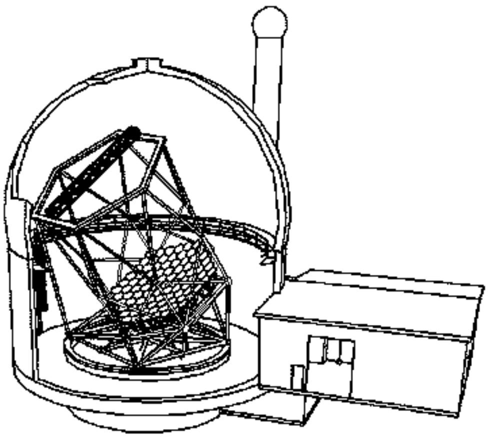
Phase II Planning for the HET

Niall Gaffney



March 6, 1997

Table of Contents

Introduction

Queued observations with the HET require a flexible language with which the user can relay what needs to be done to acquire data that adequately achieves his/her scientific goals. The goal is to allow the user to specify every aspect of the observation that is required to make it and nothing more. Some parameters are required and many are optional. For example, unless the outcome of the observation depends on it, one need not tell the HET operators the telescope azimuth in order for the observation to be made properly or efficiently. However, the PI does need to tell the operator the object's RA and Dec, as that information is crucial to achieving the scientific goals. A few parameters are highly recommended, such as the positions and magnitudes of guide stars. This recommended information will allow the operator to do things such as estimate the atmospheric transmission, or will significantly reduce the overhead time which will be charged against the PI's allocated time. Though these parameters may not directly affect the operator's ability to get the data, they do change the efficiency by which the data are acquired and hence will change how much data are gathered for each project.

What we have tried to do is create a simple scripting language made up of keywords that make up separate building blocks. That is to say one creates blocks of information that fit together in a plan to make the HET make an observation. Different concepts that belong together (such as position and radial velocity of an object) are put into the same block, leaving the related but separate information (such as instrument setups or weather conditions) to different blocks. By assembling the different blocks, one can construct a plan for an observation. The main benefit of this scheme is that the blocks are reusable. If you create a object block (hereafter called an object template) for a particular source, that object template can be used in different plans involving different instruments and even different dates. Further, that object template can be used in future scripts without having to modify it. Hence, I have broken down the HET observing system into different sub–systems that make sense standing alone and can be combined to make a complete observation. These systems are the object, the instrument, the constraints on the observational conditions, the calibration lamp, and the calibration object. These pieces are then assembled in plans which are executed. Further, these plan can have calls to macros, which are smaller subsections of a plan that can be called repeatedly without extra typing or cutting–and–pasting.

First, some definitions:

observing script:	the text file with all of the information needed to execute the observations in a project. The script is made up of a set of templates and plans. All templates for each project granted time on the HET are to be combined into a single file which is the observing script for that project. Collaborative projects that have been grated time by more than one TAC will also share a single observing script.
template:	a template is a block of text that describes a particular subsystem of an observation. They include the object, calibration object, calibration lamp, constraint, instrument, and macro.

plan:	a series of commands that, when executed in a sequence, make the telescope and instruments observe objects, calibrators, and calibrations (e.g., flats).
keyword:	keywords are the single word commands that start each line in the observing script.
parameter:	a word or other set of characters (without white spaces) that convey the knowledge to the keyword call (for example the object's RA, or an exposure time).
object:	a source for which the PI of the plan is privileged to the data. Data for an object will be given only to the PI and all time required acquiring that data will be charged against that PI's allotted time.
calibrator:	a source for which data (and charged time) are to be shared among any projects that require that data (*e.g.* a hot star for removing telluric absorption lines).

Some general rules pertaining to the parameters are:

1) All offsets are in seconds of arc.
2) Proper motions are in seconds of arc per year.
3) All positions are listed in colon separated sexigesimal format (10:22:43.2). Any time can be left off from the right (10:20.3 is 10 hours 20.3 minutes).
4) All exposure times are in seconds.
5) All times (both absolute and relative) are listed in dd/mm/yyyy hh:mm:ss format, but can be abbreviated hh:mm:ss or dd/mm/yyyy. Any time listed without "/" or ":" is assumed to be days. For example December 13, 1997 at 12:45:32 would be 13/12/1997 12:45:32. An offset from that time of 4 days 22 hours 14 minutes would be 4 22:14. An offset of another 3 hours would be 3:00. An offset of 90 minutes would be 0:90. Decimal days are also permitted.
6) When giving date constraints, dates without times are assumed to encompass the entire day listed. So saying we can start on 3/3/1997 means the observation can be done anytime on that day. A date given of 3/3/1997 delta 3:00:00 means that from three hours before the day starts to three hours after the day ends the observations are permitted. A range given of 3/3/1997 to 3/9/1997 10:45 means that the observations can be done as soon as the 3/3/1997 begins and up to 10:45 on 3/9/1997.
7) Bold face keywords are **REQUIRED** for that template to be complete.
8) Plain faced keywords are optional and only needed if required to define an observation or template correctly for the data to be acquired correctly.
9) Different parameters are separated by white spaces.
10) Parameters in <> are **REQUIRED** for keyword syntax to be correct.
11) Parameters in [] are optional.
12) All keywords end in a ":".

13) Keywords are not case sensitive (but all other words such as object names are).
14) White spaces at beginning of a line are ignored (indenting text between a begin and end keyword is recommended!!!).
15) The pound (“#”) character acts as a line comment marker; any characters on a line after # are ignored by the parser and HET operations.
16) The backslash (“\”) character acts as a line continuation character. If it appears as the last character on a line, the next line is considered to be a continuation of the previous line.
17) For any observing script to be complete, it must contain at least a summary template (that contains a PI person template), an object template, an instrument template, and a plan. It may also contain more objects and plans as well as lamps, calibrators, and macros if these are needed to make complete a plan or plans.

Templates

The following sections list the details that may be specified in the different templates. Each is listed with a summary of all commands and what they are for listed in a box. After these definitions, the commands are listed again specifying the syntax and expected information that each keyword will supply. Examples of each keyword syntax are given in `typewriter` font. After the templates are all defined, we give a section with lists of example plans.

Summary Template

This section describes the Summary template. This includes all of the information about the people involved in the project, who to contact, and how. It also should include a short abstract of the science being done, and some description of what is needed from the data (what features, what sort of S/N). This abstract will be used by the resident astronomer to judge whether the data acquired appears to be satisfactory or not. There can be only one summary template per observing script.

Summary template (required)	
begin_summary:	**marks the start of the Summary template**
begin_PI:	**marks the start of the PI information**
end_PI:	**marks the end of the PI information**
begin_coi:	marks the start of the CoI information
end_coi:	marks the end of the CoI information
begin_contact:	marks the start of the contact information
end_contact:	marks the end of the contact information
project_id:	**the main affiliation and unique program id for this project**
project_title:	**the title of this project**
project_username:	**the username for this project's account on the HET data distribution system.**
collab_id:	for collaborations, the affiliation and id of the project as considered by another TAC
begin_abstract:	**some abstract of what is being done and what is needed and why for the Resident Astronomer to use to judge data**
end_abstract:	**marks end of abstract**
num_discretionary:	number of discretionary objects needed to finish project
end_summary:	**marks the end of the summary template**

begin_summary:

This simply marks the beginning of the summary template. There are no arguments.

```
begin_summary:
```

begin_PI:
end_PI:

These two keywords mark the beginning and end of the person template (which is defined in the next section) that gives the identity and pertinent information about the PI. There are no arguments.

```
begin_PI:
            name: Niall Gaffney
            affiliation: UT
            address: RLM 15.308/University of Texas / \
                         Austin, TX 78712
            phone: (512)471-3343
            fax: (512)471-6016
            email: niall@rhea.as.utexas.edu
end_PI:
```

begin_coi:
end_coi

These two keywords mark the beginning and end of the person template (which is defined in the next section) that gives the identity and pertinent information about each of the project's co–investigators. There are no arguments.

```
begin_coi:
            name: Mark Cornell
            affiliation: UT
            address: RLM 15.308/University of Texas / \
                         Austin, TX 78712
            phone: (512)471-3423
            fax: (512)471-6016
            email: cornell@puck.as.utexas.edu
end_coi:
```

begin_contact:
end_contact:

These two keywords mark the beginning and end of the person template (which is defined in the next section) that gives the identity and pertinent information about the contact person for this project if the contact is someone other than the PI. There are no arguments.

```
begin_contact:
            name: Greg Doppmann
            affiliation: UT
            address: RLM 15.308/University of Texas / \
                        Austin, TX 78712
            phone: (512)471-4475
            fax: (512)471-6016
            email: greg@marple.as.utexas.edu
end_contact:
```

project_id: <the affiliation of the project> <the unique number assigned to this project>

This identifies the affiliation and proposal number given by the primary TAC to this project. In collaborative projects, which TAC is the primary one is determined by the investigators. It, combined with the affiliation, uniquely identifies this project. Note that for collaborative projects, this id will be the one used to uniquely identify this project. There are two arguments, the first of which is the affiliation denoted by one of the following abbreviations:

Göttingen	UG
München	UM
Stanford	SU
Penn. State	PS
Texas	UT

```
project_id: UT 96001
```

collab_id: <the affiliation of a collaborative project> <the project number of the collaborative project>

This identifies the affiliation and proposal number given by the collaborating institution's TAC to this project. This only applies if the collaborator is at another HET partner institution. It is to be used to identify what other TAC has allocated time to this project and which project is to be billed. **Note that the TACs for each institution must approve of and allocate time to this project for this keyword to be valid.** There are two arguments, the first of which is the affiliation denoted by one of the following abbreviations:

Göttingen	UG
München	UM
Stanford	SU
Penn. State	PS
Texas	UT

```
collab_id: PS 96004
```

begin_abstract:
end_abstract:

These two keywords mark the beginning and end of the abstract. The text that falls between them is not interpreted by the scheduler, but will be available to the resident astronomer to read when examining the data as it comes in. It is intended to provide the resident astronomer with enough knowledge to make a judgment on the quality of the data relative to the science that it will do.

```
begin_abstract:
     For this project we need to get spectra of the Na D and
     CaII absorption lines in the nuclei of 10 to 30
     galaxies, with S/N of order 100 to get velocity
     dispersion information at two different wavelengths. It
     is critical to get CaII data on as many sources as
     possible and then NaD to explore the effects of
     continuum absorption on the kinematics.
end_abstract:
```

num_discretionary: <number of discretionary objects needed to finish project>

This tells the operations team how many of the discretionary plans need to be completed to mark this project as completed. All unmarked projects or ones marked required **MUST** be completed to mark this project as completed.

```
num_discretionary: 15
```

end_summary:

This marks the end of the summary template. There are no arguments.

```
end_summary:
```

Person Template

This template simply contains the information for each investigator or contact person. It is to be contained within the summary template. See the summary template for the keywords that mark the beginning and end of the different person templates (e.g., the begin_PI: keyword).

