

MSI / NIS / OPNAV Sequencer Guide

Contents

SEQGEN FRAGMENTS/CASs/REQUESTS	3
1. OVERVIEW AND TERMINOLOGY	3
2. FRAGMENTS INTO CASs - INFO AND RULES	3
3. FRAGMENT LIST	4
3. MSI CAS LIST	4
4. NIS CAS LIST	5
5. OPNAV CAS LIST	6
CAS DESCRIPTIONS	6
1. GUIDANCE AND CONTROL CASs (for MSI and NIS)	7
1) <i>Inertial Hold – DS40</i>	7
CAS_MSI_POINT_FULL and CAS_NIS_POINT_FULL	7
CAS_MSI_POINT and CAS_NIS_POINT	9
2) <i>Scanning - DS56</i>	9
CAS_MSI_SCAN and CAS_NIS_SCAN	10
2. INSTRUMENT CASs	11
1) <i>MSI Turn-On CAS</i>	11
CAS_MSI_ON	11
2) <i>MSI Auto Exposure Setup CASs</i>	12
CAS_MSI_LOAD_FILTER_TABLE	12
CAS_MSI_AUTO_EXPOSE	12
3) <i>MSI Sequence Definition CAS</i>	14
CAS_MSI_LOAD_SEQ_DEF	14
4) <i>MSI Sequence Execution CASs</i>	17
CAS_MSI_SHOOT	17
CAS_MSI_SINGLE_REPEAT	18
CAS_MSI_DOUBLE_REPEAT	18
CAS_MSI_DOUBLE_STAGGERED_REPEAT	19
CAS_MSI_TRIPLE_REPEAT	20
5) <i>Miscellaneous MSI CASs</i>	21
CAS_MSI_RELEASE_ATTITUDE	21
CAS_MSI_PARK_MODE	22
6) <i>MSI Combination CASs</i>	22
CAS_MSI_POINT_SCAN_SHOOT	22
7) <i>NIS Turn-On CASs</i>	23
CAS_NIS_MAG_DPU_ON	23
8) <i>NIS Sequence Definition</i>	24
CAS_NIS_LOAD_TRANS_SEQ_DEF	24
9) <i>NIS Calibration CASs</i>	25
CAS_NIS_CRUISE_CAL	26
CAS_NIS_CAL	27
CAS_NIS_ZERO_SUN	27
CAS_NIS_OFF_SUN	28
CAS_NIS_BUFFER_FLUSH	28
10) <i>NIS Sequence Execution CASs</i>	28
CAS_NIS_EXECUTE	28
CAS_NIS_SINGLE_REPEAT	29
CAS_NIS_DOUBLE_REPEAT	30
CAS_NIS_SUPERSCAN_1 (previously called LOW_PHASE_SUPER_SCAN)	31
CAS_NIS_SUPER_SCAN2 (previously called SUPER_SCAN)	31
11) <i>Miscellaneous NIS CASs</i>	32
CAS_NIS_RELEASE_ATTITUDE	32
CAS_NIS_PARK_MODE	32
12) <i>NIS Combination CASs</i>	32
CAS_NIS_POINT_SCAN_EXECUTE	32
3. OPNAV CASs	33
CAS_OPNAV_A	33
CAS_OPNAV_B, and CAS_OPNAV_BP	34

<i>CAS_OPNAV_C (also CAS_OPNAV_CT, CAS_OPNAV_CE)</i>	35
<i>CAS_OPNAV_C_A_CRUISE</i>	37
<i>CAS_OPNAV_D</i>	38
<i>CAS_OPNAV_E</i>	38
<i>CAS_OPNAV_E_2</i>	39
<i>CAS_OPNAV_E_3</i>	39
<i>CAS_OPNAV_E_4</i>	40
<i>CAS_OPNAV_F</i>	41
<i>CAS_OPNAV_G</i>	42
<i>CAS_OPNAV_H</i>	43
<i>CAS_OPNAV_I</i>	43
<i>CAS_OPNAV_J</i>	44
<i>CAS_OPNAV_K</i>	45
Specialized Observations	47
OPNAV Star Calibrations.....	47
NLR Alignment	48
MSI Star Calibrations	48
NIS Low Phase Flyby	48
CAS_NIS_FULL_POINT_SCAN_EXECUTE	49
Global Maps	49
MPP Maps	51
CAS_NIS_FULL_POINT_EXECUTE.....	52
Latitude scans	52
CAS_NIS_SUPER_SCAN.....	52
Seqgen	53
Getting Started	54
Running Seqgen at Cornell	54
Running Seqgen at APL.....	57
Coordinating Observations	57
Common Mistakes to Double Check	57
Delivering Products	57

SEQGEN FRAGMENTS/CASs/REQUESTS

1. OVERVIEW AND TERMINOLOGY

Spacecraft Commands are the most basic building blocks of a command sequence, but we never use them directly. Rules for their use is largely beyond the scope of this document and our knowledge. Instead, we command our instruments via SEQGEN 'macros' which observe spacecraft commanding protocol and free the user from the responsibility of understanding those details. There are several levels of SEQGEN macros which are used to construct our sequences.

A fragment is a SEQGEN macro which strings together a number of spacecraft commands by a fixed set of rules and timing. They contain all of the commands necessary to load sequences and timetags into DPU memory or the Flight Computer, etc. Most of these activities are invisible to us but are required to perform our sequences. Since many of the things we do require stringing together the same sets of spacecraft commands, separated by the same timing offsets, fragments save time, and prevent error. Most fragments contain free parameters whose values may be selected at the time of use. However, some are entirely canned, and allow no parameter entry.

Fragments are combined by other sets of rules to form CASs (Canned Activity Sequences). A CAS represents a higher level of MACRO in SEQGEN which strings together a selection of fragments. In all CASs except the Opnav CASs, the complete parameter list for the CAS is simply the combination of the fragment parameter lists. The order in which these parameters must be entered is fixed for each CAS, and actually abides by a set of rules described later. Opnav CASs are special in that the parameter list for each CAS is not the sum of the fragment lists.

A CAS is the basic building block of a SEQGEN REQUEST, which, in turn, is the basic building block of the SEQGEN .sarf file. There may be multiple SEQGEN REQUESTs in an .sarf file. Each REQUEST is made up of one or more "Activities" separated from each other by a timing offset. An Activity must contain one and only one CAS (fragments may not be called in an Activity). In an Activity, the CAS name is listed, followed by the parameter list. A single REQUEST may not contain both MSI and NIS CAS's (it's one or the other).

The CAS Start Time, or Scheduling Point is both an actual time, as well as a conceptual time reference within each CAS where the timing of the various fragments it contains is determined by offsets from the Scheduling Pt. of the CAS. Sometimes these offsets are free parameters, and sometimes they are fixed (for instance, the SCAN fragment always initiates at the Scheduling Point). The Scheduling Point itself is assigned a time in the REQUEST (this is the time on the first line of the REQUEST).

2. FRAGMENTS INTO CASs - INFO AND RULES

Each instrument has a set of fragments from which we may choose to construct that instrument's CASs.. To understand how the MSI and NIS CASs have been constructed, it is useful to divide these fragments into 4 groups: 1) fragments which command the instruments, 2) fragments which command a DS40 (inertial hold), 3) the fragment which commands a DS56 (scan pattern), and 4) miscellaneous 'stand-alone' fragments.

Group 4 is a special case; these fragments are never combined with other fragments. Each is used in a CAS by itself. Generally these are instrument turn on, configuration and miscellaneous setup CASs.

Most of the MSI and NIS CASs which actually command data acquisition are a combination of fragments from Groups 1, 2 and 3. We have chosen the most useful combinations to create the MSI and NIS CASs, and self-imposed the following general rules mainly to facilitate implementation of this system in the Orbit software. A CAS may contain a fragment from one group only, from two groups, or from all three groups, but no more than one from each group. *The Opnav CASs are an exception to this and to all of the rules below. They will be explained separately.* Both MSI and NIS CASs may use the guidance and control CASs from Groups 2 and/or 3. MSI CASs may only contain MSI fragments from Group 1 and may not contain NIS fragments from Group 1. Similarly, NIS CASs may only contain NIS fragments from Group 1 and may not contain MSI fragments. Note that Groups 2 and 3 contain Guidance and Control fragments that are used identically by both instruments. In general, when a fragment is used in a CAS, it always uses the same exact set of parameters defined for that fragment. The order in which the parameters appear in the CAS parameter list is determined as follows. If a group 1 fragment is present its parameters are always

listed first; if a group 2 fragment is present its parameters are listed next; if a group 3 fragment is present its parameters are listed last. There are no additional parameters when these fragments are combined.

The Opanv CASs are special. They are not constructed by a simple combination of fragments. The section on Opanv CASs below explains in detail how each CAS works and what parameters are available.

3. FRAGMENT LIST

Below is a list all of the fragments available for use by MSI and NIS. Note that the shortcut letters in parentheses (following the fragment name) is used to systematically define the shortcut names for the CASs we have created (see the CAS lists).

MSI Group 1 Fragments: Instrument control

SHOOT - (S)
SINGLE REPEAT - (SR)
DOUBLE_REPEAT - (DR)
TRIPLE_REPEAT - (TR)
DOUBLE_STAGGERED_REPEAT - (DSR)

NIS Group 1 Fragments: Instrument control

EXECUTE - (E)
SINGLE REPEAT - (SR)
DOUBLE REPEAT - (DR)
SUPER SCAN 1 - (SU1)
SUPER SCAN 2 - (SU2)

G& C Group 2 Fragments: Inertial Hold (may be used by MSI or NIS)

POINT - (P)
POINT FULL - (PF)

G&C Group 3 Fragments: Scanning about an Inertial Hold (may be used by MSI or NIS)

SCAN - (S)

MSI Group 4 Fragments: Stand-alone fragments, never combined

SEQ DEF - (SEQ)
AUTO EXPOSE - (AUTO)
LOAD FILTER TABLE - (FILT)
CONFIG - (CONF)
RELEASE_ATTITUDE - (REL)
PARK_MODE - (PARK)

NIS Group 4 Fragments: Stand-alones – never combined

BUFFER_FLUSH - (BUF)
CONFIGURE - (CONF)
LOAD_TRANS_SEQ_DEF - (SEQ)
RELEASE_ATTITUDE - (REL)
PARK_MODE - (PARK)

3. MSI CAS LIST

The following is a complete list of MSI CASs available.

- 1. CAS_MSI_ON
- 2. CAS_MSI_OFF
- MCONF 3. CAS_MSI_CONFIG
- 4. CAS_MSI_DUMP_LOGS
- MSEQ 5. CAS_MSI_SEQ_DEF

- 6. CAS_MSI_CANCEL_IMAGE
- MAUTO 7. CAS_MSI_AUTO_EXPOSE
- MFILT 8. CAS_MSI_LOAD_FILTER_TABLE
- MSH 9. CAS_MSI_SHOOT
- MSR 10. CAS_MSI_SINGLE_REPEAT
- MDR 11. CAS_MSI_DOUBLE_REPEAT
- MTR 12. CAS_MSI_TRIPLE_REPEAT
- MDSR 13. CAS_MSI_DOUBLE_STAGGERED_REPEAT
- MPARK 14. CAS_MSI_PARK_MODE
- MREL 15. CAS_MSI_RELEASE_ATTITUDE
- 16. CAS_MSI_NADIR_POINT
- 17. CAS_MSI_NADIR_SINGLE_REPEAT
- 18. CAS_MSI_NADIR_DOUBLE_REPEAT
- 19. CAS_MSI_NADIR_DOUBLE_STAGGERED_REPEAT
- 20. CAS_MSI_NADIR_SCAN_SINGLE_REPEAT
- 21. CAS_MSI_NADIR_SCAN_DOUBLE_REPEAT
- 22. CAS_MSI_NADIR_SCAN_DOUBLE_STAGGERED_REPEAT
- MP 23. CAS_MSI_POINT
- MFP 24. CAS_MSI_POINT_FULL
- MPSH 25. CAS_MSI_POINT_SHOOT
- MPSR 26. CAS_MSI_POINT_SINGLE_REPEAT
- MPDR 27. CAS_MSI_POINT_DOUBLE_REPEAT
- MPTR 28. CAS_MSI_POINT_TRIPLE_REPEAT
- MPDSR 29. CAS_MSI_POINT_DOUBLE_STAGGERED_REPEAT
- MSSR 30. CAS_MSI_SCAN_SINGLE_REPEAT
- MSDR 31. CAS_MSI_SCAN_DOUBLE_REPEAT
- MSDSR 32. CAS_MSI_SCAN_DOUBLE_STAGGERED_REPEAT
- MPSSH 33. CAS_MSI_POINT_SCAN_SHOOT
- MPSSR 34. CAS_MSI_POINT_SCAN_SINGLE_REPEAT
- MPSDR 35. CAS_MSI_POINT_SCAN_DOUBLE_REPEAT
- MPSTR 36. CAS_MSI_POINT_SCAN_TRIPLE_REPEAT
- MPSDSR 37. CAS_MSI_POINT_SCAN_DOUBLE_STAGGERED_REPEAT

4. NIS CAS LIST

The following is list of all NIS CASs available:

- NBUF 1) CAS_NIS_BUFFER_FLUSH
- 2) CAS_NIS_CAL
- 3) CAS_NIS_CRUISE_CAL
- 4) CAS_NIS_ZERO_SUN
- 5) CAS_NIS_OFF_SUN
- 6) CAS_NIS_MAG_DPU_ON
- 7) CAS_NIS_ON
- 8) CAS_NIS_OFF
- 9) CAS_NIS_MAG_DPU_OFF
- NCONF 10) CAS_NIS_CONFIGURE
- NSEQ 11) CAS_NIS_LOAD_TRANS_SEQ_DEF
- NREL 12) CAS_NIS_RELEASE_ATTITUDE
- NPARK 13) CAS_NIS_PARK_MODE
- 14) CAS_NIS_NADIR_POINT
- NE 15) CAS_NIS_EXECUTE

NSR 16) CAS_NIS_SINGLE_REPEAT
 NDR 17) CAS_NIS_DOUBLE_REPEAT
 NPE 18) CAS_NIS_POINT_EXECUTE
 NPSE 19) CAS_NIS_POINT_SCAN_EXECUTE
 NPSR 20) CAS_NIS_POINT_SINGLE_REPEAT
 NPDR 21) CAS_NIS_POINT_DOUBLE_REPEAT
 22) CAS_NIS_NADIR_EXECUTE
 23) CAS_NIS_NADIR_SINGLE_REPEAT
 24) CAS_NIS_NADIR_DOUBLE_REPEAT
 NSSR 25) CAS_NIS_SCAN_SINGLE_REPEAT
 NSDR 26) CAS_NIS_SCAN_DOUBLE_REPEAT
 27) CAS_NIS_NADIR_SCAN_SINGLE_REPEAT
 28) CAS_NIS_NADIR_SCAN_DOUBLE_REPEAT
 NPSSR 29) CAS_NIS_POINT_SCAN_SINGLE_REPEAT
 NPSDR 30) CAS_NIS_POINT_SCAN_DOUBLE_REPEAT
 NSU2 31) CAS_NIS_SUPER_SCAN will be called CAS_NIS_SUPER_SCAN_2
 NSU 32) CAS_NIS_LOW_PHASE_SUPER_SCAN will be called CAS_NIS_SUPER_SCAN
 NFPE 33) CAS_NIS_FULL_POINT_EXECUTE
 NFPSE 34) CAS_NIS_FULL_POINT_SCAN_EXECUTE
 NFPSSU 35) CAS_NIS_FULL_POINT_SCAN_SUPER
 NFPSU 36) CAS_NIS_FULL_POINT_SUPER
 NFPSSR 37) CAS_NIS_FULL_POINT_SCAN_SINGLE_REPEAT

5. OPNAV CAS LIST

1) CAS_OPNAV_A
 2) CAS_OPNAV_BP
 3) CAS_OPNAV_B
 4) CAS_OPNAV_C
 5) CAS_OPNAV_CE
 6) CAS_OPNAV_CT
 7) CAS_OPNAV_D
 8) CAS_OPNAV_E
 9) CAS_OPNAV_E_1
 10) CAS_OPNAV_E_2
 11) CAS_OPNAV_E_3
 12) CAS_OPNAV_E_4
 13) CAS_OPNAV_F
 14) CAS_OPNAV_G
 15) CAS_OPNAV_H
 16) CAS_OPNAV_I
 17) CAS_OPNAV_J
 18) CAS_OPNAV_K
 19) CAS_OPNAV_L
 20) CAS_OPNAV_M
 21) CAS_OPNAV_N

CAS DESCRIPTIONS

As described above, the NIS and MSI CASs have been built by combining fragments that come from the instrument fragment groups and/or the guidance and control groups. The sections that follow describe the actual CASs that Karl has constructed for us using these fragments. The first section just discusses the CASs which only contain G&C fragments. These are available for use by both MSI and NIS. The second section explains all CASs which contain MSI or NIS instrument fragments only, as well as CASs which combine instrument fragments with G&C fragments. The final section explains OPNAV CASs, which are not simple combinations of the fragments listed in the previous section.

With only a few exceptions, the following CAS descriptions only give definitions for free parameters, and information about how to use them in SEQGEN. Detailed descriptions of the internal workings of each CAS are available in the near.satf file and the CAS Dictionary.

1. GUIDANCE AND CONTROL CASs (for MSI and NIS)

MSI and NIS are allowed to utilize two basic G&C spacecraft command data structures. They are the DS40, which commands the spacecraft to some inertial hold position, and the DS56, which will cause the spacecraft to modulate the pointing about that inertial hold position. Below is a description of these spacecraft capabilities followed by descriptions of the CASs we created that allow MSI and NIS to utilize those capabilities.

1) *Inertial Hold – DS40*

The DS40 command data structure points a specified spacecraft body vector, the *virtual boresight* (v.b.), toward a specified target (*aimpoint*), while attempting to maintain a specified *roll position* about the virtual boresight. Once achieved, the spacecraft will continue to attempt to keep the v.b. pointed at the aimpoint indefinitely, or until another DS40 data structure is loaded. Note that this is almost never a static position. The aimpoint is almost always moving relative to the spacecraft (except when using a J2000 aimpoint, or a very distant aimpoint such as Eros on distant approach).

When a DS40 data structure is loaded and **enabled**, the spacecraft slews as quickly as possible to the specified orientation from the previous spacecraft orientation. To perform this 'slew', it accelerates at the maximum acceleration available (about .0025 deg/sec²) for half the angle, and then decelerates at the same rate for half the angle. The direction of this slew is the shortest path available from the previous spacecraft orientation to the commanded orientation.

Here is how the DS40 works. Four basic entities are defined in the DS40 data structure: 1) Virtual Boresight – this is a unit vector in spacecraft body fixed coordinates that represents the part of the spacecraft which is to be pointed toward the aimpoint; 2) Aimpoint – this is the POINT in space you want the virtual boresight to point toward; 3) Roll Vector – this is another spacecraft body unit vector, 4) Roll Reference – a spacecraft-centered J2000 unit vector. The roll vector and the roll reference are used together to determine the 'roll orientation' (or twist) ABOUT the virtual boresight. The roll position assumed at each instant is the one which minimizes the angle between the roll vector and the roll reference. Note that the first priority is to keep the virtual boresight pointed at the aimpoint. Therefore, the only degree of freedom available for satisfying the roll algorithm is rolling ABOUT the virtual boresight. It is therefore not always possible to make the angle between the roll vector and roll reference go to zero.

An example will help. Let's say we have defined the the virtual boresight to be x' (camera boresight), aimpoint to be Eros, the roll vector to be +z (solar panel normal), the roll reference to be the sun. First, point the x' to Eros nadir. No matter what happens, the x' must always stay pointed at nadir. This means the only degree of freedom left is twisting the spacecraft about the x' (v.b.). Now imagine a spacecraft-centered J2000 vector pointing out from the spacecraft toward the sun. Then, visualize the roll vector (the z axis) as a little handle sticking out of the spacecraft (normal to the x'). Grab the little z axis handle and spin the spacecraft about the x' until the angle between the z axis (r.v.) and the s/c centered J2000 vector that points to the sun is minimized. Voila! This is the orientation commanded by that set of DS40 parameters.

Additional note: When the spacecraft can't keep up with an aimpoint (that is moving too fast in angle-space), it will simply do the best it can. Since the spacecraft has on-board representations of the target, planetary and spacecraft trajectories, it knows where it SHOULD be looking at each instant in time. It continuously attempts to be pointing where it knows it should be pointing.

