



AAO2 (ODC and IRIS 2) Sequencer Task

Contents

1	<i>Introduction</i>	2
2	<i>Sequences</i>	2
2.1	A simple single exposure example	2
2.2	Waiting for events	5
2.3	Using Tcl	6
2.4	IRIS 2	7
2.5	Locating Sequence scripts	7
3	<i>Sequencer Task Interface</i>	8
3.1	Task Actions	8
3.1.1	INITIALISE	8
3.1.2	RESET	8
3.1.3	CONFIGURE	8
3.1.4	RUN	9
3.2	Task Parameters	9
3.2.1	RUNNING	9
3.2.2	SEQUENCE	9
3.2.3	STEP	9
4	<i>Sequence Commands</i>	9
4.1	General Sequencer commands	9
4.2	DETECTOR Subsystem commands	10
4.3	DETECTOR Subsystem events	11
4.4	TELESCOPE Subsystem commands	12
4.5	TELESCOPE Subsystem events	12
4.6	IRIS2 Subsystem commands	12
4.7	IRIS2 Subsystem events	13
A	<i>IRIS 2 standard procedures</i>	13
B	<i>Real IRIS 2 Example procedures</i>	14
B.1	AAO_Nod_Alone_Slit.tcl	14
B.2	AAO_Flat_Arc.tcl	16
B.3	AAO_Spec_Camfoc.tcl	18

Revisions:

- 14-Mar-2000: First Version.
- 04-Aug-2000 New detector commands
- 05-May-2003 Proper ASD number allocated – KS.
- 01-Dec-2004 Modify for ODC Software – MVV.
- 26-Apr-2005 ACMM Version 1.2, Update and Merge IRIS 2 and ODC information. Add stuff on locating scripts and sections on IRIS 2 scrips. TJF

1 Introduction

This document describes the sequencer task for the IRIS2 and ODC systems. This task is responsible for executing sequences of operations that involve coordinated use of different subsystems, typically the detector and the telescope.

The sequencer is designed to be sufficiently flexible to be used for other instruments based around the new AAO CCD controller.

2 Sequences

Sequences are specified in the form of text files. The main action of the sequencer task is the RUN action that executes a specified sequence.

Sequences can thus be generated using any text editor. However, we envisage that there will be simpler ways of generating standard sequences using a graphical user interface.

2.1 A simple single exposure example

Here is an example of a simple single exposure sequence file

```
# Sequence to take two K band imaging runs at two telescope
# positions

DETECTOR_1 config observation_description
DETECTOR_2 config observation_description

TELESCOPE offset 0 0

# Note: for ODC, always specify the first run action with a
# nowait option and the second run action with either a nowait
# or wait (default) option depending on whether you want to
# wait for the action to complete or not.
#
# If the nowait option is not specified for the first run,
# the script will start the action on DETECTOR_1 and then wait
# for it to complete before starting the other run action on
# DETECTOR_2. This is not normally what is intended.
DETECTOR_1 run -nowait
DETECTOR_2 run

TELESCOPE offset 0 20
DETECTOR_1 run -nowait
DETECTOR_2 run

TELESCOPE offset 0 0
```

Example 1

Here is an example of a multiple exposure sequence file

```

# This example assumes that the observation description
# is setup for Nod and Shuffle runs.
DETECTOR_1 config observation_description
DETECTOR_2 config observation_description

TELESCOPE offset 0 0

# For multi-exposure (i.e nod and shuffle), the run action for
# both detectors must be specified with nowait otherwise it
# wont work
DETECTOR_1 run -nowait
DETECTOR_2 run -nowait

DETECTOR_1 expose 100 -nowait
DETECTOR_2 expose 100 -nowait

DETECTOR_1 wait exposure_end
DETECTOR_2 wait exposure_end

TELESCOPE offset 0 20

DETECTOR_1 shuffle 50
DETECTOR_2 shuffle 50

DETECTOR_1 expose 20 -nowait
DETECTOR_2 expose 20 -nowait

DETECTOR_1 wait exposure_end
DETECTOR_2 wait exposure_end

DETECTOR_1 readout -nowait
DETECTOR_2 readout -nowait

TELESCOPE offset 0 0

DETECTOR_1 wait readout_end
DETECTOR_2 wait readout_end

```

Example 2

The sequence consists of a series of commands. The first word in the command specifies the subsystem that the command applies to. There are three sub-systems used above - two detector subsystems and the telescope sub-system. The commands to access these sub-systems are:¹

DETECTOR_1 or DETECTOR_2 — Commands involved in taking data from the detectors are sent to these subsystems. The number - _1 or _2 refers to the interface

¹ The subsystems and their names are set up by the CONFIGURE action of the sequencer task.

on which the Detector is running. For IRIS 2, you should drop this and we intend to adapt the software such that it works without the underscore on any system with just one detector in use

TELESCOPE — Commands to move the telescope are sent to this subsystem.

