Keck Adaptive Optics Note 1232

RTC Upgrade: Operations Software System

Sam Ragland, Peter Wizinowich & Sylvain Cetre; 9/6/2018; Updated on 01/12/2020
(with input from, Charlotte Bond, Kelleen Casey, and Jason Chin)

Contents

1. Introduction

The real-time controllers (RTC) of the Keck I and Keck II adaptive optics (AO) systems are being replaced with GPU-based high-performance systems (KAON 1214.) This document summarizes the necessary changes for the Operations Software System (OSS) of the upgrade. The upgrade will provide the necessary flexibility for the future AO upgrades.

The RTC is one of the most crucial components of the AO system. The RTC gets wavefront aberration information from sensors (tip/tilt and higher-order wavefront sensors) and applies corrections to actuators (tip/tilt and deformable mirror) in real-time. The upgrade will be first implemented on the Keck I AO and then on the Keck II AO.

The new RTC will accommodate the existing operational mode inputs including the Shack Hartmann wavefront sensor (SH WFS) SciMeasure camera (CCD39), the STRAP and TRICK tip/tilt sensors, the transitional RTC mode pyramid wavefront sensing (PyWFS) SAPHIRA camera and the future operational mode inputs being designed for the KAPA (Keck All sky Precision Adaptive optics) and FIU (Fiber Injection Unit) projects. The new RTC accommodates the existing control devices including the tip/tilt mirrors (DTT & UTT) and the Xinetics deformable mirror (DM) actuators for the current operational or transitional modes, and a MEMS for one of the KPIC near-future modes. This document focuses on the operations software changes for the existing operational modes and the one under transition (PyWFS).

The main changes to the AO system as part of the upgrade are (1) real-time computer systems – both hardware and software, (2) new WFS cameras (OCAM2K) replacing the existing ones, (3) associated infrastructure changes such as cooling, environment sensors, etc., and (4) associated operational software changes. This document addresses the last item at a high level. The operational software changes for the upgrade will be implemented under ECR 3080.

2. Scope and Applicability

The OSS is a fairly well-defined subsystem that resides outside of the other AO software subsystems such as RTC, Motion controls System, Camera System, etc. Hence, in principle, the RTC upgrade should be transparent to the OSS at least for the existing operational modes.

We make two assumptions here: (1) The names and the data structures of the EPICS keywords and the RTC telemetry channels will be retained for the existing operational modes, and (2) no new operational tools are necessary to operate the new RTC. Under these assumptions, the changes to the existing operational modes are relatively minimal.

The OSS for the new operational modes, developed as part of the KAPA project, will require significant modifications to the existing OSS. This includes developing new software tools in some cases. The design of the operational software for the new modes for KAPA will be covered in the KAPA Operational Software Design Document.

The observatory is migrating away from IDL in favor of Python. In support of this policy, we identify potential opportunities for the transition in the OSS design for the RTC upgrade and assign a name starting with the letter 'O' (e.g. ***O-01***) for tracking purposes.

After careful review of the existing IDL and Python codes, we decided to defer these opportunities to the KAPA OSS design phase as at least some parts of the Python codes are expected to mature by then during the PyWFS facilitization phase. Similarly, we defer an opportunity to upgrade the FSTsequencer to the KAPA OSS design phase. Only one opportunity that we would like to take on at this time is some cleanup of the IDL routines/functions (O-12.) A schematic diagram of the interface between OSS and the rest of the observatory sub-systems is shown in Figure 1. The new or significantly modified components are marked as orange blocks.

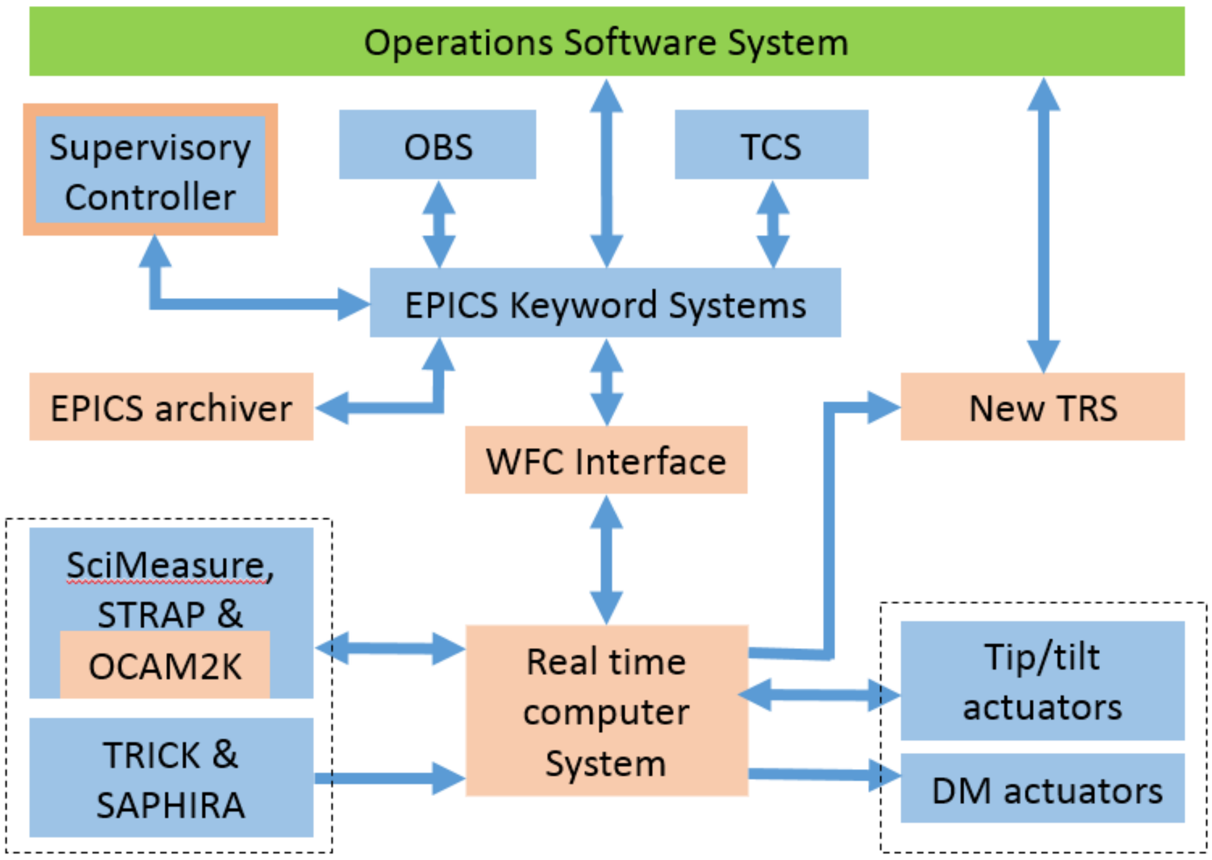


Figure 1: A schematic diagram of the interface between OSS and the rest of the operational subsystems. The new or significantly modified components are shown in orange. For clarity, the interaction between the RTC and most of the subsystems are not shown.

3. The Main Changes to the Operational System

For completeness, the main changes as part of the RTC upgrade are listed below.

- New computational engine (both hardware and software) - MG.
- New Telemetry server and associated interface - MG
- New RTC IOC and associated interface on a new physical Linux server (VM is the backup option)
- New EPICS archiver and associated interface.
- Merging of the EPICS IOC for the PWS with the RTC IOC.
- Modifications to RTC configuration files to accommodate the new operational modes.
- Supervisory Controller (SC) changes for the new RTC.
- Replacing the existing SH WFS camera with OCAM2K camera.
- New EPICS keywords for the new WFS camera (OCAM2K) and new operational modes such as LTAO, PWS, FIU-Coronagraph, etc.
- Updating the new RTC to include PyWFS (PyWFS currently uses a standalone RTC.)
- SC changes for the OCAM2K and SAPHIRA.
- Adding a high-density MEMS DM for the FIU project.