There are two fragments available to each instrument which can command a DS40, the POINT_FULL and the POINT. Each can be used together with instrument fragments and/or the SCAN fragment, or alone in a CAS as shown below.

CAS_MSI_POINT_FULL and CAS_NIS_POINT_FULL

The POINT_FULL CAS has all the parameters of the DS40 data structure. These CASs must be used with caution, and may only be used with special permission from the Mission Ops Manager. NIS must use the CAS_NIS_POINT_FULL, and MSI must use

the CAS_MSI_POINT_FULL, but the parameters are identical in both, and are only described once below. Again, once a DS40 has been loaded and enabled, the spacecraft will remain in that configuration indefinitely, or until another DS40 with different parameter values is loaded and enabled.

Parameters:

Slew_Duration: Time, in seconds (between 1 and 900) which the CAS backs up prior to the Scheduling Point to initiate the slew. The slew to take the spacecraft to the desired DS40 configuration will begin at the Scheduling Point minus the Slew-Duration. Therefore, Slew_Dur must be larger than the time it takes to complete the slew plus any settling time required. See appendix ? for a guide to making estimates for slew durations and settling time required.

DS40_coord_sys: Coordinate system for the Aimpoint. There are 5 to choose from. 0 = J2000, 1=ECI (Earth Centered Inertial), 2 = SCI (Sun Centered Inertial), 3 =ACI (Asteroid Centered), 4 = NS (Nadir-Sun), 5 = AF (Asteroid Body Fixed)

DS40_aimpt_x , DS40_aimpt_y , DS40_aimpt_z: X, y, and z components of the Aimpoint. Units for these aimpt components are a function of coordinate system selected (see below). To visualize how this works, plot the point specified by the coordinates in the specified coord. system. *The v.b. will point toward THAT point in space.*

0. J2000 – J2000 centered in the spacecraft. X, y, and z must be a unit vector.
1. Earth Centered Inertial – J2000 centered in the Earth. *Components (x, y, and z) are in KM*, measured from Earth center in that coordinate system.
2. Sun Centered Inertial – This is J2000 centered in the Sun. *Components are in KM*, measured from Sun center in that coordinate system.
3. Asteroid Centered Inertial – This is J2000 centered in the Asteroid. *Components are in KM*, measured from Asteroid center in that coordinate system. (NON-ROTATING)
4. Nadir/Azimuth – This is a polar spherical system centered in the spacecraft. Direction to asteroid nadir is the pole. X = elevation (deviation from pole in *degrees*), Y = Azimuth (measured clockwise when looking down nadir, also in *degrees*; zero azimuth is in the longitude of the sun). Z = N/A.
5. Asteroid Fixed – This is a cartesian coordinate system centered somewhere close to the center of mass of the asteroid. The spacecraft gets orientation and location of center of this coordinate system from the Navigation Asteroid Attitude file. X, y, and z components are in KM. This system ROTATES with the asteroid.

DS40_roll_vec_x ; DS40_roll_vec_y ; DS40_roll_vec_z: X, y, and z components of the Roll Vector. *Must be a unit vector.* This is a vector in the spacecraft coordinate system (where x' points down the instrument boresight, z points down the HGA, and Y = Z x X). Spacecraft attempts to minimize the angle between roll vector and roll reference by rolling about the Virtual Boresight.

DS40_aimpt_short_cut: Integer between 0 and 3. If 0, the aimpoint x, y, and z and coordinate system selected above will be used. If not 0, SEQGEN will ignore the aimpoint x, y, z values and use a short cut where 1=sun, 2=earth, 3=nadir. **Do you have to have proper coord selected when using these?**

DS40_roll_ref_x , DS40_roll_ref_y , DS40_roll_ref_z: x, y, and z components of the roll reference vector. *Must be a unit vector.* This vector is in spacecraft-centered J2000. It essentially describes an RA/Dec. The spacecraft tries to minimize the angle between the roll vector and the roll reference by rolling the s/c about the virtual boresight.

DS40_boresight_x , DS40_boresight_y , DS40_boresight_z: x, y, and z components of the virtual boresight vector. *Must be a unit vector.* This vector is in the spacecraft coordinate system and defines the Virtual Boresight, which defines the physical part of the spacecraft that will be pointed toward the aimpoint. Using values (1,0,0) means that the x' (or instrument boresight) will point at the aimpoint. . Virtual Boresights representing the exact center of the MSI, or the various NIS fields of view are available in the Appendix. To cause the center of a field of view to be offset from the aimpoint, a vector other than the instrument boresight must be input.

DS40_roll_algor: Integer between 0 and 3. If 0, use the inertial roll reference provided above. If not 0, then the roll reference is defined as follows: 1=sun, 2=earth, 3=nadir.

Notes:

1. We may only use the FULL_POINT CASs with approval of MOPs.
2. Even if you are using a roll algorithm, or an aimpoint shortcut, you must still have a unit vector in the coordinate parameters for roll reference or aimpoint, respectively.
3. **Significant figures? ...**

CAS_MSI_POINT and CAS_NIS_POINT

Same parameters as the POINT_FULL except that the parameters for roll vector, roll reference, and roll algorithm are not free in this CAS. The roll vector is usually set to be the Z axis, and the roll reference is usually set to be the Earth. This maximizes the chance that the fan beam antenna will be able to 'see' the Earth during normal data acquisition.(for doppler and tracking, etc). But take warning that this is not always the case. The Ops team can put in any roll algorithm they choose without MSI/NIS ever knowing. Orbit always sets roll reference to Earth for the POINT CASs. If Ops decides they need to use sun as roll reference for some reason, and we have used the POINT CAS, our planned observations will be different from what actually executes! They have agreed to tell us if they plan to use something other than Earth. Normally, this also means that if we wish to use sun as roll reference, we cannot use this CAS. We must use the POINT_FULL CAS.

It has been requested by Mission Ops that we use the POINT CAS under most circumstances. Once again, each instrument has it's own POINT CAS that it is required to use, but the parameters are identical.

Parameters:

Slew_Duration: (same as above)
DS40_coord_sys: (same as above)
DS40_aimpt_x: (same as above)
DS40_aimpt_y: (same as above)
DS40_aimpt_z: (same as above)
DS40_aimpt_short_cut: (same as above)
DS40_boresight_x: (same as above)
DS40_boresight_y: (same as above)
DS40_boresight_z: (same as above)

Notes:

1. Minimum slew dur we are allowed to use under normal circumstances is 90 seconds. Has to do with back up for imaging commands interfering with guidance commands. In the low phase flyby this rule was waived because there was no imaging.

Examples:

- 1) 510,4,0,0,0,1,0,0 point nadir, no boresight offset
- 2) 600,0,-0.063080,0.4880745,-0.8589599,0,0.9996294,-0.166889,0.0215065 point to canopus in J2000 coordinates but offset the frame up and to the left.

2) Scanning - DS56

We may also command the virtual boresight to make a scanning pattern about the inertial pointing position achieved with the DS40. This is accomplished through the DS56 spacecraft command data structure.

The best way to think about it is that the DS56, when enabled, causes the point that the virtual boresight is tracking to deviate from the nominal DS40 aimpoint position by some cyclical motion. There are 5 coordinate systems available to describe this motion. The scanning occurs in each dimension (or axis) of the coordinate system independently. You can almost think of it as the 'aimpoint' moving about it's nominal position. The v.b. will continue to track the new position indicated by the combination of those simultaneous movements in each axis to the best of it's physical capabilities. There is one coordinate system where this is not true. In SBF, it's actually the virtual boresight that is modulated.

The DS56 has the capability to define, for EACH dimension: 1) a scanning period, defined by a rate and a duration; and 2) a hold, or pause period, defined by a duration. During the pause, the rate for that dimension simply goes to zero. The scan and hold periods are always executed as a pair. The hold may occur before or after the scan, as defined in the hold_place parameter. The pair will be repeated for as many times as possible until the end of the scan duration. Upon each execution of hold and scan, the order that hold and scan occurs (as defined in hold_place) remains the same. But the direction of scanning (sign of the rate) may be reversed for each execution of the scan if the rev parameter is set to True. Once again, all of the above capabilities exist for each dimension independently.

In all non-Opnav MSI or NIS CASs, the scan always begins at the Scheduling Point. As is commonly available with other fragments, there is no offset parameter which allows the user to offset the start of the scan from the Scheduling Point. The scan is terminated at Scheduling Point plus Scan_duration. **When the scan is terminated, the v.b. immediately returns to DS40 position as fast as possible.** We may not load a DS56 without having a DS40 already loaded and active. It is NOT required that the DS40 and DS56 be loaded and executed in the same CAS, or even in the same REQUEST, however, it is wise to have both the POINT and SCAN in the same REQUEST to avoid timing mistakes.

CAS_MSI_SCAN and CAS_NIS_SCAN

There is only one fragment, the SCAN fragment, which can execute a DS56 data structure. The SCAN fragment may be combined with the POINT or POINT_FULL fragments, and/or combined with instrument fragments, or used alone as shown in the CAS_MSI_SCAN, and CAS_NIS_SCAN CASs. The parameters are identical for both CASs and are only listed once below.

Parameters:

Reuse_previous_scan: TRUE = ignore the following parameters and reuse parameters from most recently used DS56; FALSE = do not use previous scan values, use the parameters provided in this CAS. *Note*: you should only use TRUE when you are reusing something that was used in the same REQUEST - it's the only way to be certain which DS56 was the last used. *If TRUE is selected, we must input the parameters even though they are not used.*

Scan_duration: Amount of time the scan continues. Scan start occurs at the Scheduling Point. Scan Stop will be sent at Sched. Pt. + Scan_dur.

DS56_coord_sys: Must be compatible with DS40 coordinate system, *except* that SBF may be used with any DS40 coord. system. 0=J2000 Inertial Rotation, 1= Nadir-Sun Rotation, 2= ABF (Asteroid Body Fixed) Translation, 3= ACI (Asteroid Centered Inertial) Translation, 4= SBF (Spacecraft Body Fixed).

DS56_rev_x, DS56_rev_y, DS56_rev_z: 0 = do not reverse sign of rate, and 1 = reverse sign of rate for each subsequent execution of the scan in that dimension.

DS56_hold_place_x, DS56_hold_place_y, DS56_hold_place_z: 0=pause occurs before the scan in that dimension, and 1=pause occurs after the scan in that dimension.

DS56_rate_dur_x, DS56_rate_dur_y, DS56_rate_dur_z: The length of time to scan at specified rate in each dimension (for one scanning period).

DS56_rate_x, DS56_rate_y, DS56_rate_z: Scan rate to be used during scanning period in each dimension. Units are a function of coordinate system chosen:

- 0) J2000 Inertial - rotational system. X, y, and z point along J2000 axes. Rates cause the aimpoint to change by right-hand rule rotation about each J2000 axis. *Units are in deg/sec.*
- 1) Nadir-Sun - rotational system. X points at nadir, y points normal to x and in the 'longitude of the sun', and z is normal to both (out the back of the spacecraft generally). The coordinate system's orientation in space is continually adjusted to satisfy the above constraints. Rates cause the aimpoint to change by rotation about each axis in the N-S system. *Units are in deg/sec.*
- 2) Asteroid Body Fixed - translational system. X, y, and z are embedded in asteroid, rotate with the asteroid, and are defined by the Navigation Asteroid Attitude file. Rates cause the aimpoint to move as specified in the ABF system. *Units are in km/sec.* Example: if rate values are x=10, y=0, z=0, the aimpoint will move at 10km/s along the x axis of the asteroid.
- 3) Asteroid Centered Inertial - translational system. X, y, and z point along J2000 axes, centered in the asteroid and therefore the system *Does not rotate with the asteroid.* Rates cause the v.b. to scan relative to aimpoint in directions parallel to the each axis in the ACI system.
- 4) Spacecraft Body Fixed - rotational system using v. boresight offset. X, y, and z refer to the spacecraft body coordinate system. This one works differently from the other coordinate systems. Rates cause the virtual boresight definition to CHANGE, that is, to be offset from the x' by right-hand-rule rotation about each axis. . But the v.b. stays pointed at the nominal aimpoint, resulting in a motion of the FOV (or x') relative to the aimpoint in the opposite direction. Example: +y rate (y points 'up' in FOV, put thumb up and curl fingers, they curl left) causes v.b. to move to the 'left' of x' (as looking out the boresight). But since the v.b. always remains pointed at the aimpoint, this causes the FOV (or x') to scan to the 'right' of the aimpoint. A +z rate causes v.b. to move 'up' relative to x', but since it stays on aimpoint, this causes FOV to scan 'down' relative to aimpoint. CAN ONLY BE USED FOR SMALL ANGLE OFFSETS (up to ?). *Units are in deg/sec.*

Summary: +y rate causes FOV to scan RIGHT, -y rate causes FOV to scan LEFT
+z rate causes FOV to scan DOWN, -z rate causes FOV to scan UP

DS40_pause_dur_x, DS40_pause_dur_y, DS40_pause_dur_z: Length of time to pause scanning either before or after the scan period occurs (before or after depends on value in hold_place).

Notes:

1. The time between end of a scan and start of any subsequent slew must be AT LEAST 5 SECONDS.
2. If only one scan period is used, it is safest to put 1's in for the reverse parameter, and 100,000's in for the hold durations. The latter is especially important because if the scan disable does not get issued, the spacecraft will simply hold at the end of the scan for 100,000 seconds.
3. ABF is defined by the Navigation Asteroid Attitude file, and uploaded to the spacecraft in a different data structure. This will be updated periodically by the Ops Team as the shape model evolves.
3. S/c J2000 and ACI will not change.

Example of typical DS56 command:

1. FALSE,1680,4,0,1,1,0,1,1,0,112,112,0,0.0000097,0.0001647,0,0,0 - Perform a diagonal scan across two pixels for 112 seconds and repeat 15 times without stopping at either end

2. INSTRUMENT CASs

The following are descriptions of all CASs available to operate the MSI and NIS instruments. The different sections cover all aspects of instrument operation including instrument turn-on, configuration, sequence definition, and sequence execution, as well as other miscellaneous CASs required for SEQGEN.

Instrument turn-on and setup is very different for each of the instruments. However, one thing they have in common is that sequences (when to take data and how often) must be defined for an activity to occur. For MSI this is done with the MSI_SEQ_DEF CAS. For the NIS it's the NIS_LOAD_TRANS_SEQ_DEF command. Once a sequence is defined, it will remain in memory until it is redefined or the instrument is turned off. When a sequence definition resides in DPU memory, it is then possible to execute that sequence. Numerous fragments exist to execute MSI and NIS sequences, and whereas NIS and MSI share most of the same terminology (SINGLE_REPEAT, DOUBLE_REPEAT), sometimes they are defined slightly differently for each instrument. Each also has some instrument-specific CASs, not available at all to the other instrument. For instance, the MSI has a CAS which will execute up to 12 sequences (MSI_SHOOT) and the NIS has a command to perform a single NIS sequence once (NIS_EXECUTE).

All of the instrument execution fragments can be combined with the "point" (DS40) and "scan" (DS56) fragments in almost every possible combination.

Other miscellaneous CASs exist for various purposes. Two commonly-used CASs available to MSI and NIS are used to indicate that an MSI, NIS, or Opnav observation has concluded, allowing an other instrument to drive the spacecraft pointing: PARK_MODE and RELEASE_ATTITUDE. The PARK_MODE CAS commands the spacecraft to a "safe" configuration. The RELEASE_ATTITUDE CAS just indicates to SEQGEN that another instrument or Ops may take control of pointing the spacecraft. No commands are sent to the spacecraft in this case. It's used during heavy use times like approach and orbit.

1) MSI Turn-On CAS

CAS_MSI_ON

The MSI and NIS instruments should be turned on during an attended track (9 am - 5pm ET where EST=UTC-5 and EDT=UTC-4). **Warm up, wait time?** To turn on the MSI, one must simply issue the MSI_ON CAS.

Parameters:

None

Example:

```
request(MSI_ON,
      START_TIME, 1999-083T16:30:00.0,
      REQUESTOR, "meb",
      PROCESSOR, CTP1,
      KEY, "MSI")

  activity(1,
    SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
    MSI(CAS_MSI_ON)
  ),
end;
```

2) MSI Auto Exposure Setup CASs

Two very important **CASs** are required to be issued before MSI takes AUTO_EXPOSURE images: LOAD_FILTER_TABLE and AUTO_EXPOSE. These are not necessary if you are taking manual exposures. These and all sequence definitions are held in memory until they are reset or the instrument is turned off.

CAS_MSI_LOAD_FILTER_TABLE

Loads a specified MSI filter transmission table into memory. Parameters describe the relative sensitivity of filters 0 - 7, and are used in the autoexposure algorithm to calculate exposures for the images taken through the filters other than the one used for the test image. The values given here are the best values known to date and are currently loaded in spacecraft memory. Better values may be calculated after we get to Eros. If so, we will upload a new set of values using this CAS. Otherwise, this CAS will not be sent again unless the instrument is turned off.

(How much time before or after?)

Parameters:

Sens_F0: Relative transmission for filter 0 [15308]
Sens_F1: Relative transmission for filter 1 [2000]
Sens_F2: Relative transmission for filter 2 [498]
Sens_F3: Relative transmission for filter 3 [1587]
Sens_F4: Relative transmission for filter 4 [581]
Sens_F5: Relative transmission for filter 5 [1001]
Sens_F6: Relative transmission for filter 6 [279]
Sens_F7: Relative transmission for filter 7 [101]

Example:

```
activity(1,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  MSI(CAS_MSI_LOAD_FILTER_TABLE,16678,2000,470,1729,617,1091,300,1
09)
),
```

CAS_MSI_AUTO_EXPOSE

This defines the setup for automatic exposure control. A different setup is used for different classes of images (opnavs, monochrome, color, etc.) There are 7 parameters. Since the auto exposure parameters required are scene-dependent, as well as 'objective' dependent, it is often necessary to issue the AUTO_EXPOSE CAS prior to each observation. Our policy is that it

should be issued as the first Activity in any REQUEST where auto exposures are required. This makes the REQUEST more robust.

A test image, not downlinked, is taken some second prior to execution of a sequence. Its exposure time and filter are commanded. The image is histogrammed, and the exposure time is adjusted to reach a target DN level. The relative sensitivity of the different filters (viewing a solar-illuminated Eros) is commanded separately based on all calibration results and Eros observations to date; thus a text exposure is needed in only one filter for a color sequence

Parameters:

T_IM_DELTA: The time in seconds prior to acquisition of an image sequence at which a test exposure is taken; (0-3600), default 10 sec

FILT FOR T_IM: Filter in which the test exposure is taken; (0-7), default filter 3. If a monochrome sequence follows, use that filter (with current sequence definitions, that's 0 or 4). If a color sequence follows, use the filter with the best PSF (4).

T_IM_EXP_TIME: Test image exposure time in milliseconds; (0-999), default 200ms. These have been calculated from Beth Clark's best Eros photometry, shooting for 1/2 of the max expected DN near the terminator (1/6 of the brightest point on the asteroid).

ALLOWABLE SAT: Allowable saturation in 1/10's of percent; (0-1000), default 10 (1%). This is the number of pixels allowed to exceed the target DN in the final image. In distant opnavs, this is 0 to avoid saturating Eros. In disk-resolved data, this is 5 to allow for cosmic ray hits.

TARGET SAT DN: Target saturation DN level for science image (0-4095); default 4000. For color images (taken with slow or no slewing) this is 2500 for max signal, allowing headroom for user mixtakes in extrapolating exposures between filters. For opnavs, 3000. For monochrome images, set so that there is <1 pixel of smear when slewing at the maximum rate used for mosaicking (0.03 degrees/sec), where the brightness is 1/3 of max on the asteroid (this is close to the terminator). For filter 4 at solar distances early in the mission, this is 1800. This number will be adjusted periodically as solar distance changes.

OVR_EXP_FLBK_TIM: Overexposure fallback time in 1/10's of percent of test image exposure time; (0-1000), default 750 (75%). If the test exposure is saturated (1 or more pixels 4095), then the actual exposure used is this many thousandths of the test exposure. (For example, if the test exposure is 100 ms, using a value of 750 results in a final exposure of 75 ms.)

NOISE_OFFSET: Dark current in DN's assumed for image exposure calculations. Use 87.

