

```
// LOG 1403
// NOHALT
*
*** PRINT LIBRARY FILE TO THE SYSTEM PRINTER.
*
// LOAD $MAINT,F1
// RUN
// COPY FROM-R2,LIBRARY-ALL,NAME-DIR,TO-PRINT
```

SYSTEM DIRECTORY FROM R2 VOLUME ID R2R2R2 06/06/10

LIBRARY AREA OVERVIEW	--LOCATION-- DISK ADDRESS	-----SIZE----- TRACKS SECTORS	--NUMBER-- OF ENTRIES
SOURCE LIBRARY ALLOCATED SIZE		175 4200	
OBJECT DIRECTORY ALLOCATED SIZE		3 72	863
OBJECT LIBRARY ALLOCATED SIZE		110 2640	
START OF LIBRARIES	008-00		
SYSTEM HISTORY AREA		0	
ROLLOUT/ROLLIN AREA		0	
SCHEDULER WORK AREA		0	
END OF LIBRARIES	292-23		

SOURCE LIBRARY SECTION

START OF SOURCE LIBRARY	008-00		
DIRECTORY SPACE		1 5	
PERMANENT LIBRARY SPACE		160 3828	
ACTIVE LIBRARY SPACE		160 3828	
AVAILABLE LIBRARY SPACE	167-17	15 367	
END OF SOURCE LIBRARY	182-23		

OBJECT LIBRARY SECTION

START OF OBJECT DIRECTORY	183-00		
AVAILABLE PERMANENT DIRECTORY ENTRIES	184-01-126		557
AVAILABLE TEMPORARY DIRECTORY ENTRIES	184-01-126		557
END OF OBJECT DIRECTORY	185-23		
START OF OBJECT LIBRARY	186-00		
AVAILABLE PERMANENT LIBRARY SPACE	265-12	27 660	
UNUSED SPACE FROM DELETED PERMANENTS		0 1	
AVAILABLE TEMPORARY LIBRARY SPACE	265-12	27 660	
UNUSED SPACE FROM DELETED TEMPORARIES		0 0	
ACTIVE LIBRARY SPACE		80 1907	
PERMANENT OBJECT LIBRARY SPACE		74 1758	
PERMANENT ROUTINE LIBRARY SPACE		7 149	
ALLOCATED END OF OBJECT LIBRARY	292-23		
EXTENDED END OF OBJECT LIBRARY	292-23		

SOURCE-175,DIRSIZE-3,OBJECT-110,SYSTEM-NO

SOURCE DIRECTORY FROM R2 VOL. ID R2R2R2 06/06/10

TYPE	NAME	FIRST @	LAST @	ATTRI	# SECTORS
P	\$CG1G1	167-03	167-04	P	0002
P	\$CG1G2	167-05	167-07	P	0003
P	\$CG1G3	167-08	167-10	P	0003
P	\$CG1G4	167-11	167-11	P	0001
P	\$CG1G5	167-12	167-13	P	0002
P	\$SGPTR	167-14	167-14	P	0001
P	\$SGPT2	167-15	167-15	P	0001
P	\$SGPVR	167-16	167-16	P	0001
S	\$CANB	163-05	163-16	P	0012
S	\$CGCND	145-09	145-15	P	0007
S	\$CGEND	008-05	008-09	P	0005
S	\$CGSET	008-10	009-02	P	0017
S	\$CGSMP	009-03	009-18	P	0016
S	\$CGSST	145-16	145-22	P	0007
S	\$CG1GI	145-23	146-08	P	0010
S	\$CG1GM	158-02	158-08	P	0007
S	\$DFOB	154-16	155-21	P	0030
S	\$DTOM	155-22	158-01	P	0052
S	\$EASC	165-12	167-02	P	0039
S	\$EBEG	158-09	159-05	P	0021
S	\$EBEQ	051-17	053-22	P	0054
S	\$EBSC	009-19	012-15	P	0069
S	\$EBSD	012-16	013-09	P	0018
S	\$ECFG	159-06	160-05	P	0024
S	\$ECOM	040-06	043-15	P	0082
S	\$ECPL	045-21	047-19	P	0047
S	\$ECRL	160-06	161-06	P	0025
S	\$EDFT	066-10	066-14	P	0005
S	\$EDSM	036-17	040-05	P	0085
S	\$EEQU	034-07	035-01	P	0019
S	\$EFAC	013-10	016-03	P	0066
S	\$EFDT	067-04	067-18	P	0015
S	\$EFIL	016-04	017-19	P	0040
S	\$EFT	066-15	066-21	P	0007
S	\$EGEN	017-20	030-00	P	0293
S	\$EHLF	161-07	161-20	P	0014
S	\$EIOB	045-11	045-20	P	0010
S	\$ELCB	049-15	051-01	P	0035
S	\$EMLA	030-01	030-15	P	0015
S	\$EMLD	030-16	032-15	P	0048
S	\$EPAS	035-02	036-03	P	0026
S	\$EPCT	161-21	162-18	P	0022
S	\$EPLG	032-16	033-13	P	0022
S	\$EROL	162-19	163-04	P	0010
S	\$ESEC	033-14	034-06	P	0017
S	\$ESIT	163-17	164-06	P	0014
S	\$ETAS	047-20	048-06	P	0011
S	\$ETCB	053-23	054-13	P	0015
S	\$ETCC	043-16	045-10	P	0043
S	\$ETMG	164-07	165-11	P	0029
S	\$ETML	051-09	051-16	P	0008
S	\$ETNT	051-02	051-08	P	0007
S	\$ETT	066-22	067-03	P	0006
S	\$ETUB	048-07	049-14	P	0032
S	\$E000	054-14	055-12	P	0023
S	\$E001	055-13	056-05	P	0017
S	\$E002	056-06	056-19	P	0014

TYPE	NAME	FIRST @	LAST @	ATTRI	# SECTORS
S	\$E003	056-20	057-19	P	0024
S	\$E030	057-20	062-03	P	0104
S	\$E033	062-04	063-19	P	0040
S	\$E038	063-20	066-09	P	0062
S	\$E060	067-19	079-23	P	0293
S	\$E065	080-00	089-13	P	0230
S	\$E070	089-14	094-04	P	0111
S	\$E072	094-05	102-02	P	0190
S	\$E075	102-03	106-09	P	0103
S	\$E080	114-00	118-18	P	0115
S	\$E082	106-10	107-18	P	0033
S	\$E085	107-19	113-23	P	0149
S	\$E087	131-20	135-15	P	0092
S	\$E090	135-16	137-09	P	0042
S	\$E092	118-19	129-14	P	0260
S	\$E093	137-10	143-22	P	0157
S	\$E094	143-23	145-08	P	0034
S	\$E095	129-15	131-19	P	0053
S	\$LOGD	036-04	036-16	P	0013
S	\$NCIO	146-09	151-01	P	0113
S	\$NCOM	151-02	151-16	P	0015
S	\$NOPV	151-17	152-09	P	0017
S	\$NPL	152-10	153-03	P	0018
S	\$NPLO	153-04	153-17	P	0014
S	\$NRTV	153-18	154-15	P	0022

OBJECT DIRECTORY FROM R2 VOL. ID R2R2R2 06/06/10

TYPE	NAME	DSK	ADD	CYL/SEC	TXT-CAT	LINK	ADD	RLD	DISP	ENTRY	PNT	CORE	SEC	ATTR	LEVEL	TOT	SEC
P	O	\$CCAAM	222/09	6F/24	012		8000		3C	8000		016		8000	006	0012	
P	O	\$CCAA1	195/08	61/A0	002		4000		E5	4000		016		8000	003	0002	
P	O	\$CCAA2	203/09	65/A4	002		4000		FC	4000		016		8000	006	0002	
P	O	\$CCBBA	195/11	61/AC	002		4200		FC	4200		016		8000	008	0002	
P	O	\$CCBBB	195/13	61/B4	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCBBC	195/15	61/BC	002		4200		FC	4200		016		8000	007	0002	
P	O	\$CCBBD	195/17	61/C4	002		4200		FC	4200		002		8000	006	0002	
P	O	\$CCBBE	195/19	61/CC	002		4200		FC	4200		002		8000	007	0002	
P	O	\$CCBBF	258/01	81/04	002		4200		FD	4200		016		8000	007	0003	
P	O	\$CCBBG	258/04	81/10	003		4200		D0	4200		016		8000	008	0003	
P	O	\$CCBBI	191/06	5F/98	002		4200		00	4200		002		8000	008	0003	
P	O	\$CCBBL	196/02	62/08	002		4200		FC	4200		016		8000	008	0002	
P	O	\$CCBBP	196/04	62/10	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCBBQ	196/06	62/18	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCBBR	196/08	62/20	002		4200		FC	4200		002		8000	006	0002	
P	O	\$CCBBS	196/10	62/28	002		4200		FC	4200		016		8000	008	0002	
P	O	\$CCBBU	195/21	61/D4	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCBB0	191/09	5F/A4	002		4200		FF	4200		016		8000	006	0003	
P	O	\$CCBB5	196/15	62/3C	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCBB7	212/14	6A/38	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCBB9	203/16	65/C0	002		4200		FC	4200		016		8000	008	0002	
P	O	\$CCCGR	197/08	62/A0	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCCMB	199/10	63/A8	002		4200		FC	4200		016		8000	007	0002	
P	O	\$CCCMD	199/13	63/B4	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCCMF	199/17	63/C4	001		4200		FE	4200		016		8000	003	0001	
P	O	\$CCCMO	199/20	63/D0	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCCMP	199/22	63/D8	001		4200		FE	4200		016		8000	006	0001	
P	O	\$CCCS2	202/23	65/5C	002		4200		FC	4200		016		8000	008	0002	
P	O	\$CCWC	204/02	66/08	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCWD	204/04	66/10	001		4200		FE	4200		016		8000	006	0001	
P	O	\$CCWR	213/05	6A/94	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCDB1	194/08	61/20	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCDB2	210/20	69/50	002		4200		FC	4200		016		8000	006	0002	
P	O	\$CCEEA	204/00	66/00	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEB	205/12	66/B0	002		4000		FC	4000		016		8000	005	0002	
P	O	\$CCEEC	211/00	69/80	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEED	211/02	69/88	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEE	211/04	69/90	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEF	211/06	69/98	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEG	211/08	69/A0	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEH	211/12	69/B0	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEL	211/14	69/B8	002		4000		FC	4085		016		8000	003	0002	
P	O	\$CCEEM	197/13	62/B4	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEN	211/10	69/A8	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEEP	197/11	62/AC	002		4000		FC	4000		016		8000	006	0002	
P	O	\$CCEE1	211/18	69/C8	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEE2	211/20	69/D0	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEE3	211/22	69/D8	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEE4	212/00	6A/00	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCEE5	212/02	6A/08	002		4000		FC	4000		016		8000	003	0002	
P	O	\$CCFDB	197/01	62/84	002		4000		FC	4011		016		8000	006	0002	
P	O	\$CCFDC	197/03	62/8C	002		4000		35	400C		016		8000	006	0002	
P	O	\$CCFDD	197/05	62/94	001		4000		EE	400B		016		8000	006	0001	
P	O	\$CCFDE	197/06	62/98	002		4000		FC	400C		016		8000	003	0002	
P	O	\$CCGPC	212/22	6A/58	002		4000		FC	4000		016		8000	006	0002	

TYPE	NAME	DSK	ADD	CYL/SEC	TXT-CAT	LINK	ADD	RLD	DISP	ENTRY	PNT	CORE	SEC	ATTR	LEVEL	TOT	SEC
P	O	\$CCGPF	195/10	61/A8	001	4000		FE		40B5		016		8000	003	0001	
P	O	\$CCGPK	213/02	6A/88	001	4000		FA		4000		016		8000	005	0001	
P	O	\$CCGPM	230/19	73/4C	002	4000		FC		4000		016		8000	005	0002	
P	O	\$CCGPR	201/02	64/88	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCGPT	201/04	64/90	002	4000		FC		4000		016		8000	003	0002	
P	O	\$CCIAB	194/10	61/28	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCIAQ	194/20	61/50	002	4000		F8		4000		016		8000	007	0002	
P	O	\$CCIAX	195/04	61/90	002	4000		F8		4000		016		8000	006	0002	
P	O	\$CCIAY	195/06	61/98	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCIDG	203/22	65/D8	002	4000		FC		4000		016		8000	005	0002	
P	O	\$CCIGA	197/15	62/BC	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCIGB	197/17	62/C4	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCINC	200/02	64/08	002	4000		F8		4000		016		8000	006	0002	
P	O	\$CCIRL	201/10	64/A8	002	4000		F8		4000		016		8000	006	0002	
P	O	\$CCIRX	201/16	64/C0	002	4000		F8		4000		016		8000	006	0002	
P	O	\$CCIRY	201/18	64/C8	002	4000		F8		4000		016		8000	006	0002	
P	O	\$CCLL2	199/04	63/90	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCLL3	212/06	6A/18	002	4000		FC		4000		016		8000	007	0002	
P	O	\$CCLL4	199/06	63/98	002	4000		FC		4000		016		8000	004	0002	
P	O	\$CCMMA	199/08	63/A0	002	4200		FC		4200		016		8000	006	0002	
P	O	\$CCMMC	199/12	63/B0	001	4200		FE		4200		016		8000	006	0001	
P	O	\$CCMME	199/15	63/BC	002	4200		FC		4200		016		8000	006	0002	
P	O	\$CCMMT	199/23	63/DC	001	4200		FE		4200		016		8000	006	0001	
P	O	\$CCMM9	199/01	63/84	002	4200		00		4200		016		8000	003	0003	
P	O	\$CCMPG	201/00	64/80	002	4200		FC		4200		016		8000	006	0002	
P	O	\$CCMSB	202/07	65/1C	002	4200		FC		4200		016		8000	006	0002	
P	O	\$CCMSC	202/09	65/24	001	4200		FE		4200		016		8000	003	0001	
P	O	\$CCMSF	202/10	65/28	001	4200		FE		4200		016		8000	006	0001	
P	O	\$CCMSK	202/13	65/34	001	4200		FE		4200		016		8000	006	0001	
P	O	\$CCMSP	202/16	65/40	002	4200		FC		4200		016		8000	006	0002	
P	O	\$CCMSQ	202/18	65/48	002	4200		FC		4200		016		8000	006	0002	
P	O	\$CCMT1	203/20	65/D0	001	4200		FE		4200		016		8000	006	0001	
P	O	\$CCMT2	203/21	65/D4	001	4200		FE		4200		016		8000	003	0001	
P	O	\$CCOOD	212/08	6A/20	002	4000		FC		4000		016		8000	007	0002	
P	O	\$CCOOH	200/12	64/30	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCOOP	210/22	69/58	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCOOR	212/12	6A/30	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCOOT	200/14	64/38	002	4000		FC		4000		016		8000	007	0002	
P	O	\$CCO01	200/16	64/40	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCO02	212/10	6A/28	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCO03	214/23	6B/5C	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCO04	200/18	64/48	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCO05	200/06	64/18	002	4000		FC		4000		016		8000	003	0002	
P	O	\$CCO06	200/20	64/50	002	4000		FC		4000		016		8000	003	0002	
P	O	\$CCO07	212/18	6A/48	002	4000		FC		4000		016		8000	003	0002	
P	O	\$CCO08	212/20	6A/50	002	4000		FC		4000		016		8000	005	0002	
P	O	\$CCO09	200/22	64/58	002	4000		FC		4000		016		8000	006	0002	
P	O	\$CCP	220/15	6E/3C	038	4000		38		42C5		088		C00A	003	0042	
P	O	\$CCPAL	236/12	76/30	076	4000		E8		4006		080		8082	007	0077	
P	O	\$CCPAS	244/17	7A/44	086	4000		00		4069		086		808A	006	0089	
P	O	\$CCPAU	208/17	68/44	036	4000		1C		4000		037		8080	006	0038	
P	O	\$CCPDD	204/06	66/18	023	8000		9C		8005		024		8080	006	0025	
P	O	\$CCPDF	186/04	5D/10	012	4000		F3		4405		072		80AA	006	0013	
P	O	\$CCPDT	239/23	77/DC	042	4000		68		4006		080		8000	008	0046	
P	O	\$CCPPF	235/09	75/A4	012	4000		F3		4405		072		80AA	006	0013	
P	O	\$CCPRB	186/17	5D/44	051	4000		CC		4006		072		9008	005	0052	
P	O	\$CCRR1	201/20	64/D0	002	4000		F9		4000		016		8000	006	0002	
P	O	\$CCRR2	201/22	64/D8	001	4000		FE		4000		016		8000	006	0001	

TYPE	NAME	DSK ADD	CYL/SEC	TXT-CAT	LINK	ADD RLD	DISP	ENTRY	PNT	CORE	SEC	ATTR	LEVEL	TOT SEC
P O	\$CCRR3	239/17	77/C4	002	4000		FE	4000	016	8000	006	0003		
P O	\$CCRR4	202/01	65/04	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCRR5	202/03	65/0C	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCRR6	202/05	65/14	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCSAS	194/22	61/58	002	4000		FC	4000	016	8000	005	0002		
P O	\$CCSAT	195/00	61/80	002	4000		FC	4015	016	8000	003	0002		
P O	\$CCSAU	195/02	61/88	002	4000		FC	4000	016	8000	003	0002		
P O	\$CCSCG	196/19	62/4C	002	4000		FC	4000	016	8000	007	0002		
P O	\$CCSCJ	213/00	6A/80	002	4000		FB	41AA	016	8000	006	0002		
P O	\$CCSCX	196/23	62/5C	002	4000		A4	4180	016	8000	006	0002		
P O	\$CCSMZ	214/21	6B/54	002	4000		FC	411B	016	8000	005	0002		
P O	\$CCSRE	201/07	64/9C	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCSRF	201/09	64/A4	001	4000		FE	405B	016	8000	006	0001		
P O	\$CCSRP	201/14	64/B8	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCSSA	219/04	6D/90	001	0000		46	0000	016	8000	003	0001		
P O	\$CCSSH	241/21	78/D4	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCSSS	202/20	65/50	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCSS0	235/06	75/98	002	4200		FF	4200	016	8000	006	0003		
P O	\$CCSTE	203/03	65/8C	001	4000		FE	4000	016	8000	005	0001		
P O	\$CCSTS	203/04	65/90	002	4000		FC	4000	016	8000	005	0002		
P O	\$CCSTZ	203/14	65/B8	002	4000		FC	41A9	016	8000	005	0002		
P O	\$CCSVA	203/18	65/C8	002	4000		FC	4000	016	8000	005	0002		
P O	\$CCSVB	196/17	62/44	002	4000		FC	4000	016	8000	005	0002		
P O	\$CCTCN	196/21	62/54	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCTDM	197/10	62/A8	001	4000		FE	4000	016	8000	003	0001		
P O	\$CCTFL	212/04	6A/10	002	4000		FC	4000	016	8000	005	0002		
P O	\$CCTF2	211/16	69/C0	002	4000		FC	40C2	016	8000	003	0002		
P O	\$CCTMG	199/18	63/C8	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCTNM	200/04	64/10	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCTOF	200/08	64/20	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCTOG	200/10	64/28	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCTQQ	201/06	64/98	001	4000		FE	4000	016	8000	005	0001		
P O	\$CCTRN	201/12	64/B0	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCTSO	202/14	65/38	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCXEJ	236/09	76/24	002	4000		FE	4000	016	8000	006	0003		
P O	\$CCXTD	203/01	65/84	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCXTF	206/02	67/08	002	4000		00	4000	016	8000	006	0003		
P O	\$CCXTG	201/23	64/DC	002	4000		F6	4000	016	8000	006	0002		
P O	\$CCXTH	200/00	64/00	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCXTK	239/20	77/D0	002	4000		FD	4000	016	8000	006	0003		
P O	\$CCXTN	213/07	6A/9C	002	4000		EF	4000	016	8000	006	0002		
P O	\$CCXTP	213/03	6A/8C	002	4000		54	4000	016	8000	007	0002		
P O	\$CCXTW	212/16	6A/40	002	4000		FC	4000	016	8000	006	0002		
P O	\$CCXTY	203/06	65/98	002	4000		FE	4000	016	8000	006	0003		
P O	\$CCYBL	188/21	5E/54	009	7600		62	7637	016	8000	003	0009		
P O	\$CCYBT	258/09	81/24	020	7600		DB	767B	016	8000	008	0021		
P O	\$CCYDF	226/03	71/0C	014	7600		EF	766E	016	8000	006	0015		
P O	\$CCYEA	190/04	5F/10	003	0000		00	0000	003	8000	002	0004		
P O	\$CCYEB	190/08	5F/20	003	0000		00	0000	003	8000	006	0004		
P O	\$CCYEC	190/12	5F/30	003	0000		00	0000	003	8000	002	0004		
P O	\$CCYE1	190/16	5F/40	003	0000		00	0000	003	8000	002	0004		
P O	\$CCYE2	190/20	5F/50	003	0000		C0	0000	003	8000	002	0003		
P O	\$CCYE3	190/23	5F/5C	003	0000		C0	0000	003	8000	002	0003		
P O	\$CCYE4	191/02	5F/88	003	0000		00	0000	003	8000	002	0004		
P O	\$CCYE5	210/07	69/1C	003	0000		00	0000	003	8000	003	0004		
P O	\$CCYE6	226/18	71/48	003	0000		00	0000	003	8000	003	0004		
P O	\$CCYE7	226/22	71/58	003	0000		00	0000	003	8000	003	0004		
P O	\$CCYE8	191/15	5F/BC	003	0000		00	0000	003	8000	002	0004		

TYPE	NAME	DSK ADD	CYL/SEC	TXT-CAT	LINK	ADD RLD	DISP	ENTRY	PNT	CORE	SEC	ATTR	LEVEL	TOT SEC
P O	\$CCYE9	191/19	5F/CC	003	0000	00	0000	0000	003	8000	003	0004		
P O	\$CCYHK	227/02	71/88	012	7600	E3	7678	016	8000	006	0012			
P O	\$CCYML	192/10	60/28	008	7600	64	7633	016	8000	003	0008			
P O	\$CCYMT	233/12	74/B0	019	7600	50	7669	016	8000	004	0020			
P O	\$CCYPG	223/17	6F/C4	021	7600	FD	76A4	016	8000	006	0022			
P O	\$CCYSP	234/08	75/20	010	0000	FD	0000	016	8000	006	0011			
P O	\$CCYSS	210/11	69/2C	008	0000	A6	005C	008	8000	003	0009			
P O	\$CCYSY	227/14	71/B8	009	7600	6D	7673	016	8000	005	0009			
P O	\$CCYTA	193/04	60/90	009	7600	AF	7637	016	8000	003	0009			
P O	\$CCYTN	193/13	60/B4	012	7600	71	767A	016	8000	003	0012			
P O	\$CCZCF	224/20	70/50	028	4000	00	4012	056	800A	006	0031			
P O	\$CCZCG	242/14	79/38	028	4000	00	4011	056	800A	006	0031			
P O	\$CCZCP	189/16	5E/C0	010	4400	00	4405	018	800A	004	0012			
P O	\$CCZCQ	192/18	60/48	008	4400	00	4405	018	800A	004	0010			
P O	\$CCZCR	231/00	73/80	008	4400	00	4407	056	800A	006	0010			
P O	\$CCZCS	235/22	75/D8	009	4400	00	4407	056	800A	005	0011			
P O	\$CC1PP	219/05	6D/94	033	4000	F5	4006	036	800A	003	0034			
P O	\$CC3CR	205/07	66/9C	004	8000	D9	8006	021	8008	006	0005			
P O	\$CC3CX	213/09	6A/A4	003	4000	F8	4000	003	8008	006	0004			
P O	\$CC3CY	203/11	65/AC	003	4000	42	4000	003	8008	006	0003			
P O	\$CC3CZ	194/01	61/04	003	4000	33	4000	003	8008	006	0003			
P O	\$CC3DF	205/14	66/B8	012	8000	2F	8006	016	8008	004	0012			
P O	\$CC3DL	241/23	78/DC	013	8000	00	8005	014	8008	006	0015			
P O	\$CC3DM	206/05	67/14	004	8000	D6	8006	021	8008	006	0005			
P O	\$CC3EI	230/16	73/40	003	4000	9D	4000	016	8008	003	0003			
P O	\$CC3EJ	233/02	74/88	003	4000	C5	4000	003	8008	007	0004			
P O	\$CC3EK	208/14	68/38	003	4000	AE	4000	016	8008	006	0003			
P O	\$CC3EL	252/06	7E/18	003	4000	CF	4000	016	8008	006	0004			
P O	\$CC3EM	230/21	73/54	003	4000	44	4000	016	8008	003	0003			
P O	\$CC3FS	206/13	67/34	021	8000	72	8006	016	8008	005	0024			
P O	\$CC3FX	218/05	6D/14	007	8000	44	8006	021	8008	006	0007			
P O	\$CC3IP	257/14	80/B8	009	8000	F1	8006	021	8008	006	0011			
P O	\$CC3LD	207/20	67/D0	009	8000	C3	8006	021	800D	006	0010			
P O	\$CC3LO	208/06	68/18	004	8000	53	8005	021	8008	003	0004			
P O	\$CC3LT	186/00	5D/00	003	4000	DD	4000	003	8008	006	0004			
P O	\$CC3PX	208/10	68/28	003	8000	D6	8006	021	8008	004	0004			
P O	\$CC3RO	194/04	61/10	003	9500	00	9500	003	8008	003	0004			
P O	\$CC3TA	206/10	67/28	003	8000	62	8006	021	8008	003	0003			
P O	\$CC3TB	243/21	79/D4	018	8000	5F	8006	016	8008	005	0020			
P O	\$CC3TC	213/13	6A/B4	010	8000	0D	8006	021	8008	008	0011			
P O	\$CC3UB	214/05	6B/14	006	8000	9F	8006	016	8008	006	0007			
P O	\$CC3US	214/00	6B/00	004	8000	40	8006	016	8008	004	0005			
P O	\$CC4JA	191/12	5F/B0	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4JB	195/23	61/DC	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4JC	196/12	62/30	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4JD	197/19	62/CC	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4JE	258/07	81/1C	002	4200	80	4200	016	8000	008	0002			
P O	\$CC4J1	197/22	62/D8	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J2	198/01	63/04	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J3	198/04	63/10	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J4	198/07	63/1C	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J5	198/10	63/28	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J6	198/13	63/34	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J7	198/16	63/40	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J8	198/19	63/4C	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4J9	198/22	63/58	002	4200	00	4200	016	8000	008	0003			
P O	\$CC4SU	202/22	65/58	001	4000	FE	407A	016	8000	006	0001			
P O	\$CC4YA	204/05	66/14	001	4000	1B	4000	016	8000	003	0001			

TYPE	NAME	DSK	ADD	CYL/SEC	TXT-CAT	LINK	ADD	RLD	DISP	ENTRY	PNT	CORE	SEC	ATTR	LEVEL	TOT	SEC
P	O	\$CC5SH	248/10	7C/28	011	8000		AE	8000	005	8000	006	0012				
P	O	\$CC6CL	234/19	75/4C	010	8000		5C	8800	032	8000	006	0011				
P	O	\$CC6CS	231/10	73/A8	008	8800		0C	8800	020	8000	004	0008				
P	O	\$CC6IM	231/18	73/C8	006	8800		97	8800	024	8000	006	0006				
P	O	\$CC6MI	232/00	74/00	006	8800		13	8800	008	8000	004	0006				
P	O	\$CC6OP	224/15	70/3C	005	8000		9F	8006	008	8000	006	0005				
P	O	\$CC6SC	232/06	74/18	008	8800		4D	8E8E	024	8000	007	0008				
P	O	\$CC6SS	232/14	74/38	010	8800		0A	8800	024	8000	003	0012				
P	O	\$CGDRV	207/13	67/B4	006	0000		B3	041C	064	881A	003	0007				
P	O	\$CGNBX	215/01	6B/84	004	49C7		F3	49CE	064	881A	003	0004				
P	O	\$CGNCM	215/05	6B/94	023	43F4		AD	43FB	064	881A	003	0023				
P	O	\$CGNPE	216/04	6C/10	012	43F4		72	49C7	064	881A	003	0012				
P	O	\$CGNPS	216/16	6C/40	020	49C7		C2	4ACD	064	881A	003	0020				
P	O	\$CGNSB	214/12	6B/30	009	43F4		E8	43FB	064	881A	004	0009				
P	O	\$CGNSF	217/12	6C/B0	004	43F4		42	43FB	064	881A	003	0004				
P	O	\$CGNSS	217/16	6C/C0	005	43F4		79	43FB	064	881A	004	0005				
P	O	\$CGNSX	217/21	6C/D4	008	49C7		3F	49CE	064	881A	003	0008				
P	O	\$MPXDV	250/10	7D/28	023	4000		00	4023	048	80AA	003	0024				
P	O	\$MPXP1	249/20	7C/D0	013	0000		B3	04B1	020	9000	003	0014				
P	O	\$MPXP2	191/23	5F/DC	010	0000		A7	0000	020	9000	003	0011				
P	O	\$MPXP3	248/22	7C/58	010	0000		EA	0000	020	9000	003	0011				
P	O	\$MPXP4	249/09	7C/A4	010	0000		E0	0000	020	9000	003	0011				
P	O	\$OLAF	222/21	6F/54	020	4400		42	5553	016	8000	004	0020				
P	O	\$OLAH	251/10	7D/A8	009	4400		00	4407	016	8000	004	0010				
P	O	\$OLAJ	253/05	7E/94	013	4400		58	4FC0	016	8000	004	0013				
P	O	\$OLAP	256/05	80/14	016	4400		58	4507	016	8000	007	0016				
P	O	\$OLAR	253/18	7E/C8	014	4500		DD	51AB	016	8000	004	0014				
P	O	\$OLAT	254/08	7F/20	014	4400		BB	4407	016	8000	004	0014				
P	O	\$OLBE	189/06	5E/98	010	4400		02	4407	016	8000	005	0010				
P	O	\$OLBO	256/21	80/54	017	4400		0C	4402	016	8000	005	0017				
P	O	\$OLER	202/11	65/2C	001	0000		F1	000A	016	8000	003	0002				
P	O	\$OLFTP	194/12	61/30	007	4800		00	4802	016	8000	005	0008				
P	O	\$OLINK	252/10	7E/28	019	4000		48	4F05	040	808A	004	0019				
P	O	\$OLIN1	255/07	7F/9C	008	4F00		5F	4F0A	016	8000	004	0008				
P	O	\$OLIN2	255/20	7F/D0	008	4F00		00	4F0A	016	8000	004	0009				
P	O	\$OLIN3	255/15	7F/BC	005	4F00		F0	4F0A	016	8000	004	0005				
P	O	\$OLMSG	251/20	7D/D0	009	4400		00	4407	016	8000	004	0010				
P	O	\$OLYNX	254/22	7F/58	009	4000		AE	4405	016	8008	004	0009				
P	O	\$SGFIX	227/23	71/DC	014	4000		00	4174	032	9088	003	0015				
P	O	\$SGLOG	233/06	74/98	006	4000		7C	4006	032	9088	003	0006				
P	O	\$SGPTF	228/14	72/38	017	4000		80	4174	032	9088	003	0017				
P	O	\$SGPTR	229/07	72/9C	018	4000		88	4006	040	908A	003	0018				
P	O	\$SGPVR	230/01	73/04	015	4000		CF	4006	032	908A	003	0015				
P	O	CCPIVP	218/12	6D/30	014	8000		F9	8000	016	8000	006	0016				
P	R	\$@DE12	263/21	83/D4	000						8000	001	0015				
P	R	\$@LPOE	264/19	84/4C	000						8000	001	0003				
P	R	\$@QD54	264/13	84/34	000						8000	001	0003				
P	R	\$@SPFN	264/16	84/40	000						8000	001	0003				
P	R	\$CC\$BS	260/01	82/04	000						8000	004	0001				

TYPE	NAME	DSK ADD	CYL/SEC	TXT-CAT	LINK	ADD RLD	DISP	ENTRY	PNT	CORE	SEC	ATTR	LEVEL	TOT	SEC
P R	\$CC\$ML	260/08	82/20	000								8000	002	0001	
P R	\$CC1BF	260/09	82/24	000								8000	003	0005	
P R	\$CC4BT	260/21	82/54	000								8000	004	0001	
P R	\$CC4CP	260/22	82/58	000								8000	003	0001	
P R	\$CC4DF	262/05	83/14	000								8000	006	0016	
P R	\$CC4DQ	260/02	82/08	000								8000	003	0002	
P R	\$CC4IB	260/23	82/5C	000								8000	006	0001	
P R	\$CC4IM	261/18	82/C8	000								8000	004	0001	
P R	\$CC4MS	260/04	82/10	000								8000	006	0004	
P R	\$CC4MV	261/03	82/8C	000								8000	003	0002	
P R	\$CC4MX	261/00	82/80	000								8000	003	0003	
P R	\$CC4M1	259/08	81/A0	000								8000	008	0017	
P R	\$CC4M2	262/21	83/54	000								8000	008	0021	
P R	\$CC4NQ	261/05	82/94	000								8000	003	0003	
P R	\$CC4OC	261/08	82/A0	000								8000	003	0001	
P R	\$CC4PI	262/00	83/00	000								8000	006	0005	
P R	\$CC4TI	261/09	82/A4	000								8000	006	0003	
P R	\$CC4V1	260/18	82/48	000								8000	003	0003	
P R	\$CC4V2	260/14	82/38	000								8000	003	0004	
P R	\$CGCNM	263/18	83/C8	000								8000	001	0003	
P R	\$SGNIP	265/05	84/94	000								8000	001	0007	
P R	CCPCIO	261/19	82/CC	000								8000	002	0002	
P R	CCPFIO	261/21	82/D4	000								8000	002	0002	
P R	SUBR90	261/12	82/B0	000								8000	003	0002	
P R	SUBR91	259/06	81/98	000								8000	006	0002	
P R	SUBR92	261/14	82/B8	000								8000	006	0004	
P R	SUBR93	261/23	82/DC	000								8000	002	0001	
P R	WRTNIP	264/22	84/58	000								8000	001	0007	

// COPY FROM-R2, LIBRARY-P, NAME-ALL, TO-PRINT

MODULE-\$CG1G1, VOLUME ID-R2R2R2, DATE-06/06/10

// LOG PRINTER,NOEJECT

01000

// NOHALT

01010

*

01020

*** CCP GENERATION CARDLESS PROCEDURE ***

01030

*

01040

// CALL \$CG1G3,R1 *** PROCESS USER SPECIFICATIONS ***

01060

// CALL \$CG1G4,R1 *** PRINT RESULTS OF SPECIFICATIONS ***

01070

// CALL \$CG1G5,R1 *** CREATE SOURCE AND PROCEDURE MEMBERS ***

01080

// CALL \$CCPSA,R1 *** COMPLETE THE CCP GENERATION ***

01090

MODULE-\$CG1G2, VOLUME ID-R2R2R2, DATE-06/06/10

*	02010
*** CCP CARDLESS GENERATION PROCEDURE ***	02020
*	02030
*** COPY MACRO PROCESSOR AND REQUIRED MODULES OF OVERLAY LINKAGE	02040
*** EDITOR FROM SYSTEM PACK TO DISTRIBUTION PACK	02050
*	02060
// LOAD \$MAINT,F1	++02070
// RUN	02080
// COPY FROM-F1,TO-R1,LIBRARY-O,RETAIN-R,NAME-\$MPX.ALL	##++02090
// COPY FROM-F1,TO-R1,LIBRARY-O,RETAIN-R,NAME-\$OLYNX	##++02100
// COPY FROM-F1,TO-R1,LIBRARY-O,RETAIN-R,NAME-\$OLBO	##++02110
// COPY FROM-F1,TO-R1,LIBRARY-O,RETAIN-R,NAME-\$OLFTP	##++02120
// COPY FROM-F1,TO-R1,LIBRARY-O,RETAIN-R,NAME-\$OLER	##++02130
// END	02140

MODULE-\$CG1G3, VOLUME ID-R2R2R2, DATE-06/06/10

*	03000
*** CCP CARDLESS GENERATION PROCEDURE ***	03010
*** EXPAND USERS CCP SPECIFICATIONS ***	03020
*	03030
// LOAD \$MPXDV,R1	## 03040
// FILE NAME-\$SOURCE,RETAIN-T,UNIT-R1,PACK-PID001,TRACKS-20,	##\$\$03050
// LOCATION-386	03060
// COMPILE SOURCE-\$CG1GM,UNIT-R1	## 03070
// RUN	03080
*	03090
*	03100
* NOTE:	03110
* _____	03120
*	03130
* \$E MACRO STATEMENTS ARE ENTERED VIA THE SOURCE LIBRARY MEMBER \$CG1GM	03140
* ON THE CCP DISTRIBUTION PACK OR FROM THE SYSTEM INPUT DEVICE IF THE	03150
* '// COMPILE' STATEMENT IS REMOVED FROM THIS PROCEDURE.	03160

MODULE-\$CG1G4, VOLUME ID-R2R2R2, DATE-06/06/10

```
*                                04010
*** CCP CARDLESS GENERATION PROCEDURE *** 04020
*                                04030
*** PRINT RESULTS OF CCP GENERATION SPECIFICATIONS 04040
*                                04050
// LOAD $CC1PP,R1                ## 04060
// FILE NAME-$SOURCE,UNIT-R1,PACK-PID001  ##$04070
// RUN                            04080
```

MODULE-\$CG1G5, VOLUME ID-R2R2R2, DATE-06/06/10

*	05010
*** CCP CARDLESS GENERATION PROCEDURE ***	05020
*	05030
*** IF NO ERRORS IN GENERATION SPECIFICATIONS, LOAD \$MAINT TO CREATE	05040
*** SOURCE AND PROCEDURE MEMBERS THAT WILL COMPLETE THE CCP GENERATION	05050
*	05060
// LOAD \$MAINT,F1	++05070
// FILE NAME-\$SOURCE,UNIT-R1,PACK-PID001,RETAIN-S	###\$05080
// RUN	05090
// COPY FROM-DISK,TO-R1,RECL-96,RETAIN-R,FILE-\$SOURCE	###\$05100
// END	05110

```
MODULE-SSGPTR, VOLUME ID-R2R2R2, DATE-06/06/10
// LOAD $SGPTR,R1
// NOHALT
// FILE NAME-$WORK,UNIT-F1,PACK-F1F1F1,TRACKS-6
// RUN
```


MODULE-\$\$SGPT2, VOLUME ID-R2R2R2, DATE-06/06/10

// LOAD \$MAINT,F1

// HALT

// FILE NAME-\$WORK,UNIT-F1,PACK-F1F1F1,RETAIN-S

// RUN

// COPY FROM-DISK,TO-F1,RETAIN-R,RECL-80,FILE-\$WORK

// END

```
MODULE-$SGPVR, VOLUME ID-R2R2R2, DATE-06/06/10
// LOAD $SGPVR,R1
// FILE NAME-$WORK,UNIT-F1,PACK-F1F1F1,RETAIN-S,TRACKS-6
// RUN

// COPY FROM-R2,LIBRARY-S,NAME-ALL,TO-PRINT
```

```
*****
*
*           C C P   G E N E R A T I O N   C O M P L E T E
*
*****
*
*   TO VERIFY THAT YOUR CCP SYSTEM HAS BEEN CORRECTLY
*   GENERATED, YOU SHOULD NOW DO THE FOLLOWING --
*
*   1. FAMILIARIZE YOURSELF WITH THE PROCEDURES FOR
*   PERFORMING INSTALLATION VERIFICATION. THESE
*   PROCEDURES ARE CONTAINED IN THE CCP SYSTEM
*   REFERENCE MANUAL, GC21-7620.
*
*   2. AS THE PROGRAM FOR PERFORMING INSTALLATION
*   VERIFICATION, USE THE SUPPLIED PROGRAM
*   'CCPIVP'.
*
*   3. PERFORM AN ASSIGNMENT BUILD RUN TO CREATE
*   AN ASSIGNMENT SET SUFFICIENT FOR THE VERIFI-
*   CATION ACTIVITY. USE THE ASSIGNMENT CONTROL
*   CARDS PUNCHED OUT IN THE PREVIOUS STEP FOR
*   THIS PURPOSE.
*
*   4. START UP CCP BY USING THE OCL DECK PUNCHED
*   OUT IN THE PREVIOUS STEP, THEN INTERACTING
*   THROUGH THE CONSOLE.
*
*   5. WHEN EXECUTION BEGINS, REQUEST THE PROGRAM
*   'CCPIVP' FROM THE CONSOLE.
*
*   6. COMPARE THE RESULTS OF THE OPERATION OF THIS
*   PROGRAM WITH THOSE SPECIFIED IN THE CCP
*   SYSTEM REFERENCE MANUAL.
*
*****
```

```

MODULE-$CGSET, VOLUME ID-R2R2R2, DATE-06/06/10
*****SAMPLE ASSIGNMENT AND SAMPLE START-UP DECK***** 00010000
* 00020000
*****SAMPLE ASSIGNMENT SET ***** 00030000
* 00040000
****FILL IN UNIT 00050000
* 00060000
// LOAD $CCPAS, 00070000
* 00080000
****FILL IN PACK AND UNIT 00090000
* 00100000
// FILE NAME-$CCPFILE,RETAIN-P,UNIT- ,PACK- 00110000
* 00120000
****FILL IN PACK AND UNIT 00130000
* 00140000
// FILE NAME-$CCPWORK,RETAIN-S,TRACKS-3,UNIT- ,PACK- 00150000
// RUN 00160000
* 00170000
*****THE FOLLOWING STATEMENTS CAN BE MODIFIED FOR YOUR 00180000
*****CONFIGURATION BUT SOME MUST BE KEPT TO RUN CCPIVP. SEE THE 00190000
*****COMMENTS IN THIS DECK. 00200000
* 00210000
// SET ID-@,ACTION-CREATE,DFLTEXEC-YES 00220000
// SYSTEM MINUPA-22K,MINTPBUF-2840, 00230000
// PASSWORD-FECD, 00240000
// COMMANDL-50,DFFPACK-PROGRAM,PGMREQ-15 00250000
* 00260000
// TERMATTR ATTRID-1,TRANSLAT-NO,BLKL-512,DATAFORM-MESSAGE, 00270000
// VERIFYID-NO,DFV3270-YES 00280000
* 00290000
*****THIS STMT TYPE REQD FOR CCPIVP(OR MLTALINE STMT) 00300000
* 00310000
// BSCALINE TYPE-CS,LINENUM-1,POLLIST-'00,01,10,11' 00320000
// BSCATERM TERMID-00,TYPE-3277M2,ATTRID-1,COMMAND-YES,OFFACTN-HOLD, 00330000
// ADDRCHAR-*60604040*,POLLCHAR-*40404040* 00340000
// BSCATERM TERMID-01,TYPE-3277M2,ATTRID-1,COMMAND-YES,OFFACTN-HOLD, 00350000
// ADDRCHAR-*6060C1C1*,POLLCHAR-*4040C1C1* 00360000
// BSCATERM TERMID-10,TYPE-3277M2,ATTRID-1,COMMAND-NO, 00370000
// ADDRCHAR-*61614040*,POLLCHAR-*C1C14040* 00380000
// BSCATERM TERMID-11,TYPE-3277M2,ATTRID-1,COMMAND-NO, 00390000
// ADDRCHAR-*6161C1C1*,POLLCHAR-*C1C1C1C1* 00400000
* 00410000
// TERMNAME NAME-CU0DV0,TERMID-00 00420000
// TERMNAME NAME-CU0DV1,TERMID-01 00430000
// TERMNAME NAME-CU1DV0,TERMID-10 00440000
// TERMNAME NAME-CU1DV1,TERMID-11 00450000
* 00460000
*****THIS STMT TYPE REQD FOR CCPIVP 00470000
* 00480000
// DISKFILE NAME-CGIVFIL1,ORG-C,RECL-16 00490000
* 00500000
*****THIS STMT TYPE REQD FOR CCPIVP 00510000
* 00520000
// DISKFILE NAME-CGIVFIL2,ORG-C,RECL-16 00530000
* 00540000
*****NOTE THAT ONE DISKFILE STATEMENT -CGIVFILE- WOULD BE NEEDED 00550000
*****IF SYMBOLIC FILES ARE NOT BEING USED. 00560000
* 00570000
* THE FOLLOWING TWO DISK FILE STATEMENTS ARE FOR EXAMPLE ONLY AND 00580000
* SHOULD BE REMOVED PRIOR TO RUNNING THIS ASSIGNMENT SET. 00590000

```

```

* 00600000
// DISKFILE NAME-DUMMY1,ORG-D,RECL-256 00610000
// DISKFILE NAME-DUMMY2,ORG-I,RECL-64,KEYL-8,KEYPOS-1,MSTRINDX-YES 00620000
* 00630000
*****THIS STMT TYPE REQD FOR CCPIVP IF SYMBOLIC FILES ARE USED. 00640000
* 00650000
// SYMFILE NAME-CGIVFILE,DISKFILE-'CGIVFIL1,CGIVFIL2' 00660000
* 00670000
*****THIS STMT NECESSARY FOR CCPIVP,PACK AND PRINTER VALUES 00680000
*****CAN BE CHANGED FOR YOUR CONFIG. 00690000
* 00700000
// PROGRAM NAME-CCPIVP,PGMDATA-YES, 00710000
// FILES-'CGIVFILE/CO/NOSHR', 00720000
// PACK-PROGRAM,PRINTER-NO 00730000
* 00740000
*****NOTE THAT CCPIVP MUST BE ON CORRECT PACK AT STARTUP OF CCP. 00750000
* 00760000
* THE FOLLOWING TWO PROGRAM STATEMENTS ARE FOR EXAMPLE ONLY AND 00770000
* SHOULD BE REMOVED PRIOR TO RUNNING THIS ASSIGNMENT SET. 00780000
* 00790000
// PROGRAM NAME-DUMMY1,MRTMAX-2,PGMDATA-YES, 00800000
// FILES-'DUMMY1/DU/SHR,DUMMY2/IRUA/SHR',PACK-SYSTEM,DFFMTERM-4, 00810000
// DFFNDF-2,DFFSFDT-1006 00820000
* 00830000
// PROGRAM NAME-DUMMY2,MRTMAX-2,PGMDATA-YES, 00840000
// FILES-'DUMMY1/DU/SHR,DUMMY2/IRA/SHR',PACK-SYSTEM,DFFMTERM-2, 00850000
// DFFNDF-1,DFFSFDT-396 00860000
* 00870000
* 00880000
*/ * REPLACE WITH /* 00890000
* 00900000
*****END OF SAMPLE ASSIGNMENT DECK***** 00910000
* 00920000
*****SAMPLE START-UP OCL FOR CCPIVP***** 00930000
* 00940000
* FILL IN UNIT 00950000
* 00960000
// LOAD $CCP, 00970000
* 00980000
***FOLLOWING TWO // FILE STATEMENTS CORRESPOND TO SAMPLE ASSIGNMENT 00990000
***DECK SYMBOLIC FILES. 01000000
* 01010000
// FILE NAME-CGIVFIL1,RETAIN-T,TRACKS-1,UNIT- ,PACK- ***ANY 5444 01020000
// FILE NAME-CGIVFIL2,RETAIN-T,TRACKS-1,UNIT- ,PACK- ***ANY 5444 01030000
* 01040000
***IF SYMBOLIC FILES ARE NOT USED REPLACE THE PRECEEDING TWO STATEMENTS 01050000
***SINGLE // FILE STATEMENT***** 01060000
* 01070000
// FILE NAME-CGIVFILE,RETAIN-T,TRACKS-1,UNIT- ,PACK- ***ANY 5444 01080000
* 01090000
// RUN 01100000
*****END OF SAMPLE START-UP OCL***** 01110000

```

MODULE-\$CGSMP, VOLUME ID-R2R2R2, DATE-06/06/10

```
*****
*           I N P U T   T O   C C P   G E N E R A T I O N           *
*****
*
*
* THESE CARDS -- AFTER YOU HAVE MODIFIED THEM TO YOUR SPECIFICATIONS -- *
* WILL BE THE INPUT TO THE FIRST PASS OF CCP GENERATION              *
*
*
* DISCARD THE LEADING CARDS OF THIS DECK, FROM THE // COPY CARD AT THE *
* BEGINNING THROUGH THE CARD THAT MARKS *** END OF INSTRUCTIONS ***  *
*
* DISCARD ALSO THE // CEND CARD -- THE LAST CARD OF THE DECK        *
*
*
* YOU MUST MAKE THREE KINDS OF MODIFICATIONS TO THE REMAINING CARDS -- *
*
* 1. IN THOSE CARDS THAT ARE MARKED ++ IN COLUMNS 74-75, REPLACE    *
* ANY $$ OR ## IN THE CARD WITH THE IDENTIFICATION OF A DISK        *
* UNIT --                                                            *
*
* -- $$ WITH THE UNIT ON WHICH YOUR SYSTEM PACK IS LOCATED          *
*
* -- ## WITH THE UNIT ON WHICH THE CCP DISTRIBUTION PACK IS          *
* LOCATED                                                            *
*
* 2. REPLACE CARD NUMBER 00299 WITH A CARD PUNCHED /* IN COLUMNS 1-2 *
*
* 3. MODIFY THE GENERATION CONTROL STATEMENTS -- THOSE CARDS        *
* FOLLOWING CARD NUMBER 00206 -- TO SPECIFY THE REQUIREMENTS OF     *
* THE CCP YOU WISH TO GENERATE                                       *
*
* IF YOU DO NOT WISH TO USE THE DISTRIBUTION PACK FOR THE REQUIRED    *
* $SOURCE FILE, SPECIFY THE LOCATION OF THAT FILE BY CHANGING THE UNIT *
* AND PACK PARAMETERS IN CARDS 00204 AND 00306                      *
*
*
* WHEN YOU HAVE MADE THESE CHANGES, PLACE THE MODIFIED DECK IN THE  *
* HOPPER OF THE SYSTEM INPUT DEVICE AND BEGIN PASS 1 OF CCP GENERATION *
*
* ***** END OF INSTRUCTIONS ***** END OF INSTRUCTIONS *****
```

```
*
*
*** PROCESS SPECIFICATIONS FOR THE CCP TO BE GENERATED
*
// LOAD $MPXDV,##
// FILE NAME-$SOURCE,RETAIN-T,UNIT-##,PACK-PID001,TRACKS-20,
// LOCATION-386
// RUN
$EFAC ESCAPE-NO, -- 'CCCCC' / X'XXXXXXXXXXXX' -- X FAC00
      PGMCNT-NO, -- YES -- X FAC01
      PRUF-NO, -- YES -- X FAC02
      FORMAT-NO, -- YES -- X FAC03
      ACCEPT-NO, -- YES -- X FAC04
      BSYPRN-NO, -- YES -- X FAC05
      CPUMSG-NO -- YES -- FAC06
$EPLG LANG- , -- COBOL / RPGII / FORTRAN / ASSEM -- X PLG00
      PPUNIT- -- R1 / F1 / R2 / F2 -- PLG01
$ESEC SECURE-NO, -- CCP / USER -- X SEC00
      LUSI-0 -- 1 - 4096 IF SECURE-USER -- SEC01
```

\$EFIL	SETS-1,	-- 2 - 25 --	X	FIL00
	PROGS-10,	-- 1 - 255 --	X	FIL01
	DFILES-5,	-- 1 - 50 --	X	FIL02
	TERMS-1,	-- 2 - 254 --	X	FIL03
	DUMPS-1,	-- 2 - 9 --	X	FIL04
	CORE-48K,	-- 64K/96K/128K/160K/192K/224K/256K	X	FIL05
	FLUNIT-	-- R1 / F1 / R2 / F2 --	X	FIL06
	TRKLOC-	-- VALID TRACK NUMBER / OMITTED --	X	FIL07
	FLPACK-	-- NAME OF PACK --		FIL08
\$EMLA	LINES-	-- 0 - 8 --	X	MLA00
	XLATE-YES	-- NO --		MLA01
\$EMLD	TYPE-	-- SEE SYSTEM REFERENCE MANUAL --	X	MLD00
	XMCODE-	-- SEE SYSTEM REFERENCE MANUAL --		MLD01
\$EBSC	BSCA-	-- 0 - 2 --	X	BSC00
	DIAL-NO,	-- YES --	X	BSC01
	PP-NO,	-- YES --	X	BSC02
	MP-NO,	-- YES --	X	BSC03
	CS-NO,	-- YES --	X	BSC04
	GETMSG-NO,	-- YES --	X	BSC05
	ITB-NO,	-- YES --	X	BSC06
	RECSEP-1E,	-- TWO HEX DIGITS --	X	BSC07
	ASCII-NO,	-- YES --	X	BSC08
	EBCDIC-YES,	-- NO --	X	BSC09
	XPRNCY-NO,	-- YES --	X	BSC10
	RESPOL-NO,	-- YES --	X	BSC11
	AUTORS-NO,	-- YES --	X	BSC12
	INTPOL-NO,	-- YES --	X	BSC13
	DA-NO	-- YES --		BSC14
\$EBSD	TYPE-	-- SEE SYSTEM REFERENCE MANUAL --		BSD00
\$EGEN	DSUNIT-\$\$,	-- R1 / F1 --	X	++GEN00
	CCUNIT-	-- R1 / F1 / R2 / F2 --	X	GEN01
	WKUNIT-	-- UNIT / 'UNIT,UNIT,UNIT' --	X	GEN02
	WKPACK-	-- PACK / 'PACK,PACK,PACK' --	X	GEN03
	DIUNIT-##,	-- R1 / F1 / R2 / F2 --	X	++GEN04
	MINRES-NO	-- YES --		GEN05
/	-- REPLACE THIS CARD WITH /* IN COLUMNS 1-2 --			00299
*				00300
***	PRINT RESULTS OF CCP GENERATION PASS 1			00301
*				00302
***	IF NO ERRORS, PUNCH INPUT TO CCP GENERATION PASS 2			00303
*				00304
//	LOAD \$CC1PP,##			++00305
//	FILE NAME-\$SOURCE,RETAIN-S,UNIT-##,PACK-PID001			++00306
//	RUN			00307

MODULE-\$EBSC , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $EBSC 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 0, MODIFICATION LEVEL 0 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION FIRST PASS MACRO-INSTRUCTION -- DEFINE BSCA 00100000
.* SUPPORT. 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . BSCA-0/1/2 00150000
.* 00160000
.* NUMBER OF BSC ADAPTERS. OPERAND REQUIRED IF THIS STATEMENT 00170000
.* IS USED. 00180000
.* 00190000
.* . DIAL-YES/NO 00200000
.* 00210000
.* SPECIFIES WHETHER SWITCHED-LINE SUPPORT IS TO BE INCLUDED. 00220000
.* DEFAULT IS NO. 00230000
.* 00240000
.* . PP-YES/NO 00250000
.* 00260000
.* SPECIFIES WHETHER POINT-TO-POINT SUPPORT IS TO BE INCLUDED. 00270000
.* DEFAULT IS NO. 00280000
.* 00290000
.* . MP-YES/NO 00300000
.* 00310000
.* SPECIFIES WHETHER MULTIPOINT TRIBUTARY SUPPORT IS TO BE 00320000
.* INCLUDED. DEFAULT IS NO. 00330000
.* 00340000
.* . CS-YES/NO 00350000
.* 00360000
.* SPECIFIES WHETHER MULTIPOINT CONTROL STATION SUPPORT IS TO BE 00370000
.* INCLUDED. DEFAULT IS NO. 00380000
.* 00390000
.* . GETMSG-YES/NO 00400000
.* 00410000
.* SPECIFIES WHETHER 'GATHER READ' SUPPORT IS TO BE INCLUDED. 00420000
.* DEFAULT IS NO. 00430000
.* 00440000
.* . ITB-YES/NO 00450000
.* 00460000
.* SPECIFIES WHETHER INTERMEDIATE-TEXT-BLOCK SUPPORT IS TO BE 00470000
.* INCLUDED. DEFAULT IS NO. 00480000
.* 00490000
.* . RECSEP-XX 00500000
.* 00510000
.* SPECIFIES ALTERNATE RECORD-SEPARATOR BYTE. OPERAND IS 00520000
.* OPTIONAL. DEFAULT IS "1E". 00530000
.* 00540000
.* . ASCII-YES/NO 00550000
.* 00560000
.* SPECIFIES WHETHER SUPPORT FOR ASCII TRANSMISSION CODE IS TO 00570000
.* BE INCLUDED. DEFAULT IS NO. 00580000
.* 00590000
```



```

.* . EBCDIC-YES/NO * 00600000
.* * 00610000
.* SPECIFIES WHETHER SUPPORT FOR EBCDIC TRANSMISSION CODE IS TO * 00620000
.* BE INCLUDED. DEFAULT IS *YES*. * 00630000
.* * 00640000
.* . RESPOL-YES/NO * 00650000
.* * 00660000
.* SPECIFIES WHETHER RESIDENT POLLING IS TO BE INCLUDED FOR * 00670000
.* CONTROL-STATION LINE SUPPORT. DEFAULT IS NO. * 00680000
.* * 00690000
.* . AUTORS-YES/NO * 00700000
.* * 00710000
.* SPECIFIES WHETHER AUTO-RESPONSE SUPPORT IS TO BE INCLUDED * 00720000
.* FOR MULTIPOINT TRIBUTARY LINES. DEFAULT IS NO. * 00730000
.* * 00740000
.* . XPRNCY-YES/NO * 00750000
.* * 00760000
.* SPECIFIES WHETHER TEXT TRANSPARENCY FEATURE IS TO BE SUPPORTED * 00770000
.* -- DEFAULT IS NO. * 00780000
.* * 00790000
.* . INTPOL-YES/NO * 00800000
.* * 00810000
.* SPECIFIES WHETHER INTERVAL POLLING FEATURE IS TO BE SUPPORTED * 00820000
.* -- DEFAULT IS NO. * 00830000
.* * 00840000
.* . DA-YES/NO * 00850000
.* * 00860000
.* SPECIFIES WHETHER IDA FEATURE IS TO BE SUPPORTED 00870000
.* -- DEFAULT IS NO. 00880000
.* * 00890000
.* ***** 00900000
.* $EBC &BSCA-, &DIAL-NO, &PP-NO, &MP-NO, &CS-NO, &GETMSG-NO, &ITB-NO, X00910000
.* &RECSEP-, &ASCII-NO, &EBCDIC-YES, &RESPOL-NO, &AUTORS-NO, X00920000
.* &XPRNCY-NO, &INTPOL-NO, &DA-NO 00930000
.* * 00940000
.* GBLA &SEQ . SEQUENCE CONTROL: 00950000
.* * . MUST BE 5, 6, 7 UPON ENTRY 00960000
.* * . IF 6 THEN &MLA MUST BE 0 00970000
.* * . IS MADE 8 AFTER PROCESSING 00980000
.* GBLB &TERR . TERMINATION ERROR SWITCH 00990000
.* * 01000000
.* GBLB &BIDA . IDA SUPPORT 01010000
.* GBLA &BSC . NUMBER BSC ADAPTERS 01020000
.* GBLC &BLT . BSC LINE TYPES SUPPORTED 01030000
.* GBLC &BFA . BSCA FEATURES SUPPORTED 01040000
.* GBLC &RSB . HEX RECORD SEPARATOR 01050000
.* GBLA &MLA . NUMBER MLTA LINES 01060000
.* GBLB &INPOL . INTERVAL POLLING SUPPORT. 01070000
.* * 01080000
.* LCLA &N . NUMBER OF CHARACTERS 01090000
.* LCLC &C1, &C2 . FOR MANIPULATING CHAR VALUES 01100000
.* * 01110000
.* TABLE &DIAL . SWITCHED LINE SUPPORT 01120000
YES TABDF 1 01130000
Y TABDF 1 01140000
NO TABDF 0 01150000
N TABDF 0 01160000
.* TABDF * . INVALID PARAMETER 01170000
.* * 01180000
.* TABLE &PP . POINT-TO-POINT SUPPORT 01190000

```

YES	TABDF 1		01200000
Y	TABDF 1		01210000
NO	TABDF 0		01220000
N	TABDF 0		01230000
.	TABDF *	. INVALID PARAMETER	01240000
.	*		01250000
	TABLE &MP	. MULTIPOINT TRIBUTARY SUPPORT	01260000
YES	TABDF 1		01270000
Y	TABDF 1		01280000
NO	TABDF 0		01290000
N	TABDF 0		01300000
.	TABDF *	. INVALID PARAMETER	01310000
.	*		01320000
	TABLE &CS	. CONTROL STATION SUPPORT	01330000
YES	TABDF 1		01340000
Y	TABDF 1		01350000
NO	TABDF 0		01360000
N	TABDF 0		01370000
.	TABDF *	. INVALID PARAMETER	01380000
.	*		01390000
	TABLE &GETMSG	. GATHER-READ SUPPORT	01400000
YES	TABDF 1		01410000
Y	TABDF 1		01420000
NO	TABDF 0		01430000
N	TABDF 0		01440000
.	TABDF *	. INVALID PARAMETER	01450000
.	*		01460000
	TABLE &ITB	. INTERMEDIATE TEXT BLOCK SUPPRT	01470000
YES	TABDF 1		01480000
Y	TABDF 1		01490000
NO	TABDF 0		01500000
N	TABDF 0		01510000
.	TABDF *	. INVALID PARAMETER	01520000
.	*		01530000
	TABLE &ASCII	. ASCII TRANSMISSION CODE	01540000
YES	TABDF 1		01550000
Y	TABDF 1		01560000
NO	TABDF 0		01570000
N	TABDF 0		01580000
.	TABDF *	. INVALID PARAMETER	01590000
.	*		01600000
	TABLE &EBCDIC	. EBCDIC TRANSMISSION CODE	01610000
YES	TABDF 1		01620000
Y	TABDF 1		01630000
NO	TABDF 0		01640000
N	TABDF 0		01650000
.	TABDF *	. INVALID PARAMETER	01660000
.	*		01670000
	TABLE &RESPOL	. RESIDENT POLLING	01680000
YES	TABDF 1		01690000
Y	TABDF 1		01700000
NO	TABDF 0		01710000
N	TABDF 0		01720000
.	TABDF *	. INVALID PARAMETER	01730000
.	*		01740000
	TABLE &AUTORS	. AUTO RESPONSE	01750000
YES	TABDF 1		01760000
Y	TABDF 1		01770000
NO	TABDF 0		01780000
N	TABDF 0		01790000

	TABDF *	. INVALID PARAMETER	01800000
.*			01810000
	TABLE &XPRNCY	. TRANSPARENCY FEATURE	01820000
YES	TABDF 1		01830000
Y	TABDF 1		01840000
NO	TABDF 0		01850000
N	TABDF 0		01860000
	TABDF *	. INVALID PARAMETER	01870000
.*			01880000
	TABLE &INTPOL	. INTERVAL POLLING FEATURE	01890000
NO	TABDF 0		01900000
N	TABDF 0		01910000
YES	TABDF 1		01920000
Y	TABDF 1		01930000
	TABDF *	. INVALID PARAMETER	01940000
.*			01950000
	TABLE &DA	. DA FEATURE	01960000
YES	TABDF 1		01970000
Y	TABDF 1		01980000
NO	TABDF 0		01990000
N	TABDF 0		02000000
	TABDF *	. INVALID PARAMETER	02010000
.*			02020000
	TEXT		02030000
.*			02040000
.*	----- CHECK STATEMENT SEQUENCE -----*		02050000
.*			02060000
	AIF (&SEQ EQ '5').SEQOK	. OK IF \$EFIL PREVIOUS	02070000
	AIF (&SEQ EQ '7').SEQOK	. OK IF \$EMLD PREVIOUS	02080000
	AIF (&SEQ NE '6').SEQER	. ERROR IF NOT \$EMLA PREVIOUS	02090000
	AIF (&MLA EQ '0').SEQOK	. OK IF \$EMLA AND LINES-0	02100000
.*			02110000
.SEQER ANOP			02120000
*!400E \$EBSC	STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR		02130000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH		02140000
&BLT SETC ''	. MAKE NULL PREVIOUS SPEC		02150000
&BFA SETC ''	. MAKE NULL PREVIOUS SPEC		02160000
.*			02170000
.SEQOK ANOP	. VALID SEQUENCE		02180000
.*			02190000
.*	----- BSCA-0/1/2 -----*		02200000
.*			02210000
	AIF (T'&BSCA NE '0').BSCA	. SKIP IF BSCA SPECIFIED	02220000
.*			02230000
*!405E MISSING 'BSCA' OPERAND -- MUST BE SPECIFIED IF STATEMENT USED			02240000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH		02250000
AGO .BSEND			02260000
.*			02270000
.BSCA ANOP	. BSCA SPECIFIED		02280000
AIF (T'&BSCA NE 'N').BSERR	. ERROR IF NOT NUMERIC		02290000
AIF (K'&BSCA GT '4').BSERR	. SKIP IF LONGER THAN 4 DIGITS		02300000
&N SETA &BSCA			02310000
AIF (&N EQ '0').BS0	. SKIP IF BSCA-0		02320000
AIF (&N LE '2').BSOK	. SKIP IF BSCA-1 OR BSCA-2		02330000
.*			02340000
.BSERR ANOP			02350000
*!410E INVALID 'BSCA' PARAMETER -- MUST BE NUMBER IN RANGE 0-2			02360000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH		02370000
AGO .BSEND			02380000
.*			02390000

.BS0	ANOP	. BSCA-0	02400000
	AIF	('&DIAL&PP&MP&CS&GETMSG&ITB' NE '000000').BS0ER	02410000
	AIF	('&ASCII&EBCDIC&RESPOL&AUTORS' NE '0100').BS0ER	02420000
	AIF	(&RECSEP EQ 'NO').END	02430000
	AIF	(&RECSEP EQ 'N').END	02440000
.BS0ER	ANOP		02450000
*!415E	BSCA-0,	BUT OTHER OPERAND SPECIFIED WITH NON-DEFAULT PARAMETER	02460000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	02470000
.	*		02480000
.BSOK	ANOP	. BSCA-1 OR BSCA-2	02490000
&BSC	SETA &BSCA	. NUMBER BSC ADAPTERS	02500000
.	*		02510000
.BSEND	ANOP	. END OPERAND 'BSCA'	02520000
.	*		02530000
.	*	----- DA-YES/Y/NO/N (YES=1, NO=0) -----*	02540000
.	*		02550000
	AIF	(&DA NE '*').DAOK	02560000
*!417E	INVALID 'DA'	PARAMETER -- MUST BE YES/Y/NO/N	02570000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	02580000
	AGO	.DAEND	02590000
.	*		02600000
.DAOK	ANOP		02610000
	AIF	(&DA NE '1').DAEND	02620000
	AIF	(&BSC GT '0').BIDK	02630000
*!419E	IF DA-YES,	MUST SPECIFY BSCA-1 OR BSCA-2	02640000
&TERR	SETB 1		02650000
	AGO	.DAEND	02660000
.	*		02670000
.BIDK	ANOP	. SET 'IDA' SUPPORT	02680000
&BIDA	SETB 1		02690000
.	*		02700000
.DAEND	ANOP	. END OPERAND 'DA'	02710000
.	*		02720000
.	*	----- PP-YES/Y/NO/N (YES=1, NO=0) -----*	02730000
.	*		02740000
	AIF	(&PP NE '*').PPOK	02750000
.	*	. SKIP IF VALID PARAMETER	02760000
*!425E	INVALID 'PP'	PARAMETER -- MUST BE YES/Y/NO/N	02770000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	02780000
	AGO	.PPEND	02790000
.	*		02800000
.PPOK	ANOP	. VALID PARAMETER	02810000
&BLT	SETC '&PP'	. SET X	02820000
.	*		02830000
.PPEND	ANOP	. END OPERAND 'PP'	02840000
.	*		02850000
.	*	----- MP-YES/Y/NO/N (YES=1, NO=0) -----*	02860000
.	*		02870000
	AIF	(&MP NE '*').MPOK	02880000
.	*	. SKIP IF VALID PARAMETER	02890000
*!430E	INVALID 'MP'	PARAMETER -- MUST BE YES/Y/NO/N	02900000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	02910000
	AGO	.MPEND	02920000
.	*		02930000
.MPOK	ANOP	. VALID PARAMETER	02940000
&BLT	SETC '&BLT&MP'	. SET .X	02950000
.	*		02960000
.MPEND	ANOP	. END OPERAND 'MP'	02970000
.	*		02980000
.	*	----- CS-YES/Y/NO/N (YES=1, NO=0) -----*	02990000

```

.*
AIF (&CS NE '*').CSOK . SKIP IF VALID PARAMETER 03000000
.* 03010000
.* 03020000
*!435E INVALID 'CS' PARAMETER -- MUST BE YES/Y/NO/N 03030000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03040000
AGO .CSEND 03050000
.* 03060000
.CSOK ANOP . VALID PARAMETER 03070000
&BLT SETC '&BLT&CS' . SET ..X 03080000
.* 03090000
.CSEND ANOP . END OPERAND 'CS' 03100000
.* 03110000
.*----- DIAL-YES/Y/NO/N (YES=1, NO=0) -----* 03120000
.* 03130000
AIF (&DIAL NE '*').DIOK . SKIP IF VALID PARAMETER 03140000
.* 03150000
*!420E INVALID 'DIAL' PARAMETER -- MUST BE YES/Y/NO/N 03160000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03170000
AGO .DIEND 03180000
.* 03190000
.DIOK ANOP 03200000
&BLT SETC '&BLT&DIAL' . SET ...X 03210000
.* 03220000
.DIEND ANOP . END OPERAND 'DIAL' 03230000
.* 03240000
.*----- ASSURE AT LEAST ONE LINE TYPE SPECIFIED -----* 03250000
.* 03260000
AIF (&BLT NE '0000').LTEND . SKIP IF AT LEAST 1 TYPE 03270000
.* 03280000
*!440E BSCA PRESENT BUT NO LINE TYPES SPECIFIED 03290000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03300000
.* 03310000
.LTEND ANOP . END CHECK OF LINE TYPES 03320000
.* 03330000
.*----- GETMSG-YES/Y/NO/N (YES=1, NO=0) -----* 03340000
.* 03350000
AIF (&GETMSG NE '*').GMOK . SKIP IF VALID PARAMETER 03360000
.* 03370000
*!445E INVALID 'GETMSG' PARAMETER -- MUST BE YES/Y/NO/N 03380000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03390000
AGO .GMEND 03400000
.* 03410000
.GMOK ANOP . VALID PARAMETER 03420000
&BFA SETC '&GETMSG' . SET X 03430000
.* 03440000
.GMEND ANOP . END OPERAND 'GETMSG' 03450000
.* 03460000
.*----- ITB-YES/Y/NO/N (YES=1, NO=0) -----* 03470000
.* 03480000
AIF (&ITB NE '*').ITOK . SKIP IF VALID PARAMETER 03490000
.* 03500000
*!450E INVALID 'ITB' PARAMETER -- MUST BE YES/Y/NO/N 03510000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03520000
.* 03530000
.ITOK ANOP . VALID PARAMETER 03540000
&BFA SETC '&BFA&ITB' . SET .X 03550000
.* 03560000
.ITEND ANOP . END OPERAND 'ITB' 03570000
.* 03580000
.*----- RECSEP-XX -----* 03590000

```

```

.*
AIF (T'&RECSEP EQ 'O').RSNO . SKIP IF RECSEP OMITTED 03600000
&N SETA K'&RECSEP . LENGTH OF PARAMETER 03610000
AIF (&N NE '2').RSERR . ERROR IF NOT 2 CHARS 03620000
.*
AIF ('&RECSEP'(1,1) GT '9').RSERR . ERR IF GT 9 03630000
AIF ('&RECSEP'(1,1) LT 'A').RSERR . ERR IF LT 'A' 03640000
AIF ('&RECSEP'(1,1) LE 'F').RS2 . OK IF IN RANGE A-F 03650000
AIF ('&RECSEP'(1,1) LT '0').RSERR . ERR IF NOT IN RANGE 0-9 03660000
.RS2 ANOP 03670000
AIF ('&RECSEP'(2,1) GT '9').RSERR . ERR IF GT 9 03680000
AIF ('&RECSEP'(2,1) LT 'A').RSERR . ERR IF LT 'A' 03690000
AIF ('&RECSEP'(2,1) LE 'F').RSYES . OK IF IN RANGE A-F 03700000
AIF ('&RECSEP'(2,1) GE '0').RSYES . OK IF IN RANGE 0-9 03710000
.*
.RSERR ANOP . INVALID PARAMETER 03720000
*!455E INVALID 'RECSEP' PARAMETER -- MUST BE 2 HEX DIGITS 03730000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03740000
AGO .RSEND 03750000
.*
.RSNO ANOP . RECSEP-NO 03760000
&BFA SETC '&BFA.0' . SET ..X 03770000
AGO .RSEND 03780000
.*
.RSYES ANOP . RECSEP-XX 03790000
&BFA SETC '&BFA.1' 03800000
&RSB SETC '1E' . SET DEFAULT RECORD-SEPARATOR 03810000
&RSB SETC '&RECSEP' . RECSEP CHARACTER 03820000
.*
.RSEND ANOP . END OPERAND 'RECSEP' 03830000
.*
*----- INTPOL-YES/Y/NO/N (YES=1, NO=0) -----* 03840000
.*
AIF (&INTPOL NE '*').IPOK . SKIP IF VALID PARAMETER 03850000
.*
*!481E INVALID 'INTPOL' PARAMETER -- MUST BE YES/Y/NO/N 03860000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03870000
AGO .IPEND 03880000
.*
.IPOK ANOP . VALID PARAMETER 03890000
AIF (&INTPOL EQ '0').IPEND . SKIP IF INTPOL-NO 03900000
&INPOL SETB 1 . SET INTERVAL POLLINE 03910000
.*
.IPEND ANOP . END OPERAND 'INTPOL' 03920000
.*
*----- RESPOL-YES/Y/NO/N (YES=1, NO=0) -----* 03930000
.*
AIF (&RESPOL NE '*').RPOK . SKIP IF VALID PARAMETER 03940000
.*
*!480E INVALID 'RESPOL' PARAMETER -- MUST BE YES/Y/NO/N 03950000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03960000
AGO .RPEND 03970000
.*
.RPOK ANOP . VALID PARAMETER 03980000
&BFA SETC '&BFA&RESPOL' . SET ...X 03990000
AIF (&RESPOL NE '1').RPEND . SKIP UNLESS RESPOL-YES 04000000
AIF (&CS NE '0').RPEND . SKIP UNLESS RESPOL-YES/CS-NO 04010000
.*
*!482W RESPOL-YES SPECIFIED WITH CS-NO -- TREATED AS RESPOL-NO 04020000
.*
04030000
04040000
04050000
04060000
04070000
04080000
04090000
04100000
04110000
04120000
04130000
04140000
04150000
04160000
04170000
04180000
04190000

```

```

.RPEND ANOP          . END OPERAND 'RESPOL'          04200000
.*                  04210000
.*----- AUTORS-YES/Y/NO/N (YES=1, NO=0) -----* 04220000
.*                  04230000
      AIF  (&AUTORS NE '*').AUOK          . SKIP IF VALID PARAMETER 04240000
.*                  04250000
*!485E INVALID 'AUTORS' PARAMETER -- MUST BE YES/Y/NO/N 04260000
&TERR SETB 1          . SET TERMINATION ERROR SWITCH 04270000
      AGO  .AUEND          04280000
.*                  04290000
.AUOK ANOP          . VALID PARAMETER          04300000
&BFA SETC '&BFA&AUTORS' . SET ....X          04310000
      AIF  (&AUTORS NE '1').AUEND . SKIP UNLESS AUTORS-YES 04320000
      AIF  (&MP NE '0').AUEND . SKIP UNLESS AUTORS-YES/MP-NO 04330000
.*                  04340000
*!487W AUTORS-YES SPECIFIED WITH CS-NO -- TREATED AS AUTORS-NO 04350000
.*                  04360000
.AUEND ANOP          . END OPERAND 'AUTORS'          04370000
.*                  04380000
.*----- EBCDIC-YES/Y/NO/N (YES=NULL=1, NO=0) -----* 04390000
.*                  04400000
      AIF  (&EBCDIC NE '*').EBOK          . SKIP IF VALID PARAMETER 04410000
.*                  04420000
*!465E INVALID 'EBCDIC' PARAMETER -- MUST BE YES/Y/NO/N 04430000
&TERR SETB 1          . SET TERMINATION ERROR SWITCH 04440000
      AGO  .EBEND          04450000
.*                  04460000
.EBOK ANOP          . VALID PARAMETER          04470000
&BFA SETC '&BFA&EBCDIC' . SET .....X          04480000
.*                  04490000
.EBEND ANOP          . END OPERAND 'EBCDIC'          04500000
.*                  04510000
.*----- ASCII-YES/Y/NO/N (YES=1, NO=0) -----* 04520000
.*                  04530000
      AIF  (&ASCII NE '*').ASOK          . SKIP IF VALID PARAMETER 04540000
.*                  04550000
*!460E INVALID 'ASCII' PARAMETER -- MUST BE YES/Y/NO/N 04560000
&TERR SETB 1          . SET TERMINATION ERROR SWITCH 04570000
      AGO  .ASEND          04580000
.*                  04590000
.ASOK ANOP          . VALID PARAMETER          04600000
&BFA SETC '&BFA&ASCII' . SET .....X          04610000
.*                  04620000
      AIF  ('&BFA'(6,2) NE '00').ASCK2          04630000
.*                  04640000
*!470E BSCA PRESENT BUT NEITHER TRANSMISSION CODE IS USED 04650000
&TERR SETB 1          . SET TERMINATION ERROR SWITCH 04660000
      AGO  .ASEND          04670000
.*                  04680000
.ASCK2 ANOP          04690000
      AIF  ('&BFA'(6,2) NE '11').ASEND . SKIP UNLESS ASCII/EBCDIC-Y 04700000
      AIF  (&BSCA NE '1').ASEND . SKIP UNLESS BSCA-1          04710000
.*                  04720000
*!472E EBCDIC-YES AND ASCII-YES SPECIFIED WITH BSCA-1 -- NOT POSSIBLE 04730000
&TERR SETB 1          . SET TERMINATION ERROR SWITCH 04740000
.*                  04750000
.ASEND ANOP          . END OPERAND 'ASCII'          04760000
.*                  04770000
.*----- XPRNCY-YES/Y/NO/N (YES=1, NO=0) -----* 04780000
.*                  04790000

```

```

AIF (&XPRNCY NE '*').XPOK . SKIP IF VALID PARAMETER 04800000
.* 04810000
*!490E INVALID 'XPRNCY' PARAMETER -- MUST BE YES/Y/NO/N 04820000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 04830000
AGO .XPEND 04840000
.* 04850000
.XPOK ANOP . VALID PARAMETER 04860000
&BFA SETC '&BFA&XPRNCY' . SET .....X 04870000
.* 04880000
AIF (&XPRNCY EQ '0').XPEND . SKIP IF XPRNCY-NO 04890000
.* 04900000
AIF (&EBCDIC NE '0').XPEND . SKIP UNLESS EBCDIC-NO 04910000
.* 04920000
*!492E XPRNCY-YES SPECIFIED WITH EBCDIC-NO -- NOT VALID 04930000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 04940000
.* 04950000
.XPEND ANOP . END OPERAND 'XPRNCY' 04960000
.* 04970000
*----- SET STATEMENT SEQUENCE -----* 04980000
.* 04990000
.END ANOP 05000000
&SEQ SETA 8 . INDICATE $EBSC PROCESSED 05010000
.* 05020000
MEND 05030000

```



```

MACRO                                00010000
*****                                00020000
.*                                     * 00030000
.* NAME: $EBSD                         * 00040000
.* RELEASE 4                            * 00050000
.*                                     * 00060000
.* MODIFICATION LEVEL: VERSION 1, MODIFICATION LEVEL 0 OF 5702-SC1 * 00070000
.*                                     * 00080000
.* FUNCTION:                            * 00090000
.*                                     * 00100000
.* . CCP GENERATION FIRST PASS MACRO-INSTRUCTION -- DEFINE A      * 00110000
.*   BSCA DEVICE TO BE SUPPORTED.                                     * 00120000
.*                                     * 00130000
.* INPUT OPERANDS:                                                  * 00140000
.*                                     * 00150000
.* . TYPE-3275M1/3277M1/3284M1/3286M1/3275M2/3277M2/3284M2/3286M2/ * 00160000
.*   3735/CPU/3741                                                  * 00170000
.*                                     * 00180000
.* SPECIFIES A DEVICE TYPE TO BE SUPPORTED.  REQUIRED OPERAND.     * 00190000
.*                                     * 00200000
*****                                00210000
$EBSD &TYPE-                        00220000
.*                                     00230000
GBLA  &SEQ                            . SEQUENCE CONTROL:      * 00240000
.*                                     . MUST BE 8 OR 9 UPON ENTRY  00250000
.*                                     . IS MADE 9 AFTER PROCESSING 00260000
GBLB  &TERR                            . TERMINATION ERROR SWITCH 00270000
.*                                     00280000
GBLA  &BSC                            . NUMBER BSC ADAPTERS      00290000
GBLC  &BD1,&BD2,&BLT                    . BSC DEVICE AND LINE SUPPORT 00300000
.*                                     00310000
LCLC  &C1,&C2                          . USED FOR CHAR MANIPULATION 00320000
LCLA  &N1,&N2,&N3                        . USED FOR CHAR MANIPULATION 00330000
.*                                     00340000
TABLE &TYPE                            . DEVICE TYPE SUPPORTED    00350000
3275M1 TABDF 10                          BYTE=1, BIT=0              00360000
3277M1 TABDF 11                          BYTE=1, BIT=1              00370000
3284M1 TABDF 12                          BYTE=1, BIT=2              00380000
3286M1 TABDF 13                          BYTE=1, BIT=3              00390000
3275M2 TABDF 14                          BYTE=1, BIT=4              00400000
3277M2 TABDF 15                          BYTE=1, BIT=5              00410000
3284M2 TABDF 16                          BYTE=1, BIT=6              00420000
3286M2 TABDF 17                          BYTE=1, BIT=7              00430000
3735   TABDF 20                          BYTE=2, BIT=0              00440000
CPU    TABDF 21                          BYTE=2, BIT=1              00450000
3741   TABDF 22                          BYTE=2, BIT=2              00460000
' '    TABDF ??                          00470000
TABDF *                                  00480000
.*                                     00490000
TEXT                                     00500000
.*                                     00510000
.*----- CHECK STATEMENT SEQUENCE -----* 00520000
.*                                     00530000
AIF   (&SEQ EQ '9').SEQOK                . SKIP IF SUCCEEDING $EBSD 00540000
AIF   (&SEQ EQ '8').SEQ1                 . SKIP IF FIRST $EBSD     00550000
.*                                     00560000
*!500E $EBSD STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR 00570000
&TERR SETB 1                             . SET TERMINATION ERROR SWITCH 00580000
.*                                     00590000

```

```

.SEQ1 ANOP . FIRST $EBSD STATEMENT 00600000
&BD1 SETC '00000000' 00610000
&BD2 SETC '00000000' 00620000
.* 00630000
.SEQOK ANOP . VALID SEQUENCE 00640000
AIF (&BSC NE '0').BDOK . SKIP UNLESS BSCA-0 00650000
.* 00660000
*!505E $EBSD STATEMENT USED, BUT NO BSC ADAPTERS SPECIFIED 00670000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00680000
.* 00690000
.BDOK ANOP 00700000
.* 00710000
.*----- TYPE-XXXXXX -----* 00720000
.* 00730000
AIF (&TYPE NE '??').TYYES . SKIP IF TYPE SPECIFIED 00740000
.* 00750000
*!510E MISSING 'TYPE' OPERAND -- MUST BE SPECIFIED 00760000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00770000
AGO .END 00780000
.TYYES ANOP . TYPE SPECIFIED 00790000
AIF (&TYPE NE '*').TYOK . SKIP IF VALID PARAMETER 00800000
.* 00810000
*!515E INVALID 'TYPE' PARAMETER -- MUST BE BSCA TERMINAL DESIGNATION 00820000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00830000
AGO .END 00840000
.* 00850000
.TYOK ANOP . VALID PARAMETER 00860000
&C1 SETC '&TYPE'(2,1) . GET THE BIT POSITION 00870000
&N1 SETA &C1 . LENGTH OF PRECEDING 00880000
&N2 SETA &N1+2 . POSITION OF FOLLOWING 00890000
&N3 SETA 9-&N2 . LENGTH OF FOLLOWING 00900000
AIF ('&TYPE'(1,1) EQ '2').BD2 . SKIP IF 3735, CPU OR 3741 00910000
.* 00920000
.* TYPE IS A 3270. 00930000
.* 'TYPE' 3270 REQUIRES CS LINES (3275 MAY BE ON SWITCHED LINES) 00940000
.* 00950000
AIF ('&BLT'(3,1) EQ '1').TYCS SKIP IF CS LINE SUPPORTED. 00960000
AIF ('&TYPE'(2,1) EQ '0').TYSW .CHECK FOR SW LINE IF 3275 00970000
AIF ('&TYPE'(2,1) EQ '4').TYSW .CHECK FOR SW LINE IF 3275 00980000
*!520E 3271 CONTROL UNIT REQUIRES CONTROL STATION SUPPORT 00990000
&TERR SETB 1 01000000
.* 01010000
AGO .TYCS . SKIP CHECK FOR SWITCHED LINE 01020000
.TYSW ANOP 01030000
AIF ('&BLT'(4,1) EQ '1').TYCS . SKIP IF 3275 AND SWITCHED 01040000
*!525E 'TYPE' 3275 REQUIRES CONTROL STATION OR SWITCHED SUPPORT 01050000
&TERR SETB 1 01060000
.* 01070000
.TYCS ANOP CONTINUE HERE 01080000
.* 01090000
&C1 SETC '&BD1'(1,&N1) 01100000
&C2 SETC '&BD1'(&N2,&N3) 01110000
&BD1 SETC '&C1.1&C2' . INSERT THE DEVICE TYPE 01120000
AGO .END 01130000
.* 01140000
.BD2 ANOP 01150000
&C1 SETC '&BD2'(1,&N1) 01160000
&C2 SETC '&BD2'(&N2,&N3) 01170000
&BD2 SETC '&C1.1&C2' . SET THE DEVICE TYPE 01180000
.* 01190000

```

```
.TYEND ANOP                                01200000
.*                                           01210000
.*----- SET STATEMENT SEQUENCE -----* 01220000
.*                                           01230000
.END ANOP                                    01240000
&SEQ SETA 9 . INDICATE $EBSD PROCESSED    01250000
.*                                           01260000
      MEND                                   01270000
```

MODULE-\$EFAC , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* * 00030000
.* NAME: $EFAC * 00040000
.* * 00050000
.* MODIFICATION LEVEL: VERSION 7, MODIFICATION LEVEL 0 * 00060000
.* * 00070000
.* FUNCTION: * 00080000
.* * 00090000
.* . CCP GENERATION FIRST PASS MACRO-INSTRUCTION -- DEFINE * 00100000
.* FUNCTIONAL FACILITIES DESIRED IN USER'S VERSION OF CCP. * 00110000
.* * 00120000
.* INPUT OPERANDS: * 00130000
.* * 00140000
.* * 00150000
.* . ESCAPE-NO/'6 CHARACTERS'/X'12 HEX DIGITS' * 00160000
.* * 00170000
.* DATA MODE ESCAPE FEATURE IS TO BE SUPPORTED AND THE INPUT * 00180000
.* STRING FROM A TERMINAL WHICH SIGNALS THE REQUEST FOR ESCAPE * 00190000
.* IS SPECIFIED. DEFAULT IS NO (FEATURE IS NOT TO BE SUPPORTED. * 00200000
.* * 00210000
.* . PGMCNT-YES/Y/NO/N * 00220000
.* * 00230000
.* COUNTS ARE TO BE KEPT, PER USER PROGRAM, OF REQUESTS FOR THAT * 00240000
.* PROGRAM. DEFAULT IS NO. * 00250000
.* * 00260000
.* . SYMFIL-YES/Y/NO/N * 00270000
.* * 00280000
.* THE SYMBOLIC FILE REFERENCE FACILITY IS TO BE SUPPORTED. * 00290000
.* DEFAULT IS NO. * 00300000
.* * 00310000
.* . FORMAT-YES/Y/NO/N * 00320000
.* * 00330000
.* THE DISPLAY FORMAT FACILITY FOR 3270 IS TO BE INCLUDED. * 00340000
.* DEFAULT IS NO. * 00350000
.* * 00360000
.* . RUF-YES/Y/NO/N * 00370000
.* * 00380000
.* THE 'READ UNDER FORMAT' FACILITY IS TO BE INCLUDED. * 00390000
.* DEFAULT IS NO. * 00400000
.* * 00410000
.* . ACCEPT-YES/Y/NO/N 00420000
.* * 00430000
.* THE 'RESIDENT ACCEPT INPUT' FACILITY IS TO BE SUPPORTED. 00440000
.* DEFAULT IS NO. 00450000
.* * 00450900
.* . BSYPRN-YES/Y/NO/N 00451800
.* * 00452700
.* THE '328X BUSY PRINTER' FACILITY IS TO BE SUPPORTED. 00453600
.* DEFAULT IS NO. 00454500
.* * 00455400
.* . CPUMSG-YES/Y/NO/N 00456300
.* * 00457200
.* ALLOW S TYPE (INFORMATIONAL) MESSAGES TO CPU TERMINALS. 00458100
.* DEFAULT IS NO. 00459000
.* * 00460000
.* ***** 00470000
.* $EFAC &ESCAPE-NO,&PGMCNT-NO,&SYMFIL-NO,&FORMAT-NO,&PRUF-NO, X00480000
.* &ACCEPT-NO,&BSYPRT-NO,&CPUMSG-NO 00490000
```

```

.*
GBLB &TERR . TERMINATION ERROR SWITCH 00500000
GBLA &SEQ . STATEMENT SEQUENCE CONTROL 00510000
.* . MUST BE 0 UPON ENTRY 00520000
.* . IS MADE 2 AFTER PROCESSING 00530000
.* 00540000
.* 00550000
GBLB &FDME . DATA-MODE ESCAPE USED (ESCAPE) 00560000
GBLB &FPGC . PGM-RQST COUNTS USED (PGMCNT) 00570000
GBLB &FSYM . SYMBOLIC FILES USED (SYMFIL) 00580000
GBLB &FDFD . DISPLAY FORMAT USED (FORMAT) 00590000
GBLB &FRUF . 'READ UNDER FORMAT' (RUF) 00600000
GBLB &FAI . RESIDENT ACCEPT INPUT (ACCEPT) 00610000
GBLC &X1DME,&X2DME . DATA MAOE ESCAPE STRING 00620000
GBLB &FPBSY . 328X PRINTER BUSY SUPPORT 00623000
GBLB &FCPUM . SEND S TYPE MESSAGES TO CPU'S 00626000
.* 00630000
LCLA &N . USED FOR LENGTH OF ESCAPE CHAR 00640000
LCLA &NUM FOR TESTING NUMERIC OPERANDS 00650000
.* 00660000
.*----- PARAMETER TRANSFORMATION TABLES -----* 00670000
.* 00680000
.* 00690000
TABLE &SYMFIL . SYMBOLIC DISK FILE FEATURE 00700000
YES TABDF 1 00710000
Y TABDF 1 00720000
NO TABDF 0 00730000
N TABDF 0 00740000
TABDF * . ERRONEOUS PARAMETER 00750000
.* 00760000
TABLE &PGMCNT . PROGRAM-REQUEST COUNTS 00770000
YES TABDF 1 00780000
Y TABDF 1 00790000
NO TABDF 0 00800000
N TABDF 0 00810000
TABDF * . ERRONEOUS PARAMETER 00820000
.* 00830000
TABLE &FORMAT . DISPLAY FORMAT FACILITY 00840000
YES TABDF 1 00850000
Y TABDF 1 00860000
NO TABDF 0 00870000
N TABDF 0 00880000
TABDF * . ERRONEOUS PARAMETER 00890000
.* 00900000
TABLE &PRUF . READ UNDER FORMAT 00910000
YES TABDF 1 00920000
Y TABDF 1 00930000
NO TABDF 0 00940000
N TABDF 0 00950000
TABDF * . ERRONEOUS PARAMETER 00960000
.* 00970000
TABLE &ACCEPT . RESIDENT ACCEPT INPUT FEATURE 00980000
YES TABDF 1 00990000
Y TABDF 1 01000000
NO TABDF 0 01010000
N TABDF 0 01020000
TABDF * . ERRONEOUS PARAMETER 01030000
.* 01030600
TABLE &BSYPRT . 328X PRINTER BUSY FACILITY 01031200
YES TABDF 1 01031800
Y TABDF 1 01032400

```

NO	TABDF	0			01033000
N	TABDF	0			01033600
	TABDF	*	. ERRONEOUS PARAMETER		01034200
.*					01034800
	TABLE	&CPUMSG	. SEND S MESSAGES TO CPU'S		01035400
YES	TABDF	1			01036000
Y	TABDF	1			01036600
NO	TABDF	0			01037200
N	TABDF	0			01037800
	TABDF	*	. ERRONEOUS PARAMETER		01038400
.*					01039000
	TEXT				01040000
.*					01050000
.*	-----	CHECK STATEMENT SEQUENCE	-----*		01060000
.*					01070000
	AIF	(&SEQ EQ '0').SEQOK	. SKIP IF GOOD SEQUENCE		01080000
.*					01090000
*!050E	\$EFAC	STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR			01100000
&TERR	SETB	1			01110000
&FDME	SETB	0	. ERROR--MAKE NULL PREVIOUS SPEC		01120000
&FPGC	SETB	0	. ERROR--MAKE NULL PREVIOUS SPEC		01130000
&FSYM	SETB	0	. ERROR--MAKE NULL PREVIOUS SPEC		01140000
&FDFF	SETB	0	. ERROR--MAKE NULL PREVIOUS SPEC		01150000
&FRUF	SETB	0	. ERROR--MAKE NULL PREVIOUS SPEC		01160000
&FAI	SETB	0	. ERROR--MAKE NULL PREVIOUS SPEC		01170000
&X1DME	SETC	' '	. ERROR--MAKE NULL PREVIOUS SPEC		01180000
&X2DME	SETC	' '	. ERROR--MAKE NULL PREVIOUS SPEC		01190000
.*					01200000
.SEQOK	ANOP				01210000
.*					01220000
.*	-----	SYMFIL-YES/NO (YES-1, NO-0)	-----*		01230000
.*					01240000
	AIF	(&SYMFIL NE '*').SFOK	. SKIP IF VALID PARAMETER		01250000
.*					01260000
*!075E	INVALID	'SYMFIL' PARAMETER -- MUST BE YES/Y/NO/N			01270000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH		01280000
	AGO	.SFEND			01290000
.*					01300000
.SFOK	ANOP		. VALID PARAMETER		01310000
	AIF	(&SYMFIL EQ '0').SFEND	. SKIP IF SYMFIL-NO		01320000
.*					01330000
&FSYM	SETB	1	. SYMFIL-YES		01340000
.*					01350000
.SFEND	ANOP		. END OPERAND 'SYMFIL'		01360000
.*					01370000
.*	-----	PGMCNT-YES/NO (YES-1, NO-0)	-----*		01380000
.*					01390000
	AIF	(&PGMCNT NE '*').PCOK	. SKIP IF VALID PARAMETER		01400000
.*					01410000
*!080E	INVALID	'PGMCNT' PARAMETER -- MUST BE YES/Y/NO/N			01420000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH		01430000
	AGO	.PCEND			01440000
.*					01450000
.PCOK	ANOP		. VALID PARAMETER		01460000
	AIF	(&PGMCNT EQ '0').PCEND	. SKIP IF PGMCNT-NO		01470000
.*					01480000
&FPGC	SETB	1	. PGMCNT-YES		01490000
.*					01500000
.PCEND	ANOP		. END OPERAND 'PGMCNT'		01510000
.*					01520000

```

.*----- ESCAPE-NO/'CCCCC'/X'XXXXXXXXXXXX' -----* 01530000
.* 01540000
&N SETA K'&ESCAPE . NUMBER OF CHARS WRITTEN 01550000
AIF (&ESCAPE EQ 'NO').DMEND . SKIP IF ESCAPE-NO 01560000
AIF (&ESCAPE EQ 'N').DMEND . SKIP IF ESCAPE-N 01570000
AIF ('&ESCAPE'(1,2) EQ 'X''').DMX . SKIP IF ESCAPE-HEX 01580000
.* 01590000
.DMC ANOP . ESCAPE-'CCCCC' 01600000
AIF (&N LT '6').DMERR . SKIP IF LT 6 CHARS -- ERROR 01610000
.* 01620000
&FDME SETB 1 01630000
&X2DME SETC '&ESCAPE'(1,4) . GET 1ST FOUR CHARS 01640000
&X1DME SETC 'CL6'&X2DME' . CL6'XXXX 01650000
&X2DME SETC '&ESCAPE'(5,8) . GET REMAINING CHARACTERS 01660000
AIF ('&ESCAPE'(1,1) NE ''').DM2 . SKIP IF THIS CHAR *NOT* ' 01670000
AIF ('&ESCAPE'(2,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 01680000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 01690000
AGO .DM3 . WAS QUOTE -- SKIP A CHARACTER 01700000
.DM2 ANOP 01710000
AIF ('&ESCAPE'(2,1) NE ''').DM3 . SKIP IF THIS CHAR *NOT* ' 01720000
AIF ('&ESCAPE'(3,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 01730000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 01740000
AGO .DM4 . WAS QUOTE -- SKIP A CHARACTER 01750000
.DM3 ANOP 01760000
AIF ('&ESCAPE'(3,1) NE ''').DM4 . SKIP IF THIS CHAR *NOT* ' 01770000
AIF ('&ESCAPE'(4,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 01780000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 01790000
AGO .DM5 . WAS QUOTE -- SKIP A CHARACTER 01800000
.DM4 ANOP 01810000
AIF ('&ESCAPE'(4,1) NE ''').DM5 . SKIP IF THIS CHAR *NOT* ' 01820000
AIF ('&ESCAPE'(5,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 01830000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 01840000
AGO .DM6 . WAS QUOTE -- SKIP A CHARACTER 01850000
.DM5 ANOP 01860000
AIF ('&ESCAPE'(5,1) NE ''').DM6 . SKIP IF THIS CHAR *NOT* ' 01870000
AIF ('&ESCAPE'(6,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 01880000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 01890000
AGO .DM7 . WAS QUOTE -- SKIP A CHARACTER 01900000
.DM6 ANOP 01910000
AIF ('&ESCAPE'(6,1) NE ''').DM7 . SKIP IF THIS CHAR *NOT* ' 01920000
AIF ('&ESCAPE'(7,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 01930000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 01940000
AGO .DM8 . WAS QUOTE -- SKIP A CHARACTER 01950000
.DM7 ANOP 01960000
AIF ('&ESCAPE'(7,1) NE ''').DM8 . SKIP IF THIS CHAR *NOT* ' 01970000
AIF ('&ESCAPE'(8,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 01980000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 01990000
AGO .DM9 . WAS QUOTE -- SKIP A CHARACTER 02000000
.DM8 ANOP 02010000
AIF ('&ESCAPE'(8,1) NE ''').DM9 . SKIP IF THIS CHAR *NOT* ' 02020000
AIF ('&ESCAPE'(9,1) NE ''').DMERR . ERROR IF NEXT NOT QUOTE 02030000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 02040000
AGO .DM10 . WAS QUOTE -- SKIP A CHARACTER 02050000
.DM9 ANOP 02060000
AIF ('&ESCAPE'(9,1) NE ''').DM10 . SKIP IF THIS CHAR *NOT* ' 02070000
AIF ('&ESCAPE'(10,1) NE ''').DMERR . ERR IF NEXT NOT QUOTE 02080000
&N SETA &N-1 . REDUCE BY 1 FOR DOUBLE QUOTE 02090000
AGO .DM11 . WAS QUOTE -- SKIP A CHARACTER 02100000
.DM10 ANOP 02110000
AIF ('&ESCAPE'(10,1) NE ''').DM11 . SKIP IF THIS *NOT* ' 02120000

```

&N	AIF	('&ESCAPE'(11,1) NE ''').DMERR	. ERR IF NEXT NOT QUOTE	02130000
	SETA	&N-1	. REDUCE BY 1 FOR DOUBLE QUOTE	02140000
	AGO	.DMCK6	. WAS QUOTE -- GO CHECK COUNT	02150000
.DM11	ANOP			02160000
	AIF	('&ESCAPE'(11,1) NE ''').DMCK6	. SKIP IF THIS NOT QUOTE	02170000
	AIF	('&ESCAPE'(12,1) NE ''').DMERR	. ERR IF NEXT NOT QUOTE	02180000
&N	SETA	&N-1	. REDUCE BY 1 FOR DOUBLE QUOTE	02190000
.*				02200000
.DMCK6	ANOP		. NOW CHECK THE COUNT FOR REAL	02210000
	AIF	(&N NE '6').DMERR	. ERROR IF NOT EXACTLY 6	02220000
.*				02230000
	AGO	.DMEND		02240000
.*				02250000
.DMX	ANOP		. ESCAPE-X'XXXXXXXXXXXXX'	02260000
	AIF	(&N NE '15').DMERR	. SKIP IF INVALID NO. CHARS	02270000
	AIF	('&ESCAPE'(15,1) NE ''').DMERR	. SKIP IF LAST CHAR NOT '	02280000
.*				02290000
&FDME	SETB	1		02300000
&X2DME	SETC	'&ESCAPE'(2,5)		02310000
&X1DME	SETC	'XL6&X2DME'		02320000
&X2DME	SETC	'&ESCAPE'(7,8)		02330000
.*				02340000
	AIF	('&ESCAPE'(3,1) GT '9').DMERR	. ERROR IF GT 9	02350000
	AIF	('&ESCAPE'(3,1) LT 'A').DMERR	. ERROR IF LT 'A'	02360000
	AIF	('&ESCAPE'(3,1) GE '0').DX4	. OK IF IN RANGE 0-9	02370000
	AIF	('&ESCAPE'(3,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02380000
.DX4	ANOP			02390000
	AIF	('&ESCAPE'(4,1) GT '9').DMERR	. ERROR IF GT 9	02400000
	AIF	('&ESCAPE'(4,1) LT 'A').DMERR	. ERROR IF LT 'A'	02410000
	AIF	('&ESCAPE'(4,1) GE '0').DX5	. OK IF IN RANGE 0-9	02420000
	AIF	('&ESCAPE'(4,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02430000
.DX5	ANOP			02440000
	AIF	('&ESCAPE'(5,1) GT '9').DMERR	. ERROR IF GT 9	02450000
	AIF	('&ESCAPE'(5,1) LT 'A').DMERR	. ERROR IF LT 'A'	02460000
	AIF	('&ESCAPE'(5,1) GE '0').DX6	. OK IF IN RANGE 0-9	02470000
	AIF	('&ESCAPE'(5,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02480000
.DX6	ANOP			02490000
	AIF	('&ESCAPE'(6,1) GT '9').DMERR	. ERROR IF GT 9	02500000
	AIF	('&ESCAPE'(6,1) LT 'A').DMERR	. ERROR IF LT 'A'	02510000
	AIF	('&ESCAPE'(6,1) GE '0').DX7	. OK IF IN RANGE 0-9	02520000
	AIF	('&ESCAPE'(6,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02530000
.DX7	ANOP			02540000
	AIF	('&ESCAPE'(7,1) GT '9').DMERR	. ERROR IF GT 9	02550000
	AIF	('&ESCAPE'(7,1) LT 'A').DMERR	. ERROR IF LT 'A'	02560000
	AIF	('&ESCAPE'(7,1) GE '0').DX8	. OK IF IN RANGE 0-9	02570000
	AIF	('&ESCAPE'(7,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02580000
.DX8	ANOP			02590000
	AIF	('&ESCAPE'(8,1) GT '9').DMERR	. ERROR IF GT 9	02600000
	AIF	('&ESCAPE'(8,1) LT 'A').DMERR	. ERROR IF LT 'A'	02610000
	AIF	('&ESCAPE'(8,1) GE '0').DX9	. OK IF IN RANGE 0-9	02620000
	AIF	('&ESCAPE'(8,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02630000
.DX9	ANOP			02640000
	AIF	('&ESCAPE'(9,1) GT '9').DMERR	. ERROR IF GT 9	02650000
	AIF	('&ESCAPE'(9,1) LT 'A').DMERR	. ERROR IF LT 'A'	02660000
	AIF	('&ESCAPE'(9,1) GE '0').DX10	. OK IF IN RANGE 0-9	02670000
	AIF	('&ESCAPE'(9,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02680000
.DX10	ANOP			02690000
	AIF	('&ESCAPE'(10,1) GT '9').DMERR	. ERROR IF GT 9	02700000
	AIF	('&ESCAPE'(10,1) LT 'A').DMERR	. ERROR IF LT 'A'	02710000
	AIF	('&ESCAPE'(10,1) GE '0').DX11	. OK IF IN RANGE 0-9	02720000

.DX11	AIF	('&ESCAPE'(10,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02730000
	ANOP			02740000
	AIF	('&ESCAPE'(11,1) GT '9').DMERR	. ERROR IF GT 9	02750000
	AIF	('&ESCAPE'(11,1) LT 'A').DMERR	. ERROR IF LT 'A'	02760000
	AIF	('&ESCAPE'(11,1) GE '0').DX12	. OK IF IN RANGE 0-9	02770000
	AIF	('&ESCAPE'(11,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02780000
.DX12	ANOP			02790000
	AIF	('&ESCAPE'(12,1) GT '9').DMERR	. ERROR IF GT 9	02800000
	AIF	('&ESCAPE'(12,1) LT 'A').DMERR	. ERROR IF LT 'A'	02810000
	AIF	('&ESCAPE'(12,1) GE '0').DX13	. OK IF IN RANGE 0-9	02820000
	AIF	('&ESCAPE'(12,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02830000
.DX13	ANOP			02840000
	AIF	('&ESCAPE'(13,1) GT '9').DMERR	. ERROR IF GT 9	02850000
	AIF	('&ESCAPE'(13,1) LT 'A').DMERR	. ERROR IF LT 'A'	02860000
	AIF	('&ESCAPE'(13,1) GE '0').DX14	. OK IF IN RANGE 0-9	02870000
	AIF	('&ESCAPE'(13,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02880000
.DX14	ANOP			02890000
	AIF	('&ESCAPE'(14,1) GT '9').DMERR	. ERROR IF GT 9	02900000
	AIF	('&ESCAPE'(14,1) LT 'A').DMERR	. ERROR IF LT 'A'	02910000
	AIF	('&ESCAPE'(14,1) GE '0').DMEND	. OK IF IN RANGE 0-9	02920000
	AIF	('&ESCAPE'(14,1) GT 'F').DMERR	. ERR IF NOT IN RANGE A-F	02930000
	AGO	.DMEND		02940000
.*				02950000
.DMERR	ANOP	. INVALID PARAMETER		02960000
*!085E	INVALID	'ESCAPE' PARAMETER -- MUST BE '6 CHARS'/X'12 CHARS'/NO		02970000
&TERR	SETB	1 . SET TERMINATION ERROR SWITCH		02980000
.*				02990000
.DMEND	ANOP	. END OPERAND 'ESCAPE'		03000000
.*				03010000
.*	-----	FORMAT-YES/NO (YES-1, NO-0) -----*		03020000
.*				03030000
	AIF	(&FORMAT NE '*').FMOK	. SKIP IF VALID PARAMETER	03040000
.*				03050000
*!090E	INVALID	'FORMAT' PARAMETER -- MUST BE YES/Y/NO/N		03060000
&TERR	SETB	1 . SET TERMINATION ERROR SWITCH		03070000
	AGO	.FMEND		03080000
.*				03090000
.FMOK	ANOP	. VALID PARAMETER		03100000
	AIF	(&FORMAT EQ '0').FMEND	. SKIP IF FORMAT-NO	03110000
.*				03120000
&FDFE	SETB	1 . FORMAT-YES		03130000
.*				03140000
.FMEND	ANOP	. END OPERAND 'FORMAT'		03150000
.*				03160000
.*	-----	PRUF-YES/NO (YES-1, NO-0) -----*		03170000
.*				03180000
	AIF	(&PRUF NE '*').RFMOK	. SKIP IF VALID PARAMETER	03190000
.*				03200000
*!091E	INVALID	'PRUF' PARAMETER ---- MUST BE YES/Y/NO/N		03210000
&TERR	SETB	1 . SET TERMINATION ERROR SWITCH		03220000
	AGO	.RFEND		03230000
.*				03240000
.RFMOK	ANOP	. VALID PARAMETER		03250000
	AIF	(&PRUF EQ '0').RFEND	. SKIP IF PRUF-NO	03260000
.*				03270000
&FRUF	SETB	1 . PRUF-YES		03280000
.*				03290000
.RFEND	ANOP	. END OPERAND 'PRUF'		03300000
.*				03310000
.*	-----	ACCEPT-YES/NO (YES-1,NO-0) -----*		03320000

```

.*
      AIF      (&ACCEPT NE '*').RAIOK      . SKIP IF VALID PARAMETER      03330000
.*
      AIF      (&ACCEPT NE '*').RAIOK      . SKIP IF VALID PARAMETER      03340000
.*
      03350000
*!093E INVALID 'ACCEPT' PARAMETER -- MUST BE YES/Y/NO/N      03360000
&TERR SETB 1      . SET TERMINATION ERROR SWITCH      03370000
      AGO      .RAEND      03380000
.*
      03390000
.RAIOK ANOP      . VALID PARAMETER      03400000
      AIF      (&ACCEPT EQ '0').RAEND      . SKIP IF ACCEPT-NO      03410000
.*
      03420000
&FAI SETB 1      . ACCEPT-YES      03430000
.*
      03440000
.RAEND ANOP      . END OPERAND 'ACCEPT'      03450000
.*
      03460000
.*----- BSYPRT-YES/NO (YES-1,NO-0) -----*      03460300
.*
      03460600
      AIF      (&BSYPRT NE '*').BYOK      . SKIP IF VALID PARAMETER      03460900
.*
      03461200
*!097E INVALID 'BSYPRT' PARAMETER -- MUST BE YES/Y/NO/N      03461500
&TERR SETB 1      . SET TERMINATION ERROR SWITCH      03461800
      AGO      .BYEND      03462100
.*
      03462400
      BYOK ANOP      . VALID PARAMETER      03462700
      AIF      (&BSYPRT EQ '0').BYEND      . SKIP IF ACCEPT-NO      03463000
.*
      03463300
&FPBSY SETB 1      . BSYPRT-YES      03463600
.*
      03463900
      BYEND ANOP      . END OPERAND 'BSYPRT'      03464200
.*
      03464500
.*----- CPUMSG-YES/NO (YES-1,NO-0) -----*      03464800
.*
      03465100
      AIF      (&CPUMSG NE '*').MSOK      . SKIP IF VALID PARAMETER      03465400
.*
      03465700
*!099E INVALID 'CPUMSG' PARAMETER -- MUST BE YES/Y/NO/N      03466000
&TERR SETB 1      . SET TERMINATION ERROR SWITCH      03466300
      AGO      .MSEND      03466600
.*
      03466900
      MSOK ANOP      . VALID PARAMETER      03467200
      AIF      (&CPUMSG EQ '0').MSEND      . SKIP IF CPUMSG-NO      03467500
.*
      03467800
&FCPUM SETB 1      . CPUMSG-YES      03468100
.*
      03468400
      MSEND ANOP      . END OPERAND 'CPUMSG'      03468700
.*
      03469000
.*----- SET STATEMENT SEQUENCE -----*      03470000
.*
      03480000
&SEQ SETA 2      . INDICATE $EFAC PROCESSED      03490000
.*
      03500000
      MEND      03510000

```

```

MODULE-$EFIL , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO 00010000
***** 00020000
.* * 00030000
.* NAME: $EFIL * 00040000
.* * 00050000
.* MODIFICATION LEVEL: VERSION 0, MODIFICATION LEVEL 0 * 00060000
.* * 00070000
.* FUNCTION: * 00080000
.* * 00090000
.* . CCP GENERATION FIRST PASS MACRO-INSTRUCTION -- DEFINE THE * 00100000
.* SPACE REQUIREMENTS OF THE USER'S $CCPFILE. * 00110000
.* * 00120000
.* INPUT OPERANDS: * 00130000
.* * 00140000
.* . FLUNIT-R1/F1/R2/F2 * 00150000
.* * 00160000
.* DISK UNIT ONTO WHICH AN INITIALIZED (BUT EMPTY) $CCPFILE IS * 00170000
.* TO BE CREATED BY GENERATION. REQUIRED OPERAND. * 00180000
.* * 00190000
.* . FLPACK-PACKNAME * 00200000
.* * 00210000
.* VOLUME ID OF THE PACK ONTO WHICH $CCPFILE IS TO BE GENERATED. * 00220000
.* REQUIRED OPERAND. * 00230000
.* * 00240000
.* . SETS-1/N * 00250000
.* * 00260000
.* NUMBER OF ASSIGNMENT SETS PLANNED. DEFAULT IS 1. * 00270000
.* * 00280000
.* . PROGS-10/N * 00290000
.* * 00300000
.* PLANNED NUMBER OF PROGRAMS PER ASSIGNMENT SET. DEFAULT IS 10. * 00310000
.* * 00320000
.* . DFILES-5/N * 00330000
.* * 00340000
.* NUMBER OF DISK FILES PLANNED PER ASSIGNMENT SET. DEFAULT IS 5. * 00350000
.* * 00360000
.* . TERMS-1/N * 00370000
.* * 00380000
.* NUMBER OF TERMINALS PLANNED PER ASSIGNMENT SET. DEFAULT IS 1. * 00390000
.* * 00400000
.* . DUMPS-1/N * 00410000
.* * 00420000
.* MAXIMUM NUMBER OF DUMPS TO DISK PER CCP EXECUTION. DEFAULT * 00430000
.* IS 1. * 00440000
.* * 00450000
.* . CORE-48K/64K/96K/128K * 00460000
.* * 00470000
.* MAIN STORAGE SIZE OF CPU ON WHICH CCP WILL BE EXECUTED. * 00480000
.* DEFAULT IS 48K. * 00490000
.* * 00500000
.* . TRKLOC-TRACKNUMBER * 00510000
.* * 00520000
.* NUMBER OF THE TRACK ON WHICH $CCPFILE SHOULD BE GENERATED. * 00530000
.* IF NOT SPECIFIED, THE STANDARD ALLOCATION ALGORITHM OF * 00540000
.* DISK SYSTEM MANAGEMENT WILL BE USED TO ASSIGN THE LOCATION ON * 00550000
.* THE SPECIFIED PACK. * 00560000
.* * 00570000
.* ***** 00580000
$EFIL &SETS-1,&PROGS-10,&DFILES-5,&TERMS-1,&DUMPS-1, X00590000

```

```

        &CORE-48K, &FLUNIT-, &FLPACK-, &TRKLOC-          00600000
.*                                                    00610000
        GBLA &SEQ          . SEQUENCE CONTROL:          00620000
.*                                                    . MUST BE 3 OR 4 UPON ENTRY 00630000
.*                                                    . IS MADE 5 AFTER PROCESSING 00640000
        GBLB &TERR          . TERMINATION ERROR SWITCH 00650000
        GBLA &NS            . NUMBER OF SETS            00660000
        GBLA &NPM           . NUMBER OF PROGRAMS        00670000
        GBLA &NDF           . NUMBER OF DISK FILES       00680000
        GBLA &NT            . NUMBER OF TERMINALS        00690000
        GBLA &DMP           . NUMBER OF DUMPS            00700000
        GBLA &COR           . CORE SIZE IN BYTES--       00710000
        GBLA &TFIL          . TRACK LOCATION            00720000
        GBLC &UFIL          . UNIT FOR $CCPFILE         00730000
        GBLC &PFIL          . PACK FOR $CCPFILE         00740000
        LCLA &N             . USED TO COUNT CHARACTERS   00750000
        LCLA &NUM           TO TEST NUMERIC OPERANDS    00760000
.*                                                    00770000
        TABLE &CORE          . MAIN STORAGE SIZE      00780000
48K  TABDF 192             48K                          00790000
64K  TABDF 256             64K                          00800000
96K  TABDF 384             96K                          00810000
128K TABDF 512            128K                         00820000
160K TABDF 640            160K                         00830000
192K TABDF 768            192K                         00840000
224K TABDF 896            224K                         00850000
256K TABDF 1024          256K                         00860000
        TABDF *                          00870000
.*                                                    00880000
        TABLE &FLUNIT          . UNIT FOR $CCPFILE    00890000
R1   TABDF R1              00900000
F1   TABDF F1              00910000
R2   TABDF R2              00920000
F2   TABDF F2              00930000
' '  TABDF ??              . MISSING OPERAND      00940000
        TABDF **              . ERROR PARAMETER      00950000
.*                                                    00960000
        TEXT                    00970000
.*                                                    00980000
.*----- CHECK STATEMENT SEQUENCE -----*          00990000
.*                                                    01000000
        AIF (&SEQ EQ '3').SEQOK 01010000
        AIF (&SEQ EQ '4').SEQOK 01020000
.*                                                    01030000
*!200E $EFIL STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR 01040000
&TERR SETB 1              . SET TERMINATION ERROR SWITCH 01050000
.*                                                    01060000
.SEQOK ANOP                . STATEMENT IN PROPER SEQUENCE 01070000
.*                                                    01080000
.*----- SETS-1/N (N = 1-25) -----*              01090000
.*                                                    01100000
        AIF (T'&SETS NE 'N').SEERR . SKIP IF SETS NOT NUMERIC 01110000
        AIF (K'&SETS GT '4').SEERR ERR IF MORE THAN 4 DIGITS 01120000
&NUM SETA &SETS           SET NUMERIC VALUE          01130000
        AIF (&NUM LT '1').SEERR . SKIP IF SETS-0 -- ERROR 01140000
        AIF (&NUM LE '25').SEOK . SKIP IF IN RANGE 1-25 01150000
.*                                                    01160000
.SEERR ANOP                01170000
*!205E INVALID 'SETS' PARAMETER -- MUST BE NUMBER IN RANGE 1-25 01180000
&TERR SETB 1              . SET TERMINATION ERROR SWITCH 01190000

```

```

AGO .SEEND 01200000
.* 01210000
.SEOK ANOP . VALID PARAMETER 01220000
&NS SETA &SETS 01230000
.* 01240000
.SEEND ANOP . END OPERAND 'SETS' 01250000
.* 01260000
.*----- PROGS-1/N (N = 1-255) -----* 01270000
.* 01280000
AIF (T'&PROGS NE 'N').PRERR . SKIP IF NOT NUMERIC 01290000
AIF (K'&PROGS GT '4').PRERR . SKIP IF MORE THAN 4 DIGITS-ERR 01300000
&NUM SETA &PROGS . SET NUMERIC VALUE 01310000
AIF (&NUM LT '1').PRERR . SKIP IF PROGS-0 01320000
AIF (&NUM LE '255').PROK . SKIP IF IN RANGE 1-255 01330000
.* 01340000
.PRERR ANOP 01350000
*!210E INVALID 'PROGS' PARAMETER -- MUST BE NUMBER IN RANGE 1-255 01360000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 01370000
AGO .PREND 01380000
.* 01390000
.PROK ANOP . VALID PARAMETER 01400000
&NPM SETA &PROGS 01410000
.* 01420000
.PREND ANOP . END OPERAND 'PROGS' 01430000
.* 01440000
.*----- DFILES-0/N (N = 0-50) -----* 01450000
.* 01460000
AIF (T'&DFILES NE 'N').DFERR . SKIP IF NOT NUMERIC 01470000
AIF (K'&DFILES GT '4').DFERR . SKIP IF MORE THAN 4 DIGITS-ERR 01480000
&NUM SETA &DFILES . SET NUMERIC VALUE 01490000
AIF (&NUM LE '50').DFOK . SKIP IF IN RANGE 0-50 01500000
.* 01510000
.DFERR ANOP 01520000
*!215E INVALID 'DFILES' PARAMETER -- MUST BE NUMBER IN RANGE 0-50 01530000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 01540000
AGO .DFEND 01550000
.* 01560000
.DFOK ANOP . VALID PARAMETER 01570000
&NDF SETA &DFILES 01580000
.* 01590000
.DFEND ANOP . END OPERAND 'DFILES' 01600000
.* 01610000
.*----- TERMS-1/N (N = 1-254) -----* 01620000
.* 01630000
AIF (T'&TERMS NE 'N').TEERR . SKIP IF NOT NUMERIC 01640000
AIF (K'&TERMS GT '4').TEERR . SKIP IF MORE THAN 4 DIGITS-ERR 01650000
&NUM SETA &TERMS . SET NUMERIC VALUE 01660000
AIF (&NUM LT '1').TEERR . SKIP IF TERMS-0 01670000
AIF (&NUM LE '254').TEOK . SKIP IF IN RANGE 1-254 01680000
.* 01690000
.TEERR ANOP 01700000
*!220E INVALID 'TERMS' PARAMETER -- MUST BE NUMBER IN RANGE 1-254 01710000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 01720000
AGO .TEEND 01730000
.* 01740000
.TEOK ANOP . VALID PARAMETER 01750000
&NT SETA &TERMS 01760000
.* 01770000
.TEEND ANOP . END OPERAND 'TERMS' 01780000
.* 01790000
.*----- DUMPS-1/N (N = 1-9) -----*

```

```

.*
AIF (T'&DUMPS NE 'N').DUERR . SKIP IF NOT NUMERIC 01800000
AIF (K'&DUMPS GT '4').DUERR . SKIP IF MORE THAN 4 DIGITS-ERR 01810000
&NUM SETA &DUMPS . SET NUMERIC VALUE 01820000
AIF (&NUM LT '1').DUERR . SKIP IF DUMPS-0 01830000
AIF (&NUM LE '9').DUOK . SKIP IF IN RANGE 1-9 01840000
.*
.DUERR ANOP 01850000
*!225E INVALID 'DUMPS' PARAMETER -- MUST BE NUMBER IN RANGE 1-9 01860000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 01870000
AGO .DUEND 01880000
.*
.DUOK ANOP . VALID PARAMETER 01890000
&DMP SETA &DUMPS 01900000
.*
.DUEND ANOP . END OPERAND 'DUMPS' 01910000
.*
*----- CORE-48K/64K/96K/128K -----* 01920000
.*
AIF (&CORE NE '*').COOK . SKIP IF VALID PARAMETER 01930000
.*
*!230E INVALID 'CORE' -- MUST BE 48K/64K/96K/128K 01940000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 01950000
AGO .COEND 01960000
.*
.COOK ANOP . VALID PARAMETER 01970000
&COR SETA &CORE 01980000
.*
.COEND ANOP . END OPERAND 'CORE' 01990000
.*
*----- FLUNIT-R1/F1/R2/F2 -----* 02000000
.*
AIF (&FLUNIT NE '??').FU001 . SKIP IF OPERAND SPECIFIED 02010000
.*
*!240E MISSING 'FLUNIT' OPERAND -- MUST BE SPECIFIED 02020000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 02030000
AGO .FUEND 02040000
.*
.FU001 AIF (&FLUNIT NE '**').FU002 . SKIP IF VALID PARAMETER 02050000
.*
*!242E INVALID 'FLUNIT' PARAMETER -- MUST BE R1/F1/R2/F2 02060000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 02070000
AGO .FUEND 02080000
.*
.FU002 ANOP . VALID PARAMETER 02090000
&UFIL SETC '&FLUNIT' 02100000
.*
.FUEND ANOP . END OPERAND 'FLUNIT' 02110000
.*
*----- FLPACK-PACKANME (1-6 CHARACTERS) -----* 02120000
.*
AIF (T'&FLPACK NE 'O').FP001 . SKIP IF OPERAND SPECIFIED 02130000
.*
*!244E MISSING 'FLPACK' OPERAND -- MUST BE SPECIFIED 02140000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 02150000
AGO .FPEND 02160000
.*
.FP001 ANOP . OPERAND SPECIFIED 02170000
&N SETA K'&FLPACK . NUMBER OF CHARS SPECIFIED 02180000
AIF (&N LE '6').FPOK . SKIP IF 1-6 CHARACTERS 02190000

```

```

.*
*!246E INVALID 'FLPACK' PARAMETER -- MUST BE 1-6 CHARACTERS 02400000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 02410000
AGO .FPEND 02420000
.* 02430000
.FPOK ANOP . VALID PARAMETER 02440000
&PFIL SETC '&FLPACK' 02450000
.* 02460000
.FPEND ANOP . END OPERAND 'FLPACK' 02470000
.* 02480000
*----- TRKLOC-NNN (N = 8-405) -----* 02490000
.* 02500000
AIF (T'&TRKLOC EQ '0').TLEND . SKIP IF NOT SPECIFIED 02510000
.* 02520000
AIF (T'&TRKLOC NE 'N').TLERR . SKIP IF NOT NUMERIC 02530000
AIF (K'&TRKLOC GT '4').TLERR . SKIP IF MORE THAN 4 DIGITS-ERR 02540000
&NUM SETA &TRKLOC . SET NUMERIC VALUE 02550000
AIF (&NUM LT '8').TLERR . SKIP IF BELOW 8 02560000
AIF (&NUM LE '405').TLOK . SKIP IF IN RANGE 8-405 02570000
.* 02580000
.TLERR ANOP 02590000
*!248E INVALID 'TRKLOC' PARAMETER -- MUST BE NUMBER IN RANGE 8-405 02600000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 02610000
AGO .TLEND 02620000
.* 02630000
.TLOK ANOP . VALID PARAMETER SPECIFIED 02640000
&TFIL SETA &TRKLOC 02650000
.* 02660000
.TLEND ANOP . END OPERAND 'TRKLOC' 02670000
.* 02680000
*----- SET STATEMENT SEQUENCE -----* 02690000
.* 02700000
&SEQ SETA 5 . INDICATE $EFIL PROCESSED 02710000
.* 02720000
MEND 02730000
.* 02740000

```

MODULE-\$EGEN , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $EGEN SYS/3 MOD 15 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 7, MODIFICATION LEVEL 0 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION FIRST PASS MACRO-INSTRUCTION -- MAKE FINAL 00100000
.* CHECKS AND GENERATE SECOND PASS INPUT. 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . DSUNIT-R1/F1 00150000
.* 00160000
.* SPECIFIES UNIT ON WHICH RESIDE: 00170000
.* . LIBRARY MAINTENANCE PROGRAM ($MAINT) 00180000
.* . OVERLAY LINKAGE EDITOR ($OLINK) 00190000
.* . RELOCATABLE MODULES OF THE COMMUNICATIONS IOCS ($$ML., 00200000
.* $$BS., $$BM..) 00210000
.* 00220000
.* . CCUNIT-R1/F1/R2/F2 00230000
.* 00240000
.* . SPECIFIES UNIT ONTO WHICH ARE TO BE GENERATED ALL CCP LOAD 00250000
.* MODULES (THAT IS, ALL DISK OUTPUT FROM GENERATION EXCEPT 00260000
.* $CCPFILE, LANGUAGE SUPPORT ROUTINES, AND TEMPORARILY CREATED 00270000
.* MODULES). 00280000
.* 00290000
.* . WKUNIT-UNIT/'UNIT1,UNIT2,UNIT3' 00300000
.* 00310000
.* SPECIFIES UNIT OR UNITS ON WHICH WORK FILES ARE TO BE 00320000
.* ALLOCATED DURING GENERATION. IF A SINGLE UNIT IS SPECIFIED, 00330000
.* ALL WORK FILES ARE ALLOCATED ON THAT UNIT. IF THREE UNITS 00340000
.* ARE SPECIFIED, THEY ARE USED, RESPECTIVELY, FOR $SOURCE, 00350000
.* $WORK, AND $WORK2. 00360000
.* 00370000
.* . WKPACK-NAME/'NAME1,NAME2,NAME3' 00380000
.* 00390000
.* SPECIFIES THE PACK NAMES THAT CORRESPOND WITH THE UNIT NAMES 00400000
.* PREVIOUSLY SPECIFIED. 00410000
.* 00420000
.* . DIUNIT-R1/F1/R2/F2 00430000
.* 00440000
.* SPECIFIES UNIT ON WHICH IS MOUNTED THE PID DISTRIBUTION PACK. 00450000
.* 00460000
.* . MINRES-YES/NO 00470000
.* 00480000
.* SPECIFIES WHETHER TO GENERATE THE NORMAL RESIDENT CODE OR TO 00490000
.* *MINIMIZE* THE RESIDENT CODE BY REMOVING CERTAIN ITEMS AND BY 00500000
.* MAKING SEVERAL FUNCTIONS TRANSIENT. 00510000
.* 00520000
.* YES INDICATES MINIMIZE RESIDENT CODE -- DEFAULT IS NO. 00530000
.* 00540000
.* . CARD-YES/NO 00550000
.* 00560000
.* SPECIFIES WHETHER TO GENERATE PASS 2 STATEMENTS AS A CARD 00570000
.* DECK OR AS SOURCE AND PROCEDURE MEMBERS. 00580000
.* 00590000
```



```

***** 00600000
$EGEN &DSUNIT-,&CCUNIT-,&WKUNIT-,&WKPACK-,&DIUNIT-,&MINRES-NO, X00610000
      &CARD-YES 00620000
.* 00630000
.*----- COMMON GLOBAL VARIABLES -----* 00640000
.* 00650000
GBLA &SEQ . SEQUENCE CONTROL: 00660000
.* . MUST BE 7/8/9 UPON ENTRY 00670000
.* . IF 8, THEN &BSC MUST BE 0 00680000
.* . IS MADE 10 AFTER PROCESSING 00690000
.* 00700000
GBLB &TERR . TERMINATION ERROR SWITCH 00710000
.* 00720000
.*----- GLOBAL VARIABLES FROM $EFAC -----* 00730000
.* 00740000
GBLB &FDME . BOOLEAN--FROM ESCAPE-VALUE/NO 00750000
.* . 1=STRING SPECIFIED / 0=NO 00760000
.* 00770000
GBLC &X1DME,&X2DME . CHARACTER (8+2/8+8)--ESCAPE 00780000
.* . THE DATA MODE ESCAPE STRING, 00790000
.* . AS A CHARACTER OR HEX 00800000
.* . DC OPERAND LESS FINAL QUOTE: 00810000
.* . CHAR -- CL6 'CCCCCC' 00820000
.* . HEX -- XL6 'XXXXXXXXXXXX' 00830000
.* 00840000
GBLB &FPGC . BOOLEAN--FROM PGMCNT-YES/NO 00850000
.* . 1=YES / 0=NO 00860000
.* 00870000
GBLB &FDFD . BOOLEAN--FROM FORMAT-YES/NO 00880000
.* . 1=YES / 0=NO 00890000
.* 00900000
GBLB &FRUF . BOOLEAN--FROM RUF-YES/NO 00910000
.* . 1=YES / 0=NO 00920000
.* 00930000
GBLB &FAI . BOOLEAN--FROM ACCEPT-YES/NO 00930000
.* . 1=YES / 0=NO 00940000
.* 00950000
GBLB &FPBSY . BOOLEAN--FROM BSYPRY-YES/NO 00951000
.* . 1=YES / 0=NO 00952000
.* 00953000
GBLB &FCPUM . BOOLEAN--FROM CPUMSG-YES/NO 00954000
.* . 1=YES / 0=NO 00955000
.* 00956000
.*----- GLOBAL VARIABLES FROM $EPLG -----* 00960000
.* 00970000
GBLC &PLG . CHARACTER(4)--FROM LANG-XXXXX 00980000
.* . '1' = SUPPORTED / '0' = NOT 00990000
.* . (1) COBOL (3) ASSEM 01000000
.* . (2) FORTRAN (4) RPGII 01010000
.* 01020000
GBLC &UPCBL . CHARACTER(2)--PPUNIT (COBOL) 01030000
.* 01040000
GBLC &UPFOR . CHARACTER(2)--PPUNIT (FORTRAN) 01050000
.* 01060000
GBLC &UPASM . CHARACTER(2)--PPUNIT (ASSEM) 01070000
.* 01080000
GBLC &UPRPG . CHARACTER(2)--PPUNIT (RPG) 01090000
.* 01100000
.*----- GLOBAL VARIABLES FROM $ESEC -----* 01110000
.* 01120000
GBLB &CPW . BOOLEAN--FROM SECURE-CCP 01130000

```

```

.*          .      1 IF SECURE-CCP / ELSE 0      01140000
.*          .      01150000
.*      GBLB  &UPW      .  BOOLEAN--FROM SECURE-USER      01160000
.*          .      1 IF SECURE-USER / ELSE 0      01170000
.*          .      01180000
.*      GBLA  &LUS      .  ARITHMETIC--NUMBER: LUSI-#      01190000
.*          .      01200000
.*----- GLOBAL VARIABLES FROM $EFIL -----* 01210000
.*          .      01220000
.*      GBLA  &NS      .  ARITHMETIC--NUMBER: SETS-#      01230000
.*          .      01240000
.*      GBLA  &NPM      .  ARITHMETIC--NUMBER: PROGS-#      01250000
.*          .      01260000
.*      GBLA  &NDF      .  ARITHMETIC--NUMBER: DFILES-#      01270000
.*          .      01280000
.*      GBLA  &NT      .  ARITHMETIC--NUMBER: TERMS-#      01290000
.*          .      01300000
.*      GBLA  &DMP      .  ARITHMETIC--NUMBER: DUMPS-#      01310000
.*          .      01320000
.*      GBLA  &COR      .  ARITHMETIC--FROM CORE-NNK      01330000
.*          .      192=48K   384=96K      01340000
.*          .      256=64K   512=128K      01350000
.*          .      1024=256K      01360000
.*          .      01370000
.*      GBLC  &UFIL      .  CHARACTER(2)--FROM FLUNIT-XX      01380000
.*          .      01390000
.*      GBLC  &PFIL      .  CHARACTER(6)--FROM FLPACK-XXX      01400000
.*          .      01410000
.*      GBLA  &TFIL      .  ARITHMETIC--NUMBER: TRKLOC-#      01420000
.*          .      0 IF NOT SPECIFIED      01430000
.*          .      01440000
.*----- GLOBAL VARIABLES FROM $EMLA -----* 01450000
.*          .      01460000
.*      GBLA  &MLA      .  ARITHMETIC--NUMBER: LINES-#      01470000
.*          .      0 IF STATEMENT NOT USED      01480000
.*          .      01490000
.*      GBLB  &MNOX      .  BOOLEAN--FROM XLATE-YES/NO      01500000
.*          .      1=NO / 0=YES * NOTE VALUE * 01510000
.*          .      01520000
.*----- GLOBAL VARIABLES FROM $EMLD -----* 01530000
.*          .      01540000
.*      GBLB  &MT40      .  BOOLEAN--SUPPORT 2740 DEVICE      01550000
.*          .      1=SUPPORTED / 0=NOT      01560000
.*          .      01570000
.*      GBLB  &MT41      .  BOOLEAN--SUPPORT 2741 DEVICE      01580000
.*          .      1=SUPPORTED / 0=NOT      01590000
.*          .      01600000
.*      GBLB  &MT50      .  BOOLEAN--SUPPORT 1050 DEVICE      01610000
.*          .      1=SUPPORTED / 0=NOT      01620000
.*          .      01630000
.*      GBLB  &MFSC      .  BOOLEAN--SUPPORT STATION CNTRL 01640000
.*          .      1=SUPPORTED / 0=NOT      01650000
.*          .      01660000
.*      GBLB  &MFSW      .  BOOLEAN--SUPPORT SWITCHED LINE 01670000
.*          .      1=SUPPORTED / 0=NOT      01680000
.*          .      01690000
.*      GBLB  &MFBR      .  BOOLEAN--SUPPORT BUF RCV      01700000
.*          .      1=SUPPORTED / 0=NOT      01710000
.*          .      01720000
.*      GBLB  &MF'TC     .  BOOLEAN--SUPPORT TRANSMIT CTL 01730000

```

```

.*          . 1=SUPPORTED / 0=NOT          01740000
.*          .                                01750000
.* GBLB &MFCK . BOOLEAN--SUPPORT CHECKING TERM 01760000
.*          . 1=SUPPORTED / 0=NOT          01770000
.*          .                                01780000
.* GBLB &MFNK . BOOLEAN--SUPPORT NON-CHK TERM 01790000
.*          . 1=SUPPORTED / 0=NOT          01800000
.* GBLB &XM4E . BOOLEAN--SUPPORT 2740 PTTCEBCD 01810000
.*          . 1=SUPPORTED / 0=NOT          01820000
.*          .                                01830000
.* GBLC &MD1  . CHARACTER(8)--MLTA DEVICES    01840000
.*          . '1' = SUPPORTED / '0' = NOT 01850000
.*          . (1) 1050 (5) 2740C           01860000
.*          . (2) 1050D (6) 2740SC         01870000
.*          . (3) 2740 (7) 2740D          01880000
.*          . (4) 2740S (8) 2740DT         01890000
.*          .                                01900000
.* GBLC &MD2  . CHARACTER(8)--MLTA DEVICES    01910000
.*          . '1' = SUPPORTED / '0' = NOT 01920000
.*          . (1) 2740DC (5) 2740M2SC      01930000
.*          . (2) 2740DTC (6) 2740M2SCB    01940000
.*          . (3) 2740M2S (7) 2741         01950000
.*          . (4) 2740M2SB (8) 2741D       01960000
.*          .                                01970000
.* GBLC &MD3  . CHARACTER(4)--MLTA DEVICES    01980000
.*          . '1' = SUPPORTED / '0' = NOT 01990000
.*          . (1) SYS7C (3) SYS7DC         02000000
.*          . (2) SYS7SC (4) CMCSTD        02010000
.*          .                                02020000
.* GBLC &MXC  . CHARACTER(3)--FROM XMCODE     02030000
.*          . '1' = SUPPORTED / '0' = NOT 02040000
.*          . (1) XMCODE-CORR              02050000
.*          . (2) XMCODE-PTTCEBCD (2740/1) 02060000
.*          . (3) XMCODE-PTTCBCD           02070000
.*          . (4) XMCODE-PTTCEBCD (1050)   02080000
.*          .                                02090000
.*----- GLOBAL VARIABLES FROM $EBSC -----* 02100000
.*          .                                02110000
.* GBLB &INPOL . BOOLEAN--INTERVAL POLLING    02120000
.*          . 0=NOT / 1=SUPPORT            02130000
.* GBLA &BSC   . ARITHMETIC--NUMBER: BSCA-#   02140000
.*          . 0 IF STATEMENT NOT USED      02150000
.*          .                                02160000
.* GBLB &BIDA  . BOOLEAN--DA SUPPORT          02170000
.*          . 1=SUPPORT / 0=NOT            02180000
.*          .                                02190000
.* GBLC &BLT   . CHARACTER(4)--BSCA LINE TYPES 02200000
.*          . '1' = SUPPORTED / '0' = NOT 02210000
.*          . (1) PP-YES (3) CS-YES        02220000
.*          . (2) MP-YES (4) DIAL-YES      02230000
.*          .                                02240000
.* GBLC &BFA   . CHARACTER(8)--BSCA FEATURES 02250000
.*          . '1' = SUPPORTED / '0' = NOT 02260000
.*          . (1) GETMSG (5) AUTORS        02270000
.*          . (2) ITB (6) EBCDIC           02280000
.*          . (3) RECSEP-XX (7) ASCII      02290000
.*          . (4) RESPOL (8) XPRNCY       02300000
.*          .                                02310000
.* GBLC &RSB   . CHARACTER(2)--REC SEP CHAR (X) 02320000
.*          .                                02330000

```

```

.*----- GLOBAL VARIABLES FROM $EBSD -----* 02340000
.*
.*      GBLC  &BD1          . CHARACTER(8)--BSCA DEVICES 02350000
.*                                     . '1' = SUPPORTED / '0' = NOT 02360000
.*                                     . (1) 3275M1   (5) 3275M2 02370000
.*                                     . (2) 3277M1   (6) 3277M2 02380000
.*                                     . (3) 3284M1   (7) 3284M2 02390000
.*                                     . (4) 3286M1   (8) 3286M2 02400000
.*                                     .                                     02410000
.*                                     .                                     02420000
.*      GBLC  &BD2          . CHARACTER(8)--BSCA DEVICES 02430000
.*                                     . '1' = SUPPORTED / '0' = NOT 02440000
.*                                     . (1) 3735   (2) CPU 02450000
.*                                     . (3) 3741   (4-8) NOT USED 02460000
.*                                     .                                     02470000
.*----- LOCAL VARIABLES WITHIN $EGEN -----* 02480000
.*
.*      LCLC  &C,&D          . CCUNIT, DIUNIT RESPECTIVELY 02490000
.*      LCLB  &MIN          . MINIMUM RESIDENT SYSTEM 02500000
.*      LCLA  &N,&N1,&N2,&N3,&N4 02510000
.*      LCLC  &C1,&C2          .                                     02520000
.*      LCLA  &LVL          . CURRENT SYSTEM LEVEL 02530000
.*      LCLC  &C3          .                                     02540000
.*      LCLC  &UWSRC,&UWWRK,&UWWK2 . UNITS FOR WORK FILES 02550000
.*      LCLC  &PWSRC,&PWWRK,&PWWK2 . PACKS FOR WORK FILES 02560000
.*      LCLA  &ASIZE        . SIZE OF ASSIGNMENT FILE 02570000
.*      LCLC  &#            . FOR SINGLE BLANK 02580000
.*      LCLB  &CL          . CARD GENERATION 02590000
.*                                     .                                     02600000
.*                                     .                                     02610000
.*----- PARAMETER TRANSFORMATION TABLES -----* 02620000
.*
.*      TABLE &DSUNIT      . DSM UNIT 02630000
F1      TABDF F1          .                                     02640000
R1      TABDF R1          .                                     02650000
' '     TABDF ??         . MISSING OPERAND 02660000
' '     TABDF **         . INVALID PARAMETER 02670000
.*                                     .                                     02680000
.*                                     .                                     02690000
.*      TABLE &CCUNIT      . CCP PRODUCTION UNIT 02700000
R1      TABDF R1          .                                     02710000
F1      TABDF F1          .                                     02720000
R2      TABDF R2          .                                     02730000
F2      TABDF F2          .                                     02740000
' '     TABDF ??         . MISSING OPERAND 02750000
' '     TABDF **         . INVALID PARAMETER 02760000
.*                                     .                                     02770000
.*      TABLE &DIUNIT      . PID DISTRIBUTION UNIT 02780000
R1      TABDF R1          .                                     02790000
F1      TABDF F1          .                                     02800000
R2      TABDF R2          .                                     02810000
F2      TABDF F2          .                                     02820000
' '     TABDF ??         . MISSING OPERAND 02830000
' '     TABDF **         . INVALID PARAMETER 02840000
.*                                     .                                     02850000
.*      TABLE &MINRES      . MINIMUM RESIDENT FUNCTION 02860000
YES     TABDF 1          .                                     02870000
Y       TABDF 1          .                                     02880000
NO      TABDF 0          .                                     02890000
N       TABDF 0          .                                     02900000
' '     TABDF *          . INVALID PARAMETER 02910000
.*                                     .                                     02920000
.*      TABLE &CARD        . CARD CCP GENERATION 02930000

```

YES	TABDF 1		02940000
Y	TABDF 1		02950000
NO	TABDF 0		02960000
N	TABDF 0		02970000
	TABDF *	. INVALID PARAMETER	02980000
	TEXT		02990000
.*			03000000
.*	-----	SET CURRENT SYSTEM LEVEL -----*	03010000
.*			03020000
&LVL	SETA 3	. SET LEVEL OF CURRENT RELEASE	03030000
.*			03040000
.*	-----	CHECK STATEMENT SEQUENCE -----*	03050000
.*			03060000
	AIF (&SEQ LE '9').SEQ1	. SKIP UNLESS DUPLICATE	03070000
.*			03080000
*!605E	DUPLICATE \$EGEN STATEMENT -- CONTENTS IGNORED		03090000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	03100000
*? ***	THIS RECORD INDICATES TO \$CC1PP THAT \$EGEN WAS PROCESSED ***		03110000
	MEXIT		03120000
.*			03130000
.SEQ1	ANOP		03140000
	AIF (&SEQ LT '7').SEQER	. SKIP IF MISSING PREV STMTS	03150000
	AIF (&SEQ EQ '7').SEQOK	. SKIP IF \$EMLD LAST	03160000
	AIF (&SEQ EQ '9').SEQOK	. SKIP IF \$EBSD LAST	03170000
	AIF (&BSC EQ '0').SEQOK	. SKIP IF NULL \$EBSC LAST	03180000
.*			03190000
.SEQER	ANOP		03200000
*!600E	\$EGEN STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR		03210000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	03220000
*? ***	THIS RECORD INDICATES TO \$CC1PP THAT \$EGEN WAS PROCESSED ***		03230000
.*			03240000
.SEQOK	ANOP	. VALID SEQUENCE	03250000
.*			03260000
.*	-----	DSUNIT-R1/F1 -----*	03270000
.*			03280000
	AIF (&DSUNIT NE '??').DSYES	. SKIP IF DSUNIT SPECIFIED	03290000
.*			03300000
*!610E	MISSING 'DSUNIT' OPERAND -- MUST BE SPECIFIED		03310000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	03320000
	AGO .DSEND		03330000
.*			03340000
.DSYES	ANOP	. DSUNIT SPECIFIED	03350000
	AIF (&DSUNIT NE '**').DSEND	. SKIP IF VALID PARAMETER	03360000
.*			03370000
*!615E	INVALID 'DSUNIT' PARAMETER -- MUST BE R1/F1		03380000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	03390000
.*			03400000
.DSEND	ANOP	. END OPERAND 'DSUNIT'	03410000
.*			03420000
.*	-----	CCUNIT-R1/F1/R2/F2 -----*	03430000
.*			03440000
	AIF (&CCUNIT NE '??').CCYES	. SKIP IF CCUNIT SPECIFIED	03450000
.*			03460000
*!620E	MISSING 'CCUNIT' OPERAND -- MUST BE SPECIFIED		03470000
&TERR	SETB 1	. SET TERMINATION ERROR SWITCH	03480000
	AGO .CCEND		03490000
.*			03500000
.CCYES	ANOP	. CCUNIT SPECIFIED	03510000
	AIF (&CCUNIT NE '**').CCOK	. SKIP IF VALID PARAMETER	03520000
.*			03530000

*!625E	INVALID 'CCUNIT' PARAMETER -- MUST BE R1/F1/R2/F2	03540000
&TERR	SETB 1 . SET TERMINATION ERROR SWITCH	03550000
	AGO .CCEND	03560000
.*		03570000
.CCOK	ANOP . VALID PARAMETER	03580000
&C	SETC '&CCUNIT'	03590000
.*		03600000
.CCEND	ANOP . END OPERAND 'CCUNIT'	03610000
.*		03620000
.*	----- WKUNIT-UNIT/'UNIT1,UNIT2,UNIT3' -----*	03630000
.*		03640000
	AIF (T'&WKUNIT NE 'O').WUYES . SKIP IF WKUNIT SPECIFIED	03650000
.*		03660000
*!630E	MISSING 'WKUNIT' OPERAND -- MUST BE SPECIFIED	03670000
&TERR	SETB 1 . SET TERMINATION ERROR SWITCH	03680000
	AGO .WUEND	03690000
.*		03700000
.WUYES	ANOP . WKUNIT SPECIFIED	03710000
&N	SETA K'&WKUNIT . LENGTH OF PARAMETER	03720000
	AIF (&N NE '2').WU3 . SKIP IF MULTIPLE UNITS	03730000
.*		03740000
&UWSRC	SETC '&WKUNIT'	03750000
&UWWRK	SETC '&WKUNIT'	03760000
&UWVK2	SETC '&WKUNIT'	03770000
	AGO .WUCHK	03780000
.*		03790000
.WU3	ANOP . WKUNIT SERIES	03800000
	AIF (&N NE '8').WUERR . SKIP IF NOT 8 CHARS	03810000
&UWSRC	SETC '&WKUNIT'(1,2)	03820000
&UWWRK	SETC '&WKUNIT'(4,2)	03830000
&UWVK2	SETC '&WKUNIT'(7,2)	03840000
	AIF ('&WKUNIT'(3,1) NE ',').WUERR . SKIP IF NOT COMMA	03850000
	AIF ('&WKUNIT'(6,1) NE ',').WUERR . SKIP IF NOT COMMA	03860000
.*		03870000
.WUCHK	ANOP . CHECK THAT EACH UNIT IS VALID	03880000
	AIF (&UWSRC EQ 'R1').WUCK2	03890000
	AIF (&UWSRC EQ 'F1').WUCK2	03900000
	AIF (&UWSRC EQ 'R2').WUCK2	03910000
	AIF (&UWSRC EQ 'F2').WUCK2	03920000
	AIF (&UWSRC EQ 'D1').WUCK2	03930000
	AIF (&UWSRC EQ 'D2').WUCK2	03940000
	AIF (&UWSRC EQ 'D3').WUCK2	03950000
	AIF (&UWSRC EQ 'D4').WUCK2	03960000
	AGO .WUERR . SKIP IF ERROR	03970000
.*		03980000
.WUCK2	ANOP	03990000
	AIF (&UWWRK EQ 'R1').WUCK3	04000000
	AIF (&UWWRK EQ 'F1').WUCK3	04010000
	AIF (&UWWRK EQ 'R2').WUCK3	04020000
	AIF (&UWWRK EQ 'F2').WUCK3	04030000
	AIF (&UWWRK EQ 'D1').WUCK3	04040000
	AIF (&UWWRK EQ 'D2').WUCK3	04050000
	AIF (&UWWRK EQ 'D3').WUCK3	04060000
	AIF (&UWWRK EQ 'D4').WUCK3	04070000
	AGO .WUERR . SKIP IF ERROR	04080000
.*		04090000
.WUCK3	ANOP	04100000
	AIF (&UWVK2 EQ 'R1').WUEND	04110000
	AIF (&UWVK2 EQ 'F1').WUEND	04120000
	AIF (&UWVK2 EQ 'R2').WUEND	04130000

AIF	(&UWWK2 EQ 'F2').WUEND	04140000
AIF	(&UWWK2 EQ 'D1').WUEND	04150000
AIF	(&UWWK2 EQ 'D2').WUEND	04160000
AIF	(&UWWK2 EQ 'D3').WUEND	04170000
AIF	(&UWWK2 EQ 'D4').WUEND	04180000
.*		04190000
.WUERR	ANOP	04200000
*!635E	INVALID 'WKUNIT' PARAMETER -- MUST BE R1/F1/R2/F2/D1/D2/D3/D4	04210000
*!635E	-- OR SERIES OF 3.	04220000
&TERR	SETB 1 . SET TERMINATION ERROR SWITCH	04230000
.*		04240000
.WUEND	ANOP . END OPERAND 'WKUNIT'	04250000
.*		04260000
.*	----- WKPACK-NAME/'NAME1,NAME2,NAME3' -----*	04270000
.*		04280000
AIF	(T'&WKPACK NE 'O').WPYES . SKIP IF WKPACK SPECIFIED	04290000
.*		04300000
*!640E	MISSING 'WKPACK' OPERAND -- MUST BE SPECIFIED	04310000
&TERR	SETB 1 . SET TERMINATION ERROR SWITCH	04320000
AGO	.WPEND	04330000
.*		04340000
.WPYES	ANOP . WKPACK SPECIFIED	04350000
&N	SETA K'&WKPACK . LENGTH OF PARAMETER	04360000
AIF	(&N GT '20').WPERR . ERROR IF GT 20 CHARACTERS	04370000
.*		04380000
&N1	SETA 1 . INITIALIZE BASE POINTER	04390000
&N2	SETA &N1 . SET RUNNING POINTER	04400000
AIF	('&WKPACK'(&N2,1) EQ ',').WPERR . ERR IF 1ST CHAR COMMA	04410000
.*		04420000
&N2	SETA &N2+1 . ADVANCE POINTER	04430000
AIF	(&N2 GT &N).WP11 . SKIP IF END-OF-VALUE	04440000
AIF	('&WKPACK'(&N2,1) EQ ',').WP1 . 2ND CHAR	04450000
&N2	SETA &N2+1 . ADVANCE POINTER	04460000
AIF	(&N2 GT &N).WP11 . SKIP IF END-OF-VALUE	04470000
AIF	('&WKPACK'(&N2,1) EQ ',').WP1 . 3RD CHAR	04480000
&N2	SETA &N2+1 . ADVANCE POINTER	04490000
AIF	(&N2 GT &N).WP11 . SKIP IF END-OF-VALUE	04500000
AIF	('&WKPACK'(&N2,1) EQ ',').WP1 . 4TH CHAR	04510000
&N2	SETA &N2+1 . ADVANCE POINTER	04520000
AIF	(&N2 GT &N).WP11 . SKIP IF END-OF-VALUE	04530000
AIF	('&WKPACK'(&N2,1) EQ ',').WP1 . 5TH CHAR	04540000
&N2	SETA &N2+1 . ADVANCE POINTER	04550000
AIF	(&N2 GT &N).WP11 . SKIP IF END-OF-VALUE	04560000
AIF	('&WKPACK'(&N2,1) EQ ',').WP1 . 6TH CHAR	04570000
&N2	SETA &N2+1 . ADVANCE POINTER	04580000
AIF	(&N2 GT &N).WP11 . SKIP IF END-OF-VALUE	04590000
AIF	('&WKPACK'(&N2,1) EQ ',').WP1 . 7TH CHAR	04600000
AIF	(&N GE &N2).WPERR . ERROR IF 1 NAME GT 6 CHARS	04610000
.*		04620000
.WP11	ANOP . ONLY ONE NAME GIVEN.	04630000
&PWSRC	SETC '&WKPACK' . ONE NAME FOR \$SOURCE	04640000
&PWWRK	SETC '&WKPACK' . SAME NAME FOR \$WORK	04650000
&PWWK2	SETC '&WKPACK' . SAME NAME FOR \$WORK2	04660000
AGO	.WPEND . FINISHED THE OPERAND	04670000
.*		04680000
.WP1	ANOP . END OF SUBPARAMETER #1	04690000
&N3	SETA &N2-&N1 . LENGTH OF SUBPARAMETER	04700000
&PWSRC	SETC '&WKPACK' (&N1,&N3) . SET SUBPARAMETER #1	04710000
.*		04720000
AIF	(&N EQ &N2).WPERR . ERROR IF COMMA LAST	04730000

&N1	SETA	&N2+1	. RESET BASE POINTER	04740000
&N2	SETA	&N1	. RESET RUNNING POINTER	04750000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WPERR . ERR IF 1ST CHAR COMMA	04760000
.*				04770000
&N2	SETA	&N2+1	. ADVANCE POINTER	04780000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WP2 . 2ND CHAR	04790000
&N2	SETA	&N2+1	. ADVANCE POINTER	04800000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WP2 . 3RD CHAR	04810000
&N2	SETA	&N2+1	. ADVANCE POINTER	04820000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WP2 . 4TH CHAR	04830000
&N2	SETA	&N2+1	. ADVANCE POINTER	04840000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WP2 . 5TH CHAR	04850000
&N2	SETA	&N2+1	. ADVANCE POINTER	04860000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WP2 . 6TH CHAR	04870000
&N2	SETA	&N2+1	. ADVANCE POINTER	04880000
	AIF	('&WKPAC'(&N2,1) NE ',')	. WPERR . ERR IF NAME GT 6 CHARS	04890000
.*				04900000
.WP2	ANOP		. END OF SUBPARAMETER #2	04910000
&N3	SETA	&N2-&N1	. LENGTH OF SUBPARAMETER	04920000
&PWWRK	SETC	'&WKPAC'(&N1,&N3)	. SET SUBPARAMETER #2	04930000
.*				04940000
	AIF	(&N EQ &N2)	. WPERR . ERROR IF COMMA LAST	04950000
&N1	SETA	&N2+1	. RESET BASE POINTER	04960000
&N2	SETA	&N1	. RESET RUNNING POINTER	04970000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WPERR . ERR IF 1ST CHAR COMMA	04980000
.*				04990000
&N2	SETA	&N2+1	. ADVANCE POINTER	05000000
	AIF	(&N2 GT &N)	. WP3 . SKIP IF END-OF-STRING	05010000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WPERR . 2ND CHAR--ERR IF COMMA	05020000
&N2	SETA	&N2+1	. ADVANCE POINTER	05030000
	AIF	(&N2 GT &N)	. WP3 . SKIP IF END-OF-STRING	05040000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WPERR . 3RD CHAR--ERR IF COMMA	05050000
&N2	SETA	&N2+1	. ADVANCE POINTER	05060000
	AIF	(&N2 GT &N)	. WP3 . SKIP IF END-OF-STRING	05070000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WPERR . 4TH CHAR--ERR IF COMMA	05080000
&N2	SETA	&N2+1	. ADVANCE POINTER	05090000
	AIF	(&N2 GT &N)	. WP3 . SKIP IF END-OF-STRING	05100000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WPERR . 5TH CHAR--ERR IF COMMA	05110000
&N2	SETA	&N2+1	. ADVANCE POINTER	05120000
	AIF	(&N2 GT &N)	. WP3 . SKIP IF END-OF-STRING	05130000
	AIF	('&WKPAC'(&N2,1) EQ ',')	. WPERR . 6TH CHAR--ERR IF COMMA	05140000
&N2	SETA	&N2+1	. ADVANCE POINTER	05150000
	AIF	(&N2 GT &N)	. WP3 . SKIP IF END-OF-STRING	05160000
	AGO	. WPERR	. ERR IF NAME GT 6 CHARS	05170000
.*				05180000
.WP3	ANOP		. END OF SUBPARAMETER #3	05190000
&N3	SETA	&N2-&N1	. LENGTH OF SUBPARAMETER #3	05200000
&PWWK2	SETC	'&WKPAC'(&N1,&N3)	. SET SUBPARAMETER #3	05210000
	AGO	. WPEND		05220000
.*				05230000
.WPERR	ANOP		. ERR IN WKPAC PARAMETER	05240000
*!645E	INVALID	'WKPAC' PARAMETER--MUST BE 1-6 CHAR NAME OR SERIES OF 3		05250000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	05260000
.*				05270000
.WPEND	ANOP		. END OPERAND 'WKPAC'	05280000
.*				05290000
.*-----		DETERMINE NO MISMATCH OF WKUNIT/WKPAC -----*		05300000
.*				05310000
.WK1	ANOP		. CHECK \$SOURCE/\$WORK	05320000
	AIF	(&UWSRC EQ '')	. WKEND . SKIP IF NOT VALIDLY SPECIFIED	05330000

.*	AIF	(&PWWK2 EQ ' ').WKEND	. SKIP IF NOT VALIDLY SPECIFIED	05340000
.*				05350000
&C1	SETC	'&UWSRC'		05360000
	AIF	(&UWSRC NE &UWWRK).WK2		05370000
	AIF	(&PWSRC EQ &PWWRK).WK2		05380000
.*				05390000
*!637E	WKUNIT/WKPACK	ERROR -- PACKS &PWSRC AND &PWWRK BOTH ON UNIT &C1		05400000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	05410000
	AGO	.WKEND		05420000
.*				05430000
.WK2	ANOP		. CHECK \$SOURCE/\$WORK2	05440000
	AIF	(&UWSRC NE &UWWRK2).WK3		05450000
	AIF	(&PWSRC EQ &PWWK2).WK3		05460000
.*				05470000
*!637E	WKUNIT/WKPACK	ERROR -- PACKS &PWSRC AND &PWWK2 BOTH ON UNIT &C1		05480000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	05490000
	AGO	.WKEND		05500000
.*				05510000
.WK3	ANOP		. CHECK \$WORK/\$WORK2	05520000
&C1	SETC	'&UWWRK'		05530000
.*				05540000
	AIF	(&UWWRK NE &UWWRK2).WKEND		05550000
	AIF	(&PWWRK EQ &PWWK2).WKEND		05560000
.*				05570000
*!637E	WKUNIT/WKPACK	ERROR -- PACKS &PWWRK AND &PWWK2 BOTH ON UNIT &C1		05580000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	05590000
.*				05600000
.WKEND	ANOP		. END OPERAND 'WKPACK'	05610000
.*				05620000
.*	-----	DIUNIT-R1/F1/R2/F2	-----*	05630000
.*				05640000
	AIF	(&DIUNIT NE '??').DIYES	. SKIP IF DIUNIT SPECIFIED	05650000
.*				05660000
*!650E	MISSING	'DIUNIT' OPERAND -- MUST BE SPECIFIED		05670000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	05680000
	AGO	.DIEND		05690000
.*				05700000
.DIYES	ANOP		. DIUNIT SPECIFIED	05710000
	AIF	(&DIUNIT NE '**').DIOK	. SKIP IF VALID PARAMETER	05720000
.*				05730000
*!655E	INVALID	'DIUNIT' PARAMETER -- MUST BE R1/F1/R2/F2		05740000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	05750000
	AGO	.DIEND		05760000
.*				05770000
.DIOK	ANOP		. VALID PARAMETER	05780000
&D	SETC	'&DIUNIT'		05790000
.*				05800000
.*	-----	CHECK CONFLICTS OF DIUNIT WITH CCUNIT AND PPUNITS	-----*	05810000
.*				05820000
	AIF	(&CCUNIT NE &DIUNIT).DI2	. SKIP IF NO CCUNIT CONFLICT	05830000
.*				05840000
*!660E	'CCUNIT'	SAME AS 'DIUNIT' -- NOT PERMITTED		05850000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	05860000
.*				05870000
.DI2	ANOP			05880000
	AIF	(&DIUNIT EQ &UPCBL).DIUPX	. IF MISMATCH WITH COBOL UNIT	05890000
	AIF	(&DIUNIT EQ &UPRPG).DIUPX	. IF MISMATCH WITH RPG UNIT	05900000
	AIF	(&DIUNIT EQ &UPFOR).DIUPX	. IF MISMATCH WITH FORTRAN UNIT	05910000
	AIF	(&DIUNIT NE &UPASM).DIEND	. UNLESS MISMATCH WITH ASSEM	05920000
.*				05930000

.DIUPX ANOP		05940000
*!665E 'PPUNIT' FROM \$EPLG STATEMENT SAME AS 'DIUNIT' -- NOT PERMITTED		05950000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH	05960000
.*		05970000
.DIEND ANOP	. END OPERAND 'DIUNIT'	05980000
.*		05990000
.*----- MINRES-YES/Y/NO/N (YES=1, NO=0) -----*		06000000
.*		06010000
AIF (&MINRES NE '*').MROK	. SKIP IF VALID PARAMETER	06020000
.*		06030000
*!670E INVALID 'MINRES' PARAMETER -- MUST BE YES/Y/NO/N		06040000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH	06050000
AGO .MREND		06060000
.*		06070000
.MROK ANOP	. VALID PARAMETER	06080000
AIF (&MINRES NE '1').MREND	. SKIP IF MINRES-NO	06090000
.*		06100000
&MIN SETB 1	. MINRES-YES	06110000
.MREND ANOP	. END MINRES	06120000
.*		06130000
.*----- CARD-YES/Y/NO/N (YES=1, NO=0) -----*		06140000
.*		06150000
AIF (&CARD NE '*').CLOK	. SKIP IF VALID PARAMETER	06160000
.*		06170000
*!680E INVALID 'CARD' PARAMETER -- MUST BE YES/Y/NO/N		06180000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH	06190000
AGO .CLEND		06200000
.*		06210000
.CLOK ANOP	. VALID PARAMETER	06220000
AIF (&CARD EQ '0').CLEND	. SKIP IF CARD-NO	06230000
.*		06240000
&CL SETB 1	. CARD-YES	06250000
.CLEND ANOP	. END 'CARD'	06260000
.*		06270000
.*----- CHECK CONFLICTS IN SUPPORT REQUESTED -----*		06280000
.*		06290000
AIF (&MLA NE '0').XX1	. SKIP IF MLTA SPECIFIED	06300000
AIF (&BSC NE '0').XX1	. SKIP IF BSCA SPECIFIED	06310000
.*		06320000
*!700E NO MLTA OR BSCA SUPPORT SPECIFIED -- AT LEAST ONE REQUIRED		06330000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH	06340000
.*		06350000
.XX1 ANOP	. CHECK FOR DFF W/O GETMSG	06360000
AIF (&FDFF NE '1').XX3	. SKIP IF NO DFF REQUESTED	06370000
AIF ('&BFA'(1,1) EQ '1').XX2	. SKIP IF FORMAT *AND* GETMSG	06380000
.*		06390000
*!705E FORMAT-YES SPECIFIED IN \$EFAC REQUIRES GETMSG-YES IN \$EBSC		06400000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH	06410000
.*		06420000
.XX2 ANOP	. CHECK DFF AGAINST 3270 DEVICE	06430000
AIF (&BD1 GT '00000000').XX3	. SKIP UNLESS DFF, NO 3270	06440000
.*		06450000
*!715E FORMAT-YES IN \$EFAC REQUIRES 3270 DISPLAY DEVICE IN \$EBSD		06460000
&TERR SETB 1	. SET TERMINATION ERROR SWITCH	06470000
.*		06480000
.XX3 ANOP	. CHECK 'PRUF' SUPPORT.	06490000
AIF (&FRUF NE '1').XX33	. SKIP IF NO PRUF REQUESTED	06500000
AIF ('&BFA'(1,1) EQ '1').XX22	. SKIP IF PRUF *AND* GETMSG	06510000
.*		06520000
*!706E PRUF-YES SPECIFIED IN \$EFAC REQUIRES GETMSG-YES IN \$EBSC		06530000

```

&TERR SETB 1 . SET TERMINATION ERROR SWITCH 06540000
.* 06550000
.XX22 ANOP . CHECK PRUF AGAINST 3270 DEVICE 06560000
AIF (&BD1 GT '00000000').XX33 . SKIP UNLESS DFF, NO 3270 06570000
.* 06580000
*!716E PRUF-YES IN $EFAC REQUIRES 3270 DISPLAY DEVICE IN $EBS 06590000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 06600000
.* 06610000
.XX33 ANOP . CHECK AT LEAST ONE PGM LANG 06620000
AIF (&PLG NE '0000').XX4 . SKIP IF AT LEAST 1 06630000
.* 06640000
*!710E NO PROGRAMMING LANGUAGE SUPPORTED -- AT LEAST ONE REQUIRED 06650000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 06660000
.XX4 ANOP 06670000
.* 06680000
.*----- SET STATEMENT SEQUENCE AND EXIT IF ERRORS -----* 06690000
.* 06700000
&SEQ SETA 10 . INDICATE $EGEN PROCESSED 06710000
AIF (&TERR NE '1').NOERR . SKIP IF *NO* ERRORS 06720000
.* 06730000
*? *** THIS RECORD INDICATES TO $CC1PP THAT $EGEN WAS PROCESSED *** 06740000
MEXIT . EXIT NOW IF *ERRORS* 06750000
.* 06760000
.NOERR ANOP . NO ERRORS -- PROCEED 06770000
.* 06780000
.*----- HOUSEKEEP FOR UNSPECIFIED INFORMATION -----* 06790000
.* 06800000
&# SETC ' ' . SINGLE BLANK 06810000
AIF (&BD1 NE ' ').HK010 . SKIP IF $EBS 06820000
.* 06830000
&BD1 SETC '00000000' . SET BSCA DEVICES BYTE 1 06840000
&BD2 SETC '00000000' . SET BSCA DEVICES BYTE 2 06850000
&BFA SETC '00000000' . SET BSCA FACILITIES 06860000
&BLT SETC '0000' . SET BSCA LINE TYPES 06870000
&INPOL SETB 0 . SET NO INTERVAL POLLING 06880000
.* 06890000
.HK010 ANOP 06900000
AIF (&RSB NE ' ').HK015 . SKIP IF RECSEP BYTE KNOWN 06910000
.* 06920000
&RSB SETC '1E' . SET DEFAULT RECSEP BYTE 06930000
.* 06940000
.HK015 ANOP 06950000
AIF (&MD1 NE ' ').HK020 . SKIP IF $EMLD SPECIFIED 06960000
.* 06970000
&MD1 SETC '00000000' . SET MLTA DEVICES BYTE 1 06980000
&MD2 SETC '00000000' . SET MLTA DEVICES BYTE 2 06990000
&MD3 SETC '0000' . SET MLTA DEVICES BYTE 3 07000000
&MXC SETC '0000' . SET MLTA TRANSMISSION CODES 07010000
.* 07020000
.HK020 ANOP . END HOUSEKEEPING 07030000
.* 07040000
.*----- DETERMINE THE SIZE REQUIRED FOR $CCPFILE -----* 07050000
.* 07060000
&ASIZE SETA 1 . SECTORS OF 1 SIT 07070000
&ASIZE SETA &ASIZE+(5*&NT+255)/256 . ADD SECTORS OF 1 TAT 07080000
&ASIZE SETA &ASIZE+(40*(&MLA+&BSC)+255)/256 . ADD SECTORS OF 1 LCT 07090000
&ASIZE SETA &ASIZE+(18*&NT+255)/256 . ADD SECTORS OF 1 TUT 07100000
&ASIZE SETA &ASIZE+(22+11*&NT+255)/256 . ADD SECTORS OF 1 TNT 07110000
.* 07120000
AIF (&MFSW).AZSTT . SKIP IF SWITCHED LINES 07130000

```

```

      AIF ('&BLT'(4,1) NE '1').AZNSW . SKIP IF *NO* SWITCHED LINES 07140000
.AZSTT ANOP . PERFORM IF SWITCHED LINES 07150000
&ASIZE SETA &ASIZE+(25*&NT+255)/256 . ADD SECTORS OF 1 STT 07160000
.* 07170000
.AZNSW ANOP . CONTINUE HERE 07180000
&ASIZE SETA &ASIZE+(20*&NDF+255)/256 . ADD SECTORS OF 1 FCT 07190000
&ASIZE SETA &ASIZE+(&NPM+4)/5 . ADD SECTORS OF 1 PCT 07200000
&ASIZE SETA (2+&NS*&ASIZE+23)/24 . HEADER+ASSIGNMENT **TRACKS** 07210000
&ASIZE SETA &ASIZE+(&DMP*&COR)/24 . ADD TRACKS OF DUMP SPACE. 07220000
.* 07230000
.AZCOM ANOP . CONTINUE HERE 07240000
.AZEND ANOP . END DETERMINATION OF $CCPFILE 07250000
.* 07260000
.*----- ASSURE THAT $CCPFILE WILL FIT IF TRKLOC SPECIFIED -----* 07270000
.* 07280000
&N SETA &TFIL+&ASIZE-1 . HIGHEST TRACK USED 07290000
AIF (&N LE '405').TRKOK . SKIP IF NOT BEYOND END 07300000
.* 07310000
&TFIL SETA 406-&ASIZE . SET NEW TRACK 07320000
*!720W TRACK LOCATION OF $CCPFILE CHANGED TO &TFIL BECAUSE OF FILE SIZE 07330000
.* 07340000
.TRKOK ANOP . END CHECK OF $CCPFILE TRKLOC 07350000
.* 07360000
.*----- OUTPUT THE $CCPFILE ESTABLISHMENT GROUP -----* 07370000
.* 07380000
      AIF (&CL EQ '1').CL10 . SKIP IF CARD GENERATION. 07390000
*P// COPY NAME-$CC1FC,LIBRARY-P 07400000
.CL10 ANOP . CONTINUE HERE. 07410000
.* 07420000
* --- PREPARE THE INITIAL CONTENTS OF $CCPFILE --- 07430000
* 07440000
*** FIRST, USE THE SCP GENERATOR TO CONVERT THE CONTENTS OF THE 07450000
*** CCP CONFIGURATION RECORD AND THE ASSIGNMENT DIRECTORY TO ENCODED 07460000
*** FORM -- SAVE THIS ENCODED INFORMATION AS A RELOCATABLE MODULE 07470000
*** NAMED $CC1FC -- THE INFORMATION WILL LATER BE WRITTEN TO THE 07480000
*** FIRST TWO SECTORS OF $CCPFILE AS THE CCP CONFIGURATION RECORD 07490000
*** AND THE ASSIGNMENT SET DIRECTORY 07500000
* 07510000
*P// NOHALT 07520000
*P// LOAD $CGDRV,&DIUNIT 07530000
*P// SWITCH 0XXXXXXX 07540000
*P// FILE NAME-$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-10,RETAIN-S 07550000
*P// FILE NAME-$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-5,RETAIN-S 07560000
*P// FILE NAME-$WORK2,UNIT-&UWVK2,PACK-&PWVK2,TRACKS-5,RETAIN-S 07570000
.* 07580000
      AIF (&CL EQ '1').CL20 . SKIP IF CARD GENERATION. 07590000
*P// COMPILE UNIT-&DIUNIT,SOURCE-$CC1FC 07600000
*P// RUN 07610000
*P// CEND 07620000
*P// COPY NAME-$CC1FC,LIBRARY-S 07630000
      AGO .CL30 07640000
.CL20 ANOP . CONTINUE HERE. 07650000
*P// RUN 07660000
.CL30 ANOP . CONTINUE HERE. 07670000
.* 07680000
$CC1FC TITLE 'CCP&#.CONFIGURATION&#.RECORD&#.AND&#.DIRECTORY' 07690000
$CC1FC START 0 MODULE NAME 07700000
      EXTRN $CC1BF REFERENCE TO MODULE WHICH INITIS 07710000
      EJECT 07720000
***** 07730000

```

```

*          C C P   C O N F I G U R A T I O N   R E C O R D          * 07740000
*****
SPACE 2 07750000
* THIS SET OF 256 BYTES -- A SECTOR ON DISK -- INDICATES TO CCP THOSE 07760000
* DEVICES AND SYSTEM FACILITIES YOU SPECIFIED DURING CCP GENERATION 07770000
SPACE 1 07780000
CFGVRF DC   XL2'EBD7'          TWO BYTES *ALWAYS* HEX 'EBD7' 07790000
SPACE 1 07800000
CFGRV1 DC   6XL1'00'          *RESERVED* 07810000
SPACE 1 07820000
CFGFA2 DC   BL1'0&FPGC.0&CPW&UPW&FDME&FDFE&FRUF'  FACILITIES--BYTE 2 07830000
CFGRV2 DC   2XL1'00'          *RESERVED* 07840000
CFGPLG DC   BL1'&PLG.0000'     PROGRAMMING LANGUAGES SUPPORTED 07850000
SPACE 1 07860000
CFGCBO DC   XL2'00C7'         CONSOLE INTERRUPT BRANCH OFFSET 07870000
AIF      (&FDME).CFDME        . SKIP IF DME 07880000
CFGESC DC   CL6'&#'           *NO* DATA MODE ESCAPE STRING 07890000
AGO      .CFCM1               . SKIP TO COMMON CODE 07900000
.CFDME ANOP                    . IF DME 07910000
CFGESC DC   &X1DME&X2DME'     . DATA MODE ESCAPE CHAR STRING 07920000
.CFCM1 ANOP                    07930000
CFGLUS DC   IL2'&LUS'         LENGTH OF USER SECURITY DATA 07940000
CFGRV3 DC   4XL1'00'          *RESERVED* 07950000
SPACE 1 07960000
CFGNS  DC   IL1'&NS'          MAX NUMBER SETS 07970000
CFGNPM DC   IL1'&NPM'         MAX NUMBER PROGRAMS 07980000
CFGNDF DC   IL1'&NDF'         MAX NUMBER DISK FILES 07990000
CFGNT  DC   IL1'&NT'          MAX NUMBER TERMINALS 08000000
CFGDMP DC   IL1'&DMP'         MAX NUMBER CORE DUMPS 08010000
CFGSC  DC   IL2'&COR'         MEMORY # SECTORS (192-512) 08020000
CFGRV4 DC   3XL1'00'          *RESERVED* 08030000
SPACE 1 08040000
CFGTP  DC   AL1(16*&MLA+&BSC)  NUMBER COMMUNICATIONS LINES 08050000
SPACE 1 08060000
CFGMD1 DC   BL1'&MD1'         MLTA DEVICES--BYTE 1 08070000
CFGMD2 DC   BL1'&MD2'         MLTA DEVICES--BYTE 2 08080000
CFGMD3 DC   BL1'&MD3.0000'     MLTA DEVICES--BYTE 3 08090000
CFGRV5 DC   2XL1'00'          *RESERVED* 08100000
CFGMXC DC   BL1'&MXC.000&MNOX' MLTA TRANSMISSION CODES 08110000
CFGRV6 DC   2XL1'00'          *RESERVED* 08120000
SPACE 1 08130000
CFGBLT DC   BL1'&BLT.0000'     BSC LINE TYPES SUPPORTED 08140000
CFGBFA DC   BL1'&BFA'         BSC FACILITIES 08150000
CFGBF1 DC   BL1'&INPOL.0000000' BSC FACILITIES CONTINUED 08160000
CFGRV7 DC   1XL1'00'          *RESERVED* 08170000
CFGRSB DC   XL1'&RSB'         RECORD SEPARATOR BYTE 08180000
CFGBD1 DC   BL1'&BD1'         BSCA DEVICES--BYTE 1 08190000
CFGBD2 DC   BL1'&BD2'         BSCA DEVICES--BYTE 2 08200000
SPACE 1 08210000
DC      204XL1'FF'           FILLED WITH HEX 'FF' 08220000
EJECT 08230000
***** 08240000
*          $ C C P F I L E   D I R E C T O R Y          * 08250000
*****
SPACE 2 08260000
* THIS SET OF 256 BYTES -- A SECOND SECTOR ON DISK -- SERVES AS A 08270000
* DIRECTORY TO THE CONTENTS OF THIS FILE FOR THE CCP ASSIGNMENT AND 08280000
* STARTUP PROGRAMS -- EACH FIELD IS INITIALLY ZERO 08290000
SPACE 1 08300000
DIRNSA DC   IL1'0'           CURRENT NUMBER ASSIGNMENT SETS 08310000
08320000
08330000

```

```

DIRRS1 DC    IL2'0'          * RESERVED *          08340000
DIRDAT DC    XL3'000000'    DATE LAST MODIFIED 08350000
DIRDID DC    XL1'00'       EXECUTION DEFAULT SET 08360000
DIRNFX DC    XL1'00'       *** UNUSED ***      08370000
DIRRS2 DC    IL2'0'          * RESERVED *          08380000
DIRRS3 DC    IL1'0'          * RESERVED *          08390000
DIRBD@ DC    XL2'0000'     LOCATION BEGIN DUMP AREA 08400000
DIRED@ DC    XL2'0000'     LOCATION END DUMP AREA  08410000
DIRRS5 DC    XL2'0000'     * RESERVED *          08420000
SPACE 1                                           08430000
* ASSIGNMENT SET DIRECTORY ENTRIES START HERE    08440000
SPACE 1                                           08450000
DC 239XL1'FF'          INITIALLY UNUSED PORTION 08460000
SPACE 5                                           08470000
END $CC1FC                                          08480000
.*
AIF (&CL EQ '1').CL40 . SKIP IF CARD GENERATION. 08500000
*P// CEND                                          08510000
*P// COPY NAME-$CC1LC,LIBRARY-P                   08520000
AGO .CL50                                          08530000
.CL40 ANOP . CONTINUE HERE.                       08540000
*P/*
.CL50 ANOP . CONTINUE HERE.                       08550000
.*
*** USE THE OVERLAY LINKAGE EDITOR TO JOIN THE ABOVE MODULE WITH 08580000
*** ANOTHER MODULE CAPABLE OF WRITING THE INFORMATION TO $CCPFILE -- 08590000
*** SAVE THE RESULTING LOAD MODULE $CC1BF ON THE DISTRIBUTION PACK 08600000
*** FOR LATER INITIALIZATION OF $CCPFILE          08610000
*
*P// LOAD $OLINK,&DIUNIT                            08630000
*P// FILE NAME-$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-10,RETAIN-S 08640000
*P// FILE NAME-$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-10,RETAIN-S 08650000
*P// RUN                                           08660000
*P// PHASE NAME-$CC1BF,UNIT-&DIUNIT,RETAIN-R,LINKADD-X'4000' 08670000
*P// OPTIONS ENTRY-EXECBF,LEVEL-&LVL              08680000
*P// INCLUDE NAME-'$CC1FC,$CC1BF',UNIT-&DIUNIT    08690000
*P// END                                           08700000
.*
AIF (&CL EQ '1').CL70 . SKIP IF CARD GENERATION. 08720000
*P// CEND                                          08730000
*P// COPY NAME-$CC1#1,LIBRARY-P                   08740000
.CL70 ANOP . CONTINUE HERE.                       08750000
.*
.*----- BUILD THE STRING THAT DETERMINES TRANSLATE SUPPORT -----* 08770000
.*
&C1 SETC '' . MAKE INITIALLY NULL                 08790000
&C2 SETC '' . MAKE INITIALLY NULL                 08800000
.*
.* CREATE THE FIRST MXL STRING                    08820000
.*
&C3 SETC '&MXC'(3,1) . '1'=2740/1 PTTBCD / '0'=NO 08840000
&C1 SETC '&C1&C3' .                               08850000
&C1 SETC '&C1&XM4E' .                             08860000
&C3 SETC '&MXC'(1,1) . '1'=CORR / '0'=NO         08870000
&C1 SETC '&C1&C3' .                               08880000
&C1 SETC '&C1&MT50' .                             08890000
&C1 SETC '&C1&XM4E' .                             08900000
&C3 SETC '&MXC'(1,1) . '1'=CORR / '0'=NO         08910000
&C1 SETC '&C1&C3' .                               08920000
&C3 SETC '&MXC'(3,1) . '1'=2740/1 PTTBCD / '0'=NO 08930000

```

&C1	SETC	'&C1&C3'		08940000
&C1	SETC	'&C1&MT50'		08950000
.	*			08960000
.	*	CREATE THE SECOND MXL STRING		08970000
.	*			08980000
&C3	SETC	'&MXC'(3,1)	. '1'=2740/1 PTTBCD / '0'=NO	08990000
&C2	SETC	'&C2&C3'		09000000
&C2	SETC	'&C2&XM4E'		09010000
&C3	SETC	'&MXC'(1,1)	. '1'=CORR / '0'=NO	09020000
&C2	SETC	'&C2&C3'		09030000
&C2	SETC	'&C2&MT50'		09040000
	AIF	('&BFA'(8,1) NE '1').MXL00	. SKIP IF *NO* BSCA ASCII	09050000
.	*			09060000
&C2	SETC	'&C2.11'	. INCLUDE ASCII SUPPORT	09070000
	AGO	.MXL99		09080000
.	*			09090000
.MXL00	ANOP		. *NO* ASCII SUPPORT	09100000
&C2	SETC	'&C2.00'	. EXCLUDE ASCII SUPPORT	09110000
.	*			09120000
.MXL99	ANOP		. END BUILD TRANSLATE LIST	09130000
.	*			09140000
.	*	----- OUTPUT THE RESIDENT MODULE CREATION GROUP -----*		09150000
.	*			09160000
*??	***	BEGIN NEW PAGE IN OUTPUT LISTING ***		09170000
*	---	CREATE THE EXECUTION STAGE RESIDENT MODULES ---		09180000
*				09190000
***	USE THE MACRO PROCESSOR TO EXPAND SOURCE CODE OF THE FIRST			09200000
***	OF THE RESIDENT MODULES - \$CC4#1			09210000
*				09220000
*P//	LOAD	\$MPXDV,&DIUNIT		09230000
*P//	FILE	NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-130,RETAIN-T		09240000
.	*			09250000
	AIF	(&CL EQ '1').CL80	. SKIP IF CARD GENERATION.	09260000
*P//	COMPILE	UNIT-&DIUNIT,SOURCE-\$CC1#1		09270000
*P//	RUN			09280000
*P//	CEND			09290000
*P//	COPY	NAME-\$CC1#1,LIBRARY-S		09300000
	AGO	.CL85		09310000
.CL80	ANOP		. CONTINUE HERE.	09320000
*P//	RUN			09330000
.CL85	ANOP		. CONTINUE HERE.	09340000
.	*			09350000
\$CC4#1	TITLE	'BEGIN&#.CCP&#.SYSTEM&#.CONTROL&#.MODULE'		09360000
\$CC4#1	START	X'4400'	CCP SYSTEM CONTROL MODULE	09370000
	\$E000	#M-&MLA,#B-&BSC,DF-&FDFD,DE-&FDME,MS-&MIN,RF-&FRUF,RS-&FAI		09380000
	\$E001	PC-&FPGC,SO-&CPW&UPW		09390000
	\$E002	ML-&MFSC&MFSW,MT-&MT50&MT41&MFBR,MF-&MNOX,MS-&FCPUM		09400000
	\$E003	BL-&BLT,BF-&BFA,BT-&BD1&BD2,INP-&INPOL,BIA-&BIDA,BY-&FPBSY		09410000
	TITLE	'EQUATES&#.--&#.COMMON&#.VALUES'		09420000
	\$EEQU			09430000
	TITLE	'EQUATES&#.--&#.CCP&#.COMMUNICATIONS&#.AREA'		09440000
	\$ECOM	ID-0,DF-&FDFD		09450000
	TITLE	'EQUATES&#.--&#.TASK&#.CONTROL&#.BLOCK'		09460000
	\$ETCB			09470000
	TITLE	'EQUATES&#.--&#.FOR&#.PROGRAM&#.APPENDED&#.STORAGE'		09480000
	\$EPAS			09490000
	TITLE	'EQUATES&#.--&#.FOR&#.SYSLOG'		09500000
	\$LOGD			09510000
	TITLE	'EQUATES&#.--&#.FOR&#.SYSTEM&#.COMM&#.AREAS'		09520000
	\$EDSM	RB-Y,SP-Y		09530000

TITLE	'EQUATES&#.--&#.TASK&#.COMPLETION&#.CODES'	09540000
\$ETCC	DF-&FDF	09550000
TITLE	'EQUATES&#.--&#.COMMUNICATIONS&#.PARAMETER&#.LIST'	09560000
\$ECPL		09570000
TITLE	'EQUATES&#.--&#.TERMINAL&#.ATTRIBUTES&#.SET'	09580000
\$ETAS		09590000
TITLE	'EQUATES&#.--&#.TERMINAL&#.UNIT&#.BLOCK'	09600000
\$ETUB		09610000
TITLE	'EQUATES&#.--&#.LINE&#.CONTROL&#.BLOCK'	09620000
\$ELCB		09630000
.*		09640000
AIF	(&BSC EQ '0').GEN10 . SKIP IF *NO* BSCA SUPPORT	09650000
.*		09660000
TITLE	'EQUATES&#.--&#.BSCA&#.DTF'	09670000
\$DFOB		09680000
.*		09690000
AIF	(&FDF NE '1').XDFF . SKIP IF NO DFF.	09700000
.*		09710000
TITLE	'EQUATES&#.--&#.DFF&#.--&#.TERMINATION&#.CODES'	09720000
\$EDFT		09730000
TITLE	'EQUATES&#.--&#.FORMAT&#.TABLES'	09740000
\$EFT		09750000
TITLE	'EQUATES&#.--&#.TERMINAL&#.TABLE'	09760000
\$ETT		09770000
TITLE	'EQUATES&#.--&#.DFF&#.FDT'	09780000
\$EFD		09790000
TITLE	'EQUATES&#.--&#.DISK&#.IOB'	09800000
\$EIOB		09810000
.*		09820000
.XDFF	ANOP . CONTINUE HERE.	09830000
AIF	(&MLA EQ '0').GEN10 SKIP IF NO MLTA.	09840000
.*		09850000
\$DTOM	AT1-Y,AT2-Y,CMP-Y,OPC-Y,OSC-Y,SNS-Y,TFT-Y	09860000
.*		09870000
.GEN10	ANOP . CONTINUE HERE	09880000
TITLE	'EQUATES&#.--&#.TERMINAL&#.NAME&#.TABLE'	09890000
\$ETNT		09900000
TITLE	'EQUATES&#.--&#.TRANSLATE/MOVE&#.LIST'	09910000
\$ETML		09920000
TITLE	'CCP&#.COMMON&#.AREA'	09930000
\$E030		09940000
\$E038	MXL-&C1&C2,CON-1	09950000
\$E060		09960000
.*		09970000
AIF	(&FDF NE '1').XXDFF . SKIP IF NO DFF.	09980000
.*		09990000
\$E065		10000000
.XXDFF	ANOP	10010000
END	\$CC4#1	10020000
.*		10030000
AIF	(&CL EQ '1').CL90 . SKIP IF CARD GENERATION.	10040000
*P//	CEND	10050000
*P//	COPY NAME-\$CC1#A,LIBRARY-P	10060000
AGO	.CL100	10070000
.CL90	ANOP	10080000
P/		10090000
.CL100	ANOP . CONTINUE HERE.	10100000
***	USE THE SCP GENERATOR TO PROCESS THIS SOURCE CODE INTO A	10110000
***	RELOCATABLE MODULE	10120000
*		10130000

*P// LOAD \$CGDRV,&DIUNIT	10140000
*P// SWITCH 1XXXXXXX	10150000
*P// FILE NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,RETAIN-S	10160000
*P// FILE NAME-\$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-040,RETAIN-S	10170000
*P// FILE NAME-\$WORK2,UNIT-&UWVK2,PACK-&PWVK2,TRACKS-040,RETAIN-S	10180000
*P// RUN	10190000
.*	10200000
AIF (&CL EQ '1').CL105 . SKIP IF CARD GENERATION.	10210000
*P// CEND	10220000
*P// COPY NAME-\$CC1L1,LIBRARY-P	10230000
.CL105 ANOP . CONTINUE HERE.	10240000
*** USE THE OVERLAY LINKAGE EDITOR TO FORM THE FIRST RESIDENT MODULE	10250000
*** AS A LOAD MODULE NAMED \$CC4#1.	10260000
*	10270000
*P// LOAD \$SOLINK,&DIUNIT	10280000
*P// FILE NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-100,RETAIN-S	10290000
*P// FILE NAME-\$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-040,RETAIN-S	10300000
*P// RUN	10310000
*P// OPTIONS ENTRY-\$CC4II,MAP-XREF,LEVEL-&LVL	10320000
*P// PHASE NAME-\$CC4#1,UNIT-&CCUNIT,RETAIN-R,LINKADD-X'4400',RLD-NO	10330000
*P// CATEGORY NAME-\$CC4V1,VALUE-100	10340000
*P// INCLUDE NAME-'\$CC4#1,\$CC4PI,\$CC4MS,\$CC4V1',UNIT-&DIUNIT	10350000
.*	10360000
AIF (&FDFE NE '1').NODFF . SKIP IF NO DFF.	10370000
.*	10380000
*P// INCLUDE NAME-\$CC4MX,UNIT-&DIUNIT	10390000
AGO .GEN11 SKIP NO DFF MODULE.	10400000
.*	10410000
.NODFF ANOP . CONTINUE HERE IF NO DFF.	10420000
*P// INCLUDE NAME-\$CC4MV,UNIT-&DIUNIT	10430000
.*	10440000
.GEN11 ANOP . CONTINUE HERE.	10450000
*P// INCLUDE NAME-'\$CC4TI,\$CC4CP,\$CC4OC,\$CC4NQ,\$CC4DQ',UNIT-&DIUNIT	10460000
*P// END	10470000
.*	10480000
AIF (&CL EQ '1').CL120 . SKIP IF CARD GENERATION.	10490000
*P// CEND	10500000
*P// COPY NAME-\$CC1VT,LIBRARY-P	10510000
.CL120 ANOP . CONTINUE HERE.	10520000
.*	10530000
*** USE THE MACRO PROCESSOR TO EXPAND SOURCE CODE OF THE ADDRESS	10540000
*** VECTOR TABLE - \$CC4VT	10550000
*	10560000
*P// LOAD \$MPXDV,&DIUNIT	10570000
*P// SWITCH 1XXXXXXX	10580000
*P// FILE NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-50,RETAIN-T	10590000
.*	10600000
AIF (&CL EQ '1').CL125 . SKIP IF CARD GENERATION.	10610000
*P// COMPILE UNIT-&DIUNIT,SOURCE-\$CC1VT	10620000
*P// RUN	10630000
*P// CEND	10640000
*P// COPY NAME-\$CC1VT,LIBRARY-S	10650000
AGO .CL130	10660000
.CL125 ANOP . CONTINUE HERE.	10670000
*P// RUN	10680000
.CL130 ANOP . CONTINUE HERE.	10690000
.*	10700000
\$CC4VT TITLE 'ADDRESS&#.--&#.VECTOR&#.--&#.TABLE'	10710000
\$CC4VT START 0	10720000
\$E000 #M-&MLA,#B-&BSC,DF-&FDFE,DE-&FDME,MS-&MIN	10730000

\$E002 ML-&MFSC&MFSW,MT-&MT50&MT41&MFBR,MF-&MNOX,MS-&FCPUM	10740000
\$E003 BL-&BLT,BF-&BFA,BT-&BD1&BD2,INP-&INPOL,BIA-&BIDA,BY-&FPBSY	10750000
\$E033 MD1-&MD1,MD2-&MD2,MD3-&MD3,BF-&BFA	10760000
END \$CC4VT	10770000
.*	10780000
AIF (&CL EQ '1').CL150 . SKIP IF CARD GENERATION.	10790000
*P// CEND	10800000
*P// COPY NAME-\$CC1VS,LIBRARY-P	10810000
AGO .CL155	10820000
.CL150 ANOP . CONTINUE HERE.	10830000
.*	10840000
P/	10850000
.CL155 ANOP	10860000
.*	10870000
*** USE THE SCP GENERATOR TO PROCESS THIS SOURCE CODE INTO A	10880000
*** RELOCATABLE MODULE - \$CC4VT	10890000
*	10900000
*P// LOAD \$CGDRV,&DIUNIT	10910000
*P// SWITCH 1XXXXXXX	10920000
*P// FILE NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,RETAIN-S	10930000
*P// FILE NAME-\$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-040,RETAIN-S	10940000
*P// FILE NAME-\$WORK2,UNIT-&UWVK2,PACK-&PWVK2,TRACKS-040,RETAIN-S	10950000
*P// RUN	10960000
.*	10970000
AIF (&CL EQ '1').CL180 . SKIP IF CARD GENERATION.	10980000
*P// CEND	10990000
*P// COPY NAME-\$CC1#2,LIBRARY-P	11000000
.CL180 ANOP . CONTINUE HERE.	11010000
.*	11020000
*** USE THE MACRO PROCESSOR TO EXPAND SOURCE CODE OF THE SECOND	11030000
*** RESIDENT MODULE - \$CC4#2	11040000
*	11050000
*P// LOAD \$MPXDV,&DIUNIT	11060000
*P// SWITCH 1XXXXXXX	11070000
*P// FILE NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-180,RETAIN-T	11080000
.*	11090000
AIF (&CL EQ '1').CL200 . SKIP IF CARD GENERATION.	11100000
*P// COMPILE UNIT-&DIUNIT,SOURCE-\$CC1#2	11110000
*P// RUN	11120000
*P// CEND	11130000
*P// COPY NAME-\$CC1#2,LIBRARY-S	11140000
AGO .CL210	11150000
.CL200 ANOP . CONTINUE HERE.	11160000
*P// RUN	11170000
.CL210 ANOP . CONTINUE HERE.	11180000
.*	11190000
\$CC4#2 TITLE 'BEGIN&#.CCP&#.COMMUNICATIONS&#.CONTROL&#.MODULE'	11200000
\$CC4#2 START 0 CCP COMMUNICATIONS CNTRL MODULE	11210000
\$E000 #M-&MLA,#B-&BSC,DF-&FDFE,DE-&FDME,MS-&MIN,RF-&FRUF,RS-&FAI	11220000
\$E002 ML-&MFSC&MFSW,MT-&MT50&MT41&MFBR,MF-&MNOX,MS-&FCPUM	11230000
\$E003 BL-&BLT,BF-&BFA,BT-&BD1&BD2,INP-&INPOL,BIA-&BIDA,BY-&FPBSY	11240000
TITLE 'EQUATES&#.--&#.COMMON&#.VALUES'	11250000
\$EEQU	11260000
TITLE 'EQUATES&#.--&#.CCP&#.COMMUNICATIONS&#.AREA'	11270000
\$ECOM ID-0,DF-&FDFE,CP-Y,AM-Y,TM-Y,CM-Y	11280000
TITLE 'EQUATES&#.--&#.TASK&#.CONTROL&#.BLOCK'	11290000
\$ETCB	11300000
TITLE 'EQUATES&#.--&#.DSM&#.TCB'	11310000
\$EDSM NP-Y,NC-Y,SP-N	11320000
TITLE 'EQUATES&#.--&#.COMMUNICATIONS&#.PARAMETER&#.LIST'	11330000

\$ECPL	RTNCD-EXCP		11340000
TITLE	'EQUATES&#.--&#.TERMINAL&#.ATTRIBUTES&#.SET'		11350000
\$ETAS			11360000
TITLE	'EQUATES&#.--&#.TERMINAL&#.UNIT&#.BLOCK'		11370000
\$ETUB			11380000
TITLE	'EQUATES&#.--&#.LINE&#.CONTROL&#.BLOCK'		11390000
\$ELCB	MLTA-&MLA,BSCA-&BSC		11400000
TITLE	'EQUATES&#.--&#.TRANSLATE/MOVE&#.LIST'		11410000
\$ETML			11420000
.*			11430000
AIF	(&BSC EQ '0').GEN20	. SKIP IF *NO* BSCA SUPPORT	11440000
.*			11450000
TITLE	'EQUATES&#.--&#.BSCA&#.COMMON&#.VALUES'		11460000
\$EBEQ	IOB-Y,WKA-Y,POL-Y,CKL-Y,CMD-CM,MIN-Y		11470000
TITLE	'EQUATES&#.--&#.BSCA&#.DTF'		11480000
\$DFOB			11490000
.*			11500000
.GEN20	ANOP	. CONTINUE HERE	11510000
AIF	(&MLA EQ '0').GEN15	. SKIP IF *NO* MLTA SUPPORT	11520000
.*			11530000
TITLE	'EQUATES&#.--&#.MLTA&#.DTF'		11540000
\$DTOM	AT1-Y,AT2-Y,CMP-Y,OPC-Y,OSC-Y,SNS-Y,TFT-Y		11550000
.*			11560000
.GEN15	ANOP	. CONTINUE HERE	11570000
\$E038	MXL-&C1&C2		11580000
\$E070			11590000
AIF	(&BSC EQ '0').GEN25	. SKIP IF *NO* BSCA SUPPORT	11600000
\$E072			11610000
\$E080			11620000
\$E085			11630000
.GEN25	ANOP	. CONTINUE HERE	11640000
AIF	(&MLA EQ '0').GEN26	. SKIP IF *NO* MLTA SUPPORT	11650000
\$E075			11660000
\$E082			11670000
\$E087			11680000
.GEN26	ANOP	. CONTINUE HERE	11690000
\$E090			11700000
\$E092			11710000
AIF	(&BSC EQ '0').GEN30	. SKIP IF *NO* BSCA SUPPORT	11720000
\$E093			11730000
.GEN30	ANOP	. CONTINUE HERE	11740000
AIF	(&MLA EQ '0').GEN40	. SKIP IF *NO* MLTA SUPPORT	11750000
\$E094			11760000
.GEN40	ANOP	. CONTINUE HERE	11770000
AIF	(&FDME NE '1').GEN95	. SKIP IF *NO* DATA MODE ESCAPE	11780000
\$E095	DMESTR-&X1DME&X2DME'		11790000
AGO	.GEN99		11800000
.GEN95	ANOP	. *NO* DATA MODE ESCAPE	11810000
\$E095			11820000
.GEN99	ANOP	. CONTINUE HERE	11830000
END	\$CC4#2		11840000
.*			11850000
AIF	(&CL EQ '1').CL220	. SKIP IF CARD GENERATION.	11860000
*P//	CEND		11870000
*P//	COPY NAME-\$CC1#B,LIBRARY-P		11880000
AGO	.CL230		11890000
.CL220	ANOP		11900000
P/			11910000
.CL230	ANOP	. CONTINUE HERE.	11920000
.*			11930000