4. Overview of the Operational Software System

The OSS consists of six major components. A block diagram of the OSS is shown in Figure 2. A brief description of the six components is given below from the perspective of the changes needed for the RTC upgrade. More details on the affected EPICS keywords are presented in Appendix B.

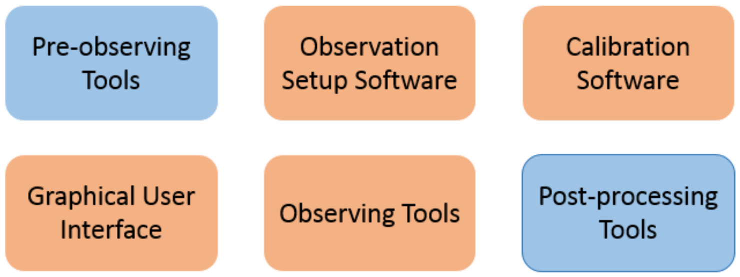


Figure 2: The major components of the Keck AO Operational Software. The software components with significant changes are shown in orange.

The main changes are:

- Updates to the shell scripts for rebooting & initializing the RTC.
- Defining the 40 x 40 (or 80 x 80 in non-binning mode) subarray location of the OCAM2K.
- Updates to the WFS camera setup for OCAM2K.
- Updates to the PyWFS operations software still under transition.

In addition, we anticipate significant troubleshooting efforts needed to validate the operations software for the RTC upgrade.

4.1 Pre-observing Tools

The pre-observing tools include (1) AO Guide Star tool for target selection including the upgraded AO Guide Star tool for TRICK, (2) Keck AO web pages for the observers, and (3) AO science instrument pre-observing web pages. The last two items require an update in terms of performance numbers and the limiting magnitudes of the operational modes. This will be performed at the end of the on-sky performance validation of the RTC upgrade.

4.2 Observation Setup Software

This software includes startup and shutdown scripts, setup bench, setup cameras, wavefront controller, etc. The startup and shutdown scripts specific to the RTC hardware and software will be modified for the new RTC. Similarly, the IDL routine for the SciMeasure camera configuration needs to be updated for the OCAM2K camera.

The following two shell scripts require an update or elimination for the new RTC:

- resetNGWFC - to reset the VME (/kroot/rel/epcom/default/bin/solaris/resetVME) will be removed.
- /export/home/trs/default/ctl on the k#ngwfc-trs (to stop/start the trs server) will be updated.

The relevant EPICS keywords in the AO service to initialize and setup the WFC are:

- wcstat: WFC status keyword; gives an integer number from 0 to 4 representing the status in the following order: ['OFF','STANDBY','INITING','NORMAL','FAULTED']
- wcoff: WFC power off keyword; to set to 1 to power off.
- wcboot: WFC boot keyword; to be set to 1 to boot the WFC.
- wcinit: WFC initialization keyword; to be set to 1 for initialization after a boot.
- wcoper: WFC setup keyword; to be set to 0 or 1 refering to standby or normal mode 1.
- wcup: WFC status keyword; gives 0 or 1 to represent the state (off or on)

The relevant EPICS keywords in the AO service associated with the STRAP sensor are:

- ststate to check STRAP status
- stinttim to set STRAP integration time
- stbkgnd to set the currently used STRAP background values
- stgate to set the STRAP gate (open or close)

The associated IDL tools are listed below.

WFC setup & initialization routines/functions:

- initwfc.pro – initializes the WFC
    - Used by calsetup.pro, faultdetector.pro, rebootwfc.pro, setwcoper.pro & start_initwfc.pro
- setupwfc.pro - setup the parameter for the CCD 39 & the NGWFC
    - Used by calsetup.pro, cameragui.pro, makedarks.pro, ~~setngsao_targ.pro~~ & setframerate.pro
- setwcoper.pro - sets the WFC to 0 or 1 state
    - Used by initwfc.pro, loadconfig.pro, setupwfc.pro, setwfcparms.pro & wfcttinfmat.pro
- rebootwfc.pro by start_rebootwfc.pro
    - Uses setwfcparms.pro
    - Uses the shell scripts: resetNGWFC & ctl stop

  The reboot script will be modified as follows: set "wcoff" to 1 and wait for 5 seconds. Then set "wcboot" to 1 and wait for "wcstate" for the status to change to "LOADED". Then, set "wcinit" to 1 and wait for "wcstat" to change to "NORMAL". KAON is updated.
- setwfcparms.pro
    - Used by calsetup.pro & rebootwfc.pro

O-01: The IDL setup tools for the RTC may be written in Python and called from IDL.

WFS camera setup routines/functions:

- wfsconfig.pro - sets WFS platescale
    - Used by aoacq.pro, center_wls.pro, configgui.pro, loadconfig.pro, setngsao_targ.pro, wfs_cal.pro & start_wfsconfig.pro
    - Uses check_and_modify.pro (used by setngsao_targ.pro as well)
- setframerate.pro – sets the WFS framerate (binning, program & gain)
    - Used by acamtool.pro, aoacq.pro, setngsao_vmag.pro, wcs_focus.pro & wfs_cal.pro
- cameragui.pro
    - Uses calculateframerate.pro
- calculateframerate.pro – calculate allowed WFS framerate, repetition, program for given binning & demand framerate – script this for KAPA
    - Used by cameragui.pro & setframerate.pro

  OCAM2K uses a single clock and the full array is read. The RTC extracts the 80 x 80 or 40 x 40 subarray for the unbinned or 2x2 binned modes, respectively. The subarray location is controlled through EPICS keywords, "WS1OROIPOS" and the binning is set through the keyword, "OCAM2K-BIN" - 0 for no binning and 1 for binning modes. The camera has 8 pre-amp outputs. The unbinned and binned modes require separate bias settings. The wavefront sensor frame-rate is the only parameter to be changed as per the flux level. This can be done without the need to turn off and on the camera.

  The existing EPICS keywords, wssmprg & wssmrep are not relevant for the OCAM2K.

O-02: The OCAM2K configuration routines could be written in Python instead of IDL. The routines could be written in a generic way to accommodate the SAPHIRA as well.

STRAP setup routines/functions:

- setup_strap.pro - sets SFW, APD temperature, and integration time of STRAP for a given tip-tilt star.
    - Used by aoacq.pro, lbwfscals.pro & setup_lbwfs.pro
    - Uses read_strap_par.pro & set_strap_gain.pro
- init_strap_params.pro – sets strap gain based on the psd - We should rename this tool as set_strap_params.pro & update aoacq.
    - Used by aoacq.pro

        - Uses strap_calcerror.pro & set_strap_gain.pro
    - set_strap_gain.pro
        - Used by bw_widget.pro, init_strap_params.pro & setup_strap.pro
    - set_strap_int.pro - sets strap integration time and scales the existing background
        - Used by setup_strap.pro - should include this routine in setup_strap.pro for clarify.
    - strap_calcerror.pro
        - Used by bw_widget.pro & init_strap_params.pro
        - Uses strap_noise.pro – We should append this function to the strap_calcerror.pro.
- resetaodcs.pro by start_resetaodcs.pro

General setup routines/functions:

- Switchtofiber.pro
- Switchtostar.pro
- Switchtohalt.pro
- fsmoffload.pro
    - Used by aoacq.pro, dm_wls_reg.pro, wfcttinfmat.pro, wfs_cal.pro & wfsstatus.pro
- loadlbcog.pro
    - Used by check_lbwfs_reg.pro, lbimgacq.pro, lbsetup.pro, lbwfscals.pro, lbwfscals_subs.pro & setup_lbwfs.pro
- restoreaostate.pro
    - Used by checkopenwindow.pro & checkzenithwindow.pro
- utt_zero.pro
    - Used by acamtool.pro, aoacq.pro & auto_man_acq.pro

4.3 Calibration Software

This includes (1) camera calibrations (refered to as *Calibration #1* in this document and similarly, the other calibrations listed below are refered with the corresponding calibration number), (2) pupil registration, (3) pointing origin calibration, (4) non-common path aberration estimation, (5) focus calibration, (6) WFS centroid origin calibration, (7) interaction matrix calibration, and (8) DM influence function calibration. These IDL routines need to be modified for OCAM2K. The associated IDL tools are listed below.