NOTE:

- 1. For all above parameters, value -1 means unchanged.
- 2. Must wait 20 seconds before executing a sequence which uses autoexposure.

Example:

```
activity(2,  
    SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,  
    MSI(CAS_MSI_AUTO_EXPOSE,10,1,20,0,2500,750,87)  
)
```

Notes:

1. How autoexposure works: At T_Im_Delta prior to first image in a Sequence that requires autoexposure, the filter wheel is set to the test image filter, and a test image is taken using T_Im_Exp_Time as exposure level. This should be set to try to underexpose the image. The test image is read into an image buffer and analyzed using the Allowable_Sat parameter to determine if the image was under or overexposed. If the test image was underexposed, the correct exposure to achieve Target_Sat_DN is calculated by scaling the test exposure up. If the test image was overexposed, the proper exposure time is calculated by scaling the test exposure down using Ovr_Exp_Flbk_Tim (usually 75%). Either method results in an estimated proper exposure for the filter used in the test image. The exposure time for the other filters are calculated using the filter transmission parameters.

2. Generally we set T_Im_Delta to be 10 sec. It only takes 2 seconds to set up the filter wheel (occurs at -10sec before first image in real Sequence). Then the test image is taken and correct exposures determined. It only takes 5 seconds to do this. **...?**
SCOTT?

3. **IMPORTANT:** Only one test image is only taken prior to the first image taken by the execution of a Sequence Definition. The exposures of all other frames taken in that Seq.Def. will be based on that test image. If we use a single Seq Def to cover multiple positions in a mosaic, and the scene content of the first position is very different from the others (like, for instance, if the first image contains mostly black sky, and the others contain more of the asteroid), then the exposures for the other positions in the mosaic will not be correct.

TABLE OF AUTOEXPOSURE SETUP VALUES:

Below is the table of AUTO_EXPOSE parameters for the encounter. Scott is the AUTO_EXPOSE guru and he should always double-check the values that you use. (The table needs to be updated Scott.)

Table 1: AUTO_EXPOSE values to use NOW

Time Period	Sequence	Seqs	t_i m_d elta	t_im_filt	t_im_exp	allow_sat	Target_sat _dn	ovr_exp_ flbk_tim	noise_o ffset
1/00-4/00	Approach opnav B	26,28, 8	10	0	2	0	3000	500	87
"	Spectral/ Clean	25,24, 22,20	5	4	28	0	2500	750	87
"	Filter 4 - Monochr, Opnav C-K	27, 18	5	4	28	0	1800	750	87
	Filter 6- Monochr	16	10	6	50	5	900	750	87

3) MSI Sequence Definition CAS

An MSI Sequence Definition defines the format by which up to 8 images may be taken. It defines the type of image compression to be used, the time delta between images, a unique filter value for each image, and whether the exposures should be taken using autoexposure or manual exposures. In the latter case, the manual exposure values are also supplied. The MSI SEQ_DEF CAS does not execute this sequence, it just loads the parameters for a single Sequence Definition into the MSI DPU. The DPU can store up to 30 unique sequence definitions at one time. We may overwrite, or redefine one or more of these sequence definitions at any time, however, to avoid confusion, we have adopted the policy that we will load all 30 sequences at the start of the week, and make every attempt to use only those 30 sequences during each week. It may be possible to avoid changing these 30 sequences for months at a time. (see tables below).

As mentioned above, image compression is specified in the sequence definition. The MSI observations are downlink hogs, and it is always necessary to fiddle with the number of images taken or compression type to fit the data into the downlink. See Appendix A for description of what compression to use when.

The sequence definition fragment is never combined with other fragments. It is always used alone in its own CAS. It is called CAS_MSI_LOAD_SEQ_DEF. Only ONE sequence is defined in each call of this CAS, and it must be a separate activity within the REQUEST. In addition to loading all 30 Seq Defs at the start of the each week-long sequence load, it is useful to put the required SEQ Def CASs at the beginning of each REQUEST. This is because if the camera is turned off by accident, or intent, each REQUEST will work if the sequence is restarted in the middle (where the initial load of all 30 sequences would not occur).

CAS_MSI_LOAD_SEQ_DEF

All values are 16 bit? 65535?

Parameters:

seq_num: Unique identifier (RANGE, \1...30\)

num_images: Number of images to take during single execution (RANGE,\1...8\)

interval: Time between image exposures (RANGE,\1...255\)

comp_table: Tables for Lossy compression (12-8 bit conversion):

"NONE": Do not use Lossy compression (12 bit data)

"TABLE1":

"TABLE2":

"TABLE3":

"TABLE4":

"TABLE5":

"TABLE6":

"TABLE7":

Tables 2-6 for 8-bit data with greater compression ratios resulting from subsequent application of lossless compression.

Table 7 converts output to all zeros from throwaway image at end of double nominal-zero exposure sequence (triple repeats).

flag_dpcm: (data pulse code modulation) Flag to use lossless decompression. "OFF" or "ON" . Always on if comp_alg=FAST, always off if comp_alg= NONE.

comp_alg: "NONE", or two types of Lossless compression:"FAST", "RICE" (we never use RICE).

pixels: Number of bits/pixel to take image? (default=16)

mode: "AUTO" for autoexposure mode using the last valid autoexposure setup, or "MAN" for manual exposure using the following exposure times.

im1_dur; im2_dur; im3_dur; im4_dur; im5_dur; im6_dur; im7_dur; im8_dur: Exposure times in msec (RANGE,\0...999\), first num_images values are used. The rest must be zero.

im1_filt; im2_filt; im3_filt; im4_filt; im5_filt; im6_filt; im7_filt; im8_filt: Filter numbers (RANGE,\0...7\), first num_images values are used. The rest must be zero.

Notes:

1. Must load a sequence definition at least 2 minutes prior to use.

Example:

In this example, definitions are assigned for Seq Defs 30, 29 and 28.

```
activity(3,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  MSI(CAS_MSI_SEQ_DEF,30,1,1,"NONE","OFF","NONE",16,"MAN",999,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
),
activity(4,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  MSI(CAS_MSI_SEQ_DEF,29,1,1,"NONE","ON","FAST",16,"MAN",999,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
),
activity(5,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  MSI(CAS_MSI_SEQ_DEF,28,5,1,"NONE","ON","FAST",16,"AUTO",0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
),
```

SEQUENCE DEFINITION TABLE:

Below is a table for approach and high orbit sequence definitions. Remember that seq_num 30, 29, 28, 27, 9 and 8 are reserved for OPNAV's (See Table 3), although 9 and 8 are used for OPNAV-B (a cruise OPNAV cas that will not be used).

Table 2a: Approach and High Orbit Sequence Definitions

	8-bit monochr	Satsrch resolved	8-filter MS	4-filter MS	Canopus filter 7	7-filter MS	7-filter clean	3-filter MS	3-filter clean	8-bit monochr
Seq_num	27	26	25	24	23	22	21	20	19	18
Num_images	1	1	8	4	1	7	7	3	3	1
Interval	1	1	2	2	1	2	2	2	2	1
Comp_table	TABLE 5	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	TABLE 5
Flag_dpcm	ON	ON	ON	ON	ON	OFF	ON	ON	ON	ON
Comp_alg	FAST	FAST	FAST	FAST	FAST	NONE	FAST	FAST	FAST	FAST
Pixels	16	16	16	16	16	16	16	16	16	16
Mode	AUTO	AUTO	AUTO	AUTO	MAN	AUTO	MAN	AUTO	MAN	AUTO
Im_dur1	0	0	0	0	999	0	0	0	0	0
Im_dur2	0	0	0	0	0	0	0	0	0	0
Im_dur3	0	0	0	0	0	0	0	0	0	0
Im_dur4	0	0	0	0	0	0	0	0	0	0
Im_dur5	0	0	0	0	0	0	0	0	0	0
Im_dur6	0	0	0	0	0	0	0	0	0	0
Im_dur7	0	0	0	0	0	0	0	0	0	0
Im_dur8	0	0	0	0	0	0	0	0	0	0
Im_filt1	4	0	0	1	7	1	1	1	1	4

Im_filt2	0	0	1	3	0	2	2	3	3	0
Im_filt3	0	0	2	4	0	3	3	4	4	0
Im_filt4	0	0	3	6	0	4	4	0	0	0
Im_filt5	0	0	4	0	0	5	5	0	0	0
Im_filt6	0	0	5	0	0	6	6	0	0	0
Im_filt7	0	0	6	0	0	7	7	0	0	0
Im_filt8	0	0	7	0	0	0	0	0	0	0

Table 2b: Approach and High Orbit Sequence Definitions

	NLR coalign	1000 µm map	4-filter clean	8-bit monchr clean	12-bit monchr clean	Canopus filter 6	Canopus filter 5	Throw away in Triple Repeat	Sat search - sub pix
Seq_num	17	16	15	14	13	12	11	10	7
Num_images	4	1	4	1	1	1	1	1	3
Interval	1	1	2	1	1	1	1	1	3
Comp_table	NONE	TABLE 5	NONE	TABLE 5	NONE	NONE	NONE	TABLE 7	NONE
Flag_dpcm	ON	ON	ON	ON	ON	ON	ON	ON	ON
Comp_alg	FAST	FAST	FAST	FAST	FAST	FAST	FAST	FAST	FAST
Pixels	16	16	16	16	16	16	16	16	16
Mode	MAN	AUTO	MAN	MAN	MAN	MAN	MAN	MAN	MAN
Im_dur1	999	0	0	0	0	999	999	0	999
Im_dur2	999	0	0	0	0	0	0	0	999
Im_dur3	999	0	0	0	0	0	0	0	999
Im_dur4	999	0	0	0	0	0	0	0	0
Im_dur5	0	0	0	0	0	0	0	0	0
Im_dur6	0	0	0	0	0	0	0	0	0
Im_dur7	0	0	0	0	0	0	0	0	0
Im_dur8	0	0	0	0	0	0	0	0	0
Im_filt1	7	6	1	4	4	6	5	1	0
Im_filt2	7	0	3	0	0	0	0	0	0
Im_filt3	7	0	4	0	0	0	0	0	0
Im_filt4	7	0	6	0	0	0	0	0	0
Im_filt5	0	0	0	0	0	0	0	0	0
Im_filt6	0	0	0	0	0	0	0	0	0
Im_filt7	0	0	0	0	0	0	0	0	0
Im_filt8	0	0	0	0	0	0	0	0	0

Canopus filter 4	Canopus filter 3	Canopus filter 2	Canopus filter 1	USE?	12 bit monochr
6	5	4	3	2	1
1	1	1	1	1	1
1	1	1	1	1	1
NONE	NONE	NONE	NONE	NONE	NONE
ON	ON	ON	ON	ON	ON
FAST	FAST	FAST	FAST	FAST	FAST
16	16	16	16	16	16
MAN	MAN	MAN	MAN	MAN	AUTO
999	999	999	999	250	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

0	0	0	0	0	0
0	0	0	0	0	0
4	3	2	1	0	4
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Create table for Low orbit?

COMPRESSION RATES:

In the following execution CASs it is necessary to leave enough time for compression to take place.

NEED A TABLE: Lossy Buffer Filled etc..... SCOTT, could you please write this up??

4) MSI Sequence Execution CASs

There are 5 different fragments available for executing MSI sequences: SHOOT, SINGLE_REPEAT, DOUBLE_REPEAT, DOUBLE_STAGGERED_REPEAT, and TRIPLE_REPEAT. No more than one of the above may be used in any single given CAS, however, each of these may be combined with the POINT (or POINT_FULL) fragments and/or the SCAN fragment, and/or used alone. Below we go through each of these MSI fragments when used alone in a CAS.

Note about IM_TYPE. This parameter occurs in all of these CASs. It is a parameter that is filled in by the sequence builder, which SEQGEN uses to estimate data volume. This parameter is not sent to the spacecraft. The actual compression type used when the images are taken is specified in the MSI Sequence Definition. IM_TYPE estimates how much actual compression we can expect given the compression type used, and the expected scene. Below is a table which can be used to determine what IM_TYPE value to select. Also, see Appendix A.

CAS_MSI_SHOOT

With this CAS we can execute up to 12 sequences spaced by intervals that can vary. For each execution we can specify the seq id of the sequence to be executed, a timing delta which indicates when to execute the sequence relative to the Scheduling Point of the CAS, and the compression algorithm for SEQGEN to use in calculating data volume.

Parameters:

num_executes: Number of executes, up to 12. Specifies how many sequences we will execute in this SHOOT CAS.

seq_num_1: Sequence number of 1st sequence to execute.

im_delta_1: Seconds from start of CAS to execute 1st sequence.

IM_TYPE_1: Estimated compression (RANGE,\"2.5\",\"2.4\",\"2.3\",\"2.2\",\"2.1\",\"2.0\",\"1.9\",\"1.8\",\"1.7\",\"1.6\",\"1.5\",\"1.4\",\"1.3\", \"1.2\",\"1.1\",\"1.0\")

seq_num_2: Sequence number.

im_delta_2: Seconds from start of CAS to execution of 2nd sequence.

IM_TYPE_2: Compression.

seq_num_3: Sequence number.

im_delta_3: Seconds from start of CAS to execute 3rd sequence.

IM_TYPE_3: Compression.

... etc

seq_num 12: Sequence number.

im_delta 12: Seconds from start of CAS to execute 1st sequence.

IM_TYPE 12: Compression.

CAS_MSI_SINGLE_REPEAT

This CAS will repeatedly execute a single sequence at constant intervals. We specify the sequence id to use, the total number of executions, time delta between executions, and compression estimate. Note that the images are taken at the beginning of the time interval. So if there is a 20 second interval, the first set is taken in the first second, the second execution begins in the 21st second, etc.

(I want to put in a little graph here that shows executions on a horizontal timeline, you remember those little drawings we made?)

Parameters:

START_IMAGING: Time in seconds relative to start of CAS that the first execution occurs.

SEQ_1: Sequence to be repeated (RANGE,\1...30\,).

DELTA_1: Time in seconds between executions of SEQ_1.

ITERATIONS: Number of times to execute sequence. (includes first execution!) MAY NOT BE "1"! Use a SHOOT if need it to be 1.

IMAGE_TYPE: Compression algorithm for SEQGEN data volume calculations.

(RANGE,\"2.5\",\"2.4\",\"2.3\",\"2.2\",\"2.1\",\"2.0\",\"1.9\",\"1.8\",\"1.7\",\"1.6\",\"1.5\",\"1.4\",\"1.3\",
\"1.2\",\"1.1\",\"1.0\").

Example:

In the following REQUEST the SINGLE_REPEAT is combined with a NADIR (mo, we will no longer have the NADIR CAS... I mean, ORBIT cannot do NADIR CASs) point to get a light curve. Note that there is no boresight offset to align with NIS. Sequence 26 is executed every 461 seconds, 46 times.

```
request(MSI_Lightcrv3,  
        START_TIME, 1998-349T02:22:00.0,  
        REQUESTOR, "MSI",  
        PROCESSOR, CTP1,  
        KEY, "MSI")  
  
    activity(1,  
        SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,  
        MSI(CAS_MSI_AUTO_EXPOSE,10,0,50,0,3000,750,87)  
    ),  
    activity(2,  
        SCHEDULED_TIME,\600\,FROM_PREVIOUS_START,  
        MSI(CAS_MSI_NADIR_SINGLE_REPEAT,0,26,461,46,"2.5",600)  
    ),  
    activity(4,  
        SCHEDULED_TIME,\20748\,FROM_PREVIOUS_START,  
        MSI(CAS_MSI_PARK_MODE)  
    ),  
end;
```

CAS_MSI_DOUBLE_REPEAT

The DOUBLE_REPEAT fragment repeatedly executes a pair of MSI sequences at a constant intervals. The time spacing between the pair is also constant.

Parameters:

START_IMAGING: Time in seconds relative to start of CAS that the first sequence is executed.

seq_1: First sequence number in pair (RANGE,\1...30\).

delta_1: Time in seconds between executions of SEQ_1 and SEQ_2.

seq_2: Second sequence number in pair (RANGE,\1...30\).

delta_2: Time in seconds between executions of SEQ_2 and SEQ_1.

Iterations: Number of times to execute sequence pair, including first execution.(May not be 1)

IMAGE_TYPE: Compression algorithm (RANGE,\"2.5\",\"2.4\",\"2.3\",\"2.2\",\"2.1\",\"2.0\",\"1.9\",\"1.8\",\"1.7\",\"1.6\",\"1.5\",\"1.4\",\"1.3\", \"1.2\",\"1.1\",\"1.0\").

Example:

In the GlobalMorph, Seq 25 is executed at the beginning of the CAS, then 527 seconds later Seq 6, then 1054 seconds later Seq 25 again etc. for a total of 13 iterations.

```
request(MSI_GlobalMorph2,
        START_TIME, 1999-012T19:00:00.0,
        REQUESTOR, "MSI",
        PROCESSOR, CTP1,
        KEY, "MSI")

        activity(1,
            SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
            MSI(CAS_MSI_AUTO_EXPOSE,10,0,2,0,3000,750,87)
        ),
        activity(2,
            SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
            MSI(CAS_MSI_SEQ_DEF,25,8,2,"NONE","ON","FAST",16,"AUTO",0,0,0,0,0,0,0,0,1,2,3,4,5,6,7)
        ),
        activity(3,
            SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
            MSI(CAS_MSI_SEQ_DEF,6,2,527,"NONE","ON","FAST",16,"AUTO",0,0,0,0,0,0,0,0,6,6,0,0,0,0,0)
        ),
        activity(4,
            SCHEDULED_TIME,\600\,FROM_PREVIOUS_START,
            MSI(CAS_MSI_POINT_DOUBLE_REPEAT,0,25,527,6,1054,13,"1.7",600,4,0,0,0,0,1,0,0)
        ),
        activity(5,
            SCHEDULED_TIME,\41106\,FROM_PREVIOUS_START,
            MSI(CAS_MSI_RELEASE_ATTITUDE)
        ),
end;
```

CAS_MSI_DOUBLE_STAGGERED_REPEAT

The DOUBLE_STAGGERED_REPEAT fragment repeatedly executes 2 different sequences, but unlike the DOUBLE_REPEAT, the sequences are not executed as a pair. Each sequence has its own time delta for offsetting the first execution from CAS start, as well as its own time delta between executions, and number of iterations. It is essentially like running two SINGLE_REPEATs at the same time (which is otherwise not legal). The user must be very careful with this CAS to avoid causing the sequence executions to fall on top of one another (occurring at the same time). *Note that there is only one parameter for Image Type*. This means that the Image Type specified will be used for all images taken in this CAS. Have to calculate an 'average' or effective im_type value if different im-types are required.

Parameters:

START_SEQ_1_IMAGING: Time in seconds relative to start of CAS to first execution of sequence 1.

START_SEQ_2_IMAGING: Time in seconds relative to start of CAS to first execution of sequence 2.

seq_1: First sequence number (RANGE,\1...30\).

delta_1: Time in seconds between executions of SEQ_1.

seq_1_iterations: Number of times to execute SEQ_1.

seq_2: Second sequence number (RANGE,\1...30\).

delta_2: Time in seconds between executions of SEQ_2.

seq_2_iterations: Number of times to execute SEQ_2.

IMAGE_TYPE: Compression algorithm (RANGE,\"2.5\",\"2.4\",\"2.3\",\"2.2\",\"2.1\",\"2.0\",\"1.9\",\"1.8\",\"1.7\",\"1.6\",\"1.5\",\"1.4\",\"1.3\",

"1.2","1.1","1.0").

Example:

In this example the DOUBLE_STAGGERED_REPEAT fragment is combined with the POINT fragment (note the boresight alignment for the NIS) to take a multi-filter light curve. Seq 16 executes at the beginning of the CAS and every 527 seconds for 39 iterations. Seq 6 executes 15 seconds after the beginning of the CAS and every 1581 seconds for 13 iterations.

```
request(MSI_MSLtCv3_GM,
        START_TIME, 1999-005T23:00:00.0,
        REQUESTOR, "MSI",
        PROCESSOR, CTP1,
        KEY, "MSI")

activity(1,
        SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_AUTO_EXPOSE,10,1,12,0,2500,750,87)
),
activity(2,
        SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_SEQ_DEF,16,1,1,"NONE","ON","FAST",16,"AUTO",0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
),
activity(3,
        SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_SEQ_DEF,6,7,2,"NONE","ON","FAST",16,"AUTO",0,0,0,0,0,0,0,0,0,1,2,3,4,5,7,0)
),
activity(4,
        SCHEDULED_TIME,\600\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_POINT_DOUBLE_STAGGERED_REPEAT,0,15,16,527,39,6,1581,13,"2.1",
595,4,0,0,0,0,0,0,0,9999939,-0.0024574,0.0024766)
),
activity(5,
        SCHEDULED_TIME,\20553\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_RELEASE_ATTITUDE)
),
end;
```

CAS_MSI_TRIPLE_REPEAT

The TRIPLE_REPEAT fragment repeatedly executes 3 sequences as a triplet. The triplet executes at constant time intervals. Time spacing between the individual sequences is also fixed.