```

*** USE THE SCP GENERATOR TO PROCESS THIS SOURCE CODE INTO A      11940000
*** RELOCATABLE MODULE - $CC4#2                                  11950000
*                                                                    11960000
*P// LOAD $CGDRV,&DIUNIT                                          11970000
*P// SWITCH 1XXXXXXX                                             11980000
*P// FILE NAME-$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,RETAIN-S          11990000
*P// FILE NAME-$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-050,RETAIN-S 12000000
*P// FILE NAME-$WORK2,UNIT-&UWVK2,PACK-&PWVK2,TRACKS-050,RETAIN-S 12010000
*P// RUN                                                         12020000
.*                                                                12030000
    AIF (&CL EQ '1').CL240 . SKIP IF CARD GENERATION.            12040000
*P// CEND                                                         12050000
*P// COPY NAME-$CC1LE,LIBRARY-P                                  12060000
.CL240 ANOP . CONTINUE HERE.                                     12070000
.*                                                                12080000
*P// LOAD $OLINK,&DIUNIT LINKEDIT $CC4#2                          12090000
*P// FILE NAME-$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-100,RETAIN-S 12100000
*P// FILE NAME-$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-040,RETAIN-S 12110000
*P// RUN                                                         12120000
*P// OPTIONS ENTRY-$CC4CM,MAP-XREF,LEVEL-&LVL                    12130000
*P// PHASE NAME-$CC4#2,UNIT-&CCUNIT,RETAIN-R,LINKADD-X'0000'    12140000
*P// CATEGORY NAME-$CC4V2,VALUE-100                              12150000
    AIF (&FDFE NE '1').NDFE . SKIP IF NO DFE                    12160000
.*                                                                12170000
*P// INCLUDE NAME-'$CC4VT,$CC4DF,$CC4#2,$CC4V2',UNIT-&DIUNIT    12180000
    AGO .NC01                                                     12190000
.*                                                                12200000
.NDFE ANOP . CONTINUE HERE.                                     12210000
*P// INCLUDE NAME-'$CC4VT,$CC4#2,$CC4V2',UNIT-&DIUNIT            12220000
.NC01 ANOP                                                       12230000
    AIF (&BSC EQ '0').GXNBM . SKIP IF *NO* BSCA SUPPORT          12240000
.*                                                                12250000
*P// INCLUDE NAME-'$CC4IB,$CC4BT',UNIT-&DIUNIT                    12260000
.*                                                                12270000
    AIF ('&BLT'(4,1) EQ '1').BSMS INCLUDE $$BSMS IF DIAL-YES     12280000
    AIF (&BD2 EQ '00000000').GXN32 SKIP IF 3270 ONLY.           12290000
.*                                                                12300000
.BSMS ANOP . INCLUDE MODULE $$BSMS                              12310000
*P// INCLUDE NAME-$$BSMS,UNIT-&DSUNIT                              12320000
    AGO .GXBS . SKIP OVER 3270 ONLY MODULE.                       12330000
.*                                                                12340000
.GXN32 ANOP . CONTINUE HERE.                                     12350000
    AIF (&MINRES EQ '1').C4M1 . SKIP IF MINRES-YES               12360000
    AIF ('&BFA'(4,1) NE '1').C4M1 . SKIP IF RESPOL-NO           12370000
.*                                                                12380000
*P// INCLUDE NAME-$CC4M2,UNIT-&DIUNIT                              12390000
    AGO .GXBS . SKIP IF RESPOL-YES                                12400000
.*                                                                12410000
.C4M1 ANOP . INCLUDE MODULE $CC4M1                               12420000
*P// INCLUDE NAME-$CC4M1,UNIT-&DIUNIT                              12430000
.*                                                                12440000
.GXBS ANOP . CONTINUE HERE                                       12450000
    AIF (&BSC LE '1').GXB1L . SKIP IF SINGLE BSCA ONLY          12460000
.*                                                                12470000
*P// INCLUDE NAME-$$BSAT,UNIT-&DSUNIT                              12480000
.*                                                                12490000
.GXB1L ANOP . CONTINUE HERE                                       12500000
    AIF ('&BLT'(3,1) NE '1').GXBNC . SKIP IF CS-NO              12510000
.*                                                                12520000
*P// INCLUDE NAME-$$BSLG,UNIT-&DSUNIT                              12530000

```

AIF ('&BFA'(4,1) NE '1').GXBNC . SKIP IF RESPOL-NO	12540000
.*	12550000
*P// INCLUDE NAME- '\$\$BSMA, \$\$BSMB, \$\$BSMC, \$\$BSMF', UNIT-&DSUNIT	12560000
.*	12570000
.GXBNC ANOP . CONTINUE HERE	12580000
AIF ('&BLT'(2,1) NE '1').GXNAF . SKIP IF MP-NO	12590000
AIF ('&BFA'(5,1) NE '1').GXNAF . SKIP IF MP-YES, AUTORS-NO	12600000
.*	12610000
*P// INCLUDE NAME- \$\$BSMD, UNIT-&DSUNIT	12620000
.*	12630000
.GXNAF ANOP	12640000
AIF (&BIDA EQ '0').GXNBM . SKIP IF NO IDA SUPPORT	12650000
.*	12660000
*P// INCLUDE NAME- \$\$BSID, UNIT-&DSUNIT	12670000
.*	12680000
.GXNBM ANOP	12690000
AIF (&MLA EQ '0').GXNOM . SKIP *NO* MLTA SUPPORT	12700000
.*	12710000
*P// INCLUDE NAME- \$CC4IM, UNIT-&DIUNIT	12720000
*P// INCLUDE NAME- '\$\$MLDI, \$\$MLRR', UNIT-&DSUNIT	12730000
.*	12740000
AIF (&MFSC NE '1').GXNSC . SKIP IF *NO* STATION CONTROL	12750000
AIF (&MFNK NE '1').GXN00 . SKIP IF *NO* NON-CHECKING TERM	12760000
AIF (&MFSW NE '1').GXN00 . SKIP IF *NO* SWITCHED SUPPORT	12770000
*P// INCLUDE NAME- '\$\$MLIA, \$\$MLIB, \$\$MLID, \$\$MLIF', UNIT-&DSUNIT	12780000
AIF (&MFSC NE '1').GXNSC . SKIP IF *NO* STATION CONTROL	12790000
AGO .GXNSW	12800000
.*	12810000
.GXN00 ANOP	12820000
*P// INCLUDE NAME- '\$\$MLIA, \$\$MLIB', UNIT-&DSUNIT	12830000
.GXNSC ANOP . CONTINUE HERE	12840000
AIF (&MFNK NE '1').GXNNK . SKIP IF *NO* NON-CHECKING TERM	12850000
AIF (&MFSW NE '1').NC05 . SKIP IF *NO* SWITCHED SUPPORT	12860000
*P// INCLUDE NAME- '\$\$MLID, \$\$MLIF', UNIT-&DSUNIT	12870000
AGO .GXNSW	12880000
.*	12890000
.NC05 ANOP	12900000
*P// INCLUDE NAME- \$\$MLID, UNIT-&DSUNIT	12910000
.*	12920000
.GXNNK ANOP . CONTINUE HERE	12930000
AIF (&MFSW NE '1').GXNSW . SKIP IF *NO* SWITCHED SUPPORT	12940000
.*	12950000
*P// INCLUDE NAME- \$\$MLIF, UNIT-&DSUNIT	12960000
.*	12970000
.GXNSW ANOP . CONTINUE HERE	12980000
AIF (&MFCK NE '1').GXNCK . SKIP IF *NO* CHECKING TERMINAL	12990000
AIF ('&MD1'(1,1) NE '1').NC10 . SKIP IF NO TYPE-1050	13000000
*P// INCLUDE NAME- '\$\$MLIG, \$\$MLTM', UNIT-&DSUNIT	13010000
AGO .MTV02	13020000
.*	13030000
.NC10 ANOP	13040000
*P// INCLUDE NAME- \$\$MLIG, UNIT-&DSUNIT	13050000
.*	13060000
.GXNCK ANOP . CONTINUE HERE	13070000
AIF ('&MD1'(1,1) NE '1').MTV02 . SKIP IF NO TYPE-1050	13080000
.*	13090000
*P// INCLUDE NAME- \$\$MLTM, UNIT-&DSUNIT	13100000
.*	13110000
.MTV02 ANOP . CONTINUE HERE	13120000
AIF ('&MD1'(2,1) NE '1').MTV03 . SKIP IF NO TYPE-1050D	13130000

```

.*
*P// INCLUDE NAME-$$MLTB,UNIT-&DSUNIT 13140000
.* 13150000
.* 13160000
.MTV03 ANOP . CONTINUE HERE 13170000
AIF ('&MD1'(4,1) EQ '1').MTVX3 . SKIP IF TYPE-2740S 13180000
AIF ('&MD2'(3,2) EQ '00').MTV04 . SKIP IF NEITHER 2740M2S/SB 13190000
.* 13200000
.MTVX3 ANOP . 2740S/2740M2S/2740M2SB 13210000
*P// INCLUDE NAME-$$MLTC,UNIT-&DSUNIT 13220000
.* 13230000
.MTV04 ANOP . CONTINUE HERE 13240000
AIF ('&MD1'(3,1) NE '1').MTV05 . SKIP IF NO TYPE-2740 13250000
AIF ('&MD1'(7,1) NE '1').NC15 . SKIP IF NO TYPE-2740D 13260000
*P// INCLUDE NAME-$$MLTD,$$MLTF',UNIT-&DSUNIT 13270000
AGO .MTV06 13280000
.* 13290000
.NC15 ANOP 13300000
*P// INCLUDE NAME-$$MLTD,UNIT-&DSUNIT 13310000
.* 13320000
.MTV05 ANOP . CONTINUE HERE 13330000
AIF ('&MD1'(7,1) NE '1').MTV06 . SKIP IF NO TYPE-2740D 13340000
.* 13350000
*P// INCLUDE NAME-$$MLTF,UNIT-&DSUNIT 13360000
.* 13370000
.MTV06 ANOP . CONTINUE HERE 13380000
AIF ('&MD1'(5,1) EQ '1').MTVX6 . SKIP IF TYPE-2740C 13390000
AIF ('&MD3'(1,1) NE '1').MTV07 . SKIP UNLESS TYPE-SYS7C 13400000
.* 13410000
.MTVX6 ANOP . TYPE-2740C/SYS7C 13420000
*P// INCLUDE NAME-$$MLTG,UNIT-&DSUNIT 13430000
.* 13440000
.MTV07 ANOP . CONTINUE HERE 13450000
AIF ('&MD2'(1,1) EQ '1').MTVX7 . SKIP IF TYPE-2740DC 13460000
AIF ('&MD3'(3,1) NE '1').MTV08 . SKIP UNLESS TYPE-SYS7DC 13470000
.* 13480000
.MTVX7 ANOP . TYPE-2740DC/SYS7DC 13490000
*P// INCLUDE NAME-$$MLTH,UNIT-&DSUNIT 13500000
.* 13510000
.MTV08 ANOP . CONTINUE HERE 13520000
AIF ('&MD1'(8,1) NE '1').MTV09 . SKIP UNLESS TYPE-2740DT 13530000
AIF ('&MD2'(2,1) NE '1').NC20 . SKIP UNLESS TYPE-2740DTC 13540000
*P// INCLUDE NAME-$$MLTI,$$MLTL',UNIT-&DSUNIT 13550000
AGO .MTV10 13560000
.* 13570000
.NC20 ANOP 13580000
*P// INCLUDE NAME-$$MLTI,UNIT-&DSUNIT 13590000
.* 13600000
.MTV09 ANOP . CONTINUE HERE 13610000
AIF ('&MD2'(2,1) NE '1').MTV10 . SKIP UNLESS TYPE-2740DTC 13620000
.* 13630000
*P// INCLUDE NAME-$$MLTL,UNIT-&DSUNIT 13640000
.* 13650000
.MTV10 ANOP . CONTINUE HERE 13660000
AIF ('&MD1'(1,1) EQ '1').MTV11 . SKIP IF TYPE-1050--INCLUDED 13670000
AIF ('&MD1'(6,1) EQ '1').MTVXA . SKIP IF TYPE-2740SC 13680000
AIF ('&MD2'(5,2) NE '00').MTVXA . SKIP IF TYPE-2740M2SC/SCB 13690000
AIF ('&MD3'(2,1) NE '1').MTV11 . SKIP UNLESS TYPE-SYS7C 13700000
.* 13710000
.MTVXA ANOP . TYPE-2740SC/2740M2SC(B)/SYS7SC 13720000
*P// INCLUDE NAME-$$MLTM,UNIT-&DSUNIT 13730000

```

```

.*
.MTV11 ANOP . CONTINUE HERE 13740000
AIF ('&MD2'(7,1) NE '1').MTV12 . SKIP UNLESS TYPE-2741 13750000
AIF ('&MD2'(8,1) EQ '1').MTV20 . SKIP IF TYPE-2741D 13760000
AIF ('&MD3'(4,1) NE '1').MTV21 . SKIP UNLESS TYPE-CMCSTD 13770000
.MTV20 ANOP 13780000
*P// INCLUDE NAME-$$MLTO,$$MLTP',UNIT-&DSUNIT 13790000
AGO .MTV99 13800000
.MTV21 ANOP 13810000
AIF ('&MD2'(7,1) NE '1').MTV12 . SKIP UNLESS TYPE-2741 13820000
.* 13830000
*P// INCLUDE NAME-$$MLTO,UNIT-&DSUNIT 13840000
.* 13850000
.MTV12 ANOP . CONTINUE HERE 13860000
AIF ('&MD2'(8,1) EQ '1').MTVXC . SKIP IF TYPE-2741D 13870000
AIF ('&MD3'(4,1) NE '1').MTV99 . SKIP UNLESS TYPE-CMCSTD 13880000
.* 13890000
.MTVXC ANOP 13900000
*P// INCLUDE NAME-$$MLTP,UNIT-&DSUNIT 13910000
.* 13920000
.MTV99 ANOP . END INCLUDE TRANSFER VECTORS 13930000
.GXNOM ANOP . CONTINUE HERE 13940000
.* 13950000
*P// END 13960000
.* 13970000
.*----- LINK EDIT THE LOADABLE COMMUNICATIONS TRACE -----* 13980000
.* 13990000
AIF (&BSC EQ '0').LET02 . SKIP IF BSCA NOT SUPPORTED 14000000
.* 14010000
AIF (&CL EQ '1').CL270 . SKIP IF CARD GENERATION. 14020000
*P// CEND 14030000
*P// COPY NAME-$CC1BT,LIBRARY-P 14040000
.CL270 ANOP . CONTINUE HERE. 14050000
.* 14060000
*** USE THE OVERLAY LINKAGE EDITOR TO CREATE THE LOADABLE BSCA TRACE 14070000
*** SERVICE AID 14080000
* 14090000
*P// LOAD $OLINK,&DIUNIT 14100000
*P// FILE NAME-$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-100,RETAIN-S 14110000
*P// FILE NAME-$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-40,RETAIN-S 14120000
*P// RUN 14130000
*P// PHASE NAME-$CC$BS,UNIT-&CCUNIT,RETAIN-R 14140000
*P// OPTIONS LEVEL-&LVL 14150000
*P// INCLUDE NAME-$CC$BS,UNIT-&DIUNIT 14160000
*P// INCLUDE NAME-$B$STT,UNIT-&DSUNIT 14170000
*P// END 14180000
.* 14190000
.LET02 ANOP . CONTINUE HERE 14200000
.* 14210000
AIF (&MLA EQ '0').LET99 . SKIP IF MLTA NOT SUPPORTED 14220000
.* 14230000
AIF (&CL EQ '1').CL300 . SKIP IF CARD GENERATION. 14240000
*P// CEND 14250000
*P// COPY NAME-$CC1MT,LIBRARY-P 14260000
.CL300 ANOP . CONTINUE HERE. 14270000
.* 14280000
*** USE THE OVERLAY LINKAGE EDITOR TO CREATE THE LOADABLE MLTA TRACE 14290000
*** SERVICE AID 14300000
* 14310000
*P// LOAD $OLINK,&DIUNIT 14320000
14330000

```

*P// FILE NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-100,RETAIN-S	14340000
*P// FILE NAME-\$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-40,RETAIN-S	14350000
*P// RUN	14360000
*P// PHASE NAME-\$CC\$ML,UNIT-&CCUNIT,RETAIN-R	14370000
*P// OPTIONS LEVEL-&LVL	14380000
*P// INCLUDE NAME-\$CC\$ML,UNIT-&DIUNIT	14390000
*P// INCLUDE NAME-\$SMLDT,UNIT-&DSUNIT	14400000
*P// END	14410000
.LET99 ANOP . CONTINUE HERE	14420000
.*	14430000
. AIF (&CL EQ '1').CL360 . SKIP IF CARD GENERATION.	14440000
*P// CEND	14450000
*P// COPY NAME-\$CC1C1,LIBRARY-P	14460000
.CL360 ANOP . CONTINUE HERE.	14470000
.*	14480000
. *----- OUTPUT THE CCP MODULE COPY GROUP -----*	14490000
.*	14500000
*?? *** BEGIN NEW PAGE IN OUTPUT LISTING ***	14510000
* --- COPY REQUIRED LOAD MODULES FOR ALL STAGES OF CCP ---	14520000
*	14530000
*P// LOAD \$MAINT,&DSUNIT	14540000
*P// RUN	14550000
*P// COPY FROM-&D,TO-&C,LIBRARY-O,RETAIN-R,NAME-\$CC1BF	14560000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCPAS	14570000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCY.ALL,NEWNAME-\$CC2	14580000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCPAL	14590000
.*	14600000
. AIF (&FDFE NE '1').AYNOF . SKIP IF NO DFF REQUESTED	14610000
.*	14620000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCPDF	14630000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCPPF	14640000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCZ.ALL,NEWNAME-\$CC2	14650000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCPDT	14660000
. AIF (&DSUNIT EQ &C).AYNOF . SKIP IF CCP AND DSM SAME @01	14670000
*P// COPY FROM-&DSUNIT,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$OLBO	14680000
*P// COPY FROM-&DSUNIT,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$OLFTP	14690000
*P// COPY FROM-&DSUNIT,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$OLER	14700000
.*	14710000
.AYNOF AIF (&UPW NE '1').AYNOU . SKIP IF NOT SECURE-USER	14720000
.*	14730000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCPAU	14740000
.*	14750000
.AYNOU ANOP . CONTINUE HERE	14760000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCP	14770000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CC3.ALL	14780000
.*	14790000
. * STARTUP TRANSIENT	14800000
.*	14810000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CC4SU	14820000
.*	14830000
. * DUMMY USER SECURITY ROUTINE	14840000
.*	14850000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CC4YA	14860000
.*	14870000
. * SYSTEM OPERATOR COMMAND ROUTINES	14880000
.*	14890000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCS.ALL,NEWNAME-\$CC4	14900000
.*	14910000
. * TERMINAL OPERATOR COMMAND ROUTINES	14920000
.*	14930000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-\$CCT.ALL,NEWNAME-\$CC4	14930000


```

.* 14940000
.* MESSAGE-ISSUING MODULES 14950000
.* 14960000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCE.ALL,NEWNAME-$CC4 14970000
.* 14980000
.* COMMUNICATIONS MANAGEMENT -- MLTA ROUTINES 14990000
.* 15000000
    AIF (&MLA EQ '0').CPXL5 . SKIP IF *NO* MLTA 15010000
    AIF (&CL EQ '1').CL365 . SKIP IF CARD GENERATION. 15020000
*P// END 15030000
*P// CEND 15040000
*P// COPY NAME-$CC1C2,LIBRARY-P 15050000
*P// LOAD $MAINT,&DSUNIT 15060000
*P// RUN 15070000
.CL365 ANOP . CONTINUE HERE 15080000
.* 15090000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCM.ALL,NEWNAME-$CC4 15100000
.* 15110000
.CPY10 ANOP . CONTINUE HERE 15120000
.* 15130000
.* COPY TRANSLATE ROUTINES 15140000
    AIF ('&MXC'(3,1) NE '1').CPXL2 .SKIP IF NO PTTCCBD CODE 15150000
.* 15160000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J1 15170000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J7 15180000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J9 15190000
.* 15200000
.CPXL2 ANOP . CONTINUE HERE 15210000
    AIF (&XM4E NE '1').CPXL3 . SKIP IF NO 2740/1 PTTCEBCD 15220000
.* 15230000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J2 15240000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J5 15250000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4JA 15260000
.* 15270000
.CPXL3 ANOP . CONTINUE HERE 15280000
    AIF ('&MXC'(1,1) NE '1').CPXL4 . SKIP IF NO CORR CODE 15290000
.* 15300000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J3 15310000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J6 15320000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4JB 15330000
.* 15340000
.CPXL4 ANOP . CONTINUE HERE 15350000
    AIF (&MT50 NE '1').CPXL5 . SKIP IF NO 1050 CODE 15360000
.* 15370000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J4 15380000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4J8 15390000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4JC 15400000
.* 15410000
.CPXL5 ANOP . CONTINUE HERE 15420000
    AIF (&CL EQ '1').CL367 . SKIP IF CARD GENERATION. 15430000
*P// END 15440000
*P// CEND 15450000
*P// COPY NAME-$CC1C3,LIBRARY-P 15460000
*P// LOAD $MAINT,&DSUNIT 15470000
*P// RUN 15480000
.CL367 ANOP . CONTINUE HERE 15490000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4JE 15495000
    AIF ('&BFA'(7,1) NE '1').CPXL9 . SKIP IF NO ASCII 15500000
.* 15510000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC4JD 15520000

```

```

.*
.CPXL9 ANOP . CONTINUE HERE 15540000
.* 15550000
.* 15560000
.* COMMUNICATIONS MANAGEMENT -- COMMON 15570000
.* 15580000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCC.ALL,NEWNAME-$CC4 15590000
.* 15600000
.* COMMUNICATIONS MANAGEMENT -- BSCA ROUTINES 15610000
.* 15620000
.* AIF (&BSC EQ '0').CPY14 . SKIP IF *NO* BSCA SUPPORT 15630000
.* 15640000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCB.ALL,NEWNAME-$CC4 15650000
.* 15660000
.* AGO .CPY16 15660800
.* 15661600
.CPY14 ANOP . CONTINUE HERE 15662400
.* 15663200
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCBBC,NEWNAME-$CC4BC 15664000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCBBF,NEWNAME-$CC4BF 15664800
.* 15665600
.* AGO .CPY20 15666400
.* 15667200
.CPY16 ANOP . CONTINUE HERE 15668000
.* 15668800
.* AIF (&MINRES NE '1').CPY20 . SKIP IF *MINRES* NO 15670000
.* 15680000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCD.ALL,NEWNAME-$CC4 15690000
.* 15700000
.CPY20 ANOP . CONTINUE HERE 15710000
.* 15720000
.* COMMUNICATIONS INTERFACE ROUTINES 15730000
.* 15740000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCI.ALL,NEWNAME-$CC4 15750000
.* 15760000
.* DFF TRANSIENT ROUTINES 15770000
.* 15780000
.* AIF (&FDFE NE '1').CPY30 . SKIP IF *NO* DFF SUPPORT 15790000
.* 15800000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCF.ALL,NEWNAME-$CC4 15810000
.* 15820000
.CPY30 ANOP . CONTINUE HERE 15830000
.* 15840000
.* COMMAND PROCESSOR MAINLINE ROUTINES 15850000
.* 15860000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCG.ALL,NEWNAME-$CC4 15870000
.* 15880000
.* PROGRAM REQUEST ROUTINES 15890000
.* 15900000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCR.ALL,NEWNAME-$CC4 15910000
.* 15920000
.* ALLOCATATION ROUTINES 15930000
.* 15940000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCA.ALL,NEWNAME-$CC4 15950000
.* 15960000
.* DATA MANAGEMENT -- OPEN 15970000
.* 15980000
.*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCO.ALL,NEWNAME-$CC4 15990000
.* 16000000
.* DATA MANAGEMENT -- CLOSE 16010000
.* 16020000

```

```

*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCL.ALL,NEWNAME-$CC4 16030000
.* 16040000
.* TASK TERMINATION/DEALLOCATION 16050000
.* 16060000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCX.ALL,NEWNAME-$CC4 16070000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC5SH 16080000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CCPDD 16090000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-CCPIVP 16110000
*P// COPY FROM-&D,TO-&C,RETAIN-R,LIBRARY-O,NAME-$CC6.ALL,NEWNAME-$CCP 16120000
    AIF (&CL EQ '1').CL395 . SKIP IF CARD GENERATION. 16130000
*P// COPY FROM-&D,TO-&C,LIBRARY-S,NAME-$CGSST,RETAIN-R 16140000
.CL395 ANOP 16150000
*P// END 16160000
.* 16170000
.*----- OUTPUT THE USER SECURITY MODULE CREATION GROUP -----* 16180000
.* 16190000
    AIF (&UPW NE '1').Z9END . SKIP IF NOT SECURE-USER 16200000
.* 16210000
    AIF (&CL EQ '1').CL400 . SKIP IF CARD GENERATION. 16220000
*P// CEND 16230000
*P// COPY NAME-$CC1US,LIBRARY-P 16240000
.CL400 ANOP . CONTINUE HERE. 16250000
.* 16260000
*?? *** BEGIN NEW PAGE IN OUTPUT LISTING *** 16270000
* --- CREATE EMPTY USER SECURITY INFORMATION MODULE --- 16280000
* 16290000
*** USE SCP GENERATOR TO CREATE THE MODULE $CC4Z9 16300000
* 16310000
*P// LOAD $CGDRV,&DIUNIT 16320000
*P// SWITCH 0XXXXXXX 16330000
*P// FILE NAME-$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-10,RETAIN-S 16340000
*P// FILE NAME-$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-10,RETAIN-S 16350000
*P// FILE NAME-$WORK2,UNIT-&UWVK2,PACK-&PWVK2,TRACKS-10,RETAIN-S 16360000
.* 16370000
    AIF (&CL EQ '1').CL410 . SKIP IF CARD GENERATION. 16380000
*P// COMPILE UNIT-&DIUNIT,SOURCE-$CC1US 16390000
*P// RUN 16400000
*P// CEND 16410000
*P// COPY NAME-$CC1US,LIBRARY-S 16420000
    AGO .CL420 16430000
.CL410 ANOP . CONTINUE HERE. 16440000
*P// RUN 16450000
.CL420 ANOP . CONTINUE HERE. 16460000
.* 16470000
$CC4Z9 TITLE 'MODULE&#.TO&#.HOLD&#.USER&#.SECURITY&#.INFORMATION' 16480000
***** 16490000
* USER SECURITY INFORMATION * 16500000
***** 16510000
    SPACE 2 16520000
* THIS MODULE IS CREATED -- WITH NULL CONTENTS -- AS THE PLACE TO 16530000
* HOLD SECURITY INFORMATION THAT WILL BE USED, DURING THE EXECUTION 16540000
* OF CCP, BY YOUR OWN SIGN-ON CHECKING ROUTINE. THE INFORMATION IS 16550000
* FILLED IN THIS MODULE BY THE SUPPLIED PROGRAM $CCPAU. 16560000
    SPACE 1 16570000
$CC4Z9 START 0 MODULE NAME 16580000
    DC &LUS.XL1'00' &LUS BYTES, INITIALLY HEX ZEROS 16590000
    END $CC4Z9 16600000
.* 16610000
    AIF (&CL EQ '1').CL430 . SKIP IF CARD GENERATION. 16620000
*P// CEND 16630000

```

*P// COPY NAME-\$CC1UL,LIBRARY-P	16640000
AGO .CL440	16650000
.CL430 ANOP	16660000
P/	16670000
.CL440 ANOP . CONTINUE HERE.	16680000
*	16690000
*** USE OVERLAY LINKAGE EDITOR TO FORM \$CC4Z9 AS A LOAD MODULE	16700000
*	16710000
*P// LOAD \$SOLINK,&DIUNIT	16720000
*P// FILE NAME-\$SOURCE,UNIT-&UWSRC,PACK-&PWSRC,TRACKS-10,RETAIN-S	16730000
*P// FILE NAME-\$WORK,UNIT-&UWWRK,PACK-&PWWRK,TRACKS-10,RETAIN-S	16740000
*P// RUN	16750000
*P// PHASE NAME-\$CC4Z9,UNIT-&CCUNIT,RETAIN-R	16760000
*P// INCLUDE NAME-\$CC4Z9,UNIT-&DIUNIT	16770000
*P// END	16780000
.Z9END ANOP . END OUTPUT USER SECURITY	16790000
*	16800000
.*----- OUTPUT THE \$CCPFILE ALLOCATION GROUP -----*	16810000
*	16820000
AIF (&CL EQ '1').CL470 . SKIP IF CARD GENERATION.	16830000
*P// CEND	16840000
*P// COPY NAME-\$CC1BF,LIBRARY-P	16850000
.CL470 ANOP	16860000
*	16870000
*?? *** BEGIN NEW PAGE IN OUTPUT LISTING ***	16880000
* --- ALLOCATE \$CCPFILE AND WRITE INITIAL CONTENTS ---	16890000
*	16900000
*** EXECUTE THE PREVIOUSLY CREATED LOAD MODULE \$CC1BF TO WRITE THE	16910000
*** INITIAL CONTENTS OF \$CCPFILE	16920000
*	16930000
AIF (&UFIL EQ &DIUNIT).FANOP . SKIP IF TO DISTRIBUTION PACK	16940000
*	16950000
*P// PAUSE ASSURE UNIT &UFIL CONTAINS PACK &PFIL -- THEN CONTINUE	16960000
*	16970000
.FANOP ANOP . CONTINUE HERE	16980000
*	16990000
*P// LOAD \$CC1BF,&DIUNIT	17000000
*	17010000
AIF (&TFIL EQ '0').FANOT . SKIP IF TRKLOC NOT SPECIFIED	17020000
*	17030000
*P// FILE NAME-\$CCPFILE,UNIT-&UFIL,PACK-&PFIL,TRACKS-&ASIZE,RETAIN-P,	17040000
*P// LOCATION-&TFIL	17050000
AGO .FACOM . JOIN COMMON CODE	17060000
*	17070000
.FANOT ANOP	17080000
*P// FILE NAME-\$CCPFILE,UNIT-&UFIL,PACK-&PFIL,TRACKS-&ASIZE,RETAIN-P	17090000
*	17100000
.FACOM ANOP . CONTINUE HERE	17110000
*P// RUN	17120000
*	17130000
.*----- OUTPUT THE COPY PROGRAM SUPPORT GROUP -----*	17140000
*	17150000
AIF ('&PLG'(1,1) NE '1').PP020 . SKIP IF NO COBOL	17160000
AIF (&CL EQ '1').CL501 . SKIP IF CARD GENERATION.	17170000
*P// CEND	17180000
*P// COPY NAME-\$CC1CO,LIBRARY-P	17190000
.CL501 ANOP . CONTINUE HERE.	17200000
*?? *** BEGIN NEW PAGE IN OUTPUT LISTING ***	17210000
*	17220000
*** COPY SUBROUTINES FOR USE WITH COBOL TO PROGRAM PREPARATION PACK	17230000

```

*
*P// PAUSE ASSURE COBOL PROGRAM PACK ON UNIT &UPCBL -- THEN CONTINUE 17240000
*P// LOAD $MAINT,&DSUNIT 17250000
*P// RUN 17260000
*P// COPY FROM-&DIUNIT,TO-&UPCBL,RETAIN-R,LIBRARY-R,NAME-CCPCIO 17270000
*P// END 17280000
.* 17290000
.* 17300000
.PP020 ANOP 17310000
    AIF ('&PLG'(2,1) NE '1').PP030 . SKIP IF NO FORTRAN 17320000
    AIF (&CL EQ '1').CL503 . SKIP IF CARD GENERATION. 17330000
*P// CEND 17340000
*P// COPY NAME-$CC1FT,LIBRARY-P 17350000
.CL503 ANOP . CONTINUE HERE. 17360000
.* 17370000
*** COPY SUBROUTINES FOR USE WITH FORTRAN TO PROGRAM PREPARATION PACK 17380000
* 17390000
*P// PAUSE ASSURE FORTRAN PROGRAM PACK ON UNIT &UPFOR -- THEN CONTINUE 17400000
*P// LOAD $MAINT,&DSUNIT 17410000
*P// RUN 17420000
*P// COPY FROM-&DIUNIT,TO-&UPFOR,RETAIN-R,LIBRARY-R,NAME-CCPFIO 17430000
*P// END 17440000
.* 17450000
.PP030 ANOP 17460000
    AIF ('&PLG'(3,1) NE '1').PP050 . SKIP IF NO ASSEM 17470000
    AIF (&CL EQ '1').CL505 . SKIP IF CARD GENERATION. 17480000
*P// CEND 17490000
*P// COPY NAME-$CC1AS,LIBRARY-P 17500000
.CL505 ANOP . CONTINUE HERE. 17510000
.* 17520000
*** COPY MACRO DEFINITIONS AND ANY REQUIRED SUBROUTINES FOR USE WITH 17530000
*** BASIC ASSEMBLER TO PROGRAM PREPARATION PACK 17540000
* 17550000
*P// PAUSE ASSURE BASIC ASSEMBLER PACK ON UNIT &UPASM -- THEN CONTINUE 17560000
*P// LOAD $MAINT,&DSUNIT 17570000
*P// RUN 17580000
*P// COPY FROM-&DIUNIT,TO-&UPASM,RETAIN-R,LIBRARY-S,NAME-$N.ALL 17590000
*P// END 17600000
.* 17610000
.PP050 ANOP . CONTINUE HERE 17620000
    AIF ('&PLG'(4,1) NE '1').PP010 . SKIP IF NO RPG SUPPORT 17630000
    AIF (&CL EQ '1').CL507 . SKIP IF CARD GENERATION. 17640000
*P// CEND 17650000
*P// COPY NAME-$CC1RG,LIBRARY-P 17660000
.CL507 ANOP . CONTINUE HERE. 17670000
.* 17680000
*** COPY SUBROUTINES FOR USE WITH RPG II TO PROGRAM PREPARATION PACK 17690000
* 17700000
*P// PAUSE ASSURE RPG II PROGRAM PACK ON UNIT &UPRPG -- THEN CONTINUE 17710000
*P// LOAD $MAINT,&DSUNIT 17720000
*P// RUN 17730000
*P// COPY FROM-&DIUNIT,TO-&UPRPG,RETAIN-R,LIBRARY-R,NAME-SUBR.ALL 17740000
*P// END 17750000
.* 17760000
.PP010 ANOP 17770000
.* 17780000
.*----- OUTPUT THE AUXILIARY MODULES GROUP -----* 17790000
.* 17800000
*?? *** BEGIN NEW PAGE IN OUTPUT LISTING *** 17810000
.* 17820000
    AIF (&CL NE '1').CL510 . SKIP IF CARDLESS GENERATION 17830000

```

```

*      --- PUNCH OUT THE AUXILIARY ELEMENTS REQUIRED IN CARD FORM --- 17840000
*
*** PUNCH SAMPLE ASSIGNMENT SET AND STARTUP OCL 17850000
*
      AGO      .CL520          . SKIP CARDLESS 17860000
.CL510 ANOP          . CONTINUE HERE 17870000
*P// CEND 17880000
*P// COPY NAME-$CC1ND,LIBRARY-P 17890000
*
*** PRINT SAMPLE ASSIGNMENT SET TO BE USED WITH INSTALLATION 17900000
*** VERIFICATION PROGRAM 17910000
*
.CL520 ANOP 17920000
*P// LOAD $MAINT,&DSUNIT 17930000
*P// RUN 17940000
*
      AIF      (&CL EQ '1').CL525 . SKIP IF CARD GENERATION. 17950000
*
*P// COPY FROM-&D,TO-PRINT,LIBRARY-S,NAME-$CGSST 17960000
*P// COPY FROM-&D,TO-PRINT,LIBRARY-S,NAME-$CGCND 17970000
      AGO      .CL530 18000000
.CL525 ANOP 18010000
*
*P// COPY FROM-&D,TO-PRTPCH,LIBRARY-S,NAME-$CGSET 18020000
*P// COPY FROM-&D,TO-PRINT,LIBRARY-S,NAME-$CGEND 18030000
.CL530 ANOP 18040000
*P// END 18050000
*
      AIF      (&CL EQ '1').CL700 . SKIP IF CARD GENERATION. 18060000
*P// CEND 18070000
*?? *** BEGIN NEW PAGE IN OUTPUT LISTING *** 18080000
*
.* BUILD THE PROCEDURE THAT WILL PERFORM CCP GENERATION PASS 2. 18090000
*
*P// COPY NAME-$CCPSA,LIBRARY-P 18100000
*** PERFORM CCP GENERATION PASS 2 *** 18110000
*
*P// CALL $CC1FC,&DIUNIT *** ASSEMBLE $CC1FC 18120000
*P// CALL $CC1LC,&DIUNIT *** CREATE R-MODULE $CC1FC 18130000
*P// CALL $CC1#1,&DIUNIT *** EXPAND $CC4#1 MOCROS 18140000
*P// CALL $CC1#A,&DIUNIT *** ASSEMBLE $CC4#1 18150000
*P// CALL $CC1L1,&DIUNIT *** CREATE $CC4#1 18160000
*P// CALL $CC1VT,&DIUNIT *** EXPAND $CC4VT MACROS 18170000
*P// CALL $CC1VS,&DIUNIT *** ASSEMBLE $CC4VT 18180000
*P// CALL $CC1#2,&DIUNIT *** EXPAND $CC4#2 MACROS 18190000
*P// CALL $CC1#B,&DIUNIT *** ASSEMBLE $CC4#2 18200000
*P// CALL $CC1LE,&DIUNIT *** CREATE $CC4#2 18210000
      AIF      (&BSC EQ '0').CL550 . SKIP IF BSCA NOT SUPPORTED 18220000
*P// CALL $CC1BT,&DIUNIT *** CREATE BSCA TRACE MODULE 18230000
.CL550 ANOP 18240000
      AIF      (&MLA EQ '0').CL560 . SKIP IF MLTA NOT SUPPORTED 18250000
*P// CALL $CC1MT,&DIUNIT *** CREATE MLTA TRACE MODULE 18260000
.CL560 ANOP 18270000
*P// CALL $CC1C1,&DIUNIT *** COPY REQUIRED CCP MODULES 18280000
      AIF      (&MLA EQ '0').CL565 . SKIP IF MLTA NOT SUPPORTED 18290000
*P// CALL $CC1C2,&DIUNIT *** COPY REQUIRED CCP MODULES 18300000
.CL565 ANOP 18310000
*P// CALL $CC1C3,&DIUNIT *** COPY REQUIRED CCP MODULES 18320000
      AIF      (&UPW NE '1').CL570 . SKIP IF NOT SECURE-USER 18330000
*P// CALL $CC1US,&DIUNIT *** ASSEMBLE USER SECURITY 18340000

```

```
*P// CALL $CC1UL,&DIUNIT          *** CREATE USER SECURITY $CC4Z9 18440000
.CL570 ANOP                        18450000
*P// CALL $CC1BF,&DIUNIT          *** CREATE $CCPFILE 18460000
AIF ('&PLG'(1,1) NE '1').CL580 . SKIP IF NO COBOL 18470000
*P// CALL $CC1CO,&DIUNIT          *** COPY COBOL MODULES 18480000
.CL580 ANOP                        18490000
AIF ('&PLG'(2,1) NE '1').CL590 . SKIP IF NO FORTRAN 18500000
*P// CALL $CC1FT,&DIUNIT          *** COPY FORTRAN MODULES 18510000
.CL590 ANOP                        18520000
AIF ('&PLG'(3,1) NE '1').CL600 . SKIP IF NO ASSEM 18530000
*P// CALL $CC1AS,&DIUNIT          *** COPY ASSEMBLER MODULES 18540000
.CL600 ANOP                        18550000
AIF ('&PLG'(4,1) NE '1').CL610 . SKIP IF NO RPG SUPPORT 18560000
*P// CALL $CC1RG,&DIUNIT          *** COPY RPGII MODULES 18570000
.CL610 ANOP                        18580000
*P// CALL $CC1ND,&DIUNIT          *** COPY SAMPLE ASSIGNMENT 18590000
*P// CEND                          18600000
*P// END                            18610000
.CL700 ANOP                        . CONTINUE HERE. 18620000
*?? *** END OF PASS 1 OUTPUT *** 18630000
MEND                               18640000
```

```

MACRO                                00010000
*****                                00020000
.*                                     * 00030000
.* NAME: $EMLA                        * 00040000
.*                                     * 00050000
.* MODIFICATION LEVEL: VERSION 0, MODIFICATION LEVEL 0 * 00060000
.*                                     * 00070000
.* FUNCTION:                          * 00080000
.*                                     * 00090000
.* . CCP GENERATION FIRST PASS MACRO INSTRUCTION -- DEFINE * 00100000
.* MLTA ADAPTER AND MLTA SUPPORT      * 00110000
.*                                     * 00120000
.* INPUT OPERANDS:                   * 00130000
.*                                     * 00140000
.* . LINES-NUMBER                    * 00150000
.*                                     * 00160000
.* SPECIFIES NUMBER OF LINES ON MLTA ADAPTER TO BE SUPPORTED. * 00170000
.* OPERAND REQUIRED IF STATEMENT USED. LINES-0 VALID TO SIGNIFY * 00180000
.* NO MLTA SUPPORT.                  * 00190000
.*                                     * 00200000
.* . XLATE-YES/NO                    * 00210000
.*                                     * 00220000
.* 'YES' SPECIFIES *ALL* MLTA OPERATIONS PERFORMED BY USER * 00230000
.* PROGRAMS WILL TRANSLATE TO/FROM EBCDIC. 'NO' SPECIFIES SOME * 00240000
.* OPERATIONS WILL *NOT* TRANSLATE. DEFAULT IS 'YES'.        * 00250000
.*                                     * 00260000
*****                                00270000
$EMLA &LINES-,&XLATE-YES            00280000
.*                                     00290000
GBLA &SEQ                            . SEQUENCE CONTROL:    00300000
.*                                     . MUST BE 5 UPON ENTRY  00310000
.*                                     . IS MADE 6 AFTER PROCESSING 00320000
GBLB &TERR                            . TERMINATION ERROR SWITCH 00330000
GBLA &MLA                             . SET TO NUMBER MLTA LINES 00340000
GBLB &MNOX                             . 1=MOVE-ONLY PERMITTED    00350000
LCLA &NUM                              . TO TEST NUMERIC OPERANDS 00360000
.*                                     00370000
TABLE &XLATE                          00380000
YES TABDF 0                            00390000
Y TABDF 0                              00400000
NO TABDF 1                             00410000
N TABDF 1                              00420000
TABDF *                                . ERROR PARAMETER        00430000
TEXT                                    00440000
.*                                     00450000
.*----- CHECK STATEMENT SEQUENCE -----* 00460000
.*                                     00470000
AIF (&SEQ EQ '5').SEQOK               . SKIP IF GOOD SEQUENCE  00480000
.*                                     00490000
*!250E $EMLA STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR 00500000
&TERR SETB 1                          . SET TERMINATION ERROR SWITCH 00510000
&MNOX SETB 0                           . MAKE NULL PREVIOUS SPEC 00520000
.*                                     00530000
.SEQOK ANOP                            . STATEMENT IN SEQUENCE  00540000
.*                                     00550000
.*----- LINES-0/N -----* 00560000
.*                                     00570000
AIF (T'&LINES NE '0').MA100           . SKIP IF OPERAND SPECIFIED 00580000
.*                                     00590000

```



```

*!260E MISSING 'LINES' OPERAND -- MUST BE SPECIFIED IF STATEMENT USED 00600000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00610000
&MLA SETA 0 . SET AS IF LINES-0 00620000
AGO .LIEND 00630000
.* 00640000
.MA100 ANOP 00650000
AIF (T'&LINES NE 'N').LI900 . SKIP IF PARM NOT NUMERIC 00660000
AIF (K'&LINES GT '4').LI900 . SKIP IF MORE THAN 4 DIGITS-ERR 00670000
&NUM SETA &LINES . SET NUMERIC VALUE 00680000
AIF (&NUM LE '8').LI200 . SKIP IF RANGE 0-8 00690000
.* 00700000
.LI900 ANOP 00710000
*!255E INVALID 'LINES' PARAMETER -- MUST BE NUMBER IN RANGE 0-8 00720000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00730000
&MLA SETA 0 . SET AS IF LINES-0 00740000
AGO .LIEND 00750000
.* 00760000
.LI200 ANOP 00770000
&MLA SETA &LINES 00780000
.* 00790000
.LIEND ANOP . END OPERAND 'LINES' 00800000
.* 00810000
*----- XLATE-YES/Y/NO/N -----* 00820000
.* 00830000
AIF (&XLATE NE '*').XLOK . SKIP IF VALID PARAMETER 00840000
.* 00850000
*!270E INVALID 'XLATE' PARAMETER -- MUST BE YES/Y/NO/N 00860000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00870000
AGO .END 00880000
.* 00890000
.XLOK ANOP . VALID PARAMETER 00900000
AIF (&XLATE EQ '0').XL0 . SKIP IF XLATE-YES 00910000
.* 00920000
&MNOX SETB 1 . XLATE-NO 00930000
AIF (&LINES NE '0').XLEND . SKIP UNLESS LINES-0 00940000
.* 00950000
*!265E LINES-0, BUT OTHER KEYWORD SPECIFIED WITH NON-DEFAULT PARAMETER 00960000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00970000
AGO .XLEND 00980000
.* 00990000
.XL0 ANOP . XLATE-YES 01000000
&MNOX SETB 0 01010000
.* 01020000
.XLEND ANOP . END OPERAND 'XLATE' 01030000
.* 01040000
*----- SET STATEMENT SEQUENCE -----* 01050000
.* 01060000
.END ANOP 01070000
&SEQ SETA 6 . INDICATE $EMLA PROCESSED 01080000
.* 01090000
MEND 01100000

```

MODULE-\$EMLD , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $EMLD 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 0, MODIFICATION LEVEL 0 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION FIRST PASS MACRO INSTRUCTION -- DEFINE AN 00100000
.* MLTA TERMINAL DEVICE TO BE SUPPORTED. 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . TYPE-1050/1050D/2740/2740S/2740C/2740SC/2740D/2740DT/2740DC/ 00150000
.* 2740DTC/2740M2S/2740M2SB/2740M2SC/2740M2SCB/2741/2741D/ 00160000
.* SYS7C/SYS7SC/CMCSTD 00170000
.* 00180000
.* DEVICE TYPE SUPPORTED. REQUIRED OPERAND. 00190000
.* 00200000
.* . XMCODE-CORR/PTTCEBCD/PTTCBCD 00210000
.* 00220000
.* TRANSMISSION CODE USED WITH THE DEVICE. REQUIRED OPERAND. 00230000
.* 00240000
***** 00250000
$EMLD &TYPE-, &XMCODE- 00260000
.* 00270000
GBLA &SEQ . SEQUENCE CONTROL: 00280000
.* . MUST BE 6 OR 7 UPON ENTRY 00290000
.* . IS MADE 7 AFTER PROCESSING 00300000
GBLB &TERR . TERMINATION ERROR SWITCH 00310000
GBLA &MLA . NUMBER MLTA LINES 00320000
GBLB &MT40, &MT41, &MT50 . TERMINALS SUPPORTED 00330000
GBLB &MFSC, &MFCK, &MFNK, &MFBR, &MFSW, &MFTC . FEATURES SUPPORTED 00340000
GBLB &XM4E . 2740/1 PTTCEBCD CODE PRESENT 00350000
GBLC &MXC . TRANSMISSION CODE SUPPORT 00360000
GBLC &MD1, &MD2, &MD3 . MLTA DEVICE SUPPORT 00370000
.* 00380000
LCLB &XCORR, &XEBCD, &XBCD . TRANSMISSION CODE SWITCHES 00390000
LCLC &C1, &C2 . USED TO CONSTRUCT XMCODES 00400000
.* 00410000
TEXT 00420000
.* 00430000
.*----- CHECK STATEMENT SEQUENCE -----* 00440000
.* 00450000
AIF (&SEQ EQ '6').SEQ1 . SKIP IF FIRST $EMLD STATEMENT 00460000
AIF (&SEQ EQ '7').SEQOK . SKIP IF SUCCEEDING $EMLD STMT 00470000
.* 00480000
*!300E $EMLD STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR 00490000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00500000
.* 00510000
.SEQ1 ANOP . FIRST $EMLD--INITIALIZE 00520000
&MD1 SETC '00000000' 00530000
&MD2 SETC '00000000' 00540000
&MD3 SETC '0000' 00550000
&MXC SETC '0000' 00560000
.* 00570000
.SEQOK ANOP 00580000
AIF (&MLA NE '0').MDOK . SKIP IF SOME MLTA DEFINED 00590000
```

```

.* 00600000
*!305E $EMLD STATEMENT USED, BUT NO MLTA LINES SPECIFIED 00610000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00620000
.* 00630000
.MDOK ANOP 00640000
.* 00650000
.*----- TYPE-XXXXXXXX -----* 00660000
.* 00670000
AIF (T'&TYPE NE 'O').MD010 . SKIP IF OPERAND SPECIFIED 00680000
.* 00690000
*!310E MISSING 'TYPE' OPERAND -- MUST BE SPECIFIED 00700000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00710000
AGO .END 00720000
.* 00730000
.MD010 ANOP . OPERAND SPECIFIED 00740000
AIF ('&TYPE'(1,4) EQ '1050').MD50 . 1050/1050D 00750000
AIF ('&TYPE'(1,4) EQ '2740').MD40 . 2740/2740... 00760000
AIF ('&TYPE'(1,4) EQ '2741').MD41 . 2741/2741D 00770000
AIF ('&TYPE'(1,4) EQ 'SYS7').MDS7 . SYS7/SYS7SC/SYS7DC 00780000
AIF (&TYPE EQ 'CMCSTD').MDCT . COMMUNICATING TYPEWRITER 00790000
AGO .MDERR . IF NONE OF ABOVE, ERROR 00800000
.* 00810000
.MD50 ANOP . TYPE-1050/1050D 00820000
&MT50 SETB 1 . 1050 TYPE SUPPORTED 00830000
&XEBCD SETB 1 . PTTCEBCD CODE VALID 00840000
&MFSC SETB 1 . STATION CONTROL SUPPORT 00850000
AIF (&TYPE NE '1050').MD110 . SKIP UNLESS TYPE-1050 00860000
.* 00870000
&C1 SETC '' 00880000
&C2 SETC '&MD1'(2,7) 00890000
&MD1 SETC '&C1.1&C2' 00900000
AGO .MDCK 00910000
.* 00920000
.MD110 AIF (&TYPE NE '1050D').MDERR . SKIP UNLESS TYPE-1050D 00930000
.* 00940000
&C1 SETC '&MD1'(1,1) 00950000
&C2 SETC '&MD1'(3,6) 00960000
&MD1 SETC '&C1.1&C2' 00970000
AGO .MDDC 00980000
.* 00990000
.MD40 ANOP . TYPE-2740... 01000000
&MT40 SETB 1 . 2740 SUPPORT INCLUDED 01010000
&XEBCD SETB 1 . PTTCEBCD CODE VALID 01020000
&XBCD SETB 1 . PTTCBCD CODE VALID 01030000
&XCORR SETB 1 . CORR CODE VALID 01040000
.* 01050000
AIF (&TYPE NE '2740').MD120 . SKIP UNLESS TYPE-2740 01060000
.* 01070000
&C1 SETC '&MD1'(1,2) 01080000
&C2 SETC '&MD1'(4,5) 01090000
&MD1 SETC '&C1.1&C2' 01100000
AGO .MDNK 01110000
.* 01120000
.MD120 AIF (&TYPE NE '2740S').MD130 . SKIP UNLESS TYPE-2740S 01130000
.* 01140000
&C1 SETC '&MD1'(1,3) 01150000
&C2 SETC '&MD1'(5,4) 01160000
&MD1 SETC '&C1.1&C2' 01170000
AGO .MDSN 01180000
.* 01190000

```

.MD130	AIF	(&TYPE NE '2740C')	.MD140 . SKIP UNLESS TYPE-2740C	01200000
.*				01210000
&C1	SETC	'&MD1'(1,4)		01220000
&C2	SETC	'&MD1'(6,3)		01230000
&MD1	SETC	'&C1.1&C2'		01240000
	AGO	.MDNK		01250000
.*				01260000
.MD140	AIF	(&TYPE NE '2740SC')	.MD150 . SKIP UNLESS TYPE-2740SC	01270000
.*				01280000
&C1	SETC	'&MD1'(1,5)		01290000
&C2	SETC	'&MD1'(7,2)		01300000
&MD1	SETC	'&C1.1&C2'		01310000
	AGO	.MDSC		01320000
.*				01330000
.MD150	AIF	(&TYPE NE '2740D')	.MD160 . SKIP UNLESS TYPE-2740D	01340000
.*				01350000
&C1	SETC	'&MD1'(1,6)		01360000
&C2	SETC	'&MD1'(8,1)		01370000
&MD1	SETC	'&C1.1&C2'		01380000
	AGO	.MDDN		01390000
.*				01400000
.MD160	AIF	(&TYPE NE '2740DT')	.MD170 . SKIP UNLESS TYPE-2740DT	01410000
.*				01420000
&C1	SETC	'&MD1'(1,7)		01430000
&C2	SETC	' '		01440000
&MD1	SETC	'&C1.1&C2'		01450000
	AGO	.MDTDN		01460000
.*				01470000
.MD170	AIF	(&TYPE NE '2740DC')	.MD180 . SKIP UNLESS TYPE-2740DC	01480000
.*				01490000
&C1	SETC	' '		01500000
&C2	SETC	'&MD2'(2,7)		01510000
&MD2	SETC	'&C1.1&C2'		01520000
	AGO	.MDDC		01530000
.*				01540000
.MD180	AIF	(&TYPE NE '2740DTC')	.MD190 . SKIP UNLESS TYPE-2740DTC	01550000
.*				01560000
&C1	SETC	'&MD2'(1,1)		01570000
&C2	SETC	'&MD2'(3,6)		01580000
&MD2	SETC	'&C1.1&C2'		01590000
	AGO	.MDTDC		01600000
.*				01610000
.MD190	AIF	(&TYPE NE '2740M2S')	.MD200 . SKIP UNLESS TYPE-2740M2S	01620000
.*				01630000
&C1	SETC	'&MD2'(1,2)		01640000
&C2	SETC	'&MD2'(4,5)		01650000
&MD2	SETC	'&C1.1&C2'		01660000
	AGO	.MDSN		01670000
.*				01680000
.MD200	AIF	(&TYPE NE '2740M2SB')	.MD210 . SKIP UNLESS TYPE-2740M2SB	01690000
.*				01700000
&C1	SETC	'&MD2'(1,3)		01710000
&C2	SETC	'&MD2'(5,4)		01720000
&MD2	SETC	'&C1.1&C2'		01730000
	AGO	.MDBSN		01740000
.*				01750000
.MD210	AIF	(&TYPE NE '2740M2SC')	.MD220 . SKIP UNLESS TYPE-2740M2SC	01760000
.*				01770000
&C1	SETC	'&MD2'(1,4)		01780000
&C2	SETC	'&MD2'(6,3)		01790000

&MD2	SETC	'&C1.1&C2'		01800000
	AGO	.MDSC		01810000
.*				01820000
.MD220	AIF	(&TYPE NE '2740M2SCB').MDERR	. SKIP UNL TYPE-2740M2SCB	01830000
.*				01840000
&C1	SETC	'&MD2'(1,5)		01850000
&C2	SETC	'&MD2'(7,2)		01860000
&MD2	SETC	'&C1.1&C2'		01870000
	AGO	.MDBSC		01880000
.*				01890000
.MD41	ANOP		. TYPE-2741...	01900000
&MT41	SETB	1	. 2741 SUPPORT	01910000
&XCORR	SETB	1	. CORR CODE VALID	01920000
&XEBCD	SETB	1	. PTTCEBCD CODE VALID	01930000
&XBCD	SETB	1	. PTTCBCD CODE VALID	01940000
.*				01950000
	AIF	(&TYPE NE '2741').MD300	. SKIP UNLESS TYPE-2741	01960000
.*				01970000
&C1	SETC	'&MD2'(1,6)		01980000
&C2	SETC	'&MD2'(8,1)		01990000
&MD2	SETC	'&C1.1&C2'		02000000
	AGO	.MDNK		02010000
.*				02020000
.MD300	AIF	(&TYPE NE '2741D').MDERR	. SKIP UNLESS TYPE-2741D	02030000
.*				02040000
&C1	SETC	'&MD2'(1,7)		02050000
&C2	SETC	' '		02060000
&MD2	SETC	'&C1.1&C2'		02070000
	AGO	.MDDN		02080000
.*				02090000
.MDS7	ANOP		. TYPE-SYS7...	02100000
&MT40	SETB	1	. 2740 SUPPORT	02110000
&XEBCD	SETB	1	. PTTCEBCD CODE VALID	02120000
.*				02130000
	AIF	(&TYPE NE 'SYS7C').MD350	. SKIP UNLESS TYPE-SYS7C	02140000
.*				02150000
&C1	SETC	' '		02160000
&C2	SETC	'&MD3'(2,7)		02170000
&MD3	SETC	'&C1.1&C2'		02180000
	AGO	.MDCK		02190000
.*				02200000
.MD350	AIF	(&TYPE NE 'SYS7SC').MD360	. SKIP UNLESS TYPE-SYS7SC	02210000
.*				02220000
&C1	SETC	'&MD3'(1,1)		02230000
&C2	SETC	'&MD3'(3,6)		02240000
&MD3	SETC	'&C1.1&C2'		02250000
	AGO	.MDSC		02260000
.*				02270000
.MD360	AIF	(&TYPE NE 'SYS7DC').MDERR	. SKIP UNLESS TYPE-SYS7DC	02280000
.*				02290000
&C1	SETC	'&MD3'(1,2)		02300000
&C2	SETC	'&MD3'(4,5)		02310000
&MD3	SETC	'&C1.1&C2'		02320000
	AGO	.MDDC		02330000
.*				02340000
.MDCT	ANOP		. COMMUNICATING TYPEWRITER	02350000
&MT41	SETB	1	. 2741 SUPPORT USED	02360000
&XCORR	SETB	1	. CORR CODE VALID	02370000
&C1	SETC	'&MD3'(1,3)		02380000
&C2	SETC	'&MD3'(5,4)		02390000

&MD3	SETC	'&C1.1&C2'		02400000
	AGO	.MDDN	. SKIP TO SET DIAL, NONCHK	02410000
.*				02420000
.MDERR	ANOP		. INVALID TYPE SPECIFIED	02430000
*!315E	INVALID	'TYPE' PARAMETER -- MUST BE MLTA TERMINAL DESIGNATION		02440000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	02450000
	AGO	.MDEND		02460000
.*				02470000
.MDBSC	ANOP		. BUFRCV, STACTL, CHECK	02480000
&MFBR	SETB	1	. BUFFER RECEIVE SUPPORT	02490000
	AGO	.MDSC	. GO SET STACTL, CHECK	02500000
.*				02510000
.MDBSN	ANOP		. BUFRCV, STACTL, NONCHK	02520000
&MFBR	SETB	1	. BUFFER RECEIVE SUPPORT	02530000
	AGO	.MDSN	. GO SET STACTL, NONCHK	02540000
.*				02550000
.MDTDC	ANOP		. XMITCTL, DIAL, CHECK	02560000
&MF7C	SETB	1	. TRANSMIT CONTROL SUPPORT	02570000
	AGO	.MDDC	. GO SET DIAL, CHECK	02580000
.*				02590000
.MDTDN	ANOP		. XMITCTL, DIAL, NONCHK	02600000
&MF7C	SETB	1	. TRANSMIT CONTROL SUPPORT	02610000
	AGO	.MDDN	. GO SET DIAL, NONCHK	02620000
.*				02630000
.MDDC	ANOP		. DIAL, CHECK	02640000
&MFSW	SETB	1	. DIAL SUPPORT	02650000
	AGO	.MDCK	. GO SET CHECK	02660000
.*				02670000
.MDDN	ANOP		. DIAL, NONCHK	02680000
&MFSW	SETB	1	. DIAL SUPPORT	02690000
	AGO	.MDNK	. GO SET NONCHK	02700000
.*				02710000
.MDSC	ANOP		. STACTL, CHECK	02720000
&MFSC	SETB	1	. STATION CONTROL SUPPORT	02730000
	AGO	.MDCK	. GO SET CHECKING	02740000
.*				02750000
.MDSN	ANOP		. STACTL, NONCHK	02760000
&MFSC	SETB	1	. STATION CONTROL SUPPORT	02770000
	AGO	.MDNK	. GO SET NON-CHECKING	02780000
.*				02790000
.MDCK	ANOP		. CHECKING TERMINAL	02800000
&MFCK	SETB	1	. CHECKING TERMINAL SUPPORT	02810000
	AGO	.MDEND		02820000
.*				02830000
.MDNK	ANOP		. NON-CHECKING TERMINAL	02840000
&MFNK	SETB	1	. NON-CHECKING TERMINAL SUPPORT	02850000
.*				02860000
.MDEND	ANOP		. END OPERAND 'TYPE'	02870000
.*				02880000
-----	XMCODE-CORR/PTTCEBCD/PTTCBCD	-----		02890000
.*				02900000
	AIF	(T'&XMCODE NE 'O').XM010	. SKIP IF XMCODE SPECIFIED	02910000
.*				02920000
*!320E	MISSING	'XMCODE' OPERAND -- MUST BE SPECIFIED		02930000
&TERR	SETB	1	. SET TERMINATION ERROR SWITCH	02940000
	AGO	.END		02950000
.*				02960000
.XM010	ANOP		. XMCODE SPECIFIED	02970000
	AIF	(&XMCODE EQ 'CORR').XMC	. SKIP IF XMCODE-CORR	02980000
	AIF	(&XMCODE EQ 'PTTCEBCD').XME	. SKIP IF XMCODE-PTTCEBCD	02990000

```

AIF (&XMCODE EQ 'PTTCBCD').XMB . SKIP IF XMCODE-PTTCBCD 03000000
.* 03010000
*!325E INVALID 'XMCODE' PARAMETER -- MUST BE CORR/PTTCBCD/PTTCBCD 03020000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03030000
AGO .END 03040000
.* 03050000
.XMC ANOP . XMCODE-CORR 03060000
AIF (&XCORR NE '1').XMERR . SKIP IF INVALID WITH TYPE 03070000
.* 03080000
&C2 SETC '&MXC'(2,3) 03090000
&MXC SETC '1&C2' 03100000
AGO .XMEND 03110000
.* 03120000
.XME ANOP . XMCODE-PTTCBCD 03130000
AIF (&XEBCD NE '1').XMERR . SKIP IF INVALID WITH TYPE 03140000
.* 03150000
AIF ('&TYPE'(1,4) EQ '1050').XME50 . SKIP IF 1050 PTTCEBCD 03160000
.* 03170000
&C1 SETC '&MXC'(1,1) 03180000
&C2 SETC '&MXC'(3,2) 03190000
&MXC SETC '&C1.1&C2' 03200000
&XM4E SETB 1 . SET 2740/1 PTTCEBCD PRESENT 03210000
AGO .XMEND 03220000
.* 03230000
.XME50 ANOP 03240000
&C1 SETC '&MXC'(1,3) 03250000
&MXC SETC '&C1.1' 03260000
AGO .XMEND 03270000
.* 03280000
.XMB ANOP . XMCODE-PTTCBCD 03290000
AIF (&XBCD NE '1').XMERR . SKIP IF INVALID WITH TYPE 03300000
&C1 SETC '&MXC'(1,2) 03310000
&C2 SETC '&MXC'(4,1) 03320000
&MXC SETC '&C1.1&C2' 03330000
AGO .XMEND 03340000
.* 03350000
.XMERR ANOP 03360000
*!330E XMCODE-&XMCODE NOT VALID FOR TERMINAL TYPE SPECIFIED 03370000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 03380000
AGO .END 03390000
.* 03400000
.XMEND ANOP . END OPERAND 'XMCODE' 03410000
.* 03420000
*----- SET STATEMENT SEQUENCE -----* 03430000
.* 03440000
.END ANOP 03450000
&SEQ SETA 7 . INDICATE $EMLD PROCESSED 03460000
.* 03470000
MEND 03480000

```

```

MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $EPLG 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 8, MODIFICATION LEVEL 0 OF 5702-SC1 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION FIRST PASS MACRO-INSTRUCTION -- DEFINE 00100000
.* PROGRAMMING LANGUAGE FOR WHICH SUPPORT IS TO BE INCLUDED. 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . LANG-COBOL/FORTRAN/ASSEM/RPGII 00150000
.* 00160000
.* NAME OF PROGRAMMING LANGUAGE TO BE SUPPORTED. THERE IS NO 00170000
.* DEFAULT. 00180000
.* 00190000
.* . PPUNIT-R1/F1/R2/F2 00200000
.* 00210000
.* UNIT ONTO WHICH THE SOURCE MODULES (MACROS) AND/OR RELOCATABLE 00220000
.* MODULES (SUBROUTINES) REQUIRED TO SUPPORT THE LANGUAGE ARE TO 00230000
.* BE PLACED BY GENERATION. 00240000
.* 00250000
***** 00260000
$EPLG &LANG-, &PPUNIT- 00270000
.* 00280000
GBLA &SEQ . SEQUENCE CONTROL 00290000
.* . MUST BE 2 OR 3 UPON ENTRY 00300000
.* . IS MADE 3 AFTER PROCESSING 00310000
GBLB &TERR . TERMINATION ERROR SWITCH 00320000
GBLC &PLG . LANGUAGES SUPPORTED 00330000
.* . 1ST CHAR -- COBOL 00340000
.* . 2ND CHAR -- FORTRAN 00350000
.* . 3RD CHAR -- ASSEMBLER 00360000
.* . 4TH CHAR -- RPG II 00370000
GBLC &UPCBL, &UPFOR, &UPASM, &UPRPG . UNIT FOR EACH PGM LANG 00380000
.* 00390000
LCLA &N . LOCAL VARIABLE FOR LENGTHS 00400000
LCLC &C1, &C2 . USED TO REBUILD LANG SUPPORT 00410000
.* 00420000
.*----- PARAMETER TRANSFORMATION TABLE -----* 00430000
.* 00440000
TABLE &PPUNIT . PROGRAM PREPARATION UNIT 00450000
R1 TABDF R1 00460000
F1 TABDF F1 00470000
R2 TABDF R2 00480000
F2 TABDF F2 00490000
' ' TABDF ?? 00500000
TABDF ** . ERROR PARAMETER 00510000
TEXT 00520000
.* 00530000
.*----- CHECK STATEMENT SEQUENCE -----* 00540000
.* 00550000
AIF (&SEQ EQ '2').SEQ1 . SKIP IF $EIOD, $EFAC BEFORE 00560000
AIF (&SEQ EQ '3').SEQOK . SKIP IF $EPLG BEFORE 00570000
.* 00580000
*!100E $EPLG STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR 00590000

```


&TERR	SETB	1	.	SET TERMINATION ERROR SWITCH	00600000
.	*				00610000
	AIF	(&PLG NE ' ').SEQOK	.	SKIP IF INITIALIZED	00620000
.	*				00630000
.SEQ1	ANOP		.	STMT IN SEQUENCE--1ST USE	00640000
&PLG	SETC	'0000'	.	INITIALIZE SUPPORT SWITCHES	00650000
.	*				00660000
.SEQOK	ANOP		.	STATEMENT IN SEQUENCE	00670000
.	*				00680000
.	*	----- LANG-COBOL/FORTRAN/ASSEM/RPGII -----	*		00690000
.	*				00700000
	AIF	(T'&LANG NE 'O').LG001	.		00710000
.	*				00720000
*!120E	MISSING	'LANG' OPERAND -- MUST BE SPECIFIED			00730000
&TERR	SETB	1	.	SET TERMINATION ERROR SWITCH	00740000
	AGO	.LGEND	.	SKIP REST OF 'LANG' PROCESSING	00750000
.	*				00760000
.LG001	ANOP		.	LANG OPERAND SPECIFIED	00770000
	AIF	(&LANG EQ 'COBOL').CBL	.	SKIP IF LANG-COBOL	00780000
	AIF	(&LANG EQ 'FORTRAN').FOR	.	SKIP IF LANG-FORTRAN	00790000
	AIF	(&LANG EQ 'ASSEM').ASM	.	SKIP IF LANG-ASSEM	00800000
	AIF	(&LANG EQ 'RPGII').RPG	.	SKIP IF LANG-RPGII	00810000
.	*				00820000
*!105E	INVALID	'LANG' PARAMETER -- MUST BE COBOL/RPGII/ASSEM/FORTRAN			00830000
&TERR	SETB	1	.	SET TERMINATION ERROR SWITCH	00840000
	AGO	.LGEND	.		00850000
.	*				00860000
.CBL	ANOP		.	LANG-COBOL	00870000
&UPCBL	SETC	'&PPUNIT'	.	PGM PREP UNIT FOR COBOL	00880000
.	*				00890000
	AIF	('&PLG'(1,1) EQ '1').DUERR	.	SKIP IF DUPLICATE LANG-COBOL	00900000
.	*				00910000
&C1	SETC	'1'			00920000
&C2	SETC	'&PLG'(2,3)			00930000
	AGO	.LGCOM			00940000
.	*				00950000
.FOR	ANOP		.	LANG-FORTRAN	00960000
&UPFOR	SETC	'&PPUNIT'	.	PGM PREP UNIT FOR FORTRAN	00970000
.	*				00980000
	AIF	('&PLG'(2,1) EQ '1').DUERR	.	SKIP IF DUPLIC LANG-FORTRAN	00990000
.	*				01000000
&C1	SETC	'&PLG'(1,1)			01010000
&C1	SETC	'&C1.1'			01020000
&C2	SETC	'&PLG'(3,2)			01030000
	AGO	.LGCOM			01040000
.	*				01050000
.ASM	ANOP		.	LANG-ASSEM	01060000
&UPASM	SETC	'&PPUNIT'	.	PGM PREP UNIT FOR ASSEMBLER	01070000
.	*				01080000
	AIF	('&PLG'(3,1) EQ '1').DUERR	.	SKIP IF DUPLICATE LANG-ASSEM	01090000
.	*				01100000
&C1	SETC	'&PLG'(1,2)			01110000
&C1	SETC	'&C1.1'			01120000
&C2	SETC	'&PLG'(4,1)			01130000
	AGO	.LGCOM			01140000
.	*				01150000
.RPG	ANOP		.	LANG-RPGII	01160000
&UPRPG	SETC	'&PPUNIT'	.	PGM PREP UNIT FOR RPG II	01170000
.	*				01180000
	AIF	('&PLG'(4,1) EQ '1').DUERR	.	SKIP IF DUPLICATE LANG-RPGII	01190000

```

.* 01200000
&C1 SETC '&PLG'(1,3) 01210000
&C2 SETC '1' 01220000
AGO .LGCOM 01230000
.DUERR ANOP . $EPLG STMT DUPLICATE FOR LANG 01240000
*!110E DUPLICATE $EPLG STATEMENT FOR &LANG LANGUAGE 01250000
AGO .LGEND . SKIP REST OF 'LANG' PROCESSING 01260000
.* 01270000
.LGCOM ANOP . COMMON FOR ALL LANGUAGES 01280000
&PLG SETC '&C1&C2' . REBUILD LANG SUPPORT SWITCHES 01290000
.* 01300000
.LGEND ANOP . END OPERAND 'LANG' 01310000
.* 01320000
.*----- PPUNIT-R1/F1/R2/F2 -----* 01330000
.* 01340000
AIF (&PPUNIT NE '??').LG500 . SKIP IF OPERAND SPECIFIED 01350000
.* 01360000
*!125E MISSING 'PPUNIT' OPERAND -- MUST BE SPECIFIED 01370000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 01380000
AGO .PPEND . SKIP REST OF 'PPUNIT' PROCESS 01390000
.* 01400000
.LG500 ANOP 01410000
AIF (&PPUNIT NE '**').PPEND . SKIP IF VALID PARAMETER 01420000
.* 01430000
*!115E INVALID 'PPUNIT' PARAMETER -- MUST BE R1/F1/R2/F2 01440000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 01450000
.* 01460000
.PPEND ANOP . END OPERAND 'PPUNIT' 01470000
.* 01480000
.*----- SET STATEMENT SEQUENCE -----* 01490000
.* 01500000
.END ANOP 01510000
&SEQ SETA 3 . INDICATE $EPLG PROCESSED 01520000
.* 01530000
MEND 01540000

```

```

MACRO                                00010000
*****                                00020000
*                                     * 00030000
* NAME: $ESEC                          * 00040000
*                                     * 00050000
* MODIFICATION LEVEL:  VERSION 0, MODIFICATION LEVEL 0  * 00060000
*                                     * 00070000
* FUNCTION:                             * 00080000
*                                     * 00090000
* . CCP GENERATION FIRST PASS MACRO INSTRUCTION -- SPECIFY * 00100000
*   SIGN-ON SECURITY SUPPORT IN USER'S VERSION OF CCP.   * 00110000
* . THIS GENERATION MACRO IS OPTIONAL -- IF OMITTED, *NO* * 00120000
*   SECURITY SUPPORT WILL BE GENERATED INTO THE USER'S CCP.* 00130000
*                                     * 00140000
* INPUT OPERANDS:                        * 00150000
*                                     * 00160000
* . SECURE-NO/CCP/USER                   * 00170000
*                                     * 00180000
* SPECIFY WHAT SECURITY CHECKING IS TO BE DONE AT SIGN-ON: * 00190000
*   NO -- NEITHER CCP PASSWORD CHECK NOR USER ROUTINE CHECK * 00200000
*   CCP -- CCP PASSWORD CHECKING          * 00210000
*   USER -- USER WILL SUPPLY HIS OWN ROUTINE (AND LUSI MUST BE * 00220000
*     SPECIFIED IN THIS MACRO-INSTRUCTION. * 00230000
*   DEFAULT IS NO.                       * 00240000
*                                     * 00250000
* . LUSI-NUMBER                          * 00260000
*                                     * 00270000
* THIS OPERAND IS SPECIFIED ONLY WHEN SECURE-USER IS SPECIFIED. * 00280000
* INDICATES NUMBER OF BYTES TO RESERVE IN LOAD MODULE '$CC4Z9' * 00290000
* AND IN MAIN STORAGE DURING CCP EXECUTION FOR USER SECURITY * 00300000
* DATA.                                 * 00310000
*                                     * 00320000
*****                                00330000
$ESEC &SECURE-NO,&LUSI-0              00340000
*                                     00350000
GBLB &TERR . TERMINATION ERROR SWITCH 00360000
GBLA &SEQ . SEQUENCE CONTROL:          00370000
*                                     . MUST BE 3 UPON ENTRY 00380000
*                                     . IS MADE 4 AFTER PROCESSING 00390000
GBLB &CPW,&UPW . CCP CHECKING / USER CHECKING 00400000
GBLA &LUS . LENGTH OF USER SECURITY DATA 00410000
TEXT                                     00420000
*                                     00430000
*----- CHECK STATEMENT SEQUENCE -----* 00440000
*                                     00450000
AIF (&SEQ EQ '3').SEQOK . SKIP IF IN SEQUENCE 00460000
*                                     00470000
*!150E $ESEC STATEMENT OUT OF SEQUENCE -- OR PRECEDING STATEMENT ERROR 00480000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00490000
&CPW SETB 0 . MAKE NULL PREVIOUS SPEC 00500000
&UPW SETB 0 . MAKE NULL PREVIOUS SPEC 00510000
*                                     00520000
.SEQOK ANOP . STATEMENT IN PROPER SEQUENCE 00530000
*                                     00540000
*----- SECURE-NO/CCP/USER              00550000
*                                     00560000
AIF (&SECURE EQ 'N').LUEQ0 . SKIP IF SECURE-N 00570000
AIF (&SECURE EQ 'NO').LUEQ0 . SKIP IF SECURE-NO 00580000
AIF (&SECURE EQ 'CCP').CCP . SKIP IF SECURE-CCP 00590000

```

```

AIF (&SECURE EQ 'USER').USER . SKIP IF SECURE-USER 00600000
.* 00610000
*!155E INVALID 'SECURE' PARAMETER -- MUST BE CCP/USER/NO 00620000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00630000
AGO .LUEQ0 . SKIP TO CHECK LUSI OPERAND 00640000
.* 00650000
.CCP ANOP . SECURE-CCP 00660000
&CPW SETB 1 00670000
.* 00680000
.LUEQ0 ANOP . CHECK LUSI-0 OR OMITTED 00690000
AIF (&LUSI EQ '0').SECND . SKIP IF 0 OR OMITTED 00700000
.* 00710000
*!160E INVALID 'LUSI' PARAMETER -- ONLY 0 IS VALID UNLESS SECURE-USER 00720000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00730000
AGO .SECND 00740000
.* 00750000
.USER ANOP . SECURE-USER 00760000
.* 00770000
.*----- LUSI-NUMBER -----* 00780000
.* 00790000
AIF (&LUSI NE '0').LUOK1 . SKIP IF VALUE SPECIFIED 00800000
.* 00810000
*!165E LUSI-0 SPECIFIED, OR 'LUSI' OPERAND OMITTED, WITH SECURE-USER 00820000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00830000
AGO .SECND 00840000
.* 00850000
.LUOK1 AIF (T'&LUSI EQ 'N').LUOK2 . SKIP IF VALUE IS NUMERIC 00860000
.* 00870000
*!170E INVALID 'LUSI' PARAMETER -- MUST BE NUMBER IN RANGE 1-4096 00880000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00890000
AGO .SECND 00900000
.* 00910000
.LUOK2 AIF (&LUSI LE '4096').LUOK . SKIP IF IN RANGE 1 - 4096 00920000
.* 00930000
*!170E INVALID 'LUSI' PARAMETER -- MUST BE NUMBER IN RANGE 1 - 4096 00940000
&TERR SETB 1 . SET TERMINATION ERROR SWITCH 00950000
AGO .SECND 00960000
.* 00970000
.LUOK ANOP . VALID LUSI PARAMETER 00980000
&LUS SETA &LUSI 00990000
&UPW SETB 1 01000000
.* 01010000
.*----- SET STATEMENT SEQUENCE -----* 01020000
.* 01030000
.SECND ANOP . END PROCESSING $ESEC STATEMENT 01040000
&SEQ SETA 4 . INDICATE $ESEC PROCESSED 01050000
.* 01060000
MEND 01070000

```


IAR	EQU	16	PROGRAM LEVEL IAR	00600000
PMR	EQU	48	PROGRAM MODE REGISTER	00610000
	SPACE	1		00620000
LVIIAR	EQU	X'80'	DPF INTERRUPT LEVEL IAR	00630000
LVMIAI	EQU	X'90'	MLTA INTERRUPT LEVEL IAR	00640000
LVBIAR	EQU	X'A0'	BSCA INTERRUPT LEVEL IAR	00650000
LVCIAI	EQU	X'C0'	CONSOLE INTERRUPT LEVEL IAR	00660000
.BRQ	AIF	(&BRQ EQ 'N').CCDE		00670000
	SPACE	2		00680000
* Q-BYTE EQUATES FOR BRANCH/NOP				00690000
NOP	EQU	X'07'	NO-OP	00700000
NOOP	EQU	X'80'	NO-OP	00710000
UNCON	EQU	X'FF'	ALWAYS BRANCH	00720000
BRNOP	EQU	X'80'	BRANCH NO-OP	00730000
BR	EQU	X'87'	ALWAYS BRANCH	00740000
BR97	EQU	X'97'	BRANCH AND RESET FALSE BIT	00750000
	SPACE	1		00760000
* EQUATES FOR MODIFIED FIELDS IN AN INSTRUCTION				00770000
	SPACE	1		00780000
#	EQU	X'00'	ANY FIELD	00790000
##	EQU	X'00'	SINGLE-BYTE FIELD	00800000
###	EQU	X'00'	ONE- OR TWO-BYTE FIELD	00810000
####	EQU	X'0000'	TWO-BYTE FIELD	00820000
.CCDE	AIF	(&CCDE EQ 'N').MODE		00830000
	SPACE	2		00840000
* CONDITION CODE EQUATES				00850000
ANY	EQU	X'80'	IF *ANY* CONDITION HOLDS	00860000
NONE	EQU	X'00'	IF *NO* CONDITION HOLDS	00870000
BOVFL	EQU	X'20'	BINARY OVERFLOW	00880000
FALSE	EQU	X'10'	FALSE	00890000
DOVFL	EQU	X'08'	DECIMAL OVERFLOW	00900000
HI	EQU	X'04'	FIRST OPERAND HIGH	00910000
LO	EQU	X'02'	FIRST OPERAND LOW	00920000
EQ	EQU	X'01'	OPERANDS EQUAL	00930000
.MODE	AIF	(&MODE EQ 'N').TRACE		00940000
PMRI12	EQU	112	OP2,OP1,I-CYCLE TRANSLATE	00950000
PMROP2	EQU	64	OPERAND 2 E-CYCLES TRANSLATION	00960000
PMROP1	EQU	32	OPERAND 1 E-CYCLES TRANSLATION	00970000
PMRINS	EQU	16	INSTRUCTION-CYCLES TRANSLATION	00980000
PMRPRV	EQU	8	PRIVELEGED	00990000
PMRPRT	EQU	2	STORAGE PROTECT	01000000
PMRINT	EQU	1	MASK INTERRUPTS	01010000
.TRACE	AIF	(&TRACE EQ 'N').END		01020000
	SPACE	2		01030000
* EQUATES FOR TYPES OF CCP TRACE ENTRIES				01040000
TTTRAN	EQU	X'E7'	TRACE ID FOR TRANSIENT CALLS	01050000
TTGETM	EQU	X'E8'	TRACE ID FOR GETMAIN	01060000
TTFREE	EQU	X'E9'	TRACE ID FOR FREEMAIN	01070000
TTMOPN	EQU	X'EA'	TRACE ID FOR TP CHECK ROUTINE	01080000
TTMSIO	EQU	X'EB'	TRACE ID FOR MLTA START IO	01090000
TTBSIO	EQU	X'FB'	TRACE ID FOR BSCA START IO	01100000
TTII	EQU	X'EC'	TRACE ID FOR \$CC4II	01110000
TTIS	EQU	X'ED'	TRACE ID FOR \$CC4IS	01120000
TTIIRT	EQU	X'EE'	RETURN FROM \$CC4II TO USER	01130000
TTDFEN	EQU	X'FC'	TRACE ID FOR ENTRY TO DFF TASK	01140000
TTDFEX	EQU	X'FE'	TRACE ID FOR EXIT FROM DFF TASK	01150000
.END	ANOP			01160000
	MEND			01170000

```

MACRO                                00010000
*****                                00020000
.* STATUS:  RELEASE 2                 00030000
.*                                     * 00040000
.* NAME:  $EPAS                        * 00050000
.*                                     * 00060000
.* FUNCTION:  DEFINE THE LABELS NEEDED TO REFERENCE THE FIELDS IN * 00070000
.* USER PROGRAM APPENDED STORAGE.     * 00080000
.*                                     * 00090000
.* INPUT OPERANDS:  NONE               * 00100000
*****                                00110000
$EPAS                                00120000
TEXT                                  00130000
* * * * *                             00140000
*                                     * 00150000
* EQUATES TO OFFSETS IN THE 'PROGRAM APPENDED STORAGE' AREA. * 00160000
* 01/08/74  MODEL 15                  * 00170000
*                                     * 00180000
*                                     * 00190000
* THE PAS IS DIVIDED UP INTO THREE MAIN STORAGE AREAS... * 00200000
* 1. PAS CONSTANTS AND WORK AREAS. THIS AREA CONTAINS * 00210000
* ALL THE INFORMATION PERTAENT TO A TASK. * 00220000
* THIS AREA STARTS ON A SECTOR BOUNDARY AND IS LESS * 00230000
* THAN A SECTOR IN LENGTH. * 00240000
*                                     * 00250000
* 2. FT AND TT ENTRIES. THIS AREA IMMEDIATELY FOLLOWS * 00260000
* THE PAS CONSTANTS AND CONTAINS THE 'FORMAT TABLE' * 00270000
* AND 'TERMINAL TABLE' (TT AND FT) ENTRIES. THIS AREA * 00280000
* WILL RUN UP TO THE FIRST BYTE OF THE NEXT AREA. * 00290000
*                                     * 00300000
* 3. 'FIELD DESCRIPTOR AREA' (FDT). THIS AREA WILL START * 00310000
* ON A SECTOR BOUNDARY AND ITS LENGTH WILL BE EVENLY * 00320000
* DIVISIBLE BY SECTORS. THE 'FDT' OF A FORMAT WILL BE * 00330000
* READ INTO THIS AREA FROM THE OBJECT LIBRARY ON DISK. * 00340000
*                                     * 00350000
* * * * *                             00360000
SPACE 2                               00370000
* * * * *                             00380000
* CONSTANT AND WORK AREA. * 00390000
* * * * *                             00400000
SPACE                                  00410000
PAS EQU 0 START OF PROGRAM APPENDED STORAG 00420000
PASITT EQU PAS+1 @ OF FIRST TERMINAL TABLE ENTRY. 00430000
* 2 (TT ENTRY) 00440000
SPACE                                  00450000
PASNFT EQU PASITT+2 @ OF NEXT AVAILABLE TT OR FORMAT 00460000
* 2 TABLE (FT) ENTRY. 00470000
SPACE                                  00480000
PASEFT EQU PASNFT+2 @ OF END OF AREA TO BUILD TT AND 00490000
* 2 FT ENTRIES. (SHOULD BE END OF 00500000
* PAS AREA. 00510000
SPACE                                  00520000
PASFDT EQU PASEFT+2 @ OF FDT AREA 00530000
* 2 00540000
SPACE                                  00550000
PASTID EQU PASFDT+1 TASK ID 00560000
* 1 00570000
SPACE                                  00580000
PASFDL EQU PASTID+1 NUMBER OF 256 BYTE BLOCKS OF 00590000

```

*			1	CORE NEEDED FOR LARGEST FDT.	00600000
	SPACE				00610000
PASCT@	EQU	PASFDL+2		@ OF TT ENTRY CURRENTLY PROCESS.	00620000
*			2		00630000
	SPACE				00640000
PASCTN	EQU	PASCT@+6		NAME OF TERMINAL CURRENTLY	00650000
*			6	BEING PROCESSED.	00660000
	SPACE				00670000
PASCF@	EQU	PASCTN+2		@ OF FT FOR FDT CURRENTLY IN PAS	00680000
*			2		00690000
	SPACE				00700000
PASFR@	EQU	PASCF@+2		RETURN POINT FROM DFF000	00710000
*			2		00720000
	SPACE				00730000
PASJR@	EQU	PASFR@+2		RETURN POINT FROM DFJ000	00740000
*			2		00750000
	SPACE				00760000
PASXR@	EQU	PASJR@+2		COMMON SAVE AREA FOR RETURN	00770000
*			2	ADDRESS FOR MISC ROUTINES.	00780000
	SPACE				00790000
PASRE@	EQU	PASXR@+2		@ OF RETURN POINT TO OP CODE	00800000
*			2	LOGIC AFTER RETURN FROM \$CC4CM.	00810000
	SPACE				00820000
PASOH@	EQU	PASRE@+2		@ OF OUTPUT HOLD AREA CURRENTLY	00830000
*			2	IN USE BY THIS TASK	00840000
	SPACE				00850000
PASOHL	EQU	PASOH@+2		LENGTH OF OUTPUT HOLD AREA	00860000
*			2	CURRENTLY USING.	00870000
	SPACE				00880000
PASOHE	EQU	PASOHL+2		END ADDR OF OHA	00890000
*			2		00900000
	SPACE				00910000
PASCCC	EQU	PASOHE+1		COPY CONTROL CHARACTER	00920000
*			1		00930000
	SPACE				00940000
PASTOD	EQU	PASCCC+2		'TO'DEVICE + CONTROL UNIT	00950000
*			2		00960000
	SPACE				00970000
PASFRD	EQU	PASTOD+2		'FROM'DEVICE ADDRESS FOR 'COPY'	00980000
*			2		00990000
PASID@	EQU	PASFRD		ADDRESS OF LINE ID IN CCCOM	01000000
*			2		01010000
	SPACE				01020000
PASFLG	EQU	PASFRD+1		FLAG BYTE	01030000
*			1		01040000
PASFRT	EQU	BIT0		1-'TO'TERMINAL	01050000
*				0-'FROM'TERMINAL	01060000
PASRST	EQU	BIT2		1-RESET MDT SET IN USERS WCC	01070000
PASBLK	EQU	BIT3		0-NO BLOCKING	01080000
*				1- BLOCKING	01090000
PASFTS	EQU	BIT4		FIRST TIME SWITCH FOR BLOCKING	01100000
*				1- FIRST TIME THROUGH	01110000
PASRUF	EQU	BIT5		READ UNDER FORMAT ACCEPT INPUT	01120000
*				1-ACCEPT INPUT AND TUBRUF ON	01130000
	SPACE	2			01140000
* * * * *				DISK IOB FOR DISK READS.	01150000
	SPACE				01160000
PASIOB	EQU	PASFLG+1		BEGINING OF IOB	01170000
	SPACE				01180000
PASDBE	EQU	PASIOB+28		END OF IOB	01190000

	SPACE	2		01200000
	SPACE	2		01210000
* * * * *	CCP PARAMETER LIST STORAGE AREA FROM USER PGM.			01220000
	SPACE			01230000
PASPL@	EQU	PASDBE+2	@ OF USER PARM LIST	01240000
*		2		01250000
	SPACE			01260000
PASCPL	EQU	PASPL+1	BEGIN OF PARM LIST	01270000
	SPACE			01280000
PASCPE	EQU	PASCPL+11	END OF PARM LIST(LAST 4 BYTES	01290000
*		16	ARE NOT SAVED.	01300000
	SPACE	2		01310000
PASIX1	EQU	PASCPE+2	SAVE ARE FOR \$CC4II XR1	01320000
*		2		01330000
	SPACE			01340000
PASIAR	EQU	PASIX1+2	SAVE AREA FOR \$CC4II RETURN @	01350000
*		2		01360000
	SPACE			01370000
PASOF@	EQU	PASIAR+2	@ IN FDT CURRENTLY WORKING AT	01380000
*		2	FOR OUTPUT BUILD	01390000
	SPACE			01400000
PASHAP	EQU	PASOF@+2	POINTER TO AREA IN OUTPUT HOLD	01410000
*		2	AREA CURRENTLY WORKING AT.	01420000
	SPACE			01430000
PASRAP	EQU	PASHAP+2	POINTER TO RECORD AREA CURRENTLY	01440000
*		2	WORKING AT.	01450000
	SPACE			01460000
PASTTE	EQU	PASRAP+2	@ OF EMPTY TT ENTRY	01470000
*		2		01480000
	SPACE			01490000
PASCFT	EQU	PASTTE+6	NAME OF FORMAT DESIRED FOR 'PUT'	01500000
*		6		01510000
	SPACE			01520000
PASTC	EQU	PASCFT+1	TERMINATION CODE	01530000
*		1		01540000
	SPACE			01550000
PASTTS	EQU	PASTC+2	TOTAL TEXT SIZE FOR OUTPUT	01560000
*		2		01570000
	SPACE			01580000
PASTTM	EQU	PASTTS+2	TOTAL TEXT MOVED SO FAR	01590000
*		2		01600000
	SPACE			01610000
PASLTH	EQU	PASTTM+2	TOTAL END POSITION OF TEXT IN	01620000
*		2	OHA	01630000
	SPACE			01640000
PASUDL	EQU	PASLTH+2	USER DEFINED DATA LNG FOR OUTPUT	01650000
*		2		01660000
	SPACE			01670000
PASWCC	EQU	PASUDL+1	WCC AND CCC SAVE AREA	01680000
*		1		01690000
	SPACE			01700000
PASTAR	EQU	PASWCC+2	SAVE AREA FOR ARR OF CALLER TO	01710000
*		2	A ROUTINE IN DFCR WHICH MIGHT	01720000
*			BE INTERRUPTED AND TASK SWITCHED	01730000
	SPACE			01740000
PASWRK	EQU	PASTAR+2	WORK AREA	01750000
*		2		01760000
	SPACE			01770000
PASWKZ	EQU	PASWRK+2	WORK AREA WITH HIGH-ORDER BYTE	01780000
*		2	ALWAYS CONTAINING ZERO	01790000

PASRFN	EQU	PASWKZ+4	READ UNDER FORMAT NAME	01800000
*			6 FORMAT NAME FOR RUF	01810000
	SPACE	2		01820000
PASEND	EQU	PASRFN+3	RESERVED AREA AND END OF 'PAS'.	01830000
PASL	EQU	PASEND+1	LENGTH OF PAS CONSTANTS	01840000
	EJECT			01850000
	MEND			01860000

```

MODULE-$LOGD , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$LOGD &LABL-ALL,&TCB@-N              00180000
TEXT                                  00190000
*          *** S Y S L O G O F F S E T S *** 00200000
SPACE                                  00210000
AIF (&LABL EQ 'OUT').GOUT             00220000
AIF (T'&LABL EQ 'O').GOPNS            00230000
AIF (&LABL EQ 'OPNS').GOPNS           00240000
$LGFUN EQU 0 FUNCTION CODE.           00250000
AIF (&LABL EQ 'MSG').GSEVE            00260000
AIF (&LABL NE 'ALL').GCC              00270000
.GSEVE ANOP                           00280000
$LGDS EQU 1 DEFAULT/SEVERITY.         00290000
SPACE                                  00300000
$LGSE1 EQU X'00' SEVERITY 1           00310000
$LGSE2 EQU X'01' * 2                  00320000
$LGSE4 EQU X'02' * 4                  00330000
$LGSE8 EQU X'03' * 8                  00340000
SPACE                                  00350000
.GCC ANOP                              00360000
$LGCC EQU 3 COMPONENT IDENTIFICATION. 00370000
$LGHH EQU 5 MESSAGE IDENTIFICATION - PART 1. 00380000
AIF (&LABL EQ 'MSG').GOPNS            00390000
AIF (&LABL NE 'ALL').GWTO             00400000
.GOPNS ANOP                            00410000
$LGDO EQU 6 RESPONSE + VALID OPTIONS  00420000
SPACE                                  00430000
$LGDE0 EQU X'80' DEFAULT OPTION OR RESPONSE OF 0 00440000
$LGDE1 EQU X'40' * 1                  00450000
$LGDE2 EQU X'20' * 2                  00460000
$LGDE3 EQU X'10' * 3                  00470000
$LGOP0 EQU X'08' ALLOWABLE OPTION 0   00480000
$LGOP1 EQU X'04' * 1                  00490000
$LGOP2 EQU X'02' * 2                  00500000
$LGOP3 EQU X'01' * 3                  00510000
AIF (T'&LABL EQ 'O').GXIT             00520000
AIF (&LABL EQ 'OPNS').GXIT            00530000
.GWTO ANOP                              00540000
AIF (&LABL EQ 'WTO').GWTOR            00550000
AIF (&LABL EQ 'WTOR').GWTOR           00560000
SPACE                                  00570000
$LGII EQU 8 MESSAGE IDENTIFICATION - PART 2. 00580000
$LGCL EQU 7 TEXT LENGTH - FORMAT C    00590000
$LGCAA EQU 9 TEXT ADDRESS - FORMAT C   00600000
$LGDL EQU 9 TEXT LENGTH - FORMAT D    00610000
$LGDAE EQU 11 TEXT ADDRESS - FORMAT D  00620000
AIF (&LABL EQ 'WTO').GWTOR            00630000
AIF (&LABL EQ 'WTOR').GWTOR           00640000
AIF (&LABL NE 'ALL').GXIT            00650000
.GWTOR ANOP                             00660000
$LGOTY EQU 6 WTO(R) TYPE FUNCTION.    00670000
$LGWTO EQU 6 WTO(R) TYPE FUNCTION.    00675000
AIF (&LABL EQ 'ALL').GALL             00680000
$LGII EQU 8 MESSAGE IDENTIFICATION - PART 2 00690000
.GALL ANOP                              00700000
$LGOL EQU 9 TEXT LENGTH - WTO(R)      00710000
$LGOAA EQU 11 TEXT ADDRESS - WTO(R)    00720000
* LEFTMOST BYTE DISPLACEMENT TO THE 00730000
$LGOEC EQU 12 ECB - WTO (1-CHAR RESP) 00740000

```

\$LGI1E	EQU	13	ADDRESS OF ECB (WTOW/WTOWH)	00750000
\$LGI2E	EQU	15	ADDRESS OF ECB (WTOW/WTOWH) / TCB	00760000
\$LGJ1E	EQU	16	ADDRESS OF ECB (WTORW)	00770000
\$LGJ2E	EQU	18	ADDRESS OF ECB (WTORW) / TCB	00780000
	AIF	(&LABL EQ 'WTO').GXIT		00790000
\$LGON	EQU	12	REPLY LENGTH - WTOR	00800000
\$LGO@@	EQU	14	REPLY ADDRESS - WTOR	00810000
	AIF	(&LABL NE 'ALL').GXIT		00820000
	SPACE			00830000
.GOUT	ANOP			00840000
\$LGFUT	EQU	0	OUTPUT ONLY FUNCTION CODE	00850000
\$LGPGE	EQU	1	* PAGE CONTROL OPTION	00860000
\$LGLNG	EQU	2	* BUFFER LENGTH	00870000
\$LGADR	EQU	4	* BUFFER ADDRESS	00880000
\$LGCPLE	EQU	5	* CARRIAGE POSITION	00890000
\$LGCPR	EQU	6	* RESERVED.	00900000
.GXIT	ANOP			00910000
	AIF	(&TCB@ NE 'Y').GEXIT		00920000
\$LGATC	EQU	8	TCB ADDRESS IN FORMAT A	00930000
\$LGBTC	EQU	10	* B	00940000
\$LGCTC	EQU	11	* C	00950000
\$LGDTC	EQU	13	* D	00960000
\$LGITC	EQU	13	* I (WTO)	00970000
\$LGJTC	EQU	16	* J (WTOR)	00980000
.GEXIT	ANOP			00990000
	MEND			01000000

```

MODULE-$EDSM , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$EDSM &NP-Y,&NC-Y,&RB-N,&SP-N        00020000
TEXT                                 00030000
AIF  (&NP EQ 'N').TNC                00040000
*****                               00050000
*                                     * 00060000
* COMMUNICATIONS AREA FOR A PROGRAM LEVEL * 00070000
*                                     * 00080000
*****                               00090000
*           START OF TASK CONTROL BLOCK 00100000
TCBB EQU 0 *                           00110000
TCBNXT EQU TCB+1 2 @ OF NEXT TCB ON QUEUE 00120000
TCBPRI EQU TCBNXT+1 1 TASK PRIORITY 00130000
TCBID EQU TCBPRI+1 1 TASK ID 00140000
SPACE                                 00150000
*****                               00160000
*                                     * 00170000
* NAME, PRIORITY AND ID OF THE SYSTEM TCBS: * 00180000
*                                     * 00190000
* NAME..... PRIORITY... ID.*.. * 00200000
* CONSOLE MANAGEMENT TASK F0 10 * 00210000
* SYSTEM ERROR TASK E0 20 * 00220000
* SPOOL SUPPORT TASK D0 30 * 00230000
* SPOOL TASK CE 32 * 00240000
* PROGRAM LEVEL 2 TASK C0 40 * 00250000
* PROGRAM LEVEL 1 TASK 60 A0 * 00260000
* SYSTEM WAIT TASK 01 FF * 00270000
*                                     * 00280000
* *-ID CHARACTERS 'A-Z' AND '0-9' ARE RESERVED FOR CCP TASKS * 00290000
*****                               00300000
SPACE                                 00310000
TCBFG1 EQU TCBID+1 1 FLAG BITS 00320000
* X'80'-CCP TASK 00330000
* X'40'-NON-CANCELABLE SYSTEM TASK 00340000
* X'08'-3 OPTION ONLY HALT PENDING 00350000
* X'04'-CANCEL NOT ALLOWED 00360000
* X'02'-CANCEL DEFERRED 00370000
* X'01'-EOJ IN PROCESS 00380000
SPACE                                 00390000
TCBFG2 EQU TCBFG1+1 1 FLAG BITS 00400000
* X'01'-ATT LOAD REQUESTED 00410000
SPACE                                 00420000
TCBDS1 EQU TCBFG2+1 1 DISPATCHABILITY BITS 00430000
* X'80'-NON-DISPATCHABLE 00440000
* X'40'-SUSPENDED BY CCP 00450000
* X'20'-CCP TERMINATE IN PROGRESS 00460000
* X'02'-WAITING FOR RESOURCES 00470000
* X'01'-WAITING FOR CORE 00480000
SPACE                                 00490000
TCBDS2 EQU TCBDS1+1 1 DISPATCHABILITY BITS 00500000
EJECT                                 00510000
***** PROGRAM REQUEST BLOCK ***** 00520000
TCBRBP EQU TCBDS2+2 2 @ OF ACTIVE REQUEST BLOCK 00530000
TCBRBF EQU TCBRBP+2 2 REQUEST BLOCK FLAGS 00540000
TCBTCB EQU TCBRBF+2 2 @ OF ASSOCIATED TCB 00550000
TCBIAR EQU TCBTCB+2 2 PROGRAM LEVEL IAR 00560000
TCBPMR EQU TCBBIAR+2 2 PROGRAM LEVEL PMR 00570000
TCBPSR EQU TCBPMR+2 2 PROGRAM LEVEL PSR 00580000
TCBXR2 EQU TCBPSR+2 2 PROGRAM LEVEL XR2 00590000

```

```

TCBXR1 EQU   TCBXR2+2          2   PROGRAM LEVEL XR1          00600000
TCBARR EQU   TCBXR1+2          2   PROGRAM LEVEL ARR          00610000
      SPACE                                00620000
*           END OF WAIT TASK TCB          00630000
      SPACE                                00640000
TCBCSN EQU   TCBARR+3          3   C/S/N FOR PROGRAM          00650000
TCBRIB EQU   TCBCSN+1          1   RIB VALUE                    00660000
TCBRSV EQU   TCBRIB+1          1   RIB SAVE AREA                00670000
TCBRS1 EQU   TCBRSV+6          6   RESERVED                        00680000
***** END OF PROGRAM REQUEST BLOCK ***** 00690000
      SPACE                                00700000
TCBMAP EQU   TCBRS1+2          2   @ OF CORE MAP FOR ASSIGN/FREE 00710000
TCBTIM EQU   TCBMAP+2          2   @ OF TIMER QUEUE ELEMENT      00720000
TCBEJE EQU   TCBTIM+2          2   EOJ EXIT @                    00730000
TCBEJC EQU   TCBEJE+1          1   EOJ COMPLETION CODE           00740000
*                                           X'80'-DUMP REQUESTED           00750000
*                                           --- X'40'-SECONDARY CODE SPECIFIED 00760000
*           USER CODES                   --- X'20'-HALT/SYSLOG(2,3 OPTION) 00770000
*                                           --- X'10'-OCC CANCEL             00780000
*                                           00790000
*                                           --- X'08'-SECONDARY CODE SPECIFIED 00800000
*           SYSTEM CODES                   X'04'-RESERVED                 00810000
*                                           X'02'-INSUFFICIENT CORE STORAGE 00820000
*                                           --- X'01'-PROGRAM CHECK          00830000
*                                           X'00'-NORMAL COMPLETION         00840000
      SPACE                                00850000
TCBEJS EQU   TCBEJC+1          1   EOJ SECONDARY COMPLETION CODE 00860000
*           - SYSTEM SECONDARY CODES -     00870000
*           X'01'-INVALID ENQU SVC         00880000
*           X'02'-INVALID DEQU SVC         00890000
*           X'03'-I/O PROTECT VIOLATION    00900000
*           X'04'-LOADING BELOW NPBEG      00910000
*           X'05'-LOADING ABOVE NPEND      00920000
*           X'06'-INVALID SETON/SETOF SVC  00930000
*           X'07'-EXIO TO INVALID DEVICE   00940000
      SPACE                                00950000
TCBRTC EQU   TCBEJS+2          2   @ OF RELATED TCB FOR SPOOL    00960000
TCBATT EQU   TCBRTC+32         32  ATT SAVE AREA                00970000
*   BYTES 25 THRU 32 OF THE ATT SAVE AREA ARE DEFINED AS FOLLOWS: 00980000
TCBHAV EQU   TCBATT-7          1   RESOURCE(S) OWNED            00990000
*           X'01'-SCHEDULER INTERLOCK      01000000
*           X'02'-SPOOL FILE INTERLOCK     01010000
*           X'04'-SYSLOG BUFFER INTERLOCK  01020000
      SPACE                                01030000
TCBRS3 EQU   TCBATT-4          3   RESERVED                        01040000
TCBPCA EQU   TCBATT-2          2   PROG CHECK ADDRESS REG       01050000
TCBPCS EQU   TCBATT            2   PROG CHECK STATUS REG        01060000
      EJECT                                01070000
*           START OF PROGRAM LEVEL COMM AREA 01080000
NPPRTZ EQU   TCBATT+1          1   PRINTER SIZE                  01090000
NPLPSZ EQU   NPPRTZ+1          1   LEFT TRACTOR PAGE SIZE        01100000
NPRPSZ EQU   NPLPSZ+1          1   RIGHT TRACTOR PAGE SIZE       01110000
NPHALT EQU   NPRPSZ+1          1   HALT/SYSLOG                   01120000
*           X'08'-NO HALT DISPLAY           01130000
*           OFF-YES                         01140000
*           ON-NO                           01150000
*           X'04'-DEFAULT MODE              01160000
*           X'03'-                           01170000
*           00-0 OPTION => SEVERITY=1     01180000
*           01-1 OPTION => SEVERITY=2     01190000

```

```

*          10-2 OPTION => SEVERITY=4 01200000
*          11-3 OPTION => SEVERITY=8 01210000
SPACE                                           01220000
NPJOB EQU  NPHALT+8      8  JOB NAME           01230000
NPSTEP EQU  NPJOB+8     8  STEP NAME          01240000
NPNAME EQU  NPSTEP+6    6  PROGRAM NAME       01250000
NPSPCM EQU  NPNAME+2    2  @ OF SPOOL COMM RELATED TO TA01260000
NPEOJ EQU  NPSPCM+1    1  END OF JOB BYTE     01270000
*          X'80'-RJE ACTIVE/RETURN TO NPBEG01280000
*          X'40'-NO IPL DISK ERRORS          01290000
*          X'20'-LIB MAINT BIT               01300000
*          X'10'-OCC DUMP TAKEN              01310000
*          X'08'-QUIESCE IN PROCESS AT EJ    01320000
*          X'01'-DO NOT CLOSE DTFS AT EOJ    01330000
SPACE                                           01340000
NPEOJ@ EQU  TCBEJE      1  END OF JOB RETURN @    01350000
NPSPOL EQU  NPEOJ+1    1  RESERVED FOR SPOOL  01360000
*          X'80'-TRAPPING I/O REQUEST        01370000
*          X'40'-SPOOL SUPPORTED THIS LEVEL01380000
*          X'20'-ON - END-OF-JOB            01390000
*          OFF - END-OF-STEP                 01400000
*          X'10'-START SPOOL REQUESTED       01410000
*          X'08'-STOP SPOOL REQUESTED        01420000
SPACE                                           01430000
NPDTF@ EQU  NPSPOL+2    2  @ OF LAST OPENED DTF  01440000
NPBEG EQU  NPDTF@+3    3  REAL PROGRAM BEGIN @    01450000
NPEND EQU  NPBEG+3     3  REAL PROGRAM END @      01460000
NPBEGL EQU  NPEND+2    2  LOGICAL PROGRAM BEGIN @  01470000
NPRLF EQU  NPBEGL+2    2  PROGRAM RELOCATION FACTOR  01480000
NPCYL EQU  NPRLF+2     2  C/S OF FIRST LOAD (OVERLAYS) 01490000
NPOLIB EQU  NPCYL+2    2  C/S OF PROG OBJ LIB     01500000
NPORLF EQU  NPOLIB+2   2  OVERLAY RELOCATION FACTOR  01510000
NPTXT EQU  NPORLF+2    2  OVERLAY TEXT @         01520000
*          END OF SPOOL SUPPORT COMMON AREA  01530000
AIF (&SP EQ 'Y').TNC                          01540000
NPQ EQU  NPTXT+1      1  PROGRAM Q BYTE          01550000
NPUPSI EQU  NPQ+1     1  UPSI SWITCH            01560000
NPATTR EQU  NPUPSI+1  1  PROGRAM ATTRIBUTE BYTES  01570000
*          X'80'-DEDICATED PROGRAM RUNNING  01580000
NPDATE EQU  NPATTR+6   6  PROGRAM DATE            01590000
NPSTP# EQU  NPDATE+1   1  STEPNAME COUNTER        01600000
NPSYSI EQU  NPSTP#+3   3  SYSIN INDICATOR         01610000
*          C/S/DEVICE INFO                   01620000
*          X'01'-MFCU1                        01630000
*          X'02'-MFCU2                        01640000
*          X'83'-MFCM1 -                      01650000
*          X'84'-MFCM2 - 8X-80 COL           01660000
*          X'85'-1442 - DEVICES              01670000
*          X'86'-2501 -                      01680000
*          X'10'-CONSOLE/KYBD                 01690000
*          X'20'-RESERVED                     01700000
*          X'21'-RESERVED                     01710000
SPACE                                           01720000
NPSCH1 EQU  NPSYSI+1  1  READER/INTERPRETER SWITCHES 01730000
*          X'80'-// DATE RECEIVED(INTRA)     01740000
*          X'40'-// COMPILE RECEIVED          01750000
*          X'20'-// SWITCH RECEIVED           01760000
*          X'10'-PROCEDURE                    01770000
*          X'08'-OVERRIDE                      01780000
*          X'04'-INTRA STEP MODE              01790000

```

*			X'02'-INTER STEP MODE	01800000
*			X'01'-IPL MODE	01810000
	SPACE			01820000
NPSCH2	EQU	NPSCH1+1	1 SCHEDULER SWITCHES	01830000
*			X'80'-CONTINUATION	01840000
*			X'40'-UTIL CTL CARDS IN JSWA	01850000
*			X'20'-SLASH AMPERSAND READ	01860000
*			X'10'-FILE STMT RECEIVED	01870000
*			X'08'-EOS HALT INDICATOR	01880000
*			ON-ISSUE EOS MESSAGE	01890000
*			OFF-DON'T ISSUE EOS MSG	01900000
*			X'04'-FLUSH	01910000
*			X'02'-IMMEDIATE CANCEL	01920000
*			X'01'-CONTROLLED CANCEL	01930000
	SPACE			01940000
NPSCH3	EQU	NPSCH2+1	1 SCHEDULER SWITCHES	01950000
*			X'80'-TAG SORT REQUIRED(5444)	01960000
*			X'40'-AVAILABLE	01970000
*			X'20'-SOURCE REQUIRED	01980000
*			X'10'-LOG STATUS	01990000
*			ON-LOG TO CRT	02000000
*			OFF-LOG TO 3284 OR 1403	02010000
*			X'08'-MVF FILE ALLOCATED	02020000
*			X'04'-ADDITIONAL PROC STMT	02030000
*			X'02'-1ST 'LOAD/RUN'JOB READ	02040000
*			X'01'-PGM LEVEL	02050000
*			OFF-LEVEL 1	02060000
*			ON-LEVEL 2	02070000
	SPACE			02080000
NPLEVL	EQU	NPSCH3		02090000
NPOBJQ	EQU	NPSCH3+1	1 OBJECT DECK OUTPUT Q	02100000
NPBPSD	EQU	NPOBJQ+1	1 AVAIL STATUS OF SYSIN DEVICES	02110000
*			X'80'-MFCU	02120000
*			X'40'-MFCM	02130000
*			X'20'-1442	02140000
*			X'10'-2501	02150000
*			X'08'-CRT	02160000
*			X'04'-RESERVED	02170000
*			X'02'-RESERVED	02180000
*			X'01'-NESTED PROCEDURE	02190000
	SPACE			02200000
NPSCH4	EQU	NPBPSD+1	1 SCHEDULER SWITCHES	02210000
*			X'80'-OPERATION BIT	02220000
*			X'40'-// IMAGE RECEIVED	02230000
*			X'20'-// PRINTER RECEIVED	02240000
*			X'10'-F1 NEEDED FOR ALLOC	02250000
*			X'08'-R1 NEEDED FOR ALLOC	02260000
*			X'04'-F2 NEEDED FOR ALLOC	02270000
*			X'02'-R2 NEEDED FOR ALLOC	02280000
*			-TAPE DTFS ENCOUNTERED IN	02290000
*			RESOURCE ALLOCATE	02300000
*			X'01'-EOJ SCHED BIT	02310000
	SPACE			02320000
NPSCH5	EQU	NPSCH4+1	1 SCHED SWITCHES	02330000
*			X'80'-ROLL-IN PENDING	02340000
*			X'40'-ROLL-OUT REQUESTED	02350000
*			X'20'-ROLL-OUT HANDLED	02360000
*			X'10'-I-TYPE PROGRAM EXECUTING	02370000
*			X'08'-RESERVED FOR CCP	02380000
*			X'04'-TAPE FILE CARD REC'D	02390000

*		X'02'-CORE ROLLED OUT	02400000	
*		X'01'-UNIT RECORD INDICATOR FOR	02410000	
*		INQUIRY	02420000	
	SPACE		02430000	
NPSCH6	EQU	NPSCH5+1	1 SCHEDULER SWITCHES	02440000
*		X'80'-LOAD * RUNNING THIS LEVEL	02450000	
*		X'40'-TEMP END-OF-STEP HALT	02460000	
*		ON-ISSUE EOS MESSAGE	02470000	
*		OFF-DON'T ISSUE EOS MSG	02480000	
*		X'20'-CHANGE LINE 1 TO 2	02490000	
*		X'10'-CHANGE LINE 2 TO 1	02500000	
*		X'08'-MFCU FILE PREV OPENED	02510000	
*		X'04'-/. READ	02520000	
*		X'02'-PROCESSING MODE	02530000	
*		ON-JOB MODE	02540000	
*		OFF-STEP MODE	02550000	
*		X'01'-MFCU FILE PREV OPENED	02560000	
	SPACE		02570000	
NPSCH7	EQU	NPSCH6+1	1 SCHED SWITCHES	02580000
*		X'80'-JOB CARD REC'D	02590000	
*		X'40'-RUN CARD REC'D	02600000	
*		X'20'-READ-AHEAD DONE	02610000	
*		X'10'-EJ TO BE HANDLED	02620000	
*		X'08'-FLUSH IS COMPLETE	02630000	
*		X'04'-DO NOT HALT IF ERROR	02640000	
*		X'02'-FLUSH FROM 3 OPTION	02650000	
*		X'01'-FLUSH FROM 2 OPTION	02660000	
	SPACE		02670000	
NPSCH8	EQU	NPSCH7+1	1 SCHED SWITCHES	02680000
*		X'80'-TAG SORT FOR 5445	02690000	
*		X'40'-CONSOLE IN USE BY DM	02700000	
*		X'20'-VALID FILES CARDS REC'D	02710000	
*		X'10'-SHARED I/O PROGRAM	02720000	
*		X'08'-I OR B TYPE PROGRAM	02730000	
*		X'04'-ALLOCATE UNSUCCESSFUL	02740000	
*		X'02'-MAXIMUM REQUEST MET	02750000	
*		X'01'-MINIMUM REQUEST MET	02760000	
	SPACE		02770000	
NPSCH9	EQU	NPSCH8+1	1 DISK LOG UNIT ID FOR EOJ/	02780000
*		DEALLOCATE	02790000	
NPSCHA	EQU	NPSCH9+1	1 SCHEDULER SWITCHES	02800000
*		X'80'-INAT TO INDS F1 SORT CON	02810000	
*		X'40'-SPLIT CYLINDER SUB ALLOC	02820000	
*		X'20'-CALL TAPE DLOG AT EJ	02830000	
*		X'10'-RESERVED	02840000	
*		X'08'-ACTIVE FILE ON T1	02850000	
*		X'04'-ACTIVE FILE ON T2	02860000	
*		X'02'-ACTIVE FILE ON T3	02870000	
*		X'01'-ACTIVE FILE ON T4	02880000	
	SPACE		02890000	
NPSCHB	EQU	NPSCHA+1	1 SCHEDULER BYTE	02900000
*		X'80'-CHECKPOINT PGM EXECUTING	02910000	
*		X'40'-5445 FILE CARD PRESENT	02920000	
*		X'20'-DEFERRED ALLOCATE REQUEST	02930000	
*		X'10'-DISK DTFS ENCOUNTERED IN	02940000	
*		RESOURCE ALLOCATE	02950000	
*		X'08'-NEW FILE ALLOCATED ON D1	02960000	
*		X'04'-NEW FILE ALLOCATED ON D2	02970000	
*		X'02'-NEW FILE ALLOCATED ON D3	02980000	
*		X'01'-NEW FILE ALLOCATED ON D4	02990000	

SPACE				03000000
NPMANT EQU	NPSCHB+1	1	LIBRARY MAINTENANCE	03010000
*			X'80'-SYSPUNCH INDICATOR	03020000
*			X'40'-RUN \$\$OXRF ON R1	03030000
*			X'20'-RUN \$\$OXRF ON F1	03040000
*			X'10'-RUN \$\$OXRF ON R2	03050000
*			X'08'-RUN \$\$OXRF ON F2	03060000
*			X'04'-CONTROL CARD FOUND	03070000
*			X'02'-UNUSED	03080000
*			X'01'-PICKEREL RUNNING THIS LVL	03090000
SPACE				03100000
NPSRAA EQU	NPMANT+2	2	@ OF SCHED READ-AHEAD AREA	03110000
NPMPFL EQU	NPSRAA+1	1	MATRIX PRINTER FORMS LENGTH	03120000
NPLNK@ EQU	NPMPFL+1	1	LEFT BYTE OF LKED START @	03130000
SPACE				03140000
NPUNCH EQU	NPLNK@+1	1	SYSPUNCH ID	03150000
*			X'01'-MFCU1	03160000
*			X'02'-MFCU2	03170000
*			X'81'-1442	03180000
*			X'82'-MFCM1 8X-80 COL DEVICE	03190000
*			X'84'-MFCM2	03200000
SPACE				03210000
NPSYSP EQU	NPUNCH+1	1	SYSPRINT ID	03220000
*			X'01'-1403	03230000
*			X'02'-3284	03240000
SPACE				03250000
NPUTIL EQU	NPSYSP+1	1	UTILITY INTERLOCK	03260000
*			X'80'-X'08'-F2 . ALLOCATED	03270000
*			X'40'-X'04'-R2 . OR	03280000
*			X'20'-X'02'-F1 . IN	03290000
*			X'10',X'01'-R1 . USE	03300000
SPACE				03310000
NPUTL1 EQU	NPUTIL+1	1	TAPE INTERLOCK	03320000
*			X'80'-T1 IN USE	03330000
*			X'40'-T2 IN USE	03340000
*			X'20'-T3 IN USE	03350000
*			X'10'-T4 IN USE	03360000
*			X'08'-LIBRARY USAGE INTERLOCK	03370000
*			X'04'-OPEN/CLOSE/EOV HAS SCHED	03380000
*			INTERLOCK	03390000
*			X'02'-ALLOCATE HAS SCHED INTERLO	03400000
SPACE				03410000
NPUTL2 EQU	NPUTL1+1	1	5445 INTERLOCK	03420000
*			X'80'-X'08'-D4 . ALLOCATED	03430000
*			X'40'-X'04'-D3 . OR	03440000
*			X'20'-X'02'-D2 . IN	03450000
*			X'10'-X'01'-D1 . USE	03460000
SPACE				03470000
NPSCHC EQU	NPUTL2+2	2	@ OF START OF TAPE F1S (SD)	03480000
NPSCHD EQU	NPSCHC+2	2	@ OF START OF 5444 F1S (SD)	03490000
NPSCHE EQU	NPSCHD+2	2	@ OF START OF 5445 F1S (SD)	03500000
NPSCHF EQU	NPSCHE+2	2	@ OF START OF UNIT RECORD F1S	03510000
NPSCHG EQU	NPSCHF+2	2	@ OF NEXT AVAIL F1 (SD)	03520000
NPOCAF EQU	NPSCHG+2	2	@ OF DM A/F SPACE	03530000
NPSPLZ EQU	NPOCAF+1	1	SYSPRINT LINE SIZE	03540000
NPSPFL EQU	NPSPLZ+1	1	SYSPRINT FORMS LENGTH	03550000
NPECOM EQU	NPSPFL	-	END OF PROG LEVEL COMM AREA	03560000
EJECT				03570000
.TNC ANOP				03580000
AIF (&NC EQ 'N').TRB				03590000

```

*****
*
*   SYSTEM COMMUNICATIONS AREA
*
*****
SPACE 2
NCC EQU 0 *
NCPL1 EQU NCC+1 2 @ OF PROG LEVEL 1 COMM
NCPL2 EQU NCPL1+2 2 @ OF PROG LEVEL 2 COMM
NCTCB EQU NCPL2+2 2 @ OF HIGHEST PRIORITY TCB
NCXTAB EQU NCTCB+2 2 @ OF XSNT SCHEDULER TABLE
NC@NEW EQU NCXTAB+2 2 @ OF NEW AND OLD
* OLD=@ OF ACTIVE TCB
NCDSP1 EQU NC@NEW+1 1 TASK SUPERVISION BITS
* X'80'-TRACE ACTIVE
* X'40'-TRACE SUSPENDED
SPACE
NCDSP2 EQU NCDSP1+1 1 TASK SUPERVISION BITS
* X'08'-TASK(S) WAITING FOR
* RESOURCES
* X'04'-TRANSIENT AREA NOT
* REFRESHABLE
* X'02'-HIGHER PRIORITY TASKS ARE
* WAITING ON TRANSIENT AREA
* X'01'-TASK(S) WAITING FOR CORE
SPACE
NCSGEN EQU NCDSP2+1 - SYSTEM USAGE
* X'80'-DO NOT UPDATE SIO CTRS
* X'02'-SYSTEM MAINTENANCE
* X'01'-SYSTEM GENERATION
SPACE
NCAFML EQU NCSGEN+1 1 ASIGN/FREE MASK LENGTH
NCSLOG EQU NCAFML+3 3 SYSLOG INDICATOR C/S/DEV INFO
* X'00'-CONSOLE
* X'80'-3284
* X'40'-1403
SPACE
NCSWRK EQU NCSLOG+2 2 C/S OF SWA
NCSYSQ EQU NCSWRK+1 1 Q OF SYSTEM PACK
NCOLIB EQU NCSYSQ+2 2 C/S OF SYSTEM OBJECT LIB
NCDATE EQU NCOLIB+6 6 SYSTEM DATE
NCSCH1 EQU NCDATE+1 1 SCHEDULER SWITCHES
* X'80'-LOG STATUS
* ON-LOG TO CRT
* OFF-LOG TO 1403 OR 3284
* X'40'-SYSTEM DATE RECEIVED
* X'10'-SKD INTERLOCK PL1
* X'08'-SKD INTERLOCK PL2
* X'04'-DATE FORMAT
* OFF-MMDDY DOMESTIC
* ON-DDMMYY WORLD TRADE
* X'03'-5444 CONFIG
* 00-F1,R1
* 01-F1,R1,R2
* 11-F1,R1,R2,F2
SPACE
NCSMV1 EQU NCSCH1+1 1 DM/SKD SWITCHES
* X'80'-IPL SUCCESSFUL
* X'40'-INPUT FOR I TYPE
* X'20'-SYSLOG INTERLOCK-P1

```

```

03600000
* 03610000
* 03620000
* 03630000
03640000
03650000
03660000
03670000
03680000
03690000
03700000
03710000
03720000
03730000
03740000
03750000
03760000
03770000
03780000
03790000
03800000
03810000
03820000
03830000
03840000
03850000
03860000
03870000
03880000
03890000
03900000
03910000
03920000
03930000
03940000
03950000
03960000
03970000
03980000
03990000
04000000
04010000
04020000
04030000
04040000
04050000
04060000
04070000
04080000
04090000
04100000
04110000
04120000
04130000
04140000
04150000
04160000
04170000
04180000
04190000

```

```

*          X'10'-SYSLOG INTERLOCK-P2          04200000
*          - X'08'-OFFLINE MVF ON R1          04210000
*          - X'04'-OFFLINE MVF ON R2          04220000
*          PL 1 - X'02'-OTHER FILE ON R1      04230000
*          - X'01'-OTHER FILE ON R2          04240000
          SPACE                               04250000
NCSMV2 EQU  NCSMV1+1                          1  DM/SKD SWITCHES 04260000
*          X'80'-SPOOL IS ACTIVE              04270000
*          X'40'-SYSTEM IPL MODE              04280000
*          X'20'-SPOOL SUPPORTED              04290000
*          - X'08'-OFFLINE MVF ON R1          04300000
*          PL 2 - X'04'-OFFLINE MVF ON R2      04310000
*          - X'02'-OTHER FILE ON R1          04320000
*          - X'01'-OTHER FILE ON R2          04330000
          SPACE                               04340000
NCSCH EQU  NCSMV2+1                          1  SKD BYTE       04350000
*          X'80'-INQUIRY SUPPORTED            04360000
*          X'40'-CCP SUPPORTED                04370000
*          X'20'-TIMER SUPPORTED              04380000
          SPACE 2                              04390000
NCDSKQ EQU  NCSCH+2                          2  @ OF 5444 QUEUE 04400000
NCDSK5 EQU  NCDSKQ+2                        2  @ OF 5445 QUEUE 04410000
NCTAPQ EQU  NCDSK5+2                        2  @ OF TAPE QUEUE 04420000
NCURQ EQU  NCTAPQ+2                        2  @ OF FIRST UNIT RECORD QUEUE 04430000
NCETQ@ EQU  NCURQ+2                        2  @ OF SYSTEM ERROR TASK QUEUE 04440000
NCCMTQ EQU  NCETQ@+2                        2  @ OF CONSOLE MGT COMM AREA 04450000
          SPACE                               04460000
NRCRSS EQU  NCCMTQ+3                        3  ROLL/OUT C/S/N 04470000
NCCONF EQU  NRCRSS+1                        1  5445 AND TAPE CONFIG 04480000
*          X'80'-RESERVED                     04490000
*          X'40'-RESERVED                     04500000
*          X'38' - 100 - D1                   04510000
*          101 - D1,D2                       04520000
*          110 - D1,D2,D3                    04530000
*          111 - D1,D2,D3,D4                 04540000
*          X'07' - 100 - T1                   04550000
*          101 - T1,T2                       04560000
*          110 - T1,T2,T3                    04570000
*          111 - T1,T2,T3,T4                 04580000
          SPACE                               04590000
NCSCH3 EQU  NCCONF+1                        1  SCHEDULER BYTE 04600000
NCSCH4 EQU  NCSCH3+1                        1  SCHEDULER BYTE 04610000
NCRQE EQU  NCSCH4+2                        2  @ OF RQE TABLE 04620000
NCEXTR EQU  NCRQE+1                        1  EXTRA BYTE     04630000
NCSMV3 EQU  NCEXTR+1                        1  5445 DM MVF INTERLOCK 04640000
*          PGM LEVEL 1                       04650000
*          X'80'-MULTIVOLUME FILE ON D1      04660000
*          X'40'-MULTIVOLUME FILE ON D2      04670000
*          X'20'-MULTIVOLUME FILE ON D3      04680000
*          X'10'-MULTIVOLUME FILE ON D4      04690000
*          X'08'-OTHER TYPE FILE ON D1      04700000
*          X'04'-OTHER TYPE FILE ON D2      04710000
*          X'02'-OTHER TYPE FILE ON D3      04720000
*          X'01'-OTHER TYPE FILE ON D4      04730000
          SPACE                               04740000
NCSMV4 EQU  NCSMV3+1                        1  5445 DM MVF INTERLOCK 04750000
*          PGM LEVEL 2                       04760000
*          X'80'-MULTIVOLUME FILE ON D1      04770000
*          X'40'-MULTIVOLUME FILE ON D2      04780000
*          X'20'-MULTIVOLUME FILE ON D3      04790000

```

```

*          X'10'-MULTIVOLUME FILE ON D4      04800000
*          X'08'-OTHER TYPE FILE ON D1       04810000
*          X'04'-OTHER TYPE FILE ON D2       04820000
*          X'02'-OTHER TYPE FILE ON D3       04830000
*          X'01'-OTHER TYPE FILE ON D4       04840000
          SPACE                               04850000
NCMVT1 EQU  NCSMV4+1      4  TAPE MFV SUPPORT 04860000
NCMVT2 EQU  NCMVT1+4      4  TAPE MFV SUPPORT 04870000
NCMVT3 EQU  NCMVT2+4      4  TAPE MFV SUPPORT 04880000
NCMVT4 EQU  NCMVT3+4      4  TAPE MFV SUPPORT 04890000
NCPRTB EQU  NCMVT4+5      2  @ OF 133-BYTE SYSLOG BUFR 04900000
NCCNFG EQU  NCPRTB+1     17  START OF 17-BYTE UR CONFIG 04910000
NCUTL1 EQU  NCCNFG+17     1  RESERVED FOR SPOOL 04920000
*          X'80'-SPOOL USING D1              04930000
*          X'40'-SPOOL USING D2              04940000
*          X'20'-SPOOL USING D3              04950000
*          X'10'-SPOOL USING D4              04960000
          SPACE                               04970000
NCSIPT EQU  NCUTL1+1      1  SPOOL INPUT DEVICE 04980000
NCSprt EQU  NCSIPT+1      1  SPOOL PRINT DEVICE 04990000
NCSPCH EQU  NCSprt+1      1  SPOOL PUNCH DEVICE 05000000
NCSPVN EQU  NCSPCH+2      2  SUPERVISOR END @ 05010000
NCPFkt EQU  NCSPVN+2      2  @ OF PROG FUNCTION KEY TABLE 05020000
NCMPSZ EQU  NCPFkt+1      1  MATRIX PRINTER SIZE 05030000
NCMPLC EQU  NCMPSZ+1      1  MATRIX PRINTER LINE COUNT 05040000
NCTMRQ EQU  NCMPLC+2      2  @ OF TIMER QUEUE 05050000
NCAEND EQU  NCTMRQ       LAST BYTE OF SCA 05060000
          EJECT                               05070000
.TRB      ANOP                               05080000
          AIF (&RB NE 'Y').COM              05090000
*****                                       05100000
*          *                               05110000
*          REQUEST BLOCK EQUATES           * 05120000
*          *                               * 05130000
*****                                       05140000
          SPACE 2                             05150000
RBB      EQU  0                               * 05160000
RBRBP   EQU  RBB+1                           2  POINTER TO PREVIOUS RB 05170000
RBFLG1  EQU  RBRBP+1                           1  RB FLAG BITS 05180000
*          X'80'-RB WAITING ON ECB          05190000
          SPACE                               05200000
RBFLG2  EQU  RBFLG1+1                           1  RB TYPE 05210000
*          X'80'-TA REQUIRED (TRB)           05220000
*          X'40'-REFRESH (RRB)             05230000
*          X'04'-CCP (CRB)                 05240000
*          X'02'-SUSPEND (SRB)             05250000
*          X'01'-LOADER (LRB)              05260000
          SPACE                               05270000
RBTCB   EQU  RBFLG2+2                           2  @ OF TCB 05280000
RBIAR   EQU  RBTCB+2                           2  PROGRAM LEVEL IAR 05290000
RBPMP   EQU  RBIAR+2                           2  PROGRAM LEVEL PMR 05300000
RBPSR   EQU  RBPMP+2                           2  PROGRAM LEVEL PSR 05310000
RBXR2   EQU  RBPSR+2                           2  PROGRAM LEVEL XR2 05320000
RBXR1   EQU  RBXR2+2                           2  PROGRAM LEVEL XR1 05330000
RBARR   EQU  RBXR1+2                           2  PROGRAM LEVEL ARR 05340000
RBCSN   EQU  RBARR+3                           3  PROGRAM LEVEL C/S/N 05350000
RBRIB   EQU  RBCSN+1                           1  PROGRAM LEVEL RIB 05360000
RBRSAV  EQU  RBRIB+1                           1  RIB SAVE AREA 05370000
RBRES   EQU  RBRSAV+6                          6  RESERVED 05380000
RBWORK  EQU  RBRES+1                          35  START OF WORK AREA-35 BYTES 05390000

```

```

EJECT 05400000
.COM ANOP 05410000
***** 05420000
* * 05430000
* COMMON EQUATES * 05440000
* * 05450000
***** 05460000
NCENTR EQU 4 'BRANCH TO 4' 05470000
NCSYS@ EQU X'0011' @ OF POINTER TO SYSTEM COMM 05480000
NCMSVA EQU X'001A' @ OF MFCU PERM HISTORY TABLE 05490000
NCTRAC EQU X'002A' @ OF POINTER TO TRACE TABLE 05500000
NCTCB@ EQU X'002E' @ OF POINTER TO ACTIVE TCB (OLD) 05510000
NCTERM EQU X'0031' @ OF POINTER TO ABTERM ROUTINE 05520000
NCSVCJ EQU X'0032' @ OF AN SVC TO EOJ 05530000
NCSVCE EQU X'0036' @ OF AN SVC EXIT 05540000
NCTRAP EQU X'003B' @ OF POINTER TO TRAP TABLE 05550000
NCPENT EQU X'003F' @ OF PONTER TO POST ENTRY RTN 05560000
NCDENT EQU X'0041' @ OF POINTER TO DISP ENTRY RTN 05570000
NCCCP EQU X'0043' @ OF CCP SVC INTERCEPT ADDRESS 05580000
NCXTB@ EQU X'0045' @ OF CCP TCB AREA START ADDRESS 05590000
NCXTE@ EQU X'0047' @ OF CCP TCB AREA END ADDRESS 05600000
NCCECB EQU X'0048' @ OF CCP COMMAND PROCESSOR ECB 05610000
NCHIMG EQU X'0700' @ OF CHAIN IMAGE 05620000
NCSBUF EQU X'077C' @ OF SYSLOG PRINT BUFFER. 05630000
NCPCHK EQU X'0100' @ OF PROGRAM CHECK SAVE AREA 05640000
NCTA@ EQU X'0800' @ OF TRANSIENT AREA 05650000
NCL2PR EQU X'C0' LEVEL 2 PRIORITY 05660000
MEND 05670000

```

```

MACRO                                00010000
*****                               00020000
.* STATUS:  RELEASE 3                * 00030000
.*                                  * 00040000
.* NAME:  $ECOM                      * 00050000
.*                                  * 00060000
.* FUNCTION:  DEFINE THE LOCATIONS OF DATA AREAS WITHIN CCP COMMON. * 00070000
.* WHERE APPROPRIATE, DEFINE THE BIT SIGNIFICANCE FOR THOSE DATA * 00080000
.* AREAS.  DEFINE THE LENGTHS AND LOCATIONS OF FIELDS IN * 00090000
.* THE FOLLOWING WORK AREAS:  COMMAND PROCESSOR, ALLOCATION, * 00100000
.* TERMINATION, COMMUNICATIONS MANAGEMENT. * 00110000
.*                                  * 00120000
.* INPUT OPERANDS: * 00130000
.* * LABELS-ADDR  GENERATE ADDRESSES, NOT OFFSETS.  DEFAULT IS * 00140000
.* ADDR. * 00150000
.* -DSPL  GENERATE OFFSETS, NOT ADDRESSES. * 00160000
.* * ID-0  PROGRAM IS NOT A TRANSIENT.  GENERATE ABSOLUTE * 00170000
.* ADDRESS VALUES(BASE NOT POSSIBLE). * 00180000
.* * -1  PROGRAM WILL EXECUTE IN TRANSIENT AREA ONE. * 00190000
.* THIS IS THE DEFAULT.  THE DEFAULTED VALUE CAN * 00200000
.* ALSO BE USED BY NON-TRANSIENTS BY PRECEDING * 00210000
.* THIS MACRO WITH A $CC4$$ EQU X'4000' IF * 00220000
.* ADDRESSES(BASE NOT TO BE USED) ARE WANTED. * 00230000
.* IF A BASE IS DESIRED, $CC4$$ MUST BE DEFINED * 00240000
.* IN TERMS OF *(LOCATION COUNTER). * 00250000
.* -2  PROGRAM WILL EXECUTE IN TRANSIENT AREA TWO. * 00260000
.* * $CC4$$-Y/N  GENERATE '$CC4$$ EQU X'4000'.  DEFAULT IS N. * 00270000
.* * DF-Y/N/0/1  GENERATE LABELS FOR DISPLAY FORMAT FACILITY * 00280000
.* WORK AREA.  DEFAULT IS Y. * 00290000
.* * CP-Y/N  GENERATE LABELS FOR COMMAND PROCESSOR WORK * 00300000
.* AREA.  DEFAULT IS Y. * 00310000
.* * AM-Y/N  GENERATE LABELS FOR ALLOCATION WORK AREA. * 00320000
.* DEFAULT IS Y. * 00330000
.* * TM-Y/N  GENERATE LABELS FOR TERMINATION WORK AREA. * 00340000
.* DEFAULT IS Y. * 00350000
.* * CM-Y/N  GENERATE LABELS FOR COMMUNICATIONS MANAGEMENT * 00360000
.* WORK AREA.  DEFAULT IS Y. * 00370000
.* * 00380000
.* NOTES: * 00390000
.* * LABELS THAT REFERENCE A LOCATION IN CORE CAN BE GENERATED * 00400000
.* WITH EITHER AN ADDRESS OR AN OFFSET.  USE OF THE ADDRESS * 00410000
.* LABELS WITH A REGISTER WILL REQUIRE A BASE INSTRUCTION.  THE * 00420000
.* LABEL FOR THE BASE INSTRUCTION IS $COMON.  LABELS THAT ARE * 00430000
.* PART OF A WORK AREA ARE EITHER ADDRESSES OR OFFSETS FROM THE * 00440000
.* START OF THE WORK AREA(EXCEPT FOR THE DF WORK AREA WHICH CAN * 00450000
.* BE EITHER OFFSETS FROM $COMON OR ADDRESSES). * 00460000
.* ***** * 00470000
.* $ECOM &LABELS-ADDR, &ID-1, &DF-Y, &CP-Y, &AM-Y, &TM-Y, &CM-Y, * X00480000
.* &$CC4$$-N * 00490000
.* TABLE &ID * 00500000
1  TABDF 1024 * 00510000
2  TABDF 512 * 00520000
.* TABLE &DF * 00530000
Y  TABDF 1 * 00540000
N  TABDF 0 * 00550000
.* TEXT * 00560000
.* ***** * 00570000
.* COMMUNICATIONS CONTROL PROGRAM * 00580000
.* COMMON AREA OFFSETS * 00590000

```

```

***** 00600000
TAONE EQU X'4000' LOCATION OF TRANSIENT AREA 1 00610000
TATWO EQU TAONE+512 LOCATION OF TRANSIENT AREA 2 00620000
SPACE 00630000
AIF (&ID EQ '0').CC4$$ 00640000
AIF (&$CC4$$ NE 'Y').LABEL 00650000
.CC4$$ ANOP 00660000
$COMON EQU X'4400' ADDRESS OF CCP COMMON 00670000
AGO .CC120 00680000
.LABEL AIF (&LABELS EQ 'ADDR').CC100 00690000
$COMON EQU 0 00700000
AGO .CC120 00710000
.CC100 ANOP 00720000
$COMON EQU $CC4$$+&ID 00730000
.CC120 ANOP 00740000
SPACE 2 00750000
*-----ENTRY POINT ADDRESS DEFINITION-----* 00760000
@CC4TR EQU $COMON-1+2 TRANSIENT RETURN ADDRESS 00770000
@CC4TX EQU @CC4TR+2 TRANSIENT TRANSFER CONTROL @ 00780000
@CC4PI EQU @CC4TX+2 1ST LEVEL TRANSIENT INVOCATION @ 00790000
@CC4TA EQU @CC4TX+2 1ST LEVEL TRANSIENT INVOCATION @ 00800000
@CC4IS EQU @CC4TA+2 COMMUNICATIONS I/O INTERFACE ADR 00810000
@CC4GM EQU @CC4IS+2 GETMAIN ENTRY ADDRESS 00820000
@CC4FM EQU @CC4GM+2 FREEMAIN ENTRY ADDRESS 00830000
@MLTIO EQU @CC4FM+2 MLTA IOCS ENTRY POINT @. 00840000
@MLTOP EQU @MLTIO+2 MLTA OPEN ENTRY POINT @. 00850000
@USECW EQU @MLTOP+2 USER SECURITY DATA WORK AREA ADR 00860000
@CC4MX EQU @USECW+2 MOVE FOREVER ROUTINE ADDRESS 00870000
@C4TI2 EQU @CC4MX+2 PROGRAM TERMINATION INTERFACE @ 00880000
@CC4TI EQU @C4TI2+2 OTHER TASK TERMINATE ADDRESS 00890000
@CC4SR EQU @CC4TI+2 ENTRY ADDRESS IN MOVE ROUTINE 00900000
@CC4TH EQU @CC4SR+2 @ TERMINATION INTERFACE 00910000
@BTRAC EQU @CC4TH+2 CCP BSCA TRACE ROUTINE 00920000
@MTRAC EQU @BTRAC+2 CCP MLTA TRACE ROUTINE 00930000
SPACE 2 00940000
*-----TRANSIENT COMMUNICATION AREA-----* 00950000
SV1TAX EQU @MTRAC+2 TRANSIENT AREA 1 PARAMETER AREA 00960000
SV2TAX EQU SV1TAX+2 TRANSIENT AREA 2 PARAMETER AREA 00970000
SPACE 2 00980000
*-----ADDRESSES OF SYSTEM TASK CONTROL BLOCKS -----* 00990000
@CMTCB EQU SV2TAX+2 @ COMMUNICATIONS MANAGEMENT TCB 01000000
@DFTCB EQU @CMTCB+2 @ OF DISPLAY FORMAT FACILITY TCB 01010000
@TMTCB EQU @DFTCB+2 @ TERMINATION TCB 01020000
@CPTCB EQU @TMTCB+2 @ COMMAND PROCESSOR TCB 01030000
@AVTCB EQU @CPTCB+2 @ AVAILABLE TCB'S 01040000
EJECT 01050000
*-----SYSTEM LEVEL FLAGS -----* 01060000
$FLGA EQU @AVTCB+1 FIRST SYSTEM LEVEL FLAG BYTE 01070000
#DFFOK EQU BIT0 DFF IS SUPPORTED 01080000
CPSOB1 EQU BIT1 SIGN ON PASSWORD REQUIRED 01090000
CPSOB2 EQU BIT2 SIGN ON USER PASSWORD REQUIRED 01100000
CPSHUT EQU BIT3 SHUTDOWN HAS BEEN REQUESTED 01110000
CPSU EQU BIT4 STARTUP IS IN PROCESS 01120000
#DSKFS EQU BIT5 DISK FILE SHARING IS SUPPORTED 01130000
CPSHD EQU BIT6 SHUTDOWN IN PROCESS 01140000
CPISNW EQU BIT7 $CC4IS IS NOT TO ISSUE WAIT 01150000
* AFTER POSTING $CC4CM 01160000
SPACE 3 01170000
$FLGB EQU $FLGA+1 SECOND SYSTEM LEVEL FLAG BYTE 01180000
#SUALL EQU BIT0 SUSPEND ALL IN EFFECT 01190000

```


#SUINT	EQU	BIT1	SUSPEND INIT IN EFFECT	01200000
#SUCMD	EQU	BIT2	SUSPEND COMMANDS IN EFFECT	01210000
#CPCAN	EQU	BIT6	CCP CANCEL RECEIVED	01220000
#PUCNT	EQU	BIT7	PROGRAM USE COUNTING IS ON	01230000
	SPACE	3		01240000
\$FLGC	EQU	\$FLGB+1	THIRD SYSTEM LEVEL FLAG BYTE	01250000
#MTRAC	EQU	BIT0	MLTA TRACE IS ON	01260000
#BTRAC	EQU	BIT1	BSCA TRACE IS ON	01270000
#NTRAC	EQU	BIT2	NO TRACE (INTERVAL POLLING)	01280000
#PUTTP	EQU	BIT3	PUT ONLY TP GETMAIN REQUEST	01290000
#INVPL	EQU	BIT4	INVITE PL GETMAIN REQUEST	01300000
#D3340	EQU	BIT5	IPL FROM THE 3340 DISK	01310000
INTNSP	EQU	BIT6	NO INT POLL SUPPORT=1	01320000
#3340	EQU	BIT7	EXTENDED ADDRESSING	01330000
	SPACE	3		01340000
----- SYSTEM CONSTANTS -----				01350000
X\$0000	EQU	\$FLGC+2	CONSTANT XL2'0000'	01360000
X\$0001	EQU	X\$0000+1	CONSTANT XL2'0001'	01370000
X\$0002	EQU	X\$0001+2	CONSTANT XL2'0002'	01380000
X\$0004	EQU	X\$0002+2	CONSTANT XL2'0004'	01390000
X\$FFFF	EQU	X\$0004+2	CONSTANT XL2'FFFF'	01400000
	EJECT			01410000
----- QUEUE ADDRESS POINTERS -----				01420000
@ALOCQ	EQU	X\$FFFF+2	Q OF TASKS WAITING TO BE ALLOC'D	01430000
@WATSK	EQU	@ALOCQ+2	ALLOCATE WAIT QUEUE	01440000
@QTUBS	EQU	@WATSK+2	@ Q TUB'S WAITING FOR TCB/CORE	01450000
@GMWTQ	EQU	@QTUBS+2	@ GETMAIN TCB QUEUE	01460000
@DFEQ	EQU	@GMWTQ+2	QUEUE FOR REQUESTS TO DFF TASK	01470000
@PRLQ	EQU	@DFEQ+2	@ OF PARAMETER LIST QUED FOR CM	01480000
	SPACE	2		01490000
----- LIST ADDRESS POINTERS -----				01500000
@TALST	EQU	@PRLQ+2	@ PROGRAM LIST IN \$CC4TA	01510000
@TCORG	EQU	@TALST+2	@ OF THE TCB LIST	01520000
@FSQB	EQU	@TCORG+2	@ FIRST AVAILABLE FSQE	01530000
@LCB#1	EQU	@FSQB+2	@ FIRST LCB IN SYSTEM	01540000
@TUBQ	EQU	@LCB#1+2	@ OF THE FIRST TUB IN THE SYSTEM	01550000
@DFCT	EQU	@TUBQ+2	@ OF THE SHORT DTF ADDRESS LIST	01560000
@TNT	EQU	@DFCT+2	@ OF 1ST TERMINAL NAME TABLE ENT	01570000
@XDT	EQU	@TNT+2	@ 1ST SYMBOLIC DFCT NAME ENTRY	01580000
	SPACE	2		01590000
----- MAIN STORAGE CONTROL BLOCK(TPBUFF) -----				01600000
@BUFA	EQU	@XDT-1+2	@ OF FIRST FREE SEGMENT	01610000
#NBND	EQU	@BUFA+1+2	RESERVED - MUST BE ZEROS	01620000
@LOBND	EQU	#NBND+2	@ LO-BOUND GETMAIN AREA	01630000
@HIBND	EQU	@LOBND+2	@ HI-BOUND GETMAIN AREA	01640000
#GMS	EQU	@HIBND-1+2	SIZE OF LARGEST FREE SPACE	01650000
	SPACE	2		01660000
#TPBUF	EQU	#GMS+1+2	ORIGINAL SIZE OF TPBUFF	01670000
@UPA	EQU	#TPBUF+1	USER PROGRAM AREA ATR	01680000
#AVCOR	EQU	@UPA+1	# 2K BLOCKS NOT GIVEN TO NEP'S	01690000
@PUCNT	EQU	#AVCOR+2	@ PROGRAM USE COUNT TABLE	01700000
	SPACE	2		01710000
----- GENERAL AREAS -----				01720000
@TUSTG	EQU	@PUCNT+2	@ OF TUB IN STAGING NOW(CP TASK)	01730000
@KNTUB	EQU	@TUSTG+2	@ OF CONSOLE TUB	01740000
@PTX	EQU	@KNTUB+2	@ OF PCT MASTER INDEX	01750000
@PTXCS	EQU	@PTX+2	C/S VALUE OF PCT DISK START	01760000
#PCTLN	EQU	@PTXCS+1	LONGEST LEN PCT USED BY PGM RQST	01770000
#DFCT	EQU	#PCTLN+1	NO. ENTRIES IN SHORT DTF @ LIST	01780000
#SETID	EQU	#DFCT+1	ID OF ASSIGNMENT SET IN USE	01790000

#XDT	EQU	#SETID+1	NUMBER OF SYMFILE STATEMENTS	01800000
#RSVD1	EQU	#XDT+1	RESERVED AREA - 1 BYTE	01810000
@UALFA	EQU	#RSVD1+1	ATR - UNCHANGING START OF UPA	01820000
@TKFSB	EQU	@UALFA+2	FSB AREA FOR TCB'S	01830000
	SPACE	1		01840000
CPLPWD	EQU	@TKFSB+1	LENGTH OF SIGN ON PASSWORD	01850000
CPPSWD	EQU	CPLPWD+6	CCP SIGN ON PASSWORD	01860000
	SPACE	2		01870000
----- CCP DUMP AREA DISK ADDRESSES -----				01880000
@DSTRT	EQU	CPPSWD+2	DUMP AREA START C/S	01890000
@DEND	EQU	@DSTRT+2	DUMP AREA END C/S	01900000
@DNEXT	EQU	@DEND+2	DUMP AREA NEXT ENTRY C/S	01910000
#CPFLQ	EQU	@DNEXT+1	\$CCPFILE DEVICE Q-BYTE	01920000
	EJECT			01930000
----- FIXED LOCATIONS OF RESIDENT CODE -----				01940000
CC4TA	EQU	#CPFLQ+1	LOAD IAR OF @CC4TA	01950000
CC4PI	EQU	CC4TA	LOAD IAR OF @CC4PI (@CC4TA)	01960000
CC4IS	EQU	CC4TA+4	LOAD IAR OF @CC4IS	01970000
CC4GM	EQU	CC4IS+4	LOAD IAR OF @CC4GM	01980000
CC4FM	EQU	CC4GM+4	LOAD IAR OF @CC4FM	01990000
USECW	EQU	CC4FM+4	LOAD IAR OF @USECW	02000000
CC4MX	EQU	USECW+4	LOAD IAR OF @CC4MX	02010000
C4TI2	EQU	CC4MX+4	LOAD IAR OF @C4TI2	02020000
CC4TI	EQU	C4TI2+4	LOAD IAR OF @CC4TI	02030000
CC4TT	EQU	CC4TI+4	LOAD IAR OF \$TRACE(X'004C')	02040000
CC4SR	EQU	CC4TT+4	LOAD IAR OF @CC4MV	02050000
CC4FR	EQU	CC4SR+4	LOAD IAR OF @CC4FR	02060000
@CC4FR	EQU	CC4FR+3+2	@ OF \$CC4FR IN CM	02070000
@CC4II	EQU	@CC4FR+2	@ OF \$CC4II	02080000
	SPACE			02090000
----- MAINTENANCE SPACE ----- 2 BYTES -----				02100000
	SPACE			02110000
\$END1	EQU	@CC4II+2+1	END OF MAINTENANCE SPACE	02120000
	SPACE	3		02130000
----- WORK AREAS LENGTHS DEFINITIONS -----				02140000
#LDFWK	EQU	15	LENGTH DFF WORK AREA	02150000
#LCPWK	EQU	50	LENGTH COMND PROCESSOR WORK AREA	02160000
#LAMWK	EQU	9	LENGTH ALLOCATION WORK AREA	02170000
#LTMWK	EQU	24	LENGTH TERMINATION WORK AREA	02180000
#LCMWK	EQU	45	LENGTH COMMO MGMT WORK AREA	02190000
#LWKWK	EQU	143	LENGTH-SUM OF ALL WORK AREAS	02200000
	SPACE	3		02210000
----- CONSOLE TUB AND PARAMETER LIST -----				02220000
KNTUB	EQU	\$END1+#LWKWK	FIXED LOCATION OF CONSOLE TUB	02230000
KNPL	EQU	KNTUB+38	CONSOLE PARAMETER LIST	02240000
	SPACE	3		02250000
----- SHUTDOWN FIELDS -----				02260000
SHDECB	EQU	KNPL+19	SHUTDOWN'S ECB	02270000
FEHLT@	EQU	SHDECB+2+3	ADDRESS OF FE HALT ROUTINE	02280000
SHDSAV	EQU	FEHLT@+11	CM TCB FIELDS SAVE AREA	02290000
	SPACE	2		02300000
----- CM FIELDS AND MAINTENANCE SPACE -----				02310000
#CMTRL	EQU	SHDSAV-10+11	CM'S TRANSLATE PARAMETER LIST	02320000
#CMMVL	EQU	#CMTRL	CM'S MOVE PARAMETER LIST	02330000
	SPACE			02340000
PLTIME	EQU	#CMMVL+10+4	POLL TIME FOR CM	02350000
WATIME	EQU	PLTIME+3	WAIT TIME FOR CM	02360000
SAVLOP	EQU	WATIME+1	SAVE AREA FOR POLL LOOP COUNT	02370000
@CCPTB	EQU	SAVLOP+2	ADDRESS OF CCP PARTITION TCB	02380000
#TPPUT	EQU	@CCPTB+2	LENGTH OF PUT AREA OF TPBUFFER	02390000

```

#TPANY EQU #TPPUT+2          LENGTH OF COMMON TPBUFFER AREA 02400000
#ANYS EQU #TPANY+1          SIZE OF LARGEST COMMON AREA 02410000
CORCNT EQU #ANYS+2          NUMBER OF PL WAITING ON CORE 02420000
AIF (&DF EQ '0').CC500      02430000
EJECT 02440000
***** DFF WORK AREA 15 BYTES***** 02450000
$DFWK EQU $END1             START OF DFF WORK AREA 02460000
@DFFIX EQU $DFWK+1          C/S OF DFF FORMATS INDEX 02470000
#DFQ EQU @DFFIX+1           FDT DEVICE Q-BYTE 02480000
#DF1ID EQU #DFQ+1           ID OF TASK "OWNING" SPACE IN 02490000
* DFF COMM. AREA FOR LINE 1 02500000
SPACE 1 02510000
#DF2ID EQU #DF1ID+1         ID OF TASK "OWNING" SPACE IN 02520000
* DFF COMM. AREA FOR LINE 2 02530000
$DFECB EQU #DF2ID+1        ECB FOR DFF TASK 02540000
.CC500 AIF (&CP EQ 'N').CC600 . CONTINUE HERE 02550000
EJECT 02560000
***** COMMAND PROCESSOR WORK AREA 50 BYTES***** 02570000
AIF (&LABELS EQ 'ADDR').CC520 02580000
$CPWK EQU X'4400'+$END1+#LDFWK START OF COMMAND 02590000
$CP EQU 0 PROCESSOR'S WORK AREA 02600000
AGO .CC540 02610000
.CC520 ANOP 02620000
$CPWK EQU $END1+#LDFWK START OF COMMAND 02630000
$CP EQU $CPWK PROCESSOR'S WORK AREA 02640000
.CC540 ANOP 02650000
SPACE 2 02660000
*----- SAVE AREAS FOR THE DISPLAY TRANSIENTS -----* 02670000
$CPIAR EQU $CP+1 IAR 02680000
$CPXR1 EQU $CP+3 TUB,TNT,OR FSB 02690000
$CPDTF EQU $CP+5 DTF 02700000
$CPUSE EQU $CP+6 PROGRAM USE COUNT SAVE FIELD 02710000
SPACE 2 02720000
*----- COMMAND PROCESSOR ECB LIST -----* 02730000
$CPLST EQU $CP+7 ECB LIST. DEFINED AS FOLLOWS: 02740000
* DC AL2($CPQ) PROGRAM INITIATE-POSTED BY TERM 02750000
* DC AL2($CPPF9) PF9 - PROGRAM REQUEST 02760000
* DC AL2($CPCM) DATA IN TP BUFFER FOR CP 02770000
* DC AL2($CPOCC) OCC 02780000
* DC AL2($CP1ST/$CPWTO/65535) STARTUP/WTOR/LIST DELIMITER 02790000
* DC XL2'FFFF' LIST DELIMITER 02800000
SPACE 2 02810000
*----- COMMAND PROCESSOR ECB'S -----* 02820000
$CPQ EQU $CP+19 PROGRAM INITIATE-POSTED BY TERM 02830000
$CPPF9 EQU $CP+22 PF9 - PROGRAM REQUEST 02840000
$CPCM EQU $CP+25 DATA IN TP BUFFER FOR CP 02850000
$CPOCC EQU X'0048' OCC - IN THE DSM NUCLEUS 02860000
$CP1ST EQU $CP+28 FIRST TIME SWITCH 02870000
$CPWTO EQU 0 WTO - GETMAINED FOR IN TPBUFF 02880000
SPACE 2 02890000
*----- COMMAND PROCESSOR FLAG BYTE -----* 02900000
$CPFLG EQU $CP+29 COMMAND PROCESSOR FLAG BYTE 02910000
$CPFR EQU BIT0 FREEMAIN TO BE DONE 02920000
$CPCFR EQU BIT3 FREE NEEDED FOR CONSOLE OCC 02930000
$CPDPG EQU BIT4 PAGE 1 OF DISPLAY DONE 02940000
$CPD1S EQU BIT5 DO SECONDARY MENU 02950000
SPACE 1 02960000
*----- COMMAND PROCESSOR TASK TRANSIENT COMMUNICATION AREA -----* 02970000
$CPPRQ EQU $CP+30 LABEL FOR PROGRAM REQUEST 02980000
SPACE 02990000

```

\$PCOM	EQU	\$CP+30	LABEL FOR COMMANDS	03000000
\$CPRTC	EQU	\$CP+31	SAVE AREA FOR PLRTC	03010000
\$CPEFL	EQU	\$CP+33	SAVE AREA FOR PLEFFL	03020000
\$CPRCA	EQU	\$CP+35	SAVE AREA FOR PLRECA	03030000
	SPACE			03040000
----- STARTUP VALUES - VALID ONLY DURING STARTUP -----				03050000
\$CPMSG	EQU	\$CP+32	ADDRS OF \$CC4IG	03060000
#LSTSZ	EQU	\$CPMSG+1+1	# ENTRIES IN FIRST LEVEL LIST	03070000
@XSNT1	EQU	#LSTSZ+2	ENTRIES IN \$CC4PI	03080000
@XSNT2	EQU	@XSNT1+2	FOR START-UP	03090000
@L1TCB	EQU	@XSNT2+2	SWITCHED LCB TCB @ FOR LINE 1	03100000
@L2TCB	EQU	@L1TCB+2	SWITCHED LCB TCB @ FOR LINE 2	03110000
.CC600	AIF	(&AM EQ 'N').CC700		03120000
	EJECT			03130000
***** ALLOCATION WORK AREA 9 BYTES*****				03140000
	AIF	(&LABELS EQ 'ADDR').CC620		03150000
\$AMWK	EQU	X'4400'+\$END1+#LDFWK+#LCPWK	START OF ALLOCATION	03160000
\$AM	EQU	0	WORK AREA	03170000
	AGO	.CC640		03180000
.CC620	ANOP			03190000
\$AMWK	EQU	\$END1+#LDFWK+#LCPWK	START OF ALLOCATION	03200000
\$AM	EQU	\$AMWK	WORK AREA	03210000
.CC640	ANOP			03220000
	SPACE	2		03230000
\$AMFLG	EQU	\$AM	ALLOCATION FLAG BYTE	03240000
\$AMBSY	EQU	BIT0	1--ALLOCATION BUSY	03250000
\$APEND	EQU	BIT1	1--ALLOCATION TASK POST PENDING	03260000
\$AMDFP	EQU	BIT2	NON-RESIDENT DFP ALOC IN PROCESS	03270000
\$AMA1	EQU	BIT3	U/R DEVICE ALLOCATION IN PROCESS	03280000
\$AMA2	EQU	BIT4	REJECT IN PROCESS	03290000
\$AMPF9	EQU	BIT5	CONSOLE TUB IN USE FOR PROGRAM	03300000
	SPACE			03310000
\$AMUR	EQU	\$AMFLG+1	CCP LEVEL AND SPOOLING DEVICES.	03320000
A1SPLV	EQU	BIT0	1=CCP IS IN LEVEL 1.	03330000
*			0=CCP IS IN LEVEL 2.	03340000
A1PTRS	EQU	BIT1	PRINTER IS SPOOLED FOR CCP LVL.	03350000
*		BIT2	MFCU/M SEC IS SPOOL RDR FOR CCP	03360000
*		BIT3	MFCU/M SEC IS SPOOL PCH FOR CCP	03370000
A1741S	EQU	BIT4	3741 IS SPOOLED READER FOR CCP	03380000
A1501S	EQU	BIT5	2501 IS SPOOLED READER FOR CCP	03390000
*		BIT6	MFCU/M PRI IS SPOOL RDR FOR CCP	03400000
*		BIT7	MFCU/M PRI IS SPOOL PCH FOR CCP	03410000
\$AMPA	EQU	\$AMUR+1	PERM ALOC AND SPOOLED UR	03420000
*			DEVICES FOR CCP.	03430000
*PCTPRS		BIT0	1=WILL SHARE PRINTER.	03440000
*PCT41I		BIT1	1=3741 IS PERMANENTLY ALLOCATED	03450000
*PCT501		BIT2	1=2501 IS PERM ALLOCATED.	03460000
*PCTPRT		BIT3	1=PRINTER IS PERM ALLOCATED.	03470000
*PCT142		BIT4	1=1442 IS PERM ALLOCATED.	03480000
*PCTMFU		BIT5	1=MFCU/M IS PERM ALLOCATED.	03490000
*		BIT6	1=1442 IS SPOOL READER FOR CCP	03500000
*		BIT7	1=1442 IS SPOOL PUNCH FOR CCP	03510000
	SPACE			03520000
\$AMSA	EQU	\$AMPA+1	IN USE DEVICES	03530000
\$AMSHR	EQU	\$AMSA+1	COUNT OF PRESENT PRINT SHARERS	03540000
\$AMFSQ	EQU	\$AMSHR+1	NUMBER OF FSQE BLOCKS	03550000
.CC700	AIF	(&TM EQ 'N').CC800		03560000
	EJECT			03570000
***** TERMINATOR WORK AREA 24 BYTES*****				03580000
	AIF	(&LABELS EQ 'ADDR').CC720		03590000

\$TMWK	EQU	X'4400'+\$END1+#LDFWK+#LCPWK+#LAMWK	START OF	03600000
\$TM	EQU	0	TERMINATION'S WORK AREA	03610000
	AGO	.CC740		03620000
.CC720	ANOP			03630000
\$TMWK	EQU	\$END1+#LDFWK+#LCPWK+#LAMWK	START OF TERMINATION'S	03640000
\$TM	EQU	\$TMWK	WORK AREA	03650000
.CC740	ANOP			03660000
	SPACE	2		03670000
\$TMFLG	EQU	\$TM	TERMINATOR FLAG BYTE	03680000
\$TMSTK	EQU	BIT0	SYSTEM TASK FAILURE	03690000
\$TMBSY	EQU	BIT0	1--TERMINATION TASK IS ACTIVE	03700000
*			0--TERMINATION TASK NOT BUSY	03710000
\$TMDMP	EQU	BIT1	1--NO USER TASK ACTIVE FOR C CCP	03720000
*			0--USER TASK ACTIVE FOR C CCP	03730000
\$TMEJ	EQU	BIT2	1--\$CC4EJ (CCP END OF JOB)	03740000
*			HAS BEEN INVOKED BY \$CC4TD	03750000
*			(TERMINATION CONTROL ROUTINE)	03760000
\$TMCLZ	EQU	BIT3	TERMINATION MUST CALL CLOSE	03770000
\$TMCM	EQU	BIT4	RESERVED	03780000
\$TMDDR	EQU	BIT5	DISK DUMP REQUESTED (NOT 2A,2B)	03790000
\$TMDFL	EQU	BIT6	ON=NO MORE DUMP SPACE GUARANTEED	03800000
\$TMDER	EQU	BIT7	ON=PERMANENT DISK I/O ERROR	03810000
*			DURING TERMINATION DUMP	03820000
	SPACE	1		03830000
\$TMTCB	EQU	\$TMFLG+2	@ OF TCB CURRENTLY BEING	03840000
*			PROCESSED BY TERMINATION TASK	03850000
\$TMECB	EQU	\$TMTCB+1	TERMINATION'S ECB (3 BYTES)	03860000
\$TMDID	EQU	\$TMECB-1+3+2	DUMP ID DIGITS (2 BYTES)	03870000
	SPACE	2		03880000
-----		\$CC4TI-\$CC4TM-CC4TH	WORK AREAS-----*	03890000
TIWRK1	EQU	\$TMDID+2	RESIDENT WORK	03900000
TIWRK2	EQU	TIWRK1+2	AREAS	03910000
TIWRK3	EQU	TIWRK2+2	FOR INFORMATION	03920000
TIWRK4	EQU	TIWRK3+2	SAVE ON	03930000
TIWRK5	EQU	TIWRK4+2	SYSTEM DISASTER	03940000
TIWRK6	EQU	TIWRK5+2	CONDITIONS	03950000
.CC800	AIF	(&CM EQ 'N').CCEND		03960000
	EJECT			03970000
*****		COMMUNICATIONS MANAGEMENT WORK AREA	45 BYTES*****	03980000
	AIF	(&LABELS EQ 'ADDR').CC820		03990000
\$CMWK	EQU	X'4400'+\$END1+#LDFWK+#LCPWK+#LAMWK+#LTMWK	START OF	04000000
\$CM	EQU	0	COMMUNICATION TASK WORK AREA	04010000
	AGO	.CC840		04020000
.CC820	ANOP			04030000
\$CMWK	EQU	\$END1+#LDFWK+#LCPWK+#LAMWK+#LTMWK	START OF COMMUNICATION	04040000
\$CM	EQU	\$CMWK	MANAGEMENT WORK AREA	04050000
.CC840	ANOP			04060000
	SPACE	2		04070000
#CMDTF	EQU	\$CM+1	SAVE AREA FOR DTF ADDRESS	04080000
#CMPL	EQU	#CMDTF+2	SAVE AREA FOR PARM LIST	04090000
	SPACE			04100000
#CMTMA	EQU	#CMPL+2	SAVE AREA FOR TERMINAL ADDRESS	04110000
#CMBK1	EQU	#CMTMA	RETURN CODE FOR \$CC4BL MESSAGE	04120000
	SPACE			04130000
#CMTFT	EQU	#CMTMA+1	SAVE AREA FOR TERMINAL FEATURES	04140000
#CMBK2	EQU	#CMTFT	INTERNAL MSG TYPE FOR \$CC4BL	04150000
	SPACE			04160000
#CMPTR	EQU	#CMTFT+2	SAVE AREA FOR PREVIOUS POINTER	04170000
#CMFPL	EQU	#CMPTR+2	SAVE AREA FOR FOUND PARM LIST	04180000
#CMLSL	EQU	#CMFPL+2	LAST LCB CHECKED FOR PL NEEDING	04190000

```

*
#CMTUB EQU #CMLSL+2
#CMERP EQU #CMTUB+2
*
#CMSWT EQU #CMERP+1
#CMARR EQU BIT0
#CMFMD EQU BIT1
*
SPACE 1
#OPEND EQU #CMSWT+1
#CCMCL EQU #OPEND+2
#RUFCL EQU #CCMCL+2
@TAS EQU #RUFCL+2
#HITAS EQU @TAS+1
SPACE
@CSSTT EQU #HITAS+2
#HISTT EQU @CSSTT+1
SPACE
@MLTAD EQU #HISTT+2
*
@CKLST EQU @MLTAD+2
SPACE 1
*----- COMMUNICATIONS MANAGEMENT ECB'S -----*
$CMECB EQU @CKLST+1
$CMFM EQU $CMECB+3
@ANYTP EQU $CMFM+4
@INVPL EQU @ANYTP+2
SPACE 2
*----- COMMUNICATIONS MANAGEMENT RESERVED AREA -----*
$CMRV2 EQU @INVPL+1
.CCEND ANOP
SPACE 3
$CCEND EQU CORCNT+1
MEND

```

```

* GETMAIN 04200000
SAVE AREA FOR TUB ADDRESS 04210000
@ OF DTF TO BE RESCHEDULED AFTER 04220000
BEING IN CCP ERP 04230000
CONTROL SWITCH 04240000
1--OK TO MODIFY TCBARR 04250000
1--CM FREEMAIN DONE SINCE LAST 04260000
* CHECK. 04270000
T-P OP END COUNT 04280000
MAXIMUM COMMAND BUFFER LENGTH 04290000
MAX READ UNDER FORMAT COM LEN 04300000
TERMINAL ATTRIBUTE SET ADDRESS 04310000
HIGHEST TAS INDEX IN SYSTEM 04320000
DISK C/S OF STT 04330000
NO ENTRIES IN STT(SW TERM TBL) 04340000
ADDRESS OF MLTA ADAPTER 04350000
0 INDICATES NO MLTA ADAPTER 04360000
ADDRESS OF CHECK LIST 04370000
04380000
04390000
04400000
04410000
04420000
ECB FOR POST OF CM 04430000
FREEMAIN POST OF CM 04440000
ADDRESS OF COMMON TP FOR GETMAIN 04450000
ADDRESS OF INV PL TP FOR GETMAIN 04460000
04470000
RESERVED AREA -----* 04480000
START OF 2 BYTE RESERVED AREA 04490000
04500000
04510000
END OF $CCCOM 04520000
04530000

```

```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 2 * 00030000
.* * 00040000
.* NAME:  $ETCC * 00050000
.* * 00060000
.* FUNCTION: * 00070000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- DEFINE * 00080000
.* SYMBOLS FOR TASK COMPLETION CODES. * 00090000
.* * 00100000
.* INPUT OPERANDS: * 00110000
.* . DF-YES/Y/1/NO/N/0 * 00120000
.* SPECIFIES WHETHER DFF SUPPORTED OR NOT * 00130000
***** 00140000
.$ETCC &DF-1 00150000
.* 00160000
TABLE &DF 00170000
YES TABDF 1 00180000
Y TABDF 1 00190000
NO TABDF 0 00200000
N TABDF 0 00210000
.* 00220000
TEXT 00230000
.* 00240000
SPACE 2 00250000
***** 00260000
* TASK COMPLETION CODE SYMBOLS * 00270000
***** 00280000
SPACE 2 00290000
* CODES FOR ERRORS DETECTED BY $CC4II -- COMMUNICATIONS INTERFACE 00300000
SPACE 1 00310000
TCCIOP EQU X'01' INVALID OP CODE/MODIFIERS 00320000
TCCIOK EQU X'02' INVALID OP FOR 5471 CONSOLE 00330000
TCCNNF EQU X'03' STN NOT DEFINED IN SYSTEM 00340000
TCCNAP EQU X'04' TERMINAL NOT ALLOCATED TO PROGRA 00350000
TCCNAN EQU X'05' ALLOC TERM REFERENCED BY OTHER 00360000
* THAN ALLOCATED NAME 00370000
TCCMBL EQU X'06' BLANK STN FOR MRTS PROGRAM 00380000
TCCRBL EQU X'07' BLANK STN AND REQUESTOR RELEASED 00390000
TCCIBL EQU X'08' BLANK STN INVALID FOR THIS OP 00400000
TCCISN EQU X'09' INVALID USE OF SUB-TERMINAL NAME 00410000
TCCNAT EQU X'0A' STN NOT ASSIGNED TO TERMINAL 00420000
TCCITA EQU X'0B' TERMINAL ATTRIBUTE SET INVALID 00430000
* FOR TERMINAL SPECIFIED 00440000
TCCIOC EQU X'0C' TERMINAL DOES NOT HAVE I/O 00450000
* CAPABILITY SPECIFIED IN OPERATIN 00460000
TCCIOL EQU X'0D' INVALID OUTPUT LENGTH 00470000
TCCIIL EQU X'0E' INVALID INPUT LENGTH 00480000
TCCTPB EQU X'0F' INPUT LENGTH GT TP BUFFER SIZE 00490000
TCCIMO EQU X'10' INVALID OP WITH DATA FROM 00500000
* PROGRAM REQUEST OUTSTANDING 00510000
TCCIIO EQU X'11' INVALID OP WITH INVITE INPUT 00520000
* OUTSTANDING TO THIS TERMINAL 00530000
TCCIAN EQU X'12' ACCEPT INVALID WITH NO 00540000
* OUTSTANDING INVITES FOR NON-NEP 00550000
TCCIAC EQU X'13' NO OUTSTANDING INVITES FOR NEP 00560000
* WITH ACTIVE TERMINAL COUNT NOT 00570000
* LESS THAN MAX TERMINAL COUNT 00580000
TCCNIO EQU X'14' INVALID OP WITH NO INVITE 00590000

```

*		OUTSTANDING FOR THIS TERMINAL	00600000
TCCIGB EQU	X'15'	INPUT AREA NOT LARGE ENOUGH FOR	00610000
*		BSCA GET BLOCK	00620000
TCCICA EQU	X'16'	COPY TO TERMINAL WITHOUT MAPPING	00630000
TCCICN EQU	X'17'	COPY TO TERMINAL NAME NOT FOUND	00640000
TCCICT EQU	X'18'	COPY INVALID TO 3275	00650000
TCCIEA EQU	X'19'	ERASE TO TERMINAL WITHOUT MAPPING	00660000
TCCIPA EQU	X'1A'	PUT OVERRIDE WITHOUT MAPPING	00670000
TCCIPM EQU	X'1B'	INVALID PUT WITH MAPPING -NO EOT	00680000
TCCITB EQU	X'1C'	RECORD AREA TOO SMALL FOR BSCA	00690000
*		GET WITH ITB, OR OUTPUT	00700000
*		LENGTH TOO SMALL FOR PUT - ITB	00710000
TCCIAI EQU	X'1D'	ACCPET BUT IMPOSSIBLE TO RECEIVE	00720000
*		DATA FROM ANY TERMINAL WITH	00730000
*		INVITE CURRENTLY OUTSTANDING	00740000
TCCIKL EQU	X'1E'	OUTPUT LENGTH TO CONSOLE	00750000
*		GREATER THAN 80	00760000
TCCLRG EQU	X'1F'	OUTPUT LENGTH LARGER THAN	00770000
*		TP BUFFER SIZE	00780000
	SPACE 1		00790000
*	ERRORS DETECTED CONCERNING 'SUBR90' MOVE ROUTINE		00800000
	SPACE 1		00810000
TCCLRP EQU	X'20'	LENGTH OF FIELDS TO BE MOVED	00820000
*		IS GREATER AS THE GIVEN BUFFER-	00830000
*		LENGTH	00840000
	SPACE 1		00850000
*	ERRORS DETECTED CONCERNING AN 'ACQUIRE TERMINAL' OPERATION		00860000
	SPACE 1		00870000
TCCAST EQU	X'21'	ACQ OF SUB-TERMINAL NAME	00880000
TCCAQN EQU	X'22'	ACQUIRE CONSOLE	00890000
TCCAAS EQU	X'23'	ATTEMPT TO ACQUIRE AN OWNED	00900000
*		TERMINAL WHO IS NOT REQUESTOR,	00910000
*		OR IS THE REQUESTOR BUT THIS IS	00920000
*		NOT A SET ATTRIBUTES OPERATION	00930000
TCCADT EQU	X'24'	SET ATTRIBUTES TO OWNED TERMINAL	00940000
*		IN PROCESS OF BSCA DATA TRANSFER	00950000
TCCATS EQU	X'25'	TERMINAL ATTRIBUTE SET ERROR	00960000
	SPACE 1		00970000
*	ERRORS DETECTED CONCERNING A 'RELEASE TERMINAL' OPERATION		00980000
	SPACE 1		00990000
TCCRST EQU	X'26'	RELEASE SUB-TERMINAL NAME	01000000
TCCROW EQU	X'27'	RELEASE OF BSCA TERMINAL WHICH	01010000
*		CURRENTLY OWNS THE LINE	01020000
	SPACE 1		01030000
*	ERRORS DETECTED BY THE RPG COMMUNICATIONS SERVICE ROUTINE 'SUBR92'		01040000
	SPACE 1		01050000
TCCRPL EQU	X'28'	LENGTH SPECIFIED FOR INPUT OR	01060000
*		OUTPUT OPERATION IS GREATER THAN	01070000
*		THE RECORD LENGTH IN THE DTF	01080000
TCCRCB EQU	X'29'	PUT THEN GET OPERATION	01090000
*		PUT PENDING AND NOT FOLLOWED BY	01100000
*		A GET OF THE PUT THEN GET	01110000
	SPACE 1		01120000
*	CANCELLATION BY THE SYSTEM OPERATOR		01130000
	SPACE 1		01140000
TCCCCN EQU	X'2A'	CONSOL CANCEL	01150000
TCCCCP EQU	X'2B'	SYSTEM OPERATOR CANCELED CCP	01160000
	SPACE 1		01170000
*	ERRORS DETECTED UPON USER PROGRAM EXIT		01180000
	SPACE 1		01190000

TCCINV EQU	X'2C'	TASK HAD INVITES OUTSTANDING	01200000
*		AT END-OF-JOB	01210000
TCCEOT EQU	X'2D'	TASK HAS UNFINISHED BSCA EOT	01220000
*		OPERATION AT END-OF-JOB	01230000
	SPACE 1		01240000
* FILE	OPERATION ERROR -- INVALID ADD TO SHARED FILE		01250000
	SPACE 1		01260000
TCCIVA EQU	X'2E'	INVALID ADDER	01270000
	SPACE		01280000
*	TERMINATION CODE FOR INVALID CHARACTER SENT TO THE 3270 SYSTEM.		01290000
	SPACE		01300000
TCCHAR EQU	X'2F'	ILLEGAL CHARACTER OR ADDRESS	01310000
*		SENT TO THE 3270 SYSTEM.	01320000
	SPACE 1		01330000
* RPG	TERMINATION CODE		01340000
TCCRIS EQU	X'30'	PGM USED SUBR91 WITH 'SPECIAL'	01350000
*		FILES, OR SUBR92 WITH 'EXIT/'	01360000
*		RLABL OPERATION.	01370000
	SPACE 1		01380000
TCCNEJ EQU	X'31'	NEP EOJ BEFORE SHUTDOWN	01390000
	SPACE 1		01400000
TCCSQB EQU	X'32'	NO AVAILABLE SECTOR QUEUE BLOCK	01410000
*		FOR SHARED FILE I/O REQUEST	01420000
* UNCONDITIONAL HALT ISSUED FROM USER PROGRAM			01430000
	SPACE 1		01440000
* RPG II CODES FROM SUBR92			01450000
	SPACE 1		01460000
TCC33 EQU	X'33'	INDICATOR 91 NOT RESET FOR MORE	01470000
*		THAN TWO OPERATIONS.	01480000
	SPACE 1		01490000
TCC34 EQU	X'34'	SPECIAL FILE USES SUBR92 BUT AN	01500000
*		ARRAY IS NOT ASSOCIATED WITH	01510000
*		THE FILE.	01520000
	SPACE 1		01530000
TCCDCN EQU	X'3A'	TASK CANCEL WITH DUMP	01540000
	SPACE 1		01550000
TCCDCP EQU	X'3B'	CCP CANCEL WITH DUMP	01560000
	SPACE 1		01570000
* ADDITIONAL TERMINATION CODES FROM \$CC4II - COMMUNITCATIONS INTER.			01580000
	SPACE 1		01590000
TCCNDF EQU	X'35'	DFP TERMINAL REFERENCED BY	01600000
*		A NON-DFP TASK	01610000
TCCWAT EQU	X'36'	WAIT OPERATION DATA AREA WAS	01613000
*		NOT 10 BYTES (24 FOR RPG)	01616000
	SPACE 1		01620000
* 3741 TERMINATION CODE			01621000
TCCST1 EQU	X'40'	3741 HAS INDICATED THAT	01622000
*		STATUS WILL BE SENT BUT SENT	01623000
*		DATA INSTEAD.	01624000
	SPACE 1		01625000
* CCP TERMINATION CODES FOR PROGRAM CHECK TERMINATION.			01630000
TCCPRT EQU	X'50'	STORAGE PROTECT VIOLATION	01640000
TCCQCD EQU	X'51'	INVALID Q-CODE	01650000
TCCOPC EQU	X'52'	INVALID OPCODE	01660000
TCCADR EQU	X'53'	INVALID ADDRESS	01670000
TCCPRV EQU	X'54'	PRIVELEGED OPERATION	01680000
*	EQU	X'55'	RESERVED FOR PROGRAM CHECK
*	EQU	X'56'	RESERVED FOR PROGRAM CHECK
	SPACE		01710000
TCCHLT EQU	X'7F'	UNCONDITIONAL HALT FROM USER TSK	01720000