The first two commands in the sequence are “**DETECTOR_n config**” commands that specify the observation description file used for the data taking. Thus the “**observation_description**” file would specify the data taking mode, the exposure time, cycles, window etc. The easiest way to generate a normal observation description file is to set up your observation using the ODC GUI and then save the window (which is the observation description). The OBSGEN tool can also be used, and this is the normal way to generate an observation description appropriate for charge shuffling (where the CCD_METHOD in the observation file must be set to CCD_SHUFFLE rather than CCD_NORMAL). See Document ASD81 for details on this tool.

The remaining commands specify a series of telescope offsets and data taking runs. A run is taken, the telescope is then offset 20 arc seconds and another run taken. The telescope is then returned to its initial position.

By default, the Sequencer waits for the command to complete before executing the next one. Therefore in a multi exposure mode, it is advisable that when issuing commands (i.e expose, shuffle and readout) to the detectors, the nowait option should be used and then explicitly wait for the completion of the action/command with the wait command as shown in the example above.

Here is an example of an invalid sequence file

```
# This example assumes that the observation description
# is setup for Nod and Shuffle runs.
DETECTOR_1 config observation_description
DETECTOR_2 config observation_description

DETECTOR_1 run -nowait
DETECTOR_2 run -nowait

DETECTOR_1 expose 100 -nowait
DETECTOR_2 expose 50

DETECTOR_1 readout
DETECTOR_2 readout
```

Example 3

The problem with the above sequence file is that by the time the detector_2 finishes its 50 seconds exposure and starts executing a readout on the detector_1, it (detector_1) is still exposing. The behavior of the detector in this case is implementation dependent.

The Sequencer tries to detect such an error and in this case, it detects that there is an expose action with a `nowait` option but there is no corresponding wait (i.e. `DETECTOR_1 wait exposure_end`) and therefore, it will output an error message to the screen explaining why it has failed. It then automatically tries to abort the sequence.

Another potential problem with example 3 above is that the script may exit before the run is actually completed. This causes the Sequencer task to generate an error message saying that there are outstanding actions.

2.2 Waiting for events

While the above sequence in example 1 will work, it won't do so with optimum efficiency. The reason for this is that it will wait for each step in the sequence to complete before starting the next one. In the case of a **run** command, the run does not complete until the data file is written to disk. However, we could actually start offsetting the telescope as soon as the exposure has completed. This can be achieved with the following version.

```
# This example assumes that the observation description
# is setup for Normal (i.e not Nod and Shuffle) runs.

DETECTOR_1 config observation_file
DETECTOR_2 config observation_file

TELESCOPE offset 0 0

DETECTOR_1 run -nowait
DETECTOR_2 run -nowait
DETECTOR_1 wait exposure_end
DETECTOR_2 wait exposure_end

TELESCOPE offset 0 20

DETECTOR_1 wait readout_end
DETECTOR_2 wait readout_end

DETECTOR_1 run -nowait
DETECTOR_2 run -nowait
DETECTOR_1 wait exposure_end
DETECTOR_2 wait exposure_end

TELESCOPE offset 0 0

# The sequencer script should wait for the runs to complete
# before exiting otherwise the SEQUENCER task outputs an
# error message saying that the script has completed while
# there are outstanding actions.
after 180000; # 3 minutes
```

Example 4

In example 4, the **-nowait** option is specified on the **run** command which causes this command to start the run and then return immediately. This mechanism allows two or more operations to be performed concurrently.

Note1: The default behavior (if **-nowait** is not specified) is always to wait for completion of any operation.

