENABLE	80003466	ERL$VEC208	80004674	ERL$VEC4	80004674
ENTER_CONN_LIST	00000F53-R	ERL$VEC212	80004674	ERL$VEC40	80004674
ERL$ALLOCEMB	8000CB3E	ERL$VEC216	80004674	ERL$VEC400	800045CC
ERL$AL_BUFADDR	80003088	ERL$VEC220	80004674	ERL$VEC404	800045D0
ERL$COLDSTART	8000CB1C	ERL$VEC224	80004674	ERL$VEC408	800045D4
ERL$DEVICEATTN	8000C983	ERL$VEC228	80004674	ERL$VEC412	800045D8
ERL$DEVICERR	8000C8ED	ERL$VEC232	80004674	ERL$VEC416	800045DC
ERL$DEVICTMO	8000C8F1	ERL$VEC236	80004674	ERL$VEC420	800045E0
ERL$GB_BUFFLAG	800030C1	ERL$VEC24	80004674	ERL$VEC424	800045E4
ERL$GB_BUFIND	800030C0	ERL$VEC240	80004674	ERL$VEC428	800045E8
ERL$GB_BUFPTR	800030C2	ERL$VEC244	80004674	ERL$VEC432	800045EC
ERL$GB_BUFTIM	800030C3	ERL$VEC248	80004674	ERL$VEC436	800045F0
ERL$GETFULLNAME	8000CBDE	ERL$VEC252	80004674	ERL$VEC44	80004674
ERL$GL_ERLPID	800030C4	ERL$VEC256	800045BC	ERL$VEC440	800045F4
ERL$GL_SEQUENCE	800030C8	ERL$VEC260	800045C0	ERL$VEC444	800045F8

Symbol	Value	Symbol	Value	Symbol	Value
ERL$VEC448	800045BC	EXE$ALLOCATE	8000A8CE	EXE$CEBREFLCK	8000AF99
ERL$VEC452	800045C0	EXE$ALLOCBUF	8000A775	EXE$CHECKACL	80012000
ERL$VEC456	800045C4	EXE$ALLOCCEB	8000A714	EXE$CHECKACMODE	800121EE
ERL$VEC460	800045C8	EXE$ALLOCIRP	8000A726	EXE$CHECKCLASS	80012227
ERL$VEC464	800045CC	EXE$ALLOCJIB	8000A71D	EXE$CHECKPROT	80012165
ERL$VEC468	800045D0	EXE$ALLOCPCB	8000A72E	EXE$CHECKPROT_16	80012139
ERL$VEC472	800045D4	EXE$ALLOCTQE	8000A737	EXE$CHECK_BYPASS	8001229E
ERL$VEC476	800045D8	EXE$ALLOC_CSD	800037A0	EXE$CHKCREACCES	80010172
ERL$VEC48	80004674	EXE$ALONONPAGED	8000A801	EXE$CHKDELACCES	8001016D
ERL$VEC480	800045DC	EXE$ALONPAGVAR	8000A845	EXE$CHKEXEACCES	8001017C
ERL$VEC484	800045E0	EXE$ALONPAGWAIT	8000A7B0	EXE$CHKIMAGNAME	8001B9BA
ERL$VEC488	800045E4	EXE$ALONPAGWAITS	8000A7A3	EXE$CHKLOGACCES	8001016D
ERL$VEC492	800045E8	EXE$ALOP1IMAG	80010640	EXE$CHKPHYACCES	80010172
ERL$VEC496	800045EC	EXE$ALOP1PROC	80010622	EXE$CHKPRO	80011BAD
ERL$VEC500	800045F0	EXE$ALOPAGED	8000A894	EXE$CHKPRO_INT	80011DB5
ERL$VEC504	800045F4	EXE$ALOPAGWAIT	8000A7C3	EXE$CHKRDACCES	80010177
ERL$VEC508	800045F8	EXE$ALOPHYCNTG	8000AC34	EXE$CHKWAIT2	8000AFB8
ERL$VEC52	80004674	EXE$ALOSHARED	8000AA00	EXE$CHKWRTACCES	80010181
ERL$VEC56	80004674	EXE$ALTQUEPKT	80009B45	EXE$CLEANUP_ORB	800101CE
ERL$VEC60	80004674	EXE$AL_LOAVEC	80003410	EXE$CLI_UTILSRV	80020BF6
ERL$VEC64	80004674	EXE$AL_STACKS	80003F1C	EXE$CLOSE_MSG	8001848B
ERL$VEC68	80004674	EXE$AL_TQENOREPT	80002B20	EXE$CLOSE_RDB	800175BA
ERL$VEC72	80004674	EXE$ARITH	800046AC	EXE$CLREF	0000B114
ERL$VEC76	80004674	EXE$ASCEFC	80010C02	EXE$CMEXEC	80009303
ERL$VEC8	80004674	EXE$ASCTIM	8001321F	EXE$CMKRNL	80009313
ERL$VEC80	80004674	EXE$ASSIGN	80011111	EXE$CMODEXEC	8000FA58
ERL$VEC84	80004674	EXE$ASTDEL	80009D63	EXE$CMODEXECX	8000FA30
ERL$VEC88	80004674	EXE$ASTFLT	800046B7	EXE$CMODKRNL	8000FC08
ERL$VEC 92	80004674	EXE$ASTRET	80009D66	EXE$CMODKRNLX	8000FBE0
ERL$VEC96	80004674	EXE$A_BOOPARAM	800257E0	EXE$CMODSUPR	800046D4
ERL$VEC_RETURN	8000467C	EXE$A_SYSPARAM	80003C00	EXE$CMODUSER	800046EC
ERL$WAKE	8000CC3C	EXE$BINTIM	800132C0	EXE$CNTREG	8001250E
ERL$WARMSTART	8000CB21	EXE$BLDPKTGSR	80009A92	EXE$COMPAT	8000470C
EVT$_AST	00000000	EXE$BLDPKTGSW	80009A9A	EXE$CONTSIGNAL	80017E1A
EVT$_COLPGA	00000000	EXE$BLDPKTSWPR	80009AA2	EXE$CRELNM	8001A6F1
EVT$_EVENT	00000001	EXE$BLDPKTSWPW	80009AAA	EXE$CRELNT	800146EC
EVT$_FPGA	00000004	EXE$BOOTCB_CHK	80004403	EXE$CRELOG	800144D5
EVT$_PFCOM	00000007	EXE$BRDCST	80011ADC	EXE$CREMBX	80014761
EVT$_RESUME	00000006	EXE$BREAK	800046CC	EXE$CREPRC	800128F5
EVT$_SETPRI	00000008	EXE$BRKTHRU	80011485	EXE$CRETVA	8001244D
EVT$_SWPOUT	00000009	EXE$BUFFRQUOTA	8000A631	EXE$CRE_JGTABLE	80020E9D
EVT$_WAKE	00000005	EXE$BUFQUOPRC	8000A63D	EXE$CRMPSC	800131AA
EXE$$ASCTOID	800176D0	EXE$BUG_CHECK	8000427E	EXE$CSP_BRDCST	800037AC
EXE$$FINISH_RDB	80017795	EXE$BUILDPKTR	80009ABA	EXE$CSP_BRKTHRU	800037BE
EXE$$IDTOASC	800177F5	EXE$BUILDPKTW	80009AB2	EXE$CSP_CALL	800037B2
EXE$$IMGACT	8001C7D4	EXE$CANCEL	800091A4	EXE$CSP_COMMAND	800037B8
EXE$ABORTIO	8C009B1E	EXE$CANCELN	80009195	EXE$C_ACCGRNMSK	0000000F
EXE$AB_HEXTAB	800045A9	EXE$CANEXH	80013649	EXE$C_CMSTKSZ	00000014
EXE$ACVIOLAT	80004684	EXE$CANTIM	800092CA	EXE$C_SYSEFN	0000001F
EXE$ADJSTK	80010B6D	EXE$CANWAK	800092E8	EXE$C_SYSPARSZ	000003E4
EXE$ADJWSL	80010A65	EXE$CARRIAGE	8000960E	EXE$DACEFC	8001UBDB
EXE$ALLOC	80013BE7	EXE$CATCH_ALL	80020C0D	EXE$DALLOC	80013C8D

Symbol	Value	Symbol	Value	Symbol	Value
EXE$DASSGN	8000AFB9	EXE$FAOL	8001EB70	EXE$GL_GSDGRPBL	80002A8C
EXE$DCLAST	80011422	EXE$FINDACL	800120E7	EXE$GL_GSDGRPFL	80002A88
EXE$DCLCMH	80013606	EXE$FINISHIO	80009B2D	EXE$GL_GSDMTX	80002B4C
EXE$DCLEXH	80013678	EXE$FINISHIOC	80009B2B	EXE$GL_GSDSYSBL	80002A94
EXE$DEALLOCATE	80C0A9AA	EXE$FORCEJNL	800037E6	EXE$GL_GSDSYSFL	80002A90
EXE$DEALLOC_CSD	800037A6	EXE$FORCEJNLW	800037EC	EXE$GL_INTSTK	80003F20
EXE$DEANONPAGED	8000A8F9	EXE$FORCEX	80009B6F	EXE$GL_INTSTKLM	80002C00
EXE$DEANONPGDSIZ	8000A905	EXE$FORK	80009F48	EXE$GL_KFIMTX	80002B88
EXE$DEAP1	800106EA	EXE$FORKDSPTH	80009F9C	EXE$GL_KNOWNFIL	80002B84
EXE$DEAPAGED	8000A95D	EXE$FORK_WAIT	80008A7B	EXE$GL_KNOWN_FILES	80002B64
EXE$DEAPGDSIZ	8000A961	EXE$FRKIPL10DSP	80009F7C	EXE$GL_LOCKRTRY	80003D20
EXE$DEASHARED	8000AA1D	EXE$FRKIPL11DSP	80009F88	EXE$GL_MCHKERRS	80002BDC
EXE$DEASJNL	800037E0	EXE$FRKIPL6DSP	80009F64	EXE$GL_MEMERRS	80002BE0
EXE$DELLNM	800146F6	EXE$FRKIPL8DSP	80009F94	EXE$GL_MP	80002BF8
EXE$DELLOG	80014557	EXE$FRKIPL9DSP	80009F70	EXE$GL_MSGFLAGS	80003D9C
EXE$DELMBX	80014C99	EXE$GB_CPUDATA	80003F88	EXE$GL_NONPAGED	80002BB0
EXE$DELPRC	8000B0C2	EXE$GB_CPUTYPE	80003F98	EXE$GL_NUMNEXUS	800022A0
EXE$DELTVA	8001259F	EXE$GETCHN	800142F9	EXE$GL_PAGED	80002BC0
EXE$DEQ	8001415B	EXE$GETDEV	80014303	EXE$GL_PFAILTIM	80002AE8
EXE$DERLMB	80009334	EXE$GETDVI	8001430D	EXE$GL_PFATIM	80002AEC
EXE$DGBLSC	800130D8	EXE$GETJPI	800166D8	EXE$GL_PGDYNMTX	80002B48
EXE$DLCEFC	80010DF0	EXE$GETLKI	800166DD	EXE$GL_PQBBL	80002A64
EXE$DUMPCPUREG	80003472	EXE$GETMSG	80C1F147	EXE$GL_PQBFL	80002A60
EXE$DVI_FREEBLOCKS	8001B05D	EXE$GETPTI	80014312	EXE$GL_PWRDONE	80002214
EXE$ENQ	80014058	EXE$GETQUI	80014F98	EXE$GL_PWRINTVL	80002218
EXE$EPID_TO_IPID	8000B386	EXE$GETSYI	800166E5	EXE$GL_RPB	80003F74
EXE$EPID_TO_PCB	8000B35D	EXE$GETTIM	800166EA	EXE$GL_RTBITMAP	80002BD0
EXE$ERAPAT	8000B5EE	EXE$GL_ABSTIM	80002AD0	EXE$GL_RTIMESPT	80003D28
EXE$ERAPAT_RTN	800162EC	EXE$GL_ACLMTX	80002B5C	EXE$GL_RUFBASE	80003800
EXE$ERAPAT_VEC	80003804	EXE$GL_ACMFLAGS	80002C3C	EXE$GL_SAVEDUMP	800029E0
EXE$EXCEPTABLE	80016717	EXE$GL_ARCHFLAG	80003F84	EXE$GL_SCB	80003F80
EXE$EXCEPTION	800047CF	EXE$GL_BLAKHOLE	80002BE8	EXE$GL_SHBLIST	80002BCC
EXE$EXCMSG	80018160	EXE$GL_BOOTCB	800029DC	EXE$GL_SHMGSMTX	80002B50
EXE$EXCPTN	8000FBDC	EXE$GL_BUGCHECK	80000EC8	EXE$GL_SHMMBMTX	80002B54
EXE$EXCPTNE	8000FA0D	EXE$GL_CEBMTX	80002B44	EXE$GL_SITESPEC	80002BFC
EXE$EXIT	80016658	EXE$GL_CJFBASE	800037DC	EXE$GL_SPLITADR	80002BBC
EXE$EXIT_IMAGE	80020C00	EXE$GL_CLITABL	80003D2C	EXE$GL_STATE_FLAGS	80002944
EXE$EXPANDSTK	80017FEA	EXE$GL_CONFREG	80002298	EXE$GL_STATIC_FLAGS	80003D98
EXE$EXPREG	800122D4	EXE$GL_CONFREGL	80002294	EXE$GL_SVAPTE	80002C90
EXE$EXTENDPOOL	8000AA5C	EXE$GL_CPUNODSP	80002290	EXE$GL_SYSID_LOCK	80002B60
EXE$EXTRA1	80003440	EXE$GL_DEFFLAGS	80003D90	EXE$GL_SYSMSG	800029A8
EXE$EXTRA10	80003592	EXE$GL_DYNAMIC_FLAGS	80003D94	EXE$GL_SYSUCB	800029B4
EXE$EXTRA2	80003448	EXE$GL_ENQMTX	80002B58	EXE$GL_SYSUIC	80003D00
EXE$EXTRA3	80003450	EXE$GL_ERASEPB	800029E4	EXE$GL_SYSWCBBL	80002AAC
EXE$EXTRA4	80003458	EXE$GL_ERASEPPT	800029E8	EXE$GL_SYSWCBFL	80002AA8
EXE$EXTRA5	80003460	EXE$GL_FKWAITBL	80002994	EXE$GL_TENUSEC	80002BF0
EXE$EXTRA6	8000357A	EXE$GL_FKWAITFL	80002990	EXE$GL_TODR	80003C08
EXE$EXTRA7	80003580	EXE$GL_FLAGS	80002940	EXE$GL_TQFL	80002AF0
EXE$EXTRA8	80003586	EXE$GL_FPEXCVEC	80002954	EXE$GL_UBDELAY	80002BF4
EXE$EXTRA9	8000358C	EXE$GL_GPT	80002B00	EXE$GL_USRCHME	8000295C
EXE$FAILURE	8000FD1A	EXE$GL_GSDDELBL	80002A9C	EXE$GL_USRCHMK	80002958
EXE$FAO	8001EB63	EXE$GL_GSDDELFL	80002A98	EXE$GL_USRUNDWN	80002BAC

Symbol	Value	Symbol	Value	Symbol	Value
EXE$GL_VAXEXCVEC	80002950	EXE$JBCRSP	80015441	EXE$OUTHEX	80025F6C
EXE$GL_WCBDELBL	80002AA4	EXE$KERSTKNV	8000474C	EXE$OUTZSTRING	80025FF5
EXE$GL_WCBDELFL	80002AA0	EXE$LCKPAG	80014363	EXE$PAGRDERR	80004792
EXE$GQ_BOOTCB_D	800029D8	EXE$LCLDSKVALID	800093E9	EXE$POWERAST	80014DBF
EXE$GQ_BOOTTIME	80002AE0	EXE$LDB_SYNCH	8000FDAE	EXE$POWERFAIL	80005064
EXE$GQ_ERLMBX	80002948	EXE$LINK_VEC	80024200	EXE$PRCDELMSG	8001E3A6
EXE$GQ_GBLHOOK1	80002240	EXE$LKWSET	8001435C	EXE$PRCPURMSG	8001E39D
EXE$GQ_GBLHOOK2	80002248	EXE$LNM_SYNTAX_DAT	8000F000	EXE$PROBER	8000A684
EXE$GQ_GBLHOOK3	80002250	EXE$LOAD_EDISP	800035B7	EXE$PROBER_DSC	80010216
EXE$GQ_GBLHOOK4	80002258	EXE$LOAD_ERROR	8000359C	EXE$PROBEW	8000A6D3
EXE$GQ_GBLHOOK5	80002260	EXE$LOAD_ESPR1	800035B7	EXE$PROBEW_DSC	8001021A
EXE$GQ_GBLHOOK6	80002268	EXE$LOAD_ESPR2	800035BD	EXE$PROCIMGACT	80020B2E
EXE$GQ_GBLHOOK7	80002270	EXE$LOAD_KCJF	8000359E	EXE$PROCSTRT	8000AF3C
EXE$GQ_GBLHOOK8	80002278	EXE$LOAD_KDISP	8000359E	EXE$PURGWS	80014CC8
EXE$GQ_GBLHOOK9	80002280	EXE$LOAD_KRUF	800035A4	EXE$PUTMSG	8001676C
EXE$GQ_GBLHOOKA	80002288	EXE$LOAD_KSPR1	800035AA	EXE$PWRTIMCHK	80004256
EXE$GQ_KFE_LCKNAM	80002B7C	EXE$LOAD_KSPR2	800035B0	EXE$QIO	80009829
EXE$GQ_RIGHTSLIST	80002AB0	EXE$LOAD_NOP	8000359D	EXE$QIOACPPKT	80009B58
EXE$GQ_SYSDISK	8001FDE0	EXE$MAXACMODE	80010207	EXE$QIODRVPKT	80009B41
EXE$GQ_SYSTIME	80002AD8	EXE$MCHECK	80004750	EXE$QIORETURN	80009B77
EXE$GQ_TODCBASE	80003C00	EXE$MCHK	80003410	EXE$QXQPPKT	80009B88
EXE$GRANTID	80017577	EXE$MCHK_BUGCHK	8000485B	EXE$RADRMOD	8000479C
EXE$GT_STARTUP	80003FC4	EXE$MCHK_ERRCNT	80003598	EXE$READ	80009423
EXE$GW_PGFL_FID	80003F9C	EXE$MCHK_PRTCT	80004833	EXE$READCHK	800094E6
EXE$GW_SCANPIX	80002BA6	EXE$MCHK_TEST	8000489D	EXE$READCHKR	800094FA
EXE$HIBER	8000B24F	EXE$MGBLSC	800131B2	EXE$READEF	8000B173
EXE$HWCLKINT	80009FC8	EXE$MNTVERSHDOL	80003568	EXE$READLOCK	8000944F
EXE$IMGDELMSG	8001E29E	EXE$MNTVERSIO	80003562	EXE$READLOCKR	80009462
EXE$IMGDMP_EXEC	80020D4D	EXE$MNTVERSP1	8000356E	EXE$READP_TODR	80003550
EXE$IMGDMP_MERGE	80020D53	EXE$MNTVERSP2	80003574	EXE$READ_TODR	80003532
EXE$IMGFIX	8001DEDF	EXE$MODIFY	8000941D	EXE$REFLECT	80017C76
EXE$IMGPURMSG	8001E295	EXE$MODIFYLOCK	80009455	EXE$REGRESTOR	80003478
EXE$IMGSTA	8001E0CC	EXE$MODIFYLOCKR	80009458	EXE$REGSAVE	8000347E
EXE$INIBOOTADP	8000346C	EXE$MOUNTVER	8000355C	EXE$RESETVEC	80016C06
EXE$INIPROCREG	80003484	EXE$MTACCESS	8000B5F6	EXE$RESTART	80004000
EXE$INIT	8002467C	EXE$MTACCESS_RTN	8001631D	EXE$RESUME	8000B23A
EXE$INIT_DEVICE	800041B6	EXE$MTACCESS_VEC	8000380A	EXE$REVOKID	8001757E
EXE$INIT_TODR	8000353E	EXE$MULTIQUOTA	8000A656	EXE$RMVTIMQ	80008A09
EXE$INI_TIMWAIT	8000354A	EXE$NAMPID	8000B276	EXE$ROPRAND	800047A4
EXE$INSERTIRP	80009BC7	EXE$NETSNDERL	80015882	EXE$RUNDWN	80016987
EXE$INSIOQ	80009BA9	EXE$NULLPROC	80008A89	EXE$SCHDWK	80009656
EXE$INSTIMQ	800089E2	EXE$NUMTIM	800134E4	EXE$SEARCH_RIGHT	80012072
EXE$INT54	80003418	EXE$OVEPARM	800093DD	EXE$SENDMSG	8001581B
EXE$INT58	80003420	EXE$OPCCUS	80004758	EXE$SENSEMODE	800095E6
EXE$INT5C	80003428	EXE$OPCDEC	80004760	EXE$SETAST	80011400
EXE$INT60	80003430	EXE$OPEN_MSG	800182F9	EXE$SETCHAR	800095A9
EXE$IOFORK	80009F44	EXE$OPEN_RDB	800179AC	EXE$SETEF	8000B197
EXE$IORSNWAIT	80009384	EXE$OPRSNDERL	8001587C	EXE$SETEXV	80014D12
EXE$IPAPBKAST	80009E46	EXE$OUTBLANK	80025F86	EXE$SETIME	8001730E
EXE$IPCONTROL	8000DC4C	EXE$OUTCHAR	80025F8B	EXE$SETIME_INT	80017304
EXE$IPID_TO_EPID	8000B3B9	EXE$OUTCRLF	80025FE6	EXE$SETIMR	800096B8
EXE$IPID_TO_PCB	8000B364	EXE$OUTCSTRING	80025FF0	EXE$SETMODE	800095C0

Symbol	Value	Symbol	Value	Symbol	Value
EXE$SETOPR	800158D8	EXE$V_EXPLICITS	00000015	FIL$GQ_CACHE	800029D0
EXE$SETPFM	80016CA2	EXE$V_FATAL_BUG	0000000A	FIL$GT_DDDEV	800029B8
EXE$SETPRA	80014D8B	EXE$V_INIT	00000008	FIL$GT_DDSTRING	7FFE032C
EXE$SETPRI	80009BE5	EXE$V_JOBQUEUES	0000001B	FIL$GT_TOPSYS	800029C6
EXE$SETPRN	8000B3E6	EXE$V_MOUNTMSG	00000000	FIL$INITWCB	8001FD33
EXE$SETPRT	80014E30	EXE$V_MULTACP	0000000B	FIL$MOUNT	8001F810
EXE$SETPRV	80016FF7	EXE$V_NOAUTOCNF	00000001	FIL$OPENFILE	8001F547
EXE$SETRWM	80014D55	EXE$V_NOCLOCK	00000005	FIL$OPENFILE_1	8001F550
EXE$SETSFM	80014D5C	EXE$V_NOCLUSTER	0000000C	FIL$RDCHKFILHDR	8001FAF7
EXE$SETSSF	800170D8	EXE$V_PAGFILDMP	00000019	FIL$RDWRTLBN	8001FD02
EXE$SETSTK	800170F5	EXE$V_PGFLCRIT	00000017	FIL$READVBN	8001FBC2
EXE$SETSWM	80014D6A	EXE$V_PGFLFRAG	00000016	FIL$STATBLK	8001FC55
EXE$SET_RDIPTR	80017BC2	EXE$V_POOLPGING	00000003	FIL$WRITEVBN	8001FBBB
EXE$SHMCEBDEL	8000AF63	EXE$V_REINITQUE	0000001C	FIND_MAINT_BUFFER	000011BD-R
EXE$SHUTDWNADP	800034A2	EXE$V_RESALLOC	00000011	FIND_MAPPED_BUFFER	00001191-R
EXE$SIGTORET	80017FC3	EXE$V_SAVEDUMP	0000001A	FUNCTION_MASK	0001FFFF
EXE$SNDACC	80015533	EXE$V_SBIERR	00000007	IAC$AL_IMGACTBUF	7FFE4800
EXE$SNDERR	80015888	EXE$V_SETTIME	00000009	IAC$AL_VECADDR	7FFE2E58
EXE$SNDEVMSG	800019C8	EXE$V_SHRF11ACP	0000000F	IAC$AL_VECOPCOD	7FFE2E68
EXE$SNDJBC	80014FA0	EXE$V_SIMULATOR	00000004	IAC$AW_VECRESET	7FFE2E6C
EXE$SNDOPR	80015559	EXE$V_SSINHIBIT	00000013	IAC$AW_VECSET	7FFE2E74
EXE$SNDSMB	80015546	EXE$V_SYSPAGING	00000002	IAC$GL_FIRST_ICB	7FFE2E98
EXE$SNGLEQUOTA	8000A653	EXE$V_SYSUAFALT	0000000E	IAC$GL_ICBFL	7FFE2E8C
EXE$SRCHANDLER	80017DD1	EXE$V_SYSWRTABL	00000000	IAC$GL_IMAGCTX	7FFE2E50
EXE$SSFAIL	800047C0	EXE$V_TBCHK	00000018	IAC$GL_IMAGE_LIST	7FFE2E7C
EXE$STARTUPADP	8000349C	EXE$V_WRITESYSPARAMS	00000001	IAC$GL_MAIN_ICB	7FFE2E94
EXE$SUCCESS	8000FD22	EXE$V_XQP_RESIDENT	00000000	IAC$GL_PROCCTX	7FFE2E54
EXE$SUSPND	8000B1C4	EXE$WAITFR	8000B523	IAC$GL_STACK_SIZE	7FFE2E9C
EXE$SWAPINIT	8001E995	EXE$WAKE	8000B265	IAC$GL_WORK_LIST	7FFE2E84
EXE$SWTIMINT	8000A028	EXE$WFLAND	8000B512	IMG$ALLOCATE_ICB	8001D244
EXE$TBIT	800047AC	EXE$WFLOR	8000B519	IMG$DEALLOCATE_ICB	8001D28B
EXE$TEST_CSR	8000348A	EXE$WRITE	8000942C	IMG$DECODE_IHD	8001D454
EXE$TIMEOUT	8000A0FC	EXE$WRITECHK	800094EC	IMG$DO_WORK_LIST	8001D77C
EXE$TRNLNM	800146FB	EXE$WRITECHKR	80009558	IMG$GET_HEADER	8001CCE2
EXE$TRNLOG	80014583	EXE$WRITEJNL	800037F2	IMG$GET_NEXT_ISD	8001D5F1
EXE$T_ID_UPCASE	800175D0	EXE$WRITEJNLW	800C37F8	IMG$IS_IT_MAPPED	8001DF52
EXE$UBCLRINT	80009FD0	EXE$WRITELOCK	80009452	IMG$OPEN_IMAGE	8001CAB8
EXE$ULKPAG	8001434C	EXE$WRITELOCKR	80009469	IMG$PRVSHRIMG	8001E062
EXE$ULWSET	80014345	EXE$WRITEP_TODR	80003556	INI$ALLOC_CRB	800253AE
EXE$UNWIND	80017147	EXE$WRITE_TODR	80003538	INI$ALONONPAGED	800255C7
EXE$UPCASE_DAT	8000F800	EXE$WRTMAILBOX	80001A1A	INI$BRK	8000E95C
EXE$UPDSEC	8001594A	EXE$ZEROPARM	800093E3	INI$CONSOLE	80003544
EXE$VAL_IDNAME	8001025F	EXEC_LIST	00000121-R	INI$MASTERWAKE	800048B8
EXE$V_BUGDUMP	00000010	FIL$CACHE_INIT	8001F6BD	INI$MPMADP	80003496
EXE$V_BUGREBOOT	0000000D	FIL$CACHE_TRUNC	8001F731	INI$RDONLY	8000E965
EXE$V_CJFLOAD	0000001D	FIL$CVTFILNAM	8001F46A	INI$WRITABLE	8000E95E
EXE$V_CJFSYSRUJ	0000001E	FIL$CVT_DTB	800165E4	INIT_CREDIT	00000004
EXE$V_CLASS_PROT	00000000	FIL$CVT_HTB	800165F2	INIT_DG_BUF	00000002
EXE$V_CONCEALED	00000012	FIL$CVT_OTB	800165EB	IO$GL_SCB_INTO	80002BEC
EXE$V_CRDENABL	00000006	FIL$C_DIR_SIZE	00000024	IO$GL_UBA_INTO	80002BE4
EXE$V_DISMOUMSG	00000001	FIL$C_SIZE	00000218	IOC$ALLOSPT	8000D27D
EXE$V_EXPLICITP	00000014	FIL$FINDFILID	8001F895	IOC$ALOUBAMAP	8000CF9B

Symbol	Value	Symbol	Value	Symbol	Value
IOC$ALOUBAMAPN	8000CF94	IOC$GL_SRPBL	80002A10	IOC$REQCOM	8000CD99
IOC$ALOUBAMAPSP	8000D005	IOC$GL_SRPCNT	80002A24	IOC$REQDATAP	8000CE5E
IOC$ALOUBMAPRM	8000D0AB	IOC$GL_SRPFL	80002A0C	IOC$REQDATAPNW	8000CE70
IOC$ALOUBMAPRMN	8000D0A4	IOC$GL_SRPMIN	80002A18	IOC$REQDATAPUDA	8000CE7E
IOC$ALTREQCOM	8000CD6E	IOC$GL_SRPREM	80002A20	IOC$REQMAPREG	8000CF5F
IOC$APPLYECC	8000D7D8	IOC$GL_SRPSIZE	80002A14	IOC$REQMAPUDA	8000CF4A
IOC$BROADCAST	8000D37D	IOC$GL_SRPSPLIT	80002A1C	IOC$REQPCHANH	8000CD37
IOC$BUFPOST	800049E2	IOC$GL_TU_CDDB	80002A84	IOC$REQPCHANL	8000CD40
IOC$CANCELIO	8000CC5D	IOC$GQ_BRDCST	80002A74	IOC$REQSCHANH	8000CD23
IOC$CHKMBXQUOTA	80010574	IOC$GQ_MOUNTLST	80002A6C	IOC$REQSCHANL	8000CD2D
IOC$CHKUCBQUOTA	80010579	IOC$GW_LAMAPREG	80003D26	IOC$RETURN	8000D236
IOC$CLONE_UCB	8000DB00	IOC$GW_MAXBUF	80003D06	IOC$SCAN_IODB	8000D3EE
IOC$COPY_UCB	8000DB26	IOC$GW_MBXBFQUO	80003D08	IOC$SCAN_IODB_2P	8000D433
IOC$CREATE_UCB	80010605	IOC$GW_MBXMXMSG	80003D0A	IOC$SEARCH	800102F6
IOC$CREDIT_UCB	800105C1	IOC$GW_MBXNMMSG	80003D0C	IOC$SEARCHALL	800102F0
IOC$CTRLINIT	8000D48D	IOC$GW_MVTIMEOUT	80003D04	IOC$SEARCHCONT	8000D702
IOC$CVTLOGPHY	8000D84C	IOC$GW_XFMXRATE	80003D24	IOC$SEARCHDEV	800102EA
IOC$CVTLOGPHYU	8000D855	IOC$INITBUFWIND	8000C88E	IOC$SEARCHINT	8000D60E
IOC$CVT_DEVNAM	8000D2A4	IOC$INITDRV	800258A4	IOC$SENSEDISK	8000D96A
IOC$DALLOC_DEV	800104BE	IOC$INITIATE	8000CE31	IOC$SEVER_UCB	8000DBF4
IOC$DEBIT_UCB	800105A1	IOC$IOPOST	800048C4	IOC$TESTUNIT	8000D711
IOC$DELETE_UCB	8000DBE4	IOC$LAST_CHAN	8000CC7B	IOC$THREADCRB	80003652
IOC$DIAGBUFILL	8000CCB1	IOC$LAST_CHAN_AMBX	8000CC74	IOC$TRANDEVNAM	80010338
IOC$DIRPOST1	80004F66	IOC$LINK_UCB	8000DBB4	IOC$UNITINIT	8000D4C7
IOC$DISMOUNT	8000FE8E	IOC$LOADMBAMAP	8000D975	IOC$UNLOCK	8001052A
IOC$FFCHAN	800102A6	IOC$LOADUBAMAP	8000D9D6	IOC$UNLOCK_DEV	800104E6
IOC$FILSPT	8000C8C5	IOC$LOADUBAMAPA	8000D9BF	IOC$UPDATRANSP	8000D92E
IOC$FREE_UCB	8000DC16	IOC$LOADUBAMAPN	8000DA5D	IOC$VERIFYCHAN	8001053D
IOC$GETBYTE	8000C86C	IOC$LOCK_DEV	8001040D	IOC$WAKACP	80004DD8
IOC$GL_ADPLIST	80000ED0	IOC$LUBAUDAMAP	8000DA9C	IOC$WFIKPCH	8000D237
IOC$GL_AQBLIST	80002A68	IOC$MAPVBLK	8000D877	IOC$WFIRLCH	8000D259
IOC$GL_CRBTMOUT	80002A7C	IOC$MNTVER	8000CE28	IPL$_SCS	00000008
IOC$GL_DEVLIST	80000F48	IOC$MOVFRUSER	8000C895	KFI$GL_F11AACP	80002B8C
IOC$GL_DPTLIST	80000ED4	IOC$MOVFRUSER1	8000C8A6	LCK$BREAK_DEADLOCK	8000BA6F
IOC$GL_DU_CDDB	80002A80	IOC$MOVFRUSER2	8000C899	LCK$CANCEL_CVT	8000C4F9
IOC$GL_IRPBL	800029FC	IOC$MOVTOUSER	8000C8AD	LCK$CHECK_RSB	8000C729
IOC$GL_IRPCNT	80002A04	IOC$MOVTOUSER1	8000C8BE	LCK$COMPAT_TBL	8000BB24
IOC$GL_IRPFL	800029F8	IOC$MOVTOUSER2	8000C8B1	LCK$COMP_GGMODE	8000C3CE
IOC$GL_IRPMIN	80002A08	IOC$PARSDEVNAM	8000D4FA	LCK$CVTNOTQED	8000BCB8
IOC$GL_IRPREM	80002A00	IOC$PTETOPFN	8000DA49	LCK$CVT_GRANTED	8000BC8F
IOC$GL_LRPBL	80002A2C	IOC$PURGDATAP	80003490	LCK$CVT_ID_TO_LKB	80003704
IOC$GL_LRPCNT	80002A40	IOC$PUTBYTE	8000C87D	LCK$DEALLOC_RSB	8000C742
IOC$GL_LRPFL	80002A28	IOC$QNXTSEG	80004D2F	LCK$DEQLOCK	8000C596
IOC$GL_LRPMIN	80002A34	IOC$QNXTSEG1	80004D3B	LCK$DLCKEXIT	8000B814
IOC$GL_LRPREM	80002A3C	IOC$REINITDRV	800258AA	LCK$EXTEND_IDTBL	8000C7BB
IOC$GL_LRPSIZE	80002A30	IOC$RELCHAN	8000CCE0	LCK$EXTEND_IDTBLW	8000C7C4
IOC$GL_LRPSPLIT	80002A38	IOC$RELDATAP	8000CEE9	LCK$GB_HTBLSHFT	80002C38
IOC$GL_MUTEX	80002B40	IOC$RELDATAPUDA	8000CEDE	LCK$GB_MAXDEPTH	80002C39
IOC$GL_PFKBINT	80002A5C	IOC$RELMAPREG	8000D170	LCK$GB_REBLD_STATE	80002C3B
IOC$GL_POOLFKB	80002A44	IOC$RELMAPUDA	8000D155	LCK$GB_STALLREQS	80002C3A
IOC$GL_PSBL	800029F4	IOC$RELOC_DDT	8002594C	LCK$GL_DIRVEC	80002C20
IOC$GL_PSFL	800029F0	IOC$RELSCHAN	8000CCD6	LCK$GL_EXTRASTK	80003C2C

Symbol	Value	Symbol	Value	Symbol	Value
LCK$GL_HASHTBL	80002C10	LIB$_ATTCONSTO	0015827C	MIN_SEND_CR	00000001
LCK$GL_HTBLCNT	80002C14	LKI$SEARCH_BLOCKEDBY	8000E853	MMG$ALCPRD	80010956
LCK$GL_HTBLSIZ	80003D38	LKI$SEARCH_BLOCKING	8000E7B2	MMG$ALCSTX	8001092D
LCK$GL_IDTBL	80002C04	LKI$SND_BLKBY	800037D0	MMG$ALC_PGFLVBN	800165A7
LCK$GL_IDTBLMAX	80003D34	LKI$SND_BLKING	800037CA	MMG$ALLOCONTIG	80006A4C
LCK$GL_IDTBLSIZ	80003D30	LKI$SND_LOCKS	800037D6	MMG$ALLOCPAGFIL1	80006D71
LCK$GL_MAXID	80002C0C	LKI$SND_STDREQ	800037C4	MMG$ALLOCPAGFIL2	80006EDA
LCK$GL_NXTID	80002C08	LNM$AL_DIRTBL	800029A4	MMG$ALLOCPFN	800067EC
LCK$GL_PRLMAP	80002C24	LNM$AL_HASHTBL	80002998	MMG$ALLOCSWPAREA	80006D39
LCK$GL_TIMOUTQ	80002C18	LNM$AL_MUTEX	800029B0	MMG$ALOSHMGSD	80015C46
LCK$GL_WAITTIME	80003D3C	LNM$CHECK_PROT	80018A33	MMG$ALOSHMPAG	80015B3A
LCK$GQ_BITMAP_EXP	80002C28	LNM$CONTSEARCH	80018DCC	MMG$AL_BEGDRIVE	80001950
LCK$GQ_BITMAP_EXPLCL	80002C30	LNM$DELBLK	80018A94	MMG$AL_ENDDRIVE	80001E68
LCK$GRANTCVTS	8000C3F1	LNM$DELETE_LNMB	80018AC1	MMG$AL_FIXUPTBL	80025648
LCK$GRANTWTRS	8000C43A	LNM$DELETE_TAB	80018B01	MMG$AL_PGDCODEN	80021E00
LCK$GRANT_LOCK	8000C12A	LNM$FIRSTTAB	80018D86	MMG$AL_SYSPCB	80002800
LCK$GRANT_LOCK_ALT	8000C12F	LNM$GL_HTBLSIZP	80003D8C	MMG$A_ENDVEC	80000A00
LCK$GRANT_REM	8000C17C	LNM$GL_HTBLSIZS	80003D88	MMG$A_SYSPARAM	80003C00
LCK$HASH_SEARCH	8000C0BB	LNM$HASH	80018E41	MMG$A_SYS_END	80028800
LCK$LOCAL_CVT	8000BC33	LNM$INIT_PROT	80018B12	MMG$CALCSWAPSIZE	80007059
LCK$LOCAL_LOCK	8000BEF6	LNM$INSLOGTAB	80018B5A	MMG$CEFTRNLOG	80015E21
LCK$NORET_VALBLK	8000C01A	LNM$LOCKR	80018F75	MMG$CLR_BITMAP	80015ADB
LCK$NOT_QUEUED	8000C02C	LNM$LOCKW	80018F7D	MMG$CREDEL	8001260D
LCK$QUEUECVT	8000C2CD	LNM$PRESEARCH	80018DA2	MMG$CREPAG	800126DA
LCK$QUEUED_EXIT	8000C017	LNM$PROBER	80018F52	MMG$CRETVA	80012445
LCK$QUEUEWAIT	8000C2E7	LNM$SEARCHLOG	80018C7E	MMG$DALCBAKSTORE	80006962
LCK$QUEUE_BLKAST	8000C397	LNM$SEARCH_ONE	80018D0F	MMG$DALCPAGFIL	80006F65
LCK$QUEUE_BLOCKAST	8000C34C	LNM$SETUP	80018EC1	MMG$DALCSTX	80010916
LCK$QUEUE_REM	8000C2EC	LNM$SYSTEM_DIRECTORY	80003DE0	MMG$DALCSTXSCN	80010893
LCK$REGRANTLOCK	8000C142	LNM$TABLE	80018ED1	MMG$DALCSTXSCN1	8001088C
LCK$RET_VALBLK	8000BFF7	LNM$UNLOCK	80018F85	MMG$DALLOCPFN	800069C6
LCK$SEARCHDLCK	8000B720	LNM_SYSTEM_DIR_LNMTH	8000310B	MMG$DEALLOCPAGFIL	80006F6E
LCK$SND_BLKING	800036E0	MA$INITIAL	800034B4	MMG$DECPHDREF	80006788
LCK$SND_CVTREQ	800036BC	MA$RAVAIL	800034A8	MMG$DECPHDREF1	8000678C
LCK$SND_DEQCV	800036D4	MA$REQUEST	800034AE	MMG$DECPTREF	80006721
LCK$SND_DEQGR	800036CE	MAINLINE	000004A4-R	MMG$DECSECREF	8000709C
LCK$SND_DEQWT	800036DA	MAP_QUEUE	00000129-R	MMG$DECSHMREF	80008674
LCK$SND_DLCKFND	800036F8	MAX_BUF_LEN	00000400	MMG$DELCONPFN	80006801
LCK$SND_GRANTED	800036C8	MAX_BUF_MAP	00000002	MMG$DELGBLSEC	80007BF2
LCK$SND_LOCKREQ	800036C2	MAX_LOC_PORTS	00000002	MMG$DELGBLWCB	80013FE7
LCK$SND_REDO_SRCH	800036FE	MAX_NODES	00000010	MMG$DELPAG	80007868
LCK$SND_RMVDIR	800036E6	MB$DDT	80001950	MMG$DELPFNLST	800068FA
LCK$SND_SRCHDLCK	800036F2	MB$DPT	8000183C	MMG$DELSHMGS	80016244
LCK$SND_TIMESTAMP_RQST	800036EC	MB$GL_DDB	800013AC	MMG$DELWSLEPPG	80006611
LCK$SRCH_HSHTBL	8000C103	MB$GL_ORB1	80001564	MMG$DELWSLEX	800065ED
LCK$SRCH_RESDLCK	8000B882	MB$GL_ORB2	80001648	MMG$DGBLSC1	80013DB4
LCK$SYNCCVT_TBL	8000BB2A	MB$GL_UCB1	800014D4	MMG$EXPKSTK	80007FE7
LCK$SYNC_EXIT	8000BFE2	MB$GL_UCB2	800015B8	MMG$EXPREG	800122C6
LIB$GB_OPINFO	00000000	MB$ORB0	800013F0	MMG$EXTRADYNWS	800076D9
LIB$GB_OPINFO1	80023190	MB$UCB0	80001444	MMG$FAST_CREATE	800123A3
LIB$GB_OPINFO2	80023990	MCHK$GL_MASK	80002BD4	MMG$FINDTSTGSD	80015D3D
LIB$INS_DECODE	80022AF6	MCHK$GL_SP	80002BD8	MMG$FINDGSDPFN	800085FE

Symbol	Value	Symbol	Value	Symbol	Value
MMG$FINDGSNOTRN	80016143	MMG$IMGRESET	80010746	MMG$UNLOCK	80006CDE
MMG$FINDSHB	80015D65	MMG$INADRINI	8001255D	MMG$UPDSECAST	80015A5F
MMG$FINDSHD	80008711	MMG$INCPTREF	800066E3	MMG$VALIDATEGSD	80008696
MMG$FREEGSD	80015CD1	MMG$INCSHMREF	80008677	MMG$VFYSECFLG	80013D27
MMG$FREWSLE	80006467	MMG$INIBLDPKT	80006797	MMG$V_NOLASTUPD	00000000
MMG$FREWSLX	800064FD	MMG$ININEWPFN	80006637	MMG$V_NOWAIT	00000001
MMG$FRE_TRYSKIP	8000652A	MMG$INSPFNH	8000698A	MMG$WRITE_GSD	80015F7A
MMG$FRSTRONLY	80004000	MMG$INSPFNT	800069C8	MMG$WRTMFYPAG	80007316
MMG$GB_FREWFLGS	8000219A	MMG$IN_REGION	800124A5	MMG$WRTPGSBAK	80008306
MMG$GETGSNAM	80015D9A	MMG$IOLOCK	80006AAB	MMG$WSLEPFN	8000639A
MMG$GETNXTGSD	8000869A	MMG$IOLOCKPAG	80006BB0	MMG$WSPEAKCHK	80010B42
MMG$GETPTIPAG	80007D02	MMG$LCKULKPAG	80007DCD	MNT$CATCH_RST	00000007
MMG$GL_CTLBASVA	80003F18	MMG$LOCKPGTB	800066DD	MNT$CATCH_STR	00000008
MMG$GL_FPEMUL_BASE	800022A8	MMG$MAKEWSLE	80006679	MNT$MAINT_READ	00000005
MMG$GL_FRESVA	80003F30	MMG$MBXTRNLOG	80015E2A	MNT$MAINT_STATE	00000002
MMG$GL_GBLPAGFIL	800022B8	MMG$MOVPTLOCK	8000772B	MNT$MAINT_WRITE	00000006
MMG$GL_GBLSECFND	800022B4	MMG$MOVPTLOCK1	80007727	MNT$NORMAL_STATE	00000004
MMG$GL_GPTBASE	80003F24	MMG$MPWCHECK	800089BF	MNT$POLLER_OFF	00000000
MMG$GL_GPTE	80003F28	MMG$M_NOLASTUPD	00000001	MNT$POLLER_ON	00000001
MMG$GL_IRPNEXT	80003F58	MMG$M_NOWAIT	00000002	MNT$UNINIT_STATE	00000003
MMG$GL_LRPNEXT	80003F5C	MMG$PAGEFAULT	80005C58	MNT_MAP_QUEUE	00000131-R
MMG$GL_MAXGPTE	80003F2C	MMG$PAGETYPE	80007B70	MONITOR_LIST	00000141-R
MMG$GL_MAXMEM	80003F70	MMG$PGFLTWAIT	80005F9F	MPH$ASTDELCONT	80009CB2
MMG$GL_MAXPFIDX	8000220C	MMG$PTEADRCHK	80007197	MPH$ASTDELHK	80009CAC
MMG$GL_MAXPFN	80003F68	MMG$PTEINDX	800071CB	MPH$BUGCHKHK	80004362
MMG$GL_MAXSYSVA	80003F30	MMG$PTEINDXCHK	800071C5	MPH$INVALIDHK	80006522
MMG$GL_MINPFN	80003F6C	MMG$PTEREF	800071A0	MPH$NEWLVLHK	80009F2C
MMG$GL_NPAGEDYN	80003F50	MMG$PURGWSSCN	800080A9	MPH$QAST	80009E6B
MMG$GL_NPAGNEXT	80003F54	MMG$READ_GSD	80015F83	MPH$QEMPTYCONT	80009CC6
MMG$GL_NULLPFL	800021E4	MMG$REFCNTNEG	80006D31	MPH$RESCHED	8000A5A4
MMG$GL_PAGEDYN	80003F64	MMG$RELPFN	80006902	MPH$SCHED	8000A5CA
MMG$GL_PAGSWPVC	80002208	MMG$REMPFN	8000688E	MPW$AL_PTE	80001EF0
MMG$GL_PFNLOCK	80001EFC	MMG$REMPFNH	80006883	MPW$AW_INITVAL	80003C84
MMG$GL_PGDCOD	80003FE4	MMG$RESRCWAIT	80005F84	MPW$AW_PHVINDEX	80001EF4
MMG$GL_PHYPGCNT	80003C98	MMG$RETADRINI	8001256C	MPW$GB_PRIO	80003C8A
MMG$GL_RMSBASE	800022A4	MMG$RETRANGE	80012588	MPW$GL_BADPAGTOTAL	80001EF8
MMG$GL_SBICONF	8000229C	MMG$RET_BYT_QUOTA	8001A7F4	MPW$GL_THRESH	80003C8C
MMG$GL_SBR	80003F4C	MMG$RLPFNSAVPTE	800068E5	MPW$GL_WAITLIM	80003C90
MMG$GL_SPTBASE	80003F34	MMG$SCNWSLX	80007FAE	MPW$GW_HILIM	80003C86
MMG$GL_SPTLEN	80003F38	MMG$SETPRTPAG	80008188	MPW$GW_LOLIM	80003C88
MMG$GL_SRPNEXT	80003F60	MMG$SET_BITMAP	80015AD2	MPW$GW_MPWPFC	80003C84
MMG$GL_SYSLOA_BASE	800022AC	MMG$SHMTXLK	800161E6	MSG$_TRMBRDCST	00000053
MMG$GL_SYSPHD	80003F3C	MMG$SHMTXULK	80016223	MT$CHECK_ACCESS	80009C84
MMG$GL_SYSPHDLN	80003F40	MMG$SHRCNTNEG	80006D35	NET$WCB	800017C4
MMG$GL_VAXEMUL_BASE	800022B0	MMG$SHRINKWS	80007650	NEXT_ENTRY	000004C8-R
MMG$GSDMTXULK	80013D9B	MMG$SUBSECREF	8000708F	NL$DDT	800J1DF4
MMG$GSDSCN	80013ECD	MMG$SVAPTECHK	8000717E	NL$DPT	80001875
MMG$GSDTRNLOG	80015E33	MMG$SVPCTX	80005F56	NL$GL_DDB	8000169C
MMG$GW_BIGPFN	80003F9A	MMG$SWAPWSLE	80007F72	NL$GL_ORB0	800016E0
MMG$GW_MINPFIDX	80002210	MMG$TRY_ALL	80012353	NL$GL_UCB0	80001734
MMG$IMGACTBUF	7FFE4800	MMG$ULKGBLWSLE	80007DBC	NO$GL_DPT	80000EE0
MMG$IMGHDRBUF	7FFE6400	MMG$UNIQUEGSD	80016183	NSA$ARGLST_IMGNAM	8001658B

Symbol	Value	Symbol	Value	Symbol	Value
NSA$EVENT_AUDIT	800163BA	PIO$GB_DFNBC	7FFE022B	PMS$GL_ENQCVT_IN	800032D8
NSA$GR_ALARMVEC	80002C68	PIO$GB_RMSPROLOG	7FFE022C	PMS$GL_ENQCVT_LOC	800032D4
NSA$GR_JOURNVEC	80002C40	PIO$GL_DIRCACHE	7FFE0230	PMS$GL_ENQCVT_OUT	800032DC
NSA$T_IDT	7FFE2800	PIO$GL_DIRCFRLH	7FFE0238	PMS$GL_ENQNEW_IN	800032CC
OP$DPT	800018AE	PIO$GL_FMLH	7FFE0200	PMS$GL_ENQNEW_LOC	800032C8
OPA$CRB	80001330	PIO$GL_IIOFSPLH	7FFE0208	PMS$GL_ENQNEW_OUT	800032D0
OPA$GL_DDB	80001124	PIO$GL_NXTIRBSEQ	7FFE0240	PMS$GL_ENQNOTQD	800032F0
OPA$ORB0	80001168	PIO$GL_RULOCK	7FFE023C	PMS$GL_ENQWAIT	800032EC
OPA$UCB0	800011BC	PIO$GQ_IIODEFAULT	7FFE0324	PMS$GL_ERASEIO	80000B64
OPA$VECTOR	80001913	PIO$GT_DDSTRING	7FFE032C	PMS$GL_EXTHIT	80000B3C
P1SYSVECTORS	7FFEDE00	PIO$GT_ENDSTR	7FFE0212	PMS$GL_EXTMISS	80000B40
PAT$A_NONPGD_CODE	8000E9B2	PIO$GW_DFPROT	7FFE0222	PMS$GL_FAULTS	80001EB0
PAT$A_NONPGD_CODE_END	8000EC00	PIO$GW_IIOIMPA	7FFE0270	PMS$GL_FCP	80000A00
PAT$A_NONPGD_DATA	80003810	PIO$GW_PIOIMPA	7FFE0248	PMS$GL_FCP2	80000A00
PAT$A_PAGED_CODE	8001851C	PIO$GW_RMSEXTEND	7FFE022D	PMS$GL_FIDHIT	80000B34
PAT$A_PFN_FIXUP	800257D8	PIO$GW_STATUS	7FFE0210	PMS$GL_FIDMISS	80000B38
PAT$GL_EXP_NPG1	8000E9B2	PIO$S_EODSTR	0000001D	PMS$GL_FILHDR_HIT	80000B44
PAT$GL_EXP_NPG2	80003FE4	PMS$ABORT_RQ	8000442A	PMS$GL_FILHDR_MISS	80000B48
PFM$GETBUF	80016F24	PMS$AL_READTBL	80003380	PMS$GL_GVALID	80001EEC
PFM$MON	8000B480	PMS$AL_TRANSFLT	80001EC8	PMS$GL_HIT	80000B20
PFM$PURGE	30016EA2	PMS$AL_WRITETBL	800033AC	PMS$GL_IOPFMPDB	800032AC
PFN$AB_STATE	80003FBC	PMS$END_IO	80004445	PMS$GL_IOPFMSEQ	800032B0
PFN$AB_TYPE	80003FC0	PMS$END_RQ	80004465	PMS$GL_JNLBUFIO	80003334
PFN$AL_BAK	80003FA8	PMS$GB_PROMPT	80003408	PMS$GL_JNLBUFWR	80003344
PFN$AL_HEAD	80001E68	PMS$GL_ACCLCK	80000B78	PMS$GL_JNLCHNLS	8000331C
PFN$AL_HILIMIT	80001E90	PMS$GL_ARRLOCPK	800032B4	PMS$GL_JNLDIRIO	80003330
PFN$AL_LOLIMIT	80001E9C	PMS$GL_ARRTRAPK	800032BC	PMS$GL_JNLFORFL	80003340
PFN$AL_MFYLSTHD	80001E6C	PMS$GL_BLK_IN	800032F8	PMS$GL_JNLFORNL	8000333C
PFN$AL_PTE	80003FA4	PMS$GL_BLK_LOC	800032F4	PMS$GL_JNLJRNLS	80003318
PFN$AW_REFCNT	80003FAC	PMS$GL_BLK_OUT	800032FC	PMS$GL_JNLWRTAI	80003320
PFN$AW_SWPVBN	80003FB8	PMS$GL_BUFIO	80003294	PMS$GL_JNLWRTAT	80003328
PFN$AX_BLINK	80003FB4	PMS$GL_CHME	80003374	PMS$GL_JNLWRTBI	80003324
PFN$AX_FLINK	80003FB0	PMS$GL_CHMK	80003370	PMS$GL_JNLWRTFM	80003348
PFN$AX_SHRCNT	80003FB0	PMS$GL_COMPAT	80002ACC	PMS$GL_JNLWRTRU	8000332C
PFN$AX_WSLX	80003FB4	PMS$GL_DEPLOCPK	800032B8	PMS$GL_JNLWRTSS	80003338
PFN$A_BASE	80003FA4	PMS$GL_DEQ_IN	800032E4	PMS$GL_KFRNEL	80002AB8
PFN$C_LONG_LEN	00000016	PMS$GL_DEQ_LOC	800032E0	PMS$GL_LDPCTX	80003400
PFN$C_WORD_LEN	00000012	PMS$GL_DEQ_OUT	800032E8	PMS$GL_LOGNAM	80003298
PFN$GB_LENGTH	80003F99	PMS$GL_DIRDATA_HIT	80000B4C	PMS$GL_LRGRWP	800033E8
PFN$GL_PHYPGCNT	80001E8C	PMS$GL_DIRDATA_MISS	80000B50	PMS$GL_MBREADS	8000329C
PHV$GL_PIXBAS	80002238	PMS$GL_DIRHIT	80000B24	PMS$GL_MBWRITES	800032A0
PHV$GL_REFCBAS	8000223C	PMS$GL_DIRIO	80003290	PMS$GL_NOSTDTRM	800033F0
PIO$AL_RMSEXH	7FFE0314	PMS$GL_DIRMISS	80000B28	PMS$GL_OPEN	80000B5C
PIO$A_DIRCACHE	7FFE0800	PMS$GL_DIR_IN	80003300	PMS$GL_OPENS	80000B60
PIO$A_TRACE	7FFE0600	PMS$GL_DIR_OUT	80003304	PMS$GL_PAGES	80003378
PIO$GB_DFMBC	7FFE0224	PMS$GL_DLCKFND	80003314	PMS$GL_PASSALL	800033E0
PIO$GB_DFMBFHSH	7FFE022A	PMS$GL_DLCKMSGS_IN	80003308	PMS$GL_PREADIO	80001EB8
PIO$GB_DFMBFIDX	7FFE0229	PMS$GL_DLCKMSGS_OUT	8000330C	PMS$GL_PREADS	80001EB4
PIO$GB_DFMBFREL	7FFE0228	PMS$GL_DLCKSRCH	80003310	PMS$GL_PWRITES	80001EBC
PIO$GB_DFMBFSDK	7FFE0225	PMS$GL_DOSTATS	8000340C	PMS$GL_PWRITIO	80001EC0
PIO$GB_DFMBFSMT	7FFE0226	PMS$GL_DPTSCN	80001EE8	PMS$GL_QUOHIT	80000B2C
PIO$GB_DFMBFSUR	7FFE0227	PMS$GL_DZROFLTS	80001EC4	PMS$GL_QUOMISS	80000B30

Symbol	Value	Symbol	Value	Symbol	Value
PMS$GL_RCVBUFFL	800032C4	PQL$GDWSEXTENT	80003E18	SCH$GL_FREECNT	80001E80
PMS$GL_RDFLTS	80001EB4	PQL$GDWSQUOTA	80003E0C	SCH$GL_FREELIM	80001E9C
PMS$GL_READCNT	800033D8	PQL$GMASTLM	80003E24	SCH$GL_FREEREQ	80001E90
PMS$GL_RUFABORT	8000336C	PQL$GMBIOLM	80003E28	SCH$GL_GROWLIM	80003D18
PMS$GL_RUFACTIV	8000334C	PQL$GMBYTLM	80003E2C	SCH$GL_MAXPIX	80002928
PMS$GL_RUFCHNLS	80003354	PQL$GMCPULM	80003E30	SCH$GL_MFYCNT	80001E84
PMS$GL_RUFJNLS	80003350	PQL$GMDIOLM	80003E34	SCH$GL_MFYLIM	80001E94
PMS$GL_RUFMARK	80003364	PQL$GMENQLM	80003E50	SCH$GL_MFYLIMSV	80001EA8
PMS$GL_RUFMRKRB	80003368	PQL$GMFILLM	80003E38	SCH$GL_MFYLOLIM	80001EA0
PMS$GL_RUFREADS	8000335C	PQL$GMJTQUOTA	80003E58	SCH$GL_MFYLOSV	80001EAC
PMS$GL_RUFWRTS	80003358	PQL$GMPGFLQUOTA	80003E3C	SCH$GL_NULLPCB	80002440
PMS$GL_RUFXTNDS	80003360	PQL$GMPRCLM	80003E40	SCH$GL_PF3VEC	80002920
PMS$GL_RWP	800033E4	PQL$GMTQELM	80003E44	SCH$GL_PFRATH	80003CA0
PMS$GL_RWPNOSTD	800033F4	PQL$GMWSDEFAULT	80003E4C	SCH$GL_PFRATL	80003C9C
PMS$GL_RWPSUM	800033EC	PQL$GMWSEXTENT	80003E54	SCH$GL_PFRATS	80003CA4
PMS$GL_SPLIT	80000B1C	PQL$GMWSQUOTA	80003E48	SCH$GL_PIXLAST	8000292C
PMS$GL_STORAGMAP_HIT	80000B54	PR$_IPL	00000012	SCH$GL_PIXWIDTH	80002930
PMS$GL_STORAGMAP_MISS	80000B58	PR$_TBIA	00000039	SCH$GL_RESMASK	800021D4
PMS$GL_SWITCH	80003404	PRT$C_UREW	0000000D	SCH$GL_SEQVEC	80002924
PMS$GL_SYNCHLCK	80000B70	QUEUE_DELAY	000011E9-R	SCH$GL_SWPPCB	800026E0
PMS$GL_SYNCHWAIT	80000B74	RESPONDER_NAME	00000149-R	SCH$GL_SWPPID	80002740
PMS$GL_TRCNGLOS	800032C0	RETURN_CTP_RESPONSE	00000516-R	SCH$GL_SWPRATE	80003CB8
PMS$GL_TREADS	800032A4	RM$DIRCACHE_BLKAST	8000AF4D	SCH$GL_WSDEC	80003CAC
PMS$GL_TTY_CODE1	800033F8	RM$RESET	800109F0	SCH$GL_WSINC	80003CA8
PMS$GL_TTY_CODE2	800033FC	RM$SET	80010A31	SCH$GQ_CEBHD	800021A4
PMS$GL_TURN	80000B18	RMS$GL_SFDBASE	80002BC8	SCH$GQ_COLPGWQ	80002108
PMS$GL_TWRITES	800032A8	RMS$GW_GBLBUFQUO	80000B80	SCH$GQ_FPGWQ	80002180
PMS$GL_VOLLCK	80000B68	RMS$_STALL	00018001	SCH$GQ_HIBOWQ	8000215C
PMS$GL_VOLWAIT	80000B6C	RMS$_STR	000187BC	SCH$GQ_HIBWQ	80002150
PMS$GL_WRTCNT	800033DC	RND$V_IACLOCK	00000000	SCH$GQ_LEFOWQ	80002144
PMS$GL_XQPCACHEWAIT	80000B7C	SCH$AQ_COMH	80001F08	SCH$GQ_LEFWQ	80002138
PMS$GW_BATCH	8000337C	SCH$AQ_COMUH	80002008	SCH$GQ_MWAIT	80002114
PMS$GW_INTJOBS	8000337E	SCH$AQ_COMOT	8000200C	SCH$GQ_PFWQ	8000212C
PMS$START_IO	80004494	SCH$AQ_COMT	80001F0C	SCH$GQ_SUSP	80002168
PMS$START_RQ	800044B7	SCH$AQ_WQHDR	80002FFC	SCH$GQ_SUSPO	80002174
PQL$AB_FLAG	80003E5F	SCH$ASTDEL	80009C98	SCH$GW_AWSMIN	80003CB0
PQL$AB_SYSPQL	8001E94E	SCH$CHSE	8000A305	SCH$GW_CEBCNT	800021AC
PQL$AL_DEFAULT	80003DE4	SCH$CHSEP	8000A321	SCH$GW_DELPHDCT	800021AE
PQL$AL_MIN	80003E20	SCH$CLREF	8000B165	SCH$GW_DORMANTWAIT	80003CC6
PQL$C_SYSPQLLEN	00000046	SCH$CLREFR	8000B15F	SCH$GW_IOTA	80003CC2
PQL$GDASTLM	80003DE8	SCH$C_MAXPIX	0000003F	SCH$GW_LOCALNODE	80002934
PQL$GDBIOLM	80003DEC	SCH$C_SWPPIX	00000001	SCH$GW_LONGWAIT	80003CC4
PQL$GDBYTLM	80003DF0	SCH$FORCEDEXIT	8000A40A	SCH$GW_PROCCNT	8000219C
PQL$GDCPULM	80003DF4	SCH$GB_PRI	800021D8	SCH$GW_PROCLIM	8000219E
PQL$GDDIOLM	80003DF8	SCH$GB_RESCAN	80002199	SCH$GW_QUAN	80003C82
PQL$GDENQLM	80003E14	SCH$GB_SIP	80002198	SCH$GW_SWPFAIL	80003CC8
PQL$GDFILLM	80003DFC	SCH$GETEFC	8000B124	SCH$GW_SWPFCNT	80002236
PQL$GDJTQUOTA	80003E1C	SCH$GL_AWSTIME	80003CB4	SCH$IOLOCKR	8000ACE1
PQL$GDPGFLQUOTA	80003E00	SCH$GL_BORROWLIM	80003D1C	SCH$IOLOCKW	8000ACC7
PQL$GDPRCLM	80003E04	SCH$GL_COMQS	80002194	SCH$IOUNLOCK	8000AD55
PQL$GDTQELM	80003E08	SCH$GL_COMQS	80002190	SCH$LOCKR	8000ACE8
PQL$GDWSDEFAULT	80003E10	SCH$GL_CURPCB	8000218C	SCH$LOCKW	8000ACCE

Symbol	Value	Symbol	Value	Symbol	Value
SCH$LOCKWNOWAIT	8000ACAF	SCS$GL_MCLEN	800035E8	SGN$C_SYSVECPGS	00000005
SCH$NEWLVL	80009F09	SCS$GL_MSCP	800035F0	SGN$C_SYSWSCNT	00000060
SCH$OSWPSCHED	8000878A	SCS$GL_PDT	800035F4	SGN$GB_KFILSTCT	80003C10
SCH$POSTEF	8000ADCF	SCS$GL_RDT	800035E4	SGN$GB_PGTBPFC	80003C0E
SCH$QAST	80009E6B	SCS$GQ_CONFIG	800035C4	SGN$GB_STARTUP_P1	80003ECC
SCH$QEND	8000A3A7	SCS$GQ_DIRECT	800035CC	SGN$GB_STARTUP_P2	80003ED0
SCH$RAVAIL	8000AD45	SCS$GQ_POLL	800035D4	SGN$GB_STARTUP_P3	80003ED4
SCH$REMOVACB	80009F40	SCS$GW_BDTCNT	80003D40	SGN$GB_STARTUP_P4	80003ED8
SCH$RESCHED	8000A5A4	SCS$GW_CDTCNT	80003D42	SGN$GB_STARTUP_P5	80003EDC
SCH$RSE	8000A27D	SCS$GW_FLOWCUSH	80003D4A	SGN$GB_STARTUP_P6	80003EE0
SCH$RWAIT	8000ACA5	SCS$GW_MAXDG	80003D46	SGN$GB_STARTUP_P7	80003EE4
SCH$SCHED	8000A5CA	SCS$GW_MAXMSG	80003D48	SGN$GB_STARTUP_P8	80003EE8
SCH$SWAPACBS	80009F3D	SCS$GW_NEXTBIT	800035FA	SGN$GB_SYSPFC	80003C0F
SCH$SWPWAKE	8000A532	SCS$GW_PAPOLINT	80003D64	SGN$GB_TAILORED	80003ECA
SCH$UNLOCK	8000AD5C	SCS$GW_PAPOOLIN	80003D66	SGN$GL_BALSETCT	80003C30
SCH$UNWAIT	8000A2CA	SCS$GW_PAPPDDG	80003D60	SGN$GL_EXTRACPU	80003CFC
SCH$V_MPW	00000002	SCS$GW_PASTMOUT	80003D5E	SGN$GL_EXUSRSTK	80003C54
SCH$V_REORD	00000000	SCS$GW_PRCPOLINT	80003D5C	SGN$GL_FREEGOAL	80003D14
SCH$V_SIP	00000000	SCS$GW_RDTCNT	80003D44	SGN$GL_FREELIM	80003D10
SCH$WAIT	8000B594	SCS$LISTEN	80003640	SGN$GL_GBLPAGFIL	80003C18
SCH$WAITK	8000B59E	SCS$LKP_MSGWAIT	8000367C	SGN$GL_IRPCNT	80003C34
SCH$WAITL	8000B5A9	SCS$LKP_RDTCDRP	80003664	SGN$GL_IRPCNTV	80003C38
SCH$WAITM	8000B5AA	SCS$LKP_RDTWAIT	8000366A	SGN$GL_LOADFLAGS	80003DA0
SCH$WAKE	8000A50D	SCS$LOCLOOKUP	80003646	SGN$GL_LRPCNT	80003C58
SCS$ACCEPT	80003604	SCS$NEW_SB	80003688	SGN$GL_LRPCNTV	80003C5C
SCS$ALLOC_CDT	8000360A	SCS$POLL_MBX	8000369A	SGN$GL_LRPMIN	80003C64
SCS$ALLOC_RSPID	80003610	SCS$POLL_MODE	80003694	SGN$GL_LRPSIZE	80003C60
SCS$AL_LOAVEC	80003604	SCS$POLL_PROC	8000368E	SGN$GL_MAXGPGCT	80003C14
SCS$CANCEL_MBX	800036A0	SCS$RECYL_RSPID	80003670	SGN$GL_MAXVPGCT	80003C4C
SCS$CONFIG_PTH	80003616	SCS$REMOVE	8000364C	SGN$GL_MAXWSCNT	80003C3C
SCS$CONFIG_SYS	8000361C	SCS$RESUMEWAITR	80003658	SGN$GL_NPAGEDYN	80003C40
SCS$CONNECT	8000362?	SCS$SHUTDOWN	800036A6	SGN$GL_NPAGEVIR	80003C44
SCS$DEALL_CDT	80003628	SCS$UNSTALLUCB	8000365E	SGN$GL_P1LWCNT	80003F0C
SCS$DEALL_RSPID	8000362E	SEC$V_RESIDENT	0000000D	SGN$GL_PAGEDYN	80003C48
SCS$DIR_LOOKUP	80003682	SEC$W_FLAGS	00000014	SGN$GL_PE1	80003D6C
SCS$DISCONNECT	80003634	SGN$C_BALSETCNT	00000018	SGN$GL_PE2	80003D70
SCS$ENTER	8000363A	SGN$C_DFWSCNT	00000064	SGN$GL_PE3	80003D74
SCS$FIND_RDTE	80003676	SGN$C_DFWSQUOTA	00000078	SGN$GL_PE4	80003D78
SCS$GA_DFLTMSK	800035F8	SGN$C_GBLSECCNT	00000028	SGN$GL_PE5	80003D7C
SCS$GA_EXISTS	800035FC	SGN$C_MAXGPGCNT	00000800	SGN$GL_PE6	80003D80
SCS$GA_LOCALSB	80000EF4	SGN$C_MAXPAGCNT	00004000	SGN$GL_PHDAPCNT	80003F04
SCS$GB_NODENAME	80003D54	SGN$C_MAXPGFL	00001000	SGN$GL_PHDLWCNT	80003F08
SCS$GB_PAMXPORT	80003D63	SGN$C_MAXPSTCNT	00000005	SGN$GL_PHDPAGCT	80003F10
SCS$GB_PANOPOLL	80003D69	SGN$C_MAXVPGCNT	00002000	SGN$GL_PTPAGCNT	80003F14
SCS$GB_PANPOLL	80003D62	SGN$C_MAXWSCNT	00000400	SGN$GL_SPTREQ	80003C50
SCS$GB_PASANITY	80003D68	SGN$C_MINWSCNT	0000000A	SGN$GL_SRPCNT	80003C68
SCS$GB_SYSTEMID	80003D4C	SGN$C_NPAGEDYN	00006800	SGN$GL_SRPCNTV	80003C6C
SCS$GB_SYSTEMIDH	80003D50	SGN$C_NPROCS	00000040	SGN$GL_SRPMIN	80003C74
SCS$GB_UDABURST	80003D86	SGN$C_PAGEDYN	00004000	SGN$GL_SRPSIZE	80003C70
SCS$GL_BDT	800035DC	SGN$C_PFNPATSIZ	00000010	SGN$GL_USER3	80003CF4
SCS$GL_CDT	800035E0	SGN$C_PHYPAGCNT	00001000	SGN$GL_USER4	80003CF8
SCS$GL_MCADR	800035EC	SGN$C_SYSDWSCNT	00000028	SGN$GL_USERD1	80003CEC

