

===== > **NMODPROC.01E** =====>

***** Genral procedures and utilities to be run before any other PROC file
*

PROC GMRTSTART
* PROC GMRTSTART define variables used in GMRT procs
SCALAR TSV,USV,VSV,WSV,XSV,YSV,ZSV,IPA,JPA,KPA
FINI

PROC SLEEP(IPA)
* PROC SLEEP(IPA) sleep for IPA sec
XSV=COMM
COMM=IPA
T4VERB
COMM=XSV
FINI

PROC SVDC
* PROC SVDC save DEST and COMM in XSV&YSV
XSV=DEST;YSV=COMM
FINI

PROC RSTDC
* PROC RSTDC restore DEST & COMM from XSV & YSV
DEST=XSV;COMM=YSV
FINI

PROC SVCDC
* PROC SVCDC SVDC+ZSV=CPA(1)
SVDC;ZSV=CPA(1)
FINI

PROC RSTCDC
* PROC RSTCDC RSTDC + CPA(1)=ZSV
RSTDC;CPA(1)=ZSV
FINI

PROC SVCDCS
*PROC SVCDCS
SVCDC;TSV=SUBAR
FINI

PROC RSTCDCS
*PROC RSTCDCS
RSTCDC;SUBAR=TSV
FINI

PROC SVDCA

* PROC SVDC
SVDC;TSV=AXIS
FINI

SVDC+USV=AXIS

PROC RSTDCA
* PROC RSTDCA
RSTDC;AXIS=TSV
FINI

RSTDC + AXIS=USV

PROC SVCDCA
* PROC SVCDCA
SVCDC;TSV=AXIS
FINI

SVCDC+USV=AXIS

PROC RSTCDCA
* PROC RSTCDCA
RSTCDC;AXIS=TSV
FINI

RSTCDC + AXIS=USV

PROC SVDCS
* PROC SVDCS
SVDC;TSV=SUBA
FINI

SVDC + USV=SUBARRAY

PROC RSTDCA
* PROC RSTDCA
RSTDC;SUBA=TSV
FINI

RSTDC + SUBARR=USV

PROC SVDCAT
* PROC SVDCAT
SVDCAT; USV=TARGET(1);VSV=TARGET(2);WSV=TARGET(3)
FINI

SVDCAT + TARGET in UVWSV

PROC RSTDCA
* PROC RSTDCA
RSTDCA;TARGET(1)=USV;TARGET(2)=VSV;TARGET(3)=WSV
FINI

RSTDCA + TARGET FROM UVWSV

PROC SVDCATD
* PROC SVDCATD
SVDCAT;TSV=SRVCRD
FINI

SVDCAT + SRVCRD in TSV

PROC RSTDCA
* PROC RSTDCA
RSTDCA;SRVCRD=TSV
FINI
*

RSTDCA +SRVCRD in TSV

***** PROCEDURES FOR SUN *****

*

```
PROC DEFSUB(IPA)
* PROC DEFSUB(IPA)          define antennas associated with subarray IPA
SVCD
DEST=50
COMM=20
SUBA=IPA
T3V
RSTCD
FINI
```

```
PROC SHSUB(IPA)
* PROC SHSUB(IPA)          show antennas associated with subarray IPA
SVCD
DEST=50; COMM=21
SUBA=IPA;CPA(1)=0
T3V
RSTCD
FINI
```

```
PROC STRTPROC(STRA3)
* PROC STRTPROC           starts specified process
SVDC
COMM=30;DEST=50;OUTFIL=STRA3
T3V
RSTDC
FINI
```

```
PROC ABRTPROC(STRA3)
* PROC ABRTPROC           aborts starting of specified process
SVDC
COMM=31;DEST=50;OUTFIL=STRA3
T3V
RSTDC
FINI
```

```
PROC HLTPROC(STRA3)
* PROC HLTPROC           stops specified process
SVDC
COMM=32;DEST=50;OUTFIL=STRA3
T3V
RSTDC
FINI
```

```
PROC USERCONN
* PROC USERCONN          Connect User to COMMHAND
SVDC
```

COMM=36;DEST=50;T3V
RSTDC
FINI

PROC ONDBGUNIC
* PROC ONDBGUNIC set debug on in unixcomh
SVDC
COMM 40;DEST=50;T3V
RSTDC
FINI

PROC OFFDBGUNI
* PROC OFFDBGUNI turn off debug in unixcomh
SVDC
COMM=42;DEST=50;T3V
RSTDC
FINI

PROC LOGPKT(STRA3)
* PROC LOGPKT(STRA3) start pkts logging in file STRA3
SVDC
COMM 50;DEST 50;OUTFIL=STRA3;
T3V;RSTDC
FINI

PROC HLTPKTLOG(IPA)
* PROC HLTPKTLOG(IPA) stop logging packets for file no IPA
SVDC; COMM=52;CP(1)=IPA
DEST=50;T3V;RSTCDC
FINI

PROC SHLOGLIST
* PROC SHLOGLIST list opened log files for packet log
SVDC; COMM=54;
DEST=50;T3V;RSTCDC
FINI

*****= PROCEDURES FOR HANDLING THE SERVO *****

***** Operational Command Procedures *****

PROC COLDSTART
* PROC COLDSTART Cold start of antenna
SVDC
DEST=16;COMM=40X;T3V
RSTDC
FINI

PROC MV(IPA,JPA)

```
* PROC MV(IPA,JPA)                                Move antenna to az=xsv,el=ysv (astr)
SVDCAT
TARG(2)=IPA;TARG(3)=JPA
DEST=16;COMM=42X;AXIS=0;T3V
RSTDCAT
FINI
```

```
PROC MVAZIM(IPA)
* PROC MVAZIM(IPA)                                Move antenna to az=wsv (astr)
SVDCAT
TARG(2)=IPA
DEST=16;COMM=42X;AXIS=1;T3V
RSTDCAT
FINI
```

```
PROC MVELEV(IPA)
* PROC MVELEV(IPA)           Move antenna to el=wsv (astr)
SVDCAT
TARG(3)=IPA
DEST=16;COMM=42X;AXIS=2;T3V
RSTDCAT
FINI
```

```
PROC AMV(IPA,JPA)
* PROC AMV(IPA,JPA)           move antenna to az=xsv,el=ysv (antenna)
SVDCATD
TARG(2)=IPA;TARG(3)=JPA
DEST=16;COMM=42X;AXIS=0;SRVCRD=10;T3V
RSTDCATD
FINI
```

```
PROC AMVAZIM(IPA)
* PROC AMVAZIM(IPA)                move antenna to az=xsv (antenna)
SVDCATD
TARG(2)=IPA
DEST=16;COMM=42X;AXIS=1;SRVCRD=10;T3V
RSTDCATD
FINI
```

```
PROC GOINNER
* PROC GOINNER          request antenna to move on inner track
SRVCRD=0
FINI
```

```
PROC GOOUTER
* PROC GOOUTER          request antenna to move on outer track
SRVCRD=1
```

FINI

```
PROC TRACK(JPA,KPA,IPA)
* PROC TRACK(IPA,JPA,KPA)          servo track , parms=2 angles & time
SVDCAT
TARG(1)=IPA;TARG(2)=JPA;TARG(3)=KPA
DEST=16;COMM=44x;AXIS=0;T3V
RSTDCAT
FINI
```

```
PROC TRKAZIM(JPA,IPA)
* PROC TRKAZIM(IPA,JPA)          servo track , parms=azim and time
SVDCAT
TARG(1)=IPA;TARG(2)=JPA
DEST=16;COMM=44x;AXIS=1;T3V
RSTDCAT
FINI
```

```
PROC TRKELEV(JPA,IPA)
* PROC TRKELEV(IPA,JPA)          servo track , parms=elev and time
SVDCAT
TARG(1)=IPA;TARG(3)=JPA
DEST=16;COMM=44x;AXIS=2;T3V
RSTDCAT
FINI
```

```
PROC ATRACK(JPA,KPA,IPA)
* PROC ATRACK(IPA,JPA,KPA)      servo track in ant cord, parms=azim elev& time
SVDCAT
TARG(1)=IPA;TARG(2)=JPA;TARG(3)=KPA
DEST=16;COMM=44x;AXIS=0;SRVCRD=10;T3V
RSTDCAT
FINI
```

```
PROC ATRKAZIM(JPA,IPA)
* PROC ATRKAZIM(IPA,JPA)      servo track in ant cord, parms=azim and time
SVDCAT
TARG(1)=IPA;TARG(2)=JPA
DEST=16;COMM=44x;AXIS=1;SRVCRD=10;T3V
RSTDCAT
FINI
```

```
PROC HOLD
* PROC HOLD                    request servo to release brakes and hold
SVDCA
DEST=16;COMM=46X;AXIS=0;T3V
RSTDCA
FINI
```