Note2: It is advisable to specify **nowait** option for the run, expose and readout actions when performing nod and shuffle (multi exposure mode i.e CCD_METHOD = CCD_SHUFFLE). See example 2.

2.3 Using Tcl

The sequence files used by the sequencer task are actually in the Tcl language and are run by a Tcl interpreter incorporated in the task. Thus any Tcl features can be used within them. Here is an example which uses Tcl procedures and looping constructs.

```

proc nod {n} {

    DETECTOR_1      config nod_n_shuffle_obs
    DETECTOR_2      config nod_n_shuffle_obs

    TELESCOPE offset 0 0

    set i 0
    set initial 0
    while {$i <= $n} {
        DETECTOR_1 run -nowait
        DETECTOR_2 run -nowait

        DETECTOR_1 expose 10 -nowait
        DETECTOR_2 expose 10 -nowait

        DETECTOR_1 wait exposure_end
        DETECTOR_2 wait exposure_end

        DETECTOR_1 shuffle 100
        DETECTOR_2 shuffle 100

        set initial [expr $initial + 20]
        TELESCOPE offset 0 $initial

        DETECTOR_1 expose 50 -nowait
        DETECTOR_2 expose 50 -nowait

        DETECTOR_1 wait exposure_end
        DETECTOR_2 wait exposure_end

        DETECTOR_1 readout -nowait
        DETECTOR_2 readout -nowait

        DETECTOR_1 wait readout_end
        DETECTOR_2 wait readout_end

        incr i
    }
    TELESCOPE offset 0 0
}

```

```
# Execute the above procedure with n=3

nod 3
```

2.4 IRIS 2

The sequencer task was initially commissioning with IRIS 2. Currently (April 2005) IRIS 2 uses its original version of the sequencer (documented by ACMM Version 1.1 of ASD104). We intend to move IRIS 2 to using the current version of the sequencer and to ensure that all original IRIS 2 sequences continue to work. IRIS 2 has only one detector and uses DETECTOR rather the DETECTOR_1 to access the detector. As mentioned above, DETECTOR will become the way of accessing the detector if there is only one detector in the system. It adds an instrument sub-system IRIS2, used to access the IRIS 2 spectrograph.

The example below is a typical IRIS 2 sequence.

```
# Sequence to take two K band imaging runs at two telescope
# positions

DETECTOR config iris_obs2
IRIS2 config k_imaging

TELESCOPE offset 0 0

DETECTOR run
TELESCOPE offset 0 20
DETECTOR run
TELESCOPE offset 0 0
```

Note that use of the IRIS2 command to control the IRIS 2 spectrograph.

2.5 Locating Sequence scripts

In the ODC (CCD) system, sequences are loaded automatically from the directory pointed to by the environment variable SEQUENCES_DIR. For IRIS2, the environment variable IRIS2_STANDARD_DIR is used.

In both cases, this environment variable must be defined before the DRAMA network tasks are started. In addition, if not defined, the default directory is used.

As things currently stand (29/Apr/2005) these environment variables are not being used at the AAT, the default (login) directory of the instrument account is used. This will change – but for the moment, the directories are

Instrument	Account	Login Machine	Directory
IRIS2	aatobs	aatvme10	/export/home/aatobs
CCD Systems	aatinst	All machines	/instsoft/instusers/aatinst

Please note that the “aatobs” account, currently used for IRIS 2, has a different directory if you log into that account on say aatssy.

3 Sequencer Task Interface

3.1 Task Actions

3.1.1 INITIALISE

Initialise the sequencer task.

3.1.2 RESET

Reset the sequencer task.

3.1.3 CONFIGURE

Configure the sequencer task. The main purpose of this action is to specify the DRAMA tasks which control the various subsystems. It has three arguments:

DETECTOR_TASK	<p>A string giving the name of the run control task (RCT) controlling the detector system.</p> <p>OR</p> <p>A structure listing one or more such tasks, with the component names being the names by which the detectors will be known to the sequencer, and the values being the DRAMA task names.</p>
TELESCOPE_TASK	<p>A string giving the name of the telescope task (PTCS) controlling the telescope system.</p> <p>OR</p> <p>A structure listing one or more such tasks, with the component names being the names by which the telescopes will be known to the sequencer, and the values being the DRAMA task names.</p>
INSTRUMENT_TASK ²	<p>A string giving the name of the instrument task controlling the instrument.</p> <p>OR</p> <p>A structure listing one or more such tasks, with the component names being the names by which the instruments will be known to the sequencer, and the values being the DRAMA task names.</p>

² The operation of the INSTRUMENT_TASK option needs to be checked out with the ODC system. TJF, 26 April 2005. It may have got broken.

