

## Introduction for the terms used in this document

**COMMAND** is set by the user to specify to T3VERB the nature of the command. The combination of COMMAND and DESTIN uniquely identifies a command ( apart from additional adverbs). The actual interpretation of COMMAND will depend on **DESTIN**.

**DESTIN** is set by the user to specify to T3VERB the destination of the command. The combination of DESTIN and COMMAND uniquely identifies a command ( apart from additional adverbs)

Acceptable values of DESTIN are

0-15 for MCM numbers 0 to 15  
16 for SERVO  
17 for ABC  
20 for COMH  
33 for Command-Monitor  
50 for SUN system  
60 for DISPLAY handler  
70 for SUBARRAY controller  
80 for OFFline DEBUG mode

### **General procedures available for the GMRT ONLINE system**

The following temporary scalars have been defined for usage in GMRT procedures

**USAV, VSAV, WSAV, XSAV, YSAV, ZSAV, IPA, JPA**

List of available procedures and their corresponding subroutines are :

Procedures	Subroutine
ABCPROC	ABCCMD
SUNPROC	SUNCMD
COMHPROC	COMHCMD
SERVOPROC	SERVOCMD
FPSPROC	FPSCMD

DISPPROC	DISPCMD
TESTPROC	TESTCMD
ASTROPROC	ASTROCMD
GENPROC	GENPROC
MCMPROC	MCMCMD
SACPROC	SACCMD

The above procedures and their subroutines are explained in **Appendix** i.e. at the end of this document.

## Procedures related to the New DAS control (Destination = 91)

All these commands are T3V commands.

LNKNDASQ(Destination = 91 Command No = 10)

**Initialise/ links DAS(Data Acqisition System) control for sub-array IPA with correlator.**

This command first refers a **NEWDASCMD** subroutine in the **CMDFRM.FOR** file, and initialises/links a DAS for current sub-array(i.e. Suba4,Suba5 etc.) if DAS is already started. Otherwise it shouts.

<STRTNDAS>(Destination = 91 Command No = 11)

**Start DAS scan for the sub-array.**

This command first refers a **NEWDASCMD** subroutine in the **CMDFRM.FOR** file, and starts a DAS for current sub-array (i.e. Suba4,Suba5 etc.) There are various checks related which this subroutine does for this command the checks are :

- 1.If DAS is not initialised then it gives a message that **":DAS START SCAN FAILED, DAS NOT INITTED ... CMD FAILED"**.
- 2.If DAS is started but project is not initialised then gives a message **":DAS START SCAN FAILED, NO PROJECT RUNNING"**.
- 3.If DAS and Project are initialised and scan had given strtndas command then it gives message **":DAS START SCAN FAILED, SCAN ALREADY RUNNING"**.
- 4.If DAS and project are initialised and scan is not started then it starts the scan but after starting scan it checks the source name which is given before this command searches the source in the lists added in the user console and then if the source doesn't exist in the sky or the source name is incorrect then it shouts.

STPNNDAS(Destination = 91 Command No = 12)

**Stops DAS scan for the sub-array.**

This command first refers a **NEWDASCMD** subroutine in the **CMDFRM.FOR** file, and stops a DAS for current sub-array (i.e. Suba4,Suba5 etc.) There are various checks related which this subroutine does for this command the checks are :

- 1.If DAS is not initialised then it gives a message that **":DAS STOP SCAN FAILED, DAS NOT INITTED ... CMD FAILED"**.

2.If DAS is started but project is not initialised then gives a message **":DAS STOP SCAN FAILED, NO PROJECT RUNNING"**.

3.If DAS and Project are initialised and scan had already given stpnDas command then it gives message **":DAS STOP SCAN FAILED, SCAN NOT RUNNING"**.

**SNDNDASSTR(OUTFIL)**(Destination = 91 Command No = 13)

**This command sends command through a string of DAS.**

This command first refers a **NEWDASCMD** subroutine in the **CMDFRM.FOR** file.

**PRJFREQ**(Destination = 91 Command No = 24)

**Set das frequencies in the format RF1 RF2 L01 L01 L04 L04.**

This command first refers a **NEWDASCMD** subroutine in the **CMDFRM.FOR** file. It give inputs of frequencies and source information to the scan fields in the recording file.

**<LDSRCCODE(OUTFIL)**(Destination = 91 Command No = 26)

**Set source code in the outfile.**

This command first refers a **NEWDASCMD** subroutine in the **CMDFRM.FOR** file.

## Procedures for Sub-array Controller (Destination = 70)

**Note :- All commands are in this destination are T3V commands.**

**SNDSACSRC**(Destination = 70 Command No = 10)

**Sends source co-ordinate to the subarray controller.**

If the command destination matches with 70 and command no. 10 then it will refer to the subroutine **SUBACMD**(in the CMDFRM.FOR) which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND**(in IPCAIPS.FOR) to the relevant sub-array.

**GOSACIN**(Destination = 70 Command No = 8)

**Requests Sub-array controller to track inner track.**

It refers subroutine **SUBACMD** generate command and send it through subroutine **SUBACSND** to the sub-array.

The effect of this command will be seen after giving **sndsacsrc** command.

**GOSACOUT**(Destination = 70 Command No = 6)

**Requests Sub-array controller to track outer track.**

It refers subroutine **SUBACMD** generate command and send it through subroutine **SUBACSND** to the sub-array.

The effect of this command will be seen after giving **sndsacsrc** command.

**TRKELOFF**(SCO(17)=IPA;Destination = 70 Command No = 10)

**Tracks source at offsets=IPA in Elevation.**

**(SCO(17)=IPA+Dest=70+COMM=10->TRKELOFF(IPA))->SUBACMD(CMDFRM.FOR)**

**SUBACMD(IPCAIPS.FOR)->SUBACSND**

If SCO(17) + Dest=70 + Command No. = 10 then refers to tracking offsets in Elevation.

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**TRKAZOFF**(SCO(15)=IPA;Destination = 70 Command No = 10)

**Tracks source at offsets=IPA in Azimuth.**

**(SCO(15)=IPA+Dest=70+COMM=10->TRKAZOFF(IPA))->SUBACMD(CMDFRM.FOR)**

**SUBACMD(IPCAIPS.FOR)->SUBACSND**

If SCO(15) + Dest=70 + Command No. = 10 then refers to tracking offsets

in Azimuth.

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**TRKANTOFF**(SCO(15)=JPA;SCO(17)=IPA;Destination = 70 Command No = 10)

**Tracks source at**

**AZ offsets = JPA**

**EL offsets=IPA.**

(SCO(15)=JPA+SCO(17)=IPA;Dest=70+COMM=10->TRKANTOFF(JPA,IPA))-  
>SUBACMD(CMDFRM.FOR)

**SUBACMD(IPCAIPS.FOR)->SUBACSND**

If SCO(15) + SCO(17) + Dest=70 + Command No. = 10 then refers to tracking offsets in Azimuth and Elevation.

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**TRKRAOFF**(SCO(10)=IPA;Destination = 70 Command No = 10)

**Tracks source at offsets=IPA in Right Ascension.**

(SCO(10)=IPA+Dest=70+COMM=10->TRKRAOFF(IPA))->SUBACMD(CMDFRM.FOR)

**SUBACMD(IPCAIPS.FOR)->SUBACSND**

If SCO(10) + Dest=70 + Command No. = 10 then refers to tracking offsets in Right Ascension.

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**TRKDECOFF**(SCO(12)=IPA;Destination = 70 Command No = 10)

**Tracks source at offsets=IPA in Declination.**

(SCO(12)=IPA+Dest=70+COMM=10->TRKDECOFF(IPA))->SUBACMD(CMDFRM.FOR)

**SUBACMD(IPCAIPS.FOR)->SUBACSND**

If SCO(12) + Dest=70 + Command No. = 10 then refers to tracking offsets in Declination.