“Person” Template (PI, CoI, or contact)	
name:	**the name of the person**
affiliation:	**the affiliation of the person**
email:	**the email address of the person**
address:	**the postal address for this person**
phone:	**the day time phone number for this person**
home_phone:	the home phone number for this person
fax:	the fax number for this person

name: <the name of the person>

This gives the name of this person. There is one argument.

```
name: Niall Gaffney
```

affiliation: <the affiliation of the person>

This gives the affiliation of the person. There is one argument that is to be selected from the following abbreviations:

Göttingen	UG
München	UM
Stanford	SU
Penn. State	PS
Texas	UT

```
affiliation: UT
```

email: <the email address of the person>

This gives the email address of the person. There is one argument.

```
email: niall@rhea.as.utexas.edu
```

phone: <the day time phone number of the person>

This gives the day time (office) phone number of the person. There is one argument.

```
phone: (512)471-3343
```

home_phone: <the home phone number of the person>

This gives the night time (home) phone number of the person. There is one argument

```
home_phone: (512)555-1212
```

address: <the postal address of the person>

This gives the postal address of the person. Use the “/” character to note line breaks and the “\” character to continue the address on another line. There is one argument.

```
address: RLM 15.308/University of Texas at Austin / \
            Austin, TX  78712
```

fax: <the fax number of the person>

This gives the fax number of the person. It has one argument.

```
fax: (512)471-6016
```

Object Template

The next template is the object template. It is used to define parameters associated with objects to be observed. It includes the essential parameters such as position, epoch, proper motion, brightness, and radial velocity. It also contains the position of (at least) two guide stars and the name of the electronic finding chart for this object. Further provisions are made for making offsets from bright objects to the object to be observed, as well as having a positional ephemeris listing (though how one then lists an ephemeris of guide stars is still TBD). Note that no information regarding observations are in this template, it is simply information on how to find the object and how to determine if we have the right one.

object template (required)	
begin_object:	**marks the start of a object listing**
begin_comment:	marks the start of the comment about this object for the operator
end_comment:	marks the end of the comment about this object for the operator
description:	**star, galaxy, HII region, planet, moon…**
acq_descript:	identifies the type of object used in an offset acquisition
RA:	**Right Ascension**
Dec:	**Declination**
Equinox:	**equinox of position**
Epoch:	epoch of position if different from equinox
RA_PM:	proper motion in RA
Dec_PM:	proper motion in Dec
RA_offset:	RA offset from acquisition object to object to be observed
Dec_offset:	Dec offset from acquisition object to object to be observed
begin_ephemeris:	begin an ephemeris table
end_ephemeris:	marks the end of an ephemeris table
a:	semi–major axis of orbit relative to the sun
e:	eccentricity of orbit relative to the sun
I:	inclination of orbit relative to the sun
T:	time of perihelion passage of orbit relative to the sun
P:	period of orbit relative to the sun in years
big_omega:	longitude of ascending node of orbit relative to the sun
little_omega:	argument of periastron of orbit relative to the sun

object template (continued)	
RV:	radial velocity
Flux:	**flux at wavelength of target object**
V:	**apparent V magnitude of acquisition object**
B–V:	**B–V color of acquisition object**
phase_zero:	date of the zero point in a variable’s phase
phase_period:	the period of the variability
phase_pdot:	the unitless rate of change of the period
GS_equinox:	**equinox for all guide stars**
GS1_RA:	**GS #1 Right Ascension**
GS1_Dec:	**GS #1 Declination**
GS1_magnitude:	**GS #1 V magnitude**
GS2_RA	**GS #2 Right Ascension**
GS2_Dec:	**GS #2 Declination**
GS2_magnitude:	**GS #2 V magnitude**
GS3_RA	GS #3 Right Ascension
GS3_Dec:	GS #3 Declination
GS3_magnitude:	GS #3 V magnitude
finding_chart:	**file name of finding chart in a unique format**
end_object:	**marks the end of an object setup**

begin_object: <object name>

This marks the beginning of an object template. It takes a single one–word argument which is the identifier for this object. This identifier will be used in the planning stage to call up this object’s position and other information.

```
begin_object: M82
```

begin_comment:
end_comment:

These mark the beginning and end of a comment that will be passed on to the Resident Astronomer **at the time the object is to be observed**. It is NOT to be used to instruct HET operations on when to schedule the object, but rather is to allow the user to give some details of the observation that cannot be given within the structure of the template. Note that the following example will cost the user valuable observing time, as the PI will be billed for all of the overhead associated with fulfilling the requested test exposure and figuring out the integration time needed. Further, the PI should have specified elsewhere an integration time that will be considered to be the LONGEST time allowed for this observation in order to allocate a long enough time in the schedule.

```
begin_comment:
             This object may be brighter than I anticipated
             due to supernovae. Please take a 1 minute
             integration and scale the time to give a
             signal to noise of 50 at 5400A.
end_comment:
```

description: <object type>

This gives a keyword description of what the object is. It is intended to be used during acquisition to aid in finding the correct object.

```
description: galaxy
```

acq_descript: < acquisition object type>

This gives a keyword description of what the acquisition object is. This is the object from which a “blind” offset will be done to acquire an object that is to faint to acquire directly.

```
acq_descript: star
```

RA: <Right Ascension in sexigesimal format>

This gives the right ascension of the object in hours of time given in sexigesimal form. The fields of this should separated by a “:” character.

```
RA: 9:59:55.2
```

Dec: <Dec in sexigesimal format>

This gives the declination of the object in degrees given in sexigesimal form. The fields of this should separated by a “:” character.

```
Dec: +69:55:00
```

Equinox: <equinox of position>

This lists the equinox system in which the coordinates are given (both individual pairs and ephemeris positions) . Dates 2000 or later are considered to be Julian if no “J” or “B” is specified. Earlier dates are assumed to be Besselian, if no “J” or “B” is specified.

```
Equinox: B1950
```

Epoch: <epoch of position>

This lists the epoch of the coordinates given for this object, if different from the equinox.

```
Epoch: 1950
```

RA_PM: <annual PM in RA in seconds of arc per year>

This gives the proper motion in right ascension in arcseconds per year.

```
RA_PM: 0.01
```

Dec_PM: <annual PM in Dec in seconds of arc per year>

This gives the proper motion in declination in arcseconds per year.

```
Dec_PM: -0.03
```

RA_offset: <offset in RA in seconds of arc to object to be observed>

This gives an offset in right ascension from the primary position to the location of the actual object in arcseconds. It is assumed that the primary location will be an optically identifiable feature such as a star, a galactic nucleus, or other sharp bright feature.

```
RA_offset: 10.2
```

Dec_offset: <offset in Dec in seconds of arc to object to be observed>

This gives an offset in declination from the primary position to the location of the actual object in arcseconds. It is assumed that the primary location will be an optically identifiable feature such as a star, a galactic nucleus, or other sharp bright feature.

```
Dec_offset: -5.3
```

begin_ephemeris:
end_ephemeris:

These mark the beginning and end of a table listing ephemeris positions for a moving object. The ephemeris is listed between them in a tab delineated table of RA, declination, magnitude, and epoch of this position. Note that the equinox of the ephemeris must be given in the object template using the equinox keyword. The ephemeris must be complete enough that a simple linear interpolation of the position between ephemeris points will allow us to find the object.

```
begin_ephemeris:
9:50:22      69:44:23     11.4 1994.43
9:55:23      70:22:43     10.3 1994.52
end_ephemeris:
```

a: <orbital semimajor axis in AU>

This lists the orbital element for the semimajor axis in AU relative to the sun.

```
a: 42.44
```

e: <orbital eccentricity>

This lists the eccentricity of the orbit.

```
e: 0.94
```

I: <orbital inclination in degrees>

This lists the inclination of the orbit in degrees relative to the sun.

```
I: 10.211
```

T: <time of perihelion passage as a Julian date>

Sets the time of a perihelion passage of the orbit as a Julian date.

```
T: 2450423.43244
```

P: <orbital period in years>

Gives the period of the orbit in years.

```
P: 22.4324
```

big_omega: <longitude of ascending node in degrees>

This gives the longitude of the ascending node of the orbit in degrees.

```
big_omega: 94.3242
```

little_omega: <argument of periastron in degrees>

This lists the argument of periastron of the orbit in degrees.

```
little_omega: 42.432
```

RV: <Radial Velocity> <unit>

This gives the radial velocity of the object. It will be needed by the resident astronomer to identify features in any spectrum that has a significant velocity shift relative to the resolution of the spectrometer. Possible units are km/s and Z (redshift).

```
RV: 220 km/s
```

Flux: <wavelength> <feature> <flux> <flux unit>

This gives a flux measurement for the target object within the aperture of the instrument being used at the wavelength of interest. Multiple Flux entries will be needed if an object is to be observed at different wavelengths. The wavelength unit is Angstroms. The feature is either continuum or line (for an emission line flux above the continuum). The flux is then given with its units (either mag for magnitudes, ergs for ergs/(s×cm^2), or Watts for W/m^2 (all are per Å for continuum listings)).

```
Flux: 3500 continuum 19.2 mag
Flux: 5007 line 10e-15 ergs
```

V: <apparent V magnitude>

This gives the apparent V magnitude of the acquisition object, to estimate the integration time needed for acquisition of the object with the CCD camera.

```
V: 19.2
```

B–V: <B–V color>

This gives the observed B–V color of the acquisition object, to estimate the integration time needed for acquisition.

```
B-V: 4.2
```

phase_zero: <Date> [unit]