Symbol	Value	Symbol	Value	Symbol	Value
SGN$GL_USERD2	80003CF0	SWP$C_KSTACK_EX	00000004	SYS$CRELNM	7FFEE480
SGN$GL_VMS5	80003CDC	SWP$C_KSTACK_EX_WSL	0000005F	SYS$CREMEX	7FFEDEB8
SGN$GL_VMS6	80003CE0	SWP$C_KSTACK_WSL	00000063	SYS$CREPRC	7FFEDEC0
SGN$GL_VMS7	80003CE4	SWP$C_NDYN	00000001	SYS$CRETVA	7FFEDEC8
SGN$GL_VMS8	80003CE8	SWP$C_SHELLPFIL	00000068	SYS$CRMPSC	7FFEDE78
SGN$GL_VMSD1	80003CCC	SWP$C_SHLFPTE	00000081	SYS$C_JOBCTLMB	3141424D
SGN$GL_VMSD2	80003CD0	SWP$C_SHLP1PT	00000002	SYS$C_MBXUCBSIZ	00000090
SGN$GL_VMSD3	80003CD4	SWP$GB_ISWPRI	800021BE	SYS$C_OPRMBX	3241424D
SGN$GL_VMSD4	80003CD8	SWP$GB_PRIO	80003C8B	SYS$DACEFC	7FFEDED0
SGN$GW_CTLIMGLIM	80003C7E	SWP$GB_SHLP1PT	80003EF6	SYS$DALLOC	7FFEDED8
SGN$GW_CTLPAGES	80003C7C	SWP$GL_BALBASE	80003F44	SYS$DASSGN	7FFEDEE0
SGN$GW_DFPFC	80003C0C	SWP$GL_BALSPT	80003F48	SYS$DCLAST	7FFEDEE8
SGN$GW_DFWSCNT	80003E10	SWP$GL_BSLOTSZ	80003EF8	SYS$DELLNM	7FFEE488
SGN$GW_GBLSECNT	80003C12	SWP$GL_HISWPCNT	800021D0	SYS$DELMBX	7FFEDF00
SGN$GW_IMGIOCNT	80003C80	SWP$GL_HOSWPCNT	800021CC	SYS$DELPRC	7FFEDF08
SGN$GW_ISPPGCT	80003C2A	SWP$GL_INPCB	800021B4	SYS$DELTVA	7FFEDF10
SGN$GW_MAXPRCCT	80003C1C	SWP$GL_ISPAGCNT	800021B8	SYS$DEQ	7FFEE3C8
SGN$GW_MAXPSTCT	80003C20	SWP$GL_ISWPCNT	800021C4	SYS$DISCONNECT	7FFEE1D0
SGN$GW_MINWSCNT	80003C22	SWP$GL_ISWPPAGES	800021C0	SYS$ENQ	7FFEE3C0
SGN$GW_PAGFILCT	80003C24	SWP$GL_MAP	80003EFC	SYS$ENQW	7FFEE3D0
SGN$GW_PCHANCNT	80003C78	SWP$GL_OSWPCNT	800021C8	SYS$EXIT	7FFEDF40
SGN$GW_PIOPAGES	80003C7A	SWP$GL_PHDBASVA	80003F00	SYS$EXPREG	7FFEDF48
SGN$GW_PIXSCAN	80003C1E	SWP$GL_SHELIO	8000A62D	SYS$FAO	7FFEDF50
SGN$GW_SWPFILCT	80002210	SWP$GL_SHELL	800021B0	SYS$FAOL	7FFEDF58
SGN$GW_SWPFILES	80003C26	SWP$GL_SHELLBAS	80021000	SYS$FORCEX	7FFEDF60
SGN$GW_SYSDWSCT	80003C28	SWP$GL_SHELLSIZ	80003EEC	SYS$GB_BRK_LIM	80003E9F
SGN$GW_TPWAIT	80003D84	SWP$GL_SLOTCNT	800021A0	SYS$GB_DEFPRI	80003E88
SGN$GW_WSLMXSKP	80003C94	SWP$GL_SWPPGCNT	80003CBC	SYS$GB_DEFQUEPRI	80003E94
SGN$V_LOADCHKPRT	00000001	SWP$GL_SWTIME	800021DC	SYS$GB_DFMBC	80003DD8
SGN$V_LOADERAPAT	00000000	SWP$GL_BAKPTE	80003EF0	SYS$GB_DFMBFHSH	80003DDE
SGN$V_LOADMTACCESS	00000002	SWP$GW_BALCNT	80002234	SYS$GB_DFMBFIDX	80003DDD
SS$_ACCVIO	0000000C	SWP$GW_EMPTPTE	80003EF2	SYS$GB_DFMBFREL	80003DDC
SS$_BADPARAM	00000014	SWP$GW_IBALSETX	800021BC	SYS$GB_DFMBFSDK	80003DD9
SS$_DUPLNAM	00000094	SWP$GW_SWPINC	80003CC0	SYS$GB_DFMBFSMT	80003DDA
SS$_ENDOFFILE	00000870	SWP$GW_WSLPTE	80003EF4	SYS$GB_DFMBFSUR	80003DDB
SS$_EXPRCLM	00002A34	SWP$K_KSTKSZ	000000A0	SYS$GB_DFNBC	80003DE6
SS$_EXQUOTA	0000001C	SWP$SHELINIT	00000A00	SYS$GB_KMASK	8000FE37
SS$_INSSWAPSPACE	00002264	SYS$ADJSTK	7FFEDE20	SYS$GB_KRNLNARG	8000FDE0
SS$_IVLOGNAM	00000154	SYS$ADJWSL	7FFEDE28	SYS$GB_MAXQUEPRI	80003E98
SS$_IVQUOTAL	00000164	SYS$ASCTIM	7FFEDE48	SYS$GB_PWD_TMO	80003E9C
SS$_IVSTSFLG	0000017C	SYS$ASCTOID	7FFEC4B0	SYS$GB_RETRY_LIM	80003E9D
SS$_NOLOGNAM	000001BC	SYS$ASSIGN	7FFEDE50	SYS$GB_RETRY_TMO	80003E9E
SS$_NOPRIV	00000024	SYS$BINTIM	7FFEDE58	SYS$GB_RMSPROLOG	80003DDF
SS$_NORMAL	00000001	SYS$BRKTHRUW	7FFEE4E8	SYS$GET	7FFEE180
SS$_NOSHRIMG	000021BC	SYS$CALL_HANDL	80000010	SYS$GETCHN	7FFEE0C8
SS$_NOSLOT	0000039C	SYS$CANCEL	7FFEDE60	SYS$GETDVIW	7FFEE418
SWI$GL_FQBL	80002964	SYS$CANTIM	7FFEDE68	SYS$GETJPI	7FFEE0D8
SWI$GL_FQFL	80002960	SYS$CANWAK	7FFEDE70	SYS$GETLKIW	7FFEE4A0
SWP$AL_PTRPAG	00000600	SYS$CLOSE	7FFEE1B8	SYS$GETMSG	7FFEE0B0
SWP$A_KSTK	80000EC0	SYS$CMKRNL	7FFEDE90	SYS$GL_BOOTDDB	80000F5C
SWP$C_DBGPTCNT	00000001	SYS$CONNECT	7FFEE1C0	SYS$GL_BOOTORB	80000FA0
SWP$C_KSTACK	00000003	SYS$CREATE	7FFEE1C8	SYS$GL_BOOTUCB	80000FF4

Symbol	Value	Symbol	Value	Symbol	Value
SYS$GL_BRK_TMO	80003EA0	TIMER_INTR	00001213-R		
SYS$GL_FALLBACK	80000EF0	TIMER_QUEUE	00000139-R		
SYS$GL_HID_TIM	80003EA4	TTY$GB_AUTOCHAR	80003DD0		
SYS$GL_JOBCTLMB	800014D4	TTY$GB_DEFSPEED	80003DA9		
SYS$GL_OPRMBX	800015B8	TTY$GB_DIALTYP	80003DA8		
SYS$GL_UIS	80000EE8	TTY$GB_PARITY	80003DAB		
SYS$GQ_PWD	80002CA0	TTY$GB_RSPEED	80003DAA		
SYS$GQ_VERSION	80002B98	TTY$GB_SILOTIME	80003DCA		
SYS$GT_ANNOUNCE	80025864	TTY$GL_DEFCHAR	80003DB0		
SYS$GW_BJOBCNT	80002BA4	TTY$GL_DEFCHAR2	80003DB4		
SYS$GW_BJOBLIM	80003E8C	TTY$GL_DEFPORT	80003DD4		
SYS$GW_FILEPROT	80003DE2	TTY$GL_DELTA	80003DA4		
SYS$GW_GBLBUFQUO	80003DE4	TTY$GL_DPT	80000EDC		
SYS$GW_IJOBCNT	80002BA0	TTY$GL_JOBCTLMB	80000EE4		
SYS$GW_IJOBLIM	80003E8A	TTY$GL_OWNUIC	80003DC4		
SYS$GW_NJOBCNT	80002BA2	TTY$GL_TIMEOUT	80003DCC		
SYS$GW_NJOBLIM	80003E8E	TTY$GW_ALTALARM	80003DBC		
SYS$GW_RJOBLIM	80003E90	TTY$GW_ALTYPAHD	80003DBA		
SYS$GW_RMSEXTEND	80003DE0	TTY$GW_CLASSNAM	80003DC8		
SYS$HIBER	7FFEDF88	TTY$GW_DEFBUF	80003DAC		
SYS$IDTOASC	7FFEE4C0	TTY$GW_DMASIZE	80003DBE		
SYS$IMGACT	7FFEDF90	TTY$GW_PROT	80003DC0		
SYS$IMGFIX	7FFEE400	TTY$GW_TYPAHDSZ	80003DB8		
SYS$K_VERSION	394D3258	UBA$UNEXINT	80003438		
SYS$MGBLSC	7FFEDFA8	UIS$GL_USB	80000EEC		
SYS$NUMTIM	7FFEDFB8	VA$S_VFN	00000015		
SYS$OPEN	7FFEE208	VA$V_VPN	00000009		
SYS$PARSE	7FFEE230	XDELBPT	80022988		
SYS$PUT	7FFEE188	XDELIBRK	800220D8		
SYS$PUTMSG	7FFEE0E0	XDELTBIT	80022A20		
SYS$QIO	7FFEDFC8	XDEL_LOADBASE	80022151		
SYS$QIOW	7FFEDE00	XDS$GL_XESTRING	8002217D		
SYS$RMSRUNDWN	7FFEE268	XDS$GL_XFSTRING	80022181		
SYS$RUNDWN	7FFEDFE0	XDS$GT_LONG_PFN	80021EF8		
SYS$SEARCH	7FFEE248	XDS$GT_WORD_PFN	80021E00		
SYS$SETAST	7FFEDFF8	XQP$BLOCK_ROUTINE	80009093		
SYS$SETEF	7FFEE000	XQP$DEQBLOCKER	800090EF		
SYS$SETEXV	7FFEE008	XQP$FCBSTALE	80009169		
SYS$SETIMR	7FFEE020	XQP$GL_DZRO	80002C98		
SYS$SETPFM	7FFEE0A8	XQP$GL_FILESERVER	80002C9C		
SYS$SETPRI	7FFEE028	XQP$GL_SECTIONS	80002C94		
SYS$SETPRT	7FFEE030	XQP$REL_QUOTA	80009103		
SYS$SETPRV	7FFEE100	XQP$UNLOCK_QUOTA	8000911A		
SYS$SETRWM	7FFEE038				
SYS$SETSFM	7FFEE040				
SYS$SYNCH	7FFEE440				
SYS$TRNLNM	7FFEE490				
SYS$TRNLOG	7FFEE058				
SYS$UNWIND	7FFEE070				
SYS$WAIT	7FFEE1A8				
SYS$WAITFR	7FFEE078				
SYSL$CLRSBIA	80003520				

Symbols By Value

Value	Symbols...			
00000000	ACP$V_READCHK	ACP$V_SWAPSYS	CTP$REVISION	EVT$_AST
	EVT$_COLPGA	EXE$V_CLASS_PROT	EXE$V_MOUNTMSG	EXE$V_SYSWRTABL
	EXE$V_XQP_RESIDENT	LIB$GB_OPINFO	MMG$V_NOLASTUPD	MNT$POLLER_OFF
	RND$V_IACLOCK	SCH$V_REORD	SCH$V_SIP	SGN$V_LOADERAPAT
00000001	ACP$V_SWAPGRP	ACP$V_WRITECHK	DT$_RUJNL	DYN$C_ADP
	DYN$C_CD_CDDB	DYN$C_CI_BDT	DYN$C_CLU_CSB	DYN$C_JNL_ABL
	DYN$C_NON_PAGED	DYN$C_PCBVEC	DYN$C_PGD_F11BC	DYN$C_SCS_CDL
	EVT$_EVENT	EXE$V_DISMOUMSG	EXE$V_NOAUTOCNF	EXE$V_WRITESYSPARAMS
	MIN_SEND_CR	MMG$M_NOLASTUPD	MMG$V_NOWAIT	MNT$POLLER_ON
	SCH$C_SWPPIX	SGN$V_LOADCHKPRT	SS$_NORMAL	SWP$C_DBGPTCNT
	SWP$C_NDYN			
00000002	ACP$V_SWAPPRV	DYN$C_ACB	DYN$C_CD_BBRPG	DYN$C_CI_FQDT
	DYN$C_CLU_CLUVEC	DYN$C_JNL_ADL	DYN$C_PAGED	DYN$C_PHVEC
	DYN$C_SCS_CDT	EXE$V_SYSPAGING	INIT_DG_BUF	MAX_BUF_MAP
	MAX_LOC_PORTS	MMG$M_NOWAIT	MNT$MAINT_STATE	SCH$V_MPW
	SGN$V_LOADMTACCESS	SWP$C_SHLPIPT		
00000003	ACP$V_SWAPMAG	CTP$VERSION	DYN$C_AQB	DYN$C_CLU_CLUB
	DYN$C_JNL_BCB	DYN$C_LC_MP	DYN$C_SCS_DIR	DYN$C_SWPMAP
	EXE$V_POOLPGING	MNT$UNINIT_STATE	SWP$C_KSTACK	
00000004	CTL$C_KRP_COUNT	DYN$C_CEB	DYN$C_CLU_BTX	DYN$C_JNL_ACBM
	DYN$C_LC_SCS	DYN$C_MPWMAP	DYN$C_SCS_PB	EVT$_FPGA
	EXE$V_SIMULATOR	INIT_CREDIT	MNT$NORMAL_STATE	SWP$C_KSTACK_EX
00000005	AT$_NULL	DYN$C_CLU_CLUDCB	DYN$C_CRB	DYN$C_JNL_BUF
	DYN$C_LC_CLS	DYN$C_PRCMAP	DYN$C_SCS_PDT	EVT$_WAKE
	EXE$V_NOCLOCK	MNT$MAINT_READ	SGN$C_MAXPSTCNT	SGN$C_SYSVECPGS
00000006	DYN$C_BOOTCB	DYN$C_CLU_CLUOPT	DYN$C_DDB	DYN$C_JNL_DB
	DYN$C_LC_CHREML	DYN$C_SCS_RDT	EVT$_RESUME	EXE$V_CRDENABL
	MNT$MAINT_WRITE			
00000007	DYN$C_CLU_LCKDIR	DYN$C_CONF	DYN$C_FCB	DYN$C_JNL_SFT
	DYN$C_LC_FPEMUL	DYN$C_SCS_SB	EVT$_PFCOM	EXE$V_SBIERR
	MNT$CATCH_RST			
00000008	BUG$_ACPMBFAIL	DYN$C_CSP	DYN$C_FRK	DYN$C_JNL_NDL
	DYN$C_LC_MSCP	DYN$C_SCS_SPPB	EVT$_SETPRI	EXE$V_INIT
	IPL$_SCS	MNT$CATCH_STR		
00000009	DYN$C_IDB	DYN$C_JNL_JMT	DYN$C_LC_SYSL	DYN$C_SCS_SPNB
	EVT$_SWPOUT	EXE$V_SETTIME	VA$V_VPN	
0000000A	DYN$C_IRP	DYN$C_JNL_RM	DYN$C_SCS_UQB	EXE$V_FATAL_BUG
	SGN$C_MINWSCNT			
0000000B	DYN$C_JNL_RRP	DYN$C_LOG	DYN$C_SCS_HQB	EXE$V_MULTACP
0000000C	DYN$C_JNL_RCPC	DYN$C_PCB	EXE$V_NOCLUSTER	SS$_ACCVIO
0000000D	DYN$C_JNL_RUL	DYN$C_PQB	EXE$V_BUGREBOOT	PRT$C_UREW
	SEC$V_RESIDENT			
0000000E	DYN$C_JNL_VCL	DYN$C_RVT	EXE$V_SYSUAFALT	
0000000F	DYN$C_JNL_VLE	DYN$C_TQE	EXE$C_ALCGRNMSK	EXE$V_SHRF11ACP
00000010	BUG$_ACPVAFAIL	DYN$C_JNL_CWQ	DYN$C_UCB	EXE$V_BUGDUMP
	MAX_NODES	PIO$S_EODSTR	SGN$C_PFNPATSIZ	
00000011	DYN$C_JNL_RC	DYN$C_VCB	EXE$V_RESALLOC	
00000012	DYN$C_JNL_MSG	DYN$C_WCB	EXE$V_CONCEALED	PFN$C_WORD_LEN
	PR$_IPL			

Value	Symbols...			
00000013	DYN$C_BUFIO	DYN$C_JNL_BXSTS	EXE$V_SSINHIBIT	
00000014	DYN$C_JNL_MSGDATA	DYN$C_TYPAHD	EXE$C_CMSTKSZ	EXE$V_EXPLICITP
	SEC$W_FLAGS	SS$_BADPARAM		
00000015	DYN$C_GSD	DYN$C_JNL_DIOREAD	EXE$V_EXPLICITS	VA$S_VPN
00000016	DYN$C_MVL	EXE$V_PGFLFRAG	PFN$C_LONG_LEN	
00000017	DYN$C_NET	EXE$V_PGFLCRIT		
00000018	BUG$_ALCPHD	DYN$C_KFE	EXE$V_TBCHK	SGN$C_BALSETCNT
00000019	DYN$C_MTL	EXE$V_PAGFILDMP		
0000001A	DYN$C_BRDCST	EXE$V_SAVEDUMP		
0000001B	DYN$C_CXB	EXE$V_JOBQUEUES		
0000001C	DYN$C_NDB	EXE$V_REINITQUE	SS$_EXQUOTA	
0000001D	DYN$C_SSB	EXE$V_CJFLOAD		
0000001E	DYN$C_DPT	EXE$V_CJFSYSRUJ		
0000001F	DYN$C_JPB	EXE$C_SYSEFN		
00000020	BUG$_ALCSMBCLR	DYN$C_PBH		
00000021	DYN$C_PDB			
00000022	DYN$C_PIB			
00000023	DYN$C_PFL			
00000024	DYN$C_JNLWCB	FIL$C_DIR_SIZE	SS$_NOPRIV	
00000025	DYN$C_PTR			
00000026	DYN$C_KFRH			
00000027	DYN$C_DCCE			
00000028	BUG$_APTREFHIGH	DYN$C_EXTGSD	SGN$C_GBLSECCN	SGN$C_SYSDWSCNT
00000029	DYN$C_SHMGSD			
0000002A	DYN$C_SHB			
0000002B	DYN$C_MBX			
0000002C	DYN$C_IRPE			
0000002D	DYN$C_SLAVCEB			
0000002E	DYN$C_SHMCEB			
0000002F	DYN$C_JIB			
00000030	BUG$_APTWRTERR	DYN$C_TWP		
00000031	DYN$C_RBM			
00000032	DYN$C_VCA			
00000033	DYN$C_CDB			
00000034	BDL$L_SYSDLOG	DYN$C_LPD		
00000035	DYN$C_LKB			
00000036	DYN$C_RSB			
00000037	DYN$C_LKID			
00000038	BUG$_ASYNCWRTER	DYN$C_RSHT		
00000039	DYN$C_CDRP	PR$_TBIA		
0000003A	DYN$C_ERP			
0000003B	DYN$C_CIDG			
0000003C	DYN$C_CIMSG			
0000003D	DYN$C_XWB			
0000003E	DYN$C_WQE			
0000003F	DYN$C_ACL	SCH$C_MAXPIX		
00000040	BUG$_BADALORQSZ	DYN$C_LNM	SGN$C_NPROCS	
00000041	DYN$C_UNUSED_2			
00000042	DYN$C_RIGHTSLIST			
00000043	DYN$C_KFD			
00000044	DYN$C_KFPB			
00000045	DYN$C_CIA			

Value		Symbols...		
00000046	DYN$C_FMB	PQL$C_SYSPQLLEN		
00000047	DYN$C_PFB			
00000048	BUG$_BADBUFADR	DYN$C_CHIP		
00000049	DYN$C_ORB			
00000050	BUG$_BADBUFTYP			
00000053	MSG$_TRMBRDCST			
00000058	BUG$_BADDALRQSZ	R-CY$DDT		
0000005F	SWP$C_KSTACK_EX_WSL			
00000060	BUG$_BADFID	DYN$C_SCS	DYN$C_SUBTYPE	SGN$C_SYSWSCNT
00000061	DYN$C_CI			
00000062	DYN$C_LOADCODE			
00000063	DYN$C_INIT	SWP$C_KSTACK_WSL		
00000064	DYN$C_CLASSDRV	SGN$C_DFWSCNT		
00000065	DYN$C_CLU			
00000066	DYN$C_PGD			
00000067	DYN$C_JNL			
00000068	BUG$_BADFORKIPL	SWP$C_SHELLPFIL		
00000070	BUG$_BADLCKWSLE			
00000074	BDL$$_CRELNM_ITMLST			
00000078	BUG$_BADMCKCOD	SGN$C_DFWSQUOTA		
00000080	BUG$_BADPAGFILA	DYN$C_SHRBUFIO	DYN$C_SPECIAL	
00000081	SWP$C_SHLFPTE			
00000084	BOO$C_BOOPARSZ			
00000088	BUG$_BADPAGFILD			
00000090	BUG$_BADPAGTYPE	SYS$C_MBXUCBSIZ		
00000094	SS$_DUPLNAM			
00000098	BUG$_BADRSEIPL			
000000A0	BUG$_BADSBMBLK	SWP$K_KSTKSZ		
000000A1	R-CONN_LIST			
000000A8	BUG$_BADSWPVBN			
000000B0	BUG$_BADWCBPT			
000000B8	BUG$_CHMONIS			
000000C0	BUG$_CONTRACT			
000000C8	BUG$_DBLERR			
000000D0	BUG$_DECPTREF			
000000D8	BUG$_DELCONPFN			
000000E0	BUG$_DELGBLSEC			
000000E8	BUG$_DELGBLWCB			
000000F0	BUG$_BADBOOTCB			
000000F8	BUG$_DELWSLEX			
00000100	BUG$_DIRENTRY			
00000108	BUG$_DOUBLDALOC			
00000110	BUG$_DOUBLDEALO			
00000118	BUG$_ERRHALT			
00000120	BUG$_EXHFUL			
00000121	P-EXEC_LIST			
00000128	BUG$_EXPANDPHD			
00000129	R-MAP_QUEUE			
00000130	BUG$_FATALEXCPT			
00000131	R-MNT_MAP_QUEUE			
00000138	BUG$_FREEPAGREF			
00000139	R-TIMER_QUEUE			

Value	Symbols...	
00000140	BUG$_FREWSLX	
00000141	R-MONITOR_LIST	
00000148	BUG$_GBLPAGSZRO	
00000149	R-RESPONDER_NAME	
00000150	BUG$_GBLWSLXERR	
00000154	SS$_IVLOGNAM	
00000158	BUG$_GPGNULPGFL	
00000159	R-CONTROLLER_NAME	
00000160	BUG$_HALT	
00000164	SS$_IVQUOTAL	
00000168	BUG$_HDRNOTMAP	
00000169	R-CONNECT_DATA	
00000170	BUG$_ICONPFNDAT	
00000178	BUG$_ICPAGELOC	
0000017C	SS$_IVSTSFLG	
00000180	BUG$_IFREPAGCNT	
00000188	BUG$_ILLEVTNUM	
00000190	BUG$_ILLVEC	
00000198	BUG$_INCONSTATE	
00000199	R-CY$L_SAVED_UCB	
0000019D	R-CY$L_SAVED_CRB	
000001A0	BUG$_INCPTREF	
000001A1	R-CY$L_SAVED_CDT	
000001A5	R-CY$L_SPARE_CDRP	
000001A8	BUG$_INSNFREPAG	
000001AD	R-CY$INIT	
000001B0	BUG$_INSSWPFIL	
000001B8	BUG$_INSWAPERR	
000001BC	SS$_NOLOGNAM	
000001C0	BUG$_INVCHAN	
000001C8	BUG$_INVEXCEPTN	
000001D0	BUG$_INVPTEFMT	
000001D8	BUG$_INVTQEFMT	
000001E0	BUG$_IVBAKADIO	
000001E8	BUG$_IVGBLTYP	
000001F0	BUG$_IVLISTK	
000001F8	BUG$_IVSSRVRQST	
00000200	BUG$_IVWSETLIST	CTL$C_KRP_SIZE
00000208	BUG$_KRNLSTAKNV	
00000210	BUG$_MACHINECHK	
00000218	BUG$_MAKEWSLE	FIL$C_SIZE
00000220	BUG$_MODRELNBAK	
00000228	BUG$_MFYNULPGFL	
00000230	BUG$_MPWALCIRP	
00000238	BUG$_MTXCNTNONZ	
00000240	BUG$_NETNOBUF	
00000248	BUG$_NETNOSTATE	
00000250	BUG$_NETRCVPKT	
00000258	BUG$_NETSYSSRV	
00000260	BUG$_NETTRANCNT	
00000268	BUG$_NOACPCHAN	
00000270	BUG$_NOACPMAIL	

Value	Symbols...	
00000278	BUG$_NOAQBACP	
00000280	BUG$_NOBUFPCKT	
00000288	BUG$_NOBVPVCB	
00000290	BUG$_NOMULTBK	
00000294	R-CY$CONNECT_REQ	
00000298	BUG$_NONEXSTACP	
000002A0	BUG$_NORCVBUF	
000002A8	BUG$_NOTDDBDDB	
000002B0	BUG$_NOTFCBFCB	
000002B8	BUG$_NOTFCBWCB	
000002C0	BUG$_NOTFCPWCB	
000002C8	BUG$_NOTIRPAQB	
000002D0	BUG$_NOTMTLMTL	
000002D8	BUG$_NOTPCB	
000002E0	BUG$_NOTRVTVCB	
000002E8	BUG$_NOTUCBIRP	
000002F0	BUG$_NOTUCBRVT	
000002F8	BUG$_NOTUCBUCB	
00000300	BUG$_NOTVCBUCB	
00000308	BUG$_NOTVVPVCB	
00000310	BUG$_NOTWCBIRP	
00000318	BUG$_NOUSRWCS	
00000320	BUG$_OUTSWPERR	
00000328	BUG$_PAGEREDERR	
0000032D	R-CY$CONNECT_ERR	R-CY$DISCONNECT
00000330	BUG$_PAGEWRTERR	
00000338	BUG$_PAGNTRNVAL	
00000340	BUG$_PFNLISTCNT	
00000348	BUG$_PFNREFNZRO	
00000350	BUG$_PGFGBLBAD	
00000358	BUG$_PGFIPLHI	
00000360	BUG$_PGFLOCBAD	
00000368	BUG$_PROCGONE	
00000370	BUG$_PTELENVIOL	
00000378	BUG$_PTRCNT	
00000380	BUG$_PURGWSSCN	
00000388	BUG$_QUEUEMPTY	
00000390	BUG$_RDSNONRES	
00000398	BUG$_REFCNTNEG	
0000039C	SS$_NOSLOT	
000003A0	BUG$_RMSBUG	
000003A8	BUG$_SCANDEADPT	
000003B0	BUG$_SECREFNEG	
000003B8	BUG$_SHRCNTNEG	
000003BF	R-CY$DG_INPUT	R-CY$MSG_INPUT
000003C0	BUG$_SSRVEXCEPT	
000003C8	BUG$_STRNOTWCB	
000003D0	BUG$_SWAPWSLE	
000003D8	BUG$_SYSADJWSL	
000003E0	BUG$_SYSTRMERR	
000003E4	EXE$C_SYSPARSZ	
000003F8	BOO$C_SYSPARSZ	BUG$_TIPCUFLOW

Value Symbols...

000003F0 BUG$_UBMAPEXCED
000003F8 BUG$_UNABLCREVA
00000400 BUG$_UNEXUBAINT MAX_BUF_LEN SGN$C_MAXWSCNT
00000408 BUG$_UNKRSTRT
00000410 BUG$_UNXINTEXC
00000418 BUG$_UNXSIGNAL
00000420 BUG$_VBNMAPFAIL
00000428 BUG$_WACKQEMPTY
00000430 BUG$_WRTINVBUF
00000438 BUG$_WRTINVHDR
00000440 BUG$_WRTPGSBAK
00000448 BUG$_WSLENOVAL
00000450 BUG$_WSLPAGCNT
00000458 BUG$_WSLVANVAL
00000460 BUG$_WSLXVANMAT
00000468 BUG$_ZEROPAGE
00000470 BUG$_OPERATOR
00000478 BUG$_BADQHDR
00000480 BUG$_UNKNPRQ
00000488 BUG$_BDPPURGERR
00000490 BUG$_NOSHMGSD
00000498 BUG$_CEBREFNEG
000004A0 BUG$_BRDMSGLOST
000004A4 R-MAINLINE
000004A8 BUG$_MBACBHUNG
000004B0 BUG$_NEGSHBREF
000004B8 BUG$_ACPRECURS
000004C0 BUG$_ACPUNSTAK
000004C8 BUG$_BADRVNWCB R-NEXT_ENTRY
000004D0 BUG$_ERRCACHFUL
000004D8 BUG$_EXTCACHIV
000004E0 BUG$_MAPCNTZER
000004E8 BUG$_NOTUCBWCB
000004F0 BUG$_CHMVEC
000004F8 BUG$_FILCNTNONZ
00000500 BUG$_WSSIZEERR
00000508 BUG$_DEQSUBLCKS
00000510 BUG$_LKBREFNEG
00000516 R-RETURN_CTP_RESPONSE
00000518 BUG$_RSBREFNEG
00000520 BUG$_RSBREFNZRO
00000528 BUG$_SCBRDERR
00000530 BUG$_STATENTSVD
00000538 BUG$_MPBADMCK
00000540 BUG$_MPMCHECK
00000548 BUG$_MPASYNCWRT
00000550 BUG$_MPUNKRSTRT
00000558 BUG$_MPIVLISTK
00000560 BUG$_MPDBLERR
00000568 BUG$_MPHALT
00000570 BUG$_MPILLVEC
00000578 BUG$_MPNOUSRWCS

Value	Symbols...	
00000580	BUG$_MPERRHALT	
00000588	BUG$_MPCHMONIS	
00000590	BUG$_MPCHMVEC	
00000598	BUG$_MPSCBRDERR	
000005A0	BUG$_MPKNLSTKNV	
000005A8	BUG$_MPUNEXPINT	
000005B0	BUG$_LKBGRANTED	
000005B8	BUG$_NOTLKB	
000005C0	BUG$_INVRSPID	
000005C8	BUG$_WCBFCBMNG	
000005D0	BUG$_NOTWCBWCB	
000005D8	BUG$_UDAPORT	
000005E0	BUG$_DISKCLASS	
000005E8	BUG$_CIPORT	
000005F0	BUG$_NOTRULUCB	
000005F8	BUG$_INVCJFIOD	
00000600	BUG$_NOHDJMT	SWP$AL_PTRPAG
00000608	BUG$_JNLDRV	
00000610	BUG$_JNLACP	
00000618	BUG$_NORCP	
00000620	BUG$_RUF	
00000628	BUG$_TAPECLASS	
00000630	BUG$_LOCKMGRERR	
00000638	BUG$_CNXMGRERR	
00000640	BUG$_XQPERR	
00000648	BUG$_INVLOCKID	
00000650	BUG$_SBIAERROR	
00000658	BUG$_WCSCORR	
00000660	BUG$_CPUCEASED	
00000668	BUG$_CLUEXIT	
00000670	BUG$_UNSUPRTCPU	
00000678	BUG$_CJF	
00000680	BUG$_CJFFAILOVR	
00000688	BUG$_OUTOFSYNC	
00000690	BUG$_IVBYTEALGN	
00000698	BUG$_ACCVIOMCHK	
000006A0	BUG$_ACCVIOKSTK	
000006A8	BUG$_MSCPSERV	
000006B0	BUG$_RESEXH	
000006B8	BUG$_CONSOLRX50	
000006C0	BUG$_KRPEMPTY	
000006C8	BUG$_MSCPCLASS	
000006D0	BUG$_ICONCLUDAT	
000006D8	BUG$_OPERCRASH	
00000800	SGN$C_MAXGPGCNT	
00000870	SS$_ENDOFFILE	
00000A00	SWP$SHELINIT	
00000A50	R-CY$NOACT	
00000A73	R-CY$MNT_STATE	
00000AEF	R-CY$MNT_BUFMAP	
00000B61	R-CY$MNT_BUFUNM	
00000B93	R-CY$MNT_MOVBUF	

Value	Symbols...		
00000BF8	R-ALLOC_CB		
00000C23	R-ALLOC_RB		
00000C35	R-ALLOC_CDRP		
00000C5B	R-ALLOC_IRP_CDRP		
00000CA1	R-ALLOC_MNT_CDRP		
00000CC2	R-ALLOC_BUFFER		
00000CD8	R-CLEAR_BUFFER		
00000CE4	R-CLEAN_CONN		
00000D4F	R-DEALLOC_RB		
00000D79	R-CLEAN_CDRP		
00000DD2	R-DEALLOC_MSG_DG		
00000DE8	R-CHECK_CONN_RSP		
00000E1D	R-CHECK_CONN_CTLR		
00000E53	R-ENTER_CONN_LIST		
00000E69	R-CHECK_CTP_REQ		
00001000	SGN$C_MAXPGFL	SGN$C_PHYPAGCNT	
000010CF	R-BUILD_DG	R-BUILD_MSG	
00001143	R-BUILD_BUFFER		
00001191	R-FIND_MAPPED_BUFFER		
000011BD	R-FIND_MAINT_BUFFER		
000011E9	R-QUEUE_DELAY		
00001213	R-TIMER_INTR		
0000126D	R-CY$MONITOR		
000012A0	R-CY$END		
000015A4	CTL$C_CLIDATASZ		
00002000	SGN$C_MAXVPGCNT		
000021BC	SS$_NOSHRIMG		
00002264	SS$_INSSWAPSPACE		
00002A34	SS$_EXPRCLM		
00004000	SGN$C_MAXPAGCNT	SGN$C_PAGEDYN	
00006800	SGN$C_NPAGEDYN		
00018001	RMS$_STALL		
000187BC	RMS$_STR		
0001FFFF	FUNCTION_MASK		
00040000	DEV$M_AVL		
0015827C	LIB$_ATTCONSTO		
04000000	DEV$M_IDV		
08000000	DEV$M_ODV		
3141424D	SYS$C_JOBCTLMB		
3241424D	SYS$C_OPRMBX		
394D3258	SYS$K_VERSION		
7FFE0200	PIO$GL_FMLH		
7FFE0208	PIO$GL_IIOFSPLH		
7FFE0210	PIO$GW_STATUS		
7FFE0212	PIO$GT_ENDSTR		
7FFE0222	PIO$GW_DFPROT		
7FFE0224	PIO$GB_DFMBC		
7FFE0225	PIO$GB_DFMBFSDK		
7FFE0226	PIO$GB_DFMBFSMT		
7FFE0227	PIO$GB_DFMBFSUR		
7FFE0228	PIO$GB_DFMBFREL		
7FFE0229	PIO$GB_DFMBFIDX		

Value	Symbols...	
7FFE022A	PIO$GB_DFMBFHSH	
7FFE022B	PIO$GB_DFNBC	
7FFE022C	PIO$GB_RMSPROLOG	
7FFE022D	PIO$GW_RMSEXTEND	
7FFE0230	PIO$GL_DIRCACHE	
7FFE0238	PIO$GL_DIRCFRLH	
7FFE023C	PIO$GL_RULOCK	
7FFE0240	PIO$GL_NXTIRBSEQ	
7FFE0248	PIO$GW_PIOIMPA	
7FFE0270	PIO$GW_IIOIMPA	
7FFE0314	PIO$AL_RMSEXH	
7FFE0324	PIO$GQ_IIODEFAULT	
7FFE032C	FIL$GT_DDSTRING	PIO$GT_DDSTRING
7FFE0600	PIO$A_TRACE	
7FFE0800	PIO$A_DIRCACHE	
7FFE1600	CTL$A_COMMON	
7FFE1E00	CTL$AG_CMEDATA	CTL$AL_CMCNTX
7FFE2200	CTL$GL_DCLPRSOWN	
7FFE2204	CTL$GL_CLINTOWN	
7FFE2800	NSA$T_IDT	
7FFE2E00	CTL$GL_IAFLINK	
7FFE2E04	CTL$GL_IAFLAST	
7FFE2E08	CTL$GL_FIXUPLNK	
7FFE2E0C	CTL$GL_P1MERGE	
7FFE2E10	CTL$GL_IAFEXE	CTL$GL_IAFPERM
7FFE2E50	IAC$GL_IMAGCTX	
7FFE2E54	IAC$GL_PROCCTX	
7FFE2E58	IAC$AL_VECADDR	
7FFE2E68	IAC$AL_VECOPCOD	
7FFE2E6C	IAC$AW_VECRESET	
7FFE2E74	IAC$AW_VECSET	
7FFE2E7C	IAC$GL_IMAGE_LIST	
7FFE2E84	IAC$GL_WORK_LIST	
7FFE2E8C	IAC$GL_ICBFL	
7FFE2E94	IAC$GL_MAIN_ICB	
7FFE2E98	IAC$GL_FIRST_ICB	
7FFE2E9C	IAC$GL_STACK_SIZE	
7FFE3000	CTL$AL_CLICALBK	
7FFE3008	CTL$AG_CLIMAGE	
7FFE3010	CTL$AG_CLITABLE	
7FFE3018	CTL$GL_UAF_FLAGS	
7FFE301C	CTL$GT_CLINAME	
7FFE303C	CTL$GT_TABLENAME	
7FFE313C	CTL$GT_SPAWNCLI	
7FFE315C	CTL$GT_SPAWNTABLE	
7FFE325C	CTL$AG_CLIDATA	
7FFE4800	IAC$AL_IMGACTBUF	MMG$IMGACTBUF
7FFE6000	CTL$A_DISPVEC	
7FFE6400	MMG$IMGHDRBUF	
7FFE6600	CTL$GL_KRP	
7FFE7000	CTL$GL_KSTKBASEXP	
7FFE7800	CTL$GL_KSTKBAS	

Value	Symbols...	
7FFE7E00	CTL$GL_KSPINI	
7FFEDE00	P1SYSVECTORS	SYS$QIOW
7FFEDE20	SYS$ADJSTK	
7FFEDE28	SYS$ADJWSL	
7FFEDE48	SYS$ASCTIM	
7FFEDF50	SYS$ASSIGN	
7FFEDE58	SYS$BINTIM	
7FFEDE60	SYS$CANCEL	
7FFEDE68	SYS$CANTIM	
7FFEDE70	SYS$CANWAK	
7FFEDE78	SYS$CRMPSC	
7FFEDE90	SYS$CMKRNL	
7FFEDEB8	SYS$CREMBX	
7FFEDEC0	SYS$CREPRC	
7FFEDEC8	SYS$CRETVA	
7FFEDED0	SYS$DACEFC	
7FFEDED8	SYS$DALLOC	
7FFEDEE0	SYS$DASSGN	
7FFEDEE8	SYS$DCLAST	
7FFEDF00	SYS$DELMBX	
7FFEDF08	SYS$DELPRC	
7FFEDF10	SYS$DELTVA	
7FFEDF40	SYS$EXIT	
7FFEDF48	SYS$EXPREG	
7FFEDF50	SYS$FAO	
7FFEDF58	SYS$FAOL	
7FFEDF60	SYS$FORCEX	
7FFEDF88	SYS$HIBER	
7FFEDF90	SYS$IMGACT	
7FFEDFA8	SYS$MGBLSC	
7FFEDFB8	SYS$NUMTIM	
7FFEDFC8	SYS$QIO	
7FFEDFE0	SYS$RUNDWN	
7FFEDFF8	SYS$SETAST	
7FFEE000	SYS$SETEF	
7FFEE008	SYS$SETEXV	
7FFEE020	SYS$SETIMR	
7FFEE028	SYS$SETPRI	
7FFEE030	SYS$SETPRT	
7FFEE038	SYS$SETRWM	
7FFEE040	SYS$SETSFM	
7FFEE058	SYS$TRNLOG	
7FFEE070	SYS$UNWIND	
7FFEE078	SYS$WAITFR	
7FFEE0A8	SYS$SETPFM	
7FFEE0B0	SYS$GETMSG	
7FFEE0C8	SYS$GETCHN	
7FFEE0D8	SYS$GETJPI	
7FFEE0E0	SYS$PUTMSG	
7FFEE100	SYS$SETPRV	
7FFEE180	SYS$GET	
7FFEE188	SYS$PUT	

Value	Symbols...	
7FFEE1A8	SYS$WAIT	
7FFEE1B8	SYS$CLOSE	
7FFEE1C0	SYS$CONNECT	
7FFEE1C8	SYS$CREATE	
7FFEE1D0	SYS$DISCONNECT	
7FFEE208	SYS$OPEN	
7FFEE230	SYS$PARSE	
7FFEE248	SYS$SEARCH	
7FFEE268	SYS$RMSRUNDWN	
7FFEE3C0	SYS$ENQ	
7FFEE3C8	SYS$DEQ	
7FFEE3D0	SYS$ENQW	
7FFEE400	SYS$IMGFIX	
7FFEE418	SYS$GETDVIW	
7FFEE440	SYS$SYNCH	
7FFEE480	SYS$CRELNM	
7FFEE488	SYS$DELLNM	
7FFEE490	SYS$TRNLNM	
7FFEE4A0	SYS$GETLKIW	
7FFEE4B0	SYS$ASCTOID	
7FFEE4C0	SYS$IDTOASC	
7FFEE4E8	SYS$BRKTHRUW	
7FFEE518	CJF$ASSJNL	
7FFEE538	CJF$DEASJNL	
7FFEE548	CJF$DELJNL	
7FFEFE00	CTL$GL_VECTORS	CTL$GW_NMIOCH
7FFEFE02	CTL$GW_CHINDX	
7FFEFE04	CTL$GL_LNMHASH	
7FFEFE08	CTL$GL_LNMDIRECT	
7FFEFE10	CTL$AL_STACK	
7FFEFE28	CTL$GL_CMSUPR	
7FFEFE2C	CTL$GL_CMUSER	
7FFEFE30	CTL$GL_CMHANDLR	
7FFEFE34	CTL$AQ_EXCVEC	
7FFEFE54	CTL$GL_THEXEC	
7FFEFE58	CTL$GL_THSUPR	
7FFEFE60	CTL$GQ_COMMON	
7FFEFE68	CTL$GL_GETMSG	
7FFEFE6C	CTL$AL_STACKLIM	
7FFEFE7C	CTL$GL_CTLBASVA	
7FFEFE80	CTL$GL_IMGHDRBF	
7FFEFE84	CTL$GL_IMGLSTPTR	
7FFEFE88	CTL$GL_PHD	
7FFEFE8C	CTL$GQ_ALLOCREG	
7FFEFE94	CTL$GQ_MOUNTLST	
7FFEFE9C	CTL$T_USERNAME	
7FFEFEA8	CTL$T_ACCOUNT	
7FFEFEB0	CTL$GQ_LOGIN	
7FFEFEB8	CTL$GL_FINALSTS	
7FFEFEBC	CTL$GL_WSPEAK	
7FFEFEC0	CTL$GL_VIRTPEAK	
7FFEFEC4	CTL$GL_VOLUMES	

Value	Symbols...		
7FFEFEC8	CTL$GQ_ISTART		
7FFEFED0	CTL$GL_ICPUTIM		
7FFEFED4	CTL$GL_IFAULTS		
7FFEFED8	CTL$GL_IFAULTIO		
7FFEFEDC	CTL$GL_IWSPEAK		
7FFEFEE0	CTL$GL_IPAGEFL		
7FFEFEE4	CTL$GL_IDIOCNT		
7FFEFEE8	CTL$GL_IBIOCNT		
7FFEFEEC	CTL$GL_IVOLUMES		
7FFEFEF0	CTL$T_NODEADDR		
7FFEFEF7	CTL$T_NODENAME		
7FFEFEFE	CTL$T_REMOTEID		
7FFEFF10	CTL$GQ_PROCPRIV		
7FFEFF18	CTL$GL_USRCHMK		
7FFEFF1C	CTL$GL_USRCHME		
7FFEFF20	CTL$GL_POWERAST		
7FFEFF24	CTL$GB_PWRMODE		
7FFEFF25	CTL$GB_SSFILTER		
7FFEFF28	CTL$AL_FINALEXC		
7FFEFF38	CTL$GL_CCBBASE		
7FFEFF3C	CTL$GQ_DBGAREA		
7FFEFF44	CTL$GL_RMSBASE		
7FFEFF48	CTL$GL_PPMSG		
7FFEFF50	CTL$GB_MSGMASK		
7FFEFF51	CTL$GB_DEFLANG		
7FFEFF52	CTL$GW_PPMSGCHN		
7FFEFF54	CTL$GL_USRUNDWN		
7FFEFF58	CTL$GL_PCB		
7FFEFF5C	CTL$GL_RUF		
7FFEFF60	CTL$GL_SITESPEC		
7FFEFF64	CTL$GL_KNOWNFIL		
7FFEFF68	CTL$AL_IPASTVEC		
7FFEFF88	CTL$GL_CMCNTX		
7FFEFF8C	CTL$GL_IAFLNKPTR		
7FFEFF90	CTL$GL_F11BXQP		
7FFEFF94	CTL$GQ_POALLOC		
7FFEFF9C	CTL$GL_PRCALLCNT		
7FFEFFA0	CTL$GL_RDIPTR		
7FFEFFA8	CTL$GQ_HELPFLAGS		
7FFEFFB0	CTL$GQ_TERMCHAR		
7FFEFFB8	CTL$GL_KRPFL		
7FFEFFBC	CTL$GL_KRPBL		
7FFEFFC0	CTL$GL_CREPRC_FLAGS		
80000010	SYS$CALL_HANDL		
80000A00	MMG$A_ENDVEC	PMS$GL_FCP	PMS$GL_FCP2
80000B18	PMS$GL_TURN		
80000B1C	PMS$GL_SPLIT		
80000B20	PMS$GL_HIT		
80000B24	PMS$GL_DIRHIT		
80000B28	PMS$GL_DIRMISS		
80000B2C	PMS$GL_QUOHIT		
80000B30	PMS$GL_QUOMISS		

Value	Symbols...	
80000B34	PMS$GL_FIDHIT	
80000B38	PMS$GL_FIDMISS	
80000B3C	PMS$GL_EXTHIT	
80000B40	PMS$GL_EXTMISS	
80000B44	PMS$GL_FILHDR_HIT	
80000B48	PMS$GL_FILHDR_MISS	
80000B4C	PMS$GL_DIRDATA_HIT	
80000B50	PMS$GL_DIRDATA_MISS	
80000B54	PMS$GL_STORAGMAP_HIT	
80000B58	PMS$GL_STORAGMAP_MISS	
80000B5C	PMS$GL_OPEN	
80000B60	PMS$GL_OPENS	
80000B64	PMS$GL_ERASEIO	
80000B68	PMS$GL_VOLLCK	
80000B6C	PMS$GL_VOLWAIT	
80000B70	PMS$GL_SYNCHLCK	
80000B74	PMS$GL_SYNCHWAIT	
80000B78	PMS$GL_ACCLCK	
80000B7C	PMS$GL_XQPCACHEWAIT	
80000B80	RMS$GW_GBLBUFQUO	
80000EC0	SWP$A_KSTK	
80000EC8	EXE$GL_BUGCHECK	
80000ED0	IOC$GL_ADPLIST	
80000ED4	IOC$GL_DPTLIST	
80000EDC	TTY$GL_DPT	
80000EE0	NO$GL_DPT	
80000EE4	TTY$GL_JOBCTLMB	
80000EE8	SYS$GL_UIS	
80000EEC	UIS$GL_USB	
80000EF0	SYS$GL_FALLBACK	
80000EF4	SCS$GA_LOCALSB	
80000F48	IOC$GL_DEVLIST	
80000F5C	SYS$GL_BOOTDDB	
80000FA0	SYS$GL_BOOTORB	
80000FF4	SYS$GL_BOOTUCB	
80001124	OPA$GL_DDB	
80001168	OPA$ORB0	
8000118C	OPA$UCB0	
80001330	OPA$CRB	
80001354	CON$INTDISI	
80001378	CON$INTDISO	
800013AC	MB$GL_DDB	
800013F0	MB$ORB0	
80001444	MB$UCB0	
800014D4	MB$GL_UCB1	SYS$GL_JOBCTLMB
80001564	MB$GL_ORB1	
800015B8	MB$GL_UCB2	SYS$GL_OPRMBX
80001648	MB$GL_ORB2	
8000169C	NL$GL_DDB	
800016E0	NL$GL_ORB0	
80001734	NL$GL_UCB0	
800017C4	NET$WCB	

Value	Symbols...	
8000183C	MB$DPT	
80001875	NL$DPT	
800018AE	OP$DPT	
80001913	OPA$VECTOR	
80001950	MB$DDT	MMG$AL_BEGDRIVE
800019C8	EXE$SNDEVMSG	
80001A1A	EXE$WRTMAILBOX	
80001DF4	NL$DDT	
80001E68	MMG$AL_ENDDRIVE	PFN$AL_HEAD
80001E6C	PFN$AL_MFYLSTHD	
80001E80	SCH$GL_FREECNT	
80001E84	SCH$GL_MFYCNT	
80001E8C	PFN$GL_PHYPGCNT	
80001E90	PFN$AL_HILIMIT	SCH$GL_FREEREQ
80001E94	SCH$GL_MFYLIM	
80001E9C	PFN$AL_LOLIMIT	SCH$GL_FREELIM
80001EA0	SCH$GL_MFYLOLIM	
80001EA8	SCH$GL_MFYLIMSV	
80001EAC	SCH$GL_MFYLOSV	
80001EB0	PMS$GL_FAULTS	
80001EB4	PMS$GL_PREADS	PMS$GL_RDFLTS
80001EB8	PMS$GL_PREADIO	
80001EBC	PMS$GL_PWRITES	
80001EC0	PMS$GL_PWRITIO	
80001EC4	PMS$GL_DZROFLTS	
80001EC8	PMS$AL_TRANSFLT	
80001EE8	PMS$GL_DPTSCN	
80001EEC	PMS$GL_GVALID	
80001EF0	MPW$AL_PTE	
80001EF4	MPW$AW_PHVINDEX	
80001EF8	MPW$GL_BADPAGTOTAL	
80001EFC	MMG$GL_PFNLOCK	
80001F08	SCH$AQ_COMH	
80001F0C	SCH$AQ_COMT	
80002008	SCH$AQ_COMOH	
8000200C	SCH$AQ_COMOT	
80002FFC	SCH$AQ_WQHDR	
80002108	SCH$GQ_COLPGWQ	
80002114	SCH$GQ_MWAIT	
8000212C	SCH$GQ_PFWQ	
80002138	SCH$GQ_LEFWQ	
80002144	SCH$GQ_LEFOWQ	
80002150	SCH$GQ_HIBWQ	
8000215C	SCH$GQ_HIBOWQ	
80002168	SCH$GQ_SUSP	
80002174	SCH$GQ_SUSPO	
80002180	SCH$GQ_FPGWQ	
8000218C	SCH$GL_CURPCB	
80002190	SCH$GL_COMQS	
80002194	SCH$GL_COMQQS	
80002198	SCH$GB_SIP	
80002199	SCH$GB_RESCAN	

Value	Symbols...	
8000219A	MMG$GB_FREWFLGS	
8000219C	SCH$GW_PROCCNT	
8000219E	SCH$GW_PROCLIM	
800021A0	SWP$GL_SLOTCNT	
800021A4	SCH$GQ_CEBHD	
800021AC	SCH$GW_CEBCNT	
800021AE	SCH$GW_DELPHDCT	
800021B0	SWP$GL_SHELL	
800021B4	SWP$GL_INPCB	
800021B8	SWP$GL_ISPAGCNT	
800021BC	SWP$GW_IBALSETX	
800021BE	SWP$GB_ISWPRI	
800021C0	SWP$GL_ISWPPAGES	
800021C4	SWP$GL_ISWPCNT	
800021C8	SWP$GL_OSWPCNT	
800021CC	SWP$GL_HOSWPCNT	
800021D0	SWP$GL_HISWPCNT	
800021D4	SCH$GL_RESMASK	
800021D8	SCH$GB_PRI	
800021DC	SWP$GL_SWTIME	
800021E4	MMG$GL_NULLPFL	
80002208	MMG$GL_PAGSWPVC	
8000220C	MMG$GL_MAXPFIDX	
80002210	MMG$GW_MINPFIDX	SGN$GW_SWPFILCT
80002214	EXE$GL_PWRDONE	
80002218	EXE$GL_PWRINTVL	
80002234	SWP$GW_BALCNT	
80002236	SCH$GW_SWPFCNT	
80002238	PHV$GL_PIXBAS	
8000223C	PHV$GL_REFCBAS	
80002240	EXE$GQ_GBLHOOK1	
80002248	EXE$GQ_GBLHOOK2	
80002250	EXE$GQ_GBLHOOK3	
80002258	EXE$GQ_GBLHOOK4	
80002260	EXE$GQ_GBLHOOK5	
80002268	EXE$GQ_GBLHOOK6	
80002270	EXE$GQ_GBLHOOK7	
80002278	EXE$GQ_GBLHOOK8	
80002280	EXE$GQ_GBLHOOK9	
80002288	EXE$GQ_GBLHOOKA	
80002290	EXE$GL_CPUNODSP	
80002294	EXE$GL_CONFREGL	
80002298	EXE$GL_CONFREG	
8000229C	MMG$GL_SBICONF	
800022A0	EXE$GL_NUMNEXUS	
800022A4	MMG$GL_RMSBASE	
800022A8	MMG$GL_FPEMUL_BASE	
800022AC	MMG$GL_SYSLOA_BASE	
800022B0	MMG$GL_VAXEMUL_BASE	
800022B4	MMG$GL_GBLSECFND	
800022B8	MMG$GL_GBLPAGFIL	
80002440	SCH$GL_NULLPCB	

Value	Symbols...
800026E0	SCH$GL_SWPPCB
80002740	SCH$GL_SWPPID
80002800	MMG$AL_SYSPCB
80002920	SCH$GL_PCBVEC
80002924	SCH$GL_SEQVEC
80002928	SCH$GL_MAXPIX
8000292C	SCH$GL_PIXLAST
80002930	SCH$GL_PIXWIDTH
80002934	SCH$GW_LOCALNODE
80002940	EXE$GL_FLAGS
80002944	EXE$GL_STATE_FLAGS
80002948	EXE$GQ_ERLMBX
80002950	EXE$GL_VAXEXCVEC
80002954	EXE$GL_FPEXCVEC
80002958	EXE$GL_USRCHMK
8000295C	EXE$GL_USRCHME
80002960	SWI$GL_FQFL
80002964	SWI$GL_FQBL
80002990	EXE$GL_FKWAITFL
80002994	EXE$GL_FKWAITBL
80002998	LNM$AL_HASHTBL
800029A4	LNM$AL_DIRTBL
800029B0	LNM$AL_MUTEX
800029B4	EXE$GL_SYSUCB
800029B8	FIL$GT_DDDEV
800029C6	FIL$GT_TOPSYS
800029D0	FIL$GQ_CACHE
800029D8	EXE$GQ_BOOTCB_D
800029DC	EXE$GL_BOOTCB
800029E0	EXE$GL_SAVEDUMP
800029E4	EXE$GL_ERASEPB
800029E8	EXE$GL_ERASEPPT
800029F0	IOC$GL_PSFL
800029F4	IOC$GL_PSBL
800029F8	IOC$GL_IRPFL
800029FC	IOC$GL_IRPBL
80002A00	IOC$GL_IRPREM
80002A04	IOC$GL_IRPCNT
80002A08	IOC$GL_IRPMIN
80002A0C	IOC$GL_SRPFL
80002A10	IOC$GL_SRPBL
80002A14	IOC$GL_SRPSIZE
80002A18	IOC$GL_SRPMIN
80002A1C	IOC$GL_SRPSPLIT
80002A20	IOC$GL_SRPREM
80002A24	IOC$GL_SRPCNT
80002A28	IOC$GL_LRPFL
80002A2C	IOC$GL_LRPBL
80002A30	IOC$GL_LRPSIZE
80002A34	IOC$GL_LRPMIN
80002A38	IOC$GL_LRPSPLIT
80002A3C	IOC$GL_LRPREM

Value	Symbols...
80002A40	IOC$GL_LRPCNT
80002A44	IOC$GL_POOLFKB
80002A5C	IOC$GL_PFKBINT
80002A60	EXE$GL_PQBFL
80002A64	EXE$GL_PQBBL
80002A68	IOC$GL_AQBLIST
80002A6C	IOC$GQ_MOUNTLST
80002A74	IOC$GQ_BRDCST
80002A7C	IOC$GL_CRBTMOUT
80002A80	IOC$GL_DU_CDDB
80002A84	IOC$GL_TU_CDDB
80002A88	EXE$GL_GSDGRPFL
80002A8C	EXE$GL_GSDGRPBL
80002A90	EXE$GL_GSDSYSFL
80002A94	EXE$GL_GSDSYSBL
80002A98	EXE$GL_GSDDELFL
80002A9C	EXE$GL_GSDDELBL
80002AA0	EXE$GL_WCBDELFL
80002AA4	EXE$GL_WCBDELBL
80002AA8	EXE$GL_SYSWCBFL
80002AAC	EXE$GL_SYSWCBBL
80002AB0	EXE$GQ_RIGHTSLIST
80002AB8	PMS$GL_KERNEL
80002ACC	PMS$GL_COMPAT
80002AD0	EXE$GL_ABSTIM
80002AD8	EXE$GQ_SYSTIME
80002AE0	EXE$GQ_BOOTTIME
80002AE8	EXE$GL_PFAILTIM
80002AEC	EXE$GL_PFATIM
80002AF0	EXE$GL_TQFL
80002B20	EXE$AL_TQENOREPT
80002B40	IOC$GL_MUTEX
80002B44	EXE$GL_CEBMTX
80002B48	EXE$GL_PGDYNMTX
80002B4C	EXE$GL_GSDMTX
80002B50	EXE$GL_SHMGSMTX
80002B54	EXE$GL_SHMMBMTX
80002B58	EXE$GL_ENQMTX
80002B5C	EXE$GL_ACLMTX
80002B60	EXE$GL_SYSID_LOCK
80002B64	EXE$GL_KNOWN_FILES
80002B7C	EXE$GQ_KFE_LCKNAM
80002B84	EXE$GL_KNOWNFIL
80002B88	EXE$GL_KFIMTX
80002B8C	KFI$GL_F11AACP
80002B90	EXE$GL_GPT
80002B98	SYS$GQ_VERSION
80002BA0	SYS$GW_IJOBCNT
80002BA2	SYS$GW_NJOBCNT
80002BA4	SYS$GW_BJOBCNT
80002BA6	EXE$GW_SCANPIX
80002BA8	EXE$GL_SYSMSG

Value	Symbols...
80002BAC	EXE$GL_USRUNDWN
80002BB0	EXE$GL_NONPAGED
80002BBC	EXE$GL_SPLITADR
80002BC0	EXE$GL_PAGED
80002BC8	RMS$GL_SFDBASE
80002BCC	EXE$GL_SHBLIST
80002BD0	EXE$GL_RTBITMAP
80002BD4	MCHK$GL_MASK
80002BD8	MCHK$GL_SP
80002BDC	EXE$GL_MCHKERRS
80002BE0	EXE$GL_MEMERRS
80002BE4	IO$GL_UBA_INTD
80002BE8	EXE$GL_BLACKHOLE
80002BEC	IO$GL_SCB_INTD
80002BF0	EXE$GL_TENUSEC
80002BF4	EXE$GL_UBDELAY
80002BF8	EXE$GL_MP
80002BFC	EXE$GL_SITESPEC
80002C00	EXE$GL_INTSTKLM
80002C04	LCK$GL_IDTBL
80002C08	LCK$GL_NXTID
80002C0C	LCK$GL_MAXID
80002C10	LCK$GL_HASHTBL
80002C14	LCK$GL_HTBLCNT
80002C18	LCK$GL_TIMOUTQ
80002C20	LCK$GL_DIRVEC
80002C24	LCK$GL_PRCMAP
80002C28	LCK$GQ_BITMAP_EXP
80002C30	LCK$GQ_BITMAP_EXPLCL
80002C38	LCK$GB_HTBLSHFT
80002C39	LCK$GB_MAXDEPTH
80002C3A	LCK$GB_STALLREQS
80002C3B	LCK$GB_REBLD_STATE
80002C3C	EXE$GL_ACMFLAGS
80002C40	NSA$GR_JOURNVEC
80002C68	NSA$GR_ALARMVEC
80002C90	EXE$GL_SVAPTE
80002C94	XQP$GL_SECTIONS
80002C98	XQP$GL_DZRO
80002C9C	XQP$GL_FILESERVER
80002CA0	SYS$GQ_PWD
80002CA8	CIA$GL_MUTEX
80002CAC	CIA$GQ_INTRUDER
800030B8	ERL$AL_BUFADDR
800030C0	ERL$GB_BUFIND
800030C1	ERL$GB_BUFFLAG
800030C2	ERL$GB_BUFPTR
800030C3	ERL$GB_BUFTIM
800030C4	ERL$GL_ERLPID
800030C8	ERL$GL_SEQUENCE
800030E0	LNM$SYSTEM_DIRECTORY
80003108	LNM_SYSTEM_DIR_LNMTH

Value	Symbols...
80003290	PMS$GL_DIRIO
80003294	PMS$GL_BUFIO
80003298	PMS$GL_LOGNAM
8000329C	PMS$GL_MBREADS
800032A0	PMS$GL_MBWRITES
800032A4	PMS$GL_TREADS
800032A8	PMS$GL_TWRITES
800032AC	PMS$GL_IOPFMPDB
800032B0	PMS$GL_IOPFMSEQ
800032B4	PMS$GL_ARRLOCPK
800032B8	PMS$GL_DEPLOCPK
800032BC	PMS$GL_ARRTRAPK
800032C0	PMS$GL_TRCNGLOS
800032C4	PMS$GL_RCVBUFFL
800032C8	PMS$GL_ENQNEW_LOC
800032CC	PMS$GL_ENQNEW_IN
800032D0	PMS$GL_ENQNEW_OUT
800032D4	PMS$GL_ENQCVT_LOC
800032D8	PMS$GL_ENQCVT_IN
800032DC	PMS$GL_ENQCVT_OUT
800032E0	PMS$GL_DEQ_LOC
800032E4	PMS$GL_DEQ_IN
800032E8	PMS$GL_DEQ_OUT
800032EC	PMS$GL_ENQWAIT
800032F0	PMS$GL_ENQNOTQD
800032F4	PMS$GL_BLK_LOC
800032F8	PMS$GL_BLK_IN
800032FC	PMS$GL_BLK_OUT
80003300	PMS$GL_DIQ_IN
80003304	PMS$GL_DIQ_OUT
80003308	PMS$GL_DEQKMSGS_IN
8000330C	PMS$GL_DEQKMSGS_OUT
80003310	PMS$GL_DEQKSRCH
80003314	PMS$GL_DEQEND
80003318	PMS$GL_JNLIRNLS
8000331C	PMS$GL_JNLCHNLS
80003320	PMS$GL_JNLWRTAI
80003324	PMS$GL_JNLWRTBI
80003328	PMS$GL_JNLWRTAT
8000332C	PMS$GL_JNLWRTRU
80003330	PMS$GL_JNLDIRIO
80003334	PMS$GL_JNLBUFIO
80003338	PMS$GL_JNLWRTSS
8000333C	PMS$GL_JNLFORNL
80003340	PMS$GL_JNLFOREL
80003344	PMS$GL_JNLBUFWR
80003348	PMS$GL_JNLWRTFM
8000334C	PMS$GL_BUFACTIV
80003350	PMS$GL_BUFJNLS
80003354	PMS$GL_BUFCHNLS
80003358	PMS$GL_BUFWRTS
8000335C	PMS$GL_BUFREADS