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SCANELSRC**(Destination = 70 Command No = 10)

**Scans source in Elevation with scanrate=ipa, peak-time=jpa.**

It also takes following internal commands :

SCO(18)=IPA;SCO(19)=JPA

CP(1)=1;CP(2)=1h;CP(3)=0

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SCANAZSRC** (Destination = 70 Command No = 10)

**Scans source in Azimuth with scanrate=ipa, peak-time=jpa.**

It also takes following internal commands :

SCO(16)=IPA;SCO(19)=JPA

CP(1)=1;CP(2)=1h;CP(3)=0

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SCANRASRC** (Destination = 70 Command No = 10)

**Scans source in Right Ascention with scanrate=ipa, peak-time=jpa.**

It also takes following internal commands :

SCO(11)=IPA;SCO(14)=JPA

CP(1)=1;CP(2)=1h;CP(3)=0

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SCANDEC** (Destination = 70 Command No = 10)

**Scans source in Declination with scanrate=ipa, peak-time=jpa.**

It also takes following internal commands :

SCO(13)=IPA;SCO(14)=JPA

CP(1)=1;CP(2)=1h;CP(3)=0

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SNDSACCMD** (Destination = 70 Command No = 2)

**Send local pops (i.e. user shell) commands through a sub-array-controller from a given command file.**

It refers to the subroutine **SUBACMD** which will generate command for the

SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**STRTSACFIL**(Destination = 70 Command No = 64)

**Gives remote control of the sub-array-controller to the opened command file in the user shell.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**STPSACFIL**(Destination = 70 Command No = 66)

**Stops remote control given to the sub-array-controller to the opened command file in the user shell.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**OPSACFILE**(Destination = 70 Command No = 60)

**Opens sub-array-controller file in the user shell.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**CLSACFILE**(Destination = 70 Command No = 62)

**Close sub-array-controller file in the user shell.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SHSACFILE**(Destination = 70 Command No = 70)

**Shows a sub-array-controller which is open in the user shell.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SHSACLIN**(Destination = 70 Command No = 71)

**Shows a current line running for the running sub-array-controller file in the user shell.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.



**RWSACFILE**(Destination = 70 Command No = 72)

**Rewind sub-array-controller file in the user shell.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**MVSACCON(IPA)**(Destination = 70 Command No = 73)

**Move control to the point i.e. (IPA) in SAC file.**

It takes the argument which is in the bracket i.e.

CPA(1)=IPA

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SKPSACLINE(IPA)**(Destination = 70 Command No = 74)

**Skip no of lines i.e. (IPA) from SAC file.**

It takes the argument which is in the bracket i.e.

CPA(1)=IPA

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**MVSACCON(IPA)**(Destination = 70 Command No = 74)

**Step by one line i.e. (IPA) from SAC file.**

It takes the argument which is in the bracket i.e.

CPA(1)=IPA

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

**SUBHANDLE**(Destination = 70 Command No = 55)

**Sub-array handle to catter tracking generic command.**

It refers to the subroutine **SUBACMD** which will generate command for the SUBARRAY controller and send them through a subroutine **SUBACSND** to the relevant sub-array.

## Procedures for Command-Monitor (Destination = 33)

**STMCMOFF(IPA)** (Destination = 33 Command No = 0)

**Set IPA seconds as a command monitoring offset i.e. it generates a set offset command for command monitoring.** It refers subroutine **MONCMD** and writes message in the command console as "'OFFSET FOR CMD MONITOR '".

**ENACMDMON** (Destination = 33 Command No = 1)

**Enables Command monitoring.**

It refers subroutine **MONCMD** and set command check value to 0 to enable command monitoring.

**DISCMDMON** (Destination = 33 Command No = 1)

**Disables Command monitoring.**

It refers subroutine **MONCMD** and set command check value to 1 to disable command monitoring.

**ENACMINFO** (Destination = 33 Command No = 2)

**Enables Command monitor information log.**

It refers subroutine **MONCMD** and set command check value to 1(i.e. CPA(1)=1). Also writes message in the command console as "'ENABLE CMD MONITOR INFO'".

**DISCMINFO** (Destination = 33 Command No = 2)

**Disables Command monitor information log.**

It refers subroutine **MONCMD** and set command check value to 0(i.e. CPA(1)=0). Also writes message in the command console as "'DISABLE CMD MONITOR INFO'".

**ABORTDNLD** (Destination = 33 Command No = 3)

**Abort ABC download operation for a sub-array.**

It refers subroutine **MONCMD** and writes message in the command console as "'ABRTING DOWNLOAD'".

**STDNLDPKT(IPA)** (Destination = 33 Command No = 4)

**Set ABC program download packet size=IPA (CPA(1)=IPA is as argument).**

It refers subroutine **MONCMD** and writes message in the command console as "'DNLOAD PKT SIZE = Entered IPA value'".

## Procedures for FPS (Destination = 21)

**NULLFPS**(Destination = 21 Command No = 0)

**This issues a null command.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This subroutine checks that if command no. is 0 then it sends a null command to the FPS.

**STTNGPNT(IPA)**(Destination = 21 Command No = 100)

**It issues a command to set turning point to cp(1): turning point angle/pulse.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine **NEWFPSCMD** has one check i.e. if **FPSMODE = 0** then it selects 0<sup>th</sup> command of this subroutine+fpsmode and mode 0 i.e. **FPSMODE=0** that is why this command has command no. 100

**CP(1)=IPA** which takes angle per pulse.

**STRMPDCNT**(Destination = 21 Command No = 110)

**It issues a command to set ramp down count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine **NEWFPSCMD** has one check i.e. if **FPSMODE = 0** then it selects 1<sup>st</sup> command of this subroutine+fpsmode and mode 0 i.e. **FPSMODE=0** that is why this command has command no. 110.

**STLRPMLMT**(Destination = 21 Command No = 120)

**It issues a command to set lower RPM limit.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine **NEWFPSCMD** has one check i.e. if **FPSMODE = 0** then it selects 2<sup>nd</sup> command of this subroutine+fpsmode and mode 0 i.e. **FPSMODE=0** that is why this command has command no. 120.

**STBCTDIF(IPA)**(Destination = 21 Command No = 130)

**It issues a command to set brake count difference cp(1) : pulses.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine **NEWFPSCMD** has one check i.e. if **FPSMODE = 0** then it selects 3<sup>rd</sup> command of this subroutine+fpsmode and mode 0 i.e. **FPSMODE=0** that is why this command has command no. 130.

**CP(1)=IPA**

**STRUPCNT**(Destination = 21 Command No = 140)

**It issues a command to set ramp up count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine NEWFPSCMD has one check i.e. if FPSMODE = 0 then it selects 4<sup>th</sup> command of this subroutine+fpsmode and mode 0 i.e. FPSMODE=0 that is why this command has command no. 140.

**STSTPTCT**(Destination = 21 Command No = 150)

**It issues a command to set stop time count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine NEWFPSCMD has one check i.e. if FPSMODE = 0 then it selects 5<sup>th</sup> command of this subroutine+fpsmode and mode 0 i.e. FPSMODE=0 that is why this command has command no. 150.

**STMAXPWM**(Destination = 21 Command No = 160)

**It issues a command to set maximum PWM command.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine NEWFPSCMD has one check i.e. if FPSMODE = 0 then it selects 6<sup>th</sup> command of this subroutine+fpsmode and mode 0 i.e. FPSMODE=0 that is why this command has command no. 160.

**STMAXANG(IPA)**(Destination = 21 Command No = 170)

**It issues a command to set maximum angle through argument cp(1)=max angle.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine NEWFPSCMD has one check i.e. if FPSMODE = 0 then it selects 7<sup>th</sup> command of this subroutine+fpsmode and mode 0 i.e. FPSMODE=0 that is why this command has command no. 170.

CP(1)=IPA;CP(2)=270

**STMINANG(IPA)**(Destination = 21 Command No = 180)

**It issues a command to set minimum angle through argument cp(1)=minimum angle.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file. The subroutine NEWFPSCMD has one check i.e. if FPSMODE = 0 then it selects 8<sup>th</sup> command of this subroutine+fpsmode and mode 0 i.e. FPSMODE=0 that is why this command has command no. 180.