This CAS was designed for a specific purpose and is primarily used in the 200x200 orbit color observations. One of the means of 'cleaning' color images is to take the image twice, once with the normal exposure, followed immediately by a zero exposure frame through the same filter. This was originally accomplished by using a double repeat. We take the first set of filters with autoexposure, and then the same set of filters all with manual exposure, 0ms. The problem was that during all but the last image of the second set, the filter wheel is repositioning during the readout. Using the TRIPLE_REPEAT, we can take a single frame immediately after that last manual exposure image of the second set so that the filter wheel is in motion during the readout of that last frame. Then we set the compression to be a special Table 7 which will compress the spot out of that frame (32:1) so it takes up almost no space on the recorder.

(are we going to use this Scott?)

Note that in this CAS, IM_TYPE is defined separately for each of the three sequences.

Parameters:

START IMAGING: Time in seconds relative to start of cas to begin sequencing.

seq_1: First sequence number in pair (RANGE,\1...30\).

delta_1: Time in seconds between executions of SEQ_1 and SEQ_2.

IMAGE_TYPE_1: Compression (RANGE,\"2.5\",\"2.4\",\"2.3\",\"2.2\",\"2.1\",\"2.0\",\"1.9\",\"1.8\",\"1.7\",\"1.6\",\"1.5\",

"1.4","1.3","1.2","1.1","1.0").

seq_2: Second sequence number in pair (RANGE,\1...30\).

delta_2: Time in seconds between executions of SEQ_2 and SEQ_3.

IMAGE_TYPE_3: Compression (RANGE,\"2.5\",\"2.4\",\"2.3\",\"2.2\",\"2.1\",\"2.0\",\"1.9\",\"1.8\",\"1.7\",\"1.6\",\"1.5\",\"1.4\",\"1.3\",
"1.2","1.1","1.0").

seq_3: First sequence number in pair (RANGE,\1...30\).

delta_3: Time in seconds between executions of SEQ_3 and SEQ_1.

IMAGE_TYPE_3: Compression (RANGE,\"2.5\",\"2.4\",\"2.3\",\"2.2\",\"2.1\",\"2.0\",\"1.9\",\"1.8\",\"1.7\",\"1.6\",\"1.5\",
"1.4","1.3","1.2","1.1","1.0").

Iterations: Number of times to execute sequence triplet

Example:

In the following example, Seq 24 is executed at the beginning of the CAS, 8 seconds later Seq 23 executes, 8 seconds later Seq 10 executes, and 759 seconds after that Seq 24 executes again to begin the second iteration of the triplet of sequences. This continues for a total of 24 iterations.

```
request(MSI_lat_mid,
        START_TIME, 1999-030T10:45:00.0,
        REQUESTOR, "MSI",
        PROCESSOR, CTP1,
        KEY, "MSI")

    activity(1,
        SCHEDULED_TIME,\0\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_AUTO_EXPOSE,10,1,12,5,2500,750,87)
    ),
    activity(2,
        SCHEDULED_TIME,\20\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_POINT_TRIPLE_REPEAT,0,24,8,"1.3",23,8,"1.7",10,759,"32",24,300,4,0,0,0,0,1,0,0)
    ),
    activity(3,
        SCHEDULED_TIME,\18575\,FROM_PREVIOUS_START,
        MSI(CAS_MSI_RELEASE_ATTITUDE)
    ),
end;
```

5) Miscellaneous MSI CASs

CAS_MSI_RELEASE_ATTITUDE

This CAS must be included as the last activity of every REQUEST that contains pointing CASs, unless we are doing back-to-back MSI and/or NIS observations. Basically, it has to be there when we are finished using s/c pointing for any period of time greater than a few minutes. It signals to SEQGEN that MSI is releasing attitude control. No commands are sent to the spacecraft as a result of issuing this CAS.

Parameters:

None

CAS_MSI_PARK_MODE

This CAS commands the spacecraft to a safe configuration ... as Karl exactly what it does!!
GS2 point?

Parameters:

None

Duration:

10 minutes

6) MSI Combination CASs

Combining instrument fragments with G&C fragments is pretty simple. No additional parameters are added, you simply string the fragment-level parameters together in the following order (instrument, POINT, SCAN).

Example:

CAS_MSI_POINT_SCAN_SHOOT

Parameters:

START IMAGING:

SEQ 1:

DELTA 1:

ITERATIONS:

IMAGE TYPE:

Slew Duration:

DS40 coord sys:

DS40 aimpt x:

DS40 aimpt y:

DS40 aimpt z:

DS40 aimpt short cut:

DS40 boresight x:

DS40 boresight y:

DS40 boresight z:

Reuse previous scan:

Scan duration:

DS56 coord sys:

DS56 rev x:, DS56 rev y:, DS56 rev z:

DS56 hold place x, DS56 hold place y, DS56 hold place z:

DS56 rate dur x, DS56 rate dur y, DS56 rate dur z:

DS56 rate x, DS56 rate y, DS56 rate z:

DS56 pause dur x, DS56 pause dur y, DS560 pause dur z:

Example:

In the following example, the SHOOT fragment is combined with the POINT_SCAN to put Eros in each corner of the FOV, where an image is taken each time. First, the virtual boresight definition in the POINT causes the MSI FOV to slew to an initial position slightly down and to the left of nadir, putting Eros in the upper right corner of the FOV for the first frame. The SCAN causes the FOV to scan up, to the right, down, and finally back to the left to the starting position. The star images occur in the upper right, lower right, lower left, upper left, and finally the upper right part of the FOV. Images are taken at 0, 33, 99, 132, and 198 seconds from the beginning of the CAS. These times correspond to the end points of each leg of the mosaic scanning.

```
request(MSI_SBF_TEST,  
        START_TIME, 1998-324T08:50:00.0,  
        REQUESTOR, "MEB",  
        PROCESSOR, CTP1,  
        KEY, "MSI")
```

```

activity(1,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  MSI(CAS_MSI_AUTO_EXPOSE,10,0,2,0,3000,750,87)
),
activity(2,
  SCHEDULED_TIME,\600\,FROM_PREVIOUS_START,
  MSI(CAS_MSI_POINT_SCAN_SHOOT,5,29,0,"2.5",29,33,"2.5",29,99,"2.5",29,132,"2.5",
29,198,"2.5",0,00:00:00,"1.3",0,00:00:00,"1.3",0,00:00:00,"1.3",0,00:00:00,"1.3",0,00:00:00,"1.3",
0,00:00:00,"1.3",0,00:00:00,"1.3",.595,4,0,0,0,0,0.99980962,0.00872654,0.01745174,FALSE,200,4,0,1,1,0,0,1,0,66,33,0,.03,-
.03,0,33,66)
),
activity(3,
  SCHEDULED_TIME,\203\,FROM_PREVIOUS_START,
  MSI(CAS_MSI_PARK_MODE)
),
end;

```

7) NIS Turn-On CASs

Remember, the MSI and NIS instruments should be turned on during an attended track (9 am - 5pm ET where EST=UTC-5 and EDT=UTC-4).

The NIS has two "on" CASs. NIS_MAG_DPU_ON CAS turns on the DPU as well as the NIS. The NIS_ON command only turns on the NIS-- it assumes that the DPU is already on. Turn on of the NIS should be coordinated with the MAG sequencer (Brian Anderson). MAG usually turns on 15 minutes after the NIS (it shares the DPU).

CAS_NIS_MAG_DPU_ON

Parameters:

None

Example:

```

request(NIS_MAG_DPU_ON,
  START_TIME, 1999-083T01:30:00.0,
  REQUESTOR, "meb",
  PROCESSOR, CTP1,
  KEY, "NIS")

activity(1,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_MAG_DPU_ON)
),
end;

```

To turn on ONLY the NIS, use:

```

request(NIS_ON,
  START_TIME, 1999-083T01:30:00.0,
  REQUESTOR, "meb",
  PROCESSOR, CTP1,
  KEY, "NIS")

activity(1,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,

```

```
        NIS(CAS_NIS_ON)
    ),
end;
```

8) NIS Sequence Definition

As with MSI, we must define NIS data acquisition sequences and load them into DPU memory before they can be executed. The NIS_LOAD_TRANS_SEQ_DEF CAS loads the parameters which define a single NIS sequence into DPU memory.

Will we be loading these in a REQUEST at start of sequence, or just in individual REQUESTs? Need to write a blurb on that?

CAS_NIS_LOAD_TRANS_SEQ_DEF

A NIS sequence consists of a sequence identifier (Seq_num) followed by parameters which control instrument operation. These parameters define a 'scan', during which a set of 'observations' are taken. The mirror may be stepped fixed number of steps between each observation for the duration of the scan. Multiple identical scans may be performed. If so, the mirror is repositioned to its starting position prior to the start of each new scan. Periodically, the integration time of an observation may be truncated by a specified amount, and a rest period and dark period inserted.

Sequence numbers 0, 1 and 2 are hard-coded into the NIS flight s/w. We should never try to define a sequence using those numbers. We can redefine sequences 3 - 15 only. If a sequence has been defined in the past and the NIS or the NIS/MAG DPU is shut off (is this true? Or is it only if the DPU is shut off as well? Or is the DPU always shut off if the NIS is shut off?), it will need to be redefined.

Parameters:

Seq_num: Enter 3 - 15. Be careful not to overwrite, unless you really want to overwrite!

Num_Scans(nscans): **Range??** 1 - 65535. The number times to execute the scan. A 'scan' consists of a set of Num_Obs observations, each followed by a dead period where the mirror may be stepped each time by Mirror_steps_between_obs positions. After the final observation, the mirror is repositioned to its starting position, and the scan is repeated if NUM_SCANS > 1. Note, Num_Scans should NEVER be set to 0. If Num_Scans = 0, then the sequence will repeat infinitely!!

Seconds between scans (SBS): **Range??** Seconds between scans. This is a wait time at the end of a scan before the start of the next. It is used for repositioning the scan mirror back to the starting position. Currently the stepping rate is set for 4 mirror positions per second. Karl recommends adding an extra second (on top of that) for margin. So, to reposition the mirror 8 positions, we should make SBS = 8/4 = 2 seconds, plus one for margin = 3 seconds. The mirror IS repositioned after the last scan in the sequence. Remember to leave time for that.

Num_Obs(nobs): Number of observations in a scan (or number of mirror positions if the mirror is stepping).

Cal_interval (CALINT): **Range??** Calibration interval. If Cal_interval = n, then on the nth observation from the start of the sequence, and every n observations thereafter, the integration time of that observation is truncated by Num_rests + Num_darks. A rest period of length (Num_rest) followed by the acquisition of Num_darks darks is inserted into the observation following the data acquisition period. The total time spent during this observation remains equal to Spectra_per_obs. Cal_interval cannot be 0. If you do not want any darks taken, then enter 1 here, and 0 for Num_darks.

Spectra_per_obs (NSPEC): Integer between 0 and 63. In observations where no darks are acquired, this is the number of spectra to accumulate (sum) in each observation. **A single spectrum is 1 second long.** If darks are taken, the total duration of the observation still remains equal to Spectra_per_obs. The integration time is truncated such that integration time= Spectra_per_obs - Num_Rests+Num_darks.

Num_rests (NRESTS): Integer between 0 and 63. Number of rest seconds (or number of dead seconds) inserted between spectra and dark acquisition during a calibration observation. Gives time to close shutter and allow for fading of bright spectra.

Num_darks (NDARK): Integer between 0 and 63. Number of dark spectra (1 sec each) to take during a calibration observation.

Mirror_steps_between_obs (NSTEPS): **Range??** Number of mirror positions to step between each observation (sign indicates direction). Positive moves to higher mirror positions (moves mirror to the 'left'). Negative moves mirror to lower positions (to the 'right'). One mirror position step is .4 deg (approximately equal to the width of a NIS FOV).

Seconds between obs (SBO): **Range??** Seconds of dead time between observations (for mirror stepping and "memory" fade).

This dead interval is inserted following each observation in the scan EXCEPT after the last observation of the scan. To calculate time for mirror position stepping, use 1 second for 4 mirror positions.

Notes:

1. Important note: Always be sure that the time that the sequence definition command occurs does not coincide with the time of a beginning of slew. This is actually not a problem on the spacecraft, but it gives SEQGEN a belly-ache. Give at least 5 seconds for a definition before starting a point or scan or execute.

2. Also, remember that the NIS Sequence Definition *does not specify the starting mirror position, gain setting, or desired aperture*. This is done within the CASs which have been designed to execute NIS Sequences.

3. To calculate how much time a sequence will take to execute use the following formula:

$$\text{Time} = \text{nscans} * (\text{nobs} * (\text{nspec} + \text{sbo}) - \text{sbo} + \text{sbscans}) - \text{sbscans}$$

Predefined Sequences:

These sequences are coded into the NIS flight s/w.

0 (DEFAULT) - Seq 0 is an infinitely repeating sequence with zero observations. It is the basic idle mode of the spectrometer and produces no science records. This is the state that NIS is always in unless we are executing a specified sequence (1-15). Going into this mode is automatic (the on-board software does it). In other words, we do not command NIS to go to this state, it knows to do it.

1 (SNAP) Take 1 spectrum of 1 second length (used for marker between scans). Karl has programmed most of the NIS execution CASs to issue Seq 1 immediately following the execution of any normal sequence. Noam suggested this so that it would serve as a marker in the data stream for when the normal data acquisitions sequences end.

(how much time does it take?)

These have been removed from the SUPERSCAN cas's.

2 (CONTINUOUS) - Seq 2 is an indefinitely repeating sequence of 1 second spectra to produce science records at maximum rate. This is called the 'debug' sequence.

USE?

Examples for defining other sequences:

3,1,0,300,1,1,0,0,0,0 - Take 300 1 second spectra once. No mirror position changes. Terminology used sometimes for this is when someone says to take 300 'individual' spectra. That means, 1 second spectrum per each observation. No summing occurs.

7,1,0,11,1,50,2,16,0,2 - Take 11 observations each with 32 s integration, followed by 2 seconds of rest, followed by 16 sec of darks. Total time for each observation is 50 sec. No mirror position change.

13,257,2,7,7,8,2,4,1,1 - Take 7 observations in each scan, each with 8s integration. On the 7th observation of each scan, truncate the integration period to 2 seconds, and follow it by 2 sec of rest, and 4 sec of darks. Move mirror one step between observations during a scan, then reposition back to the starting position after each scan. Perform 257 such scans.

Notes:

To calculate the time that an observation like this will take uses the following formula:

$$\begin{aligned} \text{Time} &= \text{nscans} * (\text{nobs} * (\text{nspec} + \text{sbo}) - \text{sbo} + \text{sbscans}) - \text{sbscans} \\ \text{Time} &= 257 * (7 * (8 + 1) - 1 + 2) - 2 = 16446 \text{ s (IN ABOVE EXAMPLE)} \end{aligned}$$

9) NIS Calibration CASs

Several canned calibration CASs are available to NIS. The caltarget observation is a common calibration sequence used by NIS. There are two CASs available which can command a caltarget, the NIS_CRUISE_CAL and NIS_CAL.

There are two other CASs that were used for test in November, 1998, NIS_ZERO_SUN and NIS_OFF_SUN. They do what NOAM? These are not to be used unless given permission by the OPS gang.

NEED PRECISE DESCRIPTION OF WHAT HAPPENS IN THESE CALS (i.e., what happens with pointing, what sequences are called, separated by what timing deltas, etc.!)

Can insert into each CAS section..

CAS_NIS_CRUISE_CAL

The CAS_NIS_CRUISE_CAL is used primarily during 'cruise', when Eros is not close enough to add to the solar illuminated ... (needs more specific info here about why and when it is used... noam and Scott?)

Parameters:

None

Details:

SFSP = Seconds From Scheduling Point *

-900 Start slew to (DS40 = 900,2,0,0,1,1,0,0,1,0,0,1,.08716, 0,.99619,0)

→ coord=SCI, aimpt=n/a, rollvect=X, aimshort=sun, rollref = (0,0,1),
boresight = (.08716, 0, 99619) or mostly z but a little x,
roll alg = use vector (z axis of J2k).

So, panels move 5 deg off sun toward -x. X is roll vect, close to J2k Z axis.

```
-1 Define NIS Sequence 10,1,0,10,1,16,2,4,0,2 (takes 178 sec to complete)
0 Execute NIS Seq. 10 at MirPos=40, Wide, Gain = 1X
220 Execute NIS Seq. 10 at MirPos = 0, Wide, Gain = 1X
420 Execute NIS Seq. 10 at MirPos = 0, Narrow, Gain = 1X
620 Execute NIS Seq. 10 at Mir Pos = 0, Wide, Gain = 10X
820 Execute NIS Seq. 10 at MirPos = 0, Narrow, Gain = 10X
998 End
```

DURATION = 31 min 38 sec

Notes:

Example:

The following is a real REQUEST for a NIS CRUISE_CAL. The NIS_CRUISE_CAL CAS takes about 32 minutes to complete. I added 8 minutes just to be sure. (8 sounds like a lot of time?) The BUFFER_FLUSH CAS must be issued at the end of every NIS REQUEST which causes data to be recorded (see BUFFER_FLUSH CAS for more details).

```
request(NIS_Caltarg_n5,
        START_TIME, 1999-109T10:30:00.0,
        REQUESTOR, "meb",
        PROCESSOR, CTP1,
        KEY, "NIS")

    activity(1,
        SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_CRUISE_CAL)
    ),
    activity(2,
        SCHEDULED_TIME,\0:40:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_BUFFER_FLUSH)
    ),
    activity(2,
        SCHEDULED_TIME,\30\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_PARK_MODE)
    ),
end;
```

CAS_NIS_CAL

Once Eros is large enough to add light to the caltarget, (what is the specific size? Or SNR?) the NIS_CAL CAS is used (it points the spacecraft to obscure Eros). Everything else is the same as in the CAS_NIS_CRUISE_CAL.

Parameters:

None

SFSP = Seconds From Scheduling Point *

```
-900 Start slew to (DS40 = 900,3,0,0,1,-1,0,0,1,0,0,1,.08716, 0,.99619,3)
      coord = ACI, aimpt = n/a, rollvect= - X, aimptsel = sun, rollref= j2k z axis,
      boresight = mostly z, 5 deg toward +x. So panels will tilt 5 deg
-1   Define NIS Sequence 10,1,0,10,1,16,2,4,0,2 (takes 178 sec to complete)
    0   Execute NIS Seq. 10 at MirPos=40, Wide, Gain = 1X
    220 Execute NIS Seq. 10 at MirPos = 0, Wide, Gain = 1X
    420 Execute NIS Seq. 10 at MirPos = 0, Narrow, Gain = 1X
    620 Execute NIS Seq. 10 at Mir Pos = 0, Wide, Gain = 10X
    820 Execute NIS Seq. 10 at MirPos = 0, Narrow, Gain = 10X
    998 End
DURATION = 31 min 38 sec
```

The DS40 in this case

Example:

```
request(NIS_Caltarg_n5,
        START_TIME, 1999-109T10:30:00.0,
        REQUESTOR, "meh",
        PROCESSOR, CTP1,
        KEY, "NIS")

activity(1,
        SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_CAL)
),
activity(2,
        SCHEDULED_TIME,\0:40:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_BUFFER_FLUSH)
),
activity(2,
        SCHEDULED_TIME,\30\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_RELEASE_ATTITUDE)
),
end;
```

CAS_NIS_ZERO_SUN

Description?