Value	Symbols...	
80003360	PMS$GL_RUFXTNDS	
80003364	PMS$GL_RUFMARK	
80003368	PMS$GL_RUFMRKRB	
8000336C	PMS$GL_RUFABORT	
80003370	PMS$GL_CHMK	
80003374	PMS$GL_CHME	
80003378	PMS$GL_PAGES	
8000337C	PMS$GW_BATCH	
8000337E	PMS$GW_INTJOBS	
80003380	PMS$AL_READTBL	
800033AC	PMS$AL_WRITETBL	
800033D8	PMS$GL_READCNT	
800033DC	PMS$GL_WRTCNT	
800033E0	PMS$GL_PASSALL	
800033E4	PMS$GL_RWP	
800033E8	PMS$GL_LRGRWP	
800033EC	PMS$GL_RWPSUM	
800033F0	PMS$GL_NOSTDTRM	
800033F4	PMS$GL_RWPNOSTD	
800033F8	PMS$GL_TTY_CODE1	
800033FC	PMS$GL_TTY_CODE2	
80003400	PMS$GL_LDPCTX	
80003404	PMS$GL_SWITCH	
80003408	PMS$GB_PROMPT	
8000340C	PMS$GL_DOSTATS	
80003410	EXE$AL_LOAVEC	EXE$MCHK
80003418	EXE$INT54	
8000342C	EXE$INT58	
80003428	EXE$INT5C	
80003430	EXE$INT60	
80003438	UBA$UNEXINT	
80003440	EXE$_XTRA1	
80003448	EXE$EXTRA2	
80003450	EXE$EXTRA3	
80003458	EXE$EXTRA4	
80003460	EXE$EXTRA5	
80003466	ECC$REENABLE	
8000346C	EXE$INIBOOTADP	
80003472	EXE$DUMPCPUREG	
80003478	EXE$REGRESTOR	
8000347E	EXE$REGSAVE	
80003484	EXE$INIPROCREG	
8000348A	EXE$TEST_CSR	
80003490	IOC$PURGDATAP	
80003496	INI$MPMADP	
8000349C	EXE$STARTUPADP	
800034A2	EXE$SHUTDWNADP	
800034A8	MA$RAVAIL	
800034AE	MA$REQUEST	
800034B4	MA$INITIAL	
800034BA	CON$STARTIO	
800034C0	CON$SET_LINE	

Value	Symbols...	
800034C6	CON$DS_SET	
800034CC	CON$XON	
800034D2	CON$XOFF	
800034D8	CON$STOP	
800034DE	CON$STOP2	
800034E4	CON$ABORT	
800034EA	CON$RESUME	
800034F0	CON$SET_MODEM	
800034F6	CON$NULL	
800034FC	CON$DISCONNECT	
80003502	CON$INITIAL	
80003508	CON$INITLINE	
8000350E	CON$INTINP	
80003514	CON$INTOUT	
8000351A	CON$SENDCONSCMD	
80003520	SYSL$CLUSBIA	
80003526	CON$DWNCTY	
8000352C	CON$RELEASECTY	
80003532	EXE$READ_TODR	
80003538	EXE$WRITE_TODR	
8000353E	EXE$INIT_TODR	
80003544	INI$CONSOLE	
8000354A	EXE$INI_TIMWAIT	
80003550	EXE$READP_TODR	
80003556	EXE$WRITEP_TODR	
8000355C	EXE$MOUNTVER	
80003562	EXE$MNTVERSIO	
80003568	EXE$MNTVERSHDOL	
8000356E	EXE$MNTVERSP1	
80003574	EXE$MNTVERSP2	
8000357A	EXE$EXTRA6	
80003580	EXE$EXTRA7	
80003586	EXE$EXTRA8	
8000358C	EXE$EXTRA9	
80003592	EXE$EXTRA10	
80003598	EXE$MCHK_ERRCNT	
8000359C	EXE$LOAD_ERROR	
8000359D	EXE$LOAD_NOP	
8000359E	EXE$LOAD_KCJF	EXE$LOAD_KDISP
800035A4	EXE$LOAD_KRUF	
800035AA	EXE$LOAD_KSPR1	
800035B0	EXE$LOAD_KSPR2	
800035B7	EXE$LOAD_EDISP	EXE$LOAD_ESPR1
800035BD	EXE$LOAD_ESPR2	
800035C4	SCS$GL_CONFIG	
800035CC	SCS$GL_DIRECT	
800035D4	SCS$GL_POLL	
800035DC	SCS$GL_BDT	
800035E0	SCS$GL_CDT	
800035E4	SCS$GL_RDT	
800035E8	SCS$GL_MCLEN	
800035EC	SCS$GL_MCADR	

Value	Symbols...		
800035F0	SCS$GL_MSCP		
800035F4	SCS$GL_PDT		
800035F8	SCS$GA_DFLTMSK		
800035FA	SCS$GW_NEXTBIT		
800035FC	SCS$GA_EXISTS		
80003604	SCS$ACCEPT	SCS$AL_LOAVEC	
8000360A	SCS$ALLOC_CDT		
80003610	SCS$ALLOC_RSPID		
80003616	SCS$CONFIG_PTH		
8000361C	SCS$CONFIG_SYS		
80003622	SCS$CONNECT		
80003628	SCS$DEALL_CDT		
8000362E	SCS$DEALL_RSPID		
80003634	SCS$DISCONNECT		
8000363A	SCS$ENTER		
80003640	SCS$LISTEN		
80003646	SCS$LOCLCONID		
8000364C	SCS$REMOVE		
80003652	IOC$THREAD_IRB		
80003658	SCS$RESUME_WAITR		
8000365E	SCS$UNSTALL_UCB		
80003664	SCS$LKP_RD_CDRP		
8000366A	SCS$LKP_RD_WAIT		
80003670	SCS$RECVL_RSPID		
80003676	SCS$FIND_RDTE		
8000367C	SCS$LKP_MSCWAIT		
80003682	SCS$DIR_LOOKUP		
80003688	SCS$NEW_SB		
8000368E	SCS$POLL_PADC		
80003694	SCS$POLL_MODE		
8000369A	SCS$POLL_MBX		
800036A0	SCS$CANCEL_MBX		
800036A6	SCS$SHUTDOWN		
800036AC	CLU$GL_CLUB		
800036B0	CLU$GL_CLUSVEC		
800036B4	CLU$GW_MAXINDEX		
800036B8	CLU$AL_LOAVEC	CLU$AL_LOAVEC	CLU$GL_LOA_ADDR
800036BC	LCK$SND_CVTREQ		
800036C2	LCK$SND_LOCKREQ		
800036C8	LCK$SND_GRANTED		
800036CE	LCK$SND_DEQGR		
800036D4	LCK$SND_DEQCV		
800036DA	LCK$SND_DEQNT		
800036E0	LCK$SND_BLKING		
800036E6	LCK$SND_RMVDIR		
800036EC	LCK$SND_TIMESTAMP_RQST		
800036F2	LCK$SND_SRCHDLCK		
800036F8	LCK$SND_DEQSEND		
800036FE	LCK$SND_RSBQ_SRCH		
80003704	LCK$CVT_ID_TO_LKB		
8000370A	CNX$ALLOC_CDRP		
80003710	CNX$ALLOC_CDRP_ONLY		

Value	Symbols...		
80003716	CNX$ALLOC_WARMCDRP		
8000371C	CNX$ALLOC_WARMCDRP_CSB		
80003722	CNX$DEALL_MSG_BUF_CSB		
80003728	CNX$DEALL_WARMCDRP_CSB		
8000372E	CNX$INIT_CDRP		
80003734	CNX$SEND_MNY_MSGS		
8000373A	CNX$SEND_MSG		
80003740	CNX$SEND_MSG_CSB		
80003746	CNX$SEND_MSG_RESP		
8000374C	CNX$SEND_MSG_RSPID		
80003752	CNX$BLOCK_XFER		
80003758	CNX$BLOCK_XFER_IRP		
8000375E	CNX$PARTNER_INIT_CSB		
80003764	CNX$BLOCK_READ		
8000376A	CNX$BLOCK_READ_IRP		
80003770	CNX$BLOCK_WRITE		
80003776	CNX$BLOCK_WRITE_IRP		
8000377C	CNX$PARTNER_FINISH		
80003782	CNX$PARTNER_RESPOND		
80003788	CNX$CHANGE_QUORUM		
8000378E	CNX$SHUTDOWN		
80003794	CNX$POWER_FAIL		
8000379A	CNX$DISK_CHANGE		
800037A0	EXE$ALLOC_CSD		
800037A6	EXE$DEALLOC_CSD		
800037AC	EXE$CSP_BRDCST		
800037B2	EXE$CSP_CALL		
800037B8	EXE$CSP_COMMAND		
800037BE	EXE$CSP_BRKTHRU		
800037C4	LKI$SND_STOREQ		
800037CA	LKI$SND_BLKING		
800037D0	LKI$SND_BLKBY		
800037D6	LKI$SND_LOCKS		
800037DC	EXE$GL_CDTBASE		
800037E0	EXE$DEASJNL		
800037E6	EXE$FORCEJNL		
800037EC	EXE$FORCEJNLW		
800037F2	EXE$WRITEJNL		
800037F8	EXE$WRITEJNLW		
80003800	EXE$GL_RDTBASE		
80003804	EXE$ERRPAT_VEC		
8000380A	EXE$MTACCESS_VEC		
80003810	PAT$A_NONPGD_DATA		
80003C00	EXE$A_SYSPARAM	EXE$GQ_TODCBASE	MMG$A_SYSPARAM
80003C08	EXE$GL_TODR		
80003C0C	SGN$GW_DEVPFC		
80003C0E	SGN$GB_DSTBPFC		
80003C0F	SGN$GB_SYSPFC		
80003C10	SGN$GB_KFILSTCT		
80003C12	SGN$GW_GBLSECNT		
80003C14	SGN$GL_MAXGPGCT		
80003C18	SGN$GL_GBLPAGFIL		

Value	Symbols...	
80003C1C	SGN$GW_MAXPRCCT	
80003C1E	SGN$GW_PIXSCAN	
80003C20	SGN$GW_MAXPSTCT	
80003C22	SGN$GW_MINWSCNT	
80003C24	SGN$GW_PAGFILCT	
80003C26	SGN$GW_SWPFILES	
80003C28	SGN$GW_SYSDWSCT	
80003C2A	SGN$GW_ISPPGCT	
80003C2C	LCK$GL_EXTRASTK	
80003C30	SGN$GL_BALSETCT	
80003C34	SGN$GL_IRPCNT	
80003C38	SGN$GL_IRPCNTV	
80003C3C	SGN$GL_MAXWSCNT	
80003C40	SGN$GL_NPAGEDYN	
80003C44	SGN$GL_NPAGEVIR	
80003C48	SGN$GL_PAGEDYN	
80003C4C	SGN$GL_MAXVPGCT	
80003C50	SGN$GL_SDTREQ	
80003C54	SGN$GL_[illegible]SRSTK	
80003C58	SGN$GL_LRPCNT	
80003C5C	SGN$GL_LRPCNTV	
80003C60	SGN$GL_LRPSIZE	
80003C64	SGN$GL_LRPMIN	
80003C68	SGN$GL_SRPCNT	
80003C6C	SGN$GL_SRPCNTV	
80003C70	SGN$GL_SRPSIZE	
80003C74	SGN$GL_SRPMIN	
80003C78	SGN$GW_PQUANCNT	
80003C7A	SGN$GW_PIOPAGES	
80003C7C	SGN$GW_CTLPAGES	
80003C7E	SGN$GW_[illegible]IMGLIM	
80003C80	SGN$GW_IMGIOCNT	
80003C82	SCH$GW_QUAN	
80003C84	MPW$AW_INITVAL	MPW$GW_MPWPFC
80003C86	MPW$GW_HILIM	
80003C88	MPW$GW_LOLIM	
80003C8A	MPW$GB_PRIO	
80003C8B	SWP$GB_PRIO	
80003C8C	MPW$GL_THRESH	
80003C90	MPW$GL_WAITLIM	
80003C94	SGN$GW_WSLMXSKP	
80003C98	MMG$GL_PHYPGCNT	
80003C9C	SCH$GL_PFRATL	
80003CA0	SCH$GL_PFRATH	
80003CA4	SCH$GL_PFRATS	
80003CA8	SCH$GL_WSINC	
80003CAC	SCH$GL_WSDEC	
80003CB0	SCH$GW_AWSMIN	
80003CB4	SCH$GL_AWSTIME	
80003CB8	SCH$GL_SWPRATE	
80003CBC	SWP$GL_SWPPGCNT	
80003CC0	SWP$GW_SWPINC	

Value	Symbols...
80003CC2	SCH$GW_IOTA
80003CC4	SCH$GW_LONGWAIT
80003CC6	SCH$GW_DORMANTWAIT
80003CC8	SCH$GW_SWPFAIL
80003CCC	SGN$GL_VMSD1
80003CDC	SGN$GL_VMSD2
80003CD4	SGN$GL_VMSD3
80003CD8	SGN$GL_VMSD4
80003CDC	SGN$GL_VMS5
80003CE0	SGN$GL_VMS6
80003CE4	SGN$GL_VMS7
80003CE8	SGN$GL_VMS8
80003CEC	SGN$GL_USERD1
80003CF0	SGN$GL_USERD2
80003CF4	SGN$GL_USER3
80003CF8	SGN$GL_USER4
80003CFC	[illegible]N$GL_EXTRACPU
80003D00	EXE$GL_SYSUIC
80003D04	IOC$GW_MVTIMEOUT
80003D06	IOC$GW_MAXBUF
80003D08	IOC$GW_MBXBFCUO
80003D0A	IOC$GW_MBXMXMSG
80003D0C	IOC$GW_[illegible]
80003D10	SGN$GL_[illegible]
80003D14	SGN$GL_FREEGOAL
80003D18	SCH$GL_GROWLIM
80003D1C	SCH$GL_BORROWL
80003D20	EXE$GL_LOCKRTRY
80003D24	IOC$GW_XFMXRATE
80003D26	IOC$GW_LAMAPREG
80003D28	EXE$GL_RTIMESPT
80003D2C	EXE$GL_CLITABL
80003D30	LCK$GL_IDTBLSIZ
80003D34	LCK$GL_IDTBLMAX
80003D38	LCK$GL_HTBLSIZ
80003D3C	LCK$GL_WAITTIME
80003D40	SCS$GW_BDTCNT
80003D42	SCS$GW_CDTCNT
80003D44	SCS$GW_RDTCNT
80003D46	SCS$GW_MAXDG
80003D48	SCS$GW_MAXMSG
80003D4A	SCS$GW_FLOWCUSH
80003D4C	SCS$GB_SYSTEMID
80003D50	SCS$GB_SYSTEMIDH
80003D54	SCS$GB_NODENAME
80003D5C	SCS$GW_PRCPOLINT
80003D5E	SCS$GW_PASTMOUT
80003D60	SCS$GW_PAPPDDG
80003D62	SCS$GB_PANPOLL
80003D63	SCS$GB_PAMXPORT
80003D64	SCS$GW_PAPOLINT
80003D66	SCS$GW_PAPOOLIN

Value	Symbols...	
80003D68	SCS$GB_PASANITY	
80003D69	SCS$GB_PANOPOLL	
80003D6C	SGN$GL_PE1	
80003D70	SGN$GL_PE2	
80003D74	SGN$GL_PE3	
80003D78	SGN$GL_PE4	
80003D7C	SGN$GL_PE5	
80003D80	SGN$GL_PE6	
80003D84	SGN$GW_TPWAIT	
80003D86	SCS$GB_UDABURST	
80003D88	LNM$GL_HTBLSIZS	
80003D8C	LNM$GL_HTBLSIZP	
80003D90	EXE$GL_DEFFLAGS	
80003D94	EXE$GL_DYNAMIC_FLAGS	
80003D98	EXE$GL_STATIC_FLAGS	
80003D9C	EXE$GL_MSGFLACS	
80003DA0	SGN$GL_LOADFLAGS	
80003DA4	TTY$GL_DELTA	
80003DA8	TTY$GB_DIALTYP	
80003DA9	TTY$GB_DEFSPEED	
80003DAA	TTY$GB_RSPEED	
80003DAB	TTY$GB_PARITY	
80003DAC	TTY$GW_DEFBUF	
80003DB0	TTY$GL_DEFCHAR	
80003DB4	TTY$GL_DEFCHAR2	
80003DB8	TTY$GW_TYPAHDSZ	
80003DBA	TTY$GW_ALTYPAHD	
80003DBC	TTY$GW_ALTALARM	
80003DBE	TTY$GW_DMASIZF	
80003DC0	TTY$GW_PROT	
80003DC4	TTY$GL_OWNUIC	
80003DC8	TTY$GW_CLASSNAM	
80003DCA	TTY$GB_SILOTIME	
80003DCC	TTY$GL_TIMEOUT	
80003DD0	TTY$GB_AUTOCHAR	
80003DD4	TTY$GL_DEFPORT	
80003DD8	SYS$GB_DFMBC	
80003DD9	SYS$GB_DFMBFSDK	
80003DDA	SYS$GB_DFMBFSMT	
80003DDB	SYS$GB_DFMBFSUR	
80003DDC	SYS$GB_DFMBFREL	
80003DDD	SYS$GB_DFMBFIDX	
80003DDE	SYS$GB_DFMBFHSH	
80003DDF	SYS$GB_RMSPROLOG	
80003DE0	SYS$GW_RMSEXTEND	
80003DE2	SYS$GW_FILEPROT	
80003DE4	PQL$AL_DEFAULT	SYS$GW_GBLBUFQUO
80003DE6	SYS$GB_DFNBC	
80003DE8	PQL$GDASTLM	
80003DEC	PQL$GDBIOLM	
80003DF0	PQL$GDBYTLM	
80003DF4	PQL$GDCPULM	

Value	Symbols...	
80003DF8	PQL$GDDIOLM	
80003DFC	PQL$GDFILLM	
80003E00	PQL$GDPGFLQUOTA	
80003E04	PQL$GDPRCLM	
80003E08	PQL$GDTQELM	
80003E0C	PQL$GDWSQUOTA	
80003E10	PQL$GDWSDEFAULT	SGN$GW_DFWSCNT
80003E14	PQL$GDENQLM	
80003E18	PQL$GDWSEXTENT	
80003E1C	PQL$GDJTQUOTA	
80003E20	PQL$AL_MIN	
80003E24	PQL$GMASTLM	
80003E28	PQL$GMBIOLM	
80003E2C	PQL$GMBYTLM	
80003E30	PQL$GMCPULM	
80003E34	PQL$GMDIOLM	
80003E38	PQL$GMFILLM	
80003E3C	PQL$GMPGFLQUOTA	
80003E40	PQL$GMPRCLM	
80003E44	PQL$GMTQELM	
80003E48	PQL$GMWSQUOTA	
80003E4C	PQL$GMWSDEFAULT	
80003E50	PQL$GMENQLM	
80003E54	PQL$GMWSEXTENT	
80003E58	PQL$GMJTQUOTA	
80003E5F	PQL$AB_FLAG	
80003E70	ACP$GW_MAPCACHE	
80003E72	ACP$GW_HDRCACHE	
80003E74	ACP$GW_DIRCACHE	
80003E76	ACP$GW_WORKSET	
80003E78	ACP$GW_FIDCACHE	
80003E7A	ACP$GW_EXTCACHE	
80003F7C	ACP$GW_EXTLIMIT	
80003E7E	ACP$GW_QUOCACHE	
80003E80	ACP$GW_SYSACC	
80003E82	ACP$GB_MAXREAD	
80003E83	ACP$GB_WINDOW	
80003E84	ACP$GB_WRITBACK	
80003E85	ACP$GB_DATACHK	
80003E86	ACP$GB_BASEPRIO	
80003E87	ACP$GB_SWAPFLGS	
80003E88	SYS$GB_DEFPRI	
80003E8A	SYS$GW_IJOBLIM	
80003E8C	SYS$GW_BJOBLIM	
80003E8E	SYS$GW_NJOBLIM	
80003E90	SYS$GW_RJOBLIM	
80003E94	SYS$GB_DEFQUEPRI	
80003E98	SYS$GB_MAXQUEPRI	
80003E9C	SYS$GB_PWD_TMO	
80003E9D	SYS$GB_RETRY_LIM	
80003E9E	SYS$GB_RETRY_TMO	
80003E9F	SYS$GB_BRK_LIM	

Value	Symbols...	
80003EA0	SYS$GL_BRK_TMO	
80003EA4	SYS$GL_HID_TIM	
80003EA8	CLU$GB_VAXCLUSTER	
80003EAA	CLU$GW_QUORUM	
80003EAC	CLU$GW_VOTES	
80003EAE	CLU$GW_RECNXINT	
80003EB0	CLU$GB_QDISK	
80003EC0	CLU$GW_QDSKVOTES	
80003EC2	CLU$GW_QDSKINTERVAL	
80003EC4	CLU$GL_ALLOCLS	
80003EC8	CLU$GW_LCKDIRWT	
80003ECA	SGN$GB_TAILORED	
80003ECC	SGN$GB_STARTUP_P1	
80003ED0	SGN$GB_STARTUP_P2	
80003ED4	SGN$GB_STARTUP_P3	
80003ED8	SGN$GB_STARTUP_P4	
80003EDC	SGN$GB_STARTUP_P5	
80003EE0	SGN$GB_STARTUP_P6	
80003EE4	SGN$GB_STARTUP_P7	
80003EE8	SGN$GB_STARTUP_P8	
80003EEC	SWP$GL_SHELLSIZ	
80003EF0	SWP$GW_BAKPTE	
80003EF2	SWP$GW_EMPTPTE	
80003EF4	SWP$GW_WSLPTE	
80003EF5	SWP$GB_SHLP1PT	
80003EF8	SWP$GL_BSLOTSZ	
80003EFC	SWP$GL_MAP	
80003F00	SWP$GL_PHDBASVA	
80003F04	SGN$GL_PHDAPCNT	
80003F08	SGN$GL_PHDLWCNT	
80003F0C	SGN$GL_P1LWCNT	
80003F10	SGN$GL_PHDPAGCT	
80003F14	SGN$GL_PTPAGCNT	
80003F18	MMG$GL_CTLBASVA	
80003F1C	EXE$AL_STACKS	
80003F20	EXE$GL_INTSTK	
80003F24	MMG$GL_GPTBASE	
80003F28	MMG$GL_GPTE	
80003F2C	MMG$GL_MAXGPTE	
80003F30	MMG$GL_FRESVA	MMG$GL_MAXSYSVA
80003F34	MMG$GL_SPTBASE	
80003F38	MMG$GL_SPTLEN	
80003F3C	MMG$GL_SYSPHD	
80003F40	MMG$GL_SYSPHDLN	
80003F44	SWP$GL_BALBASE	
80003F48	SWP$GL_BALSPT	
80003F4C	MMG$GL_SBR	
80003F50	MMG$GL_NPAGEDYN	
80003F54	MMG$GL_NPAGNEXT	
80003F58	MMG$GL_IRPNEXT	
80003F5C	MMG$GL_LRPNEXT	
80003F60	MMG$GL_SRPNEXT	

Value	Symbols...			
80003F64	MMG$GL_PAGEDYN			
80003F68	MMG$GL_MAXPFN			
80003F6C	MMG$GL_MINPFN			
80003F70	MMG$GL_MAXMEM			
80003F74	EXE$GL_RPB			
80003F78	BOO$GL_SPTFREL			
80003F7C	BOO$GL_SPTFREH			
80003F80	EXE$GL_SCB			
80003F84	EXE$GL_ARCHFLAG			
80003F88	EXE$GB_CPUDATA			
80003F98	EXE$GB_CPUTYPE			
80003F99	PFN$GB_LENGTH			
80003F9A	MMG$GW_BIGPFN			
80003F9C	EXE$GW_PGFL_FID			
80003FA4	PFN$AL_PTE	PFN$A_BASE		
80003FA8	PFN$AL_BAK			
80003FAC	PFN$AW_REFCNT			
80003FB0	PFN$AX_FLINK	PFN$AX_SHRCNT		
80003FB4	PFN$AX_BLINK	PFN$AX_WSLX		
80003FB8	PFN$AW_SWPVBN			
80003FBC	PFN$AB_STATE			
80003FC0	PFN$AB_TYPE			
80003FC4	EXE$GT_STARTUP			
80003FE4	MMG$GL_PGDCOD	PAT$GL_EXP_NPG2		
80004000	EXE$RESTART	MMG$FRSTRONLY		
800041B6	EXE$INIT_DEVICE			
80004256	EXE$PWRTIMCHK			
8000427E	EXE$BUG_CHECK			
80004362	MPH$BUGCHKHK			
80004403	EXE$BOOTCB_CHK			
8000442A	PMS$ABORT_RQ			
80004445	PMS$END_IO			
80004465	PMS$END_RQ			
80004494	PMS$START_IO			
800044B7	PMS$START_RQ			
800045A9	EXE$AB_HEXTAB			
800045BC	ERL$VEC256	ERL$VEC320	ERL$VEC384	ERL$VEC448
800045C0	ERL$VEC260	ERL$VEC324	ERL$VEC388	ERL$VEC452
800045C4	ERL$VEC264	ERL$VEC328	ERL$VEC392	ERL$VEC456
800045C8	ERL$VEC268	ERL$VEC332	ERL$VEC396	ERL$VEC460
800045CC	ERL$VEC272	ERL$VEC336	ERL$VEC400	ERL$VEC464
800045D0	ERL$VEC276	ERL$VEC340	ERL$VEC404	ERL$VEC468
800045D4	ERL$VEC280	ERL$VEC344	ERL$VEC408	ERL$VEC472
800045D8	ERL$VEC284	ERL$VEC348	ERL$VEC412	ERL$VEC476
800045DC	ERL$VEC288	ERL$VEC352	ERL$VEC416	ERL$VEC480
800045E0	ERL$VEC292	ERL$VEC356	ERL$VEC420	ERL$VEC484
800045E4	ERL$VEC296	ERL$VEC360	ERL$VEC424	ERL$VEC488
800045E8	ERL$VEC300	ERL$VEC364	ERL$VEC428	ERL$VEC492
800045EC	ERL$VEC304	ERL$VEC368	ERL$VEC432	ERL$VEC496
800045F0	ERL$VEC308	ERL$VEC372	ERL$VEC436	ERL$VEC500
800045F4	ERL$VEC312	ERL$VEC376	ERL$VEC440	ERL$VEC504
800045F8	ERL$VEC316	ERL$VEC380	ERL$VEC444	ERL$VEC508

Value		Symbols...			
80004674	ERL$UNEXP	ERL$VEC0	ERL$VEC100	ERL$VEC104	
	ERL$VEC108	ERL$VEC112	ERL$VEC116	ERL$VEC12	
	ERL$VEC120	ERL$VEC124	ERL$VEC128	ERL$VEC132	
	ERL$VEC136	ERL$VEC140	ERL$VEC144	ERL$VEC148	
	ERL$VEC152	ERL$VEC156	ERL$VEC16	ERL$VEC160	
	ERL$VEC164	ERL$VEC168	ERL$VEC172	ERL$VEC176	
	ERL$VEC180	ERL$VEC184	ERL$VEC188	ERL$VEC192	
	ERL$VEC196	ERL$VEC20	ERL$VEC200	ERL$VEC204	
	ERL$VEC208	ERL$VEC212	ERL$VEC216	ERL$VEC220	
	ERL$VEC224	ERL$VEC228	ERL$VEC232	ERL$VEC236	
	ERL$VEC24	ERL$VEC240	ERL$VEC244	ERL$VEC248	
	ERL$VEC252	ERL$VEC28	ERL$VEC32	ERL$VEC36	
	ERL$VEC4	ERL$VEC40	ERL$VEC44	ERL$VEC48	
	ERL$VEC52	ERL$VEC56	ERL$VEC60	ERL$VEC64	
	ERL$VEC68	ERL$VEC72	ERL$VEC76	ERL$VEC8	
	ERL$VEC80	ERL$VEC84	ERL$VEC88	ERL$VEC92	
	ERL$VEC96				
8000467C	ERL$VEC_RETURN				
80004684	EXE$ACVIOLAT				
800046AC	EXE$ARITH				
800046B7	EXE$ASTFLT				
800046CC	EXE$BREAK				
800046D4	EXE$CMODSUPR				
800046EC	EXE$CMODUSER				
8000470C	EXE$COMPAT				
8000474C	EXE$KERSTKNV				
80004750	EXE$MCHECK				
80004758	EXE$OPCCUS				
80004760	EXE$OPCDEC				
80004792	EXE$PAGRDERR				
8000479C	EXE$RADRMOD				
800047A4	EXE$ROPRAND				
800047AC	EXE$TBIT				
800047C0	EXE$SSFAIL				
800047CF	EXE$EXCEPTION				
80004833	EXE$MCHK_PRTCT				
8000485B	EXE$MCHK_BUGCHK				
8000489D	EXE$MCHK_TEST				
800048B8	INI$MASTERWAKE				
800048C4	IOC$IOPOST				
800049E2	IOC$BUFPOST				
80004D2F	IOC$QNXTSEG				
80004D3B	IOC$QNXTSEG1				
80004DD8	IOC$WAKACP				
80004F66	IOC$DIRPOST1				
80005064	EXE$POWERFAIL				
80005C58	MMG$PAGEFAULT				
80005F56	MMG$SVPCTX				
80005F84	MMG$RESRCWAIT				
80005F9F	MMG$PGFLTWAIT				
8000639A	MMG$WSLEPEN				
80006467	MMG$FREWSLE				

Value	Symbols...
800064FD	MMG$FREWSLX
80006522	MPH$INVALIDHK
8000652A	MMG$FRE_TRYSKIP
800065ED	MMG$DELWSLEX
80006611	MMG$DELWSLEPPG
80006637	MMG$ININEWPFN
80006679	MMG$MAKEWSLE
800066DD	MMG$LOCKPGTB
800066E3	MMG$INCPTREF
80006721	MMG$DECPTREF
80006788	MMG$DECPHDREF
8000678C	MMG$DECPHDREF1
80006797	MMG$INIBLDPKT
800067EC	MMG$ALLOCPFN
80006801	MMG$DELCONPFN
80006883	MMG$REMPFNH
8000688E	MMG$REMPFN
800068E5	MMG$RLPFNSAVPTE
800068FA	MMG$DELPFNLST
80006902	MMG$RELPFN
80006962	MMG$DALCBAKSTORE
8000698A	MMG$INSPFNH
800069C6	MMG$DALLOCPFN
800069C8	MMG$INSPFNT
80006A4C	MMG$ALLOCONTIG
80006AAB	MMG$IOLOCK
80006BB0	MMG$IOLOCKPAG
80006CDE	MMG$UNLOCK
80006D31	MMG$REFCNTNEG
80006D35	MMG$SHRCNTNEG
80006D39	MMG$ALLOCSWPAREA
80006D71	MMG$ALLOCPAGFIL1
80006EDA	MMG$ALLOCPAGFIL2
80006F65	MMG$DALCPAGFIL
80006F6E	MMG$DEALLOCPAGFIL
80007059	MMG$CALCSWAPSIZE
8000708F	MMG$SUBSECREF
8000709C	MMG$DECSECREF
8000717E	MMG$SVAPTECHK
80007197	MMG$PTEADRCHK
800071A0	MMG$PTEREF
800071C5	MMG$PTEINDXCHK
800071CB	MMG$PTEINDX
80007316	MMG$WRTMFYPAG
80007650	MMG$SHRINKWS
800076D9	MMG$EXTRADYNWS
80007727	MMG$MOVPTLOCK1
80007728	MMG$MOVPTLOCK
80007868	MMG$DELPAG
80007B70	MMG$PAGETYPE
80007BF2	MMG$DELGBLSEC
80007D02	MMG$GETPTIPAG

Value		Symbols...
80007DBC	MMG$ULKGBLWSLE	
80007DCD	MMG$LCKULKPAG	
80007F72	MMG$SWAPWSLE	
80007FAE	MMG$SCNWSLX	
80007FE7	MMG$EXPKSTK	
800080A9	MMG$PURGWSSCN	
80008188	MMG$SETPRTPAG	
80008306	MMG$WRTPGSBAK	
800085FE	MMG$FINDGSDPFN	
80008674	MMG$DECSHMREF	
80008677	MMG$INCSHMREF	
80008696	MMG$VALIDATEGSD	
8000869A	MMG$GETNXTGSD	
80008711	MMG$FINDSHD	
8000878A	SCH$OSWPSCHED	
800089BF	MMG$MPWCHECK	
800089E2	BUG$FATAL	EXE$INSTIMQ
80008A09	EXE$RMVTIMQ	
80008A7B	EXE$FORK_WAIT	
80008A89	EXE$NULLPROC	
80008A8E	ACP$ACCESS	
80008A93	ACP$ACCESSNET	
80008ABF	ACP$DEACCESS	
80008B15	ACP$MODIFY	
80008B1D	ACP$MOUNT	
80008B3C	ACP$READBLK	
80008B78	ACP$WRITEBLK	
80009093	XQP$BLOCK_ROUTINE	
800090EF	XQP$DEQBLOCKER	
80009103	XQP$REL_QUOTA	
8000911A	XQP$UNLOCK_QUOTA	
80009169	XQP$FCBSTALE	
80009195	EXE$CANCELN	
800091A4	EXE$CANCEL	
800092CA	EXE$CANTIM	
800092E8	EXE$CANWAK	
80009303	EXE$CMEXEC	
80009313	EXE$CMKRNL	
80009334	EXE$DERLMB	
8000936F	EXE$FORCEX	
800093B4	EXE$IORSNWAIT	
800093DD	EXE$ONEPARM	
800093E3	EXE$ZEROPARM	
800093E9	EXE$LCLDSKVALID	
8000941D	EXE$MODIFY	
80009423	EXE$READ	
8000942C	EXE$WRITE	
8000944F	EXE$READLOCK	
80009452	EXE$WRITELOCK	
80009455	EXE$MODIFYLOCK	
80009458	EXE$MODIFYLOCKR	
80009462	EXE$READLOCKR	

Value	Symbols...	
80009469	EXE$WRITELOCKR	
800094E6	EXE$READCHK	
800094EC	EXE$WRITECHK	
800094FA	EXE$READCHKR	
80009558	EXE$WRITECHKR	
800095A9	EXE$SETCHAR	
800095C0	EXE$SETMODE	
800095E6	EXE$SENSEMODE	
8000960E	EXE$CARRIAGE	
80009656	EXE$SCHDWK	
800096B8	EXE$SETIMR	
80009829	EXE$QIO	
80009A92	EXE$BLDPKTGSR	
80009A9A	EXE$BLDPKTGSW	
80009AA2	EXE$BLDPKTSWPR	
80009AAA	EXE$BLDPKTSWPW	
80009AB2	EXE$BUILDPKTW	
80009ABA	EXE$BUILDPKTR	
80009B1E	EXE$ABORTIO	
80009B2B	EXE$FINISHIOC	
80009B2D	EXE$FINISHIO	
80009B41	EXE$QIODRVPKT	
80009B45	EXE$ALTQUEPKT	
80009B58	EXE$QIOACPPKT	
80009B77	EXE$QIORETURN	
80009B88	EXE$QXQPPKT	
80009BA9	EXE$INSIOQ	
80009BC7	EXE$INSERTIRP	
80009BE5	EXE$SETPRI	
80009C84	MT$CHECK_ACCESS	
80009C98	SCH$ASTDEL	
80009CAC	MPH$ASTDELHK	
80009CB2	MPH$ASTDELCONT	
80009CC6	MPH$QEMPTYCONT	
80009D63	EXE$ASTDEL	
80009D66	EXE$ASTRET	
80009E46	EXE$IPAPBKAST	
80009E6B	MPH$QAST	SCH$QAST
80009F09	SCH$NEWLVL	
80009F2C	MPH$NEWLVLHK	
80009F3D	SCH$SWAPACBS	
80009F40	SCH$REMOVACB	
80009F44	EXE$IOFORK	
80009F48	EXE$FORK	
80009F64	EXE$FRKIPL6DSP	
80009F70	EXE$FRKIPL9DSP	
80009F7C	EXE$FRKIPL10DSP	
80009F88	EXE$FRKIPL11DSP	
80009F94	EXE$FRKIPL8DSP	
80009F9C	EXE$FORKDSPTH	
80009FC8	EXE$HWCLKINT	
80009FD0	EXE$UBCLKINT	

Value	Symbols...	
8000A028	EXE$SWTIMINT	
8000A0FC	EXE$TIMEOUT	
8000A27D	SCH$RSE	
8000A2CA	SCH$UNWAIT	
8000A305	SCH$CHSE	
8000A321	SCH$CHSEP	
8000A3A7	SCH$QEND	
8000A40A	SCH$FORCEDEXIT	
8000A50D	SCH$WAKE	
8000A532	SCH$SWPWAKE	
8000A5A4	MPH$RESCHED	SCH$RESCHED
8000A5CA	MPH$SCHED	SCH$SCHED
8000A62D	SWP$GL_SHELIO	
8000A631	EXE$BUFFRQUOTA	
8000A63D	EXE$BUFQUOPRC	
8000A653	EXE$SNGLEQUOTA	
8000A656	EXE$MULTIQUOTA	
8000A68A	EXE$PROBER	
8000A6D3	EXE$PROBEW	
8000A714	EXE$ALLOCCEB	
8000A71D	EXE$ALLOCJIB	
8000A726	EXE$ALLOCIRP	
8000A72E	EXE$ALLOCPCB	
8000A737	EXE$ALLOCTQE	
8000A775	EXE$ALLOCBUF	
8000A7A3	EXE$ALONPAGWAITS	
8000A7B0	EXE$ALONPAGWAIT	
8000A7C3	EXE$ALOPAGWAIT	
8000A801	EXE$ALONONPAGED	
8000A845	EXE$ALONPAGVAR	
8000A894	EXE$ALOPAGED	
8000A8CE	EXE$ALLOCATE	
8000A8F9	EXE$DEANONPAGED	
8000A905	EXE$DEANONPGDSIZ	
8000A95D	EXE$DEAPAGED	
8000A961	EXE$DEAPGDSIZ	
8000A9AA	EXE$DEALLOCATE	
8000AA00	EXE$ALOSHARED	
8000AA1D	EXE$DEASHARED	
8000AA5C	EXE$EXTENDPOOL	
8000AC34	EXE$ALOPHYCNTG	
8000ACA5	SCH$RWAIT	
8000ACAF	SCH$LOCKWNOWAIT	
8000ACC7	SCH$IOLOCKW	
8000ACCE	SCH$LOCKW	
8000ACE1	SCH$IOLOCKR	
8000ACE8	SCH$LOCKR	
8000AD45	SCH$RAVAIL	
8000AD55	SCH$IOUNLOCK	
8000AD5C	SCH$UNLOCK	
8000ADCF	SCH$POSTEF	
8000AEB8	EXE$CHKWAIT2	

Value	Symbols...
8000AF3C	EXE$PROCSTRT
8000AF4D	RM$DIRCACHE_BLKAST
8000AF63	EXE$SHMCEBDEL
8000AF99	EXE$CEBREFLCK
8000AFB9	EXE$DASSGN
8000B0C2	EXE$DELPRC
8000B114	EXE$CLREF
8000B124	SCH$GETEFC
8000B15F	SCH$CLREFR
8000B165	SCH$CLREF
8000B173	EXE$READEF
8000B197	EXE$SETEF
8000B1C4	EXE$SUSPND
8000B23A	EXE$RESUME
8000B24F	EXE$HIBER
8000B265	EXE$WAKE
8000B276	EXE$NAMPID
8000B35D	EXE$EPID_TO_PCB
8000B364	EXE$IPID_TO_PCB
8000B386	EXE$EPID_TO_IPID
8000B3B9	EXE$IPID_TO_EPID
8000B3E6	EXE$SETPRN
8000B480	PFM$MON
8000B512	EXE$WFLAND
8000B519	EXE$WFLOR
8000B523	EXE$WAITFR
8000B594	SCH$WAIT
8000B59E	SCH$WAITK
8000B5A9	SCH$WAITL
8000B5AA	SCH$WAITM
8000B5EE	EXE$ERAPAT
8000B5F6	EXE$MTACCESS
8000B720	LCK$SEARCHDLCK
8000B814	LCK$DLCKEXIT
8000B882	LCK$SRCH_RESDLCK
8000BA6F	LCK$BREAK_DEADLOCK
8000BB24	LCK$COMPAT_TBL
8000BB2A	LCK$SYNCCVT_TBL
8000BC33	LCK$LOCAL_CVT
8000BC8F	LCK$CVT_GRANTED
8000BCB8	LCK$CVTNOTQED
8000BEF6	LCK$LOCAL_LOCK
8000BFE2	LCK$SYNC_EXIT
8000BFF7	LCK$RET_VALBLK
8000C017	LCK$QUEUED_EXIT
8000C01A	LCK$NORET_VALBLK
8000C02C	LCK$NOT_QUEUED
8000C0BB	LCK$HASH_SEARCH
8000C103	LCK$SRCH_HSHTBL
8000C12A	LCK$GRANT_LOCK
8000C12F	LCK$GRANT_LOCK_ALT
8000C142	LCK$REGRANTLOCK

Value	Symbols...
8000C17C	LCK$GRANT_REM
8000C2CD	LCK$QUEUECVT
8000C2E7	LCK$QUEUEWAIT
8000C2EC	LCK$QUEUE_REM
8000C34C	LCK$QUEUE_BLOCKAST
8000C397	LCK$QUEUE_BLKAST
8000C3CE	LCK$COMP_CGMODE
8000C3F1	LCK$GRANTCVTS
8000C43A	LCK$GRANTWTRS
8000C4F9	LCK$CANCEL_CVT
8000C596	LCK$DEQLOCK
8000C729	LCK$CHECK_RSB
8000C742	LCK$DEALLOC_RSB
8000C7BB	LCK$EXTEND_IDTBL
8000C7C4	LCK$EXTEND_IDTBLW
8000C86C	IOC$GETBYTE
8000C87D	IOC$PUTBYTE
8000C88E	IOC$INITBUFWIND
8000C895	IOC$MOVFRUSER
8000C899	IOC$MOVFRUSER2
8000C8A6	IOC$MOVFRUSER1
8000C8AD	IOC$MOVTOUSER
8000C8C1	IOC$MOVTOUSER2
8000C8BE	IOC$MOVTOUSER1
8000C8C5	IOC$FILSPT
8000C8ED	ERL$DEVICERR
8000C8F1	ERL$DEVICTMO
8000C983	ERL$DEVICEATTN
8000C9FB	ERL$LOGSTATUS
8000CA76	ERL$LOGMESSAGE
8000CAD5	ERL$LOG_TMSCP
8000CAE0	ERL$LOG_DMSCP
8000CB1C	ERL$COLDSTART
8000CB21	ERL$WARMSTART
8000CB3E	ERL$ALLOCEMB
8000CBDE	ERL$GETFULLNAME
8000CC12	ERL$RELEASEMB
8000CC3C	ERL$WAKE
8000CC5D	IOC$CANCELIO
8000CC74	IOC$LAST_CHAN_AMBX
8000CC7B	IOC$LAST_CHAN
8000CCB1	IOC$DIAGBUFILL
8000CCD6	IOC$RELSCHAN
8000CCE0	IOC$RELCHAN
8000CD23	IOC$REQSCHANH
8000CD2D	IOC$REQSCHANL
8000CD37	IOC$REQPCHANH
8000CD40	IOC$REQPCHANL
8000CD6E	IOC$ALTREQCOM
8000CD99	IOC$REQCOM
8000CE28	IOC$MNTVER
8000CE31	IOC$INITIATE

Value	Symbols...
8000CE5E	IOC$REQDATAP
8000CE70	IOC$REQDATAPNW
8000CE7E	IOC$REQDATAPUDA
8000CEDE	IOC$RELDATAPUDA
8000CEE9	IOC$RELDATAP
8000CF4A	IOC$REQMAPUDA
8000CF5F	IOC$REQMAPREG
8000CF94	IOC$ALOUBAMAPN
8000CF9B	IOC$ALOUBAMAP
8000D005	IOC$ALOUBAMAPSP
8000D0A4	IOC$ALOUBMAPRMN
8000D0AB	IOC$ALCUBMAPRM
8000D155	IOC$RELMAPUDA
8000D170	IOC$RELMAPREG
8000D236	IOC$RETURN
8000D237	IOC$WFIKPCH
8000D259	IOC$WFIRLCH
8000D27D	IOC$ALLOSPT
8000D2A4	IOC$CVT_DEVNAM
8000D37D	IOC$BROADCAST
8000D3EE	IOC$SCAN_IODB
8000D433	IOC$SCAN_IODB_2P
8000D48D	IOC$CTRLINIT
8000D4C7	IOC$UNITINIT
8000D4FA	IOC$PARSDEVNAM
8000D60E	IOC$SEARCHINT
8000D702	IOC$SEARCHCONT
8000D711	IOC$TESTUNIT
8000D7D8	IOC$APPLYECC
8000D84C	IOC$CVTLOGPHY
8000D855	IOC$CVTLOGPHYU
8000D877	IOC$MAPVBLK
8000D92E	IOC$UPDATRANSP
8000D96A	IOC$SENSEDISK
8000D975	IOC$LOADMBAMAP
8000D9BF	IOC$LOADUBAMAPA
8000D9D6	IOC$LOADUBAMAP
8000DA49	IOC$PTETOPFN
8000DA5D	IOC$LOADUBAMAPN
8000DA9C	IOC$LUBAUDAMAP
8000DB00	IOC$CLONE_UCB
8000DB26	IOC$COPY_UCB
8000DBB4	IOC$LINK_UCB
8000DBE4	IOC$DELETE_UCB
8000DBF4	IOC$SEVER_UCB
8000DC16	IOC$FREE_UCB
8000DC4C	EXE$IPCONTROL
8000DE76	COM$DELATTNAST
8000DE7E	COM$DELATTNASTP
8000DEC5	COM$FLUSHATTNS
8000DEF7	COM$POST
8000DF05	COM$DRVDEALMEM

Value	Symbols...	
8000DF4B	COM$SETATTNAST	
8000DFB8	COM$DELCTRLAST	
8000DFC0	COM$DELCTRLASTP	
8000E05F	COM$FLUSHCTRLS	
8000E0AB	COM$SETCTRLAST	
8000E7B2	LKI$SEARCH_BLOCKING	
8000E853	LKI$SEARCH_BLOCKEDBY	
8000E95C	INI$BRK	
8000E95E	INI$WRITABLE	
8000E965	INI$RDONLY	
8000E9B2	PAT$A_NONPGD_CODE	PAT$GL_EXP_NPG1
8000EC00	PAT$A_NONPGD_CODE_END	
8000F800	EXE$UPCASE_DAT	
8000F900	EXE$LNM_SYNTAX_DAT	
8000FA0D	EXE$EXCPTNE	
8000FA30	EXE$CMODEXECX	
8000FA58	EXE$CMODEXEC	
8000FBD2	EXE$EXCPTN	
8000FBE0	EXE$CMODKRNLX	
8000FC08	EXE$CMODKRNL	
8000FD1A	EXE$FAILURE	
8000FD22	EXE$SUCCESS	
8000FDAE	EXE$LDB_SYNCH	
8000FDE0	SYS$GB_KRNLNARG	
8000FE37	SYS$GB_KMASK	
8000FE8E	IOC$DISMOUNT	
80010160	EXE$CHKDELACCES	EXE$CHKLOGACCES
80010172	EXE$CHKCREACCES	EXE$CHKPHYACCES
80010177	EXE$CHKRDACCES	
8001017C	EXE$CHKEXEACCES	
80010181	EXE$CHKWRTACCES	
800101CE	EXE$CLEANUP_ORB	
80010207	EXE$MAXACMODE	
80010216	EXE$PROBER_DSC	
8001021A	EXE$PROBEW_DSC	
8001025F	EXE$VAL_IDNAME	
800102A6	IOC$FFCHAN	
800102FA	IOC$SEARCHDEV	
800102F0	IOC$SEARCHALL	
800102F6	IOC$SEARCH	
80010338	IOC$TRANDEVNAM	
8001040D	IOC$LOCK_DEV	
800104BE	IOC$DALLOC_DEV	
800104E6	IOC$UNLOCK_DEV	
8001052A	IOC$UNLOCK	
8001053D	IOC$VERIFYCHAN	
80010574	IOC$CHKMBXQUOTA	
80010579	IOC$CHKUCBQUOTA	
800105A1	IOC$DEBIT_UCB	
800105C1	IOC$CREDIT_UCB	
80010605	IOC$CREATE_UCB	
80010622	EXE$ALOP1PROC	

Value	Symbols...
80010640	EXE$ALOP1IMAG
800106EA	EXE$DEAP1
80010746	MMG$IMGRESET
8001088C	MMG$DALCSTXSCN1
80010893	MMG$DALCSTXSCN
80010916	MMG$DALCSTX
8001092D	MMG$ALCSTX
80010956	MMG$ALCPHD
80010AF0	RM$RESET
80010A31	RM$SET
80010A65	EXE$ADJWSL
80010B42	MMG$WSPEAKCHK
80010B6D	EXE$ADJSTK
80010BDB	EXE$DACEFC
80010C02	EXE$ASCEFC
80010DF0	EXE$DLCEFC
80011111	EXE$ASSIGN
80011400	EXE$SETAST
80011422	EXE$DCLAST
80011485	EXE$BRKTHRU
80011ADC	EXE$BRDCST
80011BAD	EXE$CHKPRO
80011DB5	EXE$CHKPRO_INT
80012000	EXE$CHECKACL
80012072	EXE$SEARCH_RIGHT
800120E7	EXE$FINDACE
80012139	EXE$CHECKPROT_16
80012165	EXE$CHECKPROT
800121EE	EXE$CHECKACMODE
80012227	EXE$CHECKCLASS
8001229E	EXE$CHECK_BYPASS
800122C6	MMG$EXPREG
800122D4	EXE$EXPREG
80012353	MMG$TRY_ALL
800123A3	MMG$FAST_CREATE
80012445	MMG$CRETVA
8001244D	EXE$CRETVA
800124A5	MMG$IN_REGION
8001250E	EXE$CNTREG
8001255D	MMG$INADRINI
8001256C	MMG$RETADRINI
80012588	MMG$RETRANGE
8001259F	EXE$DELTVA
8001260D	MMG$CREDEL
800126DA	MMG$CREPAG
800128F5	EXE$CREPRC
800131AA	EXE$CRMPSC
800131B2	EXE$MGBLSC
8001321F	EXE$ASCTIM
800132C0	EXE$BINTIM
800134E4	EXE$NUMTIM
80013606	EXE$DCLCMH

Value	Symbols...
80013649	EXE$CANEXH
8001367B	EXE$DCLEXH
80013BE7	EXE$ALLOC
80013C8D	EXE$DALLOC
80013D27	MMG$VFYSECFLG
80013D9B	MMG$GSDMTXULK
80013DB4	MMG$DGBLSC1
80013DBB	EXE$DGBLSC
80013ECD	MMG$GSDSCN
80013FE7	MMG$DELGBLWCB
80014058	EXE$ENQ
8001415B	EXE$DEQ
800142F9	EXE$GETCHN
80014303	EXE$GETDEV
8001430D	EXE$GETDVI
80014312	EXE$GETPTI
80014345	EXE$ULWSET
8001434C	EXE$ULKPAG
8001435C	EXE$LKWSET
80014363	EXE$LCKPAG
800144D5	EXE$CRELOG
80014557	EXE$DELLOG
80014583	EXE$TRNLOG
800146EC	EXE$CRELNT
800146F1	EXE$CRELNM
800146F6	EXE$DELLNM
800146FB	EXE$TRNLNM
80014761	EXE$CREMBX
80014C99	EXE$DELMBX
80014CC8	EXE$PURGWS
80014D12	EXE$SETEXV
80014D55	EXE$SETRWM
80014D5C	EXE$SETSFM
80014D6A	EXE$SETSWM
80014D8B	EXE$SETPRA
80014DBF	EXE$POWERAST
80014E30	EXE$SETPRT
80014F98	EXE$GETQUI
80014FA0	EXE$SNDJBC
80015441	EXE$JBCRSP
80015533	EXE$SNDACC
80015546	EXE$SNDSMB
80015559	EXE$SNDOPR
8001581B	EXE$SENDMSG
8001587C	EXE$OPRSNDERL
80015882	EXE$NETSNDERL
80015888	EXE$SNDERR
800158D8	EXE$SETOPR
8001594A	EXE$UPDSEC
80015A5F	MMG$UPDSECAST
80015AD2	MMG$SET_BITMAP
80015ADB	MMG$CLR_BITMAP

Value	Symbols...
80015B3A	MMG$ALOSHMPAG
80015C46	MMG$ALOSHMGSD
80015CD1	MMG$FREEGSD
80015D3D	MMG$FIND1STGSD
80015D65	MMG$FINDSHB
80015D9A	MMG$GETGSNAM
80015E21	MMG$CEFTRNLOG
80015E2A	MMG$MBXTRNLOG
80015E33	MMG$GSDTRNLOG
80015F7A	MMG$WRITE_GSD
80015F83	MMG$READ_GSD
80016143	MMG$FINDGSNOTRN
80016183	MMG$UNIQUEGSD
800161E6	MMG$SHMTXLK
80016223	MMG$SHMTXULK
80016244	MMG$DELSHMGS
800162EC	EXE$ERAPAT_RTN
8001631D	EXE$MTACCESS_RTN
800163BA	NSA$EVENT_AUDIT
8001658B	NSA$ARGLST_IMGNAM
800165A7	MMG$ALC_PGFLVBN
800165E4	FIL$CVT_DTB
800165EB	FIL$CVT_OTB
800165F2	FIL$CVT_HTB
80016658	EXE$EXIT
800166D8	EXE$GETJPI
800166DD	EXE$GETLKI
800166E5	EXE$GETSYI
800166EA	EXE$GETTIM
80016717	EXE$EXCEPTABLE
8001676C	EXE$PUTMSG
80016987	EXE$RUNDWN
80016C06	EXE$RESETVEC
80016CA2	EXE$SETPFM
80016EA2	PFM$PURGE
80016F24	PFM$GETBUF
80016FF7	EXE$SETPRV
800170D8	EXE$SETSSF
800170F5	EXE$SETSTK
80017147	EXE$UNWIND
80017304	EXE$SETIME_INT
8001730E	EXE$SETIME
80017577	EXE$GRANTID
8001757E	EXE$REVOKID
800175D0	EXE$T_ID_UPCASE
800176D0	EXE$$ASCTOID
80017795	EXE$$FINISH_RDB
800177F5	EXE$$IDTOASC
800179AC	EXE$OPEN_RDB
80017B6A	EXE$CLOSE_RDB
80017BC2	EXE$SET_RDIPTR
80017C76	EXE$REFLECT

Value	Symbols...
80017DD1	EXE$SRCHANDLER
80017E1A	EXE$CONTSIGNAL
80017FC3	EXE$SIGTORET
80017FEA	EXE$EXPANDSTK
80018160	EXE$EXCMSG
800182F9	EXE$OPEN_MSG
8001848B	EXE$CLOSE_MSG
8001851C	PAT$A_PAGED_CODE
80018A33	LNM$CHECK_PROT
80018A94	LNM$DELBLK
80018AC1	LNM$DELETE_LNMB
80018B01	LNM$DELETE_TAB
80018B12	LNM$INIT_PROT
80018B5A	LNM$INSLOGTAB
80018C7E	LNM$SEARCHLOG
80018D0F	LNM$SEARCH_ONE
80018D86	LNM$FIRSTTAB
80018DA2	LNM$PRESEARCH
80018DCC	LNM$CONTSEARCH
80018E41	LNM$HASH
80018EC1	LNM$SETUP
80018ED1	LNM$TABLE
80018F52	LNM$PROBER
80018F75	LNM$LOCKR
80018F7D	LNM$LOCKW
80018F85	LNM$UNLOCK
8001A7F4	MMG$RET_BYT_QUOTA
8001B05D	EXE$DVI_FREEBLOCKS
8001B9BA	EXE$CHKIMAGNAME
8001C7D4	EXE$$IMGACT
8001CAB8	IMG$OPEN_IMAGE
8001CCE2	IMG$GET_HEADER
8001D244	IMG$ALLOCATE_ICB
8001D28B	IMG$DEALLOCATE_ICB
8001D454	IMG$DECODE_IHD
8001D5F1	IMG$GET_NEXT_ISD
8001D77C	IMG$DO_WORK_LIST
8001DEDF	EXE$IMGFIX
8001DF52	IMG$IS_IT_MAPPED
8001E062	IMG$PRVSHRIMG
8001E0CC	EXE$IMGSTA
8001E295	EXE$IMGPURMSG
8001E29E	EXE$IMGDELMSG
8001E39D	EXE$PRCPURMSG
8001E3A6	EXE$PRCDELMSG
8001E94E	PQL$AB_SYSPQL
8001E995	EXE$SWAPINIT
8001EB63	EXE$FAO
8001EB70	EXE$FAOL
8001F147	EXE$GETMSG
8001F46A	FIL$CVTFILNAM
8001F547	FIL$OPENFILE

Value	Symbols...		
8001F550	FIL$OPENFILE_1		
8001F6BD	FIL$CACHE_INIT		
8001F731	FIL$CACHE_TRUNC		
8001F810	FIL$MOUNT		
8001F895	FIL$FINDFILID		
8001FAF7	FIL$RDCHKFILHDR		
8001FBBB	FIL$WRITEVBN		
8001FBC2	FIL$READVBN		
8001FC55	FIL$STATBLK		
8001FC02	FIL$RDWRTLBN		
8001FD33	FIL$INIWCB		
8001FDE0	EXE$GQ_SYSDISK		
80020B2E	EXE$PROCIMGACT		
80020BF6	EXE$CLI_UTILSRV		
80020C00	EXE$EXIT_IMAGE		
80020C0D	EXE$CATCH_ALL		
80020D4D	EXE$IMGDMP_EXEC		
80020D53	EXE$IMGDMP_MERGE		
80020E9D	EXE$CRE_JGTABLE		
80021000	SWP$GL_SHELLBAS		
80021E00	MMG$AL_PGDCODEN	XDS$GT_WORD_PFN	
80021EF8	XDS$GT_LONG_PFN		
800220D8	XDELIBRK		
80022151	XDEL_LOADBASE		
8002217D	XDS$GL_XESTRING		
80022181	XDS$GL_XFSTRING		
80022988	XDELBPT		
80022A20	XDELTBIT		
80022AF6	LIB$INS_DECODE		
80023190	LIB$GB_OPINFO1		
80023990	LIB$GB_OPINFO2		
80024200	EXE$LINK_VEC		
80024400	BDL$GL_DISK_LOG		
8002467C	EXE$INIT		
800253AE	INI$ALLOC_CRB		
800255C7	INI$ALONONPAGED		
80025648	MMG$AL_FIXUPTBL		
800257D8	PAT$A_PFN_FIXUP		
800257E0	BOO$A_BOOPARAM	BOO$GL_DSKDRV	EXE$A_BOOPARAM
800257E4	BOO$GL_SYSLOA		
800257E8	BOO$GL_TRMDRV		
800257EC	BOO$GQ_INILOA		
800257F4	BOO$GL_NPAGEDYN		
800257F8	BOO$GL_SPLITADR		
800257FC	BOO$GL_IRPCNT		
80025800	BOO$GL_LRPSIZE		
80025804	BOO$GL_LRPMIN		
80025808	BOO$GL_LRPSPLIT		
8002580C	BOO$GL_LRPCNT		
80025810	BOO$GL_SRPSPLIT		
80025814	BOO$GL_SRPCNT		
80025818	BOO$GQ_FILCACHE		

Value Symbols...

```
80025820        BOO$GL_BOOTCB
80025824        BOO$GT_TOPSYS
8002582E        BOO$GB_SYSTEMID
80025834        BOO$GL_PRTDRV
80025838        BOO$GL_UCODE
8002583C        BOO$GL_SCSLOA
80025840        BOO$GL_CLSLOA
80025844        BOO$GL_ERAPATLOA
80025848        BOO$GL_CHKPRTLOA
8002584C        BOO$GL_MTACCESSLOA
80025850        BOO$GB_NODENAME
80025858        BOO$GL_VAXEMUL
8002585C        BOO$GL_FPEMUL
80025860        BOO$GL_DEVNAME
80025864        SYS$GT_ANNOUNCE
800258A4        IOC$INITDRV
800258AA        IOC$REINITDRV
8002594C        IOC$RELOC_DDT
80025A00        BUG$A_PAGED
80025F6C        EXE$OUTHEX
80025F86        EXE$OUTBLANK
80025F8B        EXE$OUTCHAR
80025FE6        EXE$OUTCRLF
80025FF0        EXE$OUTCSTRING
80025FF5        EXE$OUTZSTRING
80026004        BUG$T_MESSAGES
80028750        BUG$A_PAGEDEND
80028800        MMG$A_SYS_END
```

Key for special characters above:

*	- undefined
U	- Universal
R	- Relocatable
X	- External

```
                        +-----------------+
                        ! Image Synopsis !
                        +-----------------+

Virtual memory allocated:                00000000 000013FF 00001400 (5120. bytes, 10. pages)
Stack size:                                     0. pages
Image header virtual block limits:              1.        1. (    1. block)
Image binary virtual block limits:              2.       11. (   10. blocks)
Image name and identification:           CYDRIVER V6-000
Number of files:                               10.
Number of modules:                              9.
Number of program sections:                     5.
Number of global symbols:                    2581.
Number of image sections:                       1.
Image type:                              NOPIC, SHAREABLE, Global Section Match=EQUAL, Ident, Major=91, Minor=7664928
Map format:                              FULL in file DRB2:[SHULL.EVXCI.CYDRIVER]CYDRIVER.MAP;21
Estimated map length:                    341. blocks
                      +----------------------+
                      ! Link Run Statistics !
                      +----------------------+

Performance Indicators                   Page Faults   CPU Time        Elapsed Time
----------------------                   -----------   --------        ------------
    Command processing:                          143   00:00:01.11     00:00:01.25
    Pass 1:                                      905   00:00:08.47     00:00:12.76
    Allocation/Relocation:                        23   00:00:00.19     00:00:00.69
    Pass 2:                                       41   00:00:01.97     00:00:03.13
    Map data after object module synopsis:        53   00:00:12.33     00:00:13.52
    Symbol table output:                           7   00:00:00.17     00:00:00.67
Total run values:                               1172   00:00:24.24     00:00:32.02

Using a working set limited to 2048 pages and 1047 pages of data storage (excluding image)

Total number object records read (both passes):   728
    of which 46 were in libraries and 14 were DEBUG data records containing 328 bytes

Number of modules extracted explicitly              = 0
    with 1 extracted to resolve undefined symbols

0 library searches were for symbols not in the library searched

A total of 8 global symbol table records was written

LINK/MAP=CYDRIVER/FULL/SHARE=CYDRIVER/CONTIG/SYMBOL_TABLE=CYDRIVER CYTABLES,CYINIT,CYINPUT,CYCMD,CYMAINT,CYMISC,SYS$SYSTEM:SYS.STB,S
YS$SYSTEM:SYSDEF.STB/SEL,SYS$INPUT/OPTION
```