CP(1)=IPA;CP(2)=270

**RDTPGPNP**(Destination = 21 Command No = 200)

**It issues a command to read turning point.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 100.

**RDRMPDCT**(Destination = 21 Command No = 210)

**It issues a command to read ramp count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 110.

**RDLRPMLMT**(Destination = 21 Command No = 220)

**It issues a command to read low RPM count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 120.

**RDBRCTDIF**(Destination = 21 Command No = 230)

**It issues a command to read brake count difference.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 130.

**RDRMPUPCT**(Destination = 21 Command No = 240)

**It issues a command to read ramp up time count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 140.

**RDSTPTMCT**(Destination = 21 Command No = 250)

**It issues a command to read stop time count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 150.

**RDMAXPWM**(Destination = 21 Command No = 260)

**It issues a command to read maximum pwm count.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 160.

**RDMAXANG**(Destination = 21 Command No = 270)

**It issues a command to read maximum angle.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 170.

**RDMINANG**(Destination = 21 Command No = 280)

**It issues a command to read minimum angle.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 180.

**RDVERSION**(Destination = 21 Command No = 290)

**It issues a command to read version.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 190.

**RDVERSION**(Destination = 21 Command No = 290)

**It issues a command to read version.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

This command is related to the previous command no. 190.

**RDUAOANG**(Destination = 21 Command No = 700)

**It issues a command to read UAO angle.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

**GETUAOANG**(Destination = 21 Command No = 700)

**It issues a command to get the first set UAO angle.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

**UAOCLBRT(IPA)**(Destination = 21 Command No = 360)

**It issues a command to calibrate to UAO angle if CP=1 then it will calibrate in clockwise direction i.e. -10 deg side and if CP=0 then it will calibrate in anti-clockwise direction.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

CPA(1)=IPA.

**RUNCLBRT**(Destination = 21 Command No = 300)

**It issues a command to calibrate the feed.**

This procedure refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

**FREER10**(Destination = 21 Command No = 310)

**It issues a command to free run the feed towards -10deg limit switch.**  
This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.  
CPA(1)=1 is an inbuilt parameter.

**FREE280**(Destination = 21 Command No = 310)

**It issues a command to free run the feed towards 280deg limit switch.**  
This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.  
CPA(1)=0 is an inbuilt parameter.

**RUNDPREST(IPA)**(Destination = 21 Command No = 320)

**It issues a command to run the feed to preset value, IPA is the target angle.**  
This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.  
This takes an argument CP(1)=IPA;CP(2)=270.

**RUNCPREST(IPA)**(Destination = 21 Command No = 320)

**It issues a command to run the feed to preset value, IPA is the target count.**  
This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.  
This takes an argument CP(1)=IPA;CP(2)=17000.

**FINECTUNE(IPA,JPA)**(Destination = 21 Command No = 330)

**It issues a command to run the feed to preset value, IPA is the target count and JPA is PWM counts.**  
This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.  
This takes an argument CP(1)=IPA; CP(2)=17000; CPA(3)=JPA.

**FINEDTUNE(IPA,JPA)**(Destination = 21 Command No = 330)

**It issues a command to run the feed to preset value, IPA is the target angle and JPA is PWM counts.**  
This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.  
This takes an argument CP(1)=IPA; CP(2)=270; CPA(3)=JPA.

**RUNPASSWD**(Destination = 21 Command No = 340)

**It issues a command to run password.**  
This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

**FPSBOOT**(Destination = 21 Command No = 500)

**It issues a command to reboot fps.**

This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

**FPSSTOP**(Destination = 21 Command No = 600)

**It issues a command to stop fps.**

This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

**PRSTCFPS(IPA)**(Destination = 21 Command No = 800)

**It issues a command to run a feed to the known feed counts.**

This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

FPSMODE=1; TPARAM(1)=IPA.

**PRSTDFPS(IPA)**(Destination = 21 Command No = 800)

**It issues a command to run a feed to the known feed angle.**

This procedures refers **NEWFPSCMD** subroutine in **CMDFRM.FOR** file.

FPSMODE=0; TPARAM(1)=IPA.



## Procedures for ABC (Destination = 17)

These procedures refers **ABCCMD** subroutine in **CMDFRM.FOR** file.

**<LDFPSPOS>**(Destination = 17 Command No = 37)

Loads the FPS counts in given TPA.

This procedures refers **ABCCMD** subroutine in **CMDFRM.FOR** file.

In this TPARM should have the 4 FPS encoder positions.

**MVFPS610**(Destination = 17 Command No = 38)

**MVFPS150**(Destination = 17 Command No = 38)

**MVFPS1420**(Destination = 17 Command No = 38)

**MVFPS325**(Destination = 17 Command No = 38)

Moves FPS to 610/150/1420/325 to counts loaded by the LDFPSPOS command.

This procedures refers **ABCCMD** subroutine in **CMDFRM.FOR** file.

**INITFPS**(Destination = 17 Command No = 21)

Does add MCM 14 set minimum angle to -10deg, set lower rpm limit.

**STMCM**(Destination = 17 Command No = 0)

Configures/Sets MCMs for ABC using mpa(I).

**STABCTIME**(Destination = 17 Command No = 1)

Set ABC(Antenna Base Computer) and Servo (SCC) time to the online machine's time.

**STABCDLY**(Destination = 17 Command No = 2)

Set ABC communication delays to values set in TPARM(1-18).

**STABCCYC**(Destination = 17 Command No = 2)

Set ABC default delays and cycle values to values.

i.e. TPARM values are

TPA 1,4,4,4,10,2,10,5,2,0,1,1,5,10,30,14,15,3

**ENABCQ(IPA)**(Destination = 17 Command No = 3)

Start queing ABC commands for destination=IPA. Gives messages  
"WHATEVER COMMANDS YOU ENTER NOW WILL BE QUED".

ENAMCMQ(Destination = 17 Command No = 3)

Enables MCM qued commands.

Dismcmq(Destination = 17 Command No = 3)

Disables MCM command queing.

STABCDLG(IPA)(Destination = 17 Command No = 4)

Set ABC in debug mode to given IPA.

SABRTABCQ(IPA,JPA)(Destination = 17 Command No = 5)

Aborts JPA commands from IPA in ABC queue.

Talk(IPA,OUTFIL)(Destination = 17 Command No = 7)

Send talk message to ABC also internally refers a different subroutine  
MKANTMSK(POTERR,LEN,TMPSTR,NINT(XSUBAR)).

RDABCVER(Destination = 17 Command No = 10)

Read ABC program version i.e. It gets ABC version number from ABCCMD  
subroutine.

RDABCDLG(Destination = 17 Command No = 11)

Read ABC error statistics.

RDABCDLY(Destination = 17 Command No = 12)

Get ABC timing information.

DISSRVLNK(Destination = 17 Command No = 14)

Disables servo communication for antenna.

ENASRVLNK(Destination = 17 Command No = 14)

Enables servo communication for antenna.

ABRTPRKANT(Destination = 17 Command No = 15)

**Aborts antenna parking sequence.**

**MCHABCCTR**(Destination = 17 Command No = 16)

**Match command response counters at ABC.**

**GOABCAPPL**(Destination = 17 Command No = 17)

**Ask ABC kernel to goto application and execute.**

**GOPROMAPPL**(Destination = 17 Command No = 17)

**Ask kernel to transfer from PROM and execute.**

**STRTABCDNL**(Destination = 17 Command No = 17)

**It starts ABC code down loading(where object dat in current area).**

**TPA(1)=255;TPA(2)=300.**

**RDQTIME**(Destination = 17 Command No = 19)

**It reads ABC and Servo queue timing information.**

**ENALO1MON**(Destination = 17 Command No = 20)

**Enables LO1 monitoring through MCM-2.**

**Input parameters are as follows :**

**TPA(1)=1;TPA(2)=2;**

**TPA(3)=0**

**ENALO2MON**(Destination = 17 Command No = 20)

**Enables LO1 monitoring through MCM-3.**

**Input parameters are as follows :**

**TPA(1)=1;TPA(2)=3;**

**TPA(3)=0.**

**DISLO1MON**(Destination = 17 Command No = 20)

**Disables LO1 monitoring through MCM-2.**

**Input parameters are as follows :**

**TPA(1)=0;TPA(2)=2.**

**DISLO2MON**(Destination = 17 Command No = 20)

**Disables LO2 monitoring through MCM-3.**

Input parameters are as follows :

TPA(1)=0;TPA(2)=3;

**ENAIFMON**(Destination = 17 Command No = 20)

**Enable IF monitoring and command goes for MCM-10.**

Input parameters are as follows :

TPA(1)=1;TPA(2)=10;

TPA(3)=1.