Standard WFS/STRAP calibration routines/functions:

- wfcidl.pro – the top-level calibration GUI
- calsetup.pro – sets the AO bench for AO calibrations or daytime tests.
    - Uses check_au_kwds.pro, which_tel.pro, xmessage.pro, sense_set.pro, set_fcs_for_inst.pro

- lightcontrl.pro
- take_strap_bkg.pro
    - Used by aoacq.pro & wfsstatus.pro
- take_wfs_bkg.pro
    - Uses makebkgnd.pro – should append this routine to take_wfs_bkg.pro
    - Used by acamtool.pro, aoacq.pro, center_wls.pro, dm_wls_reg.pro, makedarks.pro, take_strap_bkg.pro, wfs_cal.pro & wfsstatus.pro

O-03: A generic Python tool could be written that work for the OCAM2K & SAPHIRA camera calibrations (*calibration #1*.)

- dm_wls_reg.pro

O-04: The pupil registration of the SH WFS (*Calibration #2*) could be rewritten in Python with a possibility of using the tool for the pupil stabilization in the PyWFS configurations.

- dm_control.pro
    - Uses dm_flatten_edge - We should append this routine to dm_control.pro for clarity.
- image_sharpening.pro
    - Uses set_mgs_params.pro, set_nirc2_params.pro, set_osiris_params.pro, keck_mgs.pro, gen_act_coor.pro & imshp_auto.pro
    - keck_mgs.pro
        - Uses gskeck.pro, noise_filter.pro, frescale.pro & zern_fit
    - gskeck.pro
        - Uses phaseunwrap2d.pro
- move_sfp_to_fiber.pro
    - Used by center_wls.pro, dm_wls_reg.pro & wfs_cal.pro
- update_fsm_origin.pro
    - Uses loadfsmori.pro
- zernike_control.pro
    - Used by cog_sharp.pro & man_sharp.pro
- get_dm_phase.pro
    - Used by dm_control.pro & read_wyko.pro
- wfs_cal.pro – generates fast WFS cog file(s) for a given DM shape
    - Used by calsetup.pro, dm_wls_reg.pro, set_fcs_for_inst.pro & wfsconfig.pro

O-05: The Python tool to take PyWFS centroid origin calibration (*calibration #6*) may be modified for the SH WFS configurations.

- acamtool.pro - LGS operation, daytime calibration & diagnosis
    - focus_lta: uses acam_cents

LBWFS Calibration routines/functions:

- lbwfscals.pro – does LBWFS calibrations (LBWFS setup, DM to lenslet registration and the generation of LBWFS cog files for a given DM shape.)
    - Used by lbimgacq.pro, lbwfscals_subs.pro & start_lbwfscals.pro
    - Uses move_dev.pro, lbwfscals_subs, lbsreg, writelbcog, loadlbcog, dialog_message, xmessage & wait_for_complete, align_tss, dm_waffle, check_lbwfs_reg, readmr
- lbwfscals_subs.pro – routines specifically for lbwfscals.pro
    - Contains record_img, record_dark, ~~record_bkgrd, gen_lbwfs_reg_matrix, save_xyoffset, calc_dm_lbwfs_reg, apply_waffle~~, align_lbwfs & widupdate_text.
        - The crossed-out routines/functions are not used. We should move them to the development folder as part of the IDL cleanup.
- lbsreg.pro – calculates the motion needed for LBS to be consistent with the TSS and makes the move (used for nighttime ops as well)
    - Used by aoacq.pro, lbimgacq.pro, lbwfscals.pro & setup_lbwfs.pro
- lbunstack.pro – extracts LBWFS SH spots as a 16x16x304 array (used for nighttime ops as well)
    - Used by calclbsx0y0.pro, check_lbwfs_reg.pro, lbimgacq.pro & lbwfscals_subs.pro
- lbimgacq.pro – LBWFS image acquisition tool (used for nighttime ops as well)
    - Used by lbmngr.pro & start_lbimgacq.pro
- lbmngr.pro – LBWFS Manager (used for nighttime ops as well)
    - Used by lbimgacq.pro & start_lbmngr.pro
- lbimgproc.pro (used for nighttime ops as well)
    - Used by check_lbwfs_reg.pro, lbimgacq.pro, lbunstack.pro & lbwfscals_subs.pro
- lbimg.pro
    - Used by check_lbwfs_reg.pro, lbdrk.pro, lbimgacq.pro, lbimgproc.pro, lbmngr.pro, lbunstack.pro, lbwfscals_subs.pro, start_lbimgacq.pro & take_strap_bkg.pro
- lbcaldrk.pro – (1) takes LBWFS darks with a variety of integration times and binning, (2) saves median darks, and (3) updates dark catalogue.
    - Uses lbmeddrk, lbdrk & sxaddpar
- lbmeddrk.pro
    - Used by lbcaldrk.pro
    - Uses lbdrk.pro & sxaddpar

- lbdrk.pro
    - Used by lbcaldrk.pro, lbimgproc.pro, lbmeddrk.pro & lbsetup.pro
- lbfoc_offload2alt.pro
    - Used by aoacq.pro, lbimgacq.pro & lbmngr.pro
- setloopn.pro - We should rename this routine as lbsetloopn.pro for clarity & update lbimgacq.pro.
    - Used by lbimgacq.pro

IDL tools for infrequent AO calibration:

- wfcttinfmat.pro – calibrates TT interaction matrix
- strapmatcal.pro – calibrates STRAP control matrix
- ***O-06:*** Generic Python tools could be written that work for the SH WFS, STRAP & the PyWFS interaction matrix calibration for the tip/tilt controller (*calibration #7*.)

  O-07: A generic Python tool could be written that work for the SH WFS & the PyWFS DM influence function calibration (*calibration #8*.)
- wcs_focus.pro
- center_wls.pro
- definepos.pro - defines the pointing origin
    - Used by acamtool

4.4 Graphical User Interface

This category covers telescope, AO, and instrument tools primarily for visualization purposes during AO observations. This includes the MAGIC GUI, and telescope status GUIs, AO monitoring tools such as tip/tilt graphs, WFS/STRAP/TRICK/SAPHIRA intensity displays, TBAD, alarm handler, etc. The alarm handler tool needs to be modified to incorporate the new EPICS keywords. Also, the WFS intensity display (both the time-averaged and the raw intensity versions) routines need to be modified for the OCAM2K.

O-08: The Python tools to display the SAPHIRA camera intensities could be generalized to accommodate the OCAM2K camera.

The following are the IDL tools used for AO status monitoring:

- check_au_kwds.pro: Used by calsetup, faultdetector, resetaodcs & which_tel
- check_device.pro: Used by calsetup, faultdetector, resetaodcs & check_dev
- check_fcs.pro, check_fsm.pro, check_tss.pro, checkwfc.pro & detectwfcfault: Used by faultdetector
- check_au_kwds.pro, check_device.pro, check_dev.pro

- rawimdisp.pro - needs update for OCAM2K

> ***O-9:*** The raw image display tool could be written in python instead of modifying the IDL tool.