Table of contents

```
0000     1          .TITLE  CYMAC
0000     2          .IDENT  'V6-000'
0000     3
0000     4  ;*****************************************************************************
0000     5  ;
0000     6  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     9  ;
0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    15  ;
0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    18  ;
0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    21  ;
0000    22  ;*****************************************************************************
0000    23  ;
0000    24  ;+
0000    25  ;
0000    26  ; FACILITY:      VAX/VMS DIAGNOSTIC CLASS DRIVER
0000    27  ;
0000    28  ; ABSTRACT:      This module defines several data structures and
0000    29  ;                macros used by the CI responder class driver.
0000    30  ;
0000    31  ; AUTHOR:        Jim Klumpp  16-JUL-81
0000    32  ;
0000    33  ; MODIFIED BY:   Jim Klumpp   6-MAY-83
0000    34  ;
0000    35  ;       6-000    Dave Shull      07-July-1984
0000    36  ;                VMS V4 Modifications/Release
0000    37  ;
0000    38  ;       4-001    Dave Shull      6-June-1984
0000    39  ;                Added RB$L_DGBUF to RB Structure Definition.  This will be used
0000    40  ;                to save DG Buffer Address returned by ALLOC_DG_BUF.
0000    41  ;
0000    42  ;-
0000    43
0000    44
0000    45
```

```
0000     47          .SBTTL  $CYCDRPDEF,     CYDRIVER CDRP definition
0000     48 ;+
0000     49 ;
0000     50 ;  $CYCDRPDEF
0000     51 ;
0000     52 ;  The responder uses an extension of the standard VMS CDRP defined by
0000     53 ;  $CDRPDEF. Two different types of CDRP's are use by the responder.
0000     54 ;  Normal type CDRP's are used for most interfacing to SCS, and are
0000     55 ;  associated one for one with request blocks. Mapped buffer type CDRP's
0000     56 ;  actually consist of an IRP portion and a CDRP portion. They are used
0000     57 ;  for SCS MAP and buffer transfer calls.
0000     58 ;
0000     59 ;-
0000     60
0000     61          .MACRO  $CYCDRPDEF
0000     62          $DEFINI CDRP
0000     63
0000     64          ASSUME  CDRP$C_BT_LEN   EQ      64
0000     65          . = 64
0000     66
0000     67 $DEF     CDRP$B_FLAGS            .BLKB
0000     68                                  .BLKB   3
0000     69 $DEF     CDRP$T_CY_LBUFHNDL      .BLKB   12      ; Local buffer handle
0000     70 $DEF     CDRP$T_CY_RBUFHNDL      .BLKB   12      ; Remote buffer handle
0000     71 $DEF     CDRP$L_BUFADR           .BLKL           ; Mapped buffer address
0000     72 $DEF     CDRP$L_BUFLEN           .BLKL           ; Mapped buffer length
0000     73 $DEF     CDRP$L_MNT_NAM          .BLKL           ; Maintenance buffer name
0000     74 $DEF     CDRP$L_RB_FL            .BLKL           ; Request block flink
0000     75 $DEF     CDRP$L_RB_BL            .BLKL           ; Request block blink
0000     76 $DEF     CDRP$L_CB_FL            .BLKL           ; Connection block flink
0000     77 $DEF     CDRP$L_CB_BL            .BLKL           ; Connection block blink
0000     78 $DEF     CDRP$L_MAPQ_FL          .BLKL           ; Mapped buf queue flink
0000     79 $DEF     CDRP$L_MAPQ_BL          .BLKL           ; Mapped buf queue blink
0000     80 $DEF     CDRP$K_CY_LEN                           ; CYdriver CDRP length
0000     81 $DEF     CDRP$C_CY_LEN                           ; CYdriver CDRP length
0000     82
0000     83          $DEFEND CDRP                            ; End of CDRP additions
0000     84          .ENDM   $CYCDRPDEF
0000     85
0000     86
0000     87
```

```
                                        .SBTTL  $CBDEF,         Connection block definition
                                ;+
                                ;
                                ;  $CBDEF
                                ;
                                ;  Connection blocks are used by the responder to hold connection specific
                                ;  information, and are dynamically allocated from non-paged pool by
                                ;  the responder each time a connection is accepted. Connection blocks are
                                ;  queued to the connection list at the location detemined by the remote
                                ;  port over which the connection is made.
                                ;
                                ;  Responder connection blocks are deallocated when a disconnect request
                                ;  from another node is received, or when a connection with another node
                                ;  crashes.
                                ;
                                ;-

                                        .MACRO  $CBDEF
                                        $DEFINI CB

                                .=0                                             ; Responder specific structure

                        $DEF    CB$L_FLINK      .BLKL   1                       ; Connection block flink
                        $DEF    CB$L_BLINK      .BLKL   1                       ; Connection block blink
                        $DEF    CB$W_SIZE       .BLKW   1                       ; Size of structure
                        $DEF    CB$B_TYPE       .BLKB   1                       ; Type
                        $DEF    CB$B_STATUS     .BLKB   1                       ; CB status
                        $DEF    CB$L_CDT        .BLKL   1                       ; Address of CDT
                        $DEF    CB$L_PDT        .BLKL   1                       ; Address of PDT
                        $DEF    CB$Q_MAPQ       .BLKQ   1                       ; Mapped buffer queue
                        $DEF    CB$B_BUFMAP_CNT .BLKB   1                       ; Count of buffers mapped
                        $DEF    CB$Q_MNTMAPQ    .BLKQ   1                       ; Maintenance buffer queue
                        $DEF    CB$B_LPRTNUM    .BLKB   1                       ; Local port number
                        $DEF    CB$B_RPRTNUM    .BLKQ   1                       ; Remote port number
                        $DEF    CB$T_LPRTNAM    .BLKL   1                       ; Local port name
                        $DEF    CB$T_RPROCNAM   .BLKB   16                      ; Remote process name
                        $DEF    CB$L_RB         .BLKL   1                       ; Saved request block address
                        $DEF    CB$K_LENGTH                                     ; Length of this structure

                                _VIELD  -                                       ; Connect block status value
                                        CB,0,-                                  ;
                                        <-                                      ;
                                        <CONN,,M>,-                             ; Connect in progress
                                        <DISC,,M>,-                             ; Disconnect in progress
                                        >                                       ;

                                $DEFEND CB                                      ; End of CB definition
                                .ENDM   $CBDEF
```

```
0000    141          .SBTTL  $RBDEF,          Request block definition
0000    142 ;+
0000    143 ;
0000    144 ;  $RBDEF
0000    145 ;
0000    146 ;  CTP request blocks are used by the responder to hold CTP request specific
0000    147 ;  information, and are dynamically allocated from non-paged pool by
0000    148 ;  the responder each time a CTP request is received. Information is
0000    149 ;  copied from the CTP request into these blocks so that the message (or
0000    150 ;  datagram) buffer holding the request can be used to build the CTP
0000    151 ;  response.
0000    152 ;
0000    153 ;  CTP request blocks also contain execution list foreward and backward
0000    154 ;  pointers, which are used by the mainline routine to determine the order
0000    155 ;  of execution of CTP requests.
0000    156 ;
0000    157 ;  Responder connection blocks are deallocated when a CTP response is
0000    158 ;  returned to the controller, or when the connection over which the
0000    159 ;  request was received crashes.
0000    160 ;
0000    161 ;-
0000    162
0000    163          .MACRO  $RBDEF
0000    164          $DEFINI RB
0000    165
0000    166 .=0                                          ; Responder specific structure
0000    167
0000    168 $DEF     RB$L_FLINK      .BLKL               ; Execution list flink
0000    169 $DEF     RB$L_BLINK      .BLKL               ; Execution list blink
0000    170 $DEF     RB$W_SIZE       .BLKW               ; Size of structure
0000    171 $DEF     RB$B_TYPE       .BLKB               ; Type
0000    172 $DEF     RB$B_STATUS     .BLKB               ; RB status
0000    173 $DEF     RB$L_CB         .BLKL               ; Address of CB
0000    174 $DEF     RB$L_CTPBUF     .BLKL               ; Address of CTP request
0000    175 $DEF     RB$L_CDT        .BLKL               ; Address of CDT
0000    176 $DEF     RB$L_PDT        .BLKL               ; Address of PDT
0000    177 $DEF     RB$L_CDRP       .BLKL               ; Address of CDRP
0000    178 $DEF     RB$W_DELAY      .BLKW               ; Delay count of CTP request
0000    179 $DEF     RB$W_REPEAT     .BLKW               ; Repeat count of CTP request
0000    180 $DEF     RB$W_ACTCNT     .BLKW               ; Actual messages/dg sent
0000    181 $DEF     RB$L_DGBUF      .BLKL               ; Address of DG Buffer          [4-001]
0000    182 $DEF     RB$K_LENGTH                         ; Length of this structure
0000    183
0000    184          _VIELD  -                           ; Request block status values
0000    185                  RB,0,-                      ;
0000    186                  <-                          ;
0000    187                  <CONN,,M>,-                 ; These two flags are used when a
0000    188                  <CANCEL,,M>,-               ; connect command is being processed.
0000    189                  <FPD,,M>,-                  ; First part done (for set/maint cmd)
0000    190                  >
0000    191
0000    192          $DEFEND RB                              ; End of RB definition
0000    193          .ENDM   $RBDEF
0000    194
0000    195
```

```
                        0000    197          .SBTTL  $TQBDEF,        Timer queue block definition
                        0000    198 ;+
                        0000    199 ;
                        0000    200 ;  $TQBDEF
                        0000    201 ;
                        0000    202 ;  Timer queue blocks are used by the responder to keep track of delay times
                        0000    203 ;  and packet timeouts. These blocks are linked onto the timer queue, with
                        0000    204 ;  the block due to expire soonest at the head of the queue.
                        0000    205 ;
                        0000    206 ;-
                        0000    207
                        0000    208          .MACRO  $TQBDEF
                        0000    209          $DEFINI TQB
                        0000    210
                        0000    211 $DEF     TQB$L_FLINK     .BLKL                   ; TQB flink
                        0000    212 $DEF     TQB$L_BLINK     .BLKL                   ; TQB blink
                        0000    213 $DEF     TQB$W_SIZE      .BLKW                   ; Size
                        0000    214 $DEF     TQB$B_TYPE      .BLKB                   ; Type
                        0000    215 $DEF     TQB$B_STATUS    .BLKB                   ; Status
                        0000    216 $DEF     TQB$L_DUETIM    .BLKL                   ; Due time
                        0000    217 $DEF     TQB$L_RB        .BLKL                   ; Request block address
                        0000    218 $DEF     TQB$K_LENGTH                            ; Length
                        0000    219
                        0000    220          $VIELD  -                               ; Define TQB status values
                        0000    221                  TQB,0,-                         ;
                        0000    222                  <-                              ;
                        0000    223                  <STS_DELAY,,M>,-                ; Delay TQB
                        0000    224                  <STS_TMOUT,,M>,-                ; Timeout TQB
                        0000    225                  >
                        0000    226
                        0000    227          $DEFEND TQB
                        0000    228          .ENDM
                        0000    229
                        0000    230
                        0000    231
```

```
0000    233          .SBTTL  FUNCTION_SET,   Define responder function mask
0000    234 ;+
0000    235 ;
0000    236 ;  FUNCTION_SET
0000    237 ;
0000    238 ;  Macro to define the responder's function mask. The symbol FUNCTION_MASK
0000    239 ;  is equated to the function mask. The maximum CTP command opcode is assumed
0000    240 ;  to be less than 32. If CTP expands beyond this value, this macro will have
0000    241 ;  to be modified appropriately.
0000    242 ;
0000    243 ;-
0000    244
0000    245          .MACRO  FUNCTION_SET,VECTOR
0000    246
0000    247                  ASSUME  CTP$MAXCMDOPC LT 32
0000    248
0000    249                  FUNCTION_MASK == 0
0000    250                  .IRP ENTRY,<VECTOR>
0000    251                  FUNCTION_SET1 ENTRY
0000    252                  .ENDR
0000    253
0000    254          .ENDM   FUNCTION_SET
0000    255
0000    256
0000    257          .MACRO  FUNCTION_SET1,OPCODE,FLAG
0000    258
0000    259                  .IF IDN FLAG,Y
0000    260                  FUNCTION_MASK == FUNCTION_MASK ! <1 @ OPCODE>
0000    261                  .ENDC
0000    262
0000    263          .ENDM   FUNCTION_SET1
0000    264
0000    265
0000    266
0000    267
```

```
                        .SBTTL  CLEAR_ACTIVITY, Clear activity state
                ;+
                ;
                ;  CLEAR_ACTIVITY
                ;
                ;  Macro to clear all status in the responder that reflects that activity
                ;  is taking place.
                ;
                ;-

                .MACRO  CLEAR_ACTIVITY

                        CLRB    ACTIVITY_FLAG
                        CLRL    ACT_BUF_ADDR
                        CLRW    ACT_BUF_OFFSET
                        CLRB    ACT_BUF_CUR
                        CLRB    ACT_BUF_TOT
                        CLRL    ACT_RB_ADDR

                .ENDM   CLEAR_ACTIVITY
```

```
0000    293          .SBTTL  CONDITIONAL BRANCH INSTRUCTION MACROS
0000    294
0000    295 ;+
0000    296 ;  CONDITIONAL BRANCH INSTRUCTIONS MACROS
0000    297 ;
0000    298 ;  The following macros are used when a conditional branch instruction needs
0000    299 ;  a word, rather than byte, displacement. They don't save any code; they just
0000    300 ;  make it easier to code.
0000    301 ;-
0000    302
0000    303          .MACRO  BNEQW,DISPLACEMENT,?SKIP
0000    304
0000    305                  BEQL    SKIP
0000    306                  BRW     DISPLACEMENT
0000    307 SKIP:
0000    308
0000    309          .ENDM
0000    310
0000    311          .MACRO  BNEQUW,DISPLACEMENT,?SKIP
0000    312
0000    313                  BEQLU   SKIP
0000    314                  BRW     DISPLACEMENT
0000    315 SKIP:
0000    316
0000    317          .ENDM
0000    318
0000    319          .MACRO  BEQLW,DISPLACEMENT,?SKIP
0000    320
0000    321                  BNEQ    SKIP
0000    322                  BRW     DISPLACEMENT
0000    323 SKIP:
0000    324
0000    325          .ENDM
0000    326
0000    327          .MACRO  BEQLUW,DISPLACEMENT,?SKIP
0000    328
0000    329                  BNEQU   SKIP
0000    330                  BRW     DISPLACEMENT
0000    331 SKIP:
0000    332
0000    333          .ENDM
0000    334
0000    335          .MACRO  BGTRW,DISPLACEMENT,?SKIP
0000    336
0000    337                  BLEQ    SKIP
0000    338                  BRW     DISPLACEMENT
0000    339 SKIP:
0000    340
0000    341          .ENDM
0000    342
0000    343          .MACRO  BLEQW,DISPLACEMENT,?SKIP
0000    344
0000    345                  BGTR    SKIP
0000    346                  BRW     DISPLACEMENT
0000    347 SKIP:
0000    348
0000    349          .ENDM
```

```
                        0000   350
                        0000   351          .MACRO  BGEQW,DISPLACEMENT,?SKIP
                        0000   352
                        0000   353                  BLSS    SKIP
                        0000   354                  BRW     DISPLACEMENT
                        0000   355 SKIP:
                        0000   356
                        0000   357          .ENDM
                        0000   358
                        0000   359          .MACRO  BLSSW,DISPLACEMENT,?SKIP
                        0000   360
                        0000   361                  BGEQ    SKIP
                        0000   362                  BRW     DISPLACEMENT
                        0000   363 SKIP:
                        0000   364
                        0000   365          .ENDM
                        0000   366
                        0000   367          .MACRO  BGTRUW,DISPLACEMENT,?SKIP
                        0000   368
                        0000   369                  BLEQU   SKIP
                        0000   370                  BRW     DISPLACEMENT
                        0000   371 SKIP:
                        0000   372
                        0000   373          .ENDM
                        0000   374
                        0000   375          .MACRO  BLEQUW,DISPLACEMENT,?SKIP
                        0000   376
                        0000   377                  BGTRU   SKIP
                        0000   378                  BRW     DISPLACEMENT
                        0000   379 SKIP:
                        0000   380
                        0000   381          .ENDM
                        0000   382
                        0000   383          .MACRO  BVCW,DISPLACEMENT,?SKIP
                        0000   384
                        0000   385                  BVS     SKIP
                        0000   386                  BRW     DISPLACEMENT
                        0000   387 SKIP:
                        0000   388
                        0000   389          .ENDM
                        0000   390
                        0000   391          .MACRO  BVSW,DISPLACEMENT,?SKIP
                        0000   392
                        0000   393                  BVC     SKIP
                        0000   394                  BRW     DISPLACEMENT
                        0000   395 SKIP:
                        0000   396
                        0000   397          .ENDM
                        0000   398
                        0000   399          .MACRO  BGEQUW,DISPLACEMENT,?SKIP
                        0000   400
                        0000   401                  BLSSU   SKIP
                        0000   402                  BRW     DISPLACEMENT
                        0000   403 SKIP:
                        0000   404
                        0000   405          .ENDM
                        0000   406
```

```
0000    407          .MACRO  BCCW,DISPLACEMENT,?SKIP
0000    408
0000    409                  BCS     SKIP
0000    410                  BRW     DISPLACEMENT
0000    411 SKIP:
0000    412
0000    413          .ENDM
0000    414
0000    415          .MACRO  BCSW,DISPLACEMENT,?SKIP
0000    416
0000    417                  BCC     SKIP
0000    418                  BRW     DISPLACEMENT
0000    419 SKIP:
0000    420
0000    421          .ENDM
0000    422
0000    423          .MACRO  BLSSUW,DISPLACEMENT,?SKIP
0000    424
0000    425                  BGEQU   SKIP
0000    426                  BRW     DISPLACEMENT
0000    427 SKIP:
0000    428
0000    429          .ENDM
0000    430
0000    431          .MACRO  BLBSW,REGISTER,DISPLACEMENT,?SKIP
0000    432
0000    433                  BLBC    REGISTER,SKIP
0000    434                  BRW     DISPLACEMENT
0000    435 SKIP:
0000    436
0000    437          .ENDM
0000    438
0000    439          .MACRO  BLBCW,REGISTER,DISPLACEMENT,?SKIP
0000    440
0000    441                  BLBS    REGISTER,SKIP
0000    442                  BRW     DISPLACEMENT
0000    443 SKIP:
0000    444
0000    445          .ENDM
0000    446
0000    447          .MACRO BBCW,POS,BASE,DISPLACEMENT,?SKIP
0000    448
0000    449                  BBS     POS,BASE,SKIP
0000    450                  BRW     DISPLACEMENT
0000    451 SKIP:
0000    452
0000    453          .ENDM   BBCW
0000    454
0000    455          .MACRO BBSSW,POS,BASE,DISPLACEMENT,?SKIP
0000    456
0000    457                  BBCS    POS,BASE,SKIP
0000    458                  BRW     DISPLACEMENT
0000    459 SKIP:
0000    460
0000    461          .ENDM   BBSSW
0000    462
0000    463          .MACRO BBSW,POS,BASE,DISPLACEMENT,?SKIP
```

```
                0000    464
                0000    465                     BBC     POS,BASE,SKIP
                0000    466                     BRW     DISPLACEMENT
                0000    467 SKIP:
                0000    468
                0000    469             .ENDM   BBSW
                0000    470
                0000    471
                0000    472             .END
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	104	00:00:00.42	00:00:02.84
Command processing	55	00:00:00.19	00:00:01.67
Pass 1	111	00:00:00.95	00:00:03.66
Symbol table sort	0	00:00:00.01	00:00:00.00
Pass 2	77	00:00:00.67	00:00:01.51
Symbol table output	0	00:00:00.00	00:00:00.00
Psect synopsis output	3	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	353	00:00:02.26	00:00:09.72

The working set limit was 900 pages.
2982 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 0 pages of symbol table space allocated to hold 0 non-local and 0 local symbols.
472 source lines were read in Pass 1, producing 0 object records in Pass 2.
34 pages of virtual memory were used to define 28 macros.

Macro library statistics

Macro library name	Macros defined
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/NOOBJ/LIST CYMAC

Table of contents

```
0000     1          .TITLE  CYCMD
0000     2          .IDENT  'V6-000'
0000     3  ;*****************************************************************************
0000     4  ;
0000     5  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     6  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     7  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     8  ;
0000     9  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    10  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    11  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    12  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    13  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    14  ;
0000    15  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    16  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    17  ;
0000    18  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    19  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    20  ;
0000    21  ;*****************************************************************************
0000    22  ;
0000    23  ;+
0000    24  ;
0000    25  ; FACILITY:      VAX/VMS DIAGNOSTIC CLASS DRIVER
0000    26  ;
0000    27  ; ABSTRACT:      This module contains the mainline and command execution
0000    28  ;                routines for the CI responder class driver.
0000    29  ;
0000    30  ; AUTHOR:        Jim Klumpp  16-JUL-81
0000    31  ;
0000    32  ; MODIFIED BY:   Jim Klumpp   6-MAY-83
0000    33  ;
0000    34  ;       6-000    Dave Shull      07-July-1984
0000    35  ;                VMS V4 Modifications/Release
0000    36  ;
0000    37  ;       4-006    Dave Shull      14-June-1984
0000    38  ;                Modified routine MAINLINE and CY$FINISH to save/restore the
0000    39  ;                CDT address in R3.  Also, fixed a bug in CY$FINISH where it
0000    40  ;                was trying to restore RB address using R1 as base address,
0000    41  ;                should have been R3.
0000    42  ;
0000    43  ;       4-004    Dave Shull      06-June-1984
0000    44  ;                Modified routine CY$COUNTS to save DG Buffer address returned
0000    45  ;                by ALLOC_DG_BUF routine in Request Block and then restore that
0000    46  ;                address prior to calling DEALLOC_DG_BUF routine.
0000    47  ;
0000    48  ;       4-003    Dave Shull      04-June-1984
0000    49  ;                Modified routine CY$COUNTS to pass the Application Message
0000    50  ;                pointer in R2 versus the pointer to the Packet Envelope.  This
0000    51  ;                is   V4-FT1 Change ONLY.
0000    52  ;
0000    53  ;       4-002    Dave Shull      22-March-1984
0000    54  ;                Removed $PAPBDEF
0000    55  ;
0000    56  ;       4-001    Dave Shull      9-Feb-1984
0000    57  ;                Modified use of SCS$K_APPL_BASE to SCS$K_APPL_BASE-SCS$B_PPD
```

```
0000     58 ;                  as in VMS V4 all SCS$ symbols reference from the Application
0000     59 ;                  Data area versus Packet Pointer.
0000     60 ;
0000     61 ;--
0000     62
```

```
                     0000     64
                     0000     65
          00000001   0000     66 SOFTWARE_DEBUG=1                        ; Enable assembly of debug code
                     0000     67
                     0000     68         .DEFAULT DISPLACEMENT,WORD
                     0000     69         .ENABLE  SUPPRESSION
                     0000     70         .SBTTL   DEFINITIONS
                     0000     71
                     0000     72
                     0000     73 ;  Set PSECT to driver code:
                     0000     74
                     0000     75         .PSECT   $$$115_DRIVER,LONG
                     0000     76
                     0000     77
                     0000     78 ;  System definitions (LIB.MLB):
                     0000     79
                     0000     80         $CDRPDEF                        ; Define CDRP offsets
                     0000     81         $CDTDEF                         ; Define CDT offsets
                     0000     82         $CIBHANDEF                      ; Define CI buffer handles
                     0000     83         $CRBDEF                         ; Define CRB offsets
                     0000     84         $DYNDEF          GLOBAL         ; Define DYN offsets
                     0000     85         $IRPDEF                         ; Define IRP offsets
                     0000     86         $PBDEF                          ; Define PB offsets
                     0000     87         $PDTDEF                         ; Define PDT offsets
                     0000     88         $SBDEF                          ; Define SB offsets
                     0000     89         $SCSDEF                         ; Define SCS offsets
                     0000     90         $SSDEF                          ; Define SS offsets
                     0000     91         $SYSAPDEF                       ; Define SYSAP offsets
                     0000     92
                     0000     93
                     0000     94 ;  PADRIVER definitions (PALIB.MLB):
                     0000     95
                     0000     96         $PAPDTDEF                       ; Define PA PDT offsets
                     0000     97         $PAMAINTDEF                     ; Define PA maint definitions
                     0000     98         $PPDDEF                         ; Define PPD offsets
                     0000     99
                     0000    100
                     0000    101 ;  CYDRIVER definitions (CYLIB.MLB):
                     0000    102
                     0000    103         $CBDEF                          ; Define CB offsets
                     0000    104         $CTPDEF                         ; Define CTP symbols
                     0000    105         $CYCDRPDEF                      ; Define CYDRIVER CDRP extension
                     0000    106         $RBDEF                          ; Define RB offsets
                     0000    107         $TQBDEF                         ; Define TQB offsets
                     0000    108
                     0000    109
```

```
                                0000   111          .SBTTL  MAINLINE,       Responder mainline routine
                                0000   112 ;+
                                0000   113 ;
                                0000   114 ;  MAINLINE
                                0000   115 ;
                                0000   116 ;  This routine moniters the responder executicn list, looking for CTP
                                0000   117 ;  requests that require processing. If the execution list becomes empty,
                                0000   118 ;  the responder discards the current thread by RSBing to SCS. Otherwise,
                                0000   119 ;  the command at the top of the execution list is executed, its repeat count
                                0000   120 ;  decremented, and, if the repeat count reaches zero, the command is removed
                                0000   121 ;  from the execution list and a CTP response is returned to the remote
                                0000   122 ;  process that sent the CTP request.
                                0000   123 ;
                                0000   124 ;  In addition, each time through this routine, the responder forks at
                                0000   125 ;  IPL$_SCS, putting the current thread at the end of that fork queue, and
                                0000   126 ;  allowing more CTP requests to be incorporated into the command execution
                                0000   127 ;  list. This also allows the responder's connection error routine to be
                                0000   128 ;  entered, which can possibly drain the execution list.
                                0000   129 ;
                                0000   130 ;  INPUTS:
                                0000   131 ;
                                0000   132 ;       IPL                     - IPL$_SCS
                                0000   133 ;       R3                      - CDT address
                                0000   134 ;       R4                      - PDT address
                                0000   135 ;
                                0000   136 ;  OUTPUTS:
                                0000   137 ;
                                0000   138 ;       IPL,R6-R14 Preserved
                                0000   139 ;
                                0000   140 ;-
                                0000   141
                                0000   142 MAINLINE::
                                0000   143
0000'DF   0000'CF   D1          0000   144          CMPL    EXEC_LIST,@EXEC_LIST    ; Is execution list empty
                    1A          0007   145          BEQL    10$                     ; Yes, branch
          0000'CF   53   D0     0009   146          MOVL    R3,CY$L_SAVED_CDT       ; Save address of CDT           [4-006]
     55   0000'CF   D0          000E   147          MOVL    CY$L_SAVED_UCB,R5       ; Restore UCB address
                               0013   148          IOFORK                          ; Fork to allow more inputs
     53   0000'CF   D0          0019   149          MOVL    EXEC_LIST,R3            ; Get first RB in execution list
          53   63   D1          001E   150          CMPL    (R3),R3                 ; Execution list empty?
               10   12          0021   151          BNEQ    EXECUTE_CMD             ; No, branch
                    05          0023   152 10$:     RSB                             ; Return to SCS (discard this thread)
                               0024   153
                               0024   154 NEXT_ENTRY::
                               0024   155
     53   0000'DF   0F          0024   156          REMQUE  @EXEC_LIST,R3           ; Remove RB from top of execution list
                    01   1C     0029   157          BVC     10$                     ; Execution list NOT empty, branch
                    05          002B   158          RSB                             ; Discard this thread
0004'DF   63   0E          002C   159 10$:     INSQUE  (R3),@EXEC_LIST+4       ; Place RB at the bottom of list
               CD   11          0031   160          BRB     MAINLINE                ; Branch
                               0033   161
                               0033   162 EXECUTE_CMD:
                               0033   163
                               0033   164          ASSUME  RB$L_PDT+4 EQ RB$L_CDRP
                               0033   165
  52   10 A3   D0          0033   166          MOVL    RB$L_CTPBUF(R3),R2      ; Get CTP request address
  54   18 A3   7D          0037   167          MOVQ    RB$L_PDT(R3),R4         ; Get PDT, CDRP address
```

```
                20 A3   B5  003B   168          TSTW    RB$W_DELAY(R3)          ; Delay to perform?
                            003E   169          BNEQW   QUEUE_DELAY             ; Yes, branch
                            0043   170
                 FFBA'  30  0043   171          BSBW    CHECK_CTP_REQ           ; Check for valid CTP request fields
                            0046   172          BLBCW   R0,RETURN_CTP_RESPONSE  ; Error, branch
                            004C   173
                            004C   174          $DISPATCH -                             ; Dispatch off the CTP Opcode
                            004C   175                  CTP$OPCODE(R2),TYPE=B,-         ; to the CTP request specific
                            004C   176                  <-                              ; routine to execute the CTP
                            004C   177                  <CTP$FUNCTREQ,  CY$FUNCT>,-     ; request.
                            004C   178                  <CTP$BUFMAPREQ, CY$BUFMAP>,-
                            004C   179                  <CTP$BUFUNMREQ, CY$BUFUNM>,-
                            004C   180                  <CTP$MOVBUFREQ, CY$MOVBUF>,-
                            004C   181                  <CTP$GENMSGREQ, CY$GENMSG>,-
                            004C   182                  <CTP$GENDGRREQ, CY$GENDG>,-
                            004C   183                  <CTP$GENRSTREQ, CY$GENRST>,-
                            004C   184                  <CTP$GENSTRREQ, CY$GENSTR>,-
                            004C   185                  <CTP$NOACTREQ,  CY$NOACT>,-
                            004C   186                  <CTP$CONFIGREQ, CY$CONFIG>,-
                            004C   187                  <CTP$COUNTSREQ, CY$COUNTS>,-
                            004C   188                  <CTP$CONNECTREQ,CY$CONNECT>,-
                            004C   189                  <CTP$FINISHREQ, CY$FINISH>,-
                            004C   190                  <CTP$MBUFMAPREQ,CY$MNT_BUFMAP>,-
                            004C   191                  <CTP$MBUFUNMREQ,CY$MNT_BUFUNM>,-
                            004C   192                  <CTP$MOVMBUFREQ,CY$MNT_MOVBUF>,-
                            004C   193                  <CTP$MSTATEREQ, CY$MNT_STATE>,-
                            004C   194                  >
                            0072   195
                            0072   196
                            0072   197 RETURN_CTP_RESPONSE::
                            0072   198
                            0072   199          ASSUME  RB$L_PDT+4 EQ RB$L_CDRP
                            0072   200
          53    0000'CF     D0   0072   201          MOVL    EXEC_LIST,R3            ; Get current request block address
            52    10 A3     D0   0077   202          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP buffer address
            54    18 A3     7D   007B   203          MOVQ    RB$L_PDT(R3),R4         ; Restore PDT, CDRP address
              1C A5   52      D0   007F   204          MOVL    R2,CDRP$L_MSG_BUF(R5)   ; Copy CTP response address to CDRP
                 10 A3      D4   0083   205          CLRL    RB$L_CTPBUF(R3)         ; Show CTP buffer is gone in RB
            62    40 8F     80   0086   206          ADDB2   #64,CTP$OPCODE(R2)      ; Change request to response
          50    52    20    C3   008A   207          SUBL3   #SCS$K_APPL_BASE - SCS$B_PPD,R2,R0
                            008E   208                                          ; Back up to top of MSG/DG buffer[4-001]
                    3B      91   008E   209          CMPB    #DYN$C_CIDG,-           ; Was request sent as datagram?
                  0A A0             0090   210                  PPD$B_TYPE(R0)          ;
                    08      13   0092   211          BEQL    10$                     ; Yes, branch
                            0094   212
                            0094   213          RECYCL_MSG_BUF                  ; Recycle the message buffer
                            0097   214          SEND_MSG_BUF                    ; Send the CTP response
                    0E      11   009A   215          BRB     20$                     ; Branch around datagram code
                            009C   216
                    0E      A1   009C   217 10$:    ADDW3   #CTP$GENDATA,-          ; Move appl data length to R1
            51    0C A2             009E   218                  CTP$GENLENGTH(R2),R1    ; Get application data length
              51    51      3C   00A1   219          MOVZWL  R1,R1                   ; Make word into longword
                            00A4   220          SEND_DG_BUF -                   ; Return final CTP response
                            00A4   221                  FLAG=#SYSAP$C_DISPQ     ; Return datagram buffer to free queue
                            00AA   222
                 FF53'      30   00AA   223 20$:    BSBW    DEALLOC_RB              ; Clean up this request block
                 FF50       31   00AD   224          BRW     MAINLINE                ; Go execute next request in exec list
```

```
                                0080    225
                                0080    226
                                0080    227
                                0080    228
```

```
                        00B0   230          .SBTTL  CY$FUNCT,       Function set routine
                        00B0   231 ;+
                        00B0   232 ;
                        00B0   233 ;  CY$FUNCT
                        00B0   234 ;
                        00B0   235 ;  Routine to execute a CTP function set request. The function set response
                        00B0   236 ;  contains a 32 byte array, with each bit corresponding to a CTP request
                        00B0   237 ;  opcode. If the bit is set, then the corresponding CTP request IS implemented
                        00B0   238 ;  by the responder.
                        00B0   239 ;
                        00B0   240 ;  The maximum CTP command opcode is assumed to be less than 32. If CTP is
                        00B0   241 ;  expanded beyond 32 commands (unlikely) then the FUNCTION_SET macro and this
                        00B0   242 ;  routine will have to be changed appropriately.
                        00B0   243 ;
                        00B0   244 ;  INPUTS:
                        00B0   245 ;
                        00B0   246 ;       R2                      - CTP request address
                        00B0   247 ;       R3                      - RB address
                        00B0   248 ;       R4                      - PDT address
                        00B0   249 ;       R5                      - CDRP address
                        00B0   250 ;
                        00B0   251 ;  OUTPUTS:
                        00B0   252 ;
                        00B0   253 ;       R0-R5                   - Destroyed
                        00B0   254 ;       All other registers     - Preserved
                        00B0   255 ;
                        00B0   256 ;-
                        00B0   257
                        00B0   258 CY$FUNCT:
                        00B0   259
                        00B0   260 ;  The following macro call doesn't produce any code, it simply equates
                        00B0   261 ;  the symbol FUNCTION_MASK to the responders function mask. "Y" indicates
                        00B0   262 ;  the CTP command is implemented.
                        00B0   263
                        00B0   264          FUNCTION_SET -                  ; Define responder function set mask
                        00B0   265                  <-                      ;
                        00B0   266                  <CTP$FUNCTREQ,  Y>,-    ;   Function set request
                        00B0   267                  <CTP$BUFMAPREQ, Y>,-    ;   Buffer map request
                        00B0   268                  <CTP$BUFUNMREQ, Y>,-    ;   Buffer unmap request
                        00B0   269                  <CTP$MOVBUFREQ, Y>,-    ;   Move buffer request
                        00B0   270                  <CTP$GENMSGREQ, Y>,-    ;   Generate message request
                        00B0   271                  <CTP$GENDGRREQ, Y>,-    ;   Generate datagram request
                        00B0   272                  <CTP$GENRSTREQ, Y>,-    ;   Generate reset request
                        00B0   273                  <CTP$GENSTRREQ, Y>,-    ;   Generate start request
                        00B0   274                  <CTP$NOACTREQ,  Y>,-    ;   Stop poller request
                        00B0   275                  <CTP$CONFIGREQ, Y>,-    ;   Configuration data request
                        00B0   276                  <CTP$COUNTSREQ, Y>,-    ;   Counter read request
                        00B0   277                  <CTP$CONNECTREQ,Y>,-    ;   Third party connect request
                        00B0   278                  <CTP$FINISHREQ, Y>,-    ;   Listener disconnect request
                        00B0   279                  <CTP$MBUFMAPREQ,Y>,-    ;   Map maint buffer request
                        00B0   280                  <CTP$MBUFUNMREQ,Y>,-    ;   Unmap maint buffer request
                        00B0   281                  <CTP$MOVMBUFREQ,Y>,-    ;   Move maint buffer request
                        00B0   282                  <CTP$MSTATEREQ, Y>,-    ;   Maintenance state request
                        00B0   283                  >
                        00B0   284
                        00B0   285          ASSUME  CTP$MAXCMDOPC LT 32
                        00B0   286          ASSUME  CTP$SUCCESS EQ 0
```

```
                           00B0   287
        51    20     9A    00B0   288        MOVZBL  #32,R1                  ; Get length of function mask (bytes)
     52    06 A2     DE    00B3   289        MOVAL   CTP$FMASK(R2),R2        ; Get address of function msak field
            FF46'    30    00B7   290        BSBW    CLEAR_BUFFER            ; Clear out the entire function mask
     52    10 A3     D0    00BA   291        MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
   0001FFFF 8F       D0    00BE   292        MOVL    #FUNCTION_MASK,-        ; Fill in relavent part of function
           06 A2           00C4   293                CTP$FMASK(R2)           ; mask
           05 A2     94    00C6   294        CLRB    CTP$STATUS(R2)          ; Set success status
            FFA6     31    00C9   295        BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                           00CC   296
                           00CC   297
                           00CC   298
```

```
                                00CC   300          .SBTTL  CY$GENMSG,      Generate message routine
                                00CC   301 ;+
                                00CC   302 ;
                                00CC   303 ;  CY$GENMSG
                                00CC   304 ;
                                00CC   305 ;  Routine to execute a CTP generate message request. Each time through this
                                00CC   306 ;  routine, one CTP generate message response is returned to the controller.
                                00CC   307 ;  The total number of responses sent is one more than the repeat count in
                                00CC   308 ;  the request because the last response is not considered part of the action
                                00CC   309 ;  specified in the request, but rather as CTP protocol. The final CTP response
                                00CC   310 ;  is sent using the CTP request message buffer.
                                00CC   311 ;
                                00CC   312 ;  WARNING - Any suspendable calls to SCS require the stack to be clean, i.e.
                                00CC   313 ;  nothing can be saved on the stack when the call is made.
                                00CC   314 ;
                                00CC   315 ;  INPUTS:
                                00CC   316 ;
                                00CC   317 ;       R2                          - CTP request address
                                00CC   318 ;       R3                          - RB address
                                00CC   319 ;       R4                          - PDT address
                                00CC   320 ;       R5                          - CDRP address
                                00CC   321 ;
                                00CC   322 ;  OUTPUTS:
                                00CC   323 ;
                                00CC   324 ;       R0-R5                       - Destroyed
                                00CC   325 ;       All other registers         - Preserved
                                00CC   326 ;
                                00CC   327 ;-
                                00CC   328
                                00CC   329 CY$GENMSG:
                                00CC   330
          22 A3    B5           00CC   331          TSTW    RB$W_REPEAT(R3)         ; Last response to send?
             1C    13           00CF   332          BEQL    20$                     ; Yes, branch
                                00D1   333
                                00D1   334          ALLOC_MSG_BUF                   ; Allocate a mesage buffer
51  00000000'GF    3C           00D4   335          MOVZWL  G^SCS$GW_MAXMSG,R1      ; Get appl length of message buffer
           FF22'    30           00DB   336          BSBW    CLEAR_BUFFER            ; Clear the message buffer
           FF1F'    30           00DE   337          BSBW    BUILD_MSG               ; Build the CTP response
                                00E1   338          SEND_MSG_BUF                    ; Send the message
          24 A3    B6           00E4   339          INCW    RB$W_ACTCNT(R3)         ; Increment the actual count
          22 A3    B7           00E7   340          DECW    RB$W_REPEAT(R3)         ; Decrement CTP repeat count
           FF37    31           00EA   341          BRW     NEXT_ENTRY              ; Execute the next entry
                                00ED   342
           FF10'    30           00ED   343 20$:     BSBW    BUILD_MSG               ; Build the message
       62  40 8F    82           00F0   344          SUBB    #64,CTP$OPCODE(R2)      ; Opcode is changed to response later
           FF7B    31           00F4   345          BRW     RETURN_CTP_RESPONSE     ; Return the final CTP response
                                00F7   346
                                00F7   347
                                00F7   348
```

```
                                00F7   350          .SBTTL  CY$GENDG,       Generate datagram routine
                                00F7   351 ;+
                                00F7   352 ;
                                00F7   353 ;  CY$GENDG
                                00F7   354 ;
                                00F7   355 ;  Routine to execute a CTP generate datagram request. Each time through this
                                00F7   356 ;  routine, one datagram is returned to the controller. When the repeat count
                                00F7   357 ;  reaches zero, a final CTP generate datagram response is returned to
                                00F7   358 ;  the controller using the same service (message or datagram) used to send the
                                00F7   359 ;  CTP request. Thus, if the CTP request was received as a message, the FINAL
                                00F7   360 ;  response is returned as a message.
                                00F7   361 ;
                                00F7   362 ;  The total number of responses sent is one greater than the repeat count
                                00F7   363 ;  field specified in the request because the final CTP response is considered
                                00F7   364 ;  to be part of the CTP protocol, rather than part of the action specified in
                                00F7   365 ;  the request.
                                00F7   366 ;
                                00F7   367 ;  WARNING - Any suspendable calls to SCS require the stack to be clean, i.e.
                                00F7   368 ;  nothing can be saved on the stack when the call is made.
                                00F7   369 ;
                                00F7   370 ;  INPUTS:
                                00F7   371 ;
                                00F7   372 ;       R2                      - CTP request address
                                00F7   373 ;       R3                      - RB address
                                00F7   374 ;       R4                      - PDT address
                                00F7   375 ;       R5                      - CDRP address
                                00F7   376 ;
                                00F7   377 ;  OUTPUTS:
                                00F7   378 ;
                                00F7   379 ;       R0-R5                   - Destroyed
                                00F7   380 ;       All other registers     - Preserved
                                00F7   381 ;
                                00F7   382 ;-
                                00F7   383
                                00F7   384 CY$GENDG:
                                00F7   385
          22 A3    B5           00F7   386          TSTW    RB$W_REPEAT(R3)         ; Last response to send?
             2A    13           00FA   387          BEQL    20$                     ; Yes, branch
                                00FC   388
                                00FC   389          ALLOC_DG_BUF                    ; Allocate the datagram buffer
          2E 50    E9           00FF   390          BLBC    R0,30$                  ; On error, branch
51  00000000'GF    3C           0102   391          MOVZWL  G^SCS$GW_MAXDG,R1       ; Move length to R1
            FEF7'    30         0109   392          BSBW    CLEAR_BUFFER            ; Clear the datagram buffer
            FEF1'    30         010C   393          BSBW    BUILD_DG                ; Build the CTP response
              CE     A1         010F   394          ADDW3   #CTP$GENDATA,-          ; Move appl data length to R1
       51    0C A2               0111   395                  CTP$GENLENGTH(R2),R1
          51    51  3C           0114   396          MOVZWL  R1,R1                   ; Make word into longword
                                0117   397          SEND_DG_BUF -                   ; Send the datagram
                                0117   398                  FLAG=#SYSAP$C_DISPPO    ; Return datagram buffer to pool
          24 A3    B6           011D   399          INCW    RB$W_ACTCNT(R3)         ; Increment the actual count
          22 A3    B7           0120   400          DECW    RB$W_REPEAT(R3)         ; Decrement CTP repeat count
            FEFE   31           0123   401          BRW     NEXT_ENTRY              ; Execute the next entry
                                0126   402
            FED2'    30         0126   403 20$:     BSBW    BUILD_MSG               ; Build the MSG/DG buffer
       6E   40 AE    82         0129   404          SUBB    #64,CTP$OPCODE(R2)      ; Opcode is changed to response later
            FF42   31           012D   405          BRW     RETURN_CTP_RESPONSE     ; Return the final CTP response
                                0130   406
```

```
                52    10 A3   D0   0130    407 30$:    MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
                      FD 8F   90   0134    408         MOVB    #CTP$NORESOURCE,-       ; Set appropriate status
                      05 A2        0137    409                 CTP$STATUS(R2)
                      24 A3   B0   0139    410         MOVW    RB$W_ACTCNT(R3),-       ; Copy actual count to rsp
                      06 A2        013C    411                 CTP$ACTCOUNT(R2)
                       FF31   31   013E    412         BRW     RETURN_CTP_RESPONSE
                                   0141    413
                                   0141    414
                                   0141    415
```

```
                     0141   417          .SBTTL  CY$CONFIG,      Configuration routine
                     0141   418 ;+
                     0141   419 ;
                     0141   420 ;  CY$CONFIG
                     0141   421 ;
                     0141   422 ;  This routine gathers path status information about a specified port on the
                     0141   423 ;  CI. To do so, an SCS CONFIG_PTH call is issued. The CONFIG_PTH routine
                     0141   424 ;  expects to see the following input parameter block, which is built on the
                     0141   425 ;  stack by this routine:
                     0141   426 ;
                     0141   427 ;                                +-------------------------+
                     0141   428 ;                                |   Remote port number    |
                     0141   429 ;                                +-------------------------+
                     0141   430 ;                                |          MBZ            |
                     0141   431 ;                                +-------------------------+
                     0141   432 ;                                |   Local port name       |
                     0141   433 ;                                +-------------------------+
                     0141   434 ;
                     0141   435 ;  The CONFIG_PTH call returns the address of a path block, which contains
                     0141   436 ;  the path status information. The path status information is copied to the
                     0141   437 ;  CTP response, which is then returned to the controller.
                     0141   438 ;
                     0141   439 ;  INPUTS:
                     0141   440 ;
                     0141   441 ;       R2                      - CTP request address
                     0141   442 ;       R3                      - RB address
                     0141   443 ;       R4                      - PDT address
                     0141   444 ;       R5                      - CDRP address
                     0141   445 ;
                     0141   446 ;  OUTPUTS:
                     0141   447 ;
                     0141   448 ;       R0-R5                   - Destroyed
                     0141   449 ;       All other registers     - Preserved
                     0141   450 ;
                     0141   451 ;-
                     0141   452
                     0141   453 CY$CONFIG:
                     0141   454
50   14 A3   D0      0141   455          MOVL    RB$L_CDT(R3),R0             ; Get CDT address
50   1C A0   D0      0145   456          MOVL    CDT$L_PB(R0),R0             ; Get LOCAL path block address
     24 A0   DD      0149   457          PUSHL   PB$T_PORT_NAME(R0)           ; Put local port name on stack
        7E   D4      014C   458          CLRL    -(SP)                       ; Clear next longword
7E   09 A2   9A      014E   459          MOVZBL  CTP$OTHERNODE(R2),-(SP)     ; Put remote node number on stack
                     0152   460          CONFIG_PTH -                        ; Issue config path call
                     0152   461                  STAADR=(SP),-               ; Input array
                     0152   462                  OUTBUF=0                    ; No output needed, PB addr in R1
   5E    0C   C0      015D   463          ADDL    #12,SP                      ; Clean up stack
52   10 A3   D0      0160   464          MOVL    RB$L_CTPBUF(R3),R2          ; Restore CTP request address
                     0164   465          BLBCW   R0,100$                     ; Failure, branch
                     016A   466
                     016A   467          ASSUME  PB$B_P0_STS+1 EQ PB$B_P1_STS
                     016A   468          ASSUME  CTP$CFGP0STS+1 EQ CTP$CFGP1STS
                     016A   469          ASSUME  CTP$SUCCESS EQ 0
                     016A   470
     09 A1   B0      016A   471          MOVW    PB$B_P0_STS(R1),-           ; Copy path status fields from
     04 A2           016D   472                  CTP$CFGP0STS(R2)            ; path block to CTP response
     05 A2   94      016F   473          CLRB    CTP$STATUS(R2)              ; Set success status
```

```
                FEFD    31  0172    474             BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                                0175    475
                F8 8F   90  0175    476 100$:       MOVB    #CTP$NODEUNKNOWN,-      ; Set appropriate status
                05 A2       0178    477                     CTP$STATUS(R2)
                FEF5    31  017A    478             BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                                017D    479
                                017D    480
                                017D    481
```

```
                                017D    483          .SBTTL  CY$BUFMAP,      Buffer map routine
                                017D    484 ;+
                                017D    485 ;
                                017D    486 ;  CY$BUFMAP
                                017D    487 ;
                                017D    488 ;  Routine to execute a CTP buffer map request. The mapped buffer information
                                017D    489 ;  is stored in an IRP/CDRP block, which is linked to both the responder wide
                                017D    490 ;  mapped buffer queue and the connection block for the connection from which
                                017D    491 ;  the buffer map request originated. This allows the buffer to be unmapped if
                                017D    492 ;  the connection to the controller dies.
                                017D    493 ;
                                017D    494 ;  A limits is placed on the number of buffer allowed to be mapped for each
                                017D    495 ;  connection. If this limit is exceeded, a buffer is not mapped. Instead,
                                017D    496 ;  a CTP response is returned to the controller with the appropriate status.
                                017D    497 ;
                                017D    498 ;  To perform the actual buffer map, an SCS MAP call is issued. The map call
                                017D    499 ;  expects to see the following input array in the IRP/CDRP:
                                017D    500 ;
                                017D    501 ;                          +-----------------------+
                                017D    502 ;                          |        SVAPTE         |
                                017D    503 ;                          +-----------+-----------+
                                017D    504 ;                          | BCNT l.o. |    BOFF   |
                                017D    505 ;                          +-----------+-----------+
                                017D    506 ;                          |           | BCNT h.o. |
                                017D    507 ;                          +-----------+-----------+
                                017D    508 ;
                                017D    509 ;  The SVAPTE (system virtual address of the page table entry) is calculated
                                017D    510 ;  by extracting the virtual page number from the virtual address of the mapped
                                017D    511 ;  buffer, and using this value to index into the system page table. The BOFF
                                017D    512 ;  (byte offset) is merely the low order nine bits of the virtual address of
                                017D    513 ;  the buffer. The BCNT (byte count) is taken from the CTP request. In
                                017D    514 ;  addition, the access mode of the buffer is set to KERNEL.
                                017D    515 ;
                                017D    516 ;  The map call returns in the IRP/CDRP a local buffer handle, which contains,
                                017D    517 ;  among other things, a local buffer name. This name will eventually be used
                                017D    518 ;  by the port to access the mapped buffer. The buffer name is copied to the
                                017D    519 ;  CTP response, and the response is returned to the controller.
                                017D    520 ;
                                017D    521 ;  INPUTS:
                                017D    522 ;
                                017D    523 ;       R2                      - CTP request address
                                017D    524 ;       R3                      - RB address
                                017D    525 ;       R4                      - PDT address
                                017D    526 ;       R5                      - CDRP address
                                017D    527 ;
                                017D    528 ;  OUTPUTS:
                                017D    529 ;
                                017D    530 ;       R0-R5                   - Destroyed
                                017D    531 ;       All other registers     - Preserved
                                017D    532 ;
                                017D    533 ;-
                                017D    534
                                017D    535 CY$BUFMAP:
                                017D    536
                          50    017D    537          MOVL    RB$L_CB(R3),R0          ; Get connection block address
                          1C A0 91 0181    538          CMPB    CB$B_BUFMAP_CNT(R0),-   ; Exceeded buffer map limit?
                          00'8F    0184    539                  #MAX_BUF_MAP            ;
```

```
                                0186    540          BGEQUW  100$                    ; Yes, branch
                                0188    541
                     FE72'  30  0188    542          BSBW    ALLOC_IRP_CDRP          ; Allocate a IRP/CDRP combo
                                018E    543          BLBCW   R0,160$                 ; Failure, branch
                                0194    544
          51   0C A2   0C  C1  0194    545          ADDL3   #12,CTP$BUFLENGTH(R2),R1; Get length of buffer to allocate
                     FE64'  30  0199    546          BSBW    ALLOC_BUFFER            ; Allocate the buffer
                                019C    547          BLBCW   R0,150$                 ; Failure, branch
            5C A5   0C A2   DE  01A2    548          MOVAL   12(R2),CDRP$L_BUFADR(R5); Save address beyond VMS header
          60 A5   51   0C   C3  01A7    549          SUBL3   #12,R1,CDRP$L_BUFLEN(R5); Save length of buffer
               52   10 A3   D0  01AC    550          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
                     FE4D'  30  01B0    551          BSBW    BUILD_BUFFER            ; Build XFR buffer with specified data
                                01B3    552
                                01B3    553          ASSUME  CDRP$L_SVAPTE+4 EQ CDRP$W_BOFF
                                01B3    554          ASSUME  CDRP$W_BOFF+2 EQ CDRP$L_BCNT
                                01B3    555
               50   5C A5   D0  01B3    556          MOVL    CDRP$L_BUFADR(R5),R0    ; Get XFR buffer address
               51   D2 A5   DE  01B7    557          MOVAL   CDRP$L_BCNT(R5),R1      ; Point to end of array
               61   60 A5   D0  01BB    558          MOVL    CDRP$L_BUFLEN(R5),(R1)  ; Copy in byte count
          71   50   FE00 8F   AB  01BF    559          BICW3   #^XFFFFFE00,R0,-(R1)    ; Copy in byte offset
                         00'  EF  01C5    560          EXTZV   S^#VA$V_VPN,-           ; Get virtual page number
               50   50   00'      01C7    561                  S^#VA$S_VPN,R0,R0      ;
          52   00000000'9F   D0  01CA    562          MOVL    @#MMG$GL_SPTBASE,R2     ; Base address of SPT
                 71   6240   DE  01D1    563          MOVAL   (R2)[R0],-(R1)          ; Get SVA of PTE
                      AB A5   94  01D5    564          CLRB    CDRP$B_RMOD(R5)         ; Set access mode to kernel
                                01D8    565          MAP_IRP                         ; Map the read/write buffer
                                01DB    566
               52   10 A3   D0  01DB    567          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
                                01DF    568 ;        MOVL    CDRP$L_BUFLEN(R5),-     ; Copy actual buffer length
                                01DF    569 ;                CTP$BUFLENGTH(R2)       ; CTP response
               50   2C A5   D0  01DF    570          MOVL    CDRP$L_LBUFH_AD(R5),R0  ; Get local buffer handle address
                    04 A0   D0  01E3    571          MOVL    CIBHAN$L_BNAME(R0),-    ; Copy buffer name to CTP response
                    10 A2       01E6    572                  CTP$BUFLNAME(R2)        ;
                     FE87   31  01E8    573          BRW     RETURN_CTP_RESPONSE     ; Return the response
                                01EB    574
                                01EB    575
                    FA 8F   90  01EB    576 100$:    MOVB    #CTP$BUFLIMIT,-         ; Set appropriate status
                    05 A2       01EE    577                  CTP$STATUS(R2)
                     FE7F   31  01F0    578          BRW     RETURN_CTP_RESPONSE     ; Return the response
                                01F3    579
                                01F3    580          ASSUME  CDRP$L_CB_FL+8 EQ CDRP$L_MAPQ_FL
                                01F3    581
               50   0C A3   D0  01F3    582 150$:    MOVL    RB$L_CB(R3),R0          ; Get connection block address
                    1C A0   97  01F7    583          DECB    CB$B_BUFMAP_CNT(R0)     ; Decrement the buffer map count
               55   70 A5   0F  01FA    584          REMQUE  CDRP$L_CB_FL(R5),R5     ; Remove IRP/CDRP from CB
               55   08 A5   0F  01FE    585          REMQUE  8(R5),R5                ; and from mapped buffer queue
          50   FF28 C5   DE  0202    586          MOVAL   -<CDRP$L_MAPQ_FL+-      ; Back up to top of IRP
                               0207    587                  IRP$K_CDRP>(R5),R0      ;
          00000000'GF   16  0207    588          JSB     G^COM$DRVDEALMEM        ; Deallocate the IRP/DCRP
                               020D    589
               52   10 A3   D0  020D    590 160$:    MOVL    RB$L_CTPBUF(R3),R2      ; Get CTP request address
                    FD 8F   90  0211    591          MOVB    #CTP$NORESOURCE,-       ; Fill in status field
                    05 A2       0214    592                  CTP$STATUS(R2)          ;
                     FE59   31  0216    593          BRW     RETURN_CTP_RESPONSE     ; Return the response
                               0219    594
                               0219    595
                               0219    596
```

```
                                0219    598          .SBTTL  CY$MOVBUF,      Move buffer routine
                                0219    599 ;+
                                0219    600 ;
                                0219    601 ;  CY$MOVBUF
                                0219    602 ;
                                0219    603 ;  Routine to execute a CTP move buffer request. The request specifies the
                                0219    604 ;  direction of transfer, and the remote node to or from which the transfer
                                0219    605 ;  is to be made. Each time through this routine one buffer transfer is
                                0219    606 ;  performed and the repeat count is decremented. When the repeat count
                                0219    607 ;  reaches zero, a CTP buffer move response is returned to the controller.
                                0219    608 ;
                                0219    609 ;  Third party buffer transfers occur when the remote node in the CTP request
                                0219    610 ;  is not the same as the node in which the controller that sent the move
                                0219    611 ;  buffer request resides. In this case, a connection to that remote node must
                                0219    612 ;  already exist. If not, a CTP response is returned to the controller with
                                0219    613 ;  the appropriate error status.
                                0219    614 ;
                                0219    615 ;  A connection is required to the third party node so that the remote CONID
                                0219    616 ;  from the CDT can be copied to the local mapped buffer CDRP.
                                0219    617 ;
                                0219    618 ;  WARNING - Any suspendable calls to SCS require the stack to be clean, i.e.
                                0219    619 ;  nothing can be saved on the stack when the call is made.
                                0219    620 ;
                                0219    621 ;  INPUTS:
                                0219    622 ;
                                0219    623 ;       R2                      - CTP request address
                                0219    624 ;       R3                      - RB address
                                0219    625 ;       R4                      - PDT address
                                0219    626 ;       R5                      - CDRP address
                                0219    627 ;
                                0219    628 ;  OUTPUTS:
                                0219    629 ;
                                0219    630 ;       R0-R5                   - Destroyed
                                0219    631 ;       All other registers     - Preserved
                                0219    632 ;
                                0219    633 ;-
                                0219    634
                                0219    635 CY$MOVBUF:
                                0219    636
55    10 A2    DE               0219    637          MOVAL   CTP$BUFLNAME(R2),R5     ; Get address of local buffer name
      FDE0'    30               021D    638          BSBW    FIND_MAPPED_BUFFER      ; Find local mapped buffer CDRP
                                0220    639          BLBCW   R0,100$                 ; Failure, branch
                                0226    640
51    0C A3    D0               0226    641          MOVL    RB$L_CB(R3),R1          ; Get connection block address
      26 A1    91               022A    642          CMPB    CB$B_RPRTNUM(R1),-      ; Is this a second or third
      09 A2                     022D    643                  CTP$OTHERNODE(R2)       ; party transfer?
         24    13               022F    644          BEQL    20$                     ; Second party, branch
                                0231    645
7E    09 A2    9A               0231    646          MOVZBL  CTP$OTHERNODE(R2),-(SP) ; Push remote port number on stack
7E    2E A1    D0               0235    647          MOVL    CB$T_LPRTNAM(R1),-(SP)  ; Push local port name on stack
      FDC4'    30               0239    648          BSBW    CHECK_CONN_RSP          ; Get CB address for responder
   5E    08    C0               023C    649          ADDL    #8,SP                   ; Clean off stack
                                023F    650          BLBCW   R0,150$                 ; No connection, branch
                                0245    651          BBSW    #CB_V_CONN,-            ; Branch if CONNECT in progress for
                                0245    652                  CB$B_STATUS(R1),250$    ; this connection
                                024D    653          BBSW    #CB_V_DISC,-            ; Branch if DISCONNECT in progress for
                                024D    654                  CB$B_STATUS(R1),250$    ; this connection
```

```
                        0255   655
51   0C A1   D0         0255   656 20$:    MOVL    CB$L_CDT(R1),R1          ; Get CDT address
24 A5   51   D0         0259   657         MOVL    R1,CDRP$L_CDT(R5)        ; Fill in CDT address,
     0C A2   D0         025D   658         MOVL    CTP$BUFLENGTH(R2),-      ; buffer transfer length,
     3C A5              0260   659                 CDRP$L_XCT_LEN(R5)       ;
     14 A2   D0         0262   660         MOVL    CTP$BUFLOFSET(R2),-      ; local buffer offset,
     30 A5              0265   661                 CDRP$L_LBOFF(R5)         ;
     1C A2   D0         0267   662         MOVL    CTP$BUFROFSET(R2),-      ; remote buffer offset
     38 A5              026A   663                 CDRP$L_RBOFF(R5)
                        026C   664
50   2C A5   D0         026C   665         MOVL    CDRP$L_LBUFH_AD(R5),R0   ; Get local buffer handle addr
     14 A1   D0         0270   666         MOVL    CDT$L_RCONID(R1),-       ; Copy RCONID field from CDT
     08 A0              0273   667                 CIBHAN$L_RCONID(R0)
                        0275   668
                        0275   669         ASSUME  CIBHAN$L_BOFF+4 EQ CIBHAN$L_BNAME
                        0275   670         ASSUME  CIBHAN$L_BNAME+4 EQ CIBHAN$L_RCONID
                        0275   671
50   34 A5   D0         0275   672         MOVL    CDRP$L_RBUFH_AD(R5),R0   ; Get remote buffer handle addr
          80   D4       0279   673         CLRL    (R0)+                    ; Clear first longword
80   18 A2   D0         027B   674         MOVL    CTP$BUFRNAME(R2),(R0)+   ; Copy remote buffer name
60   18 A1   D0         027F   675         MOVL    CDT$L_LCONID(R1),(R0)    ; Copy LCONID field from CDT
                        0283   676
                        0283   677 ;       ASSUME  CTP$PS512 EQ 0
                        0283   678 ;
                        0283   679 ;       CLRB    CDRP$B_FLAGS(R5)         ; Assume 512 byte packets
                        0283   680 ;       TSTB    CTP$PKTSIZ(R2)           ; 512 byte packets?
                        0283   681 ;       BEQL    25$                      ; Yes, branch
                        0283   682 ;       MOVB    #^X80,CDRP$B_FLAGS(R5)   ; Set up for 576 byte packets
                        0283   683 ;25$:
                        0283   684
                        0283   685         ASSUME  CTP$MOVEFROM EQ 0
                        0283   686
   08 A2   95           0283   687         TSTB    CTP$MOVETYPE(R2)         ; Is this a buffer read request?
      11   12           0286   688         BNEQ    30$                      ; No, buffer write, branch
                        0288   689         REQUEST_DATA                     ; Issue buffer read
      0F   11           0297   690         BRB     40$                      ; Skip buffer write code
                        0299   691 30$:    SEND_DATA                        ; Issue buffer write
                        02A8   692
                        02A8   693 40$:    BLBCW   R0,200                   ; Failure, branch
   24 A3   B6           02AE   694         INCW    RB$W_ACTCNT(R3)          ; Increment the actual count
   22 A3   B7           02B1   695         DECW    RB$W_REPEAT(R3)          ; Decrement the repeat count
                        02B4   696         BNEQW   NEXT_ENTRY               ; Request not complete, branch
                        02B9   697
                        02B9   698         ASSUME  CTP$SUCCESS EQ 0
                        02B9   699
52 10 A3   D0           02B9   700         MOVL    RB$L_CTPBUF(R3),R2       ; Restore CTP request address
   05 A2   94           02BD   701         CLRB    CTP$STATUS(R2)           ; Set success status
   24 A3   B0           02C0   702         MOVW    RB$W_ACTCNT(R3),-        ; Copy actual count to response
   06 A2                02C3   703                 CTP$ACTCOUNT(R2)
    FDAA   31           02C5   704         BRW     RETURN_CTP_RESPONSE      ; Return the response
                        02C8   705
                        02C8   706
   F9 8F   90           02C8   707 100$:   MOVB    #CTP$NOBUFMAPPED,-       ; Set appropriate status
   05 A2                02CB   708                 CTP$STATUS(R2)           ;
    FDA2   31           02CD   709         BRW     RETURN_CTP_RESPONSE      ; Return the response and clean
                        02D0   710                                          ; up its associated resources
                        02D0   711
```

```
                F6 8F    90  02D0    712 150$:   MOVB    #CTP$NOCONNECT,-        ; Set appropriate status
                05 A2        02D3    713                 CTP$STATUS(R2)          ;
                 FD9A    31  02D5    714         BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                             02D8    715
                FB 8F    90  02D8    716 200$:   MOVB    #CTP$FAILURE,-          ; Set appropriate status
                05 A2        02DB    717                 CTP$STATUS(R2)          ;
                 FD92    31  02DD    718         BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                             02E0    719
                F5 8F    90  02E0    720 250$:   MOVB    #CTP$CONNTRANS,-        ; Show connection is in state
                05 A2        02E3    721                 CTP$STATUS(R2)          ; of trasnition
                 FD8A    31  02E5    722         BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                             02E8    723
                             02E8    724
```

```
                                02E8   726          .SBTTL  CY$BUFUNM,      Buffer unmap routine
                                02E8   727 ;+
                                02E8   728 ;
                                02E8   729 ;  CY$BUFUNM
                                02E8   730 ;
                                02E8   731 ;  Routine to execute a CTP buffer unmap request. The name of the buffer,
                                02E8   732 ;  supplied in the CTP request is used to look through the buffer map list to
                                02E8   733 ;  find the specified buffer. The buffer is then unmapped and deallocated, and
                                02E8   734 ;  a CTP response is reuturned to the controller.
                                02E8   735 ;
                                02E8   736 ;  If the buffer can't be found on the mapped buffer list the CTP response
                                02E8   737 ;  is returned to the controller with NOBUFMAPPED status.
                                02E8   738 ;
                                02E8   739 ;  INPUTS:
                                02E8   740 ;
                                02E8   741 ;       R2                      - CTP request address
                                02E8   742 ;       R3                      - RB address
                                02E8   743 ;       R4                      - PDT address
                                02E8   744 ;       R5                      - CDRP address
                                02E8   745 ;
                                02E8   746 ;  OUTPUTS:
                                02E8   747 ;
                                02E8   748 ;       R0-R5                   - Destroyed
                                02E8   749 ;       All other registers     - Preserved
                                02E8   750 ;
                                02E8   751 ;-
                                02E8   752
                                02E8   753 CY$BUFUNM:
                                02E8   754
    55    10 A2    DE           02E8   755          MOVAL   CTP$BUFLNAME(R2),R5     ; Get address of buffer name
          FD11'    30           02EC   756          BSBW    FIND_MAPPED_BUFFER      ; Find the mapped buffer
          2E 50    E9           02EF   757          BLBC    R0,100$                 ; Buffer can't be found, branch
    50    0C A3    D0           02F2   758          MOVL    RB$L_CB(R3),R0          ; Get connection block address
          1C A0    97           02F6   759          DECB    CB$B_BUFMAP_CNT(R0)     ; One less buffer mapped
                                02F9   760
                                02F9   761          ASSUME  CDRP$L_CB_FL+8 EQ CDRP$L_MAPQ_FL
                                02F9   762
                                02F9   763          UNMAP                           ; Unmap the buffer
    55    70 A5    0F           02FC   764          REMQUE  CDRP$L_CB_FL(R5),R5     ; Remove IRP/CDRP from CB
    55    08 A5    0F           0300   765          REMQUE  8(R5),R5                ; and from mapped buffer queue
    55    88 A5    DE           0304   766          MOVAL   -CDRP$L_MAPQ_FL(R5),R5  ; Back up to base of CDRP
50  50    5C A5    C3           0308   767          SUBL3   #12,CDRP$L_BUFADR(R5),R0; Get unmapped buffer address
    00000000'GF    16           030D   768          JSB     G^COM$DRVDEALMEM        ; Deallocate it
    50    A0 A5    DE           0313   769          MOVAL   -IRP$K_CDRP(R5),R0      ; Back up to base of IRP/CDRP
    00000000'GF    16           0317   770          JSB     G^COM$DRVDEALMEM        ; Deallocate it
                   31           031D   771          BRW     RETURN_CTP_RESPONSE     ; Return the response
                                0320   772
          F9 8F    90           0320   773 100$:    MOVB    #CTP$NOBUFMAPPED,-      ; Set appropriate status
          05 A2                 0323   774          CTP$STATUS(R2)          ;
           FD4A    31           0325   775          BRW     RETURN_CTP_RESPONSE     ; Return the response
                                0328   776
                                0328   777
                                0328   778
```

```
                        0328   780          .SBTTL  CY$GENRST,      Generate reset routine
                        0328   781 ;+
                        0328   782 ;
                        0328   783 ;  CY$GENRST
                        0328   784 ;
                        0328   785 ;  Routine to execute a CTP generate reset request. Each time through this
                        0328   786 ;  routine, one reset is sent to the specified remote port, and the repeat
                        0328   787 ;  count is decremented. When the repeat count reaches zero, a CTP response
                        0328   788 ;  is returned to the controller.
                        0328   789 ;
                        0328   790 ;  Reset packets have a forced/conditional flag, which is passed from the
                        0328   791 ;  CTP request to SCS.
                        0328   792 ;
                        0328   793 ;  WARNING - Sending resets can be extremely hazardous to the well being of
                        0328   794 ;  nodes on the CI.
                        0328   795 ;
                        0328   796 ;  INPUTS:
                        0328   797 ;
                        0328   798 ;       R2                      - CTP request address
                        0328   799 ;       R3                      - RB address
                        0328   800 ;       R4                      - PDT address
                        0328   801 ;       R5                      - CDRP address
                        0328   802 ;
                        0328   803 ;  OUTPUTS:
                        0328   804 ;
                        0328   805 ;       R0-R5                   - Destroyed
                        0328   806 ;       All other registers     - Preserved
                        0328   807 ;
                        0328   808 ;-
                        0328   809
                        0328   810 CY$GENRST:
                        0328   811
                        0328   812          ASSUME  CTP$UNCONDRST EQ 0
                        0328   813          ASSUME  CTP$CONDRST EQ 1
                        0328   814
08 A3   01      8C      0328   815          XORB2   #^X01,CTP$EXTEND(R3)     ; Change polarity of force flag
                        032C   816          MRESET  -                       ; Send a reset packet
                        032C   817                  RSTADR=CTP$OTHERNODE(R2),- ; Remote station address
                        032C   818                  FLAG=CTP$EXTEND(R2)     ; Forced/Conditional flag
    24 A3       B6      0337   819          INCW    RB$W_ACTCNT(R3)         ; Increment the actual count
    22 A3       B7      033A   820          DECW    RB$W_REPEAT(R3)         ; Decrement the repeat count
                        033D   821          BNEQW   NEXT_ENTRY              ; Request not done, branch
                        0342   822
                        0342   823          ASSUME  CTP$SUCCESS EQ 0
                        0342   824
52  10 A3       D0      0342   825          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
    05 A2       94      0346   826          CLRB    CTP$STATUS(R2)          ; Set success status
     FD26       31      0349   827          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                        034C   828
                        034C   829
                        034C   830
```