**DISIFMON**(Destination = 17 Command No = 20)

**Disables IF monitoring for the antenna.**

**Note :** This is an old command which was given for MCM-9

The Input parameters are as follows :

TPA(1)=0;TPA(2)=9

**LDANTPARAM**(Destination = 17 Command No = 44)

**Loads antenna specific parameters for local track (current hardcoded in proc in TPA array)**

The TPA array is as follows :

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

**LDANTOFFS(IPA,JPA)**(Destination = 17 Command No = 44)

**Load antenna offsets parameters for local track (current hardcoded in proc in TPA array).**

**This is an old command which was used for loading antenna offset values nowadays we use /pntmod and run pmanto.**

The TPA parameters are as follows :

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

**LDSRCPARAM(IPA,JPA)**(Destination = 17 Command No = 45)

**Load source specific parameters for local track where IPA=1=OUT**

**,JPA=tracking time.**

The TPA parameters given for this command are as follows :

TPA(1)=19.1d;TPA(2)=IPA

TPA(3)=0;TPA(4)=JPA;

TPA(5)=0

**LDTIMETRK**(Destination = 17 Command No = 27)

**Load time parameters for local track.**

**STIFABC(IPA)**(Destination = 17 Command No = 40)

**Set IF attenuations loaded in ABC for IPA frequency.**

**STRTLOCTRK**(Destination = 17 Command No = 28)

**START local track mode for antenna/antennas in SAC(Sub-Array-Controller).**

The value of TPA given is :

TPA(1)=1.

**STPLOCTRK**(Destination = 17 Command No = 28)

**STOP local track mode for antenna/antennas in SAC(Sub-Array-Controller).**

The value of TPA given is :

TPA(A)=1.

**ADDMCMS**(Destination = 17 Command No = 29)

**Sdditionally configure MCMS defined in MPA array.**

**DELMCMS**(Destination = 17 Command No = 30)

**For ABC deselect only those MCMS defined in MPA array.**

**GOABCKER**(Destination = 17 Command No = 31)

**Ask ABC to go to KERNAL mode.**

**RSTABERR**(Destination = 17 Command No = 32)

**It counts the reset errors of ABC.**

**RDANTPARA**(Destination = 17 Command No = 33)

**Read antenna parameters for local track.**

**RDSRCPARA**(Destination = 17 Command No = 34)

**Read source parameters for local track.**

**RDTIMTRK**(Destination = 17 Command No = 35)

**Read time parameters for local track.**

**STRTLOCTRK**(Destination = 17 Command No = 28)

**START local track mode for antenna/antennas in SAC(Sub-Array-Controller).**

**INITABCMD**(Destination = 17 Command No = 23)

**Init ABC cmd: tally command response counters, reset flag reset, stoptalk, rdabcv, stabct, all these commands are given through one single command.**

**Following are the commands send thorough the above command :**

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

**ENAABCCMD**(Destination = 17 Command No = 23)

**Start accepting ABC commands for ABCs (init ABC after ABC RESET).**

**Following are the TPA parameters given through this command :**

TPA 1,4,4,4,10,2,10,5,2,0,1,1,5,10,30,14,15,3

STABCDLY

## Procedures for SUN (Destination = 50)

**DEFSUB(IPA)**(Destination = 50 Command No = 20)

**This command defines antennas associated with the subarray given in the argument=IPA and defines antenna mask for the same subarray.**

This procedure refers **SUNCMD** subroutine, further this subroutine calls **MKANTMSK** in the same script i.e. **CMDFRM.FOR** file.

The argument SUBA=IPA.

**SHSUB(IPA)**(Destination = 50 Command No = 21)

**This command shows antennas associated with the subarray given in the argument=IPA.**

This procedure refers **SUNCMD** subroutine, further this subroutine calls **GETANTMSK** in the same script which gets antenna mask information and displays through **DSPANTMSK** i.e. display antenna mask. All these subroutines are in **CMDFRM.FOR** file.

SUBA=IPA;CPA(1)=0.

**STRTPROC(STRA3)**(Destination = 50 Command No = 30)

**This command starts specific process. Like subaray,scctask etc.**

This procedure refers **SUNCMD** subroutine, further this subroutine calls **START\_PROCESS** to start the given process. Both subroutine run in the same script i.e. **CMDFRM.FOR** file.

The argument OUTFIL=STRA3.

**ABRTPROC(STRA3)**(Destination = 50 Command No = 31)

**This command aborts started process. Like subaray,scctask etc.**

This procedure refers **SUNCMD** subroutine, and finds the process running for the online and kills them using their IPCs. This command is existing in **CMDFRM.FOR** file.

The argument OUTFIL=STRA3.

**HLTPROC(STRA3)**(Destination = 50 Command No = 32)

**This command halts process. Like subaray,scctask etc.**

This procedure refers **SUNCMD** subroutine, and finds the process running for the online and close it. This command is existing in **CMDFRM.FOR** file.

The argument OUTFIL=STRA3.

**USERCON**(Destination = 50 Command No = 36)

**This command connects the user to the command handler.**

This procedure refers **SUNCMD** subroutine. This command connects the user to the command handler via shared memory i.e.

- i) It checks error while connecting to the shared memory and give message '**ERROR STARTING SHARED MEMORY**',
- ii) If the connection is already existed then it doesn't do any action and give message ' **ALREADY STARTED.... NO ACTION**'.
- iii) Or " **ERROR IN INIT\_IPC, ISTAT=**" and then stops "SUNCMD IN CMDFRM BEFORE STARTUP"
- iv) Or give message " **ERROR IN SECSTART, ICOMM=36, CMDFRM**"

All these commands are existing in **CMDFRM.FOR** file.

**ONDBGUNIC**(Destination = 50 Command No = 40)

**This command sets debug mode ON in unixcomh.**

This procedure refers **SUNCMD** subroutine.

**OFFDBGUNI**(Destination = 50 Command No = 42)

**This command sets debug mode OFF in unixcomh.**

This procedure refers **SUNCMD** subroutine.

**LOGPKT(STRA3)**(Destination = 50 Command No = 50)

**This command starts packet logging in to the output file given in the argument.**

This procedure refers **SUNCMD** subroutine.

The argument **OUTFIL=STRA3**.

**HLPKTLOG(IPA)**(Destination = 50 Command No = 52)

**This command stops packet logging in to the output file given in the argument.**

This procedure refers **SUNCMD** subroutine.

The argument **CP(1)=IPA**.

**SHLOGLIST**(Destination = 50 Command No = 54)

**This command shows list of log files opened for logging packets for the files which are argumented in the IPA.**

This procedure refers **SUNCMD** subroutine.



## Procedures for SERVO (Destination = 16)

**COLDSTART**(Destination = 16 Command No = 40X)

**Cold start of an antenna.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

**MV(IPA,JPA)**(Destination = 16 Command No = 42X)

**Move antenna to given target astronomical azimuth and target astronomical elevation angles.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(2)=IPA;TARG(3)=JPA,

AXIS=0.

**MVAZIM(IPA)**(Destination = 16 Command No = 42X)

**Move antenna to the given target astronomical azimuth angle.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(2)=IPA,

AXIS=1.

**MVAELE(IPA)**(Destination = 16 Command No = 42X)

**Move antenna to the given target astronomical elevation angle.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(3)=IPA,

AXIS=2.

**AMV(IPA,JPA)**(Destination = 16 Command No = 42X)

**Move antenna to the given target antenna azimuth and antenna elevation co-ordinate.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(2)=IPA;TARG(3)=JPA,

AXIS=0,SRVCRD=10 for SRVCRD please refer **Appendix**.

**AMVAZIM(IPA)**(Destination = 16 Command No = 42X)

**Move antenna to the given target antenna azimuth and co-ordinate.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(2)=IPA;

AXIS=1,SRVCRD=10 for SRVCRD please refer **Appendix**.

**GOINNER**(Destination = 16)

**This command requests antenna to move on inner track.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

SRVCRD=0 for SRVCRD please refer **Appendix**.