Parameters:

None

Details:

SFSP = Seconds From Scheduling Point *

```
-900 Start setup slew, which takes the NIS target (and panels) 5 deg off sun (in what direction?)
      (DS40 = 900,2,0,0,1,1,0,0,1,0,0,1,.08716, 0,.99619,3)
-1   Define NIS Sequence 10,1,0,10,1,16,2,4,0,2 (takes 178 sec to complete)
    0   Execute NIS Seq. 10 at MirPos=0, Wide, Gain = 1X
    220 Select Mirror Position 40
    245 Execute NIS Seq. 10 at MirPos = 0, Wide, Gain = 1X
    465 Select MirPos = 40
    490 Execute NIS Seq. 10 at MirPos = 0, Wide, Gain = 1X
```

710 Select MirPos = 40
735 Execute NIS Seq. 10 at Mir Pos = 0, Wide, Gain = 1X
955 Select MirPos = 40
980 Execute NIS Seq. 10 at MirPos = 0, Wide, Gain = 1X
1200 Select MirPos = 40
1225 Execute NIS Seq. 10 at MirPos = 0, Wide, Gain = 1X
1445 End
DURATION = 24m5s

Duration: 24m5s

CAS_NIS_OFF_SUN

Description?

Parameters:

None

SFSP = Seconds From Scheduling Point *

-900 -15m Start setup slew, takes the NIS target (and panels) 5 deg off sun (in what direction?)
(DS40 = 900,2,0,0,1,1,0,0,1,0,0,1,.08716, 0,.99619,0)
-1 -1 Define NIS Sequence 10,1,0,10,1,16,2,4,0,2 (takes 178 sec to complete)
0 0 Execute NIS Seq. 10 at MirPos=0, Wide, Gain = 1X
200 3m20 Command GS5 Point
540 9m Begin slew (DS40 = 600,2,0,0,1,1,0,0,1,0,0,1,.08716, 0,.99619,0)
1140 19m Execute NIS Seq. 10 at MirPos=0, Wide, Gain = 1X
1340 22:20 Command GS5 Point
1680 28m Begin slew to (DS40 = 600,2,0,0,1,1,0,0,1,0,0,1,.08716, 0,.99619,0)
2280 38m Execute NIS Seq. 10 at MirPos=0, Wide, Gain = 1X

Duration: 41:30

CAS_NIS_BUFFER_FLUSH

The BUFFER_FLUSH CAS must be issued at the end of every NIS REQUEST which causes data to be recorded. ?? If the NIS buffer, which holds 1 packet (2 1/2 spectra 2864 bits) is not full it will not send the packet out to the tape and so that packet will not be read down. The buffer flush sends everything from the buffer to the tape to be read down.

Parameters:

None

Duration: ??

10) NIS Sequence Execution CASs

As with MSI, several fragments are available to execute one or more pre-defined NIS sequences.

CAS_NIS_EXECUTE

CAS_NIS_EXECUTE executes a single NIS sequence once. In general, the NIS_EXECUTE CAS is used more than any other because it uses much less of the command memory to run than the others, even the single repeat. Also, often the NIS is riding along with an MSI observation that is commanding the pointing.

The sequence always begins to execute at the Scheduling Point (it is not possible to offset the start of the execution fragment as with some other fragments). Set up of mirror position, aperture, and gain occurs prior to the Scheduling Point.

Parameters:

START SEQUENCING: Number of seconds prior to sequencing the CAS backs up to begin set up (mainly needed for moving mirror to required position). **Need more info on how much time to give for various things...**

seq_num: Be sure you have the right one.

mirror_position: Mirror position (1 - 350). Used to setup starting mirror position prior to execution of the sequence. **Need to put in times for stepping mirror!!**

aperture_state: Used to setup starting aperture state for the sequence There are 4 options: "SLIT" = narrow, "BOTH_IN" = Both slits closed (dark), "BOTH_OUT" = wide, "SHUTTER" = shutter closed (dark).

gain: Two options, "1X" and "10X". Normal data collection uses "10X".

Example:

```
request(NIS_GlobalColor2,
        START_TIME, 1999-018T18:00:00.0,
        REQUESTOR, "NIS",
        PROCESSOR, CTP1,
        KEY, "NIS")

        activity(1,
                SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
                NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,11,286,2,5,5,12,2,4,1,1)
        ),
        activity(2,
                SCHEDULED_TIME,\10\,FROM_PREVIOUS_START,
                NIS(CAS_NIS_EXECUTE,0,11,"73","SLIT","10X")
        ),
        activity(3,
                SCHEDULED_TIME,\20553\,FROM_PREVIOUS_START,
                NIS(CAS_NIS_BUFFER_FLUSH)
        ),
end;
```

In this case we start at mirror position 73 and go to 77 (5 positions). The integration time is 8 seconds at each position. There is a trade off between the coverage of the MSI FOV and the integration time. In this case we don't cover the full FOV but we get a good integration time. We calculate the number of scans (286) to cover the entire period of time for the MSI observation. When in doubt, give it a couple of extra scans.

Duration of sequence = nscans * (nobs * (nspec + sbo) - sbo + sbscans) - sbscans

$$20553 = \text{nscans} * (5 * (12 + 1) - 1 + 2) - 2$$

CAS_NIS_SINGLE_REPEAT

Normally, there is no reason to use a single repeat because the same thing can usually be accomplished by setting the Num_scans parameters to be > 1 in the Sequence definition. We should avoid using the SINGLE_REPEAT because it wastes memory on the spacecraft. However, there were some special circumstances which required it's use in the past. **(weren't there?)**

Parameters:

START SEQUENCING: Time in seconds from Scheduling Point to begin first execution of seq_num 1.

SET UP: Seconds prior to Scheduling Point to begin configuration

mirror_step: starting mirror position (mirror is stepped to this position during setup time)

aperture_state: aperture state

gain: Gain state

seq_num_1: Sequence Id of sequence to be executed

interval_1: Time in seconds between starts of seq_num_1

iterations: Total number of times seq_num_1 is executed

CAS_NIS_DOUBLE_REPEAT

The DOUBLE_REPEAT is used to repeatedly execute a pair of sequences. It is generally used when we need to alternate execution of two different sequences, or when the same sequence is repeated but some or all of the setup parameters must toggle during alternate executions. For instance, the DOUBLE_REPEAT is used during the MSI light curves because NIS changes apertures between sequence executions (switching back and forth between narrow and wide).

Notice that there are three sets of "setup" parameters (mirror position, aperture, gain state) for the double repeat. The first set of setup parameters is used to get the instrument in the correct configuration prior to the initial execution of the first sequence (mainly needed for setting up mirror position). **it can take up to xxx (sec/ pos)?**. The second set of setup parameters is used whenever the first sequence is executed. The third set is used whenever the second sequence is executed.

Be careful with the use of "NO CHANGE" in the second and third sets of setup parameters. If the mirror position, aperture, or gain must alternate or change between execution of seq_num_1 and seq_num_2, none of these parameters in either seq_num_1 or seq_num_2 should contain "NO CHANGE". Remember that the initial setup (first of the three configuration sets) is only good for the very first execution of seq_num_1. For instance, if you had "NO CHANGE" in the seq_num_1 setups, and then explicit values for any of the parameters in the seq_num_2 setup, those seq_num_2 values would carry through to the second and all subsequent executions of seq_num_1. In fact, all executions of seq_num_1 AND seq_num_2 would end up using the seq_num_2 setup values.

Parameters:

START SEQUENCING: Time in seconds from Scheduling Point to begin first execution of seq_num_1.

SET UP: Seconds prior to Scheduling Point to begin configuration

mirror_step: Starting mirror position

aperture_state: Starting aperture state

gain: Starting gain state

seq_num_1: Sequence ID of first sequence in pair.

mirror_pos_1: Mirror position used whenever seq_num_1 is executed

aperture_1: Aperture used whenever seq_num_1 is executed

gain_1: Gain state used whenever seq_num_1 is executed

interval_1: Time in seconds between start of execution of seq_num_1 and seq_num_2.

seq_num_2: Sequence ID of second sequence in pair

mirror_pos_2: Mirror position used whenever seq_num_2 is executed

aperture_2: Aperture used whenever seq_num_2 is executed

gain_2: Gain state used whenever seq_num_2 is executed

interval_2: Time in seconds between start of execution of seq_num_2 and seq_num_1.

iterations: Number of times to execute the pair of sequences.

Example:

```
request(NIS_MSLtCv2,
        START_TIME, 1999-002T02:15:00.0,
        REQUESTOR, "NIS",
        PROCESSOR, CTP1,
        KEY, "NIS")

    activity(1,
        SCHEDULED_TIME,\0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,11,1,0,4,4,16,2,4,0,0)
    ),
    activity(2,
        SCHEDULED_TIME,\600\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_DOUBLE_REPEAT,0,120,75,"SLIT","10X",11,"75","SLIT","NOCHANGE",
70,11,"75","BOTH_OUT","NOCHANGE",70,146)
    ),
    activity(3,
        SCHEDULED_TIME,\20563\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_BUFFER_FLUSH)
    ),
end;
```

CAS_NIS_SUPERSCAN_1 (previously called LOW_PHASE_SUPER_SCAN)

This CAS was created for use during the Low Phase Flyby InGaAs-Optimized Global Map observations where for each 'mirror position' in a 'scan', we needed to be able to execute four 16 second observations (each of which has 6 seconds of darks taken). In a normal sequence, the mirror would step between each observation. Here we wanted to step only after every 4th observation. So, we created the concept of a 'Super Scan' where you can execute an entire sequence at each mirror position in this Super Scan. The mirror steps one position between each execution of the sequence. We may specify the sequence to be executed at each position in the Super Scan, the total number of positions in the Super Scan, and the total number of Super Scans (or, in other words, number of times the Super Scan is executed).

Another feature of this CAS is that the execution of sequence 1, which usually follows every sequence execution, does **not** occur in this CAS.

Parameters:

Start sequencing: delay time from Scheduling point to start of sequence)... only occurs in CAS_FULL_POINT_SCAN_SUPERSCAN_1 and CAS_FULL_POINT_SUPERSCAN_1)

seq_num: Sequence number to execute at each mirror position in the Superscan.

init_mir_pos: Mirror position that each Superscan will begin on.

aperture: Starting aperture state

gain: Starting gain state

positions_per_super_scan: Number of mirror positions in the Superscan. One sequence execute at each of these positions. One mirror step between each position (hard-coded).

number_of_super_scans: Number of times to execute above Superscan. The mirror is repositioned to initial starting position before start of each Superscan. **HOW MUCH TIME BETWEEN SUPERSCANS, KARL? AND HOW MUCH TIME BETWEEN MIRROR POSITIONS?**

Notes:

1. There are three CASs which use the SUPERSCAN_1 fragment: 1) CAS_NIS_SUPERSCAN_1,
- 2) CAS_NIS_FULL_POINT_SUPERSCAN_1, and 3) CAS_NIS_FULL_POINT_SCAN_SUPERSCAN_1

PROBLEM: There is an inconsistency in fragment parameters in SEQGEN. In CAS_NIS_SUPERSCAN_1, there is no START_SEQUENCING parameter. But it is there for the FULL_POINT_SUPERSCAN_1 and the FULL_POINT_SCAN_SUPERSCAN_1. Orbit can only handle one way or the other. Brian put the START_SEQUENCING parameter into the SUPERSCAN_1 fragment. Therefore, WE MAY NEVER USE CAS_SUPERSCAN_1. We can use the others, though.

Notes:

1. The sequence used must have mirror_steps_between_obs = 0.
2. In this CAS, the mirror is always stepped *one* position between sequences (not selectable), and the mirror stepping direction during the Super Scan is always FORWARD (increasing mirror position numbers), and also not changeable.

CAS_NIS_SUPER_SCAN2 (previously called SUPER_SCAN)

This CAS was designed to allow us to build the NIS Mirror Geometry Test in which we desire to slew Eros through as many of the NIS mirror position FOVs as is allowed by sun constraints. We take 60 individual 1 second spectra while slewing Eros across each FOV; then we step two positions and do another 60 seconds of individual spectra. This way we can do odd mirror positions in one direction, and then even on the return.

SUPER_SCAN2 differs from SUPER_SCAN1 in that it steps two mirror positions between each execution of the sequence (instead of 1). It is also possible to *select the direction of stepping* (you can go backward or forward through the mirror positions).

Parameters:

seq_num: Sequence number to execute at each mirror position in the Super Scan.

init_mir_pos: Mirror position that each Super Scan will begin on.

aperture: Starting aperture state

gain: Starting gain state

positions_per_super_scan: Number of mirror positions in the Super Scan

step_dir: Direction of stepping. Forward means mirror position increases. Reverse means mirror position decreases during each scan.

Notes:

1. This fragment does not have a Start_sequencing parameter.
2. There is only one CAS which contains this fragment, and that is CAS_NIS_SUPER_SCAN2.

11) Miscellaneous NIS CASs

As with MSI, several fragments are available to execute one or more pre-defined NIS sequences.

CAS_NIS_RELEASE_ATTITUDE

This CAS must be included as the last activity of every REQUEST that contains pointing CASs, unless we are doing back-to-back NIS and/or MSI observations. Basically, it has to be there when we are finished using s/c pointing for any period of time greater than a few minutes. It signals to SEQGEN that NIS is releasing attitude control. No commands are sent to the spacecraft as a result of issuing this CAS.

Parameters:

none

Duration: ?

CAS_NIS_PARK_MODE

This CAS commands the spacecraft to a safe configuration ... as Karl exactly what it does!!

GS2 point?

The slew begins at start of CAS?

Parameters: None

Duration: 10 minutes

12) NIS Combination CASs

As with MSI, combining NIS fragments with G&C fragments is pretty simple. No additional parameters are added, you simply string the fragment-level parameters together in the following order (instrument, POINT, SCAN).

Example:

CAS_NIS_POINT_SCAN_EXECUTE

Parameters:START_SEQ:SEQID:MIRROR_STEP:APERTURE_STATE:GAIN_STATE:Slew_Duration: (see above)DS40_coord_sys:DS40_aimpt_x:DS40_aimpt_y:DS40_aimpt_z:DS40_aimpt_short_cut:DS40_boresight_x:DS40_boresight_y:DS40_boresight_z:Reuse_previous_scan:Scan_duration:DS56_coord_sys:DS56_rev_x, DS56_rev_y, DS56_rev_z:DS56_hold_place_x, DS56_hold_place_y, DS56_hold_place_z:DS56_rate_dur_x, DS56_rate_dur_y, DS56_rate_dur_z:DS56_rate_x, DS56_rate_y, DS56_rate_z:DS40_pause_dur_x, DS40_pause_dur_y, DS40_pause_dur_z:

3. OPNAV CASs

Process:

The JPL NAV Team is responsible for determining when and what type of Opnav CASs should be executed throughout the mission; however, the MSI/NIS Team has agreed to fill in target information and submit the sequence requests for all Opnavs. This requires close interaction between MSI/NIS and the NAV Team.

We have agreed upon the following process for scheduling and generation of Opnavs. One week or more prior to the delivery of the Operations Initial Files for a given week of sequencing, an Opnav planning meeting will take place with representatives from Mission Ops, MSI/NIS, and the NAV Team. During this meeting, Bill Owen specifies exactly what types, frequency, and timing of Opnavs he desires. The Operations folks then go off and schedule the generic Opnav CASs desired (without specific pointing parameters filled in). These are delivered in the xxxxx_OPNAV.sasf file.

We will read this file into Orbit. Orbit will create Orbit-compatible REQ files, and then we will fill in the targeting (DS40) parameters based on guidelines set by the nav team. When we write all of our observations for that week into an .sasf file, these Opnav REQs are merged with all of the other MSI and NIS observation requests into the same .sasf file.

ALL Opnavs (except Opnavs A and B) call Seq 27 which is nominally defined as a single filter 4 image, which is autoexp, fast compression, lossy (TABLE 5) compression. FOR ALL OF THE OPNAVs, the sequences are defined outside of the CAS (see Table 3 below). It is therefore possible to redefine this sequence just prior to execution of any Opnav CAS if it were so desired. The autoexposure setup parameters are also defined outside of the CAS. Similarly, it will be possible to appropriately select correct parameters for each execution of an Opnav CAS. We are planning to include an autoexp setup CAS within each REQUEST that calls an Opnav CAS.

Each OPNAV has a PRIORITY parameter. If FALSE, the OPNAV is recorded only once (in the normal data stream). They are played back once in the normal playback data stream. If TRUE, OPNAVs are recorded in two places: first, in the normal data stream, but also in a special buffer, where TCMs and OPNAVs may be recorded. This buffer is dumped first at beginning of next scheduled playback.

CAS_OPNAV_A

OPNAV_A is used during cruise or on approach when Eros is subpixel. It slews the MSI FOV to nadir, then takes 16 pics (2 sec apart) while scanning slowly across the diagonal of a 2x2 pixel rectangle. This slew occurs at a scan rate that moves the boresight 322 microrad in Y direction, and 190 microrad in the Z direction over 32 seconds. We allow a 4 second runup before the first image is taken, and a 4 second rundown, so the scan actually goes for 40 seconds in total.

Parameters:

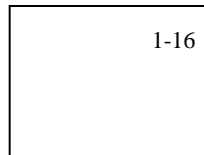
SLEW_DUR: Rng, 1-900sec. (def= 510 sec)

Priority: True or False, (def=false)

Notes:

Detailed Description:

0	(0)	Start setup slew
slew_dur+0	(510)	Start 2x2 scan (see DS56 description below)
slew_dur+4	(514)	MSISR Seq 30, delta=2sec, it = 16
slew_dur+36	(550)	(1 second after last image)
slew_dur+40	(554)	End 2x2 scan
slew_dur+90	(600)	Attitude Release



↖ Slew direction

Opnav A

Duration: 600 sec (using defaults)

DS56: (40, 4, 0, 0, 0, 1, 1, 1, 40, 40, 40, 0.0, 0.0003402, 0.0005765, 0, 0, 0)

Example:

```
request(OPNA_N3,
  START_TIME, 1999-003T22:35:00,
  REQUESTOR, "shp",
  PROCESSOR, CTP1,
  KEY, "OPNAV")

activity(1,
  SCHEDULED_TIME,(0:0:0),FROM_PREVIOUS_START,
  MSI(CAS_OPNAV_A,510,FALSE)
),
end;
```

CAS_OPNAV_B, and CAS_OPNAV_BP

OPNAV_B and OPNAV_BP are used between the time when Eros is approximately equal to one pixel, through some undetermined point when it is less than one FOV. It slews to a selectable aimpoint (nadir is default), and then takes 8 frames. OPNAV_B and BP differ in what frames are taken, but the input parameters are the same for both CASSs.

Parameters:

Slew_Dur: Rng=1-900 sec, (Def=510)

Priority: True or False. (def=false)

Image_Type: Compression estimate based on commanded compression type and scene. Range "1.0" to "2.5" by .1. Used by SEQGEN to estimate data volume. (def=1.3)

DS40_coord_sys: (4)

DS40_aimpt_x: (0)

DS40_aimpt_y: (0)

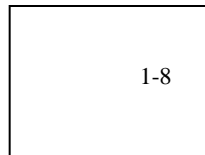
DS40_aimpt_z: (0)

DS40_aimpt_short_cut: (0)

DS40_boresight_x: (1)

DS40_boresight_y: (0)

DS40_boresight_z: (0)



Opnav B and BP

OPNAV_B

The 8 frames include 3 taken with clear filter, manual exposure, 999ms (exposed to see stars), and 5 clear filter autoexposure frames which will be exposed for Eros. OPNAV_B relies entirely on autoexposure for the Eros images.

Detailed Description:

0	(0)	Begin setup slew to nadir
slew_dur+0	(510)	Seq 29 (1 clear filter manexp 999ms for stars)
slew_dur+15	(525)	Seq 28 (5 clear filter autoexp for Eros, delta=1sec)
slew_dur+27	(537)	Seq 29 (1 clear filter manexp 999ms for stars)
slew_dur+30	(540)	Seq 29 (1 clear filter manexp 999ms for stars)
slew_dur+90	(600)	Attitude Release

Duration: 600 sec (using defaults)

OPNAV_BP

The 8 frames include 3 taken with clear filter, manual exposure, 999ms (exposed to see stars), followed by 3 clear filter autoexposure frames which will be exposed for Eros, followed by 2 clear filter manual exposure images exposed for Eros. In OPNAV_BP, two of the 5 original autoexposures in OPNAV_B have been converted to manual exposures as a back-up in case the autoexposure doesn't work for some reason.