PROC HLDAZIM
* PROC HLDAZIM release azim brakes and hold
SVDCA
DEST=16;COMM=46X;AXIS=1;T3V
RSTDCA
FINI

PROC HLDELEV
* PROC HLDELEV release elev brakes and hold
SVDCA
DEST=16;COMM=46X;AXIS=2;T3V
RSTDCA
FINI

PROC BRAKES
* PROC BRAKES apply brakes on both axis
SVDCA
DEST=16;COMM=48x;AXIS=0;T3V
RSTDCA
FINI

PROC BRKAZIM
* PROC BRKAZIM apply brakes on azim axis
SVDCA
DEST=16;COMM=48x;AXIS=1;T3V
RSTDCA
FINI

PROC BRKELEV
* PROC BRKELEV apply brakes on elev axis
SVDCA
DEST=16;COMM=48x;AXIS=2;T3V
RSTDCA
FINI

PROC CLOSE
* PROC CLOSE close down observations
SVCDC
DEST=16;COMM=4ax;T3V
RSTCDC
FINI

PROC STOW
* PROC STOW stow the antenna
SVDCA
DEST=16;COMM=4cx;AXIS=0;T3V
RSTDCA
FINI

PROC SWELEV	
* PROC SWELEV	stow antenna in elev
SVDC	
DEST=16;COMM=4cx;AXIS=2;T3V	
RSTDCA	
FINI	
PROC SWRELE	
* PROC SWRELE	release antenna from stowed pos
SVDC	
DEST=16;COMM=4ex;AXIS=0;T3V	
RSTDCA	
FINI	
PROC SWRELEL	
* PROC SWRELEL	release antenna elevation stow
SVDC	
DEST=16;COMM=4ex;AXIS=2;T3V	
RSTDCA	
FINI	
PROC STOP	
* PROC STOP	abort servo's previous command(to stop in moving both
axes)	
SVDC	
DEST=16;COMM=50x;T3V	
RSTDC	
FINI	
PROC ABRTSRVCMD	
* PROC ABRTSRVCMD	abort servo's previous command
SVDC	
DEST=16;COMM=50x;T3V	
RSTDC	
FINI	
PROC RDSRVSPC	
* PROC RDSRVSPC	read servo set param (in srvs, servo sp. window)
SVDC	
DEST=16;COMM=3ax;t3v	
COMM=36x;t3v	
RSTDC	
FINI	
PROC RSTSERVO	
* PROC RSTSERVO	reset servo computer
SVDC	
COMM=6cx;DEST=16;T3V	

RSTDC
FINI

* Procedure to check STP flag appearance & dis-appearance

* PROC STPCHK(KPA)

PROC STPCHK(KPA)

SVDC

SRVCRD=1

MVEL(88d)

type hms(KPA)

for i=1 to 3

KPA=KPA+20m

SRVCRD=2

TRKELEV(92d,KPA)

sleep(600)

sleep(600)

KPA=KPA+20m

SRVCRD=1

TRKELEV(88d,KPA)

sleep(600)

sleep(600)

END

RSTDC

FINI

***** Procedures for Set Mode Commands *****

PROC STSRVTIM

* PROC STSRVTIM set servo time IPA sec ahead

SVDC

DEST=16;COMM=52x;T3V

RSTDC

FINI

*** Other servo set parameter procedures are in STSERVPAR.001

*

***** Procedures for ABC *****

*

PROC STMCM

* PROC STMCM set MCM for ABC using mpa(I)

SVDC; COMM=0;DEST=17;T3V;RSTDC

FINI

PROC STABCTIM

* PROC STABCTIM set ABC and Servo time, and set LST on ABC

SVDC;

COMM=1;DEST=17

T3V

RSTDC;

FINI

```
PROC STABCDLY
* PROC STABCDLY          set ABC delays to values set in TPARM(1-18)
SVDC;
COMM=2;DEST=17;
T3V;RSTDC
FINI
```

```
PROC STABCCYC
* PROC STABCDLY          set ABC default delays and cycle values
SVDC;
COMM=2;DEST=17;
TPA 1,4,4,4,10,2,10,5,2,0,1,1,5,10,30,14,15,3
T3V;RSTDC
FINI
```

```
**PROC STABC3CYC
* PROC STABC3CYC          set ABC delays and 3 sec cycle time
*SVDC;
*COMM=2;DEST=17;
*TPA 3,4,4,4,10,2,10,5,2,0,1,1,8,10,30,14,15,3
*T3V;RSTDC
*FINI
```

```
**PROC STABC4CYC
* PROC STABC4CYC          set ABC delays and 4 sec cycle time
*SVDC;
*COMM=2;DEST=17;
*TPA 4,4,4,4,10,2,10,5,2,0,1,1,8,10,30,14,15,3
*T3V;RSTDC
*FINI
```

```
PROC ENABCQ(IPA)
* PROC ENABCQ(IPA)        start queuing ABC commands for dest=IPA
SVCDC;COMM=3;DEST=17;
TPARM(1)=IPA;T3V;RSTCDC
FINI
```

```
PROC ENAMCMQ
* PROC ENAMCMQ            Enable mcm qued cmds
SVCDC
COMMAND=3;DEST=17
TPARM(1)=0;T3V
RSTCDC
FINI
```

```
PROC DISMCMQ
```

```
* PROC DISMCMQ          Disable mcm cmd quing
SV CDC
COMMA=3;DEST=17
ABCQDES=17;T3V; RSTCDC
FINI
```

```
PROC STABCDBG(IPA)
* PROC STABCDBG(IPA)      set ABC in Debug mode IPA
SV CDC;COMM=4;TPARM(1)=IPA;
DEST=17;T3V;RSTCDC
FINI
```

```
PROC ABRTABCQ(IPA,JPA)
* PROC ABRTABCQ(IPA,JPA)  abort JPA cmds fro IPA in ABC Q
SV CDC;COMM=5;TPA(1)=IPA;
TPA(2)=JPA;DEST=17;T3V;RSTCDC
FINI
```

```
PROC TALK(IPA,OUTFIL)
* PROC TALK(IPA,OUTFIL)   send talk message to ABC
SV CDC;
COMM=7;DEST=17;TPA(1)=IPA;T3V;
RSTCDC;
FINI
```

```
PROC RDABCVER
* PROC RDABCVER           read ABC program version
SVDC;COMM=10;DEST=17;T3V;RSTDC
FINI
```

```
PROC RDABCDBG
* PROC RDABCDBG          read ABC error statistics
SVDC;COMM=11;DEST=17;T3V;RSTDC
FINI
```

```
PROC RDABCDLY
* PROC RDABCDLY          read the values of ABC delay
SVDC;COMM=12;DEST=17;T3V;RSTDC
FINI
```

```
PROC DISSRVLNK
* PROC DISSRVLNK        Disable servo commuication for antenna
SV CDC
TPA(1)=0;COMM=14
DEST=17;T3VERB
RSTCDC
FINI
```

PROC ENASRVLNK	
* PROC ENASRVLNK	Enable servo communication for antenna
SVCDC	
TPA(1)=1;COMM=14	
DEST=17;T3VERB	
RSTCDC	
FINI	
PROC ABRTPRKANT	
* PROC ABRTPRKANT	Abort ante parking sequence
SVDC	
COMM=15;DEST=17;T3V	
RSTDC	
FINI	
PROC MCHABCCTR	
* PROC MCHABCCTR	match cmd -resp counters for abc
SVDC	
COMM=16;DEST=17;T3V	
RSTDC	
FINI	
PROC GOABCAPPL	
* PROC GOABCAPPL	Ask abc kernel to goto appln and execute
SVCDC	
TPA(1)=4;COMM=17	
DEST=17;T3VERB	
RSTCDC	
FINI	
PROC GOPROMAPPL	
* PROC GOPROMAPPL	Ask kernel to transfer from PROM and execute
SVCDC	
TPA(1)=5;COMM=17	
DEST=17;T3VERB	
RSTCDC	
FINI	
PROC STRTABCDNL	
* PROC STRTABCDNL	Start ABC code down loading(obj.dat in current area)
SVCDC	
TPA(1)=255;TPA(2)=300;	
COMM=17	
DEST=17;T3VERB	
RSTCDC	
FINI	
PROC RDQTIME	
* PROC RDQTIME	Read abc q timing info