**GOOUTER**(Destination = 16)

**This command requests antenna to move on outer track.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

SRVCRD=1 for SRVCRD please refer **Appendix**.

**TRACK(JPA,KPA,IPA)**(Destination = 16 Command No = 44X)

**Track antenna to the given target azimuth and elevation co-ordinate within given destination time.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(1)=IPA;TARG(2)=JPA;TARG(3)=KPA,

AXIS=0;

**TRKAZIM(JPA,IPA)**(Destination = 16 Command No = 44X)

**Track antenna to the given target azimuth within given destination time.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(1)=IPA;TARG(2)=JPA

AXIS=1;

**TRKELEV(JPA,IPA)**(Destination = 16 Command No = 44X)

**Track antenna to the given target elevation within given destination time.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(1)=IPA;TARG(3)=JPA

AXIS=2;

**ATRACK(JPA,KPA,IPA)**(Destination = 16 Command No = 44X)

**Track antenna to the given target antenna azimuth and antenna elevation co-ordinate within given destination time.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(1)=IPA;TARG(2)=JPA;TARG(3)=KPA;

AXIS=0;SRVCRD=10 for SRVCRD please refer **Appendix**.

**ATRKAZIM(JPA,IPA)**(Destination = 16 Command No = 44X)

**Track antenna to the given target antenna azimuth within given destination time.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

TARG(1)=IPA;TARG(2)=JPA;

AXIS=1;SRVCRD=10; For SRVCRD please refer **Appendix**.

**HOLD**(Destination = 16 Command No = 46X)

**This command requests servo to releases brakes for both axes and holds an antenna.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=0.

**HLDAZIM**(Destination = 16 Command No = 46X)

**This command requests servo to releases brakes for azimuth axis and holds an antenna.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=1.

**HLDELEV**(Destination = 16 Command No = 46X)

**This command requests servo to releases brakes for elevation axis and holds an antenna.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=2.

**BRAKES**(Destination = 16 Command No = 48X)

**This command applies brakes for both axes.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=0.

**BRKAZIM**(Destination = 16 Command No = 48X)

**This command applies brakes for azimuth axis.**

This procedure refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=1.

**BRKELEV**(Destination = 16 Command No = 48X)

**This command applies brakes for elevation axis.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=2.

**CLOSE**(Destination = 16 Command No = 4AX)

**Close down observation.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

**STOW**(Destination = 16 Command No = 4CX)

**Stow the antenna.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=0;

**SWELEV**(Destination = 16 Command No = 4CX)

**Stow the antenna in elevation.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=2;

**SWRELE**(Destination = 16 Command No = 4EX)

**Release antenna from the stowed position.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=0;

**SWRELEL**(Destination = 16 Command No = 4EX)

**Release antenna elevation stow.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

AXIS=2;

**STOP**(Destination = 16 Command No = 50X)

**Abort servo's previously given command especially this command is used to stop moving both axes.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

**ABRTSRVCMD**(Destination = 16 Command No = 50X)

**Abort servo's previously given command.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

**RDSRVSPC**(Destination = 16 Command No = 3AX,36X)

**Reads servo set parameters and shows in the ondisplay window by giving srvs command.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

**RSTSERVO**(Destination = 16 Command No = 6CX)

**Reset servo computer.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

**STPCHK(KPA)**(Destination = 16)

**This procedure checks STP flag appearance and disappearance.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

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

**For SRVCRD please refer Appendix.**

STSRVTIM(IPA)(Destination = 16 Command No = 52X)

**Set servo time by IPA=Argument sec ahead of the servo computer.**

This procedures refers **SERCMD** subroutine in **CMDFRM.FOR** file.

## Procedures for MCM (Destination = IPA(Variable))

The destination of the MCM commands are variable.

**NULLCMD(IPA)**(Destination = IPA Command No = 0)

**It generates a null command.**

This procedures refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**STIDLTIM(IPA)**(Destination = IPA Command No = 100)

**It generates a command to set idle time for MCM IPA.**

This procedures refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**STSCAN(IPA)**(Destination = IPA Command No = 101)

**It generates a command to set scan mode for MCM IPA.**

This procedures refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**STMEAN(IPA)**(Destination = IPA Command No = 102)

**It generates a command to set mean mode for MCM IPA. g**

This procedures refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**STTHRMD(IPA)**(Destination = IPA Command No = 103)

**It generates a command to set threshold mode for MCM IPA.**

This procedures refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**STANA(IPA)**(Destination = IPA Command No = 110)

**It generates a command to set analog mode for MCM IPA.**

This procedures refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**ST16DIG(IPA)**(Destination = IPA Command No = 120)

**It generates a command to set 16 digit digital mask.**

This procedures refers **MCMCMD** subroutine in **CMDFRM.FOR** file. First it calls **GETANTMSK** to know for how many antennas we need to pass this 16 digit digital mask. Which displays the same antenna mask using **DSPANTMSK**.

**ST32DIG(IPA)**(Destination = IPA Command No = 130)

**It generates a command to set 32 digit digital mask i.e. 2 word digital**

**mask.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file. First it calls **GETANTMSK** to know for how many antennas we need to pass this 16 digit digital mask. Which displays the same antenna mask using **DSPANTMSK**.

**ST64DIG(IPA)**(Destination = IPA Command No = 140)

**It generates a command to set 64 digit digital mask i.e. 4 word digital mask.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file. First it calls **GETANTMSK** to know for how many antennas we need to pass this 16 digit digital mask. Which displays the same antenna mask using **DSPANTMSK**.

**STMCMTHT(IPA)**(Destination = IPA Command No = 150)

**It generates a command to set threshold values for MCM IPA.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**RDANA(IPA)**(Destination = IPA Command No = 200)

**It reads analog values set for MCM IPA.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**RD16DIG(IPA)**(Destination = IPA Command No = 210)

**It reads 16 digit digital mask set for MCM IPA.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**RD32DIG(IPA)**(Destination = IPA Command No = 220)

**It reads 32 digit digital mask set for MCM IPA.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**RDMCMVER(IPA)**(Destination = IPA Command No = 230)

**It reads MCM program version set for MCM IPA.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**RD64DIG(IPA)**(Destination = IPA Command No = 240)

**It reads 64 digit digital mask set for MCM IPA.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.



**RDMODE(IPA)**(Destination = IPA Command No = 250)

**It reads current MCM mode.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**RDTHRVAL(IPA)**(Destination = IPA Command No = 260)

**It reads threshold value set for MCM IPA.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**FEEDSEL(IPA)**(Destination = IPA Command No = 300)

**It sets feed select mode for old MCM.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**RBMCMD(IPA)**(Destination = IPA Command No = 500)

**It generates a command to reboot MCM.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

**FEEDSELM(IPA)**(Destination = IPA Command No = 210)

**It sets feed select mode for modified MCM.**

This procedure refers **MCMCMD** subroutine in **CMDFRM.FOR** file.

## Appendix

### Information on ABC related commands

**DESTIN=17 PROC=ABCPROC SUBROUTINE=ABCCMD**

Adverb value is T3VERB

The actual command issued depends on the value of COMMAND

Acceptable values of COMMAND and their meaning are

Procedures	CMD NO.	Explanation
<b>STMCM</b>	0	<b>Configure MCMs at the ABC MCM</b> to be connected to the ABC slots 0-15 are taken from MPARM(I),I=1,16.IF the MCM address is positive wireback=TRUE. IF the MCM address is negative wireback=FALSE.
<b>STABCTIME</b>	1	Set ABC time to SUN time.
<b>STABCDLY</b> <b>STABCCYC</b>	2	<b>Set ABC timing parameters, values set in MPARM</b> <b>MPARM(1)=</b> Cycle time of ABC.(Unit 1sec) <b>MPARM(2)=</b> ABC to MCM Transmission Timeout value (5msec) <b>MPARM(3)=</b> Receive timeout value from MCM to ABC (5msec) <b>MPARM(4)=</b> Delay between 2 consecutive MCMs (5msec) <b>MPARM(5)=</b> I/p timeout value for ABC-SERVO communication. (IN between 2 servo messages.)(50 msec) <b>MPARM(6)=</b> Interbyte timeout value for ABC-SERVO(5msec) communication. <b>MPARM(7)=</b> Waiting for ack delay in ABC-SERVO communication.(50msec) <b>MPARM(8)=</b> O/p timeout val in ABC-SERVO communication (50 msec) <b>MPARM(9)=</b> Inter Command delay in ABC-SERVO communication (50 msec) <b>MPARM(10)=</b> Periodicity of Servo command(Read Analog Vars)(Unit 1sec) <b>MPARM(11)=</b> Periodicity of ABC debug packets (Unit 1sec) <b>MPARM(12)=</b> Duration between 2 LO MONITOR commands(Unit 1sec) <b>Default values of these are</b> MPARM 1,10,10,10,36,2,10,36,2,1,10,10 MAX consistent para vals are (TESTED) MPARM 2,10,10,10,100,5,30,100,5,10,10,10
<b>ENABCQ (IPA)</b> <b>ENAMCMQ</b> <b>DISMCMQ</b>	3	<b>Start a queue of commands to a specific device (address in CPARM(1))</b> to be issued by the ABC at the specified times. After this command, the