SPACE 1		01730000
* ERROR DETECTED BY GENERAL ENTRY INTERCEPT -- INVALID SUB-RIB		01740000
SPACE 1		01750000
TCCRIB EQU X'80'	INVALID SUB-RIB TO GENERAL ENTRY	01760000
.*		01770000
SPACE 2		01780000
-----		01790000
* PSEUDO OPEN/CLOSE/ALLOCATE TASK COMPLETION CODES		* 01800000
-----		01810000
SPACE 1		01820000
TCCOPN EQU X'82'	PSUEDO-OPEN ERROR--INVALID DTF	01830000
SPACE 1		01840000
TCCCLS EQU X'83'	PSEUDO-CLOSE ERROR	01850000
SPACE 1		01860000
TCCDER EQU X'84'	I/O ERROR WHILE PRIMING OR	01870000
*	PURGING BUFFERS DURING OPEN/CLS	01880000
SPACE 1		01890000
TCCALC EQU X'8B'	PSEUDO-ALLOCATE UNOWNED DEVICE	01900000
SPACE 2		01910000
AIF (&DF EQ '0').END		01920000
.*		01930000
SPACE 1		01940000
* ERRORS DETECTED BY DISPLAY FORMAT FACILITY		01950000
SPACE 1		01960000
TCC238 EQU X'EE'	MODIFY DATA IS NOT ALLOWED	01970000
*	FOR AN INPUT FIELD.	01980000
TCC239 EQU X'EF'	AN INVALID MODIFY DATA	01990000
*	CHARACTER WAS GIVEN FOR PUT	02000000
*	OVERRIDE.	02010000
TCC240 EQU X'F0'	AN INVALID CURSOR CHARACTER	02020000
*	WAS GIVEN FOR PUT OVERRIDE.	02030000
TCC241 EQU X'F1'	AN INVALID TYPE CHARACTER WAS	02040000
*	GIVEN FOR PUT OVERRIDE.	02050000
TCC242 EQU X'F2'	LENGTH OF OUTPUT DATA IS NOT	02060000
*	CORRECT FOR PUT MESSAGE OR	02070000
*	PUT OVERRIDE.	02080000
TCC243 EQU X'F3'	THE MODEL NUMBER OF THE FORMAT	02090000
*	DOES NOT MATCH THE MODEL	02100000
*	NUMBER OF THE TERMINAL.	02110000
TCC244 EQU X'F4'	FIELD NAME PROVIDED IN THE	02120000
*	RECORD AREA FOR A PUT OVER-	02130000
*	RIDE DOES NOT MATCH A FIELD	02140000
*	NAME IN THE FDT FOR THE CURRENT	02150000
*	FORMAT.	02160000
TCC246 EQU X'F6'	THE FIELD DESCRIPTOR TABLE	02170000
*	SIZE IN THE PROGRAM APPENDED	02180000
*	STORAGE AREA IS NOT LARGE	02190000
*	ENOUGH TO CONTAIN THE FDT FOR	02200000
*	THE DESIRED FORMAT.	02210000
TCC247 EQU X'F7'	THE MODIFY DATA OPERATION IS	02212000
*	THE ONLY PUT OVERRIDE OPERATION	02214000
*	VALID FOR PRINTER GENERATED	02216000
*	FORMATS.	02218000
TCC248 EQU X'F8'	THE OUTPUT HOLD AREA MUST BE	02220000
*	AT LEAST 512 BYTES IN LENGTH	02230000
*	FOR BLOCKING OF OUTPUT TEXT.	02240000
TCC249 EQU X'F9'	DISK I/O ERROR RECEIVED DURING	02250000
*	OPERATION OF DFCE.	02260000
TCC245 EQU X'F5'	ERASE FIELD NOT VALID FOR AN	02261000
*	OUTPUT FIELD TYPE OR A SPD	02262000

*		FIELD TYPE. IT IS ONLY VALID	02263000
*		FOR AN INPUT OR OUTPUT/INPUT	02264000
*		FIELD TYPE.	02265000
TCC250	EQU	X'FA'	02270000
*		FROM AND TO TERMINALS NOT ON	02280000
*		SAME LINE, ATTACHED TO SAME	02290000
*		CONTROL UNIT, OR NOT BOTH	02300000
*		ONLINE FOR A COPY OPERATION.	02310000
TCC251	EQU	X'FB'	02320000
*		LINE BUFFER IS NOT LARGER OR	02330000
*		EQUAL IN LENGTH TO THE OUTPUT	02340000
*		HOLD AREA FOR DFF.	02350000
TCC252	EQU	X'FC'	02360000
*		THE DESIRED DISPLAY FORMAT WAS	02370000
*		NOT FOUND ON THE SPECIFIED	02380000
*		DISK PACK.	02390000
TCC253	EQU	X'FD'	02400000
*		NON-SELECTOR PEN DETECTABLE	02410000
*		FIELDS (INPUT,OUTPUT/INPUT)	02420000
*		RECEIVED FROM A TERMINAL. THE	02430000
*		INPUT WAS CAUSED BY A SELECTOR	02440000
*		PEN ATTENTION FIELD.	02450000
TCC254	EQU	X'FE'	02460000
*		NOT ENOUGH SPACE WAS ALLOCATED	02470000
*		FOR THE PROGRAM APPENDED	02480000
*		STORAGE AREA. NEED MORE SPACE	02490000
*		FOR TERMINALS OR FORMATS USED	02500000
*		BY THIS PROGRAM.	02510000
TCC255	EQU	X'FF'	02520000
*		ATTEMPTING TO GET, INVITE INPUT,	02530000
*		COPY FROM, PUT OVERRIDES, AND	02540000
*		THE TERMINAL DID NOT RECEIVE	02550000
*		AN INITIAL FORMAT PREVIOUSLY	02560000
*		FROM THIS PROGRAM.	
.*			02520000
.END	ANOP	. COMPLETION OF MACRO	02530000
.*			02540000
	SPACE 1		02550000
	MEND		02560000

MODULE-\$EIOB , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO                                00010000
$EIOB
*****                                00040000
.*                                     00050000
.*                                     00060000
.* |-----|-----|-----|          00070000
.* |         | IOBD   | OPERANDS (DESCRIBED BELOW) |          00080000
.* |         |         |         |                     |          00090000
.*                                     00100000
.* THIS MACRO GENERATES THE FIELD EQUATES FOR THE FOLLOWING DEVICES:
.* DISK
.*                                     00130000
.* DISK=Y      CREATES IOB FIELD EQUATES FOR DISK(5444,5445)
.*                                     00340000
.*                                     00360000
.* *****                                00370000
.* TEXT                                    00380000
*****                                00410000
* IOB COMMON EQUATES *                  00420000
*****                                00430000
SPACE                                    00440000
IOBECB EQU 0      WAIT/POST BYTE -- BYTE 1 OF ECB          00450000
IOBCOM EQU 1      COMPLETION CODE -BYTE 2 OF ECB          00460000
IOBCMP EQU 2      COMPLETION CODE - BYTE 3 OF ECB          00470000
IOBCHN EQU 6      IOS QUEUE CHAIN POINTER                 00480000
IOBQB EQU 7       Q-BYTE OF SIO                          00490000
IOBRB EQU 8       R-BYTE OF SIO                          00500000
IOBEID EQU 9      ERP MODULE DISPLACEMENT ID             00510000
IOBDAT EQU 11     DATA (LIO) ADDRESS                    00520000
IOBSNS EQU 13     SENSE AREA                             00530000
IOBFLG EQU 14     FLAG BITS                              00540000
IOBERR EQU 15     ERROR COUNTS                          00550000
IOBTCB EQU 17     TCB ADDRESS                            00560000
SPACE 3                                                  01560000
* EQUATES FOR DISK (5444 AND 5445)                    01570000
IOBWRK EQU 18     IOS PARTIAL COMPLETION CODE            01580000
IOBFL2 EQU 19     5445 SECOND FLAG BYTE                  01590000
IOBCC EQU 20      5445 CYLINDER; 5444 NOT USED           01600000
IOBHH EQU 21      5445 HEAD                              01610000
IOBR EQU 22       5445 RECORD                            01620000
IOBN EQU 23       5445 NUMBER OF RECORDED (-1)           01630000
IOBDAD EQU 25     PTR TO 5445 10 BYTE DISK ADDRESS       01640000
IOBDCH EQU 27     DATA MGMT CHAIN POINTER               01650000
IOBDTF EQU 29     ADDRESS OF ASSOCIATED DTF              01660000
* OPERATED UPON                                       01670000
IOBCB EQU 21      5444 CYLINDER                          01680000
IOBSB EQU 22      5444 SECTOR                            01690000
IOBNB EQU 23      5444 NUMBER OF RECORDS (-1)           01700000
WRIDFB EQU 24     01710000
WRIDCB EQU 25     01720000
WRIDSB EQU 26     01730000
RDIDFB EQU 27     01740000
RDIDCB EQU 28     01750000
RDIDSB EQU 29     01760000
SPACE 1
* THE FOLLOWING EQUATES ARE FOR THOSE PROGRAMS WHO USE READ 01780000
* OR WRITE HA AND R0 COUNT OR WRITE COUNT KEY DATA COMMANDS. 01790000
* FIELD MUST IMMEDIATELY FOLLOW IOB IF USING THESE EQUATES. 01800000
IOBF EQU 30       5445 FLAG BYTE IN DDCF                01810000
IOBCYL EQU 32     5445 CYLINDER ADDR IN DDCF            01820000
IOBHD EQU 34     5445 HEAD IN DDCF                     01830000
```

IOBREC	EQU	35	5445 RECORD IN DDCF	01840000
IOBKEY	EQU	36	5445 KEY LENGTH IN DDCF	01850000
IOBDTA	EQU	38	5445 DATA LENGTH IN DDCF	01860000
IOBNUM	EQU	39	5445 NUMBER OF RECORDS (1)	01870000
	MEND			01900000

```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 5 AJS 00030000
.* * 00040000
.* NAME:  $ECPL * 00050000
.* * 00060000
.* FUNCTION:  DEFINE SYMBOLS FOR COMMUNICATIONS PARAMETER LIST FOR: * 00070000
.* * OFFSETS OF CELLS IN THE PARAMETER LIST. * 00080000
.* * VALUES OF THE OPERATION CODE/MODIFIER CELL. * 00090000
.* * VALUES OF THE RETURN CODE CELL. * 00100000
.* * 00110000
.* INPUT OPERANDS: * 00120000
.* * DSPL-Y/N GENERATE THE OFFSETS INTO THE PARAMETER LIST. * 00130000
.* * DEFAULT IS Y. * 00140000
.* * OPCD-Y/N GENERATE THE VALUES OF THE OPCODES OR * 00150000
.* * MODIFIERS.  DEFAULT IS Y. * 00160000
.* * RTNCD-Y/N GENERATE ALL VALUES OF THE RETURN CODES. * 00170000
.* * DEFAULT IS Y. * 00180000
.* * -EXCP GENERATE ONLY EXCEPTION RETURN CODES. * 00190000
.* * -IO GENERATE ONLY MLTA/MLMP I/O RETURN CODES. * 00200000
.* * -IO3270 GENERATE ONLY 3270 I/O RETURN CODES. * 00210000
.* * -IO3735 GENERATE ONLY 3735 I/O RETURN CODES. * 00220000
.* * 00230000
.* ***** 00240000
.* $ECPL &DSPL-Y,&OPCD-Y,&RTNCD-Y 00250000
.* TEXT 00260000
.* AIF (&DSPL EQ 'N').OPCD 00270000
.* ***** 00280000
.* C O M M U N I C A T I O N S P A R A M E T E R L I S T * 00290000
.* ***** 00300000
.* SPACE 00310000
CCPPL EQU 0 BEGINNING OF PARAMETER LIST. 00320000
.* SPACE 1 00330000
PLRTC EQU CCPPL+2-1 REQUESTERS RETURN CODE. 00340000
PLCHN EQU PLRTC PARM LIST CHAIN PTR. * CM ONLY * 00350000
.* SPACE 1 00360000
PLOPM EQU PLRTC+1 REQUESTERS OP CODE MODIFIERS. 00370000
PLOPC EQU PLOPM+1 REQUESTERS OP CODE. 00380000
.* SPACE 1 00390000
PLOUTL EQU PLOPC+2 OUTPUT LENGTH. 00400000
PLEFFL EQU PLOUTL EFFECTIVE INPUT LENGTH. 00410000
PLASID EQU PLOUTL TERMINAL ATTRIBUTE SET ID. 00420000
.* SPACE 1 00430000
PLINL EQU PLOUTL+2 MAXIMUM INPUT LENGTH. 00440000
.* SPACE 1 00450000
PLRECA EQU PLINL+2 RECORD AREA ADDRESS. 00460000
.* SPACE 1 00470000
PLTUBA EQU PLRECA+2 TUB ADDRESS. ** CM ONLY ** 00480000
.* SPACE 1 00490000
PL$OPM EQU PLTUBA+1 INTERNAL OPERATION CODE. 00500000
.* SPACE 1 00510000
PL$OPC EQU PL$OPM+1 SAVED INTERNAL OP.*CM BSCA ONLY* 00520000
PL$MCT EQU PL$OPC MCT INDICES. *CM MLTA ONLY* 00530000
.* SPACE 1 00540000
PL$RTC EQU PL$OPC+2 INTERNAL RETURN CODE. ** CM ** 00550000
PL$TNT EQU PL$RTC ADDRESS OF TNT ENTRY ** II ** 00560000
.* SPACE 1 00570000
PLECB EQU PL$RTC+1 PARAMETER LIST'S ECB. 00580000
.* SPACE 1 00590000

```

```

PLEN EQU PLECB-CCPPL+3          LENGTH OF TP REQUEST PARM LIST. 00600000
SPACE 3                          00610000
.OPCD AIF (&OPCD EQ 'N').RTNCD  00620000
*-----*                          * 00630000
*                                * 00640000
*                                * 00650000
*                                * 00660000
*                                * 00670000
*                                * 00680000
*                                * 00690000
*                                * 00700000
*                                * 00710000
*                                * 00720000
*                                * 00730000
*                                * 00740000
*                                * 00750000
*                                * 00760000
*                                * 00770000
*                                * 00780000
*-----*                          * 00790000
SPACE 1                          00800000
*-----*                          * 00810000
*                                * 00820000
*                                * 00830000
*                                * 00840000
*                                * 00850000
*                                * 00860000
*                                * 00870000
*                                * 00880000
*                                * 00890000
*                                * 00900000
*                                * 00910000
*                                * 00920000
*                                * 00930000
*                                * 00940000
*                                * 00950000
*                                * 00960000
*                                * 00970000
*                                * 00980000
*                                * 00990000
*                                * 01000000
*                                * 01010000
*                                * 01020000
*                                * 01030000
*                                * 01040000
*                                * 01050000
*                                * 01060000
*                                * 01070000
*                                * 01075000
*                                * 01076000
*                                * 01077000
*-----*                          * 01080000
SPACE 1                          01090000
*-----*                          * 01100000
*                                * 01110000
*                                * 01120000
*                                * 01130000
*                                * 01140000

```

* X'0000'	SHUTDOWN INQUIRY	* 01150000
* X'0001'	GET RECORD	* 01160000
* X'0002'	PUT RECORD	* 01170000
* X'0003'	PUT THEN GET	* 01180000
* X'0004'	ACCEPT INPUT	* 01190000
* X'0005'	INVITE INPUT	* 01200000
* X'0006'	PUT-NO-WAIT RECORD	* 01210000
* X'0008'	GET TERMINAL ATTRIBUTES	* 01220000
* X'0009'	ACQUIRE TERMINAL	* 01230000
* X'000A'	RELEASE TERMINAL (DROP)	* 01240000
* X'0011'	GET - RVI RESPONSE	* 01250000
* X'0014'	WAIT OPERATION	* 01255000
* X'0019'	ACQUIRE AND SET ATTRIBUTES	* 01260000
* X'001A'	RELEASE TERMINAL (KEEP)	* 01270000
* X'0021'	GET BLOCK	* 01280000
* X'0022'	PUT BLOCK	* 01290000
* X'0023'	PUT-THEN-GET BLOCK	* 01300000
* X'0026'	PUT-NO-WAIT BLOCK	* 01310000
* X'0031'	GET MESSAGE	* 01320000
* X'0032'	PUT MESSAGE	* 01330000
* X'0033'	PUT-THEN-GET MESSAGE	* 01340000
* X'0042'	DFP COPY OPERATION	* 01350000
* X'0052'	DFP ERASE OPERATION	* 01360000
* X'0072'	RUF PUT MESSAGE	* 01365000
* X'0401'	STOP INVITE INPUT	* 01370000
*		* 01380000
*	SYSTEM-ISSUED OPERATION CODES	* 01390000
*		* 01400000
* X'8002'	SYSTEM PUT	* 01410000
* X'8005'	SYSTEM INVITE INPUT	* 01420000
* X'8006'	SYSTEM PUT-NO-WAIT	* 01430000
* X'8007'	SYSTEM PUT-NO-WAIT INVITE	* 01440000
* X'8401'	SYSTEM STOP INVITE	* 01450000
* X'8403'	SYSTEM PURGE OF TERMINAL	* 01460000
* X'8413'	SYSTEM PURGE OF WHOLE LINE	* 01470000
* X'A006'	SYSTEM PUT DISCONNECT	* 01480000
* X'A413'	SYSTEM PURGE AND PUT DISCONNECT	* 01490000
*		* 01500000
*		* 01510000
-----		* 01510000
SPACE		01520000
*		* 01530000
*	PL\$OPM-INTERNAL OP CODE	* 01540000
OPGETM EQU BIT0	1-NEED TO GETMAIN STORAGE FOR	* 01550000
*	* THIS TP REQUEST.	* 01560000
OPGETQ EQU BIT1	1-NEED GETMAIN TO FINISH	* 01570000
*	* QUEUEING THIS REQUEST.	* 01580000
OPNPST EQU BIT2	DON'T POST PURGED PUT.MLTA ONLY	* 01590000
OPBNOP EQU BIT3	DON'T POST ERP OP END.BSCA ONLY	* 01600000
*		* 01610000
* BITS 4-7 ARE THE SAME AS FOR PLOPC.		* 01620000
-----		* 01630000
SPACE		01640000
*		* 01650000
*	PL\$OPC-SAVED INTERNAL OP ** BSCA ONLY **	* 01660000
OPREFSH EQU BIT0	SEND REFRESH MSG TO 3270.	* 01670000
OPLSNS EQU BIT1	POLL FOR TERMINAL STATUS.	* 01680000
OPUSER EQU BIT2	FUNCTION, 1-SYSTEM, 0-USER.	* 01690000
OPSTCM EQU BIT3	STOP INVITE PARM LIST HANDLED.	* 01700000
*		* 01710000
* BITS 4-7 ARE THE SAME AS FOR PLOPC.		* 01720000


```

*-----* 01730000
SPACE 3 01740000
.RTNCD AIF (&RTNCD EQ 'N').END 01750000
*-----* 01760000
* PL$RTC-INTERNAL RETURN CODE * 01770000
AIF (&RTNCD EQ 'IO').IO 01780000
AIF (&RTNCD EQ 'IO3270').IO32 01790000
AIF (&RTNCD EQ 'IO3735').IO37 01800000
* 01810000
* SUCCESSFUL OP CODE * 01820000
* PL$RTC=00XX, FOR XX AS FOLLOWS: * 01830000
RCOK EQU X'00' SUCCESSFUL OPERATION. * 01840000
* 01850000
* EXCEPTION RETURN CODES * 01860000
* PL$RTC=00XX, FOR XX AS FOLLOWS: * 01870000
RCXDTR EQU X'01' DATA TRUNCATED. * 01880000
RCXEOT EQU X'02' EOT RECEIVED. * 01890000
RCXEDT EQU X'03' EOT RECEIVED AND DATA TRUNCATED * 01900000
* DATA TRANSFER IS VALID ONLY FOR OPERATIONS WHICH YIELD RETURN * 01910000
* CODES OF 0, +1, +2, +3, OR +6. * 01920000
RCXSHD EQU X'04' SHUTDOWN REQUESTED. * 01930000
RCXDPD EQU X'05' DATA PENDING ON BSCA LINE. * 01940000
RCXRVI EQU X'06' RVI/TERMINAL INTERRUPT RECEIVED * 01950000
RCXCLR EQU X'07' 3270 CLEAR KEY RECEIVED. * 01960000
RCXNAV EQU X'08' TERMINAL NO LONGER AVAILABLE. * 01970000
RCXOFF EQU X'09' TERMINAL OFFLINE. * 01980000
RCXSPI EQU X'0A' STOP INVITE INPUT SUCCESSFUL. * 01990000
RCXNAQ EQU X'0B' ACQUIRE TERMINAL FAILED. * 02000000
RCXNIQ EQU X'7E' SPI FAILED-NO INVITE IN Q. * 02010000
RCXSPF EQU X'7F' STOP INVITE INPUT FAILED. * 02020000
AIF (&RTNCD EQ 'EXCP').REND 02030000
.IO ANOP * 02040000
* 02050000
* I/O ERROR RETURN CODES * 02060000
* PL$RTC=FFXX, FOR XX AS FOLLOWS: * 02070000
RCRDCK EQU X'FF' DATA CHECK. * 02080000
RCRTRN EQU X'FE' TRANSLATION ERROR. * 02090000
RCRLST EQU X'FD' LOST DATA. * 02100000
RCRPBS EQU X'FC' PERMANENT BI-SYNC ERROR. * 02110000
RCRABN EQU X'FB' ABNORMAL RESPONSE. * 02120000
RCRXRA EQU X'FA' TRANSMIT/RECEIVE ABORT. * 02130000
RCRATO EQU X'F9' ADDRESSING TIMEOUT. * 02140000
RCRTTO EQU X'F8' TEXT TIMEOUT. * 02150000
RCRWTO EQU X'F7' WACK/TTD EXPIRATION. * 02160000
RCRNOC EQU X'F6' NO CONNECTION. * 02170000
RCRIID EQU X'F5' INVALID ID. * 02180000
RCRABD EQU X'F4' ABORT, DISCONNECT. * 02190000
RCRADC EQU X'F3' ADAPTER CHECK. * 02200000
RCRNAK EQU X'F2' NEGATIVE RESPONSE TO ADDRESSING * 02210000
AIF (&RTNCD EQ 'IO').REND 02220000
.IO32 ANOP 02230000
* 02240000
* 3270 I/O ERROR RETURN CODES * 02250000
* PL$RTC=FFXX, FOR XX AS FOLLOWS: * 02260000
RCR2DU EQU X'EC' DEVICE UNAVAIL OR NOT READY. * 02270000
RCR2ED EQU X'EA' EQUIPMENT CHECK, DEVICE END. * 02280000
RCR2TE EQU X'E9' TCU DETECTION OF BSCA ERROR. * 02290000
RCR2CD EQU X'E8' CTL CHECK, DATA CHK, DEV BUSY. * 02300000
RCR2PD EQU X'E7' DATA CHECK ON COPY COMMAND. * 02310000
RCR2PO EQU X'E6' OPERATION CHECK ON COPY COMMAND * 02320000

```

```

RCR2PB EQU X'E5'          DEVICE BUSY ON COPY COMMAND. * 02330000
RCR2PC EQU X'E4'          CTL CHK/OP CHK/DATA CHK ON COPY* 02340000
RCR1DR EQU X'E3'          INVALID DATA RECEIVED FROM A * 02350000
*                          3270 USING DFF SUPPORT. * 02360000
      AIF (&RTNCD EQ 'IO3270').REND 02370000
.IO37 ANOP 02380000
* 02390000
* 3735 I/O ERROR RETURN CODES * 02400000
* PL$RTC=FFXX, FOR XX AS FOLLOWS: * 02410000
RCR5SR EQU X'D8'          ATTEMPTED SEND BEFORE RECEIVE. * 02420000
RCR5IC EQU X'D7'          ILLEGAL CHARACTER. * 02430000
RCR5OF EQU X'D6'          BUFFER OVERFLOW. * 02440000
RCR5DF EQU X'D5'          DISK FULL. * 02450000
RCR5RF EQU X'D4'          DIRECTORY FULL. * 02460000
RCR5UH EQU X'D3'          UNDEFINED HEADER. * 02470000
RCR5DE EQU X'D2'          3735 DISK ERROR. * 02480000
.IO41 ANOP 02480500
RCR7ZR EQU 256            BASE FOR NEGATIVE NUMBERS 02481000
RCR7TE EQU RCR7ZR-50     TRANSPERANCY ERROR OCCURED 02481500
RCR7NA EQU RCR7ZR-51     NO ACTIVITY ON THE LINE 02482000
RCR7DC EQU RCR7ZR-52     DATA CHECK 02482500
RCR7LB EQU RCR7ZR-53     RECEIVED LINE BID ERROR 02483000
RCR7WL EQU RCR7ZR-54     WRONG LENGTH ERROR 02483500
RCR7RP EQU RCR7ZR-55     RESET PRESSED ON 3741 02484000
RCR7SC EQU RCR7ZR-56     SECURITY CHECK 02484500
RCR7DO EQU RCR7ZR-57     DISK OVERFLOW 02485000
RCR7BE EQU RCR7ZR-58     BAD EXTENT ERROR 02485500
RCR7BT EQU RCR7ZR-59     BOTH STATIONS TRANSMIT. 02486000
RCR7LE EQU RCR7ZR-60     LENGTH ERROR 02486500
RCR7NF EQU RCR7ZR-61     NO RECORD FOUND 02487000
RCR7SE EQU RCR7ZR-62     SEEK ERROR 02487500
RCR7RE EQU RCR7ZR-63     READ ERROR 02488000
RCR7WE EQU RCR7ZR-64     WRITE ERROR 02488500
RCR7NR EQU RCR7ZR-65     3741 NOT READY 02489000
RCR7WP EQU RCR7ZR-66     DISKETTE IS WRITE PROTECTED. 02489500
.REND ANOP 02490000
* -----* 02500000
      SPACE 2 02510000
.END ANOP 02520000
      MEND 02530000

```

MACRO		00010000
*****		00020000
.*		* 00030000
.* NAME: \$ETAS		* 00040000
.*		* 00050000
.* MODIFICATION LEVEL: VERSION 8, MODIFICATION LEVEL 0 OF 5702-SC1		* 00060000
.*		* 00070000
.* FUNCTION:		* 00080000
.*		* 00090000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- DEFINE		* 00100000
.* SYMBOLS FOR TERMINAL ATTRIBUTE SET -- FOR:		* 00110000
.*		* 00120000
.* . OFFSETS OF FIELDS		* 00130000
.* . FOR VALUES WITHIN FIELDS		* 00140000
.*		* 00150000
.* INPUT OPERANDS:		* 00160000
.*		* 00170000
.* . NONE		* 00180000
.*		* 00190000
*****		00200000
\$ETAS		00210000
.*		00220000
TEXT		00230000
.*		00240000
*****		00250000
* TERMINAL ATTRIBUTES SET		* 00260000
*****		00270000
SPACE 2		00280000
TASCCP EQU 0	BEGINNING OF TAS ENTRY	00290000
SPACE 1		00300000
* ATTRIBUTES BYTE 1		00310000
SPACE 1		00320000
TASAT1 EQU TASCCP-1+1	ATTRIBUTE BYTE 1	00330000
SPACE 1		00340000
TASTRN EQU BIT0	0-TRANSLATE	00350000
*	1-DON'T TRANSLATE	00360000
TASCAS EQU BIT1	0-FORCE UPPER CASE TRANSLATE	00370000
*	1-DO NOT FORCE UPPER CASE	00380000
TASCNC EQU BIT2	0-ON SWITCHED LINE, ANSWER	00390000
*	1-ON SWITCHED LINE, CALL OUT	00400000
TASAUT EQU BIT3	0-ON SW LINE, MANUAL CONNECT	00410000
*	1-ON SW LINE, AUTO CONNECT	00420000
TASRSV EQU BIT4+BIT5+BIT6	RESERVED BITS	00430000
TASDFE EQU BIT7	0-DO NOT USE DFE	00440000
*	1-USE DISPLAY FORMAT FACILITY	00450000
SPACE 1		00460000
* ATTRIBUTES BYTE 2		00470000
SPACE 1		00480000
TASAT2 EQU TASAT1+1	ATTRIBUTE BYTE 2	00490000
SPACE 1		00500000
TASREC EQU BIT0	0-DATA FORMAT IS NOT RECORD MODE	00510000
*	1-DATA FORMAT IS RECORD MODE	00520000
TASBLK EQU BIT1	0-DATA FORMAT NOT BLOCK MODE	00530000
*	1-DATA FORMAT IS BLOCK MODE	00540000
TASMSG EQU BIT2	0-DATA FORMAT NOT MESSAGE MODE	00550000
*	1-DATA FORMAT IS MESSAGE MODE	00560000
TASITB EQU BIT3	0-NO ITB SUPPORT	00570000
*	1-ITB SUPPORT	00580000
TASTSP EQU BIT4	0-NON-TRANSPARENCY MODE	00590000

*			1-TRANSPARENCY MODE	00600000
TASV FY EQU	BIT5		0-VERIFY EXCHANGE ID	00610000
*			1-NO EXCHANGE ID VERIFICATION	00620000
TAS PAN EQU	BIT6		0-NO SPANNED RECORD SUPPORT	00630000
*			1-SUPPORT SPANNED RECORDS	00640000
TAS VRL EQU	BIT7		0-NO VARIABLE LENGTH SUPPORT	00650000
*			1-SUPPORT VARIABLE LENGTH RECORD	00660000
	SPACE 1			00670000
* RECORD AND	BLOCK LENGTHS			00680000
	SPACE 1			00690000
TAS RCL EQU	TASAT2+2	RECORD LENGTH		00700000
	SPACE 1			00710000
TAS BKF EQU	TASRCL+1	BLOCKING FACTOR		00720000
	SPACE 1			00730000
TAS LN EQU	TASBKF-TASCCP+1	LENGTH OF TERMINAL ATTRIBUTE SET		00740000
	SPACE 1			00750000
	MEND			00760000

```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 8 * 00030000
.* * 00040000
.* NAME:  $ETUB * 00050000
.* * 00060000
.* FUNCTION: * 00070000
.* . CCP GENERATION MACRO INSTRUCTION -- DEFINE * 00080000
.* SYMBOLS FOR TERMINAL UNIT BLOCK -- FOR: * 00090000
.* . OFFSETS OF FIELDS * 00100000
.* . VALUES WITHIN FIELDS * 00110000
.* * 00120000
.* INPUT OPERANDS: * 00130000
.* SCS-Y/N GENERATE THE BIT EQUATES FOR TUBSCS. * 00140000
.* CHR-Y/N GENERATE THE BIT EQUATES FOR TUBCHR. * 00150000
.* AT1-Y/N GENERATE THE BIT EQUATES FOR TUBAT1. * 00160000
.* AT2-Y/N GENERATE THE BIT EQUATES FOR TUBAT2. * 00170000
.* AT3-Y/N GENERATE THE BIT EQUATES FOR TUBAT3. * 00180000
.* AT4-Y/N GENERATE THE BIT EQUATES FOR TUBAT4. * 00185000
.* PHY-Y/N GENERATE THE BIT EQUATES FOR TUBPHY. * 00190000
.* ***** 00200000
.* $ETUB &SCS-Y,&CHR-Y,&AT1-Y,&AT2-Y,&AT3-Y,&AT4-Y,&PHY-Y 00210000
.* TEXT 00220000
***** 00230000
* T E R M I N A L U N I T B L O C K * 00240000
***** 00250000
TUBCCP EQU 0 BEGINNING OF TUB 00260000
TUBID EQU TUBCCP+1 TERMINAL PHYSICAL ID (2 BYTES) 00270000
SPACE 1 00280000
TUBPL@ EQU TUBID+2 LAST PARAMETER LIST FOR TUB 00290000
TUBER@ EQU TUBPL@+2 PARAMETER LIST IN ERP 00300000
TUBDM@ EQU TUBER@+2 PARAMETER LIST AT TIME OF DME 00310000
SPACE 1 00320000
TUBSID EQU TUBDM@+1 INTERNAL TERMINAL ID (BSCA ONLY) 00330000
SPACE 1 00340000
TUBSCS EQU TUBSID+1 BSCA STATUS BYTES (BSCA ONLY) 00350000
AIF (&SCS NE 'Y').TMA 00360000
TUBCLR EQU BIT0 * 1-3270 CLEAR KEY DEPRESSED 00370000
TUBDME EQU BIT1 * 1-DME CHECK SUCCESSFUL 00380000
TUBDMF EQU BIT2 * 1-DME CHECK FAILED 00390000
TUBSWA EQU BIT3 * 1-SWITCHED LINE DEVICE HANDLED 00400000
* * DURING PREVIOUS SYS PURGE. 00410000
TUBSSP EQU BIT4 * 1-STOP STATUS POLLING 00420000
TUBRUF EQU BIT5 * 1-RUF DATA ON SCREEN(3270 DFF) 00430000
TUBBPT EQU BIT7 * 1-BUSY PRINT ALLOWED 00435000
.TMA ANOP 00440000
SPACE 1 00450000
TUBTMA EQU TUBSCS 2 BYTE TERMINAL ADDR (MLTA ONLY) 00460000
* OVERLAYS TUBSID AND TUBSCS 00470000
SPACE 1 00480000
* ATTRIBUTES AND STATUS BYTES 00490000
TUBCHR EQU TUBTMA+1 TERMINAL CHARACTERISTICS 00500000
AIF (&CHR NE 'Y').AT1 00510000
TUBLNE EQU BIT0 1-BSCA LINE 00520000
TUB@SL EQU BIT1 1-TYPWTR DEVICE AT START OF LINE 00530000
TUBTYP EQU BIT2 1-MLTA TYPEWRITER DEVICE 00540000
TUBNID EQU BIT3 1-NEED IDLES AT BEGIN OF LINE 00550000
TUBCMN EQU BIT4 1-COMMAND (CAPABLE) TERMINAL 00560000
TUBMCT EQU BIT5 1-MULTI-COMPONENT TERMINAL 00570000

```

TUBOUT	EQU	BIT6	1-TERMINAL CAPABLE OF OUTPUT	00580000
TUBINP	EQU	BIT7	1-TERMINAL CAPABLE OF INPUT	00590000
.AT1	ANOP			00600000
	SPACE	1		00610000
TUBAT1	EQU	TUBCHR+1	TERMINAL ATTRIBUTES BYTE 1	00620000
	AIF	(&AT1 NE 'Y').AT2		00630000
TUBKNM	EQU	BIT0	1-THIS TUB IS FOR CONSOLE	00640000
TUBDPY	EQU	BIT1	1-IN USE BY DISPLAY	00650000
TUBONL	EQU	BIT2	1-TERMINAL IS ONLINE	00660000
TUBSGN	EQU	BIT3	1-COMMAND TERMINAL IS SIGNED ON	00670000
TUBQUE	EQU	BIT4	1-COMMAND TERMINAL IN /Q STATUS	00680000
TUBREQ	EQU	BIT5	1-TERMINAL IS REQUESTER OF PGM	00690000
TUBSWC	EQU	BIT6	1-TERMINAL ON SWITCHED LINE	00700000
TUBOFF	EQU	BIT7	1-CMD TERM SIGNOFF DEFAULT=HOLD	00710000
.AT2	ANOP			00720000
	SPACE	1		00730000
TUBAT2	EQU	TUBAT1+1	TERMINAL ATTR BYTE 2	00740000
	AIF	(&AT2 NE 'Y').AT3		00750000
TUBDTA	EQU	BIT0	1-TERMINAL IN DATA MODE	00760000
TUBCMD	EQU	BIT1	1-TERMINAL IN COMMAND MODE	00770000
*			* DTA/CMD/TERMINAL MODE	00780000
*			* 0 0 INITIAL	00790000
*			* 0 1 COMMAND	00800000
*			* 1 0 DATA	00810000
*			* 1 1 COMMAND INTERRUPT	00820000
TUBIMI	EQU	BIT2	1-OUTSTANDING DATA FROM PGM RQST	00830000
TUBIIS	EQU	BIT3	1-INVITE OR GET SCHEDULED	00840000
TUBIIQ	EQU	BIT4	1-INVITE COMPLETE -- ON TCBIQ	00850000
*			0-NO OUTSTANDING COMPLETE INVITE	00860000
TUBOLT	EQU	BIT5	1-CURRENTLY RUNNING ONLINE TEST	00870000
TUBAPP	EQU	BIT6	1-ALLOCATION PENDING ON THIS TUB	00880000
TUBOWN	EQU	BIT7	1-TERMINAL NOW COMMUNICATING ON	00890000
*			* BSCA LINE--EOT NOT SENT/RECD	00900000
.AT3	ANOP			00910000
	SPACE	1		00920000
TUBAT3	EQU	TUBAT2+1	TERMINAL ATTRIBUTE BYTE 3	00930000
	AIF	(&AT3 NE 'Y').INQ		00940000
*		BIT0	(RESERVED)	00950000
TUBSPF	EQU	BIT1	1-STOP POLLING FAILED	00960000
TUBCNC	EQU	BIT2	1-/RELEASE BY TERMINAL OPERATOR	00970000
TUBERP	EQU	BIT3	1-TERMINAL ERROR -- AWAITING ERP	00980000
TUBALC	EQU	BIT4	1-PHYSICALLY ALLOCATED	00990000
*			BUT NOT LOGICALLY	01000000
*			ALLOCATED,(NO I/O POSSIBLE)	01010000
*			0-IS ALLOCATED TO THE	01020000
*			TCB POINTED TO BY TUBTCB	01030000
TUBVFP	EQU	BIT5	1-VARY OFFLINE PENDING	01040000
TUBSPP	EQU	BIT6	1-STOP POLLING PENDING TO TERM'L	01050000
*			* IN COMMAND INTERRUPT MODE	01060000
TUBSWL	EQU	BIT7	1-DISCARD ABORTED INPUT DATA	01070000
.INQ	ANOP			01080000
	SPACE	1		01090000
* QUEUE CHAIN FIELDS				01100000
TUBINQ	EQU	TUBAT3+2	GET INVITE INPUT QUEUE	01110000
*			POINTS TO NEXT TUB IN CHAIN	01120000
TUBTCB	EQU	TUBINQ+2	@ OF TCB	01130000
TUBTUB	EQU	TUBTCB+2	NEXT TUB ADDR BY TCBTUB	01140000
TUBPST	EQU	TUBTUB	NEXT TUB ON ALLOCATION QUEUE	01150000
TUBLCB	EQU	TUBTUB+2	ADDR OF LINE DTF (LCB)	01160000
TUBDTF	EQU	TUBLCB		01170000

TUBTNT EQU	TUBLCB+2	ADDR OF TNT ENTRY	01180000
TUBFSB EQU	TUBTNT+2	ADDR OF 1ST ENTRY FOR THIS	01190000
*		TERMINAL IN FILE SPEC BLOCK	01200000
	SPACE 1		01210000
* ONLINE TEST FIELDS			01220000
TUBOTC EQU	TUBFSB+1	TERMINAL ONLINE TEST CONTROL BYT	01230000
*		M L T A SPECS *	01240000
TUBLOP EQU	BIT0	* 1-LOOPING TEST SPECIFIED	01250000
TUBALL EQU	BIT1	* 1-MULTIPLE TESTS SPECIFIED	01260000
TUBTNR EQU	BIT4+BIT5+BIT6+BIT7	BINARY NUMBER OF SINGLE TEST	01270000
*		THIS BYTE - X'FF' SPECIFIES STOP	01280000
	SPACE 1		01290000
TUBCMA EQU	TUBOTC+1	MCT COMPONENT INDEX FOR OLT	01300000
	SPACE 1		01310000
* TERMINAL ATTRIBUTE SET FIELDS			01320000
TUBTAS EQU	TUBCMA+1	INDEX OF STANDARD ATTRIBUTE SET	01330000
	SPACE 1		01340000
TUBTA1 EQU	TUBTAS+1	1ST BYTE OF TERMINAL ATTRIBUTES	01350000
TUBTA2 EQU	TUBTA1+1	2ND BYTE OF TERMINAL ATTRIBUTES	01360000
TUBRCL EQU	TUBTA2+2	BSCA RECORD LENGTH	01380000
TUBVCS EQU	TUBRCL	1-BYTE SAVE AREA FOR I/O	01390000
*		COMPONENTS FOR MLTA MCT	01400000
TUBBKF EQU	TUBRCL+1	BSCA BLOCKING FACTOR	01410000
	SPACE 1		01420000
TUBCAS EQU	TUBTAS+5	DEFINES ALL 5 PRECEDING BYTES	01430000
	SPACE 1		01440000
* TERMINAL TYPE FIELD			01450000
TUBPHY EQU	TUBCAS+1	PHYSICAL TERMINAL TYPE--VALUES:	01460000
	AIF (&PHY NE 'Y').PCS		01470000
TUBCON EQU	0	0 - CONSOLE	01480000
TUBMLT EQU	1	1 - MLTA NON-1050	01490000
TUB105 EQU	2	2 - MLTA 1050	01500000
TUB7M1 EQU	3	3 - 3277/84/86 MODEL 1 (480)	01510000
TUB7M2 EQU	4	4 - 3277/84/86 MODEL 2 (1920)	01520000
TUB5M1 EQU	5	5 - 3275 MODEL 1 (480)	01530000
TUB5M2 EQU	6	6 - 3275 MODEL 2 (1920)	01540000
TUB375 EQU	7	7 - 3735	01550000
TUBCPU EQU	8	8 - CPU	01560000
TUB374 EQU	9	9 - 3741	01565000
.PCS ANOP			01570000
	SPACE 1		01580000
* MULTI-COMPONENT TERMINAL PRINCIPAL COMPONENT INDEXES			01590000
TUBPCS EQU	TUBPHY+1	PRINCIPAL COMPONENTS FOR 1050	01600000
TUBVHR EQU	TUBPCS+1	SAVE AREA FOR TUBCHR FOR MLTA	01610000
	SPACE 1		01620700
TUBAT4 EQU	TUBVHR+1	TERMINAL ATTRIBUTE BYTE 3	01621400
	AIF (&AT4 NE 'Y').AT4		01622100
*	BIT0	(NOT USED)	01622800
*	BIT1	(NOT USED)	01623500
*	BIT2	(NOT USED)	01624200
*	BIT3	(NOT USED)	01624900
*	BIT4	(NOT USED)	01625600
*	BIT5	(NOT USED)	01626300
TUBBSY EQU	BIT6	1-TUB TO BE CHECKED FOR BUSY	01627000
TUBWAT EQU	BIT7	1-WAIT FOR RESPONSE TO BUSY	01627700
.AT4 ANOP			01628400
	SPACE 2		01630000
TUBLN EQU	TUBAT4+1	LENGTH OF DATA MODE TUB	01640000
	SPACE 1		01650000
* END OF TUB FOR DATA MODE TERMINAL			01660000

	SPACE 1		01670000	
TUBPL	EQU	TUBAT4+1	PARAMETER LIST FOR SYSTEM INVITE	01680000
*			* OR PUT-NO-WAIT INVITE	01690000
*			* 19 BYTES	01700000
	SPACE 1			01710000
TUBLNC	EQU	TUBLN+19	LENGTH OF TUB FOR COMMAND MODE	01720000
*			* TERMINAL	01730000
	SPACE 1			01740000
*	END	OF TUB FOR COMMAND MODE	TERMINAL	01750000
	SPACE 1			01760000
TUBSTP	EQU	X'04'	TUB STOPPER BYTE	01770000
	SPACE 1			01780000
	MEND			01790000


```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 7 * 00030000
.* * 00040000
.* NAME:  $ELCB * 00050000
.* * 00060000
.* FUNCTION: * 00070000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- DEFINE * 00080000
.* SYMBOLS FOR LINE CONTROL BLOCK -- FOR: * 00090000
.* . OFFSETS OF FIELDS * 00100000
.* . VALUES WITHIN FIELDS * 00110000
.* * 00120000
.* INPUT OPERANDS: * 00130000
.* . BSCA=YES/Y/1/NO/N/0 * 00140000
.* . MLTA=YES/Y/1/NO/N/0 * 00150000
***** 00160000
    $ELCB &BSCA-1,&MLTA-1 00170000
    TABLE &BSCA 00180000
YES  TABDF 1 00190000
Y  TABDF 1 00200000
NO  TABDF 0 00210000
N  TABDF 0 00220000
    TABLE &MLTA 00230000
YES  TABDF 1 00240000
Y  TABDF 1 00250000
NO  TABDF 0 00260000
N  TABDF 0 00270000
    TEXT 00280000
***** 00290000
* LINE CONTROL BLOCK * 00300000
***** 00310000
.* 00320000
    AIF (&BSCA EQ '0').COMN 00330000
.* 00340000
    SPACE 2 00350000
*----- FIRST BSCA-ONLY SEGMENT -----* 00360000
    SPACE 1 00370000
LCBSCA EQU 76 LCB BEGIN FOR BSCA. 00380000
LCBPOL EQU LCBSCA+1 ADDRESS OF POLLING LIST. 00390000
LCBSEL EQU LCBPOL+2 ADDRESS OF SELECTION LIST. 00400000
LCBNO# EQU LCBSEL @ IN SW ID LIST OF NO VERIFY ID. 00410000
LCBID# EQU LCBSEL+1 ID LAST POLL/SELECTED TERM. 00420000
LCBPL@ EQU LCBID#+2 SAVE FOR CURRENT PARM LIST @. 00430000
LCBWRK EQU LCBPL@+2 WORK AREA FOR SUBRTNS. 00440000
    SPACE 00450000
* LCBAT1 AND LCBAT2 ARE BOTH ZEROED WHEN AN EOT IS HANDLED. 00460000
    SPACE 00470000
LCBAT1 EQU LCBWRK+1 ATTRIBUTE BYTE 1. 00480000
LCBCRI EQU X'80' CANCEL RCVI, STOP POLLING. 00490000
LCBPRI EQU X'40' CANCEL RCVI, HANDLE PRIORITY PUT 00500000
LCBINT EQU X'10' INTERRUPT WITH NO PARM LIST QUED 00510000
LCBNTQ EQU X'08' ACTIVE PARM LIST REMOVED FR QUE. 00520000
LCBDEQ EQU X'04' DEQUE WHEN EOT IS FOUND. 00530000
LCBEOT EQU X'02' SEARCH EOT. READ LINE FOR EOT. 00540000
    SPACE 00545000
LCBAT2 EQU LCBAT1+1 ATTRIBUTE BYTE 2. 00570000
LCBTRC EQU X'80' TRUNCATE BLOCK, BUMP BKX TO END. 00580000
LCBSET EQU X'40' SEND EOT ON THE LINE. 00590000
LCBACT EQU X'20' LINE ACTIVE WITH WORK. 00600000

```

LCBRFT EQU	X'10'	REQUEST FOR TEST STARTED.	00610000
LCBABT EQU	X'08'	ABORT THE LINE CONNECTION.	00620000
LCBSEC EQU	X'04'	SCOND BLOCK INDICATOR.	00625000
LCBPUT EQU	X'02'	PUT PENDING ON LINE.	00630000
LCBRCI EQU	X'01'	RECEIVE INIT ON LINE.	00640000
	SPACE		00650000
LCBADJ EQU	LCBAT2+2	ADJUSTED OUTPUT LENGTH.	00660000
LCBATL EQU	LCBADJ+2	@ OF ASCII TRANSLATE BUFFER.	00670000
LCBOWN EQU	LCBATL+2	TCB ADDR OF OWNING TASK.	00680000
LCB\$LO EQU	LCBOWN+2	C/S OF LINE INIT. TRANSIENT.	00690000
LCBSRT EQU	LCB\$LO+2	@ OF START OF LINE BUFFER AREA.	00700000
LCBBND EQU	LCBSRT+2	@ OF END OF LINE BUFFER AREA.	00710000
LCBKLC EQU	LCBBND+2	BLOCK LENGTH CURRENT INPUT OP.	00720000
.*			00730000
.COMN ANOP		. CONTINUE HERE	00740000
.*			00750000
	SPACE 2		00760000
-----	SEGMENT COMMON TO MLTA AND BSCA		00770000
	SPACE 1		00780000
LCBCCP EQU	X'65'	BEGINNING OF LCB	00790000
LCBPLQ EQU	LCBCCP+2-1	ADDR OF 1ST PARM LIST IN LINE QU	00800000
LCBCHN EQU	LCBPLQ+2	CHAIN OF ALL LCB'S IN CCP SYSTEM	00810000
LCBATA EQU	LCBCHN+1	LCB ATTRIBUTE BYTE A.	00820000
LCBIGN EQU	X'80'	1-IGNORE OP END. REOPENING LINE	00830000
	SPACE		00840000
LCBELC EQU	LCBATA+1	ID XLATE XIENT EBCDIC -> LINE.	00850000
LCBRS1 EQU	LCBELC+1	RESERVED BYTE 1. (NOT USED)	00860000
	SPACE		00870000
-----	MLTA / BSCA MULTIPLIED DEFINED BYTE		00880000
	SPACE		00890000
LCBLCE EQU	LCBRS1+1	ID XLATE XIENT LINE -> TO UPPER	00900000
*		* CASE EBCDIC *MLTA ONLY*	00910000
LCBOPE EQU	LCBLCE	OP END COUNT ON LINE *BSCA ONLY*	00920000
	SPACE		00930000
LCBLID EQU	LCBLCE+1	SAVED ID OF LAST TERM POLLED	00940000
LCBLLE EQU	LCBLID+1	ID XLATE XIENT LINE -> TO	00950000
*		* LOWER CASE EBCDIC.	00960000
	SPACE		00970000
-----	MLTA / BSCA MULTIPLIED DEFINED BYTE		00980000
	SPACE		00990000
LCBOLT EQU	LCBLLE+1	POLT COUNT (RUNNING AND PENDING)	01000000
*		ON THIS TP LINE *MLTA ONLY*	01010000
	AIF (&BSCA EQ '0').ATR		01020000
LCBAT3 EQU	LCBOLT	LCB ATTRIBUTE BYTE 3 *BSCA ONLY*	01030000
LCBITB EQU	BIT0	1-ITB SUPPORTED FOR THIS DTF	01040000
LCBTSP EQU	BIT1	1-TRANSPARENCY SUPPORTED	01050000
LCBENB EQU	BIT2	1-BSCA LINE ENABLED	01060000
LCBATO EQU	BIT3	AUTO CALL HARDWARE SUPPORTED.	01070000
LCBBYP EQU	BIT4	POLLING A BUSY PRINTER	01072000
LCBSTS EQU	BIT5	POLL FOR STATUS IN OPERATION	01074000
LCBSTF EQU	BIT6	TERMINATE WITH U- NS.	01076000
.ATR	SPACE 1		01080000
LCBATR EQU	LCBOLT+1	LCB ATTRIBUTE BYTE	01090000
	SPACE 1		01100000
LCBNIT EQU	BIT0	* HAD SUCCESSFUL INITIAL	01110000
*		* OPERATION ON THIS LINE	01120000
LCBOLR EQU	BIT1	* 1-POLT CURRENTLY RUNNING	01130000
LCBGMN EQU	BIT2	* 1-GETMAIN NEEDED FOR LINE QUEU	01140000
LCBSTP EQU	BIT3	1-ABORT ISSUED TO STOP READ	01150000
LCBSWL EQU	BIT4	1-SWALLOW INPUT FROM NEXT READ	01160000

*			OP END	01170000
LCB1PL	EQU	BIT5	1-INDICATES NEXT PARM LIST	01180000
*			ISSUED TO THIS LINE IS TO BE PUT	01190000
*			AT THE TOP OF LINE QUEUE	01200000
LCBTBK	EQU	BIT6	1-BIT BUCKET DATA FROM READ OP	01210000
*			SET FOR 2741 TO DO READ INITIAL	01220000
*			WHEN WRITE IS 1ST OP TO TERMINAL	01230000
LCBTIM	EQU	BIT7	RESCHEDULE PENDING ON LINE	01235000
		SPACE 1		01240000
LCBBFL	EQU	LCBATR+2	LENGTH OF DATA AREA IN LINE BUFR	01250000
LCBRS3	EQU	LCBBFL+2	RESERVED BYTES.(NOT USED)	01260000
LCBIBL	EQU	LCBRS3+2	LENGTH OF CURRENT INVITE INPUT	01270000
*			BUFFER FOR THIS LINE	01280000
LCBIBA	EQU	LCBIBL+2	ADDRESS OF INVITE INPUT BUFFER	01290000
LCBTCB	EQU	LCBIBA+2	ADDR OF TCB WHICHS OWNS SWITCHED	01300000
*			TP LINE	01310000
LCBATC	EQU	LCBTCB+1	COUNT OF ALLOCATED TUBS ON	01320000
*			SWITCHED LINE	01330000
LCBNW#	EQU	LCBATC+1	# NEW REQUESTS FOR CM - FROM II	01340000
.	*			01350000
	AIF	(&MLTA EQ '0').BSCA		01360000
.	*			01370000
	SPACE 2			01380000
*	-----	MLTA-ONLY SEGMENT	-----*	01390000
	SPACE 1			01400000
LCBWTC	EQU	LCBNW#+1	WORLD TRADE TERMAINL TYPE BYTE	01410000
MLNWTC	EQU	X'00'	NOT A WORLD TRADE TERMINAL	01420000
ML2970	EQU	X'01'	2970 WORLD TRADE TERMINAL	01430000
ML5930	EQU	X'02'	5930 WORLD TRADE TERMINAL	01440000
	SPACE 1			01450000
LCBMLN	EQU	LCBWTC+1	LENGTH OF MLTA LCB	01460000
	SPACE			01470000
*			SPECIAL MLTA DTF EQUATES IN \$MDTTP	01480000
ML2741	EQU	X'03'	2741	01490000
	SPACE			01500000
*			SPECIAL MLTA EQUATES IN IOCS FLAG NUMBER 2	01510000
TDFFL2	EQU	57	IOCS FLAG BYTE 2	01520000
FL2OLP	EQU	X'20'	PROGRAM REQUESTED ONLINE TEST	01530000
FL2OLT	EQU	X'08'	ONLINE TEST IN PROGRESS	01540000
	SPACE			01550000
*			MLTA DTF EQUATE FOR TDFEPC - INTERNAL EVENT COMPLETION CODE	01560000
TDFEPC	EQU	69	INTERNAL EVENT COMPLETION CODE.	01570000
	SPACE			01580000
*			SPECIAL MLTA DTF EQUATES IN \$MDOLT	01590000
TDFERP	EQU	80	ERROR RECOVERY STAUTS SWITCHES	01600000
ERPCAL	EQU	X'80'	ERP HAS BEEN CALLED	01610000
	SPACE			01620000
\$MDPE8	EQU	83	HDB BYTE 7	01630000
	SPACE			01640000
*			SPECIAL MLTA DTF EQUATES IN \$MDPE7	01650000
MLDISC	EQU	X'80'	1-MLTA SW LINE IS DISCONNECTED	01660000
	SPACE			01670000
*			2740M2 BUFFERED-RECEIVE STATUS BYTES IN \$MDRSP-1	01680000
MLBID	EQU	X'04'	2740M2 IN BID STATUS	01690000
MLENTR	EQU	X'02'	2740M2 IN ENTER STATUS	01700000
MLWBYP	EQU	X'08'	2740M2 BUSY WITH WRITE (PTTC)	01710000
MLWBYC	EQU	X'10'	2740M2 BUSY WITH WRITE (CORR)	01720000
	SPACE			01730000
*			SPECIAL MLTA DTF EQUATES IN \$MDTCT	01740000
MLPTTC	EQU	X'01'	PTTC TRANSMISSION CODE	01750000

```

MLCORR EQU X'02'          CORRESPONDENCE TRANSMISSION CODE 01760000
SPACE                                01770000
.*                                  01780000
.BSCA AIF (&BSCA EQ '0').END      01790000
.*                                  01800000
SPACE 2                             01810000
*----- SECOND BSCA-ONLY SEGMENT -----* 01820000
SPACE 1                              01830000
LCBOPC EQU LCBNW#+1          LAST OP CODE ON BSCA.      01840000
LCBMVD EQU X'80'            DATA MOVED INDICATOR ON.      01850000
LCBERP EQU X'40'            LINE IN ERP MODE.(IGNORE DATA). 01860000
LCBRVI EQU X'20'            RVI IND - SEND/RECEIVE RVI.      01870000
* X'08'                      * SAVE                          01880000
* X'04'                      * AREA                          01890000
* X'02'                      * FOR CCP                        01900000
* X'01'                      * OP CODE.                      01910000
SPACE                                01920000
* THE FOLLOW EQUATES ARE FOR OUTPUT (SELECTION) ONLY.      01930000
SPACE                                01940000
LCBADL EQU LCBOPC+1          START OF LINE SELECTION LIST. 01950000
LCBADN EQU LCBADL+10        END OF SELECTION LIST.      01960000
SPACE                                01970000
* THE FOLLOWING EQUATES ARE FOR INPUT (POLLING) ONLY.      01980000
SPACE                                01990000
LCBMRL EQU LCBOPC+2          INCREMENT AREA FOR MSG LENGTH. 02000000
LCBMIL EQU LCBMRL+2          ORIGINAL MESSAGE INPUT LENGTH 02010000
* USED VIA LCBMR@ LABEL.      02020000
LCBMR@ EQU LCBMIL+2          ORIGINAL MESSAGE RECORD ADDR. 02030000
SPACE 1                             02040000
LCBBLN EQU LCBADN+1          LENGTH OF BSCA LCB                02050000
SPACE 1                             02060000
* THE FOLLOWING EQUATES ARE FOR BSCA SWITCHED AUTO CALL DTF ONLY. 02070000
SPACE                                02080000
LCBTEL EQU LCBOPC+1          BEGINING OF AURO DIAL #.      02090000
LCBTL@ EQU LCBTEL+14         END OF AUTO DIAL AREA.      02100000
LCBSWN EQU LCBTL@+1          LENGTH OF BSCA AUTO DIAL DTF. 02110000
SPACE                                02120000
.*                                  02130000
.END ANOP                      . END OF LCB                    02140000
.*                                  02150000
MEND                             02160000

```

```

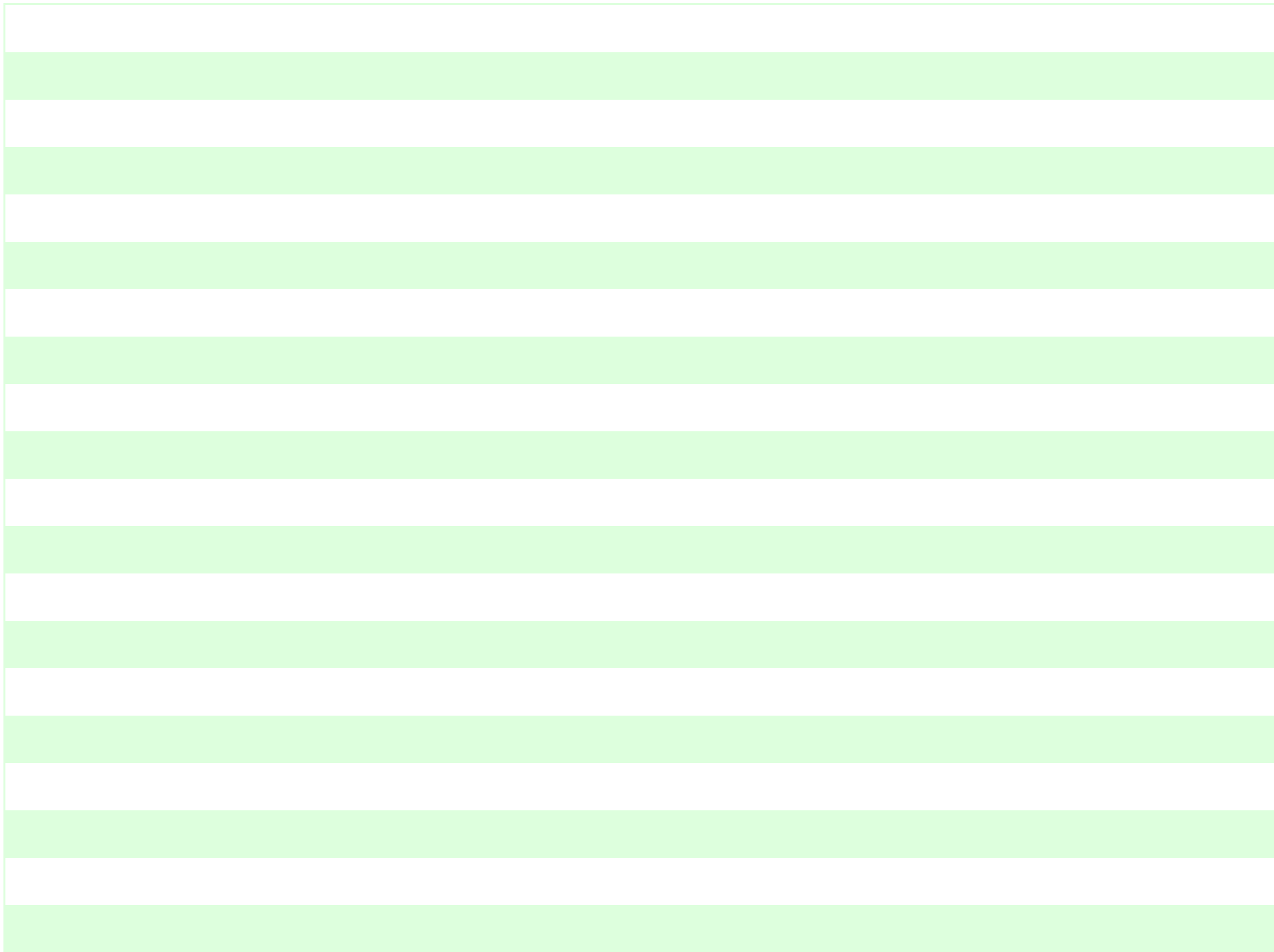
MACRO                                00010000
*****                                00020000
.* STATUS:  RELEASE 2                  * 00030000
.*                                      * 00040000
.* NAME:  $ETNT                        * 00050000
.*                                      * 00060000
.* FUNCTION:                           * 00070000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- DEFINE * 00080000
.*   SYMBOLS FOR TERMINAL NAME TABLE -- FOR:      * 00090000
.*     . OFFSETS OF FIELDS                   * 00100000
.*     . VALUES WITHIN FIELDS              * 00110000
.*                                           * 00120000
.* INPUT OPERANDS:  NONE                 * 00130000
*****                                00140000
$ETNT                                00150000
.*                                     00160000
TEXT                                  00170000
.*                                     00180000
*****                                00190000
*          T E R M I N A L   N A M E   T A B L E      * 00200000
*****                                00210000
TNTCCP EQU    0          BEGINNING OF TNT          00220000
TNTNAM EQU    TNTCCP-1+6  SYMBOLIC TERMINAL NAME FIELD 00230000
TNTTUB EQU    TNTNAM+2    ADDRESS OF ASSOCIATED TUB    00240000
*                                     * IF LEFT BYTE ZERO, UNASSIGNED 00250000
TNTTNT EQU    TNTTUB      @ MASTER TNT - FOR SUB-TERMINAL 00260000
SPACE 1                                           00270000
*----- REDEFINITION OF TNTTUB FOR DISK VERSION OF TNT -----* 00280000
TNTIID EQU    TNTTUB-2+1  TUTIID IF ASSIGNED, 0=UNASSIGNED 00290000
TNTMNO EQU    TNTIID+1    0=UNIQUE TERMINAL NAME          00300000
*                                     N=REL TNT NO. OF MASTER NAME 00310000
*-----*                                           00320000
SPACE 1                                           00330000
TNTFLG EQU    TNTTUB+1    TNT FLAG BYTE                00340000
TNTPRI EQU    BIT0        1-PRIMARY NAME FOR THIS TERMINAL 00350000
TNTBLK EQU    BIT1        1-ENTRY IS 'BLANK' TERMINAL NAME 00360000
TNTKNS EQU    BIT2        1-ENTRY IS 'CONSOL' TERM'L NAME 00370000
TNTR5 EQU    BIT3         1--TNT USED BY CURRENT PGM REQ   00380000
TNTAM EQU    BIT4         1--STOP POLLING SUCCESSFUL.      00390000
SPACE 1                                           00400000
TNTMCT EQU    TNTFLG+1    INDEXES FOR MULTI-CMPNT TERMINAL 00400000
*                                     BITS 0-3: INDEX OF INPUT CMPNT 00410000
*                                     BITS 4-7: INDEX OF OUTPUT CMPNT 00420000
TNTSTT EQU    TNTMCT+1    INDEX TO SWITCHED TERMINAL TABLE 00430000
TNTLN EQU    TNTSTT+1-TNTCCP  LENGTH OF TNT ENTRY        00440000
TNTSTP EQU    X'00'       TNT STOPPER BYTE IN CORE        00450000
MEND                                           00460000

```

```

MACRO                                00010000
*****                               00020000
.*                                     * 00030000
.* NAME: $ETML                        * 00040000
.*                                     * 00050000
.* MODIFICATION LEVEL: MOD 15 RELEASE 2. * 00060000
.*                                     * 00070000
.* FUNCTION:                           * 00080000
.*                                     * 00090000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- DEFINE * 00100000
.* EQUATES FOR THE OFFSETS OF FIELDS IN: * 00110000
.*                                     * 00120000
.* . TRANSLATE PARAMETER LIST          * 00130000
.* . MOVE PARAMETER LIST              * 00140000
.*                                     * 00150000
.* INPUT OPERANDS:                     * 00160000
.*                                     * 00170000
.* . NONE                              * 00180000
.*                                     * 00190000
*****                               00200000
$ETML                                00210000
.*                                     00220000
TEXT                                  00230000
.*                                     00240000
*****                               00250000
* TRANSLATE / MOVE LISTS              * 00260000
*****                               00270000
SPACE 2                               00280000
* TRANSLATE PARAMETER LIST            00290000
SPACE 1                               00300000
TLCCP EQU 0 BEGINNING OF TRANSLATE LIST 00310000
TLTOL EQU TLTOL-1+2 LENGTH OF TO FIELD 00320000
TLTOA EQU TLTOL+2 ADDRESS OF THE TO FIELD 00330000
TLFRMA EQU TLTOA+2 ADDRESS OF THE FROM FIELD 00340000
TLFRML EQU TLFRMA+2 LENGTH OF THE FROM FIELD 00350000
TLRTC EQU TLFRML+1 TRANSLATE RETURN CODE 00360000
SPACE 2                               00370000
* MOVE PARAMETER LIST                 00380000
SPACE 1                               00390000
MVLCCP EQU 0 BEGINNING OF MOVE LIST 00400000
MVLTOA EQU MVLCCP+2-1 LENGTH OF THE TO FIELD 00410000
MVLTOA EQU MVLTOA+2 ADDRESS OF THE TO FIELD 00420000
MVLFRMA EQU MVLTOA+2 ADDR OF THE FROM FIELD 00430000
MVLFRML EQU MVLFRMA+2 LENGTH OF THE FROM FIELD 00440000
MVLTYP EQU MVLFRML+1 TYPE OF MOVE REQUEST 00450000
* * X'00' - NO ATR SWAP NECESSARY 00460000
* (NEITHER FIELD IN UPA) 00470000
* * X'01' - FROM FIELD IN UPA 00480000
* * X'02' - TO FIELD IN UPA 00490000
* * X'03' - BOTH FIELDS IN UPA 00500000
* (SWAP ATRS). 00510000
MVLTCB EQU MVLTYP+2 ADDRESS OF TCB FOR USER TASK 00520000
* * IF EITHER TO OR FROM FIELD 00530000
* IS IN A USER PROGRAM. 00540000
* * FIELD NOT NEEDED IF MVLTYP 00550000
* IS X'00'. 00560000
SPACE 00570000
MEND 00580000

```



```

MACRO                                00010000
*****                                00020000
.* STATUS:  RELEASE 4                 * 00030000
.*                                     * 00040000
.* NAME:  $EBEQ                       * 00050000
.*                                     * 00060000
.* FUNCTION:                          * 00070000
.*                                     * 00080000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- DEFINE * 00090000
.*   SYMBOLS FOR BSCA -- FOR:         * 00100000
.*                                     * 00110000
.*     . IOB OFFSETS OFFSETS          * 00120000
.*     . BSCA WORK AREA OFFSETS       * 00130000
.*     . POLL/SELECTION LIST OFFSETS  * 00140000
.*     . CHECK LIST OFFSETS           * 00150000
.*     . MINIMUN BSCA SYSTEM TRANSIENT ID'S * 00160000
.*     . 3270/3735 COMMANDS, ORDERS,BUFFER OFFSETS * 00170000
.*                                     * 00180000
.* INPUT OPERANDS:                   * 00190000
.*                                     * 00200000
.*     . IOB-Y/N: IOB EQUATES (DEFAULT N) * 00210000
.*     . WKA-Y/N: BSCA WORK AREA EQUATES (DEFAULT N) * 00220000
.*     . POL-Y/N: POLL LIST EQUATES (DEFAULT N) * 00230000
.*     . CKL-Y/N: CHECK LIST EQUATES (DEFAULT N) * 00240000
.*     . MIN-Y/N: MINIMUN SYSTEM TRANSIENT ID'S. * 00250000
.*     . CMD-CM/Y/N: CM-SAME AS IF CMD-Y,ORDS-Y,BUFR-ALL,OUT32-EQU GIVEN * 00260000
.*       Y/N - 3270 COMMAND EQUATES (DEFAULT N) * 00270000
.*     . ORDS-Y/N: 3270 ORDER EQUATES (DEFAULT N) * 00280000
.*     . BUFR-OFFSET/AID/ALL/N: OFFSET - 3270/3735 LINE BUFFER OFFSETS * 00290000
.*       AID - 3270 AID EQUATES * 00300000
.*       ALL - BOTH OFFSETS AND AID EQUATES * 00310000
.*       N - NEITHER OF ABOVE (DEFAULT N) * 00320000
.*     . OUT32-ALL/EQU/N: ALL - IGNORES ALL OTHER KEYWORDS. GENERATE * 00330000
.*       CCP 3270 OUTPUT BUFFER FORMAT. * 00340000
.*       EQU - LENGTH OF CCP 3270 OUTPUT BUFFER FORMAT. * 00350000
.*       N - NEITHER OF THE ABOVE (DEFAULT N) * 00360000
.*                                     * 00370000
*****                                00380000
$EBEQ &IOB-N,&WKA-N,&POL-N,&CKL-N,&MIN-N,&CMD-N,&ORDS-N, X00390000
      &BUFR-N,&OUT32-N,&BITS-N,&#CMEQ-N,&ARRAD-XX,&BCEQ-N 00400000
TEXT                                00410000
.* OUT32-ALL WILL CAUSE ALL OTHER KEYWORDS TO BE IGNORED. IT WILL 00420000
.* GENERATE ONLY 3270 OUTPUT CONSTANTS AND EQUATES.                00430000
.* OUT32-EQU AND ANY OF THE OTHER KEYWORDS CAN BE USED TOGETHER. 00440000
AIF (&OUT32 EQ 'ALL').OUT32        00450000
AIF (&IOB NE 'Y').WKA              00460000
*-----*                            00470000
* BSCA IOB EQUATES(FROM BSEQU MACROS OF MLMP). * 00480000
*-----*                            00490000
SPACE                                00500000
IOBNXT EQU 1 @ NEXT IOB.          00510000
IOBQ EQU IOBNXT+1 DEVICE ADDRESS. 00520000
IOBDBL EQU IOBQ+2 BUFFER LENGTH.  00530000
IOBFLA EQU IOBDBL+1 IOB FLAG A.    00540000
TDLAY EQU X'20' TWO SEC. TIME OUT STARTED. 00550000
FIRST EQU X'04' FIRST TIME. LINE INIT NOT DONE. 00560000
DELAY EQU X'02' DELAY IOB-WAIT SEQUENCE GOING. 00570000
IOBFLG EQU IOBFLA+1 IOB FLAG B.   00580000
IOBCMP EQU IOBFLG+1 IOB COMPLETION CODE. 00590000

```


PROCES EQU	X'80'	IOB CMP, IOB IN PROCESS.	B	00600000
IOBDAT EQU	IOBCMP+2	@ DATA BUFFER.		00610000
IOBSNS EQU	IOBDAT+2	SENSE AREA.		00620000
IOBERR EQU	IOBSNS+1	ERROR COUNT.		00630000
IOBCAR EQU	IOBERR+2	CURRENT ADDRESS.		00640000
IOBTAR EQU	IOBCAR+2	TRANSITION ADDRESS.		00650000
IOBSAR EQU	IOBTAR+2	STOP ADDRESS.		00660000
IOBDTF EQU	IOBSAR+2	@ DTF.		00670000
IOBMSG EQU	IOBDTF+2	DELAY MESSAGE OFFSET.		00680000
IOBNEX EQU	IOBDTF+2	AREA FOR CALC @ NEXT IOB.		00690000
IOBDNB EQU	IOBNEX+2	AREA FOR CALL @ 2ND NEXT BUFFER.		00700000
IOB2NX EQU	IOBDNB+2	AREA FOR CALC @ 2ND NEXT IOB.		00710000
IOBL EQU	IOBDTF+1	LENGTH OF THE OPERATIONAL IOB.		00720000
.WKA AIF	(&WKA NE 'Y').POL			00730000
-----				00740000
* BSCA WORK AREA EQUATES(FROM BSEQU MACRO OF MLMP).			*	00750000
-----				00760000
SPACE				00770000
PL1EOT EQU	X'40'	RECORD TRUNCATE IND. (IN \$BPOLD)		00780000
WKDTFD EQU	20	DTF ADDRESS		00790000
BSRJ2D EQU	22	WORK AREA INDICATORS		00800000
JF2SAR EQU	X'20'	STOP AUTO RESPONSE. (IN BSRJ2D)		00810000
JF2SR2 EQU	X'10'	STOP AUTO RESPONSE. (IN BSFL4D)		00815000
WKIOBD EQU	28	ADDRESS OF LAST IOB.		00820000
WKERRD EQU	X'1A'	ERROR RETRY COUNT		00823000
DCOUNT EQU	X'7B'	DELAY COUNT		00826000
F3AUTO EQU	X'01'	AUTO RESPONSE ACTIVE.(IN \$BWFG3)		00830000
F3MOVE EQU	X'04'	RECORD MOVE INDICATOR(IN \$BWFG3)		00840000
BSFLGD EQU	35	WORK AREA FLAGS.		00850000
BSFL4D EQU	X'82'	WORK AREA FLAGS.		00855000
ACTIVE EQU	X'10'	BSCA ENABLED. (IN BSFLGD)		00860000
FWDABT EQU	X'04'	FORWARD ABORT IND. (IN BSFLGD)		00870000
ACKSD EQU	36	ACK NUMBER AND OTHER STATUS.		00880000
AKERR EQU	X'04'	ERROR, LINE DISABLED.(IN ACKSD)		00890000
WKDELL EQU	132	WORK AREA ERROR LOG LIST.		00900000
.POL AIF	(&POL NE 'Y').CKL			00910000
-----				00920000
* BSCA POLL / SELECT LIST EQUATES.			*	00930000
-----				00940000
SPACE				00950000
POLID EQU	0	ENTRY ID IN LIST.		00960000
POLCNT EQU	1	NUMBER OF TERM CHARS IN ENTRY.		00970000
POLCH1 EQU	2	1ST TERM CHAR.		00980000
POLCH2 EQU	3	2ND TERM CHAR.		00990000
POLNXT EQU	3	# CONSTANT BYTES IN EVERY ENTRY.		01000000
WRAP EQU	X'FF'	END OF WRAP LIST INDICATOR.		01010000
ONETIM EQU	X'FE'	END OF OPEN LIST INDICATOR.		01020000
POLEND EQU	X'F0'	CODES RESV. FOR SYST. F0-FF.		01030000
POLACT EQU	X'F1'	USE ONLY ACTIVE ENTRIES.		01040000
.CKL AIF	(&CKL NE 'Y').MIN			01050000
-----				01060000
* BSCA CHECK LIST EQUATES.			*	01070000
-----				01080000
SPACE				01090000
CKLSTS EQU	0	CHECK LIST STATUS BYTE.	B	01100000
CKLDTF EQU	2	CHECK DTF @.	B	01110000
CKLEN EQU	3	CHECK LIST ENTRY LENGTH.	B	01120000
CKLSKP EQU	X'80'	CHECK LIST: SKIP ENTRY BIT.	B	01130000
CKLAST EQU	X'20'	CHECK LIST: LAST ENTRY INDICATOR		01140000
.MIN AIF	(&MIN NE 'Y').NOMIN			01150000

```

*-----* 01160000
*          ID'S FOR TRANSIENT FORM OF RESIDENT FUNCTION.          * 01170000
*-----* 01180000
          SPACE 01190000
CTTASV EQU 1      TRANSIENT FORM OF 'CMTASV'. 01200000
CTREJC EQU 3      TRANSIENT FORM OF 'CMREJC'. 01210000
CTSTOR EQU 4      TRANSIENT FORM OF 'CMSTOR'. 01220000
CTBTAS EQU 5      TRANSIENT FORM OF 'CMBTAS'. 01230000
CTFORB EQU 6      TRANSIENT FORM OF 'CMFORB'. 01240000
CTRLB EQU 7      TRANSIENT FORM OF 'CMFIGL'. 01250000
.NOMIN ANOP 01260000
.TEQU AIF (&CMD EQ 'Y').TQ 01270000
      AIF (&CMD EQ 'CM').TQ 01280000
      AIF (&ORDS EQ 'Y').TQ 01290000
      AIF (&BUFR NE 'N').TQ 01300000
      AIF (&OUT32 EQ 'N').END32 01310000
.TQ ANOP 01320000
***** 01330000
*          TERMINAL DEPENDENT EQUATES.          * 01340000
***** 01350000
      AIF (&CMD EQ 'CM').CM1 01360000
      AIF (&CMD NE 'Y').ORD 01370000
.CM1 ANOP 01380000
      SPACE 01390000
*-----* 01400000
*          3270 COMMANDS.          * 01410000
*-----* 01420000
          SPACE 01430000
ESC EQU X'27'      ESCAPE IDENTIFICATION. 01440000
COPY EQU X'F7'      COPY COMMAND. 01450000
WRITE EQU X'F1'      WRITE COMMAND. 01460000
ERSUPT EQU X'6F'      ERASE UNPROTECTED COMMAND. 01470000
ERSWRT EQU X'F5'      ERASE/WRITE COMMAND. 01480000
.ORD AIF (&CMD EQ 'CM').CM2 01490000
      AIF (&ORDS NE 'Y').BUFR 01500000
.CM2 ANOP 01510000
      SPACE 01520000
*-----* 01530000
*          3270 ORDERS.          * 01540000
*-----* 01550000
          SPACE 01560000
IC EQU X'13'      INSERT CURSOR. 01570000
DUP EQU X'1C'      DUP CHAR. (KEYBOARD ONLY). 01580000
EUA EQU X'12'      ERASE UNPROTESTED TO ADDRESS. 01590000
FM EQU X'1E'      FIELD MARK (KEYBOARD ONLY). 01600000
PT EQU X'05'      PROGRAM TAB. 01610000
RA EQU X'3C'      REPEAT TO ADDRESS. 01620000
SBA EQU X'11'      SET BUFFER ADDRESS. 01630000
SF EQU X'1D'      START OF FIELD. 01640000
.BUFR ANOP 01650000
      AIF (&CMD EQ 'CM').CM3 01660000
      AIF (&BUFR EQ 'OFFSET').OFFSET 01670000
      AIF (&BUFR EQ 'ALL').OFFSET 01680000
      AIF (&BUFR EQ 'AID').AID 01690000
      AGO .OUTEQ 01700000
.OFFSET ANOP 01710000
.CM3 ANOP 01720000
      SPACE 01730000
*-----* 01740000
*          3270 BUFFER OFFSETS.          * 01750000

```

```

*-----* 01760000
SPACE 01770000
CU EQU 0 CONTROL UNIT IDENTIFICATION. 01780000
DA EQU 1 DEVICE ADDRESS. 01790000
SPACE 01800000
AID EQU 0 AID CHARACTER POSITION. 01810000
SWAID EQU 0 AID CHAR POSITION FOR SWITCHED. 01815000
AIF (&BUFR EQ 'OFFSET').CNTU 01820000
.AID ANOP 01830000
SPACE 01840000
* EBCDIC AID VALUES. 01850000
SPACE 01860000
AIDCLR EQU X'6D' AID FOR CLEAR KEY. 01870000
AIDENT EQU X'7D' AID FOR ENTER KEY. 01880000
AIDCRD EQU X'E6' AID FOR CARD READER DATA. 01883000
AIDTST EQU X'F0' AID FOR RFT KEY. 01886000
SPACE 01890000
* ASCII AID VALUES. 01900000
SPACE 01910000
ASCCLR EQU X'5F' AID FOR CLEAR KEY. 01920000
ASCENT EQU X'27' AID FOR ENTER KEY. 01930000
AIF (&BUFR EQ 'AID').OUTEQ 01940000
SPACE 01950000
.CNTU ANOP 01960000
CURSOR EQU 4 CURSOR POSITION. 01970000
TEXT EQU 5 FIRST TEXT POSITION. 01980000
SWTEXT EQU 3 FIRST TEXT POS. SWITCHED LINE. 01985000
SPACE 2 01990000
* OR 02000000
SPACE 02010000
SNSTAS EQU 1 SENSE/STATUS MESSAGE ID. 02020000
SSID EQU C'%R' SENSE/STATUS CHARACTERS. 02030000
SSTX EQU 2 STX IN STATUS MESSAGE. 02040000
SSCU EQU 3 CONTROL UNIT ID. 02050000
SSDA EQU 4 DEVICE ADDRESS. 02060000
SSBYT1 EQU 5 BYTE 1 SENSE BYTE. 02070000
SSBYT2 EQU 6 BYTE 2 STATUS BYTE. 02080000
SSETX EQU 7 ETX IN STATUS MESSAGE. 02090000
SPACE 1 02100000
*-----* 02110000
* 3735 BUFFER OFFSTES. * 02120000
*-----* 02130000
S375ID EQU 1 MESSAGE IDENTIFIER. 02140000
S375NL EQU 2 -NUL- FOLLOWING IDENTIFIER. 02150000
S375B1 EQU 3 STATUS BYTE ONE. 02160000
S375B2 EQU 4 STATUS BYTE TWO. 02170000
***** 02171000
* 02172000
* 3741 BUFFER OFFSETS * 02173000
* 02174000
***** 02175000
S374ST EQU 1 STATUS MESSAGE IDENTIFIER 02176000
AGO .OUTEQ 02180000
.OUT32 ANOP 02190000
.* NOTE: ANY CHANGE TO THE NUMBER OF FORMAT CONTROL CHARACTERS MUST 02200000
.* BE REFLECTED IN BOTH 'FORMTL' STATEMENTS BELOW. 02210000
SPACE 2 02220000
FORMAT EQU * HOLD AREA TO BUILD 3270 OUTPUT. 02230000
DC AL1(ESC) 3270 ESCAPE COMMAND. 02240000
COMAND DC AL1(WRITE) OP COMMAND BYTE. 02250000

```

```

DC      XL1'C2'          RESTORE KEYBOARD.          02260000
SPACE                                     02270000
* FOLLOWING CLEARS TOP OF SCREEN FOR 'A, E, AND R' CLASS MESSAGES. 02272000
* SETS INPUT FIELD ATTR, AND SETS CURSOR.          02274000
SPACE                                     02276000
ORD1 EQU *              ORDER FIELD ONE.          02278000
CURSR@ DC AL1(SBA)      SET FIRST FIELD          02280000
DC      XL2'4040'      *              TO R-1, C-1 (M1/M2). 02290000
DC      AL1(SF)        START FIELD / INPUT ATTRIBUTES. 02300000
DC      XL1'40'        UNPROT-ALPHAMERIC-NORMAL-MDT OFF 02310000
DC      AL1(IC)        INSERT CURSOR R-1,C-2 (M1/M2). 02310900
SPACE                                     02311800
* FOLLOWING CLEARS LAST OPERATOR INPUT (FOR CLASS A MESSAGES ONLY). 02312700
SPACE                                     02313600
ORD2 EQU *              ORDER FIELD TWO.          02314500
OPRDAT DC AL1(RA)      CLEAR INPUT TO R-03, C-02 (M1) 02315400
#OPER2 DC XL2'C1D1'    *              TO R-02, C-02 (M2). 02316300
DC      XL1'00'        * WITH NULLS.             02317200
SPACE                                     02318100
* FOLLOWING IS FRAMING FOR OUTPUT TEXT.           02319000
SPACE                                     02320000
ORD3 EQU *              ORDER FIELD THREE.        02325000
OUTFRM DC AL1(SBA)     SET BUF @ OUTPUT MESSAGE.    02330000
#OUT@ DC XL2'C1D1'    OUTPUT @: R-03, C-02 (MOD 1). 02340000
*              *              R-02, C-02 (MOD 2).    02345000
DC      AL1(SF)        START FIELD / REPLY ATTRIBUTES. 02350000
DC      XL1'60'        PROT-ALPHAMERIC-NORMAL-MDT OFF. 02360000
SPACE                                     02362000
* FOLLOWING IS THE OUTPUT TEXT AREA. (MESSAGE A11 IS DEFAULT). 02364000
SPACE                                     02366000
DATA EQU *              02370000
MSG# DC CL3'A11'      MESSAGE                     02380000
DC      XL1'40'      *              A11(BLANK)        02390000
DC      CL5'CLEAR'   *              CLEAR.           02400000
SPACE                                     02410000
* FOLLOWING NULLS OUT THE REST OF THE MESSAGE AREA. 02412000
SPACE                                     02414000
ORD4 EQU *              ORDER FIELD FOUR.          02416000
CLREST DC AL1(RA)     CLEAR REST                   02430000
#END@ DC XL2'C260'    *              TO R-5,C-1 (MOD 1). 02440000
*              *              TO R-3,C-1 (MOD 2).    02445000
DC      XL1'00'        * WITH NULLS.             02460000
ENDSCR EQU *          END OF OUTPUT TEXT STREAM.    02470000
.* NOTE: ANY CHANGE TO THE NUMBER OF FORMAT CONTROL CHARACTERS MUST 02560000
.* BE REFLECTED IN BOTH 'FORMTL' STATEMENTS BELOW. 02570000
FORMTL EQU DATA-FORMAT+ENDSCR-CLREST-1 # FORMAT CONTROL BYTES-1. 02580000
AGO .EXIT 02590000
.OUTEQ AIF (&CMD EQ 'CM').CM4 02600000
AIF (&OUT32 NE 'EQU').END32 02610000
.CM4 ANOP 02620000
SPACE 3 02630000
*-----* 02640000
* EQUATE FOR NUMBER OF CCP FORMAT CONTROL CHAR TO 3270. * 02650000
*-----* 02660000
FORMTL EQU 22 # OF FORMAT CONTROL CHARS. 02670000
.END32 ANOP 02680000
.BITS AIF (&BITS NE 'Y').NOBIT 02690000
*-----* 02700000
* HEX DEFINITION FOR BITS. * 02710000
*-----* 02720000

```

	SPACE			02730000
BIT0	EQU	X'80'	* HEX DEFINITIONS FOR A BIT.	02740000
BIT1	EQU	X'40'	*	02750000
BIT2	EQU	X'20'	*	02760000
BIT3	EQU	X'10'	*	02770000
BIT4	EQU	X'08'	*	02780000
BIT5	EQU	X'04'	*	02790000
BIT6	EQU	X'02'	*	02800000
BIT7	EQU	X'01'	*	02810000
.NOBIT	AIF	(&#CMEQ NE 'Y').NO#EQ		02820000
	SPACE			02830000
*	BSCA	EQUATES FOR \$CMWK.		02840000
	SPACE			02850000
#CMBL0	EQU	0	BSCA LOG MSG TYPE 0.	02860000
#CMBL1	EQU	1	BSCA LOG MSG TYPE 1.	02870000
#CMBL2	EQU	2	BSCA LOG MSG TYPE 2.	02880000
.NO#EQ	AIF	(&ARRAD EQ 'XX').ENDAR		02890000
	SPACE			02900000
*	ARR	INCREMENT VALUES FOR TRANSIENT RETURN TO MAINLINE ROUTINE.		02910000
	SPACE			02920000
.ARR2	AIF	(&ARRAD NE 'BE').ARR3		02930000
BEXIT1	EQU	0	POST COMPLETE, RETURN TO USER.	02940000
BEXIT2	EQU	4	RESCHEDULE WORK ON THE LINE.	02950000
BEXIT3	EQU	8	DATA AVAILABLE, HANDLE FOR USER.	02960000
	AGO	.ENDAR		02970000
.ARR3	AIF	(&ARRAD NE 'BQ').ARR4		02980000
BQXIT1	EQU	0	RESCHEDULE WORK ON THE LINE.	02990000
BQXIT2	EQU	4	STOP FAILED, COMPLETE READ OPEND	03000000
BQXIT3	EQU	7	ABORT OF THE BSCA LINE REQUIRED.	03010000
	AGO	.ENDAR		03020000
.ARR4	AIF	(&ARRAD NE 'BP').ARR5		03030000
BPXIT1	EQU	0	RESET TP SCHEDULED BITS.	03040000
BPXIT2	EQU	4	HANDLE ABORT OF THE BSCA LINE.	03050000
	AGO	.ENDAR		03060000
.ARR5	AIF	(&ARRAD NE 'BB').ARR6		03070000
BBXIT1	EQU	0	ERROR RETURN TO CM.	03080000
BBXIT2	EQU	7	NORMAL RETURN TO CM.	03090000
	AGO	.ENDAR		03100000
.ARR6	ANOP			03110000
.ENDAR	ANOP			03120000
.NOXEQ	ANOP			03130000
	AIF	(&BCEQ EQ 'CK').BFX01		03140000
	AIF	(&BCEQ NE 'Y').NOBFQ		03150000
.BFX01	ANOP			03160000
BCXRTN	EQU	0	DON'T X-CONTROL, RETURN.	03170000
BCX2S2	EQU	10	X-CONTROL TO \$CC4S2.	03180000
.NOBFQ	AIF	(&BCEQ NE 'CK').BFX02		03190000
BCXXS2	EQU	BCS2-BC	THE X-CONTROL VALUE IN BC.	03200000
.BFX02	ANOP			03210000
.EXIT	ANOP			03220000
	MEND			03230000

```

MACRO                                00010000
*****                                00020000
.* STATUS:  RELEASE 6                AJS 00030000
.*                                     * 00040000
.* NAME:  $ETCB                       * 00050000
.*                                     * 00060000
.* FUNCTION:  DEFINE THE LABELS FOR THE CCP EXTENSION TO THE TCB. * 00070000
.*                                     * 00080000
.* INPUT OPERANDS:  NONE               * 00090000
*****                                00100000
$ETCB                                00110000
TEXT                                  00120000
*****                                00130000
* EQUATES FOR THE CCP TCB EXTENSION * 00140000
*****                                00150000
TCBCAL EQU 0 OCC CANCEL WHILE TCB IN ALLOC 00160000
* CCP BIT INDICATORS APPEARING IN FIELDS IN THE STANDARD TCB. 00170000
* BIT EQUATES FOR TCBID. 00180000
TCBWID EQU B'11111111' SYSTEM WAIT TASK ID 00190000
SPACE 00200000
* BIT EQUATES FOR TCBFG1. 00210000
TCBUSR EQU B'10000000' CCP USER TASK 00220000
TCBNCL EQU B'00000100' CANCEL NOT ALLOWED 00230000
** TCBUSR+TCBNCL = CCP SYSTEM TASK 00240000
SPACE 00250000
* BIT EQUATES FOR TCBFG2. 00260000
TCBTRC EQU B'10000000' TERMINATION CALL OF CLOSE 00270000
SPACE 00280000
* BIT EQUATES FOR TCBDS1. 00290000
TCBSUS EQU B'01000000' WAIT INDICATOR FOR SUSPENDED TCB 00300000
TCBTRM EQU B'00100000' TCB IS IN CCP TERMINATION 00310000
SPACE 2 00320000
* FIELDS OVERLAYING DSM TCB FIELDS. 00330000
TCBCMP EQU X'2B' USER TASK TERMINATION CODE 00340000
SPACE 1 00350000
TCB@AS EQU X'64' PROGRAM APPENDED STORAGE ADDRESS 00360000
TCBPAS EQU X'65' SIZE (X256) OF PAS 00370000
TCBFDT EQU X'66' SIZE (X256) OF LARGEST FDT 00380000
TCBINQ EQU X'68' Q OF TUBS WITH DATA FOR ACCEPT 00390000
SPACE 1 00400000
TCBIR EQU X'A6' TRANSIENT AREA IAR SAVE FIELD 00410000
TCBPR EQU X'A8' TRANSIENT AREA PMR SAVE FIELD 00420000
TCBX2 EQU X'AA' TRANSIENT AREA REG 2 SAVE FIELD 00430000
TCBX1 EQU X'AC' TRANSIENT AREA REG 1 SAVE FIELD 00440000
SPACE 00450000
TCBDMG EQU X'AD' CCP INTERNAL FLAGS 00460000
TCBRUF EQU B'10000000' RUF PROGRAM ACTIVE 00465000
TCBKRQ EQU B'01000000' CONSOLE WAS PROGRAM REQUESTOR 00470000
TCBALC EQU B'00100000' TASK IS IN ALLOCATE 00480000
TCBNEP EQU B'00010000' TASK IS A NEVER-ENDING-PROGRAM 00490000
TCBCM EQU B'00001000' COMMUNICATIONS MANAGEMENT ID 00500000
TCBMTS EQU B'00000100' TASK IS A MULTI-TERM-SERVICER 00510000
TCBEMG EQU B'00000010' END MSG IS WANTED BY TERMINAL 00520000
TCBSHQ EQU B'00000001' SHUTDOWN COMMAND HAS BEEN KEYED 00530000
SPACE 00540000
TCBWK EQU X'AF' WORK FIELD 00550000
* END OF CCP PROGRAM LEVEL TCB(CM TASK). 00560000
SPACE 1 00570000
* BEGINNING OF CCP EXTENSION FOR TCB (ALL CCP TASKS EXCEPT CM) 00580000

```

TCBXQ	EQU	X'AF'	WAIT Q	00590000
TCBTUB	EQU	X'B1'	LIST ORIGIN - TASK'S OWNED TUBS	00600000
TCBECB	EQU	X'B2'	GENERAL PURPOSE ECB	00610000
*			CCP BITS WITHIN TCBECB	00620000
TCBACW	EQU	X'01'	THIS TASK AT ACCEPT INPUT WAIT	00630000
*				00640000
TCBIIC	EQU	X'B5'	NUMBER OF OUTSTANDING INVITES	00650000
TCBMAX	EQU	X'B6'	MAX # OF MRT Q'D INVITES	00660000
TCBATR	EQU	X'B7'	# OF Q'D TUBS	00670000
		SPACE		00680000
TCBURA	EQU	X'B8'	UNIT RECORD ALLOCATE MASK	00690000
TCBPRS	EQU	B'10000000'	PRINTER IS SHARED	AJS 00695000
TCB501	EQU	B'00100000'	PROGRAM USES 2501	00700000
TCBPRT	EQU	B'00010000'	PROGRAM USES LINE PRINTER	00710000
TCB142	EQU	B'00001000'	PROGRAM USES 1442	00720000
TCBMFU	EQU	B'00000100'	PROGRAM USES MFCU OR MFCM	00730000
TCB741	EQU	B'01000000'	3741 USED AS UNIT RECORD DEVICE	00735000
		SPACE		00740000
TCBFBM	EQU	X'BA'	FILE BIT MASK, DISPLACEMENT	00750000
TCBUSE	EQU	X'BB'	TCB USE COUNT	00760000
TCBOFG	EQU	X'BC'	CCP INTERNAL FLAGS	00763000
TCBSRT	EQU	X'80'	PROGRAM IS A SORT PROGRAM	00766000
TCBPL	EQU	X'BD'	19 BYTE TP PARAMETER LIST	00770000
TCBFRA	EQU	X'D0'	10 BYTE ASSIGN/FREE MASK	00780000
TCBINT	EQU	X'DA'	17 BYTE TIMER QUEUE ELEMENT	00790000
		MEND		00800000

MODULE-\$E000 , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $E000 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 3, MODIFICATION LEVEL 0 MODEL 15 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION SECOND PASS MACRO-INSTRUCTION -- SET SYSTEM 00100000
.* GLOBAL VARIABLES 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . #M-0/1/2/3/4/5/6/7/8 00150000
.* 00160000
.* . NUMBER OF MLTA LINES SUPPORTED 00170000
.* 00180000
.* . #B-0/1/2 00190000
.* 00200000
.* NUMBER OF BSC ADAPTERS SUPPORTED 00210000
.* 00220000
.* . DF-0/1 00230000
.* 00240000
.* 1=DISPLAY FORMAT FACILITY SUPPORTED / 0=NOT SUPPORTED 00250000
.* 00260000
.* . DE-0/1 00270000
.* 00280000
.* 1=DATA MODE ESCAPE SUPPORTED / 0=NOT SUPPORTED 00290000
.* 00300000
.* . MS-0/1 00310000
.* 00320000
.* 1=MINIMUM CORE SYSTEM SUPPORT / 0=NOT MINIMUM CORE SUPPORT 00330000
.* 00340000
.* . RF-0/1 00350000
.* 00360000
.* 1=READ UNDER FORMAT FACILITY SUPPORTED / 0= NOT SUPPORTED. 00370000
.* 00380000
.* . RS-0/1 00390000
.* 00400000
.* 1=RESIDENT ACCEPT INPUT FACILITY SUPPORTED / 0=NOT SUPPORTED 00410000
.* 00420000
***** 00430000
.* $E000 &#M-, &#B-, &DF-, &MS-, &DE-, &RF-, &RS- 00440000
.* 00450000
.*----- SYSTEM GLOBAL VARIABLES -----* 00460000
.* 00470000
.* GBLA &MLA, &BSC . # OF TP LINES. 00480000
.* 00490000
.* GBLB &MLTA . MLTA SUPPORT: 00500000
.* . 1=SUPPORTED / 0=NOT 00510000
.* 00520000
.* GBLB &NOM . INVERSE OF &MLTA 00530000
.* 00540000
.* GBLB &BSCA . BSCA SUPPORT: 00550000
.* . 1=SUPPORTED / 0=NOT 00560000
.* 00570000
.* GBLB &NOB . INVERSE OF &BSCA 00580000
.* 00590000
```



```

GBLB &ONE          . SUPPORT OF SINGLE ADAPTER TYPE 00600000
.*                .   1=MLTA *OR* BSCA / 0=BOTH      00610000
.*                .                                00620000
GBLB &DME          . DATA MODE ESCAPE SUPPORT:    00630000
.*                .   1=SUPPORTED / 0=NOT          00640000
.*                .                                00650000
GBLB &NDME         . INVERSE OF &DME              00660000
.*                .                                00670000
GBLB &DFE         . DISPLAY FORMAT FACILITY:      00680000
.*                .   1=SUPPORTED / 0=NOT          00690000
.*                .                                00700000
GBLB &NDF         . INVERSE OF &DFE              00710000
.*                .                                00720000
GBLB &RUF         . READ UNDER FORMAT FACILITY:   00730000
.*                .   1=SUPPORTED / 0=NOT          00740000
.*                .                                00750000
GBLB &NRUF        . INVERSE OF &RUF              00760000
.*                .                                00770000
GBLB &RAI         . RESIDENT ACCEPT INPUT FACILITY 00780000
.*                .   1=SUPPORTED / 0=NOT          00790000
GBLB &NRAI        . INVERSE OF &RAI              00800000
.*                .                                00810000
.*                .                                00820000
GBLB &MIN         . SUPPORT OF MINIMUM-CORE SYSTEM 00830000
.*                .   1=MINIMUM CORE / 0=NOT      00840000
.*                .                                00850000
TEXT              .                                00860000
.*                .                                00870000
.*----- SET FOR SUPPORT OF MLTA ADAPTER AND DEVICES 00880000
.*                .                                00890000
AIF (&#M EQ '0').#M0 . SKIP IF *NO* MLTA          00900000
.*                .                                00910000
&MLTA SETB 1      . MLTA SUPPORT                  00920000
&MLA  SETA &#M    . # OF MLTA LINES              00930000
AGO   .#MEND      .                                00940000
.*                .                                00950000
.#M0 ANOP        . *NO* MLTA SUPPORT             00960000
&NOM SETB 1      .                                00970000
.*                .                                00980000
.#MEND ANOP      . END OPERAND '#M'              00990000
.*                .                                01000000
.*----- SET FOR SUPPORT OF BSCA ADAPTER AND DEVICES -----* 01010000
.*                .                                01020000
AIF (&#B EQ '0').#B0 . SKIP IF *NO* BSCA SUPPORT 01030000
.*                .                                01040000
&BSCA SETB 1     . BSCA SUPPORT                  01050000
&BSC  SETA &#B   . # OF BSCA LINES              01060000
AGO   .#BEND     .                                01070000
.*                .                                01080000
.#B0 ANOP        . *NO* BSCA SUPPORT             01090000
&NOB SETB 1      .                                01100000
.*                .                                01110000
.#BEND ANOP      . END OPERAND '#B'              01120000
.*                .                                01130000
.*----- SET FOR COMBINATION OF COMMUNICATIONS SUPPORT -----* 01140000
.*                .                                01150000
AIF (&BSCA NE '1').ONE . SKIP IF NO BSCA SUPPORT 01160000
AIF (&MLTA).ONEND   . SKIP IF MLTA SUPPORT       01170000
.*                .                                01180000
.ONE ANOP        . ONLY MLTA *OR* ONLY BSCA     01190000

```



```
. *----- END SYSTEM VARIABLE SET MACRO -----* 01800000
.* 01810000
* PTF LOG NO.-214,LEVEL-04,NAME-$E085
* PTF LOG NO.-214,LEVEL-04,NAME-$E093
MEND 01820000
```

MODULE-\$E001 , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $E001 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 8, MODIFICATION LEVEL 0 OF 5702-SC1 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION SECOND PASS MACRO-INSTRUCTION -- SET GLOBAL 00100000
.* VARIABLES FOR SYSTEM CONTROL FUNCTIONS 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . DK-#### 00150000
.* 00160000
.* DISK DEVICES SUPPORTED -- EACH # IS 1 (SUPPORTED) / 0 (NOT) 00170000
.* 00180000
.* DEVICES REPRESENTED ARE, RESPCETIVELY: R2, F2, D1, D2 00190000
.* 00200000
.* . UR-#### 00210000
.* 00220000
.* U/R DEVICES SUPPORTED -- EACH # IS 1 (SUPPORTED) / 0 (NOT) 00230000
.* 00240000
.* DEVICES REPRESENTED ARE, RESPECTIVELY: MFCU, 1442, 5203, 1403 00250000
.* 00260000
.* . PC-0/1 00270000
.* 00280000
.* 1=PROGRAM REQUEST COUNTS SUPPORTED / 0=NOT SUPPORTED 00290000
.* 00300000
.* . SO-00/01/10 00310000
.* 00320000
.* 00 -- NO SIGN-ON SECURITY CHECKING WHATSOEVER 00330000
.* 01 -- USER ROUTINE PROVIDED FOR SIGN-ON SECURITY CHECKING 00340000
.* 10 -- CCP PASSWORD CHECKING AT SIGN-ON 00350000
.* 00360000
.* . SH-0/1 00370000
.* 00380000
.* 1=UPDATE FILE SHARING SUPPORTED / 0=NOT 00390000
.* 00400000
***** 00410000
.* $E001 &DK-, &UR-, &PC-, &SO-, &SH- 00420000
.* 00430000
.*----- GLOBAL VARIABLES -----* 00440000
.* 00450000
.* GBLB &SHR . UPDATE FILE SHARING: 00460000
.* . 1=SUPPORTED / 0=NOT 00470000
.* 00480000
.* GBLB &USEON . USER SIGN-ON ROUTINE: 00490000
.* . 1=SUPPORTED / 0=NOT 00500000
.* 00510000
.* GBLB &SYSON . CCP PASSWORD SECURITY: 00520000
.* . 1=SUPPORTED / 0=NOT 00530000
.* 00540000
.* GBLB &PUCNT . PROGRAM REQUEST COUNTS: 00550000
.* . 1=SUPPORTED / 0=NOT 00560000
.* 00570000
.* GBLB &URMFU . MFCU SUPPORT: 00580000
.* . 1=SUPPORTED / 0=NOT 00590000
```

```

.*
GBLB &UR142 . 1442 SUPPORT: 00600000
.* . 1=SUPPORTED / 0=NOT 00610000
.* 00620000
GBLB &URPRT . 5203/1403 SUPPORT: 00630000
.* . 1=SUPPORTED / 0=NOT 00640000
.* 00650000
GBLB &D45 . 5445 SUPPORT: 00660000
.* . 1=SUPPORTED / 0=NOT 00670000
.* 00680000
TEXT 00690000
.* 00700000
.* 00710000
.*----- SET FOR SUPPORT OF 5445 DISK -----* 00720000
.* 00730000
AIF ('&DK'(3,2) EQ '00').DKEND . SKIP IF *NO* 5445 00740000
.* 00750000
&D45 SETB 1 . 5445 SUPPORTED 00760000
.* 00770000
.DKEND ANOP . END OPERAND 'DK' 00780000
.* 00790000
.*----- SET FOR SUPPORT OF MFCU -----* 00800000
.* 00810000
AIF ('&UR'(1,1) EQ '0').MFEND . SKIP IF *NO* MFCU SUPPORT 00820000
.* 00830000
&URMFU SETB 1 . MFCU SUPPORTED 00840000
.* 00850000
.MFEND ANOP . END SUB-OPERAND 'MF' OF 'UR' 00860000
.* 00870000
.*----- SET FOR SUPPORT OF 1442 -----* 00880000
.* 00890000
AIF ('&UR'(2,1) EQ '0').RPEND . SKIP IF *NO* 1442 SUPPORT 00900000
.* 00910000
&UR142 SETB 1 . 1442 SUPPORTED 00920000
.* 00930000
.RPEND ANOP . END SUB-OPERAND 'RP' OF 'UR' 00940000
.* 00950000
.*----- SET FOR SUPPORT OF 2503 OR 1403 -----* 00960000
.* 00970000
AIF ('&UR'(3,2) EQ '00').PREND . SKIP IF *NO* PRINTER SUPPORT 00980000
.* 00990000
&URPRT SETB 1 . 5203 OR 1403 SUPPORTED 01000000
.* 01010000
.PREND ANOP . END SUB-OPERAND 'PR' OF 'UR' 01020000
.* 01030000
.*----- SET FOR SUPPORT OF SHARED FILES -----* 01040000
.* 01050000
AIF (&SH EQ '0').SHEND . SKIP IF *NO* FILE SHARING 01060000
.* 01070000
&SHR SETB 1 . FILE SHARING SUPPORTED 01080000
.* 01090000
.SHEND ANOP . END OPERAND 'SH' 01100000
.* 01110000
.*----- SET FOR SUPPORT OF PROGRAM REQUEST COUNTS -----* 01120000
.* 01130000
AIF (&PC EQ '0').PCEND . SKIP IF *NO* REQUEST COUNTING 01140000
.* 01150000
&PUCNT SETB 1 . PROGRAM REQUEST COUNTING 01160000
.* 01170000
.PCEND ANOP . END OPERAND 'PC' 01180000
.* 01190000

```

```
. *----- SET FOR SECURITY OPTION -----* 01200000
.*
.* AIF (&SO EQ '00').SOEND . SKIP IF *NO* SECURITY CHECKING 01210000
.* AIF (&SO EQ '01').USEC . SKIP IF USER SECURITY ROUTINE 01220000
.*
.* &SYSON SETB 1 . CCP PASSWORD 01230000
.* AGO .SOEND 01240000
.*
.* .USEC ANOP . USER SECURITY ROUTINE 01250000
.* &USEON SETB 1 01260000
.*
.* .SOEND ANOP . END OPERAND 'SO' 01270000
.*
.* MEND 01280000
01290000
01300000
01310000
01320000
01330000
01340000
```

```

MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $E002 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 7, MODIFICATION LEVEL 0 OF 5704-SC1 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION SECOND PASS MACRO-INSTRUCTION -- SET GLOBAL 00100000
.* SYMBOLS FOR MLTA COMMUNICATIONS SUPPORT 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . ML-## 00150000
.* 00160000
.* LINE TYPES -- EACH # IS 1 (SUPPORTED) / 0 (NOT) 00170000
.* 00180000
.* TYPES ARE, RESPECTIVELY: STATION-CONTROL, SWITCHED 00190000
.* 00200000
.* . MT-### 00210000
.* 00220000
.* TERMINAL TYPES -- #EACH # IS 1 (SUPPORTED) / 0 (NOT) 00230000
.* 00240000
.* TYPES ARE, RESPECTIVELY: 1050, 2741, BUFFERED-RECEIVE-DEVICE 00250000
.* 00260000
.* . MF-# 00270000
.* 00280000
.* FEATURES SUPPORTED -- EACH # IS 1 (SUPPORTED) / 0 (NOT) 00290000
.* 00300000
.* FEATURES ARE, RESPECTIVELY: MOVE-WITHOUT-TRANSLATE 00310000
.* 00320000
***** 00330000
.$E002 &ML-, &MT-, &MF-, &MS- 00340000
.* 00350000
.*----- GLOBAL VARIABLES -----* 00360000
.* 00370000
.* GBLB &NSCTL . *NO* STATION CONTROL SUPPORT: 00380000
.* . 1=*NO* STA CTL / 0=STA CTL 00390000
.* 00400000
.* GBLB &NSW . *NO* SWITCHED LINE SUPPORT: 00410000
.* . 1=*NO* SW LINE / 0=SW LINES 00420000
.* 00430000
.* GBLB &N1050 . *NO* 1050 SUPPORT: 00440000
.* . 1=*NO* 1050 / 0=1050 SUPPORT 00450000
.* 00460000
.* GBLB &N2741 . *NO* 2741 SUPPORT 00470000
.* . 1=*NO* 2741 / 0=2741 SUPPORT 00480000
.* 00490000
.* GBLB &NBFR . *NO* BUFFERED-RECEIVE SUPPORT: 00500000
.* . 1=*NO* BUFRCV / 0=BUFRCV 00510000
.* 00520000
.* GBLB &NMOVE . *NO* MOVE-WITHOUT-TRANSLATE: 00530000
.* . 1=*NO* MOVE / 0=MOVE SUPPORT 00540000
.* 00550000
.* GBLB &NCPUM . *NO* CPU S TYPE MESSAGES: 00552000
.* . 1=*NO* MSGS / 0=SEND MSGS 00554000
.* 00556000
.* TEXT 00560000

```



```

MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $E003 -- RELEASE 3 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 3, MODIFICATION LEVEL 0 OF 5704-SC1 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION SECOND PASS MACRO-INSTRUCTION -- SET GLOBAL 00100000
.* SYMBOLS FOR BSCA SUPPORT 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . BL-#### 00150000
.* 00160000
.* BSCA LINE TYPES -- EACH # IS 1 (SUPPORTED) / 0 (NOT) 00170000
.* 00180000
.* TYPES REPRESENTED ARE, RESPECTIVELY: PP, MP, CS, DIAL 00190000
.* 00200000
.* . BF-##### 00210000
.* 00220000
.* BSCA FEATURES SUPPORTED -- EACH # IS 1 (SUPPORTED) / 0 (NOT) 00230000
.* 00240000
.* FEATURES REPRESENTED ARE, RESPECTIVELY: GETMSG, ITB, RECSEP, 00250000
.* RESPOL, AUTORS, EBCDIC, ASCII, XPRNCY 00260000
.* 00270000
.* . BT-##### 00280000
.* 00290000
.* BSCA TERMINALS SUPPORTED -- EACH # IS 1 (SUPPORTED) / 0 (NOT) 00300000
.* 00310000
.* TYPES REPRESENTED ARE, RESPECTIVELY: 3275M1, 3277M1, 3284M1, 00320000
.* 3286M1, 3275M2, 3277M2, 3284M2, 3286M2, 3735, CPU, 3741 00330000
.* 00340000
.* THE LAST 5 POSITIONS -- ALWAYS 0 -- ARE NOT USED 00350000
.* 00360000
.* . INP-0/1 00370000
.* 00380000
.* INTERVAL POLLING SUPPORT -- 1=SUPPORT 0=NO SUPPORT 00390000
.* 00400000
***** 00410000
.$E003 &BL-,&BF-,&BT-,&INP-,&BIA-,&BY- 00420000
.* 00430000
.*----- GLOBAL VARIABLES -----* 00440000
.* 00450000
.* GBLB &NCPU . *NO* CPU SUPPORT: 00460000
.* . 1=*NO* CPU / 0=CPU SUPPORT 00470000
.* 00480000
.* GBLB &NITB . *NO* ITB SUPPORT: 00490000
.* . 1=*NO* ITB / 0=ITB SUPPORT 00500000
.* 00510000
.* GBLB &NMSG . *NO* GETMSG SUPPORT: 00520000
.* . 1=*NO* GETMSG / 0=GETMSG 00530000
.* 00540000
.* GBLB &NTSP . *NO* TRANSPARENCY SUPPORT: 00550000
.* . 1=*NO* TSP / 0=TSP SUPPORT 00560000
.* 00570000
.* GBLB &N32 . *NO* 3270 SUPPORT: 00580000
.* . 1=*NO* 3270 / 0=3270 SUPPORT 00590000

```

```

.*
.*      GBLB  &N37      . *NO* 3735 SUPPORT:      00600000
.*      . 1=*NO* 3735 / 0=3735 SUPPORT 00610000
.*      . 00620000
.*      GBLB  &N41      . *NO* 3741 SUPPORT:      00630000
.*      . 1=*NO* 3741 / 0=3741 SUPPORT 00640000
.*      . 00650000
.*      GBLB  &NAS      . *NO* ASCII SUPPORT:     00660000
.*      . 1=*NO* ASCII / 0=ASCII SUPRT 00670000
.*      . 00680000
.*      GBLB  &NPP      . *NO* POINT-TO-POINT SUPPORT: 00690000
.*      . 1=*NO* P-TO-P / 0=P-TO-P     00700000
.*      . 00710000
.*      GBLB  &NMP      . *NO* MULTIPOINT TRIBUTARY SPRT 00720000
.*      . 1=*NO* MP / 0=MP SUPPORT      00730000
.*      . 00740000
.*      GBLB  &NSWL     . *NO* SWITCHED BSCA LINES     00750000
.*      . 1=*NO* SW / 0=SW LINE SUP'RT 00760000
.*      . 00770000
.*      GBLB  &NCS      . *NO* CONTROL STATION SUPPORT: 00780000
.*      . 1=*NO* CS / 0=CS SUPPORT      00790000
.*      . 00800000
.*      GBLB  &NINT     . INTERVAL POLLING SUPPORT     00810000
.*      . 1=NO SUPPORT 0=SUPPORT       00820000
.*      . 00830000
.*      GBLB  &NBDA     . DA SUPPORT                    00840000
.*      . 00850000
.*      GBLB  &NPBY     . *NO* PRINTER BUSY SUPPORT:   00852000
.*      . 1=NO SUPPORT 0=SUPPORT       00854000
.*      . 00856000
.*      . 00860000
.*      TEXT           . 00870000
.*      . 00880000
.*      . 00890000
.*----- SET FOR LINE TYPES SUPPORTED -----* 00900000
.*      . 00910000
.*      AIF  ('&BL'(1,1) EQ '1').BL2 . SKIP IF PP SUPPORT      00920000
.*      . 00930000
&NPP  SETB  1          . SET *NO* PP SUPPORT      00940000
.*      . 00950000
.BL2  ANOP           . 00960000
.*      AIF  ('&BL'(2,1) EQ '1').BL3 . SKIP IF MP SUPPORT      00970000
.*      . 00980000
&NMP  SETB  1          . SET *NO* MP SUPPORT      00990000
.*      . 01000000
.BL3  ANOP           . 01010000
.*      AIF  ('&BL'(3,1) EQ '1').BL4 . SKIP IF CS SUPPORT      01020000
.*      . 01030000
&NCS  SETB  1          . SET *NO* CS SUPPORT      01040000
.*      . 01050000
.BL4  ANOP           . 01060000
.*      AIF  ('&BL'(4,1) EQ '1').BLEND . SKIP IF DIAL SUPPORT    01070000
.*      . 01080000
&NSWL SETB  1          . SET *NO* DIAL SUPPORT    01090000
.*      . 01100000
.BLEND ANOP          . END OPERAND 'BL'      01110000
.*      . 01120000
.*----- SET FOR BSCA FEATURE SUPPORT -----* 01130000
.*      . 01140000
.*      AIF  ('&BF'(1,1) EQ '1').BF2 . SKIP IF GETMSG=YES     01150000
.*      . 01160000

```

&NMSG	SETB	1	. SET *NO* GETMSG SUPPORT	01170000
.*				01180000
.BF2	ANOP			01190000
.*				01200000
	AIF	(&INP EQ '1').IN0	. SKIP IF INTERVAL POLLING	01210000
.*			. SUPPORT	01220000
.*				01230000
&NINT	SETB	1	. SET NO INTERVAL POLLING SUPP	01240000
.*				01250000
.IN0	ANOP			01260000
	AIF	('&BF'(2,1) EQ '1').BF3	. SKIP IF ITB SUPPORT	01270000
.*				01280000
&NITB	SETB	1	. SET *NO* ITB SUPPORT	01290000
.*				01300000
.BF3	ANOP			01310000
	AIF	('&BF'(7,1) EQ '1').BF8	. SKIP IF ASCII SUPPORTED	01320000
.*				01330000
&NAS	SETB	1	. SET *NO* ASCII SUPPORT	01340000
.*				01350000
.BF8	ANOP			01360000
	AIF	('&BF'(8,1) EQ '1').BFEND	. SKIP IF XPRNCY-YES	01370000
.*				01380000
&NTSP	SETB	1	. SET *NO* TRANSPARENCY	01390000
.*				01400000
.BFEND	ANOP		. END OPERAND 'BF'	01410000
.*				01411000
	AIF	(&BY EQ '1').BYEND	. SKIP IF PRINTER BUSY SUPPORT	01412000
.*				01413000
&NPBY	SETB	1	. SET *NO* PRINTER BUSY SUPPORT	01414000
.*				01415000
.BYEND	ANOP			01416000
.*				01420000
.*	-----	SET FOR BSCA DEVICE SUPPORT	-----*	01430000
.*				01440000
	AIF	('&BT'(1,8) NE '00000000').BT2	. SKIP IF 3270 SUPPORT	01450000
.*				01460000
&N32	SETB	1	. SET *NO* 3270 SUPPORT	01470000
.*				01480000
.BT2	ANOP			01490000
	AIF	('&BT'(9,1) EQ '1').BT3	. SKIP IF 3735 SUPPORTED	01500000
.*				01510000
&N37	SETB	1	. SET *NO* 3735 SUPPORT	01520000
.*				01530000
.BT3	ANOP			01540000
	AIF	('&BT'(10,1) EQ '1').BT4	. SKIP IF CPU SUPPORTED	01550000
.*				01560000
&NCPU	SETB	1	. SET *NO* CPU SUPPORT	01570000
.*				01580000
.BT4	ANOP			01590000
	AIF	('&BT'(11,1) EQ '1').BTEND	SKIP IF 3741 SUPPORTED	01600000
.*				01610000
&N41	SETB	1	. SET *NO* 3741 SUPPORT	01620000
.*				01630000
.BTEND	ANOP		. END OPERAND 'BT'	01640000
.*				01650000
.*	-----	SET 'DA' SUPPORT	-----*	01660000
	AIF	(&BIA EQ '1').ENDDA	. SKIP IF 'DA'-YES	01670000
.*				01680000
&NBDA	SETB	1	. SET *NO* DA SUPPORT	01690000
.*				01700000

.ENDDA ANOP

. END PARAMETER 'DA'

01710000

.*

01720000

MEND

01730000

```

MACRO                                00010000
$E030                                00020000
.*                                  00030000
.* STATUS:  RELEASE 3                * 00040000
.*                                  * 00050000
.* NAME:  $E030                      * 00060000
.*                                  * 00070000
.* FUNCTION:  DEFINE THE LOCATIONS OF DATA AREAS WITHIN CCP COMMON. * 00080000
.* WHERE APPROPRIATE, DEFINE THE BIT SIGNIFICANCE FOR THOSE DATA * 00090000
.* AREAS.  DEFINE THE LENGTHS AND LOCATIONS OF FIELDS IN          * 00100000
.* THE FOLLOWING WORK AREAS:  COMMAND PROCESSOR, ALLOCATION,        * 00110000
.* TERMINATION, COMMUNICATIONS MANAGEMENT.                        * 00120000
.*                                  * 00130000
.*                                  * 00140000
*****                                00150000
.*                                  * 00160000
.*----- SYSTEM GLOBAL VARIABLES -----* 00170000
.*                                  * 00180000
.* GBLB  &MLTA                        . MLTA SUPPORT:          00190000
.*                                  . 1=SUPPORTED / 0=NOT      00200000
.*                                  * 00210000
.* GBLB  &NOM                        . INVERSE OF &MLTA        00220000
.*                                  * 00230000
.* GBLB  &BSCA                       . BSCA SUPPORT:          00240000
.*                                  . 1=SUPPORTED / 0=NOT      00250000
.*                                  * 00260000
.* GBLB  &NOB                        . INVERSE OF &BSCA        00270000
.*                                  * 00280000
.* GBLB  &ONE                        . SUPPORT OF SINGLE ADAPTER TYPE 00290000
.*                                  . 1=MLTA *OR* BSCA / 0=BOTH 00300000
.*                                  * 00310000
.* GBLB  &DME                        . DATA MODE ESCAPE SUPPORT:    00320000
.*                                  . 1=SUPPORTED / 0=NOT      00330000
.*                                  * 00340000
.* GBLB  &NDME                       . INVERSE OF &DME          00350000
.*                                  * 00360000
.* GBLB  &DFF                        . DISPLAY FORMAT FACILITY:      00370000
.*                                  . 1=SUPPORTED / 0=NOT      00380000
.*                                  * 00390000
.* GBLB  &NDF                        . INVERSE OF &DFF          00400000
.*                                  * 00410000
.* GBLB  &MIN                        . SUPPORT OF MINIMUM-CORE SYSTEM 00420000
.*                                  . 1=MINIMUM CORE / 0=NOT    00430000
.*                                  * 00440000
.*----- GLOBAL VARIABLES -----* 00450000
.*                                  * 00460000
.* GBLB  &SHR                        . UPDATE FILE SHARING:         00470000
.*                                  . 1=SUPPORTED / 0=NOT      00480000
.*                                  * 00490000
.* GBLB  &USEON                      . USER SIGN-ON ROUTINE:        00500000
.*                                  . 1=SUPPORTED / 0=NOT      00510000
.*                                  * 00520000
.* GBLB  &SYSON                      . CCP PASSWORD SECURITY:        00530000
.*                                  . 1=SUPPORTED / 0=NOT      00540000
.*                                  * 00550000
.* GBLB  &PUCNT                      . PROGRAM REQUEST COUNTS:       00560000
.*                                  . 1=SUPPORTED / 0=NOT      00570000
.*                                  * 00580000
.* GBLB  &URMFU                      . MFCU SUPPORT:                00590000

```

```

.*          . 1=SUPPORTED / 0=NOT          00600000
.*          . 00610000
.* GBLB &UR142 . 1442 SUPPORT:          00620000
.*          . 1=SUPPORTED / 0=NOT          00630000
.*          . 00640000
.* GBLB &URPRT . 5203/1403 SUPPORT:        00650000
.*          . 1=SUPPORTED / 0=NOT          00660000
.*          . 00670000
.* GBLB &D45   . 5445 SUPPORT:          00680000
.*          . 1=SUPPORTED / 0=NOT          00690000
.*          . 00700000
.*          . 00710000
.*----- GLOBAL VARIABLES -----* 00720000
.*          . 00730000
.* GBLB &NSCTL . *NO* STATION CONTROL SUPPORT: 00740000
.*          . 1=*NO* STA CTL / 0=STA CTL 00750000
.*          . 00760000
.* GBLB &NINT  . *NO* INTERVAL POLLING SUPPORT 00770000
.*          . 1=NO INT POLL / 0=INT POLLING 00780000
.*          . 00790000
.* GBLB &NSW   . *NO* SWITCHED LINE SUPPORT: 00800000
.*          . 1=*NO* SW LINE / 0=SW LINES 00810000
.*          . 00820000
.* GBLB &N1050 . *NO* 1050 SUPPORT:          00830000
.*          . 1=*NO* 1050 / 0=1050 SUPPORT 00840000
.*          . 00850000
.* GBLB &N2741 . *NO* 2741 SUPPORT          00860000
.*          . 1=*NO* 2741 / 0=2741 SUPPORT 00870000
.*          . 00880000
.* GBLB &NBFR  . *NO* BUFFERED-RECEIVE SUPPORT: 00890000
.*          . 1=*NO* BUFRCV / 0=BUFRCV    00900000
.*          . 00910000
.* GBLB &NMOVE . *NO* MOVE-WITHOUT-TRANSLATE: 00920000
.*          . 1=*NO* MOVE / 0=MOVE SUPPORT 00930000
.*          . 00940000
.*          . 00950000
.*----- GLOBAL VARIABLES -----* 00960000
.*          . 00970000
.* GBLB &NCPU  . *NO* CPU SUPPORT:          00980000
.*          . 1=*NO* CPU / 0=CPU SUPPORT   00990000
.*          . 01000000
.* GBLB &NITB  . *NO* ITB SUPPORT:          01010000
.*          . 1=*NO* ITB / 0=ITB SUPPORT   01020000
.*          . 01030000
.* GBLB &NMSG  . *NO* GETMSG SUPPORT:          01040000
.*          . 1=*NO* GETMSG / 0=GETMSG    01050000
.*          . 01060000
.* GBLB &NTSP  . *NO* TRANSPARENCY SUPPORT: 01070000
.*          . 1=*NO* TSP / 0=TSP SUPPORT   01080000
.*          . 01090000
.* GBLB &N32   . *NO* 3270 SUPPORT:          01100000
.*          . 1=*NO* 3270 / 0=3270 SUPPORT 01110000
.*          . 01120000
.* GBLB &N37   . *NO* 3735 SUPPORT:          01130000
.*          . 1=*NO* 3735 / 0=3735 SUPPORT 01140000
.*          . 01150000
.* GBLB &NAS   . *NO* ASCII SUPPORT:          01160000
.*          . 1=*NO* ASCII / 0=ASCII SUPRT 01170000
.*          . 01180000
.* GBLB &NPP   . *NO* POINT-TO-POINT SUPPORT: 01190000

```

```

.*          . 1=*NO* P-TO-P / 0=P-TO-P          01200000
.*          . 01210000
GBLB &NMP    . *NO* MULTIPOINT TRIBUTARY SPRT 01220000
.*          . 1=*NO* MP / 0=MP SUPPORT          01230000
.*          . 01240000
GBLB &NSWL   . *NO* SWITCHED BSCA LINES          01250000
.*          . 1=*NO* SW / 0=SW LINE SUP'RT     01260000
.*          . 01270000
GBLB &NCS    . *NO* CONTROL STATION SUPPORT:    01280000
.*          . 1=*NO* CS / 0=CS SUPPORT          01290000
.*          . 01300000
TEXT        . 01310000
TITLE 'CCP COMMON AREA'                        01320000
*****
* COMMUNICATIONS CONTROL PROGRAM *             01330000
* COMMON AREA *                               01340000
*****
SPACE 2                                         01350000
***** NOTE - DO NOT CHANGE ORDER OF THESE EXTRNS WITHOUT CHANGING ***** 01360000
***** MODULE $CC4V1. *****                 01370000
SPACE 2                                         01380000
EXTRN $CC4TR(3)                                01390000
EXTRN $CC4TX(3)                                01400000
EXTRN $CC4PI(3)                                01410000
EXTRN $CC4MS(3)                                01420000
EXTRN $CC4GM(3)                                01430000
EXTRN $CC4FM(3)                                01440000
EXTRN $CC4MX(3)                                01450000
EXTRN $CC4MV(3)                                01460000
EXTRN $CC4SR(3)                                01470000
EXTRN CC4TI2(3)                                01480000
EXTRN $CC4CP(3)                                01490000
EXTRN $CC4TM(3)                                01500000
EXTRN $CCTI2(3)                                01510000
EXTRN $CC4IG(3)                                01520000
EXTRN XSNT1(3)                                 01530000
EXTRN XSNT2(3)                                 01540000
EXTRN $CC4TI(3)                                01550000
EXTRN CC4TH(3)                                 01560000
EXTRN XSNTHT(3)                                01570000
EXTRN $CC4OC(3)                                01580000
EXTRN $CC4NQ(3)                                01590000
EXTRN $CC4DQ(3)                                01600000
EXTRN ADDRS1(3)                                01610000
EXTRN $CC4V1(3)                                01620000
ENTRY $CCCOM                                  01630000
EJECT                                          01640000
$CCCOM EQU $CC4#1 CCP COMMON                  01650000
SPACE 2                                       01660000
*-----ENTRY POINT ADDRESS DEFINITION-----* 01670000
*@CC4TR *                                     01680000
DC AL2($CC4TR) TRANSIENT RETURN ADDRESS      01690000
SPACE 1                                       01700000
*@CC4TX *                                     01710000
DC AL2($CC4TX) TRANSIENT TRANSFER CONTROL @ 01720000
SPACE 1                                       01730000
*@CC4PI *                                     01740000
*@CC4TA *                                     01750000
DC AL2($CC4PI) 1ST LEVEL TRANSIENT INVOCATION @ 01760000
SPACE 1                                       01770000

```

*@CC4IS		*	01800000
DC	AL2(\$CC4IS)	COMMUNICATIONS I/O INTERFACE @	01810000
SPACE	1		01820000
*@CC4GM		*	01830000
DC	AL2(\$CC4GM)	GETMAIN ENTRY ADDRESS	01840000
SPACE	1		01850000
*@CC4FM		*	01860000
DC	AL2(\$CC4FM)	FREEMAIN ENTRY ADDRESS	01870000
SPACE	1		01880000
*@MLTIO		*	01890000
DC	AL2(ADDRS1)	MLTA IOCS ENTRY POINT @.	01900000
*		FILLED BY STARTUP AT EXECUTION	01910000
*		ADDRESS OF MAINTENANCE ADDRESS	01920000
*		TABLE AT PRE-START UP	01930000
	SPACE 1		01940000
*@MLTOP		*	01950000
DC	AL2(0)	MLTA OPEN ENTRY POINT @.	01960000
*		FILLED BY STARTUP	01970000
	SPACE 1		01980000
*@USECW		*	01990000
DC	AL2(0)	USER SECURITY DATA WORK AREA @	02000000
*		FILLED BY STARTUP	02010000
	SPACE 1		02020000
*@CC4MX		*	02030000
DC	AL2(\$CC4MX)	MOVE FOREVER ROUTINE ADDRESS	02040000
SPACE	1		02050000
*@C4TI2		*	02060000
DC	AL2(CC4TI2)	PROGRAM TERMINATION INTERFACE @	02070000
SPACE	1		02080000
*@CC4TI		*	02090000
DC	AL2(\$CC4TI)	OTHER TASK TERMINATE ADDRESS	02100000
SPACE	1		02110000
*@CC4SR		*	02120000
DC	AL2(\$CC4SR)	ENTRY ADDRESS IN MOVE ROUTINE	02130000
SPACE	1		02140000
*@CC4TH		*	02150000
DC	AL2(CC4TH)	@ TERMINATION INTERFACE	02160000
SPACE	1		02170000
*@BTRAC		*	02180000
DC	AL2(0)	CCP BSCA TRACE ROUTINE	02190000
*		ADDRESS FILLED BY STARTUP	02200000
*		IF TRACEMLMP SPECIFIED	02210000
	SPACE 1		02220000
*@MTRAC		*	02230000
DC	AL2(0)	CCP MLTA TRACE ROUTINE	02240000
*		ADDRESS FILLED BY STARTUP	02250000
*		IF TRACEMLTA SPECIFIED	02260000
	EJECT		02270000
-----TRANSIENT COMMUNICATION AREA-----			02280000
*SV1TAX		*	02290000
DC	AL2(\$CC4CP)	TRANSIENT AREA 1 PARAMETER AREA	02300000
SPACE	1		02310000
*SV2TAX		*	02320000
DC	AL2(\$CC4TM)	TRANSIENT AREA 2 PARAMETER AREA	02330000
SPACE	5		02340000
----- ADDRESSES OF SYSTEM TASK CONTROL BLOCKS -----			02350000
* THESE ADDRESSES ARE FILLED IN BY STARTUP.			02360000
	SPACE 2		02370000
*@CMTCB		*	02380000
DC	AL2(0)	@ COMMUNICATIONS MANAGEMENT TCB	02390000

SPACE 1			02400000
AIF (&NDF).NDFFS		. SKIP IF NO DFF.	02410000
*@DFTCB		*	02420000
DC AL2(DFA000)		@ OF DISPLAY FORMAT FACILITY TCB	02430000
AGO .CONTN		SKIP IF DFF SUPPORTED.	02440000
.NDFFS ANOP		CONTINUE HERE IF NO DFF.	02450000
DC XL2'0000'			02460000
.CONTN ANOP		CONTINUE HERE.	02470000
SPACE 1			02480000
*@TMTCB		*	02490000
DC AL2(0)		@ TERMINATION TCB	02500000
SPACE 1			02510000
*@CPTCB		*	02520000
DC AL2(0)		@ COMMAND PROCESSOR TCB	02530000
SPACE 1			02540000
*@AVTCB		*	02550000
DC AL2(0)		@ AVAILABLE TCB'S	02560000
EJECT			02570000
----- SYSTEM LEVEL FLAGS -----			02580000
SPACE 2			02590000
*\$FLGA		*	02600000
DC AL1(&DFF*128)		FIRST SYSTEM LEVEL FLAG BYTE	02610000
* BITS IN \$FLGA ARE SET BY STARTUP			02620000
*#DFFOK BIT0		DFF IS SUPPORTED	02630000
*#CPSOB1 BIT1		SIGN ON PASSWORD REQUIRED	02640000
*#CPSOB2 BIT2		SIGN ON USER PASSWORD REQUIRED	02650000
*#CPSHUT BIT3		SHUTDOWN HAS BEEN REQUESTED	02660000
*#CPSU BIT4		STARTUP IS IN PROCESS	02670000
*#DSKFS BIT5		DISK FILE SHARING IS SUPPORTED	02680000
*#CPISNW BIT7		\$CC4IS IS NOT TO ISSUE WAIT	02690000
*		AFTER POSTING \$CC4CM	02700000
SPACE 3			02710000
*\$FLGB		*	02720000
DC AL1(0)		SECOND SYSTEM LEVEL FLAG BYTE	02730000
* BITS IN \$FLGB ARE SET BY STARTUP			02740000
*#SUALL BIT0		SUSPEND ALL IN EFFECT	02750000
*#SUINT BIT1		SUSPEND INIT IN EFFECT	02760000
*#SUCMD BIT2		SUSPEND COMMANDS IN EFFECT	02770000
*#TDERR BIT5		* 1-PERMANENT DISK I/O ERROR	02780000
*		* TRACING TO DISK HAS BEEN STOPP	02790000
*#CPCAN BIT6		CCP CANCEL RECEIVED	02800000
*#PUCNT BIT7		PROGRAM USE COUNTING IS ON	02810000
SPACE 3			02820000
*\$FLGC		*	02830000
DC AL1(&NINT*2)		THIRD SYSTEM LEVEL FLAG BYTE	02840000
*#MTRAC BIT0		MLTA TRACE IS ON	02850000
*#BTRAC BIT1		BSCA TRACE IS ON	02860000
*#CTRAC BIT2		CCP TRACE IS ON	02870000
*#PUTTP BIT3		PUT ONLY TPBUFF FOR GETMAIN	02880000
*#INVPL BIT4		INVITE PL GETMAIN AREA OF TPBUFF	02890000
*#D3340 BIT5		IPL FROM THE 3340 DISK	02900000
*#INTNSP BIT6		INT TIMER NOT SUPPORTED	02910000
*#3340 BIT7		EXTENDED ADDRESSING IN CPU	02920000
EJECT			02930000
----- SYSTEM CONSTANTS -----			02940000
*X\$0000		*	02950000
DC XL2'0000'		CONSTANT XL2'0000'	02960000
SPACE 1			02970000
*X\$0001		*	02980000
DC XL1'01'		CONSTANT XL2'0001' ALONG WITH	02990000

*		CONSTANT X\$0000	03000000
	SPACE 1		03010000
*X\$0002		*	03020000
	DC XL2'0002'	CONSTANT XL2'0002'	03030000
	SPACE 1		03040000
*X\$0004		*	03050000
	DC XL2'0004'	CONSTANT XL2'0004'	03060000
	SPACE 1		03070000
*X\$FFFF		*	03080000
	DC XL2'FFFF'	CONSTANT XL2'FFFF'	03090000
	EJECT		03100000
-----	QUEUE ADDRESS POINTERS	-----	03110000
*@ALOCQ		*	03120000
	DC AL2(0)	Q OF TASKS WAITING TO BE ALLOC'D	03130000
	SPACE 1		03140000
*@WATSK		*	03150000
	DC AL2(0)	ALLOCATE WAIT QUEUE	03160000
	SPACE 1		03170000
*@QTUBS		*	03180000
	DC AL2(0)	@ Q TUB'S WAITING FOR TCB/CORE	03190000
	SPACE 1		03200000
*@GMWTQ		*	03210000
	DC AL2(0)	@ GETMAIN TCB QUEUE	03220000
	SPACE 1		03230000
*@DFEQ		*	03240000
	DC AL2(0)	QUEUE FOR REQUESTS TO DFF TASK	03250000
	SPACE 1		03260000
*@PRLQ		*	03270000
	DC AL2(0)	@ OF PARAMETER LIST QUED FOR CM	03280000
	EJECT		03290000
-----	LIST ADDRESS POINTERS	-----	03300000
*@TALST		*	03310000
	DC AL2(TA@PGM-10)	@ PROGRAM LIST IN \$CC4TA	03320000
	SPACE 1		03330000
*@TCORG		*	03340000
	DC AL2(0)	@ OF THE TCB LIST	03350000
	SPACE 1		03360000
*@FSQB		*	03370000
	DC AL2(0)	@ FIRST AVAILABLE FSQE	03380000
*		FILLED BY STARTUP	03390000
	SPACE 1		03400000
*@LCB#1		*	03410000
	DC AL2(0)	@ FIRST LCB IN SYSTEM	03420000
*		FILLED BY STARTUP	03430000
	SPACE 1		03440000
*@TUBQ		*	03450000
	DC AL2(0)	@ OF THE FIRST TUB IN THE SYSTEM	03460000
*		FILLED BY STARTUP	03470000
	SPACE 1		03480000
*@DFCT		*	03490000
	DC AL2(0)	@ OF THE SHORT DTF ADDRESS LIST	03500000
*		FILLED BY STARTUP	03510000
	SPACE 1		03520000
*@TNT		*	03530000
	DC AL2(0)	@ OF 1ST TERMINAL NAME TABLE ENT	03540000
*		FILLED BY STARTUP	03550000
	SPACE 1		03560000
*@XDT		*	03570000
	DC AL2(0)	@ 1ST SYMBOLIC DFCT NAME ENTRY	03580000
*		FILLED BY STARTUP	03590000

```

EJECT 03600000
*----- MAIN STORAGE CONTROL BLOCK ( TPBUFF AND UPA )-----* 03610000
* ALL FIELDS FILLED BY STARTUP 03620000
SPACE 2 03630000
*@BUFA * 03640000
DC AL2(0) @ OF FIRST FREE SEGMENT 03650000
SPACE 1 03660000
*#NBND * 03670000
DC AL2(0) RESERVED - MUST BE ZEROS 03680000
SPACE 1 03690000
*@LOBND * 03700000
DC AL2(0) @ LO-BOUND GETMAIN AREA 03710000
SPACE 1 03720000
*@HIBND * 03730000
DC AL2(0) @ HI-BOUND GETMAIN AREA 03740000
SPACE 1 03750000
*#GMS * 03760000
DC AL2(0) SIZE OF LARGEST FREE SPACE 03770000
SPACE 1 03780000
*#TPBUF * 03790000
DC AL2(0) ORIGINAL SIZE OF TPBUFF 03800000
SPACE 1 03810000
*@UPA * 03820000
DC AL1(0) USER PROGRAM AREA ATR 03830000
SPACE 1 03840000
*#AVCOR * 03850000
DC AL1(0) # 2K BLOCKS NOT GIVEN TO NEP'S 03860000
EJECT 03870000
*----- GENERAL AREAS -----* 03880000
*@PUCNT * 03890000
DC AL2(0) @ PROGRAM USE COUNT TABLE 03900000
* FILLED BY STARTUP 03910000
SPACE 1 03920000
*@TUSTG * 03930000
DC AL2(0) @ OF TUB IN STAGING NOW(CP TASK) 03940000
SPACE 1 03950000
*@KNTUB * 03960000
DC AL2(KNTUB) @ OF CONSOLE TUB 03970000
SPACE 1 03980000
*@PTX * 03990000
DC AL2(0) @ OF PCT MASTER INDEX 04000000
* FILLED BY STARTUP 04010000
SPACE 1 04020000
*@PTXCS * 04030000
DC AL2(0) C/S VALUE OF PCT DISK START 04040000
* FILLED BY STARTUP 04050000
SPACE 1 04060000
*#PCTLN * 04070000
DC AL1(0) LONGEST LEN PCT USED BY PGM RQST 04080000
* FILLED BY STARTUP 04090000
SPACE 1 04100000
*#DFCT * 04110000
DC AL1(0) NO. ENTRIES IN SHORT DTF @ LIST 04120000
* FILLED BY STARTUP 04130000
SPACE 1 04140000
*#SETID * 04150000
DC AL1(0) ID OF ASSIGNMENT SET IN USE 04160000
* FILLED BY STARTUP 04170000
SPACE 1 04180000
*#XDT * 04190000

```

DC	AL1(0)	NUMBER OF SYMFILE STATEMENTS	04200000
*		FILLED BY STARTUP	04210000
SPACE	1		04220000
*@UALFA		*	04230000
DC	AL2(0)	UNCHANGING START OF UPA	04240000
*		FILLED BY STARTUP	04250000
SPACE	1		04260000
*@TKFSB		*	04270000
DC	AL2(0)	FSB AREA FOR TCB'S	04280000
*		FILLED BY STARTUP	04290000
SPACE	1		04300000
*CPLPWD		*	04310000
DC	AL1(0)	LENGTH OF SIGN ON PASSWORD	04320000
*		FILLED BY STARTUP	04330000
SPACE	1		04340000
*CPPSWD		*	04350000
DC	CL6'	CCP SIGN ON PASSWORD	04360000
*		FILLED BY STARTUP	04370000
EJECT			04380000
-----	CCP DUMP AREA DISK ADDRESSES	-----	04390000
*	ALL FIELDS FILLED BY STARTUP		04400000
*@DSTRT		*	04410000
DC	AL2(0)	DUMP AREA START C/S	04420000
SPACE	1		04430000
*@DEND		*	04440000
DC	AL2(0)	DUMP AREA END C/S	04450000
SPACE	1		04460000
*@DNEXT		*	04470000
DC	AL2(0)	DUMP AREA NEXT ENTRY C/S	04480000
SPACE	1		04490000
*#CPFLQ		*	04500000
DC	AL1(0)	\$CCPFILE DEVICE Q-BYTE	04510000
EJECT			04520000
-----	FIXED LOCATIONS OF RESIDENT CODE	-----	04530000
*CC4TA		*	04540000
*CC4PI		*	04550000
L	@CC4TA,IAR	LOAD IAR OF \$CC4TA	04560000
SPACE	1		04570000
*CC4IS		*	04580000
L	@CC4IS,IAR	LOAD IAR OF @CC4IS	04590000
SPACE	1		04600000
*CC4GM		*	04610000
L	@CC4GM,IAR	LOAD IAR OF @CC4GM	04620000
SPACE	1		04630000
*CC4FM		*	04640000
L	@CC4FM,IAR	LOAD IAR OF @CC4FM	04650000
SPACE	1		04660000
*USECW		*	04670000
L	@USECW,IAR	LOAD IAR OF @USECW	04680000
SPACE	1		04690000
*CC4MX		*	04700000
L	@CC4MX,IAR	LOAD IAR OF @CC4MX	04710000
SPACE	1		04720000
*C4TI2		*	04730000
L	@C4TI2,IAR	LOAD IAR OF @C4TI2	04740000
SPACE	1		04750000
*CC4TI		*	04760000
L	@CC4TI,IAR	LOAD IAR OF @CC4TI	04770000
SPACE	1		04780000
*CC4TT		*	04790000

L	X'004C',IAR	LOAD IAR OF @CTRAC	04800000
SPACE 1			04810000
*CC4SR		*	04820000
L	@CC4SR,IAR	LOAD IAR OF @CC4SR	04830000
SPACE 1			04840000
*CC4FR		*	04850000
L	@CC4FR,IAR	LOAD IAR OF @CC4FR	04860000
SPACE 2			04870000
-----	ADDITIONAL ENTRY POINT ADDRESS DEFINITION	-----	04880000
*@CC4FR		*	04890000
DC	AL2(0)	SET BY STARTUP	04900000
SPACE 1			04910000
*@CC4II		*	04920000
DC	AL2(\$CC4II)	\$CC4II ENTRY POINT - NEEDED BY	04930000
*		* SHUTDOWN FOR BUFFER.	04940000
SPACE 5			04950000
-----	MAINTENANCE SPACE	----- 2 BYTES -----	04960000
SPACE 1			04970000
DC	2XL1'00'	MAINTENANCE SPACE	04980000
SPACE 1			04990000
*\$END1			05000000
TITLE	'WORK AREAS'		05010000
*****	DFW WORK AREA	15 BYTES*****	05020000
*\$DFWK			05030000
*@DFFIX		*	05040000
DC	AL2(0)	C/S OF DFF FORMATS INDEX	05050000
*		FILLED BY STARTUP	05060000
SPACE 1			05070000
*#DFQ		*	05080000
DC	AL1(0)	FDT DEVICE Q-BYTE	05090000
*		FILLED BY STARTUP	05100000
SPACE 1			05110000
*#DF1ID		ID OF TASK 'OWNING' SPACE IN	05120000
*		DFW COMM. AREA FOR LINE 1	05130000
DC	AL1(0)	TASK ID	05140000
SPACE 1			05150000
*#DF2ID		ID OF TASK 'OWNING' SPACE IN	05160000
*		DFW COMM. AREA FOR LINE 2	05170000
DC	AL1(0)	TASK ID	05180000
SPACE 1			05190000
*\$DFECB		*	05200000
DC	XL3'000000'	ECB FOR DFF TASK	05210000
DC	7XL1'00' *****		05220000
EJECT			05230000
*****	COMMAND PROCESSOR WORK AREA	50 BYTES*****	05240000
SPACE 1			05250000
*\$CPWK			05260000
SPACE 1			05270000
-----	REGISTER SAVE AREAS FOR THE DISPLAY TRANSIENT	-----	05280000
*\$CPIAR		*	05290000
DC	AL2(0)	IAR	05300000
SPACE 1			05310000
*\$CPXR1		*	05320000
DC	AL2(0)	TUB,TNT,OR FSB	05330000
SPACE 1			05340000
*\$CPDTF		*	05350000
DC	AL2(0)	DTF	05360000
SPACE 1			05370000
*\$CPUSE		*	05380000
DC	AL1(0)	PROGRAM USE COUNT SAVE FIELD	05390000

```

SPACE 2 05400000
*----- COMMAND PROCESSOR ECB LIST -----* 05410000
*$CPLST ECB LIST. 05420000
DC AL2($CPQ) PROGRAM INITIATE-POSTED BY TERM 05430000
DC AL2($CPPF9) PF9 - PROGRAM REQUEST 05440000
DC AL2($CPCM) DATA IN TP BUFFER FOR CP 05450000
DC AL2(X'0048') OCC 05460000
DC AL2($CP1ST) STARTUP/WTOR/LIST DELIMITER 05470000
DC XL2'FFFF' LIST DELIMITER 05480000
SPACE 2 05490000
*----- COMMAND PROCESSOR ECB'S -----* 05500000
*$CPQ * 05510000
DC 3XL1'00' PROGRAM INITIATE-POSTED BY TERM 05520000
SPACE 1 05530000
*$CPPF9 * 05540000
DC 3XL1'00' PF9 - PROGRAM REQUEST 05550000
SPACE 1 05560000
*$CPCM * 05570000
DC 3XL1'00' DATA IN TP BUFFER FOR CP 05580000
SPACE 1 05590000
*$CPOCC OCC - IN THE DSM NUCLEUS 05600000
* 05610000
*$CP1ST * 05620000
DC XL3'400000' FIRST TIME SWITCH 05630000
SPACE 1 05640000
*$CPFLG SECOND BYTE OF $CP1ST COMMAND PROCESSOR FLAG BYTE 05650000
*$CPFR BIT0 FREEMAIN TO BE DONE 05660000
*$CPCFR BIT3 FREE NEEDED FOR CONSOLE OCC 05670000
*$CPDPG BIT4 PAGE 1 OF DISPLAY DONE 05680000
*$CPD1S BIT5 DO SECONDARY MENU 05690000
SPACE 1 05700000
*$CPWTO WTO - GETMAINED FOR IN TPBUFF 05710000
*-----* 05720000
SPACE 2 05730000
*$CPPRQ $CP+30 START OF 20 BYTE PRGM REQ AREA 05740000
SPACE 05750000
*$CPCOM $CP+30 START OF COMMANDS COMMO AREA 05760000
*$CPRTC $CP+31 SAVE AREA FOR PLRTC 05770000
*$CPEFL $CP+33 SAVE AREA FOR PLEFFL 05780000
*$CPRCA $CP+35 SAVE AREA FOR PLRECA 05790000
SPACE 05800000
*$CPLMG 2 BYTE MESSAGE LENGTH 05810000
DC XL1'00' ALONG WITH LAST BYTE OF $CP1ST 05820000
SPACE 1 05830000
*$CPMSG $CP+32 * 05840000
DC AL2($CC4IG) MSG AREA: MSG #(NN), THEN TEXT 05850000
SPACE 1 05860000
*#LSTSZ $CPMSG+2 * 05870000
DC AL1(LSTSIZ) # ENTRIES IN FIRST LEVEL LIST 05880000
SPACE 1 05890000
*@XSNT1 #LSTSZ+2 * 05900000
DC AL2(XSNT1) ADDRESS OF TRANSIENT AREA 1 05910000
SPACE 1 05920000
*@XSNT2 @XSNT1+2 * 05930000
DC AL2(XSNT2) ADDRESS OF TRANSIENT AREA 2 05940000
SPACE 1 05950000
DC 11XL1'00' (UNUSED) 05960000
EJECT 05970000
***** ALLOCATION WORK AREA 9 BYTES***** 05980000
*$AMWK 05990000

```

*\$AM				06000000
	DC	9XL1'00'	ALLOCATION WORK AREA	06010000
	SPACE	2		06020000
*\$AMFLG		\$AM	ALLOCATION FLAG BYTE	06030000
*\$AMBSY		BIT0	1--ALLOCATION BUSY	06040000
*\$APEND		BIT1	1--ALLOCATION TASK POST PENDING	06050000
*\$AMDFD		BIT2	NON-RESIDENT DFD ALOC IN PROCESS	06060000
*\$AMA1		BIT3	U/R DEVICE ALLOCATION IN PROCESS	06070000
*\$AMA2		BIT4	REJECT IN PROCESS	06080000
*\$AMPF9		BIT5	CONSOLE TUB IN USE FOR PROGRAM	06090000
	SPACE			06100000
*\$AMUR		\$AMFLG+1	CCP LEVEL AND SPOOLING DEVICES.	06110000
*A1SPLV		BIT0	1=CCP IS IN LEVEL 1.	06120000
*			0=CCP IS IN LEVEL 2.	06130000
*		BIT2	MFCU/M SEC IS SPOOL RDR FOR CCP	06140000
*		BIT3	MFCU/M SEC IS SPOOL PCH FOR CCP	06150000
*A1741S		BIT4	3741 IS SPOOLED READER FOR CCP	06160000
*		BIT6	MFCU/M PRI IS SPOOL RDR FOR CCP	06170000
*		BIT7	MFCU/M PRI IS SPOOL PCH FOR CCP	06180000
*\$AMPA		\$AMUR+1	PERM ALOC AND SPOOLED UR	06190000
*			DEVICES FOR CCP.	06200000
*PCTPRS		BIT0	1=WILL SHARE PRINTER.	06210000
*PCT41I		BIT1	1=3741 IS PERM ALLOCATED	06220000
*PCT501		BIT2	1=2501 IS PERM ALLOCATED.	06230000
*PCTPRT		BIT3	1=PRINTER IS PERM ALLOCATED.	06240000
*PCT142		BIT4	1=1442 IS PERM ALLOCATED.	06250000
*PCTMFU		BIT5	1=MFCU/M IS PERM ALLOCATED.	06260000
*		BIT6	1=1442 IS SPOOL READER FOR CCP	06270000
*		BIT7	1=1442 IS SPOOL PUNCH FOR CCP	06280000
	SPACE			06290000
*\$AMSA		\$AMPA+1	IN USE DEVICES	06300000
*\$AMSHR		\$AMSA+1	COUNT OF PRESENT PRINT SHARERS	06310000
	EJECT			06320000
*****	TERMINATOR WORK AREA		24 BYTES*****	06330000
*\$TMWK			START OF TERMINATION WORK AREA	06340000
*\$TM			BEGINNING OF WORK AREA	06350000
	DC	24XL1'00'	TERMINATOR WORK AREA	06360000
	SPACE	2		06370000
*\$TMFLG		\$TM	TERMINATOR FLAG BYTE	06380000
*\$TMSTK		BIT0	SYSTEM TASK FAILURE	06390000
*\$TMBSY		BIT0	1--TERMINATION TASK IS ACTIVE	06400000
*			0--TERMINATION TASK NOT BUSY	06410000
*\$TMDMP		BIT1	1--DUMP HAS BEEN TAKEN	06420000
*			0--DUMP NOT TAKEN	06430000
*\$TMEJ		BIT2	1--\$CC4EJ (CCP END OF JOB)	06440000
*			HAS BEEN INVOKED BY \$CC4TD	06450000
*			(TERMINATION CONTROL ROUTINE)	06460000
*\$TMCLZ		BIT3	TERMINATION MUST CALL CLOSE	06470000
*\$TMCM		BIT4	RESERVED	06480000
*\$TMCP		BIT5	RESERVED	06490000
*\$TMDFL		BIT6	ON=NO MORE DUMP SPACE GUARANTEED	06500000
*\$TMDER		BIT7	ON=PERMANENT DISK I/O ERROR	06510000
*			DURING TERMINATION DUMP	06520000
	SPACE	1		06530000
*\$TMTCB		\$TMFLG+2	@ OF TCB CURRENTLY BEING	06540000
*			PROCESSED BY TERMINATION TASK	06550000
*\$TMECB		\$TMTCB+1	TERMINATION'S ECB (3 BYTES)	06560000
*\$TMDID		\$TMECB-1+3+2	DUMP ID DIGITS (2 BYTES)	06570000
	SPACE	2		06580000
-----	TD WORK AREAS	-----		06590000

*TDCB@	\$TMDID+2	TCB ADDRESS	06600000
*TDASAU	TDCB@+2	ATR SAVE	06610000
*TDESAU	TDASAU+2	END LENGTH	06620000
*TDQSAV	TDESAU+2	XR1	06630000
*TDXSAV	TDQSAV+2	ATR SAVE	06640000
*TDCON9	TDXSAV+2	CONSTANT 9	06650000
EJECT			06660000
*****	COMMUNICATIONS MANAGEMENT	WORK AREA 45 BYTES*****	06670000
*\$CMWK		START OF 'CM' WORK AREA	06680000
*\$CM		START OF 'CM' WORK AREA	06690000
DC	18XL1'00'	COMMUNICATION TASK WORK AREA	06700000
SPACE	2		06710000
*#CMDTF	\$CM+1	SAVE AREA FOR DTF ADDRESS	06720000
*#CMPL	#CMDTF+2	SAVE AREA FOR PARM LIST	06730000
SPACE			06740000
*#CMTMA	#CMPL+2	SAVE AREA FOR TERMINAL ADDRESS	06750000
*#CMBK1	#CMTMA	RETURN CODE FOR \$CC4BL MESSAGE	06760000
SPACE			06770000
*#CMTFT	#CMTMA+1	SAVE AREA FOR TERMINAL FEATURES	06780000
*#CMBK2	#CMTFT	INTERNAL MSG TYPE FOR \$CC4BL	06790000
SPACE			06800000
*#CMPTR	#CMTFT+2	SAVE AREA FOR PREVIOUS POINTER	06810000
*#CMFPL	#CMPTR+2	SAVE AREA FOR FOUND PARM LIST	06820000
*#CMPTX	#CMFPL+2	SPECIAL SAVE AREA FOR PREV PTR	06830000
*#CMTUB	#CMPTX+2	SAVE AREA FOR TUB ADDRESS	06840000
*#CMERP	#CMTUB+2	@ OF DTF TO BE RESCHEDULED AFTER	06850000
*		BEING IN CCP ERP	06860000
*#CMSWT	#CMERP+1	CONTROL SWITCH	06870000
*#CMARR	BIT0	1--OK TO MODIFY TCBARR	06880000
SPACE	2		06890000
*#OPEND	#CMSWT+1	*	06900000
DC	XL1'00'	T-P OP END COUNT	06910000
SPACE	1		06920000
*#CCMCL	#OPEND+2	*	06930000
DC	XL2'0000'	MAXIMUM COMMAND BUFFER LENGTH	06940000
SPACE	1		06950000
*#RUFCL	#CCMCL+2	RUF COMMAND LENGTH	06960000
DC	AL2(0)	FILLED BY STARTUP	06970000
SPACE	1		06980000
*@TAS	#RUFCL+2	TERMINAL ATTRIBUTE SET ADDRESS	06990000
DC	AL2(0)	FILLED BY STARTUP	07000000
SPACE	1		07010000
*#HITAS	@TAS+1	HIGHEST TAS INDEX IN SYSTEM	07020000
DC	AL1(0)	FILLED BY STARTUP	07030000
SPACE	1		07040000
*@CSSTT	#HITAS+2	DISK C/S OF STT	07050000
DC	AL2(0)	FILLED BY STARTUP	07060000
SPACE	1		07070000
*#HISTT	@CSSTT+1	NO ENTRIES IN STT(SW TERM TBL)	07080000
DC	AL1(0)	FILLED BY STARTUP	07090000
SPACE	1		07100000
*@MLTAD	#HISTT+2	ADDRESS OF MLTA ADAPTER	07110000
DC	AL2(0)	0 INDICATES NO MLTA ADAPTER	07120000
*		FILLED BY STARTUP	07130000
SPACE	1		07140000
*@CKLST	@MLTAD+2	ADDRESS OF CHECK LIST	07150000
DC	AL2(0)	FILLED BY STARTUP	07160000
SPACE	1		07170000
-----	COMMUNICATIONS MANAGEMENT	ECB'S -----	07180000
*\$CMECB	@CKLST+1	*	07190000

DC	XL3'000000'	GENERAL POST ECB FOR CM	07200000
SPACE	1		07210000
*\$CMFM	\$CMECB+3	*	07220000
DC	XL3'200000'	FREEMAIN'S POST OF CM	07230000
*@ANYTP	\$CMFM+4	ADDRESS OF COMMON TP BUFFER AREA	07240000
DC	AL2(0)	* FILLED BY STARTUP	07250000
*@INVPL	@ANYTP+2	ADDRESS OF INVITE PL AREA	07260000
DC	AL2(0)	* FILLED BY STARTUP	07270000
SPACE	1		07280000
DC	XL2'0000'	END OF ECB LIST	07290000
EJECT			07300000
-----	CONSOLE TUB AND PARAMETER LIST -----		07310000
*KNTUB	\$END1+#LWKWK	FIXED LOCATION OF CONSOLE TUB	07320000
DC	CL2'KN'	CONSTANT	07330000
DC	AL2(KNPL)	PARAM LIST ADDRESS	07340000
DC	XL7'0000000000000000'	UNUSED BYTES IN TUB	07350000
DC	XL1'80'	SIGNIFIES CONSOLE TUB	07360000
DC	26XL1'00'	MORE UNINITIALIZED BYTES	07370000
*KNPL	KNTUB+38	CONSOLE PARAMETER LIST	07380000
DC	XL2'0000'	PLRTC FILED OF KN PL	07390000
DC	XL2'8007'	SYS REQUEST SO NOT FREEMAINED	07400000
DC	15XL1'00'	REST OF PARM LIST	07410000
EJECT			07420000
-----	SHUTDOWN FIELDS -----		07430000
*SHDECB	KNPL+19	SHUTDOWN'S ECB	07440000
DC	XL3'000000'	SHUTDOWN'S ECB	07450000
SPACE	1		07460000
DC	XL1'00'	RESERVED	07470000
*FEHLT@	SHDECB+2+3		07480000
DC	AL2(XSNTHT)	ADDRESS OF FE HALT ROUTINE	07490000
SPACE	1		07500000
DC	11XL1'00'	SAVE AREA	07510000
*SHDSAV	FEHLT@+11	CM TCB FIELDS SAVE AREA	07520000
SPACE	2		07530000
-----	EXTENTION TO \$CC4CM WORK AREA -----		07540000
*#CMTRL	SHDSAV-10+11	*	07550000
*#CMMVL	#CMTRL	MOVE LIST FOR \$CC4CM	07560000
DC	11XL1'00'	TRANSLATE LIST FOR \$CC4CM	07570000
*PLTIME	#CMMVL+10+4	POLL TIME FOR CM	07580000
DC	XL4'00'		07590000
*WATIME	PLTIME+3	WAIT TIME FOR CM	07600000
DC	XL3'00'		07610000
*SAVLOP	WATIME+1	SAVE AREA FOR POLL LOOP COUNT	07620000
DC	XL1'00'		07630000
*@CCPTB	SAVLOP+2	CCP PARTITION TCB ADDRESS	07640000
DC	AL2(0)	ADDRESS IS FILLED BY START-UP	07650000
*#TPPUT	@CCPTB+2	LENGTH OF PUT AREA IN TPBUFFER	07660000
DC	AL2(0)	ADDRESS IS FILLED BY START-UP	07670000
*#TPANY	#TPPUT+2	LENGTH OF MIDDLE AREA IN TPBUFF	07680000
DC	AL2(0)	LENGTH FILLED BY START-UP	07690000
*#ANYS	#TPANY+1	LENGTH OF LARGEST COMMON AREA	07700000
DC	AL2(0)	INIT. LENGTH FILLED BY START-UP	07710000
*CORCNT	#ANYS+2	COUNT OF NO. PARMLISTS WAITING	07720000
*		ON CORE FROM TPBUFFER	07730000
DC	XL1'00'	INITIALIZED TO ZERO	07740000
EJECT			07750000
MEND			07760000

```

MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $E033 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 3, MODIFICATION LEVEL 0 OF 5704-SC1 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- DEFINE 00100000
.* COMMUNICATIONS POINTER TABLES FOR STARTUP BUILD OF 00110000
.* COMMUNICATIONS IOCS MODULES ENTRY POINTS. 00120000
.* 00130000
.* INPUT OPERANDS: 00140000
.* 00150000
.* . MD1-##### ,MD2-##### ,MD3-#### ,BF-##### 00160000
.* 00170000
.* SPECIFIES THE MLTA DEVICES SUPPORTED -- EACH # REPRESENTS A 00180000
.* 1 (SUPPORTED) OR A 0 (NOT SUPPORTED) 00190000
.* 00200000
.* THE DEVICES ARE, RESPECTIVELY: 00210000
.* 00220000
.* IN 'MD1': 1050, 1050D, 2740, 2740S, 2740C, 2740SC, 2740D, 00230000
.* 2740DT 00240000
.* 00250000
.* IN 'MD2' 2740DC, 2740DTC, 2740M2S, 2740M2SB, 2740M2SC, 00260000
.* 2740M2SCB, 2741, 2741D 00270000
.* 00280000
.* IN 'MD3': SYS7C, SYS7SC, SYS7DC, CMCSTD 00290000
.* 00300000
.* IN 'BF' BSCA FEATURES, GETMSG, ITB, RECSEP, RESPOL, AUTORS, 00310000
.* EBCDIC, ASCII, XPRNCY 00320000
.* 00330000
***** 00340000
$E033 &MD1-00000000,&MD2-00000000,&MD3-0000,&BF-00000000 00350000
.* 00360000
GBLB &MLTA . 1=MLTA PRESENT 00370000
GBLB &BSCA . 1=BSCA PRESENT 00380000
GBLB &NSWL . 1=NO SWITCHED LINE SUPPORT 00390000
GBLB &DFE . 1=DFE IS SUPPORTED 00400000
.* 00410000
LCLC &# . USED FOR SINGLE BLANK 00420000
.* 00430000
TEXT 00440000
.* 00450000
.*----- GENERATE THE TABLE OF GENERAL POINTERS -----* 00460000
.* 00470000
&# SETC ' ' . SINGLE BLANK 00480000
.* 00490000
TITLE 'COMMUNICATIONS&#.INITIALIZATION&#.POINTERS' 00500000
***** 00510000
* COMMUNICATIONS POINTERS * 00520000
***** 00530000
SPACE 2 00540000
* THE FOLLOWING TABLE CONTAINS ADDRESSES WHICH MUST BE SET INTO 00550000
* DTF'S DURING STARTUP. THE TABLE CONTAINS NON-ZERO VALUES IF: 00560000
.* 00570000
* . DFE IS SUPPORTED 00580000
SPACE 00590000

```

*	. MLTA IS SUPPORTED		00600000
	SPACE		00610000
*	. BSCA IS SUPPORTED WITH RESIDENT POLLING		00620000
*	. BSCA IS SUPPORTED WITH AUTO-RESPONSE		00630000
	SPACE 1		00640000
*****	NOTE - DO NOT CHANGE ORDER OF THESE EXTRNS WITHOUT CHANGING	*****	00642000
*****	MODULE \$CC4V2.	*****	00644000
	SPACE 1		00646000
	EXTRN \$CC4DF(3)	DFE-YES	00650000
	EXTRN \$CC4#2(3)	COMMUNICATIONS MANAGEMENT	00660000
	EXTRN \$CC4FR(3)	TP REQUEST PARM LIST FREEMAIN	00670000
	EXTRN ADDRS2(3)	MAINTENANCE AREA TABLE ADDRESS	00675000
	EXTRN \$CC4IB(3)	BSCA INTERRUPT HANDLER EXIT	00680000
	EXTRN \$CC4BT(3)	BSCA	00690000
	EXTRN \$\$BSMS(3)	BSCA	00700000
	EXTRN \$CC4M1(3)	BSCA - 3270 ONLY VERSION	00705000
	EXTRN \$\$BSAT(3)	TWO BSCA LINES	00710000
	EXTRN \$\$BSMA(3)	RESPOL-YES	00720000
	EXTRN \$\$BSMB(3)	RESPOL-YES	00730000
	EXTRN \$\$BSMC(3)	RESPOL-YES	00740000
	EXTRN \$\$BSMD(3)	AUTORS-YES	00750000
	EXTRN \$\$BSMF(3)	RESPOL-YES	00760000
	EXTRN \$\$BSLG(3)	RESPOL-YES	00770000
	EXTRN CMCLOZ(3)	ADDR FOR C/S OF CLOSE,OPEN	00780000
	EXTRN MSBSSA(3)	ADDR OF ENTRY POINT FOR TRACE	00785000
	EXTRN \$CC4IM(3)	MLTA INTERRUPT HANDLER EXIT	00790000
	EXTRN \$\$MLRR(3)	MLTA-YES	00800000
	EXTRN \$\$MLDI(3)	MLTA-YES	00810000
	EXTRN \$\$MLIA(3)	MLTA-YES	00820000
	EXTRN \$\$MLIB(3)	CS-YES	00830000
	EXTRN \$\$MLIC(3)	CS-YES	00840000
	EXTRN \$\$MLID(3)	CHECKING TERMINALS	00850000
	EXTRN \$\$MLIF(3)	SW TERMINALS	00860000
	EXTRN \$\$MLIG(3)	CHECKING TERMINALS	00870000
	EXTRN MLTOP1(3)	MLTA OPEN	00880000
	EXTRN MLTSP0(3)	MLTA COMMON	00890000
	EXTRN MLTIO1(3)	MLTA IOCS	00900000
	EXTRN \$\$MLTB(3)	1050D	00910000
	EXTRN \$\$MLTC(3)	2740/2740S/2740M2S(B)	00920000
	EXTRN \$\$MLTD(3)	2740	00930000
	EXTRN \$\$MLTF(3)	2740D	00940000
	EXTRN \$\$MLTG(3)	2740C/SYS7C	00950000
	EXTRN \$\$MLTH(3)	2740DC/SYS7DC	00960000
	EXTRN \$\$MLTI(3)	2740DT	00970000
	EXTRN \$\$MLTL(3)	2740DTC	00980000
	EXTRN \$\$MLTM(3)	2740(M2)SC(B)/SYS7SC	00990000
	EXTRN \$\$MLTO(3)	2741	01000000
	EXTRN \$\$MLTP(3)	2741D/CMSTD	01010000
	EXTRN \$CC4V2(3)	MAINTENANCE AREA MODULE	01015000
	EJECT		01020000
	AIF (&DFE EQ '0').S0001	SKIP IF NO DFE SUPPORT.	01030000
.	*		01040000
	DC AL2(\$CC4DF)	DFE ENTRY POINT ADDRESS.	01050000
	SPACE		01060000
.	*		01070000
	AGO .S0002		01080000
.	*		01090000
.S0001	ANOP		01100000
.	*		01110000
	DC XL2'00'	NO DFE SUPPORT.	01120000

```

.*
.S0002 ANOP          . CONTINUE HERE.          01130000
*   BSCA ENTRY POINTS          01140000
    SPACE          01150000
    DC    AL2($CC4IB)          ADDRESS OF $CC4IB IN LOAD MOD 01160000
    SPACE 1          01170000
.*          01180000
    AIF   ('&BF'(5,1) EQ '0').S0005 SKIP IF NO AUTORS 01190000
.*          01200000
    DC    AL2($$BSMD)          ADDRESS OF  $$BSMD          01210000
    SPACE 1          01220000
.*          01230000
    AGO   .S0006          01240000
.*          01250000
.S0005 ANOP          01260000
.*          01270000
    DC    XL2'00'          IF NO AUTORS          01280000
.*          01290000
.S0006 ANOP          CONTINUE HERE.          01300000
.*          01310000
    AIF   ('&BF'(4,1) EQ '0').S0007 SKIP IF NO RESPOL 01320000
.*          01330000
    DC    AL2($$BSMA)          ADDRESS OF $$BSMA IN LOAD MODULE 01340000
    SPACE 1          01350000
    DC    AL2($$BSMC)          ADDRESS OF $$BSMC          01360000
    AGO   .S0008          01370000
.*          01380000
.S0007 ANOP          01390000
.*          01400000
    DC    XL4'00'          IF NO RESPOL          01410000
.*          01420000
.S0008 ANOP          CONTINUE HERE.          01430000
    SPACE 1          01440000
    DC    AL2(CMCLOZ)          ADDRESS OF WHERE STARTUP MUST 01470000
*   PUT C/S FOR OPEN/CLOSE XIENTS 01480000
    SPACE 1          01490000
    DC    AL2(MSBSSA)          ADDRESS OF BSCA TRACE EXTRN 01500000
    SPACE 1          01503000
    DC    AL2($CC4FR)          01506000
    SPACE 1          01510000
    DC    AL2(ADDRS2)          ADDRESS OF PTF TABLE FOR #2 01512000
    DC    AL2($CC4#2)          END ADDRESS OF DFF CODE      01514000
    AIF   (&MLTA NE '1').MTV99 . SKIP IF *NO* MLTA 01516000
    SPACE 1          01520000
*   MLTA ENTRY POINTS          01530000
    SPACE 1          01540000
    DC    AL2(MLTOP1)          ADDRESS OF MLTOP1 IN LOAD MODULE 01550000
    SPACE 1          01560000
    DC    AL2($$MLRR)          ADDRESS OF $$MLRR IN LOAD MODULE 01570000
    SPACE 1          01580000
    DC    AL2(MLTSP0)          ADDRESS OF AREA IN LOAD MODULE 01590000
    SPACE 1          01600000
    DC    AL2(MLTIO1)          ADDRESS OF MLTA IOCS ENTRY POINT 01610000
.*          01620000
    AIF   ('&MD1'(1,1) NE '1').MTV02 . SKIP IF NO TYPE-1050 01630000
.*          01640000
    SPACE 1          01650000
    DC    CL1'A'          FOR $$MLTM--1050          01660000
    DC    AL2($$MLTM)          ADDRESS OF TRANSFER VECTOR 01670000
.*          01680000
          01690000

```

.MTV02	ANOP		. CONTINUE HERE	01700000
	AIF	('&MD1'(2,1) NE '1').MTV03	. SKIP IF NO TYPE-1050D	01710000
.*				01720000
	SPACE	1		01730000
	DC	CL1'B'	FOR \$\$MLTB--1050D	01740000
	DC	AL2(\$\$MLTB)	ADDRESS OF TRANSFER VECTOR	01750000
.*				01760000
.MTV03	ANOP		. CONTINUE HERE	01770000
	AIF	('&MD1'(4,1) EQ '1').MTVX3	. SKIP IF TYPE-2740S	01780000
	AIF	('&MD2'(3,2) EQ '00').MTV04	. SKIP IF NEITHER 2740M2S/SB	01790000
.*				01800000
.MTVX3	ANOP		. 2740S/2740M2S/2740M2SB	01810000
	SPACE	1		01820000
	DC	CL1'C'	FOR \$\$MLTC--2740/2740M2S(B)	01830000
	DC	AL2(\$\$MLTC)	ADDRESS OF TRANSFER VECTOR	01840000
.*				01850000
.MTV04	ANOP		. CONTINUE HERE	01860000
	AIF	('&MD1'(3,1) NE '1').MTV05	. SKIP IF NO TYPE-2740	01870000
.*				01880000
	SPACE	1		01890000
	DC	CL1'D'	FOR \$\$MLTD--2740	01900000
	DC	AL2(\$\$MLTD)	ADDRESS OF TRANSFER VECTOR	01910000
.*				01920000
.MTV05	ANOP		. CONTINUE HERE	01930000
	AIF	('&MD1'(7,1) NE '1').MTV06	. SKIP IF NO TYPE-2740D	01940000
.*				01950000
	SPACE	1		01960000
	DC	CL1'F'	FOR \$\$MLTF--2740D	01970000
	DC	AL2(\$\$MLTF)	ADDRESS OF TRANSFER VECTOR	01980000
.*				01990000
.MTV06	ANOP		. CONTINUE HERE	02000000
	AIF	('&MD1'(5,1) EQ '1').MTVX6	. SKIP IF TYPE-2740C	02010000
	AIF	('&MD3'(1,1) NE '1').MTV07	. SKIP UNLESS TYPE-SYS7C	02020000
.*				02030000
.MTVX6	ANOP		. TYPE-2740C/SYS7C	02040000
	SPACE	1		02050000
	DC	CL1'G'	FOR \$\$MLTG--2740C/SYS7C	02060000
	DC	AL2(\$\$MLTG)	ADDRESS OF TRANSFER VECTOR	02070000
.*				02080000
.MTV07	ANOP		. CONTINUE HERE	02090000
	AIF	('&MD2'(1,1) EQ '1').MTVX7	. SKIP IF TYPE-2740DC	02100000
	AIF	('&MD3'(3,1) NE '1').MTV08	. SKIP UNLESS TYPE-SYS7DC	02110000
.*				02120000
.MTVX7	ANOP		. TYPE-2740DC/SYS7DC	02130000
	SPACE	1		02140000
	DC	CL1'H'	FOR \$\$MLTH--2740DC/SYS7DC	02150000
	DC	AL2(\$\$MLTH)	ADDRESS OF TRANSFER VECTOR	02160000
.*				02170000
.MTV08	ANOP		. CONTINUE HERE	02180000
	AIF	('&MD1'(8,1) NE '1').MTV09	. SKIP UNLESS TYPE-2740DT	02190000
.*				02200000
	SPACE	1		02210000
	DC	CL1'I'	FOR \$\$MLTI--2740DT	02220000
	DC	AL2(\$\$MLTI)	ADDRESS OF TRANSFER VECTOR	02230000
.*				02240000
.MTV09	ANOP		. CONTINUE HERE	02250000
	AIF	('&MD2'(2,1) NE '1').MTV10	. SKIP UNLESS TYPE-2740DTC	02260000
.*				02270000
	SPACE	1		02280000
	DC	CL1'L'	FOR \$\$MLTL--2740DTC	02290000

```

      DC      AL2($$MLTL)          ADDRESS OF TRANSFER VECTOR      02300000
.*
.*
.MTV10 ANOP          . CONTINUE HERE      02310000
      AIF      ('&MD1'(6,1) EQ '1').MTVXA . SKIP IF TYPE-2740SC      02320000
      AIF      ('&MD2'(5,2) NE '00').MTVXA . SKIP IF TYPE-2740M2SC/SCB 02330000
      AIF      ('&MD3'(2,1) NE '1').MTV11 . SKIP UNLESS TYPE-SYS7C    02340000
.*
.*
.MTVXA ANOP          . TYPE-2740SC/2740M2SC(B)/SYS7SC 02350000
      SPACE 1
      DC      CL1'M'              FOR $$MLTM--2740(M2)SC(B)/SYS7SC 02360000
.*
.*
      AIF      ('&MD1'(1,1) EQ '1').MTVXX . SKIP IF ALREADY EXTRN'D 02370000
.*
.*
.*
.MTVXX ANOP          . CONTINUE HERE      02380000
      DC      AL2($$MLTM)        ADDRESS OF TRANSFER VECTOR      02390000
.*
.*
.MTV11 ANOP          . CONTINUE HERE      02400000
      AIF      ('&MD2'(7,1) NE '1').MTV12 . SKIP UNLESS TYPE-2741    02410000
.*
      SPACE 1
      DC      CL1'O'              FOR $$MLTO--2741                    02420000
      DC      AL2($$MLTO)        ADDRESS OF TRANSFER VECTOR      02430000
.*
.*
.MTV12 ANOP          . CONTINUE HERE      02440000
      AIF      ('&MD2'(8,1) EQ '1').MTVXC . SKIP IF TYPE-2741D      02450000
      AIF      ('&MD3'(4,1) NE '1').MTV99 . SKIP UNLESS TYPE-CMCSTD 02460000
.*
.*
.MTVXC ANOP          . CONTINUE HERE      02470000
      SPACE 1
      DC      CL1'P'              FOR $$MLTP--2741D/CMCSTD          02480000
      DC      AL2($$MLTP)        ADDRESS OF TRANSFER VECTOR      02490000
.*
.*
.MTV99 ANOP          . END INCLUDE TRANSFER VECTORS 02500000
      SPACE 1
      DC      XL1'00'            *** TABLE STOPPER ***            02510000
      SPACE 1
.*
      MEND
      02520000
      02530000
      02540000
      02550000
      02560000
      02570000
      02580000
      02590000
      02600000
      02610000
      02620000
      02630000
      02640000
      02650000
      02660000
      02670000
      02680000

```