This is simply the Julian date of the zero point in the variability of the object. Can be either UT date/time, Julian Date, or Modified Julian Date. If given in date format, it is assumed to be a UT date/time. Otherwise user must specify JD or MJD.

```
phase_zero: 2450324.2332 JD
phase_zero: 1/10/1995 10:22
```

phase_period: <some time interval>

This is simply the period of the variation of an object. It is to be given in the time/day format where entries without ":" are days and entries with ":"are hh:mm:ss. One can specify more than 60 minutes in the minutes field, or more than 60 seconds in the seconds field if one wants to give the time interval in those units.

```
phase_period: 14 10:22:14.3
phase_period: 0:3605:0
```

phase_pdot: <the unitless rate of change of the period>

This is the change in the period over time in a unitless format (also known as seconds per second). This is used to calculate the phase of a variable whose period changes with time. The formula used here is

$$T = T_0 + P \times E + 0.5 \times P \times P_{dot} \times E^2.$$

E is the phase of the observation, T_0 is the zero point in time, P is the period and P_{dot} is the change in the period over time.

```
phase_pdot: 1e-2
```

GS_equinox: <equinox for all guide stars>

This lists the equinox of the guide star positions.

```
GS_equinox: J2000
```

GS1_RA: <Right Ascension in sexigesimal format>

This gives the right ascension of the first guide star in sexigesimal format.

```
GS1_RA: 9:54:12
```

GS1_Dec: <Declination in sexigesimal format>

This gives the declination of the first guide star in sexigesimal format.

```
GS1_Dec: 69:54:01
```

GS1_magnitude: <apparent V magnitude>

This gives the apparent V magnitude of the first guide star.

```
GS1_magnitude: 10.2
```

GS2_RA <Right Ascension in sexigesimal format>

This gives the right ascension of the second guide star in sexigesimal format.

```
GS2_RA: 9:56:00
```

GS2_Dec: <Declination in sexigesimal format>

This gives the declination of the second guide star in sexigesimal format.

```
GS2_Dec: 69:50:12
```

GS2_magnitude: <apparent V magnitude>

This gives the apparent V magnitude of the second guide star.

```
GS2_magnitude: 10.2
```

GS3_RA <Right Ascension in sexigesimal format>

This gives the right ascension of the third guide star in sexigesimal format.

```
GS3_RA: 9:54:45
```

GS3_Dec: <Declination in sexigesimal format>

This gives the declination of the third guide star in sexigesimal format.

```
GS3_Dec: 69:55:23
```

GS3_magnitude: <apparent V magnitude>

This gives the apparent V magnitude of the third guide star.

```
GS3_magnitude: 10.2
```

finding_chart: <file name>

This gives the file name of the finding chart for this object. The format of this name should be the unique username of the project (given in the object template after the keyword project_username:) followed by a proposal number identifying the project (given as the second argument in the object template under the keyword project_id:), a period, and a unique number for this object. This file must be a FITS file and the acquisition object should be marked in the fits header with the keywords CRPIX1 (marking the X axis pixel number) and CRPIX2 (marking they Y axis pixel number). More details on FITS headers for these images can be found in the appendix on finding charts. For this example our project_username is `nig` and our proposal number is `96001`.

```
finding_chart: nig96001.002.fits
```

end_object:

This marks the end of the object listing. There are no arguments.

```
end_object:
```

Constraint Template

The following is an introduction to the parameters in the constraint template. This listing is not complete at this time, as we do not know all of the different constraints on the system that can be controlled/specified by the user. None of this is required to make an observation, and it is intended only to give the PI control over aspects of the observations that they may wish to take.

Constraint template (not required)	
begin_constraint:	**starts constraint set**
begin_comment:	marks the start of the comment about the constraints for the operator
end_comment:	marks the end of the comment about the constraints for the operator
azimuth :	azimuth of telescope
moon:	acceptable phases of the moon
east:	flexible azimuth but make tube east of meridian
west:	flexible azimuth but make tube west of meridian
free:	set the telescope to be free to point either east or west
min_throughput:	minimum percentage of vignetting allowable
min_net_throughput:	minimum percentage of vignetting allowed, averaged over track
max_seeing:	maximum seeing size allowed to begin observations
min_transmission:	minimum percentage transmission allowed to begin observation
min_mirrors:	the fewest mirrors required to make an observation
max_sky:	the maximum allowed sky brightness
end_constraint:	**ends constraint set**

begin_constraint: <name of constraint setup>

This simply marks the beginning of a constraint template, and gives it a name. Its one argument is a single–word name for this constraint setup.

```
begin_constraint: M82_t1
```

begin_comment:
end_comment:

These mark the beginning and end of a comment that will be passed on to the Resident Astronomer **at the time the object is to be observed**. It is NOT to be used to instruct HET operations on when to schedule the object, but rather is to allow the user to give some details of the observation that cannot be given within the structure of the template. Note that for the following example, the Resident Astronomer can only tell if this constraint is met by going to the field and looking. If the constraint were not met, the PI would be charged for the time required to determine this.

```
begin_comment:
            Please make sure that the asteroid is not
            within 15 arc seconds of a bright star.
end_comment:
```

azimuth: <azimuth in degrees> [delta <range in degrees>]

This sets the azimuth that the HET must be and gives a possible range around this azimuth, all in degrees. For example if the HET azimuth had to be at 10±5 degrees, the command would be:

```
azimuth: 10 delta 5
```

moon:

This command allows the user to specify which phases of the moon are acceptable. Possibilities are full, gray, and dark.

```
moon: dark
```

east:

This command simply specifies that the telescope must be east of the meridian.

```
east:
```

west:

This command simply specifies that the telescope must be east of the meridian.

```
west:
```

free:

This command simply undoes any east or west command, permitting the telescope to observe objects either rising or setting. There are no parameters.

```
free:
```

min_throughput: <percentage of vignetting allowable>

This allows the PI to set a maximum amount of aperture vignetting during a track. For example, if at no point during an observation should the throughput fall below 85%, one would command:

```
min_throughput: 85
```

min_net_throughput: <percentage of vignetting allowed, averaged over track>

This allows the PI to set a maximum amount of vignetting, averaged over the entire track. For example, if you wanted to have an average throughput no less than 88%, one would command:

```
min_net_throughput: 88
```

max_seeing: <seeing disk FWHM in seconds of arc>

This allows the user to set a limit to how large a seeing disk is acceptable for these observations. The seeing disk is specified by the FWHM in seconds of arc of a Gaussian fit to a stellar profile. To set a maximum FWHM seeing disk of 2.5 arcseconds, one would command:

```
max_seeing: 2.5
```

min_transmission: <percentage of transmission>

This allows the user to set a limit on how low the atmospheric transmission can fall before the conditions are unacceptable for the observations. It is given as a percentage of the transmission. Note that the transmission down the fiber is a convolution of this atmospheric transmission and the coupling function of the fiber/slit to the seeing disk.

```
min_transmission: 85
```

max_sky: <the sky brightness>

This allows the user to specify the maximum sky brightness under which observations can be made. This can either be dark, gray, or bright, or a user specified surface brightness in magnitudes per square arcsecond at V.

```
max_sky: dark
max_sky: 19.2
```

min_mirrors: <number of mirrors>

This allows the user to specify how many mirrors must be present to make the observation.

```
min_mirrors: 42
```

end_constraint:

This simply marks the end of a constraint template. There are no arguments.

Instrument Template

This is an outline of the instrument template. An instrument template is required in an observing script, else we will not know how to observe the objects you specify. For the UFOE we have set aside entries for the Fiber Feed, which grating will be used, and the wavelength center of the array. The latter may or may not be a valid variable. This section will be expanded as more information from the instrument scientists becomes available.

Instrument template (required)

begin_instrument:	**begins instrument setup**
instrument:	name of instrument
begin_comment:	marks the start of the comment about the instrument for the operator
end_comment:	marks the end of the comment about the instrument for the operator
setup:	**use predefined setup**
fiber:	which fiber to use
grating:	which grating to use
lambda:	what wavelength to center the ccd on
pa:	what position angle to orient the fiber array
firstcol:	first CCD column number to be read out
lastcol:	last CCD column number to be read out
firstrow:	first CCD row number to be read out
lastrow:	last CCD row number to be read out
colbin:	CCD column binning factor
rowbin:	CCD row binning factor
end_instrument:	**ends instrument setup**

begin_instrument: <name of setup>

This marks the beginning of an instrument template. The single–word parameter names this setup for use in the planning stage.

```
begin_instrument: M82_red
```

instrument: <name of instrument>

This selects which instrument is to be used. Currently this is only the ufoe.

```
instrument: ufoe
```

begin_comment:
end_comment:

These mark the beginning and end of a comment that will be passed on to the Resident Astronomer **at the time the object is to be observed**. It is NOT to be used to instruct HET operations on when to schedule the object, but rather is to allow the user to give some details of the observation that cannot be given within the structure of the template.

```
begin_comment:
             Please make sure that the iodine cell is in.
end_comment:
```

setup: <predefined setup name>

This selects one of the pre–defined setups for the instrument selected. Currently the following setups are defined for the UFOE mounted on the SSAC. Note, this overrides the instrument, fiber, grating, and lambda keywords. Also note this is the only way of specifying the UFOE setups in its SSAC mode.

setup name	instrument	grating	fiber size	resolution
ufoe_ssac_r2100	ufoe	150 l/mm	2"	2100
ufoe_ssac_r3400	ufoe	150 l/mm	2"	3400
ufoe_ssac_r6500	ufoe	79 l/mm	2"	6500
ufoe_ssac_r10000	ufoe	79 l/mm	2"	10000
ufoe_ssac_r13900	ufoe	79 l/mm	2"	13900

```
setup: ufoe_ssac_r13900
```

lambda: <wavelength> <units>

This is an example of a keyword to uniquely define an instrument setup. Note the UFOE lambda is fixed and cannot be specified.

```
lambda: 8500 A
```

fiber: <number of fiber feed>

This determines which of the three fiber feeds to use. For the UFOE with the SSAC there is only one option: 2.0.