```
                     034C   832          .SBTTL  CY$GENSTR,      Generate start routine
                     034C   833 ;+
                     034C   834 ;
                     034C   835 ;  CY$GENSTR
                     034C   836 ;
                     034C   837 ;  Routine to execute a CTP generate start request. Each time through this
                     034C   838 ;  routine, one start packet is sent to the specified remote port, and the
                     034C   839 ;  repeat count is decremented. When the repeat count reaches zero, a CTP
                     034C   840 ;  response is returned to the controller.
                     034C   841 ;
                     034C   842 ;  The start packet has a flag specifying which starting address should be
                     034C   843 ;  used by the remote port (default or the address sent in the packet).
                     034C   844 ;
                     034C   845 ;  WARNING - Sending starts can be extremely hazardous to the well being of
                     034C   846 ;  nodes on the CI.
                     034C   847 ;
                     034C   848 ;  INPUTS:
                     034C   849 ;
                     034C   850 ;       R2                      - CTP request address
                     034C   851 ;       R3                      - RB address
                     034C   852 ;       R4                      - PDT address
                     034C   853 ;       R5                      - CDRP address
                     034C   854 ;
                     034C   855 ;  OUTPUTS:
                     034C   856 ;
                     034C   857 ;       R0-R5                   - Destroyed
                     034C   858 ;       All other registers     - Preserved
                     034C   859 ;
                     034C   860 ;-
                     034C   861
                     034C   862 CY$GENSTR:
                     034C   863
                     034C   864          ASSUME  CTP$DEFALTADR EQ 0
                     034C   865          ASSUME  CTP$SPECIFADR EQ 1
                     034C   866
08 A3    01    8C    034C   867          XORB2   #^X01,CTP$EXTEND(R3)    ; Change polarity of address flag
                     0350   868          MSTART  -                       ; Send a start packet
                     0350   869                  RSTADR=CTP$OTHERNODE(R2),- ; Remote station address
                     0350   870                  FLAG=CTP$EXTEND(R2),-   ; Default/Specific start addr
                     0350   871                  START=CTP$STARTADR(R2)  ; Start address
   24 A3    B6    035F   872          INCW    RB$W_ACTCNT(R3)         ; Increment the actual count
   22 A3    B7    0362   873          DECW    RB$W_REPEAT(R3)         ; Decrement the repeat count
                     0365   874          BNEQW   NEXT_ENTRY              ; Request not done, branch
                     036A   875
                     036A   876          ASSUME  CTP$SUCCESS EQ 0
                     036A   877
52   10 A3    D0    036A   878          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
   05 A2    94    036E   879          CLRB    CTP$STATUS(R2)          ; Set success status
    FCFE    31    0371   880          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                     0374   881
                     0374   882
                     0374   883
```

```
                                0374   885          .SBTTL  CY$COUNTS,      Counter read routine
                                0374   886 ;+
                                0374   887 ;
                                0374   888 ;  CY$COUNTS
                                0374   889 ;
                                0374   890 ;  Routine to execute a CTP counter read request. The port's performance
                                0374   891 ;  counters are read and reset, and either locked or released, depending
                                0374   892 ;  upon flags in the CTP request. In addition, if the counters are locked,
                                0374   893 ;  they are set up for future monitoring of the port specified in the request.
                                0374   894 ;  If the counters are released, future monitoring is of all ports on the CI.
                                0374   895 ;  The following counters are kept for each path on the CI:
                                0374   896 ;
                                0374   897 ;       - ACK counter
                                0374   898 ;       - NAK counter
                                0374   899 ;       - VORSP counter
                                0374   900 ;
                                0374   901 ;  In addition, the port keeps a discarded datagram counter, used when a
                                0374   902 ;  datagram is received, but no response queue entries are available.
                                0374   903 ;
                                0374   904 ;  To read the counters, a datagram buffer most be allocated, and the address
                                0374   905 ;  of the port queue flink is passed to SCS. (The allocate datagram buffer
                                0374   906 ;  routine returns the address of the application data in the datagram buffer).
                                0374   907 ;
                                0374   908 ;  WARNING - Any suspendable calls to SCS require the stack to be clean, i.e.
                                0374   909 ;  nothing can be saved on the stack when the call is made.
                                0374   910 ;
                                0374   911 ;  INPUTS:
                                0374   912 ;
                                0374   913 ;       R2                      - CTP request address
                                0374   914 ;       R3                      - RB address
                                0374   915 ;       R4                      - PDT address
                                0374   916 ;       R5                      - CDRP address
                                0374   917 ;
                                0374   918 ;  OUTPUTS:
                                0374   919 ;
                                0374   920 ;       R0-R5                   - Destroyed
                                0374   921 ;       All other registers     - Preserved
                                0374   922 ;
                                0374   923 ;-
                                0374   924
                                0374   925 CY$COUNTS:
                                0374   926
                                0374   927          ALLOC_DG_BUF                       ; Allocate a datagram buffer
                                0377   928          BLBCW   R0,100$                    ; Failure, branch
                    26 A3   52  DC  037D   929          MOVL    R2,RB$L_DGBUF(R3)          ; Save DG Buffer Address        [4-004]
                       52   20  C2  0381   930          SUBL2   #SCS$K_APPL_BASE - SCS$B_PPD,R2 ;                          [4-005]
                                0384   931                                             ; Back up to top of datagram buffer
                                0384   932                                             ; [4-001]
                                0384   933
                                0384   934          ASSUME  CTP$RELCNT EQ 0
                                0384   935
                    50  10  A3  D0  0384   936          MOVL    RB$L_CTPBUF(R3),R0         ; Restore CTP request address
                        0A  A0  95  0388   937          TSTB    CTP$CNTFLG(R0)             ; Release the counters?
                            28  13  038B   938          BEQL    20$                        ; Yes, branch
                FF  8F  09  A0  91  038D   939          CMPB    CTP$OTHERNODE(R0),#^XFF    ; Set future monitoring for all nodes?
                            12  13  0392   940          BEQL    10$                        ; Yes branch
                                0394   941
```

```
                                0394   942          READ_COUNTERS   -                       ; Read the counters
                                0394   943                  RSTADR=CTP$OTHERNODE(R0),-  ; Remote station
                                0394   944                  LPRNAM=RESPONDER_NAME       ; Local process name
                     18 50   E8 03A0   945          BLBS    R0,30$                      ; Success, branch
                      0043   31 03A3   946          BRW     150$                        ; Otherwise, branch to error code
                             03A6   947
                             03A6   948 10$:      READ_COUNTERS   -                       ; Read counters
                             03A6   949                  RSTADR=0,-                  ; Monitor all nodes
                             03A6   950                  LPRNAM=RESPONDER_NAME       ; Local process name
                     36 50   E9 03B0   951          BLBC    R0,150$                     ; Failure, branch
                        06   11 03B3   952          BRB     30$
                             03B5   953
                             03B5   954 20$:      RLS_COUNTERS                        ; Release the counters
                     2E 50   E9 03B8   955          BLBC    R0,150$                     ; Error, branch
                             03BB   956
                50   10 A3   D0 03BB   957 30$:      MOVL    RB$L_CTPBUF(R3),R0          ; Restore CTP request address
                        3F   BB 03BF   958          PUSHR   #^M<R0,R1,R2,R3,R4,R5>      ; Save registers
                10 A2   1C   28 03C1   959          MOVC3   #28,PPD$L_PO_ACK(R2),-      ; Copy counters to CTP response
                     0C A0           03C5   960                  CTP$CNTRPOACK(R0)
                        3F   BA 03C7   961          POPR    #^M<R0,R1,R2,R3,R4,R5>      ; Restore registers
                             03C9   962
                             03C9   963          ASSUME  CTP$SUCCESS EQ 0
                             03C9   964          ASSUME  SCS$B_PPD EQ -32
                             03C9   965
                52   26 A3   D0 03C9   966          MOVL    RB$L_DGBUF(R3),R2           ; Restore DG Buffer Address     [4-004]
                     1C A5   D4 03CD   967          CLRL    CDRP$L_MSG_BUF(R5)          ; Show datagram buffer gone
                             03D0   968          DEALLOC_DG_BUF                      ; Deallocate the DG buffer
                52   10 A3   D0 03D3   969          MOVL    RB$L_CTPBUF(R3),R2          ; Restore CTP request address
                     05 A2   94 03D7   970          CLRB    CTP$STATUS(R2)              ; Set success status
                      FC9A   31 03DA   971          BRW     RETURN_CTP_RESPONSE         ; Return the response
                             03DD   972
                52   10 A3   D0 03DD   973 100$:     MOVL    RB$L_CTPBUF(R3),R2          ; Restore CTP request address
                     FD 8F   90 03E1   974          MOVB    #CTP$NORESOURCE,-           ; Fill in appropriate status
                     05 A2           03E4   975                  CTP$STATUS(R2)
                      FC89   31 03E6   976          BRW     RETURN_CTP_RESPONSE         ; Return the response
                             03E9   977
                52   26 A3   D0 03E9   978 150$:     MOVL    RB$L_DGBUF(R3),R2           ; Restore DG Buffer Address     [4-004]
                     1C A5   D4 03ED   979          CLRL    CDRP$L_MSG_BUF(R5)          ; Show datagram buffer gone
                             03F0   980          DEALLOC_DG_BUF                      ; Deallocate the DG buffer
                52   10 A3   D0 03F3   981          MOVL    RB$L_CTPBUF(R3),R2          ; Restore CTP request address
                     FB 8F   90 03F7   982          MOVB    #CTP$FAILURE,-              ; Set appropriate CTP status
                     05 A2           03FA   983                  CTP$STATUS(R2)
                      FC73   31 03FC   984          BRW     RETURN_CTP_RESPONSE         ; Return the response
                             03FF   985
```

```
                          03FF        .SBTTL  CY$CONNECT,     Third party connect routine
                          03FF  ;+
                          03FF  ;
                          03FF  ;  CY$CONNECT
                          03FF  ;
                          03FF  ;  Routine to execute a CTP connect request. A third party connection is
                          03FF  ;  established to the responser in the specified remote port. However, if
                          03FF  ;  a connection already exists, no new connection is started, but the CTP
                          03FF  ;  response is returned with success status. This situation can occur if
                          03FF  ;  two controllers are running in the clusters at once.
                          03FF  ;
                          03FF  ;  If a third party connection already exists to the specified responder, but
                          03FF  ;  it is in a state of transition (CONNECTing or DISCONNECTing), return
                          03FF  ;  "CONNECTION IN TRANSITION" status to the controller. The controller can
                          03FF  ;  the reissue the CTP connect request.
                          03FF  ;
                          03FF  ;  It is the responsibility of the responder, when it senses no connections
                          03FF  ;  exist any longer to contollers, to discard any connections it may have
                          03FF  ;  to resonders.
                          03FF  ;
                          03FF  ;  To issue an SCS CONNECT, the responder must know the system ID of the
                          03FF  ;  system in which the remote responder resides. This is obtained by issuing
                          03FF  ;  and SCS CONFIG_PTH. This routine returns the address of a path block,
                          03FF  ;  from which the system block address and system ID can be obtained.
                          03FF  ;
                          03FF  ;  When the suspended connect thread is resumed, the responder must check
                          03FF  ;  for an ACTIVE connection to the controller that send the CTP connect
                          03FF  ;  request. This connection may have gone away while the connect thread
                          03FF  ;  was suspended. If this is the case, the third party connection resources
                          03FF  ;  are deallocated, and if the third party connection succeeded, a disconnect
                          03FF  ;  is issued.
                          03FF  ;
                          03FF  ;  INPUTS:
                          03FF  ;
                          03FF  ;       R2                              - CTP request address
                          03FF  ;       R3                              - RB address
                          03FF  ;       R4                              - PDT address
                          03FF  ;       R5                              - CDRP address
                          03FF  ;
                          03FF  ;  OUTPUTS:
                          03FF  ;
                          03FF  ;       R0-R5                           - Destroyed
                          03FF  ;       All other registers             - Preserved
                          03FF  ;
                          03FF  ;-
                          03FF
                          03FF  CY$CONNECT:
                          03FF
                          03FF
7E     09 A2    9A        03FF        MOVZBL  CTP$OTHERNODE(R2),-(SP) ; Push remote port name on stack
51     0C A3    D0        0403        MOVL    RB$L_CB(R3),R1          ; Get connection block address
7E     2E A1    D0        0407        MOVL    CB$T_LPRINAM(R1),-(SP)  ; Push local port name on stack
       FBF2'    30        040B        BSBW    CHECK_CONN_RSP          ; Does connection already exist?
5E     08       C0        040E        ADDL    #8,SP                   ; Clean off stack
       43 50    E9        0411        BLBC    R0,10$                  ; No, branch
                          0414
                          0414        BBSW    #CB_V_CONN,-            ; Branch if a CONNECT is in progress
```

```
                                0414  1044                    CB$B_STATUS(R1),350$    ; for this connection
                                041C  1045          BBSW    #CB_V_DISC,-            ; Branch if a DISCONNECT is in progress
                                041C  1046                    CB$B_STATUS(R1),350$    ; for this connection
                     00DC   31  0424  1047          BRW     100$                    ; Connection active, branch
                                0427  1048
                 50  14 A3   D0  0427  1049 10$:   MOVL    RB$L_CDT(R3),R0         ; Get CDT address
                 50  1C A0   D0  042B  1050          MOVL    CDT$L_PB(R0),R0         ; Get LOCAL path block address
                     24 A0   DD  042F  1051          PUSHL   PB$T_PORT_NAME(R0)      ; Push local port name on stack
                        7E   D4  0432  1052          CLRL    -(SP)                   ; Clear a longword
                 7E  09 A2   9A  0434  1053          MOVZBL  CTP$OTHERNODE(R2),-(SP) ; Push remote port number on stack
                                0438  1054          CONFIG_PTH -                    ; Issue configuration path
                                0438  1055                  STAADR=(SP),-           ; Input array
                                0438  1056                  OUTBUF=0                ; No output needed, R1 gets PB addr
                    5E   0C   C0  0443  1057          ADDL    #12,SP                  ; Clean off stack
                                0446  1058          BLBCW   R0,150$                 ; Remote node unknown, branch
                 52  30 A1   D0  044C  1059          MOVL    PB$L_SBLINK(R1),R2      ; Save system block address
                                0450  1060
                     FBAD'  30  0450  1061          BSBW    ALLOC_CB                ; Allocate a connection block
                                0453  1062          BLBCW   R0,200$                 ; Failure, branch
                 50  10 A3   D0  0459  1063          MOVL    RB$L_CTPBUF(R3),R0      ; Get CTP request address
                 50  09 A0   9A  045D  1064          MOVZBL  CTP$OTHERNODE(R0),R0    ; Get remote port number
                     FB9C'  30  0461  1065          BSBW    ENTER_CONN_LIST         ; Put CB on node list
                                0464  1066
                 50  10 A3   D0  0464  1067          MOVL    RB$L_CTPBUF(R3),R0      ; Get CTP request address
                     09 A0   9A  0468  1068          MOVZBL  CTP$OTHERNODE(R0),-     ; Fill in remote port number
                     26 A1       046B  1069                  CB$B_RPRTNUM(R1)
                     2A A1   D4  046D  1070          CLRL    CB$B_RPRTNUM+4(R1)      ; Clear next longword
                 50  14 A3   D0  0470  1071          MOVL    RB$L_CDT(R3),R0         ; Get CDT address
                 50  1C A0   D0  0474  1072          MOVL    CDT$L_PB(R0),R0         ; Get LOCAL path block address
                     24 A0   D0  0478  1073          MOVL    PB$T_PORT_NAME(R0),-    ; Fill in local port name
                     2E A1       047B  1074                  CB$T_LPRTNAM(R1)
                                047D  1075
                        01   88  047D  1076          BISB2   #CB_M_CONN,-            ; Show that a connect is in progress
                     05 A1       047F  1077                  CB$B_STATUS(R1)         ; in the connection block
                        01   88  0481  1078          BISB2   #RB_M_CONN,-            ; Show that a third party connection
                     0B A3       0483  1079                  RB$B_STATUS(R3)         ; is in progress in the request block
                 42 A1   53   D0  0485  1080          MOVL    R3,CB$L_RB(R1)          ; Save RB address in third party CB
                                0489  1081
                    55   51   D0  0489  1082          MOVL    R1,R5                   ; Save CB address
                                048C  1083          CONNECT -                       ; Attempt connection
                                048C  1084                  MSGADR=CY$MSG_INPUT,-   ; Message input address
                                048C  1085                  DGADR=CY$DG_INPUT,-     ; Datagram input address
                                048C  1086                  ERRADR=CY$CONNECT_ERR,- ; Error input address
                                048C  1087                  RSYSID=SB$B_SYSTEMID(R2),- ; Remote system ID
                                048C  1088                  RSTADR=CB$B_RPRTNUM(R1),- ; Local port name/remote port number
                                048C  1089                  RPRNAM=RESPONDER_NAME,- ; Remote process name
                                048C  1090                  LPRNAM=RESPONDER_NAME,- ; Local process name
                                048C  1091                  INITCR=#INIT_CREDIT,-   ; Initial credits
                                048C  1092                  MINSCR=#MIN_SEND_CR,-   ; Minimum send credits
                                048C  1093                  INITDG=#INIT_DG_BUF,-   ; Initial datagram buffers
                                048C  1094                  CONDAT=CONNECT_DATA,-   ; Connection data
                                048C  1095                  AUXSTR=(R5)             ; Connection block address
                                04C9  1096
                 52  42 A5   D0  04C9  1097          MOVL    CB$L_RB(R5),R2          ; Restore request block address
                        01   8A  04CD  1098          BICB2   #CB_M_CONN,-            ; Connect no longer in progress
                     05 A5       04CF  1099                  CB$B_STATUS(R5)
                        01   8A  04D1  1100          BICB2   #RB_M_CONN,-            ; Connect no longer in progress
```

```
       CB A2        04D3                RB$B_STATUS(R2)
          01   E0   04D5          BBS   #RB_V_CANCEL,-            ; Branch if SECOND party connection
    3F 0B A2        04D7                RB$B_STATUS(R2),250$      ; failed during suspended connect call
       61 50   E9   04DA          BLBC  R0,300$                   ; If connect call failed, branch
                    04DD
                    04DD          ASSUME  CB$L_CDT+4 EQ CB$L_PDT
                    04DD          ASSUME  CTP$SUCCESS EQ 0
                    04DD
      50    55 D0   04DD          MOVL  R5,R0                     ; Get connection block address
   52    1C A2 D0   04E0          MOVL  RB$L_CTPBUF(R2),R2        ; Restore CTP request address
   50    5C A3 D0   04E4          MOVL  CDT$L_AUXSTRUC(R3),R0     ; Get 3rd party CB address
     010C D4   90   04E8          MOVB  @PDT$L_PPR(R4),-          ; Fill in local port number
        25 A0       04EC                CB$B_LPRTNUM(R0)
        09 A2  90   04EE          MOVB  CTP$OTHERNODE(R2),-       ; Fill in remote port number
        26 A0       04F1                CB$B_RPRTNUM(R0)
   0C A0    53 7D   04F3          MOVQ  R3,CB$L_CDT(R0)           ; Copy CDT, PDT address
     0000'CF   7D   04F7          MOVQ  RESPONDER_NAME,-          ; Fill in remote process name h.o.
        32 A0       04FB                CB$T_RPROCNAM(R0)
     0008'CF   7D   04FD          MOVQ  RESPONDER_NAME+8,-        ; remote process name l.o.
        3A A0       0501                CB$T_RPROCNAM+8(R0)
        05 A2  94   0503 100$:    CLRB  CTP$STATUS(R2)            ; Set success status
        FB69   31   0506          BRW   RETURN_CTP_RESPONSE       ; Return the response
                    0509
        F8 8F  90   0509 150$:    MOVB  #CTP$NODEUNKNOWN,-        ; Fill in appropriate status
        05 A2       050C                CTP$STATUS(R2)
        FB61   31   050E          BRW   RETURN_CTP_RESPONSE       ; Return the response
                    0511
        FD 8F  90   0511 200$:    MOVB  #CTP$NORESOURCE,-         ; Set appropriate status
        05 A2       0514                CTP$STATUS(R2)
        FB59   31   0516          BRW   RETURN_CTP_RESPONSE       ; Return the response
                    0519
                    0519
                    0519 ;  We arrive here if the SECOND party connection failed during the suspended
                    0519 ;  THIRD party connect call. Deallocate the request block, and throw away the
                    0519 ;  third party connection resources. If the third party connect was successful,
                    0519 ;  issue a disconnect for it. At this point, the registers look as follows:
                    0519 ;
                    0519 ;          R0      - Returned status from connect call
                    0519 ;          R2      - Request block address
                    0519 ;          R5      - Third party connection block address
                    0519
           50  DD   0519 250$:    PUSHL R0                        ; Save CONNECT status
      50    52 D0   051B          MOVL  R2,R0                     ; Get request block address
  00000000'GF  16   051E          JSB   G^COM$DRVDEALMEM          ; Deallocate the request block
        50 8ED0     0524          POPL  R0                        ; Restore CONNECT status
       0A 50   E9   0527          BLBC  R0,260$                   ; If CONNECT failed, branch
          02   88   052A          BISB2 #CB_M_DISC,-              ; Show DISCONNECT issued for this
       0B A5        052C                CB$B_STATUS(R5)           ; connection
                    052E          DISCONNECT                      ; Throw away third party connection
                    0534
      50    65 0F   0534 260$:    REMQUE (R5),R0                  ; Remove CB from connection list
  00000000'GF  16   0537          JSB   G^COM$DRVDEALMEM          ; Deallocate the third party CB
               05   053D          RSB                             ; Return to SCS
                    053E
                    053E
                    053E ;  we arrive here if the third party connect call failed. Remove the
                    053E ;  third party connection block from the connection list, deallocate it,
```

```
                                053E  1158 ;  set the apporpriate status in the CTP response, and send the response
                                053E  1159 ;  back to the controller. If the error status returned from the connect
                                053E  1160 ;  call was SS$_REMRSRC, it means the remote responder was busy accepting
                                053E  1161 ;  a connection with another process when we tried to connect with it. Tell
                                053E  1162 ;  the controller this by returning CTP$CONNTRANS status.
                                053E  1163
                 51    10 A2    D0  053E  1164 300$:   MOVL    RB$L_CTPBUF(R2),R1      ; Get CTP buffer address
                       F5 8F    90  0542  1165         MOVB    #CTP$CONNTRANS,-        ; Assume remote responder was
                          05 A1      0545  1166                 CTP$STATUS(R1)          ; busy
              50    0000206C 8F    D1  0547  1167         CMPL    #SS$_REMRSRC,R0         ; Was remote responder busy?
                          05    13  054E  1168         BEQL    310$                    ; Yes, branch
                       FB 8F    90  0550  1169         MOVB    #CTP$FAILURE,-          ; Set appropriate status
                          05 A1      0553  1170                 CTP$STATUS(R1)
                        50    65 0F  0555  1171 310$:   REMQUE  (R5),R0                 ; Remove CB from queue
                 00000000'GF    16  0558  1172         JSB     G^COM$DRVDEALMEM        ; Deallocate it
                       FB11    31  055E  1173         BRW     RETURN_CTP_RESPONSE     ; Return the response
                                0561  1174
                                0561  1175
                                0561  1176 ;  We arrive here if the third party connection is in a state of transition.
                                0561  1177 ;  That is, if the connection is either still being established (connect in
                                0561  1178 ;  progress), or if the connection is going down (disconnect in progress).
                                0561  1179 ;  Return "CONNECTION IN TRANSITION" status to the controller. The controller
                                0561  1180 ;  then has the option of retrying the connect.
                                0561  1181
                       F5 8F    90  0561  1182 350$:   MOVB    #CTP$CONNTRANS,-        ; Indicate connection is in
                          05 A2      0564  1183                 CTP$STATUS(R2)          ; transition
                        FB09    31  0566  1184         BRW     RETURN_CTP_RESPONSE     ; Return the response
                                0569  1185
                                0569  1186
                                0569  1187
```

```
                                0569  1189          .SBTTL  CY$FINISH,      Listener disconnect routine
                                0569  1190 ;+
                                0569  1191 ;
                                0569  1192 ;  CY$FINISH
                                0569  1193 ;
                                0569  1194 ;  Routine to execute a CTP disconnect request. First, a CTP disconnect
                                0569  1195 ;  response is returned to the controller because it will not get through
                                0569  1196 ;  if the connection is broken first.
                                0569  1197 ;
                                0569  1198 ;  All resources are then returned for the connection, and all an SCS
                                0569  1199 ;  DISCONNECT is issued. When the suspended disconnect thread is resumed,
                                0569  1200 ;  the connection block is deallocated and returned to pool.
                                0569  1201 ;
                                0569  1202 ;  INPUTS:
                                0569  1203 ;
                                0569  1204 ;       R2                      - CTP request address
                                0569  1205 ;       R3                      - RB address
                                0569  1206 ;       R4                      - PDT address
                                0569  1207 ;       R5                      - CDRP address
                                0569  1208 ;
                                0569  1209 ;  OUTPUTS:
                                0569  1210 ;
                                0569  1211 ;       R0-R5                   - Destroyed
                                0569  1212 ;       All other registers     - Preserved
                                0569  1213 ;
                                0569  1214 ;-
                                0569  1215
                                0569  1216 CY$FINISH:
                                0569  1217
                                0569  1218          ASSUME  CTP$SUCCESS EQ 0
                                0569  1219
  62    40 8F    80             0569  1220          ADDB2   #64,CTP$OPCODE(R2)      ; Change request to response
        05 A2    94             056D  1221          CLRB    CTP$STATUS(R2)          ; Set success status
        10 A3    D4             0570  1222          CLRL    RB$L_CTPBUF(R3)         ; Show CTP request is gone in RB
  1C A5    52    D0             0573  1223          MOVL    R2,CDRP$L_MSG_BUF(R5)   ; Copy CTP request address to CDRP
                                0577  1224          RECYCL_MSG_BUF                  ; Recycle the message buffer
                                057A  1225          SEND_MSG_BUF                    ; Return the CTP response
                                057D  1226
53    0000'CF    D0             057D  1227          MOVL    EXEC_LIST,R3            ; Restore request block address
  52    0C A3    D0             0582  1228          MOVL    RB$L_CB(R3),R2          ; Move CB address to R2          [4-006]
        FA77'    30             0586  1229          BSBW    CLEAN_CONN              ; Clean up connection resources
     55    52    D0             0589  1230          MOVL    R2,R5                   ; Save connection block address
53    0000'CF    D0             058C  1231          MOVL    CY$L_SAVED_CDT,R3       ; Restore Saved CDT Address      [4-006]
                                0591  1232          DISCONNECT #SS$_DISCONNECT      ; Disconnect
     50    65    0F             059E  1233          REMQUE  (R5),R0                 ; Remove the CB from node list
 00000000'GF    16             05A1  1234          JSB     G^COM$DRVDEALMEM        ; Deallocate the CB
        FA7A    31             05A7  1235          BRW     NEXT_ENTRY              ; Go execute next request in exec list
                                05AA  1236
                                05AA  1237          .END
```

```
$$$CURSIZ                       = 000001C4
$$$NEWSIZ                       = 0C0001D0
ALLOC_BUFFER                      ********    X   01
ALLOC_CB                          ********    X   01
ALLOC_IRP_CDRP                    ********    X   01
BUILD_BUFFER                      ********    X   01
BUILD_DG                          ********    X   01
BUILD_MSG                         ********    X   01
CB$B_BUFMAP_CNT                   0000001C
CB$B_LPRTNUM                      00000025
CB$B_RPRTNUM                      00000026
CB$B_STATUS                       0000000B
CB$B_TYPE                         0000000A
CB$K_LENGTH                       00000046
CB$L_BLINK                        00000004
CB$L_CDT                          0000000C
CB$L_FLINK                        00000000
CB$L_PDT                          00000010
CB$L_RB                           00000042
CB$Q_MAPQ                         00000014
CB$Q_MNTMAPQ                      0000001D
CB$T_LPRTNAM                      0000002E
CB$T_RPROCNAM                     00000032
CB$W_SIZE                         00000008
CB_M_CONN                       = 00000001
CB_M_DISC                       = 00000002
CB_V_CONN                       = 00000000
CB_V_DISC                       = 00000001
CDRP$B_FLAGS                      00000040
CDRP$B_RMOD                     = FFFFFFAB
CDRP$C_BT_LEN                   = 00000040
CDRP$C_CY_LEN                     00000080
CDRP$K_CY_LEN                     00000080
CDRP$L_BCNT                     = FFFFFFD2
CDRP$L_BUFADR                     0000005C
CDRP$L_BUFLEN                     00000060
CDRP$L_CB_BL                      00000074
CDRP$L_CB_FL                      00000070
CDRP$L_CDT                      = 00000024
CDRP$L_LBOFF                    = 00000030
CDRP$L_LBUFH_AD                 = 0000002C
CDRP$L_MAPQ_BL                    0000007C
CDRP$L_MAPQ_FL                    00000078
CDRP$L_MNT_NAM                    00000064
CDRP$L_MSG_BUF                  = 0000001C
CDRP$L_RBOFF                    = 00000038
CDRP$L_RBUFH_AD                 = 00000034
CDRP$L_RB_BL                      0000006C
CDRP$L_RB_FL                      00000068
CDRP$L_SVAPTE                   = FFFFFFCC
CDRP$L_XCT_LEN                  = 0000003C
CDRP$T_CY_LBUFHNDL                00000044
CDRP$T_CY_RBUFHNDL                00000050
CDRP$W_BOFF                     = FFFFFFD0
CDT$L_AUXSTRUC                  = 0000005C
CDT$L_LCONID                    = 00000018
CDT$L_PB                        = 0000001C
CDT$L_RCONID                    = 00000014
CHECK_CONN_RSP                    ********    X   01
CHECK_CTP_REQ                     ********    X   01
CIBHAN$L_BNAME                  = 00000004
CIBHAN$L_BOFF                   = 00000000
CIBHAN$L_RCONID                 = 00000008
CLEAN_CONN                        ********    X   01
CLEAR_BUFFER                      ********    X   01
COM$DRVDEALMEM                    ********    X   01
CONNECT_DATA                      ********    X   01
CTP$ACTCOUNT                      00000006
CTP$BUFLENGTH                     0000000C
CTP$BUFLIMIT                    = 000000FA
CTP$BUFLNAME                      00000010
CTP$BUFLOFSET                     00000014
CTP$BUFMAPREQ                     00000001
CTP$BUFMAPRSP                     00000041
CTP$BUFRNAME                      00000018
CTP$BUFROFSET                     0000001C
CTP$BUFTYPE                       00000005
CTP$BUFUNMREQ                     00000002
CTP$BUFUNMRSP                     00000042
CTP$CDATPREV                      00000002
CTP$CDATPTYPE                     00000000
CTP$CDATPVERS                     00000001
CTP$CFGPOSTS                      0000000B
CTP$CFGP1STS                      0000000C
CTP$CNTFLG                        0000000A
CTP$CNTRDISCDG                    00000024
CTP$CNTRPOACK                     0000000C
CTP$CNTRPONAK                     00000010
CTP$CNTRPONRSP                    00000014
CTP$CNTRP1ACK                     00000018
CTP$CNTRP1NAK                     0000001C
CTP$CNTRP1NRSP                    00000020
CTP$CONDRST                     = 00000001
CTP$CONFIGREQ                     00000009
CTP$CONFIGRSP                     00000049
CTP$CONNECTREQ                    0000000B
CTP$CONNECTRSP                    0000004B
CTP$CONNTRANS                   = 000000F5
CTP$COUNTSREQ                     0000000A
CTP$COUNTSRSP                     0000004A
CTP$DEFALTADR                   = 00000000
CTP$DELAY                         00000005
CTP$EXTEND                        00000008
CTP$FAILURE                     = 00C000FB
CTP$FINISHREQ                     0000000C
CTP$FINISHRSP                     0000004C
CTP$FMASK                         00000006
CTP$FUNCTREQ                      00000000
CTP$FUNCTRSP                      00000040
CTP$GENCONST                      00000009
CTP$GENDATA                       0000000E
CTP$GENDGRREQ                     00000005
CTP$GENDGRRSP                     00000045
CTP$GENFUNCT                      00000008
```

```
CTP$GENLENGTH                   0000000C
CTP$GENMSGREQ                   00000004
CTP$GENMSGRSP                   00000044
CTP$GENRSTREQ                   00000006
CTP$GENRSTRSP                   00000046
CTP$GENSTRREQ                   00000007
CTP$GENSTRRSP                   00000047
CTP$IMAGEDATA                   0000000E
CTP$MAXCMDOPC                   00000011
CTP$MBUFMAPREQ                  0000000D
CTP$MBUFMAPRSP                  0000004D
CTP$MBUFUNMREQ                  0000000E
CTP$MBUFUNMRSP                  0000004E
CTP$MOVBUFREQ                   00000003
CTP$MOVBUFRSP                   00000043
CTP$MOVEFROM                =   00000000
CTP$MOVETYPE                    00000008
CTP$MOVMBUFREQ                  0000000F
CTP$MOVMBUFRSP                  0000004F
CTP$MSTATEREQ                   00000010
CTP$MSTATERSP                   00000050
CTP$NOACTFLAG                   0000000A
CTP$NOACTREQ                    00000008
CTP$NOACTRSP                    00000048
CTP$NOBUFMAPPED             =   000000F9
CTP$NOCONNECT               =   000000F6
CTP$NODEUNKNOWN             =   000000F8
CTP$NORESOURCE              =   000000FD
CTP$OPCODE                      00000000
CTP$OPEXPAND                    000000FF
CTP$OTHERNODE                   00000009
CTP$PKTMULT                     0000000B
CTP$PKTSIZ                      0000000A
CTP$REFERENCE                   00000001
CTP$RELCNT                  =   00000000
CTP$REPCOUNT                    00000006
CTP$RESERV10                    0000000A
CTP$RESERV11                    0000000B
CTP$RESERV12                    0000000C
CTP$RESERV20                    00000014
CTP$RESERV5                     00000005
CTP$RESERV6                     00000006
CTP$RESERV7                     00000007
CTP$RESERV9                     00000009
CTP$REVISION                =   00000000  G
CTP$SPECIFADR               =   00000001
CTP$STARTADR                    0000000C
CTP$STATUS                      00000005
CTP$SUCCESS                 =   00000000
CTP$UNCONDRST               =   00000000
CTP$VERSION                 =   00000003  G
CY$BUFMAP                       0000017D R      01
CY$BUFUNM                       000002E8 R      01
CY$CONFIG                       00000141 R      01
CY$CONNECT                      000003FF R      01
CY$CONNECT_ERR                  ******** X      01
CY$COUNTS                       00000374 R      01
CY$DG_INPUT                     ******** X      01
CY$FINISH                       00000569 R      01
CY$FUNCT                        000000B0 R      01
CY$GENDG                        000000F7 R      01
CY$GENMSG                       000000CC R      01
CY$GENRST                       00000328 R      01
CY$GENSTR                       0000034C R      01
CY$L_SAVED_CDT                  ******** X      01
CY$L_SAVED_UCB                  ******** X      01
CY$MNT_BUFMAP                   ******** X      01
CY$MNT_BUFUNM                   ******** X      01
CY$MNT_MOVBUF                   ******** X      01
CY$MNT_STATE                    ******** X      01
CY$MOVBUF                       00000219 R      01
CY$MSG_INPUT                    ******** X      01
CY$NOACT                        ******** X      01
DEALLOC_RB                      ******** X      01
DYN$C_ACB                   =   00000002  G
DYN$C_ACL                   =   0000003F  G
DYN$C_ADP                   =   00000001  G
DYN$C_AQB                   =   00000003  G
DYN$C_BOOTCB                =   00000006  G
DYN$C_BRDCST                =   0000001A  G
DYN$C_BUFIO                 =   00000013  G
DYN$C_CDB                   =   00000033  G
DYN$C_CDRP                  =   00000039  G
DYN$C_CD_BBRPG              =   00000002  G
DYN$C_CD_CDDB               =   00000001  G
DYN$C_CEB                   =   00000004  G
DYN$C_CHIP                  =   00000048  G
DYN$C_CI                    =   00000061  G
DYN$C_CIA                   =   00000045  G
DYN$C_CIDG                  =   0000003B  G
DYN$C_CIMSG                 =   0000003C  G
DYN$C_CI_BDT                =   00000001  G
DYN$C_CI_FQDT               =   00000002  G
DYN$C_CLASSDRV              =   00000064  G
DYN$C_CLU                   =   00000065  G
DYN$C_CLU_BTX               =   00000004  G
DYN$C_CLU_CLUB              =   00000003  G
DYN$C_CLU_CLUDCB            =   00000005  G
DYN$C_CLU_CLUOPT            =   00000006  G
DYN$C_CLU_CLUVEC            =   00000002  G
DYN$C_CLU_CSB               =   00000001  G
DYN$C_CLU_LCKDIR            =   00000007  G
DYN$C_CONF                  =   00000007  G
DYN$C_CRB                   =   00C00005  G
DYN$C_CST                   =   00000008  G
DYN$C_CXB                   =   0000001B  G
DYN$C_DCCB                  =   00000027  G
DYN$C_DDB                   =   00000006  G
DYN$C_DPT                   =   0000001E  G
DYN$C_ERP                   =   0000003A  G
DYN$C_EXTGSD                =   00000028  G
DYN$C_FCB                   =   00000007  G
DYN$C_FRK                   =   00000008  G
DYN$C_GSD                   =   00000015  G
```

```
DYN$C_IDB                       =  00000009  G
DYN$C_INIT                      =  00000063  G
DYN$C_IRP                       =  0000000A  G
DYN$C_IRPE                      =  0000002C  G
DYN$C_JIB                       =  0000002F  G
DYN$C_JNL                       =  00000067  G
DYN$C_JNLWCB                    =  00000024  G
DYN$C_JNL_ABL                   =  00000001  G
DYN$C_JNL_ACBM                  =  00000004  G
DYN$C_JNL_ADL                   =  00000002  G
DYN$C_JNL_BCB                   =  00000003  G
DYN$C_JNL_BUF                   =  00000005  G
DYN$C_JNL_BXSTS                 =  00000013  G
DYN$C_JNL_CWQ                   =  00000010  G
DYN$C_JNL_DB                    =  00000006  G
DYN$C_JNL_DIOREAD               =  00000015  G
DYN$C_JNL_JMT                   =  00000009  G
DYN$C_JNL_MSG                   =  00000012  G
DYN$C_JNL_MSGDATA               =  00000014  G
DYN$C_JNL_NDL                   =  00000008  G
DYN$C_JNL_RC                    =  00000011  G
DYN$C_JNL_RCPC                  =  0000000C  G
DYN$C_JNL_RM                    =  0000000A  G
DYN$C_JNL_RRP                   =  0000000B  G
DYN$C_JNL_RUL                   =  0000000D  G
DYN$C_JNL_SFT                   =  00000007  G
DYN$C_JNL_VCL                   =  0000000E  G
DYN$C_JNL_VLE                   =  0000000F  G
DYN$C_JPB                       =  0000001F  G
DYN$C_KFD                       =  00000043  G
DYN$C_KFE                       =  00000018  G
DYN$C_KFPB                      =  00000044  G
DYN$C_KFRH                      =  00000026  G
DYN$C_LC_CHREML                 =  00000006  G
DYN$C_LC_CLS                    =  00000005  G
DYN$C_LC_FPEMUL                 =  00000007  G
DYN$C_LC_MP                     =  00000003  G
DYN$C_LC_MSCP                   =  00000008  G
DYN$C_LC_SCS                    =  00000004  G
DYN$C_LC_SYSL                   =  00000009  G
DYN$C_LKB                       =  00000035  G
DYN$C_LKID                      =  00000037  G
DYN$C_LNM                       =  00000040  G
DYN$C_LOADCODE                  =  00000062  G
DYN$C_LOG                       =  0000000B  G
DYN$C_LPD                       =  00000034  G
DYN$C_MBX                       =  0000002B  G
DYN$C_MPWMAP                    =  00000004  G
DYN$C_MTL                       =  00000019  G
DYN$C_MVL                       =  00000016  G
DYN$C_NDB                       =  0000001C  G
DYN$C_NET                       =  00000017  G
DYN$C_NON_PAGED                 =  00000001  G
DYN$C_ORB                       =  00000049  G
DYN$C_PAGED                     =  00000002  G
DYN$C_PBH                       =  00000020  G
DYN$C_PCB                       =  0000000C  G
DYN$C_PCBVEC                    =  00000001  G
DYN$C_PDB                       =  00000021  G
DYN$C_PFB                       =  00000047  G
DYN$C_PFL                       =  00000023  G
DYN$C_PGD                       =  00000066  G
DYN$C_PGD_F11BC                 =  00000001  G
DYN$C_PHVEC                     =  00000002  G
DYN$C_PIB                       =  00000022  G
DYN$C_PMB                       =  00000046  G
DYN$C_PQB                       =  0000000D  G
DYN$C_PRCMAP                    =  00000005  G
DYN$C_PTR                       =  00000025  G
DYN$C_RBM                       =  00000031  G
DYN$C_RIGHTSLIST                =  00000042  G
DYN$C_RSB                       =  00000036  G
DYN$C_RSHT                      =  00000038  G
DYN$C_RVT                       =  0000000E  G
DYN$C_SCS                       =  00000060  G
DYN$C_SCS_CDL                   =  00000001  G
DYN$C_SCS_CDT                   =  00000002  G
DYN$C_SCS_DIR                   =  00000003  G
DYN$C_SCS_HQB                   =  0000000B  G
DYN$C_SCS_PB                    =  00000004  G
DYN$C_SCS_PDT                   =  00000005  G
DYN$C_SCS_RDT                   =  00000006  G
DYN$C_SCS_SB                    =  00000007  G
DYN$C_SCS_SPNB                  =  00000009  G
DYN$C_SCS_SPPB                  =  00000008  G
DYN$C_SCS_UQB                   =  0000000A  G
DYN$C_SHB                       =  0000002A  G
DYN$C_SHMCEB                    =  0000002E  G
DYN$C_SHMGSD                    =  00000029  G
DYN$C_SHRBUFIO                  =  00000080  G
DYN$C_SLAVCEB                   =  0000002D  G
DYN$C_SPECIAL                   =  00000080  G
DYN$C_SSB                       =  0000001D  G
DYN$C_SUBTYPE                   =  00000060  G
DYN$C_SWPMAP                    =  00000003  G
DYN$C_TQE                       =  0000000F  G
DYN$C_TWP                       =  00000030  G
DYN$C_TYPAHD                    =  00000014  G
DYN$C_UCB                       =  00000010  G
DYN$C_UNUSED_2                  =  00000041  G
DYN$C_VCA                       =  00000032  G
DYN$C_VCB                       =  00000011  G
DYN$C_WCB                       =  00000012  G
DYN$C_WQE                       =  0000003E  G
DYN$C_XWB                       =  0000003D  G
ENTER_CONN_LIST                    ********    X    01
EXE$IOFORK                         ********    X    01
EXECUTE_CMD                        00000033  P      01
EXEC_LIST                          ********    X    01
FIND_MAPPED_BUFFER                 ********    X    01
FUNCTION_MASK                   =  0001FFFF  G
INIT_CREDIT                        ********    X    01
INIT_DG_BUF                        ********    X    01
IRP$K_CDRP                      =  00000060
```

```
MAINLINE                        00000000 RG    01
MAX_BUF_MAP                     ******** X     01
MIN_SEND_CR                     ******** X     01
MMG$GL_SPTBASE                  ******** X     01
MNT$CATCH_RST                 = 00000007 G
MNT$CATCH_STR                 = 00000008 G
MNT$MAINT_READ                = 00000005 G
MNT$MAINT_STATE               = 00000002 G
MNT$MAINT_WRITE               = 00000006 G
MNT$NORMAL_STATE              = 00000004 G
MNT$POLLER_OFF                = 00000000 G
MNT$POLLER_ON                 = 00000001 G
MNT$UNINIT_STATE              = 00000003 G
NEXT_ENTRY                      00000024 RG    01
PB$B_P0_STS                   = 00000029
PB$B_P1_STS                   = 0000002A
PB$L_SBLINK                   = 00000030
PB$T_LPORT_NAME               = 00000024
PDT$B_DQIMAP                    00000154
PDT$B_HSHUT_DG                  000001B0
PDT$B_MAX_PORT                  0000017C
PDT$B_NXT_PORT                  0000017E
PDT$B_P0_LBSTS                  00000180
PDT$B_P1_LBSTS                  00000181
PDT$B_PLOGMAP                   00000134
PDT$B_PORTMAP                   00000114
PDT$B_PORT_NUM                  0000017D
PDT$B_REQIDPS                   0000017F
PDT$C_LENGTH                  = 000000E4
PDT$C_PAREGBASE                 000000E4
PDT$C_PAREGEND                  00000110
PDT$C_PQB                     = 000001E0
PDT$L_ALLOCDG                 = 00000010
PDT$L_ALLOCMSG                = 00000014
PDT$L_CNF                       000000E4
PDT$L_CQ0                       000000F0
PDT$L_CQ1                       000000F4
PDT$L_DEALLOCDG               = 0000001C
PDT$L_DFQ                       000000FC
PDT$L_DFQHDR                    00000208
PDT$L_DGHDRSZ                   00000190
PDT$L_DGNETHD                   00000194
PDT$L_DQELOGOUT                 000002E0
PDT$L_GPTBASE                   0000022C
PDT$L_GPTLEN                    00000230
PDT$L_LBDG                      00000184
PDT$L_MAPIRP                  = 00000034
PDT$L_MFQ                       00000100
PDT$L_MFQHDR                    0000020C
PDT$L_MQELOGOUT                 00000320
PDT$L_MRESET                  = 00000070
PDT$L_MSTART                  = 00000074
PDT$L_MTC                       00000104
PDT$L_PFAR                      00000108
PDT$L_PMC                       000000E8
PDT$L_POLLERDUE                 0000018C
PDT$L_POOLDUE                   00000188
PDT$L_PPR                       0000010C
PDT$L_PS                        000000EC
PDT$L_PSR                       000000F8
PDT$L_RCLMSGBUF               = 00000048
PDT$L_READCOUNT               = 00000068
PDT$L_REQDATA                 = 00000050
PDT$L_RLSCOUNT                = 0000006C
PDT$L_SENDDATA                = 00000054
PDT$L_SENDDG                  = 00000058
PDT$L_SENDMSG                 = 0000005C
PDT$L_SPTBASE                   00000224
PDT$L_SPTLEN                    00000228
PDT$L_UNMAP                   = 00000064
PDT$L_VBDT                      0000021C
PDT$L_VPQB                      00000218
PDT$Q_COMQ2                     000001F0
PDT$Q_COMQ3                     000001F8
PDT$Q_COMQBASE                  000001E0
PDT$Q_COMQH                     000001E8
PDT$Q_COMQL                     000001E0
PDT$Q_DFREEQ                    000001D0
PDT$Q_FORMPB                    00000174
PDT$Q_MFREEQ                    000001D8
PDT$Q_RSPQ                      00000200
PDT$Q_TEMP_RSPQ                 00000198
PDT$W_BDTLEN                    00000220
PDT$W_DQELEN                    00000210
PDT$W_LPORT_STS                 00000110
PDT$W_MQELEN                    00000214
PDT$W_PBCOUNT                   00000112
PPD$B_DEF_ST                    0000001C
PPD$B_FLAGS                     0000000F
PPD$B_HWVERS                    00000034
PPD$B_LBDATA                    00000012
PPD$B_LCB_0                     00000012
PPD$B_LCB_LPORT                 00000010
PPD$B_LCB_NPORT                 0000000F
PPD$B_LCB_OPC                   00000011
PPD$B_LCB_PORT                  0000000E
PPD$B_OPC                       0000000E
PPD$B_PORT                      0000000C
PPD$B_PROTOCOL                  0000001A
PPD$B_RSTATE                    00000025
PPD$B_RST_PORT                  00000024
PPD$B_STATUS                    0000000D
PPD$B_SWFLAG                    0000000B
PPD$B_SYSTEMID                  00C00014
PPD$B_TYPE                      0000000A
PPD$C_LB_LENGTH                 00000046
PPD$C_LCB_DATA                  00000013
PPD$C_LENGTH                    00000012
PPD$C_MIN_DGSIZ                 00000050
PPD$K_LB_LENGTH                 00000046
PPD$K_LENGTH                    00000012
PPD$L_BLINK                     00000004
PPD$L_DG_DISC                   00000028
PPD$L_FLINK                     00000000
```

```
PPD$L_IN_VCD                    00000018
PPD$L_LBCRC                     00000042
PPD$L_P0_ACK                    00000010
PPD$L_P0_NAK                    00000014
PPD$L_P0_NRSP                   00000018
PPD$L_P1_ACK                    0000001C
PPD$L_P1_NAK                    00000020
PPD$L_P1_NRSP                   00000024
PPD$L_REC_BOFF                  00000028
PPD$L_REC_NAME                  00000024
PPD$L_RPORT_FCN                 00000020
PPD$L_RPORT_REV                 0000001C
PPD$L_RPORT_TYP                 00000018
PPD$L_SND_BOFF                  00000020
PPD$L_SND_NAME                  0000001C
PPD$L_ST_ADDR                   00000018
PPD$L_XCT_LEN                   00000018
PPD$Q_CURTIME                   00000048
PPD$Q_NODENAME                  00000040
PPD$Q_SWINCARN                  00000028
PPD$Q_XCT_ID                    00000010
PPD$T_HWTYPE                    00000030
PPD$T_SWTYPE                    00000020
PPD$T_SWVERS                    00000024
PPD$W_LCB_LEN7                  0000000C
PPD$W_LENGTH                    00000010
PPD$W_MASK                      00000010
PPD$W_MAXDG                     0000001C
PPD$W_MAXMSG                    0000001E
PPD$W_MTYPE                     00000012
PPD$W_M_VAL                     00000014
PPD$W_SIZE                      00000008
QUEUE_DELAY                     ********    X   01
RB$B_STATUS                     0000000B
RB$B_TYPE                       0000000A
RB$W_LENGTH                     0000002A
RB$L_BLINK                      00000004
RB$L_CB                         0000000C
RB$L_CDRP                       0000001C
RB$L_CDT                        00000014
RB$L_CTPBUF                     00000010
RB$L_DGBUF                      00000026
RB$L_FLINK                      00000000
RB$L_PDT                        00000018
RB$W_ACTCNT                     00000024
RB$W_DELAY                      00000020
RB$W_REPEAT                     00000022
RB$W_SIZE                       00000008
RB_M_CONN                   =   00000001
RB_V_CANCEL                 =   00000001
RESPONDER_NAME                  ********    X   01
RETURN_CTP_RESPONSE             00000072 RG     01
SB$B_SYSTEMID               =   00000018
SCS$ALLOC_RSPID                 ********    X   01
SCS$B_PPD                   =   FFFFFFE0
SCS$CONFIG_PTH                  ********    X   01
SCS$CONNECT                     ********    X   01
SCS$DISCONNECT                  ********    X   01
SCS$GW_MAXDG                    ********    X   01
SCS$GW_MAXMSG                   ********    X   01
SCS$K_APPL_BASE             =   00000000
SIZ...                      =   00000001
SOFTWARE_DEBUG              =   00000001
SS$_DISCONNECT              =   0000204C
SS$_REMRSRC                 =   0000206C
SYSAP$C_DISPPO              =   00000002
SYSAP$C_DISPQ               =   00000000
TQB$B_STATUS                    00000008
TQB$B_TYPE                      0000000A
TQB$K_LENGTH                    00000014
TQB$L_BLINK                     00000004
TQB$L_DUETIM                    0000000C
TQB$L_FLINK                     00000000
TQB$L_RB                        00000010
TQB$W_SIZE                      00000008
VA$S_VPN                        ********    X   01
VA$V_VPN                        ********    X   01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	000005AA (1450.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000360 (864.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	90	00:00:00.33	00:00:01.19
Command processing	133	00:00:00.59	00:00:01.16
Pass 1	552	00:00:24.20	00:00:54.61
Symbol table sort	5	00:00:02.64	00.00:04.09
Pass 2	229	00:00:04.42	00:00:05.77
Symbol table output	60	00:00:00.41	00:00:00.56
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1075	00:00:32.62	00:01:07.40

The working set limit was 2400 pages.
108840 bytes (213 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1641 non-local and 60 local symbols.
1237 source lines were read in Pass 1, producing 27 object records in Pass 2.
75 pages of virtual memory were used to define 65 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CYDRIVER]PALIB.MLB;1	3
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	29
DRB2:[SHULL.EVXCI.CYDRIVER]CYLIB.MLB;4	12
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	9
TOTALS (all libraries)	53

2135 GETS were required to define 53 macros.

There were no errors, warnings or information messages.

MACRO/LIS CYCMD+CYLIB/LIB+SYS$LIBRARY:LIB/LIB+CYDRIVER$DIR:PALIB/LIB

Table of contents

ZZ-CYDRIVER-6.0 CYINIT
CYINIT
V6-000

```
0000     1          .TITLE  CYINIT
0000     2          .IDENT  'V6-000'
0000     3
0000     4  ;*****************************************************************************
0000     5  ;
0000     6  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     9  ;
0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    15  ;
0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    18  ;
0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    21  ;
0000    22  ;*****************************************************************************
0000    23  ;
0000    24  ;+
0000    25  ;
0000    26  ; FACILITY:      VAX/VMS DIAGNOSTIC CLASS DRIVER
0000    27  ;
0000    28  ; ABSTRACT:      This module contains the unit initialization routine
0000    29  ;                for the CI responder class driver.
0000    30  ;
0000    31  ; AUTHOR:        Jim Klumpp  16-JUL-81
0000    32  ;
0000    33  ; MODIFIED BY:   Jim Klumpp   6-MAY-83
0000    34  ;
0000    35  ;       6-000    Dave Shull      07-July-1984
0000    36  ;                VMS V4 Modifications/Release
0000    37  ;
0000    38  ;-
0000    39
```

```
                     0000     41
           00000001  0000     42 SOFTWARE_DEBUG=1                        ; Enable assembly of debug code
                     0000     43
                     0000     44         .DEFAULT DISPLACEMENT,WORD
                     0000     45         .ENABLE SUPPRESSION
                     0000     46         .SBTTL  DEFINITIONS
                     0000     47
                     0000     48
                     0000     49 ;  Set PSECT to driver code:
                     0000     50
                00000000     51         .PSECT  $$$115_DRIVER,LONG
                     0000     52
                     0000     53
                     0000     54 ;  System definitions (LIB.MLB):
                     0000     55
                     0000     56         $CDRPDEF                        ; Define CDRP offsets
                     0000     57         $CRBDEF                         ; Define CRB offsets
                     0000     58         $UCBDEF                         ; Define UCB offsets
                     0000     59         $DYNDEF                         ; Define DYN Values
                     0000     60
                     0000     61
                     0000     62 ;  CYDRIVER definitions (CYLIB.MLB):
                     0000     63
                     0000     64         $CYCDRPDEF                      ; Define CY CDRP offsets
                     0000     65
                     0000     66
```

```
                     0000     68          .SBTTL  CY$INIT,        CYDRIVER initialization routine
                     0000     69 ;+
                     0000     70 ;
                     0000     71 ;  CY$INIT
                     0000     72 ;
                     0000     73 ;  This routine is run when the responder is loaded. It does the following:
                     0000     74 ;
                     0000     75 ;       - Sets the responder online
                     0000     76 ;       - Threads the responder's CRB onto the timer queue to allow
                     0000     77 ;         periodic wakeups. Set initial wakeup time to infinite value.
                     0000     78 ;       - Initializes the responders internal listheads
                     0000     79 ;       - Allocates one CDRP to be used if resource failures occur
                     0000     80 ;       - Issues an SCS LISTEN call
                     0000     81 ;
                     0000     82 ;  INPUTS:
                     0000     83 ;
                     0000     84 ;       R5      - Address of UCB
                     0000     85 ;
                     0000     86 ;-
                     0000     87
00000001             0000     88 INIT_FLAG:      .BLKB   1                       ; On powerfail, skip this routine.
                     0001     89                                                 ; i.e., initialize CYDRIVER only once.
                     0001     90
                     0001     91 CY$INIT::
                     0001     92
                     0001     93 ;****************************************************************************
                     0001     94 ;
                     0001     95 ;  software debug code
                     0001     96 ;
                     0001     97 ;       jsb     g^ini$brk
                     0001     98 ;
                     0001     99 ;  end software debug code
                     0001    100 ;
                     0001    101 ;****************************************************************************
                     0001    102
CO FB AF    00   E2 0001    103         BBSS    #0,INIT_FLAG,10$        ; Initialize CYDRIVER only once
                     0006    104
 0000'CF    55   D0 0006    105         MOVL    R5,CY$L_SAVED_UCB       ; Save UCB address
                     0008    106
            53   DD 0008    107         PUSHL   R3                      ; Save R3
   53    24 A5   D0 000D    108         MOVL    UCB$L_CRB(R5),R3        ; Get CRB address
 0000'CF    53   D0 0011    109         MOVL    R3,CY$L_SAVED_CRB       ; Save CRB address
   18 A3    01   CE 0016    110         MNEGL   #1,CRB$L_DUETIME(R3)    ; Set infinite dutime
      0000'CF    DE 001A    111         MOVAL   TIMER_INTR,-            ; Fill in address of timout routine
         1C A3          001E    112                 CRB$L_TOUTROUT(R3)
 00000000'GF    16   0020    113         JSB     G^IOC$THREADCRB         ; Allow timouts to occur
            53 8ED0   0026    114         POPL    R3                      ; Restore R3
                     0029    115
          003B    30   0029    116         BSBW    INIT_LISTHEADS          ; Initialize listheads
                     002C    117
   51    80 8F    9A 002C    118         MOVZBL  #CDRP$K_CY_LEN,R1       ; Move length of CDRP to R1
          0095    30 0030    119         BSBW    HIPL_ALLPOOL            ; Allocate the CDRP
        30 50    E9 0033    120         BLBC    R0,10$                  ; Error, leave unit offline, branch
   C8 A2    51    B0 0036    121         MOVW    R1,CDRP$W_CDRPSIZE(R2)  ; Set size field
            39    90 003A    122         MOVB    #DYN$C_CDRP,-           ; Set structure type
         CA A2          003C    123                 CDRP$B_CD_TYPE(R2)      ;
 0000'CF    52   D0 003E    124         MOVL    R2,CY$L_SPARE_CDRP      ; Save CDRP address for later
```

```
                                0043   125
                                0043   126          LISTEN  -                               ; Pend a listen to SCS
                                0043   127                  MSGADR=CY$CONNECT_REQ,-         ;   Connection input address
                                0043   128                  ERRADR=CY$CONNECT_ERR,-         ;   Error address
                                0043   129                  LPRNAM=RESPONDER_NAME           ;   Responder's name
                                005A   130
                     09 50   E9 005A   131          BLBC    R0,10$                          ; Failure, branch
                55   0000'CF   D0 005D   132          MOVL    CY$L_SAVED_UCB,R5               ; Get the UCB address
                          10   A8 0062   133          BISW2   #UCB$M_ONLINE,-                 ; Set responder ONLINE
                       64 A5        0064   134                  UCB$W_STS(R5)
                                0066   135
                             05 0066   136 10$:      RSB
                                0067   137
                                0067   138
                                0067   139
```

```
                                0067   141          .SBTTL  INIT_LISTHEADS, Initialize listheads
                                0067   142 ;+
                                0067   143 ;
                                0067   144 ;  INIT_LISTHEADS
                                0067   145 ;
                                0067   146 ;  Routine to initialize all the responder's list headers stored in
                                0067   147 ;  driver space. Currently, the listheads that are initialized are:
                                0067   148 ;
                                0067   149 ;       - Command execution list
                                0067   150 ;       - Connection list
                                0067   151 ;       - Timer queue
                                0067   152 ;       - Monitor list
                                0067   153 ;
                                0067   154 ;  INPUTS:
                                0067   155 ;
                                0067   156 ;  OUTPUTS:
                                0067   157 ;
                                0067   158 ;       All registers           - Preserved
                                0067   159 ;
                                0067   160 ;-
                                0067   161
                                0067   162 INIT_LISTHEADS:
                                0067   163
0000'CF   0000'CF   DE          0067   164          MOVAL   EXEC_LIST,EXEC_LIST     ; Initialize the execution list
0004'CF   0000'CF   DE          006E   165          MOVAL   EXEC_LIST,EXEC_LIST+4   ; pointers
                                0075   166
                      03        BB     0075   167          PUSHR   #^M<R0,R1>              ; Save R0,R1
                50    00'8F     9A     0077   168          MOVZBL  #MAX_NODES,R0           ; Initialize loop count
              51      0000'CF   DE     007B   169          MOVAL   CONN_LIST,R1            ; Point to top of node list
                      61  51    D0     0080   170 10$:     MOVL    R1,(R1)                 ; Initialize CB listhead flink
                    04 A1  51    D0     0083   171          MOVL    R1,4(R1)                ; Initialize CB listhead blink
                      51  08    C0     0087   172          ADDL2   #8,R1                   ; Point to next CB listhead
                        F3 50    F5     008A   173          SOBGTR  R0,10$                  ; Repeat for all CB listheads
                      03        BA     008D   174          POPR    #^M<R0,R1>              ; Restore registers
                                0090   175
0000'CF   0000'CF   DE          0090   176          MOVAL   MAP_QUEUE,MAP_QUEUE     ; Initialize the mapped buffer
0004'CF   0000'CF   DE          0097   177          MOVAL   MAP_QUEUE,MAP_QUEUE+4   ; queue
                                009E   178
0000'CF   0000'CF   DE          009E   179          MOVAL   MNT_MAP_QUEUE,MNT_MAP_QUEUE   ; Initialize the maintenance
0004'CF   0000'CF   DE          00A5   180          MOVAL   MNT_MAP_QUEUE,MNT_MAP_QUEUE+4 ; buffer map queue
                                00AC   181
0000'CF   0000'CF   DE          00AC   182          MOVAL   TIMER_QUEUE,TIMER_QUEUE ; Initialize the timer queue
0004'CF   0000'CF   DE          00B3   183          MOVAL   TIMER_QUEUE,TIMER_QUEUE+4
                                00BA   184
                                00BA   185          .IF DF  SOFTWARE_DEBUG          ; Assemble for software debug only
                                00BA   186
                  0000'CF   DE  00BA   187          MOVAL   MONITOR_LIST,-          ; Initialize the activity monitor
                  0000'CF       00BE   188                  MONITOR_LIST            ; list
                  0000'CF   DE  00C1   189          MOVAL   MONITOR_LIST,-
                  0004'CF       00C5   190                  MONITOR_LIST+4
                                00C8   191
                                00C8   192          .ENDC                           ; End of conditional assembly
                                00C8   193
                      05        00C8   194          RSB
                                00C9   195
                                00C9   196
```

```
                                00C8   198          .SBTTL  HIPL_ALLPOOL,   Allocate pool at high IPL
                                00C8   199 ;+
                                00C8   200 ;
                                00C8   201 ;  HIPL_ALLPOOL
                                00C8   202 ;
                                00C8   203 ;  This subroutine allocates nonpaged pool from IPL greater than
                                00C8   204 ;  fork IPL.  It is assumed taht the current IPL was reached by
                                00C8   205 ;  reasing IPL from a level less than, or equal to, IPL$SYNCH.
                                00C8   206 ;  Otherwise this code path might interrupt a pool allocation
                                00C8   207 ;  already in progress and corrupt pool.  In particular, this
                                00C8   208 ;  routine may be called as a result of a SYSGEN controller/unit
                                00C8   209 ;  init, INITADP controller/unit init, but not on power fail
                                00C8   210 ;  recovery.
                                00C8   211 ;
                                00C8   212 ; Inputs:
                                00C8   213 ;
                                00C8   214 ;       R1                          - Size of buffer to allocate
                                00C8   215 ;
                                00C8   216 ; Outputs:
                                00C8   217 ;
                                00C8   218 ;       R0                          - 0/1 for fail/success
                                00C8   219 ;       R1                          - Size of buffer actually allocated
                                00C8   220 ;       R2                          - Addr of buffer allocated
                                00C8   221 ;       All other registers         - Preserved
                                00C8   222 ;-
                                00C8   223
                                00C8   224 HIPL_ALLPOOL:
                                00C8   225
          7E    53    7D      00C8   226          MOVQ    R3,-(SP)                ; Save caller's registers
54  00000000'GF    DE      00CB   227          MOVAL   G^EXE$GL_NONPAGED,R4    ; Get addr of pool header area
          64    DD      00D2   228          PUSHL   (R4)                    ; Save IPL from header
64  00000000'8F    DB      00D4   229          MFPR    #PR$_IPL,(R4)           ; Put current IPL in header
    00000000'GF    16      00DB   230          JSB     G^EXE$ALONONPAGED       ; Allocate from nonpaged pool
          64  8ED0   00E1   231          POPL    (R4)                    ; Restore old IPL in header
          53    65    7D      00E4   232          MOVQ    (SP)+,R3                ; Restore caller's registers
                      05      00E7   233          RSB                             ; Return
                              00E8   234
                              00E8   235          .END
```

```
CDRP$B_CD_TYPE                  = 0000000A
CDRP$B_FLAGS                      00000040
CDRP$C_BT_LEN                   = 00000040
CDRP$C_CY_LEN                     00000080
CDRP$K_CY_LEN                     00000080
CDRP$L_BUFADR                     0000005C
CDRP$L_BUFLEN                     00000060
CDRP$L_CB_BL                      00000074
CDRP$L_CB_FL                      00000070
CDRP$L_MAPQ_BL                    0000007C
CDRP$L_MAPQ_FL                    00000078
CDRP$L_MNT_NAM                    00000064
CDRP$L_RB_BL                      0000006C
CDRP$L_RB_FL                      00000068
CDRP$T_CY_LBUFHNDL                00000044
CDRP$T_CY_RBUFHNDL                00000050
CDRP$W_CDRPSIZE                 = 00000008
CONN_LIST                         ********    X   01
CRB$L_DUETIME                   = 00000018
CRB$L_TOUTROUT                  = 0000001C
CY$CONNECT_ERR                    ********    X   01
CY$CONNECT_REQ                    ********    X   01
CY$INIT                           00000001 RG     01
CY$L_SAVED_CRB                    ********    X   01
CY$L_SAVED_UCB                    ********    X   01
CY$L_SPARE_CDRP                   ********    X   01
DYN$C_CDRP                      = 00000039
EXE$ALONONPAGED                   ********    X   01
EXE$GL_NONPAGED                   ********    X   01
EXEC_LIST                         ********    X   01
HIPL_ALLPOOL                      000000C8 R      01
INIT_FLAG                         00000000 R      01
INIT_LISTHEADS                    00000067 R      01
IOC$THREADCRB                     ********    X   01
MAP_QUEUE                         ********    X   01
MAX_NODES                         ********    X   01
MNT_MAP_QUEUE                     ********    X   01
MONITOR_LIST                      ********    X   01
PR$_IPL                           ********    X   01
RESPONDER_NAME                    ********    X   01
SCS$LISTEN                        ********   GX   01
SOFTWARE_DEBUG                  = 00000001
TIMER_INTR                        ********    X   01
TIMER_QUEUE                       ********    X   01
UCB$L_CRB                       = 00000024
UCB$M_ONLINE                    = 00000010
UCB$W_STS                       = 00000064
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	000000E8 (232.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000080 (128.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	92	00:00:00.33	00:00:01.16
Command processing	115	00:00:00.58	00:00:01.24
Pass 1	269	00:00:07.50	00:00:09.37
Symbol table sort	0	00:00:00.96	00:00:00.96
Pass 2	51	00:00:01.12	00:00:01.28
Symbol table output	5	00:00:00.07	00:00:00.07
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	537	00:00:10.59	00:00:14.10

The working set limit was 1350 pages.
33256 bytes (65 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 649 non-local and 3 local symbols.
235 source lines were read in Pass 1, producing 14 object records in Pass 2.
16 pages of virtual memory were used to define 14 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CYDRIVER]PALIB.MLB;1	0
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	5
DRB2:[SHULL.EVXCI.CYDRIVER]CYLIB.MLB;4	1
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	5
TOTALS (all libraries)	11

733 GETS were required to define 11 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS CYINIT+CYLIB/LIB+SYS$LIBRARY:LIB/LIB+CYDRIVER$DIR:PALIB/LIB
```