SVDC
COMM=19;DEST=17;T3V
RSTDC
FINI

PROC ENALO1MON
* PROC ENALO1MON Enable lo 1 monitor for antenna: MCM 2
SVDC
TPA(1)=1;TPA(2)=2;
TPA(3)=0;COMM=20
DEST=17;T3VERB
RSTCDC
FINI

PROC ENALO2MON
* PROC ENALO2MON Enable lo 2 monitor for antenna MCM 3
SVDC
TPA(1)=1;TPA(2)=3;
TPA(3)=0;COMM=20
DEST=17;T3VERB
RSTCDC
FINI

PROC DISLO1MON
* PROC DISLO1MON Disable lo 1 monitor for antenna
SVDC
TPA(1)=0;TPA(2)=2;COMM=20
DEST=17;T3VERB
RSTCDC
FINI

PROC DISLO2MON
* PROC DISLO2MON Disable lo 2 monitor for antenna
SVDC
TPA(1)=0;TPA(2)=3;COMM=20
DEST=17;T3VERB
RSTCDC
FINI

PROC ENAIFMON
* PROC ENAIFMON Enable if monitor for antenna
SVDC
TPA(1)=1;TPA(2)=10;
TPA(3)=1;COMM=20
DEST=17;T3VERB
RSTCDC
FINI

PROC DISIFMON

```
* PROC DISIFMON          Disable if monitor for antenna
SVCDC
TPA(1)=0;TPA(2)=9;COMM=20
DEST=17;T3VERB
RSTCDC
FINI
```

```
PROC LDANTPARAM
* PROC LDANTPARAM        load ant specific param for local track(current hardcoded in proc in
TPA array)
SVCDC
*COMM=25;DEST=17;
COMM=44;DEST=17;
TPA(1)=19.1d;TPA(2)=-74.1d;TPA(3)=0d;
TPA(4)=0d;TPA(5)=260d;TPA(6)=-260d;
TPA(7)=105d;TPA(8)=17d;TPA(9)=90d;
TPA(10)=0d;T3VERB
RSTCDC
FINI
```

```
PROC LDANTOFFS(IPA,JPA)
* PROC LDANTOFFS         load ant offsets param for local track(current hardcoded in proc in TPA
array)
SVCDC
*COMM=25;DEST=17;
COMM=44;DEST=17;
TPA(1)=19.1d;TPA(2)=-74.1d;TPA(3)=IPA;
TPA(4)=JPA;TPA(5)=260d;TPA(6)=-260d;
TPA(7)=105d;TPA(8)=17d;TPA(9)=90d;
TPA(10)=0d;T3VERB
RSTCDC
FINI
```

```
PROC LDSRCPARAM(IPA,JPA)
* PROC LDSRCPARAM        load src specific params for local track:IPA=1=OUT,JPA=trckg time
SVCDC
*COMM=26;DEST=17
COMM=45;DEST=17
TPA(1)=19.1d;TPA(2)=IPA
TPA(3)=0;TPA(4)=JPA;
TPA(5)=0;T3VERB
RSTCDC
FINI
```

```
PROC LDTIMTRK
* PROC LDTIMTRK          load time params for local track
SVDC
COMM=27;DEST=17;T3VE
RSTDC
```

FINI

PROC STIFABC(IPA)

* PROC STIFABC(IPA) Set IF attn loaded in ABC for IPA freq.

SVDC

TYPE '#Set IF attn loaded in ABC for IPA freq band'

TYPE '#IPA=0=>50,1=>150,2=>235,3=>325,4=>610,5=>Lband'

COMM=40;DEST=17;

TPA(1)=IPA;T3VE

RSTDC

FINI

PROC STRTLOCTRK

* PROC STRTLOCTRK START local track mode for ant or ants in sac

SVCDC

TPA(1)=1;COMM=28

DEST=17;T3VERB

RSTCDC

FINI

PROC STPLOCTRK

* PROC STPLOCTRK STOP local track mode for ant or ants in sac

SVCDC

TPA(1)=0;COMM=28

DEST=17;T3VERB

RSTCDC

FINI

PROC ADDMCMS

* PROC ADDMCMS additionally configure MCMs defined in MPA array

SVDC

COMM=29;DEST=17;T3VE

RSTDC

FINI

PROC DELMCMS

* PROC DELMCMS for abc deselect only those MCMs defined in MPA array

SVDC

COMM=30;DEST=17;T3VE

RSTDC

FINI

PROC GOABCKER

* PROC GOABCKER ask abc to go to kernal mode

SVDC;COMM=31;DEST=17;T3V;RSTDC

FINI

PROC RSTABERR

* PROC RSTABERR reset error counters of abc

SVDC;COMM=32;DEST=17;T3V;RSTDC
FINI

PROC RDANTPARA
* PROC RDANTPARA Read ant params for local track
SVDC
COMM=33;DEST=17;T3VE
RSTDC
FINI

PROC RDSRCPARA
* PROC RDSRCPARA Read src params for local track
SVDC
COMM=34;DEST=17;T3VE
RSTDC
FINI

PROC RDTIMTRK
* PROC RDTIMTRK Read time params for local track
SVDC
COMM=35;DEST=17;T3VE
RSTDC
FINI

PROC STRTLOCTRK
* PROC STRTLOCTRK START local track mode for ant or ants in sac
SVCDC
TPA(1)=1;COMM=28
DEST=17;T3VERB
RSTCDC
FINI

PROC INITABCMD
* PROC INITABCMD Init abc cmd: tally cmd-resp counters, reset flag reset, stoptalk, rdabcv, stabct
SVCDC
COMM=23;DEST=17;T3V
STABCDBG(11)
COMM=10;T3V
COMM=27;DEST=17;T3VE
COMM=1; T3V;
mpa 5 0 2 3 10 14;stmcm
COMMAND=290;DEST=21;T3V;
ana 0 0 0 03ffx
COMM 101;DEST 0;T3V
COMM 110;T3V;
stabcdbg(1);
ldantpara
TPA 1,4,4,4,10,2,10,5,2,0,1,1,5,10,30,14,15,3

COMM=103;DEST=IPA;T3V;
RSTDC
FINI

PROC STANA(IPA)
* PROC STANA(IPA) set analog mask for MCM IPA
SVDC
COMM=110;DEST=IPA;T3V;
RSTDC
FINI

PROC ST16DIG(IPA)
* PROC ST16DIG(IPA) set 16 bit digital mask
SVDC
COMM=120;DEST=IPA;T3V;
RSTDC
FINI

PROC ST32DIG(IPA)
* PROC ST32DIG(IPA) set 32 bit digital mask
SVDC
COMM=130;DEST=IPA;T3V;
RSTDC
FINI

PROC ST64DIG(IPA)
* PROC ST64DIG(IPA) set 64 bit digital mask
SVDC
COMM=140;DEST=IPA;T3V;
RSTDC
FINI

*PROC STMCMTHT ??????
*SVDC
*COMM=150;DEST=?;T3V
*RSTDC
*FINI

PROC RDANA(IPA)
* PROC RDANA(IPA) read analog values
SVDC
COMM=200;DEST=IPA;T3V;
RSTDC
FINI

PROC RD16DIG(IPA)
* PROC RD16DIG(IPA) read 16 bit digital mask
SVDC
COMM=210;DEST=IPA;T3V;

RSTDC
FINI

PROC RD32DIG(IPA)
* PROC RD32DIG(IPA) read 32 bit digital mask
SVDC
COMM=220;DEST=IPA;T3V;
RSTDC
FINI

PROC RDMCMVER(IPA)
* PROC RDMCMVER(IPA) read MCM program version
SVDC
COMM=230;DEST=IPA;T3V;
RSTDC
FINI

PROC RD64DIG(IPA)
* PROC RD64DIG(IPA) read 64 bit digital mask
SVDC
COMM=240;DEST=IPA;T3V;
RSTDC
FINI

PROC RDMODE(IPA)
* PROC RDMODE(IPA) read current mode
SVDC
COMM=250;DEST=IPA;T3V;
RSTDC
FINI

PROC RDTHRVAL(IPA)
* PROC RDTHRVAL(IPA) read threshold vals
SVDC
COMM=260;DEST=IPA;T3V;
RSTDC
FINI