fiber: 2.0

grating: <which grating to use>

This selects which of the two gratings will be used. For the UFOE it is ether the 79 l/mm R2 echelle (denoted 79 or A), or the 150 l/mm R1 echellette (denoted 150 or B).

```
grating: 79
```

pa: <position angle to orient the fiber array in degrees>

This simply is the PA the fiber must be at in degrees. Because of the limited motion of the tracker, the angle must be from 0 to 180 degrees.

```
pa: 65
```

firstcol: <column number>

This specifies the first column of the CCD to be stored when only a sub array of the entire chip is to be read out. Its single argument is the device coordinate (or column number) of the first part of the sub array to be saved.

```
firstcol: 45
```

lastcol: <column number>

This specifies the last column of the CCD to be stored when only a sub array of the entire chip is to be read out. Its single argument is the device coordinate (or column number) of the last part of the sub array to be saved.

```
lastcol: 950
```

firstrow: <row number>

This specifies the first row of the CCD to be stored when only a sub array of the entire chip is to be read out. Its single argument is the device coordinate (or row number) of the first row to be saved.

```
firstrow: 12
```

lastrow: <row number>

This specifies the last row of the CCD to be stored when only a sub array of the entire chip is to be read out. Its single argument is the device coordinate (or row number) of the last row to be saved.

```
lastrow: 845
```

colbin: <column binning>

This specifies the binning factor for the columns of the CCD. Its single argument is how many CCD pixels are to be coadded in each column and saved as a pixel in the image array.

```
colbin: 3
```

rowbin: <row binning>

This specifies the binning factor for the rows of the CCD. Its single argument is how many CCD pixels are to be coadded in each column and saved as a pixel in the image array.

```
rowbin: 4
```

end_instrument:

This marks the end of the instrument template. There are no parameters.

```
end_instrument:
```

Calibrator Template

This is a calibrator template. It is just like an object template, except if there are overlapping calibration requests from different projects or plans, one calibrator can be shared amongst those observations. Note variability is not an option in calibrators. If for some reason you need a calibration object at a particular phase, make it an object.

calibrator template (not required)	
begin_calibrator:	**begins calibrator template and gives it a name**
begin_comment:	marks the start of the comment about this calibrator for the operator
end_comment:	marks the end of the comment about this calibrator for the operator
description:	**star, galaxy, HII region, planet, moon…**
acq_descript:	identifies the type of object used in an offset acquisition
Catalog: †	**pick from this catalog and give some exposure time otherwise give positions, etc.**
RA:	**RA in sexigesimal format**
Dec:	**Dec in sexigesimal format**
Equinox:	**equinox of position**
Epoch:	epoch of position
PM_RA:	proper motion in RA
PM_Dec:	proper motion in Dec
Offset_RA:	offset in RA to object to be observed
Offset_Dec:	offset in Dec to object to be observed
Ephemeris:	begin an ephemeris table, followed by a tab delineated list of RA, Dec, magnitude, and epoch/date and time for each point
a:	semi–major axis of orbit relative to the sun in AU
e:	eccentricity of orbit relative to the sun (unitless)
I:	inclination of orbit relative to the sun in degrees
T:	time of perihelion passage of orbit relative to the sun as a Julian date
P:	Period of orbit relative to the sun in years
big_omega:	longitude of ascending node of orbit relative to the sun in degrees
little_omega:	argument of periastron of orbit relative to the sun in degrees
V:	**apparent V magnitude**
B–V:	**B–V color**

calibrator template (continued)	
GS_equinox:	**equinox for all guide star positions**
GS1_RA:	**GS #1 RA**
GS1_Dec:	**GS #1 Dec**
GS1_magnitude:	**GS #1 V magnitude**
GS2_RA	**GS #2 RA**
GS2_Dec:	**GS #2 Dec**
GS2_magnitude:	**GS #2 V magnitude**
GS3_RA	GS #3 RA
GS3_Dec:	GS #3 Dec
GS3_magnitude:	GS #3 V magnitude
finding_chart:	**unique file name**
end_calibrator:	**ends calibrator setup**

† If Catalog is used, no other parameters are required.

begin_calibrator: <calibrator name>

This marks the beginning of an calibrator template. It takes a single one–word argument which is the identifier for this calibrator. This identifier will be used in the planning stage to call up this calibrator's position and other information.

```
begin_calibrator: BS4423
```

begin_comment:
end_comment:

These mark the beginning and end of a comment that will be passed on to the Resident Astronomer **at the time the object is to be observed**. It is NOT to be used to instruct HET operations on when to schedule the object, but rather is to allow the user to give some details of the observation that cannot be given within the structure of the template.

```
begin_comment:
            Please make sure that the counts per pixel
            in the frame is no less than 5000 in the blue.
end_comment:
```

Catalog: <catalog name>

This tells the program to select an object from a catalog. Currently there are no catalogs, but I could foresee catalogs of blue rapid rotating stars, radial velocity standards, white dwarfs, and other potentially useful common calibrators. Note that if Catalog is specified, no other parameters are needed. Also note that at this time there are no catalogs available. We hope to make them available in the future once details have been worked out.

```
Catalog: Blue_star
```

description: <calibrator type>

This gives a keyword description of what the calibrator is. It is intended to be used during acquisition to aid in finding the correct calibrator.

```
description: Blue star
```

acq_descript: < acquisition object type>

This gives a keyword description of what the acquisition object is. This is the object from which a “blind” offset will be done to acquire an object that is to faint to acquire directly.

```
acq_descript: star
```

RA: <RA in sexigesimal format>

This gives the right ascension in hours of time of the calibrator in sexigesimal form. The fields of this should separated by a “:” character.

```
RA: 11:59:55.2
```

Dec: <Dec in sexigesimal format>

This gives the declination in degrees of the calibrator in sexigesimal form. The fields of this should separated by a “:” character.

```
Dec: 69:55:00
```

Equinox: <equinox of position>

This lists the equinox system in which the coordinates are given (both individual pairs and ephemeris positions) . Dates 2000 or later are considered to be Julian if no “J” or “B” is specified. Earlier dates are assumed to be Besselian, if no “J” or “B” is specified.

```
Equinox: B1950
```

Epoch: <epoch of position>

This lists the epoch of the coordinates given for this object, if different from the equinox.

```
Epoch: 1950
```

RA_PM: <annual PM in RA in seconds of arc per year>

This gives the proper motion in right ascension in arcseconds per year.

```
RA_PM: 0.01
```

Dec_PM: <annual PM in Dec in seconds of arc per year>

This gives the proper motion in declination in arcseconds per year.

```
Dec_PM: -0.03
```

RA_offset: <offset in RA in seconds of arc to object to be observed>

This gives an offset in right ascension from the primary position to the location of the actual object in arcseconds. It is assumed that the primary location will be an optically identifiable feature such as a star, a galactic nucleus, or other sharp bright feature.

```
RA_offset: 10.2
```

Dec_offset: <offset in Dec in seconds of arc to object to be observed>

This gives an offset in declination from the primary position to the location of the actual object in arcseconds. It is assumed that the primary location will be an optically identifiable feature such as a star, a galactic nucleus, or other sharp bright feature.

```
Dec_offset: -5.3
```

begin_ephemeris:
end_ephemeris:

These mark the beginning and end of a table listing ephemeris positions for a moving object. The ephemeris is listed between them in a tab delineated table of RA, declination, magnitude, and epoch of this position. Note that the equinox of the ephemeris must be given in the calibration template using the equinox keyword. The ephemeris must be complete enough that a simple linear interpolation of the position between ephemeris points will allow us to find the calibrator.

```
begin_ephemeris:
9:50:22     69:44:23    11.4 1994.43
9:55:23     70:22:43    10.3 1994.52
end_ephemeris:
```

a: <orbital semimajor axis in AU>

This lists the orbital element for the semimajor axis in AU relative to the sun.

```
a: 42.44
```

e: <orbital eccentricity>

This lists the eccentricity of the orbit.

```
e: 0.94
```

I: <orbital inclination in degrees>

This lists the inclination of the orbit in degrees relative to the sun.

```
I: 10.211
```

T: <time of perihelion passage as a Julian date>

Sets the time of a perihelion passage of the orbit as a Julian date.

```
T: 2455423.43244
```

P: <orbital period in years>

Gives the period of the orbit in years.

```
P: 22.4324
```

big_omega: <longitude of ascending node in degrees>

This gives the longitude of the ascending node of the orbit in degrees.

```
big_omega: 94.3242
```

little_omega: <argument of periastron in degrees>

This lists the argument of periastron of the orbit in degrees.

```
little_omega: 42.432
```

Flux: <wavelength> <feature> <flux> <flux unit>

This gives a flux measurement for the calibrator within the aperture of the instrument being used at the wavelength of interest. Multiple Flux entries will be needed if an calibrator is to be observed at different wavelengths. The wavelength unit is Angstroms. The feature is either continuum or line (for an emission line flux above the continuum). The flux is then given with its units (either mag for magnitudes, ergs for ergs/(s×cm^2), or Watts for W/m^2 (all are per Å for continuum listings)).

```
Flux: 3500 continuum 19.2 mag
Flux: 5007 line 10e-15 ergs
```

V: <apparent V magnitude>

This gives the apparent V magnitude of the acquisition object, to estimate the integration time needed for acquisition.

```
V: 4.2
```

B–V: <B–V color>

This gives the observed B–V color of the acquisition object, to estimate the integration time needed for acquisition.

```
B-V: -0.3
```

GS_equinox: <equinox for all guide stars>

This lists the equinox of the guide star positions.

```
GS_equinox: J2000
```

GS1_RA: <RA in sexigesimal format>

This gives the right ascension of the first guide star in sexigesimal format.

```
GS1_RA: 11:54:12
```

GS1_Dec: <Dec in sexigesimal format>

This gives the declination of the first guide star in sexigesimal format.

```
GS1_Dec: 69:54:01
```

GS1_magnitude: <apparent V magnitude>

This gives the magnitude of the first guide star.

```
GS1_magnitude: 10.4
```

GS2_RA <RA in sexigesimal format>

This gives the right ascension of the second guide star in sexigesimal format.

```
GS2_RA: 11:56:00
```

GS2_Dec: <Dec in sexigesimal format>

This gives the declination of the second guide star in sexigesimal format.

```
GS2_Dec: 69:50:12
```

GS2_magnitude: <apparent V magnitude>

This gives the magnitude of the second guide star.

```
GS2_magnitude: 10.4
```

GS3_RA <RA in sexigesimal format>

This gives the right ascension of the third guide star in sexigesimal format.

```
GS3_RA: 11:54:45
```