- start_wfsstatus.pro -> wfsstatus.pro
- start_strapstatus.pro -> strapstatus.pro
- check_lbwfs_reg.pro
- checkzenithwindow.pro (for space com in the spiral window)
- start_faultdetector.pro -> faultdetector.pro
- trickmanager.pro

4.5 Observing tools

This includes the IDL tools (1) aoacq to acquire the target on the WFS and the science instruments, (2) AO optimization tools, (3) FST sequencer to acquire the laser on the WFS, (4) reconstructor tool, (5) seeing estimation tool, (6) observing tools/scripts such as QACITS for the coronograph, (7) instrument/AO setup and observing scripts, and (8) EPICS archiver visualization tool.

The relevant IDL routines/functions are:

- nighttimescript.pro - configure AO for nighttime operation
- aoacq – target acquisition tool
  - laser_propagate.pro & laser_shutter.pro
  - effectivermag.pro – converts the stars R mag. Into effective R mag. for WFS & STRAP
- acamtool: LGS operation, daytime calibration & diagnosis
  - focus_lta: uses acam_cents
- atmogui
  - Uses estimateseeing.pro
- Reconstructor
  - recongui.pro
  - start_reconproc.pro -> reconproc.pro

- Spacecom tools
  - startup_spiral.pro -> spiral.pro > spiral_event.pro
  - spacecomclose.pro by checkopenwindow.pro & checkzenithwindow.pro
  - spiral_event.pro by checkzenithwindow.pro & startup_spiral.pro
  - spiral_picktarg.pro by spiral_event.pro
  - spiral.pro by auto_man_acq.pro, checkzenithwindow.pro, spiral_event.pro, spiral_picktarg.pro & startup_spiral.pro

- bw_widget.pro
    - Uses compensator.pro

The relevant shell Scripts are listed below:

- checktrsDiskSpace
- resetNGWFC
- make_dcs_real
- rpcKey_server -l cache_service -s idl
- caput & caget
- killRecon
- killFaultDetector
- /export/home/trs/default/ctl stop
- wfcidl.csh
- faultdetector.csh
- endofnight.csh
- trickmanager.csh

O-10: A generic reconstructor tool could be written in Python that work for the SH WFS & PyWFS.

O-11: FST sequencer may be expanded to include additional sensors (WFS, STRAP, TRICK, PyWFS, and ACAM) for further automation and reduce the load on Aoacq. This upgrade has the potential to improve the operational efficiency for the LTAO configuration with multiple laser guide stars. Essentially, a top level sequencer (aoOper) in the same platform as the FST sequencer needs to be developed to deal with the telescope, and the AO. This sequencer would in turn call the FST sequencer to propagate the laser and another sequencer for the NGS case. The aoOper would be a first step towards developing an observing sequencer to replace the aoacq.

4.6 Post-processing tools

The post-processing tools include the instrument fits header, KOA archival tools, instrument data pipeline software, and the PSF reconstruction tools including the AO telemetry extraction software. The host references of the telemetry extraction need to be updated for the existing operational modes and the tool verified during the RTC integration.
The primary changes to the TRICK system are (1) simplification in sensor configuration in the RTC (i.e. the weighting factor will be removed from the new system), and (2) simplification in the readout for the multi-ROI case.

4.7 Miscellaneous

The TRICK specific operational IDL tools are located at /home/k1bsao/mvandam/TRICKScripts/. The primary main routines are trickmanager, toda, and updated aoacq. The TRICK software needs to be run from k1aoserver-new.

There are many secondary IDL routines/functions that are listed in the Appendix A for clarity. The EPICS keywords potentially affected in this upgrade are identified for easy troubleshooting and listed in the Appendix B. Also, the IDL codes that directly access the EPICS channels are listed in Appendix C.

A new RTC telemetry channel containing sub-aperture intensity in full framerate will be available in the new system to improve WFS noise estimation for point spread function (PSF) reconstruction (KAON 1227.)

The telemetry extraction tool and the PSF reconstruction tools will also require an update to make use of the raw images full framerate. The PSF-R project is expected to make these changes.

> ***O-12:*** We propose some cleanup of the operational IDL folder as part of the RTC upgrade. Details are presented in the Appendix D.

5. Pyramid Wavefront Sensor (PyWFS)

The PyWFS specific operational tools are located at /home/prtc/dev/prtc/python and /kroot/rel/ao/prtc/default/.

The PyWFS uses the existing IDL setup tool to (1) setup the AO bench, (2) image sharpen with the dichroic, and (3) the start of the night and end of the night scripts for the AO.

The PyWFS uses Python tools to (1) setup the PyWFS, (2) perform the modulator alignment, (3) perform pupil registration, (4) perform interaction matrix calibration if needed, (5) perform focus optimization, and (6) save reference slopes for the six modulation configurations.

The operational GUIs include (1) loop control GUI (loopCtrl.py) that is currently an engineering GUI with information such as sub-aperture intensities, estimated slopes, wavefront residuals, etc., (2) overall control GUI (Pyramid Quick Access), and (3) SAPHIRA image display tool (imSumDisp.py). There are tool to perform the health checks of the SAPHIRA camera, modulator range, optical alignment, and the optical gain. The data logging tools include archiving the EPICS keywords using EPICS archiver and writing telemetry channels as FITS file per channel on-demand basis.

Setup tools:

- Shell scripts to set the switchyard optics & loads RTC reference files: prtcSetupIns -> setupInstrument
    - Associated Python routines: loadDMRef.py, loadDTRef.py, loadSlopesRef.py
- Shell scripts to move the switchyard optics: prtcInsertDichroic, prtcInsertFlatMirror, prtcRemoveOptic & prtcCheckOptics
- Shell scripts for initialization: setIMZernike, prtcInitDM & prtcInitDMFocMod
- Shell scripts for SAPHIRA camera configuration: setupBeagle, showPbServer
    - The associated Python script: initCam.py
- Python routine to setup reference slopes:

    - resetSlopesRef.py & setSlopesRef.py

Calibration tools:

- Modulator alignment: prtcSetModulation.py
- Pupil registration/Mapping: identifyPupils.py
- Interaction matrix calibration: calibrationZernike.py
- Save reference slopes: saveDMRef.py & saveDTRef.py

Operational tools:

- Python tool to access PyWFS: prtcQuickAccess.py
- Shell script for camera setting for target brightness: prtcSetupForMag
    - Associated Python tool: magnitudeEstimator.py
- Shell script for taking sky background measurements: prtcSkyBackground
    - Associated Python tool: takeBGFF.py (used for taking bench background measurements as well.)
- Script for enabling/disabling the reconstructor process: prtcTurnOnReconstructor & prtcTurnOffReconstructor
    - Associated Python routines: turnOnReconstructorMgr.py & turnOffReconstructorMgr.py
- Python routines for reconstructor computation: reconstructorMgr.py, pwfsTools.py & pupilTracker.py
- Python routines for Image display: imDisp.py, imshmDispDm.py, imshmDisp.py, imshmDispSmall.py, imshmfsDisplayFrame.py & imshmDispFull.py

6. Integration Tests

The operational system will be tested in the daytime using the AO whitelight source in preparation for the on-sky engineering for the RTC upgrade. We expect to test most of the functionalities and calibration files through daytime tests. The remaining functionalities and AO performance will be tested through on-sky engineering. This is in addition to the low-level RTC tests that will be performed during the integration.

The main daytime tests for the OSS are:

- Setup AO bench (including the RTC) for all operational modes including ACAM & LBWFS servers and verify that the positioning of the stages.
- Configure the RTC for the operational modes using the operational tools and verify the settings.
- Configure sensor(s) including the SciMeasure camera for the operational modes and verify their operation.
- Perform pupil registration (pupil stabilization in the case of PyWFS) for different operational modes.