	<p>user can modify TIME to specify the time at which the following command to the specified device should be issued. The TIME info and the commands are accumulated in a large physical packet which is sent when the user issues a second COMMAND=3 to the ABC but with DPARM(10)=17.</p> <p>CPARM(1) = Address of device to which ABC should issue the commands.</p> <p>!!!!!!!!!!!!!! Currently ABC program supports this mode only for SERVO ( addr=16)!!!!!!!!!!</p> <p>Note : When the command is first issued, DPARM(10) is loaded with the address of the specified device (CPARM(1)) and the subsequently issued command destinations are checked against this and rejected if the two don't match.</p>
<b>STABCDBG (IPA)</b>	<p><b>Set ABC to DEBUG mode with debug level defined by CPARM(1)</b></p> <p>CPARM(1)=0 No debug</p> <p>=1 Standard debug mode, in which standard ABC status packet is sent every second.</p> <p>=2 Enhanced ABC-SERVO link layer mode, in which all the link layer events are sent to Online.</p> <p>=10 Talk mode Enabled.(Debug data dumping on PC disabled)</p> <p>=11 Talk mode Disabled.</p> <p>=12 ABC debug data dump on the PC connected to MCM(addr 15) enabled. Currently Raw data traffic between ABC-Servo is monitored.Talk mode disabled.</p> <p>=13 ABC debug data dump on PC disabled.</p> <p>=14 Non Event based execution of ABC-Servo link layer Enabled.</p> <p>=15 Event Based execution of ABC-Servo link layer enabled.(Debug modes 14 &amp; 15 available only with ABC Special Debug Proms).</p> <p>=16 Local Debug data at ABC is sent to unix, once this flag is set. This debug data</p>

		<p>gets displayed in ABCDBG window.A</p> <p>=17 Local debug data transmission is disabled.</p> <p>=18 Abort Mcm Qed commands in bunch,in case of any command failure.</p> <p>=19 Dont Abort Mcm Qed commands in bunch,in case of any command failure.</p> <p>Default at power on, Talk mode is enabled for ABC &amp; it sends standard debug packet every second,MCM qed commands are aborted case of errors.</p>
<b>SABRTABCQ( IPA, J PA)</b>	5	<p><b>Abort local queued commands</b></p> <p>CPARM(1) : Destination Buffer ID(SERVO)</p> <p>CPARM(2) : Kill Level Byte.</p> <p>If kill level is FFh -&gt; Entire Servo Local Q will be flushed. For those many local commands will be popped out of the Q. (i.e for 0- the no. specified is greater than the no. of existing cmds in the Q will be flushed)</p>
<b>QABCCMD (?)</b> (This command is not found in the NMOCPROC file)	6	<p><b>Set ABC Que time check value</b></p> <p>CPARM(1): DIFF_TIME-&gt; time in minutes, by which if cmd time is ahead of abc time, cmd will be rejected.</p> <p>CPARM(2): TIME_AHEAD -&gt; time in secs, by which abc time can be ahead of cmd time &amp; still issue the command</p> <p>CPARM(3) CHECK_TIME -&gt; time in minutes ,for which if cmds are blocked, then ABC generates error event saying servo local Q blocked</p>
<b>Talk( IPA,OUTFIL )</b>	7	<b>Send a TALK message (in OUTFIL ) to ABC via MCM</b> (MESSAGE should be less than 40 characters)
<b>RDABCVER</b>	10	<b>Get ABC program version</b>
<b>RDABCDBG</b>	11	<b>Get ABC error statistics</b>
<b>RDABCDLY</b>	12	<b>Get ABC timing information</b>
Command not found	13	<b>get ABC Local debug Data</b>
<b>ENASRVLNK</b>	14	<p><b>Enable/disable ABC-SERVO communication</b></p> <p>CPARM(1) = 1 =&gt; enable communication</p> <p>CPARM(1) = 0 =&gt; disable communication</p>
<b>ABRTPRKANT</b>	15	<b>Abort Parking Of Antenna operation</b>
<b>MCHABCCTR</b>	16	<p><b>Match Command-Response counters at ABC.</b></p> <p>ANTE[1]= 1 =&gt; Preset resp counter to Command monitor's cmd cnter value</p> <p>= 0 =&gt; reset resp cnters &amp; also those corresponding in Cmd Moni</p>

		<p>ANTE[2]= dest no (0=&gt; abc, 1 for servo, 2 for mcm, 3 for</p> <p>ANTE[3]=</p> <p>This part is INCOMPLETE!!!!</p>
<b>GOABCAPPL</b> <b>GOPROMAPPL</b> <b>STRTABCDNL</b>	17	<p><b>Ask ABC kernel to goto application and execute.</b></p> <p><b>Ask kernel to transfer from PROM and execute.</b></p> <p><b>Start downloading of application code.</b></p>
<b>RDQTIME</b>	19	<b>Get Servo Q timing information.</b>
<b>ENALO1MON</b> <b>ENALO2MON</b> <b>DISLO1MON</b> <b>DISLO2MON</b> <b>ENAIFMON</b> <b>DISIFMON</b>	20	<p><b>Enable/disable autpmtic mcm channel monitoring</b></p> <p>CPARM(1)=0 =&gt; disable auto monitoring</p> <p>CPARM(1)=1 =&gt; enable auto monitoring</p> <p>CPARM(2)=x =&gt; x is mcm address for which monitor to be done .</p> <p>By default select MCM NO 2 for LO &amp; MCM NO 9 for IF Monitoring.</p> <p>CPARM(3)=0 =&gt; LO Monitoring</p> <p>              =1 =&gt; IF MOnitoring.</p> <p>For this command abcdly parameters should be set through MPARM=1 10 10 10 36 2 10 36 2 10 10 10 (default)</p>
<b>Command not found</b>	22	<p><b>Modify MCM mask .</b></p> <p>ANTE[1]= MCM No , for which mask has to be modified.</p> <p>ANTE[2]= Action , = 0 =&gt; Set mask bits</p> <p>                      = 1 =&gt; clear mask bits.</p> <p>ANTE[3]= Mask ID, = 0 =&gt; modify dig16</p> <p>                      = 1 =&gt; modify dig32</p> <p>                      = 2 =&gt; modify dig64</p> <p>                      = 3 =&gt; modify analog mask</p> <p>ANTE[4]= Range = 0 =&gt; No range but following specify total no. of c hannels to be modified &amp; their channel nos.</p> <p>                      = 1 =&gt; Specifying range, so next 2 nos. will specify range of channels to be modified.</p> <p>ANTE[5] onwards : Either range of channels or total no of channels &amp; t he nos.</p>
<b>INITABCMD</b> <b>ENAABCCMD</b>	23	<p><b>Init ABC cmds:</b> Reset flag is reset also tally cmd-resp counters. This cmd is for single ABC and not for subarray.(define single ant in a subarray and issue the cmd).</p>
<b>GOABCKER</b>	31	<b>Request ABC for the software restart.</b>
<b>RSTABERR</b>	32	<b>Reset ABC error statistics counters</b>
<b>RDANTPARA</b>	33	<b>Read antenna parameters for local track.</b>
<b>RDSRCPARA</b>	34	<b>Read source parameters for local track</b>
<b>RDTIMTRK</b>	35	<b>Read time parameters for local track.</b>