GS3_Dec: <Dec in sexigesimal format>

This gives the declination of the third guide star in sexigesimal format.

```
GS3_Dec: 69:55:23
```

GS3_magnitude: <apparent V magnitude>

This gives the magnitude of the third guide star.

```
GS3_magnitude: 10.4
```

finding_chart: <file name>

This gives the filename of the finding chart for this calibrator. The format is created in the same fashion as for the object's finding chart given above.

```
finding_chart: nig96001.003.fits
```

end_calibrator:

This marks the end of the calibrator listing. There are no arguments.

```
end_calibrator:
```

Lamp Template

The lamp template is used to specify parameters associated with calibration lamps (flats, arc lamps). Currently, we do not have specifics on the different parameters. These are just guesses. However, you will have to specify the lamp name. All other parameters are TBD.

lamp template (not required)	
begin_lamp:	**name of lamp setup**
lamp:	**standard name of lamp**
begin_comment:	marks the start of the comment about this lamp for the operator
end_comment:	marks the end of the comment about this lamp for the operator
end_lamp:	

begin_lamp: <name of setup>

This starts a lamp template. Its one argument names this particular lamp template to be used in the planning stages.

```
begin_lamp: Ar3
```

begin_comment:
end_comment:

These mark the beginning and end of a comment that will be passed on to the Resident Astronomer **at the time the object is to be observed**. It is NOT to be used to instruct HET operations on when to schedule the object, but rather is to allow the user to give some details of the observation that cannot be given within the structure of the template.

```
begin_comment:
             No less than 2500 counts please.
end_comment:
```

lamp: <lamp name>

This parameter identifies which lamp should be turned on (Flat or wavelength calibrator).

```
lamp: Ar
```

end_lamp:

This marks the end of a lamp template.

```
end_lamp:
```

Plan Templates

The plans are what actually get executed. Each plan is an independent observation that does not rely on another observation or calibration to make it complete. However, the sum of these observations make up the entire project. Each plan can be scheduled and completed independently. The basics of a plan consist of “do” commands, which are sequenced in real time (e.g., do an object and then do a lamp right after you finish the object) and “schedule” commands, which sequence at some point in the future (e.g., do this object and then at some point schedule an observation of this calibrator). By scheduling rather than doing anything but the objects, one may share calibrations and other frames within a project, or with other projects done in the same night. This will cut down on the amount of time billed for calibration. Dates and times of observations are also specified in the plan. Each plan gets assigned a priority by the PI to resolve which of these observations is more important to get first. For each object, a set of default flats, darks, and bias frames will be taken unless the user specifies there to be specific constraints put on these frames (such as number of flat fields to be taken, or that they be taken after each exposure). Finally, for flats, and lamps, a “–” in place of an integration time allows the resident astronomer to select an exposure time that will give a standard flat or lamp exposure.

plan templates (required)	
begin_plan:	**marks the beginning of one plan**
begin_comment:	marks the start of the comment about this plan for the operator
end_comment:	marks the end of the comment about this plan for the operator
priority:	sets priority of this plan
class:	either required or discretionary class observation
utdate_start:	start macro on this day/time
utdate_range:	earliest date and time to latest date and time
utdate_earliest:	earliest date plan can be started
utdate_latest:	latest date plan must be completed by
phase_start:	the phase at which to start the plan
lst_start:	LST at which to start plan
ha_start:	hour angle at which to start plan
move:	move the telescope in RA and Dec
constraint:	use these constraints
instrument:	**set the instrument up like this**
do–object:	**take an object frame**
do–flat:	take a flat frame

plan template (continued)

do–lamp:	take a lamp frame
do–dark:	take a dark frame
do–bias:	take a bias frame
do–calibrator:	take a calibrator frame
do–wait:	a real time wait for the telescope
do–macro:	execute a macro command
schedule–object:	schedule another object sometime tonight
schedule–flat:	schedule a flat for sometime tonight
schedule–lamp:	schedule a lamp for sometime tonight with the current instrument
schedule–dark:	schedule a dark frame for sometime tonight
schedule–bias:	schedule a bias for sometime tonight
schedule–calibrator:	schedule a calibrator for sometime tonight
schedule–wait:	a wait time into which other things may be scheduled
repeat:	repeats the last command
begin_set:	begins a set of observations within a plan
end_set:	ends a set of observations within a plan
begin_loop:	begins a loop of commands to be repeated
end_loop:	marks the end of the loop
end_plan:	**ends plan**

begin_plan: <name of plan>

This marks the beginning of a plan. Its argument is the name of the plan, which is used purely to identify which plan is which.

```
begin_plan: M82_CaII
```

begin_comment:
end_comment:

These mark the beginning and end of a comment that will be passed on to the Resident Astronomer **at the time the object is to be observed**. It is NOT to be used to instruct HET operations on when to schedule the object, but rather is to allow the user to give some details of the observation that cannot be given within the structure of the template.

```
begin_comment:
            This plan requires excellent water vapor
            cancellations.  Please be extra careful
            getting the blue standard star in the
            same conditions as the galaxy.
end_comment:
```

priority: <priority of this plan>

This sets the priority of this plan relative to the others in this project. In this scheme, 1 is the highest priority with larger integers indicating the decline in priority of plans. If priority is not given, it is assumed to be equal to the **LOWEST** priority given in entire project. If no priorities are in the project, then all objects have equal priority.

```
            priority: 1
```

class: <either required or discretionary class observation>

This describes the plan as being either required to complete the project, or discretionary (one of a list of plans that must be completed to fulfill the quota set in the summary template). **If this keyword is not listed, the plan is assumed to be required.**

```
class: discretionary
```

utdate_start: <start macro on this day/time> [delta <flexibility of time>]

This sets the date and time when this plan needs to be executed, and allows for a range of times. The first parameter is the date and time requested and the second optional parameter gives the window of opportunity for this. For example, if I needed to do something at 10:22:45 UT on May 1, 1996 and I had a margin of ± 1 hour around this time when the observation could be started, I would say

```
utdate_start: 01/05/1996 10:22:45 delta 1:00:00
```

utdate_range: <earliest date and time> to <latest date and time>

This gives the range of dates during which the observations can be done. For example, if a plan must be done between the 1st of April 1996 at 4:23 and the 3rd of May 1996 with no specific time, the entry would be

```
utdate_range: 1/4/1996 4:23 to 3/5/1996
```

utdate_earliest: <earliest date plan can be started>
utdate_latest: <latest date plan must be completed by>

> These two simply allow the user to specify when a plan must completed by or on what date at the earliest a plan can be started. They can be used together to specify a range of dates (though utdate_range will do this in one line) or to simply specify one limiting date.

```
utdate_earliest: 31/3/1996
utdate_latest: 9/4/1996
```

phase_start: <Object name> <phase> [delta <flexibility of phase>]

> This allows the user to define a time based on the phase information given in an object. The name is that of a predefined object and the phase is a real number from 0 to 1.

```
phase_start: RRSco .1 delta .05
```

lst_start: <time in sexigesimal format> [delta <flexibility of time>]

> This allows one to set at what LST the plan is executed. The parameters are the same as the utdate_start command: a time and a range around that time. Note that there is no date information in this command. For example, if I needed these observations to start at an LST of 10:22:14 and had a 15 minute window of opportunity in which the plan could be started, I would say

```
lst_start: 10:22:14 delta 00:15:00
```

ha_start: <hour angle in sexigesimal format> [delta <flexibility in hour angle>]

> This allows one to set at what hour angle the plan is executed. The parameters are the same as the utdate_start command: a time and a range around that time. Note that there is no date information in this command. For example, if I needed these observations to start at an hour angle of 1:22:14 and had a 15 minute window of opportunity in which the plan could be started, I would say

```
ha_start: 1:22:14 range 00:15:00
```

move: <move in RA in seconds of arc> <move in declination in seconds of arc>

> This allows the user to move the telescope during a plan, either for mapping of a region or for nodding off to sky.

```
move: 5.4 8.2
```

constraint: <constraint template name>

From this point on (until another constraint command is issued) use these parameters to describe the telescope's setup and the observing conditions under which observations can be made. The parameter is the name of a constraint template defined earlier.

```
constraint: M82_t1
```

instrument: <instrument template name>

At this point in the plan, configure the instrument according to this instrument template. The one parameter is the name of the instrument template defined earlier (or possible standard instrument setups when they become available).

```
instrument: M82_CaII
```

do–object: <name> <integration time> <scaled/fixed> [number of frames]

This makes the plan do an object exposure at this point in time. There are four parameters, the name of the object as defined earlier in the object template, the exposure time in seconds, and a flag to say that this time should be scaled for vignetting/transmission, or if it should be fixed to the time given. The final argument allows the user to specify how many frames are to be taken if more than one is required. For example, if I wanted to do two 1000 second exposures on M82 (which I defined earlier in this file) and this exposure was to be scaled to remove the effects of vignetting and transmission from the resultant signal–to–noise, I would say:

```
do-object: M82 1000 scaled 2
```

do–flat: <integration time> [number of frames]

This causes a flat exposure to be done at the current time. The first parameter is used to specify how long of an exposure is needed. If a "–" is given for an exposure time, the time is left to the resident astronomer to determine. The second argument allows the user to specify how many frames are to be taken if more than one are required. Here, I show how to request 5 sixty second flats.

```
do-flat: 60 5
```

do–lamp: <lamp setup name> <integration time> [number of frames]

This causes a lamp exposure to be done at the current time. The first two parameters are the name of the lamp template to be used, and the fixed exposure time (there is no vignetting for a lamp). If a "–" is given for the exposure time, the time is left to the resident astronomer to determine. The third argument allows the user to specify how many frames are to be taken if more than one are required. For example, if I wanted to do a single AR3 lamp (defined previously) for 600 seconds I would say:

```
do-lamp: AR3 600 1
```

do–dark: <integration time> [number of frames]

This causes a dark exposure to be done at the current time. The first parameter is the fixed exposure time. The second argument allows the user to specify how many frames are to be taken if more than one are required. For example, if I wanted to do two darks for 3600 seconds I would say:

```
do-dark: 3600 2
```

do–bias: [number of frames]