3.1.4 RUN

Execute a sequence. The action has one argument, a character string specifying the name of the sequence file to be executed.

The RUN action supports the following kicks of various types specified by a KICKTYPE argument.

STOP —	Stop execution of the sequence at the end of the current step.
ABORT	Abort the current run (if one is in progress) and stop the sequence at the end of the current step.
HOLD	Suspend execution of the sequence after completing the current step.
CONTINUE	Continue sequence execution after a HOLD.
EXPOSE	Start an exposure. Only available if the method field of the observation description is SHUFFLE i.e Nod and Shuffle run.
SHUFFLE	Shuffling charge in the CCD horizontally. Only available if the method field of the observation description is SHUFFLE i.e Nod and Shuffle run.
READOUT	Read the data from the CCD. Only available if the method field of the observation description is SHUFFLE i.e Nod and Shuffle run.

3.2 Task Parameters

3.2.1 RUNNING

This is an integer parameter which is set to one if a sequence is currently being executed or to zero at other times.

3.2.2 SEQUENCE

The file name of the sequence currently being executed.

3.2.3 STEP

The step in the sequence currently being executed.

4 Sequence Commands

4.1 General Sequencer commands

These commands are not sub-specific related, but are used in managing sequencer scripts.

Command	Parameter(s)	Purpose
PAUSE	script defined	<p>First sends a trigger message to the invoking task. The argument of the trigger message is a character string with all the command arguments (including the command itself), space separated.</p> <p>Then suspends the run as if the RUN action had received the "HOLD" kick message. To continue, the RUN must be sent the "CONTINUE" kick message (which can be done by the GUI), at which time this command will complete.</p>
ABORT	ignored	Abort the current run (if one is in progress) and stop the sequence at the end of the current step.
MESSAGE	script defined	Sends a trigger message to the invoking task. The argument of the trigger message is a character string with all the command arguments (including the command itself), space separated.

4.2 DETECTOR Subsystem commands

Command	Parameter(s)	Purpose
config	Name of observation description file	Specify the observation description to be used for subsequent runs
run	obstype (optional)	Start a run as specified by the current observation configuration. Obstype specifies the type of observation (BIAS, DARK, ARC, FLAT, OBJECT, SKY). Obstype also specifies whether it is a NORMAL or SHUFFLE run. In the case of the SHUFFLE run, issuing the RUN command does not actually starting the run. To actually start the run, the EXPOSE kick has to be issued and to readout the data, the READOUT kick has to be executed.
recipe	Name of an ORACDR reduction recipe	Specify a reduction recipe to be used for subsequent runs in the sequence.

startgroup	maxgrp (optional)	Start an ORAC DR reduction group. The optional maxgrp parameter is the maximum number of observations in the group. The group is ended by an endgroup command or another startgroup.
endgroup	—	End a group started with startgroup
fits_real	Keyword, value, comment	Add a real FITS header entry for the next or current run.
fits_integer	Keyword, value, comment	Add an integer FITS header entry for the next or current run.
fits_logical	Keyword, value, comment	Add a logical FITS header entry for the next or current run. (value can be TRUE or FALSE)
fits_string	Keyword, value, comment	Add a character string FITS header entry for the next or current run.
comment	Comment string	Add a FITS comment record for the next or current run.
object	Object name	Specify an object name – used for all subsequent runs until changed.
exposure	Exposure time in seconds	Specify a new exposure time overwriting the value in the observation description.
cycles	Number of cycles	Specify a new number of cycles overwriting the value in the observation description.
reads	Number of reads	Specify a new number of reads overwriting the value in the observation description.
expose	—	For a multiple exposure CCD run (e.g. charge shuffling method), start a new exposure.
readout	—	For a multiple exposure CCD run (e.g. charge shuffling method), start the readout.

4.3 DETECTOR Subsystem events

Name	Event
exposure_start	Start of exposure (i.e. opening of the shutter for a CCD, or initial read for IRIS).
exposure_end	End of exposure (i.e. closing of the shutter for a CCD or final read for IRIS).
readout_end	End of readout
run_end	End of run. This means all activity associated with the run has completed, including writing the data file to disk.