## Information on MCM related commands

### **DESTIN=IPA PROC=MCMPROC SUBROUTINE=MCMCMD**

Commands to a MCM are issued when DESTIN is between 0 and 15, equal to the ID of the required MCM. The actual command issued depends on the value of COMMAND . Acceptable values of COMMAND and their meaning for MCMs are

Procedures	Cmd no.	Explanation
<b>NULLCMD (IPA)</b>	0	Null command
<b>STIDLTIM (IPA)</b>	100	Set mode idle.
<b>STSCAN (IPA)</b>	101	Set mode scan.
<b>STMEAN (IPA)</b>	102	Set mode mean.(Averaging factor in cpa(1))
<b>STTHRMD (IPA)</b>	103	Set mode threshold.
<b>STANA (IPA)</b>	110	Set analog mask.(Mask in ANAMASK(I),I=1,4))
<b>ST16DIG (IPA)</b>	120	Set 16 bit digital mask. (Mask in ANAMASK(1))
<b>ST32DIG (IPA)</b>	130	Set 32 bit digital mask. (Mask in ANAMASK(I),I=1,2)
<b>ST64DIG (IPA)</b>	140	Set 64 bit digital mask. (Mask in ANAMASK(I)=1,4)
<b>STMCMTHT (IPA)</b>	150	Set threshold values.(defined in cpa starting with a count)
<b>RDANA (IPA)</b>	200	Read analog mask.
<b>RD16DIG (IPA)</b>	210	Read 16 bit digital mask.
<b>RD32DIG (IPA)</b>	220	Read 32 bit digital mask.
<b>RD64DIG (IPA)</b>	240	Read 64 bit digital mask.
<b>RDMCMVER (IPA)</b>	230	Read version version of MCM program.
<b>RDMODE (IPA)</b>	250	read current mode of MCM program
<b>RDTHRVAL (IPA)</b>	260	read threshold values
<b>FEEDSEL (IPA)</b>	300	feed select
<b>FEEDSELM (IPA)</b>	600	feed select modified
<b>RBMCM (IPA)</b>	500	reboot mcm
<b>Couldn't find</b>	700	FE-Box monitor
<b>Couldn't find</b>	710	Common Box monitor

## Information on Servo related commands

### **DESTIN=16 PROC=SERVOPROC SUBROUTINE=SERVOCMD**

Commands to a SERVO are issued when DESTIN is equal to 17 . The actual command issued depends on the value of COMMAND . Acceptable values of COMMAND and their meaning for SERVOS are

To issue any of the below commands, set DESTIN=16, COMMAND=code of command and execute the verb T3VERB. For commands requiring axes, angles, dates or time, the following adverbs should be set.

AXIS = 0 => Both axes  
1 => Azimuth axis  
2 => Elevation axis

TARGET(1)= time  
TARGET(2)= an1 , the required azimuth angle  
TARGET(3)= an2 , the required elevation angle

ODATE(1)=date  
ODATE(2)=month  
ODATE(3)=year (2 digit)

WINDVEL = windvel

Procedures	Cmd no.	Explanation
<b>COLDSTART</b>	40x	<b>Stow release and hold axes</b>
<b>MV(IPA,JPA), MVAZIM(IPA), MVAELE(IPA), AMV(IPA,JPA), AMVAZIM(IPA)</b>	42x	<b>Moves antenna to axes to specified position(s)</b>
<b>TRACK(JPA,KPA,IPA), TRKAZIM(JPA,IPA), TRKELEV(JPA,IPA), ATRACK(JPA,KPA,IPA), ATRKAZIM(JPA,IPA),</b>	44x	<b>Track axes to reach specified position(s) at specified time.</b>

<b>HOLD, HLDAZIM, HLDELEV</b>	46x	<b>Hold one or both axes at current position.</b>
<b>BRAKES, BRKAZIM, BRKELEV</b>	48x	<b>Stop drive off, brakes on one or both axes.</b>
<b>CLOSE</b>	4ax	<b>Close down the observation of park both axes.</b>
<b>STOW, SWELEV</b>	4cx	<b>Stow specified axis.</b>
<b>SWRELE, SWRELEL</b>	4ex	<b>Stow release specified axis.</b>
<b>STOP, ABRTSRVCMD</b>	50x	<b>Abort previously given command.</b>
<b>RSTSERVO</b>	6cx	<b>H/W reset servo computer.</b>
<b>RDSRVSPC</b>	3ax, 36x	<b>Reads angles, Analog var, Digitalvar, Set param, Status axis, version.</b>
<b>STSRVTIM(IPA)</b>	52x	<b>Set time of the day.</b>
<b>Command not found</b>	54x	<b>Set stow angle.</b>
<b>Command not found</b>	56x	<b>Set s/w Hi-limit software upper limit to angles.</b>
<b>Command not found</b>	58x	<b>Set s/w Low-limit software lower limit to angles.</b>
<b>Command not found</b>	5ax	<b>Set windvel limit</b>



## Information on SUN related commands

**DESTIN=50 PROC=SUNPROC SUBROUTINE=SUNCMD**

Commands to a SUN are issued when DESTIN is equal to 50 . The actual command issued depends on the value of COMMAND . Acceptable values of COMMAND and their meaning for SUNs are

Procedures	CMD NO.	Explanation
<b>Command Not found</b>	10	Startup IPC based display and communication programs
<b>Command Not found</b>	11	Shutdown IPC based display and comm programs
<b>Command Not found</b>	15	Restart IPC base display pgm if it has crashed
<b>DEFSUB (IPA)</b>	20	Make antenna mask to associate a subarray defined by SUBARRAY with a group of antennas (ANTENNA(I)=1)
<b>SHSUB (IPA)</b>	21	Display antennas associated with subarray=SUBARRAY
<b>STRTPROC (STRA3)</b>	30	Start SUBARRAY CONTROLLER
<b>ABRTPROC (STRA3)</b>	31	This command aborts started process. Like subaray,scctask etc.
<b>HLTPROC (STRA3)</b>	32	Shutdown SUBARRAY CONTROLLER
	34	Transfer control to user for current subac (host from 'stra3')
	35	Remove control to user for current subac
<b>USERCON</b>	36	Connect user for command handler to access subac (given by user only)
<b>ONDBGUNIC</b>	40	Set DEBUG on in unixcomh - detailed logging on fort.12
<b>OFFDBGUNI</b>	42	Set DEBUG off in unixcomh
<b>LOGPKT (STRA3)</b>	50	This command starts packet logging in to the output file given in the argument.
<b>HLTPKTLOG (IPA)</b>	52	This command stops packet logging in to the output file given in the argument.
<b>SHLOGLIST</b>	54	This command shows list of log files opened for logging packets for the files which are argumented in the IPA.

## Information on FPS related commands

**DESTIN=21 PROC=NEWFPSPROC SUBROUTINE=NEWFPSCMD**

Commands to a FPS are issued when DESTIN is equal to 14 . The actual command issued depends on the value of COMMAND . Acceptable values of COMMAND and their meaning for CMHs are