This command simply takes a bias frame at the current time. The single argument allows the user to specify how many frames are to be taken if more than one are required.

```
do-bias: 5
```

do–calibrator: <calibrator setup name> <integration time> <scaled/fixed> [number of frames]

This command takes a calibration frame at the current time. The parameters are the same format as the do–object command.

```
do-calibrator: BS4423 600 fixed 2
```

do–wait: <time>

This does a real time wait for a specific amount of time (with nothing scheduled during the wait). It should be used only when absolutely needed, as this wait time is billed the same as time spent doing any other setup. The single parameter of this call is a time in a sexigesimal format. For example, if I had to wait 15 minutes, I would say

```
do-wait: 00:15:00
```

do–macro: <macro name> <variable 1> <variable 2>…

This executes a predefined macro, passing the variables defined in the macro to the macro. For example, to execute the macro called M82_macro1 which has two passed variables (see the example of begin_macro listed below) I would call

```
do-macro: M82_macro1 M82 3600
```

schedule–object: <object setup name> <integration time> <scaled/fixed> [number of frames]

This the same as a do–object except this object can be scheduled any time in the night in which this plan is executed.

```
schedule-object: M82 2400 fixed 2
```

Plan Template

schedule–flat: <time> <number of frames>

This is the same as do–flat except that it can be scheduled any time during the night or even the next morning. If a “–” is given for the exposure time, the time is left to the resident astronomer to determine. If this is not specified in a plan, then a standard set of flats will be taken the next day. Here I show how to ask for five flats taken with the integration time left to the Resident Astronomer to fill in.

```
schedule-flat: - 5
```

schedule–lamp: <lamp setup name> <time> [number of frames]

This is the same as a do–lamp command except it can be scheduled any time during the night or even the next morning. If a “–” is given for the exposure time, the time is left to the resident astronomer to determine.

```
schedule-lamp: AR3 - 3
```

schedule–dark: <time> [number of frames]

This is the same as a do–dark command except it can be scheduled any time during the night or even the next morning.

```
schedule-dark: 2400 5
```

schedule–bias: [number of frames]

This is the same as a do–bias command except it can be scheduled any time during the night or even the next morning.

```
schedule-bias: 5
```

schedule–calibrator: <calibrator setup name> <time> <scaled/fixed> [number of frames]

This is the same as a do–calibrator command except it can be scheduled any time during the night. Here, as an example, I ask for a single calibration frame of 600 seconds (note that without specifying the number of frames, I am requesting a single frame).

```
schedule-calibrator: BS_4423 600 fixed
```

schedule–wait: <time period> <range of time period>

This inserts a wait into the plan into which other observations can be scheduled. It is intended to make delays such as “wait two or three days and do this again”. In this case, if other observations can be scheduled in between these observations they will be and the time is not billed against the PI’s allotted time. For example, if I wanted to wait a day ± an hour and then continue with my plan, I would say

```
schedule-wait: 1 delta 01:00:00
```

repeat: [number of times to repeat]

This command repeats the last command issued. The optional parameter is the number of times you need it repeated (defaults to 1 if it is not given).

```
repeat: 5
```

begin_set:
end_set:

A set is a subset of commands that must be executed sequentially in a plan. The set needs to be scheduled all together, but only makes up a part of the plan. Simply using individual `schedule-object` and `schedule-lamp` will not accomplish such an end, as the `schedule` commands allow the scheduler to do them at any time rather than in a sequence. Thus, each set can be executed at any time, but the commands in the set are executed sequentially at that single time. For example, if you had to observe an object and then take two flats right after that object, while during another part of the night, you needed to do another object and take two different flats, one would say:

```
begin_set:
                do-object: o1 3600 scaled
                do-lamp: l2 60
                repeat: 1
end_set:
begin_set:
                do-object: o2 2600 scaled
                do-lamp: l2 60
                repeat: 1
end_set:
```

Note that the same can be done by creating macros and scheduling them. This simply allows the user to do this without having to define macros for each unique set.

begin_loop: <number of times to loop>
end_loop:

This is a simple looping command to execute a series of commands repetitively. Its parameter is simply the number of times to repeat.

```
begin_loop: 5
                commands go in here
end_loop:
```

end_plan:

This marks the end of a plan. It has no parameters.

```
end_plan:
```

Macros

A macro template is used to define a series of commands that will be called many times. Variables that must be passed to the macro need to be named.

macro template	
begin_macro:	**define a macro to be used on many objects**
	uses same syntax as plan
end_macro:	**marks the end of the macro**

begin_macro: <macro name> [first variable id] [second variable id]…

This allows the user to create an observing macro to be executed with different objects. The first parameter names the macro. All subsequent parameters are "variables" that can be set when the macro is called. These are set aside from normal words by preceding them with a "$" character. For example, if one wanted to make a macro named M82_macro1 that had two variables passed to it, $object and $itime, one would start it with

```
begin_macro: M82_macro1 $object $itime
```

commands within a macro

Commands that are used in a macro are the same as in the plan. See that section for a listing of these commands. All keywords can have variables passed to them, or can have their parameters set. For example, a do–object command being passed the "it" variable that contains the object name with a fixed exposure time of 1000 seconds would be

```
do-object $it 1000 fixed
```

end_macro:

This marks the end of a macro template. There are no parameters.

```
end_macro:
```

Detailed examples of writing a macro with variables will be outlined in the examples section below.

Example Plans

A Simple Plan: One object and the UFOE

I have an object and I want to take a UFOE spectrum of my object. It is called N2 and is at a position of 14:22:33.2 –4:10:19 B1950. It has a V magnitude of 14.4 and B–V of 0.4. I don’t need any calibrators. I need S/N of 100 so, using the sn tool, I estimate that I need an unscaled exposure time of 2400 seconds. Because of cosmic ray events, I decided that I want to split this time into two 1200 second observations. I have found guide stars at 14:22:38 –4:11:32 (mag 14.9) and 14:22:25 –4:10:44 (mag 10.4) (epochs of B1950) using the guide star finding program. The unique ID of my project nig96001 and I have a FITS file finding chart of the object which, because it is the first finding chart, I have called nig96001.001.fits (see later section on finding charts). I don’t have any constraints on how the telescope, weather, or sky conditions must be. I will be satisfied with standard flats and wavelength calibration frames. First, I write a summary template with all my personal information:

```
begin_summary:
     begin_PI:
        name: Niall Gaffney
        affiliation: UT
        email: niall@rhea.as.utexas.edu
        phone: (512)471-3343
        address: RLM 15.308/University of Texas/ \
                  Austin, TX  78712
     end_PI:
     project_username: nig
     project_id: UT 96001
     project_title: Big time Astrophysics
     begin_abstract:
        This is just some object I want a S/N of 100 on to
        measure the flatness of the continuum from this kind
        of object without any calibrations.
     end_abstract:
end_summary:
```

Then I write an object template that has all the information needed to find the object and its guide stars.

```
begin_object: N2
     description: star
     RA: 14:22:33.2
     Dec: 64:10:19
     Equinox: B1950
     Flux: 5500 continuum 14.4 mag
     V: 14.4
     B-V: 0.4
     GS_equinox: B1950
     GS1_RA: 14:22:38
     GS1_Dec: -4:11:32
     GS1_magnitude: 14.9
```

```
        GS2_RA: 14:22:25
        GS2_Dec:-4:10:44
        GS2_magnitude: 10.4
        finding_chart: nig96001.001.fits
end_object:
```

Next I define the instrument setup for the UFOE.

```
# here is my instrument setup (it may be more complex later)
begin_instrument: I1
        instrument: UFOE
        fiber: two_fiber
        grating: 79
end_instrument:
```

And finally the plan, which simply selects the instrument setup that I want and does the object the way I want it done (two exposures).

```
# here is my simple plan
begin_plan: P1
        instrument: I1
        do-object: n2 1200 scaled 2
end_plan:
# this ends my observing script
```

Note that the PI section can be easily recycled from script to script. Also that any or all of the templates could be reused again easily if it worked in another plan. In fact, one could assemble all the parts in different files and then construct a script from them by simply concatenating the files. Finally note that adding more objects is as simple as defining them and writing a plan to observe them. However, the order in which objects and other templates appear in the file is important, as before a plan can call an object or instrument setup, it must be defined. In the order given here, the plans are valid. However if one were to put the instrument template I1 after the plan P1, the syntax checker would give an error saying that I1 was not a defined instrument template.

Adding a Calibrator

Now let us add some more details to another plan. Say we need a calibrator to remove telluric lines and 4 bias frames right after the object and then need four special Ar lamp spectra (a non–standard one) taken sometime later in the night. So in addition to the above summary and object templates, we define:

```
# here is my calibrator
begin_calibrator: C1
     description: star
     RA: 14:44:33.2
     Dec: -2:10:19
     Equinox: B1950
     Flux: 5500 continuum 5.4 mag
     V: 5.4
     B-V: 0.4
     GS_equinox: B1950
     GS1_RA: 14:45:38
     GS1_Dec: -2:11:32
     GS1_magnitude: 14.9
     GS2_RA: 14:43:25
     GS2_Dec:-2:10:44
     GS2_mangitude: 10.4
     finding_chart: nig96001.012.fits
end_calibrator:

# here is an argon lamp that I need
begin_lamp: AR
     lamp: Ar
end_lamp:

# and the new plan is here
begin_plan: p2
     priority: 1
     instrument: I1
     do-object: n2 2400 scaled 2
     do-calibrator: C1 600 scaled
     do-bias: 4
# NOTE THE DASH TELLS THE RESIDENT ASTRONOMER TO SELECT AN
# APROPRIATE EXPOSURE TIME
     schedule-lamp: AR - 4
end_plan:
```

Five Objects and a Macro

OK. That wasn’t too bad. But now say we want to do that last plan for five different objects. We could type it in five times, or use an editor to do a cut–and–paste job and replace the things that are different, or we could define a macro.