Detailed Description:

0	(0)	Begin setup slew to nadir
slew_dur+0	(510)	Seq 29 (1 clear filter manexp 999ms for stars)
slew_dur+15	(525)	Seq 8 (3 clear filter autoexp for Eros, delta=2sec)
slew_dur+23	(533)	Seq 9 (4 clear filter manexp frames, 4ms, 60ms for Eros, and 999ms, 999ms for stars, delta=2)
slew_dur_99	(600)	Attitude Release

Duration: 600 sec (using defaults)

Example:

```
request(OPNB_N3,
  START_TIME, 1999-003T22:47:00,
  REQUESTOR, "shp",
  PROCESSOR, CTP1,
  KEY, "OPNAV")

activity(1,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  MSI(CAS_OPNAV_BP,510,FALSE,"1.3",4,0,0,1,0,0)
),
end;
```

CAS_OPNAV_C (also CAS_OPNAV_CT, CAS_OPNAV_CE)

OPNAV_C is a 1x1 followed by a 2x2, one image at each position (5 frames total). In other words, first we first take a single image (Seq 27), pointing is selectable for this image. This is followed by a 2x2 mosaic, one image (Seq 27) at each position. Absolute pointing of the 2x2 is also selectable.

OPNAV_C is used under two general circumstances. 1) when the size of Eros is subpixel but when it's location is not accurately known (such as during 'Cruise' after a burn), and 2) on approach or just after Orbit insertion when the range is still greater than

400 km. OPNAV_CT and CE were variations of OPNAV_C which only differed by which autoexposure setup was used (slewing and imaging were exactly the same). Since the autoexposure setup has been taken out of all of the OPNAVs, there is no longer a need for these two. They are described below only for historical reconstruction.

The slewing for OPNAV_C is as follows. We slew to some selectable aimpoint (nadir is default), take one image (Seq 27), then slew to starting position of a 2x2 mosaic centered on another selectable aimpoint (may be different from the first aimpoint, but default is to use nadir). One image (Seq 27) is taken at each position in the 2x2. User may modify the amount of time available for the initial setup slew (FIRST_SLEW_DUR parameter), may change the amount of time for imaging at that first position prior to start of second slew which sets up 2x2 (START_SECOND_SLEW parameter), and may also change the amount of time available to complete the slew to the start of the 2x2 mosaic ("SECOND SLEW DUR" parameter). A boresight offset is included in the second DS40 so that the aimpoint you select is for the CENTER of the 2x2 (not the starting position). Mosaic layout for the 2x2 starts in upper left (position1), then go down to lower left (position2), over to lower right (position 3), up to upper right (position 4).

Parameters:

Priority: True or False (def=false)

First Slew Dur: seconds allocated to complete set up slew for 1x1 position. Rng=1,900 sec. (def = 510)

DS40_coord_sys_1: (4)

DS40_aimpt_x_1: (0)

DS40_aimpt_y_1: (0)

DS40_aimpt_z_1: (0)

DS40_aimpt_short_cut_1: (0)

DS40_boresight_x_1: (1)

DS40_boresight_y_1: (0)

DS40_boresight_z_1: (0)

Start Second Slew: Seconds from start of CAS + First_slew_dur to start of slew to 2x2 start position. (def=30 sec).

Second Slew Dur: allocation to complete slew to start of 2x2 (def=300 sec), max=900 sec.

DS40_coord_sys_2: (4)

DS40_aimpt_x_2: (1)

DS40_aimpt_y_2: (0)

DS40_aimpt_z_2: (0)

DS40_aimpt_short_cut_2: (0)

DS40_boresight_x_2: (.9996294)

DS40_boresight_y_2: (-.0166889)

DS40_boresight_z_2: (.0215065)

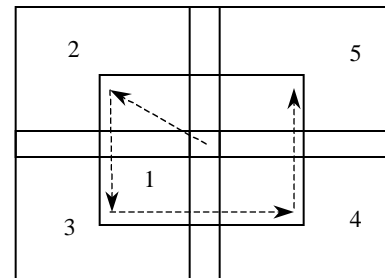
Image Type 1: Compression estimate for first image, (1.3)

Image Type 2: Compression estimate for second image, (1.3)

Image Type 3: Compression estimate for third image, (1.3)

Image Type 4: Compression estimate for fourth image, (1.3)

Image Type 5: Compression estimate for fifth image, (1.3)



Opnav C

Detailed Description:

Shortcuts: FIRST_SLEW_DUR = FSD

START_SECOND_SLEW = SSS

SECOND_SLEW_DUR = SSD

Schedule Point	(0)	Begin setup slew to nadir.
FSD(510)+0	(510)	Seq 27 (one filter 4 image at position 1)
FSD(510)+SSS(30)	(540)	Start of slew to go to start of 2x2
FSD(510)+SSS(30)+SSD(300)	(840)	Seq 27 (one filter 4 image at position 2), and begin DS56 scan
FSD(510)+SSS(30)+SSD(300)+64	(904)	Seq 27 (one filter 4 image at position 3)
FSD(510)+SSS(30)+SSD(300)+64+82	(986)	Seq 27 (one filter 4 image at position 4)
FSD(510)+SSS(30)+SSD(300)+64+82+64	(1050)	Seq 27 (one filter 4 image at position 5)
FSD(510)+SSS(30)+SSD(300)+64+82+64+75	(1125)	Attitude Release

Parameters for DS56: (4, 1, 0, 1, 1, 0, 1, 0, 82, 64, 0.0, 0.030061, 0.0298828, 0, 64, 82)

Default Duration: 1125 seconds

OPNAV_C

Designed to be used on approach after Eros has reached about 100 ?pixels in size, continuing through orbit insertion until Eros gets larger than will fit in a 2x2 (down to about 300km).

We also used this CAS during Cruise, but with Seq 27 defined using clear filter.

OPNAV_CE

This one was designed to be used for the satellite search during a contingency Eros flyby. Sequence 27 was defined externally as clear filter, and setup for autoexposure was defined internally as AUTO_EXP,10,0,3,0,3000,750,87, but has been removed from the CAS.

OPNAV_CT

This one was designed to be used to test the OPNAV C CAS during the original approach to Eros, but we were going to test it at a time when Eros was sub-pixel. Therefore we had to externally define Seq 27 to use clear filter, and setup AUTO_EXP,10,0,2,0,2000,750,87. This is a more conservative autoexposure setup than in OPNAV_CE. Autoexposure has been removed from the CAS.

Example:

```
request(OPNC_N4,
        START_TIME, 1999-005T01:20:00,
        REQUESTOR, "shp",
        PROCESSOR, CTP1,
        KEY, "OPNAV")

        activity(1,
                SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
                MSI(CAS_OPNAV_C.FALSE,510,4,0,0,0,0,1,0,0,30,300,4,0,0,0,0,0.9996294,
-0.0166889,0.0215065,"1.3","1.3","1.3","1.3","1.3")
        ),
end;
```

CAS_OPNAV_C_A_CRUISE

OPNAV_C_A_CRUISE was used during cruise routinely. It performs an OPNAV_C (filter 6 was used) followed by an OPNAV_A. Both are MANUAL exposure 999 msec. Includes a command to go to PARK_MODE at end of CAS.

Parameters:

Concatenation of OPNAV_C and OPNAV_A parameters. See above descriptions.

Example:

```
request(OPNV_C_A_7,
        START_TIME, 1999-035T01:25:00,
        REQUESTOR, "clc",
        PROCESSOR, CTP1,
        KEY, "OPNAV")

        activity(1,
                SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
                MSI(CAS_OPNAV_C_A_CRUISE,510,4,0,0,0,0,1,0,0,30,300,4,0,0,0,0,0.9996294,
-0.0166889,0.0215065,"2.5","2.5","2.5","2.5","2.5")
        ),
end;
```

CAS_OPNAV_D

OPNAV_D is a 2x2 mosaic with one filter 4 lossless/fast/autoexp image (Seq 27) at each position. A boresight offset is included in this DS40 so that the aimpoint you select is for the CENTER of the 2x2 (not the starting position of the mosaic). Mosaic layout for the 2x2 starts in upper left (position1), then lower left (position2), lower right (position 3), upper right (position 4). Maximum slew rate in z (vertical slew) is .030061 deg/sec (3.3 long pixels/sec). Maximum slew rate in y (horizontal slew) is .0298828 deg/sec (5.5 short pixels/sec).

Parameters:

Priority: True or False

Slew_Dur_1: range 1...900, (510)

DS40_coord_sys_1: (4)

DS40_aimpt_x_1: (0)

DS40_aimpt_y_1: (0)

DS40_aimpt_z_1: (0)

DS40_aimpt_short_cut_1: (0)

DS40_boresight_x_1: (.9996294)

DS40_boresight_y_1: (-.0166889)

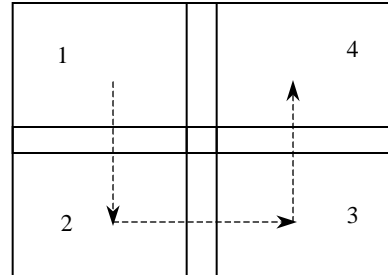
DS40_boresight_z_1: (.0215065)

Image_Type_1: Compression estimate for first image

Image_Type_2: Compression estimate for second image

Image_Type_3: Compression estimate for third image

Image_Type_4: Compression estimate for fourth image



Opanv D

Detailed Description:

Schedule Point	(0)	Begin setup slew to nadir.
Slew_dur(510)+0	(5 10)	Seq 27 (at 1 st position), begin DS56 scan
Slew_dur(510)+64	(574)	Seq 27 (at 2 nd position)
Slew_dur(510)+146	(656)	Seq 27 (at 3 rd position),
Slew_dur(510)+210	(720)	Seq 27 (at 4 th position),
Slew_dur(510)+210+75	(795)	Attitude Release

DS56 Parameters: (212, 4, 1, 0, 1, 1, 0, 1, 0, 82, 64, 0.0, 0.030061, 0.0298828, 0, 64, 82)

Default Duration: 795 seconds

CAS_OPNAV_E

OPNAV_E slews to a selectable position and takes one filter 4, lossless/fast compression image (Seq 27). After 20 seconds slew begins to a second selectable position, and another filter 4 image is taken.

Parameters:

Priority: True or False

Image_Type: (1.3)

Slew_Duration: range 1...900, (510)

DS40_coord_sys: (4)

DS40_aimpt_x: (0)

DS40_aimpt_y: (0)

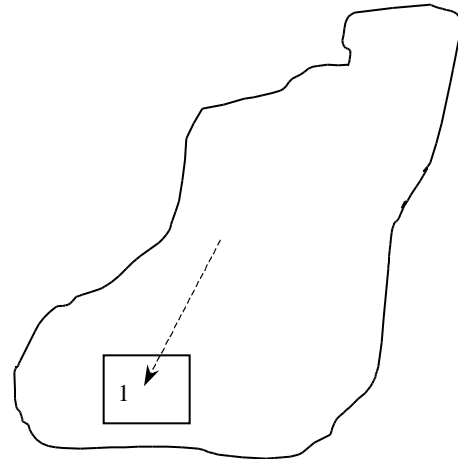
DS40_aimpt_z: (0)

DS40_aimpt_short_cut: (0)

DS40_boresight_x: (1)

DS40_boresight_y: (0)

DS40_boresight_z: (0)



Opanv E1

Detailed Description:

Schedule Point	(0)	Begin setup slew to nadir.
0+slew_dur(510)	(510)	Seq 27 (one filter 4 image)
0+slew_dur(510)+(90)	(600)	attitude release

Default Duration: 600 sec

CAS_OPNAV_E_2

OPNAV_E slews to a selectable position and takes one filter 4, lossless/fast compression image (Seq 27). After 20 seconds slew begins to a second selectable position, and another filter 4 image is taken.

Parameters:

Priority: True or False

Image Type: default (1.3)

Slew Dur 1: (range 1...900), default(510)

DS40 coord sys 1: (4)

DS40 aimpt x 1: (0)

DS40 aimpt y 1: (0)

DS40 aimpt z 1: (0)

DS40 aimpt short cut 1: (0)

DS40 boresight x 1: (1)

DS40 boresight y 1: (0)

DS40 boresight z 1: (0)

Slew Dur 2: (range 1...900), default(510)

DS40 coord sys 2: (4)

DS40 aimpt x 2: (0)

DS40 aimpt y 2: (0)

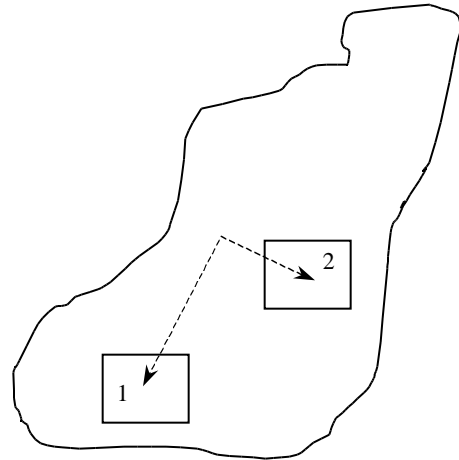
DS40 aimpt z 2: (0)

DS40 aimpt short cut 2: (0)

DS40 boresight x 2: (1)

DS40 boresight y 2: (0)

DS40 boresight z 2: (0)



Opnav E2

Detailed Description:

Schedule Point	(0)	Begin slew to 1 st position
slew_dur_1(510)	(5 10)	Seq 27 (one filter 4 image)
slew_dur_1(510)+20	(530)	Begin slew to second position
slew_dur_1(510)+20+slew_dur_2(180)	(710)	Seq 27 (one filter 4 image)
slew_dur_1(510)+90+slew_dur_2(180)	(1110)	Attitude release

Duration: 1110 sec

CAS_OPNAV_E_3

OPNAV_E slews to a selectable position and takes one filter 4, lossless/fast compression image (Seq 27). After 20 seconds slew begins to a second selectable position, and another filter 4 image is taken. After another 20 second delay we slew to a third selectable position and take another single filter 4 image.

Parameters:

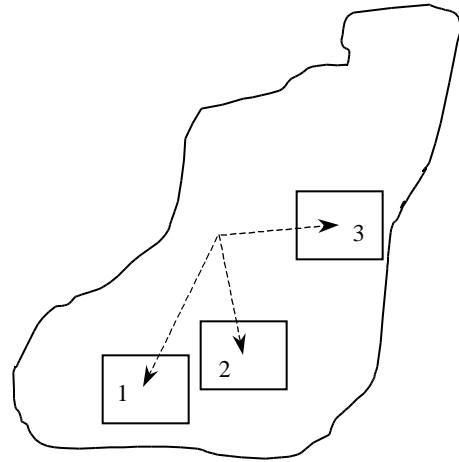
Priority: True or False

Image Type: default (1.3)

Slew Dur 1: (range 1...900), default(510)

DS40 coord sys 1: (4)

DS40 aimpt x 1: (0)
DS40 aimpt y 1: (0)
DS40 aimpt z 1: (0)
DS40 aimpt short cut 1: (0)
DS40 boresight x 1: (1)
DS40 boresight y 1: (0)
DS40 boresight z 1: (0)
Slew_Dur 2: (range 1...900), default(510)
DS40 coord_sys 2: (4)
DS40 aimpt x 2: (0)
DS40 aimpt y 2: (0)
DS40 aimpt z 2: (0)
DS40 aimpt short cut 2: (0)
DS40 boresight x 2: (1)
DS40 boresight y 2: (0)
DS40 boresight z 2: (0)
Slew_Dur 3: (range 1...900), default(510)
DS40 coord_sys 3: (4)
DS40 aimpt x 3: (0)
DS40 aimpt y 3: (0)
DS40 aimpt z 3: (0)
DS40 aimpt short cut 3: (0)
DS40 boresight x 3: (1)
DS40 boresight y 3: (0)
DS40 boresight z 3: (0)



Opnav E3

Detailed Description:

Schedule Point	(0)	Begin setup slew to 1 st position
slew_dur_1(510)	(5 10)	Seq 27 (one filter 4 image)
slew_dur_1(510)+20	(530)	Begin slew to 2 nd position
slew_dur_1(510)+20+slew_dur_2(510)	(1040)	Seq 27 (one filter 4 image)
slew_dur_1(510)+40+slew_dur_2(510)	(1060)	Begin slew to 3 rd position
slew_dur_1(510)+40+slew_dur_2(510)+slew_dur_3(510)	(1550)	Seq 27 (one filter 4 image)
slew_dur_1(510)+100+slew_dur_2(510)+slew_dur_3(510)	(1610)	Attitude Release

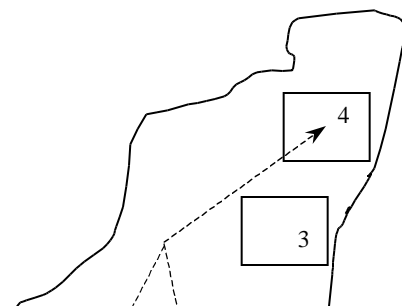
Duration: 1610 sec

CAS_OPNAV_E_4

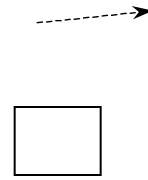
OPNAV_E_4 slews to a selectable position and takes one filter 4, lossless/fast/autoexposure image (Seq 27). After 20 seconds slew begins to a second selectable position, and another filter 4 image is taken. After another 20 second delay we slew to a third selectable position and take another single filter 4 image. After another 20 second delay we slew to a fourth selectable position and take another single filter 4 image.

Parameters:

Priority: True or False
Image_Type: default (1.3)
Slew_Dur 1: (range 1...900), default(510)
DS40 coord_sys 1: (4)
DS40 aimpt x 1: (0)
DS40 aimpt y 1: (0)
DS40 aimpt z 1: (0)
DS40 aimpt short cut 1: (0)
DS40 boresight x 1: (1)
DS40 boresight y 1: (0)
DS40 boresight z 1: (0)



Slew_Dur_2: (range 1...900), default(510)
DS40_coord_sys_2: (4)
DS40_aimpt_x_2: (0)
DS40_aimpt_y_2: (0)
DS40_aimpt_z_2: (0)
DS40_aimpt_short_cut_2: (0)
DS40_boresight_x_2: (1)
DS40_boresight_y_2: (0)
DS40_boresight_z_2: (0)
Slew_Dur_3: (range 1...900), default(510)
DS40_coord_sys_3: (4)
DS40_aimpt_x_3: (0)
DS40_aimpt_y_3: (0)
DS40_aimpt_z_3: (0)
DS40_aimpt_short_cut_3: (0)
DS40_boresight_x_3: (1)
DS40_boresight_y_3: (0)
DS40_boresight_z_3: (0)
Slew_Dur_4: (range 1...900), default(510)
DS40_coord_sys_4: (4)
DS40_aimpt_x_4: (0)
DS40_aimpt_y_4: (0)
DS40_aimpt_z_4: (0)
DS40_aimpt_short_cut_4: (0)
DS40_boresight_x_4: (1)
DS40_boresight_y_4: (0)
DS40_boresight_z_4: (0)



Opnav E4

Detailed Description:

Schedule Point = Start Time	(0) Begin slew to 1 st position
slew_dur_1(510)	(5 10) Seq 27 (one filter 4 image)
slew_dur_1(510)+20	(530) Begin slew to 2 nd position
slew_dur_1(510)+20+slew_dur_2(510)	(1040) Seq 27 (one filter 4 image)
slew_dur_1(510)+40+slew_dur_2(510)	(1060) Begin slew to 3 rd position
slew_dur_1(510)+40+slew_dur_2(510)+slew_dur_3(510)	(1570) Seq 27 (one filter 4 image)
slew_dur_1(510)+60+slew_dur_2(510)+slew_dur_3(510)	(1590) Begin slew to 4 th position
slew_dur_1(510)+60+slew_dur_2(510)+slew_dur_3(510) + slew_dur4(510)	(2100) Seq 27 (one filter 4 image)
slew_dur_1(510)+120+slew_dur_2(510)+slew_dur_3(510) + slew_dur4(510)	(2160) Attitude release

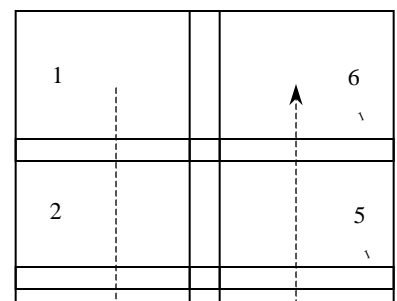
Duration: 2160 sec

CAS_OPNAV_F

OPNAV_F is a 2x3 mosaic with one filter 4 lossless/fast/autoexp image (Seq 27) at each position. Targeting is selectable for the CENTER of the 2x3 (not the starting position of the mosaic); default is nadir. Maximum slew rate in z (vertical slew) is .030061 deg/sec (3.3 long pixels/sec). Maximum slew rate in y (horizontal slew) is .0298828 deg/sec (5.5 short pixels/sec).