4.4 TELESCOPE Subsystem commands

Command	Parameter(s)	Purpose
offset	EW and NS offsets (arc seconds)	Offset the telescope by the specified amount. The offsets are relative to a base position, and are in J2000 gnomonic projection.
offset_xy	X and Y offsets (mm)	Offset the telescope by a specified amount in mm in the focal plane. The XY system is defined so that it is fixed relative to the detector however the instrument is rotated.
axis	Axis name (A, B or REF)	Change the telescope pointing axis.
focus	Focus offset in mm	Offset the telescope focus by a specified amount.
control	ON or OFF	Turn telescope control on or off. This is needed before using either of the offset commands.
autoguider	FREEZE or THAW	Freeze or thaw autoguiding.

4.5 TELESCOPE Subsystem events

Name	Event
tracking	Telescope is tracking (i.e. has completed any movement)

4.6 IRIS2 Subsystem commands

Command	Parameter(s)	Purpose
config	Name of IRIS2 configuration file	Configure the IRIS2 mechanisms using a configuration file
slit	Slit position name	Move the IRIS2 slit/aperture wheel to a named position.

filter	Filter position name	Move the IRIS2 filter wheel to a named position.
coldstop	Coldstop position name	Move the IRIS2 coldstop wheel to a named position.
grism	Grism name	Move the IRIS2 grism wheel to a named position.
focus	Focus offset in steps	Move the IRIS2 detector to a new focus offset.

4.7 IRIS2 Subsystem events

Name	Event
ready	Instrument is ready (i.e. all mechanisms have completed any requested movements).

A IRIS 2 standard procedures

There exists a set of standard IRIS 2 Tcl commands that are often used in sequencer scripts. These are kept in the file “AAO_Inc.tcl”. To include them in a script, add the following line at the top of your script

```
source /export/home/aatobs/AAO_Inc.tcl
```

This script contains the following Tcl commands.

WFIrand	<p>A Pseudo-random number generator.</p> <p>The first argument determines the mode. If the first argument is “seed” then the second argument is a seed for the random number generator. You must set the seed before generating random numbers.</p> <p>Otherwise, the first argument is a limit of the random number and the function will return an integer pseudo random number between 0 and that number.</p>
GENrandoff	<p>Generate pseudo-random offsets for the telescope.</p> <p>“GENrandoff init” – Initializes memory of past offsets and seed.</p> <p>“GENrandoff 20” - generates random</p>

	<p>offsets in box +/-20</p> <p>GENranoff uses a memory of up to 30 past random offsets to compare each generated offset against, to ensure you are more than 1/10th of the total jitter box size away from all past offsets. If it can't do this in 30 tries then it gives up and purges its memory of past offsets, and starts again.</p> <p>The offsets generated are available in the global variables GENranoff_dra and GENranoff_ddec.</p>
MOVEimaging	<p>Set IRIS 2 wheels & focus for imaging based on a single string filter name.</p> <p>Move the IRIS 2 coldstop and filter wheels and focus for imaging based on a single filter name string, and a base camera focus in the J filter which is set by the IRIS2_BASE_FOCUS global variable and which must be updated at the start of a run if the base focus is found to have changed. This routine doesn't touch the GRISM and SLIT wheels, which currently have no filters in them.</p>
MOVEspectro	<p>Set IRIS 2 wheels & focus for spectroscopy based on a single string name</p> <p>Move the IRIS 2 wheels and focus for spectroscopy based on a single filter name string, and a base camera focus in the J filter which is set by the global variable IRIS2_BASE_FOCUS and which must be updated at the start of a run if the base focus is found to have changed.</p>

B Real IRIS 2 Example procedures

This section shows some real (user written) IRIS 2 example procedures.