```

MACRO 00010000
***** 00020000
* 00030000
* $E038 - GENERATE TRANSIENT IDENTIFICATION EQUATES TO BE PASSED * 00040000
* TO THE TRANSIENT HANDLER ($CC4PI). * 00050000
* 00060000
* INPUT OPERANDS - * 00070000
* CON - 1 GENERATE CONSTANTS * 00080000
* 0 DO NOT GENERATE CONSTANTS * 00090000
* MXL - ##### * 00100000
* A STRING OF 14 BITS (0/1) WHICH INDICATES * 00110000
* WHICH MLTA TRANSLATE TRANSIENTS ARE RE- * 00120000
* QUIRED. FOR EXAMPLE, BIT 1 SAYS $CC4J1 * 00130000
* IS NEEDED, BIT 2 SAYS $CC4J2, ETC. * 00140000
* 00150000
***** 00160000
$E038 &CON-0,&MXL- 00170000
GBLB &DFB 00180000
GBLB &NAS BSCA ASCII SUPPORT 00190000
GBLB &BSCA 00200000
GBLB &N41 3741 AS A TERMINAL 00205000
GBLB &MLTA 00210000
GBLB &MIN 00220000
GBLB &NPBY 0=BUSY PRINT 1=NO SUPPORT 00225000
GBLB &DME 00230000
GBLC &# 00240000
LCLA &N RELATIVE ENTRY NUMBER 00250000
LCLA &LSIZE COUNT OF ENTRIES. 00260000
LCLB &C CONSTANTS SWITCH 00270000
TEXT 00280000
&LSIZE SETA 0 RESET COUNT 00290000
&# SETC ' ' 00300000
* BEGIN MACRO '$E038' 3/14/73 00310000
&C SETB 1 NO CONSTANTS WANTED 00320000
AIF (&CON EQ '0').NCON SET TO NO CONSTANTS GENERATED 00330000
&C SETB 0 CONSTANTS WANTED 00340000
TA@PGM EQU * START OF PGM LIST 00350000
SPACE 1 00360000
.NCON ANOP 00370000
*-----FIXED PORTION OF TRANSIENT LIST-----* 00380000
SPACE 2 00390000
CC4AB EQU 12 ACCEPT INPUT POST PRE-PROCESSOR 00400000
&N SETA 15 00410000
&LSIZE SETA &LSIZE+1 UPDATE COUNT. 00420000
AIF (&C).NO1 00430000
DC XL1'00' 00440000
DC CL2'AB' ACCEPT INPUT POST PRE-PROCESSOR 00450000
.NO1 ANOP 00460000
CC4A2 EQU &N ALLOCATION CONTROL 00470000
&N SETA &N+3 00480000
&LSIZE SETA &LSIZE+1 UPDATE COUNT. 00490000
AIF (&C).NO2 00500000
DC XL1'00' 00510000
DC CL2'A2' ALLOCATION CONTROL 00520000
.NO2 ANOP 00530000
CC4GA EQU &N GET ATTRIBUTES 00540000
&N SETA &N+3 00550000
&LSIZE SETA &LSIZE+1 UPDATE COUNT. 00560000
AIF (&C).NO3 00570000

```

	DC	XL1'00'		00580000
	DC	CL2'GA'	GET ATTRIBUTES	00590000
.NO3	ANOP			00600000
CC4OP	EQU	&N	USER FILE OPEN	00610000
&N	SETA	&N+3		00620000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	00630000
	AIF	(&C).NO10		00640000
	DC	XL1'00'		00650000
	DC	CL2'OP'	USER FILE OPEN	00660000
.NO10	ANOP			00670000
CC4R1	EQU	&N	COMMAND PROCESSOR CONTROL	00680000
&N	SETA	&N+3		00690000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	00700000
	AIF	(&C).NO11		00710000
	DC	XL1'00'		00720000
	DC	CL2'R1'	COMMAND PROCESSOR CONTROL	00730000
.NO11	ANOP			00740000
CC4PR	EQU	&N	COMMAND PROCESSOR RETURN	00750000
&N	SETA	&N+3		00760000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	00770000
	AIF	(&C).NO12		00780000
	DC	XL1'00'		00790000
	DC	CL2'PR'	COMMAND PROCESSOR RETURN	00800000
.NO12	ANOP			00810000
CC4TD	EQU	&N	TERMINATION CONTROL ROUTINE	00820000
&N	SETA	&N+3		00830000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	00840000
	AIF	(&C).NO14		00850000
	DC	XL1'00'		00860000
	DC	CL2'TD'	TERMINATION CONTROL ROUTINE	00870000
.NO14	ANOP			00880000
CC4WC	EQU	&N	USER T-P SWITCHED LINE CONNECT	00890000
&N	SETA	&N+3		00900000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	00910000
	AIF	(&C).NO16		00920000
	DC	XL1'00'		00930000
	DC	CL2'WC'	USER T-P SWITCHED LINE CONNECT	00940000
.NO16	ANOP			00950000
CC4WR	EQU	&N	T-P TRANSLATION ERROR	00960000
&N	SETA	&N+3		00970000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	00980000
	AIF	(&C).NO17		00990000
	DC	XL1'00'		01000000
	DC	CL2'WR'	T-P TRANSLATION ERROR	01010000
.NO17	ANOP			01020000
CC4MP	EQU	&N	IGNORE PUT TO ERP T-P DEVICE	01030000
&N	SETA	&N+3		01040000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01050000
	AIF	(&C).NOMP		01060000
	DC	XL1'00'		01070000
	DC	CL2'MP'	IGNORE PUT TO ERP T-P DEVICE	01080000
.NOMP	ANOP			01090000
	AIF	(&DFF NE '1').NDFF	SKIP IF NO D.F.F.	01100000
CC4DC	EQU	&N	D.F.F.	01110000
&N	SETA	&N+3		01120000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01130000
	AIF	(&C).NO20		01140000
	DC	XL1'00'		01150000
	DC	CL2'DC'	D.F.F.	01160000
.NO20	ANOP			01170000

CC4DD	EQU	&N	D.F.F.	01180000
&N	SETA	&N+3		01190000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01200000
	AIF	(&C).NO21		01210000
	DC	XL1'00'		01220000
	DC	CL2'DD'	D.F.F.	01230000
.NO21	ANOP			01240000
CC4DB	EQU	&N	D.F.F.	01250000
&N	SETA	&N+3		01260000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01270000
	AIF	(&C).NO22		01280000
	DC	XL1'00'		01290000
	DC	CL2'DB'	D.F.F.	01300000
.NO22	ANOP			01310000
.NDFD	ANOP			01320000
	AIF	(&DME NE '1').NREL	RELEASE COMMAND SUPPORTED?	01330000
CC4NC	EQU	&N	TERMINAL RELEASE COMMAND	01340000
&N	SETA	&N+3		01350000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01360000
	AIF	(&C).NO24		01370000
	DC	XL1'00'		01380000
	DC	CL2'NC'	TERMINAL RELEASE COMMAND	01390000
.NO24	ANOP			01400000
.NREL	ANOP		NO, NO DME CAPABILITY	01410000
CC4DG	EQU	&N	FILE SHARE DEQUEUE	01420000
&N	SETA	&N+3		01430000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01440000
	AIF	(&C).NO245		01450000
	DC	XL1'00'		01460000
	DC	CL2'DG'	FILE SHARE DEQUEUE	01470000
.NO245	ANOP			01480000
CC4GR	EQU	&N	GETMAIN RECOVERY TRANSIENT	01490000
&N	SETA	&N+3		01500000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01510000
	AIF	(&C).NO248		01520000
	DC	XL1'00'		01530000
	DC	CL2'GR'	GETMAIN RECOVERY TRANSIENT	01540000
.NO248	ANOP			01550000
	SPACE	1		01560000
	AIF	(&MLTA NE '1').NMLTA	SKIP IF NO MLTA	01570000
-----		MLTA TRANSIENTS PORTION OF PROGRAM LIST-----*		01580000
	SPACE	1		01590000
	AIF	('&MXL'(01,1) NE '1').NOJ1		01600000
CC4J1	EQU	&N	MLTA TRANSLATE	01610000
&N	SETA	&N+3		01620000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01630000
	AIF	(&C).NOJ1		01640000
	DC	XL1'00'		01650000
	DC	CL2'J1'	MLTA TRANSLATE	01660000
.NOJ1	ANOP			01670000
	AIF	('&MXL'(02,1) NE '1').NOJ2		01680000
CC4J2	EQU	&N	MLTA TRANSLATE	01690000
&N	SETA	&N+3		01700000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01710000
	AIF	(&C).NOJ2		01720000
	DC	XL1'00'		01730000
	DC	CL2'J2'	MLTA TRANSLATE	01740000
.NOJ2	ANOP			01750000
	AIF	('&MXL'(03,1) NE '1').NOJ3		01760000
CC4J3	EQU	&N	MLTA TRANSLATE	01770000

&N	SETA	&N+3		01780000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01790000
	AIF	(&C).NOJ3		01800000
	DC	XL1'00'		01810000
	DC	CL2'J3'	MLTA TRANSLATE	01820000
.NOJ3	ANOP			01830000
	AIF	('&MXL'(04,1) NE '1').NOJ4		01840000
CC4J4	EQU	&N	MLTA TRANSLATE	01850000
&N	SETA	&N+3		01860000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01870000
	AIF	(&C).NOJ4		01880000
	DC	XL1'00'		01890000
	DC	CL2'J4'	MLTA TRANSLATE	01900000
.NOJ4	ANOP			01910000
	AIF	('&MXL'(05,1) NE '1').NOJ5		01920000
CC4J5	EQU	&N	MLTA TRANSLATE	01930000
&N	SETA	&N+3		01940000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	01950000
	AIF	(&C).NOJ5		01960000
	DC	XL1'00'		01970000
	DC	CL2'J5'	MLTA TRANSLATE	01980000
.NOJ5	ANOP			01990000
	AIF	('&MXL'(06,1) NE '1').NOJ6		02000000
CC4J6	EQU	&N	MLTA TRANSLATE	02010000
&N	SETA	&N+3		02020000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02030000
	AIF	(&C).NOJ6		02040000
	DC	XL1'00'		02050000
	DC	CL2'J6'	MLTA TRANSLATE	02060000
.NOJ6	ANOP			02070000
	AIF	('&MXL'(07,1) NE '1').NOJ7		02080000
CC4J7	EQU	&N	MLTA TRANSLATE	02090000
&N	SETA	&N+3		02100000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02110000
	AIF	(&C).NOJ7		02120000
	DC	XL1'00'		02130000
	DC	CL2'J7'	MLTA TRANSLATE	02140000
.NOJ7	ANOP			02150000
	AIF	('&MXL'(08,1) NE '1').NOJ8		02160000
CC4J8	EQU	&N	MLTA TRANSLATE	02170000
&N	SETA	&N+3		02180000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02190000
	AIF	(&C).NOJ8		02200000
	DC	XL1'00'		02210000
	DC	CL2'J8'	MLTA TRANSLATE	02220000
.NOJ8	ANOP			02230000
	AIF	('&MXL'(09,1) NE '1').NOJ9		02240000
CC4J9	EQU	&N	MLTA TRANSLATE	02250000
&N	SETA	&N+3		02260000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02270000
	AIF	(&C).NOJ9		02280000
	DC	XL1'00'		02290000
	DC	CL2'J9'	MLTA TRANSLATE	02300000
.NOJ9	ANOP			02310000
	AIF	('&MXL'(10,1) NE '1').NOJA		02320000
CC4JA	EQU	&N	MLTA TRANSLATE	02330000
&N	SETA	&N+3		02340000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02350000
	AIF	(&C).NOJA		02360000
	DC	XL1'00'		02370000

.NOJA	DC	CL2'JA'	MLTA TRANSLATE	02380000
	ANOP			02390000
	AIF	('&MXL'(11,1) NE '1').NOJB		02400000
CC4JB	EQU	&N	MLTA TRANSLATE	02410000
&N	SETA	&N+3		02420000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02430000
	AIF	(&C).NOJB		02440000
	DC	XL1'00'		02450000
	DC	CL2'JB'	MLTA TRANSLATE	02460000
.NOJB	ANOP			02470000
	AIF	('&MXL'(12,1) NE '1').NOJC		02480000
CC4JC	EQU	&N	MLTA TRANSLATE	02490000
&N	SETA	&N+3		02500000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02510000
	AIF	(&C).NOJC		02520000
	DC	XL1'00'		02530000
	DC	CL2'JC'	MLTA TRANSLATE	02540000
.NOJC	ANOP			02550000
CC4MA	EQU	&N	MLTA 1ST LEVEL ERROR HANDLER	02560000
&N	SETA	&N+3		02570000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02580000
	AIF	(&C).NO25		02590000
	DC	XL1'00'		02600000
	DC	CL2'MA'	MLTA 1ST LEVEL ERROR HANDLER	02610000
.NO25	ANOP			02620000
CC4MC	EQU	&N	MLTA BUFR RECEIVE ERROR	02630000
&N	SETA	&N+3		02640000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02650000
	AIF	(&C).NO26		02660000
	DC	XL1'00'		02670000
	DC	CL2'MC'	MLTA BUFR RECEIVE ERROR	02680000
.NO26	ANOP			02690000
CC4ME	EQU	&N	MLTA CLOSE LINES	02700000
&N	SETA	&N+3		02710000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02720000
	AIF	(&C).NO27		02730000
	DC	XL1'00'		02740000
	DC	CL2'ME'	MLTA CLOSE LINES	02750000
.NO27	ANOP			02760000
CC4MT	EQU	&N	MLTA START OLT	02770000
&N	SETA	&N+3		02780000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02790000
	AIF	(&C).NO28		02800000
	DC	XL1'00'		02810000
	DC	CL2'MT'	MLTA START OLT	02820000
.NO28	ANOP			02830000
CC4PG	EQU	&N	MLTA PURGE I/O	02840000
&N	SETA	&N+3		02850000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02860000
	AIF	(&C).NO29		02870000
	DC	XL1'00'		02880000
	DC	CL2'PG'	MLTA PURGE I/O	02890000
.NO29	ANOP			02900000
CC4SC	EQU	&N	MLTA NON-ZERO START CODE ANALY.	02910000
&N	SETA	&N+3		02920000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	02930000
	AIF	(&C).NO30		02940000
	DC	XL1'00'		02950000
	DC	CL2'SC'	MLTA NON-ZERO START CODE ANALY.	02960000
.NO30	ANOP			02970000

CC4SK	EQU	&N	MLTA POLL SKIP BIT ROUTINE	02980000
&N	SETA	&N+3		02990000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03000000
	AIF	(&C).NO31		03010000
	DC	XL1'00'		03020000
	DC	CL2'SK'	MLTA POLL SKIP BIT ROUTINE	03030000
.NO31	ANOP			03040000
CC4SQ	EQU	&N	MLTA STOP POLLING ABORT OP-END	03050000
&N	SETA	&N+3		03060000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03070000
	AIF	(&C).NO32		03080000
	DC	XL1'00'		03090000
	DC	CL2'SQ'	MLTA STOP POLLING ABORT OP-END	03100000
.NO32	ANOP			03110000
CC4T2	EQU	&N	MLTA OLT OP-END	03120000
&N	SETA	&N+3		03130000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03140000
	AIF	(&C).NO33		03150000
	DC	XL1'00'		03160000
	DC	CL2'T2'	MLTA OLT OP-END	03170000
.NO33	ANOP			03180000
CC4T1	EQU	&N	OLT PARM LIST HANDLER	03190000
&N	SETA	&N+3		03200000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03210000
	AIF	(&C).NO15		03220000
	DC	XL1'00'		03230000
	DC	CL2'T1'	OLT PARM LIST HANDLER	03240000
.NO15	ANOP			03250000
	SPACE	1		03260000
.NMLTA	ANOP			03270000
	AIF	(&BSCA NE '1').NBSCA	SKIP IF NO BSCA	03280000
-----		BSCA PORTION OF PROGRAM LIST-----		03290000
	SPACE	1		03300000
CC4BA	EQU	&N	3270 SAVE STATUS	03310000
&N	SETA	&N+3		03320000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03330000
	AIF	(&C).NO34		03340000
	DC	XL1'00'		03350000
	DC	CL2'BA'	3270 SAVE STATUS	03360000
.NO34	ANOP			03370000
CC4BB	EQU	&N	BSCA OUTPUT RECORD FORMATTING	03380000
&N	SETA	&N+3		03390000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03400000
	AIF	(&C).NO35		03410000
	DC	XL1'00'		03420000
	DC	CL2'BB'	BSCA OUTPUT RECORD FORMATTING	03430000
.NO35	ANOP			03440000
CC4BC	EQU	&N	BSCA STOP POLLING	03450000
&N	SETA	&N+3		03460000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03470000
	AIF	(&C).NO36		03480000
	DC	XL1'00'		03490000
	DC	CL2'BC'	BSCA STOP POLLING	03500000
.NO36	ANOP			03510000
CC4BE	EQU	&N	BSCA ERROR RECOVERY	03520000
&N	SETA	&N+3		03530000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03540000
	AIF	(&C).NO37		03550000
	DC	XL1'00'		03560000
	DC	CL2'BE'	BSCA ERROR RECOVERY	03570000

.NO37	ANOP			03580000
	AIF	(&NPBY).NO375	SKIP IF NO BUSY PRINTER	03581000
CC4BG	EQU	&N	BUSY PRINTER CLEANUP	03582000
&N	SETA	&N+3		03583000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT	03584000
	AIF	(&C).NO375		03585000
	DC	XL1'00'		03586000
	DC	CL2'BG'	BUSY PRINTER CLEANUP	03587000
.NO375	ANOP			03588000
	AIF	(&DME NE '1').NBDM	SKIP IF NO DME IN SYSTEM	03590000
CC4BI	EQU	&N	BSCA DATA MODE ESCAPE	03600000
&N	SETA	&N+3		03610000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03620000
	AIF	(&C).NO38		03630000
	DC	XL1'00'		03640000
	DC	CL2'BI'	BSCA DATA MODE ESCAPE	03650000
.NO38	ANOP			03660000
.NBDM	ANOP			03670000
CC4BP	EQU	&N	BSCA PURGE	03680000
&N	SETA	&N+3		03690000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03700000
	AIF	(&C).NO39		03710000
	DC	XL1'00'		03720000
	DC	CL2'BP'	BSCA PURGE	03730000
.NO39	ANOP			03740000
CC4BQ	EQU	&N	BSCA STOP POLLING QUEUE	03750000
&N	SETA	&N+3		03760000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03770000
	AIF	(&C).NO40		03780000
	DC	XL1'00'		03790000
	DC	CL2'BQ'	BSCA STOP POLLING QUEUE	03800000
.NO40	ANOP			03810000
CC4BR	EQU	&N	BSCA USER T-P REQ VALIDATION	03820000
&N	SETA	&N+3		03830000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03840000
	AIF	(&C).NO41		03850000
	DC	XL1'00'		03860000
	DC	CL2'BR'	BSCA USER T-P REQ VALIDATION	03870000
.NO41	ANOP			03880000
CC4B5	EQU	&N	BSCA 3735 SENSE STATUS	03890000
&N	SETA	&N+3		03900000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03910000
	AIF	(&C).NO42		03920000
	DC	XL1'00'		03930000
	DC	CL2'B5'	BSCA 3735 SENSE STATUS	03940000
.NO42	ANOP			03950000
CC4B0	EQU	&N	BSCA 3270 COMMAND FORMATTING	03960000
&N	SETA	&N+3		03970000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	03980000
	AIF	(&C).NO43		03990000
	DC	XL1'00'		04000000
	DC	CL2'B0'	BSCA 3270 COMMAND FORMATTING	04010000
.NO43	ANOP			04020000
	AIF	(&MIN NE '1').NBSMN	MIN BSCA SUPPORT NEEDED?	04030000
CC4B1	EQU	&N	MIN SYS BSCA # 1	04040000
&N	SETA	&N+3		04050000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	04060000
	AIF	(&C).NO44		04070000
	DC	XL1'00'		04080000
	DC	CL2'B1'	MIN SYS BSCA # 1	04090000

.NO44	ANOP			04100000
CC4B2	EQU	&N	MIN SYS BSCA # 2	04110000
&N	SETA	&N+3		04120000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	04130000
	AIF	(&C).NO45		04140000
	DC	XL1'00'		04150000
	DC	CL2'B2'	MIN SYS BSCA # 2	04160000
.NO45	ANOP			04170000
.NBSMN	ANOP			04180000
	AIF	(&NAS).NASCI	ASCII SUPPORT IN BSCA?	04190000
CC4JD	EQU	&N	BSCA ASCII TRANSLATE	04200000
&N	SETA	&N+3		04210000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	04220000
	AIF	(&C).NO18		04230000
	DC	XL1'00'		04240000
	DC	CL2'JD'	BSCA ASCII TRANSLATE	04250000
.NO18	ANOP			04260000
CC4JE	EQU	&N	BSCA ASCII TRANSLATE	04270000
&N	SETA	&N+3		04280000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	04290000
	AIF	(&C).NO19		04300000
	DC	XL1'00'		04310000
	DC	CL2'JE'	BSCA ASCII TRANSLATE	04320000
.NO19	ANOP			04330000
.NASCI	ANOP			04340000
	AIF	(&N41).NO46	3741 SUPPORT	04341000
CC4B7	EQU	&N	3741 SUPPORT AS A TERMINAL	04342000
&N	SETA	&N+3	UPDATE COUNT	04343000
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT	04344000
	AIF	(&C).NO46		04345000
	DC	XL1'00'		04346000
	DC	CL2'B7'	3741 TERMINAL SUPPORT	04347000
.NO46	ANOP			04348000
CC4S0	EQU	&N	BSCA 3275 COMMAND FORMATTING	04348200
&N	SETA	&N+3		04348400
&LSIZE	SETA	&LSIZE+1	UPDATE COUNT.	04348600
	AIF	(&C).NO47		04348800
	DC	XL1'00'		04349000
	DC	CL2'S0'	BSCA 3275 COMMAND FORMATTING	04349200
.NO47	ANOP			04349400
	SPACE	1		04350000
.NBSCA	ANOP			04360000
.END	ANOP			04370000
-----	DEFINE THE EQUATE FOR THE # OF ENTRIES IN THE LIST-----			04380000
	SPACE	1		04390000
LSTSIZ	EQU	&LSIZE	# OF ENTRIES IN TRANSIENT LIST	04400000
*			END MACRO '\$E038'	04410000
	MEND			04420000

MODULE-\$EDFT , VOLUME ID-R2R2R2, DATE-06/06/10

MACRO			00010000
\$EDFT			00020000
TEXT			00030000
*****			00040000
*			* 00050000
*	EQUATES OF THE TERMINATION CODES USSED BY THE DISPLAY		* 00060000
*	FORMAT CONTROL ROUTINE, \$CC4DF, \$CC4DB, \$CC4DC.		* 00070000
*	THESE EQUATES REFER TO THE TERMINATION CODES CONTAINED IN		* 00080000
*	MACRO 'CPTCC'.		* 00090000
*			* 00100000
*****			00110000
SPACE 2			00120000
TC01	EQU TCC255	TERMINATION CODE	00130000
TC02	EQU TCC254	TERMINATION CODE	00140000
TC03	EQU TCC253	TERMINATION CODE	00150000
TC04	EQU TCC252	TERMINATION CODE	00160000
TC05	EQU TCC251	TERMINATION CODE	00170000
TC06	EQU TCC250	TERMINATION CODE	00180000
TC07	EQU TCC249	TERMINATION CODE	00190000
TC08	EQU TCC248	TERMINATION CODE	00200000
TC09	EQU TCC247	TERMINATION CODE	00210000
TC10	EQU TCC246	TERMINATION CODE	00220000
TC11	EQU TCC245	TERMINATION CODE	00230000
TC12	EQU TCC244	TERMINATION CODE	00240000
TC13	EQU TCC243	TERMINATION CODE	00250000
TC14	EQU TCC242	TERMINATION CODE	00260000
TC15	EQU TCC241	TERMINATION CODE	00270000
TC16	EQU TCC240	TERMINATION CODE	00280000
TC17	EQU TCC239	TERMINATION CODE	00290000
TC18	EQU TCC238	TERMINATION CODE	00300000
MEND			00310000

MODULE-\$EFT , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 2 * 00030000
.* * 00040000
.* NAME:  $EFT * 00050000
.* * 00060000
.* FUNCTION:  DEFINE THE LABELS NECESSARY TO REFERENCE THE * 00070000
.* 'FORMAT TABLE' IN PROGRAM APPENDED STORAGE. * 00080000
.* * 00090000
.* INPUT OPERANDS:  NONE * 00100000
***** 00110000
$EFT 00120000
TEXT 00130000
* * * * * 00140000
* * * * * 00150000
* EQUATES FOR ITEMS IN 'FORMAT TABLE' FOUND IN PAS (PROGRAM * 00160000
* APPENDED STORAGE) FOR DFF. * 00170000
* 02/07/73 * 00180000
* * * * * 00190000
SPACE 00200000
FT EQU 0 START OF FORMAT TABLE 00210000
* LNG CONTENTS 00220000
SPACE 00230000
FTCHN@ EQU FT+1 CHAIN @ TO NEXT FT ENTRY 00240000
* 2 00250000
SPACE 00260000
FTNAME EQU FTCHN@+6 NAME OF DISPLAY FORMAT 00270000
* 6 00280000
SPACE 00290000
FTFCS EQU FTNAME+2 ACTUAL DISK C/S OF FDT 00300000
* 2 00310000
SPACE 00320000
FTFDTL EQU FTFCS+1 # SECTORS TO READ FOR FDT 00330000
* 1 00340000
SPACE 00350000
FTTSCS EQU FTFDTL+2 DISK C/S OF TEXT STREAM FOR THIS 00360000
* 2 FORMAT. 00370000
SPACE 00380000
FTTSL EQU FTTSCS+1 # SECTORS OF TEXT STREAM FOR 00390000
* 1 OUTPUT 00400000
FTM#2 EQU BIT0 1-MODEL 2 00410000
* 0-MODEL 1 00420000
SPACE 00430000
FTOUTL EQU FTTSL+2 LENGTH IN BYTES OF OUTPUT TEXT 00440000
* 2 STREAM FOR OUTPUT. 00450000
SPACE 00460000
FTINL EQU FTOUTL+2 LENGTH OF DATA TEXT TO EXPECT 00470000
* 2 FOR A FULL INPUT BUFFER FOR 00480000
* HOLD AREA. 00490000
SPACE 00500000
FTL EQU FTINL-FT+1 LENGTH OF ONE FT ENTRY. 00510000
EJECT 00520000
MEND 00530000
```



```

MODULE-$ETT , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$ETT                                00020000
.*                                  00030000
.*                                  00040000
.*      MACRO EQUATES FOR OFFSETS TO ITEMS IN THE 00050000
.*      'TERMINAL TABLE' FOUND IN PAS.          00060000
.*      12/11/72                                00070000
.*                                  00080000
TEXT                                  00090000
* * * * *                               * 00100000
*                                  * 00110000
*      EQUATES TO ITEMS IN 'TERMINAL TABLE' IN 'PAS'. 00120000
*      12/11-72                                * 00130000
* * * * *                               * 00140000
SPACE                                  00150000
TT   EQU 0          START OF TERMINAL TABLE 00160000
*                                  * 00170000
*      LNG CONTENTS                            00180000
SPACE                                  00190000
TTCHN@ EQU TT+1    @ OF NEXT ENTRY. (CONTAINS 00200000
*                                  * 2 ZERO IN HIGH-ORDER BYTE IF LAST)
SPACE                                  00210000
TTNAME EQU TTCHN@+6 NAME OF TERMINAL        00220000
*                                  * 6
SPACE                                  00230000
TTFT@ EQU TTNAME+2 @ OF MAP TABLE ENTRY CURRENTLY 00240000
*                                  * 2 USED BY THIS TERMINAL
SPACE                                  00250000
TTIFT EQU TTFT@+25 INPUT FORMT TABLE. ONE BIT FOR 00260000
*                                  * 25 EACH INPUT, OUTPUT/IINPUT,
*                                  * DETECTABLE FIELD. FIRST FIELD
*                                  * IS RIGHT-MOST OR LOW ORDER BIT.
*                                  * CAPACITY IS 200 FIELDS.
SPACE                                  00270000
TTIFTL EQU 25      LENGTH OF IFT.            00280000
SPACE                                  00290000
TTINLH EQU TTIFT+2 LENGTH OF HOLD AREA NEEDED FOR 00300000
*                                  * 2 AN INPUT OPERATION.
SPACE                                  00310000
TTL   EQU TTINLH-TT+1 LENGTH OF ONE TT ENTRY. 00320000
EJECT                                  00330000
MEND                                  00340000

```

```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 2 * 00030000
.* * 00040000
.* NAME:  $EFDT * 00050000
.* * 00060000
.* FUNCTION:  DEFINE THE LABELS NEEDED TO REFERENCE THE 'FIELD * 00070000
.* DEFINITION TABLE' (FDT) WHICH IS THE FIRST SECTION OF A * 00080000
.* DISPLAY FORMAT. * 00090000
.* * 00100000
.* INPUT OPERANDS:  NONE * 00110000
***** 00120000
$EFDT 00130000
TEXT 00140000
* * * * * 00150000
* THE 'FIELD DESCRIPTER TABLE' OR 'FDT' FOR THE 'DISPLAY FORMAT * 00160000
* FACILITY'. (DFF) * 00170000
* * 00180000
* THE FOLLOWING ARE THE EQUATES FOR THE VERY FIRST FDT ENTRY * 00190000
* WHICH CONTAINS INFORMATION ABOUT THE ENTIRE PHYSICAL FORMAT. * 00200000
* * * * * 00210000
SPACE 2 00220000
DFFDT EQU 0 FIRST BYTE 00230000
* LNG CONTENTS 00240000
SPACE 00250000
FDTTSN EQU DFFDT NUMBER OF SECTORS TEXT STREAM 00260000
* 1 OCCUPIES ON DISK. 00270000
SPACE 00280000
FDTTSL EQU FDTTSN+2 NUMBER OF BYTES OF HOLD AREA 00290000
* 2 REQUIRED TO PUT ENTIRE FORMAT. 00300000
SPACE 00310000
FDTTSI EQU FDTTSL+2 NUMBER OF BYTES OF HOLD AREA 00320000
* 2 REQUIRED IF ALL FIELDS RECEIVED 00330000
* ON INPUT OPERATION. 00340000
SPACE 00350000
FDTM# EQU FDTTSI+1 MODEL NUMBER OF FORMAT 00360000
* 1 00370000
FDTM#2 EQU BIT6 MODEL II 00380000
FDTM#1 EQU BIT7 MODEL I 00390000
FDTNDT EQU BIT5 EXECUTION DATAZ 00400000
SPACE 00410000
FDTNS EQU FDTM#+1 NUMBER OF SECTORS OF FDT 00420000
* 1 00430000
SPACE 00440000
FDTWRK EQU FDTNS+2 WORK/CALC AREA 00450000
* 2 00460000
SPACE 00470000
FDTWK2 EQU FDTWRK+1 EXTRA WORK AREA 00480000
* 1 00490000
SPACE 00500000
FDTCFG EQU FDTWK2+1 FLAG BYTE 00510000
* 1 00520000
FDT11 EQU BIT0 1-THIS FORMAT HAS FIELD DEFINED 00530000
* IN ROW 1, COLUMN 1. 00540000
FDTPRT EQU BIT4 FORMAT GENERATED BY PFGR 00545000
SPACE 2 00550000
* * * * * 00560000
* * 00570000
* EQUATES FOR 'FIELD DEFINITION TABLE' (FDT) * 00580000

```



```

MACRO 00010000
***** 00020000
.* 00030000
.* NAME: $E060 00040000
.* 00050000
.* MODIFICATION LEVEL: VERSION 7, MODIFICATION LEVEL 0 00060000
.* 00070000
.* FUNCTION: 00080000
.* 00090000
.* . CCP GENERATION SECOND PASS MACRO INSTRUCTION -- GENERATE 00100000
.* COMMUNICATIONS I/O INTERFACE ROUTINE 00110000
.* 00120000
.* INPUT OPERANDS: 00130000
.* 00140000
.* . NONE 00150000
.* 00160000
***** 00170000
$E060 00180000
.* 00190000
.*----- GLOBAL VARIABLES REFERENCED -----* 00200000
.* 00210000
.* GBLB &MLTA . 1=MLTA SUPPORTED / 0=NOT 00220000
.* GBLB &NOM . 1=MLTA *NOT* SUPPORTED / 0=IS 00230000
.* 00240000
.* GBLB &BSCA . 1=BSCA SUPPORTED / 0=NOT 00250000
.* GBLB &NOB . 1=BSCA *NOT* SUPPORTED / 0=IS 00260000
.* 00270000
.* GBLB &DFB . 1=DFB SUPPORTED / 0=NOT 00280000
.* GBLB &NDF . 1=DFB *NOT* SUPPORTED / 0=IS 00290000
.* 00300000
.* GBLB &N1050 . 1=1050 *NOT* SUPPORTED / 0=IS 00310000
.* 00320000
.* GBLB &NDME . *NO* DATA MODE ESCAPE 00330000
.* . 1=*NO* DME / 0=DME SUPPORTED 00340000
.* 00350000
.* GBLB &NPBY . 0=BUSY PRINTER SUPPORTED 00352000
.* . 1=BUSY PRINTER NOT SUPPORTED 00354000
.* 00356000
.* GBLB &MIN . MAKE FUNCTIONS TRANSIENT/0=NO 00360000
.* 00370000
.* GBLB &RAI . 1=RESIDENT ACCEPT INPUT 00372000
.* GBLB &NRAI . 0=TRANSIENT ACCEPT INPUT 00374000
.* 00376000
.* GBLB &NCPUM . *NO* CPU S MESSAGES 00377000
.* . 1=*NO* MSGS / 0=SEND MSGS 00378000
.* 00379000
.*----- LOCAL VARIABLES USED -----* 00380000
.* 00390000
.* LCLC &# . USED FOR SINGLE BLANK 00400000
.* 00410000
.* TEXT 00420000
.* 00430000
.*&# SETC ' ' . SINGLE BLANK 00440000
.* 00450000
.* TITLE 'COMMUNICATIONS&#.I/O&#.INTERFACE&#.--&#.USER&#.I/O' 00460000
***** 00470000
.* 00480000
.* TITLE: '$CC4II' - I/O INTERFACE ROUTINE. 00490000
.* 00500000

```

```

* FUNCTION/OPERATION: THE FUNCTION OF THIS RESIDENT ROUTINE IS TO * 00510000
* PROCESS BOTH USER AND SYSTEM I/O REQUESTS. EXTENSIVE DIAGNOSTICS * 00520000
* ARE PERFORMED ON EACH REQUEST PRIOR TO SCHEDULING THE REQUEST. * 00530000
* USER REQUESTS ARE PROCESSED IN THE FOLLOWING MANNER: * 00540000
* * 00550000
* 1. GET/PUT/PUT-NO-WAIT/INVITE/STOP INVITE - FOLLOWING THE * 00560000
* DIAGNOSTICS, AN INPUT OPERATION CAUSES THE RECORD AREA TO BE * 00570000
* BLANKED, AND TUBIIS IS SET ON FOR 'CM'. THE USER'S PARAME- * 00580000
* TER LIST IS MOVED TO THE TCB FOR ADDRESSABILITY BY 'CM'. * 00590000
* THE PARAMETER LIST IS QUEUED ON @PRLQ, AND 'CM' IS POSTED * 00600000
* WITH WORK TO DO AND A WAIT ON THE ECB OCCURS. WHEN CONTROL * 00610000
* RETURNS, THE COPY OF THE PARAMETER LIST IS MOVED BACK TO THE * 00620000
* USER'S PARAMETER LIST AREA. CONTROL NOW RETURNS TO THE USER * 00630000
* VIA THE DSM DISPATCHER. * 00640000
* * 00650000
* 2. ACCEPT INPUT - WHEN AN ACCEPT INPUT REQUEST IS ISSUED, TRAN- * 00660000
* SIENT $CC4AB IS CALLED TO FIND A COMPLETED INVITE INPUT FOR * 00670000
* THIS TASK. IF NO COMPLETED INVITES ARE FOUND ON THE TCBIHQ * 00680000
* CHAIN, CONTROL RETURNS TO $CC4II TO ISSUE A WAIT FOR AN IN- * 00690000
* VITE TO COMPLETE OR SHUTDOWN TO BE REQUESTED BY THE SYSTEM * 00700000
* OPERATOR. IF CONTROL RETURNS DUE TO A SHUTDOWN REQUEST, THE * 00710000
* RETURN CODE OF THE ACCEPT PARAMETER LIST IS SET TO REFLECT * 00720000
* THIS. IF CONTROL RETURNS BECAUSE OF THE COMPLETION OF AN * 00730000
* INVITE INPUT, $CC4AB IS AGAIN INVOKED TO FIND THE COMPLETED * 00740000
* INVITE REQUEST ON THE 'TCBIHQ' CHAIN. WHEN $CC4AB RETURNS * 00750000
* TO THE RESIDENT CODE, THE INVITE INPUT PARAMETER LIST IS * 00760000
* DEQUEUED FROM THE TCBIHQ CHAIN, AND 'CM' IS POSTED TO MOVE * 00770000
* THE DATA TO THE USER'S RECORD AREA AND FREEMAIN THE INVITE * 00780000
* INPUT HOLD BUFFER AND THE INVITE PARAMETER LIST COPY. * 00790000
* * 00800000
* 3. STATUS REQUESTS (ACQUIRE TERMINAL, RELEASE TERMINAL, OR GET * 00810000
* TERMINAL ATTRIBUTES) - AFTER THESE REQUESTS ARE DIAGNOSED, * 00820000
* TRANSIENT $CC4GA IS INVOKED TO EITHER PROCESS THE REQUEST * 00830000
* (GET ATTRIBUTES) OR TO ROUTE THE REQUEST TO THE APPROPRIATE * 00840000
* ACQUIRE OR RELEASE TRANSIENT ($CC4AQ OR $CC4RL). * 00850000
* * 00860000
* 4. CONSOLE REQUESTS - FOR USER REQUESTS TO THE CONSOLE, A GET- * 00870000
* MAIN IS DONE FOR A CONSOLE WRITE TO OPERATOR (WTO) OR WRITE * 00880000
* TO OPERATOR WITH REPLY (WTOR) REQUEST. THE PARAMETER LIST * 00890000
* IS BUILT IN THE GETMAINED AREA FROM THE USER'S PARAMETER * 00900000
* LIST AND THE REQUEST IS ISSUED VIA CONSOLE DATA MANAGEMENT * 00910000
* CONTROL RETURNS TO $CC4II AFTER THE OPERATION IS COMPLETED, * 00920000
* AND A FREEMAIN OCCURS TO FREE THE PARAMETER LIST AREA AND * 00930000
* CONTROL RETURNS TO THE USER PROGRAM. * 00940000
* * 00950000
* 5. DFF REQUESTS - IF THE REQUEST IS FOR A DFF TERMINAL, THE * 00960000
* FOLLOWING ACTION TAKES PLACE: * 00970000
* * 00980000
* - RELEASE TERMINAL - DFF PROCESSING OCCURS FOLLOWING * 00990000
* THE RETURN FROM THE RELEASE TRANSIENTS. * 01000000
* * 01010000
* - GET ATTRIBUTES/ACQUIRE TERMINAL - DFF IS NOT INVOLVED * 01020000
* * 01030000
* - ACCEPT INPUT - THE DFF TASK IS INVOKED BY 'CM'. * 01040000
* * 01050000
* - OTHERS - DFF PROCESSING OCCURS IMMEDIATELY PRIOR TO * 01060000
* POSTING 'CM' THAT HE HAS WORK TO DO. THE DFF TASK IS * 01070000
* THEN INVOKED BY COMMUNICATIONS MANAGEMENT DURING HIS * 01080000
* PROCESSING. * 01090000
* * 01100000

```

```

*      6. SYSTEM REQUESTS ($CC4IS) - FOR SYSTEM TP REQUESTS, THE RE- * 01110000
*      QUEST IS MERELY PASSED ON TO $CC4CM BY MEANS OF QUEUING IT * 01120000
*      ON THE PARM CHAIN AND POSTING HIM.  IF THE OPERATION IS AN * 01130000
*      INVITE INPUT, THE PARAMETER LIST IS MOVED TO THE TUB. * 01140000
*      * 01150000
* INPUT:  INPUT TO $CC4II IS A PARAMETER LIST POINTED TO BY XR2. * 01160000
*      * 01170000
* OUTPUT:  OUTPUT FROM $CC4II WILL BE A SCHEDULED PUT OPERATION, * 01180000
*      DATA ON AN INPUT OPERATION, OR TERMINATION DUE TO ERROR WITH ONE * 01190000
*      OF THE FOLLOWING TERMINATION CODES: * 01200000
*      01 - INVALID OP CODE/MODIFIERS * 01210000
*      02 - INVALID OPERATION FOR CONSOLE * 01220000
*      03 - UNDEFINED SYMBOLIC TERMINAL NAME * 01230000
*      04 - TERMINAL NOT ALLOCATED TO PROGRAM * 01240000
*      05 - TERMINAL REFERENCED BY OTHER THAN ALLOCATED NAME * 01250000
*      06 - BLANK TERMINAL NAME FOR MRTS PROGRAM * 01260000
*      07 - BLANK STN AND REQUESTOR RELEASED * 01270000
*      08 - BLANK STN INVALID FOR THIS OPERATION * 01280000
*      0A - STN NOT ASSIGNED TO TERMINAL * 01290000
*      0C - TERMINAL I/O CAPABILITY DOES NOT MATCH OPERATION * 01300000
*      0D - INVALID OUTPUT LENGTH * 01310000
*      0E - INVALID INPUT LENGTH * 01320000
*      0F - INPUT LENGTH GREATER THAN TP BUFFER LIZE * 01330000
*      10 - INVALID OP WITH DATA FROM PROG. REQ. OUTSTANDING * 01340000
*      11 - INVALID OP WITH INVITE INPUT OUTSTANDING * 01350000
*      14 - INVALID OP WITH NO INVITES OUTSTANDING * 01360000
*      15 - INPUT AREA NOT LARGE ENOUGH FOR BSCA GET BLOCK * 01370000
*      16 - COPY REQUESTED TO TERMINAL WITHOUT MAPPING * 01380000
*      17 - COPY TO TERMINAL NAME NOT FOUND * 01390000
*      18 - COPY INVALID WITH 3275 * 01400000
*      19 - ERASE REQUESTED TO TERMINAL WITHOUT MAPPING * 01410000
*      1A - PUT OVERRIDE TO TERMINAL WITHOUT MAPPING * 01420000
*      1B - INVALID PUT WITH MAPPING - NO EOT * 01430000
*      1C - RECORD AREA TOO SMALL FOR BSCA GET WITH ITB * 01440000
*      1E - OUTPUT LENGTH TO CONSOLE TOO LARGE * 01450000
*      1F - OUTPUT LENGTH GREATER THAN TP BUFFER LENGTH * 01460000
*      35 - DFF TERMINAL REFERENCED BY NON-DFF TASK * 01470000
*      * 01480000
*      * 01490000
* ENTRY POINTS: * 01500000
*      $CC4II - ENTRY FOR USER REQUESTS * 01510000
*      $CC4IS - ENTRY POINT FOR SYSTEM REQUESTS * 01520000
*      * 01530000
* EXTERNAL REFERENCES: * 01540000
*      $CC4TT - TRACE ENTRY * 01550000
*      $CC4PI - TRANSIENT INTERFACE TO CALL $CC4NC AND $CC4AB * 01560000
*      $CC4NC - CANCEL TRANSIENT * 01570000
*      $CC4AB - TRANSIENT TO CHECK ACCEPT INPUT POSSIBILITIES * 01580000
*      $CC4GA - GET TERMINAL ATTRIBUTES TRANSIENT * 01590000
*      CC4TI2 - TERMINATION ENTRY WITH TCB COMPLETION CODE * 01600000
*      * 01610000
* EXITS-NORMAL:  A NORMAL EXIT BACK TO THE USER PROGRAM VIA THE DIS- * 01620000
*      PATCHER TAKES PLACE IF NO ERRORS OCCUR. * 01630000
*      -ERROR:  AN ERROR EXIT TO TERMINATION WITH THE APPROPRIATE * 01640000
*      TCB COMPLETION CODE OCCURS IN CASE OF ERROR. * 01650000
*      * 01660000
* TABLES/WORK AREAS:  A TABLE OF OPERATION CODES IS USED TO CHECK * 01670000
*      VALIDITY OF OP CODES. * 01680000
*      * 01690000
* NOTES:  RESIDENT CODE, REENTRANT. * 01700000
*

```

```

* CHANGE ACTIVITY - $CC4II * 01710000
* RELEASE 3 * 01720000
* @01/INCR/0ES0327 - WAIT OPERATION CODE FROM USER. * 01730000
* @02/INCR/ - PROGRAM REQUEST UNDER FORMAT SUPPORT. * 01740000
* @03/APAR/ - DEADLY EMBRACE FIX. * 01750000
* @04/INCR/ES0418 - RESIDENT ACCEPT OPTION * 01760000
* @05/APAR/S306386 - NEXT DOWN SEARCH IF ACQUIRE FAILED * 01765000
* RELEASE 5 * 01765800
* @06/APAR/S307603 - CORRECTS AN ABNORMAL TERMINATION OF MLTA. * 01766600
* @07/APAR/S307609 - CORRECTS ID VERIFICATION ON A 3275 SW LINE. * 01767400
* @08/APAR/S307694 - INCORRECT ADDITIONAL INVITES AFTER SHUTDOWN. * 01768200
* @09/INCR/ - IOS PROC CHECK ON ALTERED SAVED Q BYTE. * 01769000
* RELEASE 6 * 01769100
* @10/APAR/S309195 - SUPPRESS FREEMAIN OF 'S' MSG TO CPU RECORD AREA * 01769200
* @11/APAR/S309333 - INCORRECT MOVE OF PARM LIST ON RESCHEDULE * 01769300
* @12/APAR/S309332 - SHUTDOWN AND REINVOKE ONLY - INVALID OPCODE. * 01769400
* @13/APAR/S308295 - MRTNEP ACCEPT WITH NO INVITED OWNED TUBS. * 01769500
* @14/APAR/S308397 - GET ATTRIBUTES PC-HALT IF BAD TNT GIVEN. * 01769600
* RELEASE 7 * 01769620
* @15/APAR/S310771 - CONVERT TO USE SCP $DLOG MACRO. * 01769640
* @16/INCR/ES0705 - BUSY PRINTER SUPPORT * 01769660
* @17/INCR/ES0708 - ALLOW S TYPE MESSAGES TO CPU TERMINALS * 01769680
* * 01769700
***** 01770000
EJECT 01780000
SPACE 01790000
***** 01800000
* ENTRY POINT FOR I/O INTERFACE FOR USER REQUESTS 01810000
***** 01820000
SPACE 01830000
* UPON ENTRY FROM USER RELOCATABLE MODULE - XR2 POINTS AT PARM LIST 01840000
ENTRY $CC4II 01850000
$CC4II EQU * 01860000
SPACE 01870000
CCP DISABL,PMRQ DISABLE ALL INTERRUPTS 01880000
ST II@PRL,XR2 SAVE PARAMETER LIST ADDRESS 01890000
SPACE 01900000
***** 01910000
* VALIDATE THE OPERATION CODE PASSED BY THE USER 01920000
* OBTAIN VALIDITY BYTE FROM OP CODE VALIDITY TABLE 01930000
***** 01940000
SPACE 01950000
* XR2 POINTS AT THE PARM LIST 01960000
MNN IIMVBY+4,PLOPC(,XR2) MOVE OP CODE TO DISPLACEMENT 01970000
LA IIVTBL,XR1 POINT XR1 AT OP CODE VALID. TBL. 01980000
IIMVBY MVC IIVBYT(1),*-(,XR1) GET VALIDITY BYTE FROM TABLE 01990000
SPACE 02000000
* -----START-----@02 02010000
CLI PLOPC(,XR2),OPPNW+OPRUF IS THE OP IN THE VALID RANGE? 02020000
* -----END-----@02 02030000
JH IIVLD BRANCH TO TERMINATE IF NOT 02040000
CLI IIVBYT,IIZERO IS VALIDITY BYTE ZERO 02050000
JNE IIVLD JUMP IF NOT 02060000
SPACE 02070000
* VALIDITY BYTE ZERO SO OPERATION IS INVALID 02080000
SPACE 02090000
IIVLD B C4TI2 BRANCH TO TERMINATION 02100000
DC AL1(TCCIOP) INVALID OP COMPLETION CODE 02110000
SPACE 02120000
IIVLD EQU * OP CODE IS ONE OF CCP VALID OPS 02130000

```

```

SLC  PLCHN(2, XR2), PLCHN(, XR2)  CLEAR CHAIN/RETURN CODE IN P. L. 02140000
SLC  PL$RTC(4, XR2), PL$RTC(, XR2) CLEAR INTERNAL RETURN CODE      02150000
*                                     AND INTERNAL OP CODE          02160000
*                                     ----- START -@01             02170000
CLI  PLOPC(, XR2), OPWAIT          IS OP WAIT REQUEST             02180000
JE   IIFOND                        JUMP TO ISSUE WAIT             02190000
*                                     ----- END ---@01            02200000
CLI  PLOPC(, XR2), OPSHQ           IS OP SHUTDOWN INQUIRY        02210000
JNE  IICKAI                        JUMP IF NOT SHUTDOWN INQUIRY    02220000
SPACE                                     02230000
* SHUTDOWN INQUIRY OP CODE                                     02240000
TBN  $FLGA, CPSHUT                 HAS SHUTDOWN BEEN REQUESTED 02250000
JF   IIBBCK                        JUMP IF NOT                    02260000
SPACE                                     02270000
* SHUTDOWN REQUESTED SO PLUG IN PROPER RETURN CODE           02280000
MVI  PLRTC(, XR2), RCXSHD          PLUG SHUTDOWN REQUESTED RTN CODE 02290000
IIBBCK EQU *                       BRACN TO GO BACK                02300000
B    IITEND                        BR TO RETURN TO USER            02310000
SPACE                                     02320000
* CHECK OP CODE TO SEE IF IT'S AN ACCEPT INPUT              02330000
SPACE                                     02340000
IICKAI EQU *                       CHECK FOR ACCEPT INPUT          02350000
TBN  PLOPC(, XR2), OPACI           IS THIS AN ACCEPT INPUT?    02360000
TBF  PLOPC(, XR2), BIT4+BIT6+BIT7  AND NOTHING ELSE?            02370000
BT   IIACI                        BRANCH IF ACCEPT INPUT        02380000
SPACE                                     02390000
* NOT ACCEPT INPUT SO MUST VALIDATE THE SYMBOLIC TERMINAL NAME 02400000
* PROVIDED BY USER ALONG WITH THE OPERTAION                 02410000
EJECT                                  02420000
***** 02430000
* LOCATE SYMBOLIC TERMINAL NAME IN THE TERMINAL NAME TABLE 02440000
***** 02450000
SPACE 02460000
L    @TNT, XR1                     POINT XR1 AT FIRST TNT ENTRY 02470000
* CHECK FOR END OF TNT 02480000
* XR1 POINTS AT TNT 02490000
SPACE 02500000
L    PLRECA(, XR2), XR2            POINT XR2 AT RECORD AREA 02510000
SPACE 02520000
IITNTN CLI TNTCCP(, XR1), STOPER   ARE WE AT END OF TNT 02530000
JE   IINFND                       JUMP IF AT END TO NOT FOUND 02540000
SPACE 02550000
* CHECK THIS ENTRY IN THE TNT 02560000
SPACE 02570000
CLC  IISTNL-1(IISTNL, XR2), TNTNAM(, XR1) COMPARE TNT STN 02580000
JH   IILOOK                       JUMP HIGH TO LOOK AGAIN 02590000
JE   IIFOND                       JUMP IF FOUND IN TNT 02600000
SPACE 02610000
* NOT FOUND - HAVE PROGRAM ERROR 02620000
SPACE 02630000
IINFND EQU *                       STN NOT FOUND IN TNT 02640000
L    II@PRL, XR2                  POINT TO PARAMETER LIST 02650000
SLC  PLTUBA(2, XR2), PLTUBA(, XR2) CLEAR TUB POINTER TO ZEROS 02660000
CLI  PLOPC(, XR2), OPGTA          WAS THIS A GET ATTRIBUTES OP? 02670000
JE   IIGTA                        SKIP TERMINATION IF YES 02680000
B    C4TI2                        BRANCH TO TERMINATION 02690000
DC   AL1(TCCNNF)                 TERMINAL NAME NOT FOUND 02700000
SPACE 02710000
* LOOK SOME MORE 02720000
* DETERMINE HOW MUCH TO BUMP TNT REGITER POINTER 02730000

```


SPACE			02740000
IILOOK EQU *		LOOK AT ANOTHER ENTRY	02750000
IIBUMP LA	TNTLN(,XR1),XR1	BUMP POINTER BY TNT FIXED SIZE	02760000
B	IITNTN	BRANCH TO COMPARE AGAIN	02770000
EJECT			02780000
*****			02790000
* SYMBOLIC TERMINAL NAME WAS FOUND IN TNT			* 02800000
* DETERMINE IF OPERATION IS VALID FOR THE TYPE TNT ENTRY FOUND			* 02810000
*****			02820000
SPACE			02830000
* XR1 POINTS AT PROPER TNT ENTRY			02840000
SPACE			02850000
IIFOND EQU *		STN FOUND IN TNT	02860000
L	II@PRL,XR2	RESTORE PARM LIST POINTER	02870000
B	CC4TT	BRANCH TO TRACE ROUTINE	02880000
DC	AL1(TTII)	TRACE FOR \$CC4II	02890000
*		----- START -@01	02900000
CLI	PLOPC(,XR2),OPWAIT	IS OP WAIT REQUEST	02910000
JNE	IICKNM	JUMP IF NOT WAIT REQUEST	02920000
CLC	PLOUTL(2,XR2),IILN10	WAIT DATA AREA 10 BYTES LONG	02950000
JNL	IIWAT1	JUMP TO ISSUE WAIT	02960000
* ERROR - WAIT OPERATION DATA LENGTH IS TOO SHORT			02970000
B	C4TI2	BRANCH TO TERMINATION	02980000
DC	AL1(TCCWAT)	INVALID WAIT DATA LENGTH	02990000
IIWAT1 EQU *			03000000
ALC	PLRECA(2,XR2),CC0006	BUMP POINTER TO DATA AREA	03010000
L	PLRECA(,XR2),XR2	LOAD WAIT DATA AREA POINTER	03020000
MVI	IIZERO(,XR2),IITIMR	SET TIME INDICATOR FOR HHMMSS	03030000
SBN	IIZERO(,XR2),IITMWT	SET THE WAIT FLAG	03040000
MVI	IIWFLG(,XR2),IIZERO	INITIALIZE THE WAIT FLAG	03050000
CCP	ENABLE,PMRQ	ENABLE ALL INTERRUPTS	03060000
SVC	0	SUPERVISOR CALL	03070000
DC	XL1'15'	RIB FOR TIMER WAIT	03080000
B	IITEND	BR TO RETURN TO USER	03090000
IICKNM EQU *			03100000
*		----- END ---@01	03110000
L	PLRECA(,XR2),XR2	RESTORE RECORD AREA POINTER	03120000
SPACE			03130000
TBN	TNTFLG(,XR1),TNTBLK	IS STN ALL BLANKS	03140000
JF	IINBLK	JUMP IF NOT ALL BLNKA	03150000
SPACE			03160000
* HAVE SYMBOLIC TERMINAL NAME OF ALL BLANKS			03170000
SPACE			03180000
* THUS THE TERMINAL TO WHICH THE OPERATION IS DIRECTED IS			03190000
* THE ONE AND ONLY REQUESTOR OF A SINGLE REQUESTOR PROGRAM			03200000
SPACE			03210000
* DETERMINE IF OPERATION IS VALID FOR STN OF ALL BLANKS			03220000
SPACE			03230000
TBN	IIVBYT,IIVBLK	IS IT VALID OP FOR STN-BLANKS	03240000
JT	IICKBK	JUMP IF VALID OP	03250000
SPACE			03260000
* HAVE ERROR** INVALID OPERATION FOR STN OF BLANKS			03270000
SPACE			03280000
B	C4TI2	BRANCH TO TERMINATION	03290000
DC	AL1(TCCIBL)	INVALID BLANK NAME	03300000
EJECT			03310000
IICKBK EQU *		CHECK VALIDITY OF BLANK NAME	03320000
SPACE			03330000
* FIRST CHECK TO SEE IF PROGRAM IS NON-MRTS PROGRAM			03340000
SPACE			03350000

	L	NCTCB@,XR1	POINT TO USER'S TCB	03360000
	TBF	TCBDMG(,XR1),TCBMTS	IS IT NON-MRTS PROGRAM?	03370000
	JT	IISR	JUMP IF NON-MRTS PROGRAM	03380000
	SPACE			03390000
*		HAVE USE OF BLANK NAME FOR MRTS PROGRAM		03400000
	SPACE			03410000
	B	C4TI2	BRANCH TO TERMINATION	03420000
	DC	AL1(TCCMBL)	BLANK NAME FOR MRTS PROGRAM	03430000
	SPACE			03440000
IISR	EQU	*		03450000
	TBN	TCBDMG(,XR1),TCBKRQ	IS THIS A CONSOLE REQUESTOR TASK	03460000
	JF	IINKRQ	BRANCH IF NOT	03470000
	L	@KNTUB,XR1	LOAD POINTER TO CONSOLE TUB	03480000
	J	IIKRQ	BRANCH TO MOVE IN NAME	03490000
IINKRQ	EQU	*		03500000
	CLI	TCBTUB-1(,XR1),NOBIT	IS TUB POINTER NULL	03510000
	L	TCBTUB(,XR1),XR1	POINT XR1 AT REQUESTOR TUB	03520000
	TBN	TUBAT1(,XR1),TUBREQ	IS TUB REQUESTING TERMINAL	03530000
	JC	IIRBL,ANY+FALSE+EQ	JUMP IF NULL TUB POINTER OR	03540000
			TUB IS NOT REQUESTING	03550000
*				03560000
	SPACE			03570000
*		BLANK NAME COMPLETELY VALID		03580000
	SPACE			03590000
IIKRQ	EQU	*		03600000
	L	TUBTNT(,XR1),XR1	POINT XR1 AT THIS TUBS TNT ENTRY	03610000
	MVC	IISTNL-1(IISTNL,XR2),TNTNAM(,XR1)	MOVE STN TO RECORD AREA	03620000
	J	IINBLK	JUMP TO HANDLE FURTHER	03630000
	SPACE			03640000
*		ERROR - BLANK NAME TO NON-MRTS WITH REQUESTOR RELEASED ALREADY		03650000
	SPACE			03660000
IIRBL	EQU	*		03670000
	B	C4TI2	BRANCH TO TERMINATION	03680000
	DC	AL1(TCCRBL)	BLANK NAME WITH REQUESTOR REL.	03690000
	EJECT			03700000
*****				03710000
*		HAVE REQUEST TO A SPECIFIC TUB AT THIS POINT		03720000
*****				03730000
	SPACE			03740000
*		XR1 POINTS TO TNT ENTRY		03750000
*		ASSURE THAT THIS TNT HAS BEEN ASSIGNED TO A TUB		03760000
	SPACE			03770000
IINBLK	EQU	*	HAVE TERMINAL REQUEWT TO HANDLE	03780000
	CLI	TNTTUB-1(,XR1),NOBIT	IS TNTTUB POINTER NULL	03790000
	JNE	IINNUL	JUMP IF POINTER NOT NULL	03800000
	SPACE			03810000
*		ERROR - TNT NOT ASSINGED TO A TUB		03820000
	SPACE			03830000
	B	C4TI2	BRANCH TO TERMINATION	03840000
	DC	AL1(TCCNAT)	TNT NOT ASSIGNED TO TUB	03850000
	SPACE			03860000
IINNUL	EQU	*	HAVE VALID TNT TUB POINTER	03870000
	L	II@PRL,XR2	POINT XR2 AT PARM LIST	03880000
.*				03890000
	AIF	(&NOM).NML1	. SKIP IF *NO* MLTA	03900000
.*				03910000
*****				03920000
*		FOR 1050 ONLY		03930000
*****				03940000
	SPACE			03950000
*		DETERMINE IF THIS IS SUB-TERMINAL NAME		

```

SPACE 03960000
MVC PL$MCT(1,XR2),TNTMCT(,XR1) MOVE MCT INDICES INTO PARM LIST 03970000
CLI TNTMCT(,XR1),NOBIT IS THIS SUB-TERMINAL NAME 03980000
JE IISTNT JUMP IF NOT SUB-TERMINAL NAME 03990000
SPACE 04000000
* HAVE SUB-TERMINAL NAME SO MUST GET ADDR OF MASTER TNT FROM 04010000
* TTTTNT FIELD 04020000
SPACE 04030000
L TTTTNT(,XR1),XR1 POINT XR1 AT MASTER TNT 04040000
SPACE 04050000
.NML1 ANOP 04060000
ENTRY IISTNT 04070000
IISTNT EQU * SAVE TNT ADDR 04080000
***** 04090000
ST IIVTNT,XR1 SAVE TNT ENTRY ADDRESS 04100000
MVC PLTUBA(2,XR2),TNTTUB(,XR1) MOVE TUB @ TO PARM LIST 04110000
IIGTA ALC PLRECA(2,XR2),CC0006 ADD STN LENGTH TO BUMP RECORD 04120000
* ADDRESS TO DATA ADDRESS 04130000
EJECT 04140000
SPACE 04150000
* CHECK REQUEST TYPE - DETERMINE IF IT IS STATUS REQUEST 04160000
SPACE 04170000
TBN PLOPC(,XR2),OPSTAT IS IT STATUS OPERATION 04180000
JF IITKNS JUMP IF NOT NOT STATUS OPERATION 04190000
SPACE 04200000
* HAVE STATUS OPERATION (RELEASE OR GET ATTRIBUTES) 04210000
SPACE 04220000
ST PL$TNT(,XR2),XR1 PLUG TNT ADDR INTO PARM LIST 04230000
B CC4PI BRACNH TO BRING IN TRANSIENT 04240000
DC AL1(CC4GA) GET ATTRIBUTES TRANSIENT ID 04250000
SPACE 04260000
CCP DISABL,PMRQ DISABLE INTERUPTS. 04310000
CLI PLTUBA-1(,XR2),NOBIT TEST IF TUB WAS FOUND? @14 04312000
BE IITEND NO THEN EXIT II. @14 04314000
SPACE 04316000
L NCTCB@,XR1 POINT XR1 AT CURRENT TCB 04320000
CLI TCBCMP(,XR1),NOBIT IS TCB COMPLETION CODE ZERO 04330000
.* 04340000
AIF (&DFF).MAP1 . SKIP IF DFF SUPPORTED 04350000
.* 04360000
BE IISEND BRANCHOUT IF ZERO (OK) 04370000
.* 04380000
.MAP1 AIF (&NDF).NMAP1 . SKIP IF *NO* DFF SUPPORT 04390000
.* 04400000
JNE IIBTCB BRANCH TO HALT IF NON-ZERO 04410000
SPACE 04420000
***** 04430000
* CHECK FOR MAPPING IF GOOD COMPLETION CODE * 04440000
***** 04450000
SPACE 04460000
TBN PLOPC(,XR2),OPREL IS THIS A RELEASE REQUEST? 04470000
BT IIMAPP BRANCH TO MAPPING IF YES 04480000
B IISEND 04490000
.NMAP1 ANOP 04500000
EJECT 04510000
***** 04520000
* BAD TCB COMPLETION CODE * 04530000
***** 04540000
SPACE 04550000
ENTRY IIBTCB 04560000

```

IIBTCB EQU *			04570000
MVC IITCMP(1),TCBCMP(,XR1)	MOVE TCB COMP CODE SUCH THAT ARR		04580000
*	POINTS TO IT		04590000
SPACE			04600000
B C4TI2	BRANCH TO TERMINATE THIS TASK		04610000
IITCMP DS CL1	TCB COMPLETION CODE		04620000
SPACE			04630000
*****			04640000
* DETERMINE IF THIS TUB IS THE CONSOLE			04650000
*****			04660000
SPACE			04670000
IITKNS EQU *	TEST FOR CONSOLE TUB		04680000
TBN TNTFLG(,XR1),TNTKNS	IS IT CONSOLE		04690000
L TNTTUB(,XR1),XR1	POINT XR1 AT TUB		04700000
JT IIKNSL	JUMP IF CONSOLE TO CONSOLE HNDLR		04710000
SPACE			04720000
* HAVE REQUEST TO TERMINAL - VALIDATE IT			04730000
EJECT			04740000
*****			04750000
* PREPARE FOR PHYSICAL TP IO - MUST VALIDATE REQUEST FIRST			04760000
*****			04770000
SPACE			04780000
* VALIDATE THAT THIS TUB BEONGS TO TCB REQUESTING IT			04790000
SPACE			04800000
CLC TUBTCB(2,XR1),NCTCB@	DOES CURRENT TCB = TUBTCB?		04810000
TBF TUBAT3(,XR1),TUBALC	IS TUB PHYSICALLY ALLOCATED ?		04820000
JC IIPHYS,NONE+FALSE+HI+LO	JUMP IF OKAY		04830000
SPACE			04840000
* CURRENT TCB DOES NOT MATCH TUB TCB SO PROGRAM ERROR			04850000
* TERMINAL NOT ALLOCATED TO THIS PROGRAM			04860000
SPACE			04870000
B C4TI2	BRANCH TO TERMINATION		04880000
DC AL1(TCCNAP)	TERMINAL NOT ALLOCATED TO PROG		04890000
IIPHYS EQU *			04900000
TBN TUBAT1(,XR1),TUBONL	IS TUB ONLINE?		04910000
JT IITCBK	BRANCH IF YES		04920000
MVI PLRTC(,XR2),RCXOFF	SET COMPLETION CODE TO OFFLINE		04930000
B IIREND	BRANCH TO RETURN		04940000
SPACE			04950000
IITCBK EQU *	CHECK THAT TUB POINTS TO CORRECT		04960000
*	TNT ENTRY		04970000
CLC TUBTNT(2,XR1),IIVTNT	DOES TUB PT TO PROPER TNT		04980000
JE IITNTK	JUMP IF YES		04990000
SPACE			05000000
* ERROR TNT POINTS TO TUB BUT TUB DOES NOT POINT TO TNT			05010000
* TERMIANL NOT REFERECNED BY NAME BY WHICH IT WAS ALLOCATED			05020000
SPACE			05030000
B C4TI2	BRANCH TO TERMINATION		05040000
DC AL1(TCCNAN)	NOT ALLOCATED BY THIS NAME		05050000
EJECT			05060000
* ASSURE TERMINAL DOES NOT HAVE OUTSTANDING IMPLICIT INVITE INPUT			05070000
* ONLY ACCEPT INPUT CAN BE ISSUED TO TERMINAL WITH IMPLICIT INVITE INPT			05080000
* WHICH HAS NOT BEEN SATISFIED			05090000
SPACE			05100000
IITNTK EQU *			05110000
TBN TUBAT2(,XR1),TUBIMI	IS IMPLICIT INVITE UNSATISFIED		05120000
JF IICKIS	JUMP IF NO IMPLICIT INVITE INPUT		05130000
SPACE			05140000
* ERROR - HAVE NON ACCEPT INPUT REQUEST TO TUB WITH IMPLICIT INV. INPUT			05150000
* UNSATISFIED			05160000

* DATA FROM PROGRAM REQUEST OUTSTANDING	05170000
SPACE	05180000
B C4TI2 BRANCH TO TERMINATION	05190000
DC AL1(TCCIMO) IMPLICIT INVITE OUTSTANDING	05200000
SPACE	05210000
* DETERMINE IF INVITE INPUT IS OUTSTANDING TO THIS TERMINAL	05220000
* ERROR IF SO UNLESS THIS IS STOP INVITE INPUT	05230000
SPACE	05240000
IICKIS EQU * CHECK FOR OUTSTANDING INVITE INP	05250000
SPACE	05260000
* TEST FOR TUB IN COMMAND INTERRUPT MODE	05270000
* IF YES, THEN MUST HAVE INVITE OUTSTANDING TO TERMINAL	05280000
SPACE	05290000
TBN TUBAT2(,XR1),TUBDTA+TUBCMD IS TUB IN COM'ND INTERRUPT MODE	05300000
JT IIVSCH JUMP IF CMD INT SINCE INV SCHED.	05310000
TBN TUBAT2(,XR1),TUBIIS IS INVITE INPUT SCHEDULED ON TUB	05320000
JF IINIIS JUMP IF NO INVITE SCHEDULED	05330000
SPACE	05340000
EJECT	05350000
* HAVE INVITE INPUT SCHEDULED - THIS OP IS VALID ONLY IF IT IS	05360000
* A STOP INVITE INPUT	05370000
SPACE	05380000
IIVSCH EQU * INVITE INPUT SCHEDULED	05390000
TBN PLOPM(,XR2),OPSTOP IS THIS STOP INVITE INPUT	05400000
JT IIIOCP JUMP IF STOP SINCE OPERATION OK	05410000
SPACE	05420000
* INVALID OPERATION WITH INVITE INPUT OUTSTANDING	05430000
SPACE	05440000
B C4TI2 BRANCH TO TERMINATION	05450000
DC AL1(TCCIO) INVITE INPUT OUTSTANDING	05460000
SPACE	05470000
* NO INVITE SCHEDULED SO THIS MUST NOT BE A STOP INVITE INPUT	05480000
SPACE	05490000
IINIIS EQU * NO INVITE SCHEDULED	05500000
TBN PLOPM(,XR2),OPSTOP IS THIS STOP INVITE INPUT	05510000
JF IIIOCP JUMP IF NOT STOP INVITE	05520000
SPACE	05530000
* ERROR - HAVE STOP INVITE WITH NO INVITE OUTSTANDING	05540000
SPACE	05550000
B C4TI2 BRANCH TO TERMINATION	05560000
DC AL1(TCCNIO) NO INVITES OUTSTANDING	05570000
EJECT	05580000
III OCP EQU * CHECK FOR PROPER IO CAPABILITY	05590000
*****	05600000
MNN IIIOTS+1,PLOPC(,XR2) MOVE RIGHTMOST 4 BITS OF OP CODE	05610000
* TO TBN INSTURCTION	05620000
SBF IIIOTS+1,X'FC' SET OFF ALL BUT RIGHTMOST 2 BITS	05630000
SPACE	05640000
* NOW READY TO TEST TERMINAL CHARACTERISTICS VS REQUEST IO	05650000
SPACE	05660000
IIIOTS TBN TUBCHR(,XR1),*-* TEST TUB FOR IO CAPABILITY	05670000
JT IIRQTY JUMP IF I/O OKAY	05680000
SPACE	05690000
* I/O CAPABILTY DOES NOT MATCH REQUEST	05700000
SPACE	05710000
B C4TI2 BRANCH TO TERMINATION	05720000
DC AL1(TCCIOC) I/O CAPABILITY DOES NOT MATCH	05730000
* DETERMINE REQUEST TYPE	05740000
SPACE	05750000
* VALIDATE OP CODE AND MODIFIER TO EXTENT THAT THOSE BITS THAT MUST	05760000

* BE OFF FOR A USER REQUEST ARE OFF		05770000
SPACE		05780000
* CHECK THAT OP CODE IS VALID FOR CONSOLE AS TERMINAL NAME		05790000
SPACE		05800000
IIKNSL EQU *	CONSOLE REQUEST HANDLER	05810000
TBN IIVBYT,IIVKNS	IS OP VALID FOR CONSOLE	05820000
*	-----START-----@02	05830000
CLI PLOPC(,XR2),OPPUT+OPRUF	AND NOT A RUF PUT MESSAGE ?	05840000
JC IIKLEN,TANDNE	JUMP IF VALID	05850000
*	-----END-----@02	05860000
SPACE		05870000
* ERROR - HAVE OP CODE WHICH IS VALID OP BUT NOT FOR CONSOLE		05880000
SPACE		05890000
B C4TI2	BRANCH TO TERMINATION	05900000
DC AL1(TCCIOK)	INVALID OP FOR CONSOLE	05910000
EJECT		05920000
SPACE		05930000
* CHECK FOR OUTPUT LENGTH TO CONSOLE TOO LARGE		05940000
SPACE		05950000
IIKLEN EQU *	TEST CONSOLE LENGTH FOR > 71/107	05960000
TBN PLOPC(,XR2),OPPUT	IS THIS AN OUTPUT OPERATION?	05970000
JF IIRQTY	BRANCH IF NOT	05980000
CLC PLOUTL(2,XR2),IIMAXK	LENGTH GREATER THAN 71?	05990000
JNH IIRQTY	BRANCH IF NOT	06000000
TBN PLOPC(,XR2),OPGET	IS THIS A WTOR?	06010000
JT IITOLG	BRANCH TO ERROR IF YES	06020000
CLC PLOUTL(2,XR2),IIMAXC	IS LENGTH GREATER THAN 107?	06030000
JNH IIRQTY	BRANCH OUT IF NOT	06040000
IITOLG EQU *		06050000
SPACE		06060000
* ERROR - OUTPUT LENGTH TO CONSOLE GREATER THAN 71 FOR WTOR OR		06070000
* GREATER THAN 117 FOR A WTO.		06080000
SPACE		06090000
B C4TI2	BRANCH TO TERMINATION	06100000
DC AL1(TCCIKL)	INVALID CONSOLE LENGTH	06110000
SPACE		06120000
IIRQTY EQU *		06130000
TBF PLOPM(,XR2),OP\$SYS+OPOLT+OPDISC+OPUNAM	CHECK OP MOD BITS	06140000
*	TO ASSURE VALIDITY SET	06150000
JT IIVLNO	JUMP IF OKAY	06160000
SPACE		06170000
* ERROR - OP MODIFER HAS INVALID BITS FOR USER REQUEST		06180000
SPACE		06190000
B C4TI2	BRANCH TO TERMINATION	06200000
DC AL1(TCCIOP)	INVALID OP OR MODIFIER	06210000
EJECT		06220000
* VALIDATE RECORD LENGTHS PASSED IN THE PARM LIST FROM THE USER		06230000
SPACE		06240000
IIVLNO EQU *	VALIDATE OUTPUT LENGTH	06250000
SPACE		06260000
.*		06270000
AIF (&NOM).NML2	. SKIP IF *NO* MLTA SUPPORT	06280000
.*		06290000
B IIMCT	BRN TO SEE IF MCT AND VALID	06300000
SPACE		06310000
.NML2 ANOP		06320000
ENTRY IIVLNN		06330000
IIVLNN EQU *		06340000
TBN PLOPC(,XR2),OPPUT	DOES OP INVOLVE OUTPUT	06350000
JF IIVLNI	JUMP IF NO OUTPUT	06360000

SPACE			06370000
* CHECK OUTPUT LENGTH FOR GREATER THAN ZERO			06380000
SPACE			06390000
CLC	PLOUTL(2, XR2), X\$0000	COMPARE LENGTH VS ZERO	06400000
JH	IIFUTH	JUMP IF GREATER THAN ZERO	06410000
TBF	TUBAT1(, XR1), TUBKNM	IS THIS THE CONSOLE?	06420000
JF	IIIOL	BRANCH TO ERROR IF YES	06430000
SPACE			06440000
.*			06450000
AIF	(&NOM).NYET	. SKIP IF *NO* MLTA SUPPORT	06460000
.*			06470000
* ZERO LENGTH IS OK IF THIS IS MLTA PUT EOT/EOB			06480000
SPACE			06490000
TBF	TUBCHR(, XR1), TUBLNE	IS THIS AN MLTA REEQUEST?	06500000
TBN	PLOPC(, XR2), OPBLK	IS IT PUT EOT/EOB?	06510000
BT	IIREND	BRNACH IF YES TO RETURN	06520000
SPACE			06530000
.NYET	ANOP		06540000
.*			06550000
AIF	(&NOB).NBSC1	. SKIP IF *NO* BSCA SUPPORT	06560000
.*			06570000
* ZERO LENGTH IS OK IF THIS IS PUT-EOT/EOB AND HE OWNS THE BSCA LINE			06580000
SPACE			06590000
ENTRY	IIVLNP		06600000
IIVLNP	EQU *		06610000
TBN	PLOPC(, XR2), OPBLK	IS THIS PUT EOT/EOB?	06620000
TBN	TUBAT2(, XR1), TUBOWN	DOES HE OWN THE BSCA LINE?	06630000
JT	IIVLNI	BRANCH PAST ERROR IF YES	06640000
SPACE			06650000
.*			06660000
AIF	(&NDF).BEEP	. SKIP IF *NO* DFF SUPPORT	06670000
.*			06680000
TBN	PLOPC(, XR2), OPEAU	IS THIS AN ERASE OPERATION?	06690000
* -----START-----@02			06700000
TBF	PLOPC(, XR2), OPRUF-OPEAU	AND NOT A RUF PUT ?	06710000
JT	IIVLNI	BRANCH IF YES - LENGTH OK	06720000
* -----END-----@02			06730000
.BEEP	ANOP		06740000
.NBSC1	ANOP		06750000
* ERROR HAVE OUTPUT LENGTH NOT GREATER THAN ZERO			06760000
SPACE			06770000
ENTRY	IIIOL		06780000
IIIOL	EQU *		06790000
B	C4TI2	BRANCH TO TERMINATION	06800000
DC	AL1(TCCIOL)	INVALID OUTPUT LENGTH	06810000
EJECT			06820000
SPACE	2		06830000
* CHECK FOR OUTPUT LENGTH GREATER THAN TP BUFFER LENGTH			06840000
SPACE			06850000
IIFUTH	EQU *		06860000
AIF	(&NOB).NBTH		06870000
TBN	TUBTA2(, XR1), TASITB	IS THIS PUT WITH ITB?	06880000
JF	IINITB	BRANCH IF NOT	06890000
CLC	PLOUTL(2, XR2), TUBRCL(, XR1)	OUTPUT LENGTH < RECORD LENGTH?	06900000
JNL	IINITB	BRANCH PAST TERMINATION IF NOT	06910000
SPACE			06920000
* *ERROR* OUTPUT LENGTH LESS THAT RECORD LENGTH FOR PUT ITB			06930000
SPACE			06940000
B	C4TI2	BRANCH TO TERMINATION	06950000
DC	AL1(TCCITB)		06960000

IINITB	EQU	*		06970000
TBN	PLOPC(,XR2),OPBLK		IS THIS PUT EOT/EOB?	06980000
TBF	TUBTA1(,XR1),TASDF		AND NOT MAPPING?	06990000
TBN	TUBCHR(,XR1),TUBLNE		BSCA?	07000000
JF	IIVLNI		BRANCH OUT IF NOT ALL	07010000
MVC	IIWORK,PLOUTL(2,XR2)		MOVE LENGTH TO WORK AREA	07020000
TBN	PLOPC(,XR2),OPNOW		IS THIS A PUT-NO-WAIT?	07030000
JF	IILTER		BRANCH OUT IF NOT	07040000
ALC	IIWORK(2),II19		ADD LENGTH OF PARAMETER LIST	07050000
IILTER	EQU	*		07060000
*			-----START-----@03	07070000
CLC	IIWORK(2),#TPPUT		IS OUTPUT LENGTH GREATER?	07080000
*			-----END-----@03	07090000
JNH	IIVLNI		BRANCH PAST TERMINATION IF NOT	07100000
SPACE				07110000
* *ERROR*	OUTPUT LENGTH IS GREATER THAN TP BUFFER LENGTH			07120000
SPACE				07130000
B	C4TI2		BRNACH TO TERMINATION	07140000
DC	AL1(TCCLRG)			07150000
EJECT				07160000
SPACE				07170000
* CHECK INPUT LENGTH FOR GREATER THAN ZERO IF APPROPRIATE				07180000
SPACE				07190000
.NBTH	ANOP			07200000
ENTRY	IIVLNI			07210000
IIVLNI	EQU	*	CHECK INPUT LENGTH IF NECESSARY	07220000
TBN	PLOPC(,XR2),OPGET		DOES OP INVOLVE GET	07230000
JF	IIIREQ		JUMP IF NOT	07240000
SPACE				07250000
IISETS	EQU	*	CHECK INPUT LENGTH	07260000
*			OPERATION IS SCHEDULED ON TERMIN	07270000
CLC	PLINL(2,XR2),X\$0000		COMPARE INPUT LENGTH VS ZERO	07280000
JH	II#TPB		JUMP IF GREATER THAN ZERO	07290000
SPACE				07300000
*ERROR	- INPUT LENGTH NOT GREATER THAN ZERO			07310000
SPACE				07320000
B	C4TI2		BRANCH TO TERMINATION	07330000
DC	AL1(TCCIIL)		INVALID INPUT LENGTH	07340000
EJECT				07350000
II#TPB	EQU	*	IF INVITE-CHECK INPUT LENGTH VS	07360000
.*				07370000
AIF	(&NOB).NBS3		. SKIP IF *NO* BSCA SUPPORT	07380000
.*				07390000
*****				07400000
* IF BSCA TERMINAL - CHECK IF RECORD AREA LARGE ENOUGN IF ITB				* 07410000
*****				07420000
SPACE				07430000
TBN	TUBCHR(,XR1),TUBLNE		IS THIS A BSCA LINE?	07440000
TBN	TUBTA2(,XR1),TASITB		IS ITB SPECIFIED?	07450000
JF	IIINRT		BRANCH OUT IF NOT	07460000
CLC	PLINL(2,XR2),TUBRCL(,XR1)		IS RECORD AREA LARGE ENOUGH?	07470000
JNL	IIINRT		BRANCH OUT IF YES	07480000
SPACE				07490000
* ERROR - RECORD AREA NOT LARGE ENOUGH FOR BSCA ITB				* 07500000
SPACE				07510000
B	C4TI2		BRANCH TO TERMINATION	07520000
DC	AL1(TCCITB)		RECORD AREA TOO SMALL FOR ITB	07530000
SPACE				07540000
* CHECK FOR DATA PENDING ON A BSCA LINE				07550000
SPACE				07560000


```

.NBS3 ANOP 07570000
ENTRY IIINRT 07580000
IIINRT EQU * 07590000
TBN PLOPC(,XR2),OPINV DOES OP INVOLVE INVITE INPUT 07600000
JF IIOUTR JUMP IF NOT INVITE INPUT 07610000
.* 07620000
AIF (&NOB).NBS31 . SKIP IF *NO* BSCA SUPPORT 07630000
.* 07640000
SPACE 07650000
* CHECK FOR DATA PENDING ON THE BSCA LINE 07660000
SPACE 07670000
TBN TUBAT2(,XR1),TUBOWN DOES HE OWN A BSCA LINE? 07680000
L TUBLCB(,XR1),XR1 POINT TO HIS DTF 07690000
TBN $BDOPC(,XR1),$BOPUT WAS LAST OPERATION A PUT? 07700000
TBN LCBAT1(,XR1),LCBNTQ WAS LAST OPERATION PUT-MESSAGE? 07710000
JF IINDPD BRANCH IF NEITHER - INVITE OK 07720000
MVI PLRTC(,XR2),RCXDPD SET DATA PENDING RETURN CODE 07730000
B IIREND GO RETURN TO USER 07740000
.* 07750000
.NBS31 ANOP 07760000
.* 07770000
ENTRY IINDPD 07780000
IINDPD EQU * 07790000
SPACE 07800000
* HAVE INVITE INPUT SO INPUT LENGTH MUST NOT BE LARGER THAN THE 07810000
* SIZE FOR THE ENTIRE TP BUFFER 07820000
SPACE 07830000
L NCTCB@,XR1 POINT XR1 AT CURRENT TCB 07840000
ALC TCBIIC(1,XR1),X$0001 ADD 1 TO TCB INVITE INPUT COUNT 07850000
SPACE 07860000
* -----START-----@03 07870000
CLC PLINL(2,XR2),#TPANY COMPARE INPUT LENGTH VS TP BUF 07880000
* -----END-----@03 07890000
JNH IIIREQ JUMP IF INPUT LENGTH NOT HIGH 07900000
SPACE 07910000
* ERROR - FOR INVITE INPUT - THE INPUT LENGTH IS GT TP BUFFER SIZE 07920000
SPACE 07930000
B C4TI2 BRANCH TO TERMINATION 07940000
DC AL1(TCCTPB) TP BUFFER LENGTH PROBLEM 07950000
EJECT 07960000
SPACE 5 07970000
***** 07980000
* POST $CC4CM AND ISSUE WAIT * 07990000
* OR ISSUE WTO/WTOR FOR CONSOLE REQUEST * 08000000
***** 08010000
SPACE 08020000
IIOUTR EQU * 08030000
SPACE 08040000
ENTRY IIIREQ 08050000
IIIREQ EQU * I/O INTERFACE REQUEST 08060000
SPACE 08070000
* DETERMINE IF REQUEST IS FOR CONSOLE OR $CC4CM 08080000
SPACE 08090000
L PLTUBA(,XR2),XR1 POINT XR1 AT TUB 08100000
TBF TUBAT1(,XR1),TUBKNM IS THIS CONSOLE TUB? 08110000
.* 08120000
AIF (&DFF).NM2RT SKIP IF MAPPING 08130000
AIF (&NPBY).NBSY1 SKIP IF NOT BUSY PRINTER 08132000
JT IIBUSY NOT CONSOLE,TEST BUSY PRNT 08134000
AGO .NM2RT 08136000

```

```

.NBSY1 ANOP                                08138000
JT      IIUPST                             NOT CONSOLE, GO TO POST 08140000
.NM2RT ANOP                                08143000
AIF     (&NDF).SK1RT                       SKIP IF NOT MAPPING    08146000
JT      IICKMP                             NOT CONSOLE, CHECK FOR MAPPING 08160000
.SK1RT ANOP                                08170000
EJECT                                       08180000
SPACE 2                                    08190000
***** 08200000
*** HAVE A CONSOLE REQUEST - ISSUE THE WTO OR WTOR AS APPROPRIATE *** 08210000
***** 08220000
SPACE 3                                    08230000
* FIRST MUST BUILD A CONSOLE PARAMETER LIST IN THE TCB 08240000
SPACE                                       08250000
L       NCTCB@,XR2                         POINT TO THE TCB      08260000
LA      TCBPL(,XR2),XR2                    POINT XR2 AT PARM LIST AREA 08270000
MVC     $LGO@@(15,XR2),IICNPM              MOVE PARM LIST TO THE ECB @15 08280000
L       II@PRL,XR1                         POINT XR1 AT PARAMETER LIST 08290000
SPACE                                       08300000
* NOW FILL IN THE NECESSARY LENGTHS AND ADDRESSES IN PARM LIST 08310000
SPACE                                       08320000
MVC     $LGOL(1,XR2),PLOUTL(,XR1) MOVE IN OUTPUT LENGTH @15 08330000
MVC     $LGOAA(2,XR2),PLRECA(,XR1) MOVE IN RECORD AREA ADDRESS@15 08340000
MVC     $LGON(1,XR2),PLINL(,XR1) MOVE IN INPUT LENGTH @15 08350000
MVC     $LGO@@(2,XR2),PLRECA(,XR1) MOVE IN RECORD AREA ADDRESS@15 08360000
MVI     $LGOTY(,XR2),IIWTO                 SET OP CODE TO WTO @15 08370000
TBN     PLOPC(,XR1),OPGET                  IS THIS A WTOR?      08380000
JF      IIDOIT                              BRANCH IF YES        08390000
MVI     $LGOTY(,XR2),IIWTOR                SET OP CODE TO WTOR @15 08400000
SPACE                                       08410000
* PARAMETER LIST IS NOW BUILT - ISSUE SVC TO DO WTO OR WTOR 08420000
SPACE                                       08430000
IIDOIT EQU *                               08440000
CCP     ENABLE,PMRQ                        ENABLE ALL INTERRUPTS 08450000
SVC     0                                  ISSUE THE SVC         08460000
DC      XL1'85'                             RIB FOR LOG          08470000
SPACE                                       08480000
* MESSAGE IS SENT OUT - RETURN TO THE USER 08490000
SPACE                                       08500000
CCP     DISABL,PMRQ                        DISABLE ALL INTERRUPTS 08510000
ST      II@PRL,XR1                         RESTORE PARAMETER LIST POINTER 08520000
SLC     PLEFFL(2,XR1),PLEFFL(,XR1) SET EFFEL TO ZERO 08530000
L       PLRECA(,XR1),XR2                    POINT TO THE RECORD AREA 08540000
CLI     0(,XR2),IINODA                      WAS ANY DATA ENTERED? 08550000
JE      IICDNE                              BRANCH OUT IF YES-EFFL = 0 08560000
MVC     PLEFFL(2,XR1),PLINL(,XR1) SAVE MAXIMUM INPUT LENGTH 08570000
A       PLINL(,XR1),XR2                     POINT TO THE          08580000
IICLOP A IIIMIN1,XR2                       END OF THE AREA      08590000
CLI     0(,XR2),IIBLNK                      IS THE CHARACTER A BLANK? 08600000
JNE     IICDNE                              BRANCH OUT IF YES    08610000
SLC     PLEFFL(2,XR1),X$0001                DECREMENT EFFL BY ONE 08620000
B       IICLOP                              LOOP TO KEEP TESTING 08630000
IICDNE LA 0(,XR1),XR2                       RESTORE PARAMETER LIST POINTER 08640000
B       IIFINI                              GO TO EXIT TO USER  08650000
EJECT                                       08660000
AIF     (&NDF).NMAP2                       SKIP IF *NO* DFF     08670000
SPACE 2                                    08680000
***** 08690000
* CHECK VALIDITY OF MAPPING REQUESTS * 08700000
***** 08710000

```

	SPACE			08720000
	ENTRY IICKMP			08730000
IICKMP	EQU *			08740000
	TBN PLOPC(,XR2),OPCOPY	IS THIS A COPY COMMAND?		08750000
	TBF PLOPC(,XR2),OPKPL	AND NOT AN ERASE OR RUF PUT ?		08760000
	JF IINCPY	BRANCH IF NOT		08770000
	TBN TUBTA1(,XR1),TASDFF	IS MAPPING SPECIFIED FOR TERM?		08780000
	JT IICKTN	BRANCH IF YES		08790000
	SPACE			08800000
*	ERROR - COPY TO 3270 REQUESTED BUT NO MAPPING FOR THIS TERMINAL		*	08810000
	SPACE			08820000
	B C4TI2	BRANCH TO TERMINATION		08830000
	DC AL1(TCCICA)	INVALID COPY - ATTRIBUTES		08840000
	SPACE			08850000
IICKTN	EQU *			08860000
	L @TNT,XR1	POINT TO FIRST TNT		08870000
	L PLRECA(,XR2),XR2	POINT TO RECORD AREA		08880000
IIAGIN	CLI 0(,XR1),STOPER	IS THIS END OF TNT'S?		08890000
	JNE IINSTP	BRANCH IF NOT		08900000
	SPACE			08910000
*	ERROR - SYMBOLIC TERMINAL NAME FOR COPY NOT FOUND IN TNT'S		*	08920000
	SPACE			08930000
	B C4TI2	BRANCH TO TERMINATION		08940000
	DC AL1(TCCICN)	INVALID COPY DUE TO NAME NF		08950000
	SPACE			08960000
IINSTP	CLC IISTNL-1(IISTNL,XR2),IISTNL-1(,XR1)	IS THIS TNT FOR COPY?		08970000
	JE IITNT	BRANCH IF YES		08980000
	LA TNTLN(,XR1),XR1	POINT TO NEXT TNT		08990000
	B IIAGIN	GO CHECK THIS ONE		09000000
	EJECT			09010000
IITNT	L TTTUB(,XR1),XR1	POINT TO COPY TUB		09020000
	L II@PRL,XR2	POINT TO PARAMETER LIST		09030000
	ST PL\$RTC(,XR2),XR1	PUT SECOND TUB @ IN PARM LIST		09040000
	CLI TUBPHY(,XR1),TUB5M1	IS THIE A 3275 MODEL 1?		09050000
	JE IIINCP	BRANCH IF YES		09060000
	CLI TUBPHY(,XR1),TUB5M2	IS THIS A 3275 MODEL 2?		09070000
	JE IIINCP	BRANCH TO HALT IF YES		09080000
	L PLTUBA(,XR2),XR1	POINT TO OTHER TUB		09090000
	CLI TUBPHY(,XR1),TUB5M1	IS THIE A 3275 MODEL 1?		09100000
	JE IIINCP	BRANCH IF YES		09110000
	CLI TUBPHY(,XR1),TUB5M2	IS THIS A 3275 MODEL 2?		09120000
	JNE IICMAP	BRANCH IF YES		09130000
IIINCP	EQU *	INVALID DEVICE FOR COPY		09140000
	SPACE			09150000
*	ERROR - COPY REQUEST INVALID WITH 3275		*	09160000
	SPACE			09170000
	B C4TI2	BRANCH TO TERMINALTION		09180000
	DC AL1(TCCICT)	COPY INVALID TO TERMINAL TYPE		09190000
	SPACE			09200000
IINCPY	EQU *			09210000
	TBN TUBTA1(,XR1),TASDFF	IS MAPPING SPECIFIED?		09220000
	JT IICMAP	BRANCH IF YES		09230000
	CLI PLOPC(,XR2),OPPUT+OPEAU	IS THIS AN ERASE		09240000
	JNE IINEAU	BRANCH IF NOT		09250000
	SPACE			09260000
*	ERROR - EAU REQUESTED BUT NO MAPPING SPECIFIED FOR THIS TERMINAL			09270000
	SPACE			09280000
	B C4TI2	BRANCH TO TERMINATION		09290000
	DC AL1(TCCIEA)	INVALID ERASE REQUEST		09300000
	SPACE			09310000

IINEAU	TBN	PLOPC(,XR2),OPPUT+OPMSG	IS THIS A PUT EOT?	09320000
	TBN	PLOPM(,XR2),OPLIST	IS THIS PUT OVERRIDES?	09330000
	AIF	(&NPBY).NBSY2	SKIP IF NO BUSY PRINTER	09332000
	JF	IIBUSY	BRANCH IF NOT	09334000
	AGO	.BSY1		09336000
.NBSY2	ANOP			09338000
	JF	IIUPST	BRANCH IF NOT	09340000
.BSY1	ANOP			09345000
	SPACE			09350000
*	ERROR - PUT OVERRIDES REQUESTED BUT NO MAPPING ON TERMINAL			* 09360000
	SPACE			09370000
	B	C4TI2	BRANCH TO TERMINATION	09380000
	DC	AL1(TCCIPA)	INVALID PUT OVERRIDES	09390000
	EJECT			09400000
IICMAP	EQU	*		09410000
	TBN	PLOPC(,XR2),OPPUT	IS THE OP A PUT?	09420000
	JF	IICLMP	BRANCH IF NOT	09430000
	TBN	PLOPC(,XR2),OPMSG	IS IT A PUT-EOT?	09440000
	JT	IICLMP	BRANCH IF YES	09450000
	TBF	PLOPC(,XR2),OPCOPY	IS THIS A COPY OR ERASE?	09460000
	JF	IICLMP	BRANCH PAST ERROR IF YES	09470000
	SPACE			09480000
*	ERROR - PUT REQUESTED TO MAPPING WITHOUT THE EOT MODIFIER			09490000
	SPACE			09500000
	B	C4TI2	BRANCH TO TERMINATION	09510000
	DC	AL1(TCCIPM)	INVALID PUT TO MAPPING	09520000
	SPACE			09530000
IICLMP	EQU	*		09540000
	B	IIMAPP	BRANCH TO MAPPING INTERFACE	09550000
	SPACE			09560000
	CLI	PLRTC(,XR2),RCOK	WAS MAPPING SUCCESSFUL?	09570000
	BNE	IISEND	BRANCH PAST CM CALL IF NOT	09580000
.NMAP2	ANOP			09590000
	AIF	(&NPBY).NBSY3	SKIP IF NO BUSY PRINTER	@16 09590300
	EJECT			@16 09590600
	SPACE 1			@16 09590900
*****				@16 09591200
*	BUSY PRINTER SUPPORT			* @16 09591500
*	TEST IF DEVICE IS BUSY, IF IT IS THEN DO A SYSTEM INVITE			* @16 09591800
*	TO TEST WHETHER THE DEVICE IS STILL BUSY.			* @16 09592100
*****				@16 09592400
	SPACE 1			@16 09592700
	ENTRY IIBUSY			@16 09593000
IIBUSY	EQU	*	BUSY PRINTER TEST.	@16 09593300
	L	II@PRL,XR2	RELOAD USERS PARM LIST.	@16 09593600
	TBN	TUBAT4(,XR1),TUBBSY	IS DEVICE BUSY?	@16 09593900
	JF	IIUPST	NO, GO TO POST	@16 09594200
	SPACE 1			@16 09594500
	L	NCTCB@,XR2	POINT TO USERS TCB	@16 09594800
	SBN	TUBAT2(,XR1),TUBIIS	SET ON INVITE BIT IN TUB	@16 09595100
	SBN	TUBAT4(,XR1),TUBWAT	INDICATE WAITING FOR STATUS	@16 09595400
	SLC	TCBPL+PLLEN-1(PLLEN,XR2),TCBPL+PLLEN-1(,XR2)	BLANK IT	@16 09595700
	MVI	TCBPL+PLOPM(,XR2),OP\$SYS	INDICATE A SYSTEM OPERATION	@16 09596000
	MVI	TCBPL+PLOPC(,XR2),OPINV	INDICATE AN INVITE OPERATION	@16 09596300
	ST	TCBPL+PLTUBA(,XR2),XR1	PUT TUB ADDRESS IN PARM LIST	@16 09596600
	LA	TCBPL(,XR2),XR2	POINT TO PARM LIST	@16 09596900
	J	IIPRBQ	GO QUEUE REQUEST.	@16 09597200
.NBSY3	ANOP			@16 09597500
	SPACE			09600000
	ENTRY IIUPST			09610000

IIUPST	EQU	*	PERFORM POST FOR USER TCB	09620000
	TBN	PLOPC(,XR2),OPGET	DID OP INVOLVE A GET OPERATION?	09630000
	JF	IINOI1	BRANCH OUT IF NOT	09640000
	SBN	TUBAT2(,XR1),TUBIIS	SET ON READ BIT SCHEDULED	09650000
	TBN	PLOPC(,XR2),OPPUT	DOES OPERATION INVOLVE PUT?	09660000
	JT	IINOI1	BRANCH IF YES	09670000
	SLC	PLEFFL(2,XR2),PLEFFL(,XR2)	CLEAR INPUT LENGTH FIELD	09680000
IINOI1	EQU	*		09690000
	EJECT			09700000
	SPACE	2		09710000
**		NOW MOVE THE PARAMETER LIST TO THE TCB FOR CM TO REFERENCE		09720000
	SPACE	2		09730000
	L	NCTCB@,XR1	POINT TO THE TCB	09740000
	MVC	TCBPL+PL\$RTC(16,XR1),PL\$RTC(,XR2)	MOVE THE PARM LIST IN	09750000
	SLC	TCBPL+PLECB+2(3,XR1),TCBPL+PLECB+2(,XR1)	CLEAR THE ECB	09760000
	MVC	TCBWK(2,XR1),PLRECA(,XR2)	SAVE RECORD AREA ADDRESS IN TCB	09770000
	SLC	TCBPL+PLRECA(2,XR1),TCBPL+PLRECA(,XR1)	CLEAR RECA POINTER	09780000
	LA	TCBPL(,XR1),XR2	POINT XR2 AT PARAMETER LIST	09790000
IIPRBQ	EQU	*		09795000 @16
	B	IIQSUB	GO QUEUE THE LIST ON @PRLQ	09800000
	L	II@PRL,XR2	POINT TO PARM LIST TO SAVE IT	09810000
	SBF	\$FLGC,#NTRAC	ENABLE TRACE BEFORE CM POST	09820000
	SPACE	3		09830000
	CCP	ENABLE,PMRQ	REENABLE INTERRUPTS	09840000
	SPACE			09850000
*		NOW POST \$CC4CM THAT HE HAS A TP REQUEST TO HANDLE		09860000
	SPACE			09870000
	LA	\$CMECB,XR1	POINT TO CM'S ECB	09880000
	SVC	0	SUPERVISOR CALL	09890000
	DC	XL1'05'	POST	09900000
	SPACE			09910000
*		NOW ISSUE A WAIT FOR COMPLETION OF THE TP REQUEST		09920000
	SPACE			09930000
IIDOWT	EQU	*	ISSUE THE WAIT	09940000
	L	NCTCB@,XR1	POINT TO THE TCB	09950000
	LA	TCBPL+PLECB(,XR1),XR1	LOAD POINTER TO THE ECB	09960000
	SVC	0	SUPERVISOR CALL	09970000
	DC	XL1'03'	WAIT FOR CM'S POST	09980000
	CCP	DISABL,PMRQ	DISABLE ALL INTERRUPTS	09990000
	ST	II@PRL,XR2	RESTORE PARAMETER LIST POINTER	10000000
	L	NCTCB@,XR2	POINT TO THE TCB	10010000
	LA	TCBPL(,XR2),XR2	POINT TO THE PARAMETER LIST	10020000
	AIF	(&NPBY).NBSY4	SKIP IF NO BUSY PRINTER	@16 10021000
	SPACE	1		@16 10022000
	L	PLTUBA(,XR2),XR1	POINT TO TUB	@16 10023000
	CLI	PLOPM(,XR2),OP\$SYS	IF SYSTEM OPERATION, MUST BE	@16 10024000
	BE	IIBUSY	FOR BUSY PRINTER, SO LOOP	@16 10025000
.NBSY4	ANOP			@16 10026000
	SPACE			10030000
.*				10040000
	AIF	(&NDF).REND	. SKIP IF *NO* DFF SUPPORT	10050000
.*				10060000
	LA	DFQ030,XR1	POINT TO ACCEPT MAPPING ENTRY	10070000
	ST	IIIIIII+3,XR1	SET IN BRANCH ADDRESS	10080000
	B	IIMAPP	GO TO MAPPING INTERFACE	10090000
	LA	DFA000,XR1	POINT TO NORMAL ENTRY POINT	10100000
	ST	IIIIIII+3,XR1	RESET MAPPING BRANCH ADDRESS	10110000
.REND	ANOP			10120000
	EJECT			10130000
	SPACE	5		10140000

```

***** 10150000
* REENTER $CC4II AT THIS POINT AFTER USER REQUEST SCHEDULED 10160000
***** 10170000
    SPACE 5 10180000
    ENTRY IIREND 10190000
IIREND EQU * RETURN TO USER PROGRAM 10200000
    SPACE 2 10210000
***** 10220000
* NOW MOVE THE PARAMETER LIST BACK TO USER PROGRAM AREA * 10230000
***** 10240000
    SPACE 2 10250000
    L II@PRL,XR1 POINT TO USER'S PARAMETER LIST 10260000
    MVC PL$RTC(6,XR1),PL$RTC(,XR2) MOVE THE PARAMETER LIST 10270000
    MVC PLINL(8,XR1),PLINL(,XR2) BACK TO THE USER PROGRAM 10280000
IIUT LA 0(,XR1),XR2 RESTORE PARM LIST IN XR2 10290000
    SPACE 2 10300000
IIFINI L PLTUBA(,XR2),XR1 POINT XR1 AT TUB 10310000
    SBF TUBAT3(,XR1),TUBSPF+TUBSWL ASSURE STOP FAILED AND SWALLOW 10320000
* BITS IN TUB ARE OFF 10330000
    EJECT 10340000
IISEND EQU * RETURN 10350000
    SLC PLRECA(2,XR2),CC0006 RESTORE CORRECT PLRECA 10360000
IIAEND EQU * 10370000
.* 10380000
    AIF (&NDME).NDME1 . SKIP IF *NO* DME SUPPORT 10390000
.* 10400000
    SPACE 10410000
***** 10420000
* CANCEL REQUEST ANALYSIS * 10430000
***** 10440000
    SPACE 10450000
* XR2 WILL POINT XT THE PARM LIST (GET OR INVITE INPUT) 10460000
* IF CANCEL HAS BEEN REQUESTED BY THE TERMINAL OPERATOR - CALL IN 10470000
* TRANSIENT TO DO HIS THING 10480000
    SPACE 10490000
    L PLTUBA(,XR2),XR1 POINT XR1 AT THE TUB 10500000
    TBN TUBAT3(,XR1),TUBCNC HAS CANCEL BEEN REQUESTED 10510000
    JF IITEND JUMP IF NO CANCEL 10520000
    SPACE 10530000
* CANCEL HAS BEEN REQUESTED - BRING IN TRANSIENT 10540000
    SPACE 10550000
    B CC4PI 10560000
    DC AL1(CC4NC) CANCEL TRANSIENT FOR $CC4II 10570000
.* 10580000
.NDME1 ANOP 10590000
.* 10600000
    SPACE 6 10610000
    ENTRY IITEND 10620000
IITEND EQU * LEAVING SOON 10630000
    B CC4TT BRANCH TO TRACE RETURN TO USER 10640000
    DC AL1(TTIIRT) TRACE ID - 'EE' 10650000
    CCP ENABLE,PMRQ REENABLE ALL INTERRUPTS 10660000
    SPACE 10670000
* NOW RETURN TO THE USER PROGRAM VIA A SUPERVISOR EXIT 10680000
    SPACE 10690000
    SVC 0 SUPERVISOR CALL 10700000
    DC XL1'08' RIB TO EXIT 10710000
    EJECT 10720000
***** 10730000
* HAVE ACCEPT INPUT REQUEST * 10740000

```

```

***** 10750000
SPACE 10760000
IIACI EQU * HAVE ACCEPT INPUT 10770000
      B CC4TT BRANCH TO TRACE THE ACCEPT INPUT 10780000
      DC AL1(TTII) '0C' TRACE ID 10790000
SPACE 10800000
      AIF (&NRAI).NRES 10800080
***** 10800160
*-----RESIDENT ACCEPT INPUT IS BEING USED-----* 10800240
SPACE 2 10800320
      L NCTCB@,XR1 CURRENT TASK 10800400
      CLI PLOPM(,XR2),NOBIT NO OP-MODIFIER BITS ALLOWED 10800480
      TBF PLOPC(,XR2),BIT0+BIT1+BIT2+BIT3 HIGH FOUR BITS OFF 10800560
      JC IIVINL,X'16' OK CONTINUE 10800640
***** 10800720
*OP-CODE ERROR MUST BE X'04' ONLY * 10800800
***** 10800880
      B C4TI2 ERROR IN OP CODE 10800960
      DC AL1(TCCIOP) TERMINATE USER 10801040
*-----OP IS VALID CHECK LENGTH-----* 10801120
IIVINL CLC PLINL(2,XR2),X$0000 INPUT LENGTH > 0 10801200
      JH IITSCT YES CONTINUE 10801280
      B C4TI2 NO 10801360
      DC AL1(TCCIIL) TERMINATION CODE 10801440
IITSCT CLI TCBIIC(,XR1),0 ANY INVITES 10801520
      JH IICTOK YES CONTINUE 10801600
      TBN TCBDMG(,XR1),TCBNEP ELSE MUST BE A NEP PROGRAM 10801680
      JT IINEP OK 10801760
SPACE 2 10801840
*-----NO INVITES TERMINATE-----* 10801920
      B C4TI2 TERMINATION 10802000
      DC AL1(TCCIAN) ERROR TYPE 10802080
IINEP CLC TCBMAX(1,XR1),TCBATR(,XR1) MAX TERMINAL COUNT @13 10802160
      JH IIOK NO. GO WAIT FOR ACCEPT. @13 10802240
***** 10802320
*ALREADY AT MAXIMUM TERMINAL COUNT * 10802400
***** 10802480
      B C4TI2 TERMINATION 10802560
      DC AL1(TCCIAC) ERROR TYPE 10802640
SPACE 3 10802960
*-----RE-ENTRY POINT AFTER ACCEPT COMPLETE -----* 10803040
*-----HAS BEEN POSTED CHECK FOR TUBS ON QUEUE-----* 10803120
SPACE 2 10803200
IICTOK LA TCBINQ(,XR1),XR2 START OF TERMINAL CHAIN 10803280
      ST IITPRV,XR2 SAVE TERMINAL ADDR FOR DEQUE 10803360
      CLI TCBINQ-1(,XR1),0 ANY TUBS OP-ENDED YET 10803440
      JE IIGGWT NO GO WAIT ON ACCEPT 10803520
      L TCBINQ(,XR1),XR2 FIRST TUB 10803600
IITSTG TBN TUBAT2(,XR2),TUBIMI IMPLICIT INVITE 10803680
      JF IIHERD CHECK DFF ON ACCEPT INPUT 10803760
      TBF TCBDMG(,XR1),TCBMTS MULTIPLE REQUESTOR 10803840
      JT IIHERE NO USE TUB 10803920
      CLC TCBMAX(1,XR1),TCBATR(,XR1) YES CAN WE USE IT 10804000
      JNH IINXTB ONLY IF MAX GREATER 10804080
      TBN TUBAT1(,XR2),TUBKNM CONSOLE REQUEST 10804160
      JT IIHERE YES 10804240
      ALC TCBATR(1,XR1),X$0001 NO BUMP ACTIVE TERMINAL COUNT 10804320
SPACE 2 10804400
*-----HAVE A TUB CHECK DFF AND RUF USE-----* 10804480
IIHERE TBN TUBSCS(,XR2),TUBRUF RUF ACTIVE ON REQUEST 10804560

```

IIHERD	CLI	TCB@AS-1(,XR1),0	DFE ACTIVE FOR REQUEST	10804640
	L	II@PRL,XR1	PARM LIST ADDRESS	10804720
	MVI	PL\$OPC(,XR1),BRNOP	NO-OP BRANCH TO MAPPING	10804800
	JC	IIGISF,X'91'	NO DFE OR RUF INPUT	10804880
	MVI	PL\$OPC(,XR1),BR97	SET FOR DFE MAPPING	10804960
	J	IIGISF	GO EXIT ACCEPT	10805040
IINXTB	CLI	TUBINQ-1(,XR2),0	MORE TUBS	10805120
	JE	IIGGWT	NO GO WAIT ON ACCEPT	10805200
	LA	TUBINQ(,XR2),XR1	TUB POINTER	10805280
	ST	IITPRV,XR1	SAVE FOR DEQUE	10805360
	L	TUBINQ(,XR2),XR2	NEXT TUB	10805440
	L	NCTCB@,XR1	OUR USER	10805520
	B	IITSTG	TRY NEXT TUB	10805600
	SPACE	2		10805680
	-----	SET UP FOR ACCEPT INPUT WAIT-----		10805760
IIGGWT	L	II@PRL,XR2	RESTORE PARM LIST ADDRESS	10805840
	MVC	PL\$RTC(2,XR2),IITPRV	PRIOR TUB ADDRESS	10805920
IITUBS	L	TCBTUB(,XR1),XR1	FIRST USER TUB	10806000
IILOOP	TBN	TUBAT2(,XR1),TUBIIS	INVITE	10806080
	JF	IINSAV	NO SKIP IT	10806160
	TBN	TUBCHR(,XR1),TUBLNE	BSCA TUB	10806240
	JF	IIOK	NO IT'S OK	10806320
	TBN	TUBAT2(,XR1),TUBOWN	DOES IT OWN A LLINE	10806400
	JT	IIOK	YES OK	10806480
	L	TUBDTF(,XR1),XR2	DTF ADDRESS	10806560
	MVC	IILINE,0(1,XR2)	SAVE LINE NUMBER	10806640
	ST	IISAVE+3,XR1	SAVE TUB POINTER	10806720
	L	NCTCB@,XR1	CURRENT TCB	10806800
	L	TCBTUB(,XR1),XR1	FIRST TUB	10806880
IIIGIN	TBN	TUBAT2(,XR1),TUBOWN	OWN A LINE	10806960
	JF	IIANTH	NO DO NEXT	10807040
	L	TUBDTF(,XR1),XR2	DTF	10807120
	CLC	0(1,XR2),IILINE	OWN OUR LINE	10807200
	JE	IINEXT	YES	10807280
IIANTH	CLI	TUBTUB-1(,XR1),0	MORE TUBS	10807360
	JE	IIOK	NO	10807440
	L	TUBTUB(,XR1),XR1	NEXT TUB POINTER	10807520
	B	IIIGIN	GO CHECK IT	10807600
IINEXT	EQU	*	CHECK OTHER TUBS	10807680
IISAVE	LA	*-*,XR1	RELOAD TUB POINTER	10807760
IINSAV	CLI	TUBTUB-1(,XR1),0	MORE TUBS	10807840
	JNE	IIMORE	YES CHECK THEM	10807920
	B	C4TI2	TERMINATION	10808000
	DC	AL1(TCCIAC)	USER ERROR CODE	10808080
IIMORE	L	TUBTUB(,XR1),XR1	NEXT TUB	10808160
	B	IILOOP	CHECK NEXT TUB	10808240
IIGISF	L	II@PRL,XR1	RESTORE PARM LIST ADDRESS	10808320
	MVC	PL\$RTC(2,XR1),IITPRV	SET TO DEQUE TUB	10808400
	ST	PLTUBA(,XR1),XR2	SET TUB ADDRESS IN PARM LIST	10808480
	LA	0(,XR1),XR2	SET PARM LIST ADDRESS	10808560
	J	IIGGSF	ACCEPT SATISFIED	10808640
	SPACE	2		10808720
	-----	SET UP FOR ACCEPT INPUT WAIT FOR USER-----		10808800
IIOK	L	NCTCB@,XR1	CURR USER	10808880
	L	II@PRL,XR2	RELOAD TUB ADDRESS	10808960
	MVI	TCBECB(,XR1),TCBACW	ACCEPT WAIT INDICATOR FOR C.M.	10809040
	AGO	.RESA		10809120
.NRES	ANOP			10809200
	MVI	PL\$OPC(,XR2),NOBIT	SETP OP TO SHOW INITIAL REQUEST	10810000
IICALL	EQU	*		10820000

B	CC4PI	BRANCH TO TRANSIENT HANDLER	10830000
DC	AL1(CC4AB)	ID FOR ACCEPT TRANSIENT	10840000
B	C4TI2	BRANCH TO TERMINATION ON ERROR	10850000
DC	AL1(*-*)	ACCEPT IMPOSSIBLE TO SATISFY	10860000
SPACE			10870000
J	IIGGSF	JUMP IF ACCEPT SATISFIED	10880000
SPACE	5		10890000
AGO	.RESA		10893000
.RESA	ANOP		10896000
*	DETERMINE NOW WHETHER USER SHOULD BE TOLD ABOUT SHUTDOWN REQUEST		10900000
SPACE			10910000
*****			10920000
*	CHECK FOR SHUTDOWN	*	10930000
*****			10940000
SPACE			10950000
*	DETERMINE IF SYSTEM OPERATOR REQUESTED SHUTDOWN AND		10960000
*	ALSO WHETHER USER HAS BEEN TOLD YET WITH NON-SHUTDOWN INQUIRY OP		10970000
SPACE			10980000
TBN	\$FLGA,CPSHUT	HAS SHUTDOWN BEEN REQUESTED	10990000
JF	IIAIWT	JUMP IF NOT	11000000
SPACE			11010000
*	SHUTDOWN HAS BEEN REQUESTED - DETERMINE IF USER TOLD ALREADY		11020000
SPACE			11030000
IITELL	L NCTCB@,XR1	POINT XR1 AT CURRENT TCB	11040000
TBN	TCBDMG(,XR1),TCBSHQ	HAS USER BEEN TOLD	11050000
SBN	TCBDMG(,XR1),TCBSHQ	SET BIT THAT USER TOLD	11060000
JF	IIPGSH	JUMP IF SHUTDOWN REQUESTED	11070000
EJECT			11080000
*****			11090000
*	WAIT FOR AN INVITE COMPLETION OR A SHUTDOWN REQUEST	*	11100000
*****			11110000
SPACE	3		11120000
IIAIWT	EQU *	ISSUE WAIT	11130000
CCP	DISABL,PMRQ	DISABLE ALL INTERRUPTS	11140000
SPACE			11150000
*	IF A NEP MRT WAITING - DO FILE SHARE DEQUEUE THING		11160000
SPACE			11170000
LA	0(,XR2),XR1	POINT XR1 AT PARM LIST	11180000
L	NCTCB@,XR2	POINT TO THE TCB	11190000
CLI	TCBIIC(,XR2),NOBIT	ANY INVITES OUTSTANDING?	11200000
JNE	IINODG	BRANCH IF YES	11210000
TBN	TCBDMG(,XR2),TCBNEP+TCBMTS	IS THIS A NEP MRT?	11220000
JF	IINODG	BRANCH IF NOT	11230000
SPACE			11240000
B	CC4PI	CALL TRANSIENT HANDLER FOR CC4DG	11250000
DC	AL1(CC4DG)	ID FOR \$CC4DG	11260000
SPACE			11270000
*	MUST DO A WAIT FOR AN INVITE OR A SHUTDOWN REQUEST		11280000
SPACE			11290000
IINODG	LA 0(,XR1),XR2	POINT XR2 AT PARM LIST	11300000
L	NCTCB@,XR1	POINT TO THE TCB	11310000
LA	TCBECB(,XR1),XR1	POINT TO THE ECB	11320000
SPACE			11330000
CCP	ENABLE,PMRQ	REENABLE ALL INTERRUPTS	11340000
SVC	0	ISSUE WAIT FOR INVITE COMPLETE	11350000
*		OR SHUTDOWN REQUEST	11360000
DC	XL1'03'	RIB TO WAIT ON ECB LIST	11370000
SPACE			11380000
*	WAIT HAS COMPLETED - TEST IF SHUTDOWN OR INVITE COMPLETE		11390000
SPACE			11400000

```

CCP  DISABL,PMRQ          DISABLE INTERRUPTS          11410000
ST   II@PRL,XR2          SAVE PARAMETER LIST  POINTER  11430000
EJECT                                     11440000
SPACE 3                                     11450000
*****                                     11460000
* RETURN AFTER BEING POSTED THAT A TUB HAS BEEN PUT IN THE TCB INVITE 11470000
* INPUT TUB QUEUE                                     11480000
* OR SHUTDOWN HAS BEEN REQUESTED BY THE SYSTEM OPERATOR * 11490000
*****                                     11500000
SPACE 3                                     11510000
* DETERMINE IF POST WAS FOR INVITE INPUT COMPLETE 11520000
* IF NOT MUST HAVE BEEN SHUTDOWN REQUESTED 11530000
SPACE                                     11540000
* GET CORRECT POST AND SET SKIP BITS IN SHUTDOWN ENTRY 11550000
SBF  0(,XR1),TCBACW      RESET ACCEPT WAIT BIT 11555000
CLI  2(,XR1),IIPOST     WAS POST FOR INVITE COMPLETE? 11560000
BNE  IITELL              JUMP IF NOT INVITE POST 11570000
SPACE                                     11580000
* HAD POST OF INVITE INPUT COMPLETE 11590000
AIF  (&NRAI).NRS1       RESIDENT ACCEPT 11601000
L    NCTCB@,XR1         SET TO CURRENT TASK 11602000
B    IICTOK             GO HANDLE OP-ENDED TUB 11603000
IITPRV DC XL2'00'       PRIOR TUB SAVE AREA 11604000
IILINE DC XL1'00'       BSCA LINE NUMBER SAVE AREA 11605000
AGO  .NRS2              11606000
.NRS1 ANOP              11607000
MVI  PL$OPC(,XR2),ALLBIT SET OP TO SHOW SECONDARY ENTRY 11610000
B    IICALL             BRANCH TO RECALL TRANSIENT 11620000
.NRS2 ANOP              11625000
SPACE                                     11630000
* TELL USER SHUTDOWN REQUESTED NOW 11640000
SPACE                                     11650000
IIPGSH EQU *           SHUTDOWN HAS BEEN REQUESTED 11660000
MVI  PLRTC(,XR2),RCXSHD PLUG SHUTDOWN REQUESTED RTN CODE 11670000
B    IITEND             BR TO RETURN TO USER 11680000
EJECT                                     11690000
SPACE 2                                     11700000
*****                                     11710000
* ACCEPT INPUT SATISFIED * 11720000
*****                                     11730000
SPACE 2                                     11740000
IIGGSF EQU *           GET GENERAL SATIFIED 11750000
CCP  DISABL,PMRQ          DISABLE ALL INTERRUPTS 11760000
ST   II@PRL,XR2          STORE PARAMETER LIST ADDRESS 11770000
L    PLTUBA(,XR2),XR2    POINT AT TUB 11780000
L    NCTCB@,XR1         POINT XR1 AT CURRENT TCB 11790000
SLC  TCBIIC(1,XR1),X$0001 SUBTRACT 1 FROM INVITE COUNT 11800000
SBF  TUBAT2(,XR2),TUBIIS+TUBIIQ SET OFF TUB INVITE STATUS BITS 11810000
SPACE 2                                     11820000
* DEQUEUE THE TUB FROM INVITE INPUTE TUB QUEUE 11830000
L    II@PRL,XR1         POINT TO PARAMETER LIST 11840000
L    PL$RTC(,XR1),XR1   POINT TO PREVIOUS TUB 11850000
MVC  0(2,XR1),TUBINQ(,XR2) DEQUEUE THE TUB 11860000
L    TUBTNT(,XR2),XR2  POINT TO THE TNT 11870000
L    NCTCB@,XR1         POINT TO THE TCB 11880000
MVC  NPJOB-2(6,XR1),TNTNAM(,XR2) MOVE TERM NAME TO TCB 11890000
EJECT                                     11900000
SPACE 2                                     11910000
*****                                     11920000
* NOW POST CM TASK TO PROCESS THE ACCEPT INPUT DATA * 11930000

```

```

***** 11940000
SPACE 2 11950000
* MOVE THE PARAMETER LIST TO THE TCB 11960000
SPACE 11970000
L II@PRL,XR2 POINT XR2 AT PARAMETER LIST 11980000
MVC TCBWK(2,XR1),PLRECA(,XR2) SET PLRECA IN TCBWK 11990000
LA TCBPL(,XR1),XR1 POINT TO PARM LIST AREA IN TCB 12000000
MVC PL$RTC(16,XR1),PL$RTC(,XR2) MOVE THE PARM LIST TO THE TCB 12010000
L PLTUBA(,XR1),XR2 POINT TO THE TUB 12020000
L TUBTNT(,XR2),XR2 POINT TO THE TNT ENTRY 12030000
L PLRECA(,XR1),XR1 POINT TO THE USER'S REC. AREA 12040000
MVC 5(6,XR1),5(,XR2) MOVE IN THE SYMB. TERM. NAME 12050000
SPACE 12060000
L II@PRL,XR2 RESTORE PARAMETER LIST POINTER 12070000
AIF (&NDF).PONK SKIP IF MAPPING *NOT* SUPPORTED 12080000
MVC IIDFF,PL$OPC(1,XR2) MOVE IN BRANCH CONDITION CODE 12090000
IIDFF EQU *+1 12100000
BC IIMAPP,BR97 BRANCH TO MAPPING 12110000
.PONK ANOP 12120000
ENTRY IINDFF 12130000
IINDFF EQU * 12140000
L NCTCB@,XR2 POINT TO THE TCB 12150000
SLC TCBPL+PLECB+2(3,XR2),TCBPL+PLECB+2(,XR2) CLEAR THE ECB 12160000
LA TCBPL(,XR2),XR2 POINT TO THE PARAMETER LIST 12170000
SLC PLRECA(2,XR2),PLRECA(,XR2) CLEAR PLRECA 12180000
SPACE 12190000
* QUEUE THE ACCEPT INPUT REQUEST ON CM'S REQUEST QUEUE 12200000
SPACE 12210000
B IIQSUB GO QUEUE THE REQUEST ON @PRLQ 12220000
L II@PRL,XR2 RESTORE PARAMETER LIST POINTER 12230000
SBF $FLGC,#NTRAC REENABLE TRACE BEFORE CM POST 12240000
SPACE 2 12250000
* NOW POST THE CM WITH A REQUEST TO PROCESS 12260000
SPACE 2 12270000
LA $CMECB,XR1 POINT XR1 AT CM'S ECB 12280000
CCP ENABLE,PMRQ REENABLE ALL INTERRUPTS 12290000
SVC 0 POST DFF WITH THE REQUEST 12300000
DC XL1'05' RIB FOR POST 12310000
SPACE 2 12320000
* CM IS NOW POSTED - WAIT ON COMPLETION OF THE ACCEPT INPUT 12330000
SPACE 12340000
L NCTCB@,XR1 POINT TO THE TCB 12350000
LA TCBPL+PLECB(,XR1),XR1 POINT AT THE ECB IN THE TCB PL 12360000
SVC 0 NOW WAIT ON THE POST OF THE ECB 12370000
DC XL1'03' RIB FOR EXPLICIT WAIT 12380000
EJECT 12390000
SPACE 2 12400000
* WE ARE POSTED COMPLETE - MOVE THE PARM LIST BACK AND RETURN 12410000
SPACE 2 12420000
CCP DISABL,PMRQ DISABLE INTERRUPTS 12430000
L NCTCB@,XR1 POINT XR1 AT THE TCB 12440000
MVC PL$RTC(6,XR2),TCBPL+PL$RTC(,XR1) MOVE THE PARM LIST 12450000
MVC PLINL(8,XR2),TCBPL+PLINL(,XR1) BACK TO THE UPA 12460000
SPACE 12470000
B IIAEND BRANCH TO RETURN TO REQUESTOR 12480000
.* 12490000
.NORS AIF (&NOM).MAPP . SKIP IF *NO* MLTA SUPPORT 12500000
.* 12510000
EJECT 12520000
***** 12530000

```

```

* MCT CHECK FOR INPUT AND OUTPUT CAPABILITY * 12540000
***** 12550000
SPACE 12560000
* XR1 POINTS AT TUB 12570000
* XR2 POINTS AT PARM LIST 12580000
SPACE 12590000
IIMCT ENTRY IIMCT 12600000
EQU * 1050 CHECK 12610000
ST IIMCTR+3,ARR SAVE ARR 12620000
SPACE 12630000
* DOES THIS OP INVOLVE MULTI-COMPONENT TERMINAL 12640000
SPACE 12650000
TBN TUBCHR(,XR1),TUBMCT IS THIS MCT 12660000
JF IIMCTR JUMP IF NOT TO RETURN NOW 12670000
SPACE 12680000
* HAVE MCT - SO CHECK COMPONENT INDICES 12690000
SPACE 12700000
CLI PL$MCT(,XR2),NOBIT ARE COMPONENT INDICES IN PARM L. 12710000
JNE IIMCTC JUMP IF ALREADY IN PARM L. 12720000
MVC PL$MCT(1,XR2),TUBPCS(,XR1) PLUB MCT INDICES INTO PARM LIST 12730000
SPACE 12740000
IIMCTC EQU * CHECK MCT INDICES 12750000
* -----START-----@06 12752000
TBF PLOPM(,XR2),OP$SYS SYSTEM OPERATION AND 12754000
* -----END-----@06 12756000
TBN PLOPC(,XR2),OPPUT DOES OP INVOLVE PUT 12760000
TBF PL$MCT(,XR2),MCTOUT IS MCT OUTPUT CAPABLE 12770000
JF IIMCTG JUMP IF OUTPUT CAPABLE OR NO PUT 12780000
SPACE 12790000
* ERROR - SUB TERMINAL NAME SPECIFIED FOR OUTPUT BUT NO 12800000
* OUTPUT DEVICES ASSOCIATED WITH THIS NAME 12810000
SPACE 12820000
B C4TI2 BRANCH TO TERMINATION 12830000
DC AL1(TCCIOC) I/O CAPABILITY - NO MATCH 12840000
SPACE 12850000
IIMCTG EQU * CHECK INPUT 12860000
* -----START-----@06 12862000
TBF PLOPM(,XR2),OP$SYS SYSTEM OPERATION AND 12864000
* -----END-----@06 12866000
TBN PLOPC(,XR2),OPGET DOES OP INVOLVE GET 12870000
TBF PL$MCT(,XR2),MCTIN IS MCT INPUT CAPABLE 12880000
JF IIMCTR JUMP IF INPUT CAPABLE OR NO GET 12890000
SPACE 12900000
* ERROR - SUB TERMINAL NAME SPECIFIED FOR INPUT OPERATION BUT NO 12910000
* INPUT DEVICE ASSOCIATED WITH THIS NAME 12920000
SPACE 12930000
B C4TI2 BRANCH TO TERMINATION 12940000
DC AL1(TCCIOC) I/O CAPABILITY - NO MATCH 12950000
SPACE 12960000
IIMCTR B #### RETURN 12970000
EJECT 12980000
.* 12990000
.MAPP ANOP 13000000
AIF (&NDF).CKIN . SKIP IF *NO* DFF SUPPORT 13010000
.* 13020000
***** 13030000
* INTERFACE TO MAPPING MODULE - XR2 POINTS AT PARAMETER LIST * 13040000
***** 13050000
SPACE 13060000
ENTRY IIMAPP 13070000

```

IIMAPP	EQU	*	INTERFACE TO MAPPING	13080000
	ST	IIRTRN+3,ARR	SAVE RETURN ADDRESS	13090000
	L	PLTUBA(,XR2),XR1	POINT TO TUB	13100000
	TBN	TUBTA1(,XR1),TASDF	IS MAPPING FOR THIS TERMINAL	13110000
	JF	IIRTRN	BRANCH IF NOT MAPPING	13120000
	L	NCTCB@,XR1	POINT TO CURRENT TCB	13130000
	CLI	TCB@AS-1(,XR1),NOBIT	IS MAPPING FOR THIS TASK?	13140000
	L	PLTUBA(,XR2),XR1	RESTORE TUB POINTER	13150000
	JNE	IIXXXX	BRANCH IF YES	13160000
	TBN	PLOPC(,XR2),OPSTAT	IS THIS A STATUS REQUEST?	13170000
	JT	IIRTRN	BRANCH TO RETURN IF YES	13180000
	SPACE	2		13190000
*---	ERROR---	DF	TERMINAL REFERENCED BY NON-DF	13200000
	SPACE	2		13210000
IIDFER	EQU	*		13220000
	B	C4TI2	BRANCH TO TERMINATION	13230000
	DC	AL1(TCCNDF)		13240000
IIXXXX	EQU	*		13250000
	L	IIRTRN+3,XR1	SAVE ARR IN XR1	13260000
IIIIII	B	DFA00	BRANCH TO MAPPING MODULE	13270000
	ST	IIRTRN+3,XR1	STORE RETURN ADDRESS IN BRANCH	13280000
	L	NCTCB@,XR1	POINT TO THE TCB	13290000
	L	TCB@AS(,XR1),XR1	POINT TO THE PAS	13300000
	MVC	II@PRL,PASPL@(2,XR1)	RESTORE PARM LIST SAVE AREA	13310000
	L	PLTUBA(,XR2),XR1	RESTORE TUB POINTER	13320000
IIRTRN	B	*-*		13330000
	EJECT			13340000
.CKIN	ANOP			13350000
	SPACE	2		13360000
*****				13370000
*		QUEUE THE REQUEST ON THE @PRLQ IN CCP COMMON AREA	*	13380000
*****				13390000
	SPACE	2		13400000
	ENTRY	IIQSUB		13410000
IIQSUB	ST	IIQRET+3,ARR	SAVE RETURN ADDRESS	13420000
	L	PLTUBA(,XR2),XR1	POINT TO THE TUB	13430000
*			-----START-----@09	13432000
	TBN	TUBAT1(,XR1),TUBKNM	THIS THE CONSOLE TUB?	13434000
	JT	IICNCK	SKIP UPDATING REQUEST COUNTER	13436000
*			-----END-----@09	13438000
	L	TUBLCB(,XR1),XR1	POINT TO THE LCB	13440000
	ALC	LCBNW#(1,XR1),X\$0001	UPDATE THE NEW REQUEST COUNTER	13450000
IICNCK	LA	@PRLQ-1,XR1	POINT TO THE QUEUE CHAIN FIELD	13460000
IIRECK	CLI	PLCHN-1(,XR1),NOBIT	IS THIS THE END OF THE CHAIN?	13470000
	JE	IIQIT	BRANCH TO QUEUE IF YES	13480000
	L	PLCHN(,XR1),XR1	POINT TO THE NEXT LIST IN CHAIN	13490000
	B	IIRECK	LOOP TO LOOK FOR LAST ONE	13500000
	SPACE			13510000
*		HAVE FOUND THE END OF THE CHAIN - QUEUE THIS REQUEST ON THE END		13520000
	SPACE			13530000
IIQIT	ST	PLCHN(,XR1),XR2	ADD THE REQUEST ON THE END	13540000
	SLC	PLCHN(2,XR2),PLCHN(,XR2)	CLEAR THE CHAIN POINTER	13550000
IIQRET	B	*-*	RETURN	13560000
	TITLE	'COMMUNICATIONS&#.I/O&#.INTERFACE&#.--&#.SYSTEM&#.I/O'		13570000
*****				13580000
*		ENTRY POINT FOR I/O INTERFACE FOR THE SYSTEM		13590000
*****				13600000
	SPACE			13610000
	ENTRY	\$CC4IS		13620000
\$CC4IS	EQU	*	I/O INTERFACE FOR SYSTEM	13630000

	CCP	DISABL,PMRQ	DISABLE ALL INTERRUPTS	13640000
	A	X\$0001,ARR	ADD 1 TO ARR TO POINT AT RIGHT	13650000
*			END OF PARM LIST ADDR OF ADDR	13660000
	ST	MOVE+5,ARR	STORE ADDR OF PARM LIST ADDR IN	13670000
*			MVC INSTRUCTION	13680000
	A	X\$0001,ARR	BUMP ARR TO NSI	13690000
		SPACE		13700000
* MOVE		PARM LIST ADDR TO II@PRL		13710000
		SPACE		13720000
MOVE	MVC	II@PRL(2),*-*	MOVE PARM LIST ADDR TO II@PRL	13730000
		SPACE		13740000
*		SAVE THE REGISTERS		13750000
		SPACE		13760000
	ST	IICARR,XR2	SAVE XR2	13770000
	L	NCTCB@,XR2	POINT TO THE TCB	13780000
	ST	IIIX1(,XR2),XR1	SAVE XR1	13790000
	ST	IIIR(,XR2),ARR	SAVE THE ARR	13800000
	MVC	IIIX2(2,XR2),IICARR	SAVE XR2	13810000
		SPACE		13820000
		*****		13830000
*		POST CM WITH TP IO REQUEST		13840000
*		WAIT		13850000
*		RETURN TO INVOKER		13860000
		*****		13870000
		SPACE 2		13880000
	L	II@PRL,XR2	POINT XR2 AT PARM LIST	13890000
		SPACE		13900000
	SLC	PLCHN(2,XR2),PLCHN(,XR2)	CLEAR CHAIN POINTER IN PARM LIST	13910000
	SLC	PL\$RTC(4,XR2),PL\$RTC(,XR2)	ZERO INTERNAL RETURN CODE	13920000
*			AND INTERNAL OP-CODE	13930000
	TBF	PLOPC(,XR2),OPPUT	OPERATION INVOLVE PUT ?	13940000
	JF	IS0010	YES, BRANCH	13950000
	MVI	PLRECA-1(,XR2),NOBIT	ZERO OUT RECORD ADDRESS	13960000
IS0010	EQU	*		13970000
		SPACE		13980000
	L	PLTUBA(,XR2),XR1	POINT XR1 AT TUB	13990000
	B	CC4TT	BRANCH TO TRACE ROUTINE	14000000
	DC	AL1(TTIS)	TRACE FOR \$CC4IS	14010000
		SPACE		14020000
*		TAKE A TRACE AT THIS POINT		14030000
		EJECT		14040000
		*****		14050000
*		HAVE TPIO REQUEST FOR A TERMINAL		14060000
*		POST \$CC4CM AND ISSUE WAIT	*	14070000
		*****		14080000
		SPACE		14090000
*POST		CM FOR TP REQUEST		14100000
		SPACE		14110000
ISREQ	EQU	*	HAVE TP REQUEST FOR \$CC4CM	14120000
		SPACE		14130000
*		IF THIS IS AN INVITE OPERATION, MOVE PARM LIST TO THE TUB		14140000
		SPACE		14150000
	TBN	PLOPM(,XR2),OP\$SYS	IS THIS A SYSTEM REQUEST, AND	14160000
	TBN	PLOPC(,XR2),OPINV	IS THIS AN INVITE OPERATION?	14170000
	TBF	PLOPC(,XR2),OPJRSH	AND NOT JUST RESCHEDULED? @11	14175000
	JF	ISWTOP	BRANCH IF NOT BOTH TRUE	14180000
*			-----START-----@08	14181000
	TBF	TUBAT2(,XR1),TUBDTA	TERMINAL IN DME?	14182000
	TBN	\$FLGA,CPSHUT	. AND IS SHUTDOWN PENDING?	14183000
	JF	ISNDQI	IF BOTH ARN'T TRUE SKIP RESET	14184000

```

SBF PLOPC(,XR2),OPINV      DESCHEDULE THE SYSTEM INVITE      14185000
TBF PLOPC(,XR2),OPPUT      TEST IF INVITE ONLY OPERATION@12 14185300
JT  ISREGS                  NO-OPP INVITE IF TRUE.             @12 14185600
J   ISWTOP                  GO RESTORE TUB POINTER           14186000
*                               -----END-----@08 14187000
ISNDQI MVC TUBPL+PL$RTC(16,XR1),PL$RTC(,XR2) MOVE PARM LIST TO TUB 14190000
LA   TUBPL(,XR1),XR2       POINT XR2 AT TUB PARM LIST           14200000
SPACE                                     14210000
ISWTOP L PLTUBA(,XR2),XR1   RESTORE TUB POINTER IN XR1       14220000
.*                                     14230000
AIF (&N1050).NMCT9         . SKIP IF *NO* 1050 SUPPORT       14240000
.*                                     14250000
SPACE                                     14260000
*   FIRST PLUG PL$MCT IS THIS IS MULTI-COMPONENT TERMINAL       14270000
SPACE                                     14280000
B   IIMCT                   FOR MCT (1050) ONLY           14290000
.*                                     14300000
.NMCT9 ANOP                 . CONTINUE HERE               14310000
.*                                     14320000
SPACE                                     14330000
ENTRY ISREQ1
ISREQ1 EQU *                14340000
*                                     14350000
*   -----START-----@07 14352000
TBF PLOPC(,XR2),OPJRSH     RESCHEDULE ONLY REQUEST AND     14354000
*   -----END-----@07 14356000
TBN PLOPC(,XR2),OPGET      DOES OP INVOLVE GET             14360000
TBF PLOPM(,XR2),OPSTOP     BUT NOT STOP OP                 14370000
JF  ISMVPS                 JUMP IF GET NOT INVOLVED       14380000
SPACE                                     14390000
*   OPERATION TO TERMINAL INVOLVES GET SO SET ON TUBIIS BIT     14400000
SPACE                                     14410000
SBN TUBAT2(,XR1),TUBIIS    SET ON READ SCHEDULED BIT       14420000
SPACE                                     14430000
ISMVPS EQU *                14440000
AIF (&NCPUM EQ '0').MSG1   SKIP IF CPU S MSGS. ALLOWED @17 14445000
CLI TUBPHY(,XR1),TUBCPU    IS THE IO TO A CPU?           14450000
JNE ISPOST                 GO TO POST CM IF NOT           14460000
TBF PLOPM(,XR2),OPSTOP+OPDISC STOP INVITE OR DISCONNECT? 14470000
JF  ISPOST                 JUMP IF EITHER               14480000
L   PLRECA(,XR2),XR1       POINT TO RECORD AREA           14490000
CLI 0(,XR1),SMESG         IS THIS AN 'S' MESSAGE TO A CPU? 14500000
JNE ISPOST                 BRANCH TO POST IF NOT           14510000
SBF PLOPC(,XR2),OPPUT      SET TO IGNORE THE PUT           14520000
MVI PLRECA-1(,XR2),NOBIT   DO NOT LET CM FREEMAIN AREA @10 14525000
TBN PLOPC(,XR2),OPGET      IS A GET TO BE DONE?           14530000
JF  ISREGS                 SKIP POST AND WAIT IF YES       14540000
.MSG1 ANOP                 @17 14545000
EJECT                                     14550000
SPACE 2                                  14560000
***** 14570000
*   QUEUE THE PARAMETER LIST ON THE APPROPRIATE LCB CHAIN       14580000
***** 14590000
SPACE 2                                  14600000
ISPOST EQU *                14610000
B   IIQSUB                  BRANCH TO QUEUE THE LCB         14620000
SPACE 2                                  14630000
***** 14640000
*   BRANCH TO POST ROUTINE                                       14650000
***** 14660000
SPACE                                     14670000

```

MVI	PLECB(,XR2),NOBIT	CLEAR ECB FLAG BYTE	14680000
CCP	ENABLE,PMRQ	REENABLE INTERRUPTS	14690000
SBF	\$FLGC,#NTRAC	ALLOW TRACE	14700000
LA	\$CMECB,XR1	POINT TO CM'S ECB	14710000
SVC	0	SUPERVISOR CALL	14720000
DC	XL1'05'	RIB FOR POST	14730000
SPACE	3		14740000
* ISSUE WAIT			14750000
SPACE	3		14760000
*****			14770000
* W A I T *			14780000
*****			14790000
SPACE	2		14800000
TBN	PLOPM(,XR2),OP\$SYS	IS THIS REQUEST TO BE WAITED ON?	14810000
JF	ISREGS	BRANCH IF NOT	14820000
LA	PLECB(,XR2),XR1	POINT TO THE ECB	14830000
ISWAIT SVC	0	SUPERVISOR CALL	14840000
DC	XL1'03'	RIB FOR WAIT	14850000
TBN	0(,XR1),IIPOST	WAS MY ECB POSTED?	14860000
BF	ISWAIT	LOOP BACK TO WAIT IF NOT	14870000
SPACE			14880000
*****			14890000
* RETURN HERE AFTER WAIT SATISFIED WITH POST			14900000
*****			14910000
SPACE			14920000
ISREGS EQU *		RESTORE THE REGISTERS	14930000
B	CC4TT	BRANCH TO TRACE	14940000
DC	AL1(TTIIRT)	TRACE ID	14950000
L	NCTCB@,XR1	POINT TO THE TCB	14960000
MVC	IICARR,IIIR(2,XR1)	SAVE THE RETURN ADDRESS	14970000
L	IIIX2(,XR1),XR2	RESTORE XR2	14980000
L	IIIX1(,XR1),XR1	RESTORE XR1	14990000
L	IICARR,IAR	RETURN TO CALLER	15000000
EJECT			15010000
*****			15020000
* CONSTANTS AND WORK AREAS *			15030000
*****			15040000
SPACE			15050000
IIVTNT DS	AL2	SAVE AREA FOR TNT ENTRY ADDRESS	15060000
SMESG EQU	C'S'	IDENTIFIER FOR AN 'S' MESSAGE	15070000
STOPER EQU	X'00'	STOPPR BYTE-INDICATES END OF TNT	15080000
IISTNL EQU	6	LENGHT OF SYMBOLIC TERMINAL NAME	15090000
CC0006 DC	IL2'6'	2-BYTE BINARY 6	15100000
CC0003 DC	IL2'3'	2-BYTE BINARY 3	15110000
IIZERO EQU	0		15120000
MCTIN EQU	X'F0'	INPUT COMPONENT BITS	15130000
MCTOUT EQU	X'0F'	OUTPUT COMPONENT BITS	15140000
IIMAXK DC	AL2(0071)	MAX OUTPUT LENGTH FOR CONSOLE	15150000
IIMAXC DC	AL2(0107)	MAX LENGTH FOR WTO	15160000
ENABLE EQU	X'78'	INTERRUPTS ENABLED	15170000
DISABL EQU	X'79'	INTERRUPTS DISABLED	15180000
PMRQ EQU	X'30'	Q-CODE TO LOAD PMR	15190000
IIWTO EQU	X'10'	FUNCTION CODE FOR WTO	15200000
IIWTOR EQU	X'20'	FUNCTION CODE FOR WTOR	15210000
II@PRL DC	XL2'00'	SAVE AREA FOR PARAMETER LIST @	15220000
IIPOST EQU	X'40'	ECB POST BIT	15230000
IIWAIT EQU	X'80'	ECB WAIT BIT	15240000
IISKIP EQU	X'20'	ECB SKIP BIT	15250000
IICARR DC	XL2'00'	ARR SAVE AREA	15260000
IIIR EQU	TCBPL+1	ARR SAVE AREA IN TCB	15270000

IIIX1	EQU	TCBPL+3	XR1 SAVE AREA IN TCB	15280000
IIIX2	EQU	TCBPL+5	XR2 SAVE AREA IN TCB	15290000
IINODA	EQU	X'20'	CHARACTER FOR ZERO DATA - WTOR	15300000
IIBLNK	EQU	X'40'	CHARACTER BLANK	15310000
IIMIN1	DC	XL2'FFFF'	CONSTANT OF MINUS 1	15320000
DSFLAG	EQU	X'002F'	FLAG BYTE FOR ENVIRONMENT INDS.	15322000
ENVSAV	EQU	X'80'	SAVE ENVIRONMENT.	15324000
N2G0	EQU	X'0015'	DISPATCHER ADDRESS.	15326000
*			----- START -@01	15330000
IITIMR	EQU	X'00'	TIMER HOURS/MINUTES/SECONDS	15340000
IITMWT	EQU	X'10'	TIMER WAIT FLAG	15350000
IIWFLG	EQU	X'07'	WAIT FLAG DISPLACEMENT	15360000
IILN10	DC	XL2'000A'	WAIT DATA AREA LENGTH	15370000
*			----- END ---@01	15380000
NHIGH	EQU	LO+EQ	NOT HIGH TEST	@16 15382000
NLOW	EQU	HI+EQ	NOT LOW TEST	@16 15384000
NEQ	EQU	HI+LO	NOT EQUAL TEST	@16 15386000
	EJECT			15390000
*	WTO OR WTOR PARAMETER LIST			15400000
	SPACE			15410000
	DC	XL1'FF'	TYPE AND FORMAT OF WTO	15420000
	DC	AL1(X'00')	SYSTEM OPTIONS	15430000
	DC	CL2'CP'	COMPONENT IDENTIFICATION (CCP)	15440000
	DC	CL2'U-'	MESSAGE ID - PART 1	15450000
	DC	AL1(X'10')	FUNCTION AND OPTIONS	15460000
	DC	CL2' '	MESSAGE ID - PART 2	15470000
	DC	AL1(0)	TEXT LENGTH	15480000
	DC	AL2(0)	TEXT ADDRESS	15490000
	DC	AL1(0)	REPLY LENGTH	15500000
IICNPM	DC	AL2(0)	REPLY ADDRESS	15510000
	SPACE			15520000
.*				15530000
	AIF	(&NDF).C75	. SKIP IF *NO* DFF SUPPORT	15540000
.*				15550000
.*				15560000
.C75	AIF	(&NOB).TAB	. SKIP IF *NO* BSCA SUPPORT	15570000
.*				15580000
IIWORK	DC	XL2'00'	WORK AREA FOR BLOCK LENGTH	15590000
II19	DC	IL2'19'	ADD LENGTH OF PARM LIST GETMAIN	15600000
.TAB	ANOP			15610000
	EJECT			15620000
*****				15630000
* OP CODE VALIDITY TABLE				15640000
*****				15650000
	SPACE			15660000
IIINVD	EQU	0	INVALID OPERATION	15670000
IIVKNS	EQU	BIT5	BIT IN VALIDITY BYTE FOR VALID	15680000
*			OPERATION WITH CONSOLE	15690000
IIVBLK	EQU	BIT6	BIT IN VALIDITY BYTE FOR VALID	15700000
*			OPERATION WITH BLANK STN	15710000
IIVSTN	EQU	BIT7	BIT IN VALIDITY BYTE FOR VALID	15720000
*			OPERATION WITH STN	15730000
TANDNE	EQU	X'11'	TRUE AND NOT EQUAL CONDITION	15740000
	SPACE	3		15750000
* OP CODE VALIDITY TABLE				15760000
	SPACE			15770000
	ENTRY	IIVTBL		15780000
IIVTBL	EQU	*	USER OP STN BLANKS CONSOLE	15790000
	DC	AL1(IIVSTN+IIVBLK+IIVKNS)	00 X X X	15800000
	DC	AL1(IIVSTN+IIVBLK)	01 X X	15810000

DC	AL1 (IIVSTN+IIVBLK+IIVKNS)	02	X	X	X	15820000
DC	AL1 (IIVSTN+IIVBLK+IIVKNS)	03	X	X	X	15830000
DC	AL1 (IIVSTN+IIVBLK+IIVKNS)	04	X	X	X	15840000
DC	AL1 (IIVSTN+IIVBLK)	05	X	X		15850000
DC	AL1 (IIVSTN+IIVBLK+IIVKNS)	06	X	X	X	15860000
DC	AL1 (IIIINVD)	07				15870000
DC	AL1 (IIVSTN+IIVBLK+IIVKNS)	08	X	X	X	15880000
DC	AL1 (IIVSTN)	09	X			15890000
DC	AL1 (IIVSTN+IIVBLK)	10	X	X		15900000
DC	AL1 (IIIINVD)	11				15910000
DC	AL1 (IIIINVD)	12				15920000
DC	AL1 (IIIINVD)	13				15930000
DC	AL1 (IIIINVD)	14				15940000
DC	AL1 (IIIINVD)	15				15950000
SPACE	2					15960000
*		BIT 5 ON - VALID OP FOR CONSOLE				15970000
*		BIT 6 ON - VALID OP FOR BLANKS				15980000
*		BIT 7 ON - VALID OP FOR SYMBOLIC				15990000
*		TERMINAL NAME				16000000
SPACE	4					16010000
IIVBYT DS	CL1	VALIDITY BYTE - BIT SIGNIFICANT				16020000
*		*** END OF MACRO '\$E060 *****				16030000
MEND						16040000

```

MODULE-$E065 , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$E065                                00020000
GBLB  &DFF                            00030000
GBLB  &NDF                            00040000
TEXT                                  00050000
AIF  (&NDF).END                      00060000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE. 3270 SUPPORT FOR CCP' 00070000
SPACE                                00080000
*****                                00090000
*                                    * 00100000
*NAME- $CC4DF                        * 00110000
*                                    * 00120000
*VERSION-V3M0      MODEL 15 CCP      * 00130000
*                                    * 00140000
*TITLE- $CC4DF. DISPLAY FORMAT CONTROL ROUTINE, MAINLINE (DFCR). * 00150000
*   THIS IS THE MAIN ROUTINE TO SERVICE USER PROGRAMS WHICH USE * 00160000
*   THE DISPLAY FORMAT FACILITY (DFF). * 00170000
*                                    * 00180000
*FUNCTION- PROVIDE LOGICAL DATA MANAGEMENT SUPPORT FOR COMPONENTS OF * 00190000
*   OF THE 3270 SYSTEM. THIS MAINLINE ROUTINE WILL CONTROL A * 00200000
*   SERIES OF TRANSIENTS WHICH OPERATE IN THE 'GENERAL' TRANSIENT * 00210000
*   AREA IN CCP. * 00220000
*   THE PRIME FUNCTION IS TO SCREEN OUT THE 3270 LINE + DATA * 00230000
*   CONTROL CHARACTERS BETWEEN THE USER AND THE 3270 SYSTEM. * 00240000
*                                    * 00250000
*   THIS ROUTINE IS FIELD-ORIENTATED, TABLE-DRIVEN, * 00260000
*   SEGMENTALLY REUSABLE. IT IS CAPABLE OF SERVICING MORE THAN * 00270000
*   ONE TASK AT A TIME. * 00280000
*                                    * 00290000
*   THE FUNCTIONS PROVIDED FOR THE VARIOUS OP CODES ARE * 00300000
*   AS FOLLOWS... * 00310000
*                                    * 00320000
*   'GET MESSAGE' AND 'STOP INVITE INPUT' -- THE USER PARAMETER * 00330000
*   LIST WILL BE MODIFIED TO TELL $CC4CM HOW MUCH OF A HOLD * 00340000
*   AREA TO GETMAIN FOR THE 3270 TEXT. AFTER THE OP END * 00350000
*   OCCURS THE DATA WILL BE MOVED FROM THE DYNAMIC TP BUFFER * 00360000
*   TO THE USER'S RECORD AREA. THE ADDRESS OF THE TP AREA * 00370000
*   WHICH CONTAINS THE 3270 TEXT WILL BE PLACED IN THE USER'S * 00380000
*   PARAMETER LIST BY $CC4CM. THE PARAMETER LIST WILL BE RE- * 00390000
*   STORED TO THE ORIGINIAL CONDITION BEFORE RETURNING TO * 00400000
*   $CC4II. THE TP AREA REQUIRED FOR PASSING 3270 TEXT IS * 00410000
*   GET-MAINED BY $CC4CM AND FREED BY $CC4CM. EXCEPTIONS ARE * 00420000
*   INPUT OPERATIONS. IN THOSE CASES $CC4DF FREE MAINS THE * 00430000
*   AREAS GET-MAINED BY $CC4CM. * 00440000
*                                    * 00450000
*   'INVITE INPUT' -- THE PLRECA AND PLINL AS SET BY THE USER IS * 00460000
*   IGNORED BY THIS ROUTINE. $CC4CM WILL GETMAIN AN AREA FOR * 00470000
*   THE 3270 TEXT, AND PLINL IS FILLED IN BY THIS ROUTINE * 00480000
*   FROM TTINLH. * 00490000
*                                    * 00500000
*   'ACCEPT INPUT' -- THIS OPERATION IS HANDLED FOR THE MOST PART * 00510000
*   LIKE 'GET MESSAGE', EXCEPT THE DATA AREA ADDRESS OF THE * 00520000
*   3270 TEXT AND PLEFFL IS OBTAINED FROM THE TUB. * 00530000
*   THIS ROUTINE WILL HANDLE DATA WITH A PROGRAM REQUEST IF * 00540000
*   PRUF IS ACTIVE ON THE TERMINAL AT PROGRAM REQUEST TIME. * 00550000
*   IF PRUF IS ACTIVE, DATA WILL BE HANDLED AS NORMAL ACCEPT * 00560000
*   INPUT OPERATION. IF NOT, DATA WILL BE HANDLED ENTIRELY * 00570000
*   BY $CC4CM. * 00580000
*                                    * 00590000

```

```

*      'PUT MESSAGE' -- PLOUTL IN THE USER'S PARAMETER LIST IS      * 00600000
*      MODIFIED BY THIS ROUTINE TO REFLECT THE ACTUAL AMOUNT      * 00610000
*      OF 3270 TEXT BEING SENT BY THIS ROUTINE. THE TEXT IS      * 00620000
*      READ FROM THE DISPLAY FORMAT IN THE OBJECT LIBRARY INTO    * 00630000
*      THE TP BUFFER AREA. AFTER THE USER'S EXECUTION TIME DATA * 00640000
*      HAS BEEN MERGED WITH THE GENERATION TEXT IN TPBUFFER,      * 00650000
*      CM IS POSTED TO MOVE DATA FROM THERE TO LINE BUFFER.     * 00660000
*      ALL TEXT, BLOCKED OR NOT BLOCKED IS PUT OUT VIA          * 00670000
*      POSTING $CC4CM.                                           * 00680000
*                                                                * 00690000
*      'PUT OVERRIDE' -- THE TRANSIENT $CC4DB WILL BE EVOKED TO  * 00700000
*      DIAGNOSE THE USE OF THE OPERATION AND TO CALCULATE THE    * 00710000
*      LENGTH OF TEXT TO GENERATE, ALSO DETERMINING IF BLOCKING  * 00720000
*      IS REQUIRED. THE 3270 TEXT WILL BE GENERATED DYNAMICALLY * 00730000
*      BY THIS ROUTINE IN TPBUFFER. IF BLOCKING IS REQUIRED, THE  * 00740000
*      SAME PROCEEDURE IS USED AS THAT FOR 'PUT MESSAGE'.        * 00750000
*                                                                * 00760000
*      'COPY' -- SET UP PARAMETER LIST SO $CC4CM WILL GETMAIN AN * 00770000
*      AREA IN TPBUFFER TO BUILD COPY TEXT.                      * 00780000
*      THIS OPERATION WILL GO THROUGH $CC4CM AS A 'PUT MESSAGE' . * 00790000
*                                                                * 00800000
*      'ERASE ALL UNPROTECTED' -- THE TEXT FOR THIS OPERATION    * 00810000
*      IS CONTAINED IN PAS. THE USER'S PARAMETER LIST WILL BE  * 00820000
*      MODIFIED TO REFLECT THIS FACT. THIS OPERATION IS SENT TO * 00830000
*      $CC4CM AS A 'PUT MESSAGE' OPERATION. THE USER'S PARAMETER * 00840000
*      LIST WILL BE RESTORED BEFORE RETURNING THROUGH $CC4II     * 00850000
*      THE FINAL TIME.                                           * 00860000
*                                                                * 00870000
*      'RELEASE TERMINAL' -- THIS OPERATION INVOLVES BLANKING    * 00880000
*      OUT THE TT ENTRY FOR THE RESPECTIVE TERMINAL. NO         * 00890000
*      REPERCUSSIONS ARE TAKEN OUT ON THE USER FOR RELEASING   * 00900000
*      TERMINAL WHICH HAD NOT PREVIOUSLY BEEN USED WITH THIS    * 00910000
*      ROUTINE.                                                  * 00920000
*                                                                * 00930000
*      *ENTRY POINT- THERE IS ONLY ONE ENTRY POINT FROM $CC4II - * 00940000
*      DFA000.                                                  * 00950000
*                                                                * 00960000
*      *INPUT- XR2 MUST CONTAIN THE HIGH ORDER ADDRESS OF THE    * 00970000
*      USER CCP PARAMETER LIST.                                  * 00980000
*      A 'USER PROGRAM APPENDED STORAGE AREA' (PAS) MUST BE     * 00990000
*      AVAILABLE FOR STORING DATA SPECIFIC TO THIS TASK.       * 01000000
*                                                                * 01010000
*      DISPLAY FORMATS MUST HAVE BEEN ALREADY GENERATED AND     * 01020000
*      MUST EXIST IN THE OBJECT LIBRARY. THE FORMAT MUST BE IN  * 01030000
*      TWO PARTS. THE FIRST PART IS THE 'FIELD DESCRIPTOR TABLE' * 01040000
*      (FDT), AND THE SECOND IS THE TEXT STREAM FOR AN INITIAL   * 01050000
*      PUT TO A 3270 TERMINAL.                                   * 01060000
*                                                                * 01070000
*      FOR ALL OPERATIONS, PLRECA MUST POINT TO THE HIGH-ORDER  * 01080000
*      BYTE OF THE TERMINAL NAME FOR WHICH THE OPERATION IS     * 01090000
*      INTENDED (EXCEPT FOR ACCEPT INPUT, IN WHICH CASE THE    * 01100000
*      POINTER IS TO THE HIGH ORDER BYTE OF THE DATA AREA).    * 01110000
*      IMMEDIATELY FOLLOWING THE 6 CHARACTER TERMINAL NAME     * 01120000
*      MUST BE THE BEGINING OF THE DATA AREA.                  * 01130000
*                                                                * 01140000
*      FOR OUTPUT OPERATIONS, THE FOLLOWING INFORMATION IS     * 01150000
*      ALSO REQUIRED...                                           * 01160000
*      'PUT MSG'- DISPLAY FORMAT NAME IN FIRST 6 POSITIONS OF   * 01170000
*      THE DATA AREA, FOLLOWED BY ANY DATA FOR FIELDS WHICH    * 01180000
*      WERE DEFINED AS BEGIN DEFINED AT 'EXECUTION' TIME.      * 01190000
*                                                                *
*      'PUT OVERRIDE'- THE WCC MUST BE THE FIRST POSITION IN THE *

```

```

*          DATA AREA. THIS IS THE MINIMUM ENTRY FOR THIS OPERATION.* 01200000
*          OPTIONALLY, 9 POSITIONS SHOULD BE GIVEN FOR EACH FIELD * 01210000
*          WHICH IS TO BE OVERRIDEN. 6 FOR FIELD NAME 1 EACH FOR * 01220000
*          TYPE, CURSOR, AND MODIFY. * 01230000
* * 01240000
*          FOR ALL OUTPUT OPERATIONS, THE TEXT IS EITHER READ INTO THE * 01250000
*          TPBUFFER AREA OR BUILT DIRECTLY INTO IT. * 01260000
* * 01270000
*OUTPUT- FORMATTED DATA IN USERS RECORD AREA FOR INPUT OPERATION, OR * 01280000
*          PROPER TEXT STREAM SEQUENCE TO 3270 FOR OUTPUT OPERATIONS. * 01290000
* * 01300000
*          UPDATED ARES IN PAS TO REFLECT CURRENT STATUS OF OPERATION * 01310000
*          FOR THIS TASK. * 01320000
* * 01330000
* * 01340000
*          IF A TASK WILL BE TERMINATED, THE TRANSIENT $CC4DD WILL BE * 01350000
*          EVOKED TO DETERMINE IF THE CONSOLE MESSAGE 528 SHOULD BE * 01360000
*          ISSUED. * 01370000
* * 01380000
*DIAGNOSTICS- THE FOLLOWING DIAGNOSTICS ARE PERFORMED AND ISSUED * 01390000
*          BY THIS ROUTINE. (OTHERS ARE GIVEN BY $CC4DB AND $CC4DC). * 01400000
*          TERMINATION CODES... * 01410000
* * 01420000
*          FF -255 * 01430000
*          FE - 254 * 01440000
*          FD - 253 * 01450000
*          FC - 252 * 01460000
*          FB - 251 * 01470000
*          F9 - 249 * 01480000
*          F8 - 248 * 01490000
*          F6 - 246 * 01500000
*          F5 - 245 * 01510000
*          F3 - 243 * 01520000
*          F2 - 242 * 01530000
* * 01540000
*          * 01550000
*EXTERNAL REFERENCE- * 01560000
*          CCCOM * 01570000
*          TCB * 01580000
*          USER PARAMETER LIST * 01590000
*          TUB * 01600000
*          FDT (FIELD DESCRIPTOR TABLE) * 01610000
*          PAS (PROGRAM APPENDED STORAGE) * 01620000
*          TT (TERMINAL TABLE) * 01630000
*          FT (FORMAT TABLE) * 01640000
* * 01650000
*CALLED BY- $CC4II. THERE WILL BE TWO CALLS TO THIS ROUTINE FOR THE * 01660000
*          FOLLOWING OPERATION CODES, ONCE BEFORE EVOKING $CC4CM AND * 01670000
*          ONCE AFTER $CC4CM: * 01680000
*          PUT MESSAGE * 01690000
*          PUT OVERRIDES * 01700000
*          GET MESSAGE * 01710000
*          INVITE INPUT * 01720000
*          COPY * 01730000
*          ERASE ALL UNPROTECTED * 01740000
*          STOP INVITE INPUT * 01750000
*          ACCEPT INPUT * 01760000
*          RELEASE TERMINAL * 01770000
* * 01780000
*          * 01790000
*EXITS TO- $CC4CM COMMUNICATION MANAGEMENT

```


FOURD	EQU	4	CONSTANT EQUATE	02400000
			SPACE	02410000
*			MISCELLANEOUS EQUATES	02420000
*			REGISTER EQUATES	02430000
CCOM	EQU	1	REG FOR CCP COM AREA	02440000
COM2	EQU	2	REGISTER EQUATE	02450000
CCOM2	EQU	2	REGISTER EQUATE	02460000
PAS1	EQU	1	REGISTER EQUATE	02470000
PAS2	EQU	2	REGISTER EQUATE	02480000
FT1	EQU	1	REGISTER EQUATE	02490000
FT2	EQU	2	FORMAT TABLE REG	02500000
LRA1	EQU	1		02510000
LRA2	EQU	2		02520000
FDT2	EQU	2		02530000
FDT1	EQU	1		02540000
OPTR1	EQU	1		02550000
DTF1	EQU	1	DTF PTR	02560000
DTF2	EQU	2	DTF REGISTER	02570000
UPL1	EQU	1	PARM LIST REGISTER	02580000
UPL2	EQU	2	REGISTER EQUATE	02590000
TCB1	EQU	1	REGISTER EQUATE	02600000
TCBR1	EQU	1	TCB REG 1	02610000
TT1	EQU	1	REGISTER EQUATE	02620000
TT2	EQU	2	REG TO POINT TO TT	02630000
TUB1	EQU	1	TUB PTR	02640000
TUB2	EQU	2	TUB POINTER	02650000
GM2	EQU	2	REGISTER FOR GET-MAIN PARM LIST	02660000
OHA1	EQU	1	REGISTER EQUATE	02670000
OHA2	EQU	2	REGISTER EQUATE	02680000
OH2	EQU	2	REGISTER EQUATE	02690000
TCB2	EQU	2	REGISTER EQUATE	02700000
IOB	EQU	1	REGISTER EQUATE	02710000
TYP1	EQU	1	REGISTER EQUATE	02720000
BFR2	EQU	2	REGISTER EQUATE	02730000
			SPACE	02740000
*			MISC EQUATES	02750000
COM	EQU	X'C000'	FLAG FOR RELOCATION TO CCOM VALU	02760000
PGM	EQU	X'8000'	FLAG FOR RELOCATION TO CCP PGM	02770000
X01	EQU	X'01'	HEX EQUATE VALUE	02780000
CMD	EQU	1	HEX EQUATE VALUE	02790000
X20	EQU	X'20'	HEX EQUATE VALUE	02800000
XC0	EQU	X'C0'	HEX EQUATE VALUE	02810000
X08	EQU	X'08'	HEX EQUATE VALUE	02820000
X06	EQU	X'06'	HEX EQUATE VALUE	02830000
X27	EQU	X'27'	HEX EQUATE VALUE	02840000
X16	EQU	X'16'	HEX EQUATE VALUE	02850000
X80	EQU	X'80'	HEX EQUATE VALUE	02860000
X07D	EQU	7	HEX EQUATE VALUE	02870000
X60	EQU	X'60'	HEX VALUE 60	02880000
XF0	EQU	X'F0'	NEX F0	02890000
X0F	EQU	X'0F'	HAX 0F	02900000
EQUFF	EQU	X'FF'	HEX EQUATE VALUE	02910000
BLANKD	EQU	X'40'	EQUATE FOR A BLANK	02920000
END	EQU	X'FF'	STOPPER	02930000
MODDAT	EQU	C'M'	MODIFIED DATA SUPPLIED @02	02940000
			SPACE	02950000
*			EQUATES FOR THE FUNCTION DETERMINATION TABLE	02960000
OPTBOP	EQU	0	DISP TO OP CODE	02970000
OPTB@	EQU	2	DISP TO ADDRESS FOR OP CODE	02980000
OPTBLL	EQU	3	LENGTH OF AN ENTRY	02990000

	SPACE		03000000	
HIGH	EQU	X'84'	HIGH CONDITION CODE	03010000
NOTLOW	EQU	X'02'	NOT LOW CONDITION CODE	03020000
NOTHI	EQU	X'04'	NOT HIGH CONDITION CODE	03030000
TANDEQ	EQU	X'16'	TRUE AND EQUAL	03040000
FORNE	EQU	X'96'	FALSE OR NOT EQUAL	03050000
FOREQ	EQU	X'91'	FALSE OR EQUAL CONDITION CODE	03060000
	SPACE		03070000	
FTHD	EQU	5	L-1 OF FT NAME	03080000
TNAMLD	EQU	5	L-1 OF TERMINAL NAME	03090000
LINE1	EQU	X'80'	DEVICE CODE FOR BSCA LINE 1	03100000
DTFDEV	EQU	0	DISP IN DTF TO DEVICE CODE	03110000
NUMBIT	EQU	X'0F'	NUMERIC BITS	03120000
NEGZON	EQU	X'D0'	NEGATIVE ZONE	03130000
POSZON	EQU	X'F0'	POSITIVE ZONE	03140000
MINUS	EQU	C'-'	MINUS SIGN	03150000
IOBOK	EQU	X'40'	IOB COMPLETION CODE FOR 'NORMAL'	03160000
FNAMLD	EQU	6	LENGTH OF A FIELD NAME	03170000
	SPACE		03180000	
* EQUATES FOR THE DISPLAY FORMAT INDEX IN CCPFILE				03190000
INX2	EQU	2	REGISTER FOR FORMAT INDEX	03200000
INXNAM	EQU	5	DISP TO NAME IN INDEX ENTRY	03210000
INXCS	EQU	INXNAM+2	DISP TO DISK C/S IN INDEX	03220000
INXLNG	EQU	INXCS+1	FULL LENGTH OF AN INDEX ENTRY	03230000
INXEND	EQU	X'FF'	INDICATION OF END OF INDEX	03240000
INXNS	EQU	32	NUMBER INDEX ENTRIES IN A SECTOR	03250000
	SPACE			03260000
* EQUATES FOR CHECKING LNG OF TEXT FOR 'PUT OVERRIDES'				03270000
* THESE EQUATES WILL OVERLAY THE FIELD NAME INN THE FDT				03280000
FLDOHL	EQU	2	SAVE AREA IN FDTNAM FOR OHA LNG	03290000
FLDTLN	EQU	4	SAVE AREA IN FDTNAM FOR TEXT LNG	03300000
	SPACE			03310000
CCCDSP	EQU	6	DISP TO CCC IN LRA	03320000
WCC	EQU	2	DISP IN TEXT TO WCC	03330000
ININIT	EQU	4	DISP IN INPUT TEXT TO CURSOR @	03340000
IHAD@D	EQU	3	DISP TO TEXT @ IN IHA WHEN	03350000
*			POINTING TO BYTE BEFORE SBA	03360000
IHA2	EQU	2		03370000
LRAAID	EQU	0	DISP IN LRA TO AID	03380000
IHAID	EQU	2	DISP IN IHA TO AID	03390000
SPDCHR	EQU	C'>'	GT SIGN	03400000
HIGHBT	EQU	X'C0'	BIT 0 AND BIT 1	03410000
AIDSPA	EQU	X'7E'	AID FOR SPD ATTENTION	03420000
	SPACE			03430000
* EQUATES FOR THE WRITE CONTROL CHARACTER (WCC) FOR 3270				03440000
S3WCC	EQU	X'D0'	S/3 WCC OF 'D0'	03450000
SPCWCC	EQU	X'6A'	3270 SPECIAL WCC OF '6A'	03460000
MDTWCC	EQU	X'C1'	WCC TO RESET MDT	03470000
DUMWCC	EQU	X'40'	DUMMY WCC USED FOR SECOND AND	03480000
*			SUCCESSIVE BLOCKS OF OUTPUT.	03490000
WCCMDT	EQU	BIT7	MDT BIT IN WCC	03500000
WCCRKB	EQU	BIT6	WCC BIT TO RESET KEYBOARD	03510000
SBADSP	EQU	0	DISP IN OHA TO SBA	03520000
SBA@DP	EQU	2	DISP TO @@ AFTER SBA	03530000
ATTDSP	EQU	4	DISP TO ATTRIBUTE	03540000
SFDP	EQU	3	DISP TO SF CHAR WITH ATTR	03550000
CURDPT	EQU	5	DISP TO IC WITH 'TYPE' GIVEN	03560000
CURDP	EQU	3	DISP TO IC WITHOUT 'TYPE' GIVEN	03570000
SPDCDP	EQU	5	DISP IN TEXT TO DESIGNATOR CHAR	03580000
	SPACE			03590000


```

* GET TERMINAL NAME AND SAVE IT IN CURRENT AREA IN PAS. 04200000
SPACE 1 04210000
ST PASPL@(,PAS1),UPL2 SAVE UPL ADDRESS 04220000
MVC PASCPL+PLTUBA(,PAS1),PLTUBA(PLTUBA+1,UPL2) SAVE ORIGINAL 04230000
* VERSION OF PARM LIST IN PAS. 04240000
SPACE 04250000
CLI PLOPC(,UPL2),OPACI ACCEPT INPUT OPERATION? 04260000
JNE DFA045 NO, CONTINUE 04270000
ALC PASCPL+PLRECA(2,PAS1),SIX YES, UPDATE PTR TO DATA 04280000
J DFA047 CONTINUE, RTN CODE SET UP ALREAD 04290000
DFA045 SLC PLRTC(,UPL2),PLRTC(2,UPL2) CLEAR RETURN CODE TO ZERO 04300000
DFA047 L PASCPL+PLRECA(,PAS1),LRA2 GET LRA ADDRESS 04310000
A MIN6,LRA2 POINT BACK TO TNAME 04320000
MVC PASCTN(,PAS1),TNAMLD(6,LRA2) SAVE TERMINAL NAME 04330000
L PASPL@(,PAS1),UPL2 RELOAD UPL 04340000
SPACE 2 04350000
* NOW DETERMINE THE OP CODE AND GIVE CONTROL TO APPROPRIATE SPOT. 04360000
SPACE 1 04370000
DFA050 LA OPTBL,OPTR1 LOAD TABLE OF OP CODES + @ 04380000
CLC OPTBOP(,OPTR1),PLOPC(1,UPL2) OP CODES MATCH 04390000
JE DFA060 YES, FINISH UP 04400000
LA OPTBLL(,OPTR1),OPTR1 INCREMENT TO NEXT ELEMENT 04410000
CLI ZEROD(,OPTR1),END END OF LEGAL OP CODES? 04420000
BNE DFA050 NO, LOOP BACK 04430000
***** ERROR INVALID OP CODE SENT TO THIS ROUTINE 04440000
B DFT030 CALL TERMINATION ROUTINE 04450000
DC AL1(TCCIOP) TERMINATION CODE 04460000
SPACE 2 04470000
DFA060 EQU * 04480000
SPACE 1 04490000
* ONLY ONE OF THE VALID OP CODES IN THE TABLE ARE EXPECTED. 04500000
* NOW TRANSFER CONTROL TO THE APPROPRIATE ROUTINE. 04510000
SPACE 1 04520000
DFA070 MVC XCTLA1,OPTB@(2,OPTR1) MOVE ENTRY @ TO INSTR 04530000
LA #,PAS1 RELOAD PAS @ 04540000
PAS@A1 EQU DFA070+3 04550000
SPACE 1 04560000
DFA080 B # TRANSFER CONTROL 04570000
XCTLA1 EQU DFA080+3 04580000
EJECT 04590000
* * * * * 04600000
* * * * * 04610000
* OP CODE TABLE AND RESPECTIVE PROCESSING ROUTINES. * 04620000
* * * * * 04630000
* * * * * 04640000
SPACE 1 04650000
OPTBL EQU * 04660000
DC AL1(OPGET) 'GET' OP CODE AND 'STOP II' 04670000
DC AL2(DFG000) ENTRY POINT OF APPROPRIATE RTN 04680000
SPACE 1 04690000
DC AL1(OPINV) 'INVITE INPUT' 04700000
DC AL2(DFG000) ENTRY POINT OF APPROPRIATE RTN 04710000
SPACE 1 04720000
DC AL1(OPACI) 'ACCEPT INPUT' 04730000
DC AL2(DFG000) ENTRY POINT OF APPROPRIATE RTN 04740000
SPACE 1 04750000
DC AL1(OPPUT+OPMSG) 'PUT MSG' + 'PUT OVERRIDES' 04760000
DC AL2(DFB000) ENTRY POINT OF APPROPRIATE RTN 04770000
SPACE 1 04780000
DC AL1(OPPNW+OPMSG) 'PUT NO WAIT' + 'PUT OVERRIDES' 04790000

```


MVC	PLOUTL(,UPL2),FTOUTL(2,FT1)	MOVE TEXT LENGTH TO UPL	05400000
L	PAS@A1,PAS1	SAVE ALSO IN WORK AREA.	05410000
MVC	PASTTS(,PAS1),PLOUTL(2,UPL2)		05420000
B	DFH000	GETMAIN AN OUTPUT HOLD AREA	05430000
SPACE	2		05440000
B	DFJ000	READ FDT INTO PAS	05450000
B	DFFRTN	RETURN TO II MAINLINE	05460000
SPACE			05470000
TITLE	'DISPLAY FORMAT CONTROL ROUTINE.-- COPY OPERATION'		05480000
* * * * *	* * * * *		05490000
*			05500000
*	'COPY' OP CODE SECTION.		05510000
*			05520000
* * * * *	* * * * *		05530000
SPACE	2		05540000
DFC000 EQU	*	ENTRYP POINT TO THIS SECTION	05550000
*	VERIFY THE 'TO' TERMINAL NAME		05560000
SPACE			05570000
SBN	PASFLG(,PAS1),PASFR	INDICATE 'TO' TERMINAL NAME	05580000
B	DFJ000	CALL RTN TO FIND TERMINAL ENTRY	05590000
ST	PASTOD(,PAS1),TT2	SAVE TT ENTRY ADDRESS	05600000
SPACE			05610000
*	VERIFY 'FROM' TERMINAL NAME		05620000
SPACE			05630000
SBF	PASFLG(,PAS1),PASFR	INDICATE 'FROM' TERMINAL NAME	05640000
SPACE			05650000
MVC	PASUDL(,PAS1),PASCPL+PLOUTL(2,PAS1)	SAVE OUTPUT LENGTH	05660000
L	PASCPL+PLRECA(,PAS1),LRA2	POINT TO RCD AREA	05670000
MVC	PASCTN(,PAS1),TNAMLD(6,LRA2)	MOVE NAME TO 'CURENT' AREA	05680000
MVC	PASCCC(,PAS1),CCCDSP(1,LRA2)	MOVE CCC IF ANY	05690000
B	DFJ000	CALL TERMINAL NAME VERIFY RTN	05700000
ST	PASFRD(,PAS1),TT2	SAVE TT ENTRY @	05710000
SPACE			05720000
*	GETMAIN AN AREA FOR 3270 TEXT		05730000
SPACE			05740000
MVC	PASTTS(2,PAS1),FOUR	SET TO GETMAIN FOUR BYTES	05750000
B	DFH000	CALL GETMAIN ROUTINE	05760000
SPACE			05770000
B	DFFRTN	RETURN TO II MAINLINE	05780000
\$CC4DF TITLE	'DISPLAY FORMAT CONTROL ROUTINE. --PUT OVERRIDES--'		05790000
* * * * *	* * * * *		05800000
*	PROCESS 'PUT OVERRIDES' COAMMAND.		05810000
* * * * *	* * * * *		05820000
SPACE	2		05830000
DFD000 EQU	*		05840000
SPACE	2		05850000
*	XR1-->PAS,XR2-->UPL		05860000
SPACE			05870000
*	GO TO THE ROUTINE TO FIND THE TERMINAL AND FORMAT TABLE ENTRY		05880000
SPACE			05890000
B	DFJ000	FIND TT ENTRY	05900000
B	DFJ000	GET FDT FOR THIS FORMAT	05910000
SPACE			05920000
*	CALL TRANSIENT TO CALCULATE LENGTH OF TEXT NEEDED TO PUT OUT		05930000
*	FOR THIS TEXT STREAM.		05940000
SPACE			05950000
LA	ZEROD(,PAS1),PAS2	SAVE PAS ADDRESS IN REGISTER 2	05960000
B	CC4PI	CALL TRANSIENT CONTROL ROUTINE	05970000
DC	AL1(CC4DB)	NAME OF TRANSIENT DESIRED	05980000
CCP	DISABL,PMR	DISABLE INTERRUPTS	05990000

```

ST PAS@A1,PAS2 STORE PAS ADDRESS LOCALLY 06000000
SPACE 06010000
B DFT000 CALL TERMINATE CHECK ROUTINE 06020000
LA ZEROD(,PAS2),PAS1 RELOAD PAS REGISTER 06030000
SPACE 06040000
* GETMAIN THE OUTPUT HOLD AREA 06050000
SPACE 06060000
L PASPL@(:,PAS1),UPL2 RELOAD PARM POINTER 06070000
BC DFH000,EQUFF GO GETMAIN AREA -----@03 06080000
DFD020 B DFFRTN RETURN TO II MAINLINE 06090000
SPACE 06100000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.---ERASE ALL UNPROTECTED' 06110000
* * * * * 06120000
* 06130000
* 'ERASE ALL UNPROTECTED' (EAU) OPERATION CODE. * 06140000
* XR1-->PAS, XR2-->USER PARAMETER LIST. * 06150000
* 06160000
* * * * * 06170000
SPACE 2 06180000
DFE000 EQU * ENTRY POINT TO THIS SECTION 06190000
SPACE 06200000
* VERIFY THE TERMINAL NAME AND GET TT ENTRY 06210000
B DFF000 CALL 'FIND TT ENTRY' ROUTINE 06220000
SPACE 06230000
* GETMAIN AN AREA FOR 3270 TEXT 06240000
SPACE 06250000
MVC PASTTS(2,PAS1),TWO INDICATE NUMBER OF BYTES TO SEND 06260000
B DFH000 CALL GETMAIN ROUTINE 06270000
B DFFRTN RETURN TO II MAINLINE 06280000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE. --FIND TT ENTRY--' 06290000
* * * * * 06300000
* 06310000
* FIND AND CREATE 'TERMINAL TABLE' ENTRY. * 06320000
* INPUT IS EXPECTED TO BE AS FOLLOWS: * 06330000
* XR1--> PAS * 06340000
* XR2--> UPL * 06350000
* NAME OF TERMINAL IN PAS * 06360000
* OUTPUT WILL BE A CREATED TT ENTRY IF NONE EXISTS ALREADY AND * 06370000
* THE ADDRESS OF THE ENTRY IN PAS. * 06380000
* 06390000
* * * * * 06400000
SPACE 2 06410000
DFF000 EQU * 06420000
SPACE 1 06430000
* * * * * 06440000
* FIND A TT ENTRY FOR THIS TERMINAL IN PAS. * 06450000
* * * * * 06460000
ST PASFR@(:,PAS1),ARR SAVE RETURN POINT IN PAS 06470000
* -----START-----@01 06480000
SBF PASFLG(:,PAS1),PASRUF SET OFF READ UNDER FORMAT 06490000
* -----END-----@01 06500000
MVI PASTTE-1(:,PAS1),ZEROD INIT 'END' OF TT ENTRIES 06510000
L PASITT(:,PAS1),TT2 LOAD ADDR OF FIRST TT 06520000
DFF010 CLC PASCTN(:,PAS1),TTNAME(6,TT2) NAME IN PAS AND TT MATCH? 06530000
JE DFF100 YES, CONTINUE PROCESSING IT 06540000
SPACE 06550000
* CHECK IF THIS ENTRY USED. IF NOT AND ONE IS NEEDED FOR A NEW 06560000
* ENTRY REUSE IT. WHEN A TERMINAL IS RELEASED THE TT NAME WILL BE 06570000
* BLANKED OUT INDICATING AN UNUSED ENTRY. 06580000
SPACE 1 06590000

```



```

DFF070 EQU *          TT2 POINTS TO LAST TT ENTRY          07200000
      LA ONED(,TT2),TT2  INCT TO RIGHT OF ADDRESS          07210000
      ST STF1,TT2       SAVE THIS POINTER.                07220000
      L  PASNFT(,PAS1),TT2  LOAD NEXT AVAIL ADDRESS        07230000
      ALC PASNFT(,PAS1),TTLNG(2)  ADD LNG OF TT ENTRY      07240000
      CLC PASNFT(,PAS1),PASEFT(2,PAS1)  NEW ADDRESS PAST END? 07250000
      JNH DFF080         NO, OKAY                          07260000
      SPACE 2                                               07270000
***** ERROR          NOT ENOUGH ROOM FOR ANOTHER        07280000
DFF075 B DFT030        ENTRY. CALL TERMINATION            07290000
      DC AL1(TC02)     TERMINATION CODE                    07300000
      SPACE 2                                               07310000
* SPACE EXISTS. ALLOCATE AREA + CREATE CHAIN POINTER.     07320000
      SPACE 1                                               07330000
DFF080 ST TTCHN@+#,TT2  PUT ADDR THIS ENTRY IN LAST ONE  07340000
STF1 EQU DFF080+3
      MVI TTCHN@-1(,TT2),ZEROD  BLANK OUT CHAIN PTR. NEW LAST. 07360000
      SPACE 2                                               07370000
* NOW CREATE OR FILL IN PART OF THE ENTRY FOR OLD AND NEW ENTRY. 07380000
      SPACE 1                                               07390000
DFF090 MVC TTNAME(,TT2),PASCTN(TNAMLD+1,PAS1)  MOVE TERMINAL NAME 07400000
      MVI TTFT@-1(,TT2),ZEROD  BLANK OUT FORMAT TABLE PTR. 07410000
      J DFF110          CONTINUE BUILDING ENTRY.           07420000
      EJECT           07430000
* * * * * 07440000
* ENTRY FOUND. PUT ELEMENT ADDRESS IN PAS IF NOT 'COPY FROM' * 07450000
* TERMINAL. IF A NEW FORMAT IS TO BE USED, CREATE A FORMAT TABLE* 07460000
* ENTRY. * 07470000
* * * * * 07480000
      SPACE 1                                               07490000
DFF100 CLI PASCPL+PLOPC(,PAS1),OPCOPY+OPPUT  IS THIS 'COPY FROM'? 07500000
      TBF PASFLG(,PAS1),PASFRM  TERMINAL?                 07510000
      JC DFF150,TANDEQ        YES, NO OTHER ACTION NEEDED. 07520000
      SPACE 1                                               07530000
DFF110 ST PASCT@(,PAS1),TT2  STORE @ IN CURRENT TT PROCESSED 07540000
* -----START-----@01 07550000
      TBN PASFLG(,PAS1),PASRUF  READ UNDER FORMAT ?       07560000
      JF DFF120              NO, BRANCH                    07570000
      MVC PASCFT(4,PAS1),PASRFN(,PAS1)  MOVE IN FORMAT NAME 07580000
      J DFF125              CONTINUE PROCESSING            07590000
* -----END-----@01 07600000
DFF120 TBN PASCPL+PLOPC(,PAS1),OPPUT+OPMSG  IS THIS A PUT MESSAGE ? 07610000
      TBF PASCPL+PLOPM(,PAS1),OPLIST (NOT OVERRIDE)        07620000
      JF DFF150              NO, SKIP FOLLOWING PART        07630000
      SPACE 2                                               07640000
* * * * * 07650000
* SEE IF MAP NAME IN LRA MATCHES ONE IN FT. IF NOT CREATE OR FIND 07660000
* CORRECT FT ENTRY. * 07670000
* * * * * 07680000
      SPACE 1                                               07690000
* MOVE THE FORMAT NAME TO PAS 07700000
      L PASCPL+PLRECA(,PAS1),LRA2  LOAD LRA ADDRESS        07710000
      MVC PASCFT(,PAS1),FTLD(4,LRA2)  MOVE NAME TO PAS     07720000
      SPACE 1                                               07730000
* LOAD FT ENTRY ASSOCIATED WITH THIS TERMINAL 07740000
DFF125 L PASCT@(,PAS1),TT2  LOAD CURRENT TT ENTRY @      07750000
      L TTFT@(,TT2),FT2     LOAD RESPECTIVE FT ENTRY      07760000
      SPACE 2                                               07770000
* THE HIGH ORDER BYTE OF THE FORMAT TABLE ENTRY IS ZEROED OUT IF A MOR 07780000
* MAT DOES NOT EXIST FOR THIS ENTRY. THEREFORE A COMPARE WILL BE 07790000

```


	ST	PASCF@ (,PAS1),FT2	SAVE AT 'CURRENT' POINTER	08400000
	ALC	PASNFT (,PAS1),FTLNG(2)	INCR BY LNG OF ENTRY	08410000
	CLC	PASNFT (,PAS1),PASEFT(2,PAS1)	ENOUGH ROOM FOR THIS ENTRY?	08420000
*****	ERROR			08430000
	BH	DFE075	NO, NOT ENOUGH ROOM FOR FT ENTRY	08440000
	SPACE	2		08450000
*	ALLOCATED	SPACE FOR AN ENTRY AND SET UP POINTER TO NEXT AVAILABLE		08460000
*	ENTRY.	NOW PREPARE TO READ IN THE INDEX IN THE THE PAS FDT AREA AND		08470000
*	FIND	THE C/S OF THE FORMAT		08480000
	SPACE	1		08490000
	MVC	PASIOB+IOBQB (,PAS1),#CPFLQ(1)	MOVE Q BYTE OF CCPFILE	08500000
	MVC	PASIOB+IOBSB (,PAS1),@DFFIX(2)	MOVE C/S OF INDEX	08510000
	MVC	DFE356+1,#DFQ(1)	SAVE Q BYTE OF FORMAT PACK	08520000
	MVC	PASIOB+IOBDAT (,PAS1),PASFDT(2,PAS1)	MOVE FDT @ FOR DATA	08530000
	MVC	PASIOB+IOBNB (,PAS1),PASFDL(1,PAS1)	MOVE LNG TO IOB	08540000
	SLC	PASIOB+IOBNB (,PAS1),ONE(1)	DECR TO L - 1	08550000
DFE340	MVC	WORK,PASFDL(1,PAS1)	MOVE FDT AREA LENGTH	08560000
	SLC	PASWRK (,PAS1),PASWRK(2,PAS1)	CLEAR WORK AREA	08570000
	MVI	PASWKZ (,PAS1),INXNS	INIT NUMBER ENTRIES IN SECTOR	08580000
	SPACE			08590000
DFE345	ALC	PASWRK (,PAS1),PASWKZ(2,PAS1)	ADD SECTOR AMT TO TOTAL	08600000
	SLC	WORK,ONE(1)	DECR NUMBER OF SECTORS	08610000
	BNE	DFE345	LOOP IF NOT FINISHED	08620000
	SPACE			08630000
*	READ	INDEX INTO FDT AREA		08640000
	B	DFV000	CALL DISK IOS	08650000
	SPACE			08660000
*	NOW	SEARCH INDEX FOR THE DESIRED DISPLAY FORMAT NAME		08670000
	L	PASFDT (,PAS1),INX2		08680000
DFE350	CLC	INXNAM (,INX2),PASCF(4,PAS1)	DO NAMES COMPARE?	08690000
	JE	DFE356	YES, EXIT	08700000
	CLI	ZEROD (,INX2),INXEND	END OF INDEX? @05	08710000
	JE	DFE355	YES, INDICATE ERROR @05	08720000
	LA	INXLNG (,INX2),INX2	INCR TO NEXT ENTRY @05	08730000
	SLC	PASWRK (,PAS1),ONE(2)	DECR NUMBER SECTORS PROCESSED	08740000
	BNE	DFE350	IF MORE SECTORS TO PROCESS, LOOP	08750000
	SPACE			08760000
*	PROCESSED	ALL THE INDEXES IN CORE AND STILL NOT FOUND.		08770000
*	READ	MORE INDEXES INTO FDT AREA		08780000
	SPACE			08790000
	B	DFI000	INCR DISK ADDRESS	08800000
	B	DFE340	LOOP BACK TO READ MORE	08810000
	SPACE	2		08820000
*****	ERROR	FORMAT NAME NOT FOUND IN INDEX		08830000
DFE355	B	DFT030	CALL TERMINATE ROUTINE	08840000
	DC	AL1(TC04)		08850000
	SPACE	2		08860000
*	INDEX	FOUND, NOW READ FDT OF THIS FORMAT INTO FDT AREA IN PAS		08870000
	SPACE			08880000
DFE356	MVI	PASIOB+IOBQB (,PAS1),#	MOVE IN Q BYTE OF PACK FORMATS	08890000
	MVC	PASIOB+IOBSB (,PAS1),INXCS(2,INX2)	MOVE FORMAT C/S INTO IO	08900000
	SPACE			08910000
*	NOW	READ THE FDT OF THE DESIRED FORMAT INTO PAS		08920000
	B	DFV000	CALL DISK IOS	08930000
	SPACE	2		08940000
*	BUILD	UP FORMAT TABLE ENTRY		08950000
	SPACE	1		08960000
DFE360	L	PASCF@ (,PAS1),FT2	RELOAD @ OF THIS ENTRY	08970000
	MVI	FTCHN@-1 (,FT2),ZEROD	CLEAR CHAIN POINTER	08980000
	MVC	FTFCS (,FT2),PASIOB+IOBSB(2,PAS1)	MOVE C/S OF FDT	08990000

MVC	FTTSCS(,FT2),FTFCS(2,FT2)	INIT C/S OF TEXT	09000000	
MVC	FTNAME(,FT2),PASCFT(4,PAS1)	MOVE FMT NAME	09010000	
L	PASFDT(,PAS1),FDT1	LOAD DATA AREA @	09020000	
MVC	FTFDTL(,FT2),FDTNS(1,FDT1)	MOVE NUMBER SECTORS OF FDT	09030000	
MVC	FTTSL(,FT2),FDTTSN(1,FDT1)	MOVE # SECTORS TEXT STREAM	09040000	
MVC	FTOUTL(,FT2),FDTTSL(2,FDT1)	MOVE BYTE LNG OF TEXT STREAM	09050000	
MVC	FTINL(,FT2),FDTTSI(2,FDT1)	MOVE FULL INPUT LENGTH	09060000	
	SPACE 2		09070000	
*	CALCULATE ACTUAL C/S OF TEXT STREAM		09080000	
	SPACE 1		09090000	
*	ADD # OF SECTORS OF FDT TO FIND BEGIN C/S OF DATA TEXT.		09100000	
MVC	FDTWRK(,FDT1),FDTNS(1,FDT1)	MOVE FDT LENGTH TO 'WORK'	09110000	
*	THE FDT IS AT LEAST ONE SECTOR IN LENGTH		09120000	
	SPACE		09130000	
DFE363	ALC	FTTSCS(,FT2),FOUR(1)	ADD ONE SECTOR TO BASE C/S	09140000
	TBN	FTTSCS(,FT2),X60	OVERFLOW TO ANOTHER TRACK?	09150000
	JF	DFE370	NO, OKAY	09160000
	ALC	FTTSCS(,FT2),X0020(2)	YES, FORCE TO NEXT TRACK @	09170000
DFE370	SLC	FDTWRK(,FDT1),ONE(1)	DECR FDT LENGTH	09180000
	BNE	DFE363	IF NOT FINISHED, LOOP BACK	09190000
	SPACE 2		09200000	
*	CHECK FOR SCREEN SIZE		09210000	
TBN	FDTM#(,FDT1),FDTM#2	IS THIS FMT FOR MODEL 2?	09220000	
JF	DFE390	NO, CONTINUE	09230000	
SBN	FTTSL(,FT2),FTM#2	YES, INDICATE IN FT	09240000	
	SPACE		09250000	
*	CHECK IF THIS TERMINAL IS PROPER MODEL NUMBER FOR THE FORMAT		09260000	
DFE390	L	PAS@A1,PAS1	RELOAD PAS @	09270000
	SPACE		09280000	
L	PASCPL+PLTUBA(,PAS1),TUB1	POINT TO RESPECTIVE TUB	09290000	
TBN	FTTSL(,FT2),FTM#2	MODEL 2 FORMAT?	09300000	
JF	DFE392	NO, CHECK MODEL 1 TERMINAL	09310000	
CLI	TUBPHY(,TUB1),TUB7M2	MODEL 2 3277?	09320000	
JE	DFE397	YES, OKAY	09330000	
CLI	TUBPHY(,TUB1),TUB5M2	MODEL 2 3275?	09340000	
JE	DFE397	YES, OKAY	09350000	
J	DFE394	NO, ERROR	09360000	
	SPACE		09370000	
DFE392	EQU	*	CHECK FOR MODEL 1 TERMINAL	09380000
	CLI	TUBPHY(,TUB1),TUB7M1	MODEL 1 3277?	09390000
	JE	DFE397	YES, OKAY	09400000
	CLI	TUBPHY(,TUB1),TUB5M1	MODEL 1 3275?	09410000
	JE	DFE397	YES, OKAY	09420000
	SPACE		09430000	
*****	ERROR	TERMINAL MODEL NUMBER DOES NOT MATCH DISPLAY MODEL #	09440000	
DFE394	B	DFT030	CALL TERMINATION ROUTINE	09450000
	DC	AL1(TC13)		09460000
	SPACE 2		09470000	
*	MOVE FORMAT NAME CURRENTLY IN PAS TO AREA IN PAS, AND CHECK		09480000	
*	IF THIS FDT WILL FIT IN THE AREA RESERVED FOR IT.		09490000	
	SPACE		09500000	
DFE397	EQU	*		09510000
	L	PAS@A1,PAS1	RELOAD PAS @	09520000
	CLC	PASFDL(,PAS1),FTFDTL(1,FT2)	WILL WHOLE FDT FIT IN AREA?	09530000
	JNL	DFE400	YES, OKAY	09540000
	SPACE		09550000	
*****	ERROR	FDT WILL NOT FIT INTO THE AREA	09560000	
DFE395	EQU	*		09570000
	B	DFT030	CALL TERMINATION ROUTINE	09580000
	DC	AL1(TC10)		09590000

```

SPACE 09600000
DFF400 L PASCT@ (,PAS1),TT1 RELOAD CURRENT TT ENTRY 09610000
ST TTFT@ (,TT1),FT2 STORE @ OF RESPECTIVE FT ENTRY 09620000
DFF410 L PAS@A1,PAS1 RELOAD PAS @ 09630000
B DFF140 RETURN TO DFF 09640000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.---'INPUT' OPERATIONS--' 09650000
* * * * * 09660000
* 09670000
* INPUT SECTION FOR 'GET', 'INVITE INPUT', 'STOP INVITE INPUT', * 09680000
* 'ACCEPT INPUT'. * 09690000
* 09700000
* * * * * 09710000
SPACE 2 09720000
DFG000 EQU * ENTRY POINT TO THIS SECTION 09730000
SPACE 09740000
* FIND THE TERMINAL NAME TABLE ENTRY 09750000
SPACE 09760000
B DFF000 FT SHOULD BE CREATED ALREADY 09770000
SPACE 09780000
*IF THIS IS AN INVITE INPUT, THEN BYPASS GETTING FDT 09790000
SPACE 09800000
CLI PASCPL+PLOPC (,PAS1),OPINV INVITE INPUT OPERATION ? 09810000
JE DFG020 YES, BRANCH 09820000
B DFJ000 GO READ IN FDT 09830000
SPACE 09840000
* BLANK OUT CALLER'S RECORD AREA 09850000
SPACE 09860000
LA MOVEPL,XR2 POINT TO MOVE PARM LIST 09870000
MVC DFGLen-MOVEPL(2,XR2),PASCPL+PLINL (,PAS1) MOVE IN LENGTH 09880000
MVC DFGRF@-MOVEPL(2,XR2),PASCPL+PLRECA (,PAS1) MOVE IN ADDR 09890000
B CC4MX CALL MOVE ROUTINE 09900000
DFG020 L PASPL@ (,PAS1),UPL2 RELOAD UPL @ 09910000
CLI PASCPL+PLOPC (,PAS1),OPACI ACCEPT INPUT OPERATION? 09920000
JE DFG060 YES 09930000
SPACE 09940000
DFG030 EQU * 09950000
* OP CODE IS 'STOP II', 'GET', OR 'INVITE INPUT'. 09960000
* MOVE LENGTH OF TEXT STREAM EXPECTED INTO PARM LIST INSTEAD OF 09970000
* USER'S LENGTH. 09980000
L PASCT@ (,PAS1),TT1 LOAD CURRENT TT ENTRY 09990000
MVC PLINL (,UPL2),TTINLH(2,TT1) MOVE EXPECTED TEXT LENGTH. 10000000
CLC PLINL(2,UPL2),#TPANY INPUT LENGTH EXCEED TPBUFF ? 10010000
JNH DFG040 NO, BRANCH 10020000
B DFT030 YES, TERMINATE USER TASK 10030000
DC AL1(TCCTPB) INPUT LENGTH > TPBUFF 10040000
DFG040 EQU * 10050000
L PAS@A1,PAS1 RELOAD PAS 10060000
DFG060 B DFFRTN RETURN TO II MAINLINE 10070000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE. GETMAIN TPBUFFER--' 10080000
* * * * * 10090000
* 10100000
* GETMAIN - SET UP LENGTH OF AREA TO BE GETMAINED BY $CC4CM FROM * 10110000
* CCP'S DYNAMIC TPBUFFER AREA. * 10120000
* 10130000
* * * * * 10140000
SPACE 2 10150000
* * * * * 10160000
DFH000 EQU * ENTRY POINT 10170000
* * * * * 10180000
SPACE 1 10190000

```

```

* THE RETURN ADDR IS SAVED IN PAS SINCE A TASK SWITCH MAY OCCUR 10200000
ST PASTAR(,PAS1),ARR SAVE RETURN POINT 10210000
SPACE 10220000
* XR1 SHOULD BE POINTING TO PAS. XR2 TO UPL 10230000
SPACE 2 10240000
* FIND OUT WHICH LINE THE TERMINAL IS ON. 10250000
SPACE 1 10260000
L PASCPL+PLTUBA(,PAS1),TUB2 LOAD @ RESPECTIVE TUB 10270000
MVC PASOHL(,PAS1),TUBRCL(2,TUB2) MOVE IN SIZE OF BSCA TUB 10280000
* RECORD LENGTH. 10290000
SBF PASFLG(,PAS1),ALLBIT-PASFTS CLEAR OUT FLAG BYTE 10300000
CLC PASTTS(,PAS1),PASOHL(2,PAS1) TOTAL TEXT > RECORD LENGTH 10310000
JNH DFH050 NO, NO BLOCKING 10320000
SPACE 2 10330000
* INDICATE BLOCKING. FIND OUT WHAT THE BLOCK SIZE IS 10340000
DFH040 SBN PASFLG(,PAS1),PASBLK INDICATE BLOCKING AND FIRST TIME 10350000
DFH042 CLC PASOHL(,PAS1),X200(2) IS RECORD LENGTH AT LEAST 512 ? 10360000
JNL DFH060 YES, OKAY 10370000
SPACE 1 10380000
***** ERROR FOR BLOCKING, RECORD LENGTH MUST 10390000
DFH045 B DFT030 BE AT LEAST 512. 10400000
DC AL1(TC08) TERMINATION CODE 10410000
SPACE 10420000
DFH050 MVC PASOHL(2,PAS1),PASTTS(,PAS1) MOVE IN OUTPUT HOLD LENGTH 10430000
ALC PASOHL(2,PAS1),X00FF INCREMENT TO SECTOR BOUNDARY 10440000
SBF PASOHL(,PAS1),ALLBIT SET OFF RIGHT MOST BITS 10450000
CLC PASOHL-1(1,PAS1),#TPPUT-1 SIZE GREATER THAN PUT AREA ? 10460000
JNH DFH060 NO, EVERYTHING OK 10470000
MVC PASOHL-1(1,PAS1),#TPPUT-1 MOVE IN NEW SIZE 10480000
B DFH040 GO INDICATE BLOCKING REQUIRED 10490000
SPACE 2 10500000
* INDICATE AMOUNT OF TEXT AREA REQUIRED IN TPBUFFER 10510000
DFH060 EQU * 10520000
TBN PASCPL+PLOPC(,PAS1),OPPUT+OPMSG PUT MESSAGE ? 10530000
JF DFH062 NO, BRANCH 10540000
TBN PASCPL+PLOPM(,PAS1),OPLIST PUT OVERRIDE ? 10550000
TBF PASFLG(,PAS1),PASBLK AND NOT A BLOCKING OPERATION 10560000
JF DFH063 NO, BRANCH 10570000
DFH062 MVC PASOHL(2,PAS1),PASTTS(,PAS1) MOVE IN REQUIRED TPBUFFER SPC 10580000
DFH063 EQU * 10590000
SPACE 10600000
L PASPL@(,PAS1),UPL2 POINT TO USER PARM LIST 10610000
MVC PLOUTL(2,UPL2),PASOHL(,PAS1) MOVE IN REQ. LENGTH 10620000
DFH070 EQU * 10630000
L PASTAR(,PAS1),IAR RETURN TO CALLER 10640000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.--INCREMENT DISK ADDR' 10650000
* * * * * 10660000
* 10670000
* INCREMENT THE DISK C/S ADDRESS IN IOB TO READ IN NEXT PART OF * 10680000
* THE FDT. * 10690000
* * 10700000
* * * * * 10710000
SPACE 2 10720000
DFI000 EQU * 10730000
SPACE 10740000
* XR1 SHOULD POINT TO PAS 10750000
ST PASXR@(,PAS1),ARR SAVE RETURN ADDRESS 10760000
SPACE 10770000
* IOBCHN IS UPDATED ONE SECTOR SINCE THIS FIELD WAS UPDATED BY 10780000
* DISK IOS TO POINT TO THE LAST SECTOR READ IN BY IT. 10790000