Table of contents

```
0000     1          .TITLE  CYINPUT
0000     2          .IDENT  'V6-000'
0000     3  ;*****************************************************************************
0000     4  ;
0000     5  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     6  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     7  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     8  ;
0000     9  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    10  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    11  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    12  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    13  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    14  ;
0000    15  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    16  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    17  ;
0000    18  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    19  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    20  ;
0000    21  ;*****************************************************************************
0000    22  ;
0000    23  ;+
0000    24  ;
0000    25  ; FACILITY:      VAX/VMS DIAGNOSTIC CLASS DRIVER
0000    26  ;
0000    27  ; ABSTRACT:      This module contains the message and datagram input
0000    28  ;                routines, and the connection request and connection
0000    29  ;                error routines for CI responder class driver.
0000    30  ;
0000    31  ; AUTHOR:        Jim Klumpp  16-JUL-81
0000    32  ;
0000    33  ; MODIFIED BY:   Jim Klumpp   6-MAY-83
0000    34  ;
0000    35  ;       6-000    Dave Shull      07-July-1984
0000    36  ;                VMS V4 Modifications/Release
0000    37  ;
0000    38  ;       4-003    Dave Shull      14-June-1984
0000    39  ;                Modified routine DISC_RSP_CONNS to leave the 0 on PAx0 as
0000    40  ;                an Ascii '0'.  This is needed for VMS V4.
0000    41  ;       4-002    Dave Shull
0000    42  ;                Changed routine CY$CONNECT_REQ to use Listening CDI to get
0000    43  ;                port number issuing the connect.
0000    44  ;
0000    45  ;       4-001    Dave Shull      9-Feb-1984
0000    46  ;                Modified use of SCS$K_APPL_BASE to SCS$K_APPL_BASE-SYS$B_PPD
0000    47  ;                as in VMS V4 all SCS$ symbols reference from the Application
0000    48  ;                Data area versus Packet Pointer.
0000    49  ;
0000    50  ;-
0000    51
```

```
                     0000     53
           00000001  0000     54 SOFTWARE_DEBUG=1                        ; Enable assembly of debug code
                     0000     55
                     0000     56         .DEFAULT DISPLACEMENT,WORD
                     0000     57         .ENABLE SUPPRESSION
                     0000     58         .SBTTL  DEFINITIONS
                     0000     59
                     0000     60
                     0000     61 ;  Set PSECT to driver code:
                     0000     62
                     00000000 63         .PSECT  $$$115_DRIVER,LONG
                     0000     64
                     0000     65
                     0000     66 ;  System definitions (LIB.MLB):
                     0000     67
                     0000     68         $CDRPDEF                        ; Define CDRP offsets
                     0000     69         $CDTDEF                         ; Define CDT offsets
                     0000     70         $IRPDEF                         ; Define IRP offsets
                     0000     71         $PBDEF                          ; Define PB offsets
                     0000     72         $PDTDEF                         ; Define PDT offsets
                     0000     73         $SCSDEF                         ; Define SCS offsets
                     0000     74         $SSDEF                          ; Define SS offsets
                     0000     75         $SYSAPDEF                       ; Define SYSAP offsets
                     0000     76         $DYNDEF                         ; Define DYN Values
                     0000     77
                     0000     78
                     0000     79 ;  PADRIVER definitions (PALIB.MLB):
                     0000     80
                     0000     81         $PAPDTDEF                       ; Define PA PDT offsets
                     0000     82         $PPDDEF                         ; Define PPD offsets
                     0000     83
                     0000     84
                     0000     85 ;  CYDRIVER definitions (CYLIB.MLB):
                     0000     86
                     0000     87         $CBDEF                          ; Define CB offsets
                     0000     88         $CTPDEF                         ; Define CTP symbols
                     0000     89         $CYCDRPDEF                      ; Define CYDRIVER CDRP extension
                     0000     90         $RBDEF                          ; Define RB offsets
                     0000     91
                     0000     92
                     0000     93
```

```
                        0000          .SBTTL  CY$CONNECT_REQ, Connect request routine
                        0000     ;+
                        0000     ;
                        0000     ;  CY$CONNECT_REQ
                        0000     ;
                        0000     ;  This routine is entered when a remote process attempts to connect to the
                        0000     ;  responder. Then connection data in the connect request is checked for
                        0000     ;  a remote process name of CTP$CONTROLLER or CTP$RESPONDER. If correct, a
                        0000     ;  connection block is allocated, queued to the connection list for the
                        0000     ;  appropriate remote node, and an ACCEPT is issued.
                        0000     ;
                        0000     ;  If the connection data does not match, a REJECT is issued.
                        0000     ;
                        0000     ;  INPUTS:
                        0000     ;
                        0000     ;       R2                      - Address of local process name field in
                        0000     ;                                 the connect request message
                        0000     ;       R3                      - Address of listening CDT
                        0000     ;       R4                      - Address of PDT
                        0000     ;       16(R2)                  - Address of remote process name field in
                        0000     ;                                 connect request message
                        0000     ;
                        0000     ;  OUTPUTS:
                        0000     ;
                        0000     ;       R0-R5                   - Destroyed
                        0000     ;       All other registers     - Preserved
                        0000     ;
                        0000     ;-
                        0000
                        0000     CY$CONNECT_REQ::
                        0000
                        0000            ASSUME  SCS$T_DST_PROC+16 EQ SCS$T_SRC_PROC
                        0000
             0C    BB   0000            PUSHR   #^M<R2,R3>              ; Save regs
     0000'CF 10    29   0002            CMPC3   #16,CONTROLLER_NAME,-   ; Remote process name
          10 A2         0007                    16(R2)                  ; CTP$CONTROLLER?
             0C    BA   0009            POPR    #^M<R2,R3>              ; Restore regs
             0D    13   000B            BEQL    10$                     ; Yes, branch
             0C    BB   000D            PUSHR   #^M<R2,R3>              ; Save regs
     0000'CF 10    29   000F            CMPC3   #16,RESPONDER_NAME,-    ; Remote process name
          10 A2         0014                    16(R2)                  ; CTP$RESPONDER?
             0C    BA   0016            POPR    #^M<R2,R3>              ; Restore regs
             6B    12   0018            BNEQ    100$                    ; No, reject
                        001A
          FFE3'    30   001A     10$:   BSBW    ALLOC_CB                ; Allocate a connection block
          69 50    E9   001D            BLBC    R0,150$                 ; No resources, branch
          10 A2    7D   0020            MOVQ    16(R2),-                ; Copy process name to CB l.o.
          32 A1         0023                    CBST_RPROCNAM(R1)
          18 A2    7D   0025            MOVQ    24(R2),-                ; Copy process name to CB h.o.
          3A A1         0028                    8+CB$T_RPROCNAM(R1)
                        002A
     50   20 A3    9A   002A            MOVZBL  CDT$B_RSTATION(R3),R0   ; Get remote port number         %[4-002]
          FFCF'    30   002E            BSBW    ENTER_CONN_LIST         ; Enter CB onto connection list
                        0031
       55    51    D0   0031            MOVL    R1,R5                   ; Save CB address
             01    88   0034            BISB2   #CB_M_CONN,-            ; Show CONNECT in progress for this
          0B A5         0036                    CB$B_STATUS(R5)         ; connection
```

```
                                0038   152          ACCEPT  -                               ; Accept the connection
                                0038   153                  MSGADR=CY$MSG_INPUT,-           ; Message input routine
                                0038   154                  DGADR=CY$DG_INPUT,-             ; Datagram input routine
                                0038   155                  ERRADR=CY$CONNECT_ERR,-         ; Error input routine
                                0038   156                  INITCR=#INIT_CREDIT,-           ; Initial credits
                                0038   157                  MINSCR=#MIN_SEND_CR,-           ; Minimum send credits
                                0038   158                  INITDG=#INIT_DG_BUF,-           ; Initial datagram buffers
                                0038   159                  CONDAT=CONNECT_DATA,-           ; Connection data
                                0038   160                  AUXSTR=(R5)                     ; Aux Struc - connect block
                                0067   161
                    01   8A    0067   162          BICB2   #CB_M_CONN,-                    ; CONNECT no longer in progress for
                  0B A5         0069   163                  CB$B_STATUS(R5)                 ; this connetion
                51  55   D0     006B   164          MOVL    R5,R1                           ; Restore connection block address
                  18 50   E9     006E   165          BLBC    R0,150$                         ; Accept call faile., branch
                                0071   166
                                0071   167          ASSUME  CB$L_CDT+4 EQ CB$L_PDT
                                0071   168
                0C0C D4   90    0071   169          MOVB    @PDT$L_PPR(R4),-                ; Get local port number
                  25 A1         0075   170                  CB$B_LPRTNUM(R1)                ;
             0C A1  53   7D     0077   171          MOVQ    R3,CB$L_CDT(R1)                 ; Copy CDT, PDT address to CB
                                007B   172
                                007B   173 ;
                                007B   174 ;  Place local port name in connection block
                                007B   175 ;
             50   1C A3   D0    007B   176          MOVL    CDT$L_PB(R3),R0                 ; Get LOCAL path block address
                  24 A0   D0    007F   177          MOVL    PB$T_LPORT_NAME(R0),-           ; Fill in local port name
                  2E A1         0082   178                  CB$T_LPRTNAM(R1)
                          05    0084   179          RSB
                                0085   180
                                0085   181 100$:    REJECT                                  ; Reject the connection
                          05    0088   182          RSB
                                0089   183
                                0089   184 ;
                                0089   185 ; Connect Failed.  Perform cleanup and issue a REJECT to clean SCS.
                                0089   186 ;
                 53   52   D0    0089   187 150$:   MOVL    R2,R3                           ; Put listen CDT in correct reg
                 50   61   0F    008C   188          REMQUE  (R1),R0                         ; Remove CB from connection list
       00000000'GF    16    008F   189          JSB     G^COM$DRVDEALMEM                ; Deallocate the connection block
                                0095   190          REJECT                                  ; Issue a reject to clean up SCS
                          05    0098   191          RSB
                                0099   192
```

```
                     0099   194          .SBTTL  CY$DISCONNECT, Disconnect routine
                     0099   195          .SBTTL  CY$CONNECT_ERR, Connection error routine
                     0099   196 ;+
                     0099   197 ;
                     0099   198 ;  CY$DISCONNECT
                     0099   199 ;  CY$CONNECT_ERR
                     0099   200 ;
                     0099   201 ;  This routine is called by SCS when a conne tion fails or when a remote
                     0099   202 ;  process issues a disconnect, or if a connection fails. Connections fail
                     0099   203 ;  when virtual circuits are closed.
                     0099   204 ;
                     0099   205 ;  All resources associated with the connection are cleaned up, including
                     0099   206 ;  request blocks, CDRP's, and mapped buffers, and a disconnect is issued
                     0099   207 ;  to SCS
                     0099   208 ;
                     0099   209 ;  In addition, the connection database is checked for existing connections
                     0099   210 ;  to controller processes. If no such connections exist, the responder cleans
                     0099   211 ;  up any connections it may have to remote responsers and attempts to start
                     0099   212 ;  the poller. The poller may have been stopped during CINT test activity. If
                     0099   213 ;  the responder is talking to the CI exerciser, the attempt to start the
                     0099   214 ;  poller will fail, but will not cause any harm.
                     0099   215 ;
                     0099   216 ;  INPUTS:
                     0099   217 ;
                     0099   218 ;       R0                      - Error stautus (SS$_DISCONNECT, SS$_VCBROKEN)
                     0099   219 ;       R1                      - Additional status (Disconnect reason or
                     0099   220 ;                                 VC broken reason)
                     0099   221 ;       R3                      - CDT address
                     0099   222 ;       R4                      - PDT address
                     0099   223 ;
                     0099   224 ;  OUTPUTS:
                     0099   225 ;
                     0099   226 ;       R0-R5                   - Destroyed
                     0099   227 ;       All other registers     - Preserved
                     0099   228 ;
                     0099   229 ;-
                     0099   230
                     0099   231 CY$DISCONNECT::
                     0099   232 CY$CONNECT_ERR::
                     0099   233
 52    50 A3    D0   0099   234          MOVL    CDT$L_AUXSTRUC(R3),R2   ; Get connection block address
       07       12   009D   235          BNEQ    10$                     ; If we have one, branch
                     009F   236          DISCONNECT                      ; Otherwise, issue a disconnect
                05   00A5   237          RSB                             ; Return to SCS
                     00A6   238
       50 A3    D4   00A6   239 10$:     CLRL    CDT$L_AUXSTRUC(R3)      ; Show connection block is gone in CDT
        FF54'   30   00A9   240          BSBW    CLEAN_CONN              ; Deallocate connection resources
     50   62    0F   00AC   241          REMQUE  (R2),R0                 ; Remove the CB from node list
00000000'GF    16   00AF   242          JSB     G^COM$DRVDEALMEM        ; Deallocate the connection block
                     00B5   243          DISCONNECT                      ; Issue a disconnect
                     00BB   244
        FF42'   30   00B6   245          BSBW    CHECK_CONN_CTLR         ; Check for ANY controller connections
     0F 50      E8   00BE   246          BLBS    R0,20$                  ; Still have controller conns, branch
        000D    30   00C1   247          BSBW    DISC_RSP_CONNS          ; Clean up any connection to responders
                     00C4   248          ENABLE_POLLER                   ; Start the configuration poller
                05   00D0   249 20$:     RSB
                     00D1   250
```

```
                00D1    251
                00D1    252
                00D1    253
```

```
                                00D1   255          .SBTTL  DISC_RSP_CONNS, Clean connections to responders
                                00D1   256 ;+
                                00D1   257 ;
                                00D1   258 ;  DISC_RSP_CONNS
                                00D1   259 ;
                                00D1   260 ;  This routine is called by the connection error routine when no connections
                                00D1   261 ;  to controllers exist. It attempts to clean up any connections that may
                                00D1   262 ;  exist to responders.
                                00D1   263 ;
                                00D1   264 ;  INPUTS:
                                00D1   265 ;
                                00D1   266 ;  OUTPUTS:
                                00D1   267 ;
                                00D1   268 ;       R0-R2                    - Destroyed
                                00D1   269 ;       All other registers      - Preserved
                                00D1   270 ;
                                00D1   271 ;-
                                00D1   272
                                00D1   273 DISC_RSP_CONNS:
                                00D1   274
                00F8 8F    BB   00D1   275          PUSHR   #^M<R3,R4,R5,R6,R7>     ; Save regs
             56   FF'8F    9A   00D5   276          MOVZBL  #MAX_NODES-1,R6         ; Get largest possible number of nodes
             57   FF'8F    9A   00D9   277 5$:      MOVZBL  #MAX_LOC_PORTS-1,R7     ; Get total possible local ports
                7E   56    D0   00DD   278 10$:     MOVL    R6,-(SP)                ; Pass remote port number
          7E 30414150 8F    D0   00E0   279          MOVL    #^A'PAA0',-(SP)         ; Pass local port name
             02 AE   57     80   00E7   280          ADDB    R7,2(SP)                ; Make 'PAA ' into 'PAB ', 'PAC ', etc.
                 FF12'      30   00EB   281          BSBW    CHECK_CONN_RSP          ; Connection to remote responder?
                5E   08     C0   00EE   282          ADDL    #8,SP                   ; Clean off stack
                52   51     D0   00F1   283          MOVL    R1,R2                   ; Get connection block address
                     10     13   00F4   284          BEQL    20$                     ; No connection, branch
                     00     E0   00F6   285          BBS     #CB_V_CONN,-            ; Branch if connect is in progress
          0B 0B A2               00F8   286                  CB$B_STATUS(R2),20$
                     01     E2   00FB   287          BBSS    #CB_V_DISC,-            ; Branch if disconnect in progress
          06 0B A2               00FD   288                  CB$B_STATUS(R2),20$
                 FEFD'      30   0100   289          BSBW    CLEAN_CONN              ; Throw away connection resources
                  000B      30   0103   290          BSBW    ISSUE_DISC              ; Issue the disconnect call
                04 57       F4   0106   291 20$:     SOBGEQ  R7,10$                  ; Repeat for all local ports
                   56       F4   0109   292          SOBGEQ  R6,5$                   ; Repeat for all possible remote ports
             00F8 8F       BA   010C   293          POPR    #^M<R3,R4,R5,R6,R7>     ; Restore regs
                           05   0110   294          RSB
                                0111   295
                                0111   296 ;  By making this a subroutine call, we place a return address in the
                                0111   297 ;  DISC_RSP_CONNS routine on the stack. This allows us to maintain control
                                0111   298 ;  when SCS suspends the disconnect call. Thus, we can issue a disconnect
                                0111   299 ;  call for all the possible connections responders without giving up control.
                                0111   300
                                0111   301 ISSUE_DISC:
                                0111   302
                                0111   303          ASSUME  CB$L_CDT+4 EQ CB$L_PDT
                                0111   304
             55   52       DO   0111   305          MOVL    R2,R5                   ; Save CB address
          53   0C A5       7D   0114   306          MOVQ    CB$L_CDT(R5),R3         ; Restore CDT, PDT addresses
               5C A3       D4   0118   307          CLRL    CDT$L_AUXSTRUC(R3)      ; Show connection block is gone in CDT
                           011B   308          DISCONNECT                      ; Issue the disconnect
             5C   65       DF   0121   309          PUSHAL  (R5),R0                 ; Pull the CB off the connection list
        00000000'GF        16   0124   310          JSB     G^COM$DRVDEALMEM        ; Deallocate the connection block
                           05   012A   311          RSB
```

```
                012B    312
                012B    313
                012B    314
```

```
                        012B   316          .SBTTL  CY$MSG_INPUT,   Message input routine
                        012B   317          .SBTTL  CY$DG_INPUT,    Datagram input routine
                        012B   318 ;+
                        012B   319 ;
                        012B   320 ;  CY$MSG_INPUT
                        012B   321 ;  CY$DG_INPUT
                        012B   322 ;
                        012B   323 ;  This routine is called by SCS when a message or datagram for the
                        012B   324 ;  responder is received. A request block and a CDRP are allocated and
                        012B   325 ;  initialized appropriately. The request block is linked to the end of the
                        012B   326 ;  execution list. If the execution list had previously been empty, the
                        012B   327 ;  mainline routine is JMPed into, and command processing begins. Otherwise,
                        012B   328 ;  control is returned to SCS - the mailine routine is eventually restarted
                        012B   329 ;  by the fork dispatcher.
                        012B   330 ;
                        012B   331 ;  If the message is a CTP activity request, the activity flag is checked to
                        012B   332 ;  determine if activity is already in progress. If so, a CTP response is
                        012B   333 ;  returned with the appropriate status (the responder is designed to process
                        012B   334 ;  only one activity request at any time).
                        012B   335 ;
                        012B   336 ;  If a resource failure occurs in the input routine, the CTP response is
                        012B   337 ;  returned to the controller using the spare CDRP that was allocated by the
                        012B   338 ;  initialization routine.
                        012B   339 ;
                        012B   340 ;  If the received message happens to be a CTP response (which occurs during
                        012B   341 ;  processing of an activity request), the response is checked for validity
                        012B   342 ;  and deallocated. This allows the processing of the activity request to be
                        012B   343 ;  resumed.
                        012B   344 ;
                        012B   345 ;  INPUTS:
                        012B   346 ;
                        012B   347 ;       R1                          - Message length (bytes of application data)
                        012B   348 ;       R2                          - Message address (start of application data)
                        012B   349 ;       R3                          - CDT address
                        012B   350 ;       R4                          - PDT address
                        012B   351 ;
                        012B   352 ;  OUTPUTS:
                        012B   353 ;
                        012B   354 ;       R0-R5                       - Destroyed
                        012B   355 ;       All other registers         - Preserved
                        012B   356 ;
                        012B   357 ;-
                        012B   358
                        012B   359 CY$MSG_INPUT::
                        012B   360 CY$DG_INPUT::
                        012B   361
          FED2'  30     012B   362          BSBW    ALLOC_RB                ; Allocate a request block
                 31     012E   363          BLBCW   R0,110$                 ; Failure, branch
          FEC9'  30     0134   364          BSBW    ALLOC_CDRP              ; Allocate a CDRP
                 31     0137   365          BLBCW   R0,100$                 ; Failure, branch
                        013D   366
                        013D   367          ASSUME  RB$L_CDT+4 EQ RB$L_PDT
                        013D   368
        5C A3    D0     013D   369          MOVL    CDT$L_AUXSTRUC(R3),-    ; Copy connection block address
          0C A1         0140   370                  RB$L_CB(R1)             ;
14 A1     53    7D     0142   371          MOVQ    R3,RB$L_CDT(R1)         ; Copy CDT, PDT addresses
10 A1     52    D0     0146   372          MOVL    R2,RB$L_CTPBUF(R1)      ; Copy CTP request address
```

```
                        014A  373
                        014A  374          $DISPATCH -                          ; Dispatch off the CTP opcode to
                        014A  375                  CTP$OPCODE(R2),-              ; code which fills in the delay
                        014A  376                  TYPE=B,-                      ; and repeat count in the request
                        014A  377                  <-                            ; block.
                        014A  378                  <CTP$FUNCTREQ,  10$>,-
                        014A  379                  <CTP$BUFMAPREQ, 10$>,-
                        014A  380                  <CTP$BUFUNMREQ, 10$>,-
                        014A  381                  <CTP$MOVBUFREQ, 20$>,-
                        014A  382                  <CTP$GENMSGREQ, 20$>,-
                        014A  383                  <CTP$GENDGRREQ, 20$>,-
                        014A  384                  <CTP$GENRSTREQ, 20$>,-
                        014A  385                  <CTP$GENSTRREQ, 20$>,-
                        014A  386                  <CTP$NOACTREQ,  10$>,-
                        014A  387                  <CTP$CONFIGREQ, 10$>,-
                        014A  388                  <CTP$COUNTSREQ, 10$>,-
                        014A  389                  <CTP$CONNECTREQ,10$>,-
                        014A  390                  <CTP$FINISHREQ, 10$>,-
                        014A  391                  <CTP$MBUFMAPREQ,10$>,-
                        014A  392                  <CTP$MBUFUNMREQ,10$>,-
                        014A  393                  <CTP$MOVMBUFREQ,20$>,-
                        014A  394                  <CTP$MSTATEREQ, 10$>,-
                        014A  395                  >
                        0170  396
                        0170  397          .IF DF SOFTWARE_DEBUG                ; Assemble for software debug only
                        0170  398
005E   31               0170  399          BRW     200$                         ; Invalid opcode, branch
                        0173  400
                        0173  401          .ENDC                                ; End of conditional assembly
                        0173  402
                        0173  403 ;  CTP requests with contain neither a repeat count or delay count execute
                        0173  404 ;  this code.
                        0173  405
22 A1   01   B0         0173  406 10$:     MOVW    #1,RB$W_REPEAT(R1)           ; Set repeat count to 1
      20 A1  B4         0177  407          CLRW    RB$W_DELAY(R1)               ; Set delay count to 0
       000A  31         017A  408          BRW     40$
                        017D  409
                        017D  410 ;  CTP requests that contain both a repeat and delay count execute the
                        017D  411 ;  following code. Both fields are copied from the CTP request to the
                        017D  412 ;  request block.
                        017D  413
      06 A2  B0         017D  414 20$:     MOVW    CTP$REPCOUNT(R2),-           ; Copy repeat count to RB
      22 A1             0180  415                  RB$W_REPEAT(R1)
      05 A2  9B         0182  416          MOVZBW  CTP$DELAY(R2),-              ; Copy delay count to RB
      20 A1             0185  417                  RB$W_DELAY(R1)
                        0187  418
0004'DF   61   0E       0187  419 40$:     INSQUE  (R1),@EXEC_LIST+4            ; Insert RB at end of execution list
51  0000'CF   D1        018C  420          CMPL    EXEC_LIST,R1                 ; Was execution list previously empty?
                        0191  421          BEQLW   MAINLINE                     ; Yes, branch to mainline routine
                  05    0196  422          RSB                                  ; Otherwise, return to SCS
                        0197  423
   50   51   D0         0197  424 100$:    MOVL    R1,R0                        ; Get request block address
00000000'GF   16        019A  425          JSB     G^COM$DRVDEALMEM             ; Deallocate the request block
                        01A0  426
62   40 8F   80         01A0  427 110$:    ADDB2   #64,CTP$OPCODE(R2)           ; Change request to response
      FD 8F  90         01A4  428          MOVB    #CTP$NORESOURCE,-            ; Set appropriate status
      05 A2             01A7  429                  CTP$STATUS(R2)
```

```
                55   0000'CF   D0  01A9   430          MOVL    CY$L_SPARE_CDRP,R5      ; Get spare CDRP
                  24 A5   53   D0  01AE   431          MOVL    R3,CDRP$L_CDT(R5)       ; Fill in CDT address
                                  01B2   432
                50   52   20   C3  01B2   433          SUBL3   #SCS$K_APPL_BASE - SCS$B_PPD,R2,R0
                                  01B6   434                                           ; Back up to top of MSG/DG buffer[4-001]
                          3B   91  01B6   435          CMPB    #DYN$C_CIDG,-           ; Was request sent as datagram?
                            0A A0         01B8   436          PPD$B_TYPE(R0)          ;
                             0B  13  01BA   437          BEQL    120$                    ; Yes, branch
                                  01BC   438
                  1C A5   52   D0  01BC   439 115$:    MOVL    R2,CDRP$L_MSG_BUF(R5)   ; Fill in CTP buffer address
                                  01C0   440          RECYCL_MSG_BUF                  ; Recycle the MSG buffer
                                  01C3   441          SEND_MSG_BUF                    ; Return the CTP response
                             05   01C6   442          RSB                             ; Return to SCS
                                  01C7   443
                     51   06   9A  01C7   444 120$:    MOVZBL  #CTP$STATUS+1,R1        ; Get application data length
                                  01CA   445          SEND_DG_BUF -                   ; Send the CTP response as a datagram
                                  01CA   446                  FLAG=#SYSAP$C_DISPQ     ; Return DG buffer to free queue
                             05   01D0   447          RSB                             ; Return to SCS
                                  01D1   448
                                  01D1   449
                                  01D1   450          .IF DF SOFTWARE_DEBUG           ; Assemble for software debug only
                                  01D1   451
                        FE2C'     30  01D1   452 200$:    BSBW    DEALLOC_MSG_DG          ; Deallocate the CTP request
                     50   51      D0  01D4   453          MOVL    R1,R0                   ; Get request block address
                00000000'GF      16  01D7   454          JSB     G^COM$DRVDEALMEM        ; Deallocate the request block
                     50   55      D0  01DD   455          MOVL    R5,R0                   ; Get CDRP address
                00000000'GF      16  01E0   456          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDRP
                             05   01E6   457          RSB                             ; Return to SCS
                                  01E7   458
                                  01E7   459          .ENDC                           ; End of conditional assembly
                                  01E7   460
                     50   51      D0  01E7   461 300$:    MOVL    R1,R0                   ; Get request block address
                00000000'GF      16  01EA   462          JSB     G^COM$DRVDEALMEM        ; Deallocate the request block
                 62   40 8F      80  01F0   463          ADDB2   #64,CTP$CPCODE(R2)      ; Change opcode to response
                      FC 8F      90  01F4   464          MOVB    #CTP$REFUSE,-           ; Set appropriate status
                      05 A2          01F7   465                  CTP$STATUS(R2)
                  1C A5   52   D0  01F9   466          MOVL    R2,CDRP$L_MSG_BUF(R5)   ; Save CTP response address in CDRP
                                  01FD   467          RECYCL_MSG_BUF                  ; Recycle the MSG buffer
                                  0200   468          SEND_MSG_BUF                    ; Return the CTP response
                     50   55      D0  0203   469          MOVL    R5,R0                   ; Get CDRP address
                00000000'GF      16  0206   470          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDRP
                             05   020C   471          RSB                             ; Return to SCS
                                  020D   472
                                  020D   473          .END
```

$$$CURSIZ	= 000001C4			CTP$BUFROFSET		0000001C
$$$NEWSIZ	= 000001D0			CTP$BUFTYPE		00000005
ALLOC_CB	********	X	01	CTP$BUFUNMREQ		00000002
ALLOC_CDRP	********	X	01	CTP$BUFUNMRSP		00000042
ALLOC_RB	********	X	01	CTP$CDATPREV		00000002
CB$B_BUFMAP_CNT	0000001C			CTP$CDATPTYPE		00000000
CB$B_LPRTNUM	00000025			CTP$CDATPVERS		00000001
CB$B_RPRTNUM	00000026			CTP$CFGPOSTS		0000000B
CB$B_STATUS	0000000B			CTP$CFGP1STS		0000000C
CB$B_TYPE	0000000A			CTP$CNTFLG		0000000A
CB$K_LENGTH	00000046			CTP$CNTRDISCDG		00000024
CB$L_BLINK	00000004			CTP$CNTRP0ACK		0000000C
CB$L_CDT	0000000C			CTP$CNTRP0NAK		00000010
CB$L_FLINK	00000000			CTP$CNTRP0NRSP		00000014
CB$L_PDT	00000010			CTP$CNTRP1ACK		00000018
CB$L_RB	00000042			CTP$CNTRP1NAK		0000001C
CB$Q_MAPQ	00000014			CTP$CNTRP1NRSP		00000020
CB$Q_MNTMAPQ	0000001D			CTP$CONFIGREQ		00000009
CB$T_LPRTNAM	0000002E			CTP$CONFIGRSP		00000049
CB$T_RPROCNAM	00000032			CTP$CONNECTREQ		0000000B
CB$W_SIZE	00000008			CTP$CONNECTRSP		0000004B
CB_M_CONN	= 00000001			CTP$COUNTSREQ		0000000A
CB_V_CONN	= 00000000			CTP$COUNTSRSP		0000004A
CB_V_DISC	= 00000001			CTP$DELAY		00000005
CDRP$B_FLAGS	00000040			CTP$EXTEND		00000008
CDRP$C_BT_LEN	= 00000040			CTP$FINISHREQ		0000000C
CDRP$C_CY_LEN	00000080			CTP$FINISHRSP		0000004C
CDRP$K_CY_LEN	00000080			CTP$FMASK		00000006
CDRP$L_BUFADR	00000058			CTP$FUNCTREQ		00000000
CDRP$L_BUFLEN	00000060			CTP$FUNCTRSP		00000040
CDRP$L_CB_BL	00000074			CTP$GENCONST		00000009
CDRP$L_CB_FL	00000070			CTP$GENDATA		0000000E
CDRP$L_CDT	= 00000024			CTP$GENDGRREQ		00000005
CDRP$L_MAPQ_BL	0000007C			CTP$GENDGRRSP		00000045
CDRP$L_MAPQ_FL	00000078			CTP$GENFUNCT		00000003
CDRP$L_MNT_NAM	00000064			CTP$GENLENGTH		0000000C
CDRP$L_MSG_BUF	= 0000001C			CTP$GENMSGREQ		00000004
CDRP$L_RB_BL	0000006C			CTP$GENMSGRSP		00000044
CDRP$L_RB_FL	00000068			CTP$GENRSTREQ		00000006
CDRP$T_CY_LBUFHNDL	00000044			CTP$GENRSTRSP		00000046
CDRP$T_CY_RBUFHNDL	00000050			CTP$GENSTRREQ		00000007
CDT$B_RSTATION	= 00000020			CTP$GENSTRRSP		00000047
CDT$L_AUXSTRUC	= 0000005C			CTP$IMAGEDATA		0000000E
CDT$L_PB	= 0000001C			CTP$MAXCMDOPC		00000011
CHECK_CONN_CTLR	********	X	01	CTP$MBUFMAPREQ		0000000D
CHECK_CONN_RSP	********	X	01	CTP$MBUFMAPRSP		0000004D
CLEAN_CONN	********	X	01	CTP$MBUFUNMREQ		0000000E
COM$DRVDEALMEM	********	X	01	CTP$MBUFUNMRSP		0000004E
CONNECT_DATA	********	X	01	CTP$MOVBUFREQ		00000003
CONTROLLER_NAME	********	X	01	CTP$MOVBUFRSP		00000043
CTP$ACTCOUNT	00000006			CTP$MOVETYPE		00000008
CTP$BUFLENGTH	0000000C			CTP$MOVMBUFREQ		0000000F
CTP$BUFLNAME	00000010			CTP$MOVMBUFRSP		0000004F
CTP$BUFLOFSET	00000014			CTP$MSTATEREQ		00000010
CTP$BUFMAPREQ	00000001			CTP$MSTATERSP		00000050
CTP$BUFMAPRSP	00000041			CTP$NOACTFLAG		0000000A
CTP$BUFRNAME	00000018			CTP$NOACTREQ		00000008

```
CTP$NOACTRSP                    00000048
CTP$NORESOURCE                = 000000FD
CTP$OPCODE                      00000000
CTP$OPEXPAND                    000000FF
CTP$OTHERNODE                   00000009
CTP$PKTMULT                     0000000B
CTP$PKTSIZ                      0000000A
CTP$REFERENCE                   00000001
CTP$REFUSE                    = 000000FC
CTP$REPCOUNT                    00000006
CTP$RESERV10                    0000000A
CTP$RESERV11                    0000000B
CTP$RESERV12                    0000000C
CTP$RESERV20                    00000014
CTP$RESERV5                     00000005
CTP$RESERV6                     00000006
CTP$RESERV7                     00000007
CTP$RESERV9                     00000009
CTP$REVISION                  = 00000000   G
CTP$STARTADR                    0000000C
CTP$STATUS                      00000005
CTP$VERSION                   = 00000003   G
CY$CONNECT_ERR                  00000099 RG     01
CY$CONNECT_REQ                  00000000 RG     01
CY$DG_INPUT                     0000012B RG     01
CY$DISCONNECT                   00000099 RG     01
CY$L_SPARE_CDRP                 ********   X    01
CY$MSG_INPUT                    0000012B RG     01
DEALLOC_MSG_DG                  ********   X    01
DISC_RSP_CONNS                  000000D1 R      01
DYN$C_CIDG                    = 0000003B
ENTER_CONN_LIST                 ********   X    01
EXEC_LIST                       ********   X    01
INIT_CREDIT                     ********   X    01
INIT_DG_BUF                     ********   X    01
ISSUE_DISC                      00000111 R      01
MAINLINE                        ********   X    01
MAX_LOC_PORTS                   ********   X    01
MAX_NODES                       ********   X    01
MIN_SEND_CR                     ********   X    01
MNT$POLLER_ON                   ********   X    01
PB$T_LPORT_NAME               = 00000024
PDT$B_DQIMAP                    000C0154
PDT$B_HSHUT_DG                  000001B0
PDT$B_MAX_PORT                  0000017C
PDT$B_NXT_PORT                  0000017E
PDT$B_P0_LBSTS                  00000180
PDT$B_P1_LBSTS                  00000181
PDT$B_PLOGMAP                   00000134
PDT$B_PORTMAP                   00000114
PDT$B_PORT_NUM                  0000017D
PDT$B_REQIDPS                   0000017F
PDT$C_LENGTH                  = 000000E4
PDT$C_PAREGBASE                 000000E4
PDT$C_PAREGEND                  00000110
PDT$C_PQB                     = 000001E0
PDT$L_CNF                       000000F4
PDT$L_CQ0                       000000F0
PDT$L_CQ1                       000000F4
PDT$L_DFQ                       000000FC
PDT$L_DFQHDR                    00000208
PDT$L_DGHDRSZ                   00000190
PDT$L_DGNETHD                   00000194
PDT$L_DQELOGOUT                 000002E0
PDT$L_GPTBASE                   0000022C
PDT$L_GPTLEN                    00000230
PDT$L_LBDG                      00000184
PDT$L_MAINTFCN                = 00000078
PDT$L_MFQ                       00000100
PDT$L_MFQHDR                    0000020C
PDT$L_MQELOGOUT                 00000320
PDT$L_MTC                       00000104
PDT$L_PFAR                      00000108
PDT$L_PMC                       000000E8
PDT$L_POLLERDUE                 0000018C
PDT$L_POOLDUE                   00000188
PDT$L_PPR                       0000010C
PDT$L_PS                        000000EC
PDT$L_PSR                       000000F8
PDT$L_RCLMSGBUF               = 00000048
PDT$L_REJECT                  = 0000004C
PDT$L_SENDDG                  = 00000058
PDT$L_SENDMSG                 = 0000005C
PDT$L_SPTBASE                   00000224
PDT$L_SPTLEN                    00000228
PDT$L_VBDT                      0000021C
PDT$L_VPQB                      00000218
PDT$Q_COMQ2                     000001F0
PDT$Q_COMQ3                     000001F8
PDT$Q_COMQBASE                  000001E0
PDT$Q_COMQH                     000001E8
PDT$Q_COMQL                     000001E0
PDT$Q_DFREEQ                    000001D0
PDT$Q_FORMPB                    00000174
PDT$Q_MFREEQ                    000001D8
PDT$Q_RSPQ                      00000200
PDT$Q_TEMP_RSPQ                 00000198
PDT$W_BDTLEN                    00000220
PDT$W_DQELEN                    00000210
PDT$W_LPORT_STS                 00000110
PDT$W_MQELEN                    00000214
PDT$W_PBCOUNT                   00000112
PPD$B_DEF_ST                    0000001C
PPD$B_FLAGS                     00C0000F
PPD$B_HWVERS                    00000034
PPD$B_LBDATA                    00000012
PPD$B_LCB_0                     00000012
PPD$B_LCB_LPORT                 00000010
PPD$B_LCB_NPORT                 0000000F
PPD$B_LCB_OPC                   00000011
PPD$B_LCB_PORT                  0000000E
PPD$B_OPC                       0000000E
PPD$B_PORT                      0000000C
PPD$B_PROTOCOL                  0000001A
```

PPD$B_RSTATE	00000025	RB$L_PDT	00000018		
PPD$B_RST_PORT	00000024	RB$W_ACTCNT	00000024		
PPD$B_STATUS	0000000D	RB$W_DELAY	00000020		
PPD$B_SWFLAG	00000008	RB$W_REPEAT	00000022		
PPD$B_SYSTEMID	00000014	RB$W_SIZE	00000008		
PPD$B_TYPE	0000000A	RESPONDER_NAME	********	X	01
PPD$C_LB_LENGTH	00000046	SCS$ACCEPT	********	GX	01
PPD$C_LCB_DATA	00000013	SCS$B_PPD	= FFFFFFE0		
PPD$C_LENGTH	00000012	SCS$DISCONNECT	********	X	01
PPD$C_MIN_DGSIZ	00000050	SCS$K_APPL_BASE	= 00000000		
PPD$K_LB_LENGTH	00000046	SCS$T_DST_PROC	= 00000004		
PPD$K_LENGTH	00000012	SCS$T_SRC_PROC	= 00000014		
PPD$L_BLINK	00000004	SIZ...	= 00000001		
PPD$L_DG_DISC	00000028	SOFTWARE_DEBUG	= 00000001		
PPD$L_FLINK	00000000	SYSAP$C_DISPQ	= 00000000		
PPD$L_IN_VCD	00000018				
PPD$L_LBCRC	00000042				
PPD$L_P0_ACK	00000010				
PPD$L_P0_NAK	00000014				
PPD$L_P0_NRSP	00000018				
PPD$L_P1_ACK	0000001C				
PPD$L_P1_NAK	00000020				
PPD$L_P1_NRSP	00000024				
PPD$L_REC_BOFF	00000028				
PPD$L_REC_NAME	00000024				
PPD$L_RPORT_FCN	00000020				
PPD$L_RPORT_REV	0000001C				
PPD$L_RPORT_TYP	00000018				
PPD$L_SND_BOFF	00000020				
PPD$L_SND_NAME	0000001C				
PPD$L_ST_ADDR	00000018				
PPD$L_XCT_LEN	00000018				
PPD$Q_CURTIME	00000048				
PPD$Q_NODENAME	00000040				
PPD$Q_SWINCARN	00000028				
PPD$Q_XCT_ID	00000010				
PPD$T_HWTYPE	00000030				
PPD$T_SWTYPE	00000020				
PPD$T_SWVERS	00000024				
PPD$W_LCB_LENZ	0000000C				
PPD$W_LENGTH	00000010				
PPD$W_MASK	00000010				
PPD$W_MAXDG	0000001C				
PPD$W_MAXMSG	0000001E				
PPD$W_MTYPE	00000012				
PPD$W_M_VAL	00000014				
PPD$W_SIZE	00000008				
RB$B_STATUS	0000000B				
RB$B_TYPE	0000000A				
RB$K_LENGTH	0000002A				
RB$L_BLINK	00000004				
RB$L_CB	0000000C				
RB$L_CDRP	00000010				
RB$L_CDT	00000014				
RB$L_CIPBUF	0000C010				
RB$L_DGBUF	00000026				
RB$L_FLINK	00000000				

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	0000020D (525.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000360 (864.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	90	00:00:00.33	00:00:01.16
Command processing	114	00:00:00.57	00:00:01.12
Pass 1	450	00:00:19.47	00:00:21.39
Symbol table sort	0	00:00:02.34	00:00:02.61
Pass 2	93	00:00:02.75	00:00:02.98
Symbol table output	34	00:00:00.25	00:00:00.36
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	787	00:00:25.75	00:00:29.65

The working set limit was 1950 pages.
87467 bytes (171 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1526 non-local and 26 local symbols.
473 source lines were read in Pass 1, producing 17 object records in Pass 2.
53 pages of virtual memory were used to define 43 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CYDRIVER]PALIB.MLB;1	3
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	14
DRB2:[SHULL.EVXCI.CYDRIVER]CYLIB.MLB;4	7
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	7
TOTALS (all libraries)	31

1934 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS CYINPUT+CYLIB/LIB+SYS$LIBRARY:LIB/LIB+CYDRIVER$DIR:PALIB/LIB
```

Table of contents

```
0000     1          .TITLE  CYMAINT
0000     2          .IDENT  'V6-000'
0000     3
0000     4  ;****************************************************************************
0000     5  ;
0000     6  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     9  ;
0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    15  ;
0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    18  ;
0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    21  ;
0000    22  ;****************************************************************************
0000    23  ;
0000    24  ;+
0000    25  ;
0000    26  ; FACILITY:     VAX/VMS DIAGNOSTIC CLASS DRIVER
0000    27  ;
0000    28  ; VERSION:      4-001
0000    29  ;
0000    30  ; ABSTRACT:     This module the maintenance command execution routines
0000    31  ;               for the CI responder class driver.
0000    32  ;
0000    33  ; AUTHOR:       Jim Klumpp  16-JUL-81
0000    34  ;
0000    35  ; MODIFIED BY:  Jim Klumpp   6-MAY-83
0000    36  ;
0000    37  ;       6-000   Dave Shull      07-July-1984
0000    38  ;               VMS V4 Modifications/Release
0000    39  ;
0000    40  ;       4-001   Dave Shull      22-Mar-1984
0000    41  ;               Removed $PAPBDEF
0000    42  ;
0000    43  ;-
0000    44
```

```
                     0000     46
00000001             0000     47 SOFTWARE_DEBUG=1                        ; Enable assemble of debug code
                     0000     48
                     0000     49         .DEFAULT DISPLACEMENT,WORD
                     0000     50         .ENABLE SUPPRESSION
                     0000     51         .SBTTL  DEFINITIONS
                     0000     52
                     0000     53
                     0000     54 ;  Set PSECT to driver code:
                     0000     55
                     0000     56         .PSECT  $$$115_DRIVER,LONG
                     0000     57
                     0000     58
                     0000     59 ;  System definitions (LIB.MLB):
                     0000     60
                     0000     61         $CDRPDEF                        ; Define CDRP offsets
                     0000     62         $CDTDEF                         ; Define CDT offsets
                     0000     63         $CIBHANDEF                      ; Define CI buffer handles
                     0000     64         $CRBDEF                         ; Define CRB offsets
                     0000     65         $DYNDEF                         ; Define DYN offsets
                     0000     66         $IRPDEF                         ; Define IRP offsets
                     0000     67         $PBDEF                          ; Define PB offsets
                     0000     68         $PDTDEF                         ; Define PDT offsets
                     0000     69         $SBDEF                          ; Define SB offsets
                     0000     70         $SCSDEF                         ; Define SCS offsets
                     0000     71         $SSDEF                          ; Define SS offsets
                     0000     72         $SYSAPDEF                       ; Define SYSAP offsets
                     0000     73
                     0000     74
                     0000     75 ;  PADRIVER definitions (PALIB.MLB):
                     0000     76
                     0000     77         $PAPDTDEF                       ; Define PA PDT offsets
                     0000     78         $PAMAINTDEF                     ; Define PA maint definitions
                     0000     79         $PPDDEF                         ; Define PPD offsets
                     0000     80
                     0000     81
                     0000     82 ;  CYDRIVER definitions (CYLIB.MLB):
                     0000     83
                     0000     84         $CBDEF                          ; Define CB offsets
                     0000     85         $CTPDEF                         ; Define CTP symbols
                     0000     86         $CYCDRPDEF                      ; Define CYDRIVER CDRP extension
                     0000     87         $RBDEF                          ; Define RB offsets
                     0000     88         $TQBDEF                         ; Define TQB offsets
                     0000     89
                     0000     90
                     0000     91
```

```
                        0000     93          .SBTTL  CY$NOACT,       Start/Stop poller routine
                        0000     94 ;+
                        0000     95 ;
                        0000     96 ;  CY$NOACT
                        0000     97 ;
                        0000     98 ;  This routine executes a CTP start/stop unsolicited activity request.
                        0000     99 ;  Currently, stopping unsolicited activity involves only shutting off
                        0000    100 ;  the poller. A flag in the CTP request specifies whether the poller
                        0000    101 ;  should be started or stopped. After stopping the poller, a CTP response
                        0000    102 ;  is returned to the controller.
                        0000    103 ;
                        0000    104 ;  WARNING - Stopping the poller will keep any new virtual circuits from
                        0000    105 ;  being established. In addition, any virtual circuits that are lost
                        0000    106 ;  after the poller is stopped are lost until the poller is restarted.
                        0000    107 ;
                        0000    108 ;  INPUTS:
                        0000    109 ;
                        0000    110 ;       R2      - CTP request address
                        0000    111 ;       R3      - RB address
                        0000    112 ;       R4      - PDT address
                        0000    113 ;       R5      - CDRP address
                        0000    114 ;
                        0000    115 ;  OUTPUTS:
                        0000    116 ;
                        0000    117 ;       R6-R14  - Preserved
                        0000    118 ;-
                        0000    119
                        0000    120 CY$NOACT::
                        0000    121
                        0000    122         ASSUME  CTP$NOACTON EQ 0
                        0000    123         ASSUME  CTP$SUCCESS EQ 0
                        0000    124
CA AB   05              0000    125         TSTB    CTP$NOACTFLAG(R2)       ; Turn activity on or off?
    0D  12              0003    126         BNEQ    10$                     ; Off, branch
                        0005    127
                        0005    128         ENABLE_POLLER                   ; Turn the poller on
    0B  11              0010    129         BRB     20$                     ; Skip
                        0012    130
                        0012    131 10$:    DISABLE_POLLER                  ; Turn the poller off
                        001D    132
05 A2   94              001D    133 20$:    CLRB    CTP$STATUS(R2)          ; Set success status
  FFDD' 31              0020    134         BRW     RETURN_CTP_RESPONSE     ; Return the response
                        0023    135
                        0023    136
                        0023    137
                        0023    138
```

```
                     0023   140          .SBTTL  CY$MNT_STATE,   Set port to maintenance state
                     0023   141 ;+
                     0023   142 ;
                     0023   143 ;  CY$MNT_STATE
                     0023   144 ;
                     0023   145 ;  This routine executes a CTP set maintenance state request. A flag in the
                     0023   146 ;  request specifies whether to go to the /maintenance state or whether to
                     0023   147 ;  return from the /maintenance state. If the port is already in the state
                     0023   148 ;  specified in the request, CTP$SUCCESS status is returned. Otherwise, a
                     0023   149 ;  response with CTP$GOTO status is returned, and then the port is put into
                     0023   150 ;  the appropriate state.
                     0023   151 ;
                     0023   152 ;  A one second delay is performed between returning a response and changing
                     0023   153 ;  the port state so that the response is guaanteed to be received by the
                     0023   154 ;  remote node. The delay is performed by running through this routine twice,
                     0023   155 ;  the first time to send the response, the second to change the port state.
                     0023   156 ;
                     0023   157 ;  WARNING - All virtual circuits (and therefore all connections) to the local
                     0023   158 ;  port will be lost.
                     0023   159 ;
                     0023   160 ;  To determine the state in which the local port currently resides, the port
                     0023   161 ;  I/O space at location ^X950 is checked. Bit 8 specifies whether the port
                     0023   162 ;  is in /maint or /nomaint (1 or 0).
                     0023   163 ;
                     0023   164 ;  INPUTS:
                     0023   165 ;
                     0023   166 ;       R2              - CTP request address
                     0023   167 ;       R3              - RB address
                     0023   168 ;       R4              - PDT address
                     0023   169 ;       R5              - CDRP address
                     0023   170 ;
                     0023   171 ;  OUTPUTS:
                     0023   172 ;
                     0023   173 ;       R0-R5           - Destroyed
                     0023   174 ;
                     0023   175 ;-
                     0023   176
                     0023   177 CY$MNT_STATE::
                     0023   178
                     0023   179          ASSUME  CTP$SUCCESS EQ 0
                     0023   180
          00E4 C4    C1   0023   181          ADDL3   PDT$L_CNF(R4),-         ; Get address of maint register
50   00000950 8F         0027   182                  #^X950,R0
                     0027   183
        08 A2    91      002D   184          CMPB    CTP$GENFUNCT(R2),-      ; Are we being requested to go to
           00            0030   185                  #CTP$MAINTSTATE         ; /maint state
           39    12      0031   186          BNEQ    20$                     ; No, branch
                     0033   187
     2F 60    08   E0   0033   188          BBS     #8,(R0),10$             ; Branch if already in /maint
           02    E2      0037   189          BBSS    #RB_V_FPD,-             ; Branch if response has been returned
        23 0B A3         0039   190                  RB$B_STATUS(R3),5$      ; already
                     003C   191          ALLOC_MSG_BUF                       ; Allocate another message buffer
     50    10 A3   D0    003F   192          MOVL    RB$L_CTPBUF(R3),R0      ; Get CTP request address
     60    40 8F   81    0043   193          ADDB3   #64,CTP$OPCODE(R0),-    ; Fill in CTP response opcode
           62            0047   194                  CTP$OPCODE(R2)
        01 A0    D0    0048   195          MOVL    CTP$REFERENCE(R0),-     ; Copy command reference from CTP
        01 A2          004B   196                  CTP$REFERENCE(R2)       ; request to response
```

```
                05 A2   01   90  004D   197          MOVB    #CTP$GOTO,CTP$STATUS(R2); Show we are going to /maint
                                0051   198          SEND_MSG_BUF                    ; Return the CTP response
                52   10 A3   D0  0054   199          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
                20 A3   02   B0  0058   200          MOVW    #2,RB$W_DELAY(R3)       ; Set up delay for 1 second
                      FFA1'  31  005C   201          BRW     NEXT_ENTRY
                                005F   202
                                005F   203 5$:       SET_MAINT_STATE                 ; Put port in /maint state
                             05 0065   204          RSB                             ; Connection error code has cleaned up
                                0066   205
                          05 A2  94  0066   206 10$:      CLRB    CTP$STATUS(R2)          ; Set successful status
                           FF94' 31  0069   207          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                                006C   208
                F6 60    08   E1  006C   209 20$:      BBC     #8,(R0),10$             ; Branch if already in /nomaint
                         02   E2  0070   210          BBSS    #RB_V_FPD,-             ; Branch if response has been returned
                  23 0B A3          0072   211                  RB$B_STATUS(R3),25$     ; already
                                0075   212          ALLOC_MSG_BUF                   ; Allocate another message buffer
                50   10 A3   D0  0078   213          MOVL    RB$L_CTPBUF(R3),R0      ; Get CTP request address
                60   40 8F   81  007C   214          ADDB3   #64,CTP$OPCODE(R0),-    ; Fill in CTP response opcode
                         62          0080   215                  CTP$OPCODE(R2)
                      01 A0   D0  0081   216          MOVL    CTP$REFERENCE(R0),-     ; Copy command reference from CTP
                      01 A2          0084   217                  CTP$REFERENCE(R2)       ; request to response
                05 A2   01   90  0086   218          MOVB    #CTP$GOTO,CTP$STATUS(R2); Show we are going to /nomaint
                                008A   219          SEND_MSG_BUF                    ; Return the CTP response
                52   10 A3   D0  008D   220          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
                20 A3   02   B0  0091   221          MOVW    #2,RB$W_DELAY(R3)       ; Set up delay for 1 second
                      FF68'  31  0095   222          BRW     NEXT_ENTRY
                                0098   223
                                0098   224 25$:       CLR_MAINT_STATE                 ; Put port in /nomaint state
                             05 009E   225          RSB                             ; Connection error code has cleaned up
                                009F   226
                                009F   227
```

```
                                009F    229          .SBTTL  CY$MNT_BUFMAP,  Map a maintenance buffer
                                009F    230 ;+
                                009F    231 ;
                                009F    232 ;  CY$MNT_BUFMAP
                                009F    233 ;
                                009F    234 ;  Routine to map a maintenance buffer. Maintenance buffers are not associated
                                009F    235 ;  with a particular connection. Thus, they remained mapped until specifically
                                009F    236 ;  unmapped by the controller.
                                009F    237 ;
                                009F    238 ;  Unlike standard buffers, maintenance buffers are dsescribed by the PHYSICAL
                                009F    239 ;  address of the start of the buffer. The field CTP$BUFLNAME refers to this
                                009F    240 ;  address.
                                009F    241 ;
                                009F    242 ;  INPUTS:
                                009F    243 ;
                                009F    244 ;       R2              - CTP request address
                                009F    245 ;       R3              - RB address
                                009F    246 ;       R4              - PDT address
                                009F    247 ;       R5              - CDRP address
                                009F    248 ;
                                009F    249 ;  OUTPUTS:
                                009F    250 ;
                                009F    251 ;       R0-R5           - Destroyed
                                009F    252 ;
                                009F    253 ;-
                                009F    254
                                009F    255 CY$MNT_BUFMAP::
                                009F    256
          FF5E'  30             009F    257          BSBW    ALLOC_MNT_CDRP          ; Allocate a maintenance type CDRP
        64 50    E9             00A2    258          BLBC    R0,150$                 ; Error, branch
                                00A5    259
51  0C A2  0C   C1             00A5    260          ADDL3   #12,CTP$BUFLENGTH(R2),R1; Get size of maint buffer to allocate
          FF53'  30             00AA    261          BSBW    ALLOC_BUFFER            ; Allocate the maint buffer
        50 50    E9             00AD    262          BLBC    R0,100$                 ; Error, branch
60 A5   51  0C   C3             00B0    263          SUBL3   #12,R1,CDRP$L_BUFLEN(R5); Save maint buffer length
   5C A5  0C A2  DE             00B5    264          MOVAL   12(R2),CDRP$L_BUFADR(R5); Save maint buffer address
      52  10 A3  D0             00BA    265          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
          FF3F'  30             00BE    266          BSBW    BUILD_BUFFER            ; Build the maint buffer
                                00C1    267
      50  0C A3  D0             00C1    268          MOVL    RB$L_CB(R3),R0          ; Get connection block address
          70 A5  0E             00C5    269          INSQUE  CDRP$L_CB_FL(R5),-      ; Queue CDRP to connection block
          1D A0                 00C8    270                  CB$Q_MNTMAPQ(R0)
          78 A5  0E             00CA    271          INSQUE  CDRP$L_MAPQ_FL(R5),-    ; Queue the maintenace CDRP on the
        0000'CF                 00CD    272                  MNT_MAP_QUEUE           ; maintenance buffer map queue
                                00D0    273
      50  5C A5  D0             00D0    274          MOVL    CDRP$L_BUFADR(R5),R0    ; Get maint buffer address
51  50  FE00 8F  AB             00D4    275          BICW3   #^XFE00,R0,R1           ; Save byte offset
          00'    EF             00DA    276          EXTZV   S^#VA$V_VPN,-           ; Extract the virtual page number
    50  50  00'                 00DC    277                  S^#VA$S_VPN,R0,R0
 53  00000000'9F    D0          00DF    278          MOVL    @#MMG$GL_SPTBASE,R3     ; Get base of system page table
       50  6340    DE          00E6    279          MOVAL   (R3)[R0],R0             ; Get system virtual address of PTE
50  60  15  00     EF           00EA    280          EXTZV   #0,#21,(R0),R0          ; Extract the page frame number
    50  50  09     78           00EF    281          ASHL    #9,R0,R0                ; Shift before inserting byte offset
  10 A2  51  50    C9           00F3    282          BISL3   R0,R1,CTP$BUFLNAME(R2)  ; Form physical address
          10 A2    D0           00F8    283          MOVL    CTP$BUFLNAME(R2),-      ; Save maintenance buffer physical
          64 A5                 00FB    284                  CDRP$L_MNT_NAM(R5)      ; address (name) in CDRP
          FF00'    31           00FD    285          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
```

```
                       0100   286
                       0100   287
     50   55      D0   0100   288 100$:   MOVL    R5,R0                   ; Get maintenance CDRP address
 00000000'GF      16   0103   289         JSB     G^COM$DRVDEALMEM        ; Deallocate it
                       0109   290
        FD 8F     90   0109   291 150$:   MOVB    #CTP$NORESOURCE,-       ; Set failure status
        05 A2          010C   292                 CTP$STATUS(R2)
        FEEF'     31   010E   293         BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                       0111   294
                       0111   295
                       0111   296
```

```
                                0111   298          .SBTTL  CY$MNT_BUFUNM,  Unmap a maintenance buffer
                                0111   299 ;+
                                0111   300 ;
                                0111   301 ;  CY$MNT_BUFUNM
                                0111   302 ;
                                0111   303 ;  Routine to unmap a maintenance buffer. Maintenance buffers are not
                                0111   304 ;  associated with a particular connection. Thus, they remained mapped until
                                0111   305 ;  specifically unmapped by the controller.
                                0111   306 ;
                                0111   307 ;  Unlike standard buffers, maintenance buffers are dsescribed by the PHYSICAL
                                0111   308 ;  address of the start of the buffer. The field CTP$BUFLNAME refers to this
                                0111   309 ;  address.
                                0111   310 ;
                                0111   311 ;  INPUTS:
                                0111   312 ;
                                0111   313 ;       R2              - CTP request address
                                0111   314 ;       R3              - RB address
                                0111   315 ;       R4              - PDT address
                                0111   316 ;       R5              - CDRP address
                                0111   317 ;
                                0111   318 ;  OUTPUTS:
                                0111   319 ;
                                0111   320 ;       R0-R5           - Destroyed
                                0111   321 ;
                                0111   322 ;-
                                0111   323
                                0111   324 CY$MNT_BUFUNM::
                                0111   325
     55    10 A2    DE          0111   326          MOVAL   CTP$BUFLNAME(R2),R5     ; Get address of maint buffer name
           FEE8'    30          0115   327          BSBW    FIND_MAINT_BUFFER       ; Find the mapped maintenance buffer
           20 50    E9          0118   328          BLBC    R0,100$                 ; Error, branch
                                0118   329
     50    5C A5    0C    C3    0118   330          SUBL3   #12,CDRP$L_BUFADR(R5),R0; Get maintenance buffer address
    00000000'GF     16          0120   331          JSB     G^COM$DRVDEALMEM        ; Deallocate the maintenance buffer
                                0126   332
     50    78 A5    0F          0126   333          REMQUE  CDRP$L_MAPQ_FL(R5),R0   ; Remove the maint CDRP from map queue
     50    70 A5    0F          012A   334          REMQUE  CDRP$L_CB_FL(R5),R0     ; Remove CDRP from connection block
     50    A0 A5    DE          012E   335          MOVAL   -IRP$K_CDRP(R5),R0      ; Back up to top of IRP/CDRP
    00000000'GF     16          0132   336          JSB     G^COM$DRVDEALMEM        ; Deallocate the maint CDRP
                                0138   337
           FEC5'    31          0138   338          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                                013B   339
                                013B   340
           F9 8F    90          013B   341 100$:    MOVB    #CTP$NOBUFMAPPED,-      ; Set appropriate status
           05 A2                013E   342                  CTP$STATUS(R2)
           FEBD'    31          0140   343          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                                0143   344
                                0143   345
                                0143   346
```

```
                        0143   348          .SBTTL  CY$MNT_MOVBUF,  Perform maintenance buffer XFR
                        0143   349 ;+
                        0143   350 ;
                        0143   351 ;  CY$MNT_MOVBUF
                        0143   352 ;
                        0143   353 ;  This routine performs a maintenance buffer transfer.
                        0143   354 ;
                        0143   355 ;
                        0143   356 ;  INPUTS:
                        0143   357 ;
                        0143   358 ;       R2              - CTP request address
                        0143   359 ;       R3              - RB address
                        0143   360 ;       R4              - PDT address
                        0143   361 ;       R5              - CDRP address
                        0143   362 ;
                        0143   363 ;  OUTPUTS:
                        0143   364 ;
                        0143   365 ;       R0-R5           - Destroyed
                        0143   366 ;
                        0143   367 ;-
                        0143   368
                        0143   369 CY$MNT_MOVBUF::
                        0143   370
    55  10 A2   DE      0143   371          MOVAL   CTP$BUFLNAME(R2),R5     ; Get local buffer name address
         FEB6'  30      0147   372          $SBW    FIND_MAPPED_BUFFER      ; Find the mapped buffer CDRP
        4A 50   E9      014A   373          BLBC    R0,100$                 ; Error, branch
                        014D   374
    51  34 A5   D0      014D   375          MOVL    CDRP$L_RBUFH_AD(R5),R1  ; Get remote buffer handle address
        18 A2   D0      0151   376          MOVL    CTP$BUFRNAME(R2),-      ; Copy physical address of maint
           61           0154   377                  CIBHAN$L_BOFF(R1)       ; buffer to CDRP
 04 A1     01   D0      0155   378          MOVL    #1,CIBHAN$L_BNAME(R1)   ; Show this is a maint buffer
        0C A2   D0      0159   379          MOVL    CTP$BUFLENGTH(R2),-     ; Fill in transaction length
        3C A5           015C   380                  CDRP$L_XCT_LEN(R5)      ; field
                        015E   381
                        015E   382 ;        CLRB    CDRP$B_FLAGS(R5)        ; Assume 512 byte packets
                        015E   383 ;        TSTB    CTP$PKTSIZ(R2)          ; 512 byte packets?
                        015E   384 ;        BEQL    25$                     ; Yes, branch
                        015E   385 ;        MOVB    #^X80,CDRP$B_FLAGS(R5)  ; Set up for 576 byte packets
                        015E   386 ;25$:
                        015E   387
    53  09 A2   90      015E   388          MOVB    CTP$OTHERNODE(R2),R3    ; Pass remote port number to SCS
        08 A2   91      0162   389          CMPB    CTP$MOVETYPE(R2),-      ; Maintenance read?
           00           0165   390                  #CTP$MNTMOVEFROM
           11   13      0166   391          BEQL    10$                     ; Yes, branch
                        0168   392
                        0168   393          WRITE_MAINT_DATA                ; Perform maintenance write
        28 50   E9      0174   394          BLBC    R0,150$                 ; Error, branch
           0F   11      0177   395          BRB     20$                     ; Skip maintenance read code
                        0179   396
                        0179   397 10$:     READ_MAINT_DATA                 ; Perform maintenance read
        17 50   E9      0185   398          BLBC    R0,150$                 ; Error, branch
                        0188   399
                        0188   400          ASSUME  CTP$SUCCESS EQ 0
                        0188   401
53   0000'CF    DO      0188   402 20$:     MOVL    EXEC_LIST,R3            ; Restore request block address
 52    10 A3   D0      018D   403          MOVL    RB$L_CTPBUF(R3),R2      ; Restore CTP request address
        05 A2   94      0191   404          CLRB    CTP$STATUS(R2)          ; Set success status
```

```
                FE69'   31  0194    405          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                            0197    406
                            0197    407
                F9 8F   90  0197    408 100$:    MOVB    #CTP$NOBUFMAPPED,-      ; Set appropriate status
                05 A2       019A    409                  CTP$STATUS(R2)
                FE61'   31  019C    410          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                            019F    411
                            019F    412
                FC 8F   90  019F    413 150$:    MOVB    #CTP$NORESOURCE,-       ; Set appropriate status
                05 A2       01A2    414                  CTP$STATUS(R2)
                FE59'   31  01A4    415          BRW     RETURN_CTP_RESPONSE     ; Return the CTP response
                            01A7    416
                            01A7    417
                            01A7    418          .END
```

```
$$$CURSIZ                       =  000001C4
$$$NEWSIZ                       =  000001D0
ALLOC_BUFFER                       ********    X    01
ALLOC_MNT_CDRP                     ********    X    01
BUILD_BUFFER                       ********    X    01
CB$B_BUFMAP_CNT                    0000001C
CB$B_LPRTNUM                       00000025
CB$B_RPRTNUM                       00000026
CB$B_STATUS                        0000000B
CB$B_TYPE                          0000000A
CB$K_LENGTH                        00000046
CB$L_BLINK                         00000004
CB$L_CDT                           0000000C
CB$L_FLINK                         00000000
CB$L_PDT                           00000010
CB$L_RB                            00000042
CB$Q_MAPQ                          00000014
CB$Q_MNTMAPQ                       0000001D
CB$T_LPRTNAM                       0000002E
CB$T_RPROCNAM                      00000032
CB$W_SIZE                          00000008
CDRP$B_FLAGS                       00000040
CDRP$C_BT_LEN                   =  00000040
CDRP$C_CY_LEN                      00000080
CDRP$K_CY_LEN                      00000080
CDRP$L_BUFADR                      0000005C
CDRP$L_BUFLEN                      00000060
CDRP$L_CB_BL                       00000074
CDRP$L_CB_FL                       00000070
CDRP$L_MAPQ_BL                     0000007C
CDRP$L_MAPQ_FL                     00000078
CDRP$L_MNT_NAM                     00000064
CDRP$L_RBUFH_AD                 =  00000034
CDRP$L_RB_BL                       0000006C
CDRP$L_RB_FL                       00000068
CDRP$L_XCT_LEN                  =  0000003C
CDRP$T_CY_LBUFHNDL                 00000044
CDRP$T_CY_RBUFHNDL                 00000050
CIBHAN$L_BNAME                  =  00000004
CIBHAN$L_BOFF                   =  00000000
COM$DRVDEALMEM                     ********    X    01
CTP$ACTCOUNT                       00000006
CTP$BUFLENGTH                      0000000C
CTP$BUFLNAME                       00000010
CTP$BUFLOFSET                      00000014
CTP$BUFMAPREQ                      00000001
CTP$BUFMAPRSP                      00000041
CTP$BUFRNAME                       00000018
CTP$BUFROFSET                      0000001C
CTP$BUFTYPE                        00000005
CTP$BUFUNMREQ                      00000002
CTP$BUFUNMRSP                      00000042
CTP$CDATPREV                       00000002
CTP$CDATPTYPE                      00000000
CTP$CDATPVERS                      00000001
CTP$CFGP0STS                       00000008
CTP$CFGP1STS                       0000000C
CTP$CNTFLG                         0000000A
CTP$CNTRDISCDG                     00000024
CTP$CNTRP0ACK                      0000000C
CTP$CNTRP0NAK                      00000010
CTP$CNTRP0NRSP                     00000014
CTP$CNTRP1ACK                      00000018
CTP$CNTRP1NAK                      0000001C
CTP$CNTRP1NRSP                     00000020
CTP$CONFIGREQ                      00000009
CTP$CONFIGRSP                      00000049
CTP$CONNECTREQ                     0000000B
CTP$CONNECTRSP                     0000004B
CTP$COUNTSREQ                      0000000A
CTP$COUNTSRSP                      0000004A
CTP$DELAY                          00000005
CTP$EXTEND                         00000008
CTP$FINISHREQ                      0000000C
CTP$FINISHRSP                      0000004C
CTP$FMASK                          00000006
CTP$FUNCTREQ                       00000000
CTP$FUNCTRSP                       00000040
CTP$GENCONST                       00000009
CTP$GENDATA                        0000000E
CTP$GENDGRREQ                      00000005
CTP$GENDGRRSP                      00000045
CTP$GENFUNCT                       00000008
CTP$GENLENGTH                      0000000C
CTP$GENMSGREQ                      00000004
CTP$GENMSGRSP                      00000044
CTP$GENRSTREQ                      00000006
CTP$GENRSTRSP                      00000046
CTP$GENSTRREQ                      00000007
CTP$GENSTRRSP                      00000047
CTP$GOTO                        =  00000001
CTP$IMAGEDATA                      0000000E
CTP$MAINTSTATE                  =  00000000
CTP$MAXCMDOPC                      00000011
CTP$MBUFMAPREQ                     0000000D
CTP$MBUFMAPRSP                     0000004D
CTP$MBUFUNMREQ                     0000000E
CTP$MBUFUNMRSP                     0000004E
CTP$MNTMOVEFROM                 =  00000000
CTP$MOVBUFREQ                      00000003
CTP$MOVBUFRSP                      00000043
CTP$MOVETYPE                       00000008
CTP$MOVMBUFREQ                     0000000F
CTP$MOVMBUFRSP                     00C0004F
CTP$MSTATEREQ                      00000010
CTP$MSTATERSP                      00000050
CTP$NOACTFLAG                      0000C00A
CTP$NOACTON                     =  00000000
CTP$NOACTREQ                       00000008
CTP$NOACTRSP                       00000048
CTP$NOBUFMAPPED                 =  000000F9
CTP$NORESOURCE                  =  000000FD
CTP$OPCODE                         00000000
CTP$OPEXPAND                       000000FF
```