Procedures	CMD NO.	Explanation
<b>NULLFPS</b>	0	Issues a null command to the FPS.
<b>STTNGPNT (IPA)</b>	100	Sets turning point using CP(1).
<b>STRMPDCNT</b>	110	Sets ramp down count.
<b>STLRPMLMT</b>	120	Sets lower rpm limit.
<b>STBCTDIF (IPA)</b>	130	Sets brake count difference count using CP(1). CP(1) contains the reqd. no. of pulses.
<b>STRUPCNT</b>	140	Sets ramp up time count.
<b>STSTPTCT</b>	150	Sets stop-time count.
<b>STMAXPWM</b>	160	Issues set maximum pwm count.
<b>STMAXANG (IPA)</b>	170	Sets the maximum angle of feed rotation. CP(1)=Max. angle, CP(2)=Pcal.
<b>STMINANG (IPA)</b>	180	Sets the minimum angle of feed rotation. CP(1)=Min. angle, CP(2)=Pcal.
<b>RDNGPNG</b>	200	Reads turning point.
<b>RDRMPDCT</b>	210	Reads ramp down count.
<b>RDLRPMLMT</b>	220	Reads lower rpm limit.
<b>RDBRCTDIF</b>	230	Reads brake count difference.
<b>RDRMPUPCT</b>	240	Reads ramp up time count.
<b>RDSTPTMCT</b>	250	Reads stop time count.
<b>RDMAXPWM</b>	260	Reads maximum pwm counts.
<b>RDMAXANG</b>	270	Reads maximum angle.
<b>RDMINANG</b>	280	Reads minimum angle.
<b>RDVERSION</b>	290	Read version.
<b>RUNCLBRT</b>	300	Calibrates the feed. (Run to calibrate)
<b>FREE280</b>	310	If CPA(1)=1; then free run towards -10 deg lim sw. If CPA(1)=0; then free run towards 280 deg lim sw.
<b>RUNDPREST (IPA), RUNCPPREST (IPA)</b>	320	Issues run to preset command. CPA(1)=target angle, CPA(2)=Pcal.
<b>FINECTUNE (IPA,J</b>	330	It issues a command to run the feed to preset

<b>PA), FINEDTUNE(IPA,J PA)</b>		value, IPA is the target angle and JPA is PWM counts.
<b>RUNPASSWD</b>	340	Run passworded command. If feed has hit one of the limit switches(-10/280) ,we have to run this command so as to take it out from that position. After this, we have to calibrate the feed using 'runclbrt' command.
<b>UAOCLBRT(IPA)</b>	360	It issues a command to calibrate to UA0 angle if CP=1 then it will calibrate in clockwise direction i.e. -10 deg side and if CP=0 then it will calibrate in anti-clockwise direction.
<b>FPSBOOT</b>	500	Reboots FPS.
<b>FPSSTOP</b>	600	Stop FPS.
<b>RDUAOANG, GETUA0ANG</b>	700	Read UA0 angle and get the first set UA0 angle.

## Information on Sub-array Control related commands

### **DESTIN=70 PROC=SACPROC SUBROUTINE=SUBACMD**

The SUBARRAY controller which is spawned by GMRT ( COMM =30 DEST=50) is controlled by commands issued with DESTIN=70. Currently the only implemented function is to send track commands every minute ( 6 commands each valid for 10 sec). When the SUBARRAY controller is spawned, it knows only the IPC ids of GMRT, unixcomh and dispdecode. The SUBARRAY controller has to be fed the coordinates of the source to be tracked and the antennas in the SUBARRAY before the track command is issued.

Procedures	CMD NO.	Explanation
<b>SNDSACCMD</b>	2	Send local pops (i.e. user shell) commands through a sub-array-controller from a given command file.
<b>GOSACOUT</b>	6	Tell SUBARRAY controller to go on outer azimuth track
<b>GOSACIN</b>	8	Tell SUBARRAY controller to go on inner azimuth track
<b>SNDSACSRC, TRKELOFF, TRKAZOFF, TRKANTOFF, TRKRAOFF, TRKDECOFF, SCANELSRC, SCANAZSRC, SCANRASRC, SCANDEC,</b>	10	Send source coordinates to SUBARRAY controller (coords stored in SCOORD and filled by GETSOU())
<b>Not Found in help</b>	12	Send ANTENNA mask for the SUBARRAY to the controller
<b>Not Found in help</b>	14	Send SHOW request to SUBARRAY controller
<b>Not Found in help</b>	16	Set DEBUG on for SUBARRAY controller
<b>Not Found in help</b>	18	Set DEBUG off for SUBARRAY controller
<b>Not Found in help</b>	20	Start TRACKING
<b>Not Found in help</b>	30	Stop TRACKING
<b>Not Found in help</b>	40	Set SUBAC to BINARY mode

<b>help</b>		
<b>Not Found in help</b>	50	<b>Set SUBAC to ASCII mode</b>
<b>Not Found in help</b>	54	<b>Set tolerance from cpa</b>
<b>SUBHNDLE</b>	55	<b>Sub-array handle to catter tracking generic command.</b>
<b>OPSACFIL</b>	60	<b>Opens sub-array-controller file in the user shell.</b>
<b>CLSACFILE</b>	62	<b>Close sub-array-controller file in the user shell.</b>
<b>STRTSACFIL</b>	64	<b>Gives remote control of the sub-array-controller to the opened command file in the user shell.</b>
<b>STPSACFIL</b>	66	<b>Stops remote control given to the sub-array-controller to the opened command file in the user shell.</b>
<b>SHSACFILE</b>	70	<b>Shows a sub-array-controller which is open in the user shell.</b>
<b>SHSACLIN</b>	71	<b>Shows a current line running for the running sub-array-controller file in the user shell.</b>
<b>RWSACFILE</b>	72	<b>Rewind sub-array-controller file in the user shell.</b>
<b>MVSACCON(IPA)</b>	73	<b>Move control to the point i.e. (IPA) in SAC file.</b>
<b>SKPSACLIN(IPA)</b> <b>MVSACCON(IPA)</b>	74	<b>Skip no of lines i.e. (IPA) from SAC file. Step by one line i.e. (IPA) from SAC file.</b>

## Information on Command-monitor related commands

**DESTIN=33 PROC=CMPROC SUBROUTINE=MONCMD**

Commands to a FPS are issued when DESTIN is equal to 14 . The actual command issued depends on the value of COMMAND . Acceptable values of COMMAND and their meaning for CMHs are

Procedures	CMD NO.	Explanation
<b>STMCMOFF (IPA)</b>	0	Set IPA seconds as a command monitoring offset i.e. it generates a set offset command for command monitoring.
<b>ENACMDMON, DISCMDMON</b>	1	Enables or disable command monitoring.
<b>ENACMINFO, DISCMINFO</b>	2	Enables or disable command monitor information log.
<b>ABORTDNLD</b>	3	Abort ABC download operation for a sub-array.
<b>STDNLDPKT (IPA)</b>	4	Set ABC program download packet size=IPA (CPA(1)=IPA is as argument).

## **SCOORD**

SCOORD 1-3	Source name (12 char)
4	RA at reference epoch
5	Dec at reference epoch
6	The reference epoch (1950 def)
7	RA at target epoch
8	Dec at target epoch
9	The target epoch (def(0) date)
10	RA offset
11	Time derivative of RA
12	Dec offset
13	Time derivative of Dec
14	Reference time 11 and 12
15	Azimuth offset
16	Time derivative of azimuth
17	Elevation offset
18	Time derivative of elevation
19	Reference time for 16 and 17
20	Time (def =0 => current time)

All angles and times are in radians

The currently implemented commands to T5VERB are

COMM=1 Calculates the rise and set times for the source and date defined in SCOORD. Uses the RA and Dec at the target epoch for the calculation.

COMM=10 Precesses the coordinates of the source from the reference epoch to the target epoch. If target epoch (scoord(8)) is 0.0, the computer date id the target date.

COMM=11 Precesses the coordinates from the target epoch to the reference epoch

COMM=20 Calculates the azimuth and elevation  
for the coordinates and time in SCOORD.  
If Time=0.0, it calculates for  
the computer time. If Time not=0  
the calculation is done for the  
specified time which is reset  
to 0 after the calculation.

COMM=30 Get the coordinates of the source  
specified in the AIPS string  
variable SOURCE and fill the  
SCOORD array appropriately. The  
Coordinates are got from the  
file "source.list" in ~/bin.

### **SRVCRD**

Check if command needs one or more ANGLES

If the command is equal to track or move, depending on the value  
of SRVCRD, conversion between astronomical and antenna coordinates are  
made.

If

SRVCRD = 0, astronomical coordinates are mapped to antenna  
coordinates, with el < 90d and antenna azimuth on  
the inner track

SRVCRD = 1, same as 0, but with antenna azimuth on the  
outer track

SRVCRD = 2, same as 0, but with antenna elevation in the 90d  
to 105d region when possible

SRVCRD = 3, same as 1 and 2 combined

SRVCRD =10, given coordinates are assumed to be antenna  
coordinates and no conversion is done except safety  
checks to ensure that the antenna azimuth and  
elevation are within the allowed ranges