PROC FEEDSEL(IPA)
* PROC FEEDSEL(IPA) feed select old
SVDC
COMM=300;DEST=IPA;T3V;
RSTDC
FINI

PROC RBMCM(IPA)
* PROC RBMCM(IPA) reboot MCM
SVDC
COMM=500;DEST=IPA;T3V;

RSTDC
FINI

PROC FEEDSELM(IPA)
* PROC FEEDSELM(IPA) feed select modified
SVDC
COMM=600;DEST=IPA;T3V;
RSTDC
FINI

*

***** Procedures for astronomical functions *****

*

PROC GTSRC(STRA3)
* PROC GTSRC get source ra,dec & precess
SVDC
SOURCE=STRA3
COMM=30; T5V
COMM=10; T5V
COMM=1; T5V
OUTFIL=""
COMM=26;DEST=91;T3v
RSTDC
FINI

PROC POSN
* PROC POSN move antenna to current position of source
SVDCA
COMM=20;T5V
DEST=16;AXIS=0;COMM 42x;T3V
RSTDCA
FINI

PROC ADDLIST(OUTFIL)
* PROC ADDLIST(OUTFIL) add source list in 'outfil' to default list
COMM=50; T5V
FINI

PROC SHLIST
* PROC SHLIST show the path of source lists available
COMM=58; T5V
FINI

PROC DELLIST(IPA)
* PROC DELLIST(IPA) del source list by num IPA given by shlist
SVCDC
COMMA=52
CP(1)=IPA; T5V

RSTCDC
FINI

***** PROCEDURES FOR COMMAND MONITOR *****

PROC STCMOFF(IPA)
* PROC STCMOFF(IPA) Set IPS sec as cmd mon offset
SVCDC
CPA(1)=IPA;COMM=0
DEST=33;T3V
RSTCDC
FINI

PROC ENACMDMON
* PROC ENACMDMON Enable Command Monitor
SVCDC
CPA(1)=1;COMM=1
DEST=33;T3V
RSTCDC
FINI

PROC DISCMDMON
* PROC DISCMDMON Disable Command Monitor
SVCDC
TYPE '#!!!WARNING!!! NOWONWARDS CMD EXECUTION STATUS WILL BE
*****UNKNOWN*****!!!'
CPA(1)=0;COMM=1
DEST=33;T3V
RSTCDC
FINI

PROC ENACMINFO
* PROC ENACMINFO Enable Command Monitor Info Log
SVCDC
TYPE '# CMD EXECUTION STATUS WILL BE REPORTED . NORMAL MODE.'
CPA(1)=1;COMM=2
DEST=33;T3V
RSTCDC
FINI

PROC DISCMINFO
* PROC DISCMINFO Disable Command Monitor Info Log
SVCDC
CPA(1)=0;COMM=2
DEST=33;T3V
RSTCDC
FINI

PROC ABRTDNLD

```
* PROC ABRTDNLD          abort abc program down load
SVDC
COMM=3;DEST=33;T3V
RSTDC
FINI
```

```
PROC STDNLDPKT(IPA)
* PROC STDNLDPKT(IPA) aet abc program down load pkt size=IPA bytes
SVCDC
COMM=4;CPA(1)=IPA;
DEST=33;T3V; RSTCDC
FINI
```

```
*
***** The procedures related to display of logged data from ***
***** file  for serialdebug program are in SERDBGPRO.001 *****
*
*
***** PROCEDURES FOR Sub Array Controller (SAC) *****
*
```

```
PROC SNDSACSRC(IPA,JPA)
* PROC SNDSACSRC          send source coords to subarray controller modified to send
trk/posn,TMAX
SVDC; COMM=10;DEST=70;
CPARM(1)=IPA;CPARM(2)=JPA;
T3V;RSTDC
FINI
```

```
PROC STSACTOLR
* PROC STSACTOLR          set tolerance for subarray controller using cpa
SVDC; COMM=54;DEST=70;T3V;RSTDC
FINI
```

```
PROC SNDSACANT
* PROC SNDSACANT          send antenna mask to subarray contrommer
SVDC;CP(1)=0;COMM=12;DEST=70;
T3V;RSTCDC
FINI
```

```
PROC ADDSACANT
* PROC SNDSACANT          send antenna mask to subarray contrommer
SVCDC;CPA(1)=1;COMM=12;DEST=70;T3V;RSTCDC
FINI
```

```
PROC DELSACANT
* PROC SNDSACANT          remove antenna mask to subarray contrommer
SVCDC;CPA(1)=-1;COMM=12;DEST=70;T3V;RSTCDC
```

FINI

PROC TRKSACSRC(IPA,JPA)

* PROC TRKSACSRC request subarray controller to track source
SVCDC;COMM=20;DEST=70;CPARM(1)=IPA;
CPARM(2)=JPA;T3V;RSTCDC
FINI

PROC STPSACTRK

* PROC STPSACTRK request SAC to stop tracking source
SVDC;COMM=30;DEST=70;T3V;RSTDC
FINI

PROC GOSACOUT

* PROC GOSACOUT request SAC to track on outer track
SVDC;COMM=6;DEST=70;T3V;RSTDC
FINI

PROC GOSACINN

* PROC GOSACINN request SAC to track on inner track
SVDC;COMM=8;DEST=70;T3V;RSTDC
FINI

PROC ONSACDBG

* PROC ONSACDBG request SAC to turn on debug mode
SVDC;COMM=16;DEST=70;T3V;RSTDC
FINI

PROC OFFSACDBG

* PROC OFFSACDBG request SAC to turn off debug mode
SVDC;COMM=18;DEST=70;T3V;RSTDC
FINI

PROC TRKELOFF(IPA)

* PROC TRKELOFF(IPA) track elevation offset by IPA
CPA(1)=-1;
SCO(17)=IPA;COMM=10;DEST=70;T3V
FINI

PROC TRKAZOFF(IPA)

* PROC TRKAZOFF(IPA) track azimuth offset by IPA
CPA(1)=-1
SCO(15)=IPA;COMM=10;DEST=70;T3V
FINI

PROC TRKANTOFF(JPA,IPA)

* PROC TRKANTOFF(IPA,JPA) track azimuth, elevation offset by JPA,IPA
CPA(1)=-1;
SCO(15)=JPA;

```
SCO(17)=IPA;  
COMM=10;DEST=70;T3V  
FINI
```

```
PROC TRKRAOFF(IPA)  
* PROC TRKRAOFF(IPA)          track right ascension offset by IPA  
CPA(1)=-1  
SCO(10)=IPA;COMM=10;DEST=70;T3V  
FINI
```

```
PROC TRKDECOFF(IPA)  
* PROC TRKDECOFF(IPA)        track declination offset by IPA  
CPA(1)=-1  
SCO(12)=IPA;COMM=10;DEST=70;T3V  
FINI
```

```
PROC SCANELSRC(IPA,JPA)  
* PROC SCANELSRC(IPA,JPA)    scan src in el with derv=ipa,ptime=jpa  
SCO(18)=IPA;SCO(19)=JPA  
SVDC;CP(1)=1;CP(2)=1h;CP(3)=0  
COMM=10;DEST=70;T3V;RSTDC  
FINI
```

```
PROC SCANAZSRC(IPA,JPA)  
* PROC SCANAZSRC(IPA,JPA)    scan src in az with derv=ipa,ptime=jpa  
SCO(16)=IPA;SCO(19)=JPA  
SVDC;CP(1)=1;CP(2)=1h;CP(3)=0  
COMM=10;DEST=70;T3V;RSTDC  
FINI
```

```
PROC SCANRASRC(IPA,JPA)  
* PROC SCANRASRC(IPA,JPA)    scan src in ra with derv=ipa,ptime=jpa  
SCO(11)=IPA;SCO(14)=JPA  
SVDC;CP(1)=1;CP(2)=1h;CP(3)=0  
COMM=10;DEST=70;T3V;RSTDC  
FINI
```

```
PROC SCANDEC(IPA,JPA)  
* PROC SCANDEC(IPA,JPA)      scan src in dec with derv=ipa,ptime=jpa  
SCO(13)=IPA;SCO(14)=JPA  
SVDC;CP(1)=1;CP(2)=1h;CP(3)=0  
COMM=10;DEST=70;T3V;RSTDC  
FINI
```

```
PROC SNDSACCMD(OUTFIL)  
* PROC SNDSACCMD(OUTFIL)    send local pops cmd to sac from outfil  
SVCDC  
COMMAND=2;DEST=70  
T3VERB;RSTDC
```