```

```

SPACE 10800000
DFI010 ALC PASIOB+IOBCHN(,PAS1),FOUR(1) INCR TO NEXT SECTOR 10810000
      TBN PASIOB+IOBCHN(,PAS1),X60 IS THIS NEW @ INVALID? 10820000
      JF DFI020 NO, OKAY 10830000
      ALC PASIOB+IOBCHN(,PAS1),X0020(2) YES, INCR TO NEXT TRACK @ 10840000
      SPACE 10850000
DFI020 MVC PASIOB+IOBSB(,PAS1),PASIOB+IOBCHN(2,PAS1) MOVE NEW @ TO 10860000
*      PROPER PLACE 10870000
      SPACE 2 10880000
      L PASXR@(,PAS1),IAR RETURN TO CALLER 10890000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.---READ FDT INTO PAS' 10900000
* * * * * 10910000
* * * * * 10920000
* READ 'FIELD DESCRIPTOR TABLE' )FDT). IF THE DESIRED ONE IS NOT * 10930000
* IN THE CURRENT PAS, THEN READ IT FROM DISK. * 10940000
* XR1-->PAS, XR2 IS NOT, NOT SAVED. * 10950000
* * * * * 10960000
* * * * * 10970000
      SPACE 2 10980000
DFJ000 ST PASJR@(,PAS1),ARR SAVE RETURN ADDRESS IN PAS 10990000
      L PASCT@(,PAS1),TT2 LOAD CURRENT TT ENTRY 11000000
      CLC TTFT@(,TT2),PASCFF@(2,PAS1) @ OF FT MATCH CURRENT ONE? 11010000
      JE DFJ060 YES, TEST FOR PUT OVERRIDE 11020000
      SPACE 2 11030000
* CURRENT FDT IN PAS NOT CORRECT ONE, READ PROPER ONE IN FROM DISK. 11040000
      SPACE 1 11050000
      L TTFT@(,TT2),FT2 POINT TO FT ENTRY 11060000
      SPACE 11070000
* Q BYTE FOR IOB SET UP ALREADY IN IOB BY ALLOCATE INITIALIZATION 11080000
      SPACE 11090000
      MVC PASIOB+IOBNB(,PAS1),FTFDTL(3,FT2) MOVE C/S OF FDT+ # SCTRS 11100000
      SPACE 1 11110000
DFJ030 SLC PASIOB+IOBNB(,PAS1),ONE(1) DECR TO #-1. 11120000
      MVC PASIOB+IOBDAT(,PAS1),PASFDT(2,PAS1) MOVE FDT @ 11130000
      ST PASCFF@(,PAS1),FT2 SAVE @ OF CURRENT FT ENTRY 11140000
      SPACE 11150000
      B DFV000 CALL DISK IOS AND WAIT 11160000
      SPACE 11170000
* -----START-----@02 11180000
DFJ060 L PASFDT(,PAS1),FDT2 POINT TO FDT 11190000
      CLI PASCPL+PLOPM(,PAS1),OPLIST IS OP CODE PUT OVERRIDE 11200000
      TBN FDTCFG(,FDT2),FDTPRT IS THIS A PRINTER GEN. FORMAT 11210000
      JC DFJ062,TANDEQ JUMP TRUE AND EQUAL 11220000
      J DFJ065 NO-CONTINUE 11230000
      SPACE 11240000
DFJ062 L PASCPL+PLRECA(,PAS1),LRA1 POINT TO OVERRIDE LIST 11250000
      LA WCC-1(,LRA1),LRA1 INCR PAST WCC 11260000
      CLI PORMOD(,LRA1),BLANKD OPERATION FIELD BLANK? 11270000
      JE DFJ065 JUMP EQUAL 11280000
      CLI PORMOD(,LRA1),MODDAT MODIFY DATA OPERATION? 11290000
      JE DFJ065 JUMP EQUAL 11300000
      SPACE 11310000
***** ERROR TYPE, CURSOR POSITION OR ERASE OPERATIONS ARE NOT VALID 11320000
* WHEN ISSUING A PUT OVERRIDE TO A PFGR FORMAT 11330000
      B DFT030 CALL TERMINATION CODE 11340000
      DC AL1(TC09) 11350000
      SPACE 11360000
DFJ065 L PAS@A1,PAS1 RELOAD PAS @ 11370000
* -----END-----@02 11380000
      L PASJR@(,PAS1),IAR RETURN VIA PAS 11390000

```

```

$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.---EXIT TO $CC4II' 11400000
* * * * * 11410000
* COMMON SECTION TO RETURN TO II'S MAINLINE CODE * 11420000
* * * * * 11430000
SPACE 2 11440000
DFFRTN EQU * 11450000
MVC DFFIRT+3(2),PAS1AR(,PAS1) MOVE IN RETURN ADDRESS 11460000
L PASPL@(,PAS1),UPL2 RELOAD USER'S PARM POINTER 11470000
L PASIX1(,PAS1),XR1 RESTORE REGISTER 1 11480000
DFFIRT B # RETURN TO II'S MAINLINE 11490000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.---' 'RELEASE TERMINAL'' 11500000
* * * * * 11510000
* 11520000
* RELEASE OP CODE LOGIC. * 11530000
* 11540000
* * * * * 11550000
SPACE 2 11560000
DFR000 EQU * 11570000
* CHECK TERMINAL NAME FOR VALIDITY AND FIND IT IF IT WAS PROCESSED 11580000
* PREVIOUSLY BY THIS PROGRAM. 11590000
SPACE 11600000
B DFF000 CALL TERMINAL TABLE ROUTINE 11610000
SPACE 11620000
* IF TERMINAL NAME IS NOT FOUND, THE HIGH-ORDER BYTE OF 'CURRENT' * 11630000
* ENTRY WILL BE ZERO 11640000
CLI PASCT@-1(,PAS1),ZEROD WAS MATCH ENTRY FOUND? 11650000
JE DFR020 NO, CONTINUE 11660000
SPACE 11670000
* THE FOLLOWING IS AN ENTRY POINT FROM DFG IF A TERMINAL WAS 11680000
* RELEASED BY AN TERMINAL OPERATOR COMMAND 11690000
DFR010 MVI TTNAME-TNAMLD(,TT2),BLANKD CLEAR NAME IN TT ENTRY 11700000
MVI TTFT@-1(,TT2),ZEROD BLANK OUT FT CHAIN POINTER 11710000
SPACE 11720000
* PREPARE TO RETURN TO $CC4II 11730000
DFR020 B DFFRTN RETURN TO II'S MAINLINE 11740000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.---CHECK FOR TERMINATION' 11750000
* * * * * 11760000
* 11770000
* TERMINATE A TASK. THIS ROUTINE WILL CHECK IF A TASK SHOULD * 11780000
* BE TERMINATED. IF IT SHOULD, A TERMINATION CODE SHOULD BE * 11790000
* PLACED IN THE TASK'S PASTC. IF NOT, THAT AREA SHOULD BE X'00'. * 11800000
* 11810000
* * * * * 11820000
SPACE 2 11830000
##### 11840000
* THE FOLLOWING INSTRUCTION IS THE ENTRY POINT FOR ERRORS WHICH WERE 11850000
* GENERATED WITH IN THIS PROGRAM, NOT FROM A TRANSIENT. THE ARR 11860000
* SHOULD BE POINTING TO A TERMINATION CODE. 11870000
DFT030 ST DFT050+4,ARR SAVE ADDR OF TERMINATION CODE 11880000
L PAS@A1,PAS2 LOAD PAS @ 11890000
DFT050 MVC PASTC(,PAS2),#(1) MOVE TERM CODE TO PAS 11900000
##### 11910000
SPACE 11920000
##### 11930000
* THE FOLLOWING INSTRUCTION IS THE ENTRY FOR CHECKING THE 11940000
* THE TERMINATION CODE AFTER A CALL TO THE TRANSIENTS 11950000
##### 11960000
DFT000 CLI PASTC(,PAS2),ZEROD NO TERMINATION CODE ? 11970000
JE DFT035 RIGHT, RETURN 11980000
SPACE 11990000

```

```

* AN ERROR WAS DETECTED FROM A TRANSIENT.. CALL $CC4DD TO CHECK      12000000
* IF MSG 528 SHOULD BE ISSUED.                                     12010000
SPACE                                                                12020000
DFT010 B      CC4PI          CALL TRANSIENT HANDLER                 12030000
DC      AL1(CC4DD)         TRANSIENT NAME                          12040000
SPACE                                                                12050000
DFT020 LA     PASTC(,PAS2),PAS2   LOAD @ OF TERMINATION CODE     12060000
ST     PASWRK-PASTC(,PAS2),PAS2  SAVE THE ADDRESS          12070000
L     PASWRK-PASTC(,PAS2),ARR    POINT THE ARR TO IT           12080000
SPACE                                                                12090000
DFT025 L     DFTI2,IAR          EXIT TO TERMINATION ROUTINE      12100000
DFTI2 DC     AL2(C4TI2)        ADDRESS OF TERMINATION ROUTINE    12110000
SPACE                                                                12120000
DFT035 ST     PASXR@(,PAS2),ARR   RETURN TO CALLER             12130000
ST     PAS@A1,PAS2             SAVE PAS @ IN CASE $CC4DB OR     12140000
*                               $CC4DB WERE CALLED                12150000
DFT040 L     PASXR@(,PAS2),IAR    RETURN TO CALLER             12160000
$CC4DF TITLE 'DISPLAY FORMAT CONTROL ROUTINE.--CALL DISK IOS + WAIT' 12170000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 12180000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 12190000
* CALL DISK IOS AND WAIT, AND CHECK FOR COMPLETION CODE           * 12200000
* XR1 WILL BE SAVE AND RESTORED. XR2 WILL BE DESTROYED.          * 12210000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 12220000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 12230000
SPACE 2                                                                12240000
DFV000 EQU   *                                                                12250000
ST     PASTAR(,PAS1),ARR       SAVE CALLER'S RETURN ADDRESS     12260000
LA     ZEROD(,PAS1),PAS2      SWITCH REGISTERS                   12270000
LA     PASIOB(,PAS2),IOB      LOAD IOB ADDRESS                   12280000
SBN    IOBQB(,IOB),X01        SET ON READ NBIT                   12290000
CCP    ENABLE,PMR             ENABLE INTERRUPTS                   12300000
SPACE                                                                12310000
SVC    0                      START DISK I/O                     12320000
DFV005 DC     XL1'02'          RIB FOR EXIO                       12330000
SVC    0                      WAIT ON I/O                        12340000
DC     XL1'03'          RIB FOR WAIT                             12350000
TBN    IOBECB(,IOB),POST     IS MY ECB POSTED ?                 12360000
BF     DFV005              NO, BRANCH                            12370000
SPACE                                                                12380000
CCP    DISABL,PMR           DISABLE INTERRUPTS                   12390000
CLI    IOBCMP(,IOB),IOBOK    NORMAL COMPLETION?                 12400000
ST     PAS@A1,PAS2          STORE PAS ADDRESS LOCALLY           12410000
JE     DFV020              YES, CONTINUE                         12420000
SPACE                                                                12430000
***** ERROR NON-NORMAL DISK COMPLETION                            12440000
DFV010 EQU   *                                                                12450000
B     DFT030              CALL TERMINATION CODE                  12460000
DC     AL1(TC07)          12470000
SPACE 2                                                                12480000
DFV020 LA     ZEROD(,PAS2),PAS1  SWITCH REGISTERS AGAIN         12490000
DFV040 L     PASTAR(,PAS1),IAR    RETURN TO CALLER              12500000
$CC4DF TITLE 'CHECK RETURN CODE OF OPERATION AFTER OP-END COMPLETE' 12510000
DFQ030 ST     DFQXR1+3,XR1       STORE CALLER'S XR1             12520000
ST     DFQRTN+3,ARR           STORE CALLER'S ARR                12530000
L     NCTCB@,TCB1            POINT TO TASK'S TCB                12540000
L     TCB@AS(,TCB1),PAS1      POINT TO PAS                     12550000
ST     PASWRK(,PAS1),UPL2     STORE PARM POINTER               12560000
MVC    PLINL(6,UPL2),PASCPL+PLINL(,PAS1) RESTORE PARM LIST     12570000
MVC    PLRECA(2,UPL2),PASCPL+PLRECA(,PAS1) AND ADDRESS         12580000
CLI    PLRTC(,UPL2),RCOK      GOOD RETURN CODE ?               12590000

```

	JE	DFQ080	YES, RETURN	12600000
		SPACE		12610000
*		NON-ZERO RETURN CODE, SO CHECK IF TERMINAL SHOULD BE TREATED		12620000
*		AS IF THE FORMAT IS NOT AT THE TERMINAL. IF SO BLANK OUT		12630000
*		THE TERMINAL NAME IN THE TT ENTRY TO INDICATE PROGRAM WAS		12640000
*		NOT COMMUNICATING TO TERMINAL.		12650000
		SPACE		12660000
	CLI	PLRTC(,UPL2),RCXCLR	'CLEAR' KEY GIVEN ?	12670000
	JE	DFQ050	YES, CLEAR TT ENTRY	12680000
	CLI	PLOPC(,UPL2),OPEAU+OPPUT	'EAU' REQUEST ?	12690000
	TBF	PLOPC(,UPL2),OPGET	OR A GET REQUEST ?	12700000
	JC	DFQ080,FOREQ	'EAU' OR INPUT, IGNORE	12710000
	TBN	PLOPM(,UPL2),OPLIST	PUT OVERRIDE OP ?	12720000
	JF	DFQ050	NO, CLEAR TT ENTRY	12730000
	TBN	PASWCC(,PAS1),WCCMDT	REQUEST TO RESET MDT ?	12740000
	JF	DFQ080	NO, LEAVE AS IS	12750000
DFQ050	L	PASCT@(,PAS1),TT2	POINT TO TT ENTRY	12760000
	MVI	TTNAME-TNAMLD(,TT2),BLANKD	CLEAR OUT TT ENTRY	12770000
DFQ080	L	PASWRK(,PAS1),UPL2	RELOAD USER PARM	12780000
DFQXR1	LA	#,XR1	RELOAD CALLER'S XR1	12790000
DFQRTN	B	#	RETURN TO CALLER	12800000
.END	ANOP			12810000
	MEND			12820000


```

MACRO 00010000
***** 00020000
* 00030000
* NAME: $E070 00040000
* 00050000
* MODIFICATION LEVEL: MODEL 15, VERSION 7, MOD LEVEL 0 00060000
* 00070000
* FUNCTION: 00080000
* 00090000
* . CCP GENERATION SECOND PASS MACRO-INSTRUCTION -- GENERATE THE 00100000
* COMMUNICATION MANAGEMENT TASK. (01 OF 12). 00110000
* 00120000
* INPUT OPERANDS: NONE. 00130000
* 00140000
* SYSTEM GLOBALS: 00150000
* 00160000
* &ONE - MIXTURE OF MLTA AND BSCA. 1=ONLY ONE / 2=BOTH. 00170000
* &NDME- DATA MODE ESCAPE. 1=NOT SUPPORTED / 0=SUPPORTED. 00180000
* &MIN - MINIMUM SYSTEM. 1=MINIMUM / 0=NOT MINIMUM. 00190000
* &NPBY- BUSY PRINTER SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED 00195000
* BSCA: 00200000
* &NOB - BSCA SUPPORTED. 1=NOT SUPPORTED / 0=SUPPORTED. 00210000
* &BSC - NUMBER OF BSCA LINES. NUMBER OF BSCA LINES (DTFS). 00220000
* &NPP - BSCA PT-TO-PT SUPPORTED. 1=NOT SUPPORTED / 0=SUPPORTED. 00230000
* &NMP - BSCA MULTI-PT TRIBUTARY. 1=NOT SUPPORTED / 0=SUPPORTED. 00240000
* &NSWL- BSCA SWITCHED SUPPORTED. 1=NOT SUPPORTED / 0=SUPPORTED. 00250000
* &NCS - BSCA CONTROL STATION. 1=NOT SUPPORTED / 0=SUPPORTED. 00260000
* &NITB- BSCA ITB SUPPORTED. 1=NOT SUPPORTED / 0=SUPPORTED. 00270000
* &NTSP- BSCA TRANSPARENCY. 1=NOT SUPPORTED / 0=SUPPORTED. 00280000
* &NAS - BSCA ASCII SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00290000
* &N32 - BSCA 3270 SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00300000
* &NRUF- 3270 PGM REQUEST UNDER FMT. 1=NOT SUPPORTED / 0=SUPPORTED. 00305000
* &NDF - BSCA 3270 DFF SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00310000
* &N37 - BSCA 3735 SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00320000
* &N41 - BSCA 3741 SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00325000
* &NCPU- BSCA CPU SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00330000
* &NMSG- BSCA GET MSG SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00340000
* &NINT- BSCA INTERVAL POLLING. 1=NOT SUPPORTED / 0=SUPPORTED. 00350000
* MLTA: 00360000
* &NOM - MLTA SUPPORTED. 1=NOT SUPPORTED / 0=SUPPORTED. 00370000
* &MLA - NUMBER OF MLTA LINES. NUMBER OF MLTA LINES (DTFS). 00380000
* &NSW - MLTA SWITCHED LINES. 1=NOT SUPPORTED / 0=SUPPORTED. 00390000
* &NSCTL-MLTA STATION CONTROL. 1=NOT SUPPORTED / 0=SUPPORTED. 00400000
* &NBFR -MLTA BUFFERED RECEIVE. 1=NOT SUPPORTED / 0=SUPPORTED. 00410000
* &N1050-MLTA 1050 SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00420000
* &N2741-MLTA 2741 SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00430000
* &NMOVE-MLTA MOVE ONLY SUPPORT. 1=NOT SUPPORTED / 0=SUPPORTED. 00440000
* 00450000
* MODULE LOCALS: 00460000
* 00470000
* &MIX - WORK AREA TO DETERMINE THING IN COMBINATION. 00480000
* @39 - APAR 4915 CHECKING WRONG DA BIT BERG ANSWERED DFD FIXED 00490000
***** 00500000
$E070 00510000
GBLB &ONE , &NOB , &NOM , &NDF , &NCS , &NSWL , &NINT , &N32 , &NPBY 00530000
GBLA &BSC , &MLA 00570000
LCLA &MIX 00590000
TEXT 00600000
* R-06 , C-00 CHANGE LEVEL 00610000

```

CMBEG	EQU	*			00613000
	ENTRY	CMBEG			00616000
CMEYE	DC	CL3 '\$CM'	EYE CATCHER		00620000
	DC	CL2 '07'	MOD LEVEL		00630000
&MIX	SETA	&NOB+&NOM			00640000
	AIF	(&MIX EQ '0').START			00650000
&ONE	SETB	1			00660000
.START	ANOP				00670000
	SPACE	2			00680000
	ENTRY	\$CC4CM	ENTRY POINT FOR CM		00690000
	ENTRY	\$CC4FR	COMMON PL/RECA FREEMAIN ENTRY		00700000
*			* POINT FOR TRANSIENT AND CM.		00710000
	SPACE	1			00720000
*	ENTRY POINTS	IN CCP \$SBMCH USED FOR	INITIALIZATION BY MLTA AND MLMP		00730000
	SPACE	1			00740000
	ENTRY	\$SBMCH	ENTRY FOR CCP VERSION OF COMMON		00750000
*			* MLMP/MLTA CHECK ROUTINE.		00760000
	ENTRY	CHBMHL	LOCATION FOR MLMP/MLTA STORE		00770000
*			* BRACKET HALT (NOT ISSUED)		00780000
	AIF	(&NOM).M0001			00790000
*	ENTRY	CHBMML	ENTRY WHERE MLTA STORES ADDR	M	00800000
			* OF MLTA CHECK ROUTINE	M	00810000
	ENTRY	CHBMPC	ENTRY WHERE MLTA STORES ADDR	M	00820000
*			* FOR PCI INTERRUPTS (NOT	M	00830000
*			* EXITED TO UNDER CCP)	M	00840000
.M0001	ANOP				00850000
	AIF	(&NOB).C0001			00860000
	ENTRY	CHBMBS	ENTRY WHERE MLMP STORES ADDR	B	00870000
	ENTRY	SAVCAT	SAVE AREA FOR CAR AND TAR		00875000
*			* OF BSCA WAIT ROUTINE.	B	00880000
	SPACE	1			00890000
	AIF	(&NSWL).C0000			00900000
	ENTRY	CMCLOZ	ADDR WHERE STARTUP PLUGS MLMP.	SB	00910000
*			* CLOSE AND OPEN C/S ADDRS.	SB	00920000
.C0000	ANOP				00930000
	SPACE				00940000
*	BSCA ONLY	EXTRNS:			00950000
	EXTRN	\$SBSMS	SIO ENTRY FOR MLMP	B	00960000
	EXTRN	\$CC4BT(3)	EXTRN SO ENTRY TO TRACE WILL	B	00970000
	EXTRN	\$CC4M1(3)	EXTRN FOR MINI-MLMP		00975000
	EXTRN	\$CC4M2(3)	EXTRN FOR PRIVILEGED MINI-MLMP	B	00977000
*			* NOT CAUSE LKED HALT	B	00980000
.C0001	ANOP				00990000
	AIF	(&NOM).C0002			01000000
	SPACE	1			01010000
*	MLTA ONLY	EXTRNS:		M	01020000
	EXTRN	MLTIO1	MLTA ENTRY POINT	M	01030000
.C0002	ANOP				01040000
	TITLE	'\$E070/\$CC4CM/CONTROL LOGIC '			01050000
*****					01060000
*	C O M M U N I C A T I O N S	M A N A G E R		*	01070000
*****					01080000
	SPACE	2			01090000
*****					01100000
*				*	01110000
*	MODULE	- \$CC4CM		*	01120000
*				*	01130000
*	FUNCTION	- SCHEDULE AND CONTROL ALL TELEPROCESSING COMMUNICATION		*	01140000
*		TO AND FROM MLTA AND BSCA TERMINALS ATTACHED TO CCP.		*	01150000
*		SCHEDULE AND CONTROL ALL I/O TO THE SYSTEM CONSOLE.		*	01160000

*		* 01170000
*	ENTRY POINTS :	* 01180000
*	\$CC4CM - MAIN ENTRY POINT.	* 01190000
*	DISPATCHED BY STARTUP AT THIS POINT. AFTER INITIAL	* 01200000
*	DISPATCH \$CC4CM WILL WAIT FOR MORE WORK JUST	* 01210000
*	FOLLOWING THIS POINT AND WILL RECEIVE CONTROL	* 01220000
*	AT THIS WAIT POINT AFTER OP END OR NEW REQUEST	* 01230000
*	CAUSES \$CMECB TO BE POSTED.	* 01240000
*		* 01250000
*	\$CC4FR - FREEMAIN PL AND HOLD BUFFER ENTRY POINT.	* 01260000
*	BRANCHED TO BY \$CC4CM AND CM TRANSIENTS.	* 01270000
*		* 01280000
*	GENERATION - THIS MODULE IS ASSEMBLED AT CCP GENERATION.	* 01290000
*	\$CC4CM IS CREATED BY MACROS WHICH ARE NAMED \$E0NN WHERE	* 01300000
*	NN IS A NUMERIC IDENTIFIER OF THE MACRO.	* 01310000
*		* 01320000
*	CODE IN EACH MACRO IS EITHER INCLUDED OR EXCLUDED BASED	* 01330000
*	ON THE GLOBALS SET BY THE FIRST PASS OF GENERATION. THE	* 01340000
*	CODE IS IDENTIFIED IN COLUMNS 69-71 BY ALPHABETICS WHICH	* 01350000
*	IDENTIFIES THE GENERATION OPTION THAT CAUSES ITS INCLUSION	* 01360000
*		* 01370000
*	THE ALPHABETIC IDENTIFIERS ARE AS FOLLOWS:	* 01380000
*		* 01390000
*	COLUMN 71 :	* 01400000
*	B - BSCA	* 01410000
*	M - MLTA	* 01420000
*	2 - TEST FOR MLTA OR BSCA	* 01430000
*		* 01440000
*	COLUMN 68-70 MAY INCLUDE IN ANY ORDER OR COMBINATION:	* 01450000
*	C - BSCA OR MLTA CONTROL STATION OR STATION CONTROL	* 01470000
*	D - BSCA OR MLTA DATA MODE ESCAPE	* 01480000
*	L - CODE NECESSARY FOR COMBINATION OF LINE TYPES	* 01500000
*	MIN - CODE PRESENT ON MINRES SYSTEM ONLY.	* 01505000
*	R - RESIDENT CODE MADE TRANSIENT FOR MINRES-YES.	* 01510000
*	S - BSCA OR MLTA SWITCHED LINE SUPPORT.	* 01520000
*	Y - CODE TO HANDLE MULTIPLE DTFS. EXCLUDED IN 1 LINE SYS.	* 01530000
*	/ - OR (FOR COMBINATION OF 2 OTHER IDENTIFIERS	* 01535000
*		* 01540000
*	A - BSCA ASCII SUPPORT	* 01590000
*	F - BSCA 3270 DFF SUPPORT	* 01610000
*	G - BSCA GET MESSAGE SUPPORT	* 01640000
*	I - BSCA ITB (INTERMEDIATE TEXT BLOCK) SUPPORT	* 01645000
*	N - BSCA INTERVAL POLLING	* 01650000
*	P - BSCA POINT TO POINT SUPPORT	* 01651000
*	T - BSCA MULTI POINT TRIBUTARY SUPPORT	* 01652000
*	U - BSCA CPU TO CPU SUPPORT	* 01653000
*	W - BSCA 3270 PROGRAM REQUEST UNDER FORMAT (PRUF)	* 01654000
*	X - BSCA TRANSPARENCY SUPPORT	* 01655000
*	0 - BSCA 3270 SUPPORT	* 01656000
*	5 - BSCA 3735 SUPPORT	* 01657000
*	7 - BSCA 3741 SUPPORT	* 01658000
*		* 01660000
*	B - MLTA BUFFERED RECEIVE SUPPORT	* 01670000
*	V - MLTA MOVE SUPPORT (NO TRANSLATE)	* 01675000
*	1 - MLTA 1050 SUPPORT	* 01680000
*	4 - MLTA 2741 SUPPORT	* 01690000
*		* 01710000
*	EXTERNAL REFERENCES:	* 01720000
*	\$CCCOM:	* 01730000
*	FIELDS IN \$CMWK	* 01740000

*	@CMTCB,@CPTCB	*	01750000
*	@CKLST	*	01760000
*	@LCB#1	*	01770000
*	@PRLQ		01780000
*	#CMERP	*	01790000
*	#GMS	*	01800000
*	#OPEND	*	01810000
*		*	01820000
*	CHANGE ACTIVITY - \$CC4CM.	*	01820500
*	RELEASE 4:	*	01829400
*	@16-INCR/OES -IDA.	*	01829600
*	RELEASE 5:	*	01829620
*	@18-APAR/S307609-CORRECTS ID VERIFICATION ON 3275 SWITCHED LINE.	*	01829640
*	@19-APAR/S307681-CORRECTS AN INVALID CLEAR KEY INDICATION WITH	*	01829660
*	3270 BLOCKED DATA.	*	01829680
*	@20-APAR/S307636-FIXES U-PC AFTER 'C CCP' COMMAND BY CHECKING	*	01829700
*	CORRECT TERMINATION BIT IN TCB.	*	01829720
*	@21-APAR/S307541-FREES TERMINALS WITH SKIP BIT ON AND NOT BEING	*	01829740
*	POLLED OR IN ERP.	*	01829760
*	@22-APAR/S307691-CORRECTS INVALID 'E28 PROG NAME' MESSAGES.	*	01829780
*	@23-APAR/S306901-CORRECTS U-FE WITH EXCESSIVE PRUF REQUEST DATA.	*	01829800
*	@24-APAR/S307736-CORRECTS POLLING STOPPING WITH WAIT TIME OF 1.	*	01829820
*	@25-APAR/S307796-FIXES MESSAGE 411, RETURN CODE -08 TXT TIMEOUT.	*	01829840
*	@26-APAR/S308023-CORRECTS A LOOP IN \$CC4OP BECAUSE A USER'S	*	01829860
*	RECORD AREA WAS BLANKED WITH AN INVALID LENGTH.	*	01829880
*	RELEASE 6:	*	01829881
*	@27-APAR/S308742-CORRECTS LOOP AFTER SHUTDOWN COMMAND DUE TO	*	01829882
*	4E COMPLETION CODE.	*	01829883
*	@28-APAR/S308381-RECORD MODE TERMINALS ALWAYS GO TO MLMP.	*	01829884
*	@29-APAR/S309103-FREEMAINED SWITCHED AREA PRIOR TO SWITCHED CHK.	*	01829885
*	@30-APAR/ -POSTED WRONG TASK WHEN IN TERMINATION.	*	01829886
*	@31-APAR/S309265-GET WITH RVI FAILS TO SEND RVI	*	01829887
*	@32-APAR/S309066-WAIT TIME EXCEEDED IF LARGE BLOCKS AND SMALL	*	01829888
*	RECORDS IN THE INITIAL BLOCK TRANSMITTED	*	01829889
*	@33-APAR/ -WRONG TERMINAL POLLED ON CPU-TO-CPU TERMINAL	*	01829890
*	CONFIGURATION	*	01829891
*	@34- -ADD TRACE SUBROUTINE FOR PERFORMANCE	*	01829892
*	@35- -USER REC AREA BLANKED IF GET AND CLEAR KEY	*	01829893
*	@36- -LOOP IF T.O. INTERRUPT AND PARM LIST QUEUE	*	01829894
*	EMPTY.	*	01829895
*	RELEASE 7:	*	01829900
*	@37-APAR/S311 -STOP INVITE AND CLEAR KEY TRANSLATES LOW CORE.	*	01829920
*	@38-INCR/ES0705 -SUPPORT FOR BUSY 328X PRINTERS	*	01829940
*		*	01829960
*	*****	*	01830000
*	EJECT		01840000
*	*****	*	01850000
*	\$CC4CM ENTRY POINT	*	01860000
*	*****	*	01870000
*	SPACE 2		01880000
\$CC4CM EQU *	'CM' ENTRY POINT.		01890000
SBF CMSWIT,CMSPSI+CMFMPS	RESET STOP II AND FM POST FLAGS		01900000
SBN #CMSWT,#CMFMD	RESET \$CC4GR TO 1ST LCB.		01910000
LA ECBLST,XR1	ADDRESS OF ECB WAIT LIST		01920000
SVC 0	WAIT FOR WORK - RECEIVE CONTROL		01930000
DC AL1(WAITRB)	* WHEN:		01940000
*	* NEW TP REQUEST (\$CC4II/IS)		01950000
AIF (&NDF).D0030			01960000
*	* (\$CC4DF) FB		01970000
.D0030 ANOP			01980000

```

*          *          TP OP END          ($CC4IB/IM) 01990000
*          *          FREEMAIN DONE AND          02000000
*          *          REQUEST WAITING           02010000
*          *          FOR SPACE.          ($CC4FM) 02020000
*          *          SET OFF WAIT BITS TO PREVENT TASK FROM BEING DISPATCHED AT
*          *          ANOTHER WAIT POINT BECAUSE OF A POST TO ONE OF THESE ECBS. 02030000
*          *          SPACE 1                  02040000
*          *          TBN $CMFPM,POST          IF FREEMAIN HAS OCCURRED AND          02050000
*          *          CLI CORCNT,NOBIT        * CM IS WAITING FOR ONE,          02060000
*          *          JC CMCKCN,FLSOEQ        NO-GO CHECK COUNT                  02070000
*          *          SBN CMSWIT,CMFMPS      YES- SET INTERNAL FREEMAIN BIT. 02080000
*          *          CMCKCN EQU *            * LOCAL                          02090000
*          *          SBF $CMFPM,WAIT+POST    RESET FREEMAIN ECB WAIT/POST.        02100000
*          *          SBF $CMECB,WAIT+POST    RESET GENERAL ECB WAIT/POST.        02110000
*          *          AIF (&NINT).C0008      02120000
*          *          MVI TIMIOB+7,SKIP      SET TO SKIP TIMER POSTING      NB 02130000
*          *          .C0008 ANOP              02140000
*          *          JNE CMNOZ0              IF CORE COUNT > 0 - JUMP          02150000
*          *          MVI $CMFPM,SKIP        SET TO SKIP FREE MAIN POST        02160000
*          *          CMNOZ0 EQU *            *                                02162000
*          *          *                      *                                02164000
*          *          *                      *                                02166000
*          *          *                      *                                02170000
*          *          *                      *                                02180000
*          *          *                      *                                02190000
*          *          *                      *                                02230000
*          *          *                      *                                02240000
*          *          *                      *                                02250000
*          *          *                      *                                02260000
*          *          *                      *                                02270000
*          *          *                      *                                02280000
*          *          *                      *                                02290000
*          *          *                      *                                02300000
*          *          *                      *                                02310000
*          *          *                      *                                02320000
*          *          *                      *                                02330000
*          *          *                      *                                02340000
*          *          *                      *                                02350000
*          *          *                      *                                02360000
*          *          *                      *                                02370000
*          *          *                      *                                02380000
*          *          *                      *                                02390000
*          *          *                      *                                02400000
*          *          *                      *                                02410000
*          *          *                      *                                02420000
*          *          *                      *                                02430000
*          *          *                      *                                02440000
*          *          *                      *                                02450000
*          *          *                      *                                02460000
*          *          *                      *                                02470000
*          *          *                      *                                02480000
*          *          *                      *                                02490000
*          *          *                      *                                02500000
*          *          *                      *                                02505000
*          *          *                      *                                02510000
*          *          *                      *                                02520000
*          *          *                      *                                02530000
*          *          *                      *                                02540000
*          *          *                      *                                02545000
*          *          *                      *                                02550000
*          *          *                      *                                02560000
*          *          *                      *                                02570000

```

```

*          CMSDTF - DTF TO BE HANDLED.                * 02580000
*          CMSPL  - PL TO BE HANDLED.                * 02585000
*                                                    * 02590000
* EXTERNAL REFERENCES--                               * 02600000
*          CMDTFS - SETUP BSCA DTFS FOR CHECK ROUTINE * 02610000
*          CMBMCH - CHECK ROUTINE                    * 02620000
*          CMBSKP - BSCA SKIP BIT ON/OFF ROUTINE.    * 02630000
*          $CC4GR - GETMAIN FAIL RECOVERY TRANSINET. * 02632000
*          CMPRLS - SEARCH @PRLQ FOR NEW PL.         * 02634000
*          $@TMIH - START/STOP INTERVAL TIMER.      * 02636000
*          $TRACE - SYSTEM TRACE ROUTINE.           * 02638000
*          CMTRCE - TRACE SUBROUTINE IN $E095       * 02639000
*                                                    * 02640000
*          EXIT, NORMAL--                             * 02650000
*                                                    * 02660000
*          TO CMBOPE OR CMMOPE TO HANDLE OP END     * 02670000
*          TO CMACI IF NEW ACCEPT REQUEST.          * 02672000
*          TO CMBREQ OR CMMREQ TO HANDLE NEW REQUESTS. * 02674000
*          TO CMBSOX OR CMPNWX IF GETMAIN RETRY AT QUEUE TIME. * 02676000
*          TO CMBTBY OR CMMTBY IF GETMAIN RETRY AT SCHED TIME. * 02680000
*                                                    * 02690000
*****
          SPACE 5                                     02692000
*****
*          CHECK FOR IOCS OP ENDS.                    * 02694000
*****
          SPACE 1                                     02696000
          CMOPND EQU *                                02698000
          CLI #OPEND,NOBIT TEST OP END COUNT          02700000
          BE CMFRMN CHECK OP END COUNT FOR ZERO      02710000
          SPACE IF ZERO, BR TO FREEMAIN ANALYSIS    02720000
          AIF (&NOB).C0010                            02730000
***** B 02740000
*          SETUP ANY BSCA DTF WITH OP END            * B 02750000
***** B 02760000
          SPACE 1                                     B 02770000
          B CMDTFS DTF SETUP FOR BSCA LINES         B 02780000
          SPACE 1                                     B 02790000
          .C0010 ANOP                                  B 02800000
***** B 02810000
*          CALL CHECK ROUTINE (CCP VERSION OF $BMCH) * B 02820000
***** B 02830000
          SPACE 1                                     B 02840000
          B CMBMCH CALL OP END CHECK ROUTINE        B 02850000
          SPACE                                       B 02860000
          SLC #OPEND,X$0001(1) SUBTRACT 1 FROM OP END COUNT B 02870000
          AIF (&ONE).H4450                            B 02880000
          TBF CMSWIT,CMBSCK IS IT MLTA OR BSCA      B 02890000
          CLI $BDCMP(,DTF), $BCACD CHECK FOR NOTHING COMPLETE B 02900000
          BC CMOPND,NONE+HI+LO+FALSE BRANCH IF MLTA NOT COMPLETE B 02910000
          .H4450 ANOP                                  B 02920000
          AIF (&NOB).C0015                            B 02930000
          CLI $BDCMP(,DTF), $BCCMP '56' COMPLETION CODE ? B 02940000
          JNE CMNT56 NO-NOT 56, CHECK OTHERS.      B 02950000
          SPACE                                       B 02960000
&MIX SETA (&BSC+&MLA)                                B 02970000
          AIF (&MIX LE '1').Y0100                    B 02980000
*-----* YB 02990000
*          NO OP ENDS TO HANDLE - '56' CODE FROM CHECK ROUTINE * YB 03000000
*-----* YB 03010000
          SPACE 1                                     YB 03020000
          SPACE 1                                     YB 03030000
          SPACE 1                                     YB 03040000

```

* SEARCH CHECKLIST FOR BSC DTF AND DECREMENT ITS OP END COUNT	YB	03050000
SPACE 1		03060000
L @CKLST, XR1 POINT TO THE CHECK LIST.	YB	03070000
CMEXSK EQU * * LOCAL	YB	03080000
TBF CKLSTS(, XR1), CKLSKP SKIP BIT OFF, AND	YB	03090000
L CKLDTF(, XR1), DTF POINT TO THE DTF.	YB	03100000
AIF (&ONE).C0007		03110000
TBN \$BDDEV(, DTF), BSCA * BSCA DTF, AND	Y2	03120000
.C0007 ANOP		03130000
JF CMEXNX NO-GO EXAMINE NEXT ENTRY.	YB	03140000
.Y0100 ANOP		03150000
SLC LCBOPE(1, DTF), X\$0001 DECREMENT OP END BY ONE.	B	03160000
L LCBPL@(, DTF), PL POINT TO THE PARM LIST.	B	03170000
B CMTRCE CALL TRACE SUBROUTINE FOR THIS	B	03180000
* OP END	B	03185000
DC AL1(CCP RIB) CCP RIB		03190000
DC AL1(TRRIB) TRACE SUB RIB		03200000
DC AL1(TTMOPN) TRACE OP END ID.	B	03210000
SPACE 1		03220000
&MIX SETA (&BSC+&MLA)		03230000
AIF (&MIX LE '1').Y0250		03240000
J CMEXOP GO CHECK FOR MORE OP ENDS.	YB	03250000
SPACE 1		03260000
CMEXNX EQU * * LOCAL	YB	03270000
TBN CKLSTS(, XR1), CKLAST LAST ENTRY IN THE LIST ?	YB	03280000
LA CKLEN(, XR1), XR1 POINT TO THE NEXT ENTRY.	YB	03290000
BF CMEXSK NO-GO CHECK THE NEXT	YB	03300000
.Y0250 ANOP		03310000
CMEXOP EQU * * LOCAL	B	03320000
B CMOPND GO CHECK FOR MORE OP ENDS.	B	03330000
SPACE 1		03340000
CMNT56 EQU * * LOCAL	B	03350000
-----		03480000
* OP END TO HANDLE (NOT '56') - GO TO BSCA OR MLTA OP END ROUT.*		03490000
-----		03500000
SPACE 1		03510000
ST CMSDTF, XR2 SAVE DTF ADDR	B	03520000
AIF (&ONE).C0012		03530000
TBN \$BDDEV(, DTF), BSCA BSCA DTF ?	2	03540000
BF CMMOPE NO-GO TO CHECK MLTA ABORT.	2	03550000
.C0012 ANOP		03560000
SLC LCBOPE(1, DTF), X\$0001 DECREMENT LINE OP END COUNT.	B	03580000
L LCBPL@(, DTF), PL POINT TO THE PARM LIST.	B	03590000
&MIX SETA &NCS+&NSWL		03600000
AIF (&MIX EQ '2').S0150		03610000
TBN LCBAT2(, DTF), LCBRCI RECEIVE INITIAL ?	B	03620000
BF CMBOPE NO-GO CHECK BSCA ABORT.	B	03630000
MVI CMB#SB, SBN1 SET FOR POLL SKIP BIT ON.	B	03640000
SPACE		03650000
B CMBSKP SKIP TO POLL SKIP BIT RTN.	B	03660000
.S0150 ANOP		03670000
B CMBOPE GO HANDLE BSCA OP END	B	03680000
AGO .C0025		03680900
.C0015 ANOP		03681800
SPACE 1		03682700
* XR2 NOW POINTS AT DTF FOR WHICH OP END OCCURRED	M	03683600
* ANALYZE THE COMPLETION CODE AND ACT ACCORDINGLY	M	03684500
SPACE		03685400
CLI \$MDCMP(, XR2), \$MCRET EVENT NOT COMPLETE	M	03686300
BNL CMOPND RETEST OP END COUNT.	M	03687200

ST	CMSDTF, XR2	SAVE DTF ADDR	03688100
B	CMMOPE	GO HANDLE THE OP END.	M 03689000
.C0025	ANOP		03690000
\$CC4CM	TITLE '\$E070/CMFRMN/ CONTROL LOGIC - GETMAIN FAIL RECOVERY'		03700000
*****	*****	*****	03710000
*	CHECK FOR NEW PUT REQUEST.	*	03720000
*****	*****	*****	03730000
	SPACE 1		03770000
CMFRMN	EQU *	FREEMAIN ANALYSIS ROUTINE	04140000
CLI	@PRLQ-1, NOBIT	ANY NEW TP REQUESTS ?	04150000
JE	CMFRPS	NO - CHECK FOR FREEMAINS	04160000
	SPACE		04170000
*	HAVE TP REQUEST -IF PUT REQUEST HANDLE IT BEFORE CHECKING TO SEE		04180000
*	IF REQUESTS WAITING FOR FREEMAIN CAN BE SATISFIED NOW.		04190000
	SPACE		04200000
B	CMRPLS	SEARCH PRLQ FOR PUT	04210000
*		* XR1 - PL FOUND	04220000
*		* XR2 - LEFT BYTE OF CHAIN	04230000
*		* ADDR POINTING TO PL FOUND.	04240000
TBN	PLOPC(, PL), OPPUT	DOES IT INVOLVE PUT	04250000
JT	CMRDEQ	YES - JUMP TO HANDLE NOW	04260000
	SPACE		04261000
*****	*****	*****	04262000
*	CHECK FOR REQUEST WAITING GETMAIN THAT CAN BE RETRIED.	*	04263000
*****	*****	*****	04264000
	SPACE 1		04265000
CMFRPS	EQU *	* LOCAL	04270000
TBN	CMSWIT, CMFMPS	HAS FREEMAIN BEEN POSTED	04280000
JF	CMREQ	IF NOT, JUMP TO HANDLE TP REQST	04290000
	SPACE		04292000
*	HAVE HAD FREEMAIN - FIND REQUEST AWAITING TPBUF SPACE.		04294000
	SPACE		04296000
SVC	0	***** TRANSIENT CALL *****	04350000
DC	AL1(CCPRIB)	CCP RIB	04360000
DC	AL1(CC4GR)	GETMAIN RECOVERY TRANSIENT	04370000
	SPACE 1		04380000
*	---> RETURN HERE IF NO REQUEST NEEDING GETMAIN WAS FOUND		04390000
J	CMFMGG		04400000
	SPACE 1		04410000
*	---> RETURN HERE IF REQUEST NEEDS TO BE RETRIED (XR1 POINTS TO IT)		04420000
*	XR1-PL ADDRESS, XR2-DTF TO BE RESCHEDULED.		04430000
ST	CMSDTF, DTF	SAVE THIS DTF @	04440000
ST	CMSPL, PL	SAVE THIS PL @	04450000
TBN	PL\$OPM(, PL), OPGETQ	DOES REQ NEED GM AT QUEUE TIME	04460000
SBF	PL\$OPM(, PL), OPGETM+OPGETQ	RESET GETMAIN NEEDED BITS	04470000
AIF	(&ONE).E0403		04480000
JF	CMGMBF	NO- NEED BUFFER AT SCHED TIME	04490000
AGO	.C0175		04500000
.E0403	ANOP		04510000
AIF	(&NOM).M0780		04520000
BF	CMMTBY	NO- NEED BUF AT SCHED TIME.	M 04530000
AGO	.C0175		04540000
.M0780	ANOP		04550000
BF	CMBTBY	NO- NEED BUF AT SCHED TIME.	B 04560000
.C0175	ANOP		04570000
	SPACE 1		04580000
*	GETMAIN NEEDED AT REQUEST QUEUE TIME		04590000
ST	CMNWPL, PL	SAVE ORIG PL @ FOR NO WAIT POST	04600000
B	CMDEQ	DEQUEUE PL IN TCB	04610000
SBN	CMSWIT, CMTPRQ	NEW REQUEST	04620000

	AIF	(&ONE).E0405			04630000
	TBN	\$BDDEV(,DTF),BSCA	IS REQUEST FOR BSCA	2	04640000
	BT	CMBSOX	YES - GO HANDLE	2	04650000
.E0405	ANOP				04660000
	AIF	(&NOM).M0790			04670000
	B	CMPNWX	NO - MLTA NEW REQUEST	M	04680000
	AGO	.C0200			04690000
.M0790	ANOP				04700000
	AIF	(&NOB).C0200			04710000
	B	CMBSOX	BSCA NEW REQUEST	ONLY B	04720000
.C0200	ANOP				04730000
	AIF	(&ONE).E0410			04740000
	SPACE	1			04750000
*	GETMAIN	NEEDED AT NEW REQUEST	SCHEDULE TIME		04760000
CMGMBF	EQU	*	* LOCAL		04770000
	TBN	\$BDDEV(,DTF),BSCA	BSCA DTF ?	2	04780000
	BT	CMBTBY	YES - GO SEE IF LINE IS BUSY	2	04790000
	B	CMMTBY	JUMP IF NOT BUSY OR NON-READ	M	04800000
.E0410	ANOP				04810000
	SPACE	1			04820000
CMFMGG	EQU	*	* LOCAL		04830000
	SBF	CMSWIT,CMFMPS	TURN OFF FREEMAIN POSTED FLAG		04840000
	SPACE	2			04850000
*****					04860000
*	CHECK	FOR NEW READ REQUEST	ON @PRLQ.	*	04870000
*****					04880000
	SPACE	1			04890000
CMREQ	EQU	*	HANDLE TP REQUESTS		04920000
	CLI	@PRLQ-1,NOBIT	ANY NEW TP REQUESTS TO HANDLE ?		04930000
	JE	CMNWRK	NO - CHECK FOR TERMINAL IN		04940000
*			ERROR RECOVERY		04950000
	SPACE	1			04960000
*	CM	HAS BEEN POSTED WITH A TP REQUEST,	FIND THE NEW REQ AND HANDLE IT.		04970000
	SPACE				04980000
	B	CMPRLS	SEARCH PRLQ FOR PUT		04990000
*			* XR1 - PL FOUND		05000000
*			* XR2 - LEFT BYTE OF CHAIN		05010000
*			* ADDR POINTING TO PL FOUND.		05020000
CMRDEQ	EQU	*	* LOCAL		05030000
	MVC	1(2,XR2),PLCHN(,PL)	DEQUEUE PL OFF PRLQ		05040000
	SLC	PLCHN(2,PL),PLCHN(,PL)	ZERO CHAIN FIELD		05050000
	SPACE	1			05060000
	ST	CMSPL,PL	SAVE THE PARM LIST ADDRESS		05070000
	ST	CMNWPL,PL	SAVE PL TO POST FOR NO WAIT OP		05080000
	L	PLTUBA(,PL),XR2	POINT XR2 AT THE TUB		05090000
	AIF	(&N32).C0305			05090200
	AIF	(&ONE).C0303			05090300
	TBN	TUBCHR(,XR2),TUBLNE	BSCA LINE AND -	B	05090400
.C0303	ANOP				05090500
	AIF	(&NOB).C0305			05091000
	TBN	PLOPC(,PL),OPGET	INVITE/GET OP ?	B	05092000
	TBF	PLOPM(,PL),OPSTOP	AND NOT A STOP INVITE?		05092500
	JF	CMRSCL	NO - CONTINUE	B	05093000
	SPACE	1			05094000
*	SET	OFF THE CLEAR INDICATER IN THE TUB	FOR ANY GET TYPE OP	B	05095000
	SPACE	1			05096000
	SBF	TUBSCS(,XR2),TUBCLR	SET OFF CLEAR INDICATER	B	05097000
*				@38	05097030
*	BUSY	PRINTER SUPPORT		@38	05097060
*				@38	05097090

```

AIF (&NPBY).NBY01          BUSY PRINTER SUPPORTED      @38 05097120
SPACE 1                    @38 05097150
* SET POLL LOOP COUNT TO PROPER VALUE FOR BUSY PRINTER STATUS OPS. @38 05097180
SPACE 1                    @38 05097210
TBN TUBAT4(,XR2),TUBBSY    IS THIS A PRINTER TUB?      @38 05097240
JF CMNDA                   NO - JUMP                          @38 05097270
L TUBDTF(,XR2),DTF        XR2----> DTF                      @38 05097300
TBN $BDDEV(,DTF),BSCA     THIS BSCA DTF?                  @38 05097330
JF CMNDA                   NO - LEAVE COUNT ALONE         @38 05097360
SBN LCBAT3(,DTF),LCBBYP   INDICATE POLLING BUSY PRINTER@38 05097390
CLI $BDCNT(,DTF),ALLBIT   LOOP COUNT = CONTINUOUS?      @38 05097420
JNE CMNDA                  NO - LEAVE THIS COUNT          @38 05097450
MVC $BDCNT-1(1,DTF),$BDCNT(,DTF) SAVE LOOP COUNT @38 05097480
MVI $BDCNT(,DTF),X'02'    ELSE-- SET LOOP COUNT = 2     @38 05097510
L $BDWKA(,DTF),XR2       XR2-----> WORK AREA          @38 05097540
TBN $BWKMC(,XR2),X'01'    DA RUNNING?                    @38 @39 05097570
JF CMNDA                   NO - LEAVE $BDCNT = 2         @38 05097600
L WKDTFD(,XR2),DTF       XR2----> DTF                      @38 05097630
MVI $BDCNT(,DTF),X'FF'   ELSE - SET CONTINUOUS POLL    @38 05097660
CMNDA EQU *                @38 05097690
L PLTUBA(,PL),XR2        AND FIND TUB AGAIN            @38 05097720
.NBY01 ANOP                @38 05097750
CMRSCL EQU *                *                B 05098000
.C0305 ANOP                05099000
TBN TUBAT1(,XR2),TUBKNM   THIS A CONSOLE TUB ?          05099500
L TUBDTF(,XR2),DTF       SAVE DTF ADDRESS              05100000
ST CMSDTF,DTF            * FOR THIS REQUEST.          05110000
JT CMRNDQ                YES-DON'T DECREMENT REQ COUNT 05115000
SPACE 1                  05120000
SLC LCBNW#(,DTF),X$0001   DECREMENT NEW REQ COUNT FOR LINE 05130000
SPACE 1                  05140000
*-----*                05150000
* SET INTERNAL OP CODE AND ROUTE NEW REQUEST. *                05160000
*-----*                05170000
SPACE 1                  05180000
CMRNDQ EQU *                * LOCAL                05185000
SBF PL$OPM(,PL),ALLBIT    SET ALL OP MOD BITS OFF      05190000
MNN PL$OPM(,PL),PLOPC(,PL) MOVE LAST 4 BITS OF OP CODE 05200000
SPACE 1                  05210000
CLI PLOPC(,PL),OPACI     IS OP IS ACCEPT INPUT ?      05220000
TBF PLOPM(,PL),OPOLT     AND NOT OLT ?                05230000
BC CMACI,TRUAEQ          YES - GO HANDLE.          05240000
SPACE 1                  05250000
SBN CMSWIT,CMTPRQ        SET SWITCH TO INDICATE TP    05260000
*                          REQUEST IS NOW BEING HANDLED 05270000
AIF (&NOB).C0310          05280000
AIF (&ONE).C0300         05290000
TBN $BDDEV(,DTF),BSCA     BSCA LINE ?                2 05300000
BF CMMREQ                 NO-GO PROCESS MLTA REQUEST    2 05310000
.C0300 ANOP                05320000
B CMBREQ                  PROCESS THE BSCA REQUEST. ONLY B 05330000
AGO .C0320                05340000
.C0310 ANOP                05350000
B CMMREQ                  PROCESS THE MLTA REQUEST. ONLY M 05360000
.C0320 ANOP                05370000
EJECT                    05380000
*****                    05382000
* RESCHEDULE DTF THAT WAS IN ERP. *                05384000
*****                    05386000
SPACE 1                  05388000

```

CMNWRK	EQU	*	* LOCAL	05390000
	CLI	#CMERP-1,NOBIT	IS THERE A DTF THAT WAS IN ERP	05410000
*			* TO RESCHEDULE ?	05420000
	JE	CMWTST	NO- CHECK FOR TIMER RESCHED.	05430000
	L	#CMERP,DTF	XR2--> ERP DTF ADDRESS	05460000
	MVI	#CMERP-1,NOBIT	CLEAR ERP DTF ADDRESS	05490000
	ST	CMSDTF,DTF	SAVE DTF ADDRESS	05510000
	AIF	(&NOB).C0440		05520000
	AIF	(&ONE).C0600		05530000
	TBN	\$BDDEV(,DTF),BSCA	MLTA DTF	2 05540000
	BF	CMMTBY	YES - GO CHECK FOR MLTA BUSY	2 05550000
.C0600	ANOP			05560000
	B	CMBTBY	GO TEST LINE FOR BUSY	B 05570000
.C0440	AGO	.C0480		05580000
	ANOP			05590000
.C0480	B	CMMTBY	GO SEE IF MLTA LINE BUSY. ONLY M	05600000
	ANOP			05610000
	SPACE	1		05620000
	*****	*****	*****	05630000
*	NO WORK		*	05640000
	*****	*****	*****	05650000
	SPACE	1		05660000
CMWTST	EQU	*	* LOCAL	05670000
	SPACE	1		05690000
*	RESTART	POLLING IF NECESSARY		NB 05700000
	SPACE	1		05710000
	L	@LCB#1,DTF	XR2-->FIRST LCB	NB 05720000
CMWTS1	EQU	*	*	NB 05730000
	TBN	LCBATR(,DTF),LCBTIM	RESCHEDULE NEEDED?	05740000
	AIF	(&ONE).C0483		05743000
	JF	CMWOTR	NO-CONTINUE	05746000
.C0483	TBN	\$BDDEV(,XR2),BSCA	BSC DTF AND -	2NB 05750000
	AIF	(&NOB).C0486		05750700
	TBF	LCBAT2(,DTF),LCBACT	LINE ACTIVE ?	05751400
.C0486	JT	CMTMRS	YES-GO RESCHEDULE THIS LINE	05752100
	ANOP			05752800
	AIF	(&ONE).C0489		05753500
.C0489	TBF	\$MDDEV(,DTF),BSCA	MLTA DEVICE?	05754200
	ANOP			05754900
	AIF	(&NOM).C0492		05755600
.C0492	JT	CMTMRS	YES-GO RESCHEDULE	05756300
	ANOP			05757000
CMWOTR	EQU	*	*	05757700
	SPACE	1		05800000
&MIX	SETA	(&BSC+&MLA)		05810000
	AIF	(&MIX LE '1').Y0800		05820000
	CLI	LCBCHN-1(,XR2),NOBIT	ANY MORE DTF'S	YNB 05830000
	JE	CMWAIT	NO-GO WAIT	YNB 05840000
	SPACE	1		05850000
	L	LCBCHN(,XR2),XR2	XR2--->NEXT DTF	YNB 05860000
	B	CMWTS1	GO CHECK AGAIN	YNB 05870000
.Y0800	AGO	.Y0810		05880000
	ANOP			05890000
.Y0810	J	CMWAIT	GO WAIT	NB 05900000
	ANOP			05910000
	SPACE	1		05920000
CMTMRS	EQU	*	*	NB 05930000
	ST	CMSDTF,XR2	SAVE DTF ADDRESS FOR CM	05940000
	AIF	(&ONE).Y0820		05940200
	TBF	\$MDDEV(,XR2),BSCA	MLTA DTF?	2 05940400

	BT	CMMSCH	YES-GO RESCHEDULE MLTA	2	05940600
.Y0820	ANOP				05940800
	AIF	(&NINT).Y0830			05941000
	SBN	\$FLGC,#NTRAC	INDICATE NO TRACE	NB	05941200
*	PUT	TWO SECONDS IN TIMER IOB BEFORE RESCHEDULE	(RESCHEDULE WILL	NB	05941400
*		ACTUALLY START THE TIMER RUNNIG)		NB	05941600
	SPACE	1			05941800
	LA	TIMIOB,XR2	XR2-->TIMER IOB	NB	05942000
	MVC	TITIME(4,XR2),PLTIME	SET TIME INTERVAL AS SPEC'ED	NB	05942200
.Y0830	ANOP				05942400
	AIF	(&NOB).Y0840			05942600
	B	CMBSCH	GO RESCHEDULE BSCA LINE	2/B	05942800
	AGO	.Y0850			05943000
.Y0840	ANOP				05943200
	B	CMMSCH	GO RESCHEDULE MLTA	2/M	05943400
.Y0850	ANOP				05943600
CMWAIT	EQU	*	*		05943800
	AIF	(&NINT).Y0860			05944000
	MVI	TIMIOB+7,NOBIT	SET TO WAIT ON TIMER ALSO	NB	05944200
.Y0860	ANOP				05944400
	B	\$CC4CM			05944800
	MEND				06070000

MODULE-\$E072 , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
* NAME: $E072 * 00030000
***** 00040000
$E072 00050000
  GBLB &NOB, &MIN, &NCS, &N32, &NDME, &NSWL, &NAS, &NDF, &NRUF 00060000
  GBLB &NMSG, &N37, &NINT, &N41, &NPBY 00080000
  LCLA &MIX 00090000
  TEXT 00100000
* R-07, C-00 CHANGE LEVEL 00110000
  AIF (&NOB).C0300 00120000
  TITLE '$E072/CMBOPE---BSCA OPERATION END HANDLING' 00130000
***** 00140000
* * 00150000
* NAME--CMBOPE * 00160000
* * 00170000
* TITLE--BSCA OP END ANALYSIS * 00180000
* * 00190000
* FUNCTION--ANALYSIS EACH OP END FOR A TP LINE AND DETERMINE WHAT * 00200000
* IF ANYTHING MUST BE DONE TO COMPLETE THE CURRENT TP * 00210000
* OPERATION. ROUTE COMPLETED OPERATION BACK TO THE USER. * 00220000
* RESCHEDULE WORK ON THE LINE IF NO MORE OP ENDS TO BE * 00230000
* HANDLED. * 00240000
* * 00250000
* OPERATION-- * 00260000
* * 00270000
* . IF ABORT OPERATION, HANDLE THE ABORT UNTIL IT IS * 00280000
* COMPLETE. THEN RESCHEDULE THE LINE. * 00290000
* * 00300000
* . IF STOP INVITE REQUEST CALL $CC4BQ * 00310000
* * 00320000
* . FIND THE OP ENDED PARAMETER LIST AND SET ON THE POLL * 00330000
* SKIP BIT. * 00340000
* * 00350000
* . TRACE THE OP END AFTER CALLING CHECK * 00360000
* * 00370000
* . IF AN ERROR OCCURRED CALL $CC4BE. ON RETURN * 00380000
* EITHER POST THE RESULTS TO THE USER, RESCHEDULE THE * 00390000
* LINE, OR HANDLE THE DATA IN THE LINE BUFFER. * 00400000
* * 00410000
* . IF WRITE OP END, THEN * 00420000
* - IF BSCA RECORD WAS SHORTED THEN DEFINED AT * 00430000
* ASSIGNMENT TIME, CALL TRANSIENT TO FILL OUT THE * 00440000
* RECORD TO CORRECT LENGTH. * 00450000
* * 00460000
* - SET UP THE RETURN CODE FOR THE RESULTS OF THE * 00470000
* OPERATION. * 00480000
* * 00490000
* - IF BSCA IS NOT COMPLETE, CALL RESCHEDULE FUNCTION * 00500000
* TO COMPLETE THE OPERATION. * 00510000
* - EOT MUST BE SENT. * 00520000
* - OPERATION NOT COMPLETE BECAUSE CLEAR KEY IS * 00530000
* BEING HANDLED. * 00540000
* * 00550000
* - IF BSCA OPERATION IS COMPLETE, THEN * 00560000
* - FREE THE DFF HOLD BUFFER IF DFF PUT. * 00570000
* * 00580000
* - IF PUT-THEN-GET, THEN SET UP THE GET OPERATION IN * 00590000
* THE INTERNAL OP CODE, THEN RESCHEDULE THE LINE. * 00600000
```

*		* 00610000
*	- IF OPERATION IS COMPLETED SET UP TO POST THE	* 00620000
*	REQUESTOR OF THE RESULTS.	* 00630000
*		* 00640000
*	. IF READ OP END, THEN	* 00650000
*	- IF BSCA RECEIVE INITIAL WAS LAST OP, CHECK FOR ANY	* 00660000
*	CONFLICTING OPERATION IN THE LINE QUEUE, AND REJECT	* 00670000
*	THEM IF ANY FOUND.	* 00680000
*		* 00690000
*	- IF BSCA MESSAGE MODE INPUT, SET UP TO CONSOLIDATE	* 00700000
*	ALL BLOCKS INTO ONE MESSAGE. SET UP TO RESCHEDULE	* 00710000
*	LINE AND GET TO EOT BEFORE RETURNING TO THE USER.	* 00720000
*		* 00730000
*	- IF DATA MODE ESCAPE RECOGNIZED IN THE INPUT DATA	* 00740000
*	STREAM THEN SET UP POST OF THE COMMAND PROCESSOR.	* 00750000
*		* 00760000
*	- IF VALID INPUT DATA, MOVE, TRANSLATE, TRUNCATE THE	* 00770000
*	DATA AS REQUIRED.	* 00780000
*		* 00790000
*	- IF 3270 BSCA INPUT CHECK FOR CLEAR KEY HIT AND	* 00800000
*	HANDLE IT.	* 00810000
*		* 00820000
*	. IF COMPLETED DATA OPERATION, THEN	* 00830000
*	- REMOVE THE TP REQUEST FROM THE LINE QUEUE.	* 00840000
*	- FREE UP PUT-NO-WAIT HOLD BUFFERS.	* 00850000
*	- POST THE REQUESTOR OF THE TP OPERATION THAT IT IS	* 00860000
*	COMPLETE.	* 00870000
*	- QUEUE INVITE INPUT PARAMETER LIST ONTO THE TCB.	* 00880000
*	- CHECK FOR MORE OP ENDS AND HANDLE IF THERE ARE	* 00890000
*	SOME, OTHERWISE GO AND RESCHEDULE LINE.	* 00900000
*		* 00910000
*	ENTRY POINT--	* 00920000
*	CMBOPE	* 00930000
*		* 00940000
*	INPUT--	* 00950000
*	OUTPUT--	* 00960000
*	#OPEND - ADJUSTED FOR OP ENDS HANDLED.	* 00970000
*	CMSLCB - ADDRESS OF DTF HANDLED FOR LAST OP END.	* 00980000
*	CMSPL - ADDRESS TO TP PARAMETER LIST FOR LAST OP END.	* 00990000
*	TCBINQ - QUEUE OF INVITES THAT HAVE COMPLETED.	* 01000000
*		* 01010000
*	EXTERNAL REFERENCES--	* 01020000
*	\$CC4BQ - BSCA STOP II QUEUE ANALYSIS TRANSIENT.	* 01030000
*	\$CC4TT - CCP TRACE ROUTINE.	* 01040000
*	\$CC4BE - BSCA ERP TRANSIENT.	* 01050000
*	\$CC4BR - BSCA REJECT TRANSIENT.	* 01060000
*	\$CC4B5 - 3735 SENSE/STATUS TRANSIENT.	* 01070000
*	\$CC4BA - 3270 SENSE/STATUS TRANSIENT.	* 01080000
*	\$CC4JE - TRANSLATE ASCII TO EBCDIC	* 01090000
*	\$CC4B0 - 3270 INPUT FORMAT TRANSIENT.	* 01100000
*	\$CC4BB - BSCA RECORD BLANK TRANSIENT.	* 01110000
*	CMBSKP - BSCA SKIP BIT ON/OFF ROUTINE.	* 01120000
*	CMGINL - INPUT RECORD LENGTH ROUTINE.	* 01130000
*	CMFMRT - FREEMAIN ROUTINE.	* 01140000
*	CMGMRT - GETMAIN ROUTINE.	* 01150000
*	CMPSRQ - POST TP SCHEDULED ROUTINE.	* 01160000
*	CMPOST - POST OP END FOR READ OR PUT NO WAIT	* 01170000
*	CMPWGY - OP END FOR PUT PART OF PUT THEN GET.	* 01180000
*		* 01190000
*	EXIT, NORMAL--	* 01200000

```

*          TO CMBSCH TO RESCHEDULE THE LINE          * 01210000
*                                                    * 01220000
*****                                                    * 01230000
      SPACE 4                                          01240000
CMBOPE EQU * * * * * B 01250000
      TBN $FLGC,#NTRAC NO TRACE AND -                01251000
      CLI $BDCMP(,DTF),$BCNEG * 44 COMP CODE ?      01252000
      JC CMTRAC,TRUAEQ YES- CONTINUE                 01253000
      SBF $FLGC,#NTRAC NO-SET OFF NO TRACE IND      01254000
CMTRAC EQU * * * * * 01255000
      SPACE 1                                          01260000
* TRACE THE OP END COMPLETION CODE.                 B 01270000
      SPACE                                          01280000
      B CMTRCE CALL TRACE SUBROUTINE.                B 01290000
      DC AL1(CCP RIB) CCP RIB                         B 01300000
      DC AL1(TRRIB) TRACE SUBRIB                      B 01310000
      DC AL1(TTMOPN) TRACE OP END STATUS.            B 01320000
      SPACE 1                                          01330000
CMRSEN EQU * * * * * 01333000
      L LCBPL@(,DTF),PL XR1--> PARM LIST             01336000
      TBN PL$OPM(,PL),OPPUT PUT OPERATION?           01336500
      JT CMCABT YES - CHECK ABORT IN PROCESS         01337000
*                                                    --START-----@31 01337500
      SBF LCBOPC(,DTF),LCBRVI SET OFF SEND RVI INDICATOR. 01338000
*                                                    --END-----@31 01338500
CMCABT EQU * * * * * 01339000
      TBN LCBAT2(,DTF),LCBAPT LINE ABORT IN PROCESS ? B 01340000
      JF CMSTPX NO-GO CHECK FOR STOP RECEIVE . B 01350000
      SPACE 1                                          01360000
***** B 01370000
* ABORT IN PROCESS * B 01380000
***** B 01390000
      SPACE 1                                          01400000
      TBN LCBOPC(,DTF),OPPUT ABORT OF A PUT OPERATION ? B 01410000
      JF CMRABT NO - MUST BE READ ABORT              B 01420000
      SPACE 1                                          01430000
* ABORT OF A PUT B 01440000
      SPACE 1                                          01450000
      SLC LCBAT2(2,DTF),LCBAT2(,DTF) CLEAR LCBAT1 AND LCBAT2. B 01460000
      B CMBSCH GO RESCHEDULE THE LINE                B 01470000
      SPACE 1                                          01480000
* ABORT OF A READ B 01490000
      SPACE 1                                          01500000
CMRABT EQU * * * * * B 01510000
      CLI $BDCMP(,DTF),$BCDNE CCP PURGE COMPLETE ? B 01520000
      BE CMPBMP NO-GO BUMP TO NEXT BLOCK.            B 01530000
      L $BDWKA(,DTF),WKA POINT TO BSCA WORK AREA. B 01540000
      SBF BSFLGD(,WKA),FWDABT SET OFF FORWARD ABORT. B 01550000
      SLC LCBAT2(2,DTF),LCBAT2(,DTF) CLEAR LCBAT1 AND LCBAT2. B 01560000
      B CMBSCH GO TO RESCHEDULE THE LINE.            B 01570000
      SPACE 2                                          01580000
CMSTPX EQU * * * * * B 01590000
      TBN LCBAT1(,DTF),LCBCRI STOP INVITE REQUESTED ? B 01600000
      JF CMBCMP NO-GO HANDLE THE OP END.             B 01610000
      SPACE 1                                          01620000
***** B 01630000
* STOP INVITE * B 01640000
***** B 01650000
      SPACE 1                                          01660000
&MIX SETA &NSWL+&NCS 01670000

```

```

AIF (&MIX EQ '2').S0200 01680000
USING CMBSKP,XR1 01690000
LA CMBSKP,XR1 LOAD @ OF POLL SKIP BIT RTN.C/SB 01700000
MVI CMB#SB(,XR1),SBN1 SET OP: SET SKIP BIT ON. C/SB 01710000
.S0200 SVC 0 ##### TRANSIENT CALL ##### B 01720000
DC AL1(CCPRIB) CCP SVC RIB B 01730000
DC AL1(CC4BQ) * B 01740000
SPACE 01750000
* $CC4BQ RETURNS CONTROL TO: B 01760000
* NSI, IF LINE IS STOPPED SUCCESSFULLY. B 01770000
* NSI+4, IF STOP FAILED, BUT CONTINUE OP END HANDLING. B 01780000
* NSI+7, IF STOP FAILED, AND ABORT IS REQUIRED. 01790000
SPACE 01800000
B CMBSCH LINE STOPPED, RESCHEDULE WORK. B 01810000
J CMBCMP NOT STOPPED, HANDLE THE OP END. B 01820000
B CMBSTP NOT STOPPED, ABORT THE LINE. B 01830000
SPACE 3 01840000
***** B 01850000
* COMPLETION CODE ANALYSIS FOR BSCA * B 01860000
***** B 01870000
SPACE 01880000
CMBCMP EQU * * LOCAL B 01890000
ST CMSPL,PL SAVE THE PARM LIST REG. B 01900000
CLI $BDCMP(,DTF),$BCDNE SUCCESSFUL DATA ? B 01910000
JE CMBSCH YES-GO HANDLE IT. B 01920000
CLI $BDCMP(,DTF),$BCEOT EOT RECEIVED, AND B 01930000
TBF LCBAT2(,DTF),LCBRCI * NOT RECEIVE INITIAL ? B 01940000
JC CMBSCH,TRUAEQ YES-GO HANDLE THE EOT. B 01950000
JNE CMBERR NOT EOT-GO TO ERP TRANSIENT. B 01960000
TBF LCBAT1(,DTF),LCBCRI+LCBPRI EOT TO CANCEL REQUEST ? B 01970000
JF CMBSCH YES-EOT IS OK THEN. B 01980000
SPACE 1 01990000
* EOT TO RECEIVE INITIAL IS INVALID. B 02000000
SPACE 1 02010000
CMBERR EQU * * LOCAL B 02020000
SPACE 1 02030000
AIF (&NINT).CT010 02040000
* INSTITUTE INTERVAL TIMER POLLING IF APPLICABLE NB 02050000
SPACE 1 02060000
CLI $BDCMP(,DTF),$BCNEG 44 COMPLETION CODE NB 02070000
TBN $BDATT(,DTF),$BCINP AND POLLING ? NB 02080000
JC CMBSER,FLSNEQ NO-CALL TRANSIENT NB 02090000
SPACE 1 02100000
L PLTUBA(,PL),XR2 XR2--> TUB NB 02110000
CLI TUBPHY(,XR2),TUB375 3735'S ? NB 02120000
L TUBDTF(,DTF),XR2 XR2-->DTF NB 02130000
JE CMBSER YES-CALL TRANSIENT NB 02140000
SPACE 1 02150000
***** NB 02160000
* NO RESPONSE TO POLL OF 3270S * NB 02170000
***** NB 02180000
SPACE 02190000
* CLEAN UP THE LCB AND TUB NB 02200000
SPACE 1 02210000
SLC LCBAT2(2,DTF),LCBAT2(,DTF) SET OFF LCB ATTRIBUTES NB 02220000
L PLTUBA(,PL),XR2 XR2-->TUB NB 02230000
SBF TUBAT2(,XR2),TUBOWN RESET TUB OWNERSHIP NB 02240000
SBN $FLGC,#NTRAC IND NO TRACING TO BE DONE 02245000
SPACE 1 02250000
* SET UP THE INTERVAL TIMER FOR A WAIT NB 02260000

```


SPACE 1			02270000
LA	TIMIOB, XR2	XR2-->TIMER IOB	NB 02280000
MVI	TIFLAG(, XR2), ALLBIT	SET TO CANCEL REMAINING TIME	NB 02290000
SVC	0	*	NB 02300000
DC	AL1(TMRIB)	RETURN TIME REMAINING	NB 02310000
SPACE 1			02320000
TBN	TIMOPE, VALOPE	OP END SINCE LAST HERE?	NB 02330000
SBF	TIMOPE, VALOPE	RESET IT	NB 02340000
JT	ST1MIN	YES-GO PUT 1 MIN. IN THE IOB	NB 02350000
SPACE 1			02360000
TBN	TIMOPE, POL1MN	POLLING FROM A MINUTE AND-	NB 02370000
CLC	TITIME(2, XR2), X\$0000	TIME ZERO ?	NB 02380000
SBF	TIMOPE, POL1MN	SET OFF INDICATER	NB 02390000
JC	SETIND, TRUNEQ	NO-GO SET 'POLL FOR A MIN.'	NB 02400000
SPACE 1			02440000
MVC	TITIME(3, XR2), WATIME	MAKE IT TIME SPECIFIED	NB 02450000
USETIM EQU	*	*	NB 02460000
MVI	TIFLAG(, XR2), X'02'	INTERRUPT WHEN TIME EXPIRED	NB 02470000
*		-----START-----@24	02472000
SBF	TIMIOB+7, ALLBIT	SET TO POST ON TIMER OP END	NB 02474000
*		-----END-----@24	02476000
SVC	0	*	NB 02480000
DC	AL1(STMIB)	START TIMER RUNNING	NB 02490000
SPACE 1			02500000
* INDICATE THAT A WAIT ON THE INTERVAL TIMER HAS STARTED SO			NB 02510000
* THAT CM WILL RESCHEDULE THE LINE WHEN NEXT ENTERED			NB 02520000
SPACE 1			02530000
L	CMSDTF, XR2	XR2-->CURRENT DTF	NB 02540000
SBN	LCBATR(, DTF), LCBTIM	SET IND. IN LCB	NB 02550000
SPACE 1			02560000
B	\$CC4CM	GO ISSUE THE WAIT	NB 02580000
SPACE 1			02590000
ST1MIN EQU	*	* LOCAL	NB 02600000
MVC	TITIME(4, XR2), PLTIME	PUT POLL TIME IN TIME IOB	NB 02610000
SPACE 1			02620000
SETIND EQU	*	* LOCAL	NB 02630000
SBN	TIMOPE, POL1MN	SET IND. 'POLL FOR A MINUTE'	NB 02640000
SPACE 1			02650000
*			02650700
* BUSY PRINTER SUPPORT			02651400
*			02652100
AIF	(&NPBY).NBY03	BUSY PRINTER SUPPORT?	02652800
L	CMSDTF, DTF	RELOAD DTF REGISTER	02653500
TBN	LCBAT3(, DTF), LCBBYB	BUSY PRINTER IN QUEUE?	02654200
JF	CMNPR1	NO - DONT BOTHER TO CLEAN UP	02654900
SVC	0	#### TRANSIENT CALL ####	02655600
DC	AL1(CCPRIB)	CCP SVC RIB	02656300
DC	AL1(CC4BG)	BUSY PRINTER CLEAN UP XSIENT	02657000
CMNPR1 EQU	*		02657700
.NBY03 ANOP			02658400
B	CMBSCH	GO RESCHEDULE THE LINE	NB 02660000
SPACE 1			02670000
* HAVE BSCA ERROR (DETECTED BY MLMP), CALL BSCA ERROR TRANSIENT			B 02680000
SPACE 1			02690000
CMBSER EQU	*	*	B 02700000
SBF	\$FLGC, #NTRAC	INDICATE TRACE TO RESUME	02705000
SBN	TIMOPE, VALOPE	IND. OP END TO TIMER CODE	NB 02710000
.CT010 ANOP			02720000
*			02720500
* BUSY PRINTER SUPPORT			02721000

```

*
AIF (&NPBY).NBY04 BUSY PRINTER SUPPORT? 02721500
AIF (&NINT EQ '0').NBY04 SKIP IF INTERVAL POLLING 02722000
CLI $BDCMP(,DTF), $BCNEG * 44 COMP CODE? 02722500
TBN $BDATT(,DTF), $BCINP AND POLLING 02723000
JC CMBER1,FLSNEQ NO - CALL ERROR XSIENT 02723500
TBN LCBAT3(,DTF),LCBBYP BUSY PRINTER IN QUEUE? 02724000
JF CMBER1 NO - CALL ERROR XSIENT 02724500
SVC 0 ##### TRANSIENT CALL ##### 02725000
DC AL1(CCPRIB) CCP SVC RIB 02725500
DC AL1(CC4BG) BUSY PRINTER CLEAN UP XSIENT 02726000
B CMBSCH GO RESCHEDULE WORK ON THE LINE 02726500
CMBER1 EQU * 02727000
.NBY04 ANOP 02727500
SVC 0 ##### TRANSIENT CALL ##### B 02728000
DC AL1(CCPRIB) CCP SVC RIB B 02730000
DC AL1(CC4BE) ERROR DETERMINATION TRANSIENT. B 02740000
SPACE 1 02750000
* $CC4BE RETURNS CONTROL TO B 02760000
* NSI, POST COMPLETE AND RESCHEDULE LINE (HARD ERROR, NO DATA, B 02770000
* JUST LET USER KNOW AND FORGET IT). ALSO USED IF ERROR ON B 02780000
* SYSTEM READ AND NEED TO BYPASS NORMAL ERP. B 02790000
* NSI+4, TO ONLY RESCHEDULE THE LINE (TUB IN ERP, PL @ SAVED, B 02800000
* OPERATION MAY BE RETRIED SO DON'T FREE OR POST.) B 02810000
* NSI+8, TO ANALYZE OP END AS IF NO ERROR. ALSO FOR AUTOMATIC B 02820000
* BYPASS OF NORMAL ERP IF SYSTEM PUT THAT CANNOT BE PUT B 02830000
* INTO ERP OPERATOR WAIT. B 02840000
SPACE 1 02850000
B CMRETC SET UP TO RETURN TO USER. B 02860000
SPACE 02870000
B CMBSCH GO RESCHEDULE WORK ON THE LINE. B 02880000
EJECT 02890000
***** 02900000
* OP COMPLETED SUCCESSFULLY * B 02910000
* DETERMINE WHAT KIND OF OPERATION IT WAS AND HANDLE IT ACCORDINGLY* B 02920000
***** B 02930000
SPACE 1 02940000
CMBSCS EQU * * LOCAL * B 02950000
AIF (&NINT).CT020 02960000
SBN TIMOPE,VALOPE IND. OP END TO TIMER CODE 02970000
SBF $FLGC,#NTRAC INDICATE TRACE TO RESUME 02980000
.CT020 ANOP 02985000
TBF LCBOPC(,DTF),LCBERP IF NOT WAITING FOR EOT TO B 02990000
* * STATUS MESSAGE.(IF ON PL$OPM B 03000000
* * IS NOT ORIG OP,IT IS A INV.) B 03010000
TBN PL$OPM(,PL),OPPUT WAS IT WRITE B 03020000
BT CMWEND BRANCH IF WRITE B 03030000
$CC4CM TITLE '$E072/CMREND---READ-OP-END-HANDLER' B 03040000
***** B 03050000
* BSCA READ OP END HANDLER * B 03060000
***** B 03070000
SPACE 03080000
* CHECK FOR SEARCH EOT OPERATION. IF SEARCH FOR EOT, AND EOT IS NOT B 03090000
* FOUND, CONTINUE SEARCH. IF EOT WAS FOUND, THEN DETERMINE B 03100000
* IF PARM LIST MUST BE DEQ'D. IF EOT IS FOUND FOR GET-MSG, B 03110000
* GO TO GET-MSG EOT PROCESSING LOGIC. B 03120000
SPACE 03130000
TBN LCBAT1(,DTF),LCBEOT SEARCH FOR EOT ON LINE ? B 03140000
JF CMPSAV NO-GO CHECK COMPLETION CODE. B 03150000
CLI $BDCMP(,DTF), $BCEOT EOT FOUND ? B 03160000

```

```

&MIX JNE CMPBKL NO-GO UPDATE BLK LEN TO NXT BLKB 03180000
      SETA &N32+&N37+&N41 03183000
      AIF (&MIX EQ '3').T1000 03186000
      TBN PLOPM(,PL),OPSTOP STOP INVITE AND 0/5/7B 03190000
      TBN PL$OPC(,PL),OPLSNS * POLL FOR STATUS ? 0/5/7B 03200000
      JF CMCKDQ NO - GO CHECK FOR DEQUEUE 0/5/7B 03210000
      SBN LCBAT1(,DTF),LCBDEQ SET TO DEQUEUE 0/5/7B 03220000
      SBN LCBOPC(,DTF),LCBERP SET SEARCH FOR EOT TO 0/5/7B 03230000
* * TO INSURE DEQUEUE. 0/5/7B 03240000
      SBN CMSWIT,CMSPSI SET INTERNAL IND. ON 0/5/7B 03250000
      MVI PL$RTC(,PL),RCXSPI MOVE IN STOPPED RTN CODE. 0/5/7B 03260000
CMCKDQ EQU * * LOCAL 0/5/7B 03270000
.T1000 ANOP 03275000
      SPACE 1 03280000
*-----* B 03290000
* EOT FOUND DURING SEARCH FOR EOT * B 03300000
*-----* B 03310000
      SPACE 1 03320000
      TBN LCBAT1(,DTF),LCBDEQ DEQUE WHEN EOT FOUND ? B 03330000
      SLC LCBAT2(2,DTF),LCBAT2(,DTF) CLEAR LCBAT1 AND LCBAT2. B 03340000
      L PLTUBA(,PL),XR2 LOAD REG TO THE TUB. B 03350000
      SBF TUBAT2(,XR2),TUBOWN SET OFF TUB OWN INDICATOR. B 03360000
      BF CMREDO DON'T DEQ-GO RESCHEDULE LINE. B 03370000
      TBN PL$OPC(,PL),OPGET GET OP BEING HANDLED ? B 03380000
      JF CMRERP NO-GO CHECK FOR ERP. B 03390000
      SBF TUBAT2(,XR2),TUBIIS SET OFF INVITE SCHEDULED. B 03400000
      TBN TUBAT2(,XR2),TUBDTA+TUBCMD DME ? B 03410000
      JF CMRERP IF NOT DME, CAN REUSE LINE BUF. B 03420000
      L TUBDTF(,XR2),DTF DTF ADDRESS B 03430000
      MVI LCBIBA-1(,DTF),NOBIT IF DME - ASSIGN BUFF TO PL B 03440000
      L PLTUBA(,PL),XR2 TUB ADDRESS B 03450000
CMRERP EQU * * LOCAL B 03460000
      CLI TUBAT2(,XR2),TUBDTA+TUBCMD DATA MODE ESCAPE RECOGNIZED ? B 03470000
      L TUBDTF(,XR2),DTF POINT TO THE DTF. B 03480000
      TBF LCBOPC(,DTF),LCBERP SEARCHING FOR EOT TO STATUS MSGB 03490000
      BC CMBDEQ,FLSNLO ERP/DME-GO DEQ THE REQUEST. B 03500000
      J CMREXT GO TO EXIT FROM READ OP END. B 03510000
      SPACE 03520000
*-----* B 03530000
* EOT NOT FOUND DURING SEARCH FOR EOT * B 03540000
*-----* B 03550000
      SPACE 03560000
CMPBKL EQU * * LOCAL B 03570000
      AIF (&NMSG).E0300 03580000
      TBN SAVTA2,TASMSG * GET MSG MODE ? GB 03590000
      JF CMPBMP NO-GO TO BUMP BKX FOR NXT BLK GB 03600000
      SPACE 1 03610000
* IF GET-MSG THEN SET DATA TRUNCATED INDICATOR BEFORE CONTINUING GB 03620000
* TO SEARCH GB 03630000
      SBN PL$RTC(,PL),RCXDTR SET DATA TRUNCATED RET CODE GB 03640000
.E0300 ANOP 03650000
CMPBMP EQU * * B 03660000
      ALC $BDBKX(2,DTF),$BDBKL(,DTF) UPDATE BLK PTR TO GET NEXT BLKB 03670000
      B CMFORB GO FORM NEXT OP ON THE LINE. B 03680000
      SPACE 3 03690000
*-----* B 03700000
* NOT SEARCH FOR EOT * B 03710000
*-----* B 03720000
      SPACE 1 03730000
CMPSAV EQU * * LOCAL B 03740000

```

```

MVC  PL$RTC(1,PL), $BDCMP(,DTF) SAVE THE COMPLETION CODE.      B 03750000
SBF  PL$RTC(,PL),BIT1      SET OFF X'40' IN RETURN CODE.      B 03760000
SPACE 1                                                         03770000
*  HAVE A SUCCESSFUL READ OPERATION.                            B 03780000
*  CHECK FOR RECEIVE INITIAL AND DATA RETURNED. IF YES, CHECK FOR B 03790000
*  CONFLICTING REQUESTS IN THE BSCA LINE QUEUE.                B 03800000
*  IF NOT RECEIVE INITIAL AND DATA IS RETURNED, GO HANDLE THE DATA B 03810000
*  ONLY SITUATION.                                             B 03820000
*  IF EOT WAS RECEIVED FOR CANCEL RECEIVE INITIAL OPERATION, THEN B 03830000
*  CHECK WHETHER PRIORITY CANCEL OR STOP POLLING REQUEST.     B 03840000
*  OTHERWISE EOT WAS RECEIVED FOR DATA. CHECK FOR MESSAGE MODE B 03850000
*  PROCESSING. IF GET-MSG, CLEAN UP THE OPERATION.             B 03860000
SPACE 1                                                         03870000
TBF  PL$RTC(,PL), $BCEOT-BIT1  DATA RECEIVED FOR A          B 03880000
TBN  LCBAT2(,DTF),LCBRCI      RECEIVE INITIAL ON THE LINE ?  B 03890000
SBF  LCBAT2(,DTF),LCBRCI+LCBPUT SET OFF RCVI AND PUT IND'S.   B 03900000
JT   CMREJC                  YES-DATA HIT, CHECK CONFLICTS.  B 03910000
L    PLTUBA(,PL),TUB         LOAD TUB REG.                   B 03920000
CLI  $BDCMP(,DTF), $BCEOT    EOT RECEIVED ?                 B 03930000
BNE  CMB327                  NOT EOT, DATA FOR NON-RCVI.    B 03940000
SPACE 1                                                         03950000
*-----*                                                       B 03960000
*  EOT RECEIVED - NO DATA                                     * B 03970000
*-----*                                                       B 03980000
SPACE 1                                                         03990000
TBN  LCBAT1(,DTF),LCBPRI     PRIORITY CANCEL REQUESTED ?    B 04000000
SLC  LCBAT2(2,DTF),LCBAT2(,DTF) CLEAR LCBAT1 AND LCBAT2.    B 04010000
SBF  TUBAT2(,TUB),TUBOWN    SET OFF TUB OWN INDICATOR.      B 04020000
*  -----START-----@25 04025000
JF   CMNRX1                 NOT PRIORITY CANCEL - JUMP.      B 04030000
MVC  LCBLID(1,DTF), $BDIND(,DTF) SAVE ID OF LAST TERM POLLED. B 04032000
B    CMBSCH                 RESCHEDULE IF EOT TO PRI.        B 04034000
CMNRX1 EQU *                 * LOCAL.                        B 04036000
*  -----END-----@25 04038000
SBF  TUBAT2(,TUB),TUBIIS    SET OFF INPUT SCHEDULED.        B 04040000
SPACE 1                                                         04050000
***** B 04060000
*  COMMON EOT HANDLING CODE                                     * B 04070000
***** B 04080000
SPACE 1                                                         04090000
CMREXT EQU *                 * LOCAL                          B 04100000
L    LCBPL@(,DTF),PL        POINT TO THE PARM LIST          B 04110000
AIF  (&NRUF).F0100                                                04110800
SPACE 1                                                         04112400
*  IF RUF PGM REQUEST, HANDLE LIKE A USER GET-MSG OPERATION.  WB 04113200
SPACE 1                                                         04114000
L    PLTUBA(,PL),XR2        XR2-> TUB.                       WB 04114800
TBN  TUBSCS(,XR2),TUBRUF    RUF ON THE SCREEN ?             WB 04115600
L    TUBDTF(,XR2),DTF      XR2-> LCB(DTF).                   WB 04116400
JT   CMRGOM                YES-GO HANDLE AS USER GET-MSG.WB 04117200
.F0100 ANOP                                                         04118800
CLI  PLOPM(,PL),OP$SYS     USER REQUEST, AND                B 04120000
AIF  (&NMSG).E0350                                                04130000
TBN  SAVTA2,TASMSG         GET-MSG MODE ?                    GB 04140000
JC   CMRGOM,TRUALO        YES-GO HANDLE GET-MESSAGE.        GB 04150000
.E0350 ANOP                                                         04160000
BNL  CMBDEQ               IF SYSTEM INVITE, PASS RECA      B 04170000
*  * TO $CC4CP ALWAYS.                                         B 04180000
SPACE 1                                                         04190000
*  USER INPUT OP -- EOT AND NOT MESSAGE MODE - NO HOLD BUFFER B 04200000

```

SPACE 1				04210000
SLC	PLEFFL(2,PL),PLEFFL(,PL)	ZERO EFFL. SO II WON'T FREE	B	04220000
L	PLTUBA(,PL),XR2	POINT TO THE TUB.	B	04230000
MVI	PLRECA-1(,PL),NOBIT	MAKE SURE PLRECA 0,NO BUFFER	B	04240000
L	TUBDTF(,XR2),DTF	POINT TO THE DTF.	B	04250000
B	CMBDEQ	GO DEQUE THIS REQUEST.	B	04260000
AIF	(&NMSG).E0400			04270000
SPACE				04280000
* USER GET-MSG -- EOT RECEIVED.	RESTORE PARM LIST		GB	04290000
* RECORD ADDRESS AND IN LENGTH, AND THE DTF RECORD LENGTH.			GB	04300000
* THEN GO DIRECTLY TO TEST FOR TRANSLATION.			GB	04310000
SPACE				04320000
CMRGOM EQU *	* LOCAL		GB	04330000
MVC	PLRECA(4,PL),LCBMR@(,DTF)	RESTORE RECA AND INL.	GB	04340000
CLI	PL\$RTC(,PL),RCXCLR	CLEAR RETURN CODE GOING BACK ?	GB	04350000
JNE	CMGWOC	NO - RETURN DATA	GB	04360000
* GET MSG WITH CLEAR KEY RESPONSE.			GB	04370000
MVI	LCBIBA-1(,DTF),NOBIT	BUFFER NO LONGER IS FOR LINE.	GB	04380000
B	CMBDEQ	GO DEQUEUE	GB	04390000
CMGWOC EQU *	* LOCAL			04400000
SBF	PL\$RTC(,PL),\$BCEOT	SET OFF EOT R.C. FOR GET-MSG.	GB	04410000
MVC	\$BDREL(2,DTF),LCBMRL(,DTF)	RESTORE THE DTF RECORD LEN.	GB	04420000
B	CMRBMV	GO TO CHECK FOR XLATE.	GB	04430000
SPACE				04440000
.E0400 ANOP				04450000
EJECT				04460000
*****			B	04470000
* DATA RECEIVED FOR RECEIVE INITIAL			* B	04480000
*****			B	04490000
SPACE 1				04500000
* IF THE TASK THAT THIS REQUEST IS FOR HAS A WAIT TYPE OP OR A			B	04510000
* PUT NO WAIT BLOCK OR RECORD ON THE LINE QUEUE, THEN HE ISNT			B	04520000
* WAITING FOR THIS OP END. REJECT THAT OP ON THE LINE QUEUE SO			B	04530000
* HE CAN RECEIVE THIS DATA AND WE CAN FREE THE HOLD BUFFER.			B	04540000
SPACE 1				04550000
* REMOVE CONFLICTING ACCEPTED REQUESTS. ONLY INVITES AND/OR			B	04560000
* PUT-NO-WAITS-MESSAGE REQUESTS MAY STAY IN THE QUEUE. BECAUSE OF			B	04570000
* BSCA LINE HANDLING, AT OP END TIME NO MORE THAN ONE GET OR PUT			B	04580000
* FOR RECORD, BLOCK, MESSAGE, OR PUT-NO-WAIT FOR RECORD OR BLOCK			B	04590000
* COULD BE IN THE LINE QUEUE AND THEREFORE HAVE TO BE REMOVED.			B	04600000
SPACE				04610000
CMREJC EQU *	* LOCAL		B	04620000
AIF	(&MIN).N0100			04630000
L	LCBPL@(,DTF),PL	RELOAD THE PL REG.	RB	04640000
SPACE				04650000
* SAVE PARM LIST AND DTF DATA ADDR'S AND LENGTHS. THEY WILL BE			RB	04660000
*USED ONLY BY GET-MESSAGE PROCESSING.			RB	04670000
SPACE				04680000
AIF	(&NMSG).E0600			04690000
MVC	LCBMR@(4,DTF),PLRECA(,PL)	SAVE RECORD ADDR AND LENGTH.	RGB	04700000
SLC	LCBMRL(2,DTF),LCBMRL(,DTF)	ZERO OUT INCREMENT REC LGTH.	RGB	04710000
.E0600 ANOP				04720000
SBF	LCBAT1(,DTF),LCBPRI	SET OFF PRIORITY CANCEL IND.	RB	04730000
MVC	CMIIND,LCBOWN(2,DTF)	SAVE TCB @ OF OWNER.	RB	04740000
TBF	PL\$OPC(,PL),OPLSNS+OPRFSH	POLL FOR STATUS,OR REFRESH OR	RB	04750000
TBF	SAVTA2,TASMSG	* GET-MESSAGE MODE	RB	04760000
JF	CMRECX	YES-GO READY FOR RECORD CHECK.	RB	04770000
LA	LCBPLQ-1(,DTF),PL	LOAD REG. TO DTF PL QUEUE.	RB	04780000
CMREQR EQU *	* LOCAL		RB	04790000
CLC	PLCHN(2,PL),CMSPL	NEXT PL SAME AS OP ENDED ?	RB	04800000

L	PLCHN(,PL),PL	LOAD REG TO NEXT PARM LIST.	RB	04810000
JE	CMRZRO	YES-SKIP IT, CHECK FOR ANOTHER	RB	04820000
L	PLTUBA(,PL),XR2	LOAD WORK REG WITH TUB @.	RB	04830000
CLC	CMIIND,TUBTCB(2,XR2)	THIS TUB FOR SAME OWNER ?	RB	04840000
JNE	CMRZRO	NO-GO CHECK FOR ANOTHER.	RB	04850000
TBN	PLOPC(,PL),OPNOW	NO WAIT OPERATION ?	RB	04860000
JF	CMRCAN	NO-GO CANCEL ANY WAIT OP'S.	RB	04870000
TBN	PLOPC(,PL),OPINV	THIS AN INVITE INPUT ?	RB	04880000
JT	CMRZRO	YES-GO CHECK FOR ANOTHER.	RB	04890000
TBN	PLOPM(,PL),OP\$SYS	SYSTEM REQUEST ?	RB	04900000
JT	CMRZRO	YES-TREAT AS PUT-MSG/NOWAIT.	RB	04910000
TBN	PLOPC(,PL),OPMSG	PUT/MSG REQUEST ?	RB	04920000
JT	CMRZRO	YES - OK TO KEEP	RB	04930000
	SPACE			04940000
*	REJECT THIS OP SO TASK CAN ACCEPT THE DATA RECEIVED FOR HIM		RB	04950000
	SPACE 1			04960000
CMRCAN	EQU *	* LOCAL	RB	04970000
	SVC 0	#### TRANSIENT CALL #####	RB	04980000
	DC AL1(CCPRIB)	CCP SVC RIB	RB	04990000
	DC AL1(CC4BR)	* BSCA OP REJECT TRANSIENT	RB	05000000
	J CMRECX	GO CHECK THE RECORD AREA.	RB	05010000
	SPACE			05020000
CMRZRO	EQU *	* LOCAL	RB	05030000
	CLI PLCHN-1(,PL),NOBIT	ANOTHER PARM LIST IN CHAIN ?	RB	05040000
	BNE CMREQR	YES-GO CHECK FOR REQUEST REJ	RB	05050000
	L CMSDTF,DTF	RELOAD THE DTF REG.	RB	05060000
	L LCBPL@(,DTF),PL	RELOAD THE PL REG.	RB	05070000
CMRECX	EQU *	* LOCAL	RB	05080000
	AGO .C0145			05090000
.N0100	ANOP			05100000
	B CMREJT	CALL REJT LOGIC IN\$CC4B1. MIN B		05110000
.C0145	ANOP			05120000
	L PLTUBA(,PL),TUB	POINT TO THE TUB.	B	05130000
	EJECT			05140000
	*****		B	05150000
*	DATA RECEIVED FOR NON-RECEIVE INITIAL (ALSO EXECUTED FOR RECV INT*		B	05160000
	*****		B	05170000
	SPACE 1			05180000
CMB327	EQU *	* LOCAL		05190000
	AIF (&N32).T0010			05200000
	TBN TUBSCS(,TUB),TUBCLR	CLEAR KEY HIT ?	0B	05210000
	SBF TUBSCS(,TUB),TUBCLR	SET OFF CLEAR KEY HIT.	0B	05220000
	AIF (&NDME).D0300			05230000
	JT CMRCRC	YES-GO SET CLEAR RETURN CODE.	0DB	05240000
	AGO .T0010			05250000
.D0300	ANOP			05260000
	JF CMR327	NO-GO CHECK TERMINAL TYPE.	0B	05270000
.T0010	ANOP			05280000
	AIF (&NDME).D0500			05290000
	TBF TUBSCS(,TUB),TUBDME+TUBDMF	DME HANDLED BY INQUIRY XIENT	DB	05300000
	JT CMR327	NO-GO TO CHECK FOR 3270.	DB	05310000
	AIF (&N32).D0400			05320000
.D0400	ANOP			05330000
	TBN TUBSCS(,TUB),TUBDMF	DME FAILED ?	0DB	05330000
	SBF TUBSCS(,TUB),TUBDME+TUBDMF	RESET THE DME INDICATORS.	DB	05350000
	AIF (&N32).D0500			05360000
	JF CMREOT	NO-GO SET SEARCH FOR EOT.	0DB	05370000
*	DATA MODE ESCAPE FAILED.			05380000
.D0500	ANOP			05390000
	AIF (&N32).D0600			05400000

* CLEAR KEY.				05410000
CMRCRC EQU *		* LOCAL		0B 05420000
L LCBPL@ (,DTF),PL		POINT TO THE PARM LIST.		0B 05430000
MVI PL\$RTC (,PL),RCXCLR		SET CLEAR RETURN CODE.		0B 05440000
.D0600 ANOP				05450000
&MIX SETA &N32+&NDME				05460000
AIF (&MIX EQ '2').D0700				05470000
CMREOT EQU *		* LOCAL	0/DB	05480000
SBN LCBAT1 (,DTF),LCBEOT+LCBDEQ		SET SEARCH EOT AND THEN DEQ	0/DB	05490000
B CMFORB		GO FORM THE OPERATION.	0/DB	05500000
SPACE 1				05510000
CMR327 EQU *		* LOCAL	0/DB	05520000
.D0700 ANOP				05530000
&MIX SETA &N32+&N37				05540000
AIF (&MIX EQ '2').A0020				05550000
CLI TUBPHY (,TUB),TUB375		THIS A 3270 OR 3735 ?	0/5B	05560000
.A0020 ANOP				05560500
AIF (&N41).N4101				05561000
TBN TUBPHY (,TUB),TUB374		THIS A 3741	7B	05563000
TBF TUBPHY (,TUB),ALLBIT-TUB374		THIS A 3741	7B	05564000
.N4101 ANOP				05566000
&MIX SETA &N32+&N37+&N41				05567000
AIF (&MIX EQ '3').T0015				05568000
L \$BDWKB (,DTF),XR1		POINT TO LOGICAL DATA AREA	0/57B	05570000
.T0015 ANOP				05570200
AIF (&N41).N4102				05570500
JF CMS37E		NOT A 3741 TAKE THE JUMP	7B	05571500
CLC S374ST (2,XR1),CMSME		CHECK 3741 STATUS MESSAGE	7B	05572000
AIF (&NAS).T0020				05572500
TBN \$BDATT (,DTF),\$BCASK		TEST FOR A ASCII LINE	7AB	05573000
JF CMS374		NO,TAKE THE JUMP	7AB	05573500
CLC S374ST (2,XR1),CMSMA		CHECK 3741 STATUS MESSAGE	7AB	05574000
.T0020 ANOP				05574500
CMS374 EQU *		* LOCAL	7B	05575000
JNE CMRPLS		NO-CHECK POLL FOR STATUS	7B	05575500
SVC 0		TRANSIENT CALL	7B	05576000
DC AL1 (CCPRIB)		CCP RIB	7B	05576500
DC AL1 (CC4B7)		3741 STATUS TRANSIENT	7B	05577000
B CMFORB		GO REQ NEXT OP ON THE LINE	7B	05577500
CMS37E EQU *		* LOCAL	7B	05578000
.N4102 ANOP				05579000
AIF (&N37).T0050				05580000
JNE CMRN37		NOT 3735, GO CHECK 3270.	5B	05590000
SPACE 1				05600000
*****			5B	05610000
* 3735 - CHECK FOR STATUS MESSAGE			* 5B	05620000
*****			5B	05630000
SPACE 1				05640000
CLC S375NL (3,XR1),CMNULS		THIS NUL-S-NUL MESSAGE ?	5B	05650000
AIF (&NAS).T0040				05660000
TBN \$BDATT (,DTF),\$BCASK		THIS A ASCII LINE ?	5AB	05670000
JF CMR375		NO-GO TO 3735 JUMP COND.	5AB	05680000
CLC S375NL (3,XR1),CMNULA		THIS NUL-S-NUL (ASCII) MSG ?	5AB	05690000
CMR375 EQU *		* LOCAL	5AB	05700000
.T0040 ANOP				05710000
JNE CMRCMD		NO-GO CHECK COMMAND MODE.	5B	05720000
SVC 0		#### TRANSIENT CALL #####	5B	05730000
DC AL1 (CCPRIB)		CCP SVC RIB	5B	05740000
DC AL1 (CC4B5)		3735 STATUS TRANSIENT	5B	05750000
SPACE 1				05760000

```

      B      CMFORB      GO REQUEST NEXT OP ON THE LINE5B 05770000
      SPACE 1      05780000
CMRN37 EQU *      * LOCAL      5B 05790000
.T0050 ANOP      05800000
      AIF      (&N32).A0060      05810000
      JNL      CMRCMD      NOT 3270, GO CHECK COMMAND MOD0B 05820000
      SPACE 1      05830000
***** 0B 05840000
* 3270 - NOT CLEAR KEY * 0B 05850000
***** 0B 05860000
      SPACE 1      05870000
* IGNORE CHECKING FOR CLEAR KEY IF NOT THE FIRST TEXT BLOCK. 0B 05880000
* ALL NON-FIRST TEXT BLOCKS BEGIN WITH A 'SBA' = X'11'. 0B 05890000
      SPACE      05900000
*      -----START-----@19 05902000
      TBN      LCBAT2(,DTF),LCBSEC      SECOND BLOCK OF DATA ? 0B 05904000
      JT      CMRCMD      YES-DON'T CHECK FOR CLEAR KEY 0B 05906000
*      -----START-----@19 05908000
      AIF      (&NAS).A0030      05930000
      TBN      $BDATT(,DTF),$BCASK      THIS A ASCI LINE ? 0AB 05940000
      JF      CMRSTS      NO-GO CHECK EBCDIC STATUS MSG0AB 05950000
      CLC      SNSTAS(2,XR1),CMSTAS      THIS ASCI '%R' STATUS MSG ? 0AB 05960000
      J      CMRNEB      GO TO CONDITIONAL JUMP. 0AB 05970000
      SPACE 1      05980000
.A0030 ANOP      05990000
CMRSTS EQU *      * LOCAL      0B 06000000
      CLC      SNSTAS(2,XR1),CMSTUS      THIS A STATUS MESSAGE ? 0B 06010000
CMRNEB EQU *      * LOCAL      0B 06020000
      JNE      CMRAID      NO-GO CHECK AID. 0B 06030000
      SVC      0      ##### TRANSIENT CALL ##### 0B 06040000
      DC      AL1(CCPRIB)      CCP SVC RIB 0B 06050000
      DC      AL1(CC4BA)      THE 3270 SENSE STATUS XIENT. 0B 06060000
      SPACE      06070000
      B      CMFORB      GO FORM NEXT OP ON THE LINE. 0B 06080000
      SPACE      06090000
CMRAID EQU *      * LOCAL      0B 06100000
      AIF      (&NCS).T0052      06101000
      AIF      (&NSWL).T0051      06102000
      TBN      $BDATR(,DTF),$BCMCN      SWITCHED LINE ? 0B 06103000
      JF      CMSAID      YES - DON'T UPDATE TO CS AID 0B 06104000
.T0051 ANOP      06105000
      LA      2(,XR1),XR1      BUMP POINTER FOR CONTROL STA. 0B 06106000
CMSAID EQU *      * LOCAL      06107000
.T0052 ANOP      06108000
      AIF      (&NAS).A0050      06110000
      TBN      $BDATT(,DTF),$BCASK      THIS A ASCI LINE ? 0AB 06120000
      JF      CMRADE      NO-GO CHECK EBCDIC AID VALUE.0AB 06130000
      CLI      AID(,XR1),ASCCLR      THIS ASCI CLEAR INDICATION ? 0AB 06140000
      J      CMRPLS      GO TO CONDITIONAL JUMP. 0AB 06150000
      SPACE 1      06160000
CMRADE EQU *      * LOCAL      0AB 06170000
.A0050 ANOP      06180000
      CLI      AID(,XR1),AIDCLR      CLEAR KEY HIT ? 0B 06190000
.A0060 ANOP      06190900
&MIX SETA &N32+&N41      06191800
      AIF      (&MIX EQ '2').T0053      06192700
      SPACE 1      06193600
*-----START-----@05 06194500
***** 06195400
* WHEN ENTERING AT THIS LABEL EQUAL CONDITION MUST BE OFF 06196300

```



```

* UNLESS THE 3270 CLEAR KEY HAS BEEN DEPRESSED                                06197200
*****-----*****
CMRPLS EQU * LOCAL 0B 06200000
L CMSPL,PL GET PARM LIST @ TO. 0/7B 06210000
TBN PL$OPC(,PL),OPLSNS POLLING FOR STATUS ? 0/7B 06220000
SBF PL$OPC(,PL),OPLSNS SET OFF POLL FOR STATUS IND.0/7B 06230000
JF CMRCLR NO-GO SET CLEAR INDICATOR. 0/7B 06240000
L PLTUBA(,PL),XR2 POINT TO THE TUB. 0/7B 06250000
SBF TUBSCS(,XR2),TUBSSP SET OFF STATUS POLL IND. 0/7B 06260000
AIF (&N41).A0051 06260300
SPACE 2 06260600
* THE 3741 HAS INDICATED THAT A STATUS MESSAGE WILL BE SENT 06261200
* BUT HAS SENT DATA INSTEAD - THIS INDICATES THAT THE 3741 HAS 06261500
* MALFUNCTIONED AND WILL NOW ATTEMPT TO SEND THE ENTIRE DISKETTE. 06261800
* CCP MUST SET UP A FORWARD ABORT TO PREVENT THIS. FIRST MUST 06262100
* CANCEL USER PROGRAM AND FREE NECESSARY AREAS(PL,BUFF). 06262400
SPACE 1 06262700
TBN TUBPHY(,XR2),TUB374 IS THIS - 06263000
TBF TUBPHY(,XR2),ALLBIT-TUB374 A 3741 ? 06263300
JF CMNABT NO - JUMP 06263600
* CANCEL USER PROGRAM 06263900
L TUBTCB(,XR2),XR2 XR2--> USER TO BE CANCELED 06264200
B CC4TI GO POST TERMINATION 06264500
DC XL1'40' COMPLETION CODE 06264800
* FREE PL AND BUFFER 06265100
B $CC4FR GO FREE 06265400
* SET UP FORWARD ABORT 06265700
L CMSDTF,DTF XR2--> DTF 06266000
SBN LCBAT2(,DTF),LCBABT INDICATE FORWARD ABORT 06266300
B CMBSTP GO DO IT 06266600
CMNABT EQU * * 06266900
SPACE 1 06267500
.A0051 ANOP 06267800
L TUBDTF(,XR2),DTF RELOAD THE DTF REG. 0/7B 06270000
MNN PL$OPM(,PL),PL$OPC(,PL) RESTORE ORIGINAL OP CODE. 0/7B 06280000
CMRCLR EQU * * LOCAL 0/7B 06290000
JNE CMRCMD NOT CLEAR KEY-CHK COMD MODE 0/7B 06300000
.T0053 ANOP 06304000
AIF (&N32).T0100 06306000
SPACE 06310000
* INDICATE CLEAR KEY HIT IF NOT SYSTEM USER. 0B 06320000
SPACE 06330000
* ZERO OUT PLRECA IN THE PL(AT THIS TIME THE GETMAINED ADDRESS 06332000
* IS IN IBA AND PLRECA) 06334000
SPACE 1 06336000
MVI PLRECA-1(,PL),NOBIT ZERO OUT PLRECA 0B 06338000
TBN PLOPM(,PL),OP$SYS SYSTEM USER ? 0B 06340000
SBN LCBAT1(,DTF),LCBEOT SET SEARCH EOT, DON'T DEQUE. 0B 06350000
SBN PL$OPC(,PL),OPRFSH SET REFRESH OPERATION NEEDED. 0B 06360000
MNN PL$OPC(,PL),PL$OPM(,PL) SAVE THE CURRENT OP CODE. 0B 06370000
* --START-----@37 06370500
L PLTUBA(,PL),TUB XR1 --> TUB 0B 06372000
AIF (&NRUF).T0055 06372500
SBF TUBSCS(,TUB),TUBRUF RESET PRUF INDICATOR 0B 06373000
.T0055 ANOP 06373500
JT CMGOFB GO TO FORM OP FOR MLMP 0B 06374500
SBN TUBSCS(,TUB),TUBCLR INDICATE CLEAR KEY DEPRESSED 0B 06375000
AIF (&NDME EQ '0').T0070 SKIP IF DME SUPPORTED. 06376000
B CMRCRC GO GIVE 07 RETURN CODE 0B 06376500

```

```

.T0070 ANOP                                06377000
*
*          ---END-----@37 06377200
CMGOFB EQU *                               06377500
      B   CMFORB          GO FORM NEXT OP FOR MLMP 0B 06378000
      SPACE                                06420000
.T0100 ANOP                                06430000
      EJECT                                06440000
***** B 06450000
* ANY TERMINAL AND NOT A STATUS MESSAGE * B 06460000
***** B 06470000
      SPACE                                06480000
* ZERO LCB BUFFER ADDRESS BECAUSE BUFFER NOW ASSIGNED TO OP ENDED PL. B 06491000
      SPACE                                06492000
CMRCMD EQU *                               B 06493000
      MVI LCBIBA-1(,DTF),NOBIT          * LOCAL          ZERO HIGH ORDER BYTE OF BUF AD B 06494000
      SPACE 1                             06495000
      L   LCBPL@(,DTF),PL              POINT TO THE PARM LIST.          B 06500000
      TBN PLOPM(,PL),OP$SYS            SYSTEM REQUEST ?          B 06510000
      JF  CMRDAT                        NO-GO HANDLE DATA MODE.        B 06520000
      AIF (&NRUF).F0200                06520800
      SPACE                                06522400
* IF RUF PGM REQUEST, HANDLE LIKE A USER GET-MSG OPERATION. WB 06523200
      SPACE                                06524000
      L   PLTUBA(,PL),XR2              XR2-> TUB.                  WB 06524800
      TBN TUBSCS(,XR2),TUBRUF          RUF ON THE SCREEN ?        WB 06525600
      L   TUBDTF(,XR2),DTF            XR2-> LCB(DTF).           WB 06526400
      JT  CMRMSG                        YES-GO HANDLE AS GET-MSG.        WB 06527200
.F0200 ANOP                                06528800
      SBN LCBAT1(,DTF),LCBEOT+LCBDEQ SET SEARCH EOT AND DEQ THEN B 06530000
CMRDAT EQU *                               B 06540000
      AIF (&NMSG).E0100                06600000
      TBF PLOPM(,PL),OP$SYS            USER REQUEST, AND          GB 06610000
      TBN SAVTA2,TASMSG                * GET-MSG MODE ?          GB 06620000
      JF  CMRBMV                        NO-GO CHECK FOR XLATE NOW.      GB 06630000
      SPACE 1                             06640000
***** GB 06650000
* GET-MSG -- BLOCK DATA, DO NOT TRANSLATE UNTIL EOT RECEIVED * GB 06660000
***** GB 06670000
      SPACE 1                             06680000
CMRMSG EQU *                               GB 06685000
      CLC $BDREL(2,DTF),PLINL(,PL) THIS HUNK FILL UP REST OF AREAGB 06690000
      JNE CMRBFL                        NO-GO TO UPDATE CONTROL PTRS. GB 06700000
      SBN LCBAT1(,DTF),LCBEOT+LCBDEQ SET SEARCH EOT/ DEQ ON FIND. GB 06710000
CMRBFL EQU *                               GB 06720000
      ALC PLRECA(2,PL),$BDREL(,DTF) UP REC ADDR TO NEXT FREE AREA.GB 06730000
      SLC PLINL(2,PL),$BDREL(,DTF) DECREMENT COUNT OF FREE SPACE.GB 06740000
      ALC LCBMRL(2,DTF),$BDREL(,DTF) KEEP COUNT OF TOTAL DATA IN. GB 06750000
      B   CMFORB          GO TO GET THE NEXT BLOCK.          GB 06760000
      SPACE                                06770000
.E0100 ANOP                                06780000
* NON USER GET-MSG OR WHEN EOT RECEIVED FOR GET-MSG (CMRGOM) B 06790000
CMRBMV EQU *                               B 06800000
      SPACE 1                             06810000
***** B 06820000
* TRANSLATE DATA * B 06830000
***** B 06840000
      SPACE 1                             06850000
      MVC #CMTRL+TLFRML,$BDREL(2,DTF) SET TRANSLATE PARM LIST B 06860000
      MVC #CMTRL+TLTOL,PLINL(2,PL) * B 06870000
      MVC #CMTRL+TLFRMA,PLRECA(2,PL) * B 06880000

```

MVC	#CMTRL+TLTOA,PLRECA(2,PL) *		B	06890000
TBF	PLOPM(,PL),OP\$SYS	USER REQUEST, AND	B	06900000
TBN	SAVTA1,TASTRN	DON'T TRANSLATE ?	B	06910000
JT	CMSETL	YES-SET LENGTH	B	06920000
AIF	(&NAS).A0100			06930000
TBN	\$BDATT(,DTF),\$BCASK	ASCII ADAPTER ?	AB	06940000
JF	CMRLCX	NO-GO CHECK FOR LOWER CASE.	AB	06950000
LA	#CMTRL,XR1	POINT TO TRANSLATE LIST.	AB	06960000
SVC	0	#### TRANSIENT CALL #####	B	06970000
DC	AL1(CCPRIB)	CCP SVC RIB	B	06980000
DC	AL1(CC4JE)	* TRANSLATE ASCII TO EBCDIC.	AB	06990000
*		* HARDWARE VALIDATES ALL	AB	07000000
*		* ASCII CHARS, SO NO	AB	07010000
*		* TRANSLATION ERRORS.	AB	07020000
	SPACE			07030000
L	LCBPL@(,DTF),PL	RELOAD THE PARM LIST REG.	AB	07040000
CMRLCX EQU	*	* LOCAL	AB	07050000
.A0100 ANOP				07060000
L	PLTUBA(,PL),XR1	XR1-> TUB.	B	07062000
TBF	TUBSCS(,XR1),TUBRUF	PRUF DATA IS INPUT ?	B	07064000
L	LCBPL@(,DTF),PL	RELOAD THE PARM LIST REG.	B	07066000
TBF	SAVTA1,TASCAS	FORCE UPPER CASE CHARS ?	B	07070000
JF	CMSETL	NO-SET LENGTH	B	07080000
	SPACE			07090000
* FIND AND	USE SHORTEST OF PLINL OR \$BDREL FOR FORCE UPPER CASE.		B	07100000
	SPACE			07110000
MVC	LCBWRK(2,DTF),PLINL(,PL)	SAVE INL IN WORK AREA.	B	07120000
CLC	\$BDREL(2,DTF),PLINL(,PL)	FROM AREA GT THAN TO AREA ?	B	07130000
JH	CMRUCX	YES-THEN USE PLINL AS IS SET.	B	07140000
MVC	LCBWRK(2,DTF),\$BDREL(,DTF)	ELSE USE DTF RECORD LENGTH.	B	07150000
	SPACE 1			07160000
*****			B	07170000
* UPPER CASE TRANSLATE			*	B 07180000
*****			B	07190000
	SPACE 1			07200000
CMRUCX EQU	*	* LOCAL	B	07210000
L	PLRECA(,PL),XR1	LOAD ADDR OF START OF BUFFER.	B	07220000
CMRUPX EQU	*	* LOCAL	B	07230000
CLI	0(,XR1),BLANK	THIS CHAR BLANK OR GREATER ?	B	07240000
JL	CMRUPC	NO-GO UP TO NEXT CHARACTER.	B	07250000
SBN	0(,XR1),BLANK	SET ON UPPER CASE ZONE BIT.	B	07260000
CMRUPC EQU	*	* LOCAL	B	07270000
LA	1(,XR1),XR1	INCREMENT REG TO NEXT CHAR.	B	07280000
SLC	LCBWRK(2,DTF),X\$0001	DECREMENT COUNT, ANY LEFT ?	B	07290000
BNZ	CMRUPX	YES-GO CHECK NEXT CHAR.	B	07300000
L	LCBPL@(,DTF),PL	RELOAD THE PARM LIST REG.	B	07310000
	SPACE 1			07320000
*****			B	07330000
* SET USER RECORD LENGTH			*	B 07340000
*****			B	07350000
	SPACE 1			07360000
CMSETL EQU	*	* LOCAL	B	07370000
TBF	PLOPM(,PL),OP\$SYS	USER REQUEST, AND	B	07380000
TBN	LCBAT2(,DTF),LCBTRC	* TRUNCATED BLOCK INDICATED ?	B	07390000
*		----START-----@22		07392000
JF	CMSBFL	NO-GO SET BSCA EFFL COUNT.	B	07394000
TBF	SAVTA2,TASREC+TASBLK	MESSAGE MODE TERMINAL ?	B	07396000
JF	CMNMSG	YES-DON'T SET OFF TRUNCATED.	B	07398000
SBF	LCBAT2(,DTF),LCBTRC	SET OFF TRUNCATED IND.	B	07400000
CMNMSG EQU	*	* LOCAL	B	07410000

```

* -----END-----@22 07415000
TBN SAVTA2,TASBLK BLOCK READ OPERATION ? B 07420000
JT CMSBLK YES-GO BUMP BKX PAST THIS BLK. B 07430000
CLC PLINL(2,PL),SAVRCL INL LESS THAN TAS RECL ? B 07440000
JNL CMSBFL NO-MORE REC'S IN THE BLOCK. B 07450000
SLC $BDBKX(2,DTF),PLINL(,PL) ADJUST BKX TO GET THE NEXT REC. B 07460000
CMSBLK EQU * * B 07470000
ALC $BDBKX(2,DTF),SAVRCL BUMP TO NEXT REC/BLK. B 07480000
SBN PL$RTC(,PL),RCXDTR SET TRUNCATED IND TO USEER. B 07490000
CMSBFL EQU * * B 07500000
MVC PLEFFL(,PL),$BDREL(2,DTF) MOVE IN RETURNED RECORD LEN. B 07510000
AIF (&N32).T0210 07520000
TBN PLOPM(,PL),OP$SYS SYSTEM REQUEST, AND 0B 07530000
CLI CMSPHY,TUB5M2 A 3270 TERMINAL ? 0B 07540000
JC CMRETC,FLSOHI NO-GO SET RETURN CODE. 0B 07550000
L PLTUBA(,PL),XR2 XR2 -> TUB B 07550100
TBN TUBSCS(,XR2),TUBRUF RUF DAT ON THE SCREEN ? B 07550150
L TUBDTF(,XR2),DTF XR2 -> DTF(LCB) B 07550200
JF CMCFRM NO-CALL FORMAT TRANSIENT B 07550250
MVC CMRFCK(2),#RUFCL SAVE MAX COMMAND LEN FOR PRUF B 07550300
SLC CMRFCK(2),CMSEVN SUBTRACT SEVEN FOR RIGHT SHIFT B 07550350
CLC PLEFFL(2,PL),CMRFCK INPUT DAT > MAXIMUM LENGTH ? B 07550400
JNH CMCFRM NO-CALL FORMATTING TRANSIENT B 07550450
MVC PLEFFL(2,PL),CMRFCK SET DATA LEN TO MAX DATA B 07550500
SBN PL$RTC(,PL),RCXDTR SET TRUNCATED DATA TO USER B 07550550
CMCFRM EQU * * LOCAL B 07550600
AIF (&NSWL).T0208 07550800
AIF (&NCS).T0206 07551600
TBF $BDATR(,DTF),$BCMPT NOT MULTI-POINT LINE ? B 07552400
JF CMB0C1 NO - CALL CONTROL LINE FORMAT B 07553200
.T0206 ANOP 07554000
SVC 0 ##### TRANSIENT CALL ##### B 07554800
DC AL1(CCPRIB) CCP SVC RIB B 07555600
DC AL1(CC4S0) * 3275 SWITCHED LINE FORMATING B 07556400
J CMRETC GO SET RETURN CODE B 07557200
.T0208 ANOP 07558000
CMB0C1 EQU * * LOCAL 07558800
SVC 0 ##### TRANSIENT CALL ##### B 07560000
DC AL1(CCPRIB) CCP SVC RIB B 07570000
DC AL1(CC4B0) * 3270 COMMAND INPUT FORMATING 0B 07580000
* * ALWAYS RETURNS HERE 0B 07590000
.T0210 ANOP 07600000
J CMRETC GO SET RETURN CODE. B 07610000
SPACE 07620000
$CC4CM TITLE '$E072/CMWEND---WRITE-OP-END' 07630000
***** B 07640000
* BSCA WRITE OP END HANDLING * B 07650000
***** B 07660000
SPACE 07670000
* COME HERE ON SUCCESSFUL OP END OF A WRITE OPERATION B 07680000
* IF NOT PUT THEN GET - DEQUEUE THE PARAMETER LIST. B 07690000
* IF PUT THEN GET, REFORMAT PARM LIST AND GO BACK TO RESCHED LINE. B 07700000
SPACE 07710000
CMWEND EQU * WRITE OP END HANDLER 07720000
TBF PLOPM(,PL),OP$SYS USER REQUEST, AND B 07730000
TBN LCBAT2(,DTF),LCBTRC * BLOCK TRUNCATED IND ON ? B 07740000
SBF LCBAT2(,DTF),LCBTRC SET OFF THE TRUNCATED IND. B 07750000
JF CMWRVI NO-GO CHECK FOR RVI RECEIVED. B 07760000
CLC PLOUTL(2,PL),SAVRCL OUTL LESS THAN TAS RECL ? B 07770000
JNL CMWTRC NO-THEN OUTPUT WAS TRUNCATED. B 07780000

```

```

SPACE 1 07790000
*****
* USER REQUEST - PAD WITH BLANKS * B 07800000
***** B 07810000
***** B 07820000
SPACE 1 07830000
SVC 0 ##### TRANSIENT CALL ##### B 07840000
DC AL1(CCPRIB) CCP SVC RIB B 07850000
DC AL1(CC4BB) * PAD RECORD AREA WITH BLANKS B 07860000
DC AL2($$BSMS) ADCON FOR MLMP IOCS. B 07870000
DC AL2($$BMCH) ADCON FOR CHECK. B 07880000
SPACE 1 07890000
* BB RETURNS TO NSI+4 IF AN ERROR OCCURS. B 07900000
* BB RETURNS TO NSI+11 IF A NORMAL OPERATION OCCURRED (CMWRVI). B 07910000
SPACE 1 07920000
B CMBOPE GO HANDLE ERROR CASE. B 07930000
SPACE 1 07940000
CMWTRC EQU * * LOCAL B 07950000
* SKIPPED BY BB RETURN. B 07960000
SBN PL$RTC(,PL),RCXDTR SET TRUNCATED RETURN CODE. B 07970000
SPACE 1 07980000
CMWRVI EQU * * LOCAL B 07990000
TBN LCBOPC(,DTF),LCBRVI RVI RECEIVED ? B 08000000
* -----START-----@3 08002000
SBF LCBOPC(,DTF),LCBRVI SET OFF RVI RECEIVED. B 08004000
* -----END-----@3 08006000
JF CMWEOT NO-GO CHECK FOR SEND EOT NEED B 08010000
MVI PL$RTC(,PL),RCXRVI SET RVI RETURN CODE. B 08020000
CMWEOT EQU * * LOCAL B 08030000
TBN LCBAT2(,DTF),LCBSET SEND EOT OPERATION ? B 08040000
JF CMWETX NO-CHECK FOR SEND EOT. B 08050000
SPACE 1 08060000
***** B 08070000
* SENT EOT - OPERATION REQUIRED IT. * B 08080000
***** B 08090000
SPACE 1 08100000
SLC LCBAT2(2,DTF),LCBAT2(,DTF) CLEAR LCBAT1 AND LCBAT2. B 08110000
L PLTUBA(,PL),XR2 LOAD REG TO THE TUB. B 08120000
SBF TUBAT2(,XR2),TUBOWN SET OFF TUB OWN INDICATOR. B 08130000
AIF (&NRUF).F0300 08130500
SPACE 08131500
* SET OFF RUF INDICATION FOR A SUCCESSFUL USER PUT OPEND. WB 08132000
SPACE 08132500
TBF PLOPM(,PL),OP$SYS USER OPERATION ? WB 08133000
JF CMWRFH NO-SKIP RUF SETS. WB 08133500
SBF TUBSCS(,XR2),TUBRUF SET-OFF RUF INDICATOR. WB 08134000
SPACE 08134500
* SET ON RUF INDICATOR FOR A USER PUT/RUF-MSG OPERATION. WB 08135000
SPACE 08135500
TBN PLOPC(,PL),OPRUF RUF OPERATION WB 08136000
TBF PLOPC(,PL),OPORDR-OPRUF * CODE ? WB 08136500
JF CMWRFH NO-DON'T SET-ON RUF IND. WB 08137000
SBN TUBSCS(,XR2),TUBRUF SET-ON RUF INDICATOR. WB 08137500
CMWRFH EQU * * WB 08138000
.F0300 ANOP 08139000
L TUBDTF(,XR2),DTF RELOAD THE DTF REG. B 08140000
AIF (&N32).T0300 08150000
SPACE 08160000
* WAS THIS A EOT FOR SCREEN REFRESH OPTION, THEN GO TO RESCHEDULE. 0B 08170000
SPACE 08180000
TBN PL$OPC(,PL),OPRFSH REFRESH OPERATION ? 0B 08190000

```

SBF	PL\$OPC(,PL),OPRFSH	SET OFF REFRESH INDICATOR.	0B	08200000
JF	CMWPGX	NO-GO TEST FOR PUT THEN GET	0B	08210000
MNN	PL\$OPM(,PL),PL\$OPC(,PL)	RESTORE ORIGINAL OP CODE.	0B	08220000
B	CMREDO	GO TO REDO WORK CHECK ON LINE.	0B	08230000
SPACE				08240000
CMWPGX	EQU *	* LOCAL	B	08250000
.T0300	ANOP			08260000
SPACE				08270000
* DETERMINE IF OP WAS PUT THEN GET			B	08280000
SPACE				08290000
TBN	PLOPC(,PL),OPGET	WAS IT PUT THEN GET	B	08300000
JF	CMRETC	JUMP IF NOT	B	08310000
SPACE				08320000
B	CMWPGY	HANDLE PUT THE GET	B	08330000
SPACE				08340000
J	CMREDO	JUMP TO RESTART THE LINE	B	08350000
SPACE 1				08360000
*****			B	08370000
* NOT A SEND EOT OPERAION - DETERMINE IF EOT NEEDED ANYWAY			* B	08380000
*****			B	08390000
SPACE 1				08400000
CMWETX	EQU *	* LOCAL	B	08410000
TBN	PL\$OPC(,PL),OPUSER	SYSTEM FUNCTION ?	B	08420000
JT	CMWSET	YES-GO SET SEND EOT.	B	08430000
TBN	PLOPC(,PL),OPMSG	IS THIS A PUT-MSG	B	08440000
TBF	PLOPC(,PL),OPORDR-OPRUF	* OPERATION ?	B	08450000
JT	CMWSET	YES-GO SET SEND EOT.	B	08460000
TBN	PLOPC(,PL),OPGET	GET FOLLOW PUT OPERATION ?	B	08470000
JF	CMRETC	NO	B	08480000
CMWSET	EQU *	*	B	08490000
SBN	LCBAT2(,DTF),LCBSET	SET SEND EOT.	B	08500000
B	CMFORB	GO TO FORM NEXT OP ON LINE.	B	08510000
TITLE	'\$E072/CMRETC---READ/WRITE OP END HANDLING'			08520000
*****			B	08530000
* FINAL BSCA READ/WRITE OP END HANDLING (COULD BE IN ERP)			* B	08540000
*****			B	08550000
SPACE 1				08560000
CMRETC	EQU *	*	B	08570000
TBN	LCBAT1(,DTF),LCBEOT	SEARCH EOT ?	B	08580000
BT	CMFORB	YES-GO FORM NEXT OP FOR LINE.	B	08590000
SPACE 1				08600000
TBN	PLOPC(,PL),OPGET	GET OPERATION ?	B	08610000
JF	CMRACT	NO-GO TEST LINE ACTIVE.	B	08620000
L	PLTUBA(,PL),XR2	POINT TO THE TUB.	B	08630000
SBF	TUBAT2(,XR2),TUBIIS	SET OFF INVITE SCHEDULED.	B	08640000
L	TUBDTF(,XR2),DTF	POINT TO THE DTF.	B	08650000
CMRACT	EQU *	* LOCAL	B	08660000
TBN	LCBAT2(,DTF),LCBACT	LINE ACTIVE ?	B	08670000
JF	CMBDEQ	NO-GO AND JUST DEQUE.	B	08680000
SBN	LCBAT1(,DTF),LCBNTQ	SET ID ACTIVE PARM LST REMOVED	B	08690000
SPACE				08700000
*****			B	08710000
* DEQUEUE		THE PARAMETER LIST	* B	08720000
*****			B	08730000
SPACE				08740000
CMBDEQ	EQU *	*	B	08750000
B	CMDEQ	DEQUEUE PL FROM LINE QUEUE	B	08760000
SPACE				08770000
AIF	(&NDF).F0400			08780000
L	PLTUBA(,PL),XR2	TUB ADDRESS	FB	08790000

```

TBN TUBAT2(,XR2),TUBDTA+TUBCMD COMMAND INTERRUPT MODE?(DME) FB 08800000
JT CMGOII YES - DONT GO TO DFF TASK. FB 08810000
TBN TUBTA1(,XR2),TASDFF IF A NON-DFF REQUEST OR FB 08820000
TBF PLOPM(,PL),OP$SYS * SYSTEM REQ TO DFF TERMINAL, FB 08830000
CLI PL$OPM(,PL),OPINV * OR A DFF INVITE FB 08840000
JC CMGOII,FLSOEQ YES - BYPASS POST OF DFF TASK.FB 08850000
SPACE 1 08860000
***** FB 08870000
* DFF REQUEST - GET,PUT OR STOP INVITE. POST DFF IF NECESSARY* FB 08880000
***** FB 08890000
SPACE 1 08900000
L TUBDTF(,XR2),DTF DTF ADDRESS FB 08910000
TBN PLOPC(,PL),OPPUT PUT REQUEST FB 08920000
TBF LCBAT2(,DTF),LCBACT AND LINE INACTIVE (EOT SENT) FB 08930000
JT CMGOII YES - JUST RETURN TO II FB 08940000
CLI PL$RTC(,PL),RCXEDT IF NO DATA TRANSMITTED FB 08950000
JH CMGOII YES - JUST RETURN TO II FB 08960000
SPACE 1 08970000
* POST DFF TASK FOR THE FOLLOWING USER OPS TO DFF TERMINAL: FB 08980000
* 1. SUCCESSFUL GET. FB 08990000
* 2. STOP INVITE WITH DATA (BECAME A GET). FB 09000000
* 3. SUCCESSFUL PUT WHEN EOT NOT SENT. FB 09010000
SPACE 1 09020000
B CMDFEQ QUEUE REQ FOR DFF AND POST. FB 09030000
J CMREDO RESCHED LINE IF NEEDED FB 09040000
SPACE 09050000
CMGOII EQU * * LOCAL FB 09060000
.F0400 ANOP 09070000
SPACE 1 09080000
B CMPOST POST REQUESTOR AND FREE BUFS B 09090000
SPACE 09100000
***** B 09110000
* IF LINE NOT ACTIVE - GO RESTART LINE, OTHERWISE HANDLE AN OP END B 09120000
***** B 09130000
SPACE 09140000
CMREDO EQU * * B 09150000
L PLTUBA(,PL),XR2 POINT TO THE TUB B 09160000
L TUBDTF(,XR2),DTF POINT TO THIS LINES DTF. B 09170000
&MIX SETA &N32+&N37+&N41 09180000
AIF (&MIX EQ '3').T0500 09190000
TBN PL$OPC(,PL),OPLSNS POLLING FOR STATUS ? 7/0/5B 09200000
JT CMEACT YES-DON'T DESTROY USER OP.7/0/5B 09210000
.T0500 ANOP 09220000
MNN PL$OPC(,PL),PL$OPM(,PL) SAVE CURRENT OP FOR ERP. B 09230000
CMEACT EQU * * LOCAL B 09240000
TBN LCBAT2(,DTF),LCBACT LINE ACTIVE ? B 09250000
BT CMOPND YES-GO CHECK OP END COUNT. B 09260000
B CMBSCH BR TO RESTART THE LINE B 09270000
.C0300 ANOP 09280000
MEND 09290000

```

MODULE-\$E075 , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* NAME: $E075 * 00030000
***** 00040000
$E075 00050000
  GBLB &NOM,&NDME,&NSW,&NBFR,&NMOVE,&NSCTL,&N2741 00060000
  TEXT 00070000
.* : 00080000
* R-05,C-00 CHANGE LEVEL MASTER 00090000
  AIF (&NOM).MEND 00110000
  TITLE '$E075/CMMOPE---MLTA OP END HANDLING' 00120000
***** 00130000
* * 00140000
* NAME--CMMOPE * 00150000
* * 00160000
* TITLE--MLTA OP END ANALYSIS * 00170000
* * 00180000
* FUNCTION--ANALYSIS EACH OP END FOR A TP LINE AND DETERMINE WHAT * 00190000
* IF ANYTHING MUST BE DONE TO COMPLETE THE CURRENT TP * 00200000
* OPERATION. ROUTE COMPLETED OPERATION BACK TO THE USER. * 00210000
* RESCHEDULE WORK ON THE LINE IF NO MORE OP ENDS TO BE * 00220000
* HANDLED. * 00230000
* * 00240000
* OPERATION-- * 00250000
* * 00260000
* . IF ABORT OPERATION, HANDLE THE ABORT UNTIL IT IS * 00270000
* COMPLETE. THEN RESCHEDULE THE LINE. * 00280000
* * 00290000
* . IF STOP INVITE REQUEST CALL $CC4SQ * 00300000
* * 00310000
* . IF SWALLOW THE DATA OPERATION, THEN IGNORE DATA. * 00320000
* * 00330000
* . IF MLTA OLT OP END CALL TRANSIENT TO HANDLE THE * 00340000
* OPERATION. THEN GO TO SCHEDULE THE NEXT OPERATION. * 00350000
* * 00360000
* . FIND THE OP ENDED PARAMETER LIST AND SET ON THE POLL * 00370000
* SKIP BIT. * 00380000
* * 00390000
* . TRACE THE OP END AFTER CALLING CHECK THRU $CC4TT. * 00400000
* * 00410000
* . IF AN ERROR OCCURRED CALL $CC4MA. ON RETURN * 00420000
* EITHER POST THE RESULTS TO THE USER, RESCHEDULE THE * 00430000
* LINE, OR HANDLE THE DATA IN THE LINE BUFFER. * 00440000
* * 00450000
* . IF WRITE OP END, THEN * 00460000
* - SET UP THE RETURN CODE FOR THE RESULTS OF THE * 00470000
* OPERATION. * 00480000
* * 00490000
* - IF PUT-THEN-GET, THEN SET UP THE GET OPERATION IN * 00500000
* THE INTERNAL OP CODE, THEN RESCHEDULE THE LINE. * 00510000
* * 00520000
* - IF OPERATION IS COMPLETED SET UP TO POST THE * 00530000
* REQUESTOR OF THE RESULTS. * 00540000
* * 00550000
* . IF READ OP END, THEN * 00560000
* * 00570000
* - IF DATA MODE ESCAPE RECOGNIZED IN THE INPUT DATA * 00580000
* STREAM THEN SET UP POST OF THE COMMAND PROCESSOR. * 00590000
* * 00600000
```



```

*           - IF VALID INPUT DATA, MOVE, TRANSLATE, TRUNCATE THE * 00610000
*           DATA AS REQUIRED. * 00620000
* * 00630000
*           . IF COMPLETED DATA OPERATION, THEN * 00640000
*           - REMOVE THE TP REQUEST FROM THE LINE QUEUE. * 00650000
*           - FREE UP PUT-NO-WAIT HOLD BUFFERS. * 00660000
*           - POST THE REQUESTOR OF THE TP OPERATION THAT IT IS * 00670000
*           COMPLETE. * 00680000
* * 00690000
* ENTRY POINT-- CMMOPE: HANDLE MLTA OP END. * 00700000
* * 00710000
* INPUT-- XR2 -> DTF THAT OP ENDED. * 00720000
* * 00730000
* OUTPUT-- * 00740000
*           CMSDTF - ADDRESS OF DTF HANDLED FOR LAST OP END. * 00750000
*           CMSPL - ADDRESS TO TP PARAMETER LIST FOR LAST OP END. * 00760000
* * 00770000
* EXTERNAL REFERENCES-- * 00780000
*           $CC4SQ - MLTA STOP II QUEUE ANALYSIS TRANSIENT. * 00790000
*           $CC4SK - MLTA SKIP BIT TRANSIENT. * 00800000
*           $CC4T2 - MLTA OLT OP END TRANSIENT. * 00810000
*           $CC4TT - CCP TRACE ROUTINE. * 00820000
*           $CC4MA - MLTA ERP TRANSIENT. * 00830000
*           $CC4JX - X IS APPROPRIATE TRANSLATE TRANSIENT. * 00840000
*           $CC4WR - TRANSLATE ERROR TRANSIENT. * 00850000
*           CMFMRT - FREEMAIN ROUTINE. * 00860000
*           CMONSK - MLTA POLL LIST SKIP BIT ON ROUTINE. * 00870000
*           CMWPGY - PUT-THEN-GET SUBROUTINE. * 00880000
*           CMSTOR - DETERMINE PARM LIST STORAGE NEEDS. * 00890000
*           CMGMRT - GETMAIN ROUTINE. * 00900000
*           CMPOST - POST TP COMPLETE. * 00910000
*           CMMVRT - INTERFACE TO CCP MOVE ROUTINE. * 00920000
* * 00930000
* EXIT, NORMAL--TO CMMTBY - TEST LINE BUSY FOR RESCHEDULING. * 00940000
*           TO CMMSCH - RESCHEDULE WORK ON A LINE * 00950000
*           TO CMPAII - POST REQUESTOR AFTER ERROR OP END. * 00960000
* * 00970000
***** * 00980000
EJECT * 00990000
CMMOPE EQU * * < ENTRY POINT > M 01000000
SPACE * 01010000
***** M 01020000
* MLTA ABORT CHECK AND HANDLING * M 01030000
***** M 01040000
SPACE * 01050000
TBN LCBATR(,XR2),LCBSTP WAS ABORT TO STOP READ M 01060000
JF CMSWL JUMP IF INDICATOR OFF M 01070000
SPACE * 01080000
* BRING IN TRANSIENT TO HANDLE OP END FOR STOP READ REQUEST M 01090000
SPACE * 01100000
SVC 0 ##### TRANSIENT CALL ##### M 01110000
DC AL1(CCPRIB) CCP SVC RIB. M 01120000
DC AL1(CC4SQ) TRANSIENT ID M 01130000
SPACE * 01140000
CMSWL EQU * * LOCAL M 01150000
SPACE * 01160000
* IGNORE THE OP END IS BOTH LCB SWALLOW AND BIT BUCKET BITS ARE ON M 01170000
* NO MATTER WHAT TYPE OF OPERATION WAS SCHEDULED M 01180000
SPACE * 01190000
TBN LCBATR(,XR2),LCBSWL+LCBTBK IS OP END TO BE IGNORED M 01200000

```

```

JT      CMSWLP          JUMP IF IGNORE OP END      M 01210000
SPACE 1                                01220000
* IF 2741 BIT BUCKET BIT ON OR          M 01230000
* IF SWALLOW SET AND THE OP END IS FOR AN INPUT OPERATION M 01240000
* THEN SWALLOW (BIT BUCKET) THE DATA  M 01250000
SPACE                                01260000
TBF     $MDOPC(,XR2),MLWRIT      ASSURE IT WAS NOT WRITE OP      M 01270000
JF      CMTABT          JUMP IF WRITE            M 01280000
SPACE                                01290000
* HAVE NON-WRITE OP - IF SWALLOW OR BIT BUCKET ON      M 01300000
* BIT BUCKET THE DATA THAT CAME IN          M 01310000
SPACE                                01320000
TBF     LCBATR(,XR2),LCBSWL+LCBTBK ISBIT BUCKET OR SWALLOW ON M 01330000
JT      CMTABT          JUMP IF NEITHER SWALLOW OR BIT M 01340000
*                                           M 01350000
SPACE                                01360000
*****                                M 01370000
* BIT BUCKET THE DATA THAT JUST CAME IN          * M 01380000
*****                                M 01390000
SPACE                                01400000
CMSWLP EQU *                  * LOCAL.          01410000
SBF     LCBATR(,XR2),LCBSWL+LCBTBK RESET LCB IGNORE OP END BITS M 01420000
SPACE                                01430000
* HAVE SWALLOW BIT SET - THUS MAY HAVE SKIP BIT OFF WHICH NEEDS TO BE M 01440000
* TURNED ON                                  M 01450000
SPACE                                01460000
* THIS TRANSIENT WILL SET ALL SKIP BITS ON FOR A POLLING LIST      M 01470000
* SOME WILL BE TURNED BACK OFF WHEN WE RESCHEDULE THE LINE.        M 01480000
SPACE                                01490000
SVC     0                      ##### TRANSIENT CALL #####      M 01500000
DC      AL1(CCPRIB)            CCP SVC RIB.                      M 01510000
DC      AL1(CC4SK)            SET POLL SKIP BITS                M 01520000
SPACE                                01530000
J       CMJTBY                JUMP TO TEST FOR LINE BUSY.        M 01540000
EJECT                                01550000
* IF OPERATION WAS SUCCESSFUL ABORT - THEN BR TO RESCHEDULE THE LINE M 01560000
SPACE                                01570000
CMTABT EQU *                  * LOCAL          M 01580000
CLI     $MDOPC(,XR2),$MCABT    WAS OP AN ABORT                  M 01590000
JNE     CMDFSV                JUMP IF NOT ABORT                  M 01600000
SPACE                                01610000
* HAD ABORT ATTEMPT - IF RETURN CODE 44 - JUMP TO HANDLE          M 01620000
SPACE                                01630000
CLI     $MDCMP(,XR2),$MCTNR    WAS COMP CODE 44                  M 01640000
JNE     CMDFSV                JUMP IF UNSUCCESSFUL ABORT        M 01650000
SPACE                                01660000
* HAD ABORT ATTEMPT                                  M 01670000
* ABORT WAS SUCCESSFUL SO LINE IS NOW FREE TO SCHEDULE NEXT OP      M 01680000
* IF HOLD BUFFER HAS BEEN GETMAINED FOR THIS LINE - FREEMAIN IT NOW M 01690000
SPACE                                01700000
CLI     LCBIBA-1(,XR2),NOBIT    IS HOLD BUFFER POINTER NULL      M 01710000
JE      CMJPRS                JUMP IF NO HOLD BUFFER            M 01720000
SPACE                                01730000
* HAVE HOLD BUFFER WHICH NEEDS TO BE FREEMAINED          M 01740000
SPACE                                01750000
L       LCBIBA(,XR2),XR2        POINT XR2 AT HOLD BUFFER        M 01760000
B       CMFMRT                BRANCH TO FREEMAIN                M 01770000
SPACE                                01780000
L       CMSDTF,XR2             POINT XR2 BACK AT DTF              M 01790000
MVI     LCBIBA-1(,XR2),NOBIT    SET HOLD BUFFER ADDR NULL        M 01800000

```

SPACE				01810000
CMJPRS EQU *		* LOCAL.		01820000
B CMMSCH		BRANCH TO RESCHEDULE THE LINE	M	01830000
SPACE 2				01840000
* DETERMINE IF OP END WAS FOR ONLINE TEST			M	01850000
SPACE				01860000
CMDFSV EQU *		* LOCAL.		01870000
SBN LCBATR(,XR2),LCBNIT		SET INITIAL OP BIT ON	M	01880000
TBN LCBATR(,XR2),LCBOLR		TEST LCB BIT FOR ONLINE TEST	M	01890000
JF CMPARM		IF NOT OLT, JUMP TO WORK WITH	M	01900000
*		PARM LIST	M	01910000
SPACE				01920000
* OP END WAS FOR ONLINE TEST REQUEST			M	01930000
* FOR WHICH THERE IS NO PARM LIST			M	01940000
SPACE				01950000
SVC 0		##### TRANSIENT CALL #####	M	01960000
DC AL1(CCPRIB)		CCP SVC RIB.	M	01970000
DC AL1(CC4T2)		OP END OF ONLINE TEST	M	01980000
SPACE				01990000
CMJTBY EQU *		* LOCAL.		02000000
B CMMTBY		BRANCH TO TEST IF LINE IS BUSY	M	02010000
EJECT				02020000
*****			M	02030000
* LOCATE OP ENDED PARAMETER LIST			* M	02040000
*****			M	02050000
SPACE				02060000
* XR2 POINTS AT DTF FOR THE LINE THAT OP ENDED.			M	02070000
SPACE				02080000
CMPARM EQU *		* LOCAL.	M	02090000
L LCBPLQ(,XR2),XR1		POINT XR1 AT 1ST PL IN QUEUE	M	02100000
SPACE				02110000
*****			M	02120000
* SET UP FOR FINDING OF PARM LIST WHICH OP ENDED			* M	02130000
*****			M	02140000
SPACE				02150000
AIF (&NSCTL).C0100				02160000
MVC CMSTMA(2),\$MDTMA(,XR2)		SAVE THE TERMINAL ADDR OF THE	CM	02170000
*		TERMINAL WHICH OP ENDED	CM	02180000
SPACE				02190000
.C0100 ANOP				02200000
* SET UP TEST INSTRUCTION FOR EITHER A READ OR WRITE MLTA OP CODE			M	02210000
* DEPENDING ON THE OP CODE IN THE DTF WHICH OP ENDED			M	02220000
SPACE				02230000
MVI CMTBOP+1,OPGET		SET UP TEST INSTR FOR READ	M	02240000
SPACE				02250000
AIF (&NSW).S0100				02260000
SPACE				02270000
* WAS OP END FOR WRITE DISCONNECT TO MLTA SWITCHED LINE			SM	02280000
SPACE				02290000
CLI \$MDOPC(,XR2),\$MWTDS		WAS OP A WRITE DISCONNECT	SM	02300000
JNE CMTSWR		JUMP IF OP WAS NOT WRITE DISC	SM	02310000
SPACE				02320000
* HAD OP END OF WRITE DISCONNECT SO SET LCBNIT TO INDICATE LINE DISC			SM	02330000
SPACE				02340000
SBF LCBATR(,XR2),LCBNIT		INDICATE LINE IS DISCONNECTED	SM	02350000
J CMOPT		JUMP TO LOOK FOR WRITE OP END	SM	02360000
SPACE				02370000
.S0100 ANOP				02380000
SPACE				02390000
CMTSWR EQU *		* LOCAL.	M	02400000

```

TBN $MDOPC(,XR2),MLWRIT WAS OP END FOR A WRITE M 02410000
JF CMMVSC JUMP IF NOT WRITE OP END M 02420000
CMOPUT EQU * * LOCAL. M 02430000
MVI CMTBOP+1,OPPUT SET UP TEST FOR WRITE OP END M 02440000
SPACE 02450000
* GET THE TERMINAL FEATURE TYPES BYTE FROM MLTA DTF FOR TESTING M 02460000
SPACE 02470000
CMMVSC EQU * * LOCAL. M 02480000
AIF (&NSCTL).C0200 02490000
MVC MLSTCL(1),$MDTFT(,XR2) SAVE TERMINAL FEATURES CM 02500000
.C0200 ANOP 02510000
EJECT 02520000
***** 02530000
* FIND PARAMETER LIST WITH OP CODE THAT MATCHES OP END * 02540000
***** 02550000
* XR1 POINTS AT FIRST PARM LIST IN LINE QUEUE M 02560000
SPACE 02570000
CMTBOP TBN PL$OPM(,XR1),# TEST FOR OP CODE MATCH M 02580000
AIF (&NSCTL).C0300 02590000
JF CMNXPM JUMP IF NO MATCH CM 02600000
SPACE 02610000
***** 02620000
* FOUND PL WITH MATCHING OP CODE CM 02630000
***** 02640000
SPACE 02650000
TBN MLSTCL,$MTPLT IS IT STATION CONTROL CM 02660000
JF CMMTCH JUMP IF NOT SINCE HAVE PL CM 02670000
SPACE 02680000
***** 02690000
* STATION CONTROL -- ALSO SEARCH THE TUBS FOR THE PARAMETER CM 02700000
* LIST WHOSE TERMINAL ADDRESS MATCHES TERMINAL ADDR CM 02710000
* IN DTF WHICH OP ENDED. CM 02720000
***** 02730000
SPACE 02740000
L PLTUBA(,XR1),XR2 POINT XR2 AT TUB. CM 02750000
CLC CMSTMA-1(1),TUBTMA-1(,XR2) COMPARE DTF TMA WITH TUB TMA CM 02760000
JE CMMTCH JUMP IF THEY MATCH CM 02770000
AGO .C0400 02780000
.C0300 ANOP 02790000
JT CMMTCH JUMP IF HAVE PARM LIST M 02800000
.C0400 ANOP 02810000
SPACE 02820000
* NO MATCH SO GET NEXT PARAMETER LIST IN QUEUE TO TEST AGAIN M 02830000
SPACE 02840000
CMNXPM EQU * * LOCAL. 02850000
L PLCHN(,XR1),XR1 POINT XR1 AT NEXT PL IN QUEUE M 02860000
B CMTBOP BRANCH TO TRY TO MATCH AGAIN M 02870000
EJECT 02880000
***** M 02890000
* PUT RETURN CODE IN PARAMETER LIST * M 02900000
***** M 02910000
SPACE 02920000
* XR1 POINTS AT THE OP ENDED PARM LIST M 02930000
SPACE 02940000
CMMTCH EQU * * LOCAL. M 02950000
SPACE 02960000
L CMSDTF,XR2 POINT XR2 AT DTF M 02970000
MVC CMSRTC(1),PL$RTC(,XR1) SAVE RETURN CODE FROM START OP.M 02980000
MVC PL$RTC(1,XR1),$MDCMP(,XR2) MOVE COMPLETION CODE TO PL M 02990000
SPACE 2 03000000

```

```

***** M 03010000
* TRACE CALL * M 03020000
***** M 03030000
SPACE 03040000
SVC 0 CALL M 03050000
DC AL1(CCPRIB) * THE CCP M 03060000
DC AL1(TRRIB) * TRACE M 03070000
DC AL1(TTMOPN) * TP OP END. M 03080000
SPACE 2 03090000
AIF (&NSCTL).C0500 03100000
TBN PL$OPM(,XR1),OPGET WAS OP END FOR A READ CM 03110000
JF CMMCMP JUMP IF NOT READ CM 03120000
SPACE 03130000
***** CM 03140000
* STATION CONTROL READ -- SET SKIP BIT * CM 03150000
***** CM 03160000
SPACE 03170000
B CMONSK BR TO SET POLL SKIP BIT ON CM 03180000
.C0500 ANOP 03190000
SPACE 3 03200000
CMMCMP EQU * * LOCAL. M 03210000
SPACE 03220000
ST CMSPL,XR1 SAVE PARM LIST ADDR M 03230000
SPACE 03240000
SBF PL$RTC(,XR1),BIT1 SET BIT OFF TO FORM RET CODE M 03250000
SPACE 03260000
* TEST FOR TP ERROR (COMPLETION CODE X'40' OR X'42' ARE OK) M 03270000
SPACE 03280000
TBF PL$RTC(,XR1),MLSCCS TEST FOR SUCCESSFUL OP END M 03290000
JT CMMSCS JUMP IF SUCCESSFUL M 03300000
AIF (&N2741).T0100 03310000
SPACE 03320000
* IF TERMINAL INTERRUPT FROM 2741 - HANDLE AS GOOD OP END 4M 03330000
SPACE 03340000
CLI $MDTTP(,XR2),ML2741 IS IT 2741 4M 03350000
JNE CMMERR JUMP IF NOT 2741 4M 03360000
SPACE 03370000
CLI PL$RTC(,XR1),MLCITP WAS IT TERMINAL INTERRUPT 4M 03380000
TBN PL$OPM(,XR1),OPPUT WAS IT OUTPUT OPERATION 4M 03390000
JC CMSCC1,NONE+FALSE+HI+LO JUMP IF TERMINAL INTERRUPT 4M 03400000
* FROM THE 2741 4M 03410000
.T0100 ANOP 03420000
EJECT 03430000
***** 03440000
* TP ERROR - CALL TRANSIENT TO HANDLE 03450000
***** 03460000
SPACE 03470000
CMMERR EQU * * LOCAL. M 03480000
SVC 0 ##### TRANSIENT CALL ##### M 03490000
DC AL1(CCPRIB) CCP SVC RIB. M 03500000
DC AL1(CC4MA) MLTA ERROR HANDLER M 03510000
SPACE 03520000
* RETURN TO NSI - TO DEQUEUE THE CURRENT PARM LIST. M 03530000
* NSI+4 - TO RESCHEDULE WORK ON THE LINE. M 03540000
AIF (&NBFR).T0200 03550000
* NSI+8 - TO HANDLE SUCCESSFUL DATA RECEIVED. BM 03560000
* NSI+12 - TO POST PARM LIST AND NOT RESCHEDULE THE LINE. BM 03570000
.T0200 SPACE 03580000
*--> NSI 03590000
B CMMDEQ DEQUEUE THE PARAMETER LIST M 03600000

```

```

SPACE 03610000
*--> NSI+4 03620000
B CMMTBY GO TEST LINE FOR BUSY. M 03630000
AIF (&NBFR).T0300 03640000
.* THE FOLLOWING TWO BRANCHES ARE GENERATED ONLY FOR 2740 SYSTEMS BM 03650000
SPACE 03660000
*--> NSI+8 03670000
B CMMSCS B TO HANDLE AS IF SUCCESS BM 03680000
SPACE 03690000
*--> NSI+12 03700000
B CMPAII BRANCH AROUND RESCHEDULING BM 03710000
.T0300 ANOP 03720000
EJECT 03730000
***** M 03740000
* SUCCESSFUL OP END FOR MLTA * M 03750000
***** M 03760000
SPACE 03770000
CMMSCS EQU * * LOCAL. M 03780000
SPACE 03790000
* AT THIS POINT SET ANY BITS ON IN THE RETURN CODE FROM THE RETURN M 03800000
* CODE SAVED AS PART OF THE START OF THE OPERATION M 03810000
SPACE 03820000
SBN PL$RTC(,XR1),# RESET RETURN CODE FROM OP M 03830000
CMSRTC EQU *-2 Q BYTE OF SBN INSTRUCTION M 03840000
SPACE 03850000
CMSCC1 EQU * GOOD OP END BUT AT THIS LABEL M 03860000
* THE RETURN CODE IS NOT RESET M 03870000
* DETERMINE WHAT KIND OF OP COMPLETED BY CHECKING THE OP CODE M 03880000
* RESIDING IN THE PARAMETER LIST M 03890000
SPACE 03900000
TBN PL$OPM(,XR1),OPPUT WAS IT WRITE M 03910000
JF CMRDME BRANCH IF READ M 03920000
TITLE '$E075/CMMOPE---MLTA WRITE OP END' 03930000
***** M 03940000
* MLTA WRITE OP END HANDLING * M 03950000
***** M 03960000
SPACE 2 03970000
TBN PLOPC(,XR1),OPGET IF NOT A PUT THEN GET M 03980000
BF CMMDEQ GO DEQUEUE PARM LIST M 03990000
SPACE 04000000
B CMWPGY WRITE - CHECK PUT THEN GET M 04010000
SPACE 04020000
B CMMSCH RESCHEDULE THE LINE M 04030000
TITLE '$E075/CMMOPE---MLTA READ OP END' 04040000
***** M 04050000
* MLTA READ OP END HANDLING * M 04060000
***** M 04070000
SPACE 04080000
CMRDME EQU * * LOCAL. M 04090000
SPACE 04100000
* GET TUB ADDRESS FROM THE PARAMETER LIST M 04110000
SPACE 04120000
CLI PLOPC(,XR1),OPINV IS OPERATION AN INVITE M 04124000
L PLTUBA(,XR1),XR1 POINT XR1 TO TUB M 04130000
JE CMUGET YES. LEAVE TUBIIS ON M 04134000
SBF TUBAT2(,XR1),TUBIIS SET OFF INVITE SCHEDULED M 04140000
CMUGET EQU * M 04144000
MVC SAVTA2(2),TUBTA2(,XR1) MOVE TERM ATTR TO SAVE AREA M 04150000
EJECT 04160000
AIF (&NDME).D0100 04170000

```

```

***** DM 04180000
*          DATA MODE ESCAPE CHECK * DM 04190000
***** DM 04200000
SPACE 04210000
* IF THIS TERMINAL IS IN DATA MODE AND IS ALSO A REQUESTING TERMINAL DM 04220000
* FOR THE PROGRAM TO WHICH IT IS CURRENTLY ALLOCATED - THEN CM MUST DM 04230000
* CHECK FOR A DATA MODE ESCAPE COMMAND DM 04240000
SPACE 04250000
TBF TUBAT2(,XR1),TUBCMD TUB NOT IN COMMAND MODE, BUT DM 04260000
TBN TUBAT2(,XR1),TUBDTA * IS IN DATA MODE, AND DM 04270000
TBN TUBAT1(,XR1),TUBREQ * IS REQUESTER OF PROGRAM ? DM 04280000
JF CMMDAT JUMP IF NOT DATA MODE ESCAPE. DM 04290000
SPACE 04300000
* BUILD TRANSLATE LIST TO TRANSLATE 1ST 6 BYTES OF THE RECORD. 0 DM 04310000
SPACE 04320000
* USE THE SPECIAL TRANSLATE LIST PROVIDED JUST FOR THIS PURPOSE. DM 04330000
* ALL FIELDS IN THE TRANSLATE LIST ARE SET EXCEPT FOR THE FROM ADDR. DM 04340000
SPACE 04350000
LA CMDMTL,XR1 POINT TO XLATE PARM LIST. DM 04360000
MVC TLFMA(2,XR1),$MDCRA(,XR2) MOVE 'FROM' ADDR TO TL DM 04370000
MVC CMTDME(1),LCBLLE(,XR2) MOVE TRANSLATE TRANSIENT ID DM 04380000
SVC 0 ##### TRANSIENT CALL ##### DM 04390000
DC AL1(CCPRIB) CCP SVC RIB. DM 04400000
CMTDME DC AL1(0) TRANSLATE TRANSIENT ID DM 04410000
SPACE 04420000
* COMPARE TRANSLATE INPUT AGAINST DATA MODE ESCAPE COMMAND SEQUENCE DM 04430000
SPACE 04440000
CLC CMDMEB+5(6),CMDME CHECK FOR DME COMMAND DM 04450000
JNE CMMDAT JUMP IF NOT DATA MODE ESCAPE DM 04460000
SPACE 3 04470000
***** DM 04480000
* DATA MODE ESCAPE SEQUENCE RECEIVED * DM 04490000
***** DM 04500000
SPACE 1 04510000
L CMSPL,XR1 POINT XR1 AT PARM LIST DM 04520000
L PLTUBA(,XR1),XR2 POINT XR2 AT TUB DM 04530000
SBN TUBAT2(,XR2),TUBCMD PUT TUB IN COMMAND INTERRUPT DM 04540000
* * MODE (CIM). DM 04550000
ST TUBDM@(,XR2),XR1 STORE PARM LIST ADDR IN TUB DM 04560000
L TUBDTF(,XR2),XR2 XR2--> DTF FOR THIS LINE. DM 04570000
B CMMDEQ BR TO DEQUEUE THE PARM LIST DM 04580000
EJECT 04590000
.D0100 ANOP 04600000
CMMDAT EQU * * LOCAL. M 04610000
L CMSPL,XR1 POINT XR1 AT PARM LIST. M 04620000
SPACE 04630000
***** M 04640000
* INVITE INPUT OP END -- COMPARE HOLD BUFFER SPACE AVAIL TO NEED * M 04650000
***** M 04660000
SPACE 04670000
B CMSTOR DETERMINE STORAGE NEEDED BY OP. M 04680000
SPACE 04690000
B CMGBUF GO GET LEAST AMOUNT NEEDED. M 04700000
SPACE 04710000
L CMSPL,XR1 XR1--> TP PARM LIST. M 04720000
SPACE 04730000
* MOVE THE HOLD BUFFER ADDRESS TO THE PARM LIST M 04740000
SPACE 04750000
MVC PLRECA(2,XR1),LCBIBA(,XR2) MOVE HOLD BUF ADDR TO PL M 04760000
MVI LCBIBA-1(,XR2),NOBIT ZERO HIGH ORDER BYTE OF BUF AD M 04770000

```

```

SPACE 04780000
***** M 04790000
* SET UP PARAMETER LIST FOR MOVE OR TRANSLATE * M 04800000
***** M 04810000
SPACE 04820000
* ANALYZE THE RECORD LENGTH REQUESTED BY THE USER AGAINST THE RECORD M 04830000
* LENGTH ACTUALLY RECEIVED AS A RESULT OF THE READ M 04840000
* IF RECEIVED EQUAL OR LESS THAN REQUESTED - GIVE USER ONLY WHAT RECV M 04850000
* IF RECEIVED GREATER THAN LENGTH REQUESTED, TRANSLATE ONLY UP TO M 04860000
* LENGTH REQUESTED BY USER AND PUT INDICATION IN RETURN CODE THAT M 04870000
* ALTHOUGH READ WAS SUCCESSFUL, TERMINAL OPERATOR TRANSMITTED MORE M 04880000
* DATA THAN WAS REQUESTED BY THE USER M 04890000
SPACE 04900000
* MOVE IN 'FROM' AND 'TO' ADDRESSES WHICH ARE THE SAME FOR BOTH THE M 04910000
* TRANSLATE AND THE MOVE LIST M 04920000
SPACE 04930000
MVC #CMMVL+MVLFRA(2), $MDCRA(,XR2) MOVE FROM ADDR TO MV LIST. M 04940000
MVC #CMMVL+MVLTOA(2), PLRECA(,XR1) MOVE TO ADDR TO MOVE LIST. M 04950000
SPACE 04960000
AIF (&NMOVE).F0100 04970000
* DETERMINE IF TRANSLATE WAS REQUESTED VM 04980000
* FOR SYSTEM REQUEST - ALWAYS TRANSLATE VM 04990000
SPACE 05000000
TBF PLOPM(,XR1),OP$SYS IS IT USER REQUEST, AND VM 05010000
TBN SAVTAL,TASTRN IS DON'T TRANSLATE SPECIFIED ?VM 05020000
JT CMRMOV YES-JUMP TO STRAIGHT MOVE. VM 05030000
.F0100 ANOP 05040000
EJECT 05050000
***** M 05060000
* TRANSLATE * M 05070000
***** M 05080000
SPACE 05090000
MVC #CMTRL+TLFRML(2), $MDCRL(,XR2) 'FROM' AREA LENGTH TO TL M 05100000
MVC #CMTRL+TLTOL(2), PLINL(,XR1) 'TO' AREA LENGTH TO TL M 05110000
SPACE 05120000
* DETERMINE IF UPPER OR LOWER CASE REQUESTED. M 05130000
* FOR SYSTEM REQUEST - ALWAYS TRANSLATE TO UPPER CASE. M 05140000
SPACE 05150000
TBF PLOPM(,XR1),OP$SYS IS IT USER REQUEST, AND M 05160000
TBN SAVTAL,TASCAS * LOWER CASE XLATE SPECIFIED ? M 05170000
LA #CMTRL,XR1 POINT XR1 AT TRANSLATE LIST M 05180000
SPACE 05190000
* MOVE IN ID OF UPPER CASE TRANSLATE TRANSIENT M 05200000
SPACE 05210000
MVC CMTRD(1),LCBLCE(,XR2) MOVE UPPER CASE TRANSIENT ID M 05220000
JF CMDOTR JUMP IF NOT USER LOWER CASE XL.M 05230000
SPACE 05240000
* MOVE IN TRANSIENT ID OF TRANSLATE TRANSIENT FOR LOWER CASE EBCDIC M 05250000
SPACE 05260000
MVC CMTRD(1),LCBLLE(,XR2) MOVE IN LOWER CASE TRANS ID M 05270000
SPACE 05280000
* TRANSLATE THE DATA FROM THE LINE BUFFER TO THE RECORD AREA POINTED M 05290000
* TO BY THE PARAMETER LIST M 05300000
* THE TRANSLATE ROUTINE WILL ALWAYS POST CLEAR THE INPUT AREA TO BLKS M 05310000
SPACE 05320000
CMDOTR EQU * * LOCAL M 05330000
SVC 0 ##### TRANSIENT CALL ##### M 05340000
DC AL1(CCPRIB) CCP SVC RIB. M 05350000
CMTRD DC AL1(0) TRANSLATE TRANSIENT ID M 05360000
SPACE 05370000

```



```

*      TBN   TLRTC(,XR1),TLERR      WAS THERE TRANSLATE ERROR      M 05380000
*      L     CMSPL,XR1              X'10' - INVALID CHAR REPLACED M 05390000
      JF     CMTCR                  JUMP IF NO TRANSLATE ERROR      M 05400000
      SPACE                                     05410000
*      HAD TRANSLATE ERROR.          M 05420000
      SPACE                                     05430000
      SVC    0                      ##### TRANSIENT CALL ##### M 05440000
      DC     AL1(CCPRIB)             CCP SVC RIB.                  M 05450000
      DC     AL1(CC4WR)             TRANSLATE ERROR TRANSIENT ID M 05460000
      SPACE                                     05470000
*      RETURN TO NSI - TO DEQUEUE AND HANDLE AS A USER ERROR. M 05480000
*      NSI+4 - TO HANDLE AS TERMINAL IN CCP ERP.                M 05490000
      SPACE                                     05500000
*--> NSI.                               M 05510000
      B     CMMDEQ                  JUMP AND HANDLE AS TP ERROR M 05520000
      SPACE 1                               05530000
*--> NSI+4.                               M 05540000
      B     CMMSCH                  BR TO RESCHEDULE THE LINE M 05550000
      EJECT                               05560000
***** M 05570000
* IF LAST CHARACTER OF THIS MESSAGE A CARRIAGE RETURN, SET * M 05580000
* TUB BIT SO IDLES WILL BE SENT WITH NEXT MESSAGE. * M 05590000
***** M 05600000
      SPACE 1                               05610000
CMTCR EQU *                               * LOCAL M 05620000
      L     PLTUBA(,XR1),XR1          POINT XR1 AT THE TUB M 05630000
      SBF   TUBCHR(,XR1),TUB@SL+TUBNID SET OFF LINE LOCATION BITS M 05640000
      TBN   #CMTRL+TLRTC,TLCREL      WAS LAST CHAR A CARRIAGE RET M 05650000
      JF     CMNOCR                  JUMP IF NOT CARRIAGE RETURN M 05660000
      SPACE                                     05670000
*      LAST CHARACTER TRANSLATED WAS CARRIAGE RETURN SO TYPEWRITER M 05680000
*      IS AT THE START OF A NEW LINE. M 05690000
      SPACE                                     05700000
      SBN   TUBCHR(,XR1),TUB@SL      SET AT START OF LINE BIT ON M 05710000
      SPACE                                     05720000
*      DETERMINE FROM LENGTH OF INPUT WHETHER IDLE CHARACTERS ARE STILL M 05730000
*      NEEDED AT BEGINNING OF NEXT OUTPUT. (MUST HAVE TIME FOR CARRIAGEM 05740000
*      TO RETURN TO HOME POSITION.) M 05750000
      AIF   (&NBFR).T0400            05760000
*      BUFFERED RECEIVE TERMINALS DO NOT NEED IDLE CHARACTERS FOR CR. BM 05770000
      SPACE                                     05780000
      L     CMSDTF,XR2              POINT XR2 AT THE DTF BM 05790000
      TBF   $MDTFR(,XR2),$MTBFR     BUFFERED RECEIVE, OR LESS BM 05800000
      CLI   #CMTRL+TLTOL,CMCRTM     THAN MAX HANDLED W/O IDLES ? BM 05810000
      JC    CMNOCR,ANY+LO+FALSE     JUMP-LESS THAN MAX/OR BUFF RCVBM 05820000
      AGO   .T0500                  05830000
.T0400 ANOP                            05840000
      SPACE                                     05850000
      CLI   #CMTRL+TLTOL,CMCRTM     LESS THAN MAX HANDLED W/O IDLES M 05860000
      JL    CMNOCR                  JUMP-LESS THAN MAX W/O IDLES. M 05870000
.T0500 ANOP                            05880000
      SPACE                                     05890000
*      INPUT LINE FOLLOWED BY CARRIAGE RETURN INDICATES THAT IDLES ARE M 05900000
*      NEEDED ON NEXT OUTPUT TO THIS TERMINAL. M 05910000
      SPACE                                     05920000
      SBN   TUBCHR(,XR1),TUBNID     SET ON IDLES NEEDED BIT M 05930000
      SPACE                                     05940000
CMNOCR EQU *                               * LOCAL M 05950000
      AIF   (&NMOVE).F0200          05960000

```

```

J      CMSEFL          JUMP TO SET THE LGTH FOR USER VM 05980000
EJECT                                05990000
***** VM 06000000
* PERFORM STRAIGHT MOVE ON DATA WITHOUT TRANSLATING OR ANALYZING IT VM 06010000
*   NOTE: NO CLEARING OF END OF SHORT RECORD IS DONE. VM 06020000
***** VM 06030000
SPACE                                06040000
CMRMOV EQU *          * LOCAL. VM 06050000
SPACE                                06060000
* DETERMINE WHICH IS LESS - THE LENGTH OF THE RECORD REQUESTED, OR VM 06070000
* THE LENGTH OF THE DATA KEYED. USE THE LESSER OF THE TWO. VM 06080000
SPACE                                06090000
MVC #CMMVL+MVLTO(2), $MDCRL(,XR2) USE ACTUAL LENGTH. VM 06100000
CLC $MDCRL(2,XR2),PLINL(,XR1) COMPARE ACTUAL VS ASKED LGTH VM 06110000
JNH CMDOMV          JUMP IF ACTUAL LESS OR EQUAL. VM 06120000
SPACE                                06130000
* USE REQUESTED LENGTH. VM 06140000
SPACE                                06150000
MVC #CMMVL+MVLTO(2),PLINL(,XR1) USE REQUESTED RECORD LGTH. VM 06160000
SPACE                                06170000
CMDOMV EQU *          * LOCAL VM 06180000
MVI #CMMVL+MVLTY,NOBIT SET TYPE FOR NO ATR SWAP. VM 06190000
B CMMVRT          BR TO MOVE ROUTINE VM 06200000
.F0200 ANOP                                06210000
EJECT                                06220000
***** M 06230000
* COMPARE LENGTH REQUESTED VS LENGTH RECEIVED AND IF RECEIVED GREATER M 06240000
* SET BIT ON IN RETURN CODE TO INDICATE MESSAGE TRUNCATED M 06250000
***** M 06260000
SPACE                                06270000
CMSEFL EQU *          * LOCAL. M 06280000
L CMSPL,XR1          POINT TO THE TP PARM LIST. M 06290000
MVC PLEFFL(2,XR1),#CMTRL+TLTOL MOVE EFFECTIVE READ LENGTH M 06300000
CLC PLEFFL(2,XR1),PLINL(,XR1) COMPARE ACTUAL VS ASKED LENGTH M 06310000
JNH CMMDEQ          JUMP TO DEQ IF ACTUAL LT OR EQ. M 06320000
SBN PL$RTC(,XR1),RCXDTR SET DATA TRUNCATED. ACTUAL GT. M 06330000
MVC PLEFFL(2,XR1),PLINL(,XR1) SET EFFECTIVE LENGTH TO ASKED M 06340000
TITLE '$E075/CMMOPE---MLTA OP END HANDLING, DEQUEUE PARM LIST' 06350000
***** M 06360000
* DEQUEUE READ OR WRITE MLTA PARAMETER LIST * M 06370000
***** M 06380000
SPACE                                06390000
* TAKE CHAIN ADDR POINTER FROM THIS PARM LIST AND PLACE IN CHAIN ADDR M 06400000
* OF PARM LIST THAT POINTED TO THIS PARM LIST. M 06410000
SPACE                                06420000
CMMDEQ EQU *          * WITHIN CMMOPE. M 06430000
B CMDEQ          GO TO DEQUEUE SUBROUTINE. M 06440000
SPACE                                06450000
B CMPOST          POST REQUESTOR AS NECESSARY M 06460000
SPACE                                06470000
B CMMSCH          RESCHEDULE THE LINE M 06480000
.MEND ANOP                                06490000
MEND                                06500000

```

```

MODULE-$E082 , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO 00010000
***** 00020000
.* NAME: $E082 * 00030000
***** 00040000
$E082 00050000
  GBLB &NOM,&NSW,&N2741 00060000
  TEXT 00070000
* R-06,C-00 CHANGE LEVEL 00080000
  AIF (&NOM).MEND 00100000
  TITLE '$E082/CMMREQ---NEW MLTA REQUEST HANDLING' 00110000
***** 00120000
* * 00130000
* NAME--CMMREQ - 'ACCEPT NEW MLTA TP REQUESTS' * 00140000
* * 00150000
* FUNCTION--ACCEPT NEW TP PARAMETER LIST FOR MLTA TERMINAL. * 00160000
*   PERFORM THE FUNCTION REQUESTED IF IT CAN BE HANDLED * 00170000
*   IMMEDIATELY, OTHERWISE, PLACE THE PARAMETER INTO THE * 00180000
*   LCB LINE QUEUE OF WORK TO BE DONE. * 00190000
* * 00200000
* OPERATION-- * 00210000
* * 00220000
*   . IF THIS IS A DISCONNECT REQUEST, THEN INSURE THAT THE * 00230000
*   LINE IS NOT CONNECTED TO THE SPECIFIEC TERMINAL. IF * 00240000
*   NO INVITE REQUEST IS INVOLVED, THEN POST TP COMPLETED * 00250000
*   TO THE SYSTEM. IF AN INVITE IS INDICATED THEN TREAT * 00260000
*   LIKE AN INVITE ONLY REQUEST. * 00270000
* * 00280000
*   . IF A PUT OPERATION TO A TERMINAL IN CCP ERP, IGNORE * 00290000
*   THE PUT OPERATION. POST THE USER TP COMPLETE IF NO * 00300000
*   INPUT OPERATION INDICATED. IF A PUT-THEN-GET TREAT * 00310000
*   LIKE A GET/INVITE ONLY OPERATION. * 00320000
* * 00330000
*   . IF MLTA ONLINE TEST (OLT) SET UP THE EXECUTION OF THE * 00340000
*   MLTA OLT REQUEST. * 00350000
* * 00360000
*   . IF A STOP INVITE OR PURGE I/O CALL IN THE MLTA ($CC4PG) * 00370000
*   TRANSIENT. ON RETURN FROM THIS TRANSIENT DO * 00380000
*   WINDUP OPERATIONS OF SCHEDULING TP REQUEST. * 00390000
* * 00400000
*   . IF A TP REQUEST THAT REQUIRES DATA TRANSFER, THEN QUEUE * 00410000
*   THE REQUEST ONTO THE LINE QUEUE FOR THE APPROPRIATE * 00420000
*   LCB. * 00430000
* * 00440000
*   . AFTER QUEUEING UP A NEW TP REQUEST, CHECK THE STATUS * 00450000
*   OF LINE TO: * 00460000
*   - INCLUDE A NEW GET/INVITE IF POLLING IS ALREADY * 00470000
*   GOING ON. * 00480000
*   - STOP A POLLING SEQUENCE IF A PUT REQUEST WAS * 00490000
*   JUST RECEIVED. * 00500000
* * 00510000
*   . EXIT TO THE RESCHEDULE LOGIC TO: * 00520000
*   - POST COMPLETION OF THE TP SCHEDULED OPERATION. * 00530000
*   - TO PERFORM THE NEXT PRIORITY ACTION ON THE LINE. * 00540000
*   - TO CHECK FOR OTHER WORK WITHIN 'CM' THAT CAN BE DONE * 00550000
*   AT THIS TIME. * 00560000
* * 00570000
* ENTRY POINTS- CMMREQ: HANDLE NEW MLTA TP REQUESTS. * 00580000
*   CMMTBY: TEST LINE BUSY FOR RESCHEDULING. * 00590000
* * 00600000

```

```

* INPUT--
* XR1 - ADDRESS OF TP REQUEST PARAMETER LIST.
* XR2 - ADDRESS OF THE LCB(DTF) FOR THIS REQUEST.
*
* OUTPUT--
* CMSPL - ADDRESS OF THE TP REQUEST LAST PLACED INTO THE
* LINE QUEUE.
* CMSDTF - ADDRESS OF THE LINE DTF FOR THE LAST TP REQUEST
* SCHEDULED.
*
* EXTERNAL REFERENCES--
* $CC4T1 - TRANSIENT TO HANDLE MLTA OLT REQUEST.
* $CC4PG - TRANSIENT TO HANDLE MLTA STOP II/PURGE I/O.
* MLTIO1 - MLTA IOCS SIO ROUTINE.
* $TRACE - CCP TRACE ROUTINE.
* CMERPC - INTERFACE TO IGNORE A PUT TO A TERMINAL IN CCP ERP.
* CMQUE - QUEUE NEW TP REQUEST TO LCB LINE QUEUE.
*
* EXIT, NORMAL--TO CMPAII - POST DISCONNECTED OPERATION COMPLETED.
* TO CMMSCH - RESCHEDULE A MLTA LINE.
* TO CMMTBY - TEST MLTA LINE BUSY BEFORE RESCHEDULING.
* TO CMRQBF - RESET TP REQUEST BITS.
*
*****
EJECT
CMMREQ EQU * * < ENTRY POINT > M 00880000
TBN PLOPC(,XR1),OPPUT+OPGET IS OP A PUT THEN GET M 00890000
AIF (&NSW).S0100 00900000
JT CMPTG JUMP IF PUT THEN GET SM 00910000
SPACE 00920000
***** SM 00930000
* DISCONNECT MLTA SWITCHED LINE IF REQUESTED TO THE CURRENT TUB * SM 00940000
***** SM 00950000
* HAVE PUT ONLY OPERATION TO A TERMINAL. SM 00960000
* IF IT IS PUT DISCONNECT TO NON-CONNECTED LINE, TREAT AS IF IT SM 00970000
* WORKED. SM 00980000
SPACE 00990000
TBN PLOPM(,XR1),OPDISC IS IT PUT DISCONNECT SM 01000000
TBF $MDAT2(,XR2),$MABSY CHECK LINE NOT BUSY SM 01010000
TBF LCBATR(,XR2),LCBNIT TEST LINE NOT CONNECTED SM 01020000
JF CMOPST JUMP IF NEED TO ISSUE OP SM 01030000
SPACE 01040000
* HAVE PUT DISCONNECT TO LINE WHICH IS NOT CONNECTED SM 01050000
SPACE 01060000
B CMPAII JUMP TO POST COMPLETE SM 01070000
SPACE 01080000
AGO .S0200 01090000
.S0100 ANOP 01100000
JF CMOPST JUMP IF NOT PUT THEN GET M 01110000
.S0200 ANOP 01120000
SPACE 01130000
***** M 01140000
* PUT THEN GET SO SET CURRENT OPERATION TO PUT ONLY * M 01150000
***** M 01160000
SPACE 01170000
CMPTG EQU * * LOCAL. M 01180000
SBF PL$OPM(,XR1),OPGET SET OFF GET BIT M 01190000
SPACE 01200000
CMOPST EQU * * LOCAL M 01210000
SPACE 01220000

```

```

***** M 01230000
* IF PUT TO TERMINAL IN CCP ERP - IGNORE THE PUT. * M 01240000
***** M 01250000
SPACE 01260000
B CMERPC GO CHECK FOR AND HANDLE PUT M 01270000
* * TO TERMINAL IN ERP. M 01280000
* IF TERMINAL IS IN ERP AND OP WAS A PUT ONLY, CMERPC WILL GO M 01290000
* DIRECTLY TO CMPAII. IF IN ERP AND OP WAS A PUT THEN GET M 01300000
* CMERPC RETURNS HERE TO SCHEDULE GET. M 01310000
EJECT 01320000
TBN PLOPM(,XR1),OPOLT IS IT OLT REQUEST M 01330000
JF CMSTML JUMP IF NOT OLT M 01340000
SPACE 01350000
***** M 01360000
* ONLINE TEST REQUEST TO MLTA TERMINAL * M 01370000
***** M 01380000
SPACE 01390000
* THIS TRANSIENT WILL PLUG THE TUB AND LCB M 01400000
SPACE 01410000
SVC 0 ##### TRANSIENT CALL ##### M 01420000
DC AL1(CCPRIB) CCP SVC RIB. M 01430000
DC AL1(CC4T1) HANDLE ONLINE TEST PARM LIST M 01440000
SPACE 01450000
J CMMTBY GO TO TEST LINE BUSY M 01460000
SPACE 2 01470000
CMSTML EQU * * LOCAL M 01480000
TBN PLOPM(,XR1),OPSTOP IS STOP BIT ON M 01490000
JF CMMINV NO-MOVE PL M 01500000
SPACE 1 01510000
***** M 01520000
* STOP INVITE INPUT, OR SYSTEM PURGE I/O. * M 01530000
***** M 01540000
SPACE 1 01550000
SVC 0 ##### TRANSIENT CALL ##### M 01560000
DC AL1(CCPRIB) CCP SVC RIB. M 01570000
DC AL1(CC4PG) MLTA PURGE I/O TRANSIENT M 01580000
SPACE 01590000
B CMRQBF BR TO SET TP REQUEST BITS OFF M 01600000
EJECT 01610000
***** M 01620000
* IF THIS OPERATION IS TO A TUB IN CCP ERP, THEN CANNOT HAVE A PUT M 01630000
* PUT THAT WILL BE SCHEDULED. M 01640000
***** M 01650000
SPACE 01660000
CMMINV EQU * * LOCAL M 01670000
TBN TUBAT3(,XR2),TUBERP IS TUB IN CCP ERP M 01680000
L TUBDTF(,XR2),XR2 POINT XR2 AT THE DTF M 01690000
JT CMPNWX JUMP IF IN CCP ERP. M 01700000
SPACE 01710000
TBN PL$OPM(,XR1),OPPNW IS IT PUT NO WAIT 01720000
JF CMPNWX IF NOT JUMP TO END OF PNW CHECK. 01730000
TBF PLOPM(,XR1),OP$SYS IS IT USER REQUEST M 01740000
CLI LCBOLT(,XR2),NOBIT CHECK FOR NO ONLINE TEST M 01750000
JC CMPNWX,ANY+FALSE+EQ JUMP-NOT USER PNW WITH OLT Q'D.M 01760000
CMPNWX SBF PL$OPM(,XR1),OPNOW MAKE OPERATION A WAIT 01770000
EQU * * LOCAL 01780000
SPACE 01790000
B CMQUE QUEUE THE TP REQUEST. 01800000
EJECT 01810000
***** M 01820000

```

```

* MLTA LINE BUSY CHECK * M 01830000
***** M 01840000
SPACE 01850000
CMMTBY EQU * * < ENTRY POINT > M 01860000
L CMSDTF, XR2 ADDR OF LCB TO XR2 M 01870000
TBN $MDAT2(, XR2), $MABSY TEST LINE BUSY BIT M 01880000
BF CMMSCH RESCHEDULE THE LINE M 01890000
SPACE 01900000
***** M 01910000
* MLTA LINE BUSY * M 01920000
***** M 01930000
SPACE 01940000
TBN $MDOPC(, XR2), MLREAD IS BUSY OP A READ M 01950000
AIF (&N2741).T0100 01960000
CLI $MDTTP(, XR2), ML2741 IS THIS 2741 4M 01970000
BC CMPAII, ANY+FALSE+EQ JUMP IF NOT READ OR IF 2741 4M 01980000
AGO .T0200 01990000
.T0100 ANOP 02000000
BF CMPAII JUMP IF NOT READ M 02010000
.T0200 ANOP 02020000
SPACE 02030000
* ATTEMPT TO ABORT THE PREVIOUS READ M 02040000
SPACE 02050000
MVI $MDOPC(, XR2), $MCABT MOVE ABORT OP CODE TO DTF M 02060000
B MLTIO1 BR TO MLTA IOCS M 02070000
SPACE 02080000
B CMTRCE CALL M 02090000
DC AL1(CCPRIB) * CCP M 02100000
DC AL1(TRRIB) * TRACE M 02110000
DC AL1(TTMSIO) * FOR MLTA SIO. M 02120000
SPACE 2 02130000
B CMPAII GO EXIT FROM NEW REQUEST. 02140000
.MEND ANOP 02150000
MEND 02160000

```

MODULE-\$E085 , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
* NAME: $E085 * 00030000
***** 00040000
$E085 00050000
  GBLB &NOB,&MIN,&NINT,&NPBY 00060000
  GBLB &NPP,&NMP,&NSWL,&NCS,&NITB,&NTSP,&N32,&NAS,&NCPU 00070000
  LCLA &MIX 00090000
  TEXT 00100000
* R-08,C-00 CHANGE LEVEL 00110000
  AIF (&NOB).C0500 00120000
  TITLE '$E085/CMBSCH---RESCHEDULE-A-BSCA-LINE' 00130000
***** 00140000
* * 00150000
* NAME--CMBSCH, RESCHEDULE A BSCA LINE. * 00160000
* * 00170000
* FUNCTION--TO SCHEDULE WORK ON AN INACTIVE LINE. * 00200000
* * 00230000
* OPERATION-- * 00240000
* * 00310000
* . CLEAR ANY BSCA OP ENDS LEFT IN THE LCB, AND ADJUST THE * 00320000
* OP END TOTAL COUNT (#OPEND). * 00330000
* * 00340000
* . IF QUEUE FOR LINE IS EMPTY THEN POST THE REQUESTOR IF * 00350000
* TP WAS JUST SCHEDULED, CHECK FOR OTHER FUNCTION THAT * 00360000
* CAN BE PERFORMED BY 'CM'. * 00370000
* * 00380000
* . IF A BSCA POLL FOR STATUS OPERATION IN THE QUEUE HANDLE * 00390000
* IT BEFORE ANY OTHER OPERATION. * 00400000
* * 00410000
* . IF A PUT OPERATION CAN BE STARTED, PERFORM IT NEXT. * 00420000
* * 00430000
* . IF NEITHER OF ABOVE TWO, THEN TRY TO SCHEDULE AN INPUT * 00440000
* OPERATION. IF NO INPUT TO SCHEDULE THEN POST THE * 00450000
* REQUESTOR IF TP WAS JUST SCHEDULED. THEN CHECK FOR * 00460000
* OTHER 'CM' FUNCTION THAT CAN BE PERFORMED NOW. * 00470000
* * 00480000
* . IF A WRITE OPERATION CAN BE STARTED, THEN SET UP THE * 00490000
* OUTPUT DATA BUFFER FOR THE LINE. TRANSLATE THE DATA * 00500000
* AS REQUIRED OR SPECIFIED. ADD DEVICE DEPENDENT * 00510000
* CONTROL CHARACTERS. * 00520000
* * 00530000
* . IF NEW OPERATION ON A BSCA CARVE UP THE IOB(S) AND LINE * 00540000
* BUFFER(S) AS NECESSARY TO PERFORM THE OPERATION. SET * 00550000
* UP THE LINE DEPENDENT SECTION OF THE DTF. (POLLING/ * 00560000
* ADDRESSING CHARACTERS, SWITCH ID VERIFICATION IDS, * 00570000
* SWITCH LINE CALL/ANSWER OPTIONS). * 00580000
* * 00590000
* . IF BSCA OPERATION SET UP TO DO THE GET OR PUT-NORMAL, * 00600000
* PUT-BLOCK, PUT-END OF FILE, OR PUT EOT-TO-WACK * 00610000
* OPERATION AS APPROPRIATE. * 00620000
* * 00630000
* . ISSUE OF IOCS CALL AFTER THE DTF IS SET-UP. FOR BSCA, * 00640000
* CALL $$BSMS. * 00650000
* * 00660000
* . AFTER ISSUING THE IOCS CALL, TRACE THE RESULTS. * 00670000
* * 00690000
* . IF BSCA OPERATION IS COMPLETED WITHOUT AN OP END * 00700000
* INTERRUPT THEN FAKE AN OP END INTERRUPT TO KEEP THE * 00710000
```