B.1 AAO_Nod_Alone_Slit.tcl

This sequence is used for nodding along the slit. The intention of this (and many of the other IRIS 2 scripts) is the observer can edit the top part of the script – see the script for details. This script includes Detector control, Telescope control and specification ORAC DR recipe.

```

source /export/home/aatobs/AAO_Inc.tcl

#####
# Nod Along Slit.tcl
#####
# Sequence for observing a point source, nodding along slit.
# This sequence assumes you have already done all the acquisition
# for the object (including starting guiding if required),
# and placed it on the correct slit that you want to observe in
# for the selected configuration.
#
# Revison history:
#
# 26 July 2002: First draft (SDR).
# 29 July 2002: First draft copied from Nod Along Slit K(CGT).
# 30 July 2002: Editted to make instrument setup and telescope
# moves in parallel at start. (CGT).
# 10 July 2003: Modofied top used AAO_Inc to handle spectrsocopic
# configs and focus. Now only one script for
# all configs.
# 27 June 2004: Add a note to warn that multiple configurations
# must use the same slit. (SDR)
#
# Start by loading the default exposure configuration
#
DETECTOR config MRM NORMAL
#
#=====
# OBSERVER: Check, and if necessary modify parameters here
#=====
#
# Enter object name:
set objname "RXJ0152-13"

# Enter configuration to use. Must be one of: Js Jl J H Hs K Ks
# See AAO_Inc.tcl to see what precise configurations these
# names give you.
#
# N.B!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# You cannot combine configurations requiring different slits
# (e.g. Hs + K) since each slit requires different aperture
# definitions, and will need a new acquisition. You can however
# combine configurations which use the same slit, e.g. Hs with Jl,
# or K with Js.
set config {Hs}

# Choose exposure time parameters
# The total time for each DETECTOR run is exposure*(reads-1)*cycles
# in MRM mode. Cycles is almost always 1 for MRM observations.
DETECTOR exposure 10
DETECTOR reads 3
DETECTOR cycles 1

# How many object-sky-sky-object "quads" do you want?
set nquads 1

# Finally, specify the ORAC-DR recipe to use.
# See SUN/236 for details of the different recipes
# (can always re-reduce the data later using a different recipe).

set oracrecipe "POINT SOURCE"
#set oracrecipe "POINT SOURCE NOSTD"
#set oracrecipe "FAINT POINT SOURCE"
#set oracrecipe "FAINT POINT SOURCE NOSTD"

#=====
# DO NOT EDIT BELOW HERE
#=====

```

```

=====
#
# Send telescope to axis A
#
TELESCOPE control ON
TELESCOPE axis A -nowait
#
# Set Object name and reduction recipe
#
DETECTOR recipe ${oracrecipe}
DETECTOR object ${objname}

foreach c $config {

  #
  # Set up instrument
  #
  MOVEspectro $c

  DETECTOR startgroup

  set ntot [ expr $nquads * 4 ]

  # Cycle through the No. of quads

  for {set ii 0} {(${ii} < $nquads)} {incr ii 1} {

    set count [ expr 1 + 4 * ${ii} ]
    TELESCOPE axis A
    MESSAGE " _____ STARTING EXPOSURE $count of $ntot"
    DETECTOR run OBJECT

    #
    set count [ expr 2 + 4 * ${ii} ]
    TELESCOPE axis B
    MESSAGE " _____ STARTING EXPOSURE $count of $ntot"
    DETECTOR run OBJECT

    #
    set count [ expr 3 + 4 * ${ii} ]
    TELESCOPE axis B
    MESSAGE " _____ STARTING EXPOSURE $count of $ntot"
    DETECTOR run OBJECT

    #
    set count [ expr 4 + 4 * ${ii} ]
    TELESCOPE axis A
    MESSAGE " _____ STARTING EXPOSURE $count of $ntot"
    DETECTOR run OBJECT

    #
  }

  DETECTOR endgroup
}

TELESCOPE axis A
TELESCOPE control OFF

=====

```

B.2 AAO_Flat_Arc.tcl

This is a Sequence for taking arcs and flats in each desired setup.

```

source /export/home/aatobs/AAO_Inc.tcl

*****
# AAO_Flat_Arc.tcl
*****
# Sequence for taking arcs and flats in each desired setup.
#
#
# Revision history:
#
# 26 July 2002: First draft (SDR).
# 29 July 2002: First draft copied from Nod Along Slit K(CGT).
# 30 July 2002: Editted to make instrument setup and telescope
#                moves in parallel at start. (CGT).