FINI

PROC STRTSACFIL

* PROC STRTSACFIL start remote control for sac from opened file

SVCDC

COMMAND=64;DEST=70

T3VERB;RSTDC

FINI

PROC STPSACFIL

* PROC STPSACFIL stop remote control for sac from opened file

SVCDC

COMMAND=66;DEST=70

T3VERB;RSTDC

FINI

PROC OPSACFILE(OUTFIL)

* PROC OPSACFILE(OUTFIL) open a sac control file from OUTFIL

SVCDC

COMMAND=60;DEST=70

T3VERB;RSTDC

FINI

PROC CLSACFILE(OUTFIL)

* PROC OPSACFILE(OUTFIL) close a sac control file

SVCDC

COMMAND=62;DEST=70

T3VERB;RSTDC

FINI

PROC SHSACFILE

* PROC SHSACFILE show a sac control file which is open

SVCDC

COMMAND=70;DEST=70

T3VERB;RSTDC

FINI

PROC SHSACLINE

* PROC SHSACLINE show current line of sac control file

SVCDC

COMMAND=71;DEST=70

T3VERB;RSTDC

FINI

PROC REWSACFILE

* PROC REWSACFILE rewind the sac control file

SVCDC

COMMAND=72;DEST=70

T3VERB;RSTDC

FINI

```
PROC MVSACCON(IPA)
* PROC MVSACCON(IPA)      move control to point(ipa) in sac file
SVCDC;CPA(1)=IPA
COMMAND=73;DEST=70
T3VERB;RSTDC
FINI
```

```
PROC SKPSACLINE(IPA)
* PROC SKPSACLINE(IPA)    skip n=ipa lines frmo sac coontrol file
SVCDC;CPA(1)=IPA
COMMAND=74;DEST=70
T3VERB;RSTDC
FINI
```

```
PROC STPSACLINE(IPA)
* PROC STPSACLINE(IPA)    step by one line frmo sac coontrol file
SVCDC;CPA(1)=IPA
COMMAND=74;DEST=70
T3VERB;RSTDC
FINI
```

```
PROC SUBHNDLE
* PROC SUBHNDLE           subarray handle to catter tracking generic cmd
SVCDC;
COMMAND=55;DEST=70
T3VERB;RSTDC
FINI
```

```
*
***** FOR NEW FPS: FPS97 *****
**** These are procedures related to the control of the FEED rotation ****
*
```

```
PROC NULLFPS
* PROC NULLFPS           issue null cmd
SVCDC
COMMAND=0;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC STTNGPNT(IPA)
** PROC STTNGPT(IPA)      issue set turning point cp(1) : turning pt angle/pulse
SVCDC;FPSMODE=0
COMMAND=100;DEST=21;
CP(1)=IPA;T3V;
RSTCDC
```

FINI

PROC STRMPDCNT

** PROC STRMPDCNT issue set ramp down count

SVCDC;FPSMODE=0

COMMAND=110;DEST=21;T3V;

RSTCDC

FINI

PROC STLRPMLMT

** PROC STLRPMLMT issue set lower rpm limit

SVCDC;FPSMODE=0

COMMAND=120;DEST=21;T3V;

RSTCDC

FINI

PROC STBCTDIF(IPA)

* PROC STBRCNTDIFF(IPA) issue set brake count difference count cp(1) : pusles

SVCDC;FPSMODE=0

COMMAND=130;DEST=21;

CP(1)=IPA;T3V;

RSTCDC

FINI

PROC STRUPCNT

** PROC RMPUPTMCNT issue set ramp up time count

SVCDC;FPSMODE=0

COMMAND=140;DEST=21;T3V;

RSTCDC

FINI

PROC STSTPTCT

** PROC STPTMCNT issue set stop time count

SVCDC;FPSMODE=0

COMMAND=150;DEST=21;T3V;

RSTCDC

FINI

PROC STMAXPWM

** PROC STMAXPWM issue set Max pwm cnt

SVCDC;FPSMODE=0

COMMAND=160;DEST=21;T3V;

RSTCDC

FINI

PROC STMAXANG(IPA)

* PROC STMAXANGLE(IPA) issue set Max angle cp(1)=max ang

SVCDC;FPSMODE=0

COMMAND=170;DEST=21;

CP(1)=IPA;CP(2)=270;T3V;
RSTCDC
FINI

PROC STMINANG(IPA)
* PROC STMINANGLE(IPA) issue set Min angle cp(1)=min ang
SVCDC;FPSMODE=0
COMMAND=180;DEST=21;
CP(1)=IPA;CP(2)=270;T3V;
RSTCDC
FINI

PROC RDTNGPNT
** PROC RDSTTNGPT issue read turning point
SVCDC
COMMAND=200;DEST=21;T3V;
RSTCDC
FINI

PROC RDRMPDCT
** PROC RDRMPDCT issue read ramp down count
SVCDC
COMMAND=210;DEST=21;T3V;
RSTCDC
FINI

PROC RDLRPMLMT
** PROC RDLRPMLMT issue read lower rpm limit
SVCDC
COMMAND=220;DEST=21;T3V;
RSTCDC
FINI

PROC RDBRCTDIF
* PROC RDBRCNTDIFF issue read brake count difference count
SVCDC
COMMAND=230;DEST=21;T3V;
RSTCDC
FINI

PROC RDRMPUPCT
* PROC RDRMPUPTMCNT issue read ramp up time count
SVCDC
COMMAND=240;DEST=21;T3V;
RSTCDC
FINI

PROC RDSTPTMCT

```
** PROC RDSTPTMCNT          issue read stop time count
SVCDC
COMMAND=250;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC RDMAXPWM
** PROC RDMAXPWM          issue read Max pwm cnt
SVCDC
COMMAND=260;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC RDMAXANG
* PROC RDMAXANGLE        issue read Max angle
SVCDC
COMMAND=270;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC RDMINANG
* PROC RDMINANGLE        issue read Min angle
SVCDC
COMMAND=280;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC RDVERSION
* PROC RDVERSION          issue read version
SVCDC
COMMAND=290;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC RDUA0ANG
* PROC RDUA0ANG          issue read UA0 angle
SVCDC
COMMAND=700;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC GETUA0ANG
* PROC RDUA0ANG          get the first set UA0 angle
SVCDC
COMMAND=700;DEST=21;T3V;
RSTCDC
FINI
```

PROC UA0CLBRT(IPA)

* PROC RDUA0ANG calibrate to UA0 angle CP=1=>Clkwise i.e. -10d
side,0=>anticlkwise
SVCDC
COMMAND=360;DEST=21;
CPA(1)=IPA;T3V;
RSTCDC
FINI

PROC RUNCLBRT

* PROC RUNCLBRT issue run to calibrate
SVCDC
COMMAND=300;DEST=21;T3V;
RSTCDC
FINI

PROC FREER10

* PROC FREERUN10 issue free run towards -10deg lim s/w
SVCDC
COMMAND=310;DEST=21;CPA(1)=1;T3V;
RSTCDC
FINI

PROC FREER280

* PROC FREERUN280 issue free run towards 280 deg lim s/w
SVCDC
COMMAND=310;DEST=21;
CPA(1)=0;T3V;
RSTCDC
FINI

PROC RUNDPREST(IPA)

* PROC RUNDPREST(IPA) issue run to preset, IPA target ang
SVCDC;FPSMODE=0
COMMAND=320;DEST=21;
CP(1)=IPA;CP(2)=270;T3V;
RSTCDC
FINI

PROC RUNCPREST(IPA)

* PROC RUNCPREST(IPA) issue run to preset, IPA target counts
SVCDC;FPSMODE=1
COMMAND=320;DEST=21;
CP(1)=IPA;CP(2)=17000;T3V;
RSTCDC
FINI

PROC FINECTUNE(IPA,JPA)

* PROC FINECTUNE(IPA,JPA) issue run to preset,IPA is target counts, JPA is PWM counts

```
SVCDC;FPSMODE=1
COMMAND=330;DEST=21;
CP(1)=IPA;CP(2)=17000;
CPA(3)=JPA;T3V;
RSTCDC
FINI
```

```
PROC FINEDTUNE(IPA,JPA)
* PROC FINEDTUNE(IPA,JPA)    issue run to preset,IPA is target ang,JPA is PWM counts
SVCDC;FPSMODE=0
COMMAND=330;DEST=21;
CP(1)=IPA;CP(2)=270;
CPA(3)=JPA;T3V;
RSTCDC
FINI
```

```
PROC RUNPASSWD
* PROC RUNPASSWD            issue run passworded
SVCDC
COMMAND=340;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC FPSBOOT
* PROC FPSBOOT              issue reboot fps
SVCDC
COMMAND=500;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC FPSSTOP
* PROC FPSSTOP              issue stop fps
SVCDC
COMMAND=600;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC PRSTCFPS(IPA)
* PROC PRSTCFPS(IPA)        preset known position of feed in counts
SVCDC
COMMAND=800;DEST=21
FPSMODE=1;TPARM(1)=IPA
T3V;RSTCDC
FINI
```

```
PROC PRSTAFPS(IPA)
* PROC PRSTAFPS(IPA)        preset known position of feed in angle
SVCDC
COMMAND=800;DEST=21
```

```
FPSMODE=0;TPARM(1)=IPA
T3V;RSTCDC
FINI
```