```

*          FUNCTION GOING TIL COMPLETION.  POST THE REQUESTOR IF * 00720000
*          TP WAS JUST SCHEDULED.  CHECK FOR MORE WORK TO DO AT * 00730000
*          THIS TIME. * 00740000
* * 00750000
* ENTRY POINTS * 01000000
*   CMBSCH - SCHEDULE WORK ON A BSCA LINE * 01010000
*   CMNOBY - SCHEDULE FOLLOWING REQUEST BY OWNER OF LINE(CMBREQ) * 01020000
*   CMFORB - FORM OP CODE FOR MLMP FOR INTERNAL REQUEST. * 01030000
* * 01040000
* INPUT-- * 01050000
*   CMSDTF - ADDRESS OF DTF TO BE SCHEDULED. * 01060000
*   CMSPL - ADDRESS OF TP PARAMETER LIST TO BE SCHEDULED. * 01070000
*   #OPEND - OP END PENDING COUNT. * 01090000
* * 01110000
* OUTPUT-- * 01120000
*   CMSDTF - ADDRESS OF DTF FOR LINE SCHEDULED. * 01130000
*   CMSPL - ADDRESS OF TP REQUEST SCHEDULED. * 01140000
*   DTF(LCB),TUB - SET UP FOR THE OPERATION PERFORMED. * 01150000
* * 01160000
* EXTERNAL REFERENCES-- * 01170000
*   $CC4B0 - FORMAT 3270 COMMAND OUTPUT. * 01180000
*   $CC4JD - TRANSLATE FROM EBCDIC TO ASCII. * 01190000
*   $CC4WR - HANDLE TRANSLATE ERRORS IN OUTPUT. * 01200000
*   $CC4WC - SWITCH LINE CALL/ANSWER LOG TRANSIENT. * 01210000
*   $TRACE - CCP TRACE ROUTINE. * 01220000
*   CMSRPL - SEARCH LINE QUEUE FOR PL TO SCHEDULE ON LINE. * 01230000
*   CMTASV - SAVE TERMINAL ATTRIBUTES. * 01240000
*   CMBTAS - SETUP DTF AND IOB. * 01250000
*   CMGINL - SET UP INPUT RECORD LENGTH. * 01260000
*   CMPSCH - FIND SWITCH ID ENTRY IN SWITCH ID LIST. * 01270000
*   CMASCH - FIND ADDRESSING ENTRY IN ADDRESSING LIST. * 01280000
* * 01290000
* EXIT, NORMAL-- * 01300000
*   - TO CMFRMN IF FREEMAIN POSTED OR TP REQUEST TO BE * 01310000
*   HANDLED. * 01320000
*   - TO CMPOND IF AN OP END TO BE HANDLED. * 01330000
* * 01340000
* CHANGE ACTIVITY * 01342000
*   RELEASE 7 * 01344000
*   @01 INCR/ES0705 BUSY PRINTER SUPPORT * 01346000
* * 01348000
***** * 01350000
SPACE * 01360000
CMBSCH EQU * RESCHEDULE THE LINE B 01370000
L CMSDTF,DTF POINT XR2 AT DTF B 01380000
SPACE * 01390000
***** B 01400000
* LINE INACTIVE AFTER OP END - * B 01410000
* MAKE SURE OP END COUNTS AND OWNERSHIP FLAG ARE ZERO * B 01420000
***** B 01430000
SPACE 1 * 01440000
AIF (&NSWL).S0050 * 01450000
&MIX SETA &NPP+&NMP+&NCS * 01460000
AIF (&MIX EQ '3').S0060 * 01470000
TBN $BDATR(,DTF),$BCSWI SWITCHED SLB 01480000
TBF $BDATR(,DTF),$BCMPT * LINE ? SLB 01490000
JT CMROPE YES-DON'T CLEAR 'LCBOWN'. SLB 01500000
.S0050 ANOP * 01510000
SLC LCBOWN(2,DTF),LCBOWN(,DTF) CLEAR LINE OWNERSHIP STATUS. B 01520000
.S0060 ANOP * 01530000

```


CMROPE EQU *	* LOCAL	B 01540000
SLC #OPEND,LCBOPE(1,DTF)	REMOVE LINE OP END RESIDUAL.	B 01550000
MVI LCBOPE(,DTF),NOBIT	CLEAR LINE OP END COUNT.	B 01560000
SPACE		01570000
* FREE INPUT HOLD BUFFER NOW IN CASE A PUT NEEDS THE CORE		B 01571400
SPACE 1		B 01572100
CLI LCBIBA-1(,DTF),NOBIT	ANY TO FREE?	B 01572800
JE CMNROP	NO-SKIP FREE MAIN CALL	B 01573500
LA 0(,DTF),XR1	XR1-->DTF	B 01574200
L LCBIBA(,DTF),XR2	XR2-->AREA TO BE FREED	B 01574900
MVI LCBIBA-1(,XR1),NOBIT	ZERO OUT IBA	B 01575600
B CMFMRT	GO FREE THE AREA	B 01576300
LA 0(,XR1),XR2	RESTORE DTF POINTER	B 01577000
CMNROP EQU *	*	B 01577700
SBN LCBATR(,DTF),LCBTIM	SET DEFAULT TO RESCHED NEEDED	B 01590000
TBF CMSWIT,CMFMPS	FREEMAIN POSTED	B 01600000
CLI LCBNW#(,DTF),NOBIT	* OR NEW REQT TO HANDLE ?	B 01610000
BC CMPAII,FLSNEQ	YES - HANDLE WHILE LINE INACT.	B 01620000
SPACE		01650000
*****		B 01660000
* NEW REQUEST SCHEDULE --	*	B 01670000
* SEARCH PARAMETER LIST QUEUE FOR PUT REQUEST TO SCHEDULE OR	*	B 01680000
* IF NO PUT, ALL READS THAT CAN BE SCHEDULED.	*	B 01690000
*****		B 01700000
SPACE		01710000
B CMSRPL	LCB PL QUEUE SEARCH ROUTINE	B 01720000
*	XR1 RETURNS PL ADDRESS	B 01730000
SPACE		01740000
* XR1 CONTAINS ADDRESS OF PUT PARAMETER LIST TO BE SCHEDULED OR THE		B 01750000
* LAST READ PL THAT A BUFFER COULD BE OBTAINED FOR.		B 01760000
SPACE		01770000
B CMTASV	NOW GO SET UP TAS SAVE AREAS.	B 01780000
SPACE		01790000
CMNOBY EQU *	*	B 01800000
L CMSDTF,DTF	POINT XR2 AT DTF	B 01810000
ST LCBPL@(,DTF),PL	SAVE THE PARM LIST IN LCB.	B 01820000
SPACE 1		01830000
*****		B 01840000
* DETERMINE IF OPERATION IS A READ OR WRITE		B 01850000
*****		B 01860000
SPACE 1		01870000
TBN PL\$OPM(,PL),OPGET	IS IT READ	B 01880000
JT CMFORB	JUMP IF TRUE TO READ ROUTINE	B 01890000
L PLTUBA(,PL),XR2	XR2----> TUB	@01 01890500
*		@01 01891000
* BUSY PRINTER SUPPORT CODE		@01 01891500
*		@01 01892000
AIF (&NPBY).NBY02	BUSY PRINT SUPPORTED?	@01 01892500
* IF NON DFF TERMINAL, SKIP THIS SECTION OF BUSY PRINT CODE		01892600
TBN TUBTA1(,XR2),TASDFF	IS THIS A DFF TERMINAL?	@01 01892700
JF CMNBY2	IF NO, SKIP BUSY PRINT CODE	@01 01892800
CLI TUBPHY(,XR2),TUB5M2	TEST FOR 3735	@01 01893000
JH CMNBY2	IF YES, SKIP BUSY PRINT CODE	@01 01893500
TBN TUBSCS(,XR2),TUBBPT	IS BUSY PRINT ALLOWED	@01 01893600
JF CMNBY2	NO, SKIP BUSY PRINT CODE	@01 01893700
CLC PLOUTL(2,PL),X\$0002	IS PUT LENGTH GR THN 2	@01 01894000
JNH CMNBY2	NO, SKIP PRINTER BSY STUFF	@01 01894500
TBF PLOPM(,PL),OPREQR	USER OP?	@01 01895000
TBN PLOPC(,PL),OPPUT	AND PUT?	@01 01895500
L PLRECA(,PL),XR2	FIND RECORD AREA	@01 01896000

```

TBN WCC(,XR2),STPRT IS START PRINT BIT ON? @01 01896500
L PLTUBA(,PL),XR2 XR2----> TUB @01 01897000
JF CMNBY2 NOT ALL OF THE ABOVE - JUMP @01 01897500
SBN TUBAT4(,XR2),TUBBSY SET ON PRINTER BUSY BIT @01 01898000
CMNBY2 EQU * @01 01898500
.NBY02 ANOP @01 01899000
L TUBDTF(,XR2),DTF RESTORE DTF REGISTER @01 01899500
TITLE '$E085/CMBSCH----START-A-WRITE-OPERATION' 01900000
***** B 01910000
* START A WRITE OPERATION ON THE BSCA LINE * B 01920000
***** B 01930000
SPACE 01940000
AIF (&N32).T0810 02000000
CLI CMSPHY,TUB5M2 THIS A 3270 ? 0B 02010000
TBN PL$OPC(,PL),OPUSER SYSTEM FUNCTION ? 0B 02020000
JC CMWXLT,FLSOHI NO-GO CHECK FOR TRANSLATION. 0B 02030000
AIF (&NSWL).T0805 02030800
AIF (&NCS).T0800 02031600
TBF $BDATR(,DTF),$BCMPT NOT SWITCHED LINE ? B 02032400
JF CMB0C2 YES - CALL CONTROL LINE FORMAT B 02033200
.T0800 ANOP 02034000
SVC 0 ##### TRANSIENT CALL ##### 0B 02034800
DC AL1(CCPRIB) CCP SVC RIB 0B 02035600
DC AL1(CC4S0) BRING IN 3275 FORMAT XIENT. 0B 02036400
* -----START-----@23 02036800
J CMREFH GO CHECK FOR TRANSLATION. 0B 02037200
* -----END-----@23 02037600
.T0805 ANOP 02038000
CMB0C2 EQU * * LOCAL B 02038800
SVC 0 ##### TRANSIENT CALL ##### 0B 02040000
DC AL1(CCPRIB) CCP SVC RIB 0B 02050000
DC AL1(CC4B0) BRING IN 3270 FORMAT XIENT. 0B 02060000
* -----START-----@23 02061000
CMREFH EQU * * LOCAL 0B 02062000
TBN PL$OPC(,PL),OPRFSH REFRESH OPERATION ? 0B 02063000
JT CMWXLT YES-DON'T UPDATE LENGTH 0B 02064000
SLC LCBADJ(2,DTF),LCBSRT(,DTF) DETERMINE LENGTH OF OUTPUT 0B 02065000
* -----END-----@23 02066000
SPACE 02070000
CMWXLT EQU * * LOCAL 0B 02080000
.T0810 ANOP 02090000
AIF (&NAS).A0100 02100000
TBN $BDATT(,DTF),$BCASK IS THIS AN ASCII LINE, AND AB 02110000
TBF SAVTA1,TASTRN TAS SPECIFY TRANSLATE ? AB 02120000
JF CMFORB NO--THEN JUST DO WRITE. AB 02130000
SPACE 02140000
AIF (&N32).T0900 02150000
* IF SYSTEM OR REFRESH OPTION TO A 3270, THEN THE DATA IS IN THE 0AB 02160000
* RESERVED PORTION OF THE BSCA LINE BUFFER. 0AB 02170000
SPACE 02180000
CLI CMSPHY,TUB5M2 THIS A 3270, AND 0AB 02200000
TBN PL$OPC(,PL),OPUSER SYSTEM FUNCTION ? 0AB 02210000
JC CMWREG,FLSOHI NO-GO USE REGULAR FIELDS. 0AB 02220000
MVC #CMTRL+TLFRMA,LCBSRT(2,DTF) FM MSG IS IN RESERVED BUFR. 0AB 02230000
MVC #CMTRL+TLTOL,LCBADJ(2,DTF) TRU LENGTH IS IN THE LCB. 0AB 02240000
J CMWTBL GO FILL STANDARD TABLE DATA. 0AB 02250000
SPACE 02260000
.T0900 ANOP 02270000
AIF (&NAS).A0100 02280000
CMWREG EQU * * LOCAL AB 02290000

```

```

MVC #CMTRL+TLTOL, PLOUTL(2, PL) * AT START-UP TO TRANSLATE INTAB 02300000
MVC #CMTRL+TLFRMA, PLRECA(2, PL) BUILD TRANSLATE PARM LIST AB 02310000
AIF (&NCPU).T0840 02315000
TBN PLOPM(, PL), OPOLT ON LINE TEST REQUEST ? AUB 02320000
JF CMWTBL NO-SET UP STANDARD FIELDS. AUB 02330000
ALC #CMTRL+TLFRMA(2), OLTNG BUMP PAST OLT PL TO TEXT. AUB 02340000
.T0840 ANOP 02345000
CMWTBL EQU * * LOCAL AB 02350000
MVC #CMTRL+TLTOA, LCBATL(2, DTF) *USE SPECIAL BLOCK SET ASIDE AB 02360000
LA #CMTRL, XR1 LOAD REG TO POINT TO PARM LISTAB 02370000
SVC 0 ##### TRANSIENT CALL ##### AB 02380000
DC AL1(CCPRIB) CCP SVC RIB AB 02390000
DC AL1(CC4JD) TRANSLATE TRANSIENT AB 02400000
SPACE 02410000
TBN TLRTC(, XR1), TLERR TRANSLATE ERROR ? AB 02420000
L CMSPL, PL POINT TO THE PARM LIST. AB 02430000
JF CMFORB NO-GO GIVE OP TO MLMP. AB 02440000
SPACE 1 02450000
SVC 0 ##### TRANSIENT CALL ##### AB 02460000
DC AL1(CCPRIB) CCP SVC RIB AB 02470000
DC AL1(CC4WR) REQUEST XLATE ERROR RTN. AB 02480000
SPACE 1 02490000
TBN LCBAT2(, DTF), LCBACT LINE ACTIVE ? (BETWEEN EOT) AB 02493000
JF CMWPII NO-JUST POST IF NO WAIT OP. AB 02496000
SBN LCBAT1(, DTF), LCBNTQ SET PARM LIST NOT QUEUED. AB 02500000
CMWPII EQU * * AB 02505000
B CMPAII EXIT HANDLING TP REQUEST. AB 02510000
SPACE 1 02520000
.A0100 ANOP 02530000
TITLE '$E085/CMBSCH---FORM MLMP REQUEST FOR READ OR WRITE' 02540000
***** B 02550000
* FORM BSCA TP REQUEST FOR MLMP -- READ OR WRITE * B 02560000
***** B 02570000
SPACE 1 02580000
CMFORB EQU * * B 02590000
SPACE 02600000
* IF LINE IS ACTIVE GO FORM NEXT OPERATION - DTF ALREADY SET UP. B 02610000
SPACE 02620000
TBN LCBAT2(, DTF), LCBACT LINE ACTIVE? (BETWEEN EOT) B 02630000
BT CMFVIFY YES - GO SET UP NEXT OP. B 02640000
SPACE 1 02650000
*----- B 02660000
* LINE NOT ACTIVE * B 02670000
*----- B 02680000
SPACE 1 02690000
AIF (&MIN).N0070 02700000
.* THIS CODE WILL BE IN TRANSIENT $CC4B2 FOR MIN RES SYSTEM RB 02710000
SBF $BDATT(, DTF), $BCCNV+$BCGET SET OFF INPUT/OUTPUT IND'S. RB 02720000
SBF LCBOPC(, DTF), LCBERP RESET BSCA TERM ERP IND. RB 02730000
AIF (&NITB).I0400 02740000
MVI $BDITB(, DTF), NOBIT ZERO ITB COUNT BYTE RIB 02750000
.I0400 ANOP 02760000
MNN LCBOPC(, DTF), PL$OPM(, PL) SAVE OP FOR CNFLICT CHECK. RB 02770000
TBN PL$OPM(, PL), OPGET THIS A GET OPERATION ? RB 02780000
L PLTUBA(, PL), TUB POINT TO THE TUB. RB 02790000
JT CMFGET YES-GO SET UP DTF FOR GET OP. RB 02800000
SPACE 02810000
***** RB 02820000
* SET UP DTF FOR ** PUT ** OPERATION. * RB 02830000
***** RB 02840000

```

```

SPACE                                02850000
SBF  LCBAT2(,DTF),LCBPUT              SET OFF PUT PENDING IND.    RB 02860000
SBN  $BDATT(,DTF),$BCOUT              SET OUTPUT FILE INDICAOR.  RB 02870000
MVI  $BDOPC(,DTF),$BOPUT             SET OP CODE FOR PUT.       RB 02880000
SPACE                                02890000
*   SET CURRENT ATTRIBUTTES, BLOCK LENGTH, AND OWNERSHIP STATUS.  RB 02900000
SPACE                                02910000
B    CMBTAS                           GO SET UP DTF/IOB/TUB/LCB.  RB 02920000
SPACE                                02930000
SBN  $BDAT1(,DTF),$BCPUT              SET PUT SPAN FILE INDICATOR. RB 02940000
MVC  $BDBKL(2,DTF),LCBKLC(,DTF)      FILL IN DTF BLOCK LENGTH.  RB 02950000
AIF  (&NITB).I0600                   02960000
TBN  $BDATT(,DTF),$BCITB             ITB MODE ?                  RIB 02970000
JF   CMFLIN                           NO-GO DO COMMON LINE SETUP. RIB 02980000
MVC  $BDITB(1,DTF),TUBBKF(,TUB)      MOVE BLK FACTOR TO WRK AREA. RIB 02990000
SLC  $BDITB(1,DTF),X$0001            FIND NUMBER OF ITB CHARS.   RIB 03000000
ALC  LCBKLC(2,DTF),$BDITB(,DTF)      ADD # ITB CHAR TO BUF       RIB 03010000
AIF  (&NTSP).I0500                   03020000
TBN  $BDATT(,DTF),$BCRAN             TRANSPARENCY MODE ?         RIXB 03030000
JF   CMFTB1                           NO-GO SET ITB LENGTH TO ONE. RIXB 03040000
ALC  $BDITB(2,DTF),$BDITB(,DTF)      DOUBLE THE ITB COUNT TO ADDRIB 03050000
ALC  LCBKLC(2,DTF),$BDITB(,DTF) * 4  TIMES BLOCK FACTOR FOR RIXB 03060000
ALC  LCBKLC(2,DTF),$BDITB(,DTF) * TOTAL OF 5 TIMES BLK FAC RIXR 03070000
ALC  LCBKLC(2,DTF),X$0002            THEN ADD 2 FOR PUT-ITB-TRP  RIXB 03080000
MVC  $BDITB(2,DTF),FIVE              SET ITB LENGTH TO FIVE.     RIXB 03090000
J    CMFLIN                           GO SET UP LINE DEPENDENCIES. RIXB 03100000
SPACE                                03110000
CMFTB1 EQU *                          * LOCAL                      RIXB 03120000
.I0500 ANOP                            03130000
MVI  $BDITB(,DTF),1                  SET ITB LENGTH COUNT TO ONE. RIB 03140000
.I0600 ANOP                            03150000
J    CMFLIN                           GO DETERMINE LINE SETUP NEEDS. RB 03160000
EJECT                                  03170000
***** RB 03180000
*   SET UP DTF FOR ** GET ** OPERATION. RB 03190000
***** RB 03200000
SPACE                                03210000
CMFGET EQU *                          * LOCAL                      RB 03220000
SBN  $BDATT(,DTF),$BCINP+$BCGET      SET INPUT FILE INDICATORS.  RB 03230000
MVI  $BDOPC(,DTF),$BOGET             SET OP CODE FOR GET.       RB 03240000
MVC  $BDBKL(2,DTF),LCBBFL(,DTF)      USE MAXIMUM LINE BLOCK LEN. RB 03250000
MVC  LCBKLC(2,DTF),$BDBKL(,DTF)      FILL IN CURRENT MAX BLOCK LENRB 03260000
SBN  LCBAT2(,DTF),LCBRCI             INDICATE RECEIVE INITIAL.   RB 03270000
SPACE 2                               03280000
AGO  .N0075                           03290000
.N0070 MVI CMID2,CTFORB              MOVE IN ID OF CODE DESIRED MIN B 03300000
B    CMCAL2                           CALL $CC4B2 TO CHECK LINE MIN B 03310000
*                                       OPEN AND SPECIAL CASES     MIN B 03320000
.N0075 ANOP                            03330000
***** B 03340000
*   COMMON GET AND PUT DTF SETUP      B 03350000
***** B 03360000
SPACE                                03370000
CMFLIN EQU *                          * LOCAL                      B 03380000
AIF  (&NCS).S1000                    03390000
&MIX SETA &NPP+&NMP+&NSWL            03400000
AIF  (&MIX EQ '3').S0900             03410000
TBN  $BDATR(,DTF),$BCMCN            CONTROL STATION LINE ?     CLB 03420000
JF   CMFLCS                           NO-GO TO END OF CS CODE.   CLB 03430000
.S0900 ANOP                            03440000

```

```

SPACE 03450000
***** CB 03460000
* SET CONTROL STATION ONLY DTF FIELDS * CB 03470000
***** CB 03480000
SPACE 03490000
MVC $BDLST(2,DTF),LCBPOL(,DTF) MOVE IN POLLING LIST ADDR. CB 03500000
MVC $BDIND(1,DTF),LCBLID(,DTF) FILL DESIRED ID IN THE LIST CB 03512000
MVC $BDLID(1,DTF),LCBLID(,DTF) FILL DESIRED ID IN THE LIST CB 03514000
TBN $BDOPC(,DTF),$BOGET GET 'POLL' REQUEST ? CB 03520000
JT CMFIGR YES-GO FIGURE BUFFER CONFIG. CB 03530000
MVC LCBID#(1,DTF),TUBSID(,TUB) FILL ID OF DESIRED TERM. CB 03550000
LA LCBADL(,DTF),XR1 FILL IN SELECTION 'ADDRESSING' CB 03560000
ST $BDLST(,DTF),XR1 * LIST ADDRESS. CB 03570000
B CMASCH GO TO FIND THE SEL'T ENTRY. CB 03580000
SPACE 03590000
MVC LCBADL+8(9,DTF),8(,POL) MOVE IN MAX LEN SEL'T ENTRY. CB 03600000
MVI CMFLA+2,LCBADL+3 REFRESH INSTRUCTION DISPL. CB 03610000
ALC CMFLA+2,POLCNT(1,POL) SET TO END OF SELECT LIST. CB 03620000
CMFLA EQU * * MODIFICATION CB 03630000
MVI #(,DTF),ONETIM END OF LIST: OPEN LIST IND. CB 03640000
AIF (&MIX EQ '3').S1000 CB 03650000
J CMFIGR GO TO FIGURE BUFFER CONFIG. CLB 03660000
SPACE 03670000
CMFLCS EQU * * LOCAL CB 03680000
.S1000 ANOP CB 03690000
AIF (&NSWL).S1200 CB 03700000
&MIX SETA &NPP+&NMP+&NCS CB 03710000
AIF (&MIX EQ '3').S1100 CB 03720000
.TS1100 TBN TUBAT1(,TUB),TUBSWC SWITCHED LINE ? SLB 03730000
ANOP CB 03740000
TBF LCBAT3(,DTF),LCBENB AND LINE NOT ENABLED ? SB 03750000
JF CMFLSW NO-SKIP SWITCH LINE CODE. SB 03760000
SPACE 03770000
***** SB 03780000
* SET SWITCHED LINE ONLY DTF FIELDS * SB 03790000
***** SB 03800000
SPACE 03810000
MVC LCBID#(1,DTF),TUBSID(,TUB) USE TUB BSCA ID FOR SEARCH. SB 03820000
SBF $BDATR(,DTF),$BCMAN+$BCANS SET OFF SWL LINE TYPES. SB 03830000
TBN SAVTA1,TASCNC SWITCHED CALL ? SB 03840000
JT CMFSMN YES-GO CHECK FOR MANUAL OPERATS SB 03850000
SBN $BDATR(,DTF),$BCANS SET DTF FOR ANSWER. SB 03860000
CMFSMN EQU * * LOCAL SB 03870000
TBN SAVTA1,TASAUT AUTO OPERATION ? SB 03880000
JT CMFSRQ YES-GO CHECK REQUESTOR. SB 03890000
CMFSRQ SBN $BDATR(,DTF),$BCMAN SET MANUAL OPERATION IN DTF. SB 03900000
EQU * * LOCAL SB 03910000
L LCBPL@(,DTF),PL POINT TO THE PARM LIST. SB 03920000
TBN PLOPM(,PL),OP$SYS SYSTEM REQUEST ? SB 03930000
JT CMFSNB YES-GO SET ENABLED STATUS. SB 03940000
SPACE 03950000
* ISSUE PHONE CONNECT MESSAGE TO CONSOLE OPERATOR SB 03960000
SPACE 03970000
SVC 0 ##### TRANSIENT CALL ##### SB 03980000
DC AL1(CCPRIB) CCP SVC RIB SB 03990000
DC AL1(CC4WC) BRING IN SWITCH MESSAGE RTN. SB 04000000
SPACE 1 04010000
CMFSNB EQU * * LOCAL 04020000
SBN LCBAT3(,DTF),LCBENB SET LINE ENABLED. SB 04030000
MVC $BDRID(2,DTF),LCBPOL(,DTF) FILL IN SWITCH LIST ADDRESS. SB 04040000

```

```

SBN $BDADD(,DTF), $BCSWD SET SWITCH LIST USED IND. SB 04050000
MVI $BDRLN(,DTF), POLACT SET ID TO USE ACTIVE ENTRIES. SB 04060000
TBN $BDATR(,DTF), $BCANS ANSWER LINE, AND SB 04070000
TBN LCBOPC(,DTF), OPGET * GET OPERATION ? SB 04080000
JT CMFIGR YES-GO HANDLE ANSWER-VERIFY IDSB 04090000
SPACE 1 04100000
CMFSCLEQU * * LOCAL SB 04110000
SBF $BDADD(,DTF), $BCSWD SET OFF SWITCH LIST USED. SB 04120000
TBF SAVTA2, TASVFY RECEIVE ID VERIFY ? SB 04130000
JF CMFSNV NO-GO SET FOR NO VERIFY. SB 04140000
B CMPSCH CALL SWITCH LIST SEARCH RTN. SB 04150000
SPACE 1 04160000
* CONTROL IS RETURNED TO NSI IF THE ID IS FOUND IN THE LIST. SB 04170000
SPACE 1 04180000
J CMFSID GO SET UP ID IN THE DTF. SB 04190000
SPACE 1 04200000
* CONTROL RETURNS TO NSI+3 IF IF THE ID WAS NOT FOUND. SB 04210000
SPACE 1 04220000
CMFSNVEQU * * LOCAL SB 04230000
MVI $BDRLN(,DTF), NOBIT SET ZERO LENGTH ID TO RECEIVE. SB 04240000
J CMFIGR GO TO FIGURE THE IOB'S NEEDED. SB 04250000
SPACE 1 04260000
CMFSIDEQU * * LOCAL 04270000
MVC $BDRLN(1,DTF), POLCNT(,POL) PUT RECEIVE ID LENGTH IN DTF. SB 04280000
LA POLCH1(,POL), POL UPDATE REG TO FIRST ID CHAR. SB 04290000
ST $BDRID(,DTF), POL STORE ADDRESS OF ID IN DTF. SB 04300000
AIF (&NMP).S1110 04310000
J CMFIGR GO TO FIGURE THE IOB'S NEEDED. SB 04320000
SPACE 1 04330000
.S1110 ANOP 04340000
CMFLSWEQU * * LOCAL SB 04350000
.S1200 ANOP 04360000
AIF (&NMP).S1130 04370000
&MIX SETA &NPP+&NSWL+&NCS 04380000
AIF (&MIX EQ '3').S1120 04390000
TBN $BDATR(,DTF), $BCMPT MULTI-POINT TRIBUTARY ? TLB 04400000
JF CMFIGR NO-SKIP MP TRIB CODE. TLB 04410000
SPACE 1 04420000
* MULTI-POINT TRIBUTARY SUPPORT. TLB 04430000
SPACE 1 04440000
.S1120 ANOP 04450000
SBF $BDPSC-1(,DTF), POLBIT SET BSCA CHARACTERS FOR POLL. TB 04460000
SBF $BDPSC(,DTF), POLBIT ( I SEND - PUT, ETC ) TB 04470000
TBN $BDOPC(,DTF), $BOGET GETTING ? TB 04480000
JF CMFIGR NO-I'AM PUTTING. TB 04490000
SBN $BDPSC-1(,DTF), POLBIT SET ON FOR BEING ADDRESSED, TB 04500000
SBN $BDPSC(,DTF), POLBIT ( I RECEVIE - GET, ETC ) TB 04510000
.S1130 ANOP 04520000
SPACE 04530000
***** 04540000
* FIGURE OUT THE SPACE REQUIREMENTS FOR IOBS AND BUFFERS. B 04550000
***** 04560000
SPACE 04570000
CMFIGREQU * * B 04580000
MVC $BDIOB(2,DTF), LCBSTRT(,DTF) RESTORE BUFFER START ADDRESS. B 04590000
CLI $BDWKA-1(,DTF), NOBIT WORK AREA ADDRESS SET YET ? B 04591300
JE CMNWKA NO - DON'T SET RETRY COUNT B 04591600
L $BDWKA(,DTF), WKA XR1-->WORK AREA B 04592000
MVC WKERRD(1,WKA), $BDERR(,DTF) RESTORE ERROR RETRY COUNT B 04593000
* --START-----@32 04593500

```

```

      SLC      DCOUNT(2,WKA),DCOUNT(,WKA) ZERO DELAY COUNT.           B 04594000
*          ---END-----@32      04594200
CMNWKA EQU    *          * LOCAL                                       B 04594500
      AIF      (&N32).T1200                                           04600000
      L        LCBPL@(,DTF),PL          RELOAD THE PL REG.           0B 04610000
      SPACE                                         04620000
* IF SYSTEM OR REFRESH OF 3270 SYSTEM, THEN RESERVE THE FIRST PART 0B 04630000
* OF THE LINE BUFFER FOR THE OUTPUT.           0B 04640000
      SPACE                                         04650000
      CLI      CMSPHY,TUB5M2          THIS A 3270 TERMINAL ?         0B 04660000
      TBN      PL$OPM(,PL),OPPUT      PUT OPERATION ?              0B 04670000
      TBN      PL$OPC(,PL),OPUSER     SYSTEM FUNCTION ?           0B 04680000
      JC        CMFIGL,FLSOHI         NO-GO HANDLE REGULARILY.       0B 04690000
      ALC      $BDIOB(2,DTF),MAXMSG   RESERVE LINE BUFFER SPACE.   0B 04700000
      MVC      LCBKLC(2,DTF),LCBADJ(,DTF) USE ADJUSTED LEN FOR BLK LEN. B 04710000
CMFIGL EQU    *          * LOCAL                                       0B 04720000
      .T1200 ANOP                                           04730000
      AIF      (&MIN).N0080                                           04740000
      .* THIS FOLLOWING CODE WILL BE IN TRANSIENT $CC4B2           04750000
      L        LCBKLC(,DTF),WORK      BLOCK SIZE REQUIRED IN WORK    RB 04760000
      LA       LINFO(,WORK),WORK      ALLOW FOR MAX LINE CONTROL    RB 04770000
      ST       LCBWRK(,DTF),WORK      SAVE COMPUTED VALUE.         RB 04780000
      SPACE                                         04790000
***** RB 04800000
* INITIALIZE IOB * RB 04810000
***** RB 04820000
      SPACE                                         04830000
      L        $BDIOB(,DTF),IOB       LOAD PTR TO IOB.              RB 04840000
      MVC      IOBQ(1,IOB),$BDDEV(,DTF) BUILD IOB Q CODE.           RB 04850000
      MVC      IOBDBL(2,IOB),LCBWRK(,DTF) PUT BUFFER LENGTH INTO IOB. RB 04860000
      ST       IOBNXT(,IOB),IOB       POINT 1ST IOB TO SELF.       RB 04870000
      ST       IOBDTF(,IOB),DTF      POINT IOB BACK TO DTF.       RB 04880000
      MVI      IOBERR(,IOB),X'00'     SET ERROR COUNT TO ZERO.   RB 04890000
      ST       IOBDAT(,IOB),IOB       SET @ OF IOB                 RB 04900000
      ALC      IOBDAT(2,IOB),IOBLEN    *          DATA AREA.       RB 04910000
      MVC      IOBFLG(1,IOB),$BDATT(,DTF) SET IOB FLAG BYTE.       RB 04920000
      MVI      IOBFLA(,IOB),TXTSNT    INITIALIZE TEXT INDICATORS.  RB 04930000
      MVI      IOBCMP(,IOB),DONE      INITIALIZE BUFFER STATUS.   RB 04940000
      TBN      $BDATT(,DTF),$BCINP    GET FILE ?                   RB 04950000
      JF       CMFMOR                  NO-GO CONTINUE CARVING.       RB 04960000
      MVI      IOBCMP(,IOB),READY     ELSE SET BUFFER STATUS TO READRB 04970000
      MVI      IOBFLA(,IOB),X'00'     ZERO TEXT DIRECTION INDICATOR.RB 04980000
CMFMOR EQU    *          * LOCAL                                       RB 04990000
      MVC      IOBNEX(2,IOB),IOBDAT(,IOB) DETERMINE START ADDRESS FOR RB 05000000
      ALC      IOBNEX(2,IOB),IOBDBL(,IOB) * ANOTHER IOB.           RB 05010000
CMFMUL EQU    *          * LOCAL                                       RB 05020000
      MVC      IOBDBN(2,IOB),IOBNEX(,IOB) DETERMINE ADDRESS OF NEXT  RB 05030000
      ALC      IOBDBN(2,IOB),IOBLEN    * DATA AREA.               RB 05040000
      MVC      IOB2NX(2,IOB),IOBDBN(,IOB) DETERMINE END ADDRESS FOR  RB 05050000
      ALC      IOB2NX(2,IOB),IOBDBL(,IOB) * POSSIBLE NEXT IOB/BUFFER. RB 05060000
      CLC      IOB2NX(2,IOB),LCBBND(,DTF) ANOTHER IOB/BUFFER FIT ?  RB 05070000
      JH       CMFCLN                  NO-GO CLEAN-UP THIS OPEN.   RB 05080000
      L        IOBNEX(,IOB),IBX       LOAD REG WITH IOB NEXT @.     RB 05090000
      MVC      IOBDTF(IOBDTF+1,IBX),IOBDTF(,IOB) COPY THE IOB.     RB 05100000
      ST       IOBNXT(,IOB),IBX       CHAIN NEXT IOB TO LAST.      RB 05110000
      MVC      IOBDAT(2,IBX),IOBDBN(,IOB) MOVE IN NEW DATA BUFFER @. RB 05120000
      MVC      IOBNEX(2,IBX),IOB2NX(,IOB) MOVE PTR TO NEXT IOB AREA. RB 05130000
      LA       0(,IBX),IOB           MAKE LAST IOB IN CHAIN CURRENTRB 05140000
      L        IOBDTF(,IOB),DTF      RESTORE DTF REGISTER.         RB 05150000
      B        CMFMUL                  GO TO MULTIPLE IOB LOGIC      RB 05160000

```

```

SPACE 05170000
* IOB AND BUFFER ALLOCATION DONE, PERFORM FINAL CLEAN-UP ACTIVITY. RB 05180000
SPACE 05190000
CMFCLN EQU * LOCAL RB 05200000
MVI $BDCMP(,DTF), $BCDNE MARK DTF DONE. RB 05210000
MVC $BDINT(2,DTF), LCB$LO(,DTF) RESTORE C/S @ OF $$BSL0. RB 05220000
MVI $BDNDX(,DTF), X'00' SET LINE INIT TRANS. ID TO RB 05230000
L IOBNXT(,IOB), IOB POINT TO 1ST IOB. RB 05240000
SBN IOBFLA(,IOB), FIRST SET FIRST BUFFER INDICATOR. RB 05250000
L LCBPL@(,DTF), PL RELOAD THE PARM LIST REG RB 05260000
SPACE 1 05270000
AGO .N0085 05280000
.N0080 ANOP 05290000
MVI CMID2, CTRLB MOVE IN FLAG OF CODE TO MIN B 05300000
* EXECUTE IN THE TRANSIENT MIN B 05310000
B CMCAL2 CALL IN TRANSIENT $CC4B2 MIN B 05320000
.N0085 ANOP 05330000
SPACE 4 05340000
***** B 05350000
* FINAL SETUP OF OP CODE AND RECORD LENGTH BEFORE IOS CALL B 05360000
***** B 05370000
SPACE 05380000
CMFVFY EQU * B 05390000
TBF LCBAT2(,DTF), LCBABT OPERATION AN ABORT, OR B 05400000
TBF LCBAT1(,DTF), LCBEOT SEARCH FOR EOT INDICATED ? B 05410000
JT CMFVUR NO-GO USE USER RECORD AREA. B 05420000
SPACE 1 05430000
*-----* B 05440000
* SEARCH FOR EOT-- SET TO READ 1 CHAR INTO DUMMY BUFFER * B 05450000
*-----* B 05460000
SPACE 1 05470000
MVC $BDREL(2,DTF), X$0001 SET RECORD LENGTH TO ONE. B 05480000
MVC $BDWKB(2,DTF), FNDEOT USE DUMMY DATA AREA TO FIND EOTB 05490000
* -----START-----@09 05491000
L $BDWKA(,DTF), WKA XR1-->WORK AREA 05492000
MVI WKERRD(,WKA), BIT7 SET RETRY COUNT TO 1 05493000
L LCBPL@(,DTF), PL XR1--> ACTIVE PARM LIST 05494000
* -----END-----@09 05495000
TBN LCBOPC(,DTF), OPPUT PUT OP CODE ? B 05500000
JT CMFVPT YES-MUST BE ABORT, GO SEND EOT. B 05510000
J CMBSCL GO TO CALL MLMP IOCS. B 05520000
SPACE 05530000
CMFVUR EQU * LOCAL B 05540000
MVC $BDWKB(2,DTF), PLRECA(,PL) FILL IN DATA AREA @ IN DTF. B 05550000
TBN $BDATT(,DTF), $BCINP GET OPERATION ? B 05560000
JF CMFVPT NO-GO SET UP FINAL PUT DTF. B 05570000
SPACE 1 05580000
*-----* B 05590000
* GET OPERATION * B 05600000
*-----* B 05610000
SPACE 1 05620000
AIF (&NINT).CT050 05630000
* START THE INTERVAL TIMER RUNNING NB 05640000
SPACE 1 05650000
LA TIMIOB, XR2 XR2-->TIMER IOB NB 05660000
MVI TIFLAG(,XR2), X'02' IND. 'TIME IS IN TIMER UNITS' NB 05670000
SVC 0 * NB 05680000
DC AL1(STMRIB) START THE TIMER NB 05690000
SPACE 1 05700000
L PLTUBA(,PL), XR2 XR2-->TUB NB 05710000

```


	L	TUBDTF(,XR2),DTF	XR2--->DTF	NB	05720000
.CT050	ANOP				05730000
	TBN	PLOPC(,PL),OPRVI	RVI	B	05740000
	TBF	PLOPC(,PL),OPORDR-OPRVI	SEND OP CODE?	B	05750000
	JF	CMFGIL	NO-GO GET INPUT LENGTH.	B	05760000
	SBN	LCBOPC(,DTF),LCBRVI	SET SEND RVI INDICATOR	B	05770000
CMFGIL	EQU	*	* LOCAL	B	05780000
&MIX	SETA	&NCS+&NSWL			05790000
	AIF	(&MIX EQ '2').T1250			05800000
	TBN	LCBAT2(,DTF),LCBRCI	RECEIVE INITIAL ON A	C/SB	05810000
	JT	CMBSIO	YES-GO TO BSCA IOCS CALL.	C/SB	05820000
.T1250	ANOP				05830000
	SPACE	1			05840000
*	IF NOT RECEIVE INITIAL,MLMP MAY MOVE DATA PRIOR TO \$\$\$BMCH CALL			B	05850000
*	,THEREFORE GETMAIN AND SET UP DTF TO BE READY.			B	05860000
	SPACE	1			05870000
	B	CMGINL	COMPUTE BUFFER LENGTH AND	B	05880000
*			* GETMAIN.	B	05890000
CMBSCL	EQU	*	*		05895000
	J	CMBSIO	GO TO BSCA IOCS CALL.	B	05900000
	SPACE	2			05910000
*	-----*			B	05920000
*	PUT OPERATION			B	05930000
*	-----*			B	05940000
	SPACE	1			05950000
CMFVPT	EQU	*	* LOCAL	B	05960000
	MVI	\$BDOPC(,DTF),\$BOPUT	SET OP CODE TO NORMAL PUT.	B	05970000
	MVI	\$BDCMP(,DTF),\$BCREQ	SET CMP TO OP REQUESTED.	B	05980000
	TBN	LCBAT2(,DTF),LCBSET	SEND EOT ORDERED ?	B	05990000
	JT	CMFVET	YES-GO SETUP SEND EOT.	B	06000000
	MVC	\$BDREL(2,DTF),PLOUTL(,PL)	PUT OUTL INTO DTF RECORD LEN.	B	06010000
	AIF	(&NCPU).T1270			06015000
	TBN	PLOPM(,PL),OPOLT	ON LINE TEST REQUEST ?	UB	06020000
	JF	CMFNLT	NO-JUMP PAST OLT CODE.	UB	06030000
	SPACE				06040000
*	ONLINE TEST REQUEST FROM CONSOLE TO BE SENT TO OTHER CPU.			UB	06050000
*	PLRECA POINTS TO A 12 BYTE MLMP OLT PARAMETER LIST FOLLOWED			UB	06060000
*	BY AN OPTIONAL USER SPECIFIED OLT MSG. PLOUTL TO TOTAL LGTH.			UB	06070000
	SPACE				06080000
	MVC	\$BDRFT(2,DTF),\$BDWKB(,DTF)	FILL IN MLMP OLT PARM LIST @.	UB	06090000
	ALC	\$BDWKB(2,DTF),OLTLNG	BUMP PAST OLT PARM LIST	UB	06100000
*			* TO OLT MESSAGE.	UB	06110000
	SLC	\$BDREL(2,DTF),OLTLNG	REDUCE LENGTH TO OLT LENGTH,	UB	06120000
*			*MESSAGE ONLY, EXCLUDE OLT PL.	UB	06130000
CMFNLT	EQU	*	* LOCAL	UB	06140000
.T1270	ANOP				06145000
	SPACE				06150000
	AIF	(&N32).T1300			06160000
*	IF SYSTEM REQUEST OR REFRESH OPERATION, THEN USE RESERVED AREA IN0B			0B	06170000
*	THE LINE BUFFER.			0B	06180000
	SPACE				06190000
	CLI	CMSPHY,TUB5M2	THIS A 3270 ?	0B	06200000
	TBN	PL\$OPC(,PL),OPUSER	SYSTEM FUNCTION ?	0B	06210000
	JC	CMFVMD,FLSOHI	NO-GO VERIFY THE MODE.	0B	06220000
	MVC	\$BDWKB(2,DTF),LCBSRT(,DTF)	SET CORRECT ADDRESS FOR OUTPU0B	0B	06230000
	MVC	\$BDREL(2,DTF),LCBADJ(,DTF)	USE CORRECT LENGTH OF THE DAT0B	0B	06240000
CMFVMD	EQU	*	* LOCAL	0B	06250000
.T1300	ANOP				06260000
	AIF	(&NAS).A0200			06270000
	SPACE				06280000

```

* IF THIS IS AN ASCII TRANSLATE SITUATION, THEN USE RESERVED ASCII AB 06290000
* TRANSLATION BUFFER AS FROM ADDRESS. AB 06300000
SPACE 06310000
TBN $BDATT(,DTF), $BCASK ASCII LINE, AND AB 06320000
TBF SAVTAL, TASTRN * TRANSLATION DONE ? AB 06330000
JF CMFVOP NO-GO VERIFY OP CODE. AB 06340000
MVC $BDWKB(2,DTF), LCBATL(,DTF) USE ASCII XLATE BUFFER IN WKB. B 06350000
CMFVOP EQU * * LOCAL AB 06360000
.A0200 ANOP 06370000
AIF (&N32).T1400 06380000
TBN PLOPC(,PL), OPGET GET TO BE PERFORMED NEXT ? 0B 06390000
JT CMFPEW YES-GO DO PUT EOT TO WACK. 0B 06400000
TBN PLOPC(,PL), OPMSG IS OP REQUESTED A 0B 06410000
* -----START-----@ 06415000
TBF PLOPC(,PL), OPORDR-OPRUF * PUT-EOT ? 0B 06420000
* -----END-----@ 06425000
SPACE 1 06430000
* SET PUT-EOT TO ACK OR WACK FOR 3270 SYSTEM FOR PUT-MSG. 0B 06440000
SPACE 1 06450000
JF CMFPT0 NO-NOT PUT-MSG, USE REGULAR OP0B 06460000
CMFPEW EQU * * LOCAL 0B 06470000
L PLTUBA(,PL), TUB POINT TO THE TUB. 0B 06480000
CLI TUBPHY(,TUB), TUB5M2 3270 SYSTEM ? 0B 06490000
L LCBPL@(,DTF), PL XR1 --> PARM LIST 06495000
JH CMFPT0 NO-GO CHECK FOR PUT OF ZERO LE0B 06500000
MVI $BDOPC(,DTF), $BOPEW SET PUT-EOT TO ACK/WACK. 0B 06510000
SBN LCBAT2(,DTF), LCBSET SET SEND EOT IND. 0B 06520000
TBN PL$OPC(,PL), OPUSER IF INTERNALLY SET UP SYS OP 0B 06540000
JT CMBSIO YES - DONT DO CHECK ON PLOUTL 0B 06550000
* * DTF IS SET UP DIFFERENTY. 0B 06560000
J CMFVPM GO CHECK LENGTH 0B 06570000
SPACE 1 06580000
CMFPT0 EQU * * LOCAL 0B 06590000
.T1400 ANOP 06600000
TBF PL$OPC(,PL), OPUSER USER FUNCTION, AND B 06620000
TBF PLOPC(,PL), OPMSG RECORD MODE ? B 06630000
JT CMFVRC YES-GO HANDLE RECORDS. B 06640000
TBN PLOPC(,PL), OPMSG IS OP REQUESTED A B 06650000
TBF PLOPC(,PL), OPORDR-OPRUF * PUT EOT? B 06660000
CLC $BDREL(2,DTF), X$0000 ZERO LENGTH PUT-MSG ? B 06670000
JC CMFVPB, FLSNEQ NO-GO SET PUT BLOCK OP CODE. B 06680000
SBN LCBAT2(,DTF), LCBSET SET SEND EOT IND. B 06690000
CMFVET EQU * * LOCAL B 06700000
TBN LCBAT2(,DTF), LCBABT ABORT OF A - 06701800
TBN LCBOPC(,DTF), OPPUT * PUT ? 06702700
JF CMFABT NO-CONTINUE 06703600
L $BDWKA(,DTF), WKA XR1-->WORK AREA 06704500
MVI WKERRD(,WKA), NOBIT SET RETRY COUNT TO 0 06705400
MVC DCOUNT(2,WKA), X$FFFC SET DELAY COUNT VERY HIGH 06706300
B CMFRTN GO WAIT FOR OP END 06707200
CMFABT EQU * LOCAL 06708100
MVI $BDOPC(,DTF), $BOPEF SET PUT END OF FILE OP CODE B 06710000
B CMFAKE GO FAKE CALL TO MLMP, FORCE CHKB 06720000
SPACE 06730000
CMFVPB EQU * * LOCAL B 06740000
MVI $BDOPC(,DTF), $BOPEB SET OP CODE TO PUT END OF BLOCKB 06750000
AIF (&N32).T1405 06760000
CMFVPM EQU * * LOCAL 0B 06770000
.T1405 ANOP 06780000
CLC PLOUTL(2,PL), LCBKLC(,DTF) OUTL GREATER THAN BLOCK LEN ? B 06790000

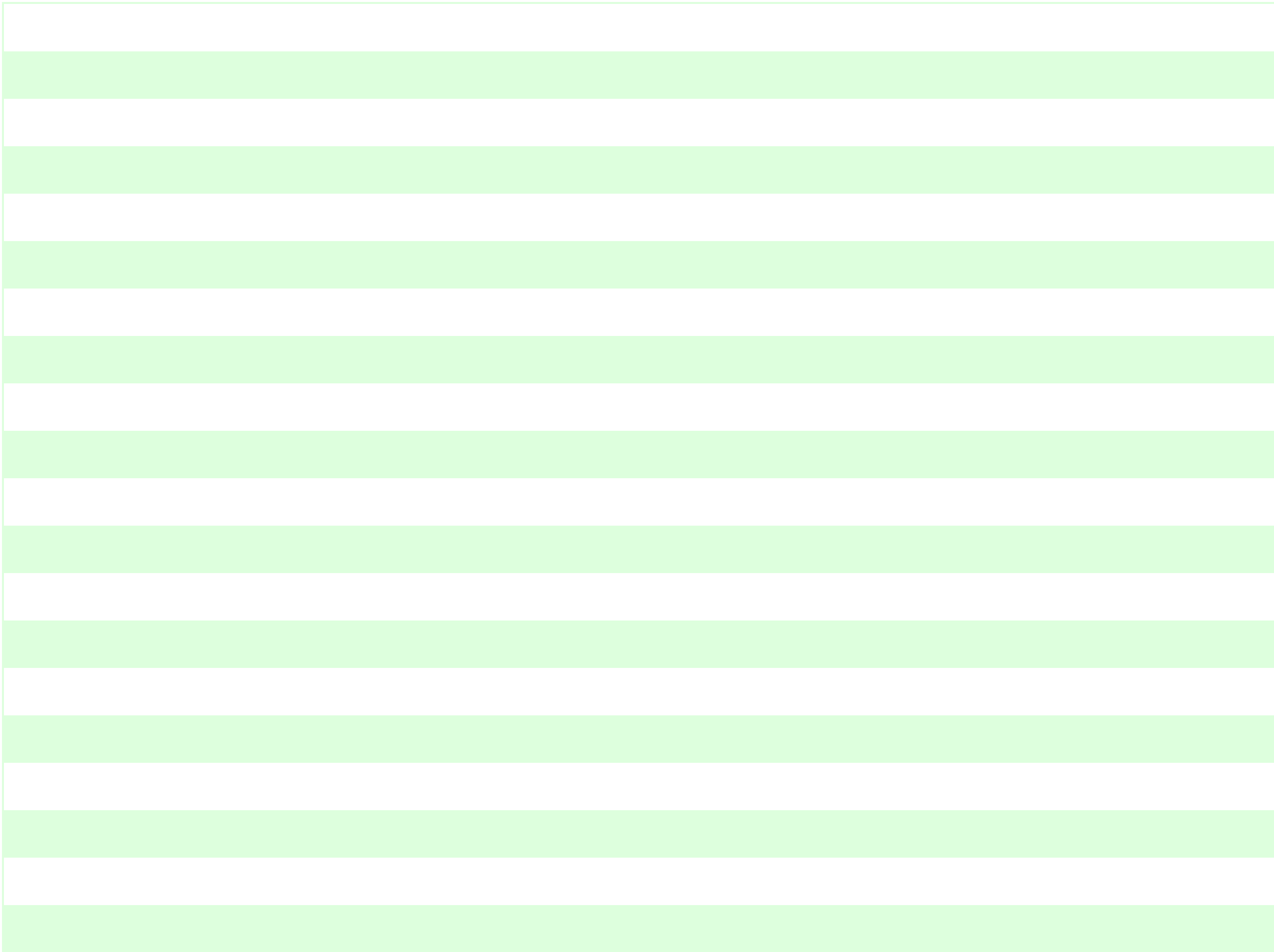
```

```

JNH CMBSIO NO-GO CALL MLMP IOCS. B 06800000
MVC $BDREL(2,DTF),LCBKLC(,DTF) TRUNCATE TO BLOCK LENGTH. B 06810000
J CMFVSM GO SET TRUNCATED IND. B 06820000
SPACE 06830000
CMFVRC EQU * * LOCAL B 06840000
L PLTUBA(,PL),TUB LOAD THE TUB REG. B 06850000
TBF SAVTA2,TASITB+TASPAN+TASVRL MLMP VARIABLE SUPPORT ? B 06860000
CLC $BDREL(2,DTF),TUBRCL(,TUB) OUTL LT TAS RECORD LEN ? B 06870000
JC CMBSIO,FLSOEQ YES-CALL MLMP, ALL IS SET. B 06880000
JL CMFVSM OUTL LT TAS RECL, BLANKS NEEDED B 06890000
MVC $BDREL(2,DTF),TUBRCL(,TUB) USE TAS RECORD LEN FOR PUT. B 06900000
CMFVSM EQU * * LOCAL B 06910000
SBN LCBAT2(,DTF),LCBTRC SET TRUNCATED INDICATOR. B 06920000
TITLE '$E085/CMBSCH---BSCA RESCHEDULE LINE - MLMP IOS CALL' 06930000
***** B 06940000
* ISSUE START I/O OPERATION TO MLMP (BSCA IOS) - READ OR WRITE* B 06950000
***** B 06960000
SPACE 1 06970000
CMBSIO EQU * * LOCAL B 06980000
AIF (&NINT).CT055 06981000
TBN LCBOPC(,DTF),OPPUT PUT OPERATION 06982000
JF CMTROF NO-CONTINUE 06983000
SBF $FLGC,#NTRAC SET OFF NO TRACE INDICATOR 06984000
CMTROF EQU * 06985000
.CT055 ANOP 06986000
SPACE 1 06987000
L LCBPL@(,DTF),PL RELOAD THE PL REG. B 06990000
* -----START-----@19 06990500
SBF LCBAT2(,DTF),LCBSEC SET OFF SECOND BLOCK IND. B 06991000
TBN LCBAT2(,DTF),LCBRCI RECEIVE INITIAL ? B 06991500
JT CMSDTX YES-GO TO MLMP B 06992000
SBN LCBAT2(,DTF),LCBSEC SET ON SECOND BLOCK IND. B 06992500
TBN PL$OPM(,PL),OPGET GET OPERATION AND B 06993000
TBF LCBAT1(,DTF),LCBEOT * NOT SEARCH EOT B 06993500
JF CMSDTX NO-CALL MLMP B 06994000
TBN LCBAT2(,DTF),LCBTRC DATA TRUNCATED ? B 06994030
SBF LCBAT2(,DTF),LCBTRC SET OFF DATA TRUNCATED. B 06994060
JT CMSDTX YES - GO TO MLMP. B 06994090
TBF SAVTA2,TASREC+TASBLK MESSAGE MODE TERMINAL ? B 06994120
JT CMBNOG YES - DON'T GO TO MLMP. B 06994150
TBN SAVTA2,TASREC RECORD MODE TERMINAL ? B 06994180
L $BDIOB(,DTF),XR1 XR1 -> IOB. B 06994210
CLI IOBCMP(,XR1),PROCES IOB IN PROCESS ? B 06994240
L LCBPL@(,DTF),PL XR1 -> PARM LIST. B 06994270
JC CMBNOG,FLSNEQ NO-DON'T FAKE AN OP END. B 06994300
ALC #OPEND(1),X$0001 UP OP END COUNT BY ONE. B 06994330
ALC LCBOPE(1,DTF),X$0001 BUMP LINE OP END COUNT BY ONE. B 06994360
CMBNOG EQU * * LOCAL. B 06994390
* GET OPERATION AND NOT RECEIVE INITIAL AND NOT SEARCH EOT, THEN B 06994500
* DON'T CALL MLMP. MLMP HAS ALREADY STARTED THE NEXT GET. WAIT FOR B 06995000
* THE OP-END AND CALL CHECK TO MOVE THE DATA. B 06995500
MVI DTFCMP(,DTF),OPACC SET COMP TO OP ACCEPTED. B 06996000
SBF $BDAT1(,DTF),$BCNOW SET OFF SPANNED RECORDS B 06996500
L $BDWKA(,DTF),WKA XR1 -> MLMP WORKAREA B 06997000
SBF $BWFG3(,WKA),F3MOVE SET OFF DATA MOVED IND. B 06997500
L LCBPL@(,DTF),PL XR1 -> PARAMETER LIST B 06998000
J CMTRGT GO AROUND CALL FOR MLMP B 06998500
CMSDTX EQU * * LOCAL B 06999000
* -----END-----@19 06999500
B $$BSMS ##### MLMP IOS CALL ##### B 07000000

```

CMTRGT	EQU	*	* LOCAL	B	07015000
	B	CMTRCE	TRACE SIO VIA TRACE SUBROUTINE	B	07020000
	DC	AL1(CCPTRIB)	CCP RIB	B	07030000
	DC	AL1(TRRIB)	TRACE SUBRIB	B	07040000
	DC	AL1(TTBSIO)	ID FOR BSCA START IO	B	07050000
	SPACE				07060000
	SBN	LCBAT2(,DTF),LCBACT	SET LINE ACTIVE IND.	B	07070000
	AIF	(&NCPU).T1500			07075000
	TBN	PLOPM(,PL),OPOLT	ON LINE TEST REQUEST ?	UB	07080000
	L	\$BDWKA(,DTF),WKA	XR1-> MLMP WORK AREA.	B	07090000
	JF	CMFAKR	NO-GO CHECK FOR ERROR POSTED.	UB	07100000
	SBN	\$BWFG3(,WKA),\$BWRFT	SET RFT STARTED IND.	UB	07110000
	SBN	LCBAT2(,DTF),LCBSET	SET SEND EOT INDICATOR.	UB	07120000
	AGO	.T1600			07122000
.T1500	L	\$BDWKA(,DTF),WKA	XR1-> MLMP WORK AREA.	B	07124000
.T1600	ANOP				07126000
CMFAKR	EQU	*	* LOCAL	B	07130000
	TBN	ACKSD(,WKA),AKERR	ERROR POST PENDING FROM MLMP ?	B	07140000
	JF	CMNFAK	NO - CHECK OTHERS	B	07150000
	SBF	ACKSD(,WKA),AKERR	SET OFF ERROR CONDITION		07152000
	CLI	LCBOPE(,DTF),NOBIT	OP ENDS = 00 ?		07154000
	JE	CMFAKE	YES - FAKE AN OP END		07156000
CMNFAK	EQU	*	* LOCAL	B	07158000
	TBN	\$BWFG3(,WKA),F3MOVE	RECORD MOVED INDICATOR ON ?	B	07160000
	JF	CMFEOT	NO-GO CHECK EOT POSTED IN IOB.	B	07170000
	SBN	LCBOPC(,DTF),LCBMVD	SET ON DATA MOVED INDICATOR.	B	07180000
	TBF	LCBOPE(,DTF),ALLBIT	OP END COUNT = 0 ?		07181200
	JT	CMFAKE	YES - GO FAKE AN OP END		07181800
	SPACE	2			07182400
	*	IF A BLOCK MODE TERMINAL,THEN MLMP HAS ALREADY MOVED THE DATA;			07182480
	*	HOWEVER,THE MAX RECORD LENGTH WAS USED. WE MUST CALCULATE THE			07182560
	*	TRUE RECORD LENGTH BEFORE POSTING THE USER.			07182640
	SPACE	2			07182720
	TBF	SAVTA2,TASMSG	NOT MESSAGE MODE ?		07182800
	JF	CMFEOT	NO-CHECK EOT POSTED		07182880
	SLC	SAVCAT-2(2),SAVCAT	SUB TAR FROM CAR		07183000
	SLC	SAVCAT-2(2),X\$0001	DECREMENT FOR SOH OR STX		07183600
	TBN	SAVTA2,TASTSP	TRANSPARENCY ?		07184200
	JF	CMNXPR	NO-GOPOST		07184800
	SLC	SAVCAT-2(2),X\$0001	DECREMENT FOR DLE		07185400
CMNXPR	EQU	*	*		07186000
	L	LCBPL@(,DTF),PL	XR1-->PARM LIST		07186600
	MVC	PLEFFL(2,PL),SAVCAT-2	PLUG NEW LENGTH		07187200
	J	CMFRTN	GO TO POSTING LOGIC		07187800
	SPACE	2			07188400
CMFEOT	EQU	*	* LOCAL	B	07210000
	L	\$BDIOB(,DTF),IOB	POINT TO THE IOB.	B	07220000
	CLI	IOBCMP(,IOB),\$BCEOT	EOT POSTED IN THE IOB ?	B	07230000
	JNE	CMFRTN	NO-GO EXIT NORMALLY.	B	07240000
CMFAKE	EQU	*	* LOCAL	B	07250000
	ALC	#OPEND(1),X\$0001	UP OP END COUNT BY ONE.	B	07260000
	ALC	LCBOPE(1,DTF),X\$0001	BUMP LINE OP END COUNT.	B	07270000
CMFRTN	EQU	*	* LOCAL	B	07280000
	AIF	(&NINT).CT060			07290000
.CT060	SBF	LCBATR(,DTF),LCBTIM	SET TIMER IND. OFF	B	07300000
	ANOP				07310000
	B	CMPAII	GO TO POSTING LOGIC	B	07320000
.C0500	ANOP				07330000
	MEND				07340000



MODULE-\$E080 , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
* NAME: $E080 * 00030000
***** 00040000
$E080 00050000
  GBLB &ONE, &NOB, &NOM, &MIN, &NDME, &NMSG, &N37, &NBDA 00060000
  GBLB &NPP, &NMP, &NSWL, &NCS, &NITB, &NTSP, &N32, &NAS, &NDF, &NCPU 00070000
  LCLA &MIX 00090000
  TEXT 00100000
* R-06, C-00 CHANGE LEVEL 00110000
  AIF (&NOB).C0470 00120000
  TITLE '$E080/CMBREQ--NEW BSCA TP REQUEST HANDLING ' 00130000
***** 00140000
* 00150000
* NAME--CMBREQ * 00160000
* 00170000
* TITLE--ACCEPT NEW BSCA TP REQUEST * 00180000
* 00190000
* FUNCTION--ACCEPT NEW TP PARAMETER LIST FOR BSCA TERMINAL. * 00200000
* PERFORM THE FUNCTION REQUESTED IF IT CAN BE HANDLED * 00210000
* IMMEDIATELY, OTHERWISE, PLACE THE PARAMETER INTO THE * 00220000
* LCB LINE QUEUE OF WORK TO BE DONE. * 00230000
* 00240000
* OPERATION-- * 00250000
* . IF THIS IS A DISCONNECT REQUEST, THEN INSURE THAT THE * 00260000
* LINE IS NOT CONNECTED TO THE SPECIFIC TERMINAL. IF * 00270000
* NO INVITE REQUEST IS INVOLVED, THEN POST TP COMPLETED * 00280000
* TO THE SYSTEM. IF AN INVITE IS INDICATED THEN TREAT * 00290000
* LIKE AN INVITE ONLY REQUEST. * 00300000
* 00310000
* . IF A PUT OPERATION TO A TERMINAL IN CCP ERP, IGNORE * 00320000
* THE PUT OPERATION. POST THE USER TP COMPLETE IF NO * 00330000
* INPUT OPERATION INDICATED. IF A PUT-THEN-GET TREAT * 00340000
* LIKE A GET/INVITE ONLY OPERATION. * 00350000
* 00360000
* . IF A STOP INVITE OR PURGE I/O CALL $CC4BP TO HANDLE. * 00410000
* ON RETURN FINISH SCHEDULING. IF THE * 00420000
* BSCA TRANSIENT INDICATES AN ABORT OF THE LINE IS * 00430000
* REQUIRED, SET UP THE ABORT OPERATION. * 00440000
* 00450000
* . IF A REGULAR TP DATA OPERATION TO A BSCA LINE, * 00460000
* CHECK TO INSURE THAT IT DOES NOT CONFLICT WITH WHAT * 00470000
* IS CURRENTLY BEING PERFORMED ON THE LINE. * 00480000
* 00490000
* . IF PUT-NO-WAIT AND SPACE IS AVAILABLE FOR PARM LIST * 00550000
* AND DATA, MOVE THE PARAMETER LIST AND DATA TO A HOLD * 00560000
* BUFFER AND USE THE HOLD BUFFER FOR ANY FURTHER * 00570000
* REFERENCE TO THIS OPERATION. OTHERWISE, TREAT THE * 00580000
* REQUEST LIKE A PUT-WAIT OPERATION. NOTE: A PUT-NO-WAIT * 00590000
* MESSAGE TO A BSCA SWITCHED LINE IS ALSO TREATED LIKE A * 00600000
* PUT-WAIT. * 00610000
* 00620000
* . IF A TP REQUEST THAT REQUIRES DATA TRANSFER, THEN QUEUE * 00630000
* THE REQUEST ONTO THE LINE QUEUE FOR THE APPROPRIATE * 00640000
* LCB. * 00650000
* 00660000
* . AFTER QUEUEING UP A NEW TP REQUEST, CHECK THE STATUS * 00670000
* OF LINE TO: * 00680000
* - INCLUDE A NEW GET/INVITE IF POLLING IS ALREADY * 00690000
```

```

*          GOING ON. * 00700000
* - STOP A POLLING SEQUENCE IF A PUT REQUEST WAS * 00710000
* JUST RECEIVED. * 00720000
* - CONTINUE OPERATION IF THE REQUEST JUST RECEIVED IS * 00730000
* THE CONTINUATION OF A PREVIOUSLY STARTED BSCA LINE * 00740000
* OPERATION. * 00750000
* * 00760000
* . EXIT TO THE RESCHEDULE LOGIC TO: * 00770000
* - POST COMPLETION OF THE TP SCHEDULED OPERATION. * 00780000
* - TO PERFORM THE NEXT PRIORITY ACTION ON THE LINE. * 00790000
* - TO CHECK FOR OTHER WORK WITHIN 'CM' THAT CAN BE DONE * 00800000
* AT THIS TIME. * 00810000
* * 00820000
* ENTRY POINTS: * 00830000
* CMBREQ - HANDLE NEW TP REQUEST. * 00840000
* CMBSTP - ABORT LINE (FROM CMBOPE). * 00850000
* CMBTBY - TEST IF LINE BUSY (FROM CMFRMN AND CMNWRK) . * 00860000
* * 00870000
* INPUT-- * 00880000
* CMSPL - ADDRESS OF THE TP REQUEST BEING HANDLED. * 00890000
* CMSDTF - ADDRESS OF THE DTF THAT PL IS FOR. * 00900000
* * 00910000
* OUTPUT-- * 00920000
* LCBPLQ - NEW REQUEST ADDED TO THIS QUEUE. ANDLED. * 00930000
* * 00960000
* EXTERNAL REFERENCES-- * 00970000
* $BSCL - CLOSE BSCA SWITCHED LINE FOR DISCONNECT OPERATION. * 00980000
* $BSOB - RE-OPEN THE BSCA SWITCHED LINE AFTER A DISCONNECT * 00990000
* OPERATION. * 01000000
* $CC4MP - TRANSIENT TO IGNORE A PUT TO A TERMINAL IN CCP ERP. * 01010000
* $CC4BP - TRANSIENT TO HANDLE BSCA STOP II/PURGE I/O. * 01020000
* $CC4BR - TRANSIENT TO REJECT A TP REQUEST THAT CONFLICTS * 01030000
* WITH A CURRENT LINE OPERATION. * 01040000
* $CC4BC - PRIORITY CANCEL OF READ ON LINE, PUT PENDING. * 01050000
* CMIVGM - HANDLE A NEW INVITE INPUT REQUEST. * 01060000
* CMPOST - POST REQUEST REJECTED. * 01070000
* CMQUE - QUEUE TP REQUEST ON LINE QUEUE. * 01080000
* * 01090000
* EXIT, NORMAL-- * 01100000
* TO CMNOBY, TO SCHEDULE ANOTHER OP FROM OWNER OF LINE. * 01100100
* TO CMNOBY, TO SCHEDULE ANOTHER OP FROM OWNER OF LINE. * 01104100
* TO CMBSCH, THE BSCA LINE RESCHEDULING FUNCTION. * 01110000
* TO CMPAII, IF LINE CAN NOT BE SCHEDULED NOW. * 01110100
* TO CMPAII, IF LINE CAN NOT BE SCHEDULED NOW. * 01114100
* * 01120000
***** * 01130000
EJECT * 01140000
CMBREQ EQU * * B 01150000
AIF (&NSWL).S0100 * 01160000
TBN PLOPM(,PL),OPDISC DISCONNECT ? SB 01170000
JF CMBPTG NO-GO HANDLE PUT-GET OP CODE. SB 01180000
SPACE 1 * 01190000
***** SB 01200000
* DISCONNECT BSC SWITCHED LINE IF REQUESTED TO THE CURRENT TUB * SB 01210000
***** SB 01220000
SBF PL$OPM(,PL),OPPUT SET OFF PUT OP IND. SB 01230000
CLC PLTUBA(2,PL),LCBOWN(,DTF) THIS TUB CONNECTED ? SB 01240000
JE CMBDIS YES - GO DO DISCONNECT SB 01250000
CLI PLOPC(,PL),OPPURG PURGE OPERATION ? SB 01252000
JE CMBDIS YES - DISCONNECT THE LINE SB 01254000

```

```

CLI   LCBOWN-1( ,DTF),NOBIT    NO OWNER OF THE LINE ?    SB 01256000
JE    CMBDIS                   YES - DISCONNECT THE LINE  SB 01258000
L     PLTUBA( ,PL),XR2         TUB ADDRESS                SB 01260000
CLC   TUBTCB(2,XR2),@TMTCB    DID TERMINATION REQUEST DISC? SB 01270000
JNE   CMBPDI                   NO - NOT CONNECTED        SB 01280000
*                                     YES - THE TASK OWNS THE LINE SB 01290000
*                                     * (MAY BE IN ID VERIFICATION) SB 01300000
CMBDIS EQU *                   * LOCAL                      SB 01310000
LA    CMOCPPL,XR2             ADDR OF MLMP OPEN/CLOSE PL  SB 01320000
*                                     * WITH DTF ADDR AT OFFSET 3. SB 01330000
SPACE 1                       01340000
SVC   0                       SVC TO LOAD CLOSE            SB 01350000
DC    AL1(LDRIB)              LOAD XIENT BY C/S          SB 01360000
CMCLOZ DS CL2                  CLOSE XIENT C/S (STARTUP SETS) SB 01370000
DC    AL1(NSECS)              NUMBER OF SECTORS - 1      SB 01380000
SPACE 1                       01390000
SVC   0                       SVC TO LOAD OPEN           SB 01400000
DC    AL1(LDRIB)              LOAD XIENT BY C/S          SB 01410000
DS    CL2                     OPEN XIENT C/S (STARTUP SETS) SB 01420000
DC    AL1(NSECS)              NUMBER OF SECTORS - 1      SB 01430000
SPACE 1                       01440000
L     CMSDTF,DTF              RESTORE DTF ADDRESS        SB 01450000
SBF   LCBATR( ,DTF),LCBNIT    SET OFF LINE CONNECTED IND. SB 01460000
SBF   LCBAT3( ,DTF),LCBENB    SET OFF LINE ENABLED IND.  SB 01470000
SLC   LCBOWN(2,DTF),LCBOWN( ,DTF) ZERO OUT TUB OWNING INDICATORSB 01480000
SLC   #OPEND,LCBOPE(1,DTF)    REMOVE RESIDUAL OP ENDS.   SB 01490000
MVI   LCBOPE( ,DTF),NOBIT     ZERO LCB OP END COUNT.     SB 01500000
TBN   PL$OPM( ,PL),OPGET      GET OPERATION REQUESTED ALSO. SB 01510000
BF    CMBSCH                  NO-GO RESCHEDULE THE LINE.  SB 01520000
J     CMBERP                   GO TO NEXT OPERATION CHECK. SB 01530000
SPACE 1                       01540000
*****                          SB 01550000
* DISCONNECT FOR TERMINAL NOT CONNECTED * SB 01560000
*****                          SB 01570000
SPACE 1                       01580000
CMBPDI EQU *                   * LOCAL                      SB 01590000
TBN   PL$OPM( ,PL),OPGET      GET REQUESTED ALSO         SB 01600000
JT    CMBERP                   YES - HANDLE IT.           SB 01610000
B     CMPAII                    NO - REQUEST FINISHED      SB 01620000
.S0100 ANOP                    01630000
CMBPTG EQU *                   * LOCAL                      B 01650000
TBN   PLOPC( ,PL),OPPUT+OPGET IS OP A PUT THEN GET      B 01660000
JF    CMBERP                    NO - SKIP SET OFF GET BIT  B 01670000
SPACE 1                       01680000
*****                          B 01690000
* IF PUT THEN GET - SCHEDULE A PUT * B 01700000
*****                          B 01710000
SPACE 1                       01720000
SBF   PL$OPM( ,PL),OPGET      SET OFF GET BIT            B 01730000
CMBERP EQU *                   * LOCAL                      B 01740000
SPACE 1                       01750000
*****                          B 01760000
* IGNORE PUT TO TERMINAL IN ERP * B 01770000
*****                          B 01780000
SPACE 1                       01790000
B     CMERPC                    GO CHECK FOR AND HANDLE PUT B 01800000
*                                     * TO TERMINAL IN ERP.      B 01810000
* IF TERMINAL IS IN ERP AND OP WAS A PUT-ONLY CMERPC WILL GO B 01820000
* DIRECTLY TO CMPAII. IF IN ERP AND OP WAS PUT THEN GET B 01830000
* CMERPC RETURNS HERE TO SCHEDULE GET. B 01840000

```



```

SPACE 01850000
TBN PLOPM(,PL),OPSTOP IS STOP BIT ON B 01860000
JF CMBNSP NO-CONTINUE NON STOP CHECKING B 01870000
SPACE 1 01880000
***** B 01890000
* STOP INVITE REQUEST * B 01900000
***** B 01910000
SPACE 01920000
LA 0(,PL),XR2 LOAD PL @ INTO XR2 FOR XIENT. B 01930000
&MIX SETA &NCS+&NSWL 01940000
AIF (&MIX EQ '2').S0125 01950000
USING CMBSKP,XR1 01960000
LA CMBSKP,XR1 LOAD @ OF SKIP BIT RTN. C/SB 01970000
.MVI CMB#SB(,XR1),SBN1 SET OP TO SET POLL BIT ON. C/SB 01980000
.S0125 SVC 0 ##### TRANSIENT CALL ##### B 01990000
DC AL1(CCPRIB) CCP SVC RIB B 02000000
DC AL1(CC4BP) ID FOR BSCA PURGE/STOP INVITE. B 02010000
SPACE 02020000
* TRANSIENT WILL RETURN HERE IF NO ABORT NEEDED, STOP SUCCESSFUL. B 02030000
* $CC4BC MAY SET UP CANCEL WHICH WILL BE HANDLED ON THE NEXT CALL B 02040000
* TO CCP CHECK ROUTINE (CMBMCH). #OPEND INCREMENTED TO CAUSE CALL. B 02050000
SPACE 1 02060000
B CMRQBF GO SET TP REQ'T BITS OFF. B 02070000
SPACE 02080000
* TRANSIENT WILL RETURN HERE IF ABORT OF LINE NEEDED. B 02090000
* DATA WAS BEING TRANSMITTED. B 02100000
SPACE 02110000
***** B 02120000
* SET UP FOR ABORT OF LINE * B 02130000
***** B 02140000
SPACE 02150000
CMBSTP EQU * * (FROM CMBOPE FOR STOP FAIL) B 02160000
TBN LCBOPC(,DTF),OPPUT ABORT OF A PUT ? B 02170000
BT CMBEOT YES-GO SET WRITE EOT. B 02180000
L $BDWKA(,DTF),WKA POINT TO BSCA WORK AREA. B 02190000
MVI WKDELL+4(,WKA),CPURGE SET CCP PURGE RETURN CODE. B 02200000
SBN BSFLGD(,WKA),FWDABT SET FORWARD ABORT REQUEST. B 02210000
L WKIOBD(,WKA),IOB POINT TO IOB OUT OF WORK AREA. B 02220000
SBF IOBFLA(,IOB),TDLAY SET OFF TIME DELAY SEQUENCE. B 02230000
B CMFORB GO TO FORM OP FOR MLMP. B 02240000
SPACE 1 02250000
CMBEOT EQU * * LOCAL B 02260000
SBN LCBAT2(,DTF),LCBSET SET SEND EOT B 02270000
ALC $BDBKX(2,DTF),$BDBKL(,DTF) UPDATE BLK PTR SO WONT GET B 02280000
* * ANY MORE DATA B 02290000
B CMFORB FORM NEXT MLMP OPERATION B 02300000
TITLE '$E080/CMBREQ--NEW BSCA TP REQUEST -- NON-STOP REQUEST' 02310000
***** B 02320000
* NON-STOP REQUEST * B 02330000
***** B 02340000
SPACE 02350000
CMBNSP EQU * * LOCAL B 02360000
* IF THIS OPERATION IS TO A TUB IN CCP ERP - THEN CANNOT HAVE A PUT B 02370000
* THAT WILL BE SCHEDULED B 02380000
SPACE 02390000
TBN TUBAT3(,XR2),TUBERP IS TUB IN CCP ERP B 02400000
L TUBDTF(,XR2),DTF POINT XR2 AT THE DTF B 02410000
JT CMBSOP JUMP IF IN CCP ERP. B 02420000
TBN PL$OPM(,PL),OPPNW IS IT PUT NO WAIT 02480000
JF CMBSOP IF NOT JUMP TO END OF PNW CHECK. 02490000

```

```

SPACE 02500000
* PUT NO WAIT REQUESTED - CONVERT IF NOT ALLOWED 02510000
SPACE 02520000
AIF (&NSWL).S0400 02530000
AIF (&MIX EQ '3').S0500 02540000
TBN $BDATR(,DTF), $BCSWI SWITCHED, AND NOT SLB 02550000
TBF $BDATR(,DTF), $BCMPT * MULTI-POINT ? SLB 02560000
JT CMBPW YES-ALL PUTS ARE PUT WAITS. SLB 02570000
.S0400 ANOP 02580000
TBF PLOPM(,PL), OP$SYS USER REQUEST, AND B 02590000
CLI PLOPC(,PL), OPMSG LESS THAN MESSAGE LEVEL PUT ? B 02600000
JC CMBSOP, ANY+FALSE+HI+EQ NO- SKIP SET TO WAIT OP . B 02610000
.S0500 ANOP 02620000
CMBPW EQU * * LOCAL 02630000
SBF PL$OPM(,PL), OPNOW MAKE OPERATION A WAIT 02640000
SPACE 2 02650000
***** B 02660000
* REJECT INVALID BSCA OPERATION REQUEST SEQUENCES. * B 02670000
***** B 02680000
SPACE 1 02690000
CMBSOP EQU * * LOCAL B 02700000
* XR1 CONTAINS ADDRESS OF PARM LIST TO BE SCHEDULED. B 02702000
SPACE 02704000
* ----- START @18 02704400
TBN $BDDEV(,DTF), BSCA BSCA LINE ? B 02704800
BF CMBSEX NO-GO TO END OF BSCA OP CHECK B 02705200
* ----- END --@18 02705600
B CMTASV GO SET UP TAS SAVE AREAS. B 02706000
SPACE 02708000
MNN CMTOPT+1, LCBOPC(,DTF) USE LAST OP ON LINE FOR CHECK. B 02710000
SBF CMTOPT+1, OPNOW SET OFF NO WAIT BIT. B 02720000
MVC CMTOP2+1(1), CMTOPT+1 SET UP TEST FOR BUSY CHECK. B 02730000
AIF (&NSWL).S0200 02740000
&MIX SETA &NMP+&NPP+&NCS 02750000
AIF (&MIX EQ '3').S0150 02760000
TBN $BDATR(,DTF), $BCSWI SWITCHED, AND SLB 02770000
TBF $BDATR(,DTF), $BCMPT * NOT MULTI-POINT ? SLB 02780000
JF CMBACT NO-GO CHECK FOR LINE ACTIVE. SLB 02790000
.S0150 ANOP 02800000
SPACE 1 02810000
* ----- * SB 02820000
* SWITCHED * SB 02830000
* ----- * SB 02840000
SPACE 1 02850000
TBN PLOPM(,PL), OP$SYS SYSTEM SB 02860000
TBN PLOPC(,PL), OPPUT * PUT TO SB 02870000
CLC PLTUBA(2,PL), LCBOWN(,DTF) * NON-CONNECTED LINE ? SB 02880000
JC CMBSII, FL$OEQ NO-GO CHECK FOR SYSTEM II. SB 02890000
SBF PL$OPM(,PL), OPPUT SET OFF PUT OPERATION. SB 02900000
* ----- START @18 02902000
SBF LCBAT2(,DTF), LCBPUT SET PUT OPERATION IND OFF SB 02904000
SBN PL$OPM(,PL), OPNOW SET NO WAIT IND FOR POST SB 02906000
* ----- END --@18 02908000
TBN PLOPC(,PL), OPGET GET OPERATION ALSO REQUESTED. SB 02910000
BF CMPAII NO-GO POST COMPLETION. SB 02920000
SBN PL$OPM(,PL), OPINV SET ON INVITE OP CODE BITS. SB 02930000
CMBSII EQU * * LOCAL SB 02940000
TBN PLOPM(,PL), OP$SYS SYSTEM SB 02950000
TBN PL$OPM(,PL), OPINV * INVITE INPUT OP ? SB 02960000
JF CMBENB NO-GO CHECK ENABLED STATUS. SB 02970000

```

```

*----- START @18 02971000
TBN PLOPC( ,PL) ,OPJRSH OPERATION TO RESCHEDULE ONLY? SB 02972000
JF CMBNJR NO-CONTINUE OP CHECKING SB 02973000
TBF LCBAT3( ,DTF) ,LCBENB LINE ENABLED ? SB 02974000
BT CMBSCH NO-GO TO RESCHEDULE SB 02975000
B CMPAII GO TO POST REQUEST SB 02976000
CMBNJR EQU * * LOCAL SB 02977000
*----- END --@18 02978000
L PLTUBA( ,PL) ,XR2 POINT TO THE TUB. SB 02980000
TBN TUBCHR( ,XR2) ,TUBCMN COMMAND CAPABLE TERMINAL ? SB 02990000
JT CMBRLD YES-GO CHECK ENABLED STATUS. SB 03000000
SBF TUBAT2( ,XR2) ,TUBIIS SET OFF INVITE SCHEDULED. SB 03010000
L TUBDTF( ,XR2) ,DTF POINT TO THE DTF. SB 03020000
TBN LCBAT3( ,DTF) ,LCBENB LINE ENABLED ? SB 03030000
BT CMPAII YES-GO POST REQUEST RESULTS. SB 03040000
B CMBSCH GO TO RESCHEDULE THE LINE. SB 03050000
SPACE 1 03060000
CMBRLD EQU * * LOCAL SB 03070000
L TUBDTF( ,XR2) ,DTF POINT TO THE DTF. SB 03080000
CMBENB EQU * * LOCAL SB 03090000
TBN LCBAT3( ,DTF) ,LCBENB LINE ENABLED ? SB 03100000
JF CMBSOX NO-GO TO END OF BSCA OP CHECK. SB 03110000
CLC PLTUBA( 2, PL) ,LCBOWN( ,DTF) THIS TUB CONNECTED ON LINE ? SB 03120000
JNE CMBCNT NO-GO CHECK CONNECTION TYPE. SB 03130000
TBN LCBAT2( ,DTF) ,LCBACT LINE ACTIVE WITH DATA ? SB 03140000
JF CMBSOX NO - FREE TO BE USED. SB 03150000
TBN PLOPC( ,PL) ,OPPUT PUT TO HANDLE AND, SB 03154100
TBN LCBAT2( ,DTF) ,LCBRCI * INITIAL RECEIVE ON LINE ? SB 03160000
BT CMBSOX YES - ACCEPT THE REQUEST. THE SB 03170000
* * ONLY WAY A PUT COULD GET SB 03180000
* * THRU TO CM WOULD BE A SYS SB 03190000
* * OLT TO CPU. CANCEL RECV. SB 03200000
J CMTOPT NO- GO CHECK OP CODE MATCH. SB 03210000
SPACE 1 03220000
CMBCNT EQU * * LOCAL SB 03230000
TBN PLOPM( ,PL) ,OP$SYS SYSTEM REQUEST ? SB 03240000
JT CMBSOX YES-GO TO END OF BSCA OP CHK. SB 03250000
L PLTUBA( ,PL) ,TUB POINT TO THE NEW REQUEST TUB. SB 03260000
L LCBOWN( ,DTF) ,XR2 POINT TO ACTIVE TUB. SB 03270000
TBF TUBTA1( ,TUB) ,TASCNC BOTH SB 03280000
TBF TUBTA1( ,XR2) ,TASCNC * ANSWER ? SB 03290000
TBF TUBTA2( ,TUB) ,TASVFY BOTH SB 03300000
TBF TUBTA2( ,XR2) ,TASVFY * VERIFY ID'S ? SB 03310000
L TUBDTF( ,TUB) ,DTF POINT TO THE DTF. SB 03320000
L CMSPL, PL POINT TO THE PARM LIST. SB 03330000
JF CMTREJ NO-GO REJECT LAST OP REQUEST. SB 03340000
J CMBSOX GO TO END OF BSCA OP CHECK. SB 03350000
SPACE 03360000
*-----* SB 03370000
* NON - SWITCHED * SB 03380000
*-----* SB 03390000
SPACE 1 03400000
AIF (&MIX EQ '3') .S0250 03410000
CMBACT EQU * * LOCAL SB 03420000
.S0200 ANOP 03430000
TBN LCBAT2( ,DTF) ,LCBACT LINE ACTIVE WITH DATA ? B 03440000
JF CMBSOX NO-GO TO END OF BSCA OP CHECK. B 03450000
SPACE 1 03460000
* LINE ACTIVE B 03470000
SPACE 1 03480000

```

TBN	LCBAT2(,DTF),LCBRCI	RECEIVE INITIAL ?	B	03490000
JT	CMBSOX	YES-TREAT LIKE LINE NOT ACTIVE.	B	03500000
L	PLTUBA(,PL),TUB	LOAD TUB @ INTO REG.	B	03510000
CLC	TUBTCB(2,TUB),LCBOWN(,DTF)	THIS TASK OWN THE LINE ?	B	03520000
L	CMSPL,PL	RELOAD PARM LIST REG.	B	03530000
JNE	CMBSOX	NO-NO OP CONFLICT, SKIP CHECKS.	B	03540000
	SPACE 1			03550000
*	ACCEPT THE OPERATION IF AN INVITE OR PUT-NO-WAIT MESSAGE.		B	03560000
	SPACE 1			03570000
CLI	PLOPC(,PL),OPINV	INVITE INPUT ?	B	03580000
TBF	PLOPM(,PL),OP\$SYS	* OR SYSTEM REQUEST ?	B	03590000
JC	CMBSOX,FLSOEQ	YES-OP OK, SKIP OTHER CHECKS.	B	03600000
TBN	PLOPC(,PL),OPMSG+OPPNW	PUT-NO WAIT/MESSAGE, AND	B	03610000
TBN	PL\$OPM(,PL),OPPNW	* STILL A PUT-NO WAIT ?	B	03620000
JT	CMBSOX	YES-ACCEPT THE OPERATION.	B	03630000
	SPACE 1			03640000
*	ACCEPT THE OPERATION IF THE CURRENT LINE OP IS INVITE-MESSAGE, OR		B	03650000
*	PUT-NO-WAIT-MESSAGE.		B	03660000
	SPACE 1			03670000
TBN	LCBOPC(,DTF),OPINV	INVITE OP ON LINE, AND	B	03680000
TBN	SAVTA2,TASMSG	* IS IT MESSAGE MODE ?	B	03690000
JT	CMBSOX	YES-ACCEPT THIS OP REQUEST.	B	03700000
TBF	LCBAT1(,DTF),LCBNTQ	PARAM LIST QUEUED, AND	B	03710000
L	LCBPL@(,DTF),XR2	POINT TO CURRENT PARM LIST.	B	03720000
TBN	PLOPC(,XR2),OPPNW+OPMSG	* IS IT A PUT-NO-WAIT-MESSAGE ?	B	03730000
JT	CMBSOX	YES-ACCEPT THIS OP REQUEST.	B	03740000
*		----- START --@27		03741000
TBF	PL\$OPC(,XR2),OPRFSH	NOT ACTIVE WAITING REFRESH OR	B	03744000
TBF	PLOPM(,XR2),OP\$SYS	NOT A SYSTEM REQUEST ?	B	03745000
JF	CMBSOX	YES-ACCEPT THIS OP REQUEST	B	03746000
*		----- END --@27		03747000
L	PLTUBA(,PL),XR2	POINT TO THE TUB.	B	03750000
TBN	TUBAT2(,XR2),TUBOWN	THIS TUB OWN THE LINE, AND	B	03760000
L	TUBDTF(,XR2),DTF	POINT TO THE DTF.	B	03770000
.S0250	ANOP			03780000
	SPACE 1			03790000
*	LINE ACTIVE WITH SAME TERMINAL AS THIS OP IS FOR - ONLY OK IF		B	03800000
*	THIS IS ANOTHER SUCH OPERATION (E.G. PUT BLOCK, GET BLOCK)		B	03810000
	SPACE 1			03820000
CMTOPT	EQU *	* LOCAL AND MODIFICATION	B	03830000
TBN	PL\$OPM(,PL),#	SAME TYPE OP AS LAST TIME ?	B	03840000
JT	CMBSOX	YES-GO CONTINUE,PARAM LIST OK.	B	03850000
	SPACE 1			03860000
*****				03870000
*	SET REJECTED OPERATION COMPLETION RETURN CODE, ALSO GO TO DEQ IT.		B	03880000
*****				03890000
	SPACE 1			03900000
CMTREJ	EQU *	* LOCAL	B	03910000
*		* XR1-PL,XR2-DTF FOR BR	B	03920000
SVC	0	##### TRANSIENT CALL #####	B	03930000
DC	AL1(CCPRIB)	CCP SVC RIB	B	03940000
DC	AL1(CC4BR)	* BSCA REJECT CLEAN-UP.	B	03950000
BLANKC	DC XL1'40'	CONSTANT OF A BLANK,	B	03960000
*		AND, INLINE PARM VALUE FOR	B	03970000
*		REJECT PARM LIST NOT Q'D.	B	03980000
SBF	CMSWIT,CMTPRQ	TURN OFF TP REQUEST SWITCH.	B	03990000
B	CMREDO	PREPARE TO RESCHEDULE	B	04000000
	SPACE 2			04010000
*****			B	04020000
*	REQUEST ACCEPTED		B	04030000

```

***** B 04040000
SPACE 04050000
CMBSOX EQU * * B 04060000
AIF (&NDF).B0075 04070000
L PLTUBA(,PL),XR2 ADDRESS OF TUB FB 04080000
TBN TUBTA1(,XR2),TASDFE IS IT A DFF REQUEST FB 04090000
TBF PLOPM(,PL),OP$SYS AND A USER REQUEST FB 04100000
TBF PLRECA-1(,PL),ALLBIT AND GETMAIN NOT DONE YET FB 04110000
* * (HAVE NOT BEEN TO DFF YET) FB 04120000
TBN PLOPC(,PL),OPPUT * AND A PUT OR COPY,THEN THIS FB 04130000
* * REQUEST MUST GO TO DFF FOR FB 04140000
* * DATA FORMATING. FB 04150000
JF CMQIT NO - GO QUEUE REQUEST. FB 04160000
SBF CMSWIT,CMTPRQ NEW REQUEST PROCESSING DONE FB 04170000
SPACE 1 04171000
* IF CURRENT LARGEST SIZE TP BUFF IS NOT BIG ENOUGH TO DO THE 04173000
* DFF PUT THEN DECREMENT PLOUTL UNTIL IT IS OR UNTIL IT IS ZERO 04174000
SPACE 1 04175000
CMGMMV EQU * * 04176000
MVC GMLIST+GMSIZE(2),PLOUTL(,PL) LENGTH OF RECORD AREA FB 04180000
ALC GMLIST+GMSIZE(2),X$0004 ADD 4 FOR FREEMAIN LIST FB 04190000
CLC #GMS+1(2),GMLIST+GMSIZE ENOUGH CORE? FB 04190400
JNL CMGMCR YES-GO DO GETMAIN FB 04190800
SBF PLOUTL(,PL),ALLBIT MAKE LNGTH=256 INCREMENT FB 04191200
TBF #GMS+1,FC GREATER THAN 3? FB 04191600
JF CMGMCK YES-JUMP FB 04192000
CMGMAG EQU * * FB 04192400
SLC PLOUTL-1(1,PL),X$0001 DECREMENT BY 256 FB 04192800
JL CMROUT PLOUTL < 256 - JUMP FB 04193200
CMGMCK EQU * * FB 04193600
CLI PLOUTL-1(,PL),HEX512 OUTL LESS THAN 512? FB 04194000
JL CMROUT YES-JUMP FB 04194400
CLC #GMS+1,PLOUTL(2,PL) ENOUGH CORE NOW? FB 04194800
BNH CMGMAG NO-GO AGAIN FB 04195200
B CMGMMV YES-GO DO GETMAIN FB 04195600
CMROUT EQU * * FB 04196000
MVC PLOUTL(2,PL),GMLIST+GMSIZE RESTORE ORIGINAL- FB 04196400
SLC PLOUTL(2,PL),X$0004 LENGTH OF OUTL FB 04196800
CMGMCR EQU * * FB 04197200
SBN $FLGC,#PUTTP SET IND.-GET FROM ANYWHERE 04197600
B CMGMRT GETMAIN HOLD BUFFER FB 04200000
JNOL CMDFFP IF GETMAIN SUCCESSFUL,POST DF.FB 04210000
SBN PL$OPM(,PL),OPGETQ GETMAIN NEEDED TO QUEUE REQ . FB 04220000
J CMQIT QUEUE TEMPORARILY TO WAIT. FB 04230000
SPACE 1 04240000
CMDFFP EQU * * LOCAL FB 04250000
MVC PLRECA(2,PL),GMLIST+GMADDR @ OF GETMAINED BUFF INTO PL FB 04260000
ALC PLRECA(2,PL),X$0004 BYPASS FREEMAIN LIST FB 04270000
SPACE 1 04280000
* USER PUT MESSAGE TO DFF TERMINAL - QUEUE FOR DFF TASK FB 04290000
SPACE 1 04300000
B CMDFFQ GO QUEUE FOR DFF AND POST FB 04310000
B CMREDO GO SEE IF LINE NEEDS SCHED FB 04320000
.B0075 ANOP 04330000
SPACE 1 04340000
***** B 04350000
* QUEUE THE REQUEST (NON- DFF) * B 04360000
***** B 04370000
SPACE 04380000
AIF (&NDF).B0080 04390000

```

```

CMQIT EQU * * QUEUE NEW REQUEST FB 04400000
SPACE 04410000
.B0080 ANOP 04420000
B CMQUE QUEUE REQUEST ON LINE QUEUE 04430000
SPACE 04440000
MNN PL$OPC(,PL),PL$OPM(,PL) SAVE INTERNAL OP CODE. B 04450000
TBN PLOPC(,PL),OPPUT PUT OPERATION ? B 04450100
JF CMBTBY NO-SKIP SETTING PUT PENDING. B 04450200
L CMSDTF,DTF XR2 --> DTF B 04450250
SBN LCBAT2(,DTF),LCBPUT SET IND. PUT PENDING Q'D. B 04450300
TITLE '$E080/CMBREQ---BSCA NEW REQUEST---LINE BUSY CHECK' B 04460000
***** B 04470000
* CHECK THE TP LINE AND SEE IF IT IS BUSY * B 04480000
***** B 04490000
SPACE 04500000
CMBTBY EQU * * ( FROM CMRQBF, CMFRMN) B 04510000
AIF (&NSWL).S0700 04520000
SPACE 1 04530000
* IF SWITCHED BSCA, AND A USER OWNS THE LINE, THEN JUST Q SYSTEM SB 04540000
* REQUESTS. SB 04550000
SPACE 1 04560000
L PLTUBA(,PL),XR2 TUB ADDRESS SB 04570000
AIF (&MIX EQ '3').S0600 04580000
TBN TUBAT1(,XR2),TUBSWC SWITCHED LINE ? SLB 04590000
JF CMBTAC NO-GO CHECK ACT STATUS. SLB 04600000
.S0600 ANOP 04610000
TBN PLOPM(,PL),OP$SYS SYSTEM REQUEST ? SB 04620000
JF CMBTAC NO-GO CHECK ACT STATUS. SB 04630000
L TUBDTF(,XR2),DTF POINT TO THE DTF. SB 04640000
CLC LCBTCB(2,DTF),X$0000 SWITCHED LINE UNOWNED ? SB 04650000
JE CMBTAC YES-GO CHECK ACT STATUS. SB 04660000
CLC LCBTCB(2,DTF),@CPTCB CP OWN THE LINE ? SB 04670000
JE CMBTAC YES-GO CHECK ACT STATUS. SB 04680000
* -----START-----@18 04682000
CLC LCBOWN(2,DTF),PLTUBA(,PL) THIS TERMINAL OWN THE LINE ? SB 04684000
JE CMBTAC YES - RESCHEDULE THE OPERATION SB 04686000
* -----END-----@18 04688000
TBN LCBAT3(,DTF),LCBENB LINE ENABLED ? SB 04690000
BT CMPAII YES-GO POST OP ACCEPTED. SB 04700000
B CMBSCH GO TO RESCHEDULE THE LINE. SB 04710000
.S0700 ANOP 04720000
SPACE 1 04730000
* IF LINE IS NOT ACT WE CAN START AN OPERATION NOW - ELSE WILL HAVE TO 04740000
* WAIT FOR AN OP END BEFORE RESCHEDULING THE LINE 04750000
SPACE 04760000
CMBTAC EQU * * LOCAL B 04770000
L CMSDTF,DTF POINT XR2 AT LCB B 04810000
TBN LCBAT2(,DTF),LCBACT LINE ACTIVE ? B 04820000
JF CMBSCH NO-GO SCHEDULE WORK ON LINE. B 04830000
TBN LCBAT2(,DTF),LCBRCI RECEIVE INITIAL ON LINE ? B 04840000
JF CMTPLQ NO-GO CHECK PARM LIST Q'D. B 04850000
TBN PL$OPM(,PL),OPPUT THIS A PUT REQUEST ? B 04860000
JT CMTRYC YES-GO TO TRY PRIORITY CANCEL. B 04870000
SPACE 1 04880000
***** B 04890000
* READ REQUEST - LINE BUSY WITH RECEIVE INITIAL * B 04900000
***** B 04910000
SPACE 04920000
ST CMSPL,PL SAVE PARM LIST ADDR B 04926000
B CMIVGM GO ASSURE SIZE FOR NEW READ. B 04930000

```

	SPACE			04940000
	L	CMSPL,PL	POINT TO PARM LIST	B 04941000
	TBN	PL\$OPM(,PL),OPGETM	GETMAIN ALREADY INDICATED ?	B 04942000
	JT	CMNOID	YES - DON'T START 2 SEC T.O.	B 04943000
	L	\$BDWKA(,DTF),WKA	XR1--> BSC WORK AREA	B 04944000
	TBN	\$BWKMC(,WKA),X'01'	IDA RUNNING NOW ?	B 04945000
	JT	CMTRYC	YES - GO TRY PROIRITY CANCEL	B 04946000
CMNOID	EQU	*	*	B 04947000
	B	CMPAII	GO TO POST RESULTS.	B 04950000
	SPACE	1		04960000
	*****			B 04970000
	* LINE BUSY WITH NON-RECEIVE INITIAL *			B 04980000
	*****			B 04990000
	SPACE			05000000
CMTPLQ	EQU	*	* LOCAL	B 05010000
	TBN	LCBAT1(,DTF),LCBNTQ	PARM LIST NOT QUEUED, AND	B 05020000
	L	PLTUBA(,PL),XR2	POINT TO THE TUB.	B 05030000
	TBN	TUBAT2(,XR2),TUBOWN	* THIS REQUEST FOR LINE OWNER?	B 05040000
	L	TUBDTF(,XR2),DTF	POINT TO THE DTF.	B 05050000
CMTOP2	TBN	PL\$OPM(,PL),#	SAME OP AS ON THE LINE ?	B 05060000
	JF	CMBBSY	NO-GO TO CHECK LINE FOR BUSY.	B 05070000
	SPACE	1		05080000
	-----			B 05090000
	* ANOTHER BLOCK OR RECORD OPERATION TO THE OWNED TERMINAL. *			B 05100000
	* IF A READ JUST GO SCHEDULE IT ON THE LINE (POLL IT). *			B 05110000
	-----			B 05120000
	SPACE	1		05130000
	SBF	LCBAT1(,DTF),LCBNTQ	PARM LIST IS BACK IN QUE.	B 05140000
	TBN	PL\$OPM(,PL),OPGET	IF A READ	B 05150000
*			-----START-----@25	05150300
	JF	CMNNGT	GO DO PUT OPERATION	B 05150600
	TBN	LCBAT1(,DTF),LCBINT	OP END WITH NO PARM LIST QUE'D	B 05150650
	SBF	LCBAT1(,DTF),LCBINT	SET OFF OP END QUE'D IND.	B 05150700
	JF	CMNPLQ	NO-DON'T FAKE AN OP END.	B 05150750
	ALC	#OPEND(1),X\$0001	UP OP END COUNT BY ONE.	B 05150800
	ALC	LCBOPE(1,DTF),X\$0001	BUMP LINE OP END COUNT.	B 05150850
CMNPLQ	EQU	*	* LOCAL.	B 05150900
	CLI	PLRECA-1(,PL),NOBIT	GETMAIN NEEDED ?	B 05151000
	JNE	CMNOBY	GO SCHEDULE GET OPERATION.	B 05153000
*			-----END-----@25	05154500
CMNYGM	EQU	*	* LOCAL	B 05155000
	ST	CMSPL,PL	SAVE PARM LIST ADDR	B 05156000
	B	CMIVGM	GO ASSURE SIZE FOR NEW GET.	B 05157000
	SPACE			05158000
	L	CMSPL,PL	POINT TO PARM LIST	B 05159000
	J	CMNOBY	GO SCHEDULE GET OPERATION.	B 05160000
CMNNGT	EQU	*	* LOCAL	B 05165000
	SPACE	1		05170000
	-----			B 05180000
	* ANOTHER PUT BLOCK OR RECORD - GETMAIN AND START OPERATION *			B 05190000
	-----			B 05200000
	SPACE			05210000
	L	PLTUBA(,PL),XR2	TUB ADDRESS	B 05220000
	TBN	TUBTA1(,XR2),TASDF	IF DFF SKIP GETMAIN	B 05230000
	JT	CMNOBY	YES - ALREADY HAVE BUFFER	B 05240000
	TBN	PL\$OPM(,PL),OPNOW	IF NO WAIT	B 05250000
	JT	CMNOBY	YES - ALREADY HAVE BUFFER	B 05260000
	SPACE	1		05270000
	B	CMGMPT	GETMAIN BUFFER, MOVE DATA IN	B 05280000
	SPACE			05290000

```

J      CMNOBY          GO START OPERATION          B 05300000
SPACE 3          05310000
*****          B 05320000
*      BSCA   TP LINE BUSY - DETERMINE IF ABORT NEEDED          * B 05330000
*****          B 05340000
SPACE          05350000
* LINE IS BUSY AND MAY HAVE MORE HONORABLE REQUEST WAITING IN THE Q B 05360000
SPACE          05370000
* IF PREVIOUS OP WAS READ - TRY TO ABORT          B 05380000
* DON'T CARE IF STARTED OR NOT BECAUSE RETURN CODE OF 44 OR NON-44 B 05390000
* WILL REFLECT THIS.          B 05400000
SPACE          05410000
CMBBSY EQU      *          * LOCAL          B 05420000
TBN   LCBAT2(,DTF),LCBRCI+LCBPUT RECEIVE INIT AND PUT PENDING. B 05430000
BF    CMPAII          NO-GO POST AS NECESSARY. B 05440000
SPACE          05450000
*****          B 05460000
*      PUT REQUEST - LINE BUSY WITH RECEIVE INITIAL - PRIORITY CANCEL B 05470000
*****          B 05480000
SPACE          05490000
CMTRYC EQU      *          *          05490400
&MIX  SETA  &NPP+&NMP+&NSWL          05490800
AIF   (&NCS).S0750          05491200
* FOR CONTROL STATION LINES PERFORM THE CANCEL FROM RESIDENT CODE C/LB 05492000
* FOR OTHER LINE TYPES - CALL $CC4BC TO STOP THE LINE          C/LB 05492400
SPACE 1          05492800
AIF   (&MIX EQ '3').S0760          05493200
TBN   $BDATR(,DTF),$BCMCN          CONTROL STATION LINE?          C/LB 05493600
JF    CMNCSL          NO-CALL $CC4BC          C/LB 05494000
.S0760 ANOP          05494400
TBF   LCBAT1(,DTF),LCBCRI+LCBPRI CANCEL ALREADY STARTED?          CB 05494800
SBN   LCBAT1(,DTF),LCBPRI          SET CANCEL STARTED          CB 05495200
JF    CMGPST          YES-JUST GO POST          CB 05495600
L     $BDWKA(,DTF),WKA          XR1--> BSC WORK AREA          CB 05496000
AIF   (&NBDA).S0761          05496020
*          -----START-----@16 05496040
TBN   $BWKMC(,WKA),X'01'          IDA RUNNING NOW ?          CB 05496080
JF    CMNIDA          NO - GO SET CANCEL FOR MLMP          CB 05496120
SBN   $BPOLD(,WKA),$BPRES          SET CANCEL FLAG FOR MLMP          CB 05496160
DC    XL3'F38806'          DO SIO TO START IDA STOP POLL          CB 05496200
J     CMNODA          JUMP OVER CANCEL SETTING          CB 05496240
CMNIDA EQU      *          *          CB 05496280
*          -----END-----@16 05496320
.S0761 ANOP          05496360
SBN   $BPOLD(,WKA),$BPRES          SET CANCEL FLAG FOR MLMP          CB 05496400
CMNODA EQU      *          * LOCAL          CB 05496600
AIF   (&MIX EQ '3').S0770          05496800
J     CMGPST          GO POST          CB 05497200
.S0750 ANOP          05497600
AIF   (&MIX EQ '3').S0770          05498000
CMNCSL EQU      *          NOT CONTROL STATION LINE          C/LB 05498400
SVC   0          ##### TRANSIENT CALL #####          B 05500000
DC    AL1(CCPRIB)          CCP SVC RIB          B 05510000
DC    AL1(CC4BC)          CANCEL OPERATION TRANSIENT.          B 05520000
DC    AL1(OPPUT)          REFRESH OP CODE VALUE.          B 05530000
*          LINKAGE INDICATOR TO 'BC', CALLB 05540000
*          IS FROM RESIDENT CODE.          B 05550000
*          TRANSIENT ONLY RETURNS HERE          B 05560000
.S0770 ANOP          05561000
CMGPST EQU      *          *          B 05562000

```



```
AIF      (&NCS) .S0780      05563000
.S0780 ANOP      05565000
      B      CMPAII      GO TO POST USER.      B 05570000
.C0470 ANOP      05580000
      MEND      05590000
```

```

MODULE-$E092 , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO 00010000
***** 00020000
.* $E092 * 00040000
***** 00050000
$E092 00060000
  GBLB &ONE, &NOB, &NOM, &MIN, &NDME, &N37, &NSCTL, &NBDA, &NPBY 000670000
  GBLB &NPP, &NMP, &NSWL, &NCS, &N32, &NDF, &NRUF, &N41 000680000
  GBLB &NSW, &N1050, &N2741, &NMOVE, &NSCTL, &NBFR 000700000
.* ARITHMETIC GLOBALS 000710000
  GBLA &MLA, &BSC 000720000
.* ARITHMETIC LOCALS 000730000
  LCLA &MIX 000740000
  TEXT 000750000
* R-08, C-00 CHANGE LEVEL 000760000
  TITLE '$E092 --- COMMON BSCA/MLTA SUBROUTINES' 000770000
  SPACE 15 000780000
***** 000790000
* * 000800000
* * 000810000
* * 000820000
* COMMON M L T A / B S C A SUBROUTINES * 000830000
* * 000840000
* * 000850000
* * 000860000
***** 000870000
  TITLE '$E092/CMPRLS---SEARCH NEW REQUEST PL QUEUE' 000880000
***** 000890000
* * 000900000
* NAME--CMPRLS * 000910000
* * 000920000
* TITLE--SEARCH NEW REQUEST PARAMETER LIST QUEUE FOR NEW REQUEST. * 000930000
* * 000940000
* FUNCTION-- * 000950000
* * 000960000
* FIND FIRST NEW PUT ON PARAMETER LIST QUEUE AND * 000970000
* AND RETURN ITS ADDRESS. IF NO NEW PUTS RETURN ADDRESS * 000980000
* OF FIRST INPUT PARAMETER LIST. * 000990000
* * 000990000
* OUTPUT * 000990000
* * 01000000
* XR1 - ADDRESS OF PL TO HANDLED NEXT. * 01000000
* * 01010000
* XR2 - ADDRESS OF POINTER TO PL TO HANDLED. * 01010000
* * 01020000
***** 01030000
  SPACE 1 01040000
CMPRLS EQU * ONLY ENTRY POINT 01050000
  ST CMPRLX+3, ARR SAVE RETURN ADDRESS 01060000
  SBF $FLGC, #NTRAC SET OFF NO TRACE IND 01065000
  LA @PRLQ-1, XR2 LEFT BYTE OF POINTER TO 1ST PL. 01070000
CMCKNX EQU * * LOCAL 01080000
  L PLCHN(, XR2), PL @ OF NEXT PARAMETER LIST IN Q 01090000
  TBN PLOPC(, PL), OPPUT IS OP A PUT ? 01100000
  TBF PLOPM(, PL), OPSTOP * AND NOT STOP ? 01110000
  JT CMPRLX YES - RETURN IT TO CALLER. 01120000
  LA 0(, PL), XR2 ADDR OF CHAIN ADDRESS IN PL 01130000
  CLI 0(, XR2), NOBIT ANY MORE PARM LISTS ON QUEUE ? 01140000
  JE CMRDPL NO - NO PUTS, CHECK FOR READ 01150000
  B CMCKNX YES - CHECK THE NEXT ONE FOR PUT 01160000
  SPACE 1 01170000
CMRDPL EQU * * LOCAL 01180000
  L @PRLQ, PL ADDRESS OF FIRST PL ON QUEUE 01190000
* * 01200000
* RETURN IT TO USER.

```

LA	@PRLQ-1,XR2	ADDRESS OF POINTER TO IT.	01210000
CMPRLX B	##		01220000
TITLE	'\$E092/CMWPGY---COMMON PUT THEN GET OP END'		01230000
*****			01240000
*			01250000
* NAME--CMWPGY			01260000
*			01270000
* TITLE--COMMON OP END ROUTINE FOR PUT PART OF PUT THEN GET.			01280000
*			01290000
* FUNCTION--			01300000
* WHEN THE PUT PART HAS OP ENDED, SCHEDULE A READ			01310000
* OPERATION USING THE SAME PARAMETER LIST THAT WAS			01320000
* USED FOR THE WRITE.			01330000
*			01340000
*****			01350000
SPACE 1			01360000
CMWPGY EQU *			01370000
ST CMWPSV+3,ARR		SAVE RETURN ADDRESS	01380000
SPACE			01390000
* REFORMAT THE PARAMETER LIST FOR THE READ PORTION			01400000
SPACE			01410000
SLC PL\$RTC(2,PL),PL\$RTC(,PL)		CLEAR INTERNAL RETURN CODE	01420000
SBF PL\$OPM(,PL),OPPUT		SET OFF PUT BIT	01430000
SBN PL\$OPM(,PL),OPGET		SET ON GET BIT	01440000
SPACE			01450000
TBF PLOPC(,PL),OPNOW		USER PUT THEN GET ?	01460000
JT CMNWFM		YES - JUST FREEMAIN, NO POST.	01470000
SPACE 2			01480000
TBN PL\$OPM(,PL),OPNOW		IF STILL A NO WAIT OPERATION	01490000
JT CMNWFM		JUMP IF YES TO FREEMAIN	01500000
SPACE 1			01510000
*****			01520000
* PUT WAIT - PART OF SYSTEM PUT NO WAIT INVITE CHANGED TO			01530000
* PUT WAIT INVITE.			01540000
*****			01550000
SPACE			01560000
* SBN PL\$OPM(,PL),OPNOW		SET NO WAIT BIT ON IN INTERNAL	01570000
		OP CODE MODIFIER FOR INVITE.	01580000
LA PLECB(,PL),XR1		ECB FOR THIS REQUEST	01590000
SVC 0		POST THIS ECB	01600000
DC AL1(POSTRB)		* POST *	01610000
SPACE 1			01620000
L CMSPL,PL		RESTORE PL ADDRESS	01630000
J CMWPGE		JUMP TO RESTART THE LINE	01640000
SPACE 2			01650000
*****			01660000
* PUT WAIT - PART OF USER PUT THEN GET -- JUST FREE PUT BUFFER *			01670000
* PUT NO WAIT - PART OF SYSTEM PUT NO WAIT INVITE			01680000
*****			01690000
SPACE			01700000
CMNWFM EQU *		FREEMAIN PUT NO WAIT AREA	01710000
L PLRECA(,PL),XR2		POINT XR2 AT RECORD AREA OF	01720000
*		AREA TO BE FREEMAINED.	01730000
B CMFMRT		BR TO FREEMAIN INTERFACE ROUT.	01740000
SPACE 1			01750000
CMWPGE EQU *		*	01755000
MVI PLRECA-1(,PL),NOBIT		ZERO HOLD BUFFER ADDRESS	01760000
CMWPSV B *-*		RETURN	01765000
\$CC4CM TITLE '\$E092/CMPOST---POST-ANALYSIS'			01780000
*****			01790000

```

* 01800000
* NAME--CMPOST 01810000
* 01820000
* TITLE--COMMON POST ROUTINE 01830000
* 01840000
* FUNCTION-- 01850000
* POST THE REQUESTOR THAT HIS REQUEST WAS 01860000
* COMPLETED AND FREE UNNEEDED GETMAINED AREAS. 01870000
* 01880000
* OPERATION: 1. IF REQUEST IS A USER INVITE, POST TCBE CB FOR OWNING 01890000
* TASK THAT THE OPERATION IS COMPLETE. 01900000
* 01910000
* 2. IF REQUEST IS SYSTEM INVITE OR COMMAND INTERRUPT 01920000
* GET POST $CC4CP. 01930000
* 3. IF REQUEST IS AN INVITE OR COMMAND INTERRUPT GET 01940000
* PUT TUB ON TCBIHQ. 01950000
* 4 IF REQUEST IS A NON COMMAND INTERRUPT GET OR A PUT 01960000
* POST PLECB OF THE REQUESTS PARAMETER LIST. 01970000
* 5. FREE ALL UNNEEDED GETMAINED AREAS. 01980000
***** 01990000
SPACE 02000000
CMPOST EQU * POST THE PROPER TCB 02010000
ST CMMKRT+3,ARR RETURN ADDRESS 02020000
SPACE 02030000
TBN PL$OPM(,PL),OPPNW PUT NO WAIT 02040000
JT CMPSFM YES - NO POST, JUST FREEMAIN 02050000
SPACE 1 02060000
L PLTUBA(,PL),XR2 POINT XR2 AT TUB 02070000
SPACE 02080000
* CHECK NO POST BIT TO SEE IF NO TCB IS TO BE POSTED DUE TO THE 02090000
* COMPLETION OF THIS OPERATION 02100000
SPACE 02110000
AIF (&NOB).C1400 02120000
TBN PL$OPM(,PL),OPBNOP IS NO POST BIT ON ? B 02130000
BT CMMKRT JUMP IF NO POST BIT IS ON. B 02140000
.C1400 ANOP 02150000
AIF (&NOM).M1400 02160000
TBN PL$OPM(,PL),OPNPST IF PURGE OF MLTA PUT M 02170000
JT CMPSFM JUST FREE BUFFER M 02180000
.M1400 ANOP 02190000
SPACE 02200000
* IF REQUESTING TASK TERMINATED DO NOT POST - THIS IS 02210000
* DETERMINED BY OP BEING A USER OPERATION BUT TERMINATION 02220000
* SYSTEM TASK TCB IS NOW POINTED TO BY TUBTCB. 02230000
SPACE 02240000
L TUBTCB(,XR2),XR2 POINT XR2 AT TUB TCB 02250000
TBN TCDFG1(,XR2),TCBUSR+TCBNCL IS IT A SYSTEM TASK 02260000
TBF PLOPM(,PL),OP$SYS IS IT USER REQUEST 02270000
* ----- START -@30 02275000
JT CMJTFR YES- NO POST BUT FREE AREAS 02280000
* ----- END ---@30 02280500
* IF TASK IS IN TERMINATION,DON'T QUEUE UP INPUT 02283000
TBN TCDFG2(,XR2),TCBTRC TASK IN TERMINATION? 02285000
JT CMJTFR YES-GO FREE(DON'T POST) 02286000
SPACE 02290000
***** 02300000
* OKAY TO POST 02310000
***** 02320000
SPACE 02330000
* DETERMINE WHICH TCB TO POST 02340000

```

```

SPACE 1 02350000
ST PSTCB, XR2 SET DEFAULT TO POST TCB FROM TUB 02360000
SPACE 1 02370000
L PLTUBA(, PL), XR2 POINT XR2 AT TUB 02380000
AIF (&NPBY).NBY04 SKIP IF NO BUSY PRINTER 02380600
* 02381200
* BUSY PRINTER SUPPORT 02381800
* 02382400
***** 02383000
* IF THIS IS THE SYSTEM INVITE FROM THE BUSY PRINTER CODE * 02383600
* NO FURTHER CHECK IS NECESSARY * 02384200
***** 02384800
SPACE 1 02385400
TBN TUBAT4(, XR2), TUBWAT THIS TASK WAITING ON PRINTER? 02386000
SBF TUBAT4(, XR2), TUBWAT SET OFF WAITING BIT 02386600
JT CMPSPL YES- GO POST PLECB. 02387200
SPACE 1 02387800
.NBY04 ANOP 02388400
AIF (&NDME).D5200 02390000
TBN TUBAT2(, XR2), TUBDTA+TUBCMD IF COMMAND INTERRUPT MODE D 02400000
* ----- START -@26 02401000
JT CMPCIM YES- HANDLE CMD INT MODE. D 02402000
.D5200 ANOP D 02403000
TBN PL$OPM(, PL), OPINV WAS OPERATION A SYSTEM D 02404000
TBN PLOPM(, PL), OP$SYS * INVITE INPUT ? D 02405000
JF CMPSII NO-POST $CC4II D 02406000
CMPCIM EQU * * LOCAL D 02407000
* ----- END ---@26 02408000
* TERMINAL IN COMMAND INTERRUPT MODE OR SYS INVITE - POST CP TSB. D 02430000
SPACE 02440000
TBN PL$OPM(, PL), OPGET DID OP INVOLVE A READ D 02460000
MVC PSTCB(2), @CPTCB USE C.P.'S TCB FOR Q'ING 02465000
JT CMQINQ IF YES, PUT REQUEST ON TCBINQ D 02470000
J CMPSPL NO - JUST POST CP, MUST BE D 02480000
* * SYS PUT NO WAIT MADE WAIT. D 02490000
SPACE 02500000
* DETERMINE WHO TO POST. 02530000
SPACE 02540000
CMPSII EQU * DETERMINE WHAT ECB TO POST 02550000
TBN PL$OPM(, PL), OPINV WAS OP AN INVITE INPUT 02560000
TBF CMSWIT, CMSPSI STOP INV AND STATUS POLL 02570000
JF CMTSGT IF NOT INVITE-CHECK FOR GET. 02580000
SPACE 02590000
***** 02600000
* INVITE INPUT OR COMMAND INTERRUPT READ - PUT TUB ON TCBINQ * 02610000
***** 02620000
SPACE 02630000
CMQINQ EQU * * LOCAL 02640000
SPACE 02650000
ST TUBPL@(, XR2), PL PL ADDRESS INTO TUB 02660000
SBN TUBAT2(, XR2), TUBIIQ+TUBIIS INVITE DATA ON TCBINQ FOR THIS 02670000
* * TERMINAL. INPUT STILL OUT- 02680000
* * STANDING UNTIL ACCEPT RECVD. 02690000
SPACE 02700000
MVI TUBINQ-1(, XR2), NOBIT ZERO THIS TUB'S CHAIN POINTER 02710000
L PSTCB, XR2 POINT XR2 AT TCB 02720000
CLI TCBINQ-1(, XR2), NOBIT ARE THERE ANY TUBS IN THE QUEUE 02730000
JNE CMTUBC IF TUBS IN QUEUE - JUMP 02740000
SPACE 02750000
* NO TUBS AT ALL IN THE QUEUE 02760000

```



```

* ANY PUT OR NON-COMMAND INTERRUPT GET - POST PLECB * 03300000
***** 03310000
SPACE 1 03320000
CMTSGT EQU * * LOCAL 03330000
TBN PL$OPM(,PL),OPGET * GET OPERATION ? 03340000
TBF PLOPM(,PL),OP$SYS IS THIS A USER 03350000
* -----START-----@3- 03353000
CLI PL$RTC(,PL),RCXEDT AND RET. CODE GREATER THAN 3 03356000
JC CMPSPL,FLSOHI YES OR SYSTEM OP - JUST POST 03360000
* -----END-----@3- 03365000
SPACE 1 03370000
* -----* 03380000
* USER GET -- MOVE DATA FROM HOLD BUFFER TO USERS RECORD AREA * 03390000
* -----* 03400000
SPACE 1 03410000
MVI #CMMVL+MVLTY,SWAPTO INDICATE TO ADDR IN USER PGM 03420000
MVC #CMMVL+MVLTO,PLINL(2,PL) DESIRED LENGTH IS TO LENGTH 03430000
MVC #CMMVL+MVLFR,PLRECA(2,PL) HOLD BUFFER ADDR IS FROM ADDR 03440000
L PLTUBA(,PL),XR2 TUB ADDRESS 03450000
L TUBTCB(,XR2),XR2 TCB ADDRESS 03460000
MVC #CMMVL+MVLTOA,TCBWK(2,XR2) USERS RECORD AREA ADDRESS 03470000
SPACE 1 03480000
B CMMVRT INTERFACE TO SYSTEM MOVE ROUTINE 03490000
SPACE 1 03500000
***** 03510000
* ANY PUT OR NON-COMMAND INTERRUPT GET - POST PLECB * 03520000
***** 03530000
SPACE 1 03540000
CMPSPL EQU * * LOCAL 03550000
LA PLECB(,PL),XR1 ADDRESS OF ECB IN PARM LIST 03560000
SPACE 1 03570000
CMPSOE EQU * * LOCAL 03580000
SVC 0 POST ECB 03590000
DC AL1(POSTRB) * POST * 03600000
SPACE 1 03610000
***** 03620000
* FREEMAIN PARAMETER LIST AND HOLD AREAS AS NECESSARY * 03630000
***** 03640000
SPACE 1 03650000
CMPSFM EQU * * LOCAL 03660000
L CMSPL,PL PL ADDRESS 03670000
TBN PL$OPM(,PL),OPINV INVITE REQUEST ? 03680000
JT CMMKRT YES- NO FREEMAIN NEEDED 03690000
* * MUST BE PRESERVED UNTIL AN 03700000
* * ACCEPT OR STOP INVITE. 03710000
SPACE 1 03720000
* -----* 03730000
* GET OR PUT -- FREE HOLD BUFFER AND PARAMETER LIST * 03740000
* -----* 03750000
SPACE 1 03760000
L PLTUBA(,PL),XR2 TUB ADDRESS 03770000
L TUBDTF(,XR2),DTF DTF ADDRESS 03780000
AIF (&NOB).C1900 03790000
TBN $BDDEV(,DTF),BSCA BSCA DTF AND B 03800000
TBN PLOPM(,PL),OPSTOP STOP INVITE AND B 03804000
TBN LCBOPC(,DTF),LCBERP IF WAITING FOR EOT TO STATUS B 03810000
JT CMMKRT YES - EXIT, NO FREEMAIN B 03820000
* IF THE ERROR OCCURED ON A SYSTEM OP - C.P. WILL FREE THE AREA B 03822000
TBN LCBOPC(,DTF),LCBERP ERROR ON - B 03824000
TBN PLOPM(,PL),OP$SYS A SYSTEM OP ? B 03826000

```

JT	CMMKRT	YES - NO FREE NEEDED	B	03828000
.C1900	ANOP			03830000
	SPACE	1		03840000
CMJTFR	EQU	*	*	03845000
	B	\$CC4FR	FREE GETMAINED AREA FOR REQUEST	03850000
	SPACE	1		03860000
CMMKRT	B	*-*	RETURN	03870000
	TITLE	'\$E092/CMIVGM---INVITE-BUFFER-ANALYSIS'		03880000
*****				03890000
*			*	03900000
*	NAME--	CMIVGM		03910000
*			*	03920000
*	TITLE--	INPUT OPERATION BUFFER ANALYSIS		03930000
*			*	03940000
*	FUNCTION--	ANALYSIS AND OBTAINING OF GETMAIN BUFFER FOR ALL		03950000
*		INPUT OPERATIONS IN THE LINE QUEUE THAT WE CAN GET		03960000
*		BUFFER SPACE FOR.		03970000
*	OPERATION--			03980000
*		. IF BSCA POLL FOR STATUS IS IN THE QUEUE SET ALL		03990000
*		POLLING SKIP BITS ON.		04000000
*		. IF LCB CURRENTLY HAS A HOLD AREA FREEMAIN IT.		04010000
*		. IF GET OPERATION TO BE SCHEDULED SET UP TO HANDLE		04020000
*		IT.		04030000
*		. IF SYSTEM INVITE TO BSCA SWITCHED LINE OWNED BY A		04040000
*		USER PROGRAM, THEN IGNORE THE SYSTEM REQUEST FOR		04050000
*		NOW.		04060000
*		. DETERMINE STORAGE REQUIREMENTS TO SCHEDULE EACH		04070000
*		INVITE OPERATION.		04080000
*		. IF STORAGE NOT AVAILABLE IGNORE THE REQUEST.		04090000
*		. IF STORAGE IS AVAILABLE SET UP TO HANDLE IT.		04100000
*		. IF BSCA POLL FOR STATUS, SCHEDULE IT BY ITSELF.		04110000
*		. GETMAIN THE SPACE NEEDED TO SCHEDULE OPERATIONS ON		04120000
*		THE LINE.		04130000
*		. SET UP THE LCB HOLD AREA TO REFLECT THE SPACE		04140000
*		ACQUIRED.		04150000
*		. SET UP GETMAIN PARAMETER LIST IN THE ACQUIRED AREA.		04160000
*				04170000
*	ENTRY POINT--	CMIVGM		04180000
*				04190000
*	INPUT--	AT LEAST ONE PARM LIST IN THE LINE QUEUE.		04200000
*				04210000
*	OUTPUT--	XR1-ADDRESS OF THE LAST TP PARAMETER LIST IN THE QUEUE.		04220000
*		XR2-ADDRESS OF THE DTF.		04230000
*		CMSPL-ADDRESS OF LAST GET/INVITE PARAMETER LIST NOT IN		04240000
*		CCPERP.		04250000
*				04260000
*	EXTERNAL REFERENCES--			04270000
*		CMFMR2-FREE LCB HOLD BUFFER BEFORE STARTING		04280000
*		ANALYSIS.		04290000
*		CMSTOR-DETERMINE STORAGE NEED FOR INVITES.		04300000
*		CMBSKP-SET BSCA POLL SKIP BITS ON/OFF.		04310000
*		CMONSK/CMOFSK-SET MLTA POLL SKIP BITS		04320000
*		ON/OFF.		04330000
*		CMGMRT-GETMAIN HOLD BUFFER TO SCHEDULE		04340000
*		INVITES ON THE LINE.		04350000
*				04360000
*	EXIT, NORMAL--	TO NSI OF CALLER.		04370000
*				04380000
*	EXIT, ERROR--	TO CPHALT WITH A 4 BLITZ HALT IF THE GETMAIN FAILS.		04390000
*				04400000


```

***** 04410000
EJECT 04420000
CMIVGM EQU * GETMAIN ANALYSIS OF INVITE 04430000
* INPUTS IN LINE QUEUE 04440000
ST CMIVGX+3,ARR SAVE RETURN ADDRESS 04450000
L CMSDTF,DTF POINT XR2 AT THE LCB 04460000
SBF CMSWIT,CMRSLN SET OFF INDICATION THAT LINE 04470000
* CAN BE SCHEDULED FOR READ 04480000
SPACE 04490000
* POINT TO FIRST PARAMETER LIST IN THE LINE QUEUE 04500000
SPACE 04510000
L LCBPLQ(,DTF),PL POINT XR1 AT 1ST PARM LIST 04520000
AIF (&N32).T2000 04530000
SPACE 1 04540000
* IF BSCA POLL FOR STATUS ON TOP OF THE QUEUE, THEN SET ALL SKIP 0B 04550000
* BITS ON IN THE POLLING LIST. 0B 04560000
SPACE 1 04570000
TBN PL$OPC(,PL),OPLSNS POLLING FOR STATUS, AND 0B 04580000
TBN $BDDEV(,DTF),BSCA BSCA LINE, AND 0B 04590000
TBN $BDATR(,DTF),$BCMCN CONTROL STATION LINE ? 0B 04600000
JF CMINOR NO-GO HANDLE NORMAL REQUESTS. 0B 04610000
L LCBPOL(,DTF),XR1 POINT TO THE POLLING LIST. 0B 04620000
CMIALL EQU * * 0B 04630000
CLI POLID(,XR1),POLEND END OF LIST ? 0B 04640000
JNL CMINOR YES-GO TO NORMAL PROCESSING. 0B 04650000
MVC CMISTS+2,POLCNT(1,XR1) SET UP REG UPDATE. 0B 04660000
CMISTS LA #(,XR1),XR1 UPDATE FOR POLL CHAR COUNT. 0B 04670000
SBN 2(,XR1),POLSKP SET ON THE SKIP BIT. 0B 04680000
LA POLNXT(,XR1),XR1 POINT TO NEXT ENTRY. 0B 04690000
B CMIALL GO CHECK NEXT ENTRY. 0B 04700000
SPACE 1 04710000
CMINOR EQU * * 0B 04720000
L LCBPLQ(,DTF),PL POINT TO FIRST PARM LIST. 0B 04730000
.T2000 ANOP 04740000
SLC CMIIND(2),CMIIND ZERO LINE BUFFER NEEDED SIZE 05010000
SPACE 05020000
* DETERMINE IF OPERATION IS A READ 05030000
SPACE 05040000
CMDTII EQU * * LOCAL (LOOP BACK FROM CMINPL 05050000
TBN PL$OPM(,PL),OPGET DOES OP INVOLVE READ 05060000
JF CMNXPL JUMP IF NO READ 05070000
SPACE 05080000
* HAVE READ - NOW ASSURE THE TERMINAL IS NOT IN ERP 05090000
SPACE 05100000
L PLTUBA(,PL),XR2 POINT XR2 ATHE TUB 05110000
TBF TUBAT3(,XR2),TUBERP IS TERMINAL IN ERROR RECOVERY 05120000
L CMSDTF,DTF POINT XR2 BACK AT LCB 05130000
JF CMNXPL JUMP IF TERMINAL IN ERP OR WAIT 05140000
SPACE 05150000
AIF (&NSWL).S0470 05160000
***** SB 05170000
* IGNORE BSCA SYS PARM LIST IF USER OWNS THE SWITCHED LINE. * SB 05180000
***** SB 05190000
SPACE 1 05200000
AIF (&ONE).E0200 05210000
TBN $BDDEV(,DTF),BSCA BSCA DTF ? S2 05220000
.E0200 ANOP 05230000
&MIX SETA &NMP+&NPP+&NCS 05240000
AIF (&MIX EQ '3').S0450 05250000
TBN $BDATR(,DTF),$BCSWI SWITCHED LINE, AND SLB 05260000

```

```

.TBFBF $BDATR(,DTF), $BCMPT * NOT MULTI-POINT LINE ? SLB 05270000
.S0450 ANOP 05280000
JF CMGMLT NO - HANDLE OP W/O MORE CHKS SB 05290000
TBN LCBATR(,DTF),LCBNIT LINE CONNECTED ? SB 05300000
JT CMCOWN YES- CHECK OWNING TUB SB 05310000
SPACE 1 05320000
* LINE NOT CONNECTED - ALLOW REQUEST UNLESS SYSTEM REQUEST AND SB 05330000
* USER TASK OWNS THE LINE. SB 05340000
SPACE 1 05350000
CLC LCBTCB(2,DTF),X$0000 SWITCH LINE OWNED ? SB 05360000
JE CMGMLT NO - HANDLE OPERATION NOW. SB 05370000
CLC LCBTCB(2,DTF),@CPTCB USER TASK OWN LINE ? SB 05380000
TBN PLOPM(,PL),OP$SYS SYSTEM REQUEST ? SB 05390000
JC CMGSBN,TRUNEQ YES-SKIP THIS PARM LIST. SB 05400000
J CMGMLT NOT CONNECTED SO SCHEDULE SB 05410000
SPACE 1 05420000
* LINE CONNECTED - CHECK TO SEE IF REQUEST IS FOR CONNECTED TUB SB 05430000
SPACE 1 05440000
CMCOWN EQU * * LOCAL SB 05450000
CLC PLTUBA(2,PL),LCBOWN(,DTF) YES - THIS TUB OWN THE LINE ? SB 05460000
JNE CMGSBN NO - LINE OWNED BY ANOTHER SB 05470000
* * TUB SO SKIP THIS PL SB 05480000
CMGMLT EQU * * LOCAL SB 05490000
.S0470 ANOP 05500000
SPACE 05500300
* --START-----@29 05500600
* DOES THIS LCB CURRENTLY POSSESS A GETMAIN AREA FOR ITS INPUTS. 05500900
* IF YES - FREEMAIN IT SO THAT THERE WILL BE A BETTER CHANCE OF 05501200
* GETTING ENOUGH STORAGE TO HANDLE ALL THE OUTSTANDING INVITE INPUTS 05501500
SPACE 05501800
CLI LCBIBA-1(,DTF),NOBIT IS INVITE BUFFER ADDR NULL 05502100
JE CMSZND JUMP IF NULL 05502400
SPACE 05502700
* HAVE GETMAIN AREA WHICH WE WILL NOW FREEMAIN 05503000
* GET ADDRESS OF RECORD AREA IN THE STORAGE AREA TO BE FREEMAINED 05503300
* FREEMAIN THE LCB HOLD BUFFER 05503600
SPACE 05503900
L LCBIBA(,DTF),XR2 POINT XR2 AT RECORD TO BE FM 05504200
B CMFMR2 BRANCH TO ISSUE FREEMAIN 05504500
* WE WILL GET BACK AT LEAST THIS 05504800
* MUCH SO BYPASS SETTING FREEMAIN 05505100
* DONE FLAG. 05505400
SPACE 05505700
* REFLECT THE FREEMAIN BY ZEROING THE LCB FIELD WHICH CONTAINS THE 05506000
* LENGTH OF THE LCB HOLD BUFFER 05506300
SPACE 05506600
L CMSDTF,DTF POINT XR2 AT THE LCB 05506900
MVI LCBIBA-1(,DTF),NOBIT ZERO HIGH ORDER BYTE OF BUFFER 05507200
* TO INDICATE NULLL BUFFER ADDR 05507500
SPACE 05507800
CMSZND EQU * SET SIZE NEEDED NOW TO ZERO 05508100
* --END-----@29 05508400
B CMSTOR BR TO DETERMINE SIZE NEEDED BY 05510000
* THIS TP REQUEST 05520000
* SIZE NEEDED RETURNED IN #BUFND 05530000
SPACE 05540000
* COMPARE WHAT IS NEEDED AGAINST WHAT IS AVAILABLE WITH A GETMAIN 05550000
SPACE 05560000
CLC #BUFND,#ANYS+1 COMPARE NEEDED VS AVAILABLE 05570000
JNH CMGMAV JUMP IF ENOUGH STORAGE AVAILABLE 05580000

```

```

SPACE 05590000
***** 05600000
*THERE IS NOT ENOUGH STORAGE AVAILABLE IN THE TP BUFFER TO 05610000
* SATISFY THIS READ -- CHECK THE NEXT ONE ON THE QUEUE 05620000
***** 05630000
SPACE 05640000
* SET GETMAIN NEEDED BIT ON 05650000
SPACE 05660000
SBN $CMFM,SKIP INDICATE NO POST FOR FREEMAIN 05666000
SBN LCBATR(,DTF),LCBGMN SET LCB GETMAIN NEEDED 05680000
TBN PL$OPM(,PL),OPGETM GETMAIN ALREADY INDICATED? 05682000
JT CMNDCR YES - 05684000
SBN PL$OPM(,PL),OPGETM SET GETMAIN NEEDED FOR PL 05686000
ALC CORCNT(1),X$0001 UP CORE COUNT +1 05688000
* ----- START -@27 05689000
SBF $CMFM,SKIP+POST SET OFF WAIT/POST IND. 05690000
* ----- END -@27 05692000
CMNDCR EQU * * 05695000
SPACE 05700000
* SET POLL SKIP BIT ON IF STATION CONTROL 05710000
SPACE 05720000
AIF (&ONE).E0552 05730000
TBN $BDDEV(,DTF),BSCA BSCA DTF ? 2 05740000
JF CMGML2 NO-GO TO HANDLE MLTA CASE. 2 05750000
.E0552 ANOP 05760000
AIF (&NOB).D0552 05770000
&MIX SETA &NCS+&NSWL 05780000
AIF (&MIX EQ '2').S0500 05790000
CMGMBN EQU * * C/SB 05800000
MVI CMB#SB,SBN1 SET OP TO SET ON SKIP BIT. C/SB 05810000
B CMBSKP GO TO BSCA SKIP BITS RTN. C/SB 05820000
SPACE 1 05830000
.S0500 ANOP 05840000
J CMNXPL GO TO CHECK NEXT PARM LIST. B 05850000
SPACE 1 05860000
AIF (&ONE).D0552 05870000
CMGML2 EQU * * 2 05880000
.D0552 ANOP 05890000
AIF (&NOM).M1600 05900000
AIF (&NSCTL).S5600 05910000
B CMONSK BR TO SET POLL SKIP BIT ON CSM 05920000
.S5600 ANOP 05930000
SPACE 05940000
J CMNXPL GO TO CHECK NEXT PARM LIST M 05950000
SPACE 05960000
.M1600 ANOP 05970000
***** 05980000
* STORAGE IS AVAILABLE * 05990000
***** 06000000
SPACE 06010000
CMGMAV EQU * STORAGE IS AVAILABLE 06020000
SBN CMSWIT,CMRSLN INDICATE READ CAN BE SCHEDULED 06030000
ST CMLPL,PL SAVE LAST SCHEDULED READ PL @ 06040000
* * WILL BE USED IN THE TRACE 06050000
SBF PL$OPM(,PL),OPGETM SET OFF GETMAIN NEEDED BIT 06060000
SPACE 06070000
* SET THE SKIP BIT OFF FOR THIS TERMINAL 06080000
SPACE 06090000
AIF (&ONE).C0553 06100000
TBN $BDDEV(,DTF),BSCA BSCA DTF ? 2 06110000

```

```

.C0553 JF      CMGML3      NO-GO TO HANDLE MLTA CASE.      2 06120000
ANOP      AIF      (&NOB).D0553      06130000
AIF      AIF      (&MIX EQ '2').S0600      06140000
*        --START-----@27      06150000
TBF      LCBAT2(,DTF),LCBRCI      NOT RECEIVE INITIAL      06151000
TBN      LCBAT2(,DTF),LCBACT      * AND LINE ACTIVE      06152000
JT      CMGNSB      YES - DON'T SET SKIP BIT.      06153000
*        --END-----@27      06154000
MVI      CMB#SB,SBF1      SET OP TO SET OFF SKIP BIT. C/SB      06155000
B      CMBSKP      GO TO BSCA SKIP BITS RTN. C/SB      06160000
SPACE 1      06170000
CMGNSB EQU *      * LOCAL      @27      06180000
.S0600 ANOP      06185000
AIF      (&NOM).M1700      06190000
J      CMNOSO      GO COMPARE BUFFER SIZES.      2 06200000
SPACE 1      06210000
AIF      (&ONE).D0553      06220000
CMGML3 EQU *      *      2 06230000
.D0553 ANOP      06240000
AIF      (&NSCTL).S5700      06250000
B      CMOFSK      BRANCH TO SET SKIP BIT OFF      CM 06260000
.S5700 ANOP      06270000
SPACE      06280000
.M1700 ANOP      06290000
CMNOSO EQU *      COMPARE HOLD BUFFER SIZE NEEDED      06300000
CLC      CMIIND(2),#BUFND      COMPARE PARM SIZE NEEDED VS      06310000
*        LINE QUEUE SIZE NEEDED      06320000
JNL      CMIV40      JUMP UNLESS PARM NEED GT LINE      06330000
SPACE      06340000
* STORAGE NEEDED BY CURRENT PARM LIST IS GREATER THAN THAT NEEDED BY      06350000
* PREVIOUS      INPUTS TESTED FOR THIS LINE      06360000
* RESET THE BUFFER SIZE NEEDED FOR THIS LINE      06370000
SPACE      06380000
MVC      CMIIND(2),#BUFND      MOVE BUFFER SIZE NEEDED TO FIELD      06390000
SPACE      06400000
* GET THE NEXT PARAMETER LIST IN THE LINE QUEUE      06410000
* FIRST TEST TO SEE IF THERE IS ANOTHER PARM LIST IN THE LINE QUEUE      06420000
SPACE      06430000
CMIV40 EQU *      06440000
AIF      (&N32).T3000      06450000
SPACE 1      06460000
*      BSCA POLL FOR STATUS IS HIGHEST PRIORITY OPERATION ON THE LINE.      0B 06470000
SPACE 1      06480000
TBN      $BDDEV(,DTF),BSCA      BSCA DTF, AND      0B 06490000
TBN      PL$OPC(,PL),OPLSNS      POLLING FOR STATUS ?      0B 06500000
JT      CMIV60      YES-GO HANDLE, TOP PRIORITY.      0B 06510000
.T3000 ANOP      06520000
CMNXPL EQU *      GET NEXT PARM LIST      06530000
CLI      PLCHN-1(,PL),NOBIT      IS CHAIN POINTER NULL      06540000
JE      CMIV60      JUMP IF NO MORE PARM LISTS      06550000
L      PLCHN(,PL),PL      LOAD ADDR OF NEXT PARM LIST      06560000
B      CMDTII      LOOP BACK TO CHECK NEW PL      06570000
SPACE      06580000
*****      06590000
*      HAVE LOOKED AT EVERY PARM LIST IN THE QUEUE      *      06600000
*****      06610000
SPACE      06620000
CMIV60 EQU *      06630000
SPACE      06640000
SPACE      06650000

```

```

* DETERMINE IF WE ARE TO DO GETMAIN BY CHECKING STORAGE AREA NEEDED 06660000
* AND AVAILABLE WITH GETMAIN 06670000
* IF GREATER THAN ZERO - DO THE GETMAIN 06680000
  SPACE 06690000
  CLC CMIIND(2),X$0000 IS SIZE GREATER THAN ZERO 06700000
  JE CMIV80 IF NOT, JUMP 06710000
  SPACE 06720000
* MUST DO GETMAIN WHICH MUST WORK ELSE HAVE CCP INTERNAL ERROR 06730000
* PUT SIZE NEEDED INTO GETMAIN LIST AND THEN GO TO GETMAIN INTERFACE 06740000
  SPACE 06750000
  MVC GMLIST+GMSIZE(2),CMIIND MOVE SIZE NEEDED TO GETMAIN LIST 06760000
  B CMGMRT BR TO GETMAIN INTERFACE ROUTINE 06770000
  SPACE 06780000
* SET UP LCBIBA AND LCBIBL FIELDS IN THE LCB 06790000
  SPACE 06800000
  MVC LCBIBA(2,DTF),GMLIST+GMADDR MOVE RECORD ADDR TO LCB 06810000
  ALC LCBIBA(2,DTF),X$0004 BYPASS FREE LIST 06820000
  MVC LCBIBL(2,DTF),CMIIND REQUESTED LENGTH 06830000
  SPACE 06840000
CMIV80 EQU * 06850000
CMIVGX B *-* RETURN 06860000
  TITLE '$E092/CMGBUF---GETMAIN RIGHT SIZE INPUT HOLD BUFFER' 06870000
***** 06880000
* NAME - CMGBUF * 06890000
* * 06900000
* FUNCTION - ADJUST INPUT HOLD BUFFER SIZE AFTER OP END BECAUSE * 06910000
* IT IS KNOWN HOW MUCH DATA CAME IN PRIOR TO CALLING * 06920000
* THE CHECK ROUTINE TO ACTUALLY CAUSE THE DATA TO BE * 06930000
* MOVED. THE BUFFER GETMAINED ORIGINALLY FOR THE LINE * 06940000
* JUST SAVES TPBUFF SPACE SO WE KNOW WE CAN GET SPACE * 06950000
* WHEN THE DATA COMES IN (OP END). * 06960000
***** 06970000
  SPACE 06980000
CMGBUF EQU * 06990000
  ST CMGBFX+3,ARR SAVE RETURN ADDRESS 07000000
  SPACE 1 07010000
  CLI LCBIBA-1(,DTF),NOBIT IS THERE A CURRENT INVITE BUF ? 07020000
  JE CMGIGM NO-GO RIGHT TO GETMAIN. 07030000
  SPACE 07040000
* IF CURRENT BUFFER FOR READ IS LARGER THAN NEED, THEN FREE THE 07050000
* CURRENT BUFFER, AND GET ONLY WHAT IS NEEDED. 07060000
  SPACE 07070000
  CLC #BUFND,LCBIBL(2,DTF) CURRENT LARGER THAN NEEDED ? 07080000
  JE CMGBFX NO-THEN MUST BE EQ, USE CURRENT 07090000
  L LCBIBA(,DTF),XR2 LOAD @ OF CURRENT BUFFER. 07100000
  B CMFMRT GO TO FREEMAIN RTN, FREE CUR'NT 07110000
  SPACE 07120000
  L CMSDTF,DTF RELOAD DTF REG. 07130000
CMGIGM EQU * 07140000
  MVI LCBIBA-1(,DTF),NOBIT ZERO INVITE BUF ADDR. * 07150000
  CLC #BUFND(2),#GMS+1 MAX AVAIL TO SATISFY GETMAIN ? 07160000
  JH CMGINO NO-GO SET UP TO WAIT ON FREEMAN 07170000
  MVC GMLIST+GMSIZE(2),#BUFND MOVE NEEDED LENGTH TO GM LIST. 07180000
  B CMGMRT GO TO GETMAIN RTN. 07190000
  SPACE 07200000
  JOL CMGINO GETMAIN FAILED, GO WAIT. 07210000
  SPACE 1 07220000
  MVC LCBIBA(2,DTF),GMLIST+GMADDR ADDR OF GOT BUF IN LCB. 07230000
  ALC LCBIBA(2,DTF),X$0004 BYPASS FREE LIST IN 1ST 4 BYTES 07240000
  MVC LCBIBL(2,DTF),#BUFND LENGTH REQUESTED 07250000

```

```

SPACE 1 07255000
* * LENGTH RECEIVED BECAUSE OF 07260000
* * ROUND UP IN GETMAIN ROUTINE. 07270000
* * IF USED AMOUNT RECVD WE 07280000
* * WOULD FREE/GETMAIN AT OP END. 07290000
CMGBFX EQU * * 07300000
B ## RETURN 07310000
SPACE 2 07320000
* INDICATE GETMAIN NEEDED FOR THIS LINE. 07330000
CMGINO EQU * * 07340000
SBN PL$OPM(,PL),OPGETM SET GETMAIN NEEDED IN PARM LST. 07350000
SBN LCBATR(,DTF),LCBGMN SET GETMAIN NEEDED FOR LINE. 07360000
* ----- START -@27 07362000
SBF $CMFM,SKIP+POST SET OFF WAIT/POST IND. 07364000
* ----- END -@27 07366000
B CMPAII GO POST USER AS NECESSARY. 07380000
TITLE '$E092/CMSTOR---GETMAIN-SIZE-DETERMINATION' 07390000
***** 07400000
* * 07410000
* NAME--CMSTOR * 07420000
* * 07430000
* TITLE--GETMAIN SIZE DETERMINATION * 07440000
* * 07450000
* FUNCTION--THIS ROUTINE CALCULATES THE AMOUNT OF MAIN STORAGE * 07460000
* NEEDED FOR A PARTICULAR TYPE OF TP REQUEST. * 07470000
* * 07480000
* OPERATION-- * 07490000
* . ALWAYS ADD FOUR BYTES FOR THE GETMAIN PARM LIST. * 07500000
* . IF A POLL FOR STATUS OPERATION ADD 20. * 07510000
* . IF SYSTEM INVITE ONLY ADD #CCMCL, AND MOVE #CCMCL * 07520000
* INTO PLINL. * 07530000
* . IF USER INVITE ONLY ADD PLINL. * 07540000
* * 07550000
* INPUT-- * 07560000
* XR1--ADDRESS OF THE TP PARAMETER LIST. * 07570000
* XR2--ADDRESS OF THE DTF FOR THE TP PARAMETER LIST. * 07580000
* * 07590000
* OUTPUT-- * 07600000
* XR1--NOT ALTERED * 07610000
* XR2--NOT ALTERED * 07620000
* #BUFND--TOTAL AMOUNT OF STORAGE NEED FOR THE TP REQUEST. * 07630000
* * 07640000
* EXIT-- TO NSI OF CALLER. * 07650000
***** 07660000
SPACE 07670000
CMSTOR EQU * * 07680000
AIF (&NOB).NB010 07690000
AIF (&MIN).N0010 07700000
.NB010 ANOP 07710000
.* FOLLOWING CODE WILL BE IN $CC4B2 FOR MIN RES R 07720000
ST CMSTOX+3,ARR SAVE RETURN ADDRESS R 07730000
MVC #BUFND(2),X$0004 ADD 4 FOR GETMNM LIST R 07740000
SPACE 07750000
* DETERMINE WHETHER IT IS SYSTEM OR USER REQUEST R 07760000
SPACE 07770000
AIF (&NOB).C0556 07780000
&MIX SETA &N32+&N37+&N41 07790000
AIF (&MIX EQ '3').T1100 07800000
AIF (&ONE).E0554 07810000
TBN $BDDEV(,DTF),BSCA BSCA DTF ? 7/R0/52 07820000

```

```

.E0554 ANOP                                07830000
      TBN  PL$OPC(,PL),OPLSNS              AND POLL FOR STATUS ?    7/R0/5B 07840000
      JF   CMSUSR                          NO-GO CHECK WHO IS USER. 7/R0/5B 07850000
      ALC  #BUFND(2),RELSNS                USE RECL FOR SENSE OP    7/R0/5B 07860000
      J    CMSTOX                          GO TO EXIT.              7/R0/5B 07870000
      SPACE                                07880000
CMSUSR EQU *                               *                          7/R0/5B 07890000
.C0556 ANOP                                07900000
.T1100 ANOP                                07910000
      TBN  PLOPM(,PL),OP$SYS              IS IT SYSTEM REQUEST    R 07920000
      JF   CMSTOY                          JUMP IF USER REQUEST    R 07930000
      AIF  (&NRUF).F0100                  07930700
      SPACE                                07932100
*   IF RUF ON SCREEN, USE MAX RUF LENGTH AS SYSTEM INPUT LENGTH.  WB 07932800
      SPACE                                07933500
      L    PLTUBA(,PL),XR2                XR2-> TUB.              WB 07934200
      TBN  TUBSCS(,XR2),TUBRUF            RUF ON THE SCREEN ?    WB 07934900
      L    TUBDTF(,XR2),DTF              XR2-> LCB(DTF).        WB 07935600
      MVC  PLINL(2,PL),#RUFCL            USE MAX RUF LENGTH.    WB 07936300
      JT   CMSTOY                          GO EXIT FROM SUBROUTINE. WB 07937000
.F0100 ANOP                                07938400
      SPACE                                07940000
*   HAVE SYSTEM READ - USE MAX COMMAND LENGTH AS INPUT LENGTH    R 07950000
      SPACE                                07960000
      MVC  PLINL(2,PL),#CCMCL            MOVE MAX COMMAND LGTH TO PARM R 07970000
CMSTOY EQU *                               R 07980000
      ALC  #BUFND(2),PLINL(,PL)          ADD INPUT LENGTH        R 07990000
CMSTOX B *-*                              R 08000000
      AGO  .G0100                        08010000
.N0010 ANOP                                08020000
      MVI  CMID2,CTSTOR                  MOVE IN ID OF CODED NEEDED. MIN 08030000
CMCAL2 ST  CMSTR@+3,ARR                  SAVE RETURN ADDRESS     MIN 08040000
      ST  CMX1S2,PL                      SAVE XR1 (PL ADDRESS).  MIN 08050000
      SPACE 1                            08060000
      SVC  0                             ##### TRANSIENT CALL ##### MIN 08070000
      DC  AL1(CCPRIB)                    CCP SVC RIB             MIN 08080000
      DC  AL1(CC4B2)                    TRANSIENT $CC4B2.      MIN 08090000
CMID2  DC  XL1'00'                      KEY FOR TYPE CODE WANTED MIN 08100000
CMX1S2 DC  XL2'00'                      PARAMETER LIST ADDRESS  MIN 08110000
#BUFND DC  XL2'00'                      BUFFER SIZE FOR LINE.   MIN 08120000
SAVT2@ DC  AL2(SAVTA2)                  ADDRESS OF SAVE TAS BYTE 2. MIN 08130000
      SPACE 1                            08140000
CMSTR@ B #                               RETURN TO CALLER.       MIN 08150000
.G0100 ANOP                                08160000
      TITLE '$E092/CMGMRT---GETMAIN-INTERFACE-ROUTINE' 08170000
*****
*
* NAME: CMGMRT
*
* FUNCTION:
* PROVIDES A GENERALIZED INTERFACE FOR THE COMMUNICATIONS MANAGER* 08250000
* TO PERFORM A GETMAIN FOR HOLD BUFFER SPACE.                  * 08260000
* IT WILL STORE THE GETMAIN LIST IN THE 1ST FOUR BYTES OF     * 08270000
* GETMAINED AREA FOR FUTURE FREEMAIN CALL.                    * 08280000
*
* ENTRY POINT: CMGMRT.
*
* INPUT:
* THE SIZE OF THE HOLD BUFFER NEEDED IS IN FIELD GMSIZE OF THE * 08360000
* GETMAIN PARAMETER LIST GMLIST IN THE COMMUNICATIONS MANAGER * 08370000

```



```

* ENTRY POINT:
*   CMFMRT - MAIN ENTRY POINT - SET CM FREEMAIN INDICATOR.
*   CMFMR2 - THIS ENTRY POINT WILL NOT SET THE INTERNAL FREEMAIN
*             INDICATION.
*
* INPUT:
*   INDEX REGISTER 2 CONTAINS THE ADDRESS OF THE 1ST BYTE BEYOND
*   THE GETMAIN/FREEMAIN PARAMETER LIST.
*
* OUTPUT:
*   THE HOLD BUFFER IS FREEMAINED
*
* EXTERNAL REFERENCES:
*   . FREEMAIN SERVICE ROUTINE ENTRY POINT ($CC4FM)
*
* EXITS, NORMAL:
*   TO INSTRUCTION FOLLOWING THE INVOKING OF THIS ROUTINE (ARR)
*
* ATTRIBUTES:
*   RESIDENT, REUSABLE
*
*****
SPACE
CMFMRT EQU *          FREEMAIN INTERFACE ROUTINE
SBN #CMSWT,#CMFMD    SET SWITCH TO INDICATE FREEMAIN
*                   OCCURRED IN CM
CMFMR2 EQU *          ENTRY POINT WHICH DOES NOT SET
*                   INTERNAL FREEMAIN INDICATION
ST CMFMRX+3,ARR      SAVE THE ARR
SPACE
* XR2 POINTS AT GETMAIN/FREEMAIN PARAMETER LIST
SPACE
SVC 0                GO TO FREEMAIN ROUTINE
DC AL1(CCPRIB)       CCP RIB
DC AL1(FMRIB)        FREEMAIN SUB RIB
SPACE
CMFMRX B *-*         RETURN
TITLE '$E092/$CC4FR--FREE GETMAINED AREAS'
*****
* NAME--$CC4FR
*
* FUNCTION : FREE GETMAINED AREAS FOR A SPECIFIC TP REQUEST.
*           ROUTINE USED BY TRANSIENTS AND RESIDENT CM CODE.
*
* OPERATION:
*   . IF SYSTEM PUT WAIT, EXIT
*   . IF PLRECA IS NON ZERO, FREE HOLD BUFFER,
*     ZERO PLRECA-1,
*     ZERO LCBIBA-1, IF IT POINTS TO AREA FREED.
*   . IF INVITE,ZERO TUBPL@-1.
*   . IF PUT NO WAIT OR USER INVITE, FREE PARAMETER LIST AREA.
*
* INPUT--
*   XR1 - PARAMETER LIST ADDRESS
*
* OUTPUT -
*   XR1,XR2 - UNCHANGED
*
* EXTERNAL ROUTINES USED:
*   CMFMRT - FREEMAIN INTERFACE.

```

```

* 09010000
* 09030000
* 09050000
* 09060000
* 09070000
* 09080000
* 09100000
* 09110000
* 09120000
* 09130000
* 09140000
* 09150000
* 09160000
* 09170000
* 09180000
* 09190000
* 09210000
* 09220000
* 09230000
* 09250000
* 09260000
* 09270000
* 09290000
* 09300000
* 09310000
* 09320000
* 09340000
* 09350000
* 09370000
* 09380000
* 09390000
* 09400000
* 09410000
* 09420000
* 09430000
* 09440000
* 09450000
* 09460000
* 09470000
* 09480000
* 09490000
* 09500000
* 09510000
* 09520000
* 09530000
* 09530100
* 09530200
* 09530300
* 09530400
* 09530500
* 09530600
* 09530700
* 09540000
* 09550000
* 09560000
* 09570000
* 09580000
* 09580100
* 09580200
* 09580300

```

```

*                               * 09580400
* EXITS: TO NEXT INSTRUCTION FOLLOWING CALL. * 09580500
*                               * 09580600
***** * 09590000
SPACE 1 * 09600000
$CC4FR EQU * 09610000
ST CMFRRT+3,ARR SAVE RETURN ADDRESS 09620000
ST CMFRX2+3,XR2 SAVE XR2 09630000
TBN PLOPM(,PL),OP$SYS IF SYSTEM 09640000
TBN PL$OPM(,PL),OPPUT * PUT 09650000
TBF PL$OPM(,PL),OPNOW * WAIT, 09660000
JT CMFRXT YES - NO FREEMAIN, NO GETMAINS 09670000
CLI PLRECA-1(,PL),NOBIT IS PLRECA NON ZERO 09680000
JE CMPLCK GO CHECK ON FREEING PL 09690000
SPACE 1 09700000
*-----* 09710000
* FREE HOLD BUFFER IF NOT A SYSTEM PUT WAIT * 09720000
*-----* 09730000
SPACE 1 09740000
L PLRECA(,PL),XR2 ADDRESS OF HOLD BUFFER. 09750000
SPACE 1 09760000
B CMFMRT GO FREE HOLD BUFFER. 09770000
SPACE 1 09780000
L PLTUBA(,PL),XR2 TUB ADDRESS 09790000
L TUBDTF(,XR2),DTF DTF ADDRESS 09800000
CLC LCBIBA(2,DTF),PLRECA(,PL) IS AREA FREED THE CURRENT BUFF 09810000
JNE CMIBOK NO - SKIP ZERO 09820000
MVI LCBIBA-1(,DTF),NOBIT ZERO LINE BUFFER ADDRESS 09830000
CMIBOK EQU * 09840000
MVI PLRECA-1(,PL),NOBIT ZERO HOLD BUFFER ADDRESS. 09850000
CMPLCK EQU * 09860000
TBF PL$OPM(,PL),OPNOW IF WAIT OPERATION ? 09870000
JT CMFRXT DO NOT FREE PARAMETER LIST. 09880000
TBN PLOPC(,PL),OPNOW ORIGINALLY A NO WAIT OP? 09883000
JF CMFRXT NO - NO FREE NEEDED 09886000
TBN PLOPC(,PL),OPINV INVITE ? 09890000
JF CMFRPL NO - JUST FREE PL 09900000
L PLTUBA(,PL),XR2 TUB ADDRESS 09910000
MVI TUBPL@-1(,XR2),NOBIT NO LONGER AN OP ENDED INVITE 09920000
* * OUTSTANDING FOR TUB. 09930000
TBN PLOPM(,PL),OP$SYS IF SYSTEM INVITE - PL IN TUB 09940000
JT CMFRXT YES - DONT FREE PL 09950000
SPACE 1 09960000
*-----* 09970000
* PUT-NO-WAIT OR USER INVITE --- FREE PARAMETER LIST * 09980000
*-----* 09990000
SPACE 1 10000000
CMFRPL EQU * * LOCAL 10010000
LA 0(,PL),XR2 ADDR OF PARAMETER LIST 10020000
SPACE 10030000
B CMFMRT FREE PL AREA 10040000
SPACE 10050000
CMFRXT EQU * * LOCAL 10060000
CMFRX2 LA *-*,XR2 RESTORE XR2 10070000
CMFRRT B *-* RETURN 10080000
SPACE 10090000
TITLE '$E092/CMGMPT--GETMAIN PUT HOLD BUFFER' 10100000
***** 10110000
* * 10120000
* NAME--CMGMPT * 10130000

```

```

* 10140000
* TITLE--GETMAIN PUT HOLD BUFFER AND MOVE DATA INTO IT. * 10150000
* * 10160000
* FUNCTION : * 10170000
* GETMAIN FOR : * 10180000
* USER PUT * 10190000
* USER PUT NO WAIT * 10200000
* SYSTEM PUT NO WAIT * 10210000
* * 10220000
***** * 10230000
SPACE 1 * 10240000
CMGMPT EQU * * 10250000
ST CMPTEX+3,ARR SAVE RETURN ADDRESS * 10260000
MVC GMLIST+GMSIZE(2),PLOUTL(,PL) LENGTH OF RECORD AREA * 10270000
ALC GMLIST+GMSIZE(2),X$0004 ADD 4 FOR FREEMAIN LIST * 10280000
SPACE 1 * 10290000
B CMGMRT GETMAIN HOLD BUFFER * 10300000
JOL CMPTEX IF FAILED, EXIT (FLAGS ARE SET) * 10310000
SPACE 1 * 10320000
* MOVE USERS RECORD AREA TO HOLD BUFFER * 10330000
SPACE 1 * 10340000
MVI #CMMVL+MVLTYT,NOBIT ASSUME SYSTEM REQ WITH NO SWAP * 10350000
MVC #CMMVL+MVLFR,PLRECA(2,PL) ASSUME SYS REQ,REC AREA IN PL * 10360000
TBN PLOPM(,PL),OP$SYS IF SYS REQUEST * 10370000
JT CMMVDT YES- MOVE DATA * 10380000
MVI #CMMVL+MVLTYT,SWAPFR FROM ADDR IS IN USER PGM AREA * 10390000
L PLTUBA(,PL),XR2 TUB ADDRESS * 10400000
L TUBTCB(,XR2),XR2 TCB ADDRESS * 10410000
MVC #CMMVL+MVLFR,TCBWK(2,XR2) MOVE USERS REC AREA @ INTO FM@ * 10420000
SPACE 1 * 10430000
CMMVDT EQU * * LOCAL * 10440000
MVC PLRECA(2,PL),GMLIST+GMADDR ADDR OF GETMAINED BUFF INTO PL * 10450000
ALC PLRECA(2,PL),X$0004 BYPASS FREEMAIN LIST * 10460000
MVC #CMMVL+MVLTOA,PLRECA(2,PL) HOLD BUFF DATA AREA IS TO @ * 10470000
MVC #CMMVL+MVLTO,PLOUTL(2,PL) LGTH OF FROM AREA IS SAME * 10480000
SPACE 1 * 10490000
B CMMVRT MOVE INTERFACE ROUTINE * 10500000
SPACE 1 * 10510000
CMPTEX B *- * RETURN * 10520000
TITLE '$E092/CMMVRT--MOVE INTERFACE ROUTINE' * 10530000
***** * 10540000
* * 10550000
* NAME--CMMVRT * 10560000
* * 10570000
* TITLE--MOVE INTERFACE ROUTINE * 10580000
* * 10590000
* FUNCTION-- * 10600000
* MOVE TO OR FROM AREA IN USERS PROGRAM * 10610000
***** * 10620000
SPACE 1 * 10630000
CMMVRT EQU * * 10640000
ST CMMVEX+3,ARR SAVE RETURN ADDRESS * 10650000
L PLTUBA(,PL),XR2 ADDRESS OF TUB * 10660000
MVC #CMMVL+MVLTCB,TUBTCB(2,XR2) USERS TCB ADDR FOR ATR S * 10670000
LA #CMMVL,XR2 ADDR OF MOVE PARAMETER LIST * 10680000
MVC MVLFR(2,XR2),PLEFFL(,PL) LENGTH OF FROM AREA FROM PL * 10690000
* * FIELD PLOUTL/PLEFFL. * 10700000
SPACE 1 * 10710000
SVC 0 * 10720000
DC AL1(CCPRIB) SVC TO MOVE ROUTINE * 10730000

```

```

DC      AL1(MVRIB)                                10740000
SPACE                                     10750000
CMMVEX B      *-*          RETURN TO CALLER      10760000
SPACE 3                                     10760400
*                                           -- START -----@34 10760800
* THIS ROUTINE SKIPS THE TRACE SVC IF TRACE IS NOT ACTIVE. 10761200
SPACE                                     10761600
CMTRCE EQU      *                                     10762000
ST      CMTR03,ARR          SAVE RETURN ADDRESS  10762400
ALC     CMTR03,X$0002(2)    POINT TO TRACE ID    10762800
CMTR03 EQU      *+5                                     10763200
MVC     CMTR09(1),#        PUT IN TRACE ID      10763600
ALC     CMTR03(2),X$0001    BUMP ARR TO RETURN ADDRESS 10764000
ST      CMTR05,XR1         SAVE REG 1           10764400
L       NCSYS@,XR1        XR1--> SYS COM       10764800
CMTR05 TBN     NCDSP1(,XR1),BIT0          TRACE ACTIVE? 10765200
EQU     *+3                                     10765600
LA      #,XR1              RESTORE REG 1        10766000
JF      CMTR20             NO RETURN            10766400
SVC     0                  SUPVSR CALL         10766800
DC      AL1(CCPRIB)        CCP RIB             10767200
DC      AL1(TRRIB)        CCP RIB FOR TRACE    10767600
CMTR09 DC      AL1(#),#    TRACE ID            10768000
CMTR20 EQU     *                                     10768400
L       CMTR03,IAR        RETURN              10768800
*                                           -- END -----@34 10769200
TITLE '$E092/CMQUE---ADD-TP-REQUEST-TO-LINE-QUEUE' 10770000
***** 10780000
* 10790000
* NAME--CMQUE 10800000
* 10810000
* TITLE--ADD TP REQUEST TO LINE QUEUE 10820000
* 10830000
* FUNCTION-- 10840000
* ADD THIS REQUEST TO THE LINE QUEUE 10850000
* MUST CHECK TO SEE IF THIS PARAMETER LIST IS TO BE ADDED TO THE 10860000
* END OF THE LINE QUEUE OR IS TO BE PUT AT THE TOP OF THE QUEUE 10870000
* IE IF RETRY OF PARM LIST IN ERP - IT WILL GO TO TOP OF QUEUE TO 10880000
* MAINTAIN THE ORIGINAL ORDER OF THE LINE QUEUE 10890000
* XR2 WILL BE USED TO POINT TO THE VARIOUS PARAMETER LISTS IN THE QUE 10900000
* POINT XR2 AT LOCATION OF 1ST PARM LIST POINTER 10910000
* 10920000
* EXIT - IF GETMAINS NECESSARY ARE SUCCESSFUL, TO NSI OF CALLER 10930000
* IF NOT, TO CMPAII. 10940000
***** 10950000
SPACE 10960000
CMQUE EQU      *          QUEUE THIS TP REQUEST 10970000
ST      CMQRET+3,ARR      RETURN ADDRESS        10980000
SBN     $FLGC,#INVPL     SET IND-G.M. FROM PL AREA ONLY 10986000
SPACE 10990000
AIF     (&NOB).C0456     11000000
L       CMSPL,PL         POINT TO THE TP PARM LIST.  B 11010000
.C0456 ANOP 11020000
SPACE 11030000
TBF     PL$OPM(,PL),OPNOW WAIT OPERATION 11040000
JT      CMQOP            YES - SKIP GETMAIN, JUST QUEUE 11050000
TBN     PLOPM(,PL),OP$SYS IF SYSTEM 11060000
CLI     PLOPC(,PL),OPINV * INVITE, PL IN TUB. 11070000
JC      CMQOP,TRUAEQ     YES - SKIP GETMAINS 11080000
TBF     PLOPM(,PL),OP$SYS IF A USER 11090000

```

TBN	PLOPC(,PL),OPINV	* INVITE,	11100000
JT	CMPLGT	YES - JUST GETMAIN FOR PL	11110000
SBF	\$FLGC,#INVPL	SET OFF INV PL ONLY IND.	11112000
SBN	\$FLGC,#PUTTP	SET IND-G.M.FROM ANYWHERE	11114000
SPACE	1		11120000
-----			11130000
* PUT NO WAIT OR SYS PNW/INVITE - GETMAIN HOLD BUFFER			11140000
-----			11150000
SPACE	1		11160000
MVC	GMLIST+GMSIZE(2),PLOUTL(,PL) PNW DATA LENGTH		11170000
ALC	GMLIST+GMSIZE(2),X\$0004	* + 4 = TOTAL GETMAIN AREA	11180000
TBN	PLOPM(,PL),OP\$SYS	SYSTEM	11190000
TBN	PLOPC(,PL),OPPUT+OPINV	* PNW/INVITE ?	11200000
JF	CMPNWP	NO-ALSO NEED PNW PARM LIST.	11210000
SPACE			11220000
* SYSTEM PUT-NO-WAIT/INVITE			11230000
SPACE			11240000
CLC	GMLIST+GMSIZE(2),#GMS+1	WILL GETMAIN WORK ?	11250000
JNH	CMPNWG	YES-GO DO.	11260000
J	CMPNWS	NO-MAKE WAIT OP AND DO IT.	11270000
SPACE			11280000
* SYSTEM PUT NO WAIT OR USER PUT NO WAIT			11290000
SPACE			11300000
CMPNWP	EQU *	*	11310000
ALC	GMLIST+GMSIZE(2),PLGMLG	* + PL LENGTH	11320000
CLC	GMLIST+GMSIZE(2),#GMS+1	IS THERE ENOUGH ROOM ?	11330000
JNH	CMPNWG	YES - GO DO GETMAINS	11340000
TBN	PLOPM(,PL),OP\$SYS	IF SYSTEM PUT NO WAIT ?	11350000
JT	CMPNWS	YES - MAKE PUT WAIT AND DO IT.	11360000
L	CMSDTF,DTF	DTF ADDRESS	11370000
SBF	PL\$OPM(,PL),OPNOW	MAKE WAIT OPERATION	11380000
AIF	(&ONE).E0570		11390000
TBN	\$BDDEV(,DTF),BSCA	IF BSCA	2 11400000
BT	CMBSOP	YES-GO CHECK REJECT	2 11410000
.E0570	ANOP		11420000
AIF	(&NOM).M1800		11430000
SBN	LCBATR(,DTF),LCBGMN	DTF NEEDED GETMAIN	M 11440000
SBN	PL\$OPM(,PL),OPGETM	NO - SET GETMAIN NEEDED FOR PL	M 11450000
ALC	CORCNT(1),X\$0001	ADD ONE TO COUNT OF WAITERS	11454000
*----- START -@27			11454500
SBF	\$CMFM,SKIP+POST	SET OFF WAIT/POST IND.	11455000
*----- END -@27			11455500
J	CMPNWF	SET OTHER FLAGS	M 11470000
AGO	.M1850		11480000
.M1800	ANOP		11490000
B	CMBSOP	GO CHECK REJECT ONLY	B 11500000
.M1850	ANOP		11510000
SPACE	1		11520000
CMPNWG	EQU *	* LOCAL	11530000
B	CMGMPT	GETMAIN HOLD BUFFER AND MOVE	11540000
* DATA INTO IT.			11550000
JNOL	CMHBOK	IF GETMAIN OK	11560000
* (PNW OP WILL NOT FAIL BECAUSE			11570000
* OF TEST ABOVE)			11580000
* HOLD BUFFER GETMAIN FAILURE			11590000
CMPNWF	EQU *	*	11600000
SBN	PL\$OPM(,PL),OPGETQ	GETMAIN FAILED AT QUEUE TIME.	11610000
CMPNWS	EQU *	*	11620000
SBF	PL\$OPM(,PL),OPNOW	MAKE WAIT OP	11630000
J	CMQOP	QUEUE IT TO WAIT FOR FREEMAIN	11640000

SPACE 1			11650000
CMHBOK EQU *		* LOCAL	11660000
*			11660500
* BUSY PRINTER SUPPORT			11661000
*			11661500
AIF	(&NPBY).NPY02	SKIP IF NO BUSY PRINTER	11662000
L	PLTUBA(,PL),XR2	POINT TO TUB	11662500
CLI	TUBPHY(,XR2),TUB5M2	TEST FOR 3735	11663000
JH	CMNBYP	IF YES, SKIP BUSY PRINT CODE	11663500
TBN	TUBSCS(,XR2),TUBBPT	IS BUSY PRINT ALLOWED?	11663600
JF	CMNBYP	NO, SKIP BUSY PRINT CODE	11663700
CLC	PLOUTL(2,PL),X\$0002	IS PUT LENGTH GR THN 2	11664000
JNH	CMNBYP	NO, SKIP PRINTER BSY STUFF	11664500
TBF	PLOPM(,PL),OPREQR	USER OPERATION AND	11665000
TBN	PLOPC(,PL),OPPUT	A PUT OPERATION AND	11665500
L	PLRECA(,PL),XR2	FIND THE RECORD AREA	11666000
TBN	WCC(,XR2),STPRT	IS THE START PRINT BIT ON?	11666500
L	PLTUBA(,PL),XR2	XR2----> TUB	11667000
JF	CMNBYP	JUMP IF NOT PRINTER OPERATION	11667500
SBN	TUBAT4(,XR2),TUBBSY	SET ON PRINTER BUSY BIT	11668000
CMNBYP EQU *			11668500
.NPY02 ANOP			11669000
TBN	PLOPC(,PL),OPPUT+OPINV	IF SYS PNW/INV ONLY GET BUFFER	11670000
JT	CMQOP	YES - GO QUEUE IT	11680000
TBN	PLOPM(,PL),OP\$SYS	IF SYSTEM	11690000
TBN	PLOPC(,PL),OPINV	* INVITE, PL IN TUB.	11700000
JT	CMQOP	YES - SKIP GETMAINS	11710000
SBN	\$FLGC,#PUTTP	SET 'GETMAIN FROM ANYWHERE'	11715000
SPACE 1			11720000
-----			11730000
* PUT-NO-WAIT OR USER INVITE - GETMAIN FOR PARAMETER LIST			11740000
-----			11750000
SPACE 1			11760000
CMPLGT EQU *		* LOCAL	11770000
MVC	GMLIST+GMSIZE(2),PLGMLG	LENGTH OF GETMAIN FOR PL	11780000
SPACE 1			11790000
B	CMGMRT	GETMAIN FOR PARAMETER LIST	11800000
JNOL	CMPTGT	IF GETMAIN OK	11810000
*		(PNW OP WILL NOT FAIL BECAUSE	11820000
*		OF TEST ABOVE)	11830000
SBN	PL\$OPM(,PL),OPGETQ	FAILED - MUST QUEUE AFTER FREE	11840000
J	CMQOP	QUEUE IT TO WAIT FOR FREEMAIN	11850000
SPACE 1			11860000
CMPTGT EQU *		* LOCAL	11870000
L	GMLIST+GMADDR,XR2	ADDRESS OF GETMAINED AREA	11880000
MVC	PLLEN+3(PLLEN,XR2),PLECB+2(,PL)	MOVE PL TO GETMAINED AREA	11890000
LA	4(,XR2),PL	ADDR OF PL INTO XR1	11900000
ST	CMSPL,PL	SAVE ADDR OF PL WORKING WITH	11910000
SPACE 1			11920000
-----			11930000
* QUEUE THE REQUEST ON LINE QUEUE			* 11940000
-----			11950000
SPACE 1			11960000
CMQOP EQU *			11970000
SBF	\$FLGC,#PUTTP+#INVPL	SET BOTH IND.S OFF	11974000
L	CMSDTF,XR2	RESTORE DTF ADDR TO XR2	11980000
TBN	LCBATR(,XR2),LCB1PL	IS THIS TO BE 1ST PL IN Q	11990000
SBF	LCBATR(,XR2),LCB1PL	SET OFF LCB1PL BIT	12000000
SPACE			12010000
LA	LCBPLQ-1(,XR2),XR2	POINT XR2 AT PARM LIST ADDR	12020000

```

SPACE 12030000
JF CMTQND JUMP IF ADD TO END OF LINE Q 12040000
SPACE 12050000
* PUT THIS PARM LIST AT THE TOP OF THE QUEUE 12060000
SPACE 12070000
MVC PLCHN(2,PL),PLCHN(,XR2) CHAIN CURRENT Q OFF THIS PL 12080000
J CMADDQ JUMP TO ADD THIS GUY TO Q 12090000
SPACE 12100000
* DETERMINE IF POINTER IS NULL 12110000
* IF NULL STORE ADDR OF THIS PARM LIST IN THAT LOCATION 12120000
SPACE 12130000
CMTQND EQU * TEST FOR END OF QUEUE 12140000
CLI PLCHN-1(,XR2),NOBIT IS PARM LIST POINTER NULL 12150000
JE CMADDQ JUMP IF NULL TO ADD TO QUEUE 12160000
SPACE 12170000
*POINTER NOT NULL - SO TRY NEXT PARM LIST POINTER 12180000
SPACE 12190000
*THIS PARM LIST PTS TO ANOTHER PARM LIST IN LINE QUEUE 12200000
* GO DOWN LIST LOOKING FOR END OF QUEUE 12210000
*(HIGH ORDER BYTE OF ADDRESS - X'00') 12220000
SPACE 12230000
L PLCHN(,XR2),XR2 POINT XR2 AT NEXT PARM LIST 12240000
* IN THE LINE QUEUE 12250000
B CMTQND BR TO TEST FOR QUEUE END 12260000
SPACE 12270000
* FOUND END OF QUEUE SO ADD THIS PARAMETER LIST TO CHAIN HERE 12280000
SPACE 12290000
CMADDQ EQU * ADD PARM LIST TO QUEUE 12300000
ST PLCHN(,XR2),PL ADD PARM LIST TO QUEUE 12310000
SPACE 1 12320000
* IF GETMAIN FAILED, LEAVE ON QUEUE UNTIL FREEMAIN DONE 12330000
TBF PL$OPM(,PL),OPGETM+OPGETQ IF GETMAIN FAILED DO NOT RETURN 12340000
BF CMPAII * TO SCHEDULE OPERATION. 12350000
CMQRET B *-* RETURN 12360000
TITLE '$E092/CMDEQ---DEQUEUE PARAMETER LIST FROM LINE' 12370000
***** 12380000
* * 12390000
* NAME: CMDEQ * 12390100
* * 12390200
* FUNCTION: DEQUEUE PARAMETER LIST FROM THE LINE QUEUE. * 12390300
* * 12390400
* OPERATION: MOVE CHAIN POINTER OF THIS PARAMETER LIST TO THE CHAIN 12390500
* POINTER OF PARAMETER LIST PRECEDING THIS ONE IN THE Q. * 12390600
* * 12390700
* INPUT: XR1 -> PL, XR2 -> DTF. * 12390800
* * 12390900
* OUTPUT: XR1 -> PL, XR2 -> CURRENT DTF. * 12391000
* * 12391100
***** 12400000
SPACE 12410000
* TAKE THE POINTER FROM THIS PARM LIST AND PLACE IN PTR ADDR OF 12420000
* LOCATION THAT POINTED TO THE COMPLETED OPERATION 12430000
SPACE 12440000
CMDEQ EQU * DEQUEUE THE PARM LIST 12450000
ST CMDEQR+3,ARR SAVE RETURN ADDRESS 12460000
LA LCBPLQ-1(,DTF),XR2 LOAD PTR TO FIRST PARM LIST. 12470000
CMDQCK EQU * LOCAL 12480000
CLC PLCHN(2,XR2),CMSPL NEXT PARM LIST TO DEQ ? 12490000
JE CMDCHN YES-GO TO DE-CHAIN IT. 12500000
L PLCHN(,XR2),XR2 UPDATE TO NEXT PARM LIST. 12510000

```

```

      B      CMDQCK          GO BACK AND CHECK THIS ONE.          12520000
* MOVE PTR TO NEXT PARM LIST TO PREVIOUS PARM LIST POINTER      12530000
CMDCHN EQU *              * LOCAL                                12540000
      MVC    1(2,XR2),PLCHN(,PL)  MOVE POINTER                  12550000
      MVC    PLRTC(2,PL),PL$RTC(,PL) SET EXTERNAL RETURN CODE NOW 12560000
*              * THAT PLCHN ISNT USED.                          12570000
      L      CMSDTF,DTF          RESTORE DTF ADDR TO XR2.        12570100
CMDEQR B #                  RETURN                                12580000
      TITLE '$E092/CMSRPL--- SEARCH PARAMETER LIST CHAIN FOR LINE' 12590000
*****
*              * 12600000
*              * 12610000
* NAME--CMSRPL          * 12620000
*              * 12630000
*      TITLE--SEARCH PARAMETER LIST CHAIN FOR LINE FOR PL TO SCHEDULE * 12640000
*              * 12650000
*      FUNCTION--          * 12660000
* LOOK FOR NON-READ IF ANY TO EXECUTE NOW          * 12670000
* ELSE SEARCH THE QUEUE FOR A READ AND START ALL READS FOR WHICH * 12680000
* THE NECESSARY HOLD BUFFER SPACE CAN BE OBTAINED * 12690000
* ASSURE NO OP SCHEDULED TO TERMINAL IN ERROR RECOVERY * 12700000
*              * 12710000
*      OUTPUT - XR1 WILL POINT TO PARAMETER LIST TO SCHEDULE * 12720000
*              * 12730000
*              * 12740000
*****
      SPACE 1          12750000
CMSRPL EQU *          12760000
      ST      CMSRRT+3,ARR          12770000
      AIF    (&NOB).C0469          12780000
      TBN    $BDDEV(,DTF),BSCA      BSC DTF ?          12790000
      JF     CMSRP1          NO-DON'T SET TIMER IND. OFF 12800000
      SBF    LCBATR(,DTF),LCBTIM    SET TIMER IND. OFF 12810000
CMSRP1 EQU *          12820000
.C0469 ANOP          12830000
      CLI    LCBPLQ-1(,XR2),NOBIT   IS QUEUE EMPTY          12840000
      JNE    CMSRP2          NO - CHECK THE QUEUE.        12840700
      AIF    (&NBDA).C0470          12841400
      TBN    $BDDEV(,DTF),BSCA      BSC DTF ?          12841700
      L      $BDWKA(,DTF),XR1      XR1 --> WORKAREA      12842100
      TBN    $BWKMC(,XR1),X'01'     IS DA WORKING?        12842800
      JF     CMSRP3          NO - DON'T CANCEL TIMEOUT.    12843500
      TBF    LCBAT1(,DTF),LCBNTQ+LCBINT NO PARM LIST OR REMOVED FROM Q? 12844200
      JF     CMSRP3          DON'T CANCEL TIMEOUT.        12844900
      DC     XL3'F38802'          SIO TO CANCEL 2 SEC. T. O. 12845600
CMSRP3 EQU *          12846300
.C0470 ANOP          12847000
      B      CMPAII          BR TO CHECK OP END COUNT.    12847700
CMSRP2 EQU *          12848400
      SPACE          12860000
*****
* REQUEST IN QUEUE FOR THE LINE READY TO GO          12870000
*****
*              * 12880000
*              * 12890000
      SPACE          12900000
      L      LCBPLQ(,XR2),PL      POINT XR1 AT 1ST PARM LIST IN Q 12910000
CMRDCK EQU *          12920000
      L      PLTUBA(,PL),XR2      POINT XR2 AT THE TUB    12930000
      AIF    (&NSWL).S0100          12940000
      SPACE 1          12950000
*      IGNORE SYSTEM REQUESTS IF A USER TASK OWNS THE SWITCHED LINE. SB 12960000
      SPACE 1          12970000
      AIF    (&ONE).E0600          12980000