We need a macro that accepts the names and integration times of the object and calibrator. We then use the same commands as in the above plan, substituting these passable variables for the names and times of the object and calibration exposures. Note that all variables start with the “$” character. Once the macro is defined, we can make up five simple plans for five different objects o1 through o5 (that I will not define here for the sake of space).

```
# Here is the macro. Note that the variables are
# $object, $time, $calibrator, and $caltime.
begin_macro: M1 $object $time $calibrator $caltime
     instrument: I1
     do-object: $object $itime scaled 1
     do-calibrator: $calibrator $caltime scaled 1
     do-bias: 3
end_macro:

begin_plan: P3
     priority: 2
     # note the syntax on the macro call has the four
     # variables that are to be passed to the macro
     do-macro: M1 o1 2000 c2 2500
end_plan: P3

begin_plan: P4
     priority: 2
     do-macro: M1 o2 1000 c3 2500
end_plan: P3

begin_plan: P5
     priority: 5
     do-macro: M1 o3 2000 c1 2500
end_plan: P3

begin_plan: P6
     priority: 3
     do-macro: M1 o4 2500 c4 2500
end_plan: P3

begin_plan: P7
     priority: 4
     do-macro: M1 o5 2000 c5 2500
end_plan: P3
```

And that is all you need. When issuing a list of many sources, using a macro and an editor with “cut–and–paste” ability will be very useful.

Finally we comment that it is possible to have a macro define an entire plan. That is to say that the begin_plan: and end_plan: commands can be embedded in a macro and then a call to that macro expands into a plan. For example:

```
begin_macro: M2 $name $object $time $calibrator $caltime
     begin_plan: $name
       instrument: I1
       do-object: $object $itime scaled 1
       do-calibrator: $calibrator $caltime scaled 1
       do-bias: 3
     end_plan:
end_macro:

do-macro: PL3 M2 o1 2000 c2 2500
do-macro: PL4 M2 o2 1000 c3 2500
do-macro: PL5 M2 o3 2000 c1 2500
do-macro: PL6 M2 o4 2500 c4 2500
do-macro: PL7 M2 o5 2000 c5 2500
```

would expand into a complete plans called PL3, PL4, PL5, PL6, and PL7 that are identical to the ones listed as P3 through P7. One could them simply write many one line do–macro calls that expand into their own scripts, saving time and keystrokes.

In fact, macros will allow the user to write simple observation sets on one line. By putting the object and plan inside of a macro, one can create an object and its plan in one step. We need to observe some large set of objects with the UFOE with the slit fibers and the 79 lines/mm grating. All have the same setup but different locations (all positions are J2000) on the sky and different exposure times. Hence we write the following:

```
begin_instrument: I4
     instrument: UFOE
     fiber: slit
     grating: 79
end_instrument:

begin_macro: m1 $name $ra $dec $itime
     begin_object: $name
       RA: $ra
       Dec: $dec
       Equinox: J2000
     end_object:
     begin_plan: $name
       instrument: I4
       do-object: $name $itime scaled 1
     end_plan:
end_macro:

do-macro: M1 BS1842 3:14:43 65:22:14 1200
do-macro: M1 BS2440 5:12:12 54:47:22 600
do-macro: M1 BS4221 12:23:22 10:12:44 1800
do-macro: M1 BS4222 12:24:33 -5:22:23 1000
do-macro: M1 BS8644 20:44:13 12:44:55 500
```

Note that this is the absolute minimum amount of information needed to take a generic exposure of an object. No information is given regarding finding charts, guide star positions, object brightness, object color, or object type. **All time used to find such an object without any assistance from the PI will be billed to the PI.** Hence we suggest users use more advanced versions of this macro that include at least these bits of information about the object being observed. A more complex macro could be written that included this other information. However, even if there are ten or so entries on the do-macro line, this is clearly a time and space saver when doing the same thing for a large list of objects.

A Complex Plan

And now for a complex plan. Let us begin anew. Let us say we have three objects, o1, o2 and o3, two instrument setups, i1 and i2, two different calibrators, c1 and c2, and a flat and two arc lamps defined as templates. We have found that we can do all three of these objects in a night in two different sets of observations if we start with i1 and do o1 and o2 together with a calibrator c1, wait a while then do o3 and a calibrator c2 roughly three hours later, then wait again for two hours and change setups to setup i2 and do o3 and a calibrator c2, then wait again and do o1 and o2 with the calibrator c1 sometime in the next two hours. All of these schedule flat lamps and arc lamps to be acquired later in the night. These observations then need to be repeated in three weeks plus or minus 2 days. We have defined two constraint templates, east and free, to catch the objects rising and then to remove the constraint of east in the plan. This is needed to get all the observations into one night.

```
begin_constraint: east
      east:
end_constraint:

begin_constraint: free
     free:
end_constraint:

# this is the plan

begin_plan: p1
     instrument: I1
     # start this with the objects rising...
     constraint: east
     do-object: o1 1600 scaled
     do-object: o2 1600 scaled
     do-calibrator: c1 600 scaled
     schedule-flat: flat 60 4
     schedule-lamp: Ar 20

     # note how the schedule-wait with the large margin of
     # error really lets us schedule it anytime
     # as it is not time critical. It just keeps the
     # observation in the
     # same night. Further we don't care if the tube is east
     # or west as now we have started the process in the early
     # part of the night
     constraint: free
     schedule-wait: 3:00:00 delta 2:00:00

     do-object: o3 1600 scaled
     do-calibration: c2 600 scaled
     schedule-wait: 1:00:00 delta 1:00:00
     instrument: i2
     do-object: o3 1600 scaled
     do-calibrator: c2 600 scaled
```

```
        do-object: o1 1600 scaled
        do-object: o2 1600 scaled
        do-calibrator: c1 600 scaled

        # here is a schedule wait for 14±2 days.

        schedule-wait 14 delta 2
        instrument: i1
        constraint: east
        do-object: o1 1600 scaled
        do-object: o2 1600 scaled
        do-calibrator: c1 600 scaled
        schedule-flat: flat 60 4
        schedule-lamp: Ar 20
        schedule-wait: 3:00:00 delta 2:00:00
        do-object: o3 1600 scaled
        do-calibrator: c2 600 scaled

        constraint: free
        schedule-wait: 1:00:00 delta 1:00:00
        instrument: i2
        do-object o3 1600 scaled
        do-calibrator c2 600 scaled
        do-object: o1 1600 scaled
        do-object: o2 1600 scaled
        do-calibrator: c1 600 scaled

    end_plan:
```

Of course if one wanted to, one could write a macro for the nightly observation and then call the do–macro command with the schedule–wait command, to save some typing.

A Time Critical Observation

Next let us schedule something time critical. We have found that by going to a non–optimal azimuth, we can observe object o1 at azimuth 20° on June 14, 1997 starting at 14:22 UT with a range of ± 10 minutes around this time. We have selected the azimuth here because if the telescope is NOT at this azimuth, the observation cannot span the time we require. If we start then, we can get an hour on the object. So we define a constraint template and write the plan as follows

```
begin_constraint: az20
     azimuth: 20
end_constraint:

begin_plan:
     instrument: I1
     constraint: az20
     utdate_start: 14/06/1997 14:22 delta 0:10
     do-object: o1 3600 fixed
end_plan:
```

Note we really didn't have to specify the azimuth, as the resident astronomer can calculate what azimuth will allow you to get the object at that time, but if the user wants a particular track that they have calculated, they can specify the azimuth and the start time (which, with an object's RA and declination, defines a single track). This may be essential in doing any precision photometry with the HET, for instance.

Phase Critical Observations

There are basically two classes of phase critical observations. The first requires that the object be observed at a given date or phase and then follow–up observations are taken at fixed periods after this observation. The second is simply that observations need to be made at some time during a particular phase of an object. Note that for all these examples, I have not put in all the required keywords (instrument, telescope) to save space.

The first of these two classes actually has two sub classes. The first is that the user knows the date on which the project must start and then knows how frequently to monitor the object. This is the simplest case. We want to start on October 4, 1997 (universal date) at 23:00 hours ± 2 hours and then return in 4 ± 1 days, and then 10 ± 2 days after the second observation. We use utdate_start to mark when to start the plan and then schedule–wait to put in the needed breaks in the observations. During these schedule waits, other observations can take place (and no time is billed to this plan). This would read

```
begin_plan: hh1tau
     utdate_start: 4/10/1997 23:00 delta 2:00
     do-object: hhtau 600 scaled 3
     schedule-wait: 4 delta 1
     do-object: hhtau 600 scaled 3
     schedule-wait: 10 delta 2
     do-object: hhtau 600 scaled 3
end_plan:
```

The second kind of monitoring observation requires one to start at a particular phase of the object's variability (any one will do) and then do some follow–up observations to monitor its progress (using the same times from above). This would read very similarly, with the phase_start keyword replacing the utdate_start.

```
begin_plan: rr2tau
     phase_start: rrtau 0 delta 0.05
     do-object: rrtau 600 scaled 3
     schedule-wait: 4 delta 1
     do-object: rrtau 600 scaled 3
     schedule-wait: 10 delta 2
     do-object: rrtau 600 scaled 3
end_plan:
```

The other kind of monitoring project requires sampling of an object at different phases of its variability, but not necessarily sequentially. This is achieved through using different plans with different phase_start parameters. Say we had to get an object at phases 0, 0.3, and 0.7 with tolerances of 0.1 in phase. That would read:

```
begin_plan: x1.0
     phase_start: x1 0 delta 0.1
     do-object: x1 600 scaled 3
end_plan:
```

```
begin_plan: x1.3
     phase_start: x1 0.3 delta 0.1
     do-object: x1 600 scaled 3
end_plan:
begin_plan: x1.7
     phase_start: x1 0.7 delta 0.1
     do-object: x1 600 scaled 3
end_plan:
```

These three plans show when a single plan is needed (when an observation is only good if a prior goal has been achieved) and when multiple plans are needed (here where the data are good even if some prior condition has not been achieved). It would be possible to write the latter plan using schedule–wait commands. The problem would then be that that plan has less flexibility in how it is scheduled. The more leeway the queue is given, the easier it is to complete a project.

Note that many phase dependent plans are not possible because of the limited motions of the HET. It is up to the PI to test to see whether their observations can be done given the limited time any object can be observed.

Some Confusing but Potentially Useful Commands

What follows now are some examples of some rather useful but potentially confusing commands. First, schedule–object.