*

**** These are procedures related to the NEW DAS Control **

*

```
PROC LNKNDASQ
* PROC LNKNDASQ    init das cntrl for subarray IPA
SVCDC
COMMA=10;DEST=91;
T3V;
RSTCDC
FINI
```

```
PROC STRTNDASC
* PROC STRTDASCAN    start das scan for subarray
SVCDC
COMMA=11;DEST=91;T3V;
RSTCDC
FINI
```

```
PROC STPNDESC
* PROC STPDASCAN    start das scan for subarray
SVCDC
COMMA=12;DEST=91;T3V;
RSTCDC
FINI
```

```
PROC SNDNDASSTR(OUTFIL)
* PROC SNDASSTR(OUTFIL) send command thro string for das
SVCDC
COMMA=13;DEST=91;T3V;
RSTCDC
FINI
```

```
PROC PRJFREQ
* PROC PRJTITLE    set das frequencies (in tPARM) with Code in OUTFIL
SVCDC
TYPE '#TPARM SHOULD HAVE RF(2) IF(2) BB(2) REST FR(2) LSR(2)(MHZ) Bandmask'
COMMA=24;DEST=91;T3V;
RSTCDC
FINI
```

```
PROC LDSRCCODE(OUTFIL)
* PROC LDSRCCODE(OUTFIL) set in outfil the source code
SVDC
COMMA=26; DEST=91; T3V;
RSTDC
```


FINI

*

**** These are procedures related to the FPS commands **

*

PROC LDFPSPOS

* PROC LDFPSPOS(IPA) Loads the fps counts in TPA

* TPARM should have the 4 FPS encoder positions

SVCDC

FPSMODE=1

* FPSMODE=IPA

COMMAND=37;DEST=17;T3V;

RSTCDC

FINI

PROC MVFPS610

* PROC MVFPS610 Mv FPS to 610 to counts loaded by above command

SVCDC

TPA=0

COMMA=38;DEST=17;T3V;

RSTCDC

FINI

PROC MVFPS150

* PROC MVFPS150 Mv FPS to 150 to counts loaded by above command

SVCDC

TPA=1

COMMA=38;DEST=17;T3V;

RSTCDC

FINI

PROC MVFPS1420

* PROC MVFPS1420 Mv FPS to 1420 to counts loaded by above command

SVCDC

TPA=2

COMMA=38;DEST=17;T3V;

RSTCDC

FINI

PROC MVFPS325

* PROC MVFPS325 Mv FPS to 325 to counts loaded by above command

SVCDC

TPA=3

COMMA=38;DEST=17;T3V;

RSTCDC

FINI

PROC INITFPS

* PROC INITFPS Does add MCM 14,rdv, set min angle to -10d, stlrpm. respectively

```
SVCDC
COMMAND=290;DEST=21;T3V;
FPSMODE=0;COMMAND=180;DEST=21;
CP(1)=-18;CP(2)=270;T3V;
COMMAND=280;DEST=21;T3V;
CP(1)=550;
COMMAND=120;DEST=21;T3V;
RSTCDC
FINI
```

```
PROC ALLANT
*PROC ALLANT    Defines all ante except abc0 in antenna array
SVDC
FOR I=2 to 31;ANTE(I)=I-1;end
ANTE(1)=30
RSTDC
FINI
```

```
PROC YANT
*PROC YANT      Defines all Y antenna array
SVDC
ANTE 16 26 27 28 29 30 17 18 21 22 23 8 13 14 15 16 25
RSTDC
FINI
```

```
PROC CANT
*PROC CANT      Defines all CSQ antenna array
SVDC
ANTE 14 7 6 5 1 3 19 20 24 4 12 9 2 11 10
RSTDC
FINI
```

```
PROC INITNFPS
*PROC INITNFPS  new init fps
SVDC
RDVER
CPA(1)=700;STLRPM
STMINA(-18)
RSTDC
FINI
```

```
* Read current time
PROC GETTIME
SVDC
COMMAND=-1;T4V
type scalr1
type hms(scalr1)
RSTDC
FINI
```

** PULSAR DAS PROCEDURES , dated : 17mar2005.

* initialise pulsar (to be run in USER0)

```
PROC INITPSR
SVCDC
COMMAND=1;DEST=92;T3V;
RSTCDC
FINI
```

* link to pulsar das

```
PROC LNKPSRQ
SVCDC
COMMAND=10;DEST=92;T3V;
RSTCDC
FINI
```

* ia das init

```
PROC IAINIT
SVCDC
OUTFIL 'DASIA INIT IA_INIT.HDR'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

* ia das start

```
PROC IASTRT
SVCDC
OUTFIL 'DASIA START IA_SCAN.HDR'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

* ia das start

```
PROC IASTRT1(STRA2)
SVCDC
OUTFIL 'DASIA START IA_SCAN.HDR'
OUTFIL=OUTFIL!! STRA2
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

* ia das stop

```
PROC IASTP
SVCDC
OUTFIL 'DASIA STOP'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* ia das finish
PROC IAFIN
SVCDC
OUTFIL 'DASIA FINISH'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* pa das init
PROC PAINIT
SVCDC
OUTFIL 'DASPA INIT PA_INIT.HDR'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* pa das start
PROC PASTRT
SVCDC
OUTFIL 'DASPA START PA_SCAN.HDR'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* pa das start
PROC PASTRT1(STRA2)
SVCDC
OUTFIL 'DASPA START PA_SCAN.HDR'
OUTFIL=OUTFIL!! STRA2
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* pa das stop
PROC PASTP
SVCDC
OUTFIL 'DASPA STOP'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* pa das finish
PROC PAFIN
SVCDC
OUTFIL 'DASPA FINISH'
COMMAND=11;DEST=92;T3V;
RSTCDC
```

FINI

```
* polmtr das init
PROC PMTINIT
SVCDC
OUTFIL 'DASPMT INIT PMT_INIT.HDR'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* polmtr das start
PROC PMTSTRT
SVCDC
OUTFIL 'DASPMT START PMT_SCAN.HDR'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* polmtr das start
PROC PMTSTRT1(STRA2)
SVCDC
OUTFIL 'DASPMT START PMT_SCAN.HDR'
OUTFIL=OUTFIL!! STRA2
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* polmtr das stop
PROC PMTSTP
SVCDC
OUTFIL 'DASPMT STOP'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* polmtr das finish
PROC PMTFIN
SVCDC
OUTFIL 'DASPMT FINISH'
COMMAND=11;DEST=92;T3V;
RSTCDC
FINI
```

```
* polmtr both sidebands das init
PROC PMBINIT
SVCDC
OUTFIL 'DASPMB INIT PMT_INIT.HDR'
COMMAND=11;DEST=92;T3V;
RSTCDC
```

FINI

* polmtr both sidebands das start

PROC PMBSTRT

SVCDC

OUTFIL 'DASPMB START PMT_SCAN.HDR'

COMMAND=11;DEST=92;T3V;

RSTCDC

FINI

* polmtr both sidebands das start

PROC PMBSTRT1(STRA2)

SVCDC

OUTFIL 'DASPMB START PMT_SCAN.HDR'

OUTFIL=OUTFIL!! STRA2

COMMAND=11;DEST=92;T3V;

RSTCDC

FINI

* polmtr both sidebands das stop

PROC PMBSTP

SVCDC

OUTFIL 'DASPMB STOP'

COMMAND=11;DEST=92;T3V;

RSTCDC

FINI

* polmtr both sidebands das finish

PROC PMBFIN

SVCDC

OUTFIL 'DASPMB FINISH'

COMMAND=11;DEST=92;T3V;

RSTCDC

FINI

*

***** PROCEDURES FOR NEW BB *****

*

PROC INITBBSRV

* PROC INITBBSRV Init BB serv

SVDC

TYPE '# bbserv must be running'

COMM=1;DEST=22; T3V

RSTDC

FINI

* OLD BASEBAND PROC COMMENTED 01 MAY 2013 - JPK

* * PROC STBBWGNALL(IPA,JPA) Set for both Pol and Sidebands:IPA BW and JPA Gain