```


.E0600	TBN	TUBCHR(,XR2),TUBLNE	BSCA LINE ?	S2	12990000
	ANOP				13000000
	TBN	TUBAT1(,XR2),TUBSWC	SWITCHED LINE ?	SB	13010000
	JF	CMRFSH	NO-GO CHECK REFRESH OPTION.	SB	13020000
	L	TUBDTF(,XR2),DTF	POINT TO THE DTF.	SB	13030000
	TBN	LCBATR(,DTF),LCBNIT	LINE CONNECTED ?	SB	13040000
	JT	CMOWNC	YES- CHECK OWNING TUB	SB	13050000
	SPACE	1			13060000
*	LINE NOT	CONNECTED	- ALLOW REQUEST UNLESS SYSTEM REQUEST AND	SB	13070000
*			USER TASK OWNS THE LINE.	SB	13080000
	SPACE	1			13090000
	CLC	LCBTCB(2,DTF),X\$0000	SWITCH LINE OWNED ?	SB	13100000
	JE	CMRFSH	NO - HANDLE OPERATION NOW.	SB	13110000
	CLC	LCBTCB(2,DTF),@CPTCB	USER TASK OWN LINE ?	SB	13120000
	TBN	PLOPM(,PL),OP\$SYS	* SYSTEM REQUEST ?	SB	13130000
	JC	CMNXTP,TRUNEQ	YES-SKIP THIS PARM LIST.	SB	13140000
	J	CMRFSH	NOT CONNECTED SO SCHEDULE	SB	13150000
	SPACE	1			13160000
*	LINE CONNECTED	- CHECK TO SEE IF REQUEST IS FOR CONNECTED TUB		SB	13170000
	SPACE	1			13180000
CMOWNC	EQU	*	* LOCAL	SB	13190000
	CLC	PLTUBA(2,PL),LCBOWN(,DTF)	YES - THIS TUB OWN THE LINE ?	SB	13200000
	JNE	CMNXTP	NO - LINE OWNED BY ANOTHER	SB	13210000
*			* TUB SO SKIP THIS PL	SB	13220000
CMRFSH	EQU	*	*	SB	13230000
	L	PLTUBA(,PL),XR2	POINT TO THE TUB.	SB	13240000
.S0100	ANOP				13250000
	AIF	(&N32).T0700			13260000
	SPACE				13270000
*	IF	OPERATION IS REFRESH OF SCREEN, THE SET TEMP OP CODE FOR PUT.		0B	13280000
	SPACE				13290000
	AIF	(&ONE).E0700			13300000
	TBN	TUBCHR(,XR2),TUBLNE	BSCA LINE ?	02	13310000
.E0700	JF	CMRXPT	NO-GO CHECK PUT IN ERP.	02	13320000
	ANOP				13330000
	TBN	PLOPM(,PL),OPSTOP	STOP INVITE AND -	B	13330700
	TBN	TUBSCS(,XR2),TUBCLR	* CLEAR KEY DEPRESSED ?	B	13331400
	JF	CMRXE1	NO-CONTINUE	B	13332100
	SPACE	1			13332800
*	PUT	CLEAR	RETURN CODE INTO PARM LIST AND GO DE-QUEUE IT	B	13333500
	SPACE	1			13334200
	MVI	PL\$RTC(,PL),RCXCLR	INT. RETURN CODE = CLEAR	B	13334900
	SBF	TUBSCS(,XR2),TUBCLR	SET OFF CLEAR IND.	B	13335600
	ST	CMSPL,PL	SAVE PL ADDRESS	B	13336300
	L	TUBDTF(,XR2),XR2	XR2--> DTF	B	13337000
	B	CMRETC	GO DE-Q AND POST	B	13337700
CMRXE1	EQU	*	*	B	13338400
	SPACE	2			13339100
	TBN	PL\$OPC(,PL),OPRFSH	REFRESH OPERATION REQUESTED ?	0B	13340000
	JF	CMRXER	NO-GO TEST FOR TERM IN ERP.	0B	13350000
	MNN	PL\$OPM(,PL),CCPUT	ALTER OP CODE TO A PUT.	0B	13360000
	J	CMSCHO	GO HANDLE NOW	0B	13364000
CMRXER	EQU	*	*	0B	13370000
	SPACE	1			13380000
*	IF	POLL FOR STATUS OPERATION , HANDLE AS HIGHEST PRIORITY.		0B	13390000
	SPACE	1			13400000
	TBN	PL\$OPC(,PL),OPLSNS	POLLING FOR STATUS ?	0B	13410000
	JT	CMTSRS	YES-GO HANDLE STATUS POLL.	0B	13420000
	SPACE	1			13425000
CMRXPT	EQU	*	*	0B	13430000

```

.T0700 ANOP 13440000
      TBF  PL$OPM(,PL),OPGETM  GETMAIN NOT FAILED 13450000
      TBN  PL$OPM(,PL),OPPUT   IS IT PUT 13460000
      TBF  TUBAT3(,XR2),TUBERP ASSURE NOT IN ERROR RECOVERY 13470000
      JF   CMNXTP              NO- CHECK NEXT ONE 13480000
      CLI  PLRECA-1(,PL),NOBIT  IF BUFFER IS ALREADY GETMAINED 13490000
      JNE  CMBFOK              YES - SKIP GETMAIN 13500000
*      * SKIPPED FOR PUT NO WAIT, 13510000
*      * MLTA PUT-NOT FIRST TIME THRU, 13520000
*      * DFF PUT WAIT( ALREADY DONE)OR 13530000
*      * SYS PUT WAIT ( NONE NEEDED). 13540000
      SPACE 1 13550000
*      NON-DFF USER PUT WAIT - GETMAIN BUFFER FOR ADDRESSABILITY 13560000
      SPACE 1 13570000
      SBN  $FLGC,#PUTTP        INDICATE 'GETMAIN FOR PUT' REQ. 13575000
      B    CMGMPT              GETMAIN HOLD BUFFER AND MOVE 13580000
*      * DATA INTO IT. 13590000
      SPACE 1 13600000
CMBFOK EQU * * LOCAL 13610000
      TBN  PL$OPM(,PL),OPGETM  IF WAITING FOR GETMAIN 13620000
      JT   CMNXTP              YES - GET NEXT PL IN QUEUE 13630000
      J    CMSCHO              NO - RETURN IT TO BE SCHEDULED 13640000
      SPACE 1 13650000
CMNXTP EQU * CHECK NEXT PARM LIST 13660000
      CLI  PLCHN-1(,PL),NOBIT  ANY MORE PARM LISTS IN QUEUE 13670000
      JE   CMTSRS              NO-GO TRY TO SCHEDULE READ. 13680000
      SPACE 13690000
*MORE REQUESTS IN QUEUE-CHECK THAM FOR NON-READ 13700000
      SPACE 13710000
      L    PLCHN(,PL),PL      PT XR1 AT NEXT PARM LIST 13720000
      B    CMRDCK              BR TO CHECK FOR NON READ 13730000
      SPACE 13740000
* ALL READS IN THE QUEUE SO DO ANALYSIS IN TERMS OF ELIGIBLITY 13750000
* BECAUSE RECORD AREA OR HOLD BUFFER IS AVAILABLE 13760000
* MAY HAVE NON-READS IN QUEUE BUT THEY WILL BE TO TERMINALS IN ERROR 13770000
* RECOVERY 13780000
      SPACE 13790000
CMTSRS EQU * TEST READ ELIGIBILITY 13800000
      B    CMIVGM              BRANCH TO DO READ GETMAIN ANALYS 13810000
      SPACE 1 13820000
      TBN  CMSWIT,CMRSLN      IS RESCHEDULE SWITCH ON 13830000
      BF   CMPAII              NO- CANNOT RESCHEDULE NOW. 13840000
      SPACE 13850000
      L    CMLPL,PL          LAST READ PL SCHEDULED - WILL 13860000
*      * GO INTO TRACE FROM CMSPL 13870000
CMSCHO EQU * SCHEDULE THE PARM LIST REQUEST 13880000
      ST   CMSPL,PL          SAVE PARM LIST ADDR 13890000
      L    PLTUBA(,PL),XR2   POINT XR2 AT THE TUB 13900000
CMSRRT B *-* RETURN 13910000
      TITLE '$E092/$$BMCH--CCP-VERSION-TP-CHECK' 13920000
***** 13940000
* 13950000
*TITLE: $$BMCH - TELEPROCESSING MULTIPLE WAIT. * 13960000
* 13970000
*FUNCTION: THE FUNCTION OF THIS WAIT ROUTINE IS TO SCAN A LIST OF DTF'S* 13980000
* AND, IF NECESSARY, PASS CONTROL AND THE DTF TO THE APPROPRIATE * 13990000
* DEVICE WAIT ROUTINE (BSCA, OR MLTA). THE USER IS THEN NOTIFIED * 14000000
* OF A COMPLETION VIA THE APPROPRIATE DTF. * 14010000
* * 14020000
*INPUT: INPUT TO THE ROUTINE IS THE WAIT LIST OF DTF'S. * 14030000

```

```

*
*OUTPUT: OUTPUT FROM THE ROUTINE WILL BE REGISTER 2 POINTING AT THE
* COMPLETED DTF, WHICH CONTAINS THE COMPLETION CODE. ON RETURN WITH
* NO COMPLETIONS OR NO ACTIVE DTF'S, XR2 WILL POINT TO THE LAST DTF
* IN THE LIST, ALSO WITH THE APPROPRIATE COMPLETION CODE.
*
*EXTERNAL ROUTINES:
* MSBSCH - BSCA PROGRAM WAIT ROUTINE. (ADDR STORED AT CHBMBS BY
* MLMP).
* $MLCK0 - MLTA CHECK ROUTINE. (ADDR STORED AT CHBMML BY MLTA)
*
*****
SPACE 2
$$BMCH EQU * ENTRY POINT NAME FOR MLMP/MLTA
CMBMCH EQU * INTERNAL CM ENTRY POINT.
ST CHSTOR,ARR SAVE RETURN @
MVI CHACTV,UNCOND INITIALIZE SWITCH TO FALL THRU
L @CKLST,LISTRG POINT REGISTER AT WAIT LIST
CHSTRT EQU *
L DTFADR(,LISTRG),DTF POINT REG 2 TO DTF
TBN CHLSTS(,LISTRG),OPGONE COMPL. CODE DESTROYED ?
JF CHNODS NO - DONT RESTORE IT
MVI DTFCMP(,DTF),OPACC RESTORE COMPLETION CODE
CHNODS EQU *
TBN CHLSTS(,LISTRG),CHLSKP IS ENTRY TO BE SKIPPED ?
JT CHSTND YES - GO TEST FOR END
TBN DTFATR(,DTF),FILEPN+FILEAC THIS FILE OPEN AND ACTIVE ?
JF CHSTND NO - GO CHECK FOR END
AIF (&NOM).H0300
AIF (&NOB).H0200
CLI DTFDEV(,DTF),BSCA BSCA ADAPTER ?
JNL CHBSCA YES -
.H0200 EJECT
***** M
* PROCESS MLTA DTF * M
***** M
SPACE 2
CHMLTA EQU * * LOCAL M
CLI CHCHEK,NOMLTA IS MLTA IN THE PROGRAM ? M
JE CHSTND NO - CHECK FOR END M
MVI CHACTV,NOOP SET ACTIVE DTF FOUND SWITCH M
ST CHSV1,LISTRG SAVE WAIT LIST POINTER M
L @CKLST,LISTRG POINT TO START OF WAIT LIST M
B *-* GO TO MLTA CHECK ROUTINE M
CHCHEK EQU *-2 ADDRESS FILLED IN BY M
CHBMML EQU *-1 * MLTA IOCS OPEN M
SPACE 14540000
CHSV1 EQU *+3 * M
LA *-*,LISTRG RESTORE WAIT LIST POINTER M
TBN DTFATR(,DTF),FILEAC OPERATION POSTED COMPLETE ? M
JF CHPNTR YES - SET POINTER M
AIF (&NOB).H0400 14590000
J CHSTND GO TEST FOR END OF LIST B
.H0300 SPACE 2 14610000
***** B
* PROCESS BSCA DTF * B
***** B
SPACE 1 14650000
CHBSCA EQU * * LOCAL B
CLI CHCKBS,NOBSCA IS BSCA IN THE PROGRAM? B

```

```

JE      CHSTND      BRANCH IF NOT      B 14680000
CLI     DTFCMP(,DTF),OPACC  BSCA OPERATION PENDING ?      B 14690000
JNE     CHSTND      NO - TEST FOR END OF LIST      B 14700000
MVI     CHACTV,NOOP  SET ACTIVE DTF FOUND          B 14710000
CHCKBS EQU *+2      *                          B 14720000
B       *-*        GO TO BSCA WAIT ROUTINE      B 14730000
CHBMBS EQU *-1      ADDRESS OF MLMP WAIT(IOCS SETS) B 14740000
CLI     DTFCMP(,DTF),OPACC  COMPLETION POSTED ?          B 14750000
JNE     CHPNTR      NO - BRANCH                B 14760000
.H0400 EJECT                          14770000
*****                                14780000
*          CHECK FOR END OF WAIT LIST      * 14790000
*****                                14800000
SPACE 2                                14810000
CHSTND EQU *                          14820000
&MIX   SETA &MLA+&BSC                14830000
AIF    (&MIX LE '1').Y0100            14840000
TBN    CHLSTS(,LISTRG),CHLAST  END OF WAIT LIST ?          Y 14850000
JT     CHEND      YES - BRANCH          Y 14860000
LA     NEXT(,LISTRG),LISTRG  POINT TO NEXT LIST ENTRY    Y 14870000
B      CHSTRT     GO TEST NEXT ENTRY     Y 14880000
.Y0100 ANOP                          14890000
SPACE 2                                14900000
*****                                14910000
*          PROCESS END OF LIST CONDITIONS  * 14920000
*****                                14930000
SPACE 1                                14940000
CHEND  EQU *          *                          14950000
L      DTFADR(,LISTRG),DTF  POINT XR2 TO DTF          14960000
CLI    DTFCMP(,DTF),OPACC  IS CODE OP ACCEPTED ?      14970000
JNE    CHNOOP      NO - BRANCH          14980000
SBN    CHLSTS(,LISTRG),OPGONE  SET BIT FOR DESTROYED CODE  14990000
CHNOOP EQU *          *                          15000000
MVI    DTFCMP(,DTF),INACTV  SET COMPLETION TO NO ACT DTF'S  15010000
** THE FOLLOWING BRANCH IS SET TO NOOP IF AN ACTIVE DTF IS FOUND ** 15020000
CHACTV EQU *+1      ACTIVE DTF SWITCH          15030000
BC     CHPNTR,UNCOND  BRANCH IF NO ACTIVE DTF'S  15040000
MVI    DTFCMP(,DTF),NOCOMP  SET NO COMPLETION RETURN CODE  15050000
EJECT                          15060000
*****                                15070000
*          RESTORE POINTERS AND EXIT      * 15080000
*****                                15090000
SPACE 3                                15100000
CHPNTR EQU *          *                          15110000
L      DTFADR(,LISTRG),DTF  POINT REGISTER TO DTF          15120000
CHSTOR EQU *+3      RETURN ADDRESS PLUGGED HERE  15130000
B      #            RETURN                15140000
TITLE '$E092/CMERPC--CHECK FOR PUT TO TERMINAL IN ERP'  15150000
*****                                15160000
* NAME -- CMERPC                      * 15170000
*                                     * 15180000
* TITLE - CHECK FOR PUT TO TERMINAL IN ERP AND HANDLE IT. * 15190000
*                                     * 15200000
* OPERATION -                          * 15210000
* CHECK FOR ERP.                        * 15220000
* IF TERMINAL IN ERP GOTO TRANSIENT $CC4MP. * 15230000
* ON RETURN , CHECK FOR PUT THEN GET.    * 15240000
* IF ONLY PUT - IGNORE IT AND GO POST REQUESTER. * 15250000
* IF PUT THEN GET - RETURN TO ALLOW SCHEDULING OF GET. * 15260000
*                                     * 15270000

```

```

* INPUT : XR1 - PL ADDRESS * 15280000
* * 15290000
* OUTPUT : XR1 - PL, XR2 - TUB. * 15300000
* * 15310000
* EXITS : IF PUT ONLY TO TERMINAL IN ERP - CMPAII * 15320000
* IF TERMINAL NOT IN ERP OR NOT PUT ONLY - RETURN * 15330000
***** 15340000
SPACE 1 15350000
CMERPC EQU * * 15360000
ST CMERPR+3,ARR SAVE RETURN ADDRESS 15370000
L PLTUBA(,PL),XR2 TUB ADDRESS 15380000
TBN TUBAT3(,XR2),TUBERP IS TUB IN CCP ERP 15390000
TBN PLOPC(,PL),OPPUT DOES OP INVOLVE PUT 15400000
TBF PLOPM(,PL),OPSTOP TEST FOR NOT PURGE OPERATION 15410000
JF CMERPR JUMP IF PUT WILL NOT BE IGNORED 15420000
SPACE 15430000
***** 15440000
* IGNORE PUT OPERATION BECAUSE TERMINAL IN ERROR RECOVERY 15450000
***** 15460000
SPACE 15470000
SVC 0 ##### TRANSIENT CALL ##### 15480000
DC AL1(CCPTRIB) CCP SVC RIB 15490000
DC AL1(CC4MP) IGNORE PUT TRANSIENT 15500000
* RETURNS TO ARR IF PUT ONLY 15510000
* TO ARR+4,IF PUT THEN INVITE 15520000
SPACE 15530000
B CMPAII BRANCH TO PERFORM POST CHECK 15540000
* IF THE OPERATION INVOLVED A READ - THE TRANSIENT WILL RETURN 15550000
* HERE -- CONTINUE TO DROP THROUGH AND SCHEDULE THE READ. 15560000
SPACE 15570000
CMERPR B *-* RETURN 15580000
MEND 15590000

```

```

MACRO 00010000
***** 00020000
* NAME: $E095 * 00030000
***** 00040000
$E095 &DMESTR- 00050000
  GBLB &ONE ,&NOB ,&NOM ,&MIN ,&NDME 00060000
  GBLB &NPP ,&NMP ,&NSWL ,&NCS ,&NITB ,&NTSP ,&N32 ,&NAS ,&NDF ,&NCPU 00070000
  GBLB &NMSG ,&N37 ,&NINT ,&N41 00080000
  GBLB &NSW ,&N1050 ,&N2741 ,&NMOVE ,&NSCTL ,&NBFR 00090000
  LCLA &MIX 00100000
  TABLE &DMESTR . DATA MODE ESCAPE STRING 00110000
' ' TABDF CL6'////////' . DEFAULT STRING FOR CURRENT USE 00120000
  TEXT 00130000
* R-06,C-00 CHANGE LEVEL 00140000
  TITLE '$E095---WORK AREA' 00150000
***** 00160000
* CM WORK AREA * 00170000
***** 00180000
  SPACE 2 00190000
X$FFFC DC XL2'FFFC' CONSTANT OF FFFC 00194000
GMFAIL DC XL2'0020' PSR SETTING = BINARY OVERFLOW 00196000
CMSWIT DC BL1'0' SET OF EIGHT BINARY SWITCHES 00200000
CMFMPS EQU BIT0 CM WAS POSTED BY $CC4FM 00210000
CMRSLN EQU BIT1 1-INDICATES LINE CAN BE RESCHED 00220000
CMTPRQ EQU BIT2 1-INDICATES REQUEST JUST 00230000
* SCHEDULED WAS RESULT OF TP REQ 00240000
* POST BIT BEING SET 00250000
CMNOST EQU BIT4 1-ONLINE TEST START FAILED 00260000
CMBSCK EQU BIT5 1-BSCA DTF SET UP FOR CHECK CALL 00270000
CMSPSI EQU BIT6 1-CM HAS HANDLED STOP INVITE TO 00280000
* * A STATUS POLL TERMINAL. 00290000
* -----START-----@23 00292000
CMRFCK DC XL2'0000' PRUF PGM LENGTH CHECK FIELD 00294000
CMSEVN DC XL2'0007' PGM LENGTH FOR PRUF COMPARE 00296000
* -----END-----@23 00298000
  SPACE 2 00300000
* ECB LIST USED BY $CC4CM TO WAIT FOR MORE WORK 00310000
  SPACE 1 00320000
ECBLST EQU * * WAIT LIST FOR CM 00330000
  DC AL2($CMECB) ECB FOR II,IB,IM,DF TO POST CM. 00340000
  DC AL2($CMFM) ECB FOR $CC4FM TO POST THAT 00350000
* * FREEMAIN HAS OCCURRED. 00360000
  AIF (&NINT).CT030 00370000
  DC AL2(TIMIOB+7) INTERVAL TIMER ECB NB 00380000
.CT030 ANOP 00390000
  DC XL2'FFFF' 00400000
  SPACE 1 00410000
  AIF (&NDME).D5100 00420000
  AIF (&NOM).M2100 00430000
* SPECIAL TRANSLATE LIST FOR DATA MODE ESCAPE COMMAND CHECK DM 00440000
* IT IS SET UP TO TRANSLATE 6 BYTES INTO A PRE-DEFINED FIELD DM 00450000
* ALL FIELDS ARE PRE-SET EXCEPT FROM ADDRESS. DM 00460000
  SPACE 00470000
CMDMTL EQU * DATA MODE ESCAPE TRANS LIST DM 00480000
  DC AL2(DMELEN) FROM LENGTH DM 00490000
  DC AL2(CMDMEB) TO ADDR DM 00500000
  DS AL2 FROM ADDR DM 00510000
  DC IL2'6' TO LENGTH DM 00520000
  DS XL1 RETURN CODE DM 00530000

```

```

SPACE 2 00540000
.M2100 ANOP 00550000
AIF (&NOM).M2200 00560000
CMDMEB EQU * DATA MODE ESCAPE CHK REC AREA DM 00570000
DS CL6 TRANSLATE 1ST 6-BYTES HERE. DM 00580000
.M2200 ANOP 00590000
.D5100 ANOP 00600000
SPACE 00610000
* GETMAIN PARAMETER LIST 00620000
SPACE 1 00630000
GMLIST EQU * START OF GETMAIN PARM LIST 00640000
DS AL2 GETMAIN ADDRESS 00650000
DS XL2 SIZE OF GETMAIN AREA 00660000
SPACE 2 00670000
* SAVE AREAS 00680000
SPACE 2 00690000
* THE FOLLOWING 4 BYTES MUST REMAIN TOGETHER FOR PARAMETER LIST 00700000
CMOCPL EQU * * 4 BYTE PL FOR MLMP OPEN/CLOSE 00710000
CMSPL DS AL2 SAVE AREA FOR PARM LIST ADDR 00720000
CMSDTF DS AL2 SAVE AREA FOR TP DTFADDR 00730000
SAVCAT DC XL4'0' SAVE AREA FOR IOBCAR + IOBTAR B 00735000
SPACE 1 00740000
CMNWPL DC AL2(0) ADDR OF NO WAIT OP PL FROM II 00750000
CMLPL DC AL2(0) LAST READ PL SCHEDULED BY CMIVGM 00760000
SPACE 1 00770000
AIF (&NOB).NB030 00780000
CCPUT DC XL1'02' PUT OP CODE (FOR REFRESH) B 00785000
AIF (&MIN).N0200 00790000
.NB030 ANOP 00800000
* SAVE AREA FOR TERMINAL ATTRIBUTES R 00810000
* NOTE: SAVTA1 AND SAVTA2 ARE SET BY ONE MOVE INTO SAVRCL OR R 00820000
* SAVTA2. R 00830000
SAVTA1 DC XL1'0' TERMINAL ATTRIBUTE SET BYTE 1.R 00840000
SAVTA2 DC XL1'0' TERMINAL ATTRIBUTE SET BYTE 2.R 00850000
AIF (&NOB).C0585 00860000
SAVRCL DC XL2'00' TERM ATTR SET RECORD LENGTH. RB 00870000
SPACE 1 R 00880000
.C0585 ANOP 00890000
CMSPHY DC AL1(0) SAVE AREA FOR TUBPHY. R 00900000
#BUFND DS CL2 CONTAINS STORAGE BUFFER R 00910000
* * SIZE NEEDED. R 00920000
.N0200 ANOP 00930000
CMIIND DS XL2 SIZE OF BUFFER NEEDED FOR LINE 00940000
AIF (&NOM).M2210 00950000
CMTCHR DS XL1 SAVE AREA- TUB CHARACTERISTICS M 00960000
AIF (&NSCTL).S5940 00970000
CMSTMA DS XL2 SAVE AREA FOR MLTA TERM ADDR CM 00980000
* HAS BEEN REQUESTED CM 00990000
.S5940 ANOP 01000000
SPACE 01010000
.M2210 ANOP 01020000
* ADDRESS PLUGGED BY MLMP/MLTA IOCS - NO EXPLICITLY USED BUT 01030000
* THE IOCS(S) ASSUME IT IS ALWAYS IN $$BMCH WHICH WE REPLACE 01040000
CHBMHL DC XL3'0' BOGUS DC FOR BRACKET HALT 01050000
AIF (&NOM).M2240 01060000
CHBMPC EQU *-1 BOGUS DC FOR PCI INTERRUPTS M 01070000
.M2240 ANOP 01080000
SPACE 3 01090000
***** 01100000
* CM CONSTANT AREA * 01110000

```

```

*****
SPACE 1 01120000
AIF (&NSCTL).S5950 01130000
MLSTCL DC AL1($MTPLT) STATION CONTROL BIT CM 01140000
.S5950 ANOP 01150000
AIF (&NOM).M2250 01160000
CMTYPZ DC IL2'16' NUMBER OF TYPEWRITER CONTROL M 01170000
* CHARACTERS ADDED TO MESSAGES M 01180000
* THIS FIELD CHANGED IF BUFF REC M 01190000
* TERMINALS IN THE SYSTEM M 01200000
* (1 CARRIAGE RET + 1-15 IDLES) M 01210000
SPACE 01220000
AIF (&N1050).N2240 01230000
SPACE 01240000
* MULTI-COMPONENT TERMINAL TABLE * 1M 01250000
MCTABL EQU * MCT TABLE 1M 01260000
DC XL1'0B' KEYBOARD INDEX 1 1M 01270000
DC XL1'0D' READER 1 INDEX 2 1M 01280000
DC XL1'0E' READER 2 INDEX 3 1M 01290000
DC XL1'15' ANY INPUT DEVICE INDEX 4 1M 01300000
DC XL1'02' PRINTER 1 INDEX 5 1M 01310000
DC XL1'04' PRINTER 2 INDEX 6 1M 01320000
DC XL1'07' PUNCH 1 INDEX 7 1M 01330000
DC XL1'06' PUNCH 2 INDEX 8 1M 01340000
DC XL1'13' ALL OUTPUT COMPONENTS INDEX9 1M 01350000
SPACE 2 01360000
.N2240 ANOP 01370000
.M2250 ANOP 01380000
SPACE 1 01390000
AIF (&NOB).C0590 01400000
FNDEOT DC AL2(BBUCKET) @ OF BIT BUCKET AREA FOR SEARCHB 01410000
BBUCKET DC XL1'00' DUMMY AREA FOR SEARCH EOT. B 01420000
SPACE 2 01430000
&MIX SETA &NITB+&NTSP+5*&NDF 01440000
AIF (&MIX NE '5').T1300 01450000
TWO DC AL2(2) ITB/TRANSP BUMP CONSTANT. IXB 01460000
.T1300 ANOP 01470000
AIF (&MIN).NX020 01480000
FIVE DC XL2'0005' CONSTANT OF FIVE. RB 01490000
.NX020 AIF (&NSWL).S0100 01500000
THREE DC AL2(3) NSI BUMP CONSTANT. SB 01510000
.S0100 ANOP 01520000
OLTLNG DC AL2(7) LENGTH OF OLT PARM LIST. B 01530000
AIF (&MIN).N0400 01540000
IOBLEN DC AL2(IOBL) LENGTH OF BSCA IOB. RB 01550000
.N0400 ANOP 01560000
LINFO EQU IOBL MAX # LINE CONTROL CHARS MLMP. B 01570000
&MIX SETA &N32+&N37+&N41 01580000
AIF (&MIX EQ '3').T1400 01590000
RELSNS DC XL2'0014' MAX RECL FOR STATUS POLL. 0/57B 01600000
AIF (&N32).T1400 01610000
MAXMSG DC AL2(80+FORMTL) MAX SPACE FOR SYSTEM MSG. 0B 01620000
CMSTUS DC AL2(SSID) CHECK VALUE FOR 3270 STATUS MS0B 01630000
.T1400 ANOP 01640000
&MIX SETA &NAS+&N32 01650000
AIF (&MIX NE '0').T1500 01660000
CMSTAS DC XL2'2552' ASCII '%R', STATUS MSG. 0AB 01670000
.T1500 ANOP 01680000
AIF (&N37).T1600 01690000
CMNULS DC XL3'00E200' NUL-S-NUL HEADER, 3735 STATUS.5B 01700000
01710000

```


CMNULA	AIF	(&NAS).T1600			01720000
.T1600	DC	XL3'005300'	NUL-S-NUL (ASCII).	5AB	01730000
	ANOP				01740000
CMSME	AIF	(&N41).N3741			01741000
	DC	CL2'%V'	PERCENT-V HEADER,3741 STATUS	7B	01742000
	AIF	(&NAS).N3741			01743000
CMSMA	DC	XL2'2656'	PERCENT-V HEADER (ASCII)	7B	01744000
.N3741	ANOP				01745000
	AIF	(&NINT).CT040			01750000
TIMOPE	DC	XL1'01'	TIMER INTERNAL SWITCH	NB	01790000
VALOPE	EQU	BIT7	BIT 7=1 - DON'T WAIT ON TIMER	NB	01800000
*			* AN OP END OCCURRED.	NB	01810000
POL1MN	EQU	BIT6	BIT 6=1 - POLL UNTIL TIME = 0	NB	01820000
	SPACE	1			01830000
TIMIOB	EQU	*	INTERVAL TIMER IOB	NB	01840000
	DC	XL1'0'	FLAG BYTE	NB	01850000
*			02 = TIME IS IN TIMER UNITS	NB	01860000
*			(1 TU = 3.33 MS)	NB	01870000
*			FF = CANCEL REMAINING TIME	NB	01880000
	DC	XL6'000000000258'	TIME(INITIAL = 2 SEC.)	NB	01890000
	DC	XL3'0'	ECB FOR TIMER	NB	01900000
.CT040	ANOP				01910000
.C0590	ANOP				01920000
	SPACE	1			01930000
	AIF	(&NDME).D5120			01940000
CMDME	DC	&DMESTR	DATA MODE ESCAPE STRING	D	01950000
.D5120	ANOP				01960000
PLGMLG	DC	AL2(PLLEN+4)	PL GETMAIN LENGTH INCL FREELIST		01970000
	TITLE	'\$E095---EQUATES'			02070000
* COMMON MLTA/BSCA EQUATES					02080000
	SPACE				02090000
GMCCP	EQU	0	BEGINNING OF GETMAIN PARM LIST		02100000
GMADDR	EQU	GMCCP+2-1	GETMAIN ADDRESS		02110000
GMSIZE	EQU	GMADDR+2	SIZE OF GETMAIN AREA		02120000
	SPACE				02130000
DMELEN	EQU	6	LENGTH OF DME STRING	D	02140000
TLERR	EQU	BIT7	BIT IN TRANSLATE RETURN CODE TO		02150000
*			INDICATE INVALID CHAR TRANSLATED		02160000
HEX512	EQU	X'02'	VALUE OF HEX 02		02165000
DTF	EQU	2	DTF REG.		02170000
PL	EQU	1	PARAMETER LIST REGISTER.		02180000
TUB	EQU	1	PTR TO TERMINAL UNIT BLOCK.		02190000
SWAPFR	EQU	X'01'	MOVE TYPE FOR FROM ATR SWAP		02200000
SWAPTO	EQU	X'02'	MOVE TYPE FOR TO ATR SWAP		02210000
TRMTSK	EQU	X'E3'	ID OF TERMINATIONS TASK		02215000
*		EQUATES FOR COMMON CHECK		*	02220000
	SPACE	1			02230000
DTFADR	EQU	2	DISPLACEMENT TO DTF IN WAIT LIST		02240000
LISTRG	EQU	1	WAIT LIST POINTER - REGISTER 1		02250000
NEXT	EQU	3	DISPLACEMENT TO NEXT LIST ENTRY		02260000
NOBSCA	EQU	X'00'	MASK TO TEST FOR BSCA PRESENCE		02270000
NOMLTA	EQU	X'00'	MASK TO CHECK FOR NO MLTA PRESENT		02280000
OPGONE	EQU	X'08'	CODE DESTROYED BIT		02290000
UNCOND	EQU	X'87'	UNCONDITIONAL Q-CODE FOR BRANCH		02300000
CHLSTS	EQU	0	CHECK LIST STATUS BYTE		02310000
CHLSKP	EQU	X'80'	CHECK LIST: SKIP ENTRY BIT		02320000
CHLAST	EQU	X'20'	CHECK LIST: LAST ENTRY INDICATOR		02330000
	SPACE	1			02340000
*		DTF EQUATES.			02350000
	SPACE	1			02360000

DTFDEV	EQU	0	DTF DEVICE TYPE.	02370000
DTFATR	EQU	3	DTF ATTRIBUTE BYTE.	02380000
FILEAC	EQU	X'02'	FILE ACTIVE.	02390000
FILOPN	EQU	X'01'	FILE OPENED.	02400000
DTFCMP	EQU	14	DTF COMPLETION CODE.	02410000
OPACC	EQU	X'00'	OPERATION ACCEPTED.	02420000
NOCOMP	EQU	X'56'	NO COMPLETED DTF'S.	02430000
INACTV	EQU	X'57'	NO ACTIVE DTF'S.	02440000
	SPACE	1		02450000
* SVC	RIB	EQUATES		02460000
	SPACE	1		02470000
POSTRB	EQU	X'05'	SVC RIB FOR POST SINGLE ECB.	02480000
WAITRB	EQU	X'04'	SVC RIB FOR WAIT ON ECB LIST.	02490000
MVRIB	EQU	X'02'	SVC SUBRIB FOR MOVE ROUTINE	02500000
GMRIB	EQU	X'03'	SVC SUBRIB FOR CCP GETMAIN.	02510000
FMRIB	EQU	X'04'	SVC SUBRIB FOR CCP FREEMAIN	02520000
TRRIB	EQU	X'09'	SVC SUBRIB FOR TRACE	02530000
LDRIB	EQU	X'80'	SVC RIB FOR LOAD DSM XIENT (C/S)	02540000
TTMRIB	EQU	X'16'	SVC RIB FOR TEST TIMER	02550000
STMTRIB	EQU	X'15'	SVC RIB FOR START TIMER	02560000
CCPRIB	EQU	X'01'	SVC RIB FOR CCP REQUEST.(\$CC4IG)	02570000
	SPACE	2		02580000
FLSNEQ	EQU	X'96'	CONDITION: FALSE OR NOT EQUAL.	02590000
FLSNLO	EQU	X'95'	CONDITION: FALSE OR NOT LOW.	02600000
FLSOEQ	EQU	X'91'	CONDITION: FALSE OR EQUAL.	02610000
FLSOHI	EQU	X'94'	CONDITION: FALSE OR HI.	02620000
TRUAEQ	EQU	X'16'	CONDITION; TRUE AND EQUAL.	02630000
TRUALO	EQU	X'15'	CONDITION: TRUE AND LOW.	02640000
TRUNEQ	EQU	X'11'	CONDITION: TRUE AND NOT EQUAL.	02650000
TRUAHI	EQU	X'13'	CONDITION: TRUE AND HIGH.	02655000
	AIF	(&NOM).M2400		02660000
	EJECT			02670000
* MLTA	EQUATES		M	02680000
	SPACE			02690000
CMCR	EQU	X'5B'	LINE CODE CARRIAGE RET CHAR	M 02700000
CMIDLE	EQU	X'5E'	LINE CODE IDLE CHARACTER	M 02710000
CMCRTM	EQU	73	MAX INPUT TO ALLOW CARRIAGE TO	M 02720000
*			RETURN WITHOUT IDLE CHAR	M 02730000
MLRSET	EQU	X'01'	RESET BIT IN OP CODE	M 02740000
MLCONT	EQU	X'02'	CONTINUE OP CODE BIT	M 02750000
MLCONV	EQU	X'04'	CONVERSATIONAL OP CODE BIT	M 02760000
MLREAD	EQU	BIT3	MLTA READ OP BIT	M 02770000
MLWRIT	EQU	BIT2	MLTA WRITE OP BIT	M 02780000
TLCREL	EQU	BIT6	BIT IN RETURN CODE TO INDICATE	M 02790000
*			CARRIAGE RETURN AT END OF LINE	M 02800000
CMTYPL	EQU	X'10'	# OF TYPEWRITER CHARS ADDED	M 02810000
*			TO MESSAGE IF REQUESTED BY USERM	02820000
CMTYBF	EQU	1	NUMBER OF TYPEWRITER CONTROL	M 02830000
*			CHARS TO ADD FOR BUFFERED REC	M 02840000
MLSCCS	EQU	X'FD'	BAD BITS IN RET CODE FROM MLTA	M 02850000
MLCITP	EQU	X'06'	TERMINAL INTERRUPT	M 02860000
.M2400	ANOP			02870000
	SPACE	3		02880000
	AIF	(&NOB).C0600		02890000
	EJECT			02900000
* BSCA	EQUATES		B	02910000
	SPACE			02920000
BLANK	EQU	X'40'	MASK FOR A BLANK CHARACTER.	B 02930000
BSCA	EQU	X'80'	Q CODE FOR BSCA.	B 02940000
DONE	EQU	X'40'	NORMAL COMPLETION.	B 02950000

POLBIT	EQU	X'20'	ON-ADDRESSING, OFF-POLLING.	B	02960000
READY	EQU	X'84'	IOB READY FOR TRANSMIT.	B	02970000
SBF1	EQU	X'7B'	SBF BASED ON XR1.	B	02980000
SBN1	EQU	X'7A'	SBN BASED ON XR1.	B	02990000
SBF2	EQU	X'BB'	SBF BASED ON XR2.	B	03000000
SBN2	EQU	X'BA'	SBN BASED ON XR2.	B	03010000
STPRT	EQU	X'08'	MASK FOR START PRINT BIT IN WCC		03015000
TXTSNT	EQU	X'01'	INITIAL SETTING TEXT INDICATORSB		03020000
WCC	EQU	2	DISPLACEMENT OF WCC IN PLRECA.		03025000
IBX	EQU	2	IOB WORK REG FOR BUFFER CARVE.	B	03030000
IOB	EQU	1	PTR TO INPUT/OUTPUT BLOCK	B	03040000
POL	EQU	1	POLL LIST WORK REGISTER.	B	03050000
WKA	EQU	1	WORK AREA REGISTER.	B	03060000
WORK	EQU	1	GENERAL WORK REG.	B	03070000
FC	EQU	X'FC'	USED FOR LESS THAN 3 CHECK	FB	03075000
POLSKP	EQU	X'80'	SKIP BIT FOR POLLING LIST	CB	03080000
CPURGE	EQU	X'59'	CCP ABORT COMPLETION CODE	B	03090000
NSECS	EQU	7	NUMBER OF SECTORS-1 FOR C/S	B	03100000
*			TRANSIENT LOAD	B	03110000
	AIF	(&NINT).CT050			03120000
*	EQUATES FOR TIMER INITIATED POLLING			NB	03130000
	SPACE	1			03140000
TIFLAG	EQU	0	TIMER FLAG BYTE	NB	03150000
*			02=TIME IS IN TIMER UNITS	NB	03160000
*			FF=CANCEL REMAINING TIME	NB	03170000
TITIME	EQU	6	TIME FIELD	NB	03180000
.CT050	ANOP				03190000
.C0600	ANOP				03200000
	MEND				03210000

MODULE-\$E087 , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
* NAME: $E087 * 00030000
***** 00040000
$E087 00050000
  GBLB &ONE, &NOB, &NOM, &MIN, &NDME 00060000
  GBLB &NSW, &N1050, &N2741, &NMOVE, &NSCTL, &NBFR 00070000
  TEXT 00080000
* R-05, C-00 CHANGE LEVEL 00100000
  AIF (&NOM).M1400 00110000
  TITLE '$E087/CMMSCH---RESCHEDULE-A-MLTA-LINE' 00120000
***** 00130000
* * 00140000
* NAME--CMMSCH * 00150000
* * 00160000
* TITLE--RESCHEDULE A MLTA LINE * 00170000
* * 00180000
* FUNCTION--TO SCHEDULE WORK ON LINES THAT NOT BUSY, OR TO STOP * 00190000
* POLLING ON A LINE AND SCHEDULE OF PUT OPERATION IF PUT * 00200000
* OPERATIONS ARE PENDING IN THE LINE QUEUE. * 00210000
* * 00220000
* OPERATION-- * 00230000
* * 00240000
* . IF MLTA OLT IS IN PROCESS CALL IN TRANSIENT TO CONTINUE * 00250000
* THE OLT OPERATION. * 00260000
* * 00270000
* . IF QUEUE FOR LINE IS EMPTY THEN POST THE REQUESTOR IF * 00280000
* TP WAS JUST SCHEDULED, CHECK FOR OTHER FUNCTION THAT * 00290000
* CAN BE PERFORMED BY 'CM'. * 00300000
* * 00310000
* . IF A PUT OPERATION CAN BE STARTED, PERFORM IT NEXT. * 00320000
* * 00330000
* . IF NEITHER OF ABOVE TWO, THEN TRY TO SCHEDULE AN INPUT * 00340000
* OPERATION. IF NO INPUT TO SCHEDULE THEN POST THE * 00350000
* REQUESTOR IF TP WAS JUST SCHEDULED. THEN CHECK FOR * 00360000
* OTHER 'CM' FUNCTION THAT CAN BE PERFORMED NOW. * 00370000
* * 00380000
* . IF A WRITE OPERATION CAN BE STARTED, THEN SET UP THE * 00390000
* OUTPUT DATA BUFFER FOR THE LINE. TRANSLATE THE DATA * 00400000
* AS REQUIRED OR SPECIFIED. ADD DEVICE DEPENDENT * 00410000
* CONTROL CHARACTERS. * 00420000
* * 00430000
* . IF MLTA OPERATION SET UP TO DO THE READ, WRITE WITH * 00440000
* THE APPROPRIATE DISCONNECT, CONTINUE OR CONVERSATIONAL * 00450000
* OPTION AS REQUIRED BY THE TERMINAL AND/OR CURRENT * 00460000
* LINE CIRCUMSTANCE. (SUBROUTINE: CMFORM). * 00470000
* * 00480000
* . ISSUE OF IOCS CALL AFTER THE DTF IS SET-UP. * 00490000
* CALL MLTIO1. * 00500000
* * 00510000
* . AFTER ISSUING THE IOCS CALL, TRACE IT. * 00520000
* * 00540000
* . IF MLTA START OPERATION CODE INDICATES A START * 00550000
* FAILURE THE CAM IN A TRANSIENT TO DETERMINE THE * 00560000
* PROBLEM. OTHERWISE, POST THE REQUESTOR IF TP WAS * 00570000
* JUST SCHEDULED. * 00580000
* * 00590000
* ENTRY POINTS * 00600000
* CMMSCH - SCHEDULE WORK ON A MLTA LINE * 00610000
```

```

* 00620000
* INPUT-- * 00630000
* CMSDTF - ADDRESS OF DTF TO BE SCHEDULED. * 00640000
* CMSPL - ADDRESS OF TP PARAMETER LIST TO BE SCHEDULED. * 00650000
* * 00660000
* OUTPUT-- * 00670000
* CMSDTF - ADDRESS OF DTF FOR LINE SCHEDULED. * 00680000
* CMSPL - ADDRESS OF TP REQUEST SCHEDULED. * 00690000
* DTF(LCB),TUB - SET UP FOR THE OPERATION PERFORMED. * 00700000
* * 00710000
* EXTERNAL REFERENCES-- * 00720000
* $CC4MT - HANDLE MLTA OLT CONTINUATION. * 00730000
* $CC4JX - X IS THE APPROPRIATE LINE TRANSLATE TRANSIENT. * 00740000
* $CC4WR - HANDLE TRANSLATE ERRORS IN OUTPUT. * 00750000
* $CC4WC - CONNECT SWITCHED LINE WTO. * 00760000
* MLTIO1 - MLTA IOCS. * 00770000
* $TRACE - CCP TRACE ROUTINE. * 00780000
* $CC4SC - MLTA START CODE FAILURE TRANSIENT. * 00790000
* CMSRPL - SEARCH FOR NEXT OP TO SCHEDULE ON THE LINE. * 00810000
* CMCRID - ADD CARRIAGE RETURN AND IDLES. * 00820000
* CMMVRT - INTERFACE TO CCP MOVE ROUTINE. * 00830000
* * 00840000
* EXIT, NORMAL-- * 00850000
* - TO CMPAII - CHECK FOR POSTING OF THE REQUESTER. * 00860000
* - TO CMRQBF - CHECK FOR MORE WORK TO DO. * 00870000
* - TO CMREQ - IF NEW TP REQUEST IS FOUND IN THE QUEUE. * 00880000
* * 00890000
***** * 00900000
EJECT * 00910000
CMMSCH EQU * * < ENTRY POINT > M 00920000
L CMSDTF,DTF POINT XR2 TO DTF M 00930000
SPACE * 00940000
* SET DEFAULT TO RESCHEDULE NEEDED - FREE ANY INPUT HOLD BUFFER M 00941400
SBN LCBATR(,DTF),LCBTIM SET RESCHEDULE NEEDED M 00942100
CLI LCBIBA-1(,DTF),NOBIT ANY HOLD BUFFER? M 00942800
JE CMMSNF NO - NO FREE NEEDED M 00943500
* FREE THE HOLD BUFFER M 00944200
L LCBIBA(,DTF),XR2 XR2--> AREA TO FREE M 00944900
B CMFMRT GO FREE IT M 00945600
L CMSDTF,DTF XR2--> DTF M 00946300
MVI LCBIBA-1(,DTF),NOBIT ZERO HOLD BUFFER ADDRESS M 00947000
B CMFRMN GO CHECK FOR PL NEEDING CORE M 00947700
CMMSNF EQU * * M 00948400
CLI LCBNW#(,DTF),NOBIT ANY NEW TP REQUESTS ? M 00950000
BNE CMPAII YES - GO HANDLE NOW M 00952000
SBF LCBATR(,DTF),LCBTIM SET OFF RESCHEDULE NEEDED M 00954000
SPACE * 00970000
CLI LCBOLT(,DTF),NOBIT IS ONLINE TEST COUNT ZERO M 00980000
JE CMLKPL JUMP IF YES TO LOOK AT PL M 00990000
SPACE * 01000000
***** * 01010000
* ON-LINE TEST REQUEST TO BE STARTED M 01020000
***** * 01030000
SPACE * 01040000
SVC 0 ##### TRANSIENT CALL ##### M 01050000
DC AL1(CCPRIB) CCP SVC RIB. M 01060000
DC AL1(CC4MT) START ONLINE TEST M 01070000
SPACE * 01080000
* RETURN TO NSI - ONLINE TEST NOT STARTED. M 01090000
* NSI+4 - ONLINE TEST STARTED. M 01100000

```

```

SPACE 01110000
*--> NSI. M 01120000
SBN CMSWIT,CMNOST INDICATE LINE I/O NOT STARTED M 01130000
SPACE 01140000
*--> NSI+4. M 01150000
B CMPAII JUMP TO SEE IF POST NEEDED M 01160000
SPACE 01170000
CMLKPL EQU * * LOCAL M 01180000
SPACE 01190000
*****M 01200000
* SEARCH LINE QUEUE FOR NEXT REQUEST TO BE SCHEDULED. M 01210000
*****M 01220000
SPACE 01230000
B CMSRPL LCB PL QUEUE SEARCH ROUTINE M 01240000
* XR1-> PL, XR2-> TUB. M 01250000
SPACE 01260000
MVC CMTCHR(1),TUBCHR(,XR2) MOVE TUB CHAR TO SAVE AREA M 01270000
AIF (&NBFR).H4467 01280000
MVC TUBVHR(1,XR2),TUBCHR(,XR2) SAVE TUBCHR FIELD BM 01290000
.H4467 ANOP 01300000
AIF (&NSCTL).S5400 01310000
MVC CMSTMA(2),TUBTMA(,XR2) SAVE TERMINAL ADDRESS CM 01320000
.S5400 ANOP 01330000
MVC SAVTA2(2),TUBTA2(,XR2) SAVE TERMINAL ATTRIBUTES M 01340000
SPACE 01350000
L TUBDTF(,XR2),DTF XR2--> DTF FOR THIS LINE. M 01360000
SPACE 01370000
* DETERMINE IF OPERATION IS READ OR WRITE M 01380000
SPACE 01390000
TBN PL$OPM(,XR1),OPGET IS IT READ ? M 01400000
JF CMMPUT NO - GO HANDLE THE PUT. M 01410000
B CMIVGM GO DETERMINE INPUT OP TO START.M 01420000
SPACE 1 01430000
L CMLPL,PL LAST READ PL SCHEDULED - WILL M 01434000
* * GO INTO TRACE FROM CMSPL M 01436000
B CMMSIO GO ISSUE SIO FOR THE INPUT. M 01440000
TITLE '$E087/CMMSCH----START-A-WRITE-OPERATION' 01450000
***** M 01460000
* WRITE REQUEST TO START ON MLTA LINE * M 01470000
***** M 01480000
SPACE 01490000
CMMPUT EQU * * LOCAL 01500000
AIF (&NSW).A5300 01510000
* IF THIS IS PUT DISCONNECT - THEN THERE IS NO DATA TO BE TRANSMITED SM 01520000
SPACE 01530000
TBN PLOPM(,XR1),OPDISC IS THIS PUT DISCONNECT SM 01540000
BT CMMSIO JUMP IF PUT DISCONNECT SM 01550000
SPACE 01560000
.A5300 ANOP 01570000
SPACE 01580000
***** M 01590000
* SETUP PARAMETER LIST FOR MOVE OR TRANSLATE * M 01600000
***** M 01610000
SPACE 01620000
MVC #CMTRL+TLFRMA(2),PLRECA(,XR1) MOVE FROM ADDR TO TRAN LST M 01630000
MVC #CMTRL+TLTOA(2),$MDCRA(,DTF) MOVE TO ADDR TO TL/MV LIST M 01640000
SPACE 01650000
MVC #CMTRL+TLTOL(2),LCBBFL(,DTF) MOVE SIZE OF LINE BUFFER TO M 01660000
* THE TRANS/MOVE LIST M 01670000
SPACE 01680000

```

```

MVI  $MDCRL(,DTF),NOBIT      SET DTF RECORD LENGTH TO ZERO  M 01690000
TBN   CMTCHR,TUBTYP          IS IT TYPEWRITER TERMINAL      M 01700000
JF    CMPWLN                 JUMP IF NOT TYPEWRITER          M 01710000
SPACE                                     01720000
*****                               M 01730000
*   TYPEWRITER TERMINAL                               *   M 01740000
*****                               M 01750000
SPACE                                     01760000
AIF   (&N1050).C5000         01770000
*   IF MULTI COMPONENT (MCT) 1050 TERMINAL, CHECK TO SEE IF 1M 01780000
*   THE COMPONENT BEING WRITTEN TO IS A TYPEWRITER.        1M 01790000
SPACE                                     01800000
TBN   CMTCHR,TUBMCT          IS IT MCT                          1M 01810000
TBF   PL$MCT(,XR1),BIT5      IF BIT OFF; NOT 1050 PRINTER 1M 01820000
TBN   PL$MCT(,XR1),BIT6      IF BIT ON, NOT A 1050 PRINTER 1M 01830000
JT    CMPWLN                 JUMP IF MCT AND NON-PRINTER 1M 01840000
.C5000 ANOP                   01850000
EJECT 01860000
* HAVE TYPEWRITER DEVICE - DETERMINE WHETHER TO ADD START OF LINE M 01870000
* AND/OR END OF LINE TYPEWRITER CONTROL CHARACTERS TO THE OUTPUT LINEM 01880000
SPACE                                     01890000
AIF   (&NBFR).R5200         01900000
*****                               BM 01910000
*   IF NON-BUFFERED TERMINAL,SET CONTROL CHAR LENGTH TO INCLUDE IDLES M 01920000
*****                               BM 01930000
SPACE                                     01940000
MVI   CMTYPZ,CMTYPL          SET LENGTH TO INCLUDE IDLES.  BM 01950000
TBN   $MDTFR(,DTF),$MTBFR    IS IT BUFFERED RECEIVE TERM  BM 01960000
JF    CMTYTS                 JUMP IF NOT BUFFERED RECEIVE  BM 01970000
SPACE                                     01980000
*****                               BM 01990000
*   BUFFERED RECEIVE TERMINAL - NEED NO IDLES AFTER CARRIAGE RETURN. BM 02000000
*****                               BM 02010000
SPACE 1                           02020000
MVI   CMTYPZ,CMTYBF          SET LENGTH TO EXCLUDE IDLES.  BM 02030000
SPACE                                     02040000
.R5200 ANOP                   02050000
CMTYTS EQU *                   * LOCAL                          M 02060000
TBF   PLOPM(,XR1),OPSOL      START OF LINE CHARS WANTED ?  M 02070000
JF    CMTSEL                 JUMP IF NOT WANTED.                M 02080000
SPACE                                     02090000
TBN   CMTCHR,TUB@SL          START OF LINE ISSUED, BUT      M 02100000
TBF   CMTCHR,TUBNID          IDLES NOT NEEDED ?                M 02110000
JT    CMTSEL                 JUMP IF AT START OF LINE AND      M 02120000
*                                     NO IDLES ARE NEEDED.      M 02130000
SPACE                                     02140000
*****                               M 02150000
*   ADD # OF START-OF-LINE CHARACTERS TO MESSAGE LENGTH      *   M 02160000
*****                               M 02170000
SPACE                                     02180000
ALC   $MDCRL(1,DTF),CMTYPZ   ADD # CHARS TO DTF LENGTH      M 02190000
SLC   #CMTRL+TLTOL(2),CMTYPZ REMOVE #CHAR FROM LINE BUF CNT.M 02200000
SBN   CMSLSW+1,NOOP          SET SWITCH: SOL CHARS WANTED. M 02210000
SPACE                                     02220000
CMTSEL EQU *                   * LOCAL                          M 02230000
SBF   CMTCHR,TUB@SL          SET NOT AT START OF LINE BIT.  M 02240000
TBF   PLOPM(,XR1),OPEOL      ARE END OF LINE CHARS REQ'D ?  M 02250000
JF    CMPWLN                 JUMP IF NOT TO BE ADDED.      M 02260000
SPACE                                     02270000
*****                               M 02280000

```

```

* ADD # OF END-OF-LINE CHARACTERS TO MESSAGE LENGTH * M 02290000
***** M 02300000
SPACE 02310000
SBN CMTCHR,TUB@SL SET TERM AT START OF LINE BIT. M 02320000
ALC $MDCRL(1,DTF),CMTYPZ ADD # CHARS TO DTF RECORD LGTH M 02330000
SLC #CMTRL+TLTOL(2),CMTYPZ REMOVE #CHAR FROM LINE BUF CNT.M 02340000
SBN CMELSW+1,NOOP SET SWITCH: EOL CHARS WANTED. M 02350000
EJECT 02360000
***** M 02370000
* DETERMINE LENGTH OF MOVE OR TRANSLATE * M 02380000
***** M 02390000
SPACE 02400000
CMPWLN EQU * * LOCAL M 02410000
SPACE 02420000
* COMPARE REQUESTED LENGTH IN PARM LIST VS AVAILBALE LENGTH IN M 02430000
* TRANSLATED LIST M 02440000
* AVAILABLE LENGTH ALREADY IN PARM LIST M 02450000
SPACE 02460000
CLC PLOUTL(2,XR1),#CMTRL+TLTOL REQ'D VS AVAILABLE LENGTH ?M 02470000
JH CMTYSW JUMP IF AVAILABLE MUST BE USED.M 02480000
MVC #CMTRL+TLTOL,PLOUTL(2,XR1) USE REQUESTED LENGTH. M 02490000
J CMSLSW SKIP SETTING OF TRUNCATED BIT. M 02500000
SPACE 02510000
CMTYSW EQU * * LOCAL M 02520000
SPACE 02530000
SBN PL$RTC(,XR1),RCXDTR SET RET CODE TO DATA TRUNCATED M 02540000
EJECT 02550000
* TYPEWRITER SWITCHES M 02560000
* ON (NO JUMP) MOVE CONTROL CHARACTERS TO START/END OF LINE. M 02570000
* OFF (JUMP) JUMP AROUND ADDING OF CONTROL CHAR. M 02580000
SPACE 02590000
* START OF LINE SWITCH. 02600000
SPACE 02610000
CMSLSW EQU * * LOCAL M 02620000
JC CMELSW,X'00' JUMP IF NO START OF LINE CHARS.M 02630000
SPACE 02640000
***** M 02650000
* ADD START-OF-LINE CONTROL CHARACTERS * M 02660000
***** M 02670000
SPACE 02680000
L #CMTRL+TLTOA,XR1 POINT XR1 AT TARGET ADDR M 02690000
B CMCRID MOVE CARRIAGE RETURN AND IDLES.M 02700000
SPACE 02710000
* DETERINE WHETHER FIRST CHARACTER IS TO BE CARRIAGE RETURN OR M 02720000
* IDLE CHARACTER M 02730000
SPACE 02740000
TBN CMTCHR,TUBNID ARE IDLES NEEDED M 02750000
JF CMADDT JUMP IF IDLES NOT NEEDED M 02760000
MVI 0(,XR1),CMIDLE PLUG IDLE INTO FIRST POSITION M 02770000
SPACE 02780000
CMADDT EQU * * LOCAL M 02790000
ALC #CMTRL+TLTOA(2),CMTYPZ ADD # CHARS TO TARGET ADDR IN M 02800000
* TRANSLATE/MOVE LIST M 02810000
SBF CMSLSW+1,NOOP RESET SOL SWITCH. M 02820000
SPACE 02830000
* END OF LINE SWITCH. M 02840000
SPACE 02850000
CMELSW EQU * * LOCAL. M 02860000
JC CMTSTR,X'00' JUMP IF NO END OF LINE CHARS. M 02870000
SPACE 02880000

```



```

***** M 02890000
* ADD END-OF-LINE CONTROL CHARACTERS * M 02900000
***** M 02910000
SPACE 02920000
L #CMTRL+TLTOA,XR1 POINT XR1 AT 'TO' ADDR M 02930000
A #CMTRL+TLTOL,XR1 ADD LENGTH OF MOVE SO XR1 PTS M 02940000
* AT 1ST BYTE PAST DATA IN LINE. M 02950000
B CMCRID ADD CARRIAGE RETURN AND IDLES. M 02960000
SPACE 02970000
CMTSTR EQU * * LOCAL M 02980000
SBF CMTCHR,TUBNID SET OFF INDICATION FOR IDLES M 02990000
SBF CMELSW+1,NOOP RESET EOL SWITCH. M 03000000
SPACE 03010000
* ADD DATA LENGTH TO DTF LENGTH WHICH NOW SITS AT THE NUMBER M 03020000
* OF TYPEWRITER CONTROL CHARACTERS NEEDED. M 03030000
SPACE 03040000
ALC $MDCRL(,DTF),#CMTRL+TLTOL(1) ADD DATA LENGTH TO DTF M 03050000
L CMSPL,XR1 POINT XR1 AT PARM LIST M 03060000
EJECT 03070000
AIF (&NMOVE).M5300 03080000
* TEST FOR DATA TRANSLATION. (SYSTEM ALWAYS TRANSLATES) VM 03090000
SPACE 03100000
TBN SAVTA1,TASTRN DO WE TRANSLATE VM 03110000
TBF PLOPM(,XR1),OP$SYS IS IT USER REQUEST VM 03120000
JT CMMOVE JUMP IF NO TRANSLATE VM 03130000
SPACE 03140000
.M5300 ANOP 03150000
***** M 03160000
* TRANSLATE DATA FROM REQUESTOR RECORD AREA TO LINE BUFFER * M 03170000
***** M 03180000
SPACE 03190000
LA #CMTRL,XR1 POINT XR1 AT TRANSLATE TABLE M 03200000
MVC CMTWR(1),LCBELC(,DTF) MOVE TRANS TRANSIENT ID M 03210000
SVC 0 ##### TRANSIENT CALL ##### M 03220000
DC AL1(CCPRIB) CCP SVC RIB. M 03230000
CMTWR DC AL1(0) TRANSLATE TRANSIENT ID M 03240000
SPACE 03250000
* XR1 NOW POINTS TO TRANSLATE TABLE AND NOT PARAMETER LIST M 03260000
* PLUG TRANSLATE RETURN CODE INTO RETURN CODE WORK AREA M 03270000
SPACE 03280000
TBN TLRTC(,XR1),TLERR WAS THERE ERROR IN TRANSLATION M 03290000
L CMSPL,XR1 POINT XR1 AT PARM LIST M 03300000
JF CMDOWR JUMP IF NO ERROR M 03310000
SPACE 03320000
* HAD TRANSLATE ERROR - CALL IN TRANSIENT TO HANDLE M 03330000
SPACE 03340000
SVC 0 ##### TRANSIENT CALL ##### M 03350000
DC AL1(CCPRIB) CCP SVC RIB. M 03360000
DC AL1(CC4WR) TRANSLATE ERROR TRANSIENT. M 03370000
SPACE 03380000
B CMPAII GO POST PUT NO-WAIT. M 03390000
SPACE 03400000
AIF (&NMOVE).M5400 03410000
***** VM 03420000
* MOVE DATA WITHOUT TRANSLATE FROM REQUESTOR RECORD AREA TO LINE BUF VM 03430000
***** VM 03440000
SPACE 03450000
CMMOVE EQU * * LOCAL VM 03460000
MVI #CMMVL+MVLTYP,NOBIT SET TYPE FOR NO ATR SWAP. VM 03470000
B CMMVRT BR TO MOVE ROUTINE VM 03480000

```

```

SPACE 03490000
.M5400 ANOP 03500000
CMDOWR EQU * * LOCAL M 03510000
L PLTUBA(,XR1),XR2 POINT XR2 AT THE TUB M 03520000
MVC TUBCHR(1,XR2),CMTCHR RESET TERMINAL CHARACTERISTICS M 03530000
L CMSDTF,DTF POINT XR2 AT DTF M 03540000
TITLE '$E087/CMMSCH---START MLTA READ OR WRITE' 03550000
***** M 03560000
* MLTA IOCS START I/O FOR READ OR WRITE * M 03570000
***** M 03580000
SPACE 03590000
* THE OP CODE IS PLUGGED IN THE DTF - BEING FORMED BASED ON M 03600000
* THE PREVIOUS OPERATIONS ON THE LINE M 03610000
SPACE 03620000
* XR1 POINTS TO THE REQUEST WHICH IS BEING SCHEDULED M 03630000
SPACE 03640000
CMMSIO EQU * * WITHIN CMMSCH. M 03650000
SPACE 03660000
* AT THIS POINT MAKE SURE NEW OP NOT YET MOVED INTO DTF M 03670000
* ALSO LEAVE TERMINAL ADDR OF DTF ALONE SINCE THE PREVIOUS OP AND M 03680000
* TERM ADDR ARE USED IN THIS OP CODE DETERMINATION M 03690000
SPACE 03700000
AIF (&NSW).A5400 03710000
*-----FOR SWITCHED LINES ONLY-----* SM 03720000
SPACE 03730000
TBN PLOPM(,XR1),OPDISC IS THIS PUT DISCONNECT OP SM 03740000
JF CMFO10 JUMP IF NOT PUT DISCONNECT SM 03750000
SPACE 03760000
SBF LCBATR(,XR2),LCBNIT SET BIT TO INDICATE INITIAL OP SM 03770000
* IS NEEDED ON THIS LINE SM 03780000
SPACE 03790000
* IF THIS IS PUT ONLY - THEN PLUG WRITE DISCONNECT OP CODE SM 03800000
* IF THIS IS PUT THEN INVITE - CONTINUE THROUGH THE CODE TO FORM THE SM 03810000
* PROPER READ INITIAL OP CODE SM 03820000
SPACE 03830000
TBN PL$OPM(,XR1),OPINV IS THIS PUT DISC/INVITE SM 03840000
JT CMFO10 JUMP IF PUT DISC/INVITE INPUT SM 03850000
SPACE 03860000
* HAVE PUT DISCONNECT OP SM 03870000
SPACE 03880000
MVI CMOPWK+1,$MWTDS PLUG WITH WRITE DISCONNECT SM 03890000
J CMOPWK JUMP TO PLUG DTF WITH OP SM 03900000
SPACE 03910000
*-----END OF FOR SWITCHED LINES ONLY -----* SM 03920000
.A5400 ANOP 03930000
SPACE 03940000
CMFO10 EQU * * LOCAL. M 03950000
MVI CMOPWK+1,MLREAD MOVE READ OP TO WORK AREA M 03960000
TBN PL$OPM(,XR1),OPGET IS THIS READ OPERATION M 03970000
JT CMTBFR JUMP IF READ OPERATION M 03980000
MVI CMOPWK+1,MLWRIT MOVE WRITE OP TO WORK AREA M 03990000
SPACE 04000000
CMTBFR EQU * * LOCAL. M 04010000
MVC CMPOP+1(1),CMOPWK+1 MOVE CURRENT OP TO TBN INSTR M 04020000
EJECT 04030000
***** 04040000
* DETERMINE IF RESET OPERATION IS NEEDED. * 04050000
***** 04060000
SPACE 04070000
AIF (&NBFR).R5400 04080000

```

```

* ALWAYS RESET IF BUFFERED RECEIVE TERMINAL          BM 04090000
* IF BUFFERED RECEIVE - THEN MUST BE STATION CONTROL BM 04100000
  SPACE                                             04110000
  TBN  $MDTFR(,XR2),$MTBFR      IS THIS BUFFERED RECV TERM  BM 04120000
  JT   CMREST                   JUMP IF BUFFERED RECEIVE    BM 04130000
.R5400 ANOP                                       04140000
  SPACE                                             04150000
  AIF  (&NSCTL).S5900                               04160000
* IF THIS IS 2740 WITH STATION CONTROL AND NO CHECKING - ISSUE RESET CM 04170000
  SPACE                                             04180000
  TBN  $MDTFT(,XR2),$MTPLT      IS IT STATION CONTROL      CM 04190000
  TBF  $MDTFT(,XR2),$MTLRC      IS IT NON-CHECKING          CM 04200000
  JT   CMREST                   JUMP IF STAT CTL AND NON-CHKG CM 04210000
  SPACE                                             04220000
.S5900 ANOP                                       04230000
* IF CHECKING TERMINAL, DON'T RESET. (RESET ALLOWS OPERATOR INPUT). M 04240000
  SPACE                                             04250000
  TBN  $MDTFT(,XR2),$MTLRC      IS IT CHECKING TERMINAL ?   M 04260000
  JT   CMINIT                   JUMP IF CHECKING TERMINAL.   M 04270000
CMREST EQU  *                      * LOCAL.                 04280000
  SBN  CMOPWK+1,MLRSET          SET RESET BIT IN OP CODE ON  M 04290000
  EJECT                                                                    04300000
*****                                                                    04310000
* DETERMINE IF INITIAL OPERATION IS NEEDED. * 04320000
*****                                                                    04330000
  SPACE                                             04340000
CMINIT EQU  *                      TEST FOR INITIAL OP      M 04350000
  TBF  LCBATR(,XR2),LCBNIT      TEST INITIAL OP BIT IN LCB  M 04360000
  JF   CMTRST                   JUMP IF INITIAL NOT NEEDED  M 04370000
  SPACE                                             04380000
*****M 04390000
* NEED INITIAL OPERATION ON THIS LINE M 04400000
*****M 04410000
  SPACE                                             04420000
  AIF  (&NSW).A5500                               04430000
* IF IT IS A SWITCHED LINE AND IT IS A USER OP SM 04440000
* THEN ISSUE MESSAGE TO SYSTEM OPERATOR INFORMING HIM OF THE STATUS SM 04450000
* OF THE LINE SM 04460000
  SPACE                                             04470000
  TBN  $MDTFT(,XR2),$MMDNT      IS IT SWITCHED LINE      SM 04480000
  TBF  PLOPM(,XR1),OP$SYS       IS IT USER REQUEST      SM 04490000
  JF   CMNMSO                   JUMP IF NOT USER OP- SWC LINE SM 04500000
  SPACE                                             04510000
* HAVE USER REQUEST TO SWITCHED LINE WHICH REQUIRES CONNECTION SM 04520000
* BRING IN TRANSIENT TO ISSUE MESSAGE SM 04530000
  SPACE                                             04540000
  SVC  0                      ##### TRANSIENT CALL ##### SM 04550000
  DC   AL1(CCPRIB)             CCP SVC RIB. SM 04560000
  DC   AL1(CC4WC)              TRANSIENT ID SM 04570000
  SPACE                                             04580000
CMNMSO EQU  *                      * LOCAL. SM 04590000
  SPACE                                             04600000
.A5500 ANOP                                       04610000
  AIF  (&N2741).B5200                               04620000
* DETERMINE IF THIS IS PUT TO 2741 4M 04630000
* IF YES - CANNOT ISSUE WRITE INITIAL TO 2741 SO MUST ISSUE READ INIT4M 04640000
* AND BIT BUCKET THE DATA FROM THE READ BEFORE ISSUEING THE WRITE 4M 04650000
* TO THE 2741 4M 04660000
  SPACE                                             04670000
  TBN  PL$OPM(,XR1),OPPUT      IS IT PUT 4M 04680000

```

```

CLI    $MDTTP(,XR2),ML2741      IS IT 2741          4M 04690000
JC    CMOPFM,ANY+LO+HI+FALSE    JUMP IF NOT INITL PUT TO 2741 4M 04700000
SPACE                                     04710000
* HAVE WRITE INITIAL TO 2741 SO ISSUE READ INITIAL          4M 04720000
SPACE                                     04730000
SBN    LCBATR(,XR2),LCBTBK      SET ON LCB BIT TO BIT          4M 04740000
*                                     BUCKET DATA FROM DUMMY READ 4M 04750000
MVI    CMOPWK+1,MLREAD          PLUG READ INITL INTO OP WORK 4M 04760000
J      CMOPWK                    JUMP TO PLUG OP INTO DTF          4M 04770000
AGO    .B5250                    04780000
.B5200 ANOP                      04790000
J      CMOPFM                    JUMP TO PLUG INITL OP INTO DTF M 04800000
.B5250 ANOP                      04810000
EJECT                                     04820000
*****                                04830000
*   DETERMINE WHETHER TO ISSUE CONVERSATION OR CONTINUE OP CODE. * 04840000
*****                                04850000
SPACE                                     04860000
CMTRST EQU    *                  *   LOCAL.                      M 04870000
TBN    CMOPWK+1,MLRSET          HAS RESET BEEN SET          M 04880000
JT     CMCONV                   JUMP TO ISSUE CONVERSATIONAL M 04890000
SPACE                                     04900000
* FOR CONTINUE TO BE USED THE FOLLOWING CONDITIONS MUST BE MET M 04910000
* 1/ LAST OP ON LINE WAS OF SAME TYPE                          M 04920000
* 2/ LAST TERMINAL SCHEDULED ON LINE WAS SAME                 M 04930000
* 3/ READ TO STATION CONTROL TERMINAL WILL NEVER BE CONTINUE M 04940000
* 4/ READ CONTINUE VALID ONLY FOR A CHECKING TERM. OR A 2741. M 04950000
SPACE                                     04960000
CMPOP  TBN    $MDOPC(,XR2),#     DO THE OP CODES MATCH ?      M 04970000
JF     CMCONV                   JUMP TO ISSUE CONVERSATIONAL M 04980000
SPACE                                     04990000
AIF    (&NSCTL).S5910           05000000
CLC    $MDTMA(2,XR2),CMSTMA     OPS TO THE SAME TERMINAL ?    CM 05010000
JNE    CMCONV                   IF NO-JUMP TO ISSUE CONVERS. CM 05020000
SPACE                                     05030000
.S5910 ANOP                      05040000
* IF OPERATION IS A WRITE ISSUE A CONTINUE OP                  M 05050000
SPACE                                     05060000
TBN    $MDOPC(,XR2),MLWRIT      WAS IT WRITE                      M 05070000
JT     CMCONT                   IF YES JUMP TO FORM CONTINUE M 05080000
SPACE                                     05090000
AIF    (&NSCTL).S5920           05100000
* IF STATION CONTROL; THEN ISSUE CONVERSATIONAL OPERATION.   CM 05110000
SPACE                                     05120000
TBN    $MDTFT(,XR2),$MTPLT      STATION CONTROL ?              CM 05130000
JT     CMCONV                   YES-GO ISSUE CONVERSATIONAL. CM 05140000
.S5920 SPACE                      05150000
* CONTINUE OP VALID TO ALL CHECKING TERMINALS.                05160000
SPACE                                     05170000
TBN    $MDTFT(,XR2),$MTLRC      CHECKING TERMINAL ?            M 05180000
JT     CMCONT                   YES-CONTINUE OP IS OK.           M 05190000
SPACE                                     05200000
AIF    (&N2741).B5300           05210000
* SINCE OPERATION IS A READ TO NON-STATION CONTROL TERMINAL 4M 05220000
* THE TERMINAL MUST BE A 2741 FOR CONTINUE TO BE VALID       4M 05230000
SPACE                                     05240000
CLI    $MDTTP(,XR2),ML2741      IS IT 2741 ?          4M 05250000
JE     CMCONT                   JUMP IF 2741                    4M 05260000
SPACE                                     05270000
.B5300 ANOP                      05280000

```



```

MODULE-$E090 , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$E090                                00020000
GBLB  &NDF, &NOM, &NRUF, &NPBY      00030000
TEXT                                  00040000
* R-05, C-00 CHANGE LEVEL            00040100
TITLE '$E090 - PROCESS SATISFIED ACCEPT INPUT' 00050000
*****                                00060000
*                                     * 00070000
* TITLE: 'CMACI'                       * 00080000
*                                     * 00090000
* FUNCTION: SATISFY ACCEPT OPERATION AFTER AN INVITE OP ENDS. * 00100000
*                                     * 00110000
* OPERATION:                            * 00120000
* . SET FIELDS IN ACCEPT PARAMETER LIST TO REFLECT DATA RECEIVED. * 00130000
* . IF DFF TERMINAL, QUEUE ACCEPT REQUEST FOR $CC4DF AND EXIT. * 00140000
* . MOVE DATA TO THE ACCEPT RECORD AREA. * 00150000
* . FREEMAIN INVITE GETMAIN AREAS. * 00160000
* . EXIT TO CMPAII TO POST ACCEPT COMPLETE. * 00170000
*                                     * 00180000
* INPUT: XR1 -> ACCEPT PARAMETER LIST ADDRESS. * 00190000
*                                     * 00190100
* OUTPUT: OUTPUT WILL BE THE DATA IN THE USER RECORD AREA. * 00200000
*                                     * 00200100
* EXTERNAL ROUTINES USED: * 00200200
* CMMVRT - CM MOVE ROUTINE INTERFACE. * 00200300
* $CC4FR - CM FREEMAIN AREAS ASSOCIATED WITH PL. * 00200400
*                                     * 00200500
* EXITS-NORMAL: TO CMPAII TO POST ACCEPT SATISFIED. * 00200600
*                                     * 00210000
*****                                00220000
SPACE 2                                00230000
CMACI EQU * ACCEPT INPUT SATISFIED      00240000
SPACE 1                                00250000
*****                                00260000
* SET UP USER RECORD AREA * 00270000
*****                                00280000
SPACE                                  00290000
L PLTUBA(, PL), XR2 LOAD POINTER TO THE TUB 00300000
SPACE                                  00310000
* BEGIN TO PLUG USER'S ACCEPT INPUT PARM LIST 00320000
SPACE                                  00330000
TBN TUBAT2(, XR2), TUBIMI IS THIS A PROGRAM REQUEST? 00340000
JF CMNOPR BRANCH IF NOT 00350000
SLC PLEFFL(2, PL), PL$RTC(, PL) DECR DATA LENGTH PROGRAM NAME 00360000
* * LENGTH IN ACCEPT PL. * 00370000
AIF (&NRUF).T0200 00370600
TBN TUBCHR(, XR2), TUBLNE BSCA LINE, AND WB 00371800
TBF TUBSCS(, XR2), TUBRUF * NOT RUF, AND WB 00372400
L TUBTCB(, XR2), XR2 XR2-> TCB OWNING THIS TUB. WB 00373000
TBN TCBDMG(, XR2), TCBRUF * PROGRAM SUPPORT RUF ? WB 00373600
L PLTUBA(, PL), XR2 XR2-> TUB. WB 00374200
JF CMNOPR NO-GO HANDLE AS NORMAL. WB 00374800
L TUBPL@(, XR2), XR2 XR2-> INVITE PARM LIST. WB 00375400
MVI PLRTC-1(, PL), RCOK RETURN CODE = 0002; NON-RUF WB 00376000
MVI PLRTC(, PL), RCXEOT * FOR A RUF PROGRAM. WB 00376600
J CMBGPG GO MOVE THE DATA. WB 00377200
.T0200 ANOP 00378400
CMNOPR EQU * * LOCAL 00380000
L TUBPL@(, XR2), XR2 POINT TO 'INVITE' PARM LIST 00390000

```

	MVC	PLRTC(2,PL),PLRTC(,XR2)	MOVE RETURN CODE FROM INV PARM		00400000
*		XR1 POINTS TO 'ACCEPT' PARM LIST AND XR2 POINTS TO 'INVITE'.			00410000
		SPACE			00420000
*		DETERMINE IF OPERATION WAS SUCCESSFUL OR NOT			00430000
		SPACE			00440000
	CLI	PLRTC(,XR2),RCXEDT	CHECK AGAINST HIGHEST RETURN FOR		00450000
*			WHICH DATA ACCOMPANIES THE PL.		00460000
	JNH	CMBGPG	DATA WITH PARAMETER LIST.		00470000
		SPACE 1			00480000
*		-----*			00490000
*		NO DATA WITH INVITE PARAMETER LIST		*	00500000
*		-----*			00510000
		SPACE 1			00520000
*		ZERO INPUT LENGTH - \$CC4MX WILL BLANK USERS REC AREA IF PLEFFL=0.			00590000
		SPACE			00600000
	SLC	PLEFFL(2,XR2),PLEFFL(,XR2)	ZERO EFFECTIVE INPUT LENGTH		00620000
	AIF	(&NDF).D0100			00630000
	MVI	PL\$OPC(,XR1),BRNOP	NOOP THE BRANCH TO DFF	FB	00640000
.D0100	ANOP				00650000
		SPACE			00660000
*		-----*			00670000
*		DATA WITH INVITE PARAMETER LIST		*	00680000
*		-----*			00690000
		SPACE 1			00700000
CMBGPG	EQU	*	BEGIN TO PLUG USER PARM LIST		00710000
	MVC	#CMMVL+MVLTOA,PLINL(2,PL)	MOVE TARGET LENGTH TO MOVE LIST		00720000
	MVC	PLEFFL(2,PL),PLEFFL(,XR2)	MOVE INVITE EFFL LENGTH TO ACCEPT		00730000
	L	PLTUBA(,XR1),XR1	POINT TO THE TUB		00740000
	L	TUBTCB(,XR1),XR1	POINT TO THE USER'S TCB		00750000
	L	TCBWK(,XR1),XR1	POINT TO USER'S RECORD AREA		00760000
	LA	6(,XR1),XR1	BUMP XR1 TO 1ST BYTE OF DATA		00770000
	ST	#CMMVL+MVLTOA,XR1	STORE ADDRESS IN MOVE LIST		00780000
		SPACE			00790000
	AIF	(&NDF).D0130			00800000
	L	CMSPL,PL	RESTORE POINTER TO PARM LIST	FB	00810000
	L	PLTUBA(,PL),XR2	POINT TO THE TUB	FB	00820000
	TBN	TUBTA1(,XR2),TASDF	IS THIS A DFF TERMINAL?	FB	00830000
	L	TUBPL@(,XR2),XR2	RESTORE INVITE PARM LIST PTR	FB	00840000
	JF	CMNMAP	BRANCH IF NO DFF	FB	00850000
		SPACE			00860000
CMBRCH	MVC	CMBRCH+1,PL\$OPC(1,XR1)	SET CONDITION CODE IN BRANCH	FB	00870000
	B	CMDFEQ	Q QUEUE UP REQUEST FOR DFF	FB	00880000
		SPACE			00890000
.D0130	ANOP				00900000
	L	CMSPL,PL	RESTORE ACCEPT PL ADDRESS		00910000
	L	PLTUBA(,PL),XR2	POINT TO THE TUB		00920000
	AIF	(&NDF).D0140			00940000
	CLI	PL\$OPC(,PL),BR97	WAS DFF CALLED?	FB	00950000
	AIF	(&NRUF).T0200			00950500
	JNE	CMNLOD	NO-GO LOAD REG, CHECK EFFL.	FWB	00951000
	TBN	TUBAT2(,XR2),TUBIMI	DATA WITH PROGRAM REQUEST ?	FWB	00951500
	JF	CMBPAI	NO-GO POST RESULTS.	FWB	00952000
	L	TUBPL@(,XR2),XR2	XR2-> INVITE PARM LIST.	FWB	00952500
	L	PLRECA(,XR2),XR1	XR1-> DATA RECORD.	FWB	00953000
	A	X\$FFFF,XR1	XR1-> LAST BYTE GETMAIN PL.	FWB	00953500
CMMOVE	MVC	CMMOVE+2,PL\$RTC(1,XR2)	SET UP LEN. FOR GM PL MOVE.	FWB	00954000
	EQU	*	BUMP GETMAIN PARM LIST UP	FWB	00954500
	MVC	#(,XR1),0(4,XR1)	* TO DFF DATA AREA.	FWB	00955000
	ALC	PLRECA(,XR2),PL\$RTC(2,XR2)	BUMP RECORD @ PAST PRMNAME.	FWB	00955500
CMBPAI	EQU	*	*	FWB	00956000

```

      B      CMPAII      GO POST RESULTS OF OPERATION.FWB 00956500
      SPACE      00957000
CMNLOD EQU      *      *      FWB 00957500
      L      PLTUBA(,PL),XR2      XR2-> TUB.      FWB 00958000
      AGO      .T0400      00958500
.T0200 BE      CMPAII      YES-GO POST RESULTS TO USER. FB 00959000
.T0400 ANOP      00959500
.D0140 ANOP      00980000
      L      TUBPL@(,XR2),XR2      XR2-> INVITE PARM LIST.      00982000
CMNMAP EQU      *      *      00984000
      CLC      PLINL(2,PL),PLEFFL(,XR2)      DOES INVITE HAVE MORE DATA      00990000
*      *      THAN ACCEPT WANTS ?      01000000
      JNL      CMGTRA      NO - CAN RETURN ALL INVITE DATA.      01010000
      SPACE      01020000
* HAVE ACCEPT INPUT WHOSE INPUT LENGTH IS LESS THAN THE EFFECTIVE      01030000
* LENGTH OF THE INVITE, THEREFORE SET ACCEPT EFFECTIVE LENGTH TO ITS      01040000
* REQUESTED LENGTH, AND TRUNCATE DATA.      01050000
      SPACE      01060000
      SBN      PLRTC(,PL),RCXDTR      SET DATA TRUNCATED BIT IN RT COD      01070000
      MVC      PLEFFL(2,PL),PLINL(,PL)      SET ACCEPT EFFL TO REQ'D LENGTH      01080000
      SPACE      01090000
CMGTRA EQU      *      * LOCAL      01100000
      SPACE 1      01110000
*****      01120000
*      MOVE DATA FROM INVITE HOLD BUFFER TO USER RECORD AREA      01130000
*****      01140000
      SPACE      01150000
      MVC      #CMMVL+MVLFRA(2),PLRECA(,XR2) MOVE IN SOURCE ADDR FOR MOVE      01160000
      SPACE      01170000
      L      PLTUBA(,XR2),XR1      POINT TO THE TUB      01180000
      TBN      TUBAT2(,XR1),TUBIMI      IS PROGRAM NAME SITTING IN SOURC      01190000
*      FILED AREA      01200000
      JF      CMDOIT      JUMP IF NO PROGRAM NAME IN AREA      01210000
      SPACE      01220000
* BUMP THE SOURCE RECORD AREA ADDRESS PAST THE PROGRAM NAME      01230000
* THE AMOUNT TO BUMP THE ADDRESS BY RESIDES IN PL$RTC OF THE INVITE      01240000
* PARAMETER LIST.      01250000
      SPACE      01260000
      ALC      #CMMVL+MVLFRA(2),PL$RTC(,XR2) BUMP ADDR PAST PROGRAM NAME      01270000
      SPACE      01280000
CMDOIT EQU      *      DO THE MOVE NOW      01290000
*      ----START----- @26      01291000
      CLI      PLRTC(,XR2),RCXEDT      RETURN CODE > THREE AND      01292000
      TBN      TUBTA1(,XR1),TASDFE      THIS A DFF TERMINAL ?      01293000
      JC      CMNOBK,TRUahi      YES-DON'T BLANK USER'S REC AREA      01294000
*      ----END----- @26      01295000
      MVI      #CMMVL+MVLTYE,SWAPTO      SET TRANSLATE ON FOR MOVE      01300000
      LA      0(,XR2),XR1      POINT TO INVITE PARAMETER LIST      01310000
      SPACE 1      01320000
* MOVE DATA - FROM LENGTH IS SET FROM PLEFFL OF INVITE PARAMETER LIST      01330000
* IF ZERO (NO DATA), THE ACCEPT RECORD AREA WILL BE      01340000
* BLANKED.      01350000
      SPACE 1      01360000
      B      CMMVRT      BR TO MOVE AND CLEAR ROUTINE      01370000
      SPACE 2      01380000
*****      01390000
*      FREEMAIN THE INVITE INPUT HOLD BUFFER      01400000
*****      01410000
      SPACE      01420000
CMNOBK EQU      *      * LOCAL      01425000

```


L	CMSPL, XR1	RESTORE PARAMETER LIST POINTER	01430000
L	PLTUBA(, XR1), XR2	RESTORE TUB POINTER	01440000
SBF	TUBAT2(, XR2), TUBIMI	SET IMPLICIT INVITE BIT OFF	01450000
TBN	TUBAT1(, XR2), TUBKNM	IS CONSOLE PGM REQ DATA ?	01460000
JF	CMNOKN	NO - BYPASS RESET OF CONSOLE	01470000
SBF	\$AMFLG, \$AMPF9	YES- RESET PF9 FOR CONSOLE	01480000
CMNOKN EQU	*	* LOCAL	01490000
L	TUBPL@(, XR2), PL	POINT TO INVITE PARM LIST	01500000
SPACE	1		01510000
B	\$CC4FR	FREE INVITE GETMAINED AREAS	01520000
SPACE	1		01530000
SBN	CMSWIT, CMTPRQ	SET BIT SO ACCEPT PL WILL BE	01540000
*		* POSTED	01550000
	TITLE '\$E090/CMPAII---DETERMINE-IF-POST-NEEDED'		01560000
*****			01570000
*			* 01580000
*	NAME--CMPAII		* 01590000
*			* 01600000
*	TITLE--DETERMINE IF POST IS NEEDED		* 01610000
*			* 01620000
*	FUNCTION--POST CALLER THAT THE IO REQUEST NEEDED TO ACCOMPLISH		* 01630000
*	HIS REQUEST HAS BEEN SCHEDULED AND THEN CHECK FOR		* 01640000
*	DTF TO BE RESCHEDULED AFTER OLT FAILED.		* 01650000
*			* 01660000
*	ENTRY POINTS- CMPAII -POST REQUESTOR IF TP SCHEDULED		* 01670000
*	CMRQBF -CHECK FOR MORE WORK TO BE PERFORMED		* 01680000
*			* 01680100
*	EXITS-NORMAL:		* 01680200
*	TO CMMTBY - IF MLTA DTF TO START AFTER OLT.		* 01680300
*	TO CMOPND - CHECK FOR MORE WORK.		* 01680400
*			* 01680500
*****			01690000
	SPACE		01700000
CMPAII EQU	*	DETERMINE IS POST NEEDED	01710000
TBN	CMSWIT, CMTPRQ	WAS TP REQUEST SCHEDULED	01720000
JF	CMTSRQ	JUMP IF NOT ON	01730000
SPACE			01740000
L	CMNWPL, PL	ORIGINAL PL TO BE POSTED	01750000
*			01750200
*	BUSY PRINTER SUPPORT		01750400
*			01750600
	AIF (&NPBY).NBY06	BUSY PRINTER SUPPORTED	01750800
L	PLTUBA(, PL), XR2	XR2-----> TUB	01751000
TBN	TUBAT4(, XR2), TUBWAT	THIS A BUSY PRINTER OPERATION?	01751200
JT	CMRQBF	YES - SKIP POST AT THIS TIME	01751400
.NBY06 ANOP			01751600
*		-----@16	01752000
TBN	PLOPM(, PL), OPDISC	DISCONNECT OPERATION ?	01754000
JT	CMDISC	YES - POST	01756000
*		-----@16	01758000
TBF	PL\$OPM(, PL), OPNOW	IS IT WAIT OP	01760000
JT	CMRQBF	IF WAIT OP, SKIP POST	01770000
SPACE			01780000
*****			01790000
*	PUT NO WAIT, INVITE OR PUT DISCONNECT - POST SCHEDULED		* 01800000
*****			01810000
CMDISC EQU	*	* LOCAL	01815000
SPACE			01820000
LA	PLECB(, PL), XR1	ADDRESS OF ECB FOR REQUEST	01830000
SVC	0	POST \$CC4II OR \$CC4IS	01840000

DC	AL1(POSTRB)	* THAT NO WAIT OP SCHEDULED	01850000
SPACE 2			01860000
*****			01870000
*	DETERMINE IF OLT FAILED, IF SO RESCHEDULE LINE.	*	01880000
*	(ENTER HERE WHEN NO POST NEEDED - AFTER STOP INVITE)	*	01890000
*****			01910000
SPACE			01920000
CMRQBF EQU *		*	01930000
SBF CMSWIT,CMTPRQ		TURN OFF TP REQUEST SWITCH	01940000
CMTSRQ EQU *		* LOCAL	01950000
AIF (&NOM).M0580			01960000
TBN CMSWIT,CMNOST		DID ONLINE TEST START FAIL M	01970000
SBF CMSWIT,CMNOST		SET OFF ONLINE TEST NOT STARTEDM	01980000
BT CMMTBY		BRANCH IF OLT START FAILED M	01990000
.M0580 ANOP			02000000
B CMOPND		CHECK FOR MORE WORK	02010000
MEND			02020000

```

MODULE-$E093 , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO 00010000
***** 00020000
.* NAME $E093 * 00030000
***** 00040000
$E093 00050000
GBLB &ONE, &NOB, &N37, &MIN, &NDME, &NMSG, &N41 00070000
GBLB &NPP, &NMP, &NSWL, &NCS, &NITB, &NTSP, &N32, &NDF, &NCPU, &NRUF 00080000
GBLA &BSC, &MLA 00110000
LCLA &MIX 00130000
TEXT 00140000
* R-06, C-00 CHANGE LEVEL 00150000
AIF (&NOB).C0580 00160000
&MIX SETA &NSWL+&NCS 00280000
AIF (&MIX EQ '2').S0700 00290000
TITLE '$E093/CMBSKP---BSCA-POLL-SKIP-BIT-ROUTINE' 00300000
***** C/SB 00310000
* * C/SB 00320000
* NAME--CMBSKP * C/SB 00330000
* * C/SB 00340000
* TITLE--BSCA POLL SKIP BIT ROUTINE * C/SB 00350000
* * C/SB 00360000
* FUNCTION--SET THE SKIP ENTRY INDICATOR ON/OFF IN THE BSCA * C/SB 00370000
* POLLING LIST OR THE BSCA SWITCHED LINE ID * C/SB 00380000
* VERIFICATION LIST. * C/SB 00390000
* * C/SB 00400000
* OPERATION-- * C/SB 00410000
* . IF THIS DTF OR THIS PARTICULAR OPERATION DOES * C/SB 00420000
* NOT INVOLVE ONE OF THE BSCA LIST THEN RETURN * C/SB 00430000
* WITHOUT SETTING SKIP BIT. * C/SB 00440000
* . OTHERWISE, FIND THE ENTRY IN THE APPROP LIST. * C/SB 00450000
* . SET THE SKIP BIT AS SPECIFIED FOR ALL ENTRIES * C/SB 00460000
* FOR THE SAME TERMINAL. * C/SB 00470000
* * C/SB 00480000
* ENTRY POINTS--CMBSKP-ALL FUNCTIONS ABOVE. * C/SB 00490000
* * C/SB 00530000
* INPUT-- * C/SB 00540000
* XR1--ADDRESS OF THE TP PARAMETER LIST. * C/SB 00550000
* XR2--SAVED AD RESTORED. * C/SB 00560000
* CMB#SB--SBN2(X'7A') SET THE SKIP BIT ON, OR * C/SB 00570000
* SBF2(X'7B') SET THE SKIP BIT OFF. * C/SB 00580000
* TUBSID--ID OF THE ENTRY(S) TO PERFORM THE OPERATION * C/SB 00590000
* FOR. * C/SB 00600000
* * C/SB 00610000
* OUTPUT-- * C/SB 00620000
* XR1-ADDRESS OF THE TP PARAMETER LIST. * C/SB 00630000
* XR2-RESTORED TO ENTRY VALUE. * C/SB 00640000
* POLLING/SWITCHED ID LIST-SKIP BIT FOR APPROPRIATE * C/SB 00650000
* ID SET AS SPECIFIED. * C/SB 00660000
* * C/SB 00670000
* * C/SB 00710000
* EXIT, NORMAL--TO NSI OF CALLER. * C/SB 00720000
***** C/SB 00730000
SPACE 00730000
CMBSKP EQU * * C/SB 00740000
SPACE 00750000
ST CMBXIT+3,ARR SAVE THE ARR C/SB 00760000
ST CMON1B+3,PL SAVE XR1 C/SB 00770000
ST CMBXR2+3,XR2 SAVE XR2 C/SB 00780000
L PLTUBA(,PL),TUB POINT XR1 AT TUB C/SB 00790000
L TUBDTF(,TUB),DTF POINT XR2 AT DTF C/SB 00800000

```

&MIX	SETA	&NPP+&NMP+&NSWL			00810000
	AIF	(&MIX EQ '3').S0600			00820000
	TBN	\$BDATR(,DTF),\$BCSWI	MULTI-TERMINAL LINE ?	C/SLB	00830000
	JF	CMOEXT	NO-GO EXIT.	C/SLB	00840000
	AIF	(&NSWL).S0600			00850000
	TBF	\$BDATR(,DTF),\$BCMPT	SWITCHED	SB	00860000
	TBN	TUBTA1(,TUB),TASCNC	* CALL ?	SB	00870000
	JT	CMOEXT	YES-DON'T SET SKIP BITS.	SB	00880000
.S0600	ANOP				00890000
	MVC	LCBID#(1,DTF),TUBSID(,TUB)	MOVE POLL LIST ID TO SAVE.	C/SB	00900000
	AIF	(&NSWL).S0640			00910000
*			----- START @18		00912000
	TBN	\$BDATR(,DTF),\$BCMCM	CONTROL STATION LINE ?	SB	00914000
	JT	CMONSD	YES SKIP SWITCHED CHECKING	SB	00916000
	TBF	\$BDATR(,DTF),\$BCMPT	SWITCHED,	SB	00920000
	TBN	TUBTA2(,TUB),TASVfy	* NO ID VERIFY ?	SB	00930000
	JF	CMONSD	YES - GO PROCESS LIST.	SB	00933000
	TBF	\$BDATR(,DTF),\$BCMPT	SWITCHED AND	SB	00936000
	CLI	CMB#SB,SBF1	* ACTIVATE ENTRY ?	SB	00940000
	JC	CMONO#,TRUAEQ	YES-GO BUILD SWITCHED ENTRY.	SB	00950000
	L	LCBNO#(,DTF),POL	POINT TO SWITCHED LIST.	SB	00951000
	CLC	LCBID#(1,DTF),POLID(,POL)	THIS ENTRY IN LIST NOW ?	SB	00952000
	JNE	CMOEXT	NO - SKIP THIS ENTRY.	SB	00953000
CMONSD	EQU	*	* LOCAL	SB	00954000
*			----- END @18		00955000
.S0640	ANOP				00960000
	L	LCBPOL(,DTF),POL	LOAD PTR TO POLLING LIST	C/SB	00970000
CMOCHK	EQU	*	*	C/SB	00980000
	CLI	POLID(,POL),POLEND	END OF LIST ?	C/SB	00990000
	JNL	CMOEXT	YES-GO EXIT.	C/SB	01000000
	MVC	CMOSTS+2,POLCNT(1,POL)	SET UP LA TO STATUS BYTE	C/SB	01010000
	CLC	POLID(1,POL),LCBID#(,DTF)	THIS ENTRY SAME AS REQ D?	C/SB	01020000
CMOSTS	LA	#(,POL),POL	UP PTR TO STATUS BYTE	C/SB	01030000
	JNE	CMONXT	NO-IGNORE IT, GOTO NXT ENT.	C/SB	01040000
CMB#SB	EQU	*	*	C/SB	01050000
	SBN	POLCNT+1(,POL),POLSKP	SET POLL SKIP BIT ON / OFF.	C/SB	01060000
*--	CC4BG		MUST KNOW IF POLL HAS ACTUALLY BEEN DONE-----@39		0106010
	TBN	POLCNT+1(,POL),POLSKP	SETTING SKIP BIT ON?		0106100
	JT	CMONXT	YES- DO NOT SET POLL PENDG		0106200
	SBN	POLCNT+1(,POL),X'01'	INDICATE POLL PENDING		0106300
*--	IF		SKIP BIT IS 01 THEN POLL PENDING- TURNED OFF IN CC4M1(2)--- @39		0106400
CMONXT	EQU	*	*	C/SB	01070000
	LA	POLNXT(,POL),POL	UP PTR TO NEXT POLL LIST ENTC/SB		01080000
	B	CMOCHK	GO CHECK THIS NEXT ENTRY.	C/SB	01090000
	SPACE				01100000
	AIF	(&NSWL).S0660			01110000
CMONO#	EQU	*	*	SB	01120000
	L	LCBNO#(,DTF),POL	XR1-> NO VERIFY ID ENTRY.	SB	01130000
	TBN	POLCNT+1(,POL),POLSKP	SKIP IND ALREADY OFF ?	SB	01140000
	JF	CMOEXT	YES-NAME OF NON-VERIFY AS SET.	SB	01150000
	MVC	POLID(,POL),LCBID#(1,DTF)	SET ID/NAME OF NON-VERIFY.	SB	01160000
	SBF	POLCNT+1(,POL),POLSKP	SET OFF THE SKIP BIT.	SB	01170000
.S0660	ANOP				01180000
CMOEXT	EQU	*	*	C/SB	01190000
CMON1B	EQU	*	*	C/SB	01200000
	LA	*-* ,PL	RESTORE PARM LIST REG.	C/SB	01210000
CMBXR2	EQU	*	*	C/SB	01220000
	LA	#,XR2	RELOAD CALLER'S XR2.	C/SB	01230000
CMBXIT	EQU	*	*	C/SB	01240000
	B	#	RETURN TO CALLER.	C/SB	01250000

```

EJECT 01260000
***** C/SB 01270000
* C/SB 01280000
* ENTRY POINT- CMASCH - ADDRESS LIST SEARCH * C/SB 01290000
* CMPSCH - POLL LIST SEARCH * C/SB 01300000
* * C/SB 01310000
* FUNCTION : GIVEN ID ENTRY BYTE IN DTF SAVE AREA, FIND THE * C/SB 01320000
* CORRESPONDING ENTRY IN THE POLL LIST AND RETURN A * C/SB 01330000
* POINTER TO THE PHYSICAL TERMINAL ADDRESS. * C/SB 01340000
* * C/SB 01350000
* INPUT: XR1-> DON'T CARE, XR2-> DTF. * C/SB 01350001
* * C/SB 01350002
* OUTPUT: XR1-> POLL LIST ENTRY, XR2-> DTF. * C/SB 01350003
* * C/SB 01350004
* EXITS-NORMAL: TO NSI OF CALLER. * C/SB 01350005
* -ERROR: CMASCH/CMPSCH WILL RETURN TO NSI+3 IF THE * C/SB 01350006
* ENTRY IS NOT FOUND. * C/SB 01350007
* * C/SB 01350008
***** C/SB 01380000
SPACE 01390000
CMASCH EQU * ADDRESS LIST ENTRY POINT. C/SB 01400000
L LCBSEL(,DTF),POL LOAD ADDR LIST REGISTER. C/SB 01410000
J CMPNXT GO ENTER MAINLINE CODE. C/SB 01420000
SPACE 01430000
CMPSCH EQU * FIRST LEVEL SUBROUTINE. C/SB 01440000
L LCBPOL(,DTF),POL LOAD POLL LIST REGISTER. C/SB 01450000
CMPNXT EQU * * C/SB 01460000
ST LCBWRK(,DTF),ARR SAVE RETURN NSI. C/SB 01470000
CMPCHK EQU * * C/SB 01480000
CLC POLID(1,POL),LCBID#(,DTF) THIS WANTED TERMINAL ENTRY? C/SB 01490000
JE CMPFND YES-GO SET UP RETURN REG. C/SB 01500000
MVC CMPLA+2,POLCNT(1,POL) USE COUNT TO GET NEXT ENTRY C/SB 01510000
CMPLA EQU * * C/SB 01520000
LA #(,POL),POL UP REG PAST PHYSICAL ADDR C/SB 01530000
LA POLNXT(,POL),POL UP REG PAST ID/COUNT/STATUS C/SB 01540000
AIF (&NSWL).S0680 01550000
CLI POLID(,POL),POLEND END OF LIST ? SB 01560000
BL CMPCHK NO-CONTINUE THE SEARCH. SB 01570000
ALC LCBWRK(2,DTF),THREE ADD 3 TO NSI ADDR FOR RETURN. SB 01580000
* RETURN TO NSI +3 ONLY FOR SWITCHED ID LIST. EITHER BOTH USER , SB 01590000
* AND CMASCH ARE IN CORE, OR BOTH ARE IN THE SAME TRANSIENT. SB 01600000
SPACE 1 01610000
AGO .S0690 01620000
.S0680 ANOP 01630000
B CMPCHK GO TO CHECK NEXT ENTRY. CB 01640000
.S0690 ANOP 01650000
SPACE 01660000
CMPFND EQU * * C/SB 01670000
L LCBWRK(,DTF),IAR RETURN TO CALLER. C/SB 01680000
.S0700 ANOP 01690000
&MIX SETA (&BSC+&MLA) 01700000
AIF (&MIX LE '1').Y0050 01710000
TITLE '$E093/CMCSKP--SET-BSCA-CHECK-LIST-BIT-ON/OFF' 01720000
***** YB 01730000
* * YB 01740000
* ENTRY POINT- CMCSKP - SET CHECK LIST SKIP BIT * YB 01750000
* * YB 01760000
* FUNCTION : * YB 01770000
* SET SKIP BIT ON OR OFF IN CHECK LIST FOR BSCA DTF'S. * YB 01780000
* TO SET THE SKIP BIT ON, MOVE SBN2 INTO CMC#SB. * YB 01790000

```

```

*      TO SET THE SKIP BIT OFF, MOVE SBF2 INTO CMC#SB.          * YB 01800000
*                                                                * YB 01810000
*      ON ENTRY: XR1 -> IOB                                     * YB 01820000
*      ON EXIT: XR1=IOB, XR2=DTF.                               * YB 01830000
***** YB 01840000
      SPACE                                                    01850000
CMCSKP EQU * * YB 01860000
      ST CMCSXT+3,ARR SAVE RETURN @. YB 01870000
      L @CKLST,XR2 LOAD CHECK LIST @ IN XR2. YB 01880000
CMCSCK EQU * * YB 01890000
      CLC CKLDTF(2,XR2),IOBDTF(,IOB) THIS LIST ENTRY FOR THIS DTF? YB 01900000
      JE CMC#SB YES-GO SET SKIP BIT. YB 01910000
      LA CKLEN(,XR2),XR2 LOAD @ OF NEXT ENTRY. YB 01920000
      B CMCSCK GO AND CHECK THIS NEXT ENTRY. YB 01930000
      SPACE                                                    01940000
CMC#SB EQU * * YB 01950000
      SBN CKLSTS(,XR2),CKLSKP SET SKIP AS SET BY CALLER. YB 01960000
      L CKLDTF(,XR2),DTF RESTORE DTF REGISTER. YB 01970000
CMCSXT EQU * * YB 01980000
      B # RETURN TO CALLER. YB 01990000
.Y0050 ANOP                                                    02000000
      AIF (&MIN).N0400                                         02010000
      TITLE '$E093/CMBTAS--SET-DTF-AND-IOB-FROM-TAS'          02020000
***** RB 02030000
*                                                                * RB 02040000
* NAME--CMBTAS                                               * RB 02050000
*                                                                * RB 02060000
* TITLE--BSCA SET DTF AND IOB FROM TAS                       * RB 02070000
*                                                                * RB 02080000
* FUNCTION--SET DTF AND IOB ATTRIBUTES AND OWNERSHIP STATUS. * RB 02090000
*                                                                * RB 02100000
* ENTRY REGS: XR2=DTF                                         * RB 02110000
* EXIT REGS: XR1=TUB, XR2=DTF.                               * RB 02120000
*                                                                * RB 02130000
***** RB 02140000
      SPACE                                                    02150000
CMBTAS EQU * * RB 02160000
      ST CMBTXT+3,ARR SAVE RETURN @. RB 02170000
      SBF $BDATT(,DTF),$BCRAN+$BCITB SET OF CURRENT TRANSP AND ITBRB 02180000
      MVC CMBTS1+1(1),SAVTA2 MOVE IN TAS ATTR 2. RB 02190000
      SBF CMBTS1+1,ALLBIT-TASITB-TASTSP LEAVE TRANSP AND ITB SET RB 02200000
      ALC CMBTS1+1(1),CMBTS1+1 CORRECT TO DTF BIT DEFINITION. RB 02210000
CMBTS1 EQU * * RB 02220000
      SBN $BDATT(,DTF),# SET TAS TRANSP AND ITB RB 02230000
      MNN $BDAT1(,DTF),SAVTA2 MOVE IN SPAN AND RECSEP IND'S. RB 02240000
      SBF $BDAT1(,DTF),ALLBIT-$BCSEP-$BCSPN-$BCPLR SET OTHERS OFF. RB 02250000
      L LCBPL@(,DTF),PL LOAD PARM LIST REG. RB 02260000
      TBN PL$OPC(,PL),OPUSER SYSTEM FUNCTION ? RB 02270000
      L PLTUBA(,PL),TUB LOAD REG TO POINT TO TUB. RB 02280000
      JT CMBSYS YES-GO SET SYSTEM BLOCK LEN. RB 02290000
      SPACE                                                    02300000
* DETERMINE THE BLOCK LENGTH GIVEN THE TAS RECORD LENGTH AND RB 02310000
* BLOCKING FACTOR.                                          RB 02320000
      SPACE                                                    02330000
      MVC LCBKLC(2,DTF),TUBRCL(,TUB) MOVE IN CURRENT RECORD LGTH. RB 02340000
      MVC LCBOWN(1,DTF),TUBBK(,TUB) PUT BLOCK FACTOR IN WRK AREA. RB 02350000
CMBLAD EQU * * RB 02360000
      SLC LCBOWN(1,DTF),X$0001 DECREMENT BLOCKING FACTOR. 0 ? RB 02370000
      JE CMBOWN YES-GO SET OWNERSHIP STATUS. B RB 02380000
      ALC LCBKLC(2,DTF),TUBRCL(,TUB) ADD ANOTHER RECORD LENGTH. RB 02390000

```

	B	CMBLAD	GO TO CHECK FOR MORE RECORDS.	RB	02400000
		SPACE			02410000
CMBSYS	EQU	*	SET SYSTEM BLOCK LENGTH.	RB	02420000
	MVC	LCBKLC(2,DTF),#CCMCL	MOVE IN MAX COMMAND LENGTH.	RB	02430000
		SPACE			02440000
*		SET OWNERSHIP STATUS IN THE TUB AND THE LCB.		RB	02450000
		SPACE			02460000
CMBOWN	EQU	*	SET OWNERSHIP.	RB	02470000
*		DON'T SET TUB OWNERSHIP IF ERROR COMP CODE			02471000
		SPACE 1			02472000
	L	\$BDIOB(,DTF),XR1	XR1--> CURRENT IOB		02473000
	CLI	IOBCMP(,XR1),\$BCEOT	HIGHER THAN X'42' AND -----		02474000
	TBF	IOBCMP(,XR1),BIT0	* LESS THAN X'80'-----		02475000
	TBN	LCBAT2(,DTF),LCBACT	* AND LINE ACTIVE ?--		02475500
	L	LCBPL@(,DTF),PL	XR1--> PARM LIST		02476000
	L	PLTUBA(,PL),TUB	XR1 --> TUB		02477000
	JC	CMNOWN,TRUAIH	JUMP IF ERROR RETURN CODE<--		02478000
	SBN	TUBAT2(,TUB),TUBOWN	SET TUB OWNERSHIP.	RB	02480000
CMNOWN	EQU	*	*		02485000
	MVC	LCBOWN(2,DTF),TUBTCB(,TUB)	SAVE TCB @ OF OWNER IN LCB.	RB	02490000
	AIF	(&NSWL).S0900			02500000
&MIX	SETA	&NPP+&NMP+&NCS			02510000
	AIF	(&MIX EQ '3').S0800			02520000
	TBN	TUBAT1(,TUB),TUBSWC	SWITCHED LINE ?	RSLB	02530000
	JF	CMBTXT	NO-GO EXIT.	RSLB	02540000
.S0800	ANOP				02550000
	ST	LCBOWN(,DTF),TUB	SET ADDRESS OF OWNING TUB.	RSB	02560000
.S0900	ANOP				02570000
	J	CMBTXT	GO TO RETURN.	RB	02580000
	AGO	.N0500			02590000
.N0400	ANOP				02600000
	TITLE	'\$E093/CMTASV--INTERFACE TO \$CC4B1'			02610000
*****				MIN B	02620000
*		CALL IN TRANSIENT FORM OF 'CMTASV', 'CMBTAS' AND 'CMFORB'*		MIN B	02630000
*****				MIN B	02640000
		SPACE 1			02650000
CMTASV	EQU	*	CALL \$CC4B1 FOR TASV	MIN B	02660000
	MVI	CMTID1,CTTASV	MOVE IN ID	MIN B	02670000
	J	CMCAL1	CALL THE TRANSIENT	MIN B	02680000
		SPACE			02690000
CMREJT	MVI	CMTID1,CTREJC	MOVE IN ID	MIN B	02700000
	J	CMCAL1	CALL THE TRANSIENT	MIN B	02710000
CMBTAS	EQU	*	*	MIN B	02720000
	MVI	CMTID1,CTBTAS	MOVE IN ID FOR 'CMBTAS'.	MIN B	02730000
CMCAL1	EQU	*	*	MIN B	02740000
	ST	CMRTN1+3,ARR	SAVE THE RETURN ADDRESS.	MIN B	02750000
		SPACE 1			02760000
	SVC	0	##### TRANSIENT CALL #####	MIN B	02770000
	DC	AL1(CCPRIB)	CCP SVC RIB	MIN B	02780000
	DC	AL1(CC4B1)	CALL IN MIN XIENT #2.	MIN B	02790000
CMTID1	DC	AL1(0)	ID OF MIN SYST FUNCTION	MIN B	02800000
SAVTA1	DC	XL1'0'	TERMINAL ATTRIB BYTE 1.	MIN B	02810000
SAVTA2	DC	XL1'0'	TERMINAL ATTRIB BYTE 2.	MIN B	02820000
SAVRCL	DC	XL2'00'	TERM ATTR SET RECORD LGTH.	MIN B	02830000
CMSPHY	DC	AL1(0)	SAVE ARE FOR TUBPHY.	MIN B	02840000
		SPACE 1			02850000
CMRTN1	EQU	*	*	MIN B	02860000
	B	#	RETURN TO CALLER.	MIN B	02870000
.N0500	ANOP				02880000
	AIF	(&MIN).N0600			02890000

```

TITLE '$E093/CMTASV--SET-AND-SAVE-TAS-VALUE-IN-CORE' 02900000
***** RB 02910000
* RB 02920000
* NAME--CMTASV * RB 02930000
* RB 02940000
* TITLE--SET AND SAVE TERMINAL ATTRIBUTES * RB 02950000
* RB 02960000
* FUNCTION--SET AND SAVE TAS BYTES 1 AND 2 , TUBPHY AND TUBRCL.* RB 02970000
* RB 02980000
* ENTRY REGS: XR1-> PL. * RB 02990000
* EXIT REGS: SAME AS ENTRY. * RB 03000000
***** RB 03010000
SPACE 03020000
CMTASV EQU * RB 03030000
ST CMBTXT+3,ARR SAVE THE RETURN ADDRESS. RB 03040000
ST CMTSX1+3,PL SAVE THE PL REG. RB 03050000
SBF PL$OPC(,PL),OPUSER RESET FUNCTION INDICATOR. RB 03060000
TBF PLOPM(,PL),OP$SYS USER REQUEST, AND RB 03070000
TBF PL$OPC(,PL),OPRFSH+OPLSNS * NOT POLL STATUS OR REFRESH ?RB 03080000
L PLTUBA(,PL),TUB LOAD THE TUB REG. RB 03090000
MVC CMSPHY,TUBPHY(1,TUB) SAVE THE TUBPHY BYTE IN CORE. RB 03100000
JF CMTSYS NO-GO SET SYSTEM TAS VALUES. RB 03110000
MVC SAVRCL,TUBRCL(4,TUB) SAVE THE USER TAS VALUES. RB 03120000
J CMTSX1 GO TO EXIT. RB 03130000
SPACE 03140000
CMTSYS EQU * RB 03150000
MVI SAVTA1,TASAUT SET SET AUTO CONNECT VALUE RB 03160000
MVI SAVTA2,TASMSG SET MESSAGE MODE IN TAS 2. RB 03170000
L CMTSX1+3,PL RELOAD THE PL REG. RB 03180000
SBN PL$OPC(,PL),OPUSER INDICATE SYSTEM FUNCTION. RB 03190000
CMTSX1 EQU * RB 03200000
LA #,PL RELOAD THE USER PL REG. RB 03210000
CMBTXT EQU * RB 03220000
B # RETURN. RB 03230000
.N0600 ANOP 03240000
TITLE '$E093/CMGINL--DETERMINE-THE-OPERATION-INPUT-LENGTH' 03250000
***** B 03260000
* B 03270000
* NAME--CMGINL * B 03280000
* B 03290000
* TITLE--DETERMINE THE OPERATION INPUT LENGTH * B 03300000
* B 03310000
* FUNCTION-- * B 03320000
* DETERMINE THE INPUT RECORD LENGTH. SET TRUNCATED * B 03330000
* INDICATOR IF INPUT LENGTH IS LESS THAN ACTUAL DATA * B 03340000
* LENGTH. * B 03350000
* B 03360000
* OPERATION-- * B 03370000
* . SET THE INPUT LENGTH FOR THE OPERATION. * B 03380000
* - POLL FOR STATUS, USE 20. * B 03390000
* - RECORD MODE INPUT - USE PLINL IF PLINL LESS THAN * B 03400000
* TUBRCL * B 03410000
* - USE TUBRCL IF PLINL GREATER * B 03420000
* THAN TUBRCL. * B 03430000
* B 03440000
* . IF NOT MLMP VARIABLE LENGTH RECORD SUPPORT ADJUST * B 03450000
* INPUT LENGTH IF LESS DATA IS AVAILABLE IN THE ACTUAL * B 03460000
* LINE BUFFER THAN IS REQUESTED. * B 03470000
* B 03480000
* . IF MORE DATA IS AVAILABLE IN LINE BUFFER THAN * B 03490000

```



```

*          REQUESTED, THEN SET THE TRUNCATED DATA INDICATOR. * B 03500000
*          * B 03510000
*          . IF PURE GET REQUEST OR A MESSAGE MODE INPUT AT * B 03520000
*          OTHER THAN RECEIVE INITIAL TIME RETURN TO USER * B 03530000
*          AFTER PERFORMING THE ABOVE FUNCTIONS. * B 03540000
*          * B 03550000
*          . OTHERWISE, ADD 4 BYTES FOR GETMAIN PARAMETER LIST. * B 03560000
*          * B 03570000
*          . FREE THE CURRENT LCB HOLD BUFFER IF IT IS LARGER THAN * B 03580000
*          NEEDED, AND GETMAIN THE EXACT HOLD AREA NEEDED. * B 03590000
*          * B 03600000
*          . IF GETMAIN AREA NOT AVAILABLE, WAIT FOR IT TO BE * B 03610000
*          AVAILABLE. * B 03620000
*          * B 03630000
*          . OTHERWISE, SET UP THE ACQUIRED AREA FOR THE INVITE * B 03640000
*          INPUT OPERATION. * B 03650000
*          * B 03660000
*          * B 03670000
*          ENTRY POINT--CMGINL * B 03680000
*          * B 03690000
*          INPUT-- * B 03700000
*          XR2 - ADDRESS OF THE DTF * B 03710000
*          * B 03720000
*          OUTPUT-- * B 03730000
*          XR1 - NOT RESTORED. * B 03740000
*          XR2 - ADDRESS OF THE DTF. * B 03750000
*          $BDREL - SET UP FOR INPUT LENGTH OF THE NEXT RECORD. * B 03760000
*          * B 03770000
*          EXTERNAL REFERENCES-- * B 03780000
*          CMFMRT - FREEMAIN CURRENT HOLD BUFFER IF LARGER * B 03790000
*          THAN IS NEEDED. * B 03800000
*          CMGMRT - GETMAIN EXACT BUFFER NEEDED. * B 03810000
*          * B 03820000
*          EXIT, NORMAL--TO NSI OF CALLER. * B 03830000
*          * B 03840000
*          EXIT, ERROR--IF SPACE IS NOT AVAILABLE, EXIT TO CMPAII AND WAIT * B 03850000
*          UNTIL THE REQUIRED SPACE IS AVAILABLE. * B 03860000
*          * B 03870000
***** * B 03880000
SPACE 03880000
CMGINL EQU * * B 03890000
ST CMGIXT+3,ARR SAVE RETURN @. B 03900000
L LCBPL@(.DTF),PL LOAD PARM LIST PTR TO GET THE B 03910000
TBF $BDAT1(.DTF),$BCRES SPANNING RECORD IN PROCESS ? B 03920000
TBF LCBAT1(.DTF),LCBEOT OR SENDING EOT ? B 03930000
BF CMGIXT YES-GO EXIT EVERYTING'S DONE. B 03940000
SPACE 1 03950000
*-----* B 03960000
* NOT PROCESSING SPANNED RECORD * B 03970000
*-----* B 03980000
SPACE 1 03990000
MVC $BDREL(2,DTF),PLINL(,PL) MOVE INL TO DTF RECL AREA. B 04000000
MVC $BDWKB(2,DTF),PLRECA(,PL) MOVE CURRENT RECORD ADDR TO DTFB 04010000
&MIX SETA &N32+&N37+&N41 04020000
AIF (&MIX EQ '3').T1500 04030000
TBN PL$OPC(,PL),OPLSNS POLL FOR STATUS OP ? 7/0/5B 04040000
JF CMGIUT NOGO CHECK UNIT TYPE OF OP7/0/5B 04050000
SPACE 1 04060000
*-----7/0/5B 04070000
* POLL FOR STATUS 7/0/5B 04080000
*-----7/0/5B 04090000

```

SPACE 1			04100000
MVC	\$BDREL(2,DTF),RELSNS	USE STANDARD STATUS RECL. 7/0/5B	04110000
J	CMGIRL	GO USE THIS LENGTH FOR GET7/0/5B	04120000
.T1500 ANOP			04130000
SPACE 1			04140000
-----			B 04150000
* NOT POLL FOR STATUS			* B 04160000
-----			* B 04170000
SPACE 1			04180000
CMGIUT EQU	*	*	B 04190000
TBN	SAVTA2,TASREC	RECORD MODE ?	B 04200000
JF	CMGIBK	NO-GO CHECK DATA LEFT IN BLOCK.	B 04210000
MVC	\$BDWKB(2,DTF),LCBIBA(,DTF)	USE INVITE BUF AREA FOR MLMP.	B 04215000
L	PLTUBA(,PL),TUB	LOAD THE TUB REG.	B 04220000
CLC	\$BDREL(2,DTF),TUBRCL(,TUB)	INL LT TAS RECL ?	B 04230000
JL	CMGIMX	YES-GO FILL MAX RECL ACCEPTABLE	B 04240000
MVC	\$BDREL(2,DTF),TUBRCL(,TUB)	OTHERWISE USE TAS RECORD LEN.	B 04250000
CMGIMX EQU	*	*	B 04260000
MVC	\$BDMRL(2,DTF),\$BDREL(,DTF)	FILL MAX REC LEN ACCEPTABLE.	B 04270000
SPACE			04280000
CMGIBK EQU	*	*	B 04290000
TBF	SAVTA2,TASITB+TASVRL+TASSPAN	MLMP VARIABLE RECORD SUPPORT?	B 04300000
JF	CMGII@	YES-GO GET II RECORD @.	B 04310000
L	\$BDIOB(,DTF),IOB	POINT TO THE IOB.	B 04320000
CLI	IOBCMP(,IOB),PROCES	IOB BEING PROCESSED ?	B 04330000
JE	CMGIDX	YES-IOB READY, GO CK DATA LEN.	B 04340000
CLI	IOBCMP(,IOB),\$BCEOT	EOT OR IOB COMPLETE ?	B 04350000
TBF	SAVTA2,TASMSG	AND NOT MESSAGE MODE INPUT ?	B 04360000
JC	CMGI00,TRUAEQ	EOT TO RECORD/BLOCK, 0 IN-LEN.	B 04370000
TBF	SAVTA2,TASMSG	MESSAGE MODE?	B 04373000
JC	CMGIRL,TRUahi	REC OR BLK AND MSG MODE--JUMP	B 04376000
JNL	CMGIXT	NOT COMPLETE, WAIT FOR OP END.	B 04380000
MVC	\$BDBKX(2,DTF),IOBTAR(,IOB)	SET UP BKX LIKE MLMP WILL.	B 04390000
ALC	\$BDBKX(2,DTF),X\$0001	UPDATE FOR -STX-.	B 04400000
AIF	(&NTSP).I0600		04410000
TBN	\$BDATT(,DTF),\$BCRAN	TRANSPARENCY MODE ?	XB 04420000
JF	CMGIDX	NOT TRANSP, GO CK DATA LEGTH.	XB 04430000
ALC	\$BDBKX(2,DTF),X\$0001	UPDATE FOR -DLE-.	XB 04440000
.I0600 ANOP			04450000
CMGIDX EQU	*	*	B 04460000
CLC	\$BDBKX(2,DTF),IOBCAR(,IOB)	DATA LEFT IN BUFFER ?	B 04470000
JNL	CMGIXT	NO-GO EXIT, WAIT FOR OP END.	B 04480000
MVC	LCBWRK(2,DTF),IOBCAR(,IOB)	FIND THE NUMBER OF DATA	B 04490000
SLC	LCBWRK(2,DTF),\$BDBKX(,DTF)	* CHARACTERS LEFT IN BUFFER.	B 04500000
CLC	\$BDREL(2,DTF),LCBWRK(,DTF)	INL LT MAX LINE DATA LEFT.	B 04510000
JNL	CMGIAL	NO-GO SET TO GET ALL DATA.	B 04520000
SBN	LCBAT2(,DTF),LCBTRC	SET BLOCK TRUNCATED IND.	B 04530000
J	CMGII@	GO CHECK INVITE RECORD @.	B 04540000
SPACE			04550000
CMGI00 EQU	*	*	04560000
SLC	\$BDREL(2,DTF),\$BDREL(,DTF)	ZERO OUT INPUT RECORD LENGTH.	B 04570000
J	CMGII@	GO CHECK INVITE RECORD LENGTH.	B 04580000
SPACE			04590000
CMGIAL EQU	*	*	B 04600000
* ACTUAL AMOUNT OF DATA FOR MLMP TO MOVE, MAY BE LESS THAN PLINL			B 04610000
MVC	\$BDREL(2,DTF),LCBWRK(,DTF)	MOVE MAX LINE LEN TO RECL.	B 04620000
SPACE 1			04630000
-----			* B 04640000
* DETERMINE SIZE OF HOLD BUFFER NEEDED			* B 04650000
-----			* B 04660000

	SPACE 1			04670000
CMGII@	EQU *	*	B	04680000
	L LCBPL@ (,DTF),PL	LOAD @ OF CURRENT PARM LIST.	B	04690000
	TBN PLOPM (,PL),OP\$SYS	SYSTEM REQUEST ?	B	04700000
	AIF (&NRUF).F0100			04701000
	L PLTUBA (,PL),XR2	XR2-> TUB.	WB	04703000
	TBF TUBSCS (,XR2),TUBRUF	AND NOT A RUF ?	WB	04704000
	L TUBDTF (,XR2),DTF	XR2-> LCB (DTF).	WB	04705000
.F0100	ANOP			04707000
	JT CMGISR	YES-GO SET UP SYSTEM RECORD LENB		04710000
	AIF (&NMSG).E0100			04720000
	TBN SAVTA2,TASMSG	GET-MSG MODE ?	GB	04730000
	JT CMGIMG	YES-GO CHECK RECEIVE INIT IND.	GB	04740000
.E0100	ANOP			04750000
CMGIRL	EQU *	*	B	04760000
	MVC #BUFND,\$BDREL(2,DTF)	USE CURRENT RECORD LEN AND ADD	B	04770000
	J CMGICB	GO ADD GETMAIN CONTROL BYTES.	B	04780000
	SPACE 1			04790000
	AIF (&NMSG).E0110			04800000
*	-----*		GB	04810000
*	USER GET MESSAGE		GB	04820000
*	-----*		GB	04830000
	SPACE 1			04840000
CMGIMG	EQU *	*	GB	04850000
	TBN LCBAT2 (,DTF),LCBRCI	RECEIVE INITIAL TIME ?	GB	04860000
	JF CMGIXT	NO-GO EXIT, BUFFERS ARE SET.	GB	04870000
	SPACE 1			04880000
.E0110	ANOP			04890000
*	-----*		B	04900000
*	SYSTEM REQUEST OR USER GET MESSAGE		B	04910000
*	-----*		B	04920000
	SPACE 1			04930000
CMGISR	EQU *	*	B	04940000
	MVC #BUFND,PLINL(2,PL)	USE PARM LIST INL FOR SYST.	B	04950000
	SPACE 1			04960000
*	-----*		B	04970000
*	ANY CALL EXCEPT USER GET MESSAGE PRIOR TO 1ST OP END.		B	04980000
*	-----*		B	04990000
	SPACE 1			05000000
CMGICB	EQU *	*	B	05010000
	ALC #BUFND(2),X\$0004	* GETMAIN PARM LIST REQ'MTS.	B	05020000
	SPACE			05030000
	B CMGBUF	GETMAIN RIGHT SIZE BUFFER	B	05040000
*		IF SPACE NOT AVAILABLE, CMGBUF	B	05040100
*		*DOES NOT RETRUN, B CMPAII.	B	05040200
	SPACE			05050000
	MVC \$BDWKB(2,DTF),LCBIBA (,DTF)	USE INVITE BUF AREA FOR MLMP.	B	05060000
	L LCBPL@ (,DTF),PL	LOAD PARM LIST REG.	B	05070000
&MIX	SETA &N32+&N37+&N41			05073000
	AIF (&MIX EQ '3').T1700			05076000
	TBN PL\$OPC (,PL),OPLSNS	POLLING FOR STATUS ?	0/5/7B	05080000
	JF CMGMOV	NO-GO DO MOVE OF RECORD @.	0/5/7B	05090000
	TBN PL\$OPC (,PL),OPGET	IF OP WAS INPUT , PLRECA	0/5/7B	05100000
*		* MUST BE SET TO GET	0/5/7B	05110000
*		* DATA RATHER THAN STATUS.	0/5/7B	05120000
	JF CMGIXT	NO-NOT INVITE, SKIP MOVE.	0/5/7B	05130000
*		(PUT MUST PRESERVE REC @)	0/5/7B	05140000
CMGMOV	EQU *	* LOCAL	0/5/7B	05150000
.T1700	ANOP			05155000
	MVC PLRECA(2,PL),\$BDWKB (,DTF)	PUT INVITE @ INTO PARM LIST.	B	05160000

```

CMGIXT EQU * * B 05170000
      B ## RETURN B 05180000
      AIF (&NDF).F0200 05190000
      TITLE '$E093/CMDFFQ---QUEUE REQUEST FOR DFF TASK' 05200000
***** FB 05210000
* * FB 05220000
* NAME: CMDFFQ * FB 05230000
* * FB 05240000
* TITLE: QUEUE REQUEST FOR DFF TASK AND POST. * FB 05250000
* * FB 05260000
* INPUT: XR1-> PL * FB 05260100
* * FB 05260200
* OUTPUT: XR1- UNCHANGED * FB 05260300
* XR2- DESTROYED * FB 05260400
* * FB 05260500
* EXIT-NORMAL: TO NSI OF CALLER. * FB 05260600
* * FB 05260700
***** FB 05270000
      SPACE 1 05280000
CMDFFQ EQU * * FB 05290000
      ST CMDFRT+3,ARR RETURN ADDRESS FB 05300000
      ST CMDFR1+3,XR1 SAVE XR1 FB 05310000
* ----- START @20 05311000
      L PLTUBA(,PL),XR2 LOAD POINTER TO THE TUB. FB 05312000
      L TUBTCB(,XR2),XR2 XR2-> TCB OWNING THIS TUB. FB 05313000
      TBF TCBFG2(,XR2),TCBTRC TASK IN TERMINATION AND FB 05314000
      CLI TCBID(,XR2),TRMTSK * TERMINATION TASK ? FB 05314500
      JC CMDFNT,TRUNEQ NO - GO TO POST DFF. FB 05315000
      CLI PLOPC(,PL),OPACI IS THIS ACCEPT INPUT PL ? FB 05315100
      JNE CMDNAC NO - DON'T SET INVITE PL FB 05315200
      L PLTUBA(,PL),XR2 POINT TO THE TUB FB 05315300
      L TUBPL@(,XR2),PL POINT TO THE INVITE PL FB 05315400
CMDNAC EQU * * LOCAL FB 05315500
      B $CC4FR GO TO FREEMAIN RECORD AREA. FB 05316000
      J CMDFR1 RETURN WITHOUT POSTING DFF. FB 05317000
CMDFNT EQU * * LOCAL FB 05318000
* ----- END @20 05319000
      MVI PLCHN-1(,PL),NOBIT ZERO CHAIN FIELD OF PL FB 05320000
      LA @DFFQ-1,XR2 ADDRESS OF QUEUE FB 05330000
CMDFEN EQU * * LOCAL FB 05340000
      CLI PLCHN-1(,XR2),NOBIT IS CHAIN PTR NULL FB 05350000
      JE CMDFAD YES - ADD TO DFF QUEUE END FB 05360000
      L PLCHN(,XR2),XR2 POINT TO NEXT QUEUE ENTRY FB 05370000
      B CMDFEN TEST FOR END FB 05380000
      SPACE 1 05390000
CMDFAD EQU * * LOCAL FB 05400000
      ST PLCHN(,XR2),PL ADD PARM LIST TO END OF QUEUE FB 05410000
      SPACE 1 05420000
* POST DFF FB 05430000
      SPACE 1 05440000
      LA $DFECB,XR1 ADDRESS OF DFF ECB FB 05450000
      SVC 0 FB 05460000
      DC AL1(POSTRB) POST DFF FB 05470000
      SPACE 1 05480000
CMDFR1 LA *-*,XR1 RESTORE XR1 FB 05490000
CMDFRT B *-* RETURN FB 05500000
.F0200 ANOP 05510000
      TITLE '$E093/CMDTFS---BSCA DTF SETUP FOR CHECK' 05520000
***** B 05530000
* * B 05540000

```

```

* NAME : CMDTFS * B 05550000
* * B 05560000
* FUNCTION : SET UP DTF FOR BSCA LINE PRIOR TO GOING TO IOCS CHECK* B 05570000
* ROUTINE. * B 05580000
* * B 05590000
* OPERATION : * B 05600000
* CHECK FOR BSCA DTF'S WITH OPERATION COMPLETE, WHEN FIND ONE * B 05610000
* DO 1 OF THE FOLLOWING: * B 05620000
* 1. IF ABORTING THE LINE, GO TO CHECK WITH NO DTF SETTING UP. * B 05630000
* 2 .IF OLT RUNNING/OUTPUT SELECT/OR DATA MOVED, NO DTF SETUP * B 05640000
* NEEDED PRIOR TO CALLING CHECK. * B 05650000
* 3. IF RECEIVE INITIAL, SET UP DTF AND IOB FOR TERMINAL READING* B 05660000
* UNLESS EOT RECEIVED TO RECEIVE INITIAL, IF SO CALL CHECK* B 05670000
* 4. IF NOT RECEIVE INITIAL, VALIDATE INPUT LINE BUFFER SIZE. * B 05680000
* * B 05680100
* * B 05680200
* EXTERNAL SUBROUTINES USED: * B 05680300
* CMCSKP - SET CHECK LIST SKIP BITS. * B 05680400
* CMTASV - SAVE TERMINAL ATTRIBUTES. * B 05680500
* CMBTAS - FILL DTF/IOB OWNERSHIP. * B 05680600
* CMGINL - GETMAIN BUFFER FOR AMOUNT OF DATA RECEIVED. * B 05680700
* $CC4BI - CHECK FOR DATA MODE ESCAPE. * B 05680800
* * B 05680900
* INPUT: * B 05681000
* LCB#1 - ADDRESS LCB CHAIN. * B 05681100
* @CKLST - ADDRESS OF CHECK LIST. * B 05681200
* * B 05681300
* OUTPUT: * B 05681400
* CHECK LIST - SKIP BITS SET SO CMBMCH WILL HANDLE OP END. * B 05681500
* DTF - THE OP ENDING DTF IS SETUP FOR MLMP. * B 05681600
* XR1, XR2 - DESTROYED. * B 05681700
* * B 05681800
* * B 05681900
* * B 05682000
* * B 05682100
* * B 05682200
* * B 05690000
*****
SPACE 05700000
CMDTFS EQU * * ENTRY POINT B 05710000
ST CMDTFX+3,ARR SAVE RETURN ADDRESS B 05720000
SPACE 05730000
* LOCATE ANY BSCA DTF'S. B 05740000
SPACE 05750000
SBF CMSWIT,CMBSCK RESET BSCA DTF CHECK SCDEDULED. B 05760000
L @LCB#1,DTF LOAD @ 1ST LCB. B 05770000
CMENR EQU * * B 05780000
AIF (&ONE).C0305 05790000
TBN $BDDEV(,DTF),BSCA BSCA DTF ? 2 05800000
JF CMEANR NO-GO CHECK FOR ANOTHER LCB. 2 05810000
SPACE 05820000
***** 2 05830000
* BSCA DTF FOUND - CHECK FOR OP END * 2 05840000
***** 2 05850000
SPACE 1 05860000
.C0305 ANOP 05870000
SPACE 05880000
ST CMSDTF,DTF SAVE THE DTF ADDRESS. B 05890000
&MIX SETA (&BSC+&MLA) 05900000
AIF (&MIX LE '1').Y0100 05910000
* SET CHECK LIST TO IGNORE DTF IF DATA IS NOT READY TO GO. YB 05920000

```

```

L      $BDIOB(,DTF),IOB      LOAD PTR TO 1ST IOB.      YB 05930000
MVI    CMC#SB,SBN2          SET OP TO SET SKIP ON.    YB 05940000
SPACE                                     05950000
B      CMCSKP                GO TO CHECK LIST SKIP BIT RTN.YB 05960000
SPACE                                     05970000
*      IF THERE IS ALREADY A BSCA DTF SET UP FOR CHECK, THEN DONT SET YB 05980000
*      UP ANOTHER BSCA DTF FOR CHECK.  SOME INTERNAL FIELDS WILL BE YB 05990000
*      LOST IF ANOTHER DTF WHERE SET UP.      YB 06000000
SPACE                                     06010000
TBN    CMSWIT,CMBSCK        BSCA DTF SET FOR CHECK ?    YB 06020000
JT     CMEANR                YES-DON'T SET UP ANOTHER ONE. YB 06030000
TBF    LCBOPE(,DTF),ALLBIT  ANY OP ENDS FOR THIS LINE ? YB 06040000
JF     CMEXPM                YES-GO EXAMINE PARM LIST STAT YB 06050000
CMEANR EQU *                * (LOOP BACK FOR ANOTHER DTF) YB 06060000
CLI    LCBCHN-1(,DTF),NOBIT ANOTHER DTF ?      YB 06070000
BE     CMECHK                NO-GO TO CALL CHECK.      YB 06080000
L      LCBCHN(,DTF),DTF     POINT TO NEXT DTF.      YB 06090000
B      CMENTR                GO TEST FOR BSCA DTF.      YB 06100000
.Y0100 ANOP                  06110000
SPACE 1                      06120000
***** B 06130000
*      BSCA DTF WITH OP END FOUND - CHECK FOR PARAMETER LIST * B 06140000
***** B 06150000
SPACE 1                      06160000
*      EXAMINE THE PARM LIST, IGNORE OP END INTERRUPT IF PARM LIST NOT B 06170000
*      QUEUED.                B 06180000
SPACE                          06190000
CMEXPM EQU *                * LOCAL                B 06200000
TBN    LCBAT1(,DTF),LCBNTQ  PARM LIST REMOVED FROM QUEUE ? B 06210000
JF     CMBDTF                NO-GO SETUP DTF FOR $$BMCH B 06220000
*      * WILL RETURN TO CMEANR B 06230000
SBN    LCBAT1(,DTF),LCBINT  SET OP END W/ NO PARM LIST Q'D. B 06240000
SLC    LCBOPE(1,DTF),X$0001 DECREMENT LINE OP END COUNT. B 06250000
SLC    #OPEND,X$0001(1)     SUBTRACT 1 FROM OP END COUNT. B 06260000
B      CMOPND                GO CHECK FOR OTHER OP END'S. B 06270000
SPACE 3                      06280000
***** B 06290000
*      SET UP DTF FOR THIS LINE * B 06300000
***** B 06310000
SPACE 1                      06320000
CMBDTF EQU *                * B 06330000
SBN    CMSWIT,CMBSCK        SET DTF SCHEDULED FOR CHECK B 06340000
&MIX   SETA (&BSC+&MLA)     06350000
AIF    (&MIX LE '1').Y0150 06360000
MVI    CMC#SB,SBF2          SET OP TO SET SKIP OFF.    YB 06370000
SPACE                                     06380000
B      CMCSKP                GO TO CHECK LIST SET SKIP RTN.YB 06390000
.Y0150 ANOP                  06400000
SPACE                          06410000
***** B 06420000
*      IF LINE IS BEING ABORTED, THEN GO TO CHECK WITH NO SETTING UP.* B 06430000
***** B 06440000
SPACE                          06450000
TBN    LCBAT2(,DTF),LCBABT  LINE ABORT IN PROCESS ? B 06460000
BT     CMEBCK                YES-NO DTF SETTING UP,NEXT DTF B 06470000
SPACE                                     06480000
*      SAVE THE TAS ATTRIBUTES FROM THE TUB IN SAVE AREA. B 06490000
SPACE                          06500000
TBN    $BDAT1(,DTF),$BCRES  SPANNING RECORD IN PROCESS ? B 06510000
BT     CMEBCK                YES - DONT SETUP DTF, MLMP IS B 06520000

```

```

*          * SEARCHING FOR RECORD SEP.      B 06530000
L      LCBPL@ ( ,DTF) ,PL      LOAD PARM LIST OF ACTIVE TERM, B 06540000
SPACE                                     06550000
B      CMTASV      GO TO TAS SAVE SUBROUTINE.      B 06560000
SPACE                                     06570000
AIF    (&NSWL) .S1060      06580000
SPACE                                     06590000
*      SET ON LINE CONNECTED INDICATOR FOR SWITCHED LINES.      SLB 06600000
SPACE                                     06610000
&MIX   SETA   &NCS+&NPP+&NMP      06620000
AIF    (&MIX EQ '3') .S1050      06630000
TBN    $BDATR ( ,DTF) , $BCSWI      SWITCHED      SLB 06640000
TBF    $BDATR ( ,DTF) , $BCMPT      *          LINE ?      SLB 06650000
JF     CMEXCK      NO-DON'T SET CONNECTED.      SLB 06660000
.S1050 ANOP      06670000
SBN    LCBATR ( ,DTF) ,LCBNIT      SET SWITCHED LINE CONNECTED.      SB 06680000
CMEXCK EQU      *          * LOCAL      SB 06690000
.S1060 ANOP      06700000
SPACE                                     06710000
***** B 06720000
*      IF OUTPUT OPERATION - NO DTF SETUP NEEDED PRIOR TO CHECK      * B 06730000
***** B 06740000
SPACE                                     06750000
TBF    LCBOPC ( ,DTF) ,LCBMVD      DATA MOVED OR      B 06760000
TBF    PL$OPM ( ,PL) ,OPPUT      OUTPUT (SELECTION) OPERATION ? B 06770000
SBF    LCBOPC ( ,DTF) ,LCBMVD      SET OFF DATA MOVED IND.      B 06780000
BF     CMEBCK      YES-GO CHECK FOR ANOTHER DTF.      B 06790000
SPACE                                     06810000
TBN    LCBAT2 ( ,DTF) ,LCBRCI      RECEIVE INITIAL ON LINE ?      B 06820000
JF     CMEREC      NO-GO HANDLE DATA RECORD LENGTHB 06830000
SPACE 1      06840000
***** B 06850000
*      RECEIVE INITIAL ON THE LINE      * B 06860000
*      IF LINE INIT OR ONLINE TEST STILL IN PROGRESS - DTF SETUP      * B 06870000
*      IS ALREADY DONE.      * B 06880000
***** B 06890000
SPACE 1      06900000
L      $BDIOB ( ,DTF) ,IOB      POINT TO THE IOB      B 06910000
TBN    IOBFLA ( ,IOB) ,FIRST      LINE INIT STILL IN PROCESS ?      B 06920000
BT     CMEBCK      YES-GO TO CHECK OTHER DTF'S.      B 06930000
L      $BDWKA ( ,DTF) , $BWK      POINT TO BSCA WORK AREA.      B 06940000
TBN    LCBAT2 ( ,DTF) ,LCBRFT      WAS DTF SETUP FOR OLT ON PRIOR B 06950000
*          * PASS ?      B 06960000
JF     CMERFF      NO-CHECK FOR RFT COMING IN.      B 06970000
TBN    $BWFG3 ( , $BWK) , $BWRFT      OLT STILL RUNNING OR STATUS MSGB 06980000
BT     CMEBCK      YES-GO CHECK FOR OTHER DTF'S.      B 06990000
SBF    LCBAT2 ( ,DTF) ,LCBRFT      OLT NO LONGER RUNNING, SET OFF. B 07000000
J      CMERCI      GO HANDLE THE RECEIVE INITIAL.      B 07010000
SPACE 1      07020000
CMERFF EQU      *          * LOCAL      B 07030000
TBN    $BWFG3 ( , $BWK) , $BWRFT      OLT RUNNING OR STATUS MESSAGE B 07040000
JF     CMERCI      NO-GO HANDLE RECEIVE INITIAL.      B 07050000
SBN    LCBAT2 ( ,DTF) ,LCBRFT      SET RFT STARTED AND RUNNING.      B 07060000
CMERCI EQU      *          * LOCAL      B 07070000
SPACE                                     07080000
*-----* B 07090000
*      IF CANCEL SUCCESSFUL (NO DATA) -- NO DTF SETUP PRIOR TO CHECK * B 07100000
*-----* B 07110000
SPACE                                     07120000
L      $BDIOB ( ,DTF) ,IOB      LOAD THE IOB REG.      B 07130000

```

CLI	IOBCMP(,IOB), \$BCNEG	NEGATIVE RESPONSE ?	B	07140000
JE	CMEBCK	YES-GO CHECK OTHER DTFS	B	07150000
TBF	LCBAT1(,DTF),LCBCRI+LCBPRI	CANCEL PENDING ON THE LINE ?	B	07160000
JT	CMERCL	NO - SET UP FOR RECEIVE	B	07170000
CLI	IOBCMP(,IOB), \$BCEOT	CANCEL OK ?	B	07180000
JE	CMEBCK	YES-GO CHECK FOR OTHER DTF'S.	B	07190000
SPACE				07200000
-----			B	07210000
* DATA RECEIVED -- SET UP DTF			B	07220000
-----			B	07230000
SPACE				07240000
CMERCL EQU *		* LOCAL	B	07250000
AIF	(&NPP).P0100			07260000
AGO	.P0150			07270000
.P0100 AIF	(&NMP).P0200			07280000
.P0150 ANOP				07290000
TBF	\$BDATR(,DTF), \$BCSWI	MULTI-TERM IND OFF ?	P/TB	07300000
JT	CMEOWN	YES-PARM @ IN DTF, GO LOAD	P/TB	07310000
.P0200 ANOP				07320000
AIF	(&NCS).S1100			07330000
MVC	LCBID#(1,DTF), \$BDIND(,DTF)	MOVE IN CONTROL STATION ID.	CB	07340000
TBN	\$BDATR(,DTF), \$BCMCN	CONTROL STATION ?	CB	07350000
JT	CMECTL	YES-GO FIND PARM LIST FOR CS.	CB	07360000
.S1100 ANOP				07370000
AIF	(&NSWL).S1200			07380000
SPACE				07390000
* FILL TERM ID FOR SWITCHED LINE IF SWITCHED ID LIST WAS USED.			SB	07400000
SPACE 1				07410000
TBN	\$BDADD(,DTF), \$BCSWD	SWITCHED ID LIST USED ?	SB	07420000
JF	CMEOWN	NO-TERM ADDRESS IN 'LCBPL@'.	SB	07430000
MVC	LCBID#(1,DTF), \$BDRLN(,DTF)	MOVE IN SWITCHED TERM ID.	SB	07440000
.S1200 ANOP				07450000
&MIX SETA	&NCS+&NSWL			07460000
AIF	(&MIX EQ '2').S1300			07470000
SPACE 1				07480000
* FIND PARM LIST FOR MULTI-TERMINAL LINE.			C/SB	07490000
SPACE 1				07500000
CMECTL EQU *		* LOCAL	C/SB	07510000
L	LCBPLQ(,DTF), PL	LOAD PTR TO FIRST PARM LIST.	C/SB	07520000
CMEXID EQU *		* LOCAL	C/SB	07530000
ST	LCBPL@(,DTF), PL	SAVE @ OF CURRENT PARM LIST.	C/SB	07540000
TBN	PL\$OPC(,PL), OPSTCM	IS STOP INVITE TURNED GET	C/SB	07550000
*		* STILL RECVING DATA.	C/SB	07560000
JT	CMESIJ	YES - HANDLE IT LIKE GET.	C/SB	07570000
TBN	PL\$OPM(,PL), OPGET	THIS A GET PARM LIST ?	C/SB	07580000
TBF	PLOPM(,PL), OPSTOP	* AND NOT A STOP	C/SB	07590000
JF	CMENXT	NO-GO TO GET NEXT PARM LIST.	C/SB	07600000
CMESIJ EQU *		* LOCAL	C/SB	07610000
L	PLTUBA(,PL), TUB	LOAD @ OF THIS TERMS TUB.	C/SB	07620000
CLC	TUBSID(1,TUB), LCBID#(,DTF)	THIS RESPONDING TERM'S TUB	C/SB	07630000
L	LCBPL@(,DTF), PL	RELOAD PARM LIST @.	C/SB	07640000
JE	CMESKP	YES-GO SET SKIP BIT ON.	C/SB	07650000
CMENXT EQU *		* LOCAL	C/SB	07660000
L	PLCHN(,PL), PL	LOAD REG TO NEXT PARM LIST @	C/SB	07670000
B	CMEXID	GO BACK TO EXAMINE THIS ID.	C/SB	07680000
SPACE				07690000
CMESKP EQU *		* LOCAL	C/SB	07700000
MVC	LCBLID(1,DTF), LCBID#(,DTF)	SAVE LAST TERMINAL ID	C/SB	07704000
SPACE				07710000
* SAVE THE TAS ATTRIBUTTES FOR NEW TERMINAL.				07720000

SPACE				07730000
B	CMTASV	GO TO TAS SAVE SUBROUTINE.	B	07740000
SPACE				07750000
.S1300	ANOP			07760000
*	SET DTF AND IOB ATTRIBUTES, OWN, RECEIVE IND'S, AND BLOCK LENGTH.		B	07770000
*	ALSO SET LINE OWNERSHIP STATUS.			07780000
SPACE				07790000
CMEOWN	EQU *	* LOCAL	B	07800000
SPACE				07810000
B	CMBTAS	RTN TO FILL DTF/IOB/OWNERSHIP.	B	07820000
SPACE				07830000
AIF	(&NITB).I0800			07840000
TBN	SAVTA2,TASREC+TASITB	RECORD MODE AND ITB ?	IB	07850000
JF	CMEIOB	NO-GO CORRECT THE IOB'S.	IB	07860000
MVC	\$BDITB(1,DTF),TUBBKF(,TUB)	PUT BLOCK FACTOR IN TEMP WORKIB		07870000
AIF	(&NTSP).I0700			07880000
TBN	TUBTA2(,TUB),TASTSP	TRANSPARENCY ?	IXB	07890000
JF	CMEITB	NO-GO SET FOR ITB ONLY.	IXB	07900000
ALC	LCBKLC(2,DTF),\$BDITB(,DTF)	ADD 3 TIMES BLOCKING FACTOR	IXB	07910000
ALC	LCBKLC(2,DTF),\$BDITB(,DTF)	* BUFFER LENGTH FOR ITB	IXB	07920000
ALC	LCBKLC(2,DTF),\$BDITB(,DTF)	* TRANSPARENT GET.	IXB	07930000
MVI	\$BDITB(,DTF),3	SET ITB LENGTH TO 3.	IXB	07940000
J	CMEIOB	GO CORRECT IOB ATTRIBUTTES.	IXB	07950000
SPACE				07960000
.I0700	ANOP			07970000
CMEITB	EQU *	* LOCAL	IXB	07980000
ALC	LCBKLC(2,DTF),\$BDITB(,DTF)	ADD BLK FACTOR FOR ITB ONLY.	IB	07990000
MVI	\$BDITB(,DTF),1	SET ITB LENGTH TO 1.	IB	08000000
.I0800	ANOP			08010000
SPACE				08020000
*	CORRECT ALL IOB'S IN THE CHAIN FOR ANY NEW ATTRIBUTTES SET.			08030000
SPACE				08040000
CMEIOB	EQU *	* LOCAL	B	08050000
L	\$BDIOB(,DTF),IOB	LOAD THE IOB POINTER.	B	08060000
CMBTIB	EQU *	* LOCAL	B	08070000
MVC	IOBFLG(1,IOB),\$BDATT(,DTF)	MOVE ATTR. BYTE TO IOB.	B	08080000
CLC	IOBNXT(2,IOB),\$BDIOB(,DTF)	ANOTHER IOB IN THE CHAIN ?	B	08090000
L	IOBNXT(,IOB),IOB	LOAD @ OF NEXT IOB.	B	08100000
BNE	CMBTIB	YES-GO FILL THE NEXT IOB.	B	08110000
SPACE	1			08120000
*	IF NOT A GOOD DATA BLOCK, DON'T CHECK DME, RECORD LENGTH, ETC.		B	08130000
SPACE	1			08140000
TBF	IOBCMP(,IOB),ALLBIT-\$BCEOT	GOOD DATA BLOCK?	B	08150000
JF	CMEBCK	NO-GO CHECK OTHER DTF'S.	B	08160000
SPACE				08170000
-----			B	08180000
* GOOD DATA RECEIVED			B	08190000
-----			B	08200000
SPACE				08210000
AIF	(&NDME).D0300			08220000
SPACE	1			08230000
L	LCBPL@(,DTF),PL	POINT TO THE PARM LIST.	DB	08240000
L	PLTUBA(,PL),TUB	POINT TO THE TUB.	DB	08250000
TBF	TUBAT2(,TUB),TUBCMD	NOT COMMAND MODE,	DB	08260000
TBN	TUBAT2(,TUB),TUBDTA	, IS IN DATA MODE, AND	DB	08270000
TBN	TUBAT1(,TUB),TUBREQ	* IS THE REQUESTOR ?	DB	08280000
JF	CMEREC	NO-GO CHECK RECORD LENGTH.	DB	08290000
AIF	(&N32).D0100			08300000
TBN	TUBSCS(,TUB),TUBCLR	CLEAR KEY HIT ?	0DB	08310000
AIF	(&NCPU).D0050			08320000

```

JT      CMEBIC      YES-GO TO INQUIRY XIENT.      0DB 08330000
AGO      .D0100
.D0050 ANOP
JF      CMEREC      NO-GO CHECK RECORD LENGTH.      0DB 08360000
AGO      .D0200
.D0100 AIF      (&NCPU) .D0200
CLI      TUBPHY( ,TUB) ,TUBCPU      CPU TERMINAL ?      UDB 08390000
JNE      CMEREC      NO-GO CHECK RECORD LENGTH.      UDB 08400000
.D0200 ANOP
SPACE
*-----*
* CLEAR KEY ON DATA MODE 3270 OR CPU -- CHECK FOR DME      *
*-----*
SPACE
CMEBIC EQU      *      * LOCAL      DB 08470000
LA      CMDME ,XR1      POINT XR1 TO DME STRING.      DB 08480000
SVC      0      ##### TRANSIENT CALL #####      DB 08490000
DC      AL1(CCPRIB)      CCP SVC RIB      DB 08500000
DC      AL1(CC4BI)      BSCA DME INQUIRY.      DB 08510000
.D0300 ANOP
SPACE
***** B 08540000
* ALL READS --- DETERMINE BUFFER SIZE NEEDED - ADJUST HOLD BUFFER* B 08550000
***** B 08560000
SPACE
CMEREC EQU      *      *      B 08580000
SPACE 1
B      CMGINL      COMPUTE AMOUNT OF DATA TO MOVE B 08600000
*      GETMAIN RIGHT SIZE BUFFER      B 08610000
*      * NOW THAT WE KNOW HOW MUCH      B 08620000
*      * DATA CAME IN.      B 08630000
SPACE
CMEBCK EQU      *      * (WITHIN CMBDTF)      B 08650000
&MIX SETA      (&BSC+&MLA)      08660000
AIF      (&MIX LE '1') .Y0250      08670000
B      CMEANR      GO CHECK FOR ANOTHER DTF.      YB 08680000
SPACE 2
.Y0250 ANOP
CMECHK EQU      *      * EXIT      B 08710000
CMDTFX B      *-*      RETURN      B 08720000
.C0580 ANOP
MEND
08730000
08740000

```

```

MODULE-$E094 , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO 00010000
***** 00020000
.* NAME $E094 * 00030000
***** 00040000
$E094 00050000
GBLB &ONE, &NOB, &NOM, &MIN, &NDME 00060000
GBLB &NSW, &N1050, &N2741, &NMOVE, &NSCTL, &NBFR 00070000
TEXT 00080000
* R-02, C-01 CHANGE LEVEL 00090000
* R-02, C-02 CHANGE LEVEL DEL 00100000
AIF (&NOM).M2000 00110000
AIF (&NSCTL).S5800 00120000
TITLE '$E094/CMONSK/CMOFSK---MLTA-POLL-SKIP-BIT-ROUTINE' 00130000
***** CM 00140000
* CM 00150000
* TITLE: CMONSK/CMOFSK - 'MLTA POLL SKIP BITS ON/OFF ROUTINE.' CM 00160000
* CM 00170000
* FUNCTION: CM 00180000
* CM 00190000
* THIS ROUTINE WILL TURN ON OR OFF AS DESIGNATED, THE POLLING CM 00200000
* SKIP BIT FOR AN MLTA TERMINAL ON A CONTROL STATION LINE CM 00210000
* CM 00220000
* . A CHECK IS MADE TO ASSURE THAT THE MLTA DTF IS OPEN AND IS CM 00230000
* FOR A LINE WITH THE STATION CONTROL FEATURE CM 00240000
* CM 00250000
* . THE POLLING SKIP BIT FOR THE TERMINAL SPECIFIED BY THE CM 00260000
* PARAMETER LIST IS THEN SET ON OR OFF AS APPROPRIATE BY CM 00270000
* BRANCHING TO THE MLTA IOCS CM 00280000
* CM 00290000
* ENTRY POINTS: CM 00300000
* CM 00310000
* . CMONSK - TO SET ON THE POLLING SKIP BIT CM 00320000
* CM 00330000
* . CMOFSK - TO SET OFF THE POLLING SKIP BIT CM 00340000
* CM 00350000
* INPUT: CM 00360000
* CM 00370000
* INDEX REGISTER 1 CONTAINS THE ADDRESS OF THE PARAMETER LIST CM 00380000
* FOR THE TERMINAL FOR WHICH THE SKIP BIT IS TO BE SET CM 00390000
* CM 00400000
* OUTPUT: CM 00410000
* CM 00420000
* POLLING SKIP BIT SET ON OR OFF AS DESIGNATED CM 00430000
* CM 00440000
* EXTERNAL REFERENCES: CM 00450000
* CM 00460000
* . MLTA IOCS (MLTIO1) CM 00470000
* CM 00480000
* . 1050 SERVICE ROUTINE (CMMCT) CM 00490000
* CM 00500000
* . MLTA START CODE ERROR ($CC4SC) CM 00510000
* CM 00520000
* EXITS, NORMAL: CM 00530000
* CM 00540000
* TO INSTRUCTION FOLLOWING THE INVOKING OF THIS ROUTINE (ARR) CM 00550000
* CM 00560000
* EXITS, ERROR: CM 00570000
* CM 00580000
* TO TRANSIENT $CC4SC IF THE MLTA IOCS YIELDS A NON-ZERO CM 00590000

```

```

* OPERATION START CODE TO THE REQUEST TO SET THE POLLING SKIP BIT.
*
*
* ATTRIBUTES: RESIDES, SERIALY REUSEABLE.
*
*****
EJECT
* IF ENTER HERE - SET THE POLL SKIP BIT ON
SPACE 1
CMONSK EQU * *
SBN CMSWNF+1,BIT0 SET SWITCH TO BRANCH
J CMSVRG JUMP
SPACE
* IF ENTER HERE - SET POLL SKIP BIT OFF
SPACE
CMOFSK EQU *
SBF CMSWNF+1,BIT0 SET SWITCH TO NO OP
SPACE
CMSVRG EQU * LOCAL.
ST CMONOF+3,ARR SAVE THE ARR.
ST CMON1+3,XR1 SAVE XR1.
ST CMON2+3,XR2 SAVE XR2.
SPACE
L PLTUBA(,XR1),XR2 POINT XR2 AT THE TUB
L TUBDTF(,XR2),XR2 POINT XR2 AT THE DTF
SPACE
* ASSURE STATION CONTROL LINE, AND THAT THE LINE IS OPEN.
SPACE
TBN $MDAT2(,XR2),$MAOPN IS DTF OPEN, AND
TBN $MDTFT(,XR2),$MTPLT IS IT STATION CONTROL ?
JF CMON2 LEAVE NOW IF NOT STAION CTRL.
SPACE
* HAVE STATION CONTROL LINE, SET APPROPRIATE OP CODE IN THE DTF.
SPACE
L PLTUBA(,XR1),XR1 POINT XR1 AT TUB
SPACE
CMSWNF JC CMNNSK,X'87' JUMP TO SET SKIP ON.
MVI $MDOPC(,XR2),$MCTSF MOVE SKIP BIT OFF OP TO DTF
J CMPGTA JUMP TO PLUG TERM ADDR IN DTF
SPACE
CMNNSK EQU * LOCAL.
MVI $MDOPC(,XR2),$MCTSN MOVE SKIP BIT ON OP TO DTF
SPACE
CMPGTA EQU * LOCAL.
MVC $MDTMA(2,XR2),TUBTMA(,XR1) MOVE TERMINAL ADDR TO DTF
CMON1 LA #,XR1 RESTORE XR1
SPACE
AIF (&N1050).C5100
B CMMCT PLUG BYTE 2 OF TMA FOR 1050 1CM
.C5100 ANOP
SPACE
B MLTIO1 BR TO MLTA IOCS
SPACE
CLI $MDOSC(,XR2),NOBIT CHECK START CODE
JE CMON2 JUMP IF START CODE GOOD
SPACE
* BRING IN TRANSIENT TO ANALYZE THE NON-ZERO MLTA START CODE
SPACE
SVC 0 ##### TRANSIENT CALL #####
DC AL1(CCPRIB) CCP SVC RIB.

```

```

CM 00600000
CM 00610000
CM 00620000
CM 00630000
CM 00640000
CM 00650000
00660000
CM 00670000
00680000
CM 00690000
CM 00700000
CM 00710000
00720000
CM 00730000
00740000
CM 00750000
CM 00760000
00770000
CM 00780000
CM 00790000
CM 00800000
CM 00810000
00820000
CM 00830000
CM 00840000
00850000
CM 00860000
00870000
CM 00880000
CM 00890000
CM 00900000
00910000
CM 00920000
00930000
CM 00940000
00950000
CM 00960000
CM 00970000
CM 00980000
00990000
01000000
CM 01010000
01020000
01030000
CM 01040000
CM 01050000
01060000
01070000
01080000
01090000
01100000
CM 01110000
01120000
CM 01130000
CM 01140000
01150000
CM 01160000
01170000
CM 01180000
CM 01190000

```

```

DC AL1(CC4SC) TRANSIENT ID FOR START CODE CM 01200000
SPACE 01210000
CMON2 EQU * * LOCAL. 01220000
LA #,XR2 RESTORE XR2 CM 01230000
CMONOF B # RETURN CM 01240000
.S5800 ANOP 01250000
TITLE '$E094/CMCRID---ADD-CARRIAGE-RETURN/IDLES' 01260000
*****M 01270000
* M 01280000
* TITLE: CMCRID - 'ADD MLTA CARRAIGE RETURN AND IDLES.' M 01290000
* M 01300000
* FUNCTION: MOVE IN A CARRAIGE RETURN AND IDLE CHARACTERS NEEDED M 01310000
* TO ALLOW THE CARRIAGE TIME TO RETURN TO HOME POSITION. M 01320000
* M 01330000
* ENTRY POINT: CMCRID. M 01340000
* M 01350000
* INPUT:XR1 -> TARGET ADDRESS. (LEFTMOST DATA ADDRESS). M 01360000
* XR2 -> DTF(LCB). M 01370000
* M 01380000
* OUTPUT: CARRAGE RETURN AND IDLES ADDED TO OUTPUT DATA STREAM. M 01390000
* M 01400000
* EXTERNAL REFERENCES: N/A. M 01410000
* M 01420000
* EXITS-NORMAL: ALWAYS RETURN TO NSI OF CALLER. M 01430000
* M 01440000
*****M 01450000
SPACE 3 01460000
CMCRID EQU * * < ENTRY POINT > M 01470000
ST CMCRIZ+3,ARR STORE RETURN ADDR IN BRANCH. M 01480000
MVI 0(,XR1),CMCR MOVE CR CHAR TO LINE BUFR. M 01490000
AIF (&NBFR).R5300 01500000
TBN $MDTFR(,XR2),$MTBFR IS IT BUFFERED RECEIVE ? BM 01510000
JT CMCRIZ JUMP IF YES - NO IDLES NEEDED BM 01520000
.R5300 ANOP 01530000
MVI CMTYPL-1(,XR1),CMIDLE MOVE IDLE CHAR TO LINE BUFR. M 01540000
MVC CMTYPL-2(CMTYPL-2,XR1),CMTYPL-1(,XR1) PROPOGATE IDLE. M 01550000
CMCRIZ B # RETURN. M 01560000
AIF (&N1050).C5300 01570000
TITLE '$E094/CMMCT----MULTI-COMPONENT-ADDRESS-ANALYSIS' 01580000
*****1M 01590000
* 1M 01600000
* TITLE: CMMCT - 'MULTI-COMPONENT TERMINAL ADDRESS ANALYSIS' 1M 01610000
* 1M 01620000
* FUNCTION: 1M 01630000
* 1M 01640000
* THIS ROUTINE PERFORMS A COMPONENT ADDRESS ANALYSIS FOR A 1050 1M 01650000
* TERMINAL. IT WILL MOVE TO THE TERMINAL ADDRESS FIELD IN THE 1M 01660000
* MLTA DTF THE COMPONENT ADDRESS SPECIFIED IN THE PARAMETER LIST1M 01670000
* 1M 01680000
* . A CHECK IS MADE TO ASSURE THE TERMINAL IS A 1050 1M 01690000
* 1M 01700000
* . THE COMPONENT INDEX IN THE PARAMETER LIST IS USED TO GET THE1M 01710000
* COMPONENT ADDRESS FROM A TABLE OF 1050 COMPONENT ADDRESSES. 1M 01720000
* 1M 01730000
* . THE COMPONENT ADDRESS IS MOVED TO THE MLTA DTF. 1M 01740000
* 1M 01750000
* ENTRY POINT: CMMCT. 1M 01760000
* 1M 01770000
* INPUT: XR1 -> PL. XR2 -> DTF(LCB). 1M 01780000
* 1M 01790000

```

```

* OUTPUT: THE MLTA DTF FIELD $MDTMA HAS THE CORRECT COMPONENT ADDR. 1M 01800000
* 1M 01810000
* EXTERNAL REFERENCES: NONE. 1M 01820000
* 1M 01830000
* EXTIS, NORMAL: ALWAYS TO THE NSI OF THE CALLER. 1M 01840000
* 1M 01850000
* TABLES/WORKAREA: MCTABL - TABLE OF 1050 COMPONENT ADDRESSES. 1M 01860000
* 1M 01870000
*****1M 01880000
EJECT 01890000
CMMCT EQU * * < ENTRY POINT > 1M 01900000
ST CMMCTR+3,ARR SAVE THE ARR. 1M 01910000
ST CMMCT1+3,XR1 SAVE XR1. 1M 01920000
SPACE 01930000
* DETERMINE IF THIS IS MCT 1M 01940000
SPACE 01950000
CLI PL$MCT(,XR1),NOBIT IS IT MCT ? 1M 01960000
JE CMMCTR JUMP IF NOT MCT. 1M 01970000
SPACE 01980000
* HAVE MCT WITH INDICES IN PL$OPC. 1M 01990000
* DETERMINE WHETHER TO PLUG INPUT OR OUTPUT COMPONENT. 1M 02000000
SPACE 02010000
MNN CMMCTV+3,PL$MCT(,XR1) MOVE OUTPUT COMPONENT INDEX. 1M 02020000
TBN PL$OPM(,XR1),OPPUT IS THIS WRITE OPERATION ? 1M 02030000
JT CMMCTM JUMP IF WRITE. 1M 02040000
SPACE 02050000
* HAVE READ OR SET SKIP BIT SO MOVE INPUT COMPONENT. 1M 02060000
SPACE 02070000
MNZ CMMCTV+3,PL$MCT(,XR1) MOVE INPUT COMPONENT INDEX. 1M 02080000
SPACE 02090000
* GET THE COMPONENT ADDRESS FROM THE TABLE BY USING THE INDEX AS A 1M 02100000
* DISPLACEMNT AND PLUG THE ADDR INTO THE DTF. 1M 02110000
SPACE 02120000
CMMCTM EQU * * LOCAL. 02130000
LA MCTABL-1,XR1 POINT XR1 AT MCT TABLE. 1M 02140000
CMMCTV MVC $MDTMA(1,XR2),#(,XR1) PLUG 2ND TMA BYTE WITH COMP @.1M 02150000
SPACE 02160000
CMMCT1 LA #,XR1 RESTORE XR1. 1M 02170000
CMMCTR EQU * * LOCAL. 1M 02180000
B # RETURN 1M 02190000
.C5300 ANOP 02200000
.M2000 ANOP 02210000
MEND 02220000

```

```
*****
*
*           C C P   G E N E R A T I O N   C O M P L E T E
*
*****
*
*   TO VERIFY THAT YOUR CCP SYSTEM HAS BEEN CORRECTLY
*   GENERATED, YOU SHOULD NOW DO THE FOLLOWING --
*
*   1. FAMILIARIZE YOURSELF WITH THE PROCEDURES FOR
*   PERFORMING INSTALLATION VERIFICATION. THESE
*   PROCEDURES ARE CONTAINED IN THE CCP SYSTEM
*   REFERENCE MANUAL, GC21-7579.
*
*   2. AS THE PROGRAM FOR PERFORMING INSTALLATION
*   VERIFICATION, USE THE SUPPLIED PROGRAM
*   'CCPIVP'.
*
*   3. PERFORM AN ASSIGNMENT BUILD RUN TO CREATE
*   AN ASSIGNMENT SET SUFFICIENT FOR THE VERIFI-
*   CATION ACTIVITY. USE THE SAMPLE ASSIGNMENT
*   SET PROVIDED FOR THIS PURPOSE -- SOURCE LIB-
*   RARY MEMBER -- $CGSST.
*
*   4. START UP CCP BY ENTERING THE FOLLOWING OCL:
*       // LOAD $CCP,R1
*       // FILE NAME-CGIVFILE,UNIT-   ,PACK-   ,RETAIN-T,
*       //   TRACKS-1
*       // RUN
*
*   5. WHEN EXECUTION BEGINS, REQUEST THE PROGRAM
*   'CCPIVP' FROM THE CONSOLE.
*
*   6. COMPARE THE RESULTS OF THE OPERATION OF THIS
*   PROGRAM WITH THOSE SPECIFIED IN THE CCP
*   SYSTEM REFERENCE MANUAL.
*
*   NOTE:
*   -----
*   IF YOUR SYSTEM HAS A PUNCH DEVICE, THE SAMPLE ASSIGNMENT
*   SET ($CGSET) WITH REQUIRED OCL STATEMENTS MAY BE PRINTED
*   AND PUNCHED BY ENTERING THE FOLLOWING OCL STATEMENTS:
*
*       // LOAD $MAINT,DSUNIT
*       // RUN
*       // COPY FROM-DIUNIT,TO-PRTPCH,LIBRARY-S,NAME-$CGSET
*       // END
*
*   REFERENCE THE COMMUNICATION CONTROL PROGRAM SYSTEM REFERENCE
*   MANUAL FOR A DESCRIPTION OF THE SAMPLE ASSIGNMENT SETS.
*
*****
```

```

MODULE-$CGSST, VOLUME ID-R2R2R2, DATE-06/06/10
***** SAMPLE ASSIGNMENT SET CONTROL STATEMENTS *****
*
***** THE FOLLOWING STATEMENTS CAN BE MODIFIED FOR YOUR
***** CONFIGURATION BUT SOME MUST BE KEPT TO RUN CCPIVP.
*
*
// SET ID-@,ACTION-CREATE,DFLTEXEC-YES
// SYSTEM MINUPA-22K,MINTPBUF-2840,
// PASSWORD-FECD,
// COMMANDL-50,DFFPACK-PROGRAM,PGMREQ-15
*
// TERMATTR ATTRID-1,TRANSLAT-NO,BLKL-512,DATAFORM-MESSAGE,
// VERIFYID-NO,DF3270-YES
*
***** THIS STMT TYPE REQD FOR CCPIVP(OR MLTALINE STMT)
// BSCALINE TYPE-CS,LINENUM-1,POLLIST-'00,01,10,11'
// BSCATERM TERMID-00,TYPE-3277M2,ATTRID-1,COMMAND-YES,OFFACTN-HOLD,
// ADDRCHAR-*60604040*,POLLCHAR-*40404040*
// BSCATERM TERMID-01,TYPE-3277M2,ATTRID-1,COMMAND-YES,OFFACTN-HOLD,
// ADDRCHAR-*6060C1C1*,POLLCHAR-*4040C1C1*
// BSCATERM TERMID-10,TYPE-3277M2,ATTRID-1,COMMAND-NO,
// ADDRCHAR-*61614040*,POLLCHAR-*C1C14040*
// BSCATERM TERMID-11,TYPE-3277M2,ATTRID-1,COMMAND-NO,
// ADDRCHAR-*6161C1C1*,POLLCHAR-*C1C1C1C1*
*
// TERMNAME NAME-CU0DV0,TERMID-00
// TERMNAME NAME-CU0DV1,TERMID-01
// TERMNAME NAME-CU1DV0,TERMID-10
// TERMNAME NAME-CU1DV1,TERMID-11
*
***** THIS STMT TYPE REQD FOR CCPIVP
// DISKFILE NAME-CGIVFILE,ORG-C,RECL-16
*
***** THIS STMT NECESSARY FOR CCPIVP
// PROGRAM NAME-CCPIVP,PGMDATA-YES,PRINTER-SHR,
// FILES-'CGIVFILE/CO/NOSHR',
// PACK-PROGRAM

```

```

00010000
00020000
00030000
00040000
00050000
00060000
00070000
00080000
00090000
00100000
00110000
00120000
00130000
00140000
00150000
00160000
00170000
00180000
00190000
00200000
00210000
00220000
00230000
00240000
00250000
00260000
00270000
00280000
00290000
00300000
00310000
00320000
00330000
00340000
00350000
00360000
00370000

```



```

MODULE-$CG1G1, VOLUME ID-R2R2R2, DATE-06/06/10
***** 00000
*          CARDLESS CCP GENERATION INSTRUCTIONS * 00010
***** 00020
*          * 00030
* THE CARDLESS CCP GENERATION PROCEDURE MAKES THE FOLLOWING ASSUMPTIONS: * 00040
*          * 00050
* . THAT DSM RESIDES ON F1 * 00060
*          * 00070
* . THAT CCP DISTRIBUTION PACK IS ON R1 * 00080
*          * 00090
* . THAT THE LAST 20 TRACKS ON THE CCP DISTRIBUTION PACK ON R1 ARE * 00100
* FREE FOR THE $SOURCE FILE USED BY THE MACRO PROCESSOR * 00110
*          * 00120
* . THAT THE $E MACRO STATEMENTS WILL BE READ FROM THE SOURCE * 00130
* LIBRARY MEMBER, $CG1GM, ON THE CCP DISTRIBUTION PACK * 00140
*          * 00150
* IF ANY OF THE ABOVE ASSUMPTIONS ARE NOT TRUE, USE THE MODIFY FUNCTION * 00160
* OF $MAINT TO CHANGE THE SUPPLIED GENERATION PROCEDURES TO REFLECT THE * 00170
* REQUIREMENTS OF YOUR SYSTEM. * 00180
*          * 00190
* SUPPLIED GENERATION PROCEDURES: * 00200
*          * 00210
* $CG1G1 *** CARDLESS GENERATION CONTROL * 00220
* $CG1G3 *** PROCESS USER SPECIFICATIONS * 00240
* $CG1G4 *** PRINT RESULTS OF GENERATION SPECIFICATIONS * 00250
* $CG1G5 *** CREATE SOURCE AND PROCEDURE MEMBERS * 00260
*          * 00270
* YOU MAY MAKE FOUR KINDS OF MODIFICATIONS TO THE SUPPLIED PROCEDURES: * 00280
*          * 00290
* 1. IN THOSE STATEMENTS MARKED ++ IN POSITION 74-75 YOU MAY CHANGE * 00300
* THE DSM PACK LOCATION FROM F1 TO R1. * 00310
*          * 00320
* 2. IN THOSE STATEMENTS MARKED ## IN POSITION 72-73 YOU MAY CHANGE * 00330
* THE CCP DISTRIBUTION PACK LOCATION FROM R1 TO F1, R2 OR F2. * 00340
*          * 00350
* 3. IN THOSE STATEMENTS MARKED $$ IN POSITION 74-75 YOU MAY CHANGE * 00360
* THE $SOURCE FILE LOCATION FROM R1 TO F1, R2 OR F2. * 00370
*          * 00380
* 4. THE '// COMPILER' STATEMENT MAY BE REMOVED FROM PROCEDURE $CG1G3 * 00390
* TO SPECIFY THAT THE MACRO PROCESSOR WILL READ THE $E MACRO * 00400
* STATEMENTS FROM THE SYSTEM INPUT DEVICE. * 00410
* -- STATEMENT 03070 -- * 00420
*          * 00430
* NOTE -- IF A SOURCE MEMBER IS USED IT MUST BE PREPARED PIOR TO * 00440
* CALLING THE PROCEDURE $CG1G1. * 00450
*          * 00460
* WHEN YOU HAVE MADE THESE CHANGES; * 00470
* CALL PROCEDURE $CG1G1 BY ENTERING THE FOLLOWING OCL FROM THE * 00480
* SYSTEM INPUT DEVICE. * 00490
*          * 00500
* // CALL $CG1G1,UNIT -- CCP DISTRIBUTION PACK -- * 00510
* // RUN * 00520
*          * 00530
***** END OF INSTRUCTIONS ***** END OF INSTRUCTIONS ***** 00540

```

MODULE-\$NCIO , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* 00030000
.* TITLE: $NCIO -- PERFORM COMMUNICATIONS I/O UNDER CCP 00040000
.* 00050000
.* FUNCTION: 00060000
.* 00070000
.* . SET THE CONTENTS OF A CCP COMMUNICATIONS PARAMETER LIST 00080000
.* 00090000
.* . BRANCH TO THE CCP COMMUNICATIONS I/O FUNCTION VIA 00100000
.* GENERAL ENTRY 00110000
.* 00120000
.* INPUT: 00130000
.* 00140000
.* . OPTIONAL LABEL WHICH, IF PRESENT, IS EQUATED TO THE ADDRESS 00150000
.* OF THE FIRST INSTRUCTION GENERATED BY THIS MACRO 00160000
.* 00170000
.* . OPERANDS, ALL OPTIONAL, AS INDICATED BELOW: 00180000
.* 00190000
.* PLIST-(1)/(2)/ADDRX/DISPX(REGX)/(ADDRX)/(DISPX(REGX)) 00200000
.* 00210000
.* ADDRESS OF THE LEFTMOST BYTE OF THE COMMUNICATIONS 00220000
.* PARAMETER LIST TO BE USED 00230000
.* 00240000
.* OP-CODE/'CODE,MOD,...'/VALUEX/(1)/(ADDRX)/(DISPX(REGX)) 00250000
.* 00260000
.* OPERATION CODE/MODIFIERS FOR THE COMMUNICATIONS FUNCTION* 00270000
.* TO BE PERFORMED * 00280000
.* 00290000
.* OUTLEN-VALUEX/(1)/(ADDRX)/(DISPX(REGX)) * 00300000
.* 00310000
.* DATA LENGTH FOR AN OUTPUT OPERATION (MUTUALLY EXCLUSIVE * 00320000
.* WITH 'ATTRID' OPERAND) * 00330000
.* 00340000
.* ATTRID-VALUEX/(1)/(ADDRX)/(DISPX(REGX)) * 00350000
.* 00360000
.* IDENTIFIER OF AN ATTRIBUTE-SET WHOSE CONTENTS ARE TO BE * 00370000
.* SET FOR THE TERMINAL BEING ACQUIRED (MUTUALLY EXCLUSIVE * 00380000
.* WITH 'OUTLEN' OPERAND) * 00390000
.* 00400000
.* INLEN-VALUEX/(1)/(ADDRX)/(DISPX(REGX)) * 00410000
.* 00420000
.* MAXIMUM LENGTH FOR AN INPUT OPERATION * 00430000
.* 00440000
.* RECA-ADDRX/(1)/(ADDRX)/(DISPX(REGX)) * 00450000
.* 00460000
.* ADDRESS OF THE RECORD AREA TO BE USED IN THIS * 00470000
.* OPERATION * 00480000
.* 00490000
.* TNAME-CHARS/(ADDRX)/(DISPX(REGX)) $ 00500000
.* 00510000
.* SYMBOLIC NAME OF THE TERMINAL TO BE SELECTED BY THIS * 00520000
.* OPERATION * 00530000
.* 00540000
.* EXEC-YES/Y/NO/N * 00550000
.* 00560000
.* YES -- THE DEFAULT -- SPECIFIES THAT A BRANCH IS TO BE * 00570000
.* MADE TO PERFORM THE COMMUNICATIONS FUNCTION SPECIFIED. * 00580000
.* NO -- THE ALTERNATE -- SPECIFIES THAT THE PARAMETER LIST* 00590000
```

```

.*      (AND POSSIBLY TERMINAL NAME) ARE TO BE SET AS SPECIFIED * 00600000
.*      ABOVE, BUT A BRANCH IS *NOT* TO BE MADE TO PERFORM THE * 00610000
.*      FUNCTION. * 00620000
.* * 00630000
.* . AN INDICATION (VIA THREE GLOBAL VARIABLES) WHETHER CERTAIN * 00640000
.* REQUIRED SYMBOLS HAVE ALREADY BEEN GENERATED. * 00650000
.* * 00660000
.* . SEE SRL #GC21-7579 -- CCP PROGRAMMING REFERENCE MANUAL -- FOR * 00670000
.* FURTHER DESCRIPTION OF THE INPUT TO THIS MACRO. * 00680000
.* * 00690000
.* OUTPUT: * 00700000
.* * 00710000
.* . REQUIRED SYMBOLS (OF THE FORM $N....) IF NOT PREVIOUSLY * 00720000
.* GENERATED BY THE USER. * 00730000
.* * 00740000
.* . IF A LABEL WAS SPECIFIED IN THE MACRO-INSTRUCTION, A SYMBOL * 00750000
.* EQUATED TO THE FIRST BYTE GENERATED. * 00760000
.* * 00770000
.* . SETTING OF INDEX REGISTER 2 TO THE ADDRESS OF THE PARAMETER * 00780000
.* LIST IF THE 'PLIST' OPERAND WAS SPECIFIED. * 00790000
.* * 00800000
.* . SETTING OF INDEX REGISTER 1 TO THE ADDRESS OF THE RECORD AREA * 00810000
.* IF THE OPERAND 'TNAME' WAS SPECIFIED. * 00820000
.* * 00830000
.* . INSTRUCTIONS TO SET THE PARAMETER LIST FIELDS (AND THE * 00840000
.* TERMINAL NAME FIELD OF THE RECORD AREA) AS SPECIFIED BY THE * 00850000
.* USER IN HIS INPUT. * 00860000
.* * 00870000
.* . CONSTANTS (WHERE NECESSARY) FOR THE VALUES TO BE SET INTO * 00880000
.* THE ABOVE FIELDS. * 00890000
.* * 00900000
.* . A BRANCH TO GENERAL ENTRY WITH AN IN-LINE CCP RIB AND SUB-RIB * 00910000
.* UNLESS THE OPERAND 'EXEC-NO' WAS SPECIFIED. * 00920000
.* * 00930000
.* EXTERNAL REFERENCES: * 00940000
.* * 00950000
.* . NO EXTRNS ARE GENERATED BY THIS MACRO. * 00960000
.* * 00970000
.* . A REFERENCE TO THE DSM SUPERVISOR'S GENERAL ENTRY POINT IS * 00980000
.* MADE. * 00990000
.* * 01000000
.* . THREE GLOBAL BOOLEAN VARIABLES ARE USED TO CONTROL THE * 01010000
.* GENERATION OF REQUIRED SYMBOLS: * 01020000
.* * 01030000
.* . &$NCOM -- FOR COMMON SYMBOLS * 01040000
.* . &$NPLO -- FOR PARAMETER LIST OFFSET SYMBOLS * 01050000
.* . &$NOPV -- FOR OPERATION CODE/MODIFIER VALUE SYMBOLS * 01060000
.* * 01070000
.* ERROR MESSAGES: * 01080000
.* * 01090000
.* . THE FOLLOWING ERROR MNOTES (SEVERITY 08): * 01100000
.* * 01110000
.* N2001 CONFLICTING OPERANDS--OUTLEN/ATTRID * 01120000
.* * 01130000
.* N3001 INVALID OPERATION CODE SPECIFIED * 01140000
.* * 01150000
.* N3002 INVALID OPERATION MODIFIER SPECIFIED * 01160000
.* * 01170000
.* N3003 PARAMETER MISSING FINAL RIGHT PAREN * 01180000
.* * 01190000

```

```

.*      N3004  PARENTHESESIZED PARAMETER TOO LONG          * 01200000
.*                                                    * 01210000
.******                                                    * 01220000
&LABEL $NCIO &PLIST-(2),&OP-,&OUTLEN-,&ATTRID-,&INLEN-,&RECA-,&TNAME-, X01230000
          &EXEC-YES                                     01240000
.*                                                    01250000
.* GENERATION VARIABLES USED                               01260000
.*                                                    01270000
      GBLB  &$NCOM,&$NPLO,&$NOPV      . =1 IF SPECIFIC SYMBOLS GEN'D 01280000
      LCLA  &N                          . COUNT CONTROL VARIABLE 01290000
      LCLA  &M,&MM                       . FOR LENGTH MANIPULATION 01300000
      LCLB  &$N3,&$N5,&$N7,&$N9,&$N#,&$N$ . CONDITIONAL GEN SWITCHES 01310000
      LCLB  &CODSW                       . ACTUAL OP CODES SWITCH 01320000
      LCLB  &SYM                          . WHETHER SYMBOLS GEN'D HERE 01330000
      LCLC  &LC1,&LC2                     . USED TO CONSTRUCT STRINGS 01340000
      LCLC  &C,&OPC,&OPM1,&OPM2,&OPM3 . FOR CONSTRUCTING OP CODES/MODS 01350000
.*                                                    01360000
.* PARAMETER TRANSFORMATION TABLE                       01370000
.*                                                    01380000
      TABLE &OP                          . OPERATION CODES (WITHOUT MODS) 01390000
GET      TABDF $NCGET                       01400000
PUT      TABDF $NCPUT                       01410000
PTG      TABDF $NCPTG                       01420000
PNW      TABDF $NCPNW                       01430000
INV      TABDF $NCINV                       01440000
ACC      TABDF $NCACC                       01450000
SPI      TABDF $NCSPI                       01460000
ACQ      TABDF $NCACQ                       01470000
REL      TABDF $NCREL                       01480000
GTA      TABDF $NCGTA                       01490000
SHQ      TABDF $NCSHQ                       01500000
CPY      TABDF $NCCPY                       01510000
EAU      TABDF $NCEAU                       01520000
WAT      TABDF $NCWAT                       01525000
.*                                                    01530000
      TABLE &EXEC                          . BRANCH TO PERFORM I/O OR NOT 01540000
YES      TABDF Y                            01550000
NO       TABDF N                            01560000
N        TABDF N                            01570000
          TABDF Y                            01580000
.*                                                    01590000
      TEXT                                    01600000
.*                                                    01610000
.* BEGIN PROCEDURAL PORTION OF MACRO-DEFINITION        01620000
.*                                                    01630000
      SPACE 1                                 01640000
*                ***  MACRO-$NCIO  RELEASE-8  *** 01650000
.*                                                    01660000
.* IF COMMON EQUATES HAVE ALREADY BEEN GENERATED, SKIP TO NEXT 01670000
.* SECTION -- ELSE GENERATE THEM HERE              01680000
.*                                                    01690000
      AIF  (&$NCOM).CI019                  . SKIP IF EQUATES ALREADY GEN'D 01700000
      SPACE 1                                01710000
* COMMON SYMBOLS USED IN CCP COMMUNICATIONS OPERATIONS 01720000
      SPACE 1                                01730000
$NIXR1 EQU 1                                XR1--USED AS WORK REGISTER 01740000
$NIXR2 EQU 2                                XR2--POINTS TO PARAMETER LIST 01750000
$NSENT EQU 4                                DSM SUPERVISOR GENERAL ENTRY 01760000
$NSCCR EQU X'01'                            DSM RIB--CCP OPERATION 01770000
$NSCCS EQU X'00'                            CCP SUB-RIB--COMMUNICATIONS I/O 01780000

```

\$NLPL	EQU	16	LENGTH OF PARAMETER LIST	01790000
\$NLPLF	EQU	2	LENGTH OF A PARAMETER LIST FIELD	01800000
\$NLSTN	EQU	6	LENGTH OF SYMBOLIC TERMINAL NAME	01810000
.	*			01820000
&SYM	SETB	1	. INDICATE SYMBOLS GEN'D HERE	01830000
&\$NCOM	SETB	1	. INDICATE COMMON SYMBOLS GEN'D	01840000
.CI019	ANOP			01850000
.	*			01860000
.	*		. * IF PARAMETER LIST OFFSETS HAVE ALREADY BEEN GENERATED, SKIP TO	01870000
.	*		. * NEXT SECTION -- ELSE GENERATE THEM HERE	01880000
.	*			01890000
	AIF	(&\$NPLO).CI029	. SKIP IF OFFSETS ALREADY GEN'D	01900000
	SPACE	1		01910000
*	OFFSETS OF	COMMUNICATIONS PARAMETER LIST FIELDS		01920000
	SPACE	1		01930000
\$NPRTC	EQU	+1	RETURN CODE FIELD	01940000
\$NPOPC	EQU	+3	OPERATION CODE/MODIFIERS FIELD	01950000
\$NPOUL	EQU	+5	OUTPUT LENGTH FIELD	01960000
\$NPEFL	EQU	+5	EFFECTIVE INPUT LENGTH FIELD	01970000
\$NPATI	EQU	+5	ATTRIBUTES IDENTIFIER FIELD	01980000
\$NPINL	EQU	+7	MAXIMUM INPUT LENGTH FIELD	01990000
\$NPRAA	EQU	+9	RECORD AREA ADDRESS FIELD	02000000
\$NPWKA	EQU	+11	INTERNAL WORK FIELD	02010000
\$NPWKB	EQU	+13	INTERNAL WORK FIELD	02020000
\$NPWKC	EQU	+15	INTERNAL WORK FIELD	02030000
.	*			02040000
&SYM	SETB	1	. INDICATE SYMBOLS GEN'D HERE	02050000
&\$NPLO	SETB	1	. INDICATE SYMBOLS ALREADY GEN'D	02060000
.CI029	ANOP			02070000
.	*			02080000
.	*		. * IF OPERATION CODE VALUES HAVE ALREADY BEEN GENERATED, SKIP TO NEXT	02090000
.	*		. * SECTION -- ELSE GENERATE THEM HERE	02100000
.	*			02110000
	AIF	(&\$NOPV).CI039	. SKIP IF VALUES ALREADY GEN'D	02120000
	SPACE	1		02130000
*	CCP OPERATION CODE VALUES			02140000
	SPACE	1		02150000
\$NCSHQ	EQU	X'0000'	SHUTDOWN INQUIRY	02160000
\$NCGET	EQU	X'0001'	GET	02170000
\$NCPUT	EQU	X'0002'	PUT	02180000
\$NCPTG	EQU	X'0003'	PUT THEN GET	02190000
\$NCACC	EQU	X'0004'	ACCEPT INPUT	02200000
\$NCINV	EQU	X'0005'	INVITE INPUT	02210000
\$NCPNW	EQU	X'0006'	PUT-NO-WAIT	02220000
\$NCGTA	EQU	X'0008'	GET TERMINAL ATTRIBUTES	02230000
\$NCACQ	EQU	X'0009'	ACQUIRE TERMINAL	02240000
\$NCREL	EQU	X'000A'	RELEASE TERMINAL	02250000
\$NCCPY	EQU	X'0042'	COPY (DFF ONLY)	02260000
\$NCEAU	EQU	X'0052'	ERASE ALL UNPROTECTED (DFF ONLY)	02270000
\$NCSPI	EQU	X'0401'	STOP INVITE/GET	02280000
\$NCWAT	EQU	X'0014'	WAIT OPERATION CODE	02285000
	SPACE	1		02290000
*	CCP OPERATION MODIFIER VALUES			02300000
	SPACE	1		02310000
\$NMSTA	EQU	X'0010'	SET TERMINAL ATTRIBUTES BY ID	02320000
\$NMRVI	EQU	X'0010'	SEND REVERSE-INTERRUPT	02330000
\$NMKPL	EQU	X'0010'	KEEP THE LINE	02340000
\$NMBLK	EQU	X'0020'	END THE CURRENT OUTPUT BLOCK	02350000
\$NMMSG	EQU	X'0030'	SEND END-OF-TRANSMISSION	02360000
\$NMPRF	EQU	X'0040'	READ UNDER FORMAT	02365000

```

$NMNEL EQU X'0100' RECORD DOES NOT END CURRENT LINE 02370000
$NMNNL EQU X'0200' RECORD DOES NOT START NEW LINE 02380000
$NMOVR EQU X'0800' OVERRIDE/SELECTED-FIELDS LIST 02390000
.* 02400000
&SYM SETB 1 . INDICATE SYMBOLS GEN'D HERE 02410000
&$NOPV SETB 1 . INDICATE VALUES ALREADY GEN'D 02420000
.CI039 ANOP 02430000
.* 02440000
.*----- L A B E L -----* 02450000
.* 02460000
    AIF (&SYM).CI040 . SKIP IF ANY SYMBOLS GEN'D HERE 02470000
    AGO .CI045 . SKIP IF *NO* SYMBOLS GEN'D 02480000
.CI040 ANOP 02490000
    SPACE 2 02500000
.CI045 ANOP 02510000
.* 02520000
    AIF (T'&LABEL EQ 'O').CI050 . SKIP IF NO LABEL SPECIFIED 02530000
&LABEL EQU * INSTRUCTION LABEL 02540000
.CI050 ANOP 02550000
.* 02560000
.*----- P L I S T -----* 02570000
.* 02580000
.* IF 'PLIST' SPECIFIED AS OTHER THAN '(2)', SET XR2 TO PARAMETER LIST 02590000
.* ADDRESS 02600000
.* 02610000
    AIF (T'&PLIST EQ 'O').CI199 . SKIP TO NEXT SECTION IF ABSENT 02620000
    AIF (&PLIST EQ '(2)').CI199 . SKIP IF ALREADY IN XR2 02630000
    AIF (&PLIST EQ '(1)').CI110 . SKIP IF SPEC'D AS IN XR1 NOW 02640000
    AIF ('&PLIST'(1,1) EQ '(').CI150 . SKIP IF ADDR OF ADDR 02650000
.* 02660000
.* PLIST -- DIRECT ADDRESS 02670000
.* 02680000
    LA &PLIST,$NIXR2 SET XR2--ADDR OF PARAMETER LIST 02690000
    AGO .CI199 . SKIP TO NEXT SECTION 02700000
.* 02710000
.* PLIST -- ADDRESS IN XR1 02720000
.* 02730000
.CI110 ANOP 02740000
    LA 0(,$NIXR1),$NIXR2 SET XR2--ADDR OF PARAMETER LIST 02750000
    AGO .CI199 . SKIP TO NEXT SECTION 02760000
.* 02770000
.* PLIST -- INDIRECT ADDRESS 02780000
.* 02790000
.CI150 ANOP 02800000
&N SETA K'&PLIST . LENGTH OF PARAMETER 02810000
    AIF (&N GT '18').CIETL . SKIP IF PARAMETER TOO LONG 02820000
    AIF ('&PLIST'(&N,1) NE ')).CIERP . ERR IF NO RIGHT PAREN 02830000
.* 02840000
&M SETA &N-2 . LENGTH OF PARAMETER - 2 02850000
&MM SETA 0 . INITIALIZE VARIABLE TO ZERO 02860000
    AIF (&N LE '10').CI155 . SKIP IF LE 8 CHARS IN PARENS 02870000
&MM SETA &N-10 . LENGTH OF PARAMETER - 10 02880000
.CI155 ANOP 02890000
&LC1 SETC '&PLIST'(2,&M) . REMOVE SURROUNDING PARENS 02900000
&LC2 SETC '&PLIST'(10,&MM) . GET REMAINDER IF ANY 02910000
    L &LC1&LC2,$NIXR2 SET XR2--ADDR OF PARAMETER LIST 02920000
.* 02930000
.CI199 ANOP . END OF SECTION--PLIST 02940000
.* 02950000
.*----- O P -----* 02960000