- Perform OCAM2K calibrations and verify the calibration files and make sure that the new RTC uses the appropriate calibration data. (Perform these calibrations using SciMeasure Camera if it becomes necessary.)
- Perform PO calibrations for different configurations and verify using aoacq in simulation mode.
- Perform non-common path aberration calibration for different modes/instruments and verify the image quality.
- Perform OCAM2K, SAPHIRA & TRICK focus calibrations and verify that the calibrations gives diffraction-limited images on the science cameras. (Perform these calibrations using SciMeasure camera, if necessary.)
- Perform influence function calibration and compare the matrix with the data from the most recent calibration.
- Perform interaction matrix calibration for the OCAM2K, STRAP, TRICK & SAPHIRA configurations and compare the matrix with the data from the most recent calibration.
- Perform WFS centroid origin calibration for the SH WFS (OCAM2K), LBWFS & PyWFS configurations and compare the results to the previous calibrations. (Perform these calibrations using SciMeasure camera, if necessary.)
- Verify the monitoring GUIs such as OCAM2K, SciMeasure, STRAP, and DM actuator displays, tip/tilt graphs including the UTT, SC GUI, fault detector, and AO alarm handler.
- Verify the control GUIs such as Maori, AO acquisition, ACAM Tool, FST sequencer, LB img acq, LB manager.
- Verify that there are no dropped frames in the telemetry channels.
- Test the night-time setup, end-of-night, switch-to-fiber, and switch-to-star scripts.
- Verify that the QACITS software runs using the white light source.

The main nighttime tests are:

- Verify again the night-time setup, end-of-night, switch-to-fiber, and switch-to-star scripts.
- Verify the AO acquisition, reconstructor, and ACAM tools.
- Verify again all the monitoring GUIs such as OCAM2K, STRAP, and DM actuator displays, tip/tilt graphs including the UTT, SC GUI, fault detector, and AO alarm handler.
- Verify the performance optimization tools such as the bandwidth widget and optical gain tracking tool.
- Verify the operation of the performance verification tools such as the Strehl tool and the seeing tool.
- Verify all the laser operations tools such as FST sequencer, LUI, TBAD, LB img acq, LB manager.
- Verify the QACITS software by taking a full sequence, post-processing the data and comparing the results with expected contrast curves.
- Perform on-sky AO functionality tests making sure that the tools perform as intended.
- Perform on-sky AO performance tests and compare the results with expected performance.

- Verify again that there are no dropped frames in the telemetry channels.
- Regression test the post-observing tools.

The priority for the night time tests is to use the OCAM2K for the SH WFS configuration. The fall back is to test the SciMeasure camera.

7. Compliance Matrix

Name	Tracking ID#	Description	Design Compliance by Section
AO Acquisition Software	1133	The CSW shall modify the acquisition tools (IDL) to support the operating modes for the RTC.	Section 6
Reconstruction Matrix	5432	The operation software shall compute the reconstruction matrix for all the operation modes. *(Note SCE: this includes the pyramid mode, but will prepare the software for the future LTAO reconstructor, too.)*	Section 6
OBS Par Files	5430	The operation software shall use the existing file format of the configuration files for all the operation modes. *(Note SCE: all the new modes, include pyramid, will follow the same rule.)*	Section 6
Software Architecture	5431	The operation software shall have a faultDetector tool that monitors all the devices and all the operation modes. *(Note SCE: this includes monitoring the pyramid)*	Section 6
Wavefront Controller Functionalities	5429	The operation software shall provide calibration of the optical devices for all the operation modes. *(Note SCE: this requirement includes the calibration of the pyramid FSM.)*	Section 6

In addition, the system operations requirements (Section 3.1), system documentation requirements (Section 3.6), and performance requirements (ref) of RTC System Requirements (KAON 1229) are also relevant here but are not listed in the above table.

8. Summary

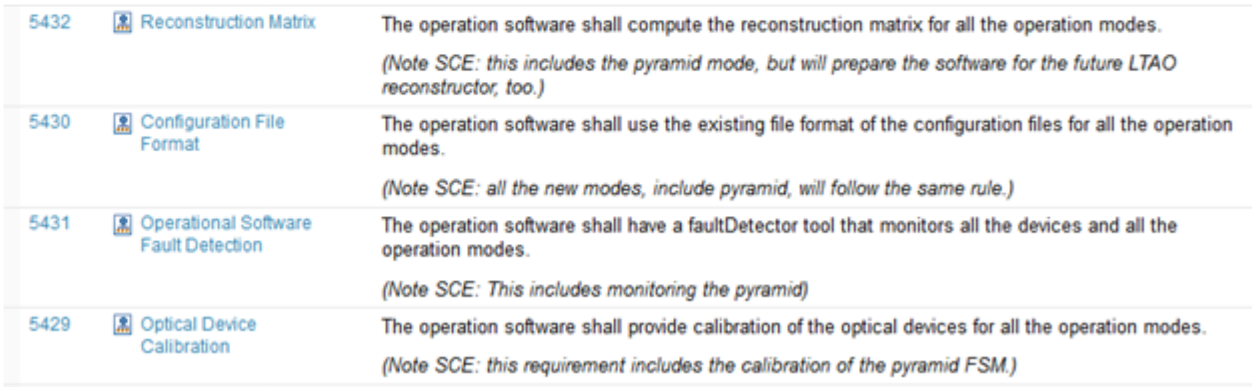
This document presents the operations software changes required for the RTC upgrade for the existing operational modes and the one under transition (PyWFS). The design of the operational software for the new modes for KAPA will be covered in the KAPA Operational Software Design Document.

The risks for the RTC upgrade for the PyWFS could be reduced significantly by performing the following four tasks in parallel perhaps through the PyWFS facilitization project.

- The reconstructor generation process for the PyWFS is currently embedded in the PyWFS RTC. This non-realtime task could be taken outside of the RTC and sky tested.
- Change the file formats the same as the existing operational modes.

- Update the faultDetector tool for the PyWFS hardware.
- Follow the calibration and operational procedures of the current FSM for the PyWFS FSM integration.

Appropriate software requirements are added to the RTC requirements (a screenshot of these requirements in DOORS is shown below.)



5432	Reconstruction Matrix	The operation software shall compute the reconstruction matrix for all the operation modes. *(Note SCE: this includes the pyramid mode, but will prepare the software for the future LTAO reconstructor, too.)*
5430	Configuration File Format	The operation software shall use the existing file format of the configuration files for all the operation modes. *(Note SCE: all the new modes, include pyramid, will follow the same rule.)*
5431	Operational Software Fault Detection	The operation software shall have a faultDetector tool that monitors all the devices and all the operation modes. *(Note SCE: This includes monitoring the pyramid)*
5429	Optical Device Calibration	The operation software shall provide calibration of the optical devices for all the operation modes. *(Note SCE: this requirement includes the calibration of the pyramid FSM.)*

Appendices

A. Secondary IDL Routines/Functions

Read/write routines/functions:

- readbackrmag.pro
    - Used by aoacq.pro, strapstatus.pro & wfsstatus.pro
- readbkgnd.pro
- readconfig.pro
- readdm.pro
    - Used by dm_clear.pro, dm_control.pro, dm_cross.pro, dm_flatten_edge.pro, dm_waffle.pro, estimateseeing.pro, image_sharpening.pro & zernike_control.pro
- readimx.pro
    - Used by reconproc.pro
- readph.pro
    - Used by dm_control.pro – should be included in dm_control.pro for clarify
- readsetup_arch.pro
- readstarlist.pro
    - Used by aoacq.pro, spiral_picktarg.pro, tss_readstarlist.pro, tss_widget_picktarg.pro & tss_widget.pro
- read_wyko_opd.pro
    - Used by get_dm_phase.pro – should be included in get_dm_phase.pro for clarity
- read_wyko.pro
    - Used by get_dm_phase.pro, image_sharpening.pro & read_wyko_opd.pro
- readlbcog.pro
    - Used by eval_subimage_disp.pro
- read_lbwfs_par.pro
    - Used by setup_lbwfs.pro
- read_strap_par.pro
    - Used by aoacq.pro & setup_strap.pro
- writecog.pro
    - Used by cent_manip.pro, cog_sharp.pro, eval_cog_update.pro, man_sharp.pro & wfs_cal.pro
- writedm.pro