```
CTP$OTHERNODE                   00000009
CTP$PKTMULT                     0000000B
CTP$PKTSIZ                      0000000A
CTP$REFERENCE                   00000001
CTP$REPCOUNT                    00000006
CTP$RESERV10                    0000000A
CTP$RESERV11                    0000000B
CTP$RESERV12                    0000000C
CTP$RESERV20                    00000014
CTP$RESERV5                     00000005
CTP$RESERV6                     00000006
CTP$RESERV7                     00000007
CTP$RESERV9                     00000009
CTP$REVISION                =   00000000  G
CTP$STARTADR                    0000000C
CTP$STATUS                      00000005
CTP$SUCCESS                 =   00000000
CTP$VERSION                 =   00000003  G
CY$MNT_BUFMAP                   0000009F RG    01
CY$MNT_BUFUNM                   00000111 RG    01
CY$MNT_MOVBUF                   00000143 RG    01
CY$MNT_STATE                    00000023 RG    01
CY$NOACT                        00000000 RG    01
EXEC_LIST                       ********   X   01
FIND_MAINT_BUFFER               ********   X   01
FIND_MAPPED_BUFFER              ********   X   01
IRP$K_CDRP                  =   00000060
MMG$GL_SPTBASE                  ********   X   01
MNT$CATCH_RST               =   00000007  G
MNT$CATCH_STR               =   00000008  G
MNT$MAINT_READ              =   00000005  G
MNT$MAINT_STATE             =   00000002  G
MNT$MAINT_WRITE             =   00000006  G
MNT$NORMAL_STATE            =   00000004  G
MNT$POLLER_OFF              =   00000000  G
MNT$POLLER_ON               =   00000001  G
MNT$UNINIT_STATE            =   00000003  G
MNT_MAP_QUEUE                   ********   X   01
NEXT_ENTRY                      ********   X   01
PDT$B_DQIMAP                    00000154
PDT$B_HSHUT_DG                  000001BC
PDT$B_MAX_PORT                  0000017C
PDT$B_NXT_PORT                  0000017E
PDT$B_PO_LBSTS                  00000180
PDT$B_P1_LBSTS                  00000181
PDT$B_PLOGMAP                   00000134
PDT$B_PORTMAP                   00000114
PDT$B_PORT_NUM                  0000017D
PDT$B_REQIDPS                   0000017F
PDT$C_LENGTH                =   000000E4
PDT$C_PAREGBASE                 000000E4
PDT$C_PAREGEND                  00000110
PDT$C_PQB                   =   000001E0
PDT$L_ALLOCDG               =   00000010
PDT$L_ALLOCMSG              =   00000014
PDT$L_CNF                       000000E4
PDT$L_CQ0                       000000F0
PDT$L_CQ1                       000000F4
PDT$L_DFQ                       000000FC
PDT$L_DFQHDR                    00000208
PDT$L_DGHDRSZ                   00000190
PDT$L_DGNETHD                   00000194
PDT$L_DQELOGOUT                 000002E0
PDT$L_GPTBASE                   0000022C
PDT$L_GPTLEN                    00000230
PDT$L_LBDG                      00000184
PDT$L_MAINTFCN              =   00000078
PDT$L_MFQ                       00000100
PDT$L_MFQHDR                    0000020C
PDT$L_MQELOGOUT                 00000320
PDT$L_MTC                       00000104
PDT$L_PFAR                      00000108
PDT$L_PMC                       000000E8
PDT$L_POLLERDUE                 0000018C
PDT$L_POOLDUE                   00000188
PDT$L_PPR                       0000010C
PDT$L_PS                        000000EC
PDT$L_PSR                       000000F8
PDT$L_SENDMSG               =   0000005C
PDT$L_SPTBASE                   00000224
PDT$L_SPTLEN                    00000228
PDT$L_VBDT                      0000021C
PDT$L_VPQB                      00000218
PDT$Q_COMQ2                     000001F0
PDT$Q_COMQ3                     000001F8
PDT$Q_COMQBASE                  000001E0
PDT$Q_COMQH                     000001E8
PDT$Q_COMQL                     000001E0
PDT$Q_DFREEQ                    000001D0
PDT$Q_FORMPB                    00000174
PDT$Q_MFREEQ                    000001D8
PDT$Q_RSPQ                      00000200
PDT$Q_TEMP_RSPQ                 00000198
PDT$W_BDTLEN                    00000220
PDT$W_DQELEN                    00000210
PDT$W_LPORT_STS                 00000110
PDT$W_MQELEN                    00000214
PDT$W_PBCOUNT                   00000112
PPD$B_DEF_ST                    0000001C
PPD$B_FLAGS                     0000000F
PPD$B_HWVERS                    00000034
PPD$B_LBDATA                    00000012
PPD$B_LCB_0                     00000012
PPD$B_LCB_LPORT                 00C00010
PPD$B_LCB_NPORT                 0000000F
PPD$B_LCB_OPC                   00000011
PPD$B_LCB_PORT                  0000000E
PPD$B_OPC                       0000000E
PPD$B_PORT                      0000000C
PPD$B_PROTOCOL                  0000001A
PPD$B_RSTATE                    00000025
PPD$B_RST_PORT                  00000024
PPD$B_STATUS                    0000000D
PPD$B_SWFLAG                    0000000B
```

```
PPD$B_SYSTEMID                  00000014
PPD$B_TYPE                      0000000A
PPD$C_LB_LENGTH                 00000046
PPD$C_LCB_DATA                  00000013
PPD$C_LENGTH                    00000012
PPD$C_MIN_DGSIZ                 00000050
PPD$K_LB_LENGTH                 00000046
PPD$K_LENGTH                    00000012
PPD$L_BLINK                     00000004
PPD$L_DG_DISC                   00000028
PPD$L_FLINK                     00000000
PPD$L_IN_VCD                    00000018
PPD$L_LBCRC                     00000042
PPD$L_PO_ACK                    00000010
PPD$L_PO_NAK                    00000014
PPD$L_PO_NRSP                   00000018
PPD$L_P1_ACK                    0000001C
PPD$L_P1_NAK                    00000020
PPD$L_P1_NRSP                   00000024
PPD$L_REC_BOFF                  00000028
PPD$L_REC_NAME                  00000024
PPD$L_RPORT_FCN                 00000020
PPD$L_RPORT_REV                 0000001C
PPD$L_RPORT_TYP                 00000018
PPD$L_SND_BOFF                  00000020
PPD$L_SND_NAME                  0000001C
PPD$L_ST_ADDR                   00000018
PPD$L_XCT_LEN                   00000018
PPD$Q_CURTIME                   00000048
PPD$Q_NODENAME                  00000040
PPD$Q_SWINCARN                  00000028
PPD$Q_XCT_ID                    00000010
PPD$T_HWTYPE                    00000030
PPD$T_SWTYPE                    00000020
PPD$T_SWVERS                    00000024
PPD$W_LCB_LEN7                  0000000C
PPD$W_LENGTH                    00000010
PPD$W_MASK                      00000010
PPD$W_MAXDG                     0000001C
PPD$W_MAXMSG                    0000001E
PPD$W_MTYPE                     00000012
PPD$W_M_VAL                     00000014
PPD$W_SIZE                      00000008
RB$B_STATUS                     0000000B
RB$B_TYPE                       0000000A
RB$K_LENGTH                     0000002A
RB$L_BLINK                      00000004
RB$L_CB                         0000000C
RB$L_CDRP                       0000001C
RB$L_CDT                        00000014
RB$L_CTPBUF                     00000010
RB$L_DGBUF                      00000026
RB$L_FLINK                      00000000
RB$L_PDT                        00000018
RB$W_ACTCNT                     00000024
RB$W_DELAY                      00000020
RB$W_REPEAT                     00000022
RB$W_SIZE                       00000008
RB_V_FPD                      = 00000002
RETURN_CTP_RESPONSE             ********    X   01
SIZ...                        = 00000001
SOFTWARE_DEBUG                = 00000001
TQB$B_STATUS                    0000000B
TQB$B_TYPE                      0000000A
TQB$K_LENGTH                    00000014
TQB$L_BLINK                     00000004
TQB$L_DUETIM                    0000000C
TQB$L_FLINK                     00000000
TQB$L_RB                        00000010
TQB$W_SIZE                      00000008
VA$S_VPN                        ********    X   01
VA$V_VPN                        ********    X   01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	000001A7 (423.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000360 (864.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	92	00:00:00.32	00:00:01.47
Command processing	115	00:00:00.62	00:00:01.46
Pass 1	453	00:00:19.30	00:00:22.51
Symbol table sort	0	00:00:02.47	00:00:02.47
Pass 2	79	00:00:02.66	00:00:02.90
Symbol table output	34	00:00:00.26	00:00:00.36
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	779	00:00:25.66	00:00:31.20

The working set limit was 1300 pages.
87986 bytes (172 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1600 non-local and 15 local symbols.
418 source lines were read in Pass 1, producing 15 object records in Pass 2.
50 pages of virtual memory were used to define 42 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CYDRIVER]PALIB.MLB;1	2
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	14
DRB2:[SHULL.EVXCI.CYDRIVER]CYLIB.MLB;4	12
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	7
TOTALS (all libraries)	35

1978 GETS were required to define 35 macros.

There were no errors, warnings or information messages.

MACRO/LIS CYMAINT+CYLIB/LIB+SYS$LIBRARY:LIB/LIB+CYDRIVER$DIR:PALIB/LIB

Table of contents

```
0000     1          .TITLE  CYMISC
0000     2          .IDENT  'V6-000'
0000     3
0000     4  ;******************************************************************************
0000     5  ;
0000     6  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     9  ;
0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    15  ;
0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    18  ;
0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    21  ;
0000    22  ;******************************************************************************
0000    23  ;
0000    24  ;+
0000    25  ;
0000    26  ; FACILITY:     VAX/VMS DIAGNOSTIC CLASS DRIVER
0000    27  ;
0000    28  ; ABSTRACT:     This module contains miscellaneous routines used by the
0000    29  ;               CI responder class driver.
0000    30  ;
0000    31  ; AUTHOR:       Jim Klumpp  16-JUL-81
0000    32  ;
0000    33  ; MODIFIED BY:  Jim Klumpp   6-MAY-83
0000    34  ;
0000    35  ;               Fred Roemer 15-JUL-83
0000    36  ;                       Added check for negative one in CTP$OTHERNODE
0000    37  ;                       field for COUNTS:
0000    38  ;
0000    39  ;       6-000   Dave Shull      07-July-1984
0000    40  ;               VMS V4 Modifications/Release
0000    41  ;
0000    42  ;       4-002   Dave Shull      9-Feb-1984
0000    43  ;               Modified use of SCS$K_APPL_BASE to SCS$K_APPL_BASE-SYS$B_PPD
0000    44  ;               as in VMS V4 all SCS$ symbols reference from the Application
0000    45  ;               Data area versus Packet Pointer.
0000    46  ;
0000    47  ;--
0000    48
```

```
                     0000     50
00000001             0000     51 SOFTWARE_DEBUG=1                        ; Enable assembly of debug code
                     0000     52
                     0000     53         .DEFAULT DISPLACEMENT,WORD
                     0000     54         .ENABLE  SUPPRESSION
                     0000     55         .SBTTL   DEFINITIONS
                     0000     56
                     0000     57
                     0000     58 ;  Set PSECT to driver code:
                     0000     59
                     0000     60         .PSECT   $$$115_DRIVER,LONG
                     0000     61
                     0000     62
                     0000     63 ;  System definitions (LIB.MLB):
                     0000     64
                     0000     65         $CDRPDEF                        ; Define CDRP offsets
                     0000     66         $CDTDEF                         ; Define CDT offsets
                     0000     67         $CIBHANDEF                      ; Define CI buffer handles
                     0000     68         $CRBDEF                         ; Define CRB offsets
                     0000     69         $DYNDEF                         ; Define DYN offsets
                     0000     70         $IRPDEF                         ; Define IRP offsets
                     0000     71         $PBDEF                          ; Define PB offsets
                     0000     72         $PDTDEF                         ; Define PDT offsets
                     0000     73         $SCSDEF                         ; Define SCS offsets
                     0000     74         $SSDEF                          ; Define SS offsets
                     0000     75
                     0000     76
                     0000     77 ;  PADRIVER definitions (PALIB.MLB):
                     0000     78
                     0000     79         $PPDDEF                         ; Define PPD offsets
                     0000     80
                     0000     81
                     0000     82 ;  CYDRIVER definitions (CYLIB.MLB):
                     0000     83
                     0000     84         $CBDEF                          ; Define CB offsets
                     0000     85         $CTPDEF                         ; Define CTP symbols
                     0000     86         $CYCDRPDEF                      ; Define CYDRIVER CDRP extension
                     0000     87         $RBDEF                          ; Define RB offsets
                     0000     88         $TQBDEF                         ; Define TQB offsets
                     0000     89
                     0000     90
                     0000     91
```

```
                                0000    93          .SBTTL  ALLOC_CB,       Allocate connection block
                                0000    94 ;+
                                0000    95 ;
                                0000    96 ;  ALLOC_CB
                                0000    97 ;
                                0000    98 ;  Routine to allocate a connection block from non-paged pool and fill
                                0000    99 ;  in the size and type fields. Also, the mapped buffer queue for the
                                0000   100 ;  connection block is initialized.
                                0000   101 ;
                                0000   102 ;  INPUTS:
                                0000   103 ;
                                0000   104 ;  OUTPUTS:
                                0000   105 ;
                                0000   106 ;       R0                      - Status code (0 or 1)
                                0000   107 ;       R1                      - Address of connection block
                                0000   108 ;       All other registers     - Preserved
                                0000   109 ;
                                0000   110 ;-
                                0000   111
                                0000   112 ALLOC_CB::
                                0000   113
          52   DD               0000   114          PUSHL   R2                      ; Save R2
51   46 8F   9A                 0002   115          MOVZBL  #CB$K_LENGTH,R1         ; Move length of CB to R1
       00C1   30                 0006   116          BSBW    ALLOC_BUFFER            ; Allocate the memory
     1B 50    E9                 0009   117          BLBC    R0,10$                  ; On error, leave
   51   52    D0                 000C   118          MOVL    R2,R1                   ; Move address of CB to R1
        01    90                 000F   119          MOVB    #DYN$C_CD_CDDB,-        ; Set structure type
  0A A1                         0011   120                  CB$B_TYPE(R1)
  14 A1       DE                0013   121          MOVAL   CB$Q_MAPQ(R1),-         ; Initialize the mapped buffer queue
  14 A1                         0016   122                  CB$Q_MAPQ(R1)
  14 A1       DE                0018   123          MOVAL   CB$Q_MAPQ(R1),-
  18 A1                         001B   124                  CB$Q_MAPQ+4(R1)
  1D A1       DE                001D   125          MOVAL   CB$Q_MNTMAPQ(R1),-      ; Initialize the maintenance buffer
  1D A1                         0020   126                  CB$Q_MNTMAPQ(R1)        ; queue
  1D A1       DE                0022   127          MOVAL   CB$Q_MNTMAPQ(R1),-
  21 A1                         0025   128                  CB$Q_MNTMAPQ+4(R1)
      52 8ED0                   0027   129 10$:     POPL    R2                      ; Restore R2
         05                     002A   130          RSB
                                002B   131
                                002B   132
```

```
                     002B   134          .SBTTL  ALLOC_RB,       Allocate request block
                     002B   135 ;+
                     002B   136 ;
                     002B   137 ;  ALLOC_RB
                     002B   138 ;
                     002B   139 ;  Routine to allocate a request block from non-paged pool and
                     002B   140 ;  fill in the size and type fields. All other fields in the buffer
                     002B   141 ;  are cleared.
                     002B   142 ;
                     002B   143 ;  OUTPUTS:
                     002B   144 ;
                     002B   145 ;       R0                          - Status code (0 or 1)
                     002B   146 ;       R1                          - Address of request block
                     002B   147 ;       All other registers         - Preserved
                     002B   148 ;
                     002B   149 ;-
                     002B   150
                     002B   151 ALLOC_RB::
                     002B   152
        52    DD     002B   153          PUSHL   R2                          ; Save R2
51      2A    9A     002D   154          MOVZBL  #RB$K_LENGTH,R1             ; Move length of CB to R1
   0097       30     0030   155          BSBW    ALLOC_BUFFER                ; Allocate the memory
  03 50       E9     0033   156          BLBC    R0,10$                      ; On error, leave
51    52      D0     0036   157          MOVL    R2,R1                       ; Move address of CB to R1
                     0039   158 ;        MOVB    #DYN$C_RB,RB$B_TYPE(R1) ; Set structure type
        52 8ED0     0039   159 10$:     POPL    R2                          ; Restore R2
             05     003C   160          RSB
                     003D   161
                     003D   162
                     003D   163
```

```
                   003D          .SBTTL  ALLOC_CDRP,     Allocate CDRP
                   003D  ;+
                   003D  ;
                   003D  ;  ALLOC_CDRP
                   003D  ;
                   003D  ;  Routine to allocate a class driver request packet from non-paged
                   003D  ;  pool. The size and type fields in the CDRP are filled in. The CDRP
                   003D  ;  CDT field is also filled in, and the CDRP address is copied to the
                   003D  ;  request block. All other CDRP fields are cleared.
                   003D  ;
                   003D  ;  INPUTS:
                   003D  ;
                   003D  ;       R1                      - Address of request block
                   003D  ;       R3                      - CDT address
                   003D  ;
                   003D  ;  OUTPUTS:
                   003D  ;
                   003D  ;       R0                      - Status code (0 or 1)
                   003D  ;       R5                      - Address of CDRP
                   003D  ;       All other registers     - Preserved
                   003D  ;
                   003D  ;-
                   003D
                   003D  ALLOC_CDRP::
                   003D
          06   BB  003D          PUSHR   #^M<R1,R2>              ; Save R1,R2
   51   80 8F  9A  003F          MOVZBL  #CDRP$K_CY_LEN,R1       ; Move length of CDRP to R1
        0084   30  0043          BSBW    ALLOC_BUFFER            ; Allocate the memory
       17 50   E9  0046          BLBC    R0,10$                  ; On error, leave
      55  52   D0  0049          MOVL    R2,R5                   ; Move address of CDRP to R5
   08 A5  51   B0  004C          MOVW    R1,CDRP$W_CDRPSIZE(R5)  ; Fill in CDRP size
          39   90  0050          MOVB    #DYN$C_CDRP,-           ; Set structure type
       0A A5       0052                  CDRP$B_CD_TYPE(R5)
          06   BA  0054          POPR    #^M<R1,R2>              ; Restore R1,R2
   1C A1  55   D0  0056          MOVL    R5,RB$L_CDRP(R1)        ; Save its address in RB
   24 A5  53   D0  005A          MOVL    R3,CDRP$L_CDT(R5)       ; Fill in the CDT field
          02   11  005E          BRB     20$                     ; R1,R2 are already restored
          06   BA  0060  10$:    POPR    #^M<R1,R2>              ; Restore R1,R2
               05  0062  20$:    RSB
                   0063
                   0063
                   0063
```

```
                                0063   208          .SBTTL  ALLOC_IRP_CDRP, Allocate IRP/CDRP
                                0063   209 ;+
                                0063   210 ;
                                0063   211 ;  ALLOC_IRP_CDRP
                                0063   212 ;
                                0063   213 ;  Routine to allocate an I/O request packet and class driver request packet
                                0063   214 ;  combination used for mapping buffers. The address of the allocated block
                                0063   215 ;  points to the division between the IRP and CDRP portions of the block.
                                0063   216 ;  Thus, when accessing this block, negative offsets access fields in the IRP
                                0063   217 ;  portion and positive offsets access fields in the CDRP extension. The size
                                0063   218 ;  of the entire block is inserted into the CDRP$W_IRP_SIZE field so that the
                                0063   219 ;  entire block can be easily deallocated.
                                0063   220 ;
                                0063   221 ;  The IRP/CDRP is linked onto the connection block over which the buffer map
                                0063   222 ;  request was received.
                                0063   223 ;
                                0063   224 ;  INPUTS:
                                0063   225 ;
                                0063   226 ;       R3                      - Address of request block
                                0063   227 ;
                                0063   228 ;  OUTPUTS:
                                0063   229 ;
                                0063   230 ;       R0                      - Status code (0 or 1)
                                0063   231 ;       R5                      - Address of IRP/CDRP (start of CDRP portion)
                                0063   232 ;       All other registers     - Preserved
                                0063   233 ;
                                0063   234 ;-
                                0063   235
                                0063   236 ALLOC_IRP_CDRP::
                                0063   237
                          06    BB  0063   238          PUSHR   #^M<R1,R2>              ; Save regs
                      51  E0 8F    9A  0065   239          MOVZBL  #<IRP$K_CDRP+ -         ; Move length of IRP/CDRP to R1
                                    0069   240                  CDRP$K_CY_LEN>,R1
                        005E        30  0069   241          BSBW    ALLOC_BUFFER            ; Allocate the memory
                       37 50        E9  006C   242          BLBC    R0,10$                  ; On error, leave
                          0A        90  006F   243          MOVB    #DYN$C_IRP,-            ; Set IRP structure type
                       0A A2            0071   244                  IRP$B_TYPE(R2)
                      55  60 A2    DE  0073   245          MOVAL   IRP$K_CDRP(R2),R5       ; Get CDRP base address
08 A5                 51  0060 8F    A3  0077   246          SUBW3   #IRP$K_CDRP,R1,-        ; Set CDRP portion size
                                    007E   247                  CDRP$W_CDRPSIZE(R5)
                          39        90  007E   248          MOVB    #DYN$C_CDRP,-           ; Set CDRP portion type
                       0A A5            0080   249                  CDRP$B_CD_TYPE(R5)
                      50  0C A3    D0  0082   250          MOVL    RB$L_CB(R3),R0          ; Get connection block address
                       70 A5    0E  0086   251          INSQUE  CDRP$L_CB_FL(R5),-      ; Queue IRP/CDRP to CB
                       14 A0            0089   252                  CB$Q_MAPQ(R0)
                       1C A0    96  008B   253          INCB    CB$B_BUFMAP_CNT(R0)     ; Increment mapped buffer count
                       78 A5    0E  008E   254          INSQUE  CDRP$L_MAPQ_FL(R5),-    ; Queue IRP/CDRP to map queue
                  0000'CF               0091   255                  MAP_QUEUE
                       14 A3    D0  0094   256          MOVL    RB$L_CDT(R3),-          ; Fill in the CDRP CDT field
                       24 A5            0097   257                  CDRP$L_CDT(R5)
                       44 A5    DE  0099   258          MOVAL   CDRP$T_CY_LBUFHNDL(R5),-; Fill in local and remote buffer
                       2C A5            009C   259                  CDRP$L_LBUFH_AD(R5)     ; handle addresses
                       50 A5    DE  009E   260          MOVAL   CDRP$T_CY_RBUFHNDL(R5),-
                       34 A5            00A1   261                  CDRP$L_RBUFH_AD(R5)
                      50  01    D0  00A3   262          MOVL    #SS$_NORMAL,R0          ; Set success status
                          06    BA  00A6   263 10$:     POPR    #^M<R1,R2>              ; Restore regs
                                05  00A8   264          RSB
```

```
                    00A9    265
                    00A9    266
                    00A9    267
                    00A9    268
```

```
                                00A9   270          .SBTTL  ALLOC_MNT_CDRP, Allocate maintenance CDRP
                                00A9   271 ;+
                                00A9   272 ;
                                00A9   273 ;  ALLOC_MNT_CDRP
                                00A9   274 ;
                                00A9   275 ;  This routine allocates a maintenance type CDRP. Maintenace CDRP's are
                                00A9   276 ;  the same size as IRP/CDRP's. The difference is that maintenace CDRP's
                                00A9   277 ;  are queued to a the maint map queue, rather than the standard map queue.
                                00A9   278 ;  The address returned is that of the CDRP portion of the IRP/CDRP.
                                00A9   279 ;
                                00A9   280 ;  INPUTS:
                                00A9   281 ;
                                00A9   282 ;  OUTPUTS:
                                00A9   283 ;
                                00A9   284 ;       R0                          - Status code (0 or 1)
                                00A9   285 ;       R5                          - Address of maintenance CDRP (if success)
                                00A9   286 ;       CDRP$L_LBUFH_AD(R5)         - Address of local buffer handle
                                00A9   287 ;       CDRP$L_RBUFH_AD(R5)         - Address of remote buffer handle
                                00A9   288 ;       All other registers         - Preserved
                                00A9   289 ;
                                00A9   290 ;-
                                00A9   291
                                00A9   292 ALLOC_MNT_CDRP::
                                00A9   293
                          06    BB    00A9   294          PUSHR   #^M<R1,R2>              ; Save regs
                    51    EF 8F   9A  00AB   295          MOVZBL  #<IRP$K_CDRP+-          ; Get length of maintenance CDRP
                                      00AF   296                  CDRP$K_CY_LEN>,R1
                        0018  30    00AF   297          BSBW    ALLOC_BUFFER            ; Allocate the memory
                       12 50  E9    00B2   298          BLBC    R0,10$                  ; Error, branch
                          0A  90    00B5   299          MOVB    #DYN$C_IRP,-            ; Set IRP structure type
                       0A A2        00B7   300                  IRP$B_TYPE(R2)
                    55 60 A2  DE    00B9   301          MOVAL   IRP$K_CDRP(R2),R5       ; Advance to base of CDRP
                       44 A5  DE    00BD   302          MOVAL   CDRP$T_CY_LBUFHNDL(R5),- ; Fill in local buffer handle
                       28 A5        00C0   303                  CDRP$L_LBUFH_AD(R5)     ; address
                       50 A5  DE    00C2   304          MOVAL   CDRP$T_CY_RBUFHNDL(R5),- ; Fill in remote buffer handle
                       34 A5        00C5   305                  CDRP$L_RBUFH_AD(R5)     ; address
                          06  BA    00C7   306 10$:      POPR    #^M<R1,R2>              ; Restore regs
                              05    00C9   307          RSB
                                      00CA   308
                                      00CA   309
                                      00CA   310
```

```
                                00CA   312          .SBTTL  ALLOC_BUFFER,   Allocate buffer from pool
                                00CA   313 ;+
                                00CA   314 ;
                                00CA   315 ;  ALLOC_BUFFER
                                00CA   316 ;
                                00CA   317 ;  Routine to allocate a buffer from non-paged pool. The size of the
                                00CA   318 ;  allocated buffer is rounded up to the nearest multiple of 10 hex.
                                00CA   319 ;
                                00CA   320 ;  INPUTS:
                                00CA   321 ;
                                00CA   322 ;       R1                          - Size of buffer to allocate
                                00CA   323 ;
                                00CA   324 ;  OUTPUTS:
                                00CA   325 ;
                                00CA   326 ;       R0                          - Status code (0 or 1)
                                00CA   327 ;       R1                          - Actual size of buffer allocated
                                00CA   328 ;       R2                          - Address of buffer
                                00CA   329 ;       All other registers         - Preserved
                                00CA   330 ;
                                00CA   331 ;-
                                00CA   332
                                00CA   333 ALLOC_BUFFER::
                                00CA   334
                         53   DD  00CA   335          PUSHL   R3                          ; Save R5
            00000000'GF   16  00CC   336          JSB     G^EXE$ALONONPAGED           ; Allocate the memory
                         53 8ED0  00D2   337          POPL    R3                          ; Restore R3
                      07 50   E9  00D5   338          BLBC    R0,10$                      ; On error, leave
                       0005   30  00D8   339          BSBW    CLEAR_BUFFER                ; Clear the buffer
                  08 A2   51   D0  00DB   340          MOVL    R1,IRP$W_SIZE(R2)           ; Set up size field
                          05   00DF   341 10$:     RSB
                               00F0   342
                               00E0   343
                               00E0   344
```

```
                                00E0   346          .SBTTL  CLEAR_BUFFER,   Clear entire buffer
                                00E0   347 ;+
                                00E0   348 ;
                                00E0   349 ;  CLEAR_BUFFER
                                00E0   350 ;
                                00E0   351 ;  Routine to clear a buffer which has been allocated from non-paged
                                00E0   352 ;  pool.
                                00E0   353 ;
                                00E0   354 ;
                                00E0   355 ;  INPUTS:
                                00E0   356 ;
                                00E0   357 ;       R1                      - Length of buffer (in bytes)
                                00E0   358 ;       R2                      - Address of buffer
                                00E0   359 ;
                                00E0   360 ;  OUTPUTS:
                                00E0   361 ;
                                00E0   362 ;       All registers           - Preserved
                                00E0   363 ;
                                00E0   364 ;-
                                00E0   365
                                00E0   366 CLEAR_BUFFER::
                                00E0   367
                          3F    BB 00E0   368          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
    62   51   00   FE AF     00    2C 00E2   369          MOVC5   #0,,#0,R1,(R2)          ; Clear the buffer
                          3F    BA 00E9   370          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore regs
                          05    00EB   371          RSB
                                00EC   372
                                00EC   373
                                00EC   374
```

```
                                00EC   376          .SBTTL  CLEAN_CONN,     Clean connection resources
                                00EC   377 ;+
                                00EC   378 ;
                                00EC   379 ;  CLEAN_CONN
                                00EC   380 ;
                                00EC   381 ;  Routine to deallocate all the resources associated with a connection.
                                00EC   382 ;  This include the following:
                                00EC   383 ;
                                00EC   384 ;       - Request blocks
                                00EC   385 ;       - Mapped buffers
                                00EC   386 ;       - Maintenance mapped buffers
                                00EC   387 ;
                                00EC   388 ;  THE CONNECTION BLOCK IS NOT DEALLOCATED.
                                00EC   389 ;
                                00EC   390 ;  INPUTS:
                                00EC   391 ;
                                00EC   392 ;       R2                          - Address of connection block
                                00EC   393 ;
                                00EC   394 ;  OUTPUTS:
                                00EC   395 ;
                                00EC   396 ;       R0,R1                       - Destroyed
                                00EC   397 ;       All other registers         - Preserved
                                00EC   398 ;
                                00EC   399 ;-
                                00EC   400
                                00EC   401 CLEAN_CONN::
                                00EC   402
                      38    BB  00EC   403          PUSHR   #^M<R3,R4,R5>           ; Save regs
                53  0000'CF   DE  00EE   404          MOVAL   EXEC_LIST,R3            ; Get top of execution list
                    50    53    D0  00F3   405          MOVL    R3,R0               ; Save it
                    53    63    D0  00F6   406 10$:     MOVL    (R3),R3             ; Get first/next RB in execution list
                    53    50    D1  00F9   407 15$:     CMPL    R0,R3               ; End of list?
                          10    13  00FC   408          BEQL    20$                 ; Yes, branch
                52    0C  A3    D1  00FE   409          CMPL    RB$L_CB(R3),R2      ; Is this RB for failed connection?
                          F2    12  0102   410          BNEQ    10$                 ; No, try next RB in execution list
                          63    DD  0104   411          PUSHL   (R3)                ; Save address of NEXT RB in list
                        004E    30  0106   412          BSBW    DEALLOC_RB          ; Clean request resources, dealloc RB
                          53  8ED0  0109   413          POPL    R3                  ; Restore address of next RB in list
                          EB    11  010C   414          BRB     15$                 ; Try next RB in execution list
                                    010E   415
                                    010E   416          ASSUME  CDRP$L_CB_FL+8 EQ CDRP$L_MAPQ_FL
                                    010E   417
                54    10  A2    D0  010E   418 20$:     MOVL    CB$L_PDT(R2),R4     ; Get PDT address
                55    14  B2    0F  0112   419 30$:     REMQUE  @CB$Q_MAPQ(R2),R5   ; Remove a mapped buffer CDRP from CB
                          17    1D  0116   420          BVS     40$                 ; Queue empty, branch
                55    08  A5    0F  0118   421          REMQUE  8(R5),R5            ; Remove it from the map queue
                55    88  A5    DE  011C   422          MOVAL   -CDRP$L_MAPQ_FL(R5),R5 ; Back up to top of CDRP
                        005E    30  0120   423          BSBW    CLEAN_CDRP          ; Clean out CDRP resources
                50    A0  A5    DE  0123   424          MOVAL   -IRP$R_CDRP(R5),R0  ; Point to top of IRP/CDRP
            00000000'GF   16  0127   425          JSB     G^COM$DRVDEALMEM    ; Deallocate the IRP/CDRP
                          E3    11  012D   426          BRB     30$                 ; Loop
                                    012F   427
                55    1D  B2    0F  012F   428 40$:     REMQUE  @CB$Q_MNTMAPQ(R2),R5 ; Remove a MNT mapped buffer CDRP
                          1F    1D  0133   429          BVS     50$                 ; Empty, branch
                55    08  A5    0F  0135   430          REMQUE  8(R5),R5            ; Remove it from the MNT map queue
                55    88  A5    DE  0139   431          MOVAL   -CDRP$L_MAPQ_FL(R5),R5 ; Back up to top of CDRP
          50    5C  A5    0C    C3  013D   432          SUBL3   #12,CDRP$L_BUFADR(R5),R0; Get MNT buffer address
```

```
                00000000'GF   16  0142   433          JSB     G^COM$DRVDEALMEM        ; Deallocate the maintenance buffer
                 50    A0 A5  DE  0148   434          MOVAL   -IRP$K_CDRP(R5),R0      ; Back up to top of IRP/CDRP
                00000000'GF   16  014C   435          JSB     G^COM$DRVDEALMEM        ; Deallocate the IRP/CDRP
                          DB  11  0152   436          BRB     40$                     ; Loop until MNT queue empty
                                  0154   437
                          38  BA  0154   438 50$:      POPR    #^M<R3,R4,R5>           ; Restore regs
                              05  0156   439          RSB
                                  0157   440
                                  0157   441
                                  0157   442
```

```
                        0157   444          .SBTTL  DEALLOC_RB,     Deallocate request block
                        0157   445 ;+
                        0157   446 ;
                        0157   447 ;  DEALLOC_RB
                        0157   448 ;
                        0157   449 ;  Routine to deallocate all the resources associated with a request block.
                        0157   450 ;  This includes the CTP request message or datagram buffer, the CDRP and
                        0157   451 ;  the request block itself.
                        0157   452 ;
                        0157   453 ;  The CDRP may be linked on a resource wait queue. If so, it must be removed
                        0157   454 ;  from the queue. The CDRP may also contain some resources itself such as
                        0157   455 ;  a message or datagram buffer, a response ID, and mapping resources. These
                        0157   456 ;  resources must also be returned.
                        0157   457 ;
                        0157   458 ;  Also, if the request block is linked into the execution list, it is
                        0157   459 ;  removed. The work count is decremented.
                        0157   460 ;
                        0157   461 ;  One special case is handled in this routine. If this request block is for
                        0157   462 ;  a third party connect request, and the third party connet is in progress,
                        0157   463 ;  do not deallocate the request block. It will be deallocated when the
                        0157   464 ;  suspended connect thread is resumed. By seing the CANCEL status bit set in
                        0157   465 ;  the request block, the resumed connect thread knows that the second party
                        0157   466 ;  connection went away while the connect thread was suspended.
                        0157   467 ;
                        0157   468 ;  INPUTS:
                        0157   469 ;
                        0157   470 ;       R3                      - Address of request block
                        0157   471 ;
                        0157   472 ;  OUTPUTS:
                        0157   473 ;
                        0157   474 ;       All registers           - Preserved
                        0157   475 ;
                        0157   476 ;-
                        0157   477
                        0157   478 DEALLOC_RB::
                        0157   479
         3F     BB      0157   480          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
   54    18 A3   7D      0159   481          MOVQ    RB$L_PDT(R3),R4        ; Restore PDT, CDRP address
         0021    30      015D   482          BSBW    CLEAN_CDRP             ; Clean up the CDRP resources
   52    10 A3   D0      0160   483          MOVL    RB$L_CTPBUF(R3),R2     ; Get CTP buffer to deallocate
         03      13      0164   484          BEQL    20$                    ; None, branch
         0071    30      0166   485          BSBW    DEALLOC_MSG_DG         ; Deallocate the MSG or DG buffer
         005B    30      0169   486 20$:     BSBW    DEALLOC_CDRP           ; Deallocate the CDRP
                         016C   487
                         016C   488
      50 63      0F      016C   489          REMQUE  (R3),R0                ; Remove the RB from execution list
         02      88      016F   490          BISB2   #RB_M_CANCEL,-         ; Show that the request block
      0B A0              0171   491                  RB$B_STATUS(R0)        ; is being deallocated
         00      E0      0173   492          BBS     #RB_V_CONN,-           ; If third party connect in progress,
   06 0B A0              0175   493                  RB$B_STATUS(R0),60$    ; branch
00000000'GF    16      0178   494 50$:     JSB     G^COM$DRVDEALMEM       ; Deallocate the request block
         3F     BA      017E   495 60$:     POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore regs
                05      0180   496          RSB
                        0181   497
                        0181   498
```

```
                        0181   500          .SBTTL  CLEAN_CDRP,     Clean CDRP resources
                        0181   501 ;+
                        0181   502 ;
                        0181   503 ;  CLEAN_CDRP
                        0181   504 ;
                        0181   505 ;  Routine to remove a CDRP from a resource wait queue (only if it is on one),
                        0181   506 ;  and deallocate all the resources associated with that CDRP. These resources
                        0181   507 ;  include message or datagram buffers, response ID's, and mapping resources.
                        0181   508 ;
                        0181   509 ;  The CDRP is NOT deallocated.
                        0181   510 ;
                        0181   511 ;  INPUTS:
                        0181   512 ;
                        0181   513 ;       R5                      - CDRP Address
                        0181   514 ;
                        0181   515 ;  OUTPUTS:
                        0181   516 ;
                        0181   517 ;       All registers           - Preserved
                        0181   518 ;
                        0181   519 ;-
                        0181   520
                        0181   521 CLEAN_CDRP::
                        0181   522
          07    BB      0181   523          PUSHR   #^M<R0,R1,R2>           ; Save registers
          65    D5      0183   524          TSTL    CDRP$L_FQFL(R5)         ; Check forward link
          16    13      0185   525          BEQL    10$                     ; Flink equal to zero, branch
       04 A5    D5      0187   526          TSTL    CDRP$L_FQBL(R5)         ; Check backward link
          11    13      018A   527          BEQL    10$                     ; Blink equal to zero, branch
    55    04 B5    D1   018C   528          CMPL    @CDRP$L_FQBL(R5),R5     ; Is CDRP in resource queue?
          0B    12      0190   529          BNEQ    10$                     ; No, branch
       55    65    0F   0192   530          REMQUE  (R5),R5                 ; Remove the CDRP from queue
       28 A5    D5      0195   531          TSTL    CDRP$L_RWCPTR(R5)       ; Any wait count
          03    13      0198   532          BEQL    10$                     ; No, branch
       28 B5    B7      019A   533          DECW    @CDRP$L_RWCPTR(R5)      ; Decrement wait count
                        019D   534
    52    10 A5    D0   019D   535 10$:     MOVL    CDRP$L_MSG_BUF(R5),R2   ; Get MSG/DG address
          07    13      01A1   536          BEQL    30$                     ; No buffer, branch
             0034    30   01A3   537          BSBW    DEALLOC_MSG_DG          ; Deallocate the MSG or DG buffer
                        01A6   538
          30 A5    D5   01A6   539 30$:     TSTL    CDRP$L_RSPID(R5)        ; Any RSPID?
          06    13      01A9   540          BEQL    40$                     ; No, branch
                        01AB   541          DEALLOC_RSPID                   ; Deallocate the response ID
                        01B1   542
          2?    A5      D5   01B1   543 40$:     TSTL    CDRP$L_LBUFH_AD(R5)     ; Any mapping resources?
          0F    13      01B4   544          BEQL    50$                     ; No, branch
                        01B6   545          UNMAP                           ; Unmap the buffer
    50    5?    A5      ??   01B9   546          SUBL3   #12,CDRP$L_BUFADR(R5),R0; Get unmapped buffer address
       00000000'GF      16   01BE   547          JSB     G^COM$DRVDEALMEM        ; Deallocate the buffer
                        01C4   548
          07    BA      01C4   549 50$:     POPR    #^M<R0,R1,R2>           ; Restore registers
                        01C6   550          RSB
                        01C7   551
                        01C?   552
                        01C?   553
```

```
                                01C7   555          .SBTTL  DEALLOC_CDRP,   Deallocate CDRP
                                01C7   556 ;+
                                01C7   557 ;
                                01C7   558 ;  DEALLOC_CDRP
                                01C7   559 ;
                                01C7   560 ;  Routine to deallocate a CDRP. The CDRP type is determined by the
                                01C7   561 ;  field CDRP$L_LBUFH_AD. If clear, the CDRP is assumed to be a normal
                                01C7   562 ;  sized CDRP. Otherwise, the CDRP is assumed to be a buffer transfer
                                01C7   563 ;  type CDRP, in which case the entire IRP/CDRP is deallocated.
                                01C7   564 ;
                                01C7   565 ;  INPUTS:
                                01C7   566 ;
                                01C7   567 ;       R5                          - CDRP address
                                01C7   568 ;
                                01C7   569 ;  OUTPUTS:
                                01C7   570 ;
                                01C7   571 ;       R0-R2                       - Destroyed
                                01C7   572 ;
                                01C7   573 ;-
                                01C7   574
                                01C7   575 DEALLOC_CDRP:
                                01C7   576
          28 A5    D5           01C7   577          TSTL    CDRP$L_LBUFH_AD(R5)     ; Normal type CDRP
             04    13           01CA   578          BEQL    10$                     ; Yes, branch
       55    A0 A5    DE        01CC   579          MOVAL   -IRP$K_CDRP(R5),R5      ; Back up to base of IRP
          50    55    D0        01D0   580 10$:     MOVL    R5,R0                   ; Prepare for call
00000000'GF       16  01D3   581          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDRP
                  05  01D9   582          RSB
                      01DA   583
                      01DA   584
                      01DA   585
```

```
                                01DA   587          .SBTTL  DEALLOC_MSG_DG, Deallocate MSG or DG buffer
                                01DA   588 ;+
                                01DA   589 ;
                                01DA   590 ;  DEALLOC_MSG_DG
                                01DA   591 ;
                                01DA   592 ;  Routine to deallocate a message or datagram buffer. The field PPD$B_TYPE
                                01DA   593 ;  is used to detemine the type of buffer to deallocate.
                                01DA   594 ;
                                01DA   595 ;  INPUTS:
                                01DA   596 ;
                                01DA   597 ;       R2                      - Address of MSG or DG buffer
                                01DA   598 ;       R4                      - PDT address
                                01DA   599 ;       R5                      - CDRP address
                                01DA   600 ;       PPD$B_TYPE(R2)          - DYN$C_CIMSG for message buffer
                                01DA   601 ;                               - DYN$C_CIDG  for datagram buffer
                                01DA   602 ;
                                01DA   603 ;  OUTPUTS:
                                01DA   604 ;
                                01DA   605 ;       R0-R2                   - Destroyed
                                01DA   606 ;       All other registers     - Preserved
                                01DA   607 ;
                                01DA   608 ;-
                                01DA   609
                                01DA   610 DEALLOC_MSG_DG::
                                01DA   611
     50    52    20    C3     01DA   612          SUBL3   #SCS$K_APPL_BASE - SCS$B_PPD,R2,R0
                                01DE   613                                          ; Back up to PPD$L_FLINK        [4-002]
                 3B    91     01DE   614          CMPB    #DYN$C_CIDG,-               ; Is this a DG or a MSG
               0A    A0        01E0   615                  PPD$B_TYPE(R0)
                 08    13     01E2   616          BEQL    20$                         ; Datagram, branch
     1C    A5    52    D0     01E4   617          MOVL    R2,CDRP$L_MSG_BUF(R5)       ; Save message buffer address
                                01E8   618          DEALLOC_MSG_BUF                     ; Deallocate the MSG buffer
                       05     01EB   619          RSB
                                01EC   620
                                01EC   621 20$:     DEALLOC_DG_BUF                      ; Deallocate the DG buffer
                       05     01EF   622          RSB
                                01F0   623
                                01F0   624
                                01F0   625
                                01F0   626
```

```
                        01F0   628          .SBTTL  CHECK_CONN_RSP, Check for connection to responder
                        01F0   629 ;+
                        01F0   630 ;
                        01F0   631 ;  CHECK_CONN_RSP
                        01F0   632 ;
                        01F0   633 ;  Routine to check for a connection to the specified remote responder through
                        01F0   634 ;  the specified local port, and return the conection block address if the
                        01F0   635 ;  connection exists.
                        01F0   636 ;
                        01F0   637 ;  Note that the local port name and remote port number are passed to this
                        01F0   638 ;  routine on the stack.
                        01F0   639 ;
                        01F0   640 ;  INPUTS:
                        01F0   641 ;
                        01F0   642 ;       (SP)                    - Return address
                        01F0   643 ;       4(SP)                   - Local port name
                        01F0   644 ;       8(SP)                   - Remote port number
                        01F0   645 ;
                        01F0   646 ;  OUTPUTS:
                        01F0   647 ;
                        01F0   648 ;       R0                      - Status
                        01F0   649 ;       R1                      - CB address if success
                        01F0   650 ;                               - 0 otherwise
                        01F0   651 ;
                        01F0   652 ;       All other registers     - Preserved
                        01F0   653 ;
                        01F0   654 ;-
                        01F0   655
                        01F0   656 CHECK_CONN_RSP::
                        01F0   657
   51    08 AE    9A    01F0   658          MOVZBL  8(SP),R1                ; Get remote port number
 50    0000'CF    DE    01F4   659          MOVAL   CONN_LIST,R0            ; Get address of port list
    50    6041    7E    01F9   660          MOVAQ   (R0)[R1],R0             ; Get listhead specified remote port
       51    60    D0    01FD   661          MOVL    (R0),R1                 ; Get first CB
       51    50    D1    0200   662 10$:     CMPL    R0,R1                   ; End of queue?
             19    13    0203   663          BEQL    20$                     ; Yes, branch
             0F    BB    0205   664          PUSHR   #^M<R0,R1,R2,R3>        ; Save registers
   32 A1     10    29    0207   665          CMPC3   #16,CB$T_RPRCNAM(R1),-  ; Is this a connection to
       0000'CF          020B   666                  RESPONDER_NAME          ; a responder?
             0F    BA    020E   667          POPR    #^M<R0,R1,R2,R3>        ; Restore registers
             07    12    0210   668          BNEQ    15$                     ; No, branch
04 AE     2E A1    D1    0212   669          CMPL    CB$T_LPRTNAM(R1),4(SP)  ; Compare local port names
             08    13    0217   670          BEQL    30$                     ; Match, branch
       51    61    D0    0219   671 15$:     MOVL    (R1),R1                 ; Get next CB in queue
             E2    11    021C   672          BRB     10$                     ; Loop
                         021E   673
             50    7C    021E   674 20$:     CLRQ    R0                      ; No connection found, return error
                   05    0220   675          RSB
                         0221   676
       50    01    90    0221   677 30$:     MOVB    #1,R0                   ; Connection found, return success
                   05    0224   678          RSB
                         0225   679
                         0225   680
```

```
                                0225   682          .SBTTL  CHECK_CONN_CTLR,Check for ANY connections to controllers
                                0225   683 ;+
                                0225   684 ;
                                0225   685 ;  CHECK_CONN_CTLR
                                0225   686 ;
                                0225   687 ;  Routine to check for ANY connections to controllers. If no connections
                                0225   688 ;  exist to controllers, then all connections to responders should be dropped.
                                0225   689 ;
                                0225   690 ;  INPUTS:
                                0225   691 ;
                                0225   692 ;  OUTPUTS:
                                0225   693 ;
                                0225   694 ;       R0                      - Status
                                0225   695 ;       All other registers     - Preserved
                                0225   696 ;
                                0225   697 ;-
                                0225   698
                                0225   699 CHECK_CONN_CTLR::
                                0225   700
                     3E   BB    0225   701          PUSHR   #^M<R1,R2,R3,R4,R5>     ; Save regs
           50    FF'8F   9A    0227   702          MOVZBL  #MAX_NODES-1,R0         ; Repeat for all remote ports
         51    0000'CF   DE    022B   703          MOVAL   CONN_LIST,R1            ; Get address of connection list
            52    6140   7E    0230   704 10$:     MOVAQ   (R1)[R0],R2             ; Get listhead for remote port
               53    52   D0    0234   705          MOVL    R2,R3                   ; Save listhead address
               53    63   D0    0237   706 20$:     MOVL    (R3),R3                 ; Get first/next CB on queue
               53    52   D1    023A   707          CMPL    R2,R3                   ; Back to start of list
                     0F   13    023D   708          BEQL    30$                     ; Try next remote port in cluster
                     0F   BB    023F   709          PUSHR   #^M<R0,R1,R2,R3>        ; Save registers
           32 A3     10   29    0241   710          CMPC3   #16,CB$T_RPROCNAM(R3),- ; Is this a connection to
              0000'CF         0245   711                  CONTROLLER_NAME          ; a controller?
                     0F   BA    0248   712          POPR    #^M<R0,R1,R2,R3>        ; Restore registers
                     09   13    024A   713          BEQL    40$                     ; Found a controller connection, branch
                     E9   11    024C   714          BRB     20$                     ; Try next CB for remote node
                  DF 50   F4    024E   715 30$:     SOBGEQ  R0,10$                  ; Repeat for all remote ports
                              0251   716
                     50   D4    0251   717          CLRL    R0                      ; No connections exist
                     03   11    0253   718          BRB     50$                     ; Skip success path
                              0255   719
                  50 01   D0    0255   720 40$:     MOVL    #SS$_NORMAL,R0          ; Return success status
                     3E   BA    0258   721 50$:     POPR    #^M<R1,R2,R3,R4,R5>     ; Restore regs
                     05   05    025A   722          RSB
                              025B   723
                              025B   724
                              025B   725
```

```
                        025B   727          .SBTTL  ENTER_CONN_LIST,Queue CB on connection list
                        025B   728 ;+
                        025B   729 ;
                        025B   730 ;  ENTER_CONN_LIST
                        025B   731 ;
                        025B   732 ;  Routine to enter the new connection block into the connection block
                        025B   733 ;  list for the particular node that the connect request came from. Also,
                        025B   734 ;  the node number field in the connection block field is filled in with
                        025B   735 ;  the source node number of the connect request.
                        025B   736 ;
                        025B   737 ;  INPUTS:
                        025B   738 ;
                        025B   739 ;       R0                      - Remote port number
                        025B   740 ;       R1                      - Address of connection block
                        025B   741 ;
                        025B   742 ;  OUTPUTS:
                        025B   743 ;
                        025B   744 ;       All registers           - Preserved
                        025B   745 ;
                        025B   746 ;-
                        025B   747
                        025B   748 ENTER_CONN_LIST::
                        025B   749
            52    DD    025B   750          PUSHL   R2                      ; Save R2
      26 A1     50    90  025D   751          MOVB    R0,CB$B_RPRTNUM(R1)     ; Copy remote port # to CB
  52    0000'CF     DE  0261   752          MOVAL   CONN_LIST,R2            ; Get base of node list
     52    6240     7E  0266   753          MOVAQ   (R2)[R0],R2             ; Get listhead for this node
        62    61    0E  026A   754          INSQUE  (R1),(R2)               ; Insert CB on CB list
              52 8ED0   026D   755          POPL    R2                      ; Restore R2
                 05     0270   756          RSB
                        0271   757
                        0271   758
```

```
                     0271   760          .SBTTL  CHECK_CTP_REQ,  Check format of CTP request
                     0271   761 ;+
                     0271   762 ;
                     0271   763 ;  CHECK_CTP_REQUEST
                     0271   764 ;
                     0271   765 ;  This routine checks for any fields that may be invalid in a CTP
                     0271   766 ;  request. This routine is used mainly for software debug purposes.
                     0271   767 ;
                     0271   768 ;  INPUTS:
                     0271   769 ;
                     0271   770 ;       R2                      - Address of CTP request
                     0271   771 ;
                     0271   772 ;  OUTPUTS:
                     0271   773 ;
                     0271   774 ;       R0                      - Status (0 - Invalid, 1 - Valid)
                     0271   775 ;       CTP$STATUS(R2)          - CTP$NONSENCE (if request invalid)
                     0271   776 ;       All other registers     - Preserved
                     0271   777 ;
                     0271   778 ;-
                     0271   779
                     0271   780 CHECK_CTP_REQ::
                     0271   781
                     0271   782          .IF NDF SOFTWARE_DEBUG          ; If software debug is DISABLED
                     0271   783
                     0271   784          BRW     VALID_REQUEST
                     0271   785
                     0271   786          .ENDC
                     0271   787
                     0271   788          $DISPATCH -                             ; Dispatch off the CTP opcode
                     0271   789                  CTP$OPCODE(R2),TYPE=B,-         ; to the specific code to
                     0271   790                  <-                              ; check the CTP request.
                     0271   791                  <CTP$FUNCTREQ,  FUNCT>,-
                     0271   792                  <CTP$BUFMAPREQ, BUFMAP>,-
                     0271   793                  <CTP$BUFUNMREQ, BUFUNM>,-
                     0271   794                  <CTP$MOVBUFREQ, MOVBUF>,-
                     0271   795                  <CTP$GENMSGREQ, GENMSG>,-
                     0271   796                  <CTP$GENDGRREQ, GENDG>,-
                     0271   797                  <CTP$GENRSTREQ, GENRST>,-
                     0271   798                  <CTP$GENSTRREQ, GENSTR>,-
                     0271   799                  <CTP$NOACTREQ,  NOACT>,-
                     0271   800                  <CTP$CONFIGREQ, CONFIG>,-
                     0271   801                  <CTP$COUNTSREQ, COUNTS>,-
                     0271   802                  <CTP$CONNECTREQ,CONNECT>,-
                     0271   803                  <CTP$FINISHREQ, FINISH>,-
                     0271   804                  <CTP$MBUFMAPREQ,MBUFMAP>,-
                     0271   805                  <CTP$MBUFUNMREQ,MBUFUNM>,-
                     0271   806                  <CTP$MOVMBUFREQ,MOVMBUF>,-
                     0271   807                  <CTP$MSTATEREQ, MSTATE>,-
                     0271   808                  >
                     0297   809
                     0297   810 FUNCT:
                     0297   811
 0239    31          0297   812          BRW     VALID_REQUEST           ; No checking done
                     029A   813
                     029A   814 BUFMAP:
                     029A   815
 05 A2   95          029A   816          TSTB    CTP*BUFTYPE(R2)         ; Valid buffer type field?
```

```
                          029D  817              BEQLW   INVALID_REQUEST          ; No, branch
          05 A2      91   02A2  818              CMPB    CTP$BUFTYPE(R2),-        ; Valid buffer type field?
             02           02A5  819                      #CTP$FAKEBUF
                          02A6  820              BGTRUW  INVALID_REQUEST          ; No, branch
          06 A2      B5   02AB  821              TSTW    CTP$RESERV6(R2)          ; Is MBZ field zero?
                          02AE  822              BNEQW   INVALID_REQUEST          ; No, branch
          08 A2      91   02B3  823              CMPB    CTP$GENFUNCT(R2),-       ; Valid generate function?
             02           02B6  824                      #CTP$GENFNODE
                          02B7  825              BGTRUW  INVALID_REQUEST          ; No, branch
          0A A2      91   02BC  826              CMPB    CTP$PKTSIZ(R2),-         ; Valid packet size?
             01           02BF  827                      #CTP$PS576
                          02C0  828              BGTRUW  INVALID_REQUEST          ; No, branch
    08    0B A2      91   02C5  829              CMPB    CTP$PKTMULT(R2),#8       ; Valid packet multiple?
                          02C9  830              BGTRUW  INVALID_REQUEST          ; No, branch
          0C A2      D1   02CE  831              CMPL    CTP$BUFLENGTH(R2),-      ; Valid mapped buffer length?
   00000000'8F            02D1  832                      #MAX_BUF_LEN
                          02D6  833              BGTRUW  INVALID_REQUEST          ; No, branch
            01F5     31   02DB  834              BRW     VALID_REQUEST            ; Valid request, branch
                          02DE  835
                          02DE  836  BUFUNM:
                          02DE  837
          05 A2      95   02DE  838              TSTB    CTP$BUFTYPE(R2)          ; Valid buffer type field?
                          02E1  839              BEQLUW  INVALID_REQUEST          ; No, branch
          05 A2      91   02E6  840              CMPB    CTP$BUFTYPE(R2),-        ; Valid buffer type field?
             02           02E9  841                      #CTP$FAKEBUF
                          02EA  842              BGTRUW  INVALID_REQUEST          ; No, branch
          06 A2      D5   02EF  843              TSTL    CTP$RESERV6(R2)          ; Valid MBZ?
                          02F2  844              BNEQUW  INVALID_REQUEST          ; No, branch
          0A A2      B5   02F7  845              TSTW    CTP$RESERV10(R2)         ; Valid MBZ?
                          02FA  846              BNEQUW  INVALID_REQUEST          ; No, branch
          0C A2      D1   02FF  847              CMPL    CTP$BUFLENGTH(R2),-      ; Valid mapped buffer length?
   00000000'8F            0302  848                      #MAX_BUF_LEN
                          0307  849              BGTRUW  INVALID_REQUEST          ; No, branch
            01C4     31   030C  850              BRW     VALID_REQUEST            ; Valid request, branch
                          030F  851
                          030F  852  MOVBUF:
                          030F  853
          08 A2      91   030F  854              CMPB    CTP$MOVETYPE(R2),-       ; Valid move type?
             01           0312  855                      #CTP$MOVETO
                          0313  856              BGTRUW  INVALID_REQUEST          ; No, branch
          09 A2      91   0318  857              CMPB    CTP$OTHERNODE(R2),-      ; Valid othernode field?
          00'8F           031B  858                      #MAX_NODES
                          031D  859              BGTRUW  INVALID_REQUEST          ; No, branch
          0A A2      91   0322  860              CMPB    CTP$PKTSIZ(R2),-         ; Valid packet size?
             01           0325  861                      #CTP$PS576
                          0326  862              BGTRUW  INVALID_REQUEST          ; No, branch
    08    0B A2      91   032B  863              CMPB    CTP$PKTMULT(R2),#8       ; Valid packet multiple?
                          032F  864              BGTRUW  INVALID_REQUEST          ; No, branch
          0C A2      D1   0334  865              CMPL    CTP$BUFLENGTH(R2),-      ; Valid mapped buffer length?
   00000000'8F            0337  866                      #MAX_BUF_LEN
                          033C  867              BGTRUW  INVALID_REQUEST          ; No, branch
            018F     31   0341  868              BRW     VALID_REQUEST            ; Valid request, branch
                          0344  869
                          0344  870  GENMSG:
                          0344  871
          08 A2      91   0344  872              CMPB    CTP$GENFUNCT(R2),-       ; Valid generate function?
             03           0347  873                      #CTP$GENFIMAGE
```

```
                                0348   874          BGTRUW  INVALID_REQUEST          ; No, branch
                     0A A2  B5  034D   875          TSTW    CTP$RESERV10(R2)         ; Valid MBZ
                                0350   876          BNEQUW  INVALID_REQUEST          ; No, branch
                        0E  A3  0355   877          SUBW3   #CTP$IMAGEDATA,-         ; Get length of fill area
           50  00000000'GF     0357   878                  G^SCS$GW_MAXMSG,R0
                50  0C A2  B1  035D   879          CMPW    CTP$GENLENGTH(R2),R0     ; Valid generate length?
                               0361   880          BGTRUW  INVALID_REQUEST          ; No, branch
                    016A  31   0366   881          BRW     VALID_REQUEST            ; Valid request, branch
                               0369   882
                               0369   883 GENDG:
                               0369   884
                     08 A2  91 0369   885          CMPB    CTP$GENFUNCT(R2),-       ; Valid generate function?
                        03     036C   886                  #CTP$GENFIMAGE
                               036D   887          BGTRUW  INVALID_REQUEST          ; No, branch
                     0A A2  B5 0372   888          TSTW    CTP$RESERV10(R2)         ; Valid MBZ
                               0375   889          BNEQW   INVALID_REQUEST          ; No, branch
                        0F  A3 037A   890          SUBW3   #CTP$IMAGEDATA,-         ; Get length of fill area
           50  00000000'GF     037C   891                  G^SCS$GW_MAXDG,R0
                50  0C A2  B1 0382   892          CMPW    CTP$GENLENGTH(R2),R0     ; Valid generate length?
                               0386   893          BGTRUW  INVALID_REQUEST          ; No, branch
                    0145  31   0388   894          BRW     VALID_REQUEST            ; Valid request, branch
                               038E   895
                               038E   896 GENRST:
                               038E   897
                     08 A2  91 038E   898          CMPB    CTP$EXTEND(R2),-         ; Valid reset flag?
                        01     0391   899                  #CTP$CONDRST
                               0392   900          BGTRUW  INVALID_REQUEST          ; No, branch
                  09 A2  91    0397   901          CMPB    CTP$OTHERNODE(R2),-      ; Valid othernode field?
                00'8F          039A   902                  #MAX_NODES
                               039C   903          BGTRUW  INVALID_REQUEST          ; No, branch
                    012F  31   03A1   904          BRW     VALID_REQUEST            ; Valid request, branch
                               03A4   905
                               03A4   906 GENSTR:
                               03A4   907
                  08 A2  91    03A4   908          CMPB    CTP$EXTEND(R2),-         ; Valid start flag?
                        01     03A7   909                  #CTP$SPECIFADR
                               03A8   910          BGTRUW  INVALID_REQUEST          ; No, branch
                  0A A2  B5    03AD   911          TSTW    CTP$RESERV10(R2)         ; Valid MBZ?
                               03B0   912          BNEQW   INVALID_REQUEST          ; No, branch
                    011B  31   03B5   913          BRW     VALID_REQUEST            ; Valid request, branch
                               03B8   914
                               03B8   915 NOACT:
                               03B8   916
                  05 A2  D5    03B8   917          TSTL    CTP$RESERV5(R2)          ; Valid MBZ?
                               03BB   918          BNEQW   INVALID_REQUEST          ; No, branch
                  09 A2  95    03C0   919          TSTB    CTP$RESERV9(R2)          ; Valid MBZ?
                               03C3   920          BNEQW   INVALID_REQUEST          ; No, branch
                  0A A2  91    03C8   921          CMPB    CTP$NOACTFLAG(R2),-      ; Valid no activity flag?
                        01     03CB   922                  #CTP$NOACTOFF
                               03CC   923          BGTRUW  INVALID_REQUEST          ; No, branch
                    00FF  31   03D1   924          BRW     VALID_REQUEST            ; Valid request, branch
                               03D4   925
                               03D4   926 CONFIG:
                               03D4   927
                  05 A2  D5    03D4   928          TSTL    CTP$RESERV5(R2)          ; Valid MBZ?
                               03D7   929          BNEQW   INVALID_REQUEST          ; No, branch
                  09 A2  91    03DC   930          CMPB    CTP$OTHERNODE(R2),-      ; Valid othernode field?
```

```
                00'8F        03DF   931                     #MAX_NODES
                             03E1   932             BGTRUW  INVALID_REQUEST          ; No, branch
                 00EA   31   03E6   933             BRW     VALID_REQUEST           ; Valid request, branch
                             03E9   934
                             03E9   935 COUNTS:
                             03E9   936
                05 A2   D5   03E9   937             TSTL    CTP$RESERV5(R2)         ; Valid MBZ?
                             03EC   938             BNEQW   INVALID_REQUEST         ; No, branch
                09 A2   91   03F1   939             CMPB    CTP$OTHERNODE(R2),-     ; Valid othernode field?
                FF 8F        03F4   940                     #^XFF                   ; (all nodes)
                   0A   13   03F6   941             BEQL    10$                     ; yes, skip next check
                09 A2   91   03F8   942             CMPB    CTP$OTHERNODE(R2),-     ; Valid othernode field?
                00'8F        03FB   943                     #MAX_NODES
                             03FD   944             BGTRUW  INVALID_REQUEST         ; No, branch
                0A A2   91   0402   945 10$:        CMPB    CTP$CNTFLG(R2),-        ; Valid counter flag?
            00000001'EF      0405   946                     CTP$KEEPCNT
                             040A   947             BGTRUW  INVALID_REQUEST         ; No, branch
                 00C1   31   040F   948             BRW     VALID_REQUEST           ; Valid request, branch
                             0412   949
                             0412   950 CONNECT:
                             0412   951
                05 A2   D5   0412   952             TSTL    CTP$RESERV5(R2)         ; Valid MBZ?
                             0415   953             BNEQW   INVALID_REQUEST         ; No, branch
                09 A2   91   041A   954             CMPB    CTP$OTHERNODE(R2),-     ; Valid othernode field?
                00'8F        041D   955                     #MAX_NODES
                             041F   956             BGTRUW  INVALID_REQUEST         ; No, branch
                 00AC   31   0424   957             BRW     VALID_REQUEST           ; Valid request, branch
                             0427   958
                             0427   959 FINISH:
                             0427   960
                 00A9   31   0427   961             BRW     VALID_REQUEST           ; Valid request, branch
                             042A   962
                             042A   963 MBUFMAP:
                             042A   964
                05 A2   91   042A   965             CMPB    CTP$BUFTYPE(R2),-       ; Valid buffer type field?
                   03        042D   966                     #CTP$MAINTBUF
                             042E   967             BNEQW   INVALID_REQUEST         ; No, branch
                06 A2   B5   0433   968             TSTW    CTP$RESERV6(R2)         ; Is MBZ field zero?
                             0436   969             BNEQW   INVALID_REQUEST         ; No, branch
                08 A2   91   043B   970             CMPB    CTP$GENFUNCT(R2),-      ; Valid generate function?
                   02        043E   971                     #CTP$GENFNODE
                             043F   972             BGTRUW  INVALID_REQUEST         ; No, branch
                0A A2   91   0444   973             CMPB    CTP$PKTSIZ(R2),-        ; Valid packet size?
                   01        0447   974                     #CTP$PS576
                             0448   975             BGTRUW  INVALID_REQUEST         ; No, branch
                0B A2   95   044D   976             TSTB    CTP$RESERV11(R2)        ; Valid MBZ?
                   29   12   0450   977             BNEQ    INVALID_REQUEST         ; No, branch
                0C A2   D1   0452   978             CMPL    CTP$BUFLENGTH(R2),-     ; Valid mapped buffer length?
            00000000'8F      0455   979                     #MAX_BUF_LEN
                             045A   980             BGTRUW  INVALID_REQUEST         ; No, branch
                 0071   31   045F   981             BRW     VALID_REQUEST           ; Valid request, branch
                             0462   982
                             0462   983 MBUFUNM:
                             0462   984
                05 A2   91   0462   985             CMPB    CTP$BUFTYPE(R2),-       ; Valid buffer type field?
                   03        0465   986                     #CTP$MAINTBUF
                   63   12   0466   987             BNEQ    INVALID_REQUEST         ; No, branch
```

```
                06 A2   D5  0468   988          TSTL    CTP$RESERV6(R2)          ; Valid MBZ?
                            0468   989          BNEQUW  INVALID_REQUEST          ; No, branch
                0A A2   B5  0470   990          TSTW    CTP$RESERV10(R2)         ; Valid MBZ?
                            0473   991          BNEQUW  INVALID_REQUEST          ; No, branch
                0C A2   D5  0478   992          TSTL    CTP$RESERV12(R2)         ; Valid MBZ?
                            047B   993          BNEQUW  INVALID_REQUEST          ; No, branch
                 0050   31  0480   994          BRW     VALID_REQUEST            ; Valid request, branch
                            0483   995
                            0483   996 MOVMBUF:
                            0483   997
                08 A2   91  0483   998          CMPB    CTP$MOVETYPE(R2),-       ; Valid move type?
                   01           0486   999                  #CTP$MNTMOVETO
                            0487  1000          BGTRUW  INVALID_REQUEST          ; No, branch
                09 A2   91  048C  1001          CMPB    CTP$OTHERNODE(R2),-      ; Valid othernode field?
                00'8F           048F  1002                  #MAX_NODES
                            0491  1003          BGTRUW  INVALID_REQUEST          ; No, branch
                0A A2   91  0496  1004          CMPB    CTP$PKTSIZ(R2),-         ; Valid packet size?
                   01           0499  1005                  #CTP$PS576
                            049A  1006          BGTRUW  INVALID_REQUEST          ; No, branch
                0B A2   95  049F  1007          TSTB    CTP$RESERV11(R2)         ; Valid MBZ
                   27   12  04A2  1008          BNEQ    INVALID_REQUEST          ; No, branch
                0C A2   D1  04A4  1009          CMPL    CTP$BUFLENGTH(R2),-      ; Valid mapped buffer length?
          00000000'8F           04A7  1010                  #MAX_BUF_LEN
                            04AC  1011          BGTRUW  INVALID_REQUEST          ; No, branch
                14 A2   D5  04B1  1012          TSTL    CTP$RESERV20(R2)         ; Valid MBZ?
                   15   12  04B4  1013          BNEQ    INVALID_REQUEST          ; No, branch
                 001A   31  04B6  1014          BRW     VALID_REQUEST            ; Valid request, branch
                            04B9  1015
                            04B9  1016 MSTATE:
                            04B9  1017
                05 A2   B5  04B9  1018          TSTW    CTP$RESERV5(R2)          ; Valid MBZ
                   0D   12  04BC  1019          BNEQ    INVALID_REQUEST          ; No, branch
                07 A2   95  04BE  1020          TSTB    CTP$RESERV7(R2)          ; Valid MBZ
                   08   12  04C1  1021          BNEQ    INVALID_REQUEST          ; No, branch
                08 A2   91  04C3  1022          CMPB    CTP$GENFUNCT(R2),-       ; Valid maint state flag?
                   01           04C6  1023                  #CTP$NORMSTATE
                   02   1A  04C7  1024          BGTRU   INVALID_REQUEST          ; No, branch
                   08   11  04C9  1025          BRB     VALID_REQUEST            ; Valid request, branch
                            04CB  1026
                            04CB  1027 INVALID_REQUEST:
                            04CB  1028
                   50   D4  04CB  1029          CLRL    R0                       ; Set error status
                FE 8F   90  04CD  1030          MOVB    #CTP$NONSENSE,-          ; Set appropriate status in
                05 A2           04D0  1031                  CTP$STATUS(R2)       ; CTP response
                        05  04D2  1032          RSB                              ; Return
                            04D3  1033
                            04D3  1034 VALID_REQUEST:
                            04D3  1035
          50    01    D0  04D3  1036          MOVL    #1,R0                    ; Set success status
                        05  04D6  1037          RSB                              ; Return
                            04D7  1038
                            04D7  1039
                            04D7  1040
```

```
                        04D7  1042          .SBTTL  BUILD_MSG,      Build message buffer
                        04D7  1043          .SBTTL  BUILD_DG,       Build datagram buffer
                        04D7  1044 ;+
                        04D7  1045 ;
                        04D7  1046 ;  BUILD_MSG
                        04D7  1047 ;  BUILD_DG
                        04D7  1048 ;
                        04D7  1049 ;  Routine to insert the proper data into CTP generate message and datagram
                        04D7  1050 ;  responses. This includes the following:
                        04D7  1051 ;
                        04D7  1052 ;       - CTP response opcode
                        04D7  1053 ;       - Command reference number (copied from request)
                        04D7  1054 ;       - CTP status of CTP$SUCCESS
                        04D7  1055 ;       - Actual count (copied from request block)
                        04D7  1056 ;       - Generation function (copied from request)
                        04D7  1057 ;       - Port status (not yet implemented)
                        04D7  1058 ;       - Generation length (copied from request)
                        04D7  1059 ;       - Generate data field
                        04D7  1060 ;
                        04D7  1061 ;
                        04D7  1062 ;  INPUTS:
                        04D7  1063 ;
                        04D7  1064 ;       R2                          - Address of CTP response message buffer
                        04D7  1065 ;       R3                          - Address of request block
                        04D7  1066 ;       RB$L_CTPBUF(R1)             - Address of CTP request
                        04D7  1067 ;
                        04D7  1068 ;  OUTPUTS:
                        04D7  1069 ;
                        04D7  1070 ;       All registers               - Preserved
                        04D7  1071 ;
                        04D7  1072 ;-
                        04D7  1073
                        04D7  1074 BUILD_MSG::
                        04D7  1075 BUILD_DG::
                        04D7  1076
        3F    BB        04D7  1077          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save registers
50  10 A3    D0        04D9  1078          MOVL    RB$L_CTPBUF(R3),R0      ; Get CTP request address
60  40 8F    81        04DD  1079          ADDB3   #64,CTP$OPCODE(R0),-    ; Fill in response opcode
               62        04E1  1080                  CTP$OPCODE(R2)
    01 A0    D0        04E2  1081          MOVL    CTP$REFERENCE(R0),-     ; Copy cmd ref #
    01 A2              04E5  1082                  CTP$REFERENCE(R2)
    24 A3    B0        04E7  1083          MOVW    RB$W_ACTCNT(R3),-       ; Copy actual count
    06 A2              04EA  1084                  CTP$ACTCOUNT(R2)
    08 A0    90        04EC  1085          MOVB    CTP$GENFUNCT(R0),-      ; Copy generate function
    08 A2              04EF  1086                  CTP$GENFUNCT(R2)
    0C A0    B0        04F1  1087          MOVW    CTP$GENLENGTH(R0),-     ; Copy generate length
    0C A2              04F4  1088                  CTP$GENLENGTH(R2)
                       04F6  1089          $DISPATCH -                             ; Dispatch on the generate
                       04F6  1090                  CTP$GENFUNCT(R2),TYPE=B,-       ; function field to the proper
                       04F6  1091                  <-                              ; code to build the buffer
                       04F6  1092                  <CTP$GENFFILL,  10$>,-
                       04F6  1093                  <CTP$GENFRPAIR, 20$>,-
                       04F6  1094                  <CTP$GENFNODE,  50$>,-
                       04F6  1095                  <CTP$GENFIMAGE, 60$>,-
                       04F6  1096                  >
                       0503  1097
                       0503  1098
```

```
                        0503  1099 ;  FILL BUFFER WITH CONSTANT
                        0503  1100 ;
                        0503  1101 ;  Fill the buffer the constant specified in the CTP request for the length
                        0503  1102 ;  specified in the CTP request.
                        0503  1103
09 A0   FE AF   00   2C 0503  1104 10$:    MOVC5   #0,,CTP$GENCONST(R0),- ; Copy the fill constant into
                0C A0       0509  1105                CTP$GENLENGTH(R0),-    ; CTP response buffer
                0E A2       050B  1106                CTP$GENDATA(R2)
                   39   11  050D  1107         BRB     BUILD_MSG_EXIT
                        050F  1108
                        050F  1109
                        050F  1110 ;  FILL BUFFER WITH BYTE PAIRS
                        050F  1111 ;
                        050F  1112 ;  In a byte pair buffer, the first word (two bytes) in the buffer is a zero,
                        050F  1113 ;  the next word a one, the next, a two, and so on. Thus, the buffer looks as
                        050F  1114 ;  follows:
                        050F  1115 ;
                        050F  1116 ;        ---------- ---------- ---------- ----------
                        050F  1117 ;       |    0     |    1     |    0     |    0     |
                        050F  1118 ;        ---------- ---------- ---------- ----------
                        050F  1119 ;       |    0     |    3     |    0     |    2     |
                        050F  1120 ;        ---------- ---------- ---------- ----------
                        050F  1121 ;       |    0     |    5     |    0     |    4     |
                        050F  1122 ;        ---------- ---------- ---------- ----------
                        050F  1123 ;       |    0     |    7     |    0     |    6     |
                        050F  1124 ;        ---------- ---------- ---------- ----------
                        050F  1125 ;       |    0     |    9     |    0     |    8     |
                        050F  1126 ;        ---------- ---------- ---------- ----------
                        050F  1127
   52   CF A2   DE       050F  1128 20$:    MOVAL   CTP$GENDATA(R2),R2      ; Point to fill area
   50   0C A0   B0       0513  1129         MOVW    CTP$GENLENGTH(R0),R0    ; Length of buffer
           54   D4       0517  1130         CLRL    R4                      ; Make first byte pair
           50   B5       0519  1131 30$:    TSTW    R0                      ; Test for zero count
           2B   13       051B  1132         BEQL    BUILD_MSG_EXIT          ; If zero, branch
      01   50   B1       051D  1133         CMPW    R0,#1                   ; Compare count to 1
           0A   13       0520  1134         BEQL    40$                     ; If one, branch
      62   54   B0       0522  1135         MOVW    R4,(R2)+                ; Fill in a byte pair
      50   02   A2       0525  1136         SUBW2   #2,R0                   ; Subtract two from count
           54   B6       0528  1137         INCW    R4                      ; Make next byte pair
           ED   11       052A  1138         BRB     30$                     ; Loop
      62   54   90       052C  1139 40$:    MOVB    R4,(R2)                 ; Fill in last byte
           17   11       052F  1140         BRB     BUILD_MSG_EXIT          ; Exit
                        0531  1141
                        0531  1142
                        0531  1143 ;  FILL BUFFER WITH LOCAL PORT NUMBER
                        0531  1144 ;
                        0531  1145 ;  The number of the local port, over which the CTP request was recieved is
                        0531  1146 ;  written into every byte of the buffer.
                        0531  1147
   53      A3   DO       0531  1148 50$:    MOVL    RB$L_CB(R3),R3          ; Get connection block address
A5 A3   FF AF   00   2C 0535  1149         MOVC5   #0,,CB$B_LPRTNUM(R3),-  ; Copy the local port number
                0C A3       053B  1150                CTP$GENLENGTH(R0),-    ; into the fill area.
                CE A3       053D  1151                CTP$GENDATA(R2)
                   57   11  053F  1152         BRB     BUILD_MSG_EXIT
                        0541  1153
                        0541  1154
                        0541  1155 ;  FILL BUFFER WITH IMAGE DATA
```

```
                 0541    ;
                 0541    ;  Image data simply means copying the exact data in the CTP request to the
                 0541    ;  CTP response. The CTP request specifies the length of the image data.
                 0541
CF A0       28   0541    60$:    MOVC3   CTP$GENLENGTH(R0),-        ; Fill response with
CE A0            0544                    CTP$IMAGEDATA(R0),-        ; image data from request.
OE A2            0546                    CTP$GENDATA(R2)
                 0548
                 0548
                 0548    BUILD_MSG_EXIT:
                 0548
   3F       BA   0548            POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore registers
            05   054A            RSB
                 054B
                 054B            .DISABLE LSB
                 054B
                 054B
                 054B
```

```
                                 054B          .SBTTL  BUILD_BUFFER,   Build mapped buffer
                                 054B     ;+
                                 054B     ;
                                 054B     ;  BUILD_BUFFER
                                 054B     ;
                                 054B     ;  Routine to insert the proper data into a buffer being mapped for
                                 054B     ;  data transfers.
                                 054B     ;
                                 054B     ;  INPUTS:
                                 054B     ;
                                 054B     ;       R2                              - Address of buffer map request
                                 054B     ;       R3                              - Address of request block
                                 054B     ;       R5                              - CDRP address
                                 054B     ;       CDRP$L_BUFADR(R5)               - Buffer address (beyond VMS header)
                                 054B     ;       CDRP$L_BUFLEN(R5)               - Buffer length (NOT including VMS overhead)
                                 054B     ;
                                 054B     ;  OUTPUTS:
                                 054B     ;
                                 054B     ;       All registers           - Preserved
                                 054B     ;
                                 054B     ;-
                                 054B
                                 054B     BUILD_BUFFER::
                                 054B
                     3E    BB    054B          PUSHR   #^M<R1,R2,R3,R4,R5>        ; Save registers
                                 054D          $DISPATCH -                        ; Dispatch on the generate
                                 054D                  CTP$GENFUNCT(R2) -         ; function field to the proper
                                 054D                  TYPE=B,-                   ; code to build the buffer
                                 054D                  <-
                                 054D                  <CTP$GENFFILL,  10$>,-
                                 054D                  <CTP$GENFBPAIR, 20$>,-
                                 054D                  <CTP$GENFNODE,  50$>,-
                                 054D                  >
                                 0558
                                 0558     ;  FILL BUFFER WITH CONSTANT
                                 0558     ;
                                 0558     ;  Fill the buffer the constant specified in the CTP request for the length
                                 0558     ;  specified in the CTP request.
                                 0558
 09 A2    FE AF    00      2C    0558     10$:  MOVC5   #0,,CTP$GENCONST(R2),- ; Copy the fill constant into
                60 A5            055E                  CDRP$L_BUFLEN(R5),-    ; the data transfer buffer
                5C B5            0560                  @CDRP$L_BUFADR(R5)
                   32      11    0562          BRB     BUILD_BUF_EXIT
                                 0564
                                 0564
                                 0564     ;  FILL BUFFER WITH BYTE PAIRS
                                 0564     ;
                                 0564     ;  In a byte pair buffer, the first word (two bytes) in the buffer is a zero,
                                 0564     ;  the next word a one, the next, a two, and so on. Thus, the buffer looks as
                                 0564     ;  follows:
                                 0564     ;
                                 0564     ;        ----------- ----------- ----------- -----------
                                 0564     ;        :    0     :     1     :     0     :     0     :
                                 0564     ;        ----------- ----------- ----------- -----------
                                 0564     ;        :    0     :     3     :     0     :     2     :
                                 0564     ;        ----------- ----------- ----------- -----------
                                 0564     ;        :    0     :     5     :     0     :     4     :
```

```
                                0564  1232 ;        ---------- ---------- ---------- ----------
                                0564  1233 ;       |    0     |    7     |    0     |    6     |
                                0564  1234 ;        ---------- ---------- ---------- ----------
                                0564  1235 ;       |    0     |    9     |    0     |    8     |
                                0564  1236 ;        ---------- ---------- ---------- ----------
                                0564  1237
52   5C A5    D0                0564  1238 20$:    MOVL    CDRP$L_BUFADR(R5),R2    ; Point to fill area
50   60 A5    B0                0568  1239         MOVW    CDRP$L_BUFLEN(R5),R0    ; Length of buffer
          54    D4               056C  1240         CLRL    R4                      ; Make first byte pair
          50    B5               056E  1241 30$:    TSTW    R0                      ; Test for zero count
          24    13               0570  1242         BEQL    BUILD_BUF_EXIT          ; If zero, branch
     01   50    B1               0572  1243         CMPW    R0,#1                   ; Compare count to 1
          0A    13               0575  1244         BEQL    40$                     ; If one, branch
     82   54    B0               0577  1245         MOVW    R4,(R2)+                ; Fill in a byte pair
     50   02    A2               057A  1246         SUBW2   #2,R0                   ; Subtract two from count
          54    B6               057D  1247         INCW    R4                      ; Make next byte pair
          ED    11               057F  1248         BRB     30$                     ; Loop
     62   54    90               0581  1249 40$:    MOVB    R4,(R2)                 ; Fill in last byte
          10    11               0584  1250         BRB     BUILD_BUF_EXIT          ; Exit
                                0586  1251
                                0586  1252
                                0586  1253 ;  FILL BUFFER WITH LOCAL PORT NUMBER
                                0586  1254 ;
                                0586  1255 ;  The number of the local port, over which the CTP request was recieved is
                                0586  1256 ;  written into every byte of the buffer.
                                0586  1257
     50   0C A3    D0           0586  1258 50$:    MOVL    RB$L_CB(R3),R0          ; Get connection block address
25 A0   FE AF    00    2C      058A  1259         MOVC5   #0,,CB$B_LPRTNUM(R0),-  ; Copy local port number into
               60 A5           0590  1260                 CDRP$L_BUFLEN(R5),-     ; the fill area
               5C B5           0592  1261                 @CDRP$L_BUFADR(R5)
          00    11               0594  1262         BRB     BUILD_BUF_EXIT
                                0596  1263
                                0596  1264
                                0596  1265 BUILD_BUF_EXIT:                       ; Common Exit
                                0596  1266
          3E    BA               0596  1267         POPR    #^M<R1,R2,R3,R4,R5>     ; Restore registers
          05    0598  1268         RSB
                                0599  1269
                                0599  1270
                                0599  1271
```

```
                                0599  1273          .SBTTL  FIND_MAPPED_BUFFER, Find mapped buffer CDRP
                                0599  1274 ;+
                                0599  1275 ;
                                0599  1276 ;  FIND_MAPPED_BUFFER
                                0599  1277 ;
                                0599  1278 ;  Routine to find the IRP/CDRP containing the mapped buffer information for
                                0599  1279 ;  the local buffer name specified.
                                0599  1280 ;
                                0599  1281 ;  The IRP/CDRP's containing the mapped buffer information are queued onto
                                0599  1282 ;  the connection block for the particular request. Each is checked for the
                                0599  1283 ;  specified buffer name until a match is found, or until all buffers have
                                0599  1284 ;  been checked.
                                0599  1285 ;
                                0599  1286 ;  INPUTS:
                                0599  1287 ;
                                0599  1288 ;       R5                      - Address of loacl buffer name
                                0599  1289 ;
                                0599  1290 ;  OUTPUTS:
                                0599  1291 ;
                                0599  1292 ;       R0                      - Status
                                0599  1293 ;       R5                      - Address of IRP/CDRP for mapped buffer (if success)
                                0599  1294 ;       All other registers     - Preserved
                                0599  1295 ;
                                0599  1296 ;-
                                0599  1297
                                0599  1298 FIND_MAPPED_BUFFER::
                                0599  1299
                                0599  1300          ASSUME  CIBHAN$L_BNAME EQ 4
                                0599  1301
            14   BB             0599  1302          PUSHR   #^M<R2,R4>              ; Save registers
50  0000'CF  DE                 059B  1303          MOVAL   MAP_QUEUE,R0            ; Get mapped buffer listhead
    52   60  D0                 05A0  1304          MOVL    (R0),R2                 ; Get a mapped buffer IRP/CDRP
    52   50  D1                 05A3  1305 10$:     CMPL    R0,R2                   ; End of list?
         18  13                 05A6  1306          BEQL    30$                     ; Yes, branch
 52   88 A2  DE                 05A8  1307          MOVAL   -CDRP$L_MAPQ_FL(R2),R2  ; Point to CDRP base
         65  D1                 05AC  1308          CMPL    (R5),-                  ; Matching buffer name?
       48 A2                    05AE  1309                  CDRP$T_CY_LBUFHNDL+4(R2)
         06  13                 05B0  1310          BEQL    20$                     ; Yes, branch
 52   78 A2  D0                 05B2  1311          MOVL    CDRP$L_MAPQ_FL(R2),R2   ; Get next IRP/CDRP in list
         EB  11                 05B6  1312          BRB     10$                     ; Loop
                                05B8  1313
    55   52  D0                 05B8  1314 20$:     MOVL    R2,R5                   ; Return CDRP address in R5
    50   01  9A                 05BB  1315          MOVZBL  #1,R0                   ; Set successful status
         02  11                 05BE  1316          BRB     40$                     ; Skip failure code
                                05C0  1317
         50  D4                 05C0  1318 30$:     CLRL    R0                      ; Set failure status
         14  BA                 05C2  1319 40$:     POPR    #^M<R2,R4>              ; Restore registers
             05                 05C4  1320          RSB
                                05C5  1321
                                05C5  1322
```

```
                                05C5  1324          .SBTTL  FIND_MAINT_BUFFER, Find maintenance buffer CDRP
                                05C5  1325 ;+
                                05C5  1326 ;
                                05C5  1327 ;  FIND_MAINT_BUFFER
                                05C5  1328 ;
                                05C5  1329 ;  Routine to find the maintenance CDRP containing the mapped maintenance
                                05C5  1330 ;  buffer information for the local buffer name specified.
                                05C5  1331 ;
                                05C5  1332 ;  INPUTS:
                                05C5  1333 ;
                                05C5  1334 ;       R5                          - Address of maintenance buffer name
                                05C5  1335 ;
                                05C5  1336 ;  OUTPUTS:
                                05C5  1337 ;
                                05C5  1338 ;       R0                          - Status
                                05C5  1339 ;       R5                          - Maintenance CDRP address (if success)
                                05C5  1340 ;       All other registers         - Preserved
                                05C5  1341 ;
                                05C5  1342 ;-
                                05C5  1343
                                05C5  1344 FIND_MAINT_BUFFER::
                                05C5  1345
                                05C5  1346          ASSUME  CIBHAN$L_BNAME EQ 4
                                05C5  1347
                     14   BB    05C5  1348          PUSHR   #^M<R2,R4>              ; Save regs
          50  0000'CF   DE    05C7  1349          MOVAL   MNT_MAP_QUEUE,R0        ; Get maint mapped buffer listhead
              52   60   D0    05CC  1350          MOVL    (R0),R2                 ; Get a maint CDRP
              52   50   D1    05CF  1351 10$:     CMPL    R0,R2                   ; End of list?
                   18   13    05D2  1352          BEQL    30$                     ; Yes, branch
          52   88 A2   DE    05D4  1353          MOVAL   -CDRP$L_MAPQ_FL(R2),R2  ; Point to CDRP base
                   65   D1    05D8  1354          CMPL    (R5),-                  ; Matching buffer name?
                 48 A2         05DA  1355                  CDRP$T_CY_LBUFHNDL+4(R2)
                   06   13    05DC  1356          BEQL    20$                     ; Yes, branch
          52   78 A2   D0    05DE  1357          MOVL    CDRP$L_MAPQ_FL(R2),R2   ; Get next IRP/CDRP in list
                   EB   11    05E2  1358          BRB     10$                     ; Loop
                        05E4  1359
              55   52   D0    05E4  1360 20$:     MOVL    R2,R5                   ; Return CDRP address in R5
              50   01   9A    05E7  1361          MOVZBL  #1,R0                   ; Set successful status
                   02   11    05EA  1362          BRB     40$                     ; Skip failure code
                        05EC  1363
                   50   D4    05EC  1364 30$:     CLRL    R0                      ; Set failure status
                   14   BA    05EE  1365 40$:     POPR    #^M<R2,R4>              ; Restore registers
                        05    05F0  1366          RSB
                              05F1  1367
                              05F1  1368
```

```
                                05F1  1370          .SBTTL  QUEUE_DELAY,    Queue request block to timer queue
                                05F1  1371 ;+
                                05F1  1372 ;
                                05F1  1373 ;  QUEUE_DELAY
                                05F1  1374 ;
                                05F1  1375 ;  This routine is used for CTP requests containing a non-zero delay count.
                                05F1  1376 ;  The request block is removed from the execution list and queued to the
                                05F1  1377 ;  timer queue.
                                05F1  1378 ;
                                05F1  1379 ;  The duetime field in the responder's CRB is set up so that the timer
                                05F1  1380 ;  interrupt routine will be awakened in one second. At that time, the delay
                                05F1  1381 ;  count in the request block is decremented by two (it represents 500 ms),
                                05F1  1382 ;  and if it drops below zero, the request block is requeued to the execution
                                05F1  1383 ;  list.
                                05F1  1384 ;
                                05F1  1385 ;  INPUTS:
                                05F1  1386 ;
                                05F1  1387 ;       R2                      - CTP request address
                                05F1  1388 ;       R3                      - Request block address
                                05F1  1389 ;       R4                      - PDT address
                                05F1  1390 ;       R5                      - CDRP address
                                05F1  1391 ;
                                05F1  1392 ;  OUTPUTS:
                                05F1  1393 ;
                                05F1  1394 ;       R0                      - Destroyed
                                05F1  1395 ;       All other registers     - Preserved     .
                                05F1  1396 ;
                                05F1  1397 ;-
                                05F1  1398
                                05F1  1399 QUEUE_DELAY::
                                05F1  1400
          53    63    0F      05F1  1401          REMQUE  (R3),R3                 ; Remove request block from exec list
    0004'DF     63    0E      05F4  1402          INSQUE  (R3),@TIMER_QUEUE+4     ; Insert it on the timer queue
       03 20 A3    E9          05F9  1403          BLBC    RB$W_DELAY(R3),10$      ; If even delay count, branch
          20 A3    96          05FD  1404          INCB    RB$W_DELAY(R3)          ; Make delay time even
                               0600  1405
          50    0000'CF    D0  0600  1406 10$:     MOVL    CY$L_SAVED_CRB,R0       ; Get CRB address
00000000'GF     01    C1      0605  1407          ADDL3   #1,G^EXE$GL_ABSTIM,-    ; Set up due time field in CRB
             18 A0            060C  1408                  CRB$L_DUETIME(R0)
                              060E  1409
0000'DF     0000'CF    D1      060E  1410          CMPL    EXEC_LIST,@EXEC_LIST    ; Is execution list empty
                               0615  1411          BNEQW   MAINLINE                ; No, branch
                         05    061A  1412          RSB                             ; Wait for timer interrupt
                               061B  1413
                               061B  1414
```

```
                                061B  1416          .SBTTL  TIMER_INTR,     Timer interrupt routine
                                061B  1417 ;+
                                061B  1418 ;
                                061B  1419 ;  TIMER_INTR
                                061B  1420 ;
                                061B  1421 ;  Routine which is woken up every second if the timer queue is not empty.
                                061B  1422 ;  The delay count field in each of request blocks in the timer queue is
                                061B  1423 ;  decremented by two. If a deleay count field in a request block reaches
                                061B  1424 ;  zero, the request block is removed from the timer queue and requeued to
                                061B  1425 ;  the execution list.
                                061B  1426 ;
                                061B  1427 ;  If the execution list was empty before requeing a request block to it, the
                                061B  1428 ;  mainline routine is invoked as a subroutine. This allows the request that
                                061B  1429 ;  was queued to it to be executed, and returns control here. The timer queue
                                061B  1430 ;  is then
                                061B  1431 ;
                                061B  1432 ;  INPUTS:
                                061B  1433 ;
                                061B  1434 ;  OUTPUTS:
                                061B  1435 ;
                                061B  1436 ;-
                                061B  1437
                                061B  1438 TIMER_INTR::
                                061B  1439
                     03   BB    061B  1440          PUSHR   #^M<R0,R1>              ; Save regs
          50  0000'CF   DE    061D  1441          MOVAL   TIMER_QUEUE,R0          ; Get timer queue listhead
              51   60   D0    0622  1442          MOVL    (R0),R1                 ; Get first RB in timer queue
              51   50   D1    0625  1443 10$:     CMPL    R0,R1                   ; End of list
                   2D   13    0628  1444          BEQL    40$                     ; Yes, branch
                                062A  1445
            20   A1   02   A2  062A  1446          SUBW2   #2,RB$W_DELAY(R1)       ; Decrement delay count
                   05   13    062E  1447          BEQL    20$                     ; Delay expired, branch
                51   61   D0    0630  1448          MOVL    (R1),R1                 ; Get next request block in timer queue
                   F0   11    0633  1449          BRB     10$
                                0635  1450
                   61   DD    0635  1451 20$:     PUSHL   (R1)                    ; Save next RB in timer queue
                51   61   0F    0637  1452          REMQUE  (R1),R1                 ; Remove request block from timer queue
          0004'DF   61   0E    063A  1453          INSQUE  (R1),@EXEC_LIST+4       ; Put it back on the execution list
0004'CF   0000'CF   D1    063F  1454          CMPL    EXEC_LIST,EXEC_LIST+4   ; Was execution list previously empty?
                   05   13    0646  1455          BEQL    30$                     ; Yes, branch
                51 8ED0    0648  1456 25$:     POPL    R1                      ; Restore next RB in timer queue
                   D8   11    064B  1457          BRB     10$
                                064D  1458
                   50   DD    064D  1459 30$:     PUSHL   R0                      ; Save reg
                F9AE'   30    064F  1460          BSBW    MAINLINE                ; Go execute mainline routine
                50 8ED0    0652  1461          POPL    R0                      ; Restore reg
                   F1   11    0655  1462          BRB     25$
                                0657  1463
          51  0000'CF   D0    0657  1464 40$:     MOVL    CY$L_SAVED_CRB,R1       ; Get CRB address
          0004'CF   50   D1    065C  1465          CMPL    R0,TIMER_QUEUE+4        ; Timer queue empty?
                   06   12    0661  1466          BNEQ    50$                     ; No, branch
            18   A1   01   CE  0663  1467          MNEGL   #1,CRB$L_DUETIME(R1)    ; Set infinite duetime
                   09   11    0667  1468          BRB     60$
                                0669  1469
00000000'GF        C1   C1    0669  1470 50$:     ADDL3   #1,G^EXE$GL_ABSTIM,-    ; Set up due time field in CRB
                18   A1         0670  1471                  CRB$L_DUETIME(R1)
                   03   BA    0672  1472 60$:     POPR    #^M<R0,R1>              ; Restore regs
```

```
                05  0674  1473          RSB
                    0675  1474
                    0675  1475
                    0675  1476
```

```
                                0675  1478          .SBTTL  CY$MONITOR,     Monitor responder activity
                                0675  1479 ;+
                                0675  1480 ;
                                0675  1481 ;  CY$MONITOR
                                0675  1482 ;
                                0675  1483 ;  Routine to monitor th activity taking place in the responder. A monitor
                                0675  1484 ;  block is allocated and filled in with the CTP opcode of the currently
                                0675  1485 ;  executing CTP request. The command reference number and repeat count
                                0675  1486 ;  are also saved. The monitor block is linked to the end of the monitor
                                0675  1487 ;  list.
                                0675  1488 ;
                                0675  1489 ;-
                                0675  1490
                                0675  1491 CY$MONITOR::
                                0675  1492
                             05 0675  1493          RSB
                                0676  1494
                          53    DD  0676  1495          PUSHL   R3
                          07    BB  0678  1496          PUSHR   #^M<R0,R1,R2>
                          0E    13  067A  1497          BEQL    10$
                    51    20    D0  067C  1498          MOVL    #^X20,R1
                       FA48     30  067F  1499          BSBW    ALLOC_BUFFER
                     1F 50      E9  0682  1500          BLBC    R0,30$
                    53  52      D0  0685  1501          MOVL    R2,R3
                          05    11  0688  1502          BRB     20$
                53  0004'DF    0F  068A  1503 10$:     REMQUE  @MONITOR_LIST+4,R3
                          07    BA  068F  1504 20$:     POPR    #^M<R0,R1,R2>
                   0C A3  62    90  0691  1505          movb    ctp$opcode(r2),12(r3)
                0E A3   24 A1   B0  0695  1506          movw    rb$w_actcnt(r1),14(r3)
                10 A3   01 A2   D0  069A  1507          movl    ctp$reference(r2),16(r3)
                 0000'CF    63   0E  069F  1508          INSQUE  (R3),MONITOR_LIST
                          53 8ED0  06A4  1509 30$:     POPL    R3
                             05    06A7  1510          RSB
                                06A8  1511
                                06A8  1512
                                06A8  1513 CY$END::
                                06A8  1514
                                06A8  1515          .END
```

```
ALLOC_BUFFER                    000000CA RG    01
ALLOC_CB                        00000000 RG    01
ALLOC_CDRP                      0000003D RG    01
ALLOC_IRP_CDRP                  00000063 RG    01
ALLOC_MNT_CDRP                  000000A9 RG    01
ALLOC_RB                        0000002B RG    01
BUFMAP                          00C0029A R     01
BUFUNM                          000002DE R     01
BUILD_BUFFER                    0000054B RG    01
BUILD_BUF_EXIT                  00000596 R     01
BUILD_DG                        000004D7 RG    01
BUILD_MSG                       000004D7 RG    01
BUILD_MSG_EXIT                  00000548 R     01
CB$B_BUFMAP_CNT                 0000001C
CB$B_LPRTNUM                    00000025
CB$B_RPRTNUM                    00000026
CB$B_STATUS                     0000000B
CB$B_TYPE                       0000000A
CB$K_LENGTH                     00000046
CB$L_BLINK                      00000004
CB$L_CDT                        0000000C
CB$L_FLINK                      00000000
CB$L_PDT                        00000010
CB$L_RB                         00000042
CB$Q_MAPQ                       00000014
CB$Q_MNTMAPQ                    0000001D
CB$T_LPRTNAM                    0000002E
CB$T_RPROCNAM                   00000032
CB$W_SIZE                       00000008
CDRP$B_CD_TYPE                = 0000000A
CDRP$B_FLAGS                    00000040
CDRP$C_BT_LEN                 = 00000040
CDRP$C_CY_LEN                   00000080
CDRP$K_CY_LEN                   00000080
CDRP$L_BUFADR                   0000005C
CDRP$L_BUFLEN                   00000060
CDRP$L_CB_BL                    00000074
CDRP$L_CB_FL                    00000070
CDRP$L_CDT                    = 00000024
CDRP$L_FQBL                   = 00000004
CDRP$L_FQFL                   = 00000000
CDRP$L_LBUFH_AD               = 0000002C
CDRP$L_MAPQ_BL                  0000007C
CDRP$L_MAPQ_FL                  00000078
CDRP$L_MNT_NAM                  00000064
CDRP$L_MSG_BUF                = 0000001C
CDRP$L_RBUFH_AD               = 00000034
CDRP$L_RB_BL                    0000006C
CDRP$L_RB_FL                    00000068
CDRP$L_RSPID                  = 00000020
CDRP$L_RWCPTR                 = 00000028
CDRP$T_CY_LBUFHNDL              00000044
CDRP$T_CY_RBUFHNDL              00000050
CDRP$W_CDRPSIZE               = 00000008
CHECK_CONN_CTLR                 00000225 RG    01
CHECK_CONN_RSP                  000001F0 RG    01
CHECK_CTP_REQ                   00000271 RG    01
CIBHAN$L_BNAME                = 00000004
CLEAN_CDRP                      00000181 RG    01
CLEAN_CONN                      000000EC RG    01
CLEAR_BUFFER                    000000E0 RG    01
COM$DRVDEALMEM                  ********   X   01
CONFIG                          000003D4 R     01
CONNECT                         00000412 R     01
CONN_LIST                       ********   X   01
CONTROLLER_NAME                 ********   X   01
COUNTS                          000003E9 R     01
CRB$L_DUETIME                 = 00000018
CTP$ACTCOUNT                    00000006
CTP$BUFLENGTH                   0000000C
CTP$BUFLNAME                    00000010
CTP$BUFLOFSET                   00000014
CTP$BUFMAPREQ                   00000001
CTP$BUFMAPRSP                   00000041
CTP$BUFRNAME                    00000018
CTP$BUFROFSET                   0000001C
CTP$BUFTYPE                     00000005
CTP$BUFUNMREQ                   00000002
CTP$BUFUNMRSP                   00000042
CTP$CDATPREV                    00000002
CTP$CDATPTYPE                   00000000
CTP$CDATPVERS                   00000001
CTP$CFGPOSTS                    0000000B
CTP$CFGP1STS                    0000000C
CTP$CNTFLG                      0000000A
CTP$CNTRDISCDG                  00000024
CTP$CNTRPOACK                   0000000C
CTP$CNTRPONAK                   00000010
CTP$CNTRPONRSP                  00000014
CTP$CNTRP1ACK                   00000018
CTP$CNTRP1NAK                   0000001C
CTP$CNTRP1NRSP                  00000020
CTP$CONDRST                   = 00000001
CTP$CONFIGREQ                   00000009
CTP$CONFIGRSP                   00000049
CTP$CONNECTREQ                  0000000B
CTP$CONNECTRSP                  0000004B
CTP$COUNTSREQ                   0000000A
CTP$COUNTSRSP                   0000004A
CTP$DELAY                       00000005
CTP$EXTEND                      00000008
CTP$FAKEBUF                   = 00000002
CTP$FINISHREQ                   0000000C
CTP$FINISHRSP                   00C0004C
CTP$FMASK                       00000006
CTP$FUNCTREQ                    00000000
CTP$FUNCTRSP                    00000040
CTP$GENCONST                    00000009
CTP$GENDATA                     0000000E
CTP$GENDGRREQ                   00000005
CTP$GENDGRRSP                   00000045
CTP$GENFBPAIR                 = 00000001
CTP$GENFFILL                  = 00000000
CTP$GENFIMAGE                 = 00000003
```

```
CTP$GENFNODE                =  00000002
CTP$GENFUNCT                   00000008
CTP$GENLENGTH                  0000000C
CTP$GENMSGREQ                  00000004
CTP$GENMSGRSP                  00000044
CTP$GENRSTREQ                  00000006
CTP$GENRSTRSP                  00000046
CTP$GENSTRREQ                  00000007
CTP$GENSTRRSP                  00000047
CTP$IMAGEDATA                  0000000E
CTP$KEEPCNT                 =  00000001
CTP$MAINTBUF                =  00000003
CTP$MAXCMDOPC                  00000011
CTP$MBUFMAPREQ                 0000000D
CTP$MBUFMAPRSP                 0000004D
CTP$MBUFUNMREQ                 0000000E
CTP$MBUFUNMRSP                 0000004E
CTP$MNTMOVETO               =  00000001
CTP$MOVBUFREQ                  00000003
CTP$MOVBUFRSP                  00000043
CTP$MOVETO                  =  00000001
CTP$MOVETYPE                   00000008
CTP$MOVMBUFREQ                 0000000F
CTP$MOVMBUFRSP                 0000004F
CTP$MSTATEREQ                  00000010
CTP$MSTATERSP                  00000050
CTP$NOACTFLAG                  0000000A
CTP$NOACTOFF                =  00000001
CTP$NOACTREQ                   00000008
CTP$NOACTRSP                   00000048
CTP$NONSENSE                =  000000FE
CTP$NORMSTATE               =  00000001
CTP$OPCODE                     00000000
CTP$OPEXPAND                   000000FF
CTP$OTHERNODE                  00000009
CTP$PKTMULT                    0000000B
CTP$PKTSIZ                     0000000A
CTP$PS576                   =  00000001
CTP$REFERENCE                  00000001
CTP$REPCOUNT                   00000006
CTP$RESERV10                   0000000A
CTP$RESERV11                   0000000B
CTP$RESERV12                   0000000C
CTP$RESERV20                   00000014
CTP$RESERV5                    00000005
CTP$RESERV6                    00000006
CTP$RESERV7                    00000007
CTP$RESERV9                    00000009
CTP$REVISION                =  00000000  G
CTP$SPECIFADR               =  00000001
CTP$STARTADR                   0000000C
CTP$STATUS                     00000005
CTP$VERSION                 =  00000003  G
CY$END                         000006A8 RG     01
CY$L_SAVED_CRB                 ******** X     01
CY$MONITOR                     00000675 RG     01
DEALLOC_CDRP                   000001C7 R      01
DEALLOC_MSG_DG                 000001DA RG     01
DEALLOC_RB                     00000157 RG     01
DYN$C_CDRP                  =  00000039
DYN$C_CD_CDDB               =  00000001
DYN$C_CIDG                  =  0000003B
DYN$C_IRP                   =  0000000A
ENTER_CONN_LIST                0000025B RG     01
EXE$ALONONPAGED                ******** X     01
EXE$GL_ABSTIM                  ******** X     01
EXEC_LIST                      ******** X     01
FIND_MAINT_BUFFER              000005C5 RG     01
FIND_MAPPED_BUFFER             00000599 RG     01
FINISH                         00000427 R      01
FUNCT                          00000297 R      01
GENDG                          00000369 R      01
GENMSG                         00000344 R      01
GENRST                         0000038E R      01
GENSTR                         000003A4 R      01
INVALID_REQUEST                000004CB R      01
IRP$B_TYPE                  =  0000000A
IRP$K_CDRP                  =  00000060
IRP$W_SIZE                  =  00000008
MAINLINE                       ******** X     01
MAP_QUEUE                      ******** X     01
MAX_BUF_LEN                    ******** X     01
MAX_NODES                      ******** X     01
MBUFMAP                        0000042A R      01
MBUFUNM                        00000462 R      01
MNT_MAP_QUEUE                  ******** X     01
MONITOR_LIST                   ******** X     01
MOVBUF                         0000030F R      01
MOVMBUF                        00000483 R      01
MSTATE                         000004B9 R      01
NOACT                          000003B8 R      01
PDT$L_DEALLOCDG             =  0000001C
PDT$L_DEALLOMSG             =  00000020
PDT$L_UNMAP                 =  00000064
PPD$B_DEF_ST                   0000001C
PPD$B_FLAGS                    0000000F
PPD$B_HWVERS                   00000034
PPD$B_LBDATA                   00000012
PPD$B_LCB_0                    00000012
PPD$B_LCB_LPORT                00000010
PPD$B_LCB_NPORT                0000000F
PPD$B_LCB_OPC                  00000011
PPD$B_LCB_PORT                 0000000E
PPD$B_OPC                      00C0000E
PPD$B_PORT                     0000000C
PPD$B_PROTOCOL                 0000001A
PPD$B_RSTATE                   00000025
PPD$B_RST_PORT                 00000024
PPD$B_STATUS                   0000000D
PPD$B_SWFLAG                   0000000B
PPD$B_SYSTEMID                 00000014
PPD$B_TYPE                     0000000A
PPD$C_LB_LENGTH                00000046
PPD$C_LCB_DATA                 00000013
```

```
PPD$C_LENGTH                    00000012
PPD$C_MIN_DGSIZ                 00000050
PPD$K_LB_LENGTH                 00000046
PPD$K_LENGTH                    00000012
PPD$L_BLINK                     00000004
PPD$L_DG_DISC                   00000028
PPD$L_FLINK                     00000000
PPD$L_IN_VCD                    00000018
PPD$L_LBCRC                     00000042
PPD$L_PO_ACK                    00000010
PPD$L_PO_NAK                    00000014
PPD$L_PO_NRSP                   00000018
PPD$L_P1_ACK                    0000001C
PPD$L_P1_NAK                    00000020
PPD$L_P1_NRSP                   00000024
PPD$L_REC_BOFF                  00000028
PPD$L_REC_NAME                  00000024
PPD$L_RPORT_FCN                 00000020
PPD$L_RPORT_REV                 0000001C
PPD$L_RPORT_TYP                 00000018
PPD$L_SND_BOFF                  00000020
PPD$L_SND_NAME                  0000001C
PPD$L_ST_ADDR                   00000018
PPD$L_XCT_LEN                   00000018
PPD$Q_CURTIME                   00000048
PPD$Q_NODENAME                  00000040
PPD$Q_SWINCARN                  00000028
PPD$Q_XCT_ID                    00000010
PPD$T_HWTYPE                    00000030
PPD$T_SWTYPE                    00000020
PPD$T_SWVERS                    00000024
PPD$W_LCB_LEN7                  0000000C
PPD$W_LENGTH                    00000010
PPD$W_MASK                      00000010
PPD$W_MAXDG                     0000001C
PPD$W_MAXMSG                    0000001E
PPD$W_MTYPE                     00000012
PPD$W_M_VAL                     00000014
PPD$W_SIZE                      00000008
QUEUE_DELAY                     000005F1 RG     01
RB$B_STATUS                     0000000B
RB$B_TYPE                       0000000A
RB$K_LENGTH                     0000002A
RB$L_BLINK                      00000004
RB$L_CB                         0000000C
RB$L_CDRP                       0000001C
RB$L_CDT                        00000014
RB$L_CTPBUF                     00000010
RB$L_DGBUF                      00000026
RB$L_FLINK                      00000000
RB$L_PDT                        00000018
RB$W_ACTCNT                     00000024
RB$W_DELAY                      00000020
RB$W_REPEAT                     00000022
RB$W_SIZE                       00000008
RB_M_CANCEL                   = 00000002
RB_V_CONN                     = 00000000
RESPONDER_NAME                  ********    X   01
SCS$B_PPD                     = FFFFFFE0
SCS$DEALL_RSPID                 ********    X   01
SCS$GW_MAXDG                    ********    X   01
SCS$GW_MAXMSG                   ********    X   01
SCS$K_APPL_BASE               = 00000000
SIZ...                        = 00000001
SOFTWARE_DEBUG                = 00000001
SS$_NORMAL                    = 00000001
TIMER_INTR                      0000061B RG     01
TIMER_QUEUE                     ********    X   01
TQB$B_STATUS                    0000000B
TQB$B_TYPE                      0000000A
TQB$K_LENGTH                    00000014
TQB$L_BLINK                     00000004
TQB$L_DUETIM                    0000000C
TQB$L_FLINK                     00000000
TQB$L_RB                        00000010
TQB$W_SIZE                      00000008
VALID_REQUEST                   000004D3 R      01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	000006A8 (1704.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	000000FF (255.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	89	00:00:00.33	00:00:01.13
Command processing	114	00:00:00.59	00:00:01.39
Pass 1	509	00:00:22.32	00:00:26.71
Symbol table sort	5	00:00:02.63	00:00:03.18
Pass 2	250	00:00:04.61	00:00:05.84
Symbol table output	34	00:00:00.27	00:00:00.46
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1007	00:00:30.21	00:00:38.76

The working set limit was 2250 pages.
106227 bytes (208 pages) of virtual memory were used to buffer the intermediate code.
There were 90 pages of symbol table space allocated to hold 1530 non-local and 127 local symbols.
1515 source lines were read in Pass 1, producing 20 object records in Pass 2.
54 pages of virtual memory were used to define 44 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CYDRIVER]PALIB.MLB;1	2
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	13
DRB2:[SHULL.EVXCI.CYDRIVER]CYLIB.MLB;4	10
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	7
TOTALS (all libraries)	32

1845 GETS were required to define 32 macros.

There were no errors, warnings or information messages.

MACRO/LIS CYMISC+CYLIB/LIB+SYS$LIBRARY:LIB/LIB+CYDRIVER$DIR:PALIB/LIB

Table of contents