The schedule–object command is intended to allow the user to schedule an observation of some object in a night if and only if another observation is scheduled for execution that night. We note that the schedule–object may be executed prior to the do–object if it fits into the schedule better. Unlike a calibration frame, these frames are proprietary to the PI, and are not shared. For example, say you wanted to observe Ganymede and if and only if you got that spectrum, then get a spectrum of Io that night (either before or after the Ganymede spectrum is taken), and a G5 solar analog spectrum as a calibrator. Then one would plan

```
begin_plan: moons
     instrument: I2
     do-object: ganymede
     schedule-object: io
     schedule-calibrator: c1 500 fixed
end_plan:
```

Note that as this reads, there is no way to make the scheduled objects have any do commands associated with it. That is to say if you needed flats right after both Ganymede and Io, you would be out of luck with this plan. To do this, you could make a macro with a do–object and a do–lamp command and schedule/do it as needed. For example

```
begin_macro: m3 $object $time
     do-object: $object $time scaled
     do-flat: 600
end_macro:

begin_plan: moons2
     do-macro: m3 ganymede 1000
     schedule-macro: m3 io 500
     schedule-calibrator: c1 500 fixed
end_plan:
```

This exemplifies a use for macros beyond their use as a space–saving device. However, one could use the begin_set and end_set to accomplish the same end. Such a plan would read:

```
begin_plan: moons3
     begin_set:
       do-object: ganymede 1000 scaled
       do-flat: 600
     end_set:
     begin_set:
       do-object: io 500 scaled
       do-flat: 600
     end_set:
     schedule-calibrator: c1 500 fixed
end_plan:
```

Checking and Submitting a Script

Plans are to be submitted and checked via an e–mail reflector. Two different e–mail accounts have been created to accept e–mailed plans from users and check the syntax. The e–mail address `check@rhea.as.utexas.edu` will accept a plan and check its syntax. Once this is done, an e–mail message will be sent to the user telling him/her that the checking is finished and, if there are any errors, will report them. Note that plans submitted to the `check@rhea.as.utexas.edu` account are not officially submitted. This account is for users to check and debug their plans before submitting them officially.

The e–mail address to which plans are to be officially submitted or revised plans are to be resubmitted is `submit@rhea.as.utexas.edu`. This account will save the last submitted plan in the users disk area and check it for syntax. If a plan from the same project is resubmitted, the old plan will be overwritten. It will also inform the Ops. team that the plan has been submitted.

Before submission, all plans, templates, and macros must be assembled in a single file. The order in which these appear in the file is important. Any object, macro, or other template that is called in a plan must be defined above it in the file. However, templates and macros can be mixed together in the file in any order. For simplicity, we also request that the first template found in the file be the summary template, for easy identification of who the submission belongs to.

Note that for both the submit and check addresses, the automated checking software will send a response to the address listed in the plan and not in the e–mail header. If the user does not get a response, then an error has occurred either in the e–mail transport or in the software. The user must be aware that their plan has not been accepted by HET Ops. until a response is received at the e–mail address listed in the PI section of the summary template. DO NOT ASSUME THAT BECAUSE E–MAIL HAS BEEN SENT THAT IT HAS BEEN RECEIVED AND VERIFIED. Please inform the HET Ops. team of any errors encountered of this nature.

Finding Charts

As the HET initially will not point extremely accurately, finding charts will be essential to making this project work efficiently. Further, because of the number of objects that are expected to be in the observation queue, use of hard copy finding charts by the observing staff is not efficient. Hence we will try to, whenever possible, use digital finding charts. These can be either created from the digitized sky survey (DSS) or from the PI's personal collection of FITS images (the PFC on the McDonald 30" will be a prime instrument for creating images of very faint objects to be observed with HET).

Submission of these digital finding charts will be by ftp to the user's account on the HET Ops machine. Hence, they must each have a unique name. To uniquely tag a finding chart, one uses the information supplied to the PI about their project. The first is the user name of the account on the data distribution computer. The second is the project number assigned to the project by the TACs. Both of these pieces of information will be mailed (using the US Post Office for PIs outside of the University of Texas) to the PI at his/her institution. This information is required in the summary template and is listed as

```
project_username: nig
project_id: UT 96001
```

Now one simply selects numbers for each finding chart. The first one might be 001. Then combining these pieces of information, we use the formula

`<project_username><project_id 2nd entry>.<finding chart number>.fits`

to create a finding chart named `nig96001.001.fits`. This is then the name of the finding chart that is submitted to the Ops. team and is the name given in the finding_chart keyword in the object or calibrator template for this object.

In addition to the right name, the FITS file must have a few other things. First, we require that the object in question be marked digitally in the file. To do this, the FITS header keywords CRPIX1 and CRPIX2 should mark the X and Y pixel where the object can be found. Further, if possible, we request that the file have a valid WCS (world coordinate system) so that image display tools (e.g., SAOimage) can solve for RA and Dec for each pixel in the frame. If given the correct coordinates for the object (which would be at the field center), the DSS extraction programs will do both of these things for you.

Extracting DSS images is fairly simple. If available locally, one can use the CD–ROMs and the program `getimage` that is supplied with the disks to extract images. Alternately, and perhaps more easily used, the web offers many different DSS image sites. SkyView can be found at the URL `http://skyview.gsfc.nasa.gov`. Simply use your favorite graphical WWW browser (Netscape, Mosaic) to go to this site and follow the directions. Another good site is `http://archive.eso.org/cgi-bin/dss`. Further, ESO has created a FITS viewing tool that will display local files as well as download DSS images directly into the tool. It has other features that allow it to download catalogs and mark the position of objects, such as guide stars, in the field. This program is called skycat. More detailed information and both source code for skycat and compiled binaries for many platforms are available from ESO at the URLs `http://archive.eso.org/skycat` or `ftp://ac1.hq.eso.org/B/skycat`. For users at the University of Texas, it has been installed under both Solaris and SunOS on the /opt/local and /local disks, respectively.

As an example of what should be in the FITS header, here is one from the DSS:

```
SIMPLE  =                    T / 20-Sep-1995 15:34:03.00
BITPIX  =                    8 / Bits per pixel.
NAXIS   =                    2 / Two dimensional image.
NAXIS1  =                  300 / Length of x axis.
NAXIS2  =                  300 / Length of y axis.
BSCALE  =              58.5267 /Scaling applied to data
BZERO   =              3665.70 /Offset applied to data
CTYPE1  = 'RA---TAN'           / X-axis type
CTYPE2  = 'DEC--TAN'           / Y-axis type
CRVAL1  =              187.278 / Reference pixel value
CRVAL2  =              2.05238 / Reference pixel value
CRPIX1  =              150.500 / Reference pixel
CRPIX2  =              150.500 / Reference pixel
CDELT1  =         -0.000333333 / Degrees/pixel
CDELT2  =          0.000333333 / Degrees/pixel
CROTA1  =              0.00000 / Rotation in degrees.
EQUINOX =              2000.00 /Equinox of coordinates
```

This header gives the coordinate system with the CRPIX1 and CRPIX2 keywords marking the X and Y pixel of the object in question. This object is at the coordinates given in degrees by the CRVAL1 and CRVAL2 keywords while the CDELT1 and CDELT2 give the plate scale, CROTA1 gives the rotation angle of the frame, and EQUINOX gives the equinox of the coordinate system.

Collaborative Projects

One peculiar feature of the HET is that collaborations between PIs at different partner institutions must somehow be billed separately while the project is worked on as a single unit. Hence we have created the collab_id keyword in the summary template. However, decisions on the part of the collaborating scientists must be made prior to plan submission. First, they must exchange the results from the TACs, including the different project numbers which have been allocated time to work on this single project. Then they must pick a PI to head this project. The head PI will go into the PI section of the summary template while the other partners will be co–investigators. If one of the other collaborators wants to be the contact person but not the PI, they can do so by being the contact person for this project. However, apart from the billing aspect of the HET, this project will be referred to in both the status reports and in the file ids as the PI's project. The entire project is then dealt with as a single project with different methods of billing. **Each collaborative project will have a single observing script submitted by the head PI even though parts of the project will be billed to each of several institutions.**

As an example, let us say we have three investigators, Moe, Larry, and Curly from UT, SU, and UG. They have been granted time in proposals 96042, 96004, and 96012 respectively. Say they appoint Larry the PI of this project. Then the summary section looks a bit like this:

```
project_title: More Great Astrophysics
project_username: larry
project_id: SU 96004
collab_id: UT 96004
collab_id: UG 96012
begin_PI:
     name: Larry
     email: larry@astro.stanford.edu
     affiliation: SU
     phone: (817)555-1212
     address: Astronomy/Stanford/California 90011
end_PI:
begin_coi:
     name: Moe
     email: moe@astro.as.utexas.edu
     affiliation: UT
     phone: (512)555-1212
     address: Astronomy/Austin/Texas 78712
end_coi:
begin_coi:
     name: Curly
     email: curly@astro.ug.de
     affiliation: UG
     phone: 011-12-55-23-33
     address: Astronomy/Goettingen/Germany
end_coi:
begin_abstract:
     We're doing great things...
end_abstract:
```

Note how I have grouped the different investigators with their affiliations and proposal numbers. This way we can determine who belongs to what allocated time (according to the TACs). Here, Larry will get the e–mail about the data coming in (regardless of which institution got billed for the time). Of course it will be allowable for the group to share the data retrieval account so all partners can check on the data. **However, we warn the users that the Ops. team will have no way of telling WHO retrieved data from the data repository. Hence if any person retrieves the data, we will assume that it has been retrieved by the GROUP and that the data can be removed from the disk.** We therefore highly recommend that only one of the collaborators retrieve the data as it comes in and then distribute it to his/her collaborators some other way .

The plans in this project will be executed based on when an individual plan can be executed and the rank it is given by each local TAC. However, the program will be run out of this single file regardless of whose time is being used. For example, let's say that project X was granted 15 hours by Texas at a rank of 5 and 23 hours from Penn State at a rank of 2. When a plan from this project can be done, it resides in the pool of potential projects twice, once as a Texas rank 5 project and once as a Penn State rank 2 project. The plan is then selected based on the selection criteria of that moment. If it were scheduled as a Texas plan, then the amount of time used would be deducted from the Texas share. Once the Texas time is used up, then the plans can only be executed as though they were Penn State time at a rank of 2 (Note there may be times when, due to the consumption of time by other projects, the Penn State rank 2 time would be scheduled ahead of the Texas rank 5 as there is no way of comparing relative ranks from the different TACs.)