    - Used by dm_control.pro, dm_flatten_edge.pro & image_sharpening.pro
- writelbcog.pro
    - Used by check_lbwfs_reg.pro, lbwfscals.pro & lbwfscals_subs.pro
- writemr.pro
    - Used by reconproc.pro

Parameter load routines/functions:

- loadbkgnd.pro by recordaosetup.pro, setngsao_targ.pro & setupwfc.pro
- loadcog.pro by aoacq.pro, calsetup.pro, center_wls.pro, cent_manip.pro, cog_sharp.pro, dm_flatten_edge.pro, eval_cog_update.pro, generate_imx.pro, lbmngr.pro, man_sharp.pro, setngsao_targ.pro & wfs_cal.pro
- loadconfig.pro by aoacq.pro
- loadflat.pro by setngsao_targ.pro & setupwfc.pro
- loadfsmori.pro by aoacq.pro, calsetup.pro, define_fsm_origin.pro, update_fsm_origin.pro
- loadmr.pro by calsetup.pro, reconproc.pro & setwfcparms.pro

Data logging routines/functions:

- recordaosetup.pro
- saveconfig.pro
    - Used by aoacq.pro
- start_systemlog.pro -> systemlog.pro
    - Uses aolog.pro & tsslog.pro
- record_lbwfs_info.pro
    - Used by lbimgacq.pro & lbimg.pro
- straplog.pro
- generate_arT.pro
    - Used by which_date.pro

General Utilities:

- abortaoacq.pro
    - Used by aoacq, laser_propagate, take_strap_bkg & take_wfs_bkg
- align_tss.pro
- dm_waffle.pro
    - Used by lbwfscals.pro, lbwfscals_subs.pro, dm_wls_reg.pro & wcs_focus.pro
- dm_clear.pro
    - Used by dm_wls_reg.pro, wcs_focus.pro, generate_imx.pro, check_lbwfs_reg.pro, lbwfscals.pro, lbwfscals_subs.pro

- zernider.pro
    - Used by image_sharpening.pro & zernike_control.pro
- zernike.pro
    - Used bw_widget.pro, cog_sharp.pro, disp2d.pro, dm_waffle.pro, estimateseeing.pro, removettp.pro, setwfcparms.pro, wcs_focus.pro & zernike_control.pro
- noisetf.pro
    - Used by bw_widget.pro, compensator.pro & optimize_gain.pro
- ttdithering.pro

General purpose routines/functions:

- atv.pro
- dm_dialog.pro
    - Used by dm_control.pro & read_wyko.pro
- center_shift.pro, centroid.pro, centgainsfp.pro
- ttmirrordynamics.pro
    - Used by compensator.pro & noisetf.pro
- tv_sub.pro by check_lbwfs_reg.pro, generate_imx.pro, lbcontrol.pro, lbdisp.pro & recongui.pro
- ttcomp.pro
- makedarks.pro
- plotObs.pro by plotObsAr.pro
- generate_arT.pro by which_date.pro
- lightcntrl.pro
- record_acam.pro by acamtool.pro & focus_lta.pro
- reduce_nirc2_data.pro by image_sharpening.pro & imshp_auto.pro
- reduce_osiris_data.pro by image_sharpening.pro & imshp_autosiris.pro
- remed.pro by acam_cents.pro, acamtool.pro & focus_lta.pro
- removettp.pro by dm_control.pro & read_wyko.pro
- sense_set.pro by calsetup.pro, dm_wls_reg.pro & wfs_cal.pro
- set_fcs_for_inst.pro by aoacq.pro, calsetup.pro, center_wls.pro, dm_wls_reg.pro & wfs_cal.pro
- setframerate.pro by acamtool.pro, aoacq.pro, setngsao_vmag.pro, start_setframerate.pro, wcs_focus.pro & wfs_cal.pro
- set_light_level.pro by center_wls.pro, dm_wls_reg.pro, wcs_focus.pro & wfs_cal.pro
- ssccd.pro by compensator.pro & noisetf.pro

- seconds_to_date.pro
- unique_file.pro
- dm_cross used by check_lbwfs_reg.pro
- getaostate.pro by checkopenwindow.pro & checkzenithwindow.pro
- gettargindex.pro by spiral_event.pro & startup_spiral.pro
- plotclosures.pro by checkopenwindow.pro, plot_z_closures.pro, spiral_event.pro, spiral.pro & startup_spiral.pro
- plot_z_closures.pro by checkzenithwindow.pro & startup_spiral.pro
- wait_for_complete.pro by align_tss.pro, center_wls.pro, centgainsfp.pro, check_device.pro, check_fcs.pro, check_fsm.pro, check_tss.pro, focustss.pro, fsmoffload.pro, lbsetup.pro, lbsreg.pro, lbwfscals.pro, lbwfscals_subs.pro, makedarks.pro, move_dev.pro, move_sfp_to_fiber.pro, resetaodcs.pro, setngsao_vmag.pro, setup_lbwfs.pro, strapmatcal.pro, take_wfs_bkg.pro, wcs_focus.pro, wfs_cal.pro & wfsstatus.pro
- expm.pro used by ssccd.pro
- ssccd.pro used by compensator.pro & noisetf.pro
- get_acam_po by acamtool.pro & focus_lta.pro
- maxloc.pro by dm_control.pro & eval_subimage_disp.pro
- move_dev.pro by center_wls.pro, cent_manip.pro, dm_wls_reg.pro, lbmngr.pro, lbsetup.pro, lbwfscals.pro, move_sfp_to_fiber.pro, set_light_level.pro, wfs_cal.pro & wfsstatus.pro
- namedposn.pro by acamtool.pro, aoacq.pro, calsetup.pro, image_sharpening.pro & lbsreg.pro
- openread.pro by aoacq.pro, centgainsfp.pro, cent_manip.pro, openwrite.pro, setwfcparms.pro, wfcttinfmat.pro & wfsconfig.pro
- openwrite.pro by centgainsfp.pro, define_fsm_origin.pro, dm_wls_reg.pro, image_sharpening.pro, lbwfscals.pro, unique_file.pro, update_fsm_origin.pro & wfcttinfmat.pro
- pixshift.pro by center_shift.pro & keck_mgs.pro
- setngsao_vmag.pro by aoacq.pro
- soundmessage.pro by aoacq.pro, copy_snd_msg.pro, faultdetector.pro, reconproc.pro & resetaodcs.pro
- trigger_diag.pro by bw_widget.pro & diagsetup.pro
- which_date.pro by generate_arT.pro & plotObsAr.pro
- which_tel.pro by aoacq.pro, atmogui.pro, bw_widget.pro, calsetup.pro, copy_snd_msg.pro, definepos.pro, get_dm_phase.pro, lbunstack.pro, readbackrmag.pro, rebootwfc.pro, reconproc.pro, set_fcs_for_inst.pro, soundmessage.pro, strap_calcerror.pro & wfsstatus.pro
- which.pro
- wobble.pro