Parameters:

Priority: True or False
Slew_Duration: (def=510, rng=1-900)
DS40_coord_sys: (def=4)
DS40_aimpt_x: (0)
DS40_aimpt_y: (0)
DS40_aimpt_z: (0)



DS40 aimpt short cut: (0)
DS40 boresight x: (.9992117)
DS40 boresight y: (-.0333732)
DS40 boresight z: (.0214975)
Image Type 1: Compression estimate for first image (1.3)
Image Type 2: Compression estimate for second image (1.3)
Image Type 3: Compression estimate for third image (1.3)
Image Type 4: Compression estimate for fourth image (1.3)
Image Type 5: Compression estimate for fifth image (1.3)
Image Type 6: Compression estimate for sixth image (1.3)

Opnav F

Detailed Description:

Schedule Point	(0)	Begin setup slew to nadir.
Slew_dur(510)+0	(510)	Seq 27 (at 1 st position), begin scan
Slew_dur(510)+64	(574)	Seq 27 (at 2 nd position)
Slew_dur(510)+128	(638)	Seq 27 (at 3 rd position),
Slew_dur(510)+210	(720)	Seq 27 (at 4 th position),
Slew_dur(510)+274	(784)	Seq 27 (at 5 th position),
Slew_dur(510)+338	(848)	Seq 27 (at 6 th position),
Slew_dur(510)+340+75	(925)	Attitude Release

DS56 Parameters: (340, 4, 1, 0, 1, 1, 0, 1, 0, 82, 128, 0.0, .030061, .0298828, 0, 128, 82)

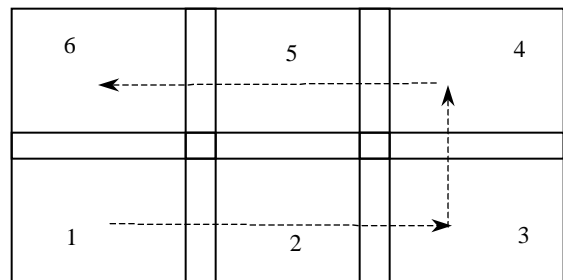
Duration: 925 seconds

CAS_OPNAV_G

OPNAV_G is a 3x2 mosaic with one filter 4 lossless/fast/autoexp image (Seq 27) at each position. Targeting is selectable for the CENTER of the 3x2 (not the starting position of the mosaic); default is nadir. Slew rate in z (vertical slew) is .030061 deg/sec. Slew rate in y (horizontal slew) is .0298828 deg/sec.

Parameters:

Priority: True or False
Slew Duration: (def=510, rng=1-900)
DS40 coord sys: (def=4)
DS40 aimpt x: (0)
DS40 aimpt y: (0)
DS40 aimpt z: (0)
DS40 aimpt short cut: (0)
DS40 boresight x: (.9989355)
DS40 boresight y: (.0166889)
DS40 boresight z: (.0430031)
Image Type 1: Compression estimate for first image (1.3)
Image Type 2: Compression estimate for second image (1.3)
Image Type 3: Compression estimate for third image (1.3)
Image Type 4: Compression estimate for fourth image (1.3)
Image Type 5: Compression estimate for fifth image (1.3)
Image Type 6: Compression estimate for sixth image (1.3)



Opnav G

Detailed Description:

Schedule Point = Start Time	(0)	Begin setup slew to nadir.
Slew_dur(510)+0	(510)	Seq 27 (at 1 st position), begin scan
Slew_dur(510)+82	(592)	Seq 27 (at 2 nd position)
Slew_dur(510)+164	(674)	Seq 27 (at 3 rd position),
Slew_dur(510)+228	(738)	Seq 27 (at 4 th position),

Slew_dur(510)+310	(820)	Seq 27 (at 5 th position),
Slew_dur(510)+392	(902)	Seq 27 (at 6 th position),
Slew_dur(510)+394+75	(979)	Attitude Release

Duration: 979 seconds

DS56 Parameters: (394, 4, 1, 1, 0, 1, 1, 0, 0, 164, 64, 0.0, 0.030061, -0.0298828, 0, 64, 164)

CAS_OPNAV_H

OPNAV_H is a 3x3 mosaic with one filter 4 lossless/fast/autoexp image (Seq 27) at each position. Targeting is selectable for the CENTER of the 3x3 (not the starting position of the mosaic); default is nadir. Slew rate in z (vertical slew) is .030061 deg/sec. Slew rate in y (horizontal slew) is .0298828 deg/sec.

Parameters:

Priority: True or False

Slew Duration: (def=510, rng=1-900)

DS40 coord sys: (def=4)

DS40 aimpt x: (0)

DS40 aimpt y: (0)

DS40 aimpt z: (0)

DS40 aimpt short cut: (0)

DS40 boresight x: (.99985182)

DS40 boresight y: (-.0333732)

DS40 boresight z: (.0429851)

Image Type 1: Compression estimate for first image (1.3)

Image Type 2: Compression estimate for second image (1.3)

Image Type 3: Compression estimate for third image (1.3)

Image Type 4: Compression estimate for fourth image (1.3)

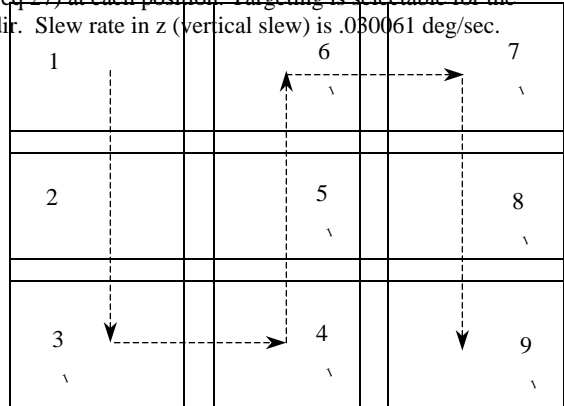
Image Type 5: Compression estimate for fifth image (1.3)

Image Type 6: Compression estimate for sixth image (1.3)

Image Type 7: Compression estimate for seventh image (1.3)

Image Type 8: Compression estimate for eighth image (1.3)

Image Type 9: Compression estimate for ninth image (1.3)



Opnav H

Detailed Description:

Schedule Point = Start Time	(0)	Begin setup slew to nadir.
Slew_dur (510)+0	(5 10)	Seq 27 (at 1 st position), begin scan
Slew_dur(510)+64	(574)	Seq 27 (at 2 nd position)
Slew_dur(510)+128	(638)	Seq 27 (at 3 rd position),
Slew_dur(510)+210	(720)	Seq 27 (at 4 th position),
Slew_dur(510)+274	(784)	Seq 27 (at 5 th position),
Slew_dur(510)+338	(848)	Seq 27 (at 6 th position),
Slew_dur(510)+420	(930)	Seq 27 (at 7 th position),
Slew_dur(510)+484	(994)	Seq 27 (at 8 th position),
Slew_dur(510)+548	(1058)	Seq 27 (at 9 th position),
Slew_dur(510)+550+75	(1135)	Attitude Release

Duration: 1135 seconds

DS56 Parameters: (550, 4, 1, 0, 1, 1, 0, 1, 0, 82, 128, 0.0, 0.030061, 0.0298828, 0, 128, 82)

CAS_OPNAV_I

OPNAV_I is a 1x2 mosaic with one filter 4, lossless/fast/autoexp image (Seq 27) at each position. Targeting is selectable for the CENTER of the 1x2 (not the starting position of the mosaic); default is nadir. Slew rate in z (vertical slew) is .030061 deg/sec. Slew rate in y (horizontal slew) is .0298828 deg/sec.

Parameters:

- Priority: True or False
- Slew_Duration: range = 1, 900 (510)
- DS40_coord_sys: (def=4)
- DS40_aimpt_x: (0)
- DS40_aimpt_y: (0)
- DS40_aimpt_z: (0)
- DS40_aimpt_short_cut: (0)
- DS40_boresight_x: (.99976864)
- DS40_boresight_y: (0.0)
- DS40_boresight_z: (0.2150952)
- Image_Type_1: Compression estimate for first image (1.3)
- Image_Type_2: Compression estimate for second image (1.3)



Opanav I

Detailed Description:

- Schedule Point (0) Begin setup slew to nadir.
- Slew_dur (510)+0 (510) Seq 27 (at 1st position), begin scan
- Slew_dur(510)+82 (592) Seq 27 (at 2nd position)
- Slew_dur(510)+84+75 (669) Attitude Release

Duration: 594 seconds

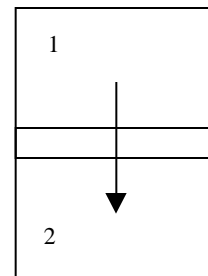
DS56 Parameters: (84, 4, 1, 1, 0, 1, 1, 0, 0, 82, 0, 0.0, 0.030061, 0.0, 0, 0, 0)

CAS_OPNAV_J

OPNAV_J is a 2x1 mosaic with one filter 4, lossless/fast/autoexp image (Seq 27) at each position. Targeting is selectable for the CENTER of the 2x1 (not the starting position of the mosaic); default is nadir. Slew rate in z (vertical slew) is .030061 deg/sec. Slew rate in y (horizontal slew) is .0298828 deg/sec.

Parameters:

- Priority: True or False
- Slew_Duration: (def=510, rng=1-900)
- DS40_coord_sys: (def=4)
- DS40_aimpt_x: (0)
- DS40_aimpt_y: (0)
- DS40_aimpt_z: (0)
- DS40_aimpt_short_cut: (0)
- DS40_boresight_x: (.9998607)
- DS40_boresight_y: (.0166889)
- DS40_boresight_z: (0)
- Image_Type_1: Compression estimate for first image (1.3)
- Image_Type_2: Compression estimate for second image (1.3)



Opanav J

Detailed Description:

- Schedule Point = Start Time (0) Begin setup slew to nadir.
- Slew_dur (510)+0 (510) Seq 27 (at 1st position), begin scan
- Slew_dur(510)+64 (574) Seq 27 (at 2nd position)
- Slew_dur(510)+66+75 (651) Attitude Release

Duration: 651 seconds

DS56 Parameters: (66, 4, 1, 1, 0, 1, 1, 0, 0, 4, 0, 0, -0.0298828, 0, 0, 0)

CAS_OPNAV_K

OPNAV_K is a design-your-own generic mosaic with one filter 4 (lossless/fast/autoexp) image at each position (Seq 27). It contains a DS40 (POINT-only, not FULL), a DS56, and an MSI_SHOOT for which all parameters are free.

Parameters:

Priority: True or False

Slew_Duration: (def=510, rng=1-900)

DS40_coord_sys: (def=4)

DS40_aimpt_x: (0)

DS40_aimpt_y: (0)

DS40_aimpt_z: (0)

DS40_aimpt_short_cut: (0)

DS40_boresight_x: (1)

DS40_boresight_y: (0)

DS40_boresight_z: (0)

Scan_duration: (400)

DS56_coord_sys: (4)

DS56_rev_x: (0)

DS56_rev_y: (1)

DS56_rev_z: (1)

DS56_hold_place_x: (0)

DS56_hold_place_y: (0)

DS56_hold_place_z: (1)

DS56_rate_dur_x: (0)

DS56_rate_dur_y: (58)

DS56_rate_dur_z: (58)

DS56_rate_x: (0)

DS56_rate_y: (.0425)

DS56_rate_z: (.032974)

DS56_pause_dur_x: (0)

DS56_pause_dur_y: (58)

DS560_pause_dur_z: (58)

FRAMES: number of image executions

Im_delta_1: time from start of scan to execute first image (0)

IM_TYPE_1: Compression estimate for first image (1.3)

Im_delta_2: time from start of scan to execute second image (58)

IM_TYPE_2: Compression estimate for second image (1.3)

Im_delta_3: time from start of scan to execute third image (116)

IM_TYPE_3: Compression estimate for third image (1.3)

Im_delta_4: time from start of scan to execute fourth image (174)

IM_TYPE_4: Compression estimate for fourth image (1.3)

Im_delta_5: time from start of scan to execute fourth image (0)

IM_TYPE_5: Compression estimate for fourth image (1.3)

...

Im_delta_12: time from start of scan to execute twelfth image (0)

IM_TYPE_12: Compression estimate for twelfth image (1.3)

Detailed Description:

Schedule Point = Start Time	(0)	Begin setup slew to nadir.
Slew_dur(510)+im_dur_1(0)	(510)	Seq 27 (at 1 st position), begin scan
Slew_dur(510)+im_dur_2(58)	(568)	Seq 27 (at 2 nd position)
Slew_dur(510)+im_dur_3(116)	(626)	Seq 27 (at 3 rd position),
Slew_dur(510)+im_dur_4(174)	(684)	Seq 27 (at 4 th position),

Slew_dur(510)+im_dur_5 (tbd) Seq 27 (at 5th position),
 Slew_dur(510)+im_dur_6 (tbd) Seq 27 (at 6th position),
 ...
 Slew_dur(510)+im_dur_12 (tbd) Seq 27 (at 12th position),
 Slew_dur(510)+scan_dur+75 (tbd) Attitude Release

Duration: tbd seconds

Table 3: OPNAV Sequence Definitions

	OPNAV -A	OPNAV-B				OPNAV_C,D,E, E2,E3,E4,F,G,H ,I,J,K
		OPNAV-BP				
Seq_num	30	29	28	9	8	27
Num_images	1	1	5	4	3	1
Interval	1	1	1	2	2	1
Comp_table	NONE	NONE	NONE	NONE	NONE	NONE
Flag_dpcm	OFF	ON	ON	ON	ON	ON
Comp_alg	NONE	FAST	FAST	FAST	FAST	FAST
Pixels	16	16	16	16	16	16
Mode	MAN	MAN	AUTO	MAN	AUTO	AUTO
Im_dur1	999	999	0	4	0	0
Im_dur2	0	0	0	60	0	0
Im_dur3	0	0	0	999	0	0
Im_dur4	0	0	0	999	0	0
Im_dur5	0	0	0	0	0	0
Im_dur6	0	0	0	0	0	0
Im_dur7	0	0	0	0	0	0
Im_dur8	0	0	0	0	0	0
Im_filt1	0	0	0	0	0	4
Im_filt2	0	0	0	0	0	0
Im_filt3	0	0	0	0	0	0
Im_filt4	0	0	0	0	0	0
Im_filt5	0	0	0	0	0	0
Im_filt6	0	0	0	0	0	0
Im_filt7	0	0	0	0	0	0
Im_filt8	0	0	0	0	0	0

During the Low Phase Flybys, which will occur on February 14, 2000 (northern latitudes), and October 14, 2000 (southern latitudes), the spacecraft will approach the asteroid to within 200 km and view it such that the incidence angle = emission angle = phase angle = 0 degrees. This is a very unique set of observations that require some fancy spacecraft driving. For this purpose, the flyby was split into 4 phases, which depend on the phase angle of the sun on the asteroid: 90-60 degrees, 60-30 degrees, 30-5 degrees and below 5 degrees.

The first flyby occurs during the approach and just before the orbit insertion burn. Time is limited and most observations will be taken below 30 degrees phase. During the October flyby, we will have many hours to obtain data at the higher phase angles as well as the lower angles.

There are primarily four types of observations that are sequenced for NIS for the Low Phase Flyby. Each of these observations has a CAS built for it. The four observations are:

Global Maps	CAS_NIS_FULL_POINT_SCAN_EXECUTE
MPP (Minimum Phase Point) Maps	CAS_NIS_FULL_POINT_SCAN_EXECUTE
Latitude Scans	CAS_NIS_FULL_POINT_EXECUTE
Super Scans	CAS_NIS_SUPER_SCAN

CAS_NIS_FULL_POINT_SCAN_EXECUTE

The FULL_POINT_SCAN_EXECUTE CAS is used for Global Maps and MPP maps, but the values of the pointing and scanning parameters are different for each. The main difference between Global Maps and MPP Maps is that the MPP Maps use the spacecraft body fixed coordinate system while the Global Maps use asteroid body fixed.

This CAS is made up of three fragments: Full Point, DS56 scan, and a NIS Execute.

Global Maps

Global Maps include a spacecraft scan (DS56) along the entire length of the asteroid's x-axis while scanning the NIS mirror back and forth across the asteroid (perpendicular to the scan motion) to create a single global coverage map.

For the **DS40 Full Point** fragment, an asteroid body fixed aimpoint is selected. This is usually at one end of the x-axis, which is one nose of the asteroid. Global maps are taken during two phases of the low phase flyby. Above 60 degrees phase, nominal low orbit pointing is sufficient, although an inertial J2000 roll reference is used and the boresight is selected in order to minimize the panel off sun angles. When the phase angle is below 30 degrees, the roll reference is selected as nadir and the NIS boresight is the negative z-axis. In this case, the global map is separated into two hemisphere maps -- the negative x-axis hemisphere and the positive x-axis hemisphere.

The following are the DS40 pointing parameter values:

For all global maps:

DS40_COORD_SYS	5 (=ABF)
DS40_AIMPT_SHORT_CUT	0 (=Use Aimpoint)

For global maps between 30 and 60 degrees phase:

DS40_AIMPT_XYZ	+x_radius	0	0
DS40_roll_vec_xyz	1	0	0
DS40_roll_ref_xyz	inertial j2k roll reference		
DS40_BORESIGHT_XYZ	sin(phase)	0	-cos(phase)
DS40_roll_algor	0 (=Use Roll Ref)		

For global maps under 30 degrees phase:

Positive x-axis hemisphere map:

DS40_AIMPT_XYZ	+x_radius	0	0
DS40_roll_vec_xyz	0	1	0

Negative x-axis hemisphere map:

DS40_AIMPT_XYZ	-0.00001	0	0
DS40_roll_vec_xyz	0	-1	0
both:			
DS40_roll_ref_xyz	0	0	0
DS40_BORESIGHT_XYZ	0	0	-1
DS40_roll_algor	3 (=Use Nadir)		

For the **DS56 Scan** fragment, there is an asteroid body fixed scan along the x-axis. The duration and the rate of this scan are calculated based on the number of mirror positions across the asteroid and the number of scans along the x-axis to globally cover the asteroid. This is calculated as follows:

obs = number of mirror positions across
 scn = number of scans down
 int = number of spectra (seconds)
 btw = number of seconds between scans

For the duration of the scan:

$$\text{dur} = ((\text{int}+1)*\text{obs}+\text{btw}-1)*\text{scn}-\text{btw}-1 \text{ [sec]}$$

From this, the scan rate can be calculated as follows:

$$\text{rate} = (\text{length of x-axis}) / \text{dur} \text{ [km/sec]}$$

The **NIS Execute** fragment has few parameters. It should be noted that the mirror position is the start position of the scan. For global maps in the 30-60 degrees phase region, the mirror position that falls on the x-axis is a function of the phase angle, according to the following simple equation:

$$\text{mirror pos} = 300 - 2.5*\text{phase}$$

The gain is usually set to "10X." The aperture is set to "SLIT" for narrow slit, and "BOTH_OUT" for wide slit.

Note that for global maps that have phase angles greater than 30 degrees, it is advantageous to select a starting time such that the asteroid x-axis is parallel to the long axis of a nixel at the mid-point of the map. This will produce the best possible map at these phase angles and will minimize the amount of mirror scanning and scan overlapping.