```

```

# 10 July 2003: Modified to use AAO_Inc to handle spectroscopic
#               configs and focus. Now only one script for
#               all configs.
# 5 Sep 2003:   Adapted from AAO_Nod_Alone_Slit.tcl (SDR)
# 11 Jul 2004: Replace some messages with pauses, to ensure observer
#               turns lamps off. (SDR)
#
# Start by loading the default exposure configuration
#
#
#=====
# OBSERVER: Check, and if necessary modify parameters here
#=====
#
PAUSE "If you want to do any darks in the next 3 hours, abort this sequence
and do them first!"

# Enter configuration to use. Must be one of: Js Jl J H Hs K Ks
# See AAO_Inc.tcl to see what precise configurations these
# names give you.
#
#set config {Jl Hs Js K}
#set probey {637 637 618 618}
set config {J_cont}
set probey {637}
# Set up for flats
DETECTOR config MRM NORMAL

# Choose exposure time parameters
#
# The total time for each DETECTOR run is exposure*(reads-1)*cycles
# in MRM mode. Cycles is almost always 1 for MRM observations.
#
DETECTOR exposure 1.5
DETECTOR reads 21
DETECTOR cycles 3
MESSAGE "Turn Domeflat lamp OFF, drive Guide Probe out"
#
foreach f $config {
  MOVEspectro $f
  DETECTOR object "$f quartz flat lamp OFF"
  DETECTOR recipe REDUCE_DARK
  PAUSE "  Turn quartz OFF  When ready press CONTINUE"
  DETECTOR run DARK

# Now turn the quartz lamp on

  DETECTOR object "$f quartz flat lamp ON"
  DETECTOR recipe REDUCE_FLAT
  PAUSE "  Turn quartz ON FULL  When ready press CONTINUE"
  DETECTOR run FLAT
}
PAUSE "  Turn Domeflat lamp OFF. When ready press CONTINUE"

#
# Do arcs AFTER all flat are completed. Residual images from arcs
# otherwise screw up flats otherwise.
#
DETECTOR config DRM NORMAL
DETECTOR exposure 1
DETECTOR cycles 5
#
foreach f $config e $probey {
  MOVEspectro $f
  DETECTOR object "$f arc Xe lamp OFF"
  DETECTOR recipe REDUCE_DARK
  PAUSE "Guide Probe 2 to 200, $e. Xe lamp OFF  When ready press CONTINUE"
  DETECTOR run DARK

# Now turn the Xe lamp on

  DETECTOR object "$f arc Xe lamp ON"
  DETECTOR recipe REDUCE_ARC
  PAUSE "  Turn Xe lamp ON!!!  When ready press CONTINUE"
  DETECTOR run ARC
}

PAUSE "  All done! Please turn off all lamps, close mirror covers, and park
Probe 2  When ready press CONTINUE"

```

```
=====
```

B.3 AAO_Spec_Camfoc.tcl

A sequence for obtaining camera focus images.

```
source /export/home/aatobs/AAO Inc.tcl
#
#
#*****
# Acquire camera focus sequence for spectrscopy setups.
# You should have set up the illumination to be
# OK in 0.6s through the J&K filters.
#
#
#*****
# Start by loading the default exposure configuration
#
DETECTOR config DRM_NORMAL
#
# ENTER OBJECT NAME
#
DETECTOR object CameraFocus

set irfilt { Js  J  J1  H Hs K Ks}
set exp    { 0.6 0.6 0.6 1 1 1 1}
#
# Use goes to indicate how many times to repeat the above
# sequences. This is mostly useful for focus sequences where
# telescope motions are being done.
set goes { 1 }
#set goes { 1 2 3 4 5 6 7 8 9 10}
set flat "yes"
#
# Take some flats?
#
#
# Take the focus sequences
#
PAUSE "Send Guide probe 2 to 195,628. Turn Xe Lamp on."
foreach g $goes {
    foreach f $irfilt e $exp {
        DETECTOR exposure $e
        DETECTOR cycles 5
        MOVEspectro $f
        DETECTOR object "CameraFocus $f"
        DETECTOR recipe QUICK LOOK
        DETECTOR startgroup
        for {set ii 275} {($ii < 650)} {incr ii 25} {
            IRIS2 focus $ii
            DETECTOR run
        }
        DETECTOR endgroup
    }
    PAUSE "Ready to start next focus sequence? Click continue"
}
}
```