- xmessage.pro by acamtool.pro, aoacq.pro, auto_man_acq.pro, calsetup.pro, centgainsfp.pro, check_device.pro, check_fcs.pro, check_fsm.pro, check_tss.pro, dm_control.pro, dm_dialog.pro, dm_wls_reg.pro, faultdetector.pro, focus_lta.pro, geevum.pro, imshp_auto.pro, imshp_autosiris.pro, lbcontrol.pro, lbmngr.pro, lbwfscals.pro, namedposn.pro,
- widupdate_text.pro by diagsetup.pro, lbsetup.pro & lbwfscals_subs.pro
- optimize_gain.pro by bw_widget.pro, strap_calcerror.pro & strap_calcerror.pro
- tss_widget_picktarg.pro
    - Used by spiral_picktarg.pro, ~~tss_widget_find_targ.pro & tss_widget.pro~~

Startup IDL Scripts

- start_acamtool.pro*
- start_aoacq.pro*
- start_atmogui.pro*
- start_bw_widget.pro*
- start_cameragui.pro*
- start_dmdisp.pro*
- start_endofnight.pro*
- start_faultdetector.pro*
- start_fixobs.pro*
- start_initwfc.pro*
- start_lbimgacq.pro*
- start_lbmngr.pro*
- start_lbwfscals.pro*
- start_logseeing.pro*
- start_nighttime.pro*
- start_rebootwfc.pro*
- start_reconproc.pro*
- start_resetaodcs.pro*
- start_setframerate.pro*
- start_strapstatus.pro*
- start_switchtofiber.pro*
- start_switchtostar.pro*
- start_systemlog.pro*

- start_tss_widget.pro*
- start_wfcidl.pro*
- start_wfsconfig.pro*
- start_wfsstatus.pro*
- startup_spiral.pro* -> rename to start_spirtal.pro

B. Relevant EPICS Keywords

The EPICS keyword system remains the same at the user level and hence the operational tools that correspond through EPICS keywords are minimally affected. The main changes are (1) additional operation mode settings and (2) WFS array size difference. The new telemetry server is different from the existing ones. We expect to have an interface tool in place to make the telemetry compatible with the operations software. In addition, we expect to have Python tools to extract the telemetry.

Identified below are the operational procedures and EPICS keywords that are associated with the RTC system. The relevant RTC telemetry channels are listed in KAON 1165 and will not be repeated here. These EPICS keywords, the telemetry channels, and the operational procedures should be validated as part of the operational transition of the RTC upgrade.

The IDL procedures/routines are located on k2aoserver at /kroot/rel.new/ao/idl/default/bin/solaris/. The python tool, Maori is located at /kroot/rel.new/ao/default/bin/solaris/.

- RTC startup scripts (shell scripts & IDL routines):
    - The commands/scripts to reboot, reset, and power cycle the RTC computer need to be tested and documented.
    - The operational tool maori.py needs to be updated and tested with the new RTC system.
    - IDL routine: ‘rebootwfc’; keyword: wcup
    - IDL routine: ‘initwfc’; keywords: wcstat, wcinit, dmlp, dtlp, utlp
- AO system startup (IDL) scripts:
    - ‘calsetup’; keywords: wsbkfn, dtdst, utdst, dtsensor, dtclxoff, dtclyoff, dtmroff, dtclp, dtgain, utgain, dmgain, stcngn, wssmrep, wsfrrt, wscngn
    - ‘nighttimescript’; keywords: wcstat, dtsensor, trsrec, wcinit, dmlp, dtlp, utlp, dtclp, utclp, dtclxoff, dtclyoff
    - ‘switchtohalt’; keywords: wcstat, dmlp, dtlp, dtclxoff, dtclyoff, trsrec, wcstby
    - ‘switchtofiber’; keywords: dmlp, dtlp, dtclxoff, dtclyoff
    - ‘switchtostar’; keywords: dmlp, dtlp, dtclxoff, dtclyoff
- AO control (IDL) tools
    - ‘acamtool’; keywords: aofox, dtlp,utlp, utclp
    - ‘align_tss’; keywords: tldtmrv, dtsensor, dtgain, strtmd, dtlp, dtgtst, stgate, dtmroff
    - ‘aoacq’; keywords: trsrec, obwpdsrc, dtsensor, dtdst, utdst, wssmbin, wsfrrt, dtlp, dmlp, utlp, ststate, stapdmn, stsnr, stbkgnd, stinttim, tldtmrv, tlsbmdint, tluterv, dtclxoff, dtclyoff,

dtclp,utclp, wscngn, stcngn, dtservo, utservo, dmservo, dtgain, utgain, dmgain, stgate, stbkgnd

- ‘auto_man_aqr’; keywords: tlutmrv, dtlp, dmlp, utlp
- ‘cameragui’; keywords: wssmprg, wssmbin, wsfrrt, wssmgn, wssmrep
- ‘eval_cog_update’; keywords: wscnorfn, dmmrfn
- ‘fsmoffload’; keywords: dtlp, dtsensor, tldterv, tluterv
- ‘getaostate’; keywords: dtlp dmlp, utlp
- ‘init_strap_params’; keywords: dtlp, dtsensor
- ‘makebkgnd’; keywords: wsbkfmd, wsbkfm, wsbkfd
- ‘readbackrmag’; keywords: tlsbmdint, wsfrrt, wssmgn, stapdmn, stinttim
- ‘recongui’; keyword: dmmrfn
- ‘reconproc’; keywords: tlsbmdint, utlp, dtlp, wscngn
- ‘record_acam’; keywords: tlsbmdint, dmgain, dtgain, wsfrrt
- ‘recordaosetup’; keywords: wssmrep, wssmprg, wssmgn, wsbkfn, wsflfn, wscnorfn, dtgain, dmgain, tlsbmdint, wscngn, wssmprg
- ‘record_lbwfs_info’; keywords: wscnorfn, dmmrfn
- ‘saveconfig’; keywords: wsfrrt, dmorfn, wsbkfn, wsflfn
- ‘setframerate’; keywords: wsfrrt, wssmprg, wssmbin, wssmgn, dtlp, dmlp, utlp, wssmrep
- ‘set_light_level’; keyword: tlsbmdint
- ‘setngsao_targ’; keywords: dtgain, dmgain, wscngn
- ‘setngsao_vmag’; keyword: wssmbin
- ‘setup_strap’:; keywords: stinit, stbkgnd, stgate
- ‘set_strap_gain’ ; keywords: dtsensor, stinttim, stbkgnd, ststate, dtgain, stdyst
- ‘setupwfc’; keywords: wssmprg, wssasz, wsclutfn, wsclutfd, wssmgn, wssmrep, wsbias, wsfrrt
- ‘setwcoper’; keyword: wcoper
- ‘setwfcparms’; keywords: stcentgn, dtclp, dtcrmat, utcrmat, stcm, stcngn, wscngn, dmmfvec, dmamfn, dmorfd, dtservo, utservo, dmservo, dtlim, utlim, wsmnsf, stdgrt, stinttim, stdyst
- ‘spacecomclose’; keywords: utlp, dmlp
- ‘strap_calcerror’; keywords: stinttim, stbkgnd, stapdmn, stcm, dtsensor
- ‘take_strap_bkg’; keywords: ststate, stinttim, dtlp, dmlp, utlp, wcdtstat, wcdmstat, wcutstat, stapdmn, stgate, stbkgnd
- ‘take_wfs_bgd’: dtlp, dmlp, utlp, wcdtstat, wcdmstat
- ‘utttool’; keywords: tluterv, tlsbmdint, utlp
- ‘uttzero’; keywords: utmroff, trsrec

