BOP Procedures
$ICE/note/xped/KRT1/notebook/procedures/BOP

BEST PRACTICES

Start by reading the flight notes. This will inform you of any unusual aspects of the data collection such as PST renamings, restarted instruments, or anything that may have affected the data collection process.
Avoid overloading the QNAP by being mindful of the I/O and bandwidth requirements of your task
Mark your initials on the media envelopes when you finish downloading or deleting.
Do not delete any original data unless it has been archived
Archiving tapes (copies and full tapes) should be distributed to avoid single point failures
Always unmount disks before removing
{_