```

* PROC STBBWGNALL(IPA,JPA)
* SVDC
* cp(9)=9;stra2=(CHAR)IPA;
* STRA3=(CHAR)JPA;
* outfil 'set,bwgn,both,both,'
* outfil=outfil!!stra2!!','!!stra3!!';
* comm 10;dest 22;t3v
* RSTDC
* FINI
*

```

```

* PROC STBBANDALL(IPA)
* *PROC STBBANDALL(IPA)      Set for both Pol and Sidebands:IPA Bandwidth MHz
* SVDC
* cp(9)=9;stra2=(CHAR)IPA;
* outfil 'set,band,both,both,'
* outfil=outfil!!stra2!!';
* comm 10;dest 22;t3v
* RSTDC
* FINI
*

```

```

* PROC STBGAINALL(IPA)
* *PROC STBGAINALL(IPA)      Set for both Pol and Sidebands:IPA Gain
* SVDC
* cp(9)=9;stra1=(CHAR)IPA;
* outfil 'set,gain,both,both,'
* outfil=outfil!!stra1!!';
* comm 10;dest 22;t3v
* RSTDC
* FINI
*

```

```

* PROC STBBNDGAIN(STRA2,STRA3,IPA,JPA)
* * USAGE : STBBNDGAIN('CHANNEL','SIDE BAND','BW','GAIN')
* *PROC STBBNDGAIN(STRA2,STRA3,IPA,JPA)      Set for STRA2 Pol and STRA3
sideband:IPA BW and JPA Gain: stra2=130,175,both; stra3=usb,lsb,both
* SVDC
* cp(9)=9;
* outfil 'set,bwgn,'
* outfil=outfil!!stra2!!','!!stra3
* stra2=(CHAR)IPA;
* outfil=outfil!!','!!STRA2
* STRA1=(CHAR)JPA;
* outfil=outfil!!','!!STRA1!!';
* comm 10;dest 22;t3v
* RSTDC
* FINI

```

```

PROC STBBLO(STRA2,STRA3)
*PROC STBBLO(STRA2,STRA3)  Set baseband lo:STRA2=LO1,STRA3=LO2
SVDC

```

```
cp(9)=9
outfil 'set,bblo,'
outfil=outfil!!stra2!!',!!stra3!!',!!'1,1'!!';'
comm 10;dest 22;t3v
RSTDC
FINI
```

```
PROC STGSBLO(STRA2,STRA3)
*PROC STGSBLO(STRA2,STRA3)      Set baseband lo:STRA2=LO1,STRA3=LO2
SVDC
cp(9)=9
outfil 'set,gsblo,'
outfil=outfil!!stra2!!',!!stra3!!',!!'1,1'!!';'
comm 12;dest 22;t3v
RSTDC
FINI
```

```
PROC INITGABSRV
* PROC INITGABSRV              Init GSB serv
SVDC
TYPE '# gab must be running'
COMM=11;DEST=23; T3V
RSTDC
FINI
```

```
PROC STGABLO(STRA2,STRA3)
*PROC STGABLO(STRA2,STRA3)      Set GAB lo:STRA2=LO1,STRA3=LO2
SVDC
cp(9)=9
outfil 'set,gablo,'
outfil=outfil!!stra2!!',!!stra3!!',!!'1,1'!!';'
comm 12;dest 23;t3v
RSTDC
FINI
```

```
PROC STGABCONF(STRA2)
*PROC STGABLO(STRA2)           set GAB 32bit Configuration word
SVDC
cp(9)=9
outfil 'set,gabconf,'
outfil=outfil!!stra2!!';'
comm 12;dest 23;t3v
RSTDC
FINI
```

```
PROC STGABATTN(STRA2,STRA3)
*PROC STGABLO(STRA2,STRA3)      Set GAB
ATTN:STRA2=CH1_ATTEN,STRA3=CH1_ATTEN
SVDC
```



```
cp(9)=9
outfil 'set,gabattn,'
outfil=outfil!!stra2!!',!!stra3!!',!!'1,1'!!';'
comm 12;dest 23;t3v
RSTDC
FINI
```

```
* ALL BB PROC SAVED IN <THIS_FILE>_working_19dec2012
*
```

```
* Procedure written to automate grid pointing on different source
* required for finding a pointing model. Please note that
* TPARM(1)=11; TPARM(2)=2 and TPARM(3)=1/2 has been enabled
* by Pramesh only for subarray 5/user5 -NGK 7Feb2006
* Modified further on 8Feb2006
* For testing use e.g. gopntg(-80,20,9,'3C48') - for 325 MHz.
* For testing use e.g. gopntg(-40,10,9,'3C48') - for 610 MHz.
* For testing use e.g. gopntg(-20,5,9,'3C48') - for 1280 MHz.
* change source name depending on availability of source
```

```
**SCRATCH DEFVAR
** DOFFA, DOFFE, DANG Diagonal SLEW OFFSETS IN AZ EL, ANGLE
**SCRATCH GOPNTG
PROC DEFVAR
  scalar gridmin,gridsp,ngrid,offset,eleva
  scalar doffa, doffe, dang
  string*20 src
FINI
```

```
* Temporary Procedures to check gopntg proc 2008 Jan 30th.
* IPA= (1:OUT, 0:IN), JPA = TRACKING HOUR
PROC TLOCTRK(IPA,JPA)
SVCDC
TPA 0;TARGET 0
COMM=45;DEST=17
TPA(1)=19.1d;TPA(2)=IPA
TPA(3)=0;TPA(4)=JPA
TPA(5)=0;T3VERB
SLEEP 4
TPA 0;TARGET 0
COMM 22;T5V
TPA(4)=TARGET(1)
TPA(5)=TARGET(2)
TPA(1)=1
TPA(2)=JPA
COMM=28;DEST=17;
T3VERB
RSTDC
FINI
```

```

PROC ELGOPNTG(GRIDMIN,GRIDSP,NGRID,SRC,IPA)
SVCDC
TPA 0;SCO 0
FOR J=0 TO NGRID-1
TPARM(1)=11; TPARM(2)=2; TPARM(3)=J
GTS SRC
OFFSET=(GRIDMIN+(J*GRIDSP))/60 *(3.14159/180)
CPA(1)=-1
SCO(17)=OFFSET
KPA=IPA
TLOCTRK(IPA,1h)
IPA=KPA
TLOCTRK(IPA,1h)
SLEEP 5
STRTNDC
SLEEP 85
STPNDC
IPA=KPA
TPA 0;SCO 0
END
RSTCDC
FINISH

```

```

** PROC TRFIAZ(GRIDMIN,GRIDSP,NGRID,SRC,IPA)
*SVCDC
*TPA 0;SCO 0
*FOR J=0 TO NGRID-1
*TPARM(1)=11; TPARM(2)=1; TPARM(3)=J
*GTS SRC
** Procedure to get the Target Elevation
*COMM=20;SCOORD(9)=0;T5V
*ELEVA=TARGET(3)
*OFFSET=(GRIDMIN+(J*GRIDSP))/60 * (3.14159/180)
*CPA(1)=-1; CPA(5)=2
*SCO(15)=OFFSET/COS(ELEVA)
*KPA=IPA
*TLOCTRK(IPA,1h)
*IPA=KPA
*TLOCTRK(IPA,1h)
*SLEEP 5
*STRTNDC
*SLEEP 10
*STPNDC
*IPA=KPA
*END
*RSTCDC
*FINISH

```

```

PROC AZGOPNTG(GRIDMIN,GRIDSP,NGRID,SRC,IPA)

```

```

SVCDC
TPA 0;SCO 0
FOR J=0 TO NGRID-1
TPARM(1)=11; TPARM(2)=1; TPARM(3)=J
GTS SRC
* Procedure to get the Target Elevation
COMM=20;SCOORD(9)=0;T5V
ELEVA=TARGET(3)
OFFSET=(GRIDMIN+(J*GRIDSP))/60 * (3.14159/180)
CPA(1)=-1; CPA(5)=2
SCO(15)=OFFSET/COS(ELEVA)
KPA=IPA
TLOCTRK(IPA,1h)
IPA=KPA
TLOCTRK(IPA,1h)
SLEEP 5
STRTNDAS
SLEEP 85
STPNDESC
IPA=KPA
END
RSTCDC
FINISH

```