- ‘wfs_focus’; keywords: trsrec, dtlp

- AO calibration (IDL) tools:
  - ‘center_wls’; keywords: tlsnmdint, tldterv
  - ‘check_lbwfs_ref’; keyword: dmmrfn
  - ‘configgui’; keyword: wscnorfn
  - ‘dm_control’; keywords: dmorvec, dmorfn, dmorfd
  - ‘dm_cross’; keyword: dmorvec
  - ‘dm_waffle’; keyword: dmorvec
  - ‘dm_wls_reg’; keywords: tlcent, dtsensor, dtlp, dmlp, wscmorfn, wscmorfd
  - ‘image_sharpening’; keyword: dmorvec
  - ‘imshp_auto.pro’; keywords: dmlp, dtlp
  - ‘lbimageacq’; keywords: dtlp, dmlp, wscnorfn, wssmbin
  - ‘lbwfscals’; keywords’: dmorvec, stgtact, dtlp, dtsensor, strtmd, stgate, stgtst,
  - ‘loadbkgnd’; keywords: wsbkfder, wsbkfm, wsbkfd
  - ‘loadcog’; keywords: wscnorfn, wscnorfd, wcldcfgfn, wcldcfg, wsfrrt, dmorfn, wsbkfn, wsflfn
  - ‘loadconfig’; keywords: wcoper, wckdcfgd
  - ‘loadflat’; keywords: wsflfder, wsflfn, wsflfd
  - ‘loadmr’; keywords: dmmrfn, dmmrfd
  - ‘loop_gui’; keyword: dtsensor
  - ‘wfcidl’; keywords: dtlp, dmlp, wcstat, ststate, trsrec, wcstby
  - ‘wfs_cal’; keywords: dmorvec, dtsensor, tldterv, tlcents, dtlp, dmlp
  - ‘wfsconfig’; keywords: wssmbin, wscngn
  - ‘wobble’; keyword: trsrec
  - ‘zernike_control’; keyword: dmorvec

- Status/Performance monitoring (IDL) tools:
  - ‘checkwfc’; keywords: wcup, wcstat, dtmroff
  - ‘atmogui’; keyword: dmlp
  - ‘detectwfcfault’; keywords: wcdtstat, wcdmstat, wcutstat, stinttim
  - ‘wfsstatus’; keywords: tlsbint, dtsensor, tldterv, tluterv,
  - ‘strapstatus’; keywords: ststate, stgtact, stapdmn, stbkgnd, stinttim, dtlp, tldterv
  - ‘dmdisp’: dmcommand telemetry
  - ‘faultdetector’; keywords: dtlp, dtsensor, stgtact, sthvstat, stocrstat, sttcstat, stotstat, stepstat, stgate

- Performance optimization/maintenance (IDL) tools
  - ‘bw_widget’; keywords: stcngn, wscngn, stinttim, trsrec, trsrec, dtgain, dmgain
  - ‘generate_imx’; keywords: tlsbmdint, dtsensor, dmorvec, tlcents, dtlp
  - ‘rawimdisp’; keywords: wcoper, wssmbin, wssasz, tlcents, trsrec
  - ‘systemlog’; keywords: dtlp, dmlp
  - ‘logseeing’; keyword: trsrec
  - ‘trigger_diag’; keywords: wssmprg, wssmrep, wssmgn, wsfrrt, dmservo, dtservo, utservo, dmmrfn, wscnorfn, wscngn, dtlp, dmlp, utlp, dtsensor, stcm, stcngn, dtclxoff, dtclyoff, stinttim
  - ‘tsslog’; keywords: stloop, stgain, stinttim, stsnr, stapdmn, stbkgnd, stcntmn, stcntsd
  - ‘ttdithering’; keywords: utdstfn, utdstfd, utdst, dtdstfn, dtdstfd, dtdst
  - ‘wfcttinfmat’; keywords: dtsensor, dtcrmat, tldterv, dtlp, dmlp, dtclp, dtmroff
  - ’straplog’ ; keywords: stapdmn, stapdsd, stcntmn, stcntsd
  - ‘strapmatcal’; keywords: stcm, stgtact, tkdterv, stcngn
  - ‘centgainsfp’; keywords: dtsensor, tldterv, wscngn, dtlp, dmlp, dtmroff
  - ‘cog_sharp’; keyword: wscnorfn
  - ‘dm_flatten_edge’; keywords: trsrec, dtlp, dmlp
  - ‘lbdisp’; keywords: dmmrfn, wscnorfn
  - ‘man_sharp’; keywords: wscnorfn, wscnorfd
- TRICK tip/tilt sensor tools
  - trickmanager: AO keywords: dtlp, trkro1px, trkro2px, trkro3px, trkro4px, trkmastr, trkenapx, trkfpspx; TRICK keywords: trkstop, trkrocpr, trkfrmmd
  - toda: AO keywords: dtsensor, aoopsmode, trkmastr, trkwtfac, stinttim, dtgain, aodtstat, dtlp, dmlp, utlp
  - trk_acq: trkro1px, tkspcszx, tkspcszy
  - trk_changepars: trkmastr, stinttim, ststate; dtgain, trkwtfac, trkrordy,
  - trk_control: trkmastr, tkcrxs, tkspcrlx, tkspcrly, trknmad1, dtgain, dtlp, , stinttim, trkwtfac, dtservo,
  - trk_dispgui: trkfpspx
  - trk_putreadmode_1: trkfrmmd, trkstat
  - trk_putreadmode_3: trkfrmmd, trkstat
  - trk_readbackirmag: trlroi1pi, trlroi2pi, trlroi3pi, trlroi4pi
  - trk_readpars: wcup, trkrdy, trkcent, trkoffst, dtgain, trkinth, trkcngn, trkro1dl, trkro2dl, trkro3dl, trkro4dl, dtsensor, trkmastr, trkwtfac, trkro1xs, trkro2xs, trkro3xs, trkro4xs, trkro1xp, trkro2xp, trkro3xp, trkro4xp, trkro1yp, trkro2yp, trkro3yp, trkro4yp, trknmad1, trknmad2, trknmad3, trknmad4

- trk_sync_spoc: trkro1xs, trkro2xs, trkro3xs, trkro4xs, trkro1xp, trkro2xp, trkro3xp, trkro4xp, trkro1yp, trkro2yp, trkro3yp, trkro4yp, trkrordy

C. IDL Codes accessing the EPICS Channels

For completeness, the IDL routines/functions that directly access the EPICS channels are listed here. The 'caget' is called by acamtool.pro, auto_man_acq.pro, calsetup.pro, nighttimescript.pro & plotObs.pro and 'caput' is called by acamtool.pro, calsetup.pro, focus_lta.pro & nighttimescript.pro. In addition to these channels, 'caget' is called by roigui_engineering.pro, trk_readpars.pro, trk_undo_distmodel.pro & undo_DistModel.pro on K1 AO.

D. IDL Codes Cleanup

- ***O-12:*** The following IDL routines/functions may be moved to a development folder:
  - utttool.pro
  - Lbcontrol.pro – sets LBWFS keywords
  - lbdisp.pro – LBWFS display
  - setup_lbwfs.pro
  - setngsao_targ.pro
  - pickaosetup.pro
    - Used by readsetup_arch.pro
  - tss_widget.pro
    - Uses tss_vis.pro & tss_widget_find_targ.pro

  The following IDL routines/functions may be deleted from the operation folder:
  - wfcprocess.pro
  - query_ltcs.pro by tss_vis.pro
  - lgsarch.pro by generate_arT.pro
  - spect.pro
  - startup.pro
  - which_trs.pro by startup.pro
  - tss_readstarlist.pro
- The IDL functions/routines specific to an operational tool may be combined in a single IDL file called 'operational tool name'_sub.pro for clarity.
- The general IDL routines/functions could be moved to a subdirectory called, 'common' for clarity and specified the subdirectory in the IDL path.
- Move the routines/functions crossed-out in the text to a development folder.
- Address the suggested changes to the IDL routines/functions that are marked in red throughout the document in consultation with the AO operation.