Example of Above 30 degrees Global Map Request:

```
activity(GlobalMapNarrow3secIntegration,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,11,28,6,21,21,3,2,1,1,1)
),
activity(GlobalMapNarrow,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_FULL_POINT_SCAN_EXECUTE,0,11,"150","SLIT","10X",600,
5,22,0,0,1,0,0,0,-0.0389639549,-0.158641383,-0.211791232,.819152044289,0,
-.573576436351,0,FALSE,2377,2,0,0,0,0,0,0,2377,0,0,-.0185107278082,0,0,0,0,0)
),
```

Example of Below 30 degrees Global Map Request:

```
activity(GlobalMapNarrow6secIntegration,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,9,13,5,19,19,6,2,4,1,1)
),
activity(GlobalMapNarrow(PosHalf),
  SCHEDULED_TIME,\0:20:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_FULL_POINT_SCAN_EXECUTE,0,9,"291","SLIT","10X",500,5
,22,0,0,0,1,0,0,1,0,0,0,0,-1,3,FALSE,1739,2,0,0,0,0,0,0,1739,0,0,-.0126509488212
```

```
,0,0,0,0)
),
activity(GlobalMapNarrow(NegHalf),
  SCHEDULED_TIME,\0:30:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_FULL_POINT_SCAN_EXECUTE,0,9,"291","SLIT","10X",40,5,
-000001,0,0,0,-1,0,0,1,0,0,0,0,-1,3,TRUE,1739,2,0,0,0,0,0,1739,0,0,-01265094
88212,0,0,0,0)
),
```

MPP Maps

MPP Maps are mosaics that cover a small region of territory around the Minimum Phase Point (MPP)-- the point on the surface that is directly beneath the spacecraft at zero degrees phase. These maps are timed so that the MPP is in the middle of the center nixel of the mosaic. This requires that there be an odd number of observations across and scans down. The MPP maps usually range in size from 5x3 to 7x5, and take 16 spectra at each position.

For the **DS40 Full Point** fragment, the difficult part is finding the correct boresight to offset the position of the scan so that the center nixel is centered around the MPP. This calculation is as follows:

First, calculate the scan rate:

$$\text{rate} = (0.76 \text{ deg}) / (\text{obs} * (\text{int} + 1 \text{ sec}) + (\text{int} - 1 \text{ sec})) \text{ [deg/sec]}$$

where

obs = number of observations across
scn = number of scans down
int = number of spectra (usually 16 seconds)

From this, we find the offset angle for the boresight in the yz-plane (mostly pointing in the -z direction):

$$\text{theta} = (\text{scn}-1)/2 * 0.76 \text{ deg} + (\text{obs}-1)/2 * \text{rate} \text{ [deg]}$$

Now, we get the xyz of the boresight offset:

x = 0
y = -sin(theta)
z = -cos(theta)

The other parameters are:

DS40_COORD_SYS	5 (=ABF)		
DS40_AIMPT_XYZ	MPP_X	MPP_Y	MPP_Z
DS40_roll_vec_xyz	0	1	0
DS40_AIMPT_SHORT_CUT	0 (=Use Aimpoint)		
DS40_roll_ref_xyz	0	0	0
DS40_roll_algor	3 (=Use Nadir)		

For the **DS56** fragment, the duration is calculated as above for global maps based on the numbers of observations and scans. The scan rate for the MPP map was calculated above as well. Just make sure that the coordinate system is set to spacecraft body fixed, which is given a value of 4.

The NIS execute fragment is the same as for the globals.

Examples of MPP Maps:

```
activity(7x5MPPObservation,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,3,5,2,7,7,16,2,4,1,1)
),
activity(7x3MPPObservation,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,4,3,2,7,7,16,2,4,1,1)
```

```

    ),
    activity(5x3MPPObservation,
        SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,5,3,2,5,5,16,2,4,1,1)
    ),

    activity(7x5MPPObservation,
        SCHEDULED_TIME,\0:40:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_FULL_POINT_SCAN_EXECUTE,0,3,"297","SLIT","10X",500,5
        ,-1.00871,2.33456,-7.91578,0,1,0,0,1,0,0,0,-0.0321608714179,-
        0.999482705378,3,FALSE,597,4,0,0,0,0,0,0,597,0,0,.00633333333,0,0,0,0)
    ),
    activity(7x3MPPObservation,
        SCHEDULED_TIME,\0:40:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_FULL_POINT_SCAN_EXECUTE,0,4,"297","SLIT","10X",500,5
        ,-1.00871,2.33456,-7.91578,0,1,0,0,1,0,0,0,-0.0189007902655,-
        0.999821364108,3,FALSE,357,4,0,0,0,0,0,0,357,0,0,.00633333333,0,0,0,0)
    ),
    activity(5x3MPPObservation,
        SCHEDULED_TIME,\0:40:0\,FROM_PREVIOUS_START,
        NIS(CAS_NIS_FULL_POINT_SCAN_EXECUTE,0,5,"298","SLIT","10X",500,5
        ,-1.00871,2.33456,-7.91578,0,1,0,0,1,0,0,0,-0.0185075511552,-
        0.999828720607,3,FALSE,255,4,0,0,0,0,0,0,255,0,0,0.0083720930233,0,0,0,0)
    ),

```

CAS_NIS_FULL_POINT_EXECUTE

This CAS combines the FULL_POINT fragment with the NIS_EXECUTE fragment.

Latitude scans

There are times when a global map is not appropriate, or when the attitude of the spacecraft must remain constant (such as when the fan beam needs to be on the earth). Latitude scans are the best way to still gather data without causing the spacecraft to perform scans and are useful between the 30 and 60 degree phase regions of the flyby.

The **DS40 Full Point** fragment is given similar parameters as the Full Point for the global maps between 30 and 60 degrees phase. The only difference is that the aimpoint is usually selected as nadir. The number of observations across should cover the long part of the asteroid, and the number of scans should be as many as needed to fill time. It is necessary to let the asteroid rotate for one full rotation in order to get global coverage, but obtaining global coverage is not always the goal for latitude scans.

The **NIS execute** parameter values are the same as for the global maps.

Examples of Lat Scans:

```

activity(LatitudeScan6secIntegration,
    SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
    NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,14,27,7,25,25,6,2,4,1,1)
),

activity(LatitudeScan,
    SCHEDULED_TIME,\0:50:0\,FROM_PREVIOUS_START,
    NIS(CAS_NIS_FULL_POINT_EXECUTE,14,"176","SLIT","10X",500,5,0,0,0
    ,1,0,0,0,-.0389639549,-.1586413831,-.2117912322,.707106781187,0,-.707106781187,0
    )
),

```

CAS_NIS_SUPER_SCAN

The NIS Super Scan is a specialized CAS for optimizing data taken in the NIS InGaAs wavelengths. It is essentially a large MPP map taking 64 spectra. The map is centered on the MPP and is made as large as sees fit by the restrictions on time and rate of

change of the phase angle. There is only one specialized fragment included in this cas with few parameters. These parameters are as follows:

- NIS Sequence
- Initial Mirror Position
- NIS Aperture
- NIS Gain
- Number of Positions per Super Scan
- Number of Super Scans

The last two parameters are the only ones which need explanation. The Super Scan cas will execute the NIS sequence specified, for each mirror position set by the Number of Positions per Super Scan. Then this scan will be repeated by the Number of Super Scans. All the calculations for the slew rates and durations are built into the cas.

It is important the NIS Sequence that the Super Scan executes is built specifically for a Super Scan. While there can be more than one observation (they will be executed at the same mirror position), only one scan is valid. Also, the number of seconds between observations and between scans should be set to zero since the Super Scan cas takes care of all the timing.

```
activity(InGaAsOptimizedLatitudeScan,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,6,1,0,4,1,16,2,6,0,0)
),
request(35PHASE_OUT_InGaAs,
  START_TIME, 1999-066T13:20:00.00,
  REQUESTOR, "NIS",
  PROCESSOR, CTP1,
  KEY, "NIS")

  activity(InGaAsOptimizedLatitudeScan,
    SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
    NIS(CAS_NIS_LOW_PHASE_SUPER_SCAN,6,199,"SLIT","10X",13,13)
  ),
end;

activity(GlobalMapNarrow45PhaseOut,
  SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
  NIS(CAS_NIS_LOAD_TRANS_SEQ_DEF,7,12,2,8,8,16,2,4,1,1)
),
request(15PHASE_OUT_InGaAs,
  REQUEST_STATE, UNEXPANDABLE,
  START_TIME, 1999-065T19:20:00.00,
  REQUESTOR, "NIS",
  PROCESSOR, CTP1,
  KEY, "NIS")

  activity(11x5InGaAsOptimizedMap,
    SCHEDULED_TIME,\0:0:0\,FROM_PREVIOUS_START,
    NIS(CAS_NIS_FULL_POINT_SCAN_SUPER,0,7,295,"SLIT","10X",11,5,500,
    5,-1.00871,2.33456,-7.91578,0,1,0,0,0,0,0,0,-0.0325440,-0.999470,3,FALSE,3581,4,
    0,0,0,0,0,3581,0,0,0.00107801,0,0,0,0,0)
  ),
end;
```

Seqgen

The program SEQGEN is maintained by the OPS people at APL (specifically Karl Whittenburg). The purpose of the program is to create an environment where simple spacecraft cases can be used to lay out a week in spacecraft time that contains many requests from all possible instrument and the OPS team. The output of the program is then translated into the very difficult spacecraft commands.

There are two versions of SEQGEN on our computer. There is the "test" version, which contains all cases and sequences that are being tested. There is the "production" version, which contains the actual cases that are acceptable to the spacecraft and the actual sequences that will be run on the spacecraft.

At Cornell the two areas are /ads01/seqgen/test and /ads01/seqgen/production.

At APL the two areas are on nearmoh5 only and are /project/nearmops/test and /project/nearmops/production.

Getting Started

Lay out of a week of sequences begins with collecting various necessary files. You need:

1) Up-to-date spreadsheet

The spreadsheet is kept up-to-date by Maureen. Deborah adds tracks when a sequence is completed. This also should be on the APL web page and the Cornell Harch web page.

2) The spice file that is on the spacecraft (.spk file for orbit)

The spice files are in /near2/SpiceOps/ops/spk. There is an up-to-date spicespec in /neartools/bin. Charlie Kowal is the OPS person who keeps track of what's on the spacecraft. If we don't have the latest files, telnet to sd-telnet.jhuapl.edu and rlogin to crossroads. Then type: cd /blackhole/navif/navigation/export/curr. Look for the latest .xsp file (or whatever Charlie tells you is the right file). Then type (note you must allow access to your account on crossroads in your .rhosts file on baritone):

```
rcp filename.xsp yourname@baritone.tn.cornell.edu:/near2/SpiceOps/ops/spk
rcp filename.txt yourname@baritone.tn.cornell.edu:/near2/SpiceOps/ops/spk
```

On baritone go to the spk directory and run

```
tobin filename.xsp
```

This will convert it to a binary file (.bsp). Then edit your spicespec file to include this filename.

3) OPS files

OPS sets up the tracking and OPNAV's and creates several files that are available on nearmoh5 in /project/nearmops/production/sasf/ops. I go to that directory and create a tar file for the week or weeks that I'm interested in (e.g. 99102).

```
tar cvf ~/move.tar 99102*
cd ~
```

```
rcp move.tar yourname@baritone.tn.cornell.edu:/ads01/seqgen/production/sasf/ops
```

Go back to baritone to the ops directory and type:

```
tar xvf move.tar
```

NOTE that you must have an up-to-date version of the "fincon" file for the previous week (in this example that would be week 99095). If you don't have it, seqgen will give you GOBS of errors.

4) Seqgen files

Periodically OPS updates the cas and model files (near.satf and near.smf). They keep the latest version on nearmoh5 in a special place just for us, /project/nearmops/production/schedule/cornell. There is a command file in my directory that updates these files (~bellma1/update1.csh). If you get strange tracking errors, try updating these files. Copy over my update file to your directory and type source update1.csh. It takes a long time.

Running Seqgen at Cornell

The version of the SEQGEN program running at Cornell has a few problems. We've found ways around them that are imperfect. But they work.

First in your .cshrc you must have a symbol to invoke SEQGEN (either the test or production versions):

```
alias test 'source /ads01/seqgen/test/seq/bin/near-seq-setup-cu'
alias prod 'source /ads01/seqgen/production/scripts/near-seq-setup-prod-cu'
```

You must also have /heartools/bin in your path.

The file near-seq-setup-prod-cu is the Cornell version of a file that defines the location of all of the files that are necessary to run SEQGEN (note the cu on the end). There is a file in my account on baritone (~mbell/SeqgenV22_5) that defines the size of the screen that you will be working in and other gui related parameters. You can try running SEQGEN- if it doesn't fit your screen, copy my file and run it again). You must have the file in you base directory.

The problems that we have running SEQGEN come in the programs ability to identify the input files. We must explicitly define the input files. The way that we do this is to create dummy files (let's say we're working on 99102):

```
prod      : work in the production mode
gen 99102 MSI_NIS  : create the input files for 99102_MSI_NIS
```

After the SEQGEN screen completes (it will probably give you errors), click and drag QUIT/SAVE. This will create a number of files named 99102_MSI_NIS with an extension. The two that are important are the .env (or environment) and the .sarf (or spacecraft ...?).

The environment file sets up the input and output files. For example:

```
CCSD3ZF0000100000001NJPL3KS0L015$$MARK$$;^M
MISSION_NAME = NEAR;^M
SPACECRAFT_NAME = NEAR;^M
DATA_SET_ID = SEQ_ENVIRONMENT;^M
FILE_NAME = 99102_MSI_NIS.env;^M
APPLICABLE_START_TIME = 1999-102T20:05:00;^M
APPLICABLE_STOP_TIME = 1999-109T20:05:00;^M
PRODUCT_CREATION_TIME = 1999-064T11:25:11;^M
PRODUCER_ID = SEQ;^M
HOST_ID =baritone;^M
CCSD3RE00000$$MARK$$NJPL3IF0M00500000001;^M
PROJECT:    NEAR
SPACECRAFT:  93
START_TIME:  1999-102T20:05:00
CUT_OFF_TIME:  1999-109T20:05:00
TIME_PER_SCREEN:  7T00:00:00
MODEL_PAGE:    7T00:00:00
SEQUENCE_ID:   99102_MSI_NIS
SSF_FORMAT:   FREE
TITLE:        99102_MSI_NIS
GCMD:         UNTIMED
Input files:
SC_MODEL:    $PSS_BASE_PATH/smf/$DELIVERED_93_SMF
CATALOG:    $PSS_BASE_PATH/sarf/$DELIVERED_93_SARF
RULES:      $PSS_BASE_PATH/fmrf/$DELIVERED_93_FMRF
LEGENDS:    $PSS_BASE_PATH/legend/$DELIVERED_93_LEGEND
CONTEXT:    $PSS_BASE_PATH/cvf/$DELIVERED_93_CVF

CLOCK:      $PSS_DATA_PATH/sclk/near.sclk
LIGHTTIME:  $PSS_DATA_PATH/ltf/near.ltf
SEQUENCE:   99102_MSI_NIS.sarf
SEQUENCE:   /ads01/seqgen/production/sarf/ops/99102_OPNAV.sarf
BG_SEQUENCE: /ads01/seqgen/production/sarf/ops/99102_OPS.sarf
Output files:
SEQUENCE:   99102_MSI_NIS.sarf
CONDITIONS: 99102_MSI_NIS.fincon
UPLINK:     99102_MSI_NIS.ssf
EVENTS:    99102_MSI_NIS.pef
RUNLOG:     99102_MSI_NIS.seqgen.log
```

\$\$EOF

The gen program creates this file but there are several problems with it. Here is a corrected version of the same file (the corrections are bold faced):

```
CCSD3ZF0000100000001NJPL3KS0L015$$MARK$$^M
MISSION_NAME = NEAR;^M
SPACECRAFT_NAME = NEAR;^M
DATA_SET_ID = SEQ_ENVIRONMENT;^M
FILE_NAME = 99102_MSI_NIS.env;^M
APPLICABLE_START_TIME = 1999-102T20:05:00;^M
APPLICABLE_STOP_TIME = 1999-109T20:05:00;^M
PRODUCT_CREATION_TIME = 1999-064T11:25:11;^M
PRODUCER_ID = SEQ;^M
HOST_ID =baritone;^M
CCSD3RE00000$$MARK$$NJPL3IF0M00500000001;^M
PROJECT:    NEAR
SPACECRAFT:    93
START_TIME:    1999-102T20:05:00
CUT_OFF_TIME:  1999-109T20:05:00
TIME_PER_SCREEN: 7T00:00:00
MODEL_PAGE:    7T00:00:00
SEQUENCE_ID:   99102_MSI_NIS
SSF_FORMAT:    FREE
TITLE:         99102_MSI_NIS
GCMD:          UNTIMED
Input files:
SC_MODEL:      $PSS_BASE_PATH/smf/$DELIVERED_93_SMF
CATALOG:       $PSS_BASE_PATH/satf/$DELIVERED_93_SATF
RULES:         $PSS_BASE_PATH/fmrf/$DELIVERED_93_FMRF
LEGENDS:       $PSS_BASE_PATH/legend/$DELIVERED_93_LEGEND
CONTEXT:       $PSS_BASE_PATH/cvf/$DELIVERED_93_CVF

CLOCK:         $PSS_DATA_PATH/sclk/near.sclk
LIGHTTIME:     $PSS_DATA_PATH/ltf/near.ltf
SEQUENCE:      99102_MSI_NIS.sasf
BG_SEQUENCE:   /ads01/seqgen/production/sasf/ops/99102_OPS.sasf
CONDITIONS: /ads01/seqgen/production/sasf/ops/99095_OPNAVMERGE.fincon
Output files:
SEQUENCE:      99102_MSI_NIS.sasf
CONDITIONS:    99102_MSI_NIS.fincon
UPLINK:        99102_MSI_NIS.ssf
RUNLOG:        99102_MSI_NIS.seqgen.log
$$EOF
```

Two lines have been removed (SEQUENCES and EVENTS) and one line has been added (CONDITIONS). These are very important changes. The OPNAV sequences line causes that file to be read and included in the 99102_MSI_NIS.sasf file (and once the OPNAVs are included you will get "Duplicate Request" Errors). The EVENTS file can be helpful but is gigantic and causes the ending of the program to take forever. The CONDITIONS file is an OPS file that gives the final conditions of the spacecraft for the preceding week.

DO NOT RUN gen AGAIN!!! If you do, your environment file will be wiped out and you'll have to re-edit it.

Now to invoke the SEQGEN program, use the following line:

```
seqgen -l 99102_MSI_NIS.env
```

Trouble Shooting:

- 1) If you type .sasf instead of .env you get the following error message:
Environment file: Invalid DATA_SET_ID in SFDU header:SPACECRAFT_ACTIVITY_SEQUENCE_SCT

In other words, it's the wrong input file!

2) If you get a lot of errors about TRACKS or other OPS functions, chances are that the fincon line is not in your .env file or that you have a bad fincon. Check your environment file. If the CONDITIONS line is there, then you need to talk to an OPS person (Rick Shelton or Carol Collins).

Running Seqgen at APL

At APL you can run SEQGEN on nearmoh5. I have copied a shell script into my directory that seqgen in production mode. You can copy the file over as well, ~bellma1/approach/gen_prod. The files that gen_prod finds are the files that are in the production areas (not the ones in the directory that you are working):

```
BG_SEQUENCE: /project/nearmops/production/sasf/ops/99102_OPS.sasf
BG_SEQUENCE: /project/nearmops/production/sasf/xgrs/99102_XRS.sasf
BG_SEQUENCE: /project/nearmops/production/sasf/mag/99102_MAG.sasf
BG_SEQUENCE: /project/nearmops/production/sasf/msi_nis/99102_MSI_NIS.sasf
```

I run this to be sure that we are within our downlink allocations (it posts all of the downlink for each instrument and engineering).

You can work on a file there by defining things much like on the Cornell computers. In your .cshrc add the lines:

```
source /project/nearmops/production/schedule/seq/bin/near-seq-setup
setenv XUSERFILESEARCHPATH ~/SeqgenV22_5
```

Then use gen or the seqgen command. You shouldn't need to edit the .env file- their version is supposed to know where all of the conditions files reside.

Coordinating Observations

Common Mistakes to Double Check

Delivering Products

A completed week of observations is due for instrument engineer review 3 weeks in advance during cruise and 4 during approach and orbit. The instrument engineer requires the .ssf file, along with the .sasf file. They are to be delivered to nearmoh5 /project/nearmops/production/sasf/msi_nsi. A final version of the week can be delivered one week later, as long as any modifications are minor. When you deliver, be sure to send email to the following list of people:

```
'Ann Harch' <ANN@cuspid.tn.cornell.edu>
cap21@cornell.edu
"Guo, Yanping" <GuoY1@SPACEMSG.JHUAPL.edu>
hawkins@ulysses.jhuapl.edu
"Izenberg, Noam" <IzenbNR1@SPACEMSG.JHUAPL.edu>
jeff.warren@jhuapl.edu
jkm@tube.jpl.nasa.gov
mbell@baritone.tn.cornell.edu
scott.murchie@jhuapl.edu
wmo@wansor.jpl.nasa.gov
```