```

PROC GOPNTG(GRIDMIN,GRIDSP,NGRID,SRC,IPA)
SVCDC
ELGOPNTG(GRIDMIN,GRIDSP,NGRID,SRC,IPA)
TIME 1S
AZGOPNTG(GRIDMIN,GRIDSP,NGRID,SRC,IPA)
RSTCDC
FINISH

```

```

* NEW LDANTOFF
* load ant specific offset for local track for all antennas one by one
*TPA(1)=Antenna Latitude, TPA(2)=Antenna Longitude
* TPA(3)=Azimuth Offset, TPA(4)=Elevation Offset

```

```

PROC LDTPANT
TPA 0;TPA(5)=260d;TPA(6)=-260d;
TPA(7)=105d;TPA(8)=17d;TPA(9)=90d;
FINISH

```

```

PROC C00OFF(JPA,KPA)
SVCDC
ANTE 1 7; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT;
TPA(1)=19.0929d;TPA(2)=-74.0570d;
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00700d;TPA(10)=TPA(10)*1000.0/15.0

```

TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C01
PROC C01OFF(JPA,KPA)
SVCDC
ANTE 1 6; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT;
TPA(1)=19.0927d;TPA(2)=-74.0536d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00357d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C02
PROC C02OFF(JPA,KPA)
SVCDC
ANTE 1 5; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT;
TPA(1)=19.0931d;TPA(2)=-74.0505d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00047d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C03
PROC C03OFF(JPA,KPA)
SVCDC
ANTE 1 1; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0944d;TPA(2)=-74.0469d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.00307d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C04
PROC C04OFF(JPA,KPA)
SVCDC
ANTE 1 3; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0931d;TPA(2)=-74.0505d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00047d;TPA(10)=TPA(10)*1000.0/15.0

TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C05
PROC C05OFF(JPA,KPA)
SVCDC
ANTE 1 19; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0908d;TPA(2)=-74.0511d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00111d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C06
PROC C06OFF(JPA,KPA)
SVCDC
ANTE 1 20; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0910d;TPA(2)=-74.0502d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00017d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C08
PROC C08OFF(JPA,KPA)
SVCDC
ANTE 1 24; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0893d;TPA(2)=-74.0531d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00314d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C09
PROC C09OFF(JPA,KPA)
SVCDC
ANTE 1 4; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0917d;TPA(2)=-74.0509d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00087d;TPA(10)=TPA(10)*1000.0/15.0

TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C10
PROC C10OFF(JPA,KPA)
SVCDC
ANTE 1 12; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0875d;TPA(2)=-74.0489d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.00110d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C11
PROC C11OFF(JPA,KPA)
SVCDC
ANTE 1 9; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0901d;TPA(2)=-74.0447d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.00527d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C12
PROC C12OFF(JPA,KPA)
SVCDC
ANTE 1 2; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0871d;TPA(2)=-74.0521d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00213d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=C13
PROC C13OFF(JPA,KPA)
SVCDC
ANTE 1 11; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0825d;TPA(2)=-74.0444d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.00561d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE

RSTCDC
FINISH

*ANT=C14
PROC C14OFF(JPA,KPA)
SVCDC
ANTE 1 10; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0871d;TPA(2)=-74.0460d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.00404d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=E02
PROC E02OFF(JPA,KPA)
SVCDC
ANTE 1 17; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1023d;TPA(2)=-74.0772d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.02721d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=E03
PROC E03OFF(JPA,KPA)
SVCDC
ANTE 1 18; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1117d;TPA(2)=-74.0940d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.04395d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

*ANT=E04
PROC E04OFF(JPA,KPA)
SVCDC
ANTE 1 21; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1209d;TPA(2)=-74.1244d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.07441d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC

FINISH

*ANT=E05

PROC E05OFF(JPA,KPA)

SVCDC

ANTE 1 22; CP 0;DEFS 0;SUBAR 0

COMM=44;DEST=17;LDTPANT

TPA(1)=19.1251d;TPA(2)=-74.1474d

TPA(3)=JPA;TPA(4)=KPA

TPA(10)=+0.09740d;TPA(10)=TPA(10)*1000.0/15.0

TPARM(12)=1;T3VE

RSTCDC

FINISH

*ANT=E06

PROC E06OFF(JPA,KPA)

SVCDC

ANTE 1 23; CP 0;DEFS 0;SUBAR 0

COMM=44;DEST=17;LDTPANT

TPA(1)=19.1365d;TPA(2)=-74.1652d

TPA(3)=JPA;TPA(4)=KPA

TPA(10)=+0.11522d;TPA(10)=TPA(10)*1000.0/15.0

TPARM(12)=1;T3VE

RSTCDC

FINISH

*ANT=S01

PROC S01OFF(JPA,KPA)

SVCDC

ANTE 1 26; CP 0;DEFS 0;SUBAR 0

COMM=44;DEST=17;LDTPANT

TPA(1)=19.0664d;TPA(2)=-74.0565d

TPA(3)=JPA;TPA(4)=KPA

TPA(10)=+0.00649d;TPA(10)=TPA(10)*1000.0/15.0

TPARM(12)=1;T3VE

RSTCDC

FINISH

*ANT=S02

PROC S02OFF(JPA,KPA)

SVCDC

ANTE 1 27; CP 0;DEFS 0;SUBAR 0

COMM=44;DEST=17;LDTPANT

TPA(1)=19.0523d;TPA(2)=-74.0470d

TPA(3)=JPA;TPA(4)=KPA

TPA(10)=-0.00302d;TPA(10)=TPA(10)*1000.0/15.0

TPARM(12)=1;T3VE

RSTCDC

FINISH


```
*ANT=S03
PROC S03OFF(JPA,KPA)
SVCDC
ANTE 1 28; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0320d;TPA(2)=-74.0536d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00363d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```
*ANT=S04
PROC S04OFF(JPA,KPA)
SVCDC
ANTE 1 29; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0074d;TPA(2)=-74.0595d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=+0.00947d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```
*ANT=S06
PROC S06OFF(JPA,KPA)
SVCDC
ANTE 1 30; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=18.9653d;TPA(2)=-74.0470d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.00304d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```
*ANT=W01
PROC W01OFF(JPA,KPA)
SVCDC
ANTE 1 8; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.0988d;TPA(2)=-74.0353d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.01466d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```
*ANT=W02
PROC W02OFF(JPA,KPA)
SVCDC
ANTE 1 13; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1066d;TPA(2)=-74.0210d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.02898d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```
*ANT=W03
PROC W03OFF(JPA,KPA)
SVCDC
ANTE 1 14; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1208d;TPA(2)=-74.0011d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.04895d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```
*ANT=W04
PROC W04OFF(JPA,KPA)
SVCDC
ANTE 1 15; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1415d;TPA(2)=-73.9836d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.06643d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```
*ANT=W05
PROC W05OFF(JPA,KPA)
SVCDC
ANTE 1 16; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1678d;TPA(2)=-73.9734d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.07656d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH
```

```

*ANT=W06
PROC W06OFF(JPA,KPA)
SVCDC
ANTE 1 25; CP 0;DEFS 0;SUBAR 0
COMM=44;DEST=17;LDTPANT
TPA(1)=19.1784d;TPA(2)=-73.9436d
TPA(3)=JPA;TPA(4)=KPA
TPA(10)=-0.10644d;TPA(10)=TPA(10)*1000.0/15.0
TPARM(12)=1;T3VE
RSTCDC
FINISH

```

```

PROC DGOPNTG(GRIDMIN,GRIDSP,NGRID,DANG,SRC,IPA)
SVCDC
FOR J=0 TO NGRID-1
TPA 0;SCO 0
TPARM(1)=11; TPARM(2)=2; TPARM(3)=J
GTS SRC
* Procedure to get the Target Elevation
COMM=20;SCOORD(9)=0;T5V
ELEVA=TARGET(3)
OFFSET=((GRIDMIN+(J*GRIDSP))/60)*cos(DANG)*(3.14159/180)
CPA(1)=-1; CPA(5)=2
SCO(15)=OFFSET/COS(ELEVA)
SCO(17)=((GRIDMIN+(J*GRIDSP))/60)*sin(DANG)*(3.14159/180)
KPA=IPA; TLOCTRK(IPA,1h)
IPA=KPA; TLOCTRK(IPA,1h)
SLEEP 5
STRTNDAS
SLEEP 85
STPNDASC
IPA=KPA
END
RSTCDC
FINISH

```