```

```

.* 02970000
.* SET OPERATION CODE AND MODIFIERS INTO BYTES 2-3 OF LIST 02980000
.* 02990000
AIF (T'&OP EQ 'O').CI299 . SKIP IF OPERAND OMITTED 03000000
.* 03010000
AIF (&OP EQ '(1)').CI201 . SKIP IF SPEC'D AS IN XR1 03020000
.* 03030000
AIF ('&OP'(1,1) EQ '(').CI210 . SKIP IF INDIRECT ADDRESS 03040000
.* 03050000
AGO .CI220 . SKIP--EXPRESSION OR CODES 03060000
.* 03070000
.* OP -- IN INDEX REGISTER 1 03080000
.* 03090000
.CI201 ANOP 03100000
ST $NPOPC(,$NIXR2),$NIXR1 SET OPERATION CODE/MODIFIERS 03110000
AGO .CI299 . SKIP TO NEXT SECTION 03120000
.* 03130000
.* OP -- INDIRECT ADDRESS 03140000
.* 03150000
.CI210 ANOP 03160000
&N SETA K'&OP . LENGTH OF PARAMETER 03170000
AIF (&N GT '18').CIETL . SKIP IF PARAMETER TOO LONG 03180000
AIF ('&OP'(&N,1) NE ')').CIERP . ERR IF NO RIGHT PAREN 03190000
.* 03200000
&M SETA &N-2 . LENGTH OF PARAMETER - 2 03210000
&MM SETA 0 . INITIALIZE VARIABLE TO ZERO 03220000
AIF (&N LE '10').CI215 . SKIP IF LE 8 CHARS IN PARENS 03230000
&MM SETA &N-10 . LENGTH OF PARAMETER - 10 03240000
.CI215 ANOP 03250000
&LC1 SETC '&OP'(2,&M) . REMOVE SURROUNDING PARENS 03260000
&LC2 SETC '&OP'(10,&MM) . GET REMAINDER IF ANY 03270000
MVC $NPOPC(2,$NIXR2),&LC1&LC2 SET OPERATION CODE/MODIFIER 03280000
AGO .CI299 . SKIP TO NEXT SECTION 03290000
.* 03300000
.* OP -- DETERMINE WHETHER EXPRESSION OR CODES 03310000
.* 03320000
.CI220 ANOP 03330000
AIF ('&OP'(4,1) EQ ',').CI230 . SKIP IF CODE WITH MODIFIERS 03340000
AGO .CI280 . SKIP IF DEFINITELY EXPRESSION 03350000
.* 03360000
.* OP -- WITH MODIFIER CODES -- DETERMINE IF VALID OP CODE 03370000
.* 03380000
.CI230 ANOP 03390000
&N SETA 1 . INITIALIZE LOOP CONTROL 03400000
&C SETC '&OP'(1,3) . INITIALIZE COMPARAND 03410000
.* 03420000
.CI235 ANOP 03430000
AIF (&C EQ 'GETINVPUTPNWPTGACQRELACCSPIGTASHQ'(&N,3)).CI240 03440000
&N SETA &N+3 . STEP LOOP CONTROL 03450000
AIFB (&N LT '34').CI235 . REITERATE IF NOT EXHAUSTED 03460000
&N SETA 1 . REINITIALIZE LOOP CONTROL 03470000
.CI236 AIF (&C EQ 'CPYEAURUF'(&N,3)).CI240 03480000
&N SETA &N+3 . STEP LOOP CONTROL 03490000
AIFB (&N LT '10').CI236 . REITERATE IF NOT EXHAUSTED 03500000
.* 03510000
.* OP -- INVALID OPERATION CODE 03520000
.* 03530000
MNOTE 08,'N3001 INVALID OPERATION CODE SPECIFIED' 03540000
AGO .CI999 . SKIP TO LEAVE MACRO 03550000
.* 03560000

```

```

.* OP -- VALID OP CODE -- COLLECT ANY MODIFIERS          03570000
.*                                                       03580000
.CI240 ANOP                                             03590000
&CODSW SETB 1 . SIGNAL ACTUAL OP CODE PRESENT          03600000
&OPC  SETC '$NC&C' . SET UP OP CODE SYMBOL            03610000
      AIF (K'&OP EQ '3').CI289 . SKIP IF ONLY OP CODE 03620000
.*                                                       03630000
&C    SETC '&OP'(5,3) . EXTRACT FIRST MODIFIER        03640000
&OPM1 SETC '+$NM&C' . CONSTRUCT 1ST MODIFIER SYMBOL    03650000
      AIF (K'&OP LE '7').CI289 . SKIP IF ONLY 1 MODIFIER 03660000
.*                                                       03670000
&C    SETC '&OP'(9,3) . EXTRACT 2ND MODIFIER          03680000
&OPM2 SETC '+$NM&C' . CONSTRUCT 2ND MODIFIER SYMBOL    03690000
      AIF (K'&OP LE '11').CI289 . SKIP IF ONLY 2 MODIFIERS 03700000
.*                                                       03710000
&C    SETC '&OP'(13,3) . EXTRACT 3RD MODIFIER         03720000
&OPM3 SETC '+$NM&C' . CONSTRUCT 3RD MODIFIER SYMBOL    03730000
      AIF (K'&OP LE '15').CI289 . SKIP IF NO MORE THAN 3 MODIFRS 03740000
.*                                                       03750000
.* OP -- TOO MANY MODIFIERS                             03760000
.*                                                       03770000
      MNOTE 08,'N3002 INVALID OPERATION MODIFIER SPECIFIED' * 03780000
      AGO .CI999 . SKIP TO END MACRO                   03790000
.*                                                       03800000
.* OP -- EXPRESSION SPECIFIED                           03810000
.*                                                       03820000
.CI280 ANOP                                             03830000
.CI289 ANOP                                             03840000
&$N3  SETB 1 . SIGNAL CONSTANT TO BE GEN'D           03850000
&$N$  SETB 1 . SIGNAL JUMP TO GE GEN'D                03860000
      MVC $NPOPC(2,$NIXR2),$N3&SYSNDX SET OPERATION CODE/MODIFIER 03870000
.*                                                       03880000
.CI299 ANOP . END OF OPERATION SECTION                 03890000
.*                                                       03900000
.*----- O U T L E N -----*                          03910000
.*                                                       03920000
.*                                                       03930000
.* SET OUTPUT LENGTH INTO BYTES 4-5 OF PARAMETER LIST 03940000
.*                                                       03950000
      AIF (T'&OUTLEN EQ 'O').CI399 . SKIP IF NO OUTPUT LENGTH 03960000
      AIF (&OUTLEN EQ '(1)').CI310 . SKIP IF LENGTH IN XR1     03970000
      AIF ('&OUTLEN'(1,1) EQ '(').CI350 . SKIP IF INDIRECT ADDRESS 03980000
.*                                                       03990000
.* OUTLEN -- ACTUAL VALUE                                04000000
.*                                                       04010000
&$N5  SETB 1 . SIGNAL CONSTANT TO BE GEN'D           04020000
&$N$  SETB 1 . SIGNAL JUMP TO BE GEN'D                04030000
      MVC $NPOUL(2,$NIXR2),$N5&SYSNDX SET OUTPUT DATA LENGTH 04040000
      AGO .CI399 . SKIP TO NEXT SECTION                 04050000
.*                                                       04060000
.* OUTLEN -- IN INDEX REGISTER 1                       04070000
.*                                                       04080000
.CI310 ANOP                                             04090000
      ST $NPOUL(,$NIXR2),$NIXR1 SET OUTPUT DATA LENGTH 04100000
      AGO .CI399 . SKIP TO NEXT SECTION                 04110000
.*                                                       04120000
.* OUTLEN -- INDIRECT ADDRESS                           04130000
.*                                                       04140000
.CI350 ANOP                                             04150000
&N    SETA K'&OUTLEN . LENGTH OF PARAMETER            04160000

```



```

AIF (&N GT '18').CIETL . SKIP IF PARAMETER TOO LONG 04170000
AIF ('&OUTLEN'(&N,1) NE ' ').CIERP . ERR IF NO RIGHT PAREN 04180000
.* 04190000
&M SETA &N-2 . LENGTH OF PARAMETER - 2 04200000
&MM SETA 0 . INITIALIZE VARIABLE TO ZERO 04210000
AIF (&N LE '10').CI355 . SKIP IF LE 8 CHARS IN PARENS 04220000
&MM SETA &N-10 . LENGTH OF PARAMETER - 10 04230000
.CI355 ANOP 04240000
&LC1 SETC '&OUTLEN'(2,&M) . REMOVE SURROUNDING PARENS 04250000
&LC2 SETC '&OUTLEN'(10,&MM) . GET REMAINDER IF ANY 04260000
MVC $NPOUL(2,$NIXR2),&LC1&LC2 SET OUTPUT DATA LENGTH 04270000
.* 04280000
.CI399 ANOP . END OF SECTION 04290000
.* 04300000
.*----- A T T R I D -----* 04310000
.* 04320000
.* SET ATTRIBUTES IDENTIFIER INTO BYTES 4-5 OF PARAMETER LIST 04330000
.* 04340000
AIF (T'&ATTRID EQ '0').CI499 . SKIP IF NO ATTRIBUTES ID 04350000
.* 04360000
AIF (T'&OUTLEN EQ '0').CI401 . SKIP IF ATTRID--NO OUTLEN 04370000
.* 04380000
.* ATTRID -- ERROR -- SPECIFIED ALONG WITH OUTLEN 04390000
.* 04400000
MNOTE 08,'N2001 CONFLICTING OPERANDS--OUTLEN/ATTRID' 04410000
AGO .CI999 . SKIP TO END MACRO 04420000
.* 04430000
.CI401 ANOP 04440000
AIF (&ATTRID EQ '(1)').CI410 . SKIP IF ATTRID IN XR1 04450000
AIF ('&ATTRID'(1,1) EQ ' ').CI450 . SKIP IF INDIRECT ADDRESS 04460000
.* 04470000
.* ATTRID -- ACTUAL VALUE 04480000
.* 04490000
&$N5 SETB 1 . SIGNAL CONSTANT TO BE GEN'D 04500000
&$N$ SETB 1 . SIGNAL JUMP TO BE GEN'D 04510000
MVC $NPATI(2,$NIXR2),$N5&SYSNDX SET ATTRIBUTES IDENTIFIER 04520000
AGO .CI499 . SKIP TO NEXT SECTION 04530000
.* 04540000
.* ATTRID -- IN INDEX REGISTER 1 04550000
.* 04560000
.CI410 ANOP 04570000
ST $NPATI(,$NIXR2),$NIXR1 SET ATTRIBUTES IDENTIFIER 04580000
AGO .CI499 . SKIP TO NEXT SECTION 04590000
.* 04600000
.* ATTRID -- INDIRECT ADDRESS 04610000
.* 04620000
.CI450 ANOP 04630000
&N SETA K'&ATTRID . LENGTH OF PARAMETER 04640000
AIF (&N GT '18').CIETL . SKIP IF PARAMETER TOO LONG 04650000
AIF ('&ATTRID'(&N,1) NE ' ').CIERP . ERR IF NO RIGHT PAREN 04660000
.* 04670000
&M SETA &N-2 . LENGTH OF PARAMETER - 2 04680000
&MM SETA 0 . INITIALIZE VARIABLE TO ZERO 04690000
AIF (&N LE '10').CI455 . SKIP IF LE 8 CHARS IN PARENS 04700000
&MM SETA &N-10 . LENGTH OF PARAMETER - 10 04710000
.CI455 ANOP 04720000
&LC1 SETC '&ATTRID'(2,&M) . REMOVE SURROUNDING PARENS 04730000
&LC2 SETC '&ATTRID'(10,&MM) . GET REMAINDER IF ANY 04740000
MVC $NPATI(2,$NIXR2),&LC1&LC2 SET ATTRIBUTES IDENTIFIER 04750000
.* 04760000

```

```

.CI499 ANOP          . END OF SECTION          04770000
.*                  04780000
.*----- I N L E N -----*                  04790000
.*                  04800000
.* SET INPUT LENGTH IN BYTES 6-7 OF PARAMETER LIST 04810000
.*                  04820000
    AIF  (T'&INLEN EQ '0').CI599  . SKIP IF NO INPUT LENGTH 04830000
    AIF  (&INLEN EQ '(1)').CI510  . SKIP IF INLEN IN XR1 04840000
    AIF  ('&INLEN'(1,1) EQ '(').CI550 . SKIP IF INDIRECT ADDRESS 04850000
.*                  04860000
.* INLEN -- ACTUAL VALUE SPECIFIED              04870000
.*                  04880000
&$N7  SETB  1          . SIGNAL CONSTANT TO BE GEN'D 04890000
&$N$  SETB  1          . SIGNAL JUMP TO BE GEN'D 04900000
    MVC  $NPINL(2,$NIXR2),$N7&SYSNDX SET MAXIMUM INPUT DATA LENGTH 04910000
    AGO  .CI599        . SKIP TO NEXT SECTION 04920000
.*                  04930000
.* INLEN -- IN INDEX REGISTER 1                04940000
.*                  04950000
.CI510 ANOP          04960000
    ST   $NPINL(,$NIXR2),$NIXR1  SET MAXIMUM INPUT DATA LENGTH 04970000
    AGO  .CI599        . SKIP TO NEXT SECTION 04980000
.*                  04990000
.* INLEN -- INDIRECT ADDRESS                  05000000
.*                  05010000
.CI550 ANOP          05020000
&N    SETA  K'&INLEN      . LENGTH OF PARAMETER 05030000
    AIF  (&N GT '18').CIETL    . SKIP IF PARAMETER TOO LONG 05040000
    AIF  ('&INLEN'(&N,1) NE ' ').CIERP . ERR IF NO RIGHT PAREN 05050000
.*                  05060000
&M    SETA  &N-2         . LENGTH OF PARAMETER - 2 05070000
&MM   SETA  0            . INITIALIZE VARIABLE TO ZERO 05080000
    AIF  (&N LE '10').CI555    . SKIP IF LE 8 CHARS IN PARENS 05090000
&MM   SETA  &N-10       . LENGTH OF PARAMETER - 10 05100000
.CI555 ANOP          05110000
&LC1  SETC  '&INLEN'(2,&M)    . REMOVE SURROUNDING PARENS 05120000
&LC2  SETC  '&INLEN'(10,&MM)  . GET REMAINDER IF ANY 05130000
    MVC  $NPINL(2,$NIXR2),&LC1&LC2 SET MAXIMUM INPUT DATA LENGTH 05140000
.*                  05150000
.CI599 ANOP          . END OF SECTION          05160000
.*                  05170000
.*----- R E C A -----*                  05180000
.*                  05190000
.* SET RECORD AREA ADDRESS IN BYTES 8-9 OF PARAMETER LIST 05200000
.*                  05210000
    AIF  (T'&RECA EQ '0').CI699  . SKIP IF NO RECORD ADDRESS 05220000
    AIF  (&RECA EQ '(1)').CI610  . SKIP IF REC ADDR IN XR1 05230000
    AIF  ('&RECA'(1,1) EQ '(').CI650 . SKIP IF INDIRECT ADDRESS 05240000
.*                  05250000
.* RECA -- ACTUAL ADDRESS SPECIFIED            05260000
.*                  05270000
    AIF  (T'&TNAME NE '0').CI605 . SKIP IF TNAME TO BE SET 05280000
.*                  05290000
.* RECA -- ACTUAL ADDRESS / *NO* TNAME        05300000
.*                  05310000
&$N9  SETB  1          . SIGNAL CONSTANT TO BE GEN'D 05320000
&$N$  SETB  1          . SIGNAL JUMP TO BE GEN'D 05330000
    MVC  $NPRAA(2,$NIXR2),$N9&SYSNDX SET RECORD AREA ADDRESS 05340000
    AGO  .CI699        . SKIP TO NEXT SECTION 05350000
.*                  05360000

```

```

.* RECA -- ACTUAL ADDRESS / *TNAME* TO BE SET          05370000
.*                                                    05380000
.CI605 ANOP                                           05390000
    LA    &RECA,$NIXR1          SET XR1--ADDR OF RECORD AREA 05400000
.*                                                    05410000
.* RECA -- IN INDEX REGISTER 1                        05420000
.*                                                    05430000
.CI610 ANOP                                           05440000
    ST    $NPRAA(,$NIXR2),$NIXR1 SET RECORD AREA ADDRESS 05450000
    AGO   .CI699                . SKIP TO NEXT SECTION      05460000
.*                                                    05470000
.* RECA -- INDIRECT ADDRESS                          05480000
.*                                                    05490000
.CI650 ANOP                                           05500000
&N      SETA  K'&RECA                . LENGTH OF PARAMETER 05510000
    AIF   (&N GT '18').CIETL        . SKIP IF PARAMETER TOO LONG 05520000
    AIF   ('&RECA'(&N,1) NE ' ').CIERP . ERR IF NO RIGHT PAREN 05530000
.*                                                    05540000
&M      SETA  &N-2                    . LENGTH OF PARAMETER - 2 05550000
&MM     SETA  0                        . INITIALIZE VARIABLE TO ZERO 05560000
    AIF   (&N LE '10').CI655        . SKIP IF LE 8 CHARS IN PARENS 05570000
&MM     SETA  &N-10                    . LENGTH OF PARAMETER - 10 05580000
.CI655 ANOP                                           05590000
&LC1    SETC  '&RECA'(2,&M)          . REMOVE SURROUNDING PARENS 05600000
&LC2    SETC  '&RECA'(10,&MM)        . GET REMAINDER IF ANY      05610000
.*                                                    05620000
    AIF   (T'&TNAME NE 'O').CI670    . SKIP IF TNAME PRESENT 05630000
.*                                                    05640000
.* RECA -- INDIRECT ADDRESS / *NO* TNAME             05650000
.*                                                    05660000
    MVC   $NPRAA(2,$NIXR2),&LC1&LC2 SET RECORD AREA ADDRESS 05670000
    AGO   .CI699                . SKIP TO NEXT SECTION      05680000
.*                                                    05690000
.* RECA -- INDIRECT ADDRESS -- *TNAME* TO BE SET    05700000
.*                                                    05710000
.CI670 ANOP                                           05720000
    L     &LC1&LC2,$NIXR1          SET XR1--ADDR OF RECORD AREA 05730000
    ST    $NPRAA(,$NIXR2),$NIXR1    SET RECORD AREA ADDRESS 05740000
.*                                                    05750000
.CI699 ANOP                                           05760000
    . END OF SECTION
.*                                                    05770000
.*----- T N A M E -----*                          05780000
.*                                                    05790000
.* SET TERMINAL NAME INTO BYTES 0-5 OF THE RECORD AREA 05800000
.*                                                    05810000
    AIF   (T'&TNAME EQ 'O').CI799    . SKIP IF NO TERMINAL NAME 05820000
    AIF   (T'&RECA NE 'O').CI701    . SKIP IF RECA *WAS* SPECIFIED 05830000
.*                                                    05840000
.* TNAME -- RECA WAS *NOT* SPECIFIED -- MUST LOAD FROM PARAMETER LIST 05850000
.*                                                    05860000
    L     $NPRAA(,$NIXR2),$NIXR1    SET XR1--ADDR OF RECORD AREA 05870000
.*                                                    05880000
.CI701 ANOP                                           05890000
    AIF   ('&TNAME'(1,1) EQ '(').CI750 . SKIP IF ADDRESS OF NAME 05900000
.*                                                    05910000
.* TNAME -- ACTUAL NAME SPECIFIED                    05920000
.*                                                    05930000
&$N#    SETB  1                        . SIGNAL CONSTANT TO BE GEN'D 05940000
&$N$    SETB  1                        SIGNAL JUMP TO BE GEN'D 05950000
    MVC   $NLSTN-1($NLSTN,$NIXR1),$N#&SYSNDX SET TERMINAL NAME 05960000

```

```

AGO .CI799 . SKIP TO NEXT SECTION 05970000
.* 05980000
.* TNAME -- ADDRESS OF THE NAME SPECIFIED 05990000
.* 06000000
.CI750 ANOP 06010000
&N SETA K'&TNAME . LENGTH OF PARAMETER 06020000
AIF (&N GT '18').CIETL . SKIP IF PARAMETER TOO LONG 06030000
AIF ('&TNAME'(&N,1) NE ')).CIERP . ERR IF NO RIGHT PAREN 06040000
.* 06050000
&M SETA &N-2 . LENGTH OF PARAMETER - 2 06060000
&MM SETA 0 . INITIALIZE VARIABLE TO ZERO 06070000
AIF (&N LE '10').CI755 . SKIP IF LE 8 CHARS IN PARENS 06080000
&MM SETA &N-10 . LENGTH OF PARAMETER - 10 06090000
.CI755 ANOP 06100000
&LC1 SETC '&TNAME'(2,&M) . REMOVE SURROUNDING PARENS 06110000
&LC2 SETC '&TNAME'(10,&MM) . GET REMAINDER IF ANY 06120000
MVC $NLSTN-1($NLSTN,$NIXR1),&LC1&LC2 SET TERMINAL NAME 06130000
.* 06140000
.CI799 ANOP . END OF SECTION 06150000
.* 06160000
.*----- C R E A T E A N Y C O N S T A N T S -----* 06170000
.* 06180000
AIF (&$N$).CI801 . SKIP IF ANY CONSTANTS TO GEN 06190000
AGO .CI899 . SKIP IF *NO* CONSTANTS TO GEN 06200000
.* 06210000
.* CONSTANTS -- GENERATE THE JUMP AROUND THEM 06220000
.* 06230000
.CI801 ANOP 06240000
J $N$&SYSNDX JUMP AROUND GENERATED CONSTANTS 06250000
SPACE 1 06260000
.* 06270000
.* CONSTANTS -- FOR OPERATION CODE 06280000
.* 06290000
AIF (&$N3).CI810 . SKIP IF OP CODE TO GEN 06300000
AGO .CI829 . SKIP IF *NO* OP CODE TO GEN 06310000
.* 06320000
.CI810 ANOP 06330000
AIF (&CODSW).CI820 . SKIP IF *ACTUAL* CODES 06340000
.* 06350000
$N3&SYSNDX DC AL2(&OP) OPERATION CODE/MODIFIERS 06360000
AGO .CI829 . SKIP TO CHECK NEXT CONSTANT 06370000
.* 06380000
.CI820 ANOP 06390000
$N3&SYSNDX DC AL2(&OPC&OPM1&OPM2&OPM3) OPERATION CODE/MODIFIERS 06400000
.* 06410000
.CI829 ANOP . END OF SUB-SECTION 06420000
.* 06430000
.* CONSTANTS -- OUTPUT DATA LENGTH /OR/ ATTRIBUTES IDENTIFIER 06440000
.* 06450000
AIF (&$N5).CI830 . SKIP IF CONSTANT TO GENERATE 06460000
AGO .CI839 . SKIP IF *NO* CONSTANT TO GEN 06470000
.* 06480000
.CI830 ANOP 06490000
AIF (T'&OUTLEN EQ 'O').CI835 . SKIP IF IT IS ATTRID TO GEN 06500000
$N5&SYSNDX DC AL2(&OUTLEN) OUTPUT DATA LENGTH 06510000
AGO .CI839 . SKIP TO NEXT SUB-SECTION 06520000
.* 06530000
.CI835 ANOP 06540000
$N5&SYSNDX DC AL2(&ATTRID) ATTRIBUTES IDENTIFIER 06550000
.* 06560000

```

```

.CI839 ANOP . END OF SUB-SECTION 06570000
.* 06580000
.* CONSTANTS -- INPUT DATA LENGTH 06590000
.* 06600000
    AIF (&$N7).CI840 . SKIP IF CONSTANT TO GENERATE 06610000
    AGO .CI849 . SKIP IF *NO* INPUT LEN TO GEN 06620000
.* 06630000
.CI840 ANOP 06640000
$N7&SYSNDX DC AL2(&INLEN) MAXIMUM INPUT DATA LENGTH 06650000
.* 06660000
.CI849 ANOP . END OF SUB-SECTION 06670000
.* 06680000
.* CONSTANTS -- RECORD AREA ADDRESS 06690000
.* 06700000
    AIF (&$N9).CI850 . SKIP IF RECA CONSTANT TO GEN 06710000
    AGO .CI859 . SKIP IF *NO* RECA CONSTANT 06720000
.* 06730000
.CI850 ANOP 06740000
$N9&SYSNDX DC AL2(&RECA) RECORD AREA ADDRESS 06750000
.* 06760000
.CI859 ANOP . END OF SUB-SECTION 06770000
.* 06780000
.* CONSTANTS -- TERMINAL NAME 06790000
.* 06800000
    AIF (&$N#).CI860 . SKIP IF NAME TO BE GEN'D 06810000
    AGO .CI869 . SKIP IF *NO* NAME TO GEN 06820000
.* 06830000
.CI860 ANOP 06840000
$N#&SYSNDX DC CL6 '&TNAME' SYMBOLIC TERMINAL NAME 06850000
.* 06860000
.CI869 ANOP . END OF SUB-SECTION 06870000
.* 06880000
.* GENERATE THE JUMP-TO SYMBOL 06890000
.* 06900000
    SPACE 1 06910000
$N$&SYSNDX EQU * JUMP TO HERE AROUND CONSTANTS 06920000
.* 06930000
.CI899 ANOP . END OF SECTION 06940000
.* 06950000
.* ----- E X E C ----- * 06960000
.* 06970000
.* UNLESS EXEC-NO WAS SPECIFIED, BRANCH TO GENERAL ENTRY WITH PROPER 06980000
.* RIB AND SUB-RIB TO PERFORM THE OPERATION. 06990000
.* 07000000
    AIF (&EXEC EQ 'N').CI999 . SKIP IF EXEC-NO SPECIFIED 07010000
.* 07020000
    SVC 0 CALL GENERAL ENTRY 07030000
    DC AL1($NSCCR) RIB FOR A CCP FUNCTION 07040000
    DC AL1($NSCCS) SUB-RIB FOR COMMUNICATIONS I/O 07050000
.CI909 AGO .CI999 . SKIP TO CONCLUDE MACRO 07060000
.* 07070000
.* SYNTAX ERROR -- EXPRESSION TOO LONG. 07080000
.* 07090000
.CIETL ANOP 07100000
    SPACE 1 07110000
    MNOTE 08, 'N3004 PARENTHESESIZED PARAMETER TOO LONG' 07120000
    SPACE 1 07130000
    AGO .CI999 07140000
.* 07150000
.* SYNTAX ERROR -- MISSING RIGHT PAREN 07160000

```

```
. * 07170000
.CIERP ANOP 07180000
  SPACE 1 07190000
  MNOTE 08, 'N3003 PARAMETER MISSING FINAL RIGHT-PAREN' 07200000
  SPACE 1 07210000
. * 07220000
. * COMMON END OF MACRO 07230000
. * 07240000
.CI999 ANOP 07250000
* *** END GENERATION--$NCIO *** 07260000
  SPACE 1 07270000
  MEND 07280000
```

MODULE-\$NCOM , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* TITLE: $NCOM -- DEFINE CCP COMMON EQUATES * 00030000
.* * 00040000
.* FUNCTION: * 00050000
.* * 00060000
.* . GENERATE SYMBOLS FOR VALUES USED COMMONLY IN CCP MACROS * 00070000
.* * 00080000
.* . PREVENT THE INADVERTANT GENERATION OF DUPLICATE SYMBOLS FOR * 00090000
.* THOSE VALUES * 00100000
.* * 00110000
.* INPUT: * 00120000
.* * 00130000
.* . NO LABEL SHOULD BE SPECIFIED FOR THIS MACRO. * 00140000
.* * 00150000
.* . THIS MACRO HAS NO OPERANDS. * 00160000
.* * 00170000
.* . SEE SRL #GC21-7579 -- CCP PROGRAMMING REFERENCE MANUAL -- FOR * 00180000
.* FURTHER DESCRIPTION OF THIS MACRO. * 00190000
.* * 00200000
.* OUTPUT: * 00210000
.* * 00220000
.* . IF THESE SYMBOLS HAVE BEEN GENERATED EARLIER IN THIS * 00230000
.* GENERATION RUN, ONLY A WARNING MNOTE (SEE ERROR MESSAGES * 00240000
.* BELOW). * 00250000
.* * 00260000
.* . ELSE, SYMBOLS FOR THE FOLLOWING: * 00270000
.* * 00280000
.* . DSM GENERAL ENTRY * 00290000
.* . DSM RIB FOR CCP * 00300000
.* . CCP SUB-RIB FOR COMMUNICATIONS OPERATION * 00310000
.* . INDEX REGISTERS * 00320000
.* . LENGTHS OF PARAMETER LIST, A PARAMETER LIST FIELD, AND * 00330000
.* A SYMBOLIC TERMINAL NAME * 00340000
.* * 00350000
.* . AN INDICATION (VIA THE GLOBAL VARIABLE &$NCOM) THAT THESE * 00360000
.* SYMBOLS HAVE BEEN GENERATED. * 00370000
.* * 00380000
.* EXTERNAL REFERENCES: * 00390000
.* * 00400000
.* . NO EXTRNS ARE GENERATED BY THIS MACRO. * 00410000
.* * 00420000
.* . THE GLOBAL BOOLEAN VARIABLE &$NCOM IS USED TO DETERMINE * 00430000
.* WHETHER THESE SYMBOLS HAVE BEEN GENERATED. WHEN GENERATED, * 00440000
.* THE VARIABLE IS SET TO THE VALUE 1. * 00450000
.* * 00460000
.* ERROR MESSAGES: * 00470000
.* * 00480000
.* . WARNING (04) MNOTE N6004 COMMON VALUES PREVIOUSLY * 00490000
.* GENERATED * 00500000
.* * 00510000
.* . IS ISSUED IF THIS MACRO-INSTRUCTION IS USED BUT THOSE SYMBOLS * 00520000
.* HAVE BEEN PREVIOUSLY GENERATED. * 00530000
.* * 00540000
***** 00550000
$NCOM 00560000
GBLB &$NCOM . =1 IF SYMBOLS ALREADY GEN'D 00570000
TEXT 00580000
SPACE 1 00590000
```

```

*                               ***  MACRO-$NCOM  RELEASE-8  *** 00600000
.*                               00610000
.* IF SYMBOLS PREVIOUSLY GENERATED, SKIP TO ISSUE WARNING MNOTE -- DO 00620000
.* NOT GENERATE SYMBOLS AGAIN 00630000
.*                               00640000
.*   AIF   (&$NCOM).COM10      . SKIP IF SYMBOLS ALREADY GEN'D 00650000
.*                               00660000
.* GENERATE THE COMMON SYMBOLS 00670000
.*                               00680000
.*   SPACE 1 00690000
* CCP COMMON VALUES -- INDEX REGISTERS 00700000
.*   SPACE 1 00710000
$NIXR1 EQU 1 XR1--USED AS WORK REGISTER 00720000
$NIXR2 EQU 2 XR2--POINTS TO PARAMETER LIST 00730000
.*   SPACE 1 00740000
* CCP COMMON VALUES -- FOR BRANCH TO GENERAL ENTRY 00750000
.*   SPACE 1 00760000
$NSENT EQU 4 DSM SUPERVISOR GENERAL ENTRY 00770000
$NSCCR EQU X'01' DSM RIB--CCP OPERATION 00780000
$NSCCS EQU X'00' CCP SUB-RIB--COMMUNICATIONS I/O 00790000
.*   SPACE 1 00800000
* CCP COMMON VALUES -- LENGTHS 00810000
.*   SPACE 1 00820000
$NLPL EQU 16 LENGTH OF PARAMETER LIST 00830000
$NLPLF EQU 2 LENGTH OF A PARAMETER LIST FIELD 00840000
$NLSTN EQU 6 LENGTH OF SYMBOLIC TERMINAL NAME 00850000
.*                               00860000
.* INDICATE THAT COMMON SYMBOLS HAVE BEEN GENERATED 00870000
.*                               00880000
&$NCOM SETB 1 . INDICATE SYMBOLS ALREADY GEN'D 00890000
.*   AGO .COM90 . SKIP TO END CURRENT MACRO 00900000
.*                               00910000
.* SYMBOLS PREVIOUSLY GENERATED -- ISSUE WARNING MNOTE 00920000
.*                               00930000
.COM10 ANOP 00940000
.*   SPACE 1 00950000
.*   MNOTE 04,'N6004 COMMON VALUES PREVIOUSLY GENERATED' 00960000
.*   SPACE 1 00970000
.*                               00980000
.* COMMON COMPLETION OF MACRO 00990000
.*                               01000000
.COM90 ANOP 01010000
*                               ***  END GENERATION--$NCOM  *** 01020000
.*   SPACE 1 01030000
.*   MEND 01040000

```


MODULE-\$NOPV , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* 00030000
.* TITLE: $NOPV -- DEFINE CCP OPERATION CODE/MODIFIER VALUE EQUATES * 00040000
.* 00050000
.* FUNCTION: * 00060000
.* * 00070000
.* . GENERATE SYMBOLS FOR THE VALUES OF OPERATION CODES / MODIFIERS * 00080000
.* FOR A CCP COMMUNICATIONS OPERATION. * 00090000
.* * 00100000
.* . PREVENT THE INADVERTANT GENERATION OF DUPLICATE SYMBOLS FOR * 00110000
.* THOSE VALUES. * 00120000
.* * 00130000
.* INPUT: * 00140000
.* * 00150000
.* . NO LABEL SHOULD BE SPECIFIED FOR THIS MACRO. * 00160000
.* * 00170000
.* . THIS MACRO HAS NO OPERANDS. * 00180000
.* * 00190000
.* . SEE SRL #GC21-7579 -- CCP PROGRAMMING REFERENCE MANUAL -- FOR * 00200000
.* FURTHER DESCRIPTION OF THIS MACRO. * 00210000
.* * 00220000
.* OUTPUT: * 00230000
.* * 00240000
.* . IF THESE SYMBOLS HAVE BEEN GENERATED EARLIER IN THIS * 00250000
.* GENERATION RUN, ONLY A WARNING MNOTE (SEE ERROR MESSAGES * 00260000
.* BELOW). * 00270000
.* * 00280000
.* . ELSE, SYMBOLS (PREFIXES $NC AND $NM) AND THEIR EQUATES FOR * 00290000
.* THE VALUES OF OPERATION CODES AND MODIFIERS. * 00300000
.* * 00310000
.* . AN INDICATION (VIA THE GLOBAL VARIABLE &$NOPV) THAT THESE * 00320000
.* SYMBOLS HAVE BEEN GENERATED. * 00330000
.* * 00340000
.* EXTERNAL REFERENCES: * 00350000
.* * 00360000
.* . NO EXTRNS ARE GENERATED BY THIS MACRO. * 00370000
.* * 00380000
.* . THE GLOBAL BOOLEAN VARIABLE &$NOPV IS USED TO DETERMINE * 00390000
.* WHETHER THESE SYMBOLS HAVE BEEN GENERATED. WHEN GENERATED, * 00400000
.* THE VARIABLE IS SET TO THE VALUE 1. * 00410000
.* * 00420000
.* ERROR MESSAGES: * 00430000
.* * 00440000
.* . WARNING (04) MNOTE N6002 OPERATION VALUES PREVIOUSLY GENERATED * 00450000
.* * 00460000
.* IS ISSUED IF THIS MACRO-INSTRUCTION IS USED BUT THOSE SYMBOLS * 00470000
.* HAVE BEEN PREVIOUSLY GENERATED. * 00480000
.* * 00490000
.* ***** 00500000
.* $NOPV 00510000
.* GBLB &$NOPV . =1 IF SYMBOLS ALREADY GEN'D 00520000
.* TEXT 00530000
.* SPACE 1 00540000
.* *** MACRO-$NOPV RELEASE-8 *** 00550000
.* * 00560000
.* IF SYMBOLS PREVIOUSLY GENERATED, SKIP TO ISSUE WARNING MNOTE -- DO 00570000
.* NOT GENERATE SYMBOLS AGAIN 00580000
.* * 00590000
```

```

AIF (&$NOPV).OPV10 . SKIP IF SYMBOLS ALREADY GEN'D 00600000
.* 00610000
.* GENERATE THE OPERATION CODE SYMBOLS 00620000
.* 00630000
SPACE 1 00640000
* CCP OPERATION CODE VALUES 00650000
SPACE 1 00660000
$NCSHQ EQU X'0000' SHUTDOWN INQUIRY 00670000
$NCGET EQU X'0001' GET 00680000
$NCPUT EQU X'0002' PUT 00690000
$NCPTG EQU X'0003' PUT THEN GET 00700000
$NCACC EQU X'0004' ACCEPT INPUT 00710000
$NCINV EQU X'0005' INVITE INPUT 00720000
$NCPNW EQU X'0006' PUT-NO-WAIT 00730000
$NCGTA EQU X'0008' GET TERMINAL ATTRIBUTES 00740000
$NCACQ EQU X'0009' ACQUIRE TERMINAL 00750000
$NCREL EQU X'000A' RELEASE TERMINAL 00760000
$NCCPY EQU X'0042' COPY (DFF ONLY) 00770000
$NCEAU EQU X'0052' ERASE ALL UNPROTECTED (DFF ONLY) 00780000
$NCSPI EQU X'0401' STOP INVITE/GET 00790000
$NCWAT EQU X'0014' WAIT OPERATION CODE 00795000
.* 00800000
.* GENERATE THE OPERATION MODIFIER SYMBOLS 00810000
.* 00820000
SPACE 1 00830000
* CCP OPERATION MODIFIER VALUES 00840000
SPACE 1 00850000
$NMSTA EQU X'0010' SET TERMINAL ATTRIBUTES BY ID 00860000
$NMRVI EQU X'0010' SEND REVERSE-INTERRUPT 00870000
$NMKPL EQU X'0010' KEEP THE LINE 00880000
$NMBLK EQU X'0020' END THE CURRENT OUTPUT BLOCK 00890000
$NMMSG EQU X'0030' SEND END-OF-TRANSMISSION 00900000
$NMPRF EQU X'0040' PROGRAM REQUEST UNDER FORMAT 00905000
$NMNEL EQU X'0100' RECORD DOES NOT END CURRENT LINE 00910000
$NMNNL EQU X'0200' RECORD DOES NOT START NEW LINE 00920000
$NMOVR EQU X'0800' OVERRIDE/SELECTED-FIELDS LIST 00930000
.* 00940000
.* INDICATE THAT SYMBOLS HAVE BEEN GENERATED 00950000
.* 00960000
&$NOPV SETB 1 . INDICATE SYMBOLS ALREADY GEN'D 00970000
AGO .OPV90 . SKIP TO END CURRENT MACRO 00980000
.* 00990000
.* SYMBOLS PREVIOUSLY GENERATED -- ISSUE WARNING MNOTE 01000000
.* 01010000
.OPV10 ANOP 01020000
SPACE 1 01030000
MNOTE 04, 'N6002 OPERATION VALUES PREVIOUSLY GENERATED' 01040000
SPACE 1 01050000
.* 01060000
.* COMMON COMPLETION OF MACRO 01070000
.* 01080000
.OPV90 ANOP 01090000
* *** END GENERATION--$NOPV *** 01100000
SPACE 1 01110000
MEND 01120000

```

```

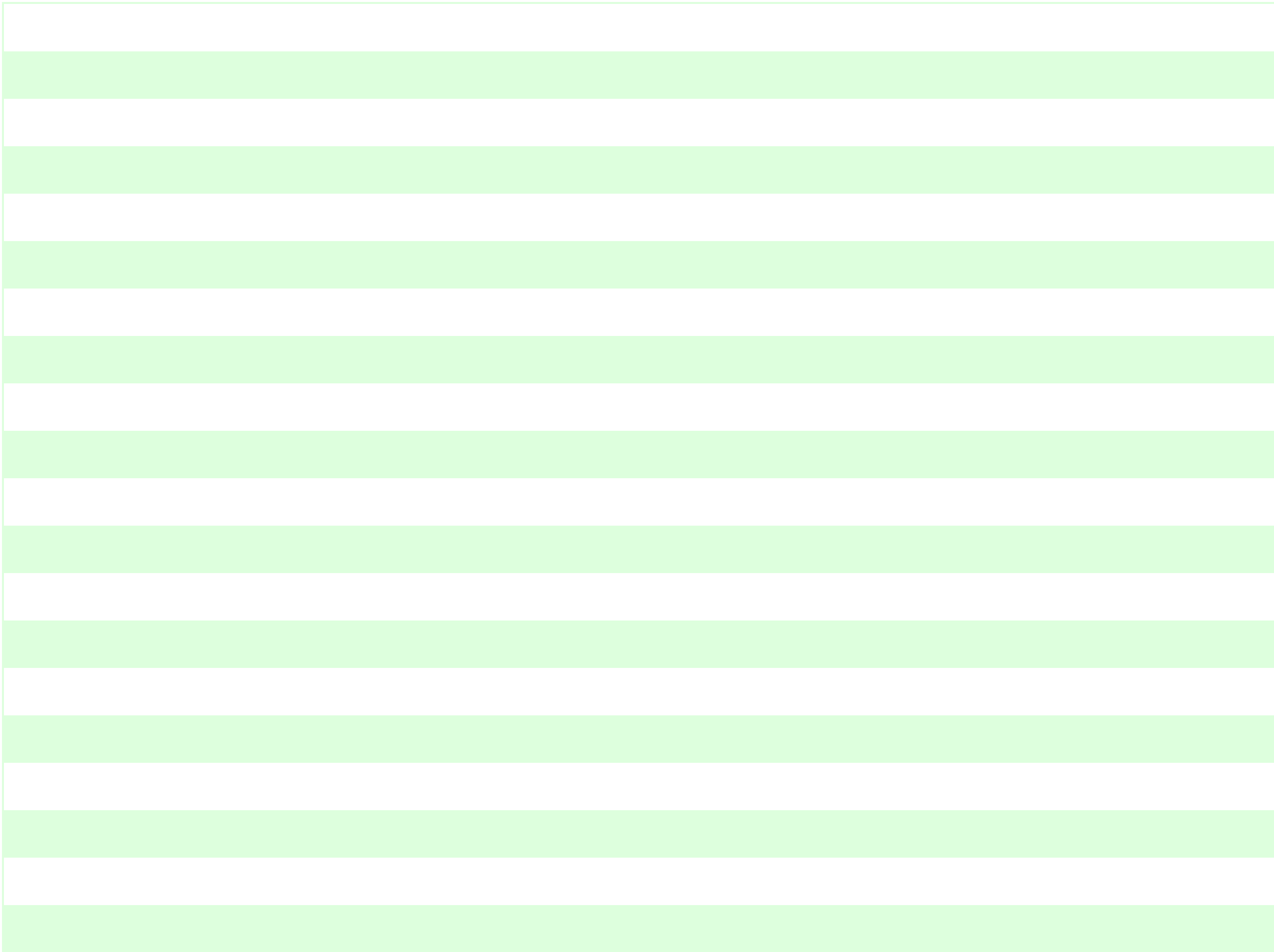
MODULE-$NPL , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO 00010000
***** 00020000
.* 00030000
.* TITLE: $NPL -- DEFINE CCP COMMUNICATIONS PARAMETER LIST * 00040000
.* 00050000
.* FUNCTION: * 00060000
.* 00070000
.* . GENERATE A 16-BYTE COMMUNICATIONS PARAMETER LIST * 00080000
.* 00090000
.* . PERMIT THE USER TO SPECIFY INITIAL VALUES FOR CERTAIN FIELDS * 00100000
.* WITHIN THAT LIST * 00110000
.* 00120000
.* INPUT: * 00130000
.* 00140000
.* . USER MAY SPECIFY A LABEL FOR THE PARAMETER LIST * 00150000
.* 00160000
.* . THE FOLLOWING OPERANDS, ALL OPTIONAL, PERMIT THE INITIALIZATION * 00170000
.* OF FIELDS WITHIN THE GENERATED PARAMETER LIST: * 00180000
.* 00190000
.* OP-ABSOLUTE EXPRESSION * 00200000
.* 00210000
.* OUTLEN-ABSOLUTE EXPRESSION * 00220000
.* 00230000
.* ATTRID-ABSOLUTE EXPRESSION * 00240000
.* 00250000
.* INLEN-ABSOLUTE EXPRESSION * 00260000
.* 00270000
.* RECA-RELOCATABLE EXPRESSION * 00280000
.* 00290000
.* . THE OPERANDS 'OUTLEN' AND 'ATTRID' ARE MUTUALLY EXCLUSIVE. * 00300000
.* 00310000
.* . SEE SRL #GC21-7579 -- CCP PROGRAMMING REFERENCE MANUAL -- FOR * 00320000
.* FURTHER DESCRIPTION OF THIS MACRO. * 00330000
.* 00340000
.* OUTPUT: * 00350000
.* 00360000
.* . EIGHT 2-BYTE DEFINED CONSTANTS * 00370000
.* 00380000
.* . A LABEL (IF SPECIFIED BY USER) EQUATED TO HI-ORDER BYTE OF THE * 00390000
.* GENERATED PARAMETER LIST * 00400000
.* 00410000
.* EXTERNAL REFERENCES: * 00420000
.* 00430000
.* . NO EXTRNS ARE GENERATED BY THIS MACRO * 00440000
.* 00450000
.* . NO GLOBAL VARIABLES ARE USED BY THIS MACRO * 00460000
.* 00470000
.* ERROR MESSAGES: * 00480000
.* 00490000
.* . ERROR MNOTE (08) N2001 CONFLICTING OPERANDS--OUTLEN/ATTRID * 00500000
.* 00510000
.* ISSUED IF THESE MUTUALLY EXCLUSIVE OPERANDS ARE BOTH PRESENT * 00520000
.* IN THE SAME MACRO-INSTRUCTION. * 00530000
.* 00540000
***** 00550000
&LABEL $NPL &OP-0,&OUTLEN-,&ATTRID-,&INLEN-0,&RECA-'*+8' 00560000
TEXT 00570000
SPACE 00580000
* *** MACRO-$NPL RELEASE-8 *** 00590000

```

```

.*
.* IF LABEL SPECIFIED, GENERATE EQUATE TO HI-ORDER OF PARAMETER LIST 00610000
.* 00620000
    AIF (T'&LABEL EQ 'O').PL10 . SKIP EQUATE IF NO LABEL SPEC'D 00630000
&LABEL EQU * HI-ORDER ADDR OF PARAMETER LIST 00640000
.PL10 ANOP . RESUME HERE 00650000
.* 00660000
.* GENERATE RETURN-CODE FIELD AS ZEROS 00670000
.* 00680000
    DC AL2(0) RTC--RETURN CODE 00690000
.* 00700000
.* GENERATE OPERATION CODE/MODIFIERS FIELD FROM USER'S SPECIFICATION 00710000
.* 00720000
    DC AL2(&OP) OPC--OPERATION CODE/MODIFIERS 00730000
.* 00740000
.* IF BOTH 'OUTLEN' AND 'ATTRID' SPECIFIED, ISSUE ERROR MNOTE 00750000
.* 00760000
    AIF (T'&ATTRID EQ 'O').PL20 . SKIP IF *NO* 'ATTRID' 00770000
    AIF (T'&OUTLEN EQ 'O').PL30 . SKIP IF ATTRID, *NO* 'OUTLEN' 00780000
    SPACE 00790000
    MNOTE 08,'N2001 CONFLICTING OPERANDS -- OUTLEN/ATTRID' 00800000
    SPACE 00810000
    AGO .PL40 . SKIP TO TREAT AS IF *NEITHER* 00820000
.* 00830000
.* NO ATTRIBUTES IDENTIFIER SPECIFIED -- IF OUTPUT LENGTH SPECIFIED, 00840000
.* GENERATE AS PER USER'S SPECIFICATIONS -- ELSE GENERATE ZEROS 00850000
.* 00860000
.PL20 ANOP 00870000
    AIF (T'&OUTLEN EQ 'O').PL40 . SKIP IF *NO* 'OUTLEN' 00880000
    DC AL2(&OUTLEN) OUL/EFL/ATI--LENGTHS OR ID 00890000
    AGO .PL50 . SKIP TO JOIN COMMON GENERATION 00900000
.* 00910000
.* ATTRIBUTES IDENTIFIER SPECIFIED -- GENERATE AS PER USER SPECS 00920000
.* 00930000
.PL30 ANOP 00940000
    DC AL2(&ATTRID) ATI/OUL/EFL--ID OR LENGTHS 00950000
    AGO .PL50 . SKIP TO JOIN COMMON GENERATION 00960000
.* 00970000
.* NEITHER 'OUTLEN' NOR 'ATTRID' SPECIFIED -- GENERATE ZEROS 00980000
.* 00990000
.PL40 ANOP 01000000
    DC AL2(0) OUL/EFL/ATI--LENGTHS OR ID 01010000
.* 01020000
.* GENERATE MAXIMUM INPUT LENGTH AS PER USER SPECIFICATION 01030000
.* 01040000
.PL50 ANOP . COMMON RESUME POINT 01050000
    DC AL2(&INLEN) INL--MAXIMUM INPUT LENGTH 01060000
.* 01070000
.* GENERATE RECORD AREA ADDRESS AS PER USER SPECIFICATION 01080000
.* 01090000
    DC AL2(&RECA) RAA--ADDRESS OF RECORD AREA 01100000
.* 01110000
.* GENERATE THREE INTERNAL WORK FIELDS AS ZEROS 01120000
.* 01130000
    DC AL2(0) WKA--INTERNAL WORK FIELD 01140000
    DC AL2(0) WKB--INTERNAL WORK FIELD 01150000
    DC AL2(0) WKC--INTERNAL WORK FIELD 01160000
* *** END GENERATION -- $NPL *** 01170000
    SPACE 01180000
    MEND 01190000

```



```

MODULE-$NPLO , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO 00010000
***** 00020000
.* 00030000
.* TITLE: $NPLO -- DEFINE CCP PARAMETER LIST OFFSET EQUATES * 00040000
.* 00050000
.* FUNCTION: * 00060000
.* * 00070000
.* . GENERATE SYMBOLS FOR THE OFFSETS OF FIELDS WITHIN A CCP * 00080000
.* COMMUNICATIONS PARAMETER LIST. * 00090000
.* * 00100000
.* . PREVENT THE INADVERTANT GENERATION OF DUPLICATE SYMBOLS FOR * 00110000
.* THESE OFFSET VALUES. * 00120000
.* * 00130000
.* INPUT: * 00140000
.* * 00150000
.* . NO LABEL SHOULD BE SPECIFIED FOR THIS MACRO * 00160000
.* * 00170000
.* . THIS MACRO HAS NO OPERANDS * 00180000
.* * 00190000
.* . SEE SRL #GC21-7579 -- CCP PROGRAMMING REFERENCE MANUAL -- FOR * 00200000
.* FURTHER DESCRIPTION OF THIS MACRO. * 00210000
.* * 00220000
.* OUTPUT: * 00230000
.* * 00240000
.* . IF THESE SYMBOLS HAVE BEEN GENERATED EARLIER IN THIS * 00250000
.* GENERATION, ONLY A WARNING MNOTE (SEE ERROR MESSAGES BELOW). * 00260000
.* * 00270000
.* . ELSE, SYMBOLS (PREFIX $NP) AND THEIR EQUATES FOR THE OFFSETS * 00280000
.* OF FIELDS WITHIN A COMMUNICATIONS PARAMETER LIST. * 00290000
.* * 00300000
.* . AN INDICATION (VIA THE GLOBAL VARIABLE &$NPLO) THAT THESE * 00310000
.* SYMBOLS HAVE BEEN GENERATED. * 00320000
.* * 00330000
.* EXTERNAL REFERENCES: * 00340000
.* * 00350000
.* . NO EXTRNS ARE GENERATED BY THIS MACRO. * 00360000
.* * 00370000
.* . THE GLOBAL BOOLEAN VARIABLE &$NPLO IS USED TO DETERMINE * 00380000
.* WHETHER THESE SYMBOLS HAVE BEEN GENERATED. WHEN GENERATED, * 00390000
.* VARIABLE IS SET TO THE VALUE 1. * 00400000
.* * 00410000
.* ERROR MESSAGES: * 00420000
.* * 00430000
.* . A WARNING (04) MNOTE N6001 OFFSET VALUES PREVIOUSLY GENERATED * 00440000
.* * 00450000
.* IS ISSUED IF THIS MACRO-INSTRUCTION IS USED BUT THOSE SYMBOLS * 00460000
.* HAVE BEEN PREVIOUSLY GENERATED. * 00470000
.* * 00480000
.* ***** 00490000
.* $NPLO 00500000
.* GBLB &$NPLO . =1 IF SYMBOLS ALREADY GEN'D 00510000
.* TEXT 00520000
.* SPACE 00530000
.* *** MACRO-$NPLO RELEASE-8 *** 00540000
.* * 00550000
.* IF SYMBOLS PREVIOUSLY GENERATED, SKIP TO ISSUE WARNING MNOTE -- DO 00560000
.* NOT GENERATE SYMBOLS AGAIN 00570000
.* * 00580000
.* AIF (&$NPLO).PLO10 . SKIP IF SYMBOLS ALREADY GEN'D 00590000

```

```

.*
.* GENERATE THE OFFSET-EQUATE SYMBOLS
.*
    SPACE 1
* OFFSETS OF COMMUNICATIONS PARAMETER LIST FIELDS
    SPACE 1
$NPRTC EQU +1          RETURN CODE FIELD
$NPOPC EQU +3          OPERATION CODE/MODIFIERS FIELD
$NPOUL EQU +5          OUTPUT LENGTH FIELD
$NPEFL EQU +5          EFFECTIVE INPUT LENGTH FIELD
$NPATI EQU +5          ATTRIBUTES IDENTIFIER FIELD
$NPINL EQU +7          MAXIMUM INPUT LENGTH FIELD
$NPRAA EQU +9          RECORD AREA ADDRESS FIELD
$NPWKA EQU +11         INTERNAL WORK FIELD
$NPWKB EQU +13         INTERNAL WORK FIELD
$NPWKC EQU +15         INTERNAL WORK FIELD
.*
.* INDICATE THAT SYMBOLS HAVE BEEN GENERATED
.*
&$NPLO SETB 1          . INDICATE SYMBOLS ALREADY GEN'D
    AGO .PLO90          . SKIP TO END CURRENT MACRO
.*
.* SYMBOLS PREVIOUSLY GENERATED -- ISSUE WARNING MNOTE
.*
.PLO10 ANOP
    SPACE
    MNOTE 04,'N6001 OFFSET VALUES PREVIOUSLY GENERATED'
    SPACE
.*
.* COMMON COMPLETION OF MACRO
.*
.PLO90 ANOP
*                      *** END GENERATION--$NPLO ***
    SPACE
    MEND

```

```

00600000
00610000
00620000
00630000
00640000
00650000
00660000
00670000
00680000
00690000
00700000
00710000
00720000
00730000
00740000
00750000
00760000
00770000
00780000
00790000
00800000
00810000
00820000
00830000
00840000
00850000
00860000
00870000
00880000
00890000
00900000
00910000
00920000
00930000
00940000

```

```

MACRO                                00010000
*****                                00020000
*                                     * 00030000
* TITLE: $NRTV -- DEFINE CCP RETURN CODE VALUE EQUATES * 00040000
*                                     * 00050000
* FUNCTION:                            * 00060000
*                                     * 00070000
* . GENERATE SYMBOLS FOR THE VALUES OF RETURN CODES FROM A CCP * 00080000
* COMMUNICATIONS OPERATION.           * 00090000
*                                     * 00100000
* . PREVENT THE INADVERTANT GENERATION OF DUPLICATE SYMBOLS FOR * 00110000
* THOSE VALUES.                      * 00120000
*                                     * 00130000
* INPUT:                                * 00140000
*                                     * 00150000
* . NO LABEL SHOULD BE SPECIFIED FOR THIS MACRO.                * 00160000
*                                     * 00170000
* . THIS MACRO HAS NO OPERANDS.      * 00180000
*                                     * 00190000
* . SEE SRL #GC21-7579 -- CCP PROGRAMMING REFERENCE MANUAL -- FOR * 00200000
* FURTHER DESCRIPTION OF THIS MACRO. * 00210000
*                                     * 00220000
* OUTPUT:                               * 00230000
*                                     * 00240000
* . IF THESE SYMBOLS HAVE BEEN GENERATED EARLIER IN THIS        * 00250000
* GENERATION RUN, ONLY A WARNING MNOTE (SEE ERROR MESSAGES      * 00260000
* BELOW).                                                         * 00270000
*                                     * 00280000
* . ELSE, SYMBOLS FOR RETURN CODE VALUES:                       * 00290000
*                                     * 00300000
* . $NOK FOR SUCCESSFUL OPERATION                                * 00310000
*                                     * 00320000
* . $NX... FOR EXCEPTIONAL CONDITIONS                          * 00330000
*                                     * 00340000
* . $NR... FOR I/O ERROR CONDITIONS                            * 00350000
*                                     * 00360000
* . AN INDICATION (VIA THE GLOBAL VARIABLE &$NRTV) THAT THESE   * 00370000
* SYMBOLS HAVE BEEN GENERATED.                                  * 00380000
*                                     * 00390000
* EXTERNAL REFERENCES:                                         * 00400000
*                                     * 00410000
* . NO EXTRNS ARE GENERATED BY THIS MACRO.                    * 00420000
*                                     * 00430000
* . THE GLOBAL BOOLEAN VARIABLE &$NRTV IS USED TO DETERMINE    * 00440000
* WHETHER THESE SYMBOLS HAVE BEEN GENERATED. WHEN GENERATED,  * 00450000
* THE VARIABLE IS SET TO THE VALUE 1.                          * 00460000
*                                     * 00470000
* ERROR MESSAGES:                                             * 00480000
*                                     * 00490000
* . WARNING (04) MNOTE N6003 RETURN-CODE VALUES PREVIOUSLY    * 00500000
* GENERATED                                                     * 00510000
*                                     * 00520000
* . IS ISSUED IF THIS MACRO-INSTRUCTION IS USED BUT THOSE SYMBOLS * 00530000
* HAVE BEEN PREVIOUSLY GENERATED.                              * 00540000
*                                     * 00550000
*****                                00560000
$NRTV                                00570000
GBLB &$NRTV . =1 IF SYMBOLS ALREADY GEN'D 00580000
TEXT                                  00590000

```



```

SPACE 1
*
*** MACRO-$NRTV RELEASE-8 ***
.*
.* IF SYMBOLS PREVIOUSLY GENERATED, SKIP TO ISSUE WARNING MNOTE -- DO
.* NOT GENERATE SYMBOLS AGAIN
.*
AIF (&$NRTV).RTV10 . SKIP IF SYMBOLS ALREADY GEN'D
.*
.* GENERATE THE SUCCESSFUL COMPLETION SYMBOL
.*
SPACE 1
* CCP RETURN CODE VALUES -- SUCCESSFUL OPERATION
SPACE 1
$NOK EQU 0 SUCCESSFUL OPERATION
.*
.* GENERATE THE EXCEPTION CONDITION SYMBOLS
.*
SPACE 1
* CCP RETURN CODE VALUES -- EXCEPTION CONDITIONS
SPACE 1
$NXDTR EQU +1 DATA TRUNCATED
$NXEOT EQU +2 EOT RECEIVED
$NXEDT EQU +3 EOT RECEIVED / DATA TRUNCATED
$NXSHD EQU +4 SHUTDOWN HAS BEEN REQUESTED
$NXDPD EQU +5 DATA PENDING ON BSCA LINE
$NXRVI EQU +6 RVI/TERMINAL INTERRUPT RECEIVED
$NXCLR EQU +7 3270 CLEAR KEY
$NXNAV EQU +8 TERMINAL NO LONGER AVAILABLE
$NXOFF EQU +9 TERMINAL OFFLINE
$NXSPI EQU +10 STOP INVITE INPUT SUCCESSFUL
$NXNAQ EQU +11 ACQUIRE TERMINAL FAILED
SPACE 1
* CCP RETURN CODE VALUES -- COMMON I/O ERRORS
SPACE 1
$NRDCK EQU -1 DATA CHECK
$NRTRN EQU -2 TRANSLATION ERROR
$NRLST EQU -3 LOST DATA
$NRPBS EQU -4 PERMANENT BI-SYNC ERROR
$NRABN EQU -5 ABNORMAL RESPONSE
$NRXRA EQU -6 TRANSMIT/RECEIVE ABORT
$NRATO EQU -7 ADDRESSING TIMEOUT
$NRTTO EQU -8 TEXT TIMEOUT
$NRWTO EQU -9 WACK/TTD EXPIRATION
$NRNOC EQU -10 NO CONNECTION
$NRIID EQU -11 INVALID ID
$NRABD EQU -12 ABORT, DISCONNECT
$NRADC EQU -13 ADAPTER CHECK
$NRNAK EQU -14 NEGATIVE RESPONSE TO ADDRESSING
SPACE 1
* CCP RETURN CODE VALUES -- I/O ERRORS UNIQUE TO 3270
SPACE 1
$NR2DU EQU -20 DEVICE UNAVAILABLE OR NOT READY
$NR2ED EQU -22 EQUIPMENT CHECK, DEVICE END
$NR2TE EQU -23 TCU DETECTION OF BSCA ERROR
$NR2CD EQU -24 CTL CHK,DATA CHK,DEV BSY,DEV END
$NR2PD EQU -25 DATA CHECK ON COPY COMMAND
$NR2PO EQU -26 OPERATION CHECK ON COPY COMMAND
$NR2PB EQU -27 DEVICE BUSY ON COPY COMMAND
$NR2PC EQU -28 CTL CHK/OP CHK/DATA CHK ON COPY
$NR2PI EQU -29 INVALID INPUT DATA FOR DFF

```

```

00600000
00610000
00620000
00630000
00640000
00650000
00660000
00670000
00680000
00690000
00700000
00710000
00720000
00730000
00740000
00750000
00760000
00770000
00780000
00790000
00800000
00810000
00820000
00830000
00840000
00850000
00860000
00870000
00880000
00890000
00900000
00910000
00920000
00930000
00940000
00950000
00960000
00970000
00980000
00990000
01000000
01010000
01020000
01030000
01040000
01050000
01060000
01070000
01080000
01090000
01100000
01110000
01120000
01130000
01140000
01150000
01160000
01170000
01180000
01190000

```

SPACE 1		01200000
* CCP RETURN CODE VALUES -- I/O ERRORS UNIQUE TO 3735		01210000
SPACE 1		01220000
\$NR5SR EQU -40	ATTEMPTED SEND BEFORE RECEIVE	01230000
\$NR5IC EQU -41	ILLEGAL CHARACTER	01240000
\$NR5BF EQU -42	BUFFER OVERFLOW	01250000
\$NR5DF EQU -43	DISK FULL	01260000
\$NR5RF EQU -44	DIRECTORY FULL	01270000
\$NR5UH EQU -45	UNDEFINED HEADER	01280000
\$NR5DE EQU -46	3735 DISK ERROR	01290000
.*		01300000
.*	INDICATE THAT SYMBOLS HAVE BEEN GENERATED	01310000
.*		01320000
&\$NRTV SETB 1	. INDICATE SYMBOLS ALREADY GEN'D	01330000
AGO .RTV90	. SKIP TO END CURRENT MACRO	01340000
.*		01350000
.*	SYMBOLS PREVIOUSLY GENERATED -- ISSUE WARNING MNOTE	01360000
.*		01370000
.RTV10 ANOP		01380000
SPACE 1		01390000
MNOTE 04, 'N6003 RETURN-CODE VALUES PREVIOUSLY GENERATED'		01400000
SPACE 1		01410000
.*		01420000
.*	COMMON COMPLETION OF MACRO	01430000
.*		01440000
.RTV90 ANOP		01450000
*	*** END GENERATION--\$NRTV ***	01460000
SPACE 1		01470000
MEND		01480000

```

MODULE-$DFOB , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$DFOB                                00110000
TEXT                                  00120000
*   BSCA EQUATES.                     RELEASE-8    00130000
    SPACE                              00140000
*   EQUATES IN IOB                    00150000
$BIFLA EQU 5                          FLAG A.    00160000
$BIFST EQU X'04'                      FIRST IN FLAG A. 00170000
$BIOBQ EQU X'02'                      OFFSET TO IOB OPERATION. 00180000
$BIRVI EQU X'83'                      RCVI OPERATION. 00190000
$BICMP EQU X'07'                      OFFSET TO IOB COMPLETION. 00200000
    SPACE                              00210000
*   EQUATES FOR WORK AREA.           00220000
$BWK EQU 1                            WORK AREA REG. 00230000
$BWFG3 EQU 29                         FLAG 3.    00240000
$BWLGD EQU 35                         FLAG BYTE  00250000
$BPATV EQU X'10'                      LINE ACTIVE INDICATOR 00260000
$BWRFT EQU X'08'                      RFT IND. IN FLAG 3. 00270000
$BPOLD EQU X'17'                      OFFSET TO POLL INDICS IN WKA. 00280000
$BPCNC EQU X'04'                      CANCEL POST INDIC. 00290000
$BPRES EQU X'10'                      RESET POLL INDIC. 00300000
$BWKMC EQU X'59'                      OFFSET FOR IDA INDICS IN WKA. 00310000
    SPACE                              00320000
*   EQUATES FOR $BCPL AND $BCSW MACROS. 00330000
$BPATR EQU 0                          CHANGE LIST ATTRIBUTE OFFSET. 00340000
$BPACT EQU X'80'                      OFF-ACTIVATE;ON-DEACTIVATE. 00350000
$BPEXT EQU X'40'                      OFF-EXACT,ON-FIRST N CHARS. 00360000
$BPDTF EQU 2                          CHANGE LIST DTF ADDR OFFSET. 00370000
$BPNUM EQU 3                          CHANGE LIST OFFSET TO LENGTH. 00380000
$BPEND EQU X'FE'                      END OF POLL/ADDR OR SW ID LIST. 00390000
$BPNOP EQU X'00'                      NO-OP JUMP INSTR. 00400000
$BPENA EQU X'80'                      ON-ACTIVE;OFF-INACT. LIST ATTR. 00410000
$BPRM1 EQU 1                          REG EQU FOR MACRO PARM LIST. 00420000
$BLST2 EQU 2                          REG EQU FOR POLL OR ID LIST- XR2. 00430000
$BLIST EQU 2                          REG EQU FOR PTR TO LIST IN XR2. 00440000
    SPACE                              00450000
*   EQUATES FOR $RFT MACRO           00460000
$BRCNT EQU 3                          COUNT OF NUMBER OF TRANSMISSIONS 00470000
$BHX0F EQU X'0F'                      MASK TO CHECK FOR DECIMAL NUMBER 00480000
    SPACE                              00490000
*   GENERAL EQUATES.                 00500000
$BDISA EQU X'80'                      ENABLE BSCA. 00510000
$BENAB EQU X'C0'                      DISABLE BSCA. 00520000
$BFOX EQU X'FF'                      EQUATE FOR 'FF'. 00530000
$BBAC1 EQU 1                          USER REGISTER SAVE (REG 1). 00540000
$BPRS2 EQU 2                          PARAMETER REGISTER SAVE (REG 2). 00550000
$B2SEC EQU X'06'                      TWO SEC TIME FOR IDA. 00560000
    SPACE 3                            00570000
*   EQUATES FOR $CANB MACRO          00580000
$BLIN2 EQU X'08'                      LINE-2    00590000
$BTREQ EQU X'16'                      TRUE AND EQUAL 00600000
$BDAON EQU X'01'                      D.A. SUPPORTED. 00610000
$BTOSC EQU X'88'                      TWO SEC TIME OUT. 00620000
$BTRNQ EQU X'11'                      TRUE AND NOT EQUAL 00630000
$BDTF EQU 2                          DTF REG. 00640000
$BIOB EQU 1                          IOB REG. 00650000
$BONE EQU 1                          CONSTANT ONE 00660000
$BTRE EQU 3                          CONSTANT THREE 00670000
    SPACE 3                            00680000

```

*		OFFSETS FOR BSCA DTF.	00690000
\$BDDEV EQU	0	DEVICE ID.	00700000
\$BDUPS EQU	1	UPSI.	00710000
\$BDATT EQU	2	ATTRIBUTE BYTE 1.	00720000
\$BCINP EQU	X'80'	INPUT FILE.	00730000
\$BCOUT EQU	X'40'	OUTPUT FILE.	00740000
\$BCCNV EQU	X'C0'	CONVERSATIONAL FILE.	00750000
\$BCITB EQU	X'20'	ITB MODE.	00760000
\$BCRAN EQU	X'10'	TRANSPARENCY.	00770000
\$BCGET EQU	X'08'	GET FILE.	00780000
\$BCASK EQU	X'04'	ON-ASCII; OFF-EBCDIC.	00790000
\$BCASM EQU	X'01'	ASSEM DTF.	00800000
\$BDATR EQU	3	ATTRIBUTE BYTE 2.	00810000
\$BCMCN EQU	X'88'	MULTIPOINT CONTROL STATION.	00820000
\$BCMPT EQU	X'80'	MULTIPOINT TRIBUTARY.	00830000
\$BCMAN EQU	X'20'	MANUAL LINE.	00840000
\$BCANS EQU	X'10'	ANSWER LINE.	00850000
\$BCSWI EQU	X'08'	SWITCHED LINE.	00860000
\$BCUSD EQU	X'04'	FILE USED.	00870000
\$BCACT EQU	X'02'	FILE ACTIVE.	00880000
\$BCOPN EQU	X'01'	FILE OPENED.	00890000
\$BDCHN EQU	5	POST OPEN DTF CHAINING PTR.	00900000
\$BDNXT EQU	7	DTF CHAINING POINTER.	00910000
\$BDWK1 EQU	9	WORK AREA.	00920000
\$BDWK2 EQU	11	WORK AREA.	00930000
\$BDWKB EQU	13	ADDRESS OF USER'S LOGICAL BUFF.	00940000
\$BDCMP EQU	14	COMPLETION CODE.	00950000
\$BCREQ EQU	X'00'	REQUEST ACCEPTED.	00960000
\$BCDNE EQU	X'40'	NORMAL COMPLETION.	00970000
\$BCUER EQU	X'41'	USER ERROR.	00980000
\$BCEOT EQU	X'42'	END OF FILE.	00990000
\$BCBID EQU	X'43'	INVALID ID.	01000000
\$BCNEG EQU	X'44'	NEGATIVE RESPONSE TO POLL/ADDR.	01010000
\$BCNON EQU	X'45'	NO RESPONSE TO POLL/ADDR.	01020000
\$BCCRP EQU	X'46'	CONV REPLY PENDING.	01030000
\$BCNDT EQU	X'47'	NO DATA FOR CONV GET.	01040000
\$BCOLT EQU	X'48'	INVALID RFT REQUEST.	01050000
\$BCNAC EQU	X'49'	NO ACT ENTRY IN POLL LIST.	01060000
\$BCIGN EQU	X'4A'	REQUEST IGNORED.	01070000
\$BCASC EQU	X'4B'	INVALID ASCII CHARACTER.	01080000
\$BCNCN EQU	X'4C'	NO-CONNECTION.	01090000
\$BCCAL EQU	X'4D'	INVALID REQUEST.	01100000
\$BCLST EQU	X'4E'	DELAY COUNT EXCEEDED.	01110000
\$BCERR EQU	X'4F'	PERM ERROR.	01120000
\$BCTIM EQU	X'50'	NO RESP FROM REMOTE DEV.	01130000
\$BCDAT EQU	X'51'	DATA CHECK.	01140000
\$BCLOS EQU	X'52'	LOST DATA.	01150000
\$BCCON EQU	X'53'	LOST CONNECTION.	01160000
\$BCRSP EQU	X'54'	INVALID RESP FROM REMOTE DEV.	01170000
\$BCADP EQU	X'55'	ADAPTER CHECK.	01180000
\$BCCMP EQU	X'56'	NO COMPLETIONS IN CHECK LIST.	01190000
\$BCACD EQU	X'57'	NO ACTIVE DTFS IN CHECK LIST.	01200000
\$BCRLE EQU	X'58'	MAXIMUM RECORD LENGTH EXCEEDED.	01210000
\$BDOPC EQU	15	OPERATION CODE.	01220000
\$BOGET EQU	X'80'	GET.	01230000
\$BOGBK EQU	X'81'	GET-BLOCK.	01240000
\$BOPUT EQU	X'40'	PUT.	01250000
\$BOPEB EQU	X'41'	PUT END OF BLOCK.	01260000
\$BOPEF EQU	X'42'	PUT END OF FILE.	01270000
\$BOPEW EQU	X'44'	PUT EOT TO WACK RESPONSE.	01280000

\$BDMRL	EQU	17	MAXIMUM RECORD LENGTH.	01290000
\$BDADD	EQU	18	SPECIAL USE INDICATORS	01300000
\$BCAA1	EQU	X'01'	ADD ON AREA ON DTF	01310000
\$BCPOL	EQU	X'02'	POLLING MODULES RESIDENT	01320000
\$BCOFL	EQU	X'04'	TRUNCATE RECORD INDICATOR.	01330000
\$BCRCL	EQU	X'08'	SPAN INDICATOR FOR RECORD LENGTH	01340000
\$BCTWO	EQU	X'10'	END OF BLOCK INDICATOR.	01350000
\$BCSWD	EQU	X'80'	ID LIST FOR SWICHD LINE	01360000
\$BDDCH	EQU	20	*ADDRESS OF DIAL NUMBER OR	01370000
\$BDPSC	EQU	20	*POLL/ADDR CHARACTERS OR	01380000
\$BDLST	EQU	20	*ADDRESS OF POLL/ADDR LIST.	01390000
\$BDDCC	EQU	21	*LENGTH OF DIAL NUMBER OR	01400000
\$BDIND	EQU	21	*POLLING/OR ADDRESSING ID.	01410000
\$BDRID	EQU	23	*ADDR OF RCV ID OR ID LIST OR	01420000
\$BDCNT	EQU	23	*LIST COUNT.	01430000
\$BDRLN	EQU	24	LEN OF RCV ID OR ENTRY SELECTOR.	01440000
\$BDLID	EQU	24	LAST ID OR POLL/ADDR FUNCTION.	01450000
\$BDSID	EQU	26	ADDRESS OF SEND ID.	01460000
\$BDSLN	EQU	27	LENGTH OF SEND ID.	01470000
\$BDDLY	EQU	29	DELAY COUNT.	01480000
\$BDREL	EQU	31	RECORD LENGTH.	01490000
\$BDBKL	EQU	33	BLOCK LENGTH.	01500000
\$BDIOB	EQU	35	ADDRESS OF IOB IN PROCESS.	01510000
\$BDBKX	EQU	37	POINT TO DATA IN BSCA BUFFER.	01520000
\$BDITB	EQU	39	ITB CHARACTER COUNT.	01530000
\$BDPRM	EQU	42	RESERVED.	01540000
\$BDRVI	EQU	45	RVI MASK AND DISPLACEMENT.	01550000
\$BDNDX	EQU	46	INDEX FOR LINE INITIALIZATION.	01560000
\$BDWKA	EQU	48	ADDRESS OF BSCA WORK AREA.	01570000
\$BDINT	EQU	50	DISK ADDR OF LINE INIT MODULE.	01580000
\$BDDED	EQU	51	WORK AREA.	01590000
\$BDAT1	EQU	52	ATTRIBUTE BYTE FOR TERMINALS.	01600000
\$BCSEP	EQU	X'01'	RECORD SEPARATOR.	01610000
\$BCSPN	EQU	X'02'	SPANNING RECORD.	01620000
\$BCNOW	EQU	X'04'	SPAN IN PROCESS.	01630000
\$BCPUT	EQU	X'08'	PUT SPAN FILE.	01640000
\$BCRES	EQU	X'10'	SPAN RESTORE NECESSARY.	01650000
\$BCPLR	EQU	X'40'	POLLING RESIDENT.	01660000
\$BDSEP	EQU	53	RECORD SEPARATOR.	01670000
\$BDSBF	EQU	55	SAVE AREA FOR USER BUFFER ADDR.	01680000
\$BDSRL	EQU	57	SAVE AREA FOR RECORD LENGTH.	01690000
\$BDRFT	EQU	59	SAVE AREA FOR OLT PARM.	01700000
\$BDTSA	EQU	61	ADDR OF TERM LOG AREA.	01710000
*		ADD ON AREA	OF DTF	01720000
\$BDRL0	EQU	63	ADDR OF RESIDENT L0.	01730000
\$BDRCL	EQU	65	ADDR OF RESIDENT CLOSE.	01740000
\$BDARA	EQU	67	AUTO RESPONSE MODULE.	01750000
\$BDERR	EQU	68	RETRY COUNT.	01760000
\$BDT1A	EQU	70	SAVE ADDRESS OF OLT CS.	01770000
\$BDEX@	EQU	72	CCP USER EXIT ADDRESS.	01780000
\$BD375	EQU	73	EOT SENT TO PERM ERROR	01790000
		MEND		01800000

```

MODULE-$DTOM , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$DTOM &AT1-N,&AT2-N,&CMP-N,&OPC-N,&OSC-N,&SNS-N,&TFT-N 00020000
GBLB &$MG                             00290000
TEXT                                 00300000
AIF (&$MG EQ '0').TEXT               00310000
PRINT OFF                             00320000
.TEXT ANOP                             00330000
*****                                00340000
*                                     * 00350000
*           DTF OFFSET EQUATES FOR THE MLTA * 00360000
*                                     * 00370000
*****                                00380000
$MDDEV EQU 0 DEVICE ID - X'28' FOR MLTA LINES 00390000
$MDUP EQU $MDDEV+1 UPSI U1-U8 CUSTOMER CONTROLLED 00400000
* PROGRAM SWITCHES 00410000
$MDAT1 EQU $MDUP+1 FILE ATTRIBUTE BYTE 1 00420000
$MDAT2 EQU $MDAT1+1 FILE ATTRIBUTE BYTE 2 00430000
$MDCHA EQU $MDAT2+2 DTF CHAIN POINTER - POST OPEN 00440000
$MDCHB EQU $MDCHA+2 DTF CHAIN POINTER - ALL DTFS 00450000
$MDARR EQU $MDCHB+2 ADDRESS RECALL REG SAVE AREA 00460000
$MDXR1 EQU $MDARR+2 REGISTER 1 SAVE AREA 00470000
$MDBAA EQU $MDXR1+2 CUSTOMER I/O BUFFER ADDRESS 00480000
$MDCMP EQU $MDBAA+1 COMPLETION CODE 00490000
$MDUWA EQU $MDCMP+2 USER WORK AREA 00500000
$MDLNO EQU $MDUWA+2 LINE NUMBER 00510000
$MDCMA EQU $MDLNO+2 MLTA COMMON DATA AREA ADDRESS 00520000
$MDOPC EQU $MDCMA+1 OPERATION CODE 00530000
$MDOSC EQU $MDOPC+1 OPERATION START CODE 00540000
$MDTTP EQU $MDOSC+1 TERMINAL TYPE 00550000
$MDTFT EQU $MDTTP+1 TERMINAL FEATURE TYPES 00560000
$MDTFR EQU $MDTFT+1 TERMINAL FEATURE RESERVED AREA 00570000
$MDTCT EQU $MDTFR+1 TERMINAL CODE TYPES 00580000
$MDRLM EQU $MDTCT+1 ERROR RETRY LIMIT 00590000
$MDLNF EQU $MDRLM+2 POST OPEN RECEIVE LENGTH + FLAGS 00600000
$MDBSL EQU $MDLNF-0 PRE OPEN LENGTH OF INPUT BUFFER 00610000
$MDBAL EQU $MDLNF+2 LENGTH OF USER BUFFER AREA 00620000
$MDCRL EQU $MDBAL+2 CURRENT RECORD LENGTH 00630000
$MDCRA EQU $MDCRL+2 CURRENT DATA BUFFER ADDRESS 00640000
$MDPBA EQU $MDCRA+2 PCI BUFFER ADDRESS 00650000
$MDPBL EQU $MDPBA+2 PCI RECORD LENGTH 00660000
$MDOLC EQU $MDPBL-1 PROGRAM REQUEST OLT CONTROL BYTE 00670000
$MDOLR EQU $MDPBL-0 PROGRAM REQUEST OLT START BYTE 00680000
$MDLPA EQU $MDPBL+2 LAST PCI BUFFER ADDRESS 00690000
$MDDIA EQU $MDLPA+2 TRANSFER VESTOR ADDRESS 00700000
$MDDLA EQU $MDDIA+2 RESERVED 00710000
$MDEBA EQU $MDDLA+2 ERROR RECORDING BLOCK ADDRESS 00720000
$MDEBC EQU $MDEBA+1 NUMBER OF SDR TABLE ENTRIES 00730000
*** THE $MDMOD AND $MDSNS FIELD CONTAIN THE STATUS FOR 00740000
*** FIRST ERROR ON THE LINE DURING THE OPERATION 00750000
$MDMOD EQU $MDEBC+1 MODE OF TERMINAL WHEN ERROR 00760000
* BIT FUNCTION 00770000
* 0 RECEIVE PCI OVERRUN 00780000
* 1 UPPER OR LOWER CASE 00790000
* 2 TEXT IN 00800000
* 3 TEXT OUT 00810000
* 2-3 BOTH ZERO FOR CONTROL 00820000
* 4-7 RESERVED 00830000
$MDSNS EQU $MDMOD+2 SENSE BYTES 00840000
$MDSN6 EQU $MDSNS-1 HIGH DENSITY BUFFER 6 00850000

```

```

$MDSN7 EQU $MDSNS-0 HIGH DENSITY BUFFER 7 00860000
$MDEXP EQU $MDSNS+2 EXPANSION AREA 00870000
$MDWRK EQU $MDEXP+40 THE IOCS USES THIS AREA TO STORE 00880000
* LINE DEPENDENT INFORMATION 00890000
$MDOLT EQU $MDEXP+26 BIT PATTERN FOR START OF ON-LINE 00900000
* TEST REQUEST 00910000
$MDPE0 EQU $MDOLT+2 CONTENTS OF HDB 0 <<* AFTER 00920000
$MDPER EQU $MDPE0+2 CONTENTS OF HDB 6,7 * THE 00930000
$MDPE6 EQU $MDPER-1 CONTENTS OF HDB6 * LAST 00940000
$MDPE7 EQU $MDPER CONTENTS OF HDB7 <<* INTERRUPT 00950007
*****EQUATES FOR POLLED TERMINALS ***** 00960000
$MDRSP EQU $MDWRK-4 RESPONSE FROM ADDRESSING 2740 00970000
* MOD 2 WITHOUT CHECKING 00980000
$MDCTN EQU $MDWRK+1 CURRENT TERMINALS RELATIVE NO. 00990000
* IN THE POLL OR ADDRESS LIST 01000000
$MDPLA EQU $MDCTN+2 POLLING LIST ADDRESS 01010000
$MDTMA EQU $MDPLA+2 TERMINAL ADDRESS IN LINE CODE 01020000
* READ---ADDRESS OF THE TERMIAL 01030000
* WITH LAST RESPONSE 01040000
* WRITE--ADDRESS OF TERMINAL TO 01050000
* RECEIVE MESSAGE 01060000
$MDCEA EQU $MDTMA+2 POLL LIST CURRENT ENTRY ADDRESS 01070000
* 01080000
***** EQUATES FOR ADAPTER DTF ***** 01090000
$MDERA EQU $MDCMA+2 ERROR RECOVERY ROUTINE ADDRESS 01100000
$MDUSA EQU $MDERA+2 APPLICATION PORGRAM INTERRUPT 01110000
* LEVEL PCI ROUTINE ADDRESS 01120000
$MDDEA EQU $MDUSA+2 APPLICATION PROGRAM TERMINATING 01130000
* ERROR ROUTINE ADDRESS 01140000
$MDFIL EQU $MDDEA+6 DISK FILE NAME FOR THE MLTA 01150000
* MICRO CODE 01160000
* AIF (&AT1 NE 'Y').AT2 01170000
* 01180000
* DTF FILE ATTRIBUTE BYTE 1 SWITCHES 01190000
$MAPCI EQU X'80' ALLOW PCI INTERRUPTS 01200000
* EQU X'40' RESERVED 01210000
* EQU X'20' RESERVED 01220000
$MATMI EQU X'10' ALLOW TERMINAL INTERRUPT DURING 01230000
* WRITE OPERATION. 01240000
* EQU X'08' RESERVED 01250000
$MAILT EQU X'04' INHIBIT LINE TIMEOUTS 01260000
$MATCF EQU X'02' TERMINAL CONTROL DEPENDENT FLAG 01270000
$MATCT EQU X'01' TERMINAL CONTROL TYPE 01280000
.AT2 AIF (&AT2 NE 'Y').CMP 01290000
* 01300000
* DTF FILE ATTRIBUTE BYTE 2 SWITCHES 01310000
$MAUSR EQU X'80' USER PCI INTERRUPT LEVEL 01320000
* PROCESSING REQUESTED 01330000
$MAFUL EQU X'40' PCI SWITCH TO INFORM APPLICATION 01340000
* PROGRAM OF FULL DATA BUFFER 01350000
$MAERP EQU X'20' USE IBM ERROR RECOVERY PROCEDURE 01360000
$MARET EQU X'10' RETRY OPERATION ISSUED BY ERP 01370000
* EQU X'08' USED BY IOCS 01380000
$MAOLT EQU X'04' ON LINE TEST ENABLED 01390000
$MABSY EQU X'02' LINE BUSY 01400000
$MAOPN EQU X'01' LINE OPENED 01410000
.CMP AIF (&CMP NE 'Y').OPC 01420000
* 01430000
* COMPLETION CODE EQUATES 01440000
$MCCP1 EQU X'40' NORMAL COMPLETION 01450000

```

\$MCOF1 EQU	X'41'	OPERATION FAILED, LINE IS OPEN	01460000
\$MCEOT EQU	X'42'	EOT RECEIVED ON SUCCESSFUL READ	01470000
\$MCOF2 EQU	X'43'	OPERATION FAILED AND LOOP TEST	01480000
*		FAILED, LINE IS CLOSED	01490000
\$MCTNR EQU	X'44'	SUCCESSFUL ABORT OR TRANSMIT	01500000
*		BREAK. TERMINAL DID NOT ANSWER	01510000
*		ADDRESS. END OF A NON-WRAP	01520000
*		POLL LIST.	01530000
\$MCOF3 EQU	X'45'	TRANSMISSION OR RECEPTION ABORT	01540000
*		ERROR CONDITION BUT LOOP TEST	01550000
*		RAN SUCCESSFULLY, LINE IS CLOSED	01560000
\$MCITP EQU	X'46'	TERMINAL INTERRUPT ON A WRITE	01570000
\$MCOF4 EQU	X'47'	TRANSMISSION OR RECEPTION ABORT	01580000
*		ERROR CONDITION AND LOOP TEST	01590000
*		FAILED, LINE IS CLOSED.	01600000
\$MCCP2 EQU	X'48'	APPLICATION PROGRAM NOT ABLE TO	01610000
*		PROCESS PCI FULL BUFFER BUT THE	01620000
*		TRANSMISSION COMPLETED WITHOUT	01630000
*		LOST DATA.	01640000
\$MCSDR EQU	X'49'	CURRENT TERMINAL ADDRESS NOT IN	01650000
*		SDR TABLE, LINE IS CLOSED.	01660000
\$MCCP3 EQU	X'4A'	EOT RECEIVED AND THE APPLICATION	01670000
*		PROGRAM WAS NOT ABLE TO PROCESS	01680000
*		PCI FULL BUFFER SEGMENT BUT THE	01690000
*		TRANSMISSION COMPLETED WITHOUT	01700000
*		LOST DATA.	01710000
\$MCOF5 EQU	X'4B'	APPLICATION PROGRAM NOT ABLE TO	01720000
*		PROCESS PCI FULL BUFFER IN TIME	01730000
*		AND A LOST DATA ERROR RESULTED.	01740000
\$MCCP4 EQU	X'4C'	SUCCESSFUL COMPLETION OF A READ.	01750000
*		ONLY ONE BUFFER SEGMENT RECEIVED	01760000
*		ON ERP ISSUED READ.	01770000
*		PREVIOUS BUFFER SEGMENTS SHOULD	01780000
*		NOT BE USED.	01790000
\$MCCP5 EQU	X'4E'	EOT RECEIVED ON SUCCESSFUL READ.	01800000
*		ONLY ONE BUFFER SEGMENT RECEIVED	01810000
*		ON ERP ISSUED READ.	01820000
*		PREVIOUS BUFFER SEGMENTS SHOULD	01830000
*		NOT BE USED.	01840000
\$MCCSL EQU	X'50'	INQUIRY PENDING FROM 5471	01850000
\$MCUER EQU	X'51'	OPERATION FAILED AND NO ERROR	01860000
*		RECOVERY WAS REQUESTED	01870000
\$MCOF6 EQU	X'53'	APPLICATION PROGRAM NOT ABLE TO	01880000
*		PROCESS PCI FULL BUFFER BUT NO	01890000
*		ERROR RECOVERY WAS REQUESTED	01900000
\$MCRET EQU	X'56'	NO COMPLETED EVENT FOUND IN THE	01910000
*		CHECK LIST.	01920000
\$MCSKP EQU	X'57'	ALL ENTRIES IN CHECK LIST HAVE	01930000
*		A COMBINATION OF THE FOLLOWING	01940000
*		THE DFT IS NOT BUSY.	01950000
*		THE DFT IS CLOSED.	01960000
*		A SKIP BIT ON THE CHECK LIST	01970000
\$MCPC1 EQU	X'81'	ERROR RECOVERY HAS ISSUED A READ	01980000
*		OPERATION AND MESSAGE IS BEING	01990000
*		RETRANSMITTED.	02000000
\$MCPC2 EQU	X'82'	PCI FULL BUFFER SEGMENT IS	02010000
*		READY FOR PROCESSING	02020000
.OPC AIF	(&OPC NE 'Y').OSC		02030000
*			02040000
*		OPERATION CODE EQUATES	02050000

\$MCABT	EQU	X'00'	ABORT CURRENT OPERATION	02060000
\$MCBRK	EQU	X'01'	TRANSMIT BREAK	02070000
\$MCTTO	EQU	X'02'	INTERVAL TIMEOUT	02080000
\$MCTLT	EQU	X'03'	PERFORM LOOP TEST	02090000
\$MCTBO	EQU	X'04'	PROGRAM ON LINE TEST REQUEST	02100000
\$MWTDS	EQU	X'05'	WRITE DISCONNECT	02110000
\$MCTSN	EQU	X'06'	TURN SKIP BIT ON IN POLL LIST	02120000
\$MCTSF	EQU	X'07'	TURN SKIP BIT OFF IN POLL LIST	02130000
\$MWTA	EQU	X'08'	WRITE POSITIVE ACKNOWLEDGEMENT	02140000
\$MWTN	EQU	X'09'	WRITE NEGATIVE ACKNOWLEDGEMENT	02150000
\$MRTI	EQU	X'10'	READ INITIAL	02160000
\$MRTIR	EQU	X'11'	READ INITIAL WITH RESET	02170000
\$MRTT	EQU	X'12'	READ CONTINUE	02180000
\$MRTTR	EQU	X'13'	READ CONTINUE WITH RESET	02190000
\$MRTV	EQU	X'14'	READ CONVERSATIONAL	02200000
\$MRTVR	EQU	X'15'	READ CONVERSATIONAL WITH RESET	02210000
\$MRTP	EQU	X'16'	READ REPEAT	02220000
\$MRTPR	EQU	X'17'	READ REPEAT WITH RESET	02230000
\$MRTS	EQU	X'18'	READ SKIP	02240000
\$MRTR	EQU	X'1A'	POLL REQUEST	02250000
\$MRTRR	EQU	X'1B'	POLL REQUEST WITH RESET	02260000
\$MRTB	EQU	X'1C'	READ FULL BUFFER	02270000
\$MRTBR	EQU	X'1D'	READ FULL BUFFER WITH RESET	02280000
\$MWTI	EQU	X'20'	WRITE INITIAL	02290000
\$MWTIR	EQU	X'21'	WRITE INITIAL WITH RESET	02300000
\$MWTT	EQU	X'22'	WRITE CONTINUE	02310000
\$MWTTR	EQU	X'23'	WRITE CONTINUE WITH RESET	02320000
\$MWTV	EQU	X'24'	WRITE CONVERSATIONAL	02330000
\$MWTVR	EQU	X'25'	WRITE CONVERSATIONAL WITH RESET	02340000
\$MWTL	EQU	X'26'	WRITE AT LINE ADDRESS	02350000
\$MWTLR	EQU	X'27'	WRITE AT LINE ADDRESS WITH RESET	02360000
\$MWTS	EQU	X'28'	WRITE ERASE	02370000
\$MWRTR	EQU	X'29'	WRITE ERASE WITH RESET	02380000
.OSC	AIF	(&OSC NE 'Y').TFT		02390000
*				02400000
*		OPEN/CLOSE COMPLETION CODES		02410000
\$MOCPL	EQU	X'00'	OPEN COMPLETED OK	02420000
\$MOOPL	EQU	X'84'	OUTSIDE PARTITION LIMITS	02430000
\$MOEXT	EQU	X'88'	UPSI BYTE DOES NOT MATCH	02440000
*			EXTERNAL SWITCH INDICATORS	02450000
\$MOILN	EQU	X'8C'	INVALID LINE NUMBER	02460000
\$MORTS	EQU	X'90'	RECEIVE WORK AREA TOO SMALL	02470000
\$MOIEX	EQU	X'94'	INVALID EXTRN TABLE FOR THE	02480000
*			TERMINAL FEATURE SPECIFIED	02490000
\$MOERP	EQU	X'98'	THE APPLICATION SPECIFIED	02500000
*			FULL BUFFER EXIT OR STANDARD	02510000
*			ERP RECOVERY BUT THE ADDRESS IN	02520000
*			THE ADAPTER DTF WAS X'FFFF'	02530000
\$MOLAO	EQU	X'9C'	LINE ALREADY OPEN	02540000
\$MOBTS	EQU	X'A0'	LOGICAL RECORD LENGTH L.T. 16	02550000
\$MOENL	EQU	X'A4'	COULD NOT ENABLE LINE	02560000
\$MODSL	EQU	X'A8'	COULD NOT DISABLE LINE	02570000
\$MOAMO	EQU	X'AC'	ADAPTER NOT OPEN AT LINE OPEN	02580000
*				02590000
*			OPERATION START CODES	02600000
*	EQU	X'00'	OPERATION STARTED OK	02610000
\$MONOP	EQU	X'04'	LINE DTF NOT OPEN	02620000
\$MOBSY	EQU	X'08'	LINE BUSY	02630000
\$MOIOP	EQU	X'0C'	INVALID OPERATION CODE	02640000
\$MOUOP	EQU	X'10'	UNSUPPORTED OPERATION CODE	02650000

\$MOASO EQU	X'14'	ALL POLL SKIP SWITCHES ARE ON	02660000
\$MOCTE EQU	X'18'	CURRENT TERMINAL NO. TOO LARGE	02670000
\$MOABE EQU	X'1C'	CABT OR CBRK NOT ACCEPTED	02680000
\$MOABN EQU	X'20'	CABT OR CBRK NOT DONE	02690000
\$MOBDC EQU	X'24'	NON-MLTA DTF SPECIFIED IN	02700000
*		DEVICE CODE	02710000
\$MOZER EQU	X'28'	ZERO LENGTH SPECIFIED FOR WRITE	02720000
\$MOSKP EQU	X'2C'	THE CTSF/CTSN OPERATION DID NOT	02730000
*		FIND A MATCHING ENTRY	02740000
\$MOBSE EQU	X'2E'	CTBO NOT STARTED BUFFER TOO SMALL	02750000
*		OR SWITCHED LINE NOT CONNECTED	02760000
\$MOIIS EQU	X'32'	INVALID INSTRUCTION SEQUENCE	02770000
.TFT AIF	(&TFT NE 'Y').SNS		02780000
*			02790000
*		DTF TERMINAL FEATURES BYTE 1	02800000
\$MTTRC EQU	X'80'	TRANSMIT CONTROL FEATURE	02810000
\$MTINT EQU	X'40'	TRANSMIT INTERRUPT FEATURE	02820000
\$MTLRC EQU	X'20'	LONGITUDINAL RECORD CHECKING	02830000
\$MMDNT EQU	X'10'	SWITCHED LINE FEATURE	02840000
\$MTBPS EQU	X'08'	LINE SPEED OF 600 BPS	02850000
\$MTPLT EQU	X'04'	STATION CONTROL FEATURE	02860000
\$MMAUP EQU	X'01'	MLTA HAS AUTOPOLL	02870000
*			02880000
*		DTF TERMINAL FEATURES BYTE 2	02890000
\$MTBFR EQU	X'80'	BUFFER RECEIVE	02900000
\$MTRSP EQU	X'40'	TEST SECOND BYTE OF ADDRESSING	02910000
*		RESPONSE FOR BUFFERED TERMINAL	02920000
.SNS AIF	(&SNS NE 'Y').MEND		02930000
*			02940000
*		DTF SENSE BYTE HDB0	02950000
\$MMCTL EQU	X'30'	BIT PATTERN FOR ALL OF CONTROL	02960000
*		FIELD BITS OFF IS CONTROL MODE	02970000
\$MMIN EQU	X'20'	TEXT-IN MODE	02980000
\$MMOUT EQU	X'10'	TEXT-OUT MODE	02990000
*			03000000
*		DTF SENSE BYTE FOR HDB6	03010000
*			03020000
\$MSOUT EQU	X'80'	TIMEOUT	03030000
\$MSDCK EQU	X'40'	DATA CHECK	03040000
\$MSTAB EQU	X'20'	TRANSMISSION ABORTED	03050000
\$MSRAB EQU	X'10'	RECEPTION ABORTED	03060000
\$MSOVR EQU	X'08'	OVERRUN	03070000
\$MSINT EQU	X'04'	TERMINAL INTERRUPT	03080000
\$MSNOP EQU	X'02'	INSTRUCTION NO-OP	03090000
\$MSLDT EQU	X'01'	LOST DATA	03100000
*			03110000
*		DTF SENSE BYTE FOR HDB7	03120000
*			03130000
\$MSMNR EQU	X'80'	MODEM NOT READY	03140000
\$MSLNR EQU	X'10'	LINE NOT READY	03150000
.MEND ANOP			03160000
*			* 03170000
*		END OF DTF EQUATES FOR THE MLTA	* 03180000
*			* 03190000
*****			03200000
AIF	(&\$MG EQ '0').DONE		03210000
PRINT ON			03220000
.DONE MEND			03230000

MODULE-\$CG1GM, VOLUME ID-R2R2R2, DATE-06/06/10

\$EFAC	ESCAPE-'////////',	-- 'CCCCC' / X'XXXXXXXXXXXX' --	X00010000
	PGMCNT-YES,	-- NO --	X00020000
	FORMAT-YES,	-- NO --	X00030000
	PRUF-NO,	-- YES --	X00030100
	ACCEPT-NO,	-- YES --	X00034100
	BSYPRT-NO,	-- YES --	X00040000
	CPUMSG-NO	-- YES --	00045000
\$EPLG	LANG-RPGII,	-- COBOL / RPGII / FORTRAN / ASSEM --	X00050000
	PPUNIT-R2	-- R1 / F1 / R2 / F2 --	00060000
\$EPLG	LANG-COBOL,	-- COBOL / RPGII / FORTRAN / ASSEM --	X00070000
	PPUNIT-R2	-- R1 / F1 / R2 / F2 --	00080000
\$ESEC	SECURE-CCP	-- CCP / USER --	00090000
\$EFIL	SETS-3,	-- 1 - 25 --	X00100000
	PROGS-15,	-- 1 - 255 --	X00110000
	DFILES-10,	-- 1 - 50 --	X00120000
	TERMS-5,	-- 2 - 254 --	X00130000
	DUMPS-2,	-- 2 - 9 --	X00140000
	CORE-64K,	--48K/64K/96K/128K/160K/192K/224K/256K	X00150000
	FLPACK-R2R2R2,	-- NAME OF PACK --	X00160000
	FLUNIT-R2	-- R1 / F1 / R2 / F2 --	00170000
\$EBSC	BSCA-1,	-- 0 - 2 --	X00180000
	DA-NO,	-- YES --	X00190000
	DIAL-NO,	-- YES --	X00200000
	PP-NO,	-- YES --	X00210000
	MP-NO,	-- YES --	X00220000
	CS-YES,	-- NO --	X00230000
	GETMSG-YES,	-- NO --	X00240000
	ITB-NO,	-- YES --	X00250000
	RECSEP-1E,	-- TWO HEX DIGITS --	X00260000
	ASCII-NO,	-- YES --	X00270000
	EBCDIC-YES,	-- NO --	X00280000
	XPRNCY-NO,	-- YES --	X00290000
	RESPOL-NO,	-- YES --	X00300000
	INTPOL-YES,	-- NO --	X00310000
	AUTORS-NO	-- YES --	00320000
\$EBSD	TYPE-3277M2	-- SEE SYSTEM REFERENCE MANUAL --	00330000
\$EGEN	DSUNIT-F1,	-- R1 / F1 --	X00340000
	CCUNIT-R2,	-- R1 / F1 / R2 / F2 --	X00350000
	WKUNIT-'R2,R1,R1',	-- 'UNIT,UNIT,UNIT' --	X00360000
	WKPACK-'R2R2R2,PID001,PID001',	-- 'PACK,PACK,PACK' --	X00370000
	DIUNIT-R1,	-- R1 / F1 / R2 / F2 -- CCP-PID	X00380000
	MINRES-NO,	-- YES --	X00390000
	CARD-NO	-- YES --	00400000

```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 8 * 00030000
.* * 00040000
.* NAME:  $EBEG * 00050000
.* * 00060000
.* FUNCTION:  THE $EBEG MACRO IS USED BY CCP TRANSIENTS TO DEFINE * 00070000
.* THE FIXED FORMAT OF THE BEGINNING OF THE TRANSIENT.  IT * 00080000
.* GENERATES THE START CARD WITH THE NAME FIELD AND THE START * 00090000
.* ADDRESS FOR ALL TRANSIENTS.  CREATES A TWO-BYTE ID(LAST TWO * 00100000
.* CHARACTERS OF THE TRANSIENT'S NAME) AND A ONE-BYTE LEVEL * 00110000
.* NUMBER).  GENERATES A JUMP AROUND THE C/S/N TABLE OF ALL * 00120000
.* TRANSIENTS CALLED.  THIS MACRO MUST BE THE FIRST STATEMENT THAT * 00130000
.* PRODUCES CODE IN ANY TRANSIENT.  BY CONVENTION THE TRANSIENT * 00140000
.* AREA SCHEDULER WILL ENTER AT OFFSET ZERO WHENEVER PASSING * 00150000
.* CONTROL TO A NEW TRANSIENT. * 00160000
.* * 00170000
.* * 00180000
.* INPUT OPERANDS: * 00190000
.* * N- LAST TWO CHARACTERS OF THE TRANSIENT'S NAME. * 00200000
.* * TID-1 TRANSIENT AREA 1.  GENERAL TRANSIENT AREA. * 00210000
.* 2 TRANSIENT AREA 2.  CM'S TRANSIENT AREA. * 00220000
.* * TR-Y/N ONLY GENERATES OFFSET DEFINITIONS IF Y. * 00230000
.* * * * * DEFAULT IS N. * 00240000
.* * JUMP- A LABEL CAN BE SUPPLIED TO MODIFY THE DEFAULT * 00250000
.* * * * * JUMP DISPLACEMENT. * 00260000
.* * XN- UPTO FIFTEEN TRANSIENT ID'S CAN BE PLACED * 00270000
.* * * * * IN A TABLE BY THIS MACRO.  'N' IS SUBSTITUTED * 00280000
.* * * * * BY THE NUMBERS 1-15. * 00290000
.* * MOD- MODIFICATION LEVEL OF THIS MODULE.  CAUSES A * 00300000
.* * * * * ONE-BYTE DC TO BE GENERATED, THE VALUE IN THE * 00310000
.* * * * * DC BEING THE DATA SPECIFIED. * 00320000
.* * * * * * 00330000
.* NOTES: * 00340000
.* * ALL $CC4 TRANSIENTS MUST USE THIS MACRO.  STARTUP WILL USE * 00350000
.* * THE $ DELIMITER TO DETERMINE THE NUMBER OF TRANSIENTS * 00360000
.* * FOR WHICH HE MUST FIND C/S/N VALUES.  THE ASSIGNMENT OF * 00370000
.* * THE N-VALUE WILL ASSURE THAT ONLY THE NUMBER OF SECTORS * 00380000
.* * ACTUALLY NEEDED BY A TRANSIENT ARE LOADED. * 00390000
.* * THE FIELD TAXPRM IS STILL PRESENT.  IN ADDITION, SV1TAX * 00400000
.* * FOR TRANSIENT AREA ONE, AND SV2TAX FOR CM'S TRANSIENT * 00410000
.* * AREA HAVE BEEN DEFINED IN THE $ECOM MACRO. * 00420000
.* ***** 00430000
.* $EBEG &N-, &TID-1, &TR-N, &JUMP-, &X1-, &X2-, &X3-, &X4-, &X5-, &X6-, X00430000
.* &X7-, &X8-, &X9-, &X10-, &X11-, &X12-, &X13-, &X14-, &X15-, &MOD-N 00440000
.* TABLE &TID 00450000
1 TABDF 16384 00460000
2 TABDF 16896 00470000
.* TEXT 00480000
.* AIF (&TR EQ 'Y').TR 00490000
$CC4&N START &TID TRANSIENT START DEFINITION 00500000
$CC4$$ EQU * $CC4$$ DEFINITION 00510000
.* LEVEL 08 00520000
.* RLD N 00530000
.* SPACE 2 00540000
.* AIF (T'&JUMP EQ 'O').BEG10 00550000
.* J &JUMP GO EXECUTE 1ST INSTRUCTION 00560000
.* AGO .XN 00570000
.*BEG10 ANOP 00580000
.* J &N.GO BYPASS PHASE ID'S, C/S'S 00590000

```

.XN	ANOP			00600000
	USING	\$CC4&N.,1	DEFINE BASE ON TRANSIENT ENTRY	00610000
TAXPRM	DC	XL2'0000'	TRANSIENT COMMUNICATION AREA	00620000
	AIF	(T'&X1 NE 'O').TID		00630000
TAXTID	DC	XL2'0000'	POSITION HOLDER	00640000
	AGO	.BYXN		00650000
.TID	ANOP			00660000
TAXTID	DC	AL2(&N&X1)	INITIALLY 1ST CSN DISPLACEMENT	00670000
&N&X1	DC	CL5'&X1'	PHASE ID, SPACE FOR C/S/N	00680000
	AIF	(T'&X2 EQ 'O').BYXN		00690000
&N&X2	DC	CL5'&X2'	PHASE ID, SPACE FOR C/S/N	00700000
	AIF	(T'&X3 EQ 'O').BYXN		00710000
&N&X3	DC	CL5'&X3'	PHASE ID, SPACE FOR C/S/N	00720000
	AIF	(T'&X4 EQ 'O').BYXN		00730000
&N&X4	DC	CL5'&X4'	PHASE ID, SPACE FOR C/S/N	00740000
	AIF	(T'&X5 EQ 'O').BYXN		00750000
&N&X5	DC	CL5'&X5'	PHASE ID, SPACE FOR C/S/N	00760000
	AIF	(T'&X6 EQ 'O').BYXN		00770000
&N&X6	DC	CL5'&X6'	PHASE ID, SPACE FOR C/S/N	00780000
	AIF	(T'&X7 EQ 'O').BYXN		00790000
&N&X7	DC	CL5'&X7'	PHASE ID, SPACE FOR C/S/N	00800000
	AIF	(T'&X8 EQ 'O').BYXN		00810000
&N&X8	DC	CL5'&X8'	PHASE ID, SPACE FOR C/S/N	00820000
	AIF	(T'&X9 EQ 'O').BYXN		00830000
&N&X9	DC	CL5'&X9'	PHASE ID, SPACE FOR C/S/N	00840000
	AIF	(T'&X10 EQ 'O').BYXN		00850000
&N&X10	DC	CL5'&X10'	PHASE ID, SPACE FOR C/S/N	00860000
	AIF	(T'&X11 EQ 'O').BYXN		00870000
&N&X11	DC	CL5'&X11'	PHASE ID, SPACE FOR C/S/N	00880000
	AIF	(T'&X12 EQ 'O').BYXN		00890000
&N&X12	DC	CL5'&X12'	PHASE ID, SPACE FOR C/S/N	00900000
	AIF	(T'&X13 EQ 'O').BYXN		00910000
&N&X13	DC	CL5'&X13'	PHASE ID, SPACE FOR C/S/N	00920000
	AIF	(T'&X14 EQ 'O').BYXN		00930000
&N&X14	DC	CL5'&X14'	PHASE ID, SPACE FOR C/S/N	00940000
	AIF	(T'&X15 EQ 'O').BYXN		00950000
&N&X15	DC	CL5'&X15'	PHASE ID, SPACE FOR C/S/N	00960000
.BYXN	ANOP			00970000
	DC	CL3'\$&N'	EYECATCHER CONSTANT	00980000
	DC	XL1'08'	RELEASE LEVEL	00990000
	AIF	(&MOD EQ 'N').GO		01000000
	DC	AL1(&MOD)	MODIFICATION LEVEL	01010000
.GO	ANOP			01020000
&N.GO	EQU	*	END OF \$EBEG CONSTANTS	01030000
	AGO	.END		01040000
.TR	ANOP			01050000
TAXPRM	EQU	4	TRANSIENT COMMUNICATION AREA	01060000
TAXTID	EQU	6	OFFSET TO CSN VALUE OF THE	01070000
*			TRANSIENT BEING CALLED.	01080000
*			DEFAULTED TO FIRST IN TABLE	01090000
TAXCID	EQU	7	OFFSET TO START OF CSN TABLE	01100000
TAXNCS	EQU	11	OFFSET TO FIRST CSN PARAMETER	01110000
TAXCLN	EQU	5	LENGTH OF A CSN TABLE ENTRY	01120000
.END	ANOP			01130000
	MEND			01140000

MODULE-\$ECFG , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 3 * 00030000
.* * 00040000
.* NAME:  $ECFG * 00050000
.* * 00060000
.* FUNCTION:  DEFINE THE LABELS NEEDED TO REFERENCE THE CCP * 00070000
.* CONFIGURATION RECORD FOUND IN $CCPFILE. * 00080000
.* * 00090000
.* INPUT OPERANDS:  NONE * 00100000
***** 00110000
$ECFG 00120000
TEXT 00130000
***** 00140000
* C C P C O N F I G U R A T I O N R E C O R D O F F S E T S * 00150000
***** 00160000
CFGCCP EQU 0 00170000
CFGVRF EQU CFGCCP+1 VERIFICATION BYTES, X'EBD7' 00180000
SPACE 00190000
CFGRV1 EQU CFGVRF+6 RESERVED 00200000
SPACE 00210000
CFGFA2 EQU CFGRV1+1 CCP FACILITIES 00220000
CFGSYM EQU BIT0 1=SYMBOLIC FILES ALLOWED 00230000
CFGPUC EQU BIT1 1=PGM REQUEST-COUNT KEPT 00240000
CFGTF EQU BIT2 RESERVED 00250000
CFGSEC EQU BIT3+BIT4 00=NO SIGN-ON SECURITY 00260000
CFGCPW EQU BIT3 10=CCP SUPPLIED PASSWORD RTN 00270000
CFGUPW EQU BIT4 01=USER SUPPLIED PASSWORD RTN 00280000
CFGDME EQU BIT5 1=DATA MODE ESCAPE ALLOWED 00290000
CFGFMT EQU BIT6 1=3270 DISPLAY FORMATTING ALLOWD 00300000
CFGGRUF EQU BIT7 1=READ UNDER FORMAT SUPPORTED 00310000
SPACE 00320000
CFGRV2 EQU CFGFA2+2 RESERVED 00330000
SPACE 00340000
CFGPLG EQU CFGRV2+1 PROGRAMMING LANGUAGE SUPPORT 00350000
CFGCOB EQU BIT0 1=COBOL 00360000
CFGFOR EQU BIT1 1=FORTRAN 00370000
CFGASM EQU BIT2 1=ASSEMBLER 00380000
CFGRPG EQU BIT3 1=RPGII 00390000
SPACE 00400000
CFGCBO EQU CFGPLG+2 OFFSET TO RIGHT END OF CCP 00410000
* BRANCH INSTRUCTION FROM START OF 00420000
* CONSOLE INTERRUPT HANDLER 00430000
SPACE 00440000
CFGESC EQU CFGCBO+6 DATA MODE ESCAPE BYTE STRING 00450000
SPACE 00460000
CFGLUS EQU CFGESC+2 LENGTH OF USER SECURITY INFO 00470000
SPACE 00480000
CFGRV3 EQU CFGLUS+4 RESERVED 00490000
SPACE 00500000
CFGNS EQU CFGRV3+1 MAX NO. SETS PLANNED 00510000
SPACE 00520000
CFGNPM EQU CFGNS+1 MAX NO. PROGRAMS IN A SET 00530000
SPACE 00540000
CFGNDF EQU CFGNPM+1 MAX NO. DISK FILES IN A SET 00550000
SPACE 00560000
CFGNT EQU CFGNDF+1 MAX NO. TERMINALS IN A SET 00570000
SPACE 00580000
CFGDMP EQU CFGNT+1 MAX NO. DYNAMIC CORE DUMPS 00590000
```

CFGSC	EQU	CFGDMP+2	MEMORY # SECTORS(0-1024)	00600000
	SPACE			00610000
CFGRV4	EQU	CFGSC+3	RESERVED	00620000
	SPACE			00630000
CFGTP	EQU	CFGRV4+1	TP ADAPTER INFORMATION	00640000
CFGMLA	EQU	BIT0+BIT1+BIT2+BIT3	NUMBER OF MLTA LINES	00650000
CFGBSC	EQU	BIT4+BIT5+BIT6+BIT7	NUMBER OF BSCA LINES	00660000
	SPACE			00670000
CFGMD1	EQU	CFGTP+1	MLTA DEVICE SUPPORT	00680000
CFG50	EQU	BIT0	1=1050	00690000
CFG50D	EQU	BIT1	1=1050D	00700000
CFG40	EQU	BIT2	1=2740	00710000
CFG40S	EQU	BIT3	1=2740S	00720000
CFG40C	EQU	BIT4	1=2740C	00730000
CFG4SC	EQU	BIT5	1=2740SC	00740000
CFG40D	EQU	BIT6	1=2740D	00750000
CFG4DT	EQU	BIT7	1=2740DT	00760000
	SPACE			00770000
CFGMD2	EQU	CFGMD1+1	MLTA DEVICE SUPPORT	00780000
CFG4DC	EQU	BIT0	1=2740DC	00790000
CFGDTC	EQU	BIT1	1=2740DTC	00800000
CFG2S	EQU	BIT2	1=2740M2S	00810000
CFG2SB	EQU	BIT3	1=2740M2SB	00820000
CFG2SC	EQU	BIT4	1=2740M2SC	00830000
CFGSCB	EQU	BIT5	1=2740M2SCB	00840000
CFG41	EQU	BIT6	1=2741	00850000
CFG41D	EQU	BIT7	1=2741D	00860000
	SPACE			00870000
CFGMD3	EQU	CFGMD2+1	MLTA DEVICE SUPPORT	00880000
CFG7C	EQU	BIT0	1=SYSTEM/7 AS 2740C	00890000
CFG7SC	EQU	BIT1	1=SYSTEM/7 AS 2740SC	00900000
CFG7DC	EQU	BIT2	1=SYSTEM/7 AS 2740DC	00910000
CFGCMC	EQU	BIT3	1=CMCST ON DIAL LINE	00920000
CFG705	EQU	BIT4	1=2970M5	00930000
CFG708	EQU	BIT5	1=2970M8	00940000
CFG709	EQU	BIT6	1=2970M9	00950000
CFG79S	EQU	BIT7	1=2970M9S	00960000
	SPACE			00970000
CFGMD4	EQU	CFGMD3+1	MLTA DEVICE SUPPORT	00980000
CFG30	EQU	BIT0	1=5930	00990000
	SPACE			01000000
CFGRV5	EQU	CFGMD4+1	RESERVED	01010000
	SPACE			01020000
CFGMXC	EQU	CFGRV5+1	MLTA TRANSMISSION CODES	01030000
CFGMCO	EQU	BIT0	1=CORRESPONDENCE CODE	01040000
CFGPEB	EQU	BIT1	1=PTTCBCD CODE	01050000
CFGPBC	EQU	BIT2	1=PTTCBCD CODE	01060000
CFGSPB	EQU	BIT3	1=PTTCBCD 1050 CODE	01070000
CFGXBF	EQU	BIT7	1=MLTA TRANSLATION REQUIRED	01080000
*			0=MLTA TRANSLATION OPTIONAL	01090000
	SPACE			01100000
CFGRV6	EQU	CFGMXC+2	RESERVED	01110000
	SPACE			01120000
CFGBLT	EQU	CFGRV6+1	BSCA LINE-TYPE SUPPORT	01130000
CFGBPP	EQU	BIT0	1=POINT TO POINT	01140000
CFGBMP	EQU	BIT1	1=MULTIPOINT	01150000
CFGBCS	EQU	BIT2	1=CONTROL STATION	01160000
CFGBD	EQU	BIT3	1=DIAL (SWITCHED) LINE	01170000
	SPACE			01180000
				01190000

CFGBFA	EQU	CFGBLT+1	BSCA FACILITIES	01200000
CFGGET	EQU	BIT0	1=GET MESSAGE	01210000
CFGITB	EQU	BIT1	1=INTERMEDIATE TEXT BLOCKS	01220000
CFGSEP	EQU	BIT2	1=RECORD SEPARATORS	01230000
CFGRSP	EQU	BIT3	1=RESIDENT POLLING	01240000
CFGARS	EQU	BIT4	1=AUTO RESPONSE	01250000
CFGELC	EQU	BIT5	1=EBCDIC LINE CODE	01260000
CFGALC	EQU	BIT6	1=ASCII LINE CODE	01270000
CFGBXP	EQU	BIT7	1=BSCA TRANSPARENCY	01280000
		SPACE		01290000
CFGBF1	EQU	CFGBFA+1	BSCA FACILITIES CONTINUED	01300000
CFGINT	EQU	BIT0	1=INTERVAL POLLING	01310000
		SPACE		01320000
CFGRV7	EQU	CFGBF1+1	RESERVED	01330000
		SPACE		01340000
CFGRSB	EQU	CFGRV7+1	RECORD SEPARATOR BYTE	01350000
		SPACE		01360000
CFGBD1	EQU	CFGRSB+1	BSCA DEVICE-TYPE SUPPORT	01370000
CFG351	EQU	BIT0	1=3275 MODEL 1	01380000
CFG371	EQU	BIT1	1=3277 MODEL 1	01390000
CFG341	EQU	BIT2	1=3284 MODEL 1	01400000
CFG361	EQU	BIT3	1=3286 MODEL 1	01410000
CFG352	EQU	BIT4	1=3275 MODEL 2	01420000
CFG372	EQU	BIT5	1=3277 MODEL 2	01430000
CFG342	EQU	BIT6	1=3284 MODEL 2	01440000
CFG362	EQU	BIT7	1=3286 MODEL 2	01450000
		SPACE		01460000
CFGBD2	EQU	CFGBD1+1	BSCA DEVICE-TYPE SUPPORT	01470000
CFG375	EQU	BIT0	1=3735	01480000
CFGCPU	EQU	BIT1	1=CPU	01490000
CFG741	EQU	BIT2	1=3741 (AS A TERMINAL)	01500000
		SPACE		01510000
CFGEND	EQU	CFGBD2	END OF CONFIGURATION RECORD	01520000
		SPACE		01530000
CFGLEN	EQU	CFGEND+1	LENGTH OF CONFIGURATION RECORD	01540000
		SPACE		01550000
CFGFLG	EQU	X'FF'	UNUSED PORTION OF SECTOR	01560000
		MEND		01570000


```
MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 5 * 00030000
.* * 00040000
.* NAME:  $ECRL * 00050000
.* * 00060000
.* FUNCTION:  DEFINE THE NAMES OF ALL '$CC4 ' MODULES THAT MUST BE * 00070000
.* CROSS-REFERENCED(CYLINDER/SECTOR/NUMBER OF SECTORS) BY STARTUP. * 00080000
.* * 00090000
.* INPUT OPERANDS:  NONE * 00100000
***** 00110000
$ECRL 00120000
TEXT 00130000
***** 00140000
* TRANSIENT CROSS REFERENCE LIST * 00150000
***** 00160000
SPCRL EQU * TWO CHARACTER ID LIST FOLLOWS 00170000
DC CL2 'AQ' TERMINAL ACQUIRE 00180000
DC CL2 'AS' ASSIGN COMMAND 00190000
DC CL2 'AT' SECOND ASSIGN 00200000
DC CL2 'AU' SECOND ASSIGN TRANSIENT 00210000
DC CL2 'AX' SECOND TERMINAL ACQUIRE 00220000
DC CL2 'A1' ALLOCATION CONTROL TRANSIENT 00230000
DC CL2 'A2' ALLOCATION DIAGNOSTICS INTERFACE 00240000
DC CL2 'BA' BSCA SENSE STATUS--3270 00250000
DC CL2 'BC' BSCA CANCEL/STOP INVITE. 00260000
DC CL2 'BD' BSCA ERP CLEAN-UP 00270000
DC CL2 'BE' BSCA ERROR RECOVERY. 00280000
DC CL2 'BF' BSCA FREEMAIN OF INVITE BUFFER 00290000
DC CL2 'BL' BSCA ERROR LOG TO OPERATOR 00300000
DC CL2 'BP' BSCA PURGE/STOP II TRANSIENT. 00310000
DC CL2 'BQ' BSCA 00320000
DC CL2 'BU' BSCA 00330000
DC CL2 'B1' MINIMUM SYSTEM XIENT ONE 00340000
DC CL2 'B5' 3735 STATUS LOG 00350000
DC CL2 'B7' 3741 SUPPORT 00355000
DC CL2 'B9' BSCA DUMP 00360000
DC CL2 'CG' CONSOLE MESSAGE COMMAND 00370000
DC CL2 'CJ' SYSTEM OPERATOR CANCEL 00380000
DC CL2 'CN' TERMINAL OPERATOR CANCEL 00390000
DC CL2 'DB' D.F.F. 00400000
DC CL2 'DM' DATA MODE ESCAPE 00410000
DC CL2 'EB' ALLOCATE MESSAGE MODULE 00420000
DC CL2 'EN' ALLOCATE MESSAGE MODULE 00430000
DC CL2 'FL' FILE COMMAND PROCESSOR 00440000
DC CL2 'F2' 00450000
DC CL2 'GA' GET ATTRIBUTES 00460000
DC CL2 'L2' CLOSE 00470000
DC CL2 'L3' CLOSE 00480000
DC CL2 'L4' CLOSE 00490000
DC CL2 'MA' MLTA ERROR HANDLER 00500000
DC CL2 'MB' MLTA SWITCHED LINE DISCONNECT 00510000
DC CL2 'MC' MLTA ERP FOR BUFFERED RCV TERMS 00520000
DC CL2 'MD' MLTA ERROR HANDLER 00530000
DC CL2 'MG' CMD PROC MESSAGE ROUTINE T-P 00540000
DC CL2 'MZ' MLTA ONLINE TEST 00550000
DC CL2 'M9' MLTA DUMP 00560000
DC CL2 'NM' NAME COMMAND PROCESSOR 00570000
DC CL2 'OF' 00580000
```

DC	CL2'OP'	OPEN/CLOSE	00590000
DC	CL2'OT'	PSEUDO TAPE OPEN	00600000
DC	CL2'O1'	OPEN	00610000
DC	CL2'O2'	OPEN	00620000
DC	CL2'O3'	OPEN	00630000
DC	CL2'O4'	OPEN	00640000
DC	CL2'O5'	OPEN	00650000
DC	CL2'O6'	OPEN	00660000
DC	CL2'O7'	OPEN	00670000
DC	CL2'O8'	OPEN	00680000
DC	CL2'O9'	OPEN	00690000
DC	CL2'PC'	CMD PROC FIRST LOAD	00700000
DC	CL2'PG'	MLTA PURGE I/O	00710000
DC	CL2'PK'	CP CONSOLE CMD VALIDATOR	00720000
DC	CL2'PM'	TIME COMMAND PROCESSOR	00725000
DC	CL2'PT'	C P TERM CMD VALIDATOR	00730000
DC	CL2'QQ'	/Q, /NOQ	00740000
DC	CL2'RE'	RESUME COMMAND	00750000
DC	CL2'RF'	RESUME TRANSIENT # 2	00760000
DC	CL2'RL'	FIRST RELEASE MODULE	00770000
DC	CL2'RN'	RUN MODULE	00780000
DC	CL2'RP'	T-P ERROR RECOVERY ROUTINE	00790000
DC	CL2'RX'	SECOND RELEASE MODULE	00800000
DC	CL2'R1'		00810000
DC	CL2'R2'		00820000
DC	CL2'R3'		00830000
DC	CL2'R4'		00840000
DC	CL2'R5'		00850000
DC	CL2'R6'	SIXTH PGM REQUEST ROUTINE	00860000
DC	CL2'SH'	SHUTDOWN COMMAND	00870000
DC	CL2'SK'	MLTA-SET ALL POLL SKIP BITS ON	00880000
DC	CL2'SO'	TERMINAL SIGN ON CMD PROCESSOR	00890000
DC	CL2'SP'	MLTA USER STOP POLLING	00900000
DC	CL2'SS'	SUSPEND COMMANDS ROUTINE	00910000
DC	CL2'S2'	STOP INVITE/PURGE	00915000
DC	CL2'TD'	1ST TERMINATION ROUTINE	00920000
DC	CL2'TE'	TRACE TO DISK COMMAND	00930000
DC	CL2'TF'	TERMINATION FILES DEALLOCATION	00940000
DC	CL2'TG'	TERMINATION	00950000
DC	CL2'TH'	TERMINATION TRANSIENT	00960000
DC	CL2'TJ'	TERMINATION DUMP DISK I/O	00970000
DC	CL2'TK'	TERMINATION DUMP MSG ROUTINE	00980000
DC	CL2'TN'	2ND TERMINATION ROUTINE	00990000
DC	CL2'TP'	TERMINATION POLLING RESTART # 1	01000000
DC	CL2'TS'	ONLINE TEST COMMAND FROM CONSOLE	01010000
DC	CL2'TW'	TERMINATION DUMP SETUP	01020000
DC	CL2'TY'	T-P POLLING RESTART # 2	01030000
DC	CL2'TZ'	BSCA OLTS	01040000
DC	CL2'T2'	MLTA ONLINE TEST OP END	01050000
DC	CL2'VA'	VARY COMMAND	01060000
DC	CL2'VB'	SECOND VARY CMD. PROCESSOR	01070000
DC	CL2'WC'	SWITCHED LINE MSG TO SYS OP	01080000
DC	CL2'WR'	TRANSLATE ERROR TRANSIENT	01090000
DC	CL2'YA'	USER FIRST SIGN-ON TRANSIENT	01100000
DC	CL2'YB'	USER TRANSIENT	01110000
DC	CL2'YC'	USER TRANSIENT	01120000
DC	CL2'YD'	USER TRANSIENT	01130000
DC	CL2'YE'	USER TRANSIENT	01140000
DC	CL2'YF'	USER TRANSIENT	01150000
DC	CL2'YG'	USER TRANSIENT	01160000

DC	CL2 'YH'	USER TRANSIENT	01170000
DC	CL2 'YI'	USER TRANSIENT	01180000
DC	CL2 'YJ'	USER TRANSIENT	01190000
DC	CL2 'YK'	USER TRANSIENT	01200000
DC	CL2 'YL'	USER TRANSIENT	01210000
DC	CL2 'YM'	USER TRANSIENT	01220000
DC	CL2 'YN'	USER TRANSIENT	01230000
DC	CL2 'YO'	USER TRANSIENT	01240000
DC	CL2 'YP'	USER TRANSIENT	01250000
DC	CL2 'YQ'	USER TRANSIENT	01260000
DC	CL2 'YR'	USER TRANSIENT	01270000
DC	CL2 'YS'	USER TRANSIENT	01280000
DC	CL2 'YT'	USER TRANSIENT	01290000
DC	CL2 'YU'	USER TRANSIENT	01300000
DC	CL2 'YV'	USER TRANSIENT	01310000
DC	CL2 'YW'	USER TRANSIENT	01320000
DC	CL2 'YX'	USER TRANSIENT	01330000
DC	CL2 'YY'	USER TRANSIENT	01340000
DC	CL2 'YZ'	USER TRANSIENT	01350000
DC	CL2 'Y0'	USER TRANSIENT	01360000
DC	CL2 'Y1'	USER TRANSIENT	01370000
DC	CL2 'Y2'	USER TRANSIENT	01380000
DC	CL2 'Y3'	USER TRANSIENT	01390000
DC	CL2 'Y4'	USER TRANSIENT	01400000
DC	CL2 'Y5'	USER TRANSIENT	01410000
DC	CL2 'Y6'	USER TRANSIENT	01420000
DC	CL2 'Y7'	USER TRANSIENT	01430000
DC	CL2 'Y8'	USER TRANSIENT	01440000
DC	CL2 'Y9'	USER TRANSIENT	01450000
DC	CL2 'Y\$'	USER TRANSIENT	01460000
DC	CL2 'Y@'	USER TRANSIENT	01470000
DC	CL2 'Y#'	USER TRANSIENT	01480000
DC	20XL2'0000'	SPARE SPACE FOR ADDITIONAL NAMES	01490000
DC	XL1 'FF'	END OF ID LIST	01500000
SPACE			01510000
MEND			01520000

```

MODULE-$EHLF , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
$EHLF                                00020000
TEXT                                  00030000
*FOLLOWING ARE SOME OF THE COMMONLY USED REGISTER EQUATES. 00040000
SPACE                                00050000
TCB EQU 1 USE R1 FOR TCB 00060000
TUB EQU 2 USE R2 FOR TUB 00070000
PTX EQU 2 USE R2 FOR TUB 00080000
PCT EQU 1 00090000
SDF EQU 2 00100000
TBL EQU 2 00110000
NC EQU 2 USE R2 FOR SYSCOM REFERENCES. 00120000
TNT EQU 2 TERMINAL NAME TABLE POINTER. 00130000
IOB EQU 1 USE R1 FOR IOB 00140000
PL EQU TUB USE FOR TP PARAMETER LIST. 00150000
SPACE 00160000
*FOLLOWING ARE SOME OF THE MISCELANEOUS EQUATES COMMONLY USED. 00170000
SPACE 00180000
# EQU 1 'INSTRUCTION-MODIFICATION' CODE. 00190000
PMRQ EQU X'30' Q-CODE FOR CCP INST TO MODIFY PMR. 00200000
DISABL EQU X'79' PMR=DSB,PRIV,ATR-ON 00210000
PRV EQU X'08' PRIVILEGED BIT IN PMR. 00220000
ENABLE EQU X'78' PMR=ENB,PRIV,ATR-ON 00230000
USER EQU X'72' PMR=ENB,NON-PRIV,ATR-ON,PROT 00240000
LPMRQ EQU X'40' LOAD PMR Q-CODE FOR LCP INST 00250000
IAR EQU X'10' LOAD/STORE Q-CODE FOR IAR 00260000
ARR EQU 8 LOAD/STORE Q-CODE FOR ARR 00270000
PSR EQU 4 LOAD/STORE Q-CODE FOR PSR 00280000
SPACE 00290000
* THE FOLLOWING ARE CP WORK AREA EQUATES FOR GENERAL USE. 00300000
SPACE 00310000
AREA EQU $CPPRQ+13 00320000
WORK EQU $CPPRQ 00330000
COMP EQU $CPPRQ+11 00340000
CTR EQU $CPPRQ+9 00350000
SPACE 00360000
* THE FOLLOWING ARE SYMBOLIC DIAGNOSTIC CODES FOR PGM REQUEST. 00370000
SPACE 00380000
A EQU * REFERENCE POINT. 00390000
SUSP EQU *-A SUSPENDED-TASK DIAGNOSTIC. 00400000
ORG *+1 00410000
ACTNEP EQU *-A REQUEST FOR ACTIVE NON-MRT NEP. 00420000
ORG *+1 00430000
PGMNF EQU *-A PROGRAM NAME INVALID OR MISSPELLED. 00440000
ORG *+1 00450000
DTANLW EQU *-A PGM NAME HAD DATA BUT WON'T ACCEPT. 00460000
ORG *+1 00470000
PCTERR EQU *-A PERMANENT IO ERROR READING PCT. 00480000
ORG *+1 00490000
NOPART EQU *-A PARTITION NOT AVAILABLE. 00500000
ORG *+1 00510000
NOTCB EQU *-A TCB NOT AVAILABLE. 00520000
ORG *+1 00530000
NOWAY EQU *-A NEP'S OWN TOO MUCH CORE FOR REQUEST 00540000
ORG *+1 00550000
MRTMAX EQU *-A MRT ALREADY HAS ENOUGH TUBS QUEUED. 00560000
ORG *+1 00570000
WATMRT EQU *-A REQUESTOR /NOQ OR ACTIVE MRT IN /Q 00580000
ORG *+1 00590000

```

SYMFIL	EQU	*-A	SYMBOLIC FILE REQUIRED, NOT GIVEN.	00600000
	ORG	*+1		00610000
CNFLCT	EQU	*-A	DISK FILE DOESN'T PERMIT SHARED USE	00620000
	ORG	*+1		00630000
NOTASN	EQU	*-A	REQUIRED TERMINAL NAME NOT ASSIGNED	00640000
	ORG	*+1		00650000
TWONAM	EQU	*-A	TERMINAL REQUIRED BY 2 NAMES.	00660000
	ORG	*+1		00670000
TRMNIA	EQU	*-A	TERMINAL NOT IMMEDIATELY AVAILABLE.	00680000
	ORG	*+1		00690000
URNA	EQU	*-A	UNIT RECORD DEVICE NOT AVAILABLE	00700000
	ORG	*+1		00710000
TRMNEP	EQU	*-A	TERMINAL OWNED BY NEP.	00720000
	ORG	*+1		00730000
SWLNA	EQU	*-A	SWITCHED TERMINAL LINE NOT AVAILABL	00740000
	ORG	*+1		00750000
TERMNA	EQU	*-A	MISCELLANEOUS NOT AVAILABLE CODE.	00760000
	ORG	*+1	* OFFLINE, SIGNED-ON, OLT IN-PROCESS, OR ERP IN-PROCESS.	00770000
	ORG	*+1		00780000
SPF	EQU	*-A	STOP-POLL FAILED.	00790000
	ORG	*+1		00800000
NOSHR	EQU	*-A	INVALID ATTEMPT TO USE DISKFILE	00810000
	ORG	*+1		00820000
INVRUF	EQU	*-A	RUF INPUT TO A NON-RUF PROGRAM	00830000
	ORG	*+1		00840000
DSMUPD	EQU	*-A	FILE BEING UPDATED BY OTHER PGM LVL	00850000
	ORG	*+1	* AJS	00853000
DISKNA	EQU	*-A	FILE BEING USED BY NEP AJS	00856000
	ORG	A		00860000
	SPACE	2		00870000
	MEND			00880000

MACRO			00010000
*****			00020000
.* STATUS: RELEASE 6			* 00030000
.*			* 00040000
.* NAME: \$EPCT			* 00050000
.*			* 00060000
.* FUNCTION: DEFINE THE LABELS NEEDED TO REFERENCE THE FIELDS IN			* 00070000
.* THE 'PROGRAM CONTROL TABLE' IN \$CCPFILE.			* 00080000
.*			* 00090000
.* INPUT OPERANDS:			* 00100000
.* * BITS-YES/Y GENERATE THE BIT INDICATORS FOR FIELDS IN THE			* 00110000
.* PCT IN ADDITION TO THE FIELD OFFSETS. THIS			* 00120000
.* IS THE DEFAULT.			* 00130000
.* NO/N DO NOT GENERATE THE BIT DEFINITIONS.			* 00140000
*****			00150000
\$EPCT			00160000
TEXT			00170000
*****			00180000
* PROGRAM CONTROL TABLE OFFSETS			* 00190000
*****			00200000
PCTCCP EQU 0	HI-ORDER OFFSET		00210000
SPACE			00220000
*****	FIXED SECTION	*****	00230000
PCTJMP EQU PCTCCP-1+2	LENGTH OF THIS ENTIRE PCT ENTRY		00240000
PCTNAM EQU PCTJMP+6	PROGRAM NAME		00250000
PCTOFF EQU PCTNAM+1-PCTCCP	@ OFFSET FOR PCT REFERENCE		00260000
*	IN TUB BUFFER		00270000
SPACE			00280000
-----	PROGRAM LOAD INFORMATION	-----	00290000
PCTCS EQU PCTNAM+2	C/S OF O.MODULE		00300000
PCT#S EQU PCTCS+1	NO. TEXT SECTORS IN O.MODULE		00310000
PCTLNK EQU PCT#S+2	LINK EDIT ADDR OF O.MODULE		00320000
PCTRLD EQU PCTLNK+1	DISPLACEMENT OF RLDS IN 1ST SCTR		00330000
PCTENT EQU PCTRLD+2	O.MODULE ENTRY POINT ADDRESS		00340000
SPACE			00350000
-----	OTHER PROGRAM ATTRIBUTES	-----	00360000
PCTLNG EQU PCTENT+1	PGM LENGTH-		00370000
*	NO. OF 256 BYTE BLOCKS - 1		00380000
PCTSIZ EQU PCTLNG+1	PGM LENGTH-		00390000
*	NO. OF 2K BLOCKS		00400000
SPACE 1			00410000
PCTAT1 EQU PCTSIZ+1	ATTRIBUTES BYTE 1		00420000
SPACE 1			00430000
PCTPRS EQU BIT0	* PGM WILL SHARE PRINTER.		00440000
PCTMTS EQU BIT1	* PGM SUPPORTS MULT REQ TERMINL		00450000
PCT501 EQU BIT2	* PGM REQUIRES 2501		00460000
PCTPRT EQU BIT3	* PGM REQUIRES LINE PRINTER		00470000
PCTNEP EQU BIT4	* PGM IS NEVER ENDING		00480000
PCTMFU EQU BIT5	* PGM REQUIRES MFCU/M		00490000
PCT14R EQU BIT6	* PGM REQUIRES 1442 AS A READER		00500000
PCT14P EQU BIT7	* PGM REQUIRES 1442 AS A PUNCH		00510000
SPACE 1			00520000
PCTAT2 EQU PCTAT1+1	ATTRIBUTES BYTE 2		00530000
SPACE 1			00540000
PCTSU EQU BIT0	* PGM SUPPRESSED BIT FROM INIT		00550000
PCTPAK EQU BIT1	1--PGM IS ON SYSTEM PACK		00560000
*	0--PGM IS ON PROGRAM PACK		00570000
PCTEMG EQU BIT2	END OF JOB MESSAGE OPTION		00580000
*	1--SEND ENDED/RELEASED MESSAGE		00590000

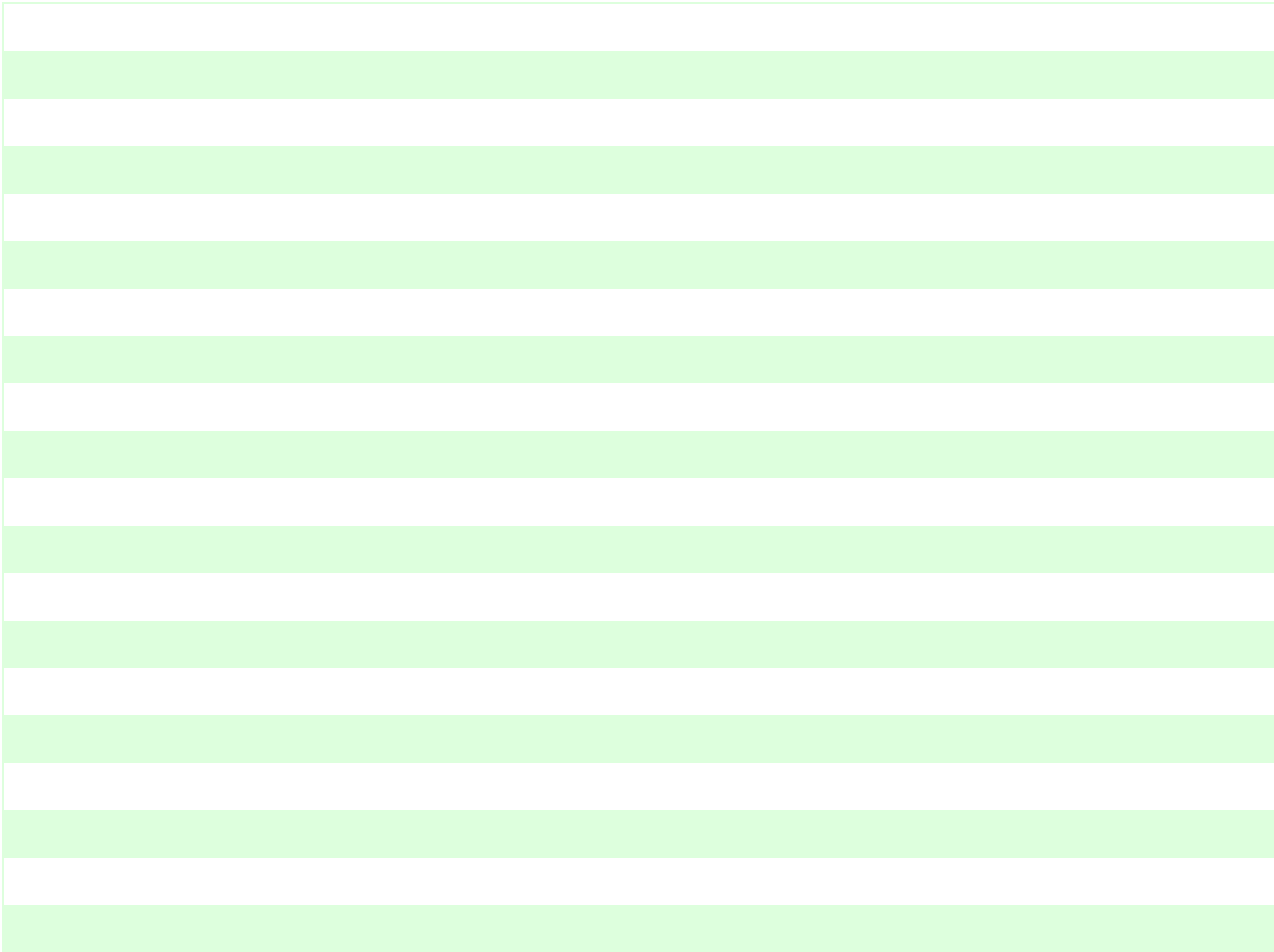

```
PCTREF EQU BIT4 * FILE IS REFERENCE ONLY 01200000
PCTLOD EQU BIT5 * ACCESS IS ORDERED INDEX LOAD 01210000
PCTIOU EQU BIT6 * ACCESS IS UNORDERED INDEX LOAD 01220000
PCTISA EQU BIT7 * ACCESS IS ISA OR ISUA 01230000
SPACE 1 01240000
PCTDTA EQU PCTMDE+1 RELATIVE NO. OF FILE--ASSIGNMENT 01250000
PCTCRL EQU PCTDTA+1 *** UNUSED *** 01260000
SPACE 1 01270000
PCTFLN EQU PCTCRL+1 LENGTH OF A PCT DISK FILE ENTRY 01280000
SPACE 01290000
*----- PROGRAM REQUEST COUNT ----- 01300000
SPACE 1 01310000
PCTDCT EQU PCTCRL+2 DISPL OF IN-CORE PROGRAM 01320000
* REQUEST COUNT TABLE 01330000
PCTNUM EQU PCTDCT+2 NO. TIMES PGM REQ (IF USED) 01340000
MEND 01350000
```



```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 3 * 00030000
.* * 00040000
.* NAME:  $EROL * 00050000
.* * 00060000
.* FUNCTION:  DEFINE ALL STARTUP PHASES SO THAT THEY CAN BE CALLED * 00070000
.* THROUGH A ROLLER LIST. * 00080000
.* * 00090000
.* INPUT OPERANDS:  NONE * 00100000
***** 00110000
    $EROL 00120000
    TEXT 00130000
    SPACE 00140000
***** 00150000
*          STARTUP PHASE ROLLER LIST - 23 ENTRIES MAXIMUM * 00160000
*          CAN BE USED BY THE ROLLER. MORE THAN 23 CAN EXIST* 00170000
*          IN THE TABLE, BUT ONLY 23 CAN BE ACCESSED. THE * 00180000
*          REST ARE INCLUDED FOR DIAGNOSTIC PURPOSES ONLY. * 00190000
***** 00200000
    SPACE 00210000
ROLIST EQU *          ID LIST FOLLOWS, 10 BYTES/ENTRY 00220000
    DC    CL10'FX--  '          CALC FREE C/S AND INDEX FORMATS 00230000
    DC    CL10'CR--  '          CROSS REFERENCE TRANSIENTS 00240000
    DC    CL10'US--  '          UNSUPPRESS FACILITIES 00250000
    DC    CL10'FS--  '          SUPPRESS FACILITIES 00260000
    DC    CL10'IP--  '          INITIALIZE PCT 00270000
    DC    CL10'LD--  '          LOAD BASE MODULE 00280000
    DC    CL10'UB--  '          BUILD USER TCB'S IN CORE 00290000
    DC    CL10'TA--  '          LOAD TERMINAL ATTRIBUTE TABLE 00300000
    DC    CL10'TB--  '          T/P CONTROL BLOCKS 00310000
    DC    CL10'TC--  '          BUILD TUBS AND TERMNAMES 00320000
    DC    CL10'DL--  '          BUILD SKELETON SHORT DTF'S 00330000
    DC    CL10'DM--  '          MULTI FILE DISAGNOSTIC PHASE 00340000
    DC    CL10'DF--  '          OPEN FILES AND COMPRESS DTF'S 00350000
    DC    CL10'PX--  '          BUILD PCT INDEX IN CORE 00360000
    SPACE 2 00370000
* 00380000
* THE FOLLOWING ENTRIES ARE NOT USED BY THE PHASE ROLLER. THEY ARE 00390000
* INCLUDED ONLY FOR THE PURPOSE OF ENSURING THEIR PRESENCE ON THE 00400000
* PACK. THEY DO NOT COUNT IN THE 23 ENTRIES MAXIMUM. THE FOLLOWING 00410000
* PHASES RUN IN THE CCP TRANSIENT AREA(S). 00420000
* 00430000
    SPACE 1 00440000
    DC    CL10'EI--  '          BUILD MLTA POLL LIST 00450000
    DC    CL10'EJ--  '          FILL DTF ADDRESSES, LOAD $CC4#2 00460000
    DC    CL10'LT--  '          LOAD TRACES 00470000
    DC    CL10'EK--  '          ALLOCATE TPBUF, USER PROG AREA 00480000
    DC    CL10'EL--  '          OPEN DTFS 00490000
    DC    CL10'EM--  '          DISPATCH CCP 00500000
    DC    CL10'CX--  '          REP CARD PROCESSOR FOR F. E. 00510000
    DC    CL10'CY--  '          REP CARD PROCESSOR FOR F. E. 00520000
    DC    CL10'CZ--  '          REP CARD PROCESSOR FOR F. E. 00530000
    DC    XL1'FF'  '          END OF ID LIST 00540000
    SPACE 00550000
ROLEN EQU 230 00560000
ROLEND EQU ROLIST+ROLEN 00570000
    SPACE 00580000
    MEND 00590000

```



```

MODULE-$CANB , VOLUME ID-R2R2R2, DATE-06/06/10
MACRO                                00010000
&LABEL $CANB &DTF-                   00020000
TEXT                                  00190000
*   CANCEL A GET REQUEST.             RELEASE-4   00200000
    AIF (T'&LABEL EQ ').A             00210000
&LABEL EQU *                           *           00220000
.A   AIF (T'&DTF EQ ').B              00230000
    LA  &DTF,$BDTF                    --> DTF     00240000
.B   TBN $BDDEV(,$BDTF),$BLIN2        Q IS THIS BSCA LINE 2? 00280000
    JF  $BH&SYSNDX                     NO - CONTINUE 00290000
    MVI $BA&SYSNDX+1,$BIRVI+$BLIN2    SET CORRECT OP CODE. 00300000
$BH&SYSNDX ST $BE&SYSNDX+$BTRE,$BBAC1 SAVE XR1. 00310000
    L   $BDIOB(,$BDTF),$BIOB          LOAD IOB ADDR IN XR1. 00320000
    CLI $BDCMP(,$BDTF),$BCREQ        Q HAS OPERATION BEEN REQUESTED? 00330000
    JE  $BG&SYSNDX                     YES-CONTINUE. 00340000
    CLI $BDCMP(,$BDTF),$BCCMP        Q NO COMPLETIONS POSTED? 00350000
$BG&SYSNDX TBN $BIFLA(,$BIOB),$BIFST  Q FIRST TIME INDIC ON? 00360000
    JC  $BK&SYSNDX,$BTREQ             YES-CONTINUE. 00370000
    TBN $BDATR(,$BDTF),$BCMCN        Q CONTROL STATION? 00380000
    JC  $BK&SYSNDX,$BTREQ             YES-CONTINUE. 00390000
    MVI $BDCMP(,$BDTF),$BCIGN        NO-POST IGNORE REQUEST. 00400000
    J   $BF&SYSNDX                     RETURN.      00410000
$BK&SYSNDX EQU *                         *           00410900
    TBN $BIOBQ(,$BIOB),$BLIN2        IS THIS LINE-2? 00411800
    L   $BDWKA(,$BDTF),$BWK          LOAD WORKAREA@. 00412700
    TBN $BWKMC(,$BWK),$BDAON        IS IDA SUPPORTED? 00413600
    JF  $BL&SYSNDX                     NO - CONTINUE 00414500
    SIO $B2SEC,$BTOSC                START TWO SEC T.O. TO IDA. 00415400
    SBN $BPOLD(,$BWK),$BPRES        INDIC POLL CANCEL 00416300
    J   $BF&SYSNDX                     RETURN      00417200
$BL&SYSNDX EQU *                         *           00418100
    L   $BDIOB(,$BDTF),$BIOB          LOAD IOB ADDR. 00419000
$BA&SYSNDX CLI $BIOBQ(,$BIOB),$BIRVI  Q SWITCHED LINE NON-RCV INIT 00450000
    TBN $BDATR(,$BDTF),$BCSWI        Q OR CONTROL STATION ? 00460000
    L   $BDWKA(,$BDTF),$BWK          LOAD WKA ADDR IN XR1. 00470000
    JC  $BD&SYSNDX,$BTRNQ            YES-SKIP ENABLE / DISABL . 00480000
    TBN $BDADD(,2),$BCAA1            Q ADD ON AREA SUPPORTED 00520000
    CLC $BDARA(2,2),X'0000'         Q AND AUTO-RES MODULE PRESENT ? 00530000
    BC  $BJ&SYSNDX,$BTRNQ            NO DISABLE THE LINE. 00540000
    SBN $BPOLD(,$BWK),$BPCNC        SET ON INDIC TO CANCEL. 00580000
    J   $BF&SYSNDX                     RETURN.      00590000
$BI&SYSNDX EQU *                         *           00600000
    SBN $BPOLD(,$BWK),$BPCNC        SET ON INDIC TO CANCEL. 00610000
    SBF $BWLGD(,$BWK),$BPATV        SET OFF LINE ACTIVE. 00620000
    J   $BF&SYSNDX                     RETURN.      00630000
$BJ&SYSNDX EQU *                         *           00670000
    MVC $BB&SYSNDX+$BONE,$BDDEV(1,$BDTF) MOVE Q-CODE TO DSBL INST 00680000
    MVC $BC&SYSNDX+$BONE,$BDDEV(1,$BDTF) MOVE Q-CODE TO ENABL INST 00690000
$BB&SYSNDX EQU *                           *           00700000
    SIO $BDISA,*-*                   DISABLE BSCA. 00710000
    TBN $BDATR(,$BDTF),$BCSWI        SWITCHED LINE RCV INITIAL ? 00750000
    BT  $BI&SYSNDX                     YES-GO SET CANCEL INDICATOR 00760000
$BC&SYSNDX EQU *                           *           00770000
    SIO $BENAB,*-*                   ENABLE BSCA. 00780000
    SBN $BPOLD(,$BWK),$BPCNC        SET ON INDIC TO CANCEL. 00790000
$BD&SYSNDX EQU *                           *           00800000
    SBN $BPOLD(,$BWK),$BPRES        SET ON RESET POLL INDICATOR. 00810000
$BF&SYSNDX EQU *                           *           00820000
$BE&SYSNDX LA *-*, $BBAC1            RESTORE XR1. 00830000

```


MACRO			00010000
*****			00020000
.* STATUS: RELEASE 5			* 00030000
.*			* 00040000
.* NAME: \$ESIT			* 00050000
.*			* 00060000
.* FUNCTION: DEFINE THE LABELS NEEDED TO REFERENCE THE 'SYSTEM			* 00070000
.* INFORMATION TABLE' IN \$CCPFILE.			* 00080000
.*			* 00090000
.* INPUT OPERANDS: NONE			* 00100000
*****			00110000
\$ESIT			00120000
TEXT			00130000
*****			00140000
* SYSTEM INFORMATION TABLE OFFSETS *			00150000
*****			00160000
SITCCP EQU 0			00170000
SITMXP EQU SITCCP-1+1	MAX NO. CONCURRENT USER PROGRAMS		00180000
SITMUA EQU SITMXP+1	MIN NO. 256-BYTE BLOCKS		00190000
*	FOR CCP USER PROGRAM AREA		00200000
SITMTB EQU SITMUA+2	MIN NO. BYTES, TP HOLD BUFR AREA		00210000
SITCPW EQU SITMTB+6	PASSWORD IF CCP PASSWORD USED,		00220000
*	OTHERWISE X'40'		00230000
SPACE			00240000
SITDED EQU SITCPW+1	DEVICES DEDICATED TO CCP		00250000
SITPRT EQU BIT0	1=PRINTER DEDICATED		00260000
SIT501 EQU BIT1	1=2501 DEDICATED READER		00270000
SITM1R EQU BIT2	1=MFCU1/MFCM1 DEDICATED READER		00280000
SITM1P EQU BIT3	1=MFCU1/MFCM1 DEDICATED PUNCH		00290000
SITM1C EQU BIT4	1=MFCU1/MFCM1 DEDICATED COMBINED		00300000
SITM2R EQU BIT5	1=MFCU2/MFCM2 DEDICATED READER		00310000
SITM2P EQU BIT6	1=MFCU2/MFCM2 DEDICATED PUNCH		00320000
SITM2C EQU BIT7	1=MFCU2/MFCM2 DEDICATED COMBINED		00330000
SPACE			00340000
SITDD2 EQU SITDED+1	CONTINUATION OF SITDED.		00350000
SIT14R EQU BIT0	1=1442 DEDICATED READER		00360000
SIT14P EQU BIT1	1=1442 DEDICATED PUNCH		00370000
SIT14C EQU BIT2	1=1442 DEDICATED COMBINED.		00380000
SIT37R EQU BIT5	1=3741 DEDICATED INPUT.		00390000
SIT37P EQU BIT6	1=3741 DEDICATED OUTPUT.		00400000
SPACE			00410000
SITFLG EQU SITDD2+1	FLAG BYTE		00420000
SITDFU EQU BIT0	PACK FOR DISPLAY FORMATS		00430000
*	0=PROGRAM PACK		00440000
*	1=SYSTEM PACK		00450000
SITFMT EQU BIT1	1=DFE SUPPORTED IN THIS SET		00460000
*	0=DFE NOT SUPPORTED IN THIS SET		00470000
SITDFF EQU BIT2	1=ALL TERMINAL ATTRIBUTES DFE		00480000
*	0=NON-DFE TERMINAL ATTRIB. EXIST		00490000
SPACE			00500000
SITRV2 EQU SITFLG+1	RESERVED		00510000
SITFSQ EQU SITRV2+1	NO. FILE SECTOR ENQUEUE BLOCKS		00520000
SITRQL EQU SITFSQ+1	MAX LENGTH PGM RQST + DATA		00530000
SITNQL EQU SITRQL+1	MAX LENGTH NON PGM RQST COMMAND		00540000
SITMCL EQU SITNQL+2	MAX TERMINAL COMMAND LENGTH		00550000
SITTAS EQU SITMCL+1	HIGHEST TAS ID VALUE		00560000
SITSTT EQU SITTAS+1	NO. OF STT ENTRIES		00570000
SITNPC EQU SITSTT+1	NO. OF PCT ENTRIES		00580000
SITLPC EQU SITNPC+1	LENGTH OF LONGEST PCT ENTRY		00590000

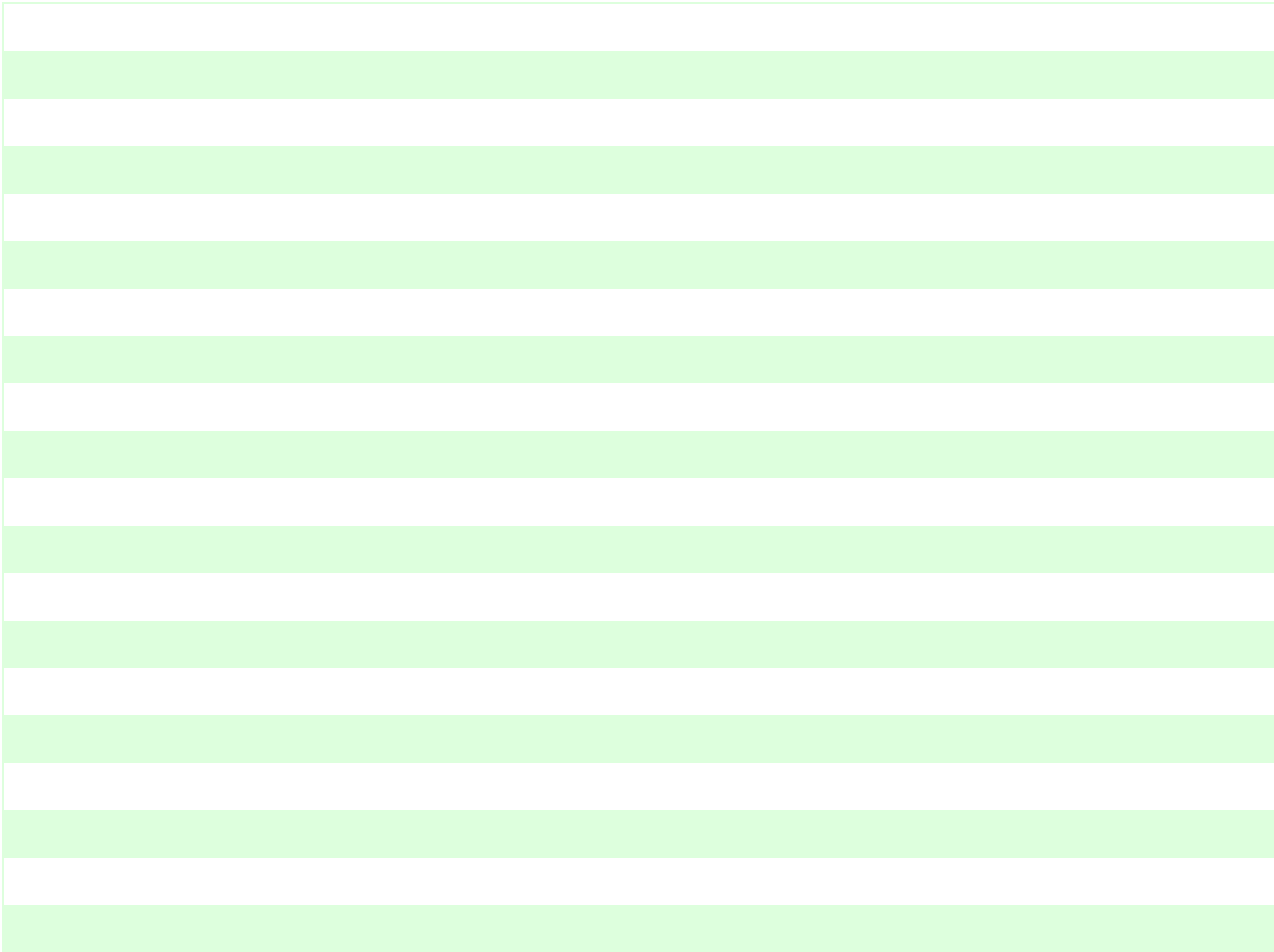
SITSID EQU	SITLPC+1	SET ID OF THIS SET	00600000
SITSYM EQU	SITSID+1	NUMBER OF SYMFILES IN SET	00610000
SITTRM EQU	SITSYM+1	NUMBER OF COMMAND TRMNL IN SET	00620000
SITRUF EQU	SITTRM+2	READ UNDER FORMAT - LENGTH	00630000
SITPOL EQU	SITRUF+4	POLL TIME IN UNITS	00640000
*		300 UNITS = 1 SECOND.	00650000
*		00000001 = 'NO'	00660000
*		00000000 = KEYWORD NOT GIVEN.	00670000
SITWIT EQU	SITPOL+2	WAIT TIME IN UNITS	00680000
*		300 UNITS = 1 SECOND.	00690000
*		0001 = 'NO'	00700000
*		0000 = KEYWORD NOT GIVEN.	00710000
	SPACE		00720000
SITRV1 EQU	SITWIT+4	RESERVED	00730000
	SPACE		00740000
SITEND EQU	SITRV1	END OF SIT	00750000
SITLEN EQU	SITEND+1	LENGTH OF SIT	00760000
SITDAT EQU	SITRV1+6	SYSTEM DATE AT TIME OF	00762000
*		ASSIGNMENT SET BUILD	00764000
SITTIM EQU	SITDAT+4	SYSTEM TIME(IF SUPPORTED) AT	00766000
*		ASSIGNMENT SET BUILD	00768000
	MEND		00770000

MODULE-\$ETMG , VOLUME ID-R2R2R2, DATE-06/06/10

```
MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 3 * 00030000
.* * 00040000
.* NAME:  $ETMG * 00050000
.* * 00060000
.* FUNCTION  GENERATE THE TEXT FOR COMMAND PROCESSOR MESSAGES TO * 00070000
.* THE TERMINAL OPERATOR. * 00080000
.* * 00090000
.* INPUT OPERANDS: * 00100000
.* * N- TWO CHARACTER SUFFIX OF THE PROGRAM IN WHICH * 00110000
.* THIS MACRO IS USED. * 00120000
.* * POS-LAST THIS IS THE LAST OF THIS BLOCK OF MESSAGES * 00130000
.* EXPANDED IN THIS PROGRAM. * 00140000
.* * MGN- THREE CHARACTER MESSAGE IDENTIFIER. * 00150000
***** 00160000
&LABEL $ETMG &N-,&POS-,&MGN-,&VAR1-,&VAR2-,&VAR3-,&VAR4-,&VAR5- 00170000
TEXT 00180000
AIF (T'&LABEL EQ 'O').NOLAB 00190000
&LABEL EQU * 00200000
.NOLAB ANOP 00210000
&N&MGN DC AL1(&N&MGN.L) ONE BYTE MESSAGE LENGTH 00220000
&N&MGN.B DC CL4'&MGN' MESSAGE NUMBER PREFIX 00230000
.A01 AIF (&MGN NE 'A01').A02 00240000
DC CL25'SIGNED ON-PROCEED' 00250000
.A02 AIF (&MGN NE 'A02').A03 00260000
DC CL20'MESSAGE SENT' 00270000
.A03 AIF (&MGN NE 'A03').A04 00280000
DC CL25'Q STATUS - PROCEED' 00290000
.A04 AIF (&MGN NE 'A04').A05 00300000
DC CL25'NOQ STATUS-PROCEED' 00310000
.A05 AIF (&MGN NE 'A05').A06 00320000
DC CL25'ACCEPTED-PROCEED' 00330000
.A06 AIF (&MGN NE 'A06').A07 00340000
DC CL25'ACCEPTED-PROCEED' 00350000
.A07 AIF (&MGN NE 'A07').A08 00360000
DC CL31'ENTER /MSG, /RELEASE, OR /RUN' 00370000
.A08 AIF (&MGN NE 'A08').A09 00380000
DC CL20'RELEASED-PROCEED' 00390000
.A09 AIF (&MGN NE 'A09').A10 00400000
DC CL20'RESUME PROGRAM' 00410000
.A10 AIF (&MGN NE 'A10').A11 00420000
DC CL11'SIGNED OFF-' 00430000
&N.&VAR1 DC 4XL1'00' 'HOLD' OR 'DROP' 00440000
DC CL15' TERMINAL' 00450000
.A11 AIF (&MGN NE 'A11').A12 00460000
DC CL10'CLEAR' 00470000
.A12 AIF (&MGN NE 'A12').E01 00480000
DC CL20'PROCEED-CANCELLED' 00490000
.E01 AIF (&MGN NE 'E01').E02 00500000
DC CL18'INVALID SYNTAX' 00510000
.E02 AIF (&MGN NE 'E02').E03 00520000
DC CL18'OPERAND MISSING' 00530000
.E03 AIF (&MGN NE 'E03').E04 00540000
DC CL30'ALREADY SIGNED ON-PROCEED' 00550000
.E04 AIF (&MGN NE 'E04').E05 00560000
DC CL20'PASSWORD ERROR' 00570000
.E05 AIF (&MGN NE 'E05').E06 00580000
DC CL30'THAT COMMAND NOT VALID NOW' 00590000
```

.E06	AIF	(&MGN NE 'E06').E07	00600000
	DC	CL20'MESSAGE TOO LONG'	00610000
.E07	AIF	(&MGN NE 'E07').E08	00620000
	DC	CL28'THAT COMMAND NOT VALID NOW'	00630000
.E08	AIF	(&MGN NE 'E08').E09	00640000
	DC	CL30'INVALID SIGN OFF OPTION'	00650000
.E09	AIF	(&MGN NE 'E09').E10	00660000
	DC	CL18'NOT SIGNED ON'	00670000
.E10	AIF	(&MGN NE 'E10').E15	00680000
	DC	CL30'THAT COMMAND NOT VALID NOW'	00690000
.E15	AIF	(&MGN NE 'E15').E16	00700000
	DC	CL31'ENTER /FILE THEN PROG REQ'	00710000
.E16	AIF	(&MGN NE 'E16').E17	00720000
	DC	CL30'THAT COMMAND NOT VALID NOW'	00730000
.E17	AIF	(&MGN NE 'E17').E18	00740000
	DC	CL25'FILE NAME TOO LONG'	00750000
.E18	AIF	(&MGN NE 'E18').E19	00760000
	DC	CL30'THAT COMMAND NOT VALID NOW'	00770000
.E19	AIF	(&MGN NE 'E19').E20	00780000
	DC	CL20'INVALID SYNTAX'	00790000
.E20	AIF	(&MGN NE 'E20').E21	00800000
	DC	CL9'FILENAME'	00810000
&N.&VAR1	DC	8XL1'0'	00820000
	DC	CL10' UNKNOWN'	00830000
.E21	AIF	(&MGN NE 'E21').E22	00840000
	DC	CL31'THIS FILE COMBINATION NOT VALID'	00850000
.E22	AIF	(&MGN NE 'E22').E23	00860000
	DC	CL30'THAT COMMAND NOT VALID NOW'	00870000
.E23	AIF	(&MGN NE 'E23').E24	00880000
	DC	CL20'NAME TOO LONG'	00890000
.E24	AIF	(&MGN NE 'E24').E25	00900000
	DC	CL25'UNKNOWN TERMINAL NAME'	00910000
.E25	AIF	(&MGN NE 'E25').E28	00920000
	DC	CL30'UNASSIGNED TERMINAL NAME'	00930000
.E28	AIF	(&MGN NE 'E28').E29	00940000
	DC	CL31'PROG NAME INVALID OR SUPPRESSED'	00950000
.E29	AIF	(&MGN NE 'E29').E30	00960000
	DC	CL18'NOT SIGNED ON'	00970000
.E30	AIF	(&MGN NE 'E30').E31	00980000
	DC	CL15'SHUTDOWN'	00990000
.E31	AIF	(&MGN NE 'E31').E32	01000000
	DC	CL31'DATA EXCESSIVE OR NOT ALLOWED'	01010000
.E32	AIF	(&MGN NE 'E32').E33	01012000
	DC	CL38'RUF INPUT WITH NON-RUF PROGRAM REQUEST'	01014000
.E33	AIF	(&MGN NE 'E33').R01	01016000
	DC	CL39'RUF INPUT FOR AN ACTIVE NON-RUF PROGRAM'	01018000
.R01	AIF	(&MGN NE 'R01').R02	01020000
	DC	CL30'COMMANDS SUSPENDED-WAIT'	01030000
.R02	AIF	(&MGN NE 'R02').R03	01040000
	DC	CL31'COMMANDS RESUMED-REENTER'	01050000
.R03	AIF	(&MGN NE 'R03').R04	01060000
	DC	CL31'REQUESTED PROG SUSPENDED'	01070000
.R04	AIF	(&MGN NE 'R04').R05	01080000
	DC	CL28'REQUESTED PROG BUSY'	01090000
.R05	AIF	(&MGN NE 'R05').R07	01100000
	DC	CL31'PROG NO LONGER AVAILABLE'	01110000
.R07	AIF	(&MGN NE 'R07').R09	01120000
	DC	CL24'SYSTEM ERROR - RETRY'	01130000
.R09	AIF	(&MGN NE 'R09').R10	01140000
	DC	CL31'UNIT RECORD DEVICE UNAVAILABLE'	01150000

.R10	AIF	(&MGN NE 'R10').R13		01160000
	DC	CL31'LOGICAL TERMINAL UNAVAILABLE'		01170000
.R13	AIF	(&MGN NE 'R13').R14		01180000
	DC	CL20'CORE UNAVAILABLE'		01190000
.R14	AIF	(&MGN NE 'R14').R16		01200000
	DC	CL28'CORE TEMP UNAVAILABLE'		01210000
.R16	AIF	(&MGN NE 'R16').R19		01220000
	DC	CL31'DISK FILE TEMP UNAVAILABLE'		01230000
.R19	AIF	(&MGN NE 'R19').R20		01240000
	DC	CL28'REQUIRED TERMINAL IN USE'		01250000
.R20	AIF	(&MGN NE 'R20').R21		01260000
	DC	CL31'REQUESTED PROG IN /Q WAIT STATE'		01270000
.R21	AIF	(&MGN NE 'R21').R23		01280000
	DC	CL31'SYSTEM TEMPORARILY BUSY'		01290000
.R23	AIF	(&MGN NE 'R23').R24		01300000
	DC	CL31'REQUIRED TERM PERM ALLOCATED'		01310000
.R24	AIF	(&MGN NE 'R24').R25		01320000
	DC	CL31'SWITCHED LINE NOT AVAILABLE'		01330000
.R25	AIF	(&MGN NE 'R25').R26		01340000
	DC	CL31'TERM NOT IMMEDIATELY AVAILABLE'		01350000
.R26	AIF	(&MGN NE 'R26').R27		01360000
	DC	CL31'DISK FILE ACCESS NOT ALLOWED'		01370000
.R27	AIF	(&MGN NE 'R27').R28		01380000
	DC	CL24'TERMINAL NOT AVAILABLE'		01390000
.R28	AIF	(&MGN NE 'R28').R29	AJS	01392000
	DC	CL21'FILE SHARE CONFLICT'		01394000
.R29	AIF	(&MGN NE 'R29').S01	AJS	01396000
	DC	CL31'REQUIRED FILE PERM ALLOCATED'	AJS	01398000
.S01	AIF	(&MGN NE 'S01').S02		01400000
	DC	CL20'PROG REL-PROCEED'		01410000
.S02	AIF	(&MGN NE 'S02').S03		01420000
	DC	CL20'PROG REL-SHUTDOWN'		01430000
.S03	AIF	(&MGN NE 'S03').S04		01440000
	DC	CL20'PROG END-PROCEED'		01450000
.S04	AIF	(&MGN NE 'S04').S05		01460000
	DC	CL10'SHUTDOWN'		01470000
.S05	AIF	(&MGN NE 'S05').S06		01480000
	DC	CL20'PROG END-SHUTDOWN'		01490000
.S06	AIF	(&MGN NE 'S06').S07		01500000
	DC	CL30'PROG CANCELLED-SHUTDOWN'		01510000
.S07	AIF	(&MGN NE 'S07').S08		01520000
	DC	CL17'NOW ONLINE-NAMED '		01530000
&N.&VAR1	DC	6XL1'0'		01540000
.S08	AIF	(&MGN NE 'S08').S09		01550000
	DC	CL15'NOW OFFLINE'		01560000
.S09	AIF	(&MGN NE 'S09').S11		01570000
	DC	CL30'PROG CANCELLED-PROCEED'		01580000
.S11	AIF	(&MGN NE 'S11').S12		01590000
	DC	CL25'INPUT IGNORED-PROCEED'		01600000
.S12	AIF	(&MGN NE 'S12').S13		01610000
	DC	CL20'TP ERROR ON INPUT'		01620000
.S13	AIF	(&MGN NE 'S13').S99		01630000
	DC	CL20'INPUT IGNORED'		01640000
.S99	ANOP			01650000
&N&MGN.E	EQU	*		01660000
&N&MGN.L	EQU	&N&MGN.E-&N&MGN.B+3	MESSAGE LENGTH EQUATE	01670000
	AIF	(&POS NE 'LAST').NOFF		01680000
	DC	XL1'FF'	MESSAGES DELIMITER	01690000
.NOFF	ANOP			01700000
	MEND			01710000



```

MACRO 00010000
***** 00020000
.* STATUS:  RELEASE 3 * 00030000
.* * 00040000
.* NAME:  $EASC * 00050000
.* * 00060000
.* FUNCTION:  DEFINE THE TABLE OF VALID ASSIGNMENT STATEMENT NUMBERS.* 00070000
.* DEFINE THE EQUATES NEEDED FOR COMMON COMMUNICATION BETWEEN * 00080000
.* ASSIGNMENT PHASES. * 00090000
.* * 00100000
.* INPUT OPERANDS:  NONE * 00110000
***** 00120000
$EASC 00130000
TEXT 00140000
* * * * * 00150000
* * * * * 00160000
* TABLE OF VALID STATEMENT NUMBERS * 00170000
* * * * * 00180000
* * * * * 00190000
SPACE 00200000
SSET EQU 1 SET STATEMENT 00210000
SSYS EQU 2 SYSTEM STATEMENT 00220000
STAT EQU 3 TERMATTR STATEMENT 00230000
SBLINE EQU 4 BSCALINE STATEMENT 00240000
SBTERM EQU 5 BSCATERM STATEMENT 00250000
SMLINE EQU 6 MLTALINE STATEMENT 00260000
SMTERM EQU 7 MLTATERM STATEMENT 00270000
STNAME EQU 8 TEAMNAME STATEMENT 00280000
SDFILE EQU 9 DISKFILE STATEMENT 00290000
SPGM EQU 10 PROGRAM STATEMENT 00300000
SEOF EQU X'4B' END OF FILE STATEMENT (/*) 00310000
SOURCE EQU 12 SOURCE STATEMENT. 00320000
SYMFIL EQU 13 SYMFILE STATEMENT. 00330000
SPACE 5 00340000
* * * * * 00350000
* * * * * 00360000
* EQUATES VALUES FOR COMMON COMUNICATION AREA BETWEEN PHASES. * 00370000
* * * * * 00380000
* * * * * 00390000
SPACE 00400000
ASCBF@ EQU 1 ADDR OF INPUT BUFFER 00410000
ASCBD@ EQU ASCBF@+2 ADDR OF BFR POS POINTER. 00420000
ASCER@ EQU ASCBD@+2 ADDR OF ERROR LOG ROUTINE 00430000
ASCBFS EQU ASCER@+2 POSITION NOW POINT AT IN BFR. 00440000
ASCSS@ EQU ASCBFS+2 ENTRY ADDR TO SYNTAX SCAN ROUTIN 00450000
ASCMSF EQU ASCSS@+1 MIS. FLAG 00460000
ASCWGS EQU ASCMSF+2 WRONG STMT ROUTINE 00470000
ASCDKB EQU ASCWGS+2 DISK BUFFER 00480000
ASCCIN EQU ASCDKB+2 CAM INPUT ADDRESS 00490000
ASCCOT EQU ASCCIN+2 CAM OUTPUT ADDRESS 00500000
ASCFND EQU ASCCOT+2 @ OF SET ID FIND ROUTINE. 00510000
ASCDEL EQU ASCFND+2 @ OF DELETE ROUTINE. 00520000
ASC$CP EQU ASCDEL+2 @ OF $CCPFILE DTF. 00530000
ASC$WK EQU ASC$CP+2 ADDR OF $WORKFILE DTF. 00540000
ASCNCS EQU ASC$WK+2 NEXT SECTER IN $CCPWORK, 00550000
* RELATIVE NUMBER. 00560000
ASCRD@ EQU ASCNCS+2 READ ROUTINE FOR CTRL STMTS. 00570000
ASCEOF EQU ASCRD@+2 END OF FILE ROUTINE (/*). 00580000
ASCCFG EQU ASCEOF+2 @ OF CONFIGURATION RCD IN CORE. 00590000

```

ASCDIR EQU	ASCCFG+2	@ OF DIRECTORY RCD IN CORE.	00600000
ASCTUS EQU	ASCDIR+2	@ OF TAT USE TABLE.	00610000
ASCTAT EQU	ASCTUS+2	BEGINNING ADDR FOR TAT TABLE	00620000
ASCLCT EQU	ASCTAT+2	BEGIN ADDR OF LCT COMPRESSED TBL	00630000
ASCTUT EQU	ASCLCT+2	BEGIN @ OF TUT COMPRESSED TABLE	00640000
ASCTNT EQU	ASCTUT+2	BEGIN @ OF TNT COMPRESSED TABLE	00650000
ASCFCT EQU	ASCTNT+2	BEGIN @ OF FCT COMPRESSED TABLE	00660000
ASCPCT EQU	ASCFCT+2	BEGIN @ OF PCT COMPRESSED TABLE	00670000
ASCTDR EQU	ASCPCT+2	BEGIN @ OF TEMP DIR ENTRY.	00680000
ASCPAR EQU	ASCTDR+2	@ OF SYS PRINT PARM LIST.	00690000
ASCBUF EQU	ASCPAR+2	@ OF SYS PRINT OUTPUT BFR.	00700000
ASCRTN EQU	ASCBUF+2	@ OF SYS PRINT ROUTINE.	00710000
	SPACE 4		00720000
* * * * *	* * * * *	* * * * *	00730000
*			00740000
*	FORMAT OF COMPRESSED IN CORE TABLE ENTRYS.		00750000
*			00760000
* * * * *	* * * * *	* * * * *	00770000
	SPACE 2		00780000
*	TAT ENTRIES ARE NOT COMPRESSED, REMOVED BEFORE PROCESSING		00790000
*	FIRST TERMNAME STMT.		00800000
	SPACE		00810000
*	TAT USE TABLE, TABLE OF TAT ENTRIES AND WHICH LINE USED ON.		00820000
	SPACE		00830000
TATBSC EQU	0-1+1	BSCA LINE USE.	00840000
TATMLT EQU	TATBSC+1	MLTA LINE USE.	00850000
TATLNG EQU	TATMLT+1	TAT USE ENTRY LNG.	00860000
	SPACE		00870000
*	LCT COMPRESSED INCORE TABLE ENTRY FORMAT		00880000
	SPACE		00890000
LTJMP EQU	-1+2	COMPRESSED ENTRY LNG.	00900000
LTNLN EQU	LTJMP+1	LINE #, AND TYPE.	00910000
LTAT1 EQU	LTNLN+1	LINE ATTRIBUTES 1.	00920000
LTSND EQU	BIT1	IDEXSEND GIVEN.	00930000
LTPLL EQU	BIT6	POLL LIST GIVEN.	00940000
LTAT2 EQU	LTAT1+1	LINE ATTRIBUTES 2.	00950000
LTTCD EQU	LTAT2+1	TRANSMISSION CODE.	00960000
LTLNA EQU	LTTCD+1	LINE CONFIGURATION.	00970000
LTBLK EQU	LTLNA+2	LINE BLOCK LNG.	00980000
LTDFE EQU	LTBLK+2	DFFOHA VALUE.	00990000
LTLNG EQU	LTDFE+1	MIN LNG OF A COMPRESSED ENTRY.	01000000
LTPL EQU	LTDFE+2	POLL LIST, 2 BYTE ENTRIES, THE	01010000
*		2 CHAR ID GIVEN ON THE CTRL STMT	01020000
	SPACE 2		01030000
*	TUT COMPRESSED INCORE TABLE ENTRY FORMAT.		01040000
	SPACE 2		01050000
TTID EQU	-1+2	TERMINAL ID.	01060000
TTIID EQU	TTID+1	INTERNAL ID.	01070000
TTCHR EQU	TTIID+1	TERMINAL CHARACTERISTICS.	01080000
TPPRI EQU	BIT1	PRIMARY TERMINAL ASSIGNED.	01090000
TPPH# EQU	BIT3	PHONE # WILL BE REQUIRED.	01100000
TTLIN EQU	TTCHR+1	LINE # FOR THIS TERMINAL.	01110000
TTADR EQU	TTLIN+1	MLTA TERMINAL ADDR.	01120000
TTLNG EQU	TTADR+1	LNG OF A COMPRESSED ENTRY.	01130000
	SPACE 2		01140000
*	TNT COMPRESSED INCORE TABLE ENTRY FORMAT		01150000
	SPACE 2		01160000
TNCCP EQU	0	BEGINNING OF ENTRY.	01170000
TNNAM EQU	TNCCP-1+6	SYMBOLIC TERMINAL NAME.	01180000
TNTUB EQU	TNNAM+1	TUTIID FOR THIS NAME.	01190000

TNLIN	EQU	TNTUB+1	LINE # FOR THIS TERMINAL.	01200000
TNFLG	EQU	TNLIN+1	FLAG BYTE FOR DIAG.	01210000
TNPG	EQU	BIT3	ALREADY REFERENCED BY THIS PROG.	01220000
TNLNG	EQU	TNFLG+1	LNG OF A COMPRESSED ENTRY.	01230000
	SPACE	2		01240000
*	FCT	COMPRESSED INCORE TABLE ENTRY FORMAT.		01250000
	SPACE	2		01260000
FTCCP	EQU	0	BEGINNING OF ENTRY.	01270000
FTJMP	EQU	FTCCP-1+1	FILLER, NOT USED.	01280000
FTMCL	EQU	FTJMP+1	MISC. FLAG BYTE	01290000
FTNAM	EQU	FTMCL+8	FILE NAME.	01300000
FTAT1	EQU	FTNAM+1	FILE ATTRIBUTES 1.	01310000
FTAT2	EQU	FTAT1+1	ATTRIBUTES 2.	01320000
FT256	EQU	BIT7	RECL-256 INDICATOR FOR SORT	01330000
	SPACE	1		01340000
FTSTR	EQU	FTAT2+1	RELATIVE SECTOR # OF ENTRY IN	01350000
*			WORKFILE IF SYMBOLIC NAME.	01360000
FTDISP	EQU	FTSTR+1	DISPLACEMENT INTO SECTOR.	01370000
FTLNG	EQU	FTDISP+1	LNG OF COMPRESSED ENTRY.	01380000
*	-----*			01390000
*	FOLLOWING EQUATES ARE USED ONLY WHILE BUILDING FCT ENTRIES			* 01400000
*	-----*			01410000
FTRCL	EQU	FTAT2+2	RECORD LNG.	01420000
FTKYL	EQU	FTRCL+1	KEY LNG.	01430000
FTKYD	EQU	FTKYL+2	KEY DISP.	01440000
FTS	EQU	FTKYD+1	HOLD AREA FOR RELATIVE SECT.	01450000
FTD	EQU	FTS+1	HOLD AREA FOR DISPLACEMENT.	01460000
FTLLG	EQU	FTD+1	LNG OF A BUILD COMPRESSED ENTRY.	01470000
	SPACE	2		01480000
*	PCT	COMPRESSED INCORE TABLE ENTRY FORMAT.		01490000
	SPACE	2		01500000
PTCCP	EQU	0	BEGINNING OF ENTRY.	01510000
PTNAM	EQU	PTCCP-1+6	PROG NAME.	01520000
PTSTR	EQU	PTNAM+1	RELATIVE SECTOR LOC OF ENTRY.	01530000
PTDISP	EQU	PTSTR+1	DISP INTO SECTOR.	01540000
PTLNG	EQU	PTDISP+1	LNG OF COMPRESSED ENTRY.	01550000
	SPACE	2		01560000
*	*	*	*	01570000
*	*	*	*	01580000
*	*	*	*	01590000
	SPACE	2		01600000
ID	EQU	-1+1	SET ID.	01610000
*			# OF SECTORS IN EACH TABLE.	01620000
SIT	EQU	ID+1	SYSTEM INFORMATION TABLE.	01630000
TAT	EQU	SIT+1	TERMINAL ATTRIBUTES TABLE.	01640000
LCT	EQU	TAT+1	LINE CONTROL TABLE.	01650000
TUT	EQU	LCT+1	TERMINAL USED TABLE.	01660000
TNT	EQU	TUT+1	TERMINAL NAME TABLE.	01670000
STT	EQU	TNT+1	SWITCHED TERMINAL TABLE.	01680000
FCT	EQU	STT+1	FILE CONTROL TABLE.	01690000
PCT	EQU	FCT+1	PROGRAM CONTROL TABLE.	01700000
	SPACE			01710000
WRK	EQU	PCT+1	BEGINNING OF COMMON WORK AREA.	01720000
	SPACE	2		01730000
*	FORMAT OF COMMON WORK AREA.			01740000
	SPACE	2		01750000
ACTION	EQU	+1-1	ACTION INDICATOR.	01760000
	SPACE			01770000
SETSTM	EQU	BIT0	'SET' STMT READ.	01780000
ACTCRE	EQU	BIT7	ACTION-CREATE	01790000

ACTREP	EQU	BIT6	ACTION-REPLACE	01800000
ACTMOD	EQU	BIT5	ACTION-MODIFY	01810000
ACTDEL	EQU	0	ACTION-DELETE	01820000
CMD32	EQU	BIT4	3270 COMMAND TERMINAL FLAG.	01830000
		SPACE		01840000
PBYTU	EQU	ACTION+1	# OF BYTES USED IN LAST SECT.	01850000
LTDISP	EQU	PBYTU+1	DISP. TO LAST LCT ENTRY IN SECT.	01860000
CSTR#	EQU	LTDISP+1	CURRENT LCT MIN ENTRY RELATIVE	01870000
*			SECT #.	01880000
SITR#	EQU	LTDISP-1+2	RELATIVE SECTOR # OF SIT TABLE	01890000
*			IS \$CCPFILE ON ACTION-SYSMOD.	01900000
CNSTR	EQU	CSTR#+1	# OF SECTORS USED FOR CURRENT	01910000
*			LCT ENTRY.	01920000
TTUTWF	EQU	CNSTR+2	SECT # FOR TEMP TUT TABLE.	01930000
LPCT	EQU	CNSTR+2	LNG OF THE LONGEST PCT ENTRY.	01940000
TUTSTR	EQU	TTUTWF+2	NEXT AVAIL SECT IN TEMP TUT TAB.	01950000
CTT@	EQU	TUTSTR+2	@ OF NEXT AVAIL POS IN TEMP	01960000
*			COMPRESSED TUT TABLE.	01970000
BTUT@	EQU	CTT@+2	@ OF NEXT AVAIL POS IN BUILD	01980000
*			TUT TABLE.	01990000
		SPACE		02000000
WKFLG	EQU	BTUT@+1	COMMON WORK FLAG BYTE.	02010000
DFFYES	EQU	BIT0	DFE SUPPORTED ON SYSTEM STMT.	02020000
DFFATT	EQU	BIT1	DFE ON A ATTRIBUTE STMT.	02030000
DFFLIN	EQU	BIT2	DFE ON A BSCALINE STMT.	02040000
		SPACE		02050000
ERRMSG	EQU	WKFLG+3	COUNT OF TERMINATION MSG.	02060000
WRNMSG	EQU	ERRMSG+3	COUNT OF WARNING MSG.	02070000
		SPACE		02080000
SYMCT	EQU	WRNMSG+3	# OF REFERENCED SYMFILE STMTS.	02090000
CMDTRM	EQU	SYMCT+1	# OF COMMAND TERMINALS.	02100000
RUFLNG	EQU	CMDTRM+2	LNG OF LONGEST DFE FORMAT REF'ED	02110000
MLBS	EQU	RUFLNG+2	MAX. LINE BUFFER SIZE @01	02120000
IPCT	EQU	MLBS+1	# OF INPUT CAPABLE TERMINALS @01	02130000
BLKLSV	EQU	IPCT+2	BLKL SAVE AREA @01	02140000
		SPACE		02150000
RESV1	EQU	BLKLSV+1	RESERVED. @01	02160000
CWEND	EQU	RESV1+1	LNG OF COMMON WORK AREA.	02170000
		MEND		02180000

// END

@2 CT@EJ I
PROGRAM END

\$MAINT01

*
// PAUSE

2 CR@90 D 0 3 STEP
// PAUSE