```
0000     1          .TITLE  CYTABLES
0000     2          .IDENT  'V6-000'
0000     3  ;*****************************************************************************
0000     4  ;
0000     5  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     6  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     7  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     8  ;
0000     9  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000    10  ; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    11  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    12  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON.  NO
0000    13  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    14  ;
0000    15  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    16  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    17  ;
0000    18  ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000    19  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    20  ;
0000    21  ;*****************************************************************************
0000    22  ;
0000    23  ;+
0000    24  ;
0000    25  ; FACILITY:      VAX/VMS DIAGNOSTIC CLASS DRIVER
0000    26  ;
0000    27  ; VERSION:       4-002
0000    28  ;
0000    29  ; ABSTRACT:      This module contains the driver tables, as well as the
0000    30  ;                driver internal storage for the CI responder class driver.
0000    31  ;
0000    32  ; AUTHOR:        Jim Klumpp  16-JUL-81
0000    33  ;
0000    34  ; MODIFIED BY:   Jim Klumpp   6-MAY-83
0000    35  ;
0000    36  ;       6-000    Dave Shull      07-July-1984
0000    37  ;                VMS V4 Modifications/Release
0000    38  ;
0000    39  ;       4-002    Dave Shull      14-June-1984
0000    40  ;                Added CY$L_SAVED_CDT for saving address of Connect Descriptor
0000    41  ;                Table.
0000    42  ;       4-001    Dave Shull      06-June-1984
0000    43  ;                Change Module Revision to causes CYDRIVER Revision change,
0000    44  ;                in the header, when linked.
0000    45  ;-
0000    46
```

```
                     0000     48
00000001             0000     49 SOFTWARE_DEBUG=1                        ; Enable assembly of debug code
                     0000     50
                     0000     51         .DEFAULT DISPLACEMENT,WORD
                     0000     52         .ENABLE  SUPPRESSION
                     0000     53         .SBTTL   DEFINITIONS
                     0000     54
                     0000     55
                     0000     56 ;  System definitions (LIB.MLB):
                     0000     57
                     0000     58         $CRBDEF                         ; Define CRB offsets
                     0000     59         $DDBDEF                         ; Define DDB offsets
                     0000     60         $DPTDEF                         ; Define DPT offsets
                     0000     61         $UCBDEF                         ; Define UCB offsets
                     0000     62         $VECDEF                         ; Define IDB offsets
                     0000     63         $DYNDEF                         ; Define DYN Values
                     0000     64
                     0000     65
                     0000     66 ;  PADRIVER definitions (PALIB.MLB):
                     0000     67
                     0000     68
                     0000     69
                     0000     70 ;  CYDRIVER definitions (CYLIB.MLB):
                     0000     71
                     0000     72         $CTPDEF                         ; Define CTP symbols
                     0000     73
                     0000     74
```

```
                        0000     76
                        0000     77          .SBTTL  DRIVER PROLOGUE TABLE
                        0000     78
                        0000     79          DPTAB   -                                       ; DPT creation macro
                        0000     80                  END=CY$END,-                            ; End of driver
                        0000     81                  ADAPTER=NULL,-                          ; Adapter type
                        0000     82  ;               FLAGS=DPT$M_SCS,-                       ; SCS driver required
                        0000     83                  UCBSIZE=UCB$K_LENGTH,-                  ; Standard UCB length
                        0000     84                  MAXUNITS=1,-                            ;
                        0000     85                  NAME=CYDRIVER                           ; Driver name
                        0038     86
                        0038     87          DPT_STORE INIT                                  ; DPT initial values
                        0038     88
                        0038     89          DPT_STORE UCB,UCB$B_FIPL,B,IPL$_SCS             ; Fork IPL in UCB
                        003C     90          DPT_STORE UCB,UCB$L_DEVCHAR,L,-                 ; Device characteristics
                        003C     91                    <DEV$M_AVL!-                          ; Device available
                        003C     92                     DEV$M_IDV!-                          ; Input device
                        003C     93                     DEV$M_ODV>                           ; Output device
                        0043     94          DPT_STORE UCB,UCB$B_DIPL,B,21                   ; Device IPL
                        0047     95          DPT_STORE CRB,CRB$L_INTD+4,D,CY$INTR            ; Responder interrupt routine
                        004C     96
                        004C     97          DPT_STORE REINIT                                ; DPT reload values
                        004C     98
                        004C     99          DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,-       ; Responder init routine
                        004C    100                    D,CY$INIT
                        0051    101          DPT_STORE DDB,DDB$L_DDT,D,CY$DDT                ; DDT address
                        0056    102
                        0056    103          DPT_STORE END                                   ; End of DPT
                        0000    104
                        0000    105
                        0000    106
                        0000    107          .SBTTL  DRIVER DISPATCH TABLE
                        0000    108
                        0000    109          DDTAB   DEVNAM=CY,-                             ; Driver dispatch table
                        0000    110                  START=FATAL_QIO,-                       ; Start I/O routine
                        0000    111                  FUNCTB=CY$FUNCTABLE                     ; Function decision table
                        0038    112  ;               UNITINIT=CY$INIT                        ; Responder init routine.
                        0038    113
                        0038    114
                        0038    115
                        0038    116          .SBTTL  FUNCTION DECISION TABLE
                        0038    117
                        0038    118  CY$FUNCTABLE:
                        0038    119
                        0038    120          FUNCTAB,-                                       ; Valid functions:
                        0038    121                  <>                                      ; None
                        0040    122          FUNCTAB,-                                       ; Buffered functions:
                        0040    123                  <>                                      ; None
                        0048    124
                        0048    125
                        0048    126  FATAL_QIO:
                        0048    127  CY$INTR:
                        0048    128
                  05    0048    129          RSB
                        0049    130
                        0049    131
                        0049    132
```

```
                        0049   134
                        0049   135          .SBTTL  TUNING PARAMETERS
                        0049   136 ;++
                        0049   137 ;
                        0049   138 ;  Parameters that affect the operation of the responder and the quality
                        0049   139 ;  (speed, efficiency) of communications between it and other nodes on
                        0049   140 ;  the CI.
                        0049   141 ;
                        0049   142 ;--
                        0049   143
00000004                0049   144 INIT_CREDIT     ==      4                       ; Initial credits
00000001                0049   145 MIN_SEND_CR     ==      1                       ; Minimum send credits
00000010                0049   146 MAX_NODES       ==      16                      ; Max nodes in the cluster
00000002                0049   147 MAX_LOC_PORTS   ==      2                       ; Max local ports
00000002                0049   148 INIT_DG_BUF     ==      2                       ; Initial datagram buffers
00000400                0049   149 MAX_BUF_LEN     ==      1024                    ; Max read/write buf length
00000002                0049   150 MAX_BUF_MAP     ==      2                       ; Max mapped buffers per conn
                        0049   151
                        0049   152
                        0049   153
                        0049   154          .SBTTL  CONN_LIST,      Connection list
                        0049   155 ;+
                        0049   156 ;
                        0049   157 ;  CONN_LIST
                        0049   158 ;
                        0049   159 ;  Below are the connection block queues for each of the possible
                        0049   160 ;  ports in the cluster. Each time a connection is accepted with another
                        0049   161 ;  port in the cluster, a connection block is allocated and is linked to
                        0049   162 ;  the appropriate connection block queue.
                        0049   163 ;
                        0049   164 ;  The connection block queues are initialized when the driver is
                        0049   165 ;  loaded (by the initialization routine).
                        0049   166 ;
                        0049   167 ;-
                        0049   168
000000C9                0049   169 CONN_LIST::      .BLKQ   MAX_NODES
                        00C9   170
                        00C9   171
                        00C9   172
                        00C9   173          .SBTTL  EXEC_LIST,      Responder execution list
                        00C9   174 ;+
                        00C9   175 ;
                        00C9   176 ;  EXEC_LIST
                        00C9   177 ;
                        00C9   178 ;  When CTP requests are received, the request block associated with the
                        00C9   179 ;  request is queued to the END of the execution list. The mainline routine
                        00C9   180 ;  simply takes the request at the TOP of the execution list, executes one
                        00C9   181 ;  repetition of the specified command, and requeues the request block at
                        00C9   182 ;  the end of the execution list. When a CTP request is complete, a CTP
                        00C9   183 ;  response is returned to the controller and the request block is removed
                        00C9   184 ;  from the execution list and deallocated.
                        00C9   185 ;
                        00C9   186 ;-
                        00C9   187
000000D1                00C9   188 EXEC_LIST::      .BLKQ
                        00D1   189
                        00D1   190
```

```
                        00D1    191
                        00D1    192             .SBTTL  MAP_QUEUE,      Mapped buffer queue
                        00D1    193 ;+
                        00D1    194 ;
                        00D1    195 ;  MAP QUEUE
                        00D1    196 ;
                        00D1    197 ;  When buffer are mapped, they are queued to the responder's map queue so
                        00D1    198 ;  the buffer can found easily prior to performing buffer transfers. However,
                        00D1    199 ;  mapped buffers are also queued to the particular connection block from
                        00D1    200 ;  which the buffer map request was received so that the buffer can be unmapped
                        00D1    201 ;  when the connection goes away.
                        00D1    202 ;
                        00D1    203 ;-
                        00D1    204
                000000D9 00D1   205 MAP_QUEUE::     .BLKQ
                        00D9    206
                        00D9    207
                        00D9    208
                        00D9    209             .SBTTL  MNT_MAP_QUEUE,  Maintenance buffer mapped queue
                        00D9    210 ;+
                        00D9    211 ;
                        00D9    212 ;  MNT_MAP_QUEUE
                        00D9    213 ;
                        00D9    214 ;  This queue contains CDRPs for mapped maintenance buffers. Unlike
                        00D9    215 ;  standard buffers, maintenance buffers must be explicitly unmapped
                        00D9    216 ;  by the controller.
                        00D9    217 ;
                        00D9    218 ;-
                        00D9    219
                000000E1 00D9   220 MNT_MAP_QUEUE:: .BLKQ
                        00E1    221
                        00E1    222
                        00E1    223
                        00E1    224
                        00E1    225             .SBTTL  TIMER_QUEUE,    Delay timer queue
                        00E1    226 ;+
                        00E1    227 ;
                        00E1    228 ;  TIMER QUEUE
                        00E1    229 ;
                        00E1    230 ;
                        00E1    231 ;  Timer queue blocks are linked on the timer queue to cause delays before
                        00E1    232 ;  executing CTP requests, and to cause timeouts when the responder is
                        00E1    233 ;  expecting to be returned CTP responses (in the case of the activity
                        00E1    234 ;  request).
                        00E1    235 ;
                        00E1    236 ;  The routine TIMER_INT is woken periodically if the timer queue is not
                        00E1    237 ;  empty and checks the first timer queue block on the timer queue. It the
                        00E1    238 ;  duetime field has expired, the timer queue block is removed, and the
                        00E1    239 ;  appropriate action is taken.
                        00E1    240 ;
                        00E1    241 ;-
                        00E1    242
                000000E9 00E1   243 TIMER_QUEUE::   .BLKQ
                        00E9    244
                        00E9    245
                        00E9    246
                        00E9    247             .SBTTL  MONITOR_LIST,   Activity monitor list
```

```
                                    00E9    248 ;+
                                    00E9    249 ;
                                    00E9    250 ;  MONITOR LIST
                                    00F9    251 ;
                                    00E9    252 ;  The following two longwords are used as the header for a monitor list.
                                    00E9    253 ;  Each time through the mainline routine, a monitor block is allocated
                                    00E9    254 ;  and linked to the end of this list. The monitor block is used to store
                                    00F9    255 ;  the opcode of the CTP request that is being executed, as well as the
                                    00E9    256 ;  command reference number and repeat count.
                                    00E9    257 ;
                                    00E9    258 ;-
                                    00E9    259
                                    00E9    260         .IF DF SOFTWARE_DEBUG           ; Assemble for software debug only
                                    00E9    261
                           000000F1 00E9    262 MONITOR_LIST::  .BLKQ
                                    00F1    263
                                    00F1    264         .ENDC                           ; End of conditional assembly
                                    00F1    265
                                    00F1    266
                                    00F1    267
45 44 4E 4F 50 53 45 52 24 50 54 43 00F1    268 RESPONDER_NAME::  .ASCII /CTP$RESPONDER   /     ; Responder process name
                     20 20 20 52    00FD
4C 4C 4F 52 54 4E 4F 43 24 50 54 43 0101    269 CONTROLLER_NAME:: .ASCII /CTP$CONTROLLER  /     ; Controller process name
                     20 20 52 45    010D
                                    0111    270
                                    0111    271 CONNECT_DATA::                          ; Connection data
                                00  0111    272                 .BYTE   CTP$CTP         ; Protocol type
                                03  0112    273                 .BYTE   CTP$VERSION     ; Protocol version
                                00  0113    274                 .BYTE   CTP$REVISION    ; Protocol revision
                          00000141  0114    275                 .BLKB   45              ; MBZ
                                    0141    276
                          00000145  0141    277 CY$L_SAVED_UCB::  .BLKL                 ; Saved UCB address
                          00000149  0145    278 CY$L_SAVED_CRB::  .BLKL                 ; Saved CRB address
                          0000014D  0149    279 CY$L_SAVED_CDT::  .BLKL                 ; Saved CDT address                 [4-002]
                          00000151  014D    280 CY$L_SPARE_CDRP:: .BLKL                 ; Address of spare CDRP
                                    0151    281
                                    0151    282
                                    0151    283         .END
```

$$$	= 00000020	R	02	CTP$IMAGEDATA	0000000E				
$$OP	= 00000002			CTP$MAXCMDOPC	00000011				
AT$_NULL	********	X	02	CTP$MBUFMAPREQ	000C000D				
CONNECT_DATA	00000111	RG	03	CTP$MBUFMAPRSP	0000004D				
CONN_LIST	00000049	RG	03	CTP$MBUFUNMREQ	0000000E				
CONTROLLER_NAME	00000101	RG	03	CTP$MBUFUNMRSP	0000004E				
CRB$L_INTD	= 00000024			CTP$MOVBUFREQ	00000003				
CTP$ACTCOUNT	00000006			CTP$MOVBUFRSP	00000043				
CTP$BUFLENGTH	0000000C			CTP$MOVETYPE	00000008				
CTP$BUFLNAME	00000010			CTP$MOVMBUFREQ	0000000F				
CTP$BUFLOFSET	00000014			CTP$MOVMBUFRSP	0000004F				
CTP$BUFMAPREQ	00000001			CTP$MSTATEREQ	00000010				
CTP$BUFMAPRSP	00000041			CTP$MSTATERSP	00000050				
CTP$BUFRNAME	00000018			CTP$NOACTFLAG	0000000A				
CTP$BUFROFSET	0000001C			CTP$NOACTREQ	00000008				
CTP$BUFTYPE	00000005			CTP$NOACTRSP	00000048				
CTP$BUFUNMREQ	00000002			CTP$OPCODE	00000000				
CTP$BUFUNMRSP	00000042			CTP$OPEXPAND	000000FF				
CTP$CDATPREV	00000002			CTP$OTHERNODE	00000009				
CTP$CDATPTYPE	00000000			CTP$PKTMULT	0000000B				
CTP$CDATPVERS	00000001			CTP$PKTSIZ	0000000A				
CTP$CFGPOSTS	0000000B			CTP$REFERENCE	00000001				
CTP$CFGP1STS	0000000C			CTP$REPCOUNT	00000006				
CTP$CNTFLG	0000000A			CTP$RESERV1C	0000000A				
CTP$CNTRDISCDG	00000024			CTP$RESERV11	0000000B				
CTP$CNTRPOACK	0000000C			CTP$RESERV12	0000000C				
CTP$CNTRPONAK	00000010			CTP$RESERV20	00000014				
CTP$CNTRPONRSP	00000014			CTP$RESERV5	00000005				
CTP$CNTRP1ACK	00000018			CTP$RESERV6	00000006				
CTP$CNTRP1NAK	0000001C			CTP$RESERV7	00000007				
CTP$CNTRP1NRSP	00000020			CTP$RESERV9	00000009				
CTP$CONFIGREQ	00000009			CTP$REVISION	= 00000000	G			
CTP$CONFIGRSP	00000049			CTP$STARTADR	0000000C				
CTP$CONNECTREQ	0000000B			CTP$STATUS	00000005				
CTP$CONNECTRSP	0000004B			CTP$VERSION	= 00000003	G			
CTP$COUNTSREQ	0000000A			CY$DDT	00000000	RG	03		
CTP$COUNTSRSP	0000004A			CY$END	********	X	02		
CTP$CTP	= 00000000			CY$FUNCTABLE	00000038	R	03		
CTP$DELAY	00000005			CY$INIT	********	X	02		
CTP$EXTEND	00000008			CY$INTR	00000048	R	03		
CTP$FINISHREQ	0000000C			CY$L_SAVED_CDT	00000149	RG	03		
CTP$FINISHRSP	0000004C			CY$L_SAVED_CRB	00000145	RG	03		
CTP$FMASK	00000006			CY$L_SAVED_UCB	00000141	RG	03		
CTP$FUNCTREQ	00000000			CY$L_SPARE_CDRP	0000014D	RG	03		
CTP$FUNCTRSP	00000040			DDB$C_DDT	= 0000000C				
CTP$GENCONST	00000009			DEV$M_AVL	********	X	02		
CTP$GENDATA	0000000E			DEV$M_IDV	********	X	02		
CTP$GENDGRREQ	00000005			DEV$M_ODV	********	X	02		
CTP$GENDGRRSP	00000045			DPT$C_LENGTH	= 00000038				
CTP$GENFUNCT	00000008			DPT$C_VERSION	= 00000004				
CTP$GENLENGTH	0000000C			DPT$INITAB	00000038	R	02		
CTP$GENMSGREQ	00000004			DPT$REINITAB	0000004C	R	02		
CTP$GENMSGRSP	00000044			DPT$TAB	00000000	R	02		
CTP$GENRSTREQ	00000006			DYN$C_CRB	= 00000005				
CTP$GENRSTRSP	00000046			DYN$C_DDB	= 00000006				
CTP$GENSTRREQ	00000007			DYN$C_DPT	= 0000001E				
CTP$GENSTRRSP	00000047			DYN$C_UCB	= 00000010				

```
EXEC_LIST                       00000019 RG    03
FATAL_QIO                       00000048 R     03
FUNCTAB_LEN                 =   00000010
INIT_CREDIT                 =   00000004  G
INIT_DG_BUF                 =   00000002  G
IOC$MNTVER                      ********    X  03
IOC$RETURN                      ********    X  03
IPL$_SCS                        ********    X  02
MAP_QUEUE                       000000D1 RG    03
MASKH                       =   00000C00
MASKL                       =   00000000
MAX_BUF_LEN                 =   00000400  G
MAX_BUF_MAP                 =   00000002  G
MAX_LOC_PORTS               =   00000002  G
MAX_NODES                   =   00000010  G
MIN_SEND_CR                 =   00000001  G
MNT_MAP_QUEUE                   000000D9 RG    03
MONITOR_LIST                    000000E9 RG    03
RESPONDER_NAME                  000000F1 RG    03
SOFTWARE_DEBUG              =   00000001
TIMER_QUEUE                     000000E1 RG    03
UCB$B_DIPL                  =   0000005E
UCB$B_FIPL                  =   0000000B
UCB$K_LENGTH                =   00000090
UCB$L_DEVCHAR               =   00000038
VEC$L_UNITINIT              =   00000018
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	000000FF (255.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$$$105_PROLOGUE	00000057 (87.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
$$$115_DRIVER	00000151 (337.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	94	00:00:00.36	00:00:01.14
Command processing	104	00:00:00.59	00:00:01.22
Pass 1	303	00:00:10.15	00:00:11.32
Symbol table sort	0	00:00:01.10	00:00:01.83
Pass 2	54	00:00:01.51	00:00:01.97
Symbol table output	17	00:00:00.13	00:00:00.13
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	577	00:00:13.87	00:00:17.64

The working set limit was 1500 pages.
43540 bytes (86 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 747 non-local and 0 local symbols.

283 source lines were read in Pass 1, producing 16 object records in Pass 2.
31 pages of virtual memory were used to define 22 macros.

+----------------------------+
! Macro library statistics !
+----------------------------+

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CYDRIVER ALIB.MLB;1	0
SYS$SYSROOT:[SYSLIB]LIB.MLB,1	11
DRB2:[SHULL.EVXCI.CYDRIVER]CYLIB.MLB;4	1
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	4
TOTALS (all libraries)	16

1023 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS CYTABLES+CYLIB/LIB+SYS$LIBRARY:LIB/LIB+CYDRIVER$DIR:PALIB/LIB
```

B 1 Directory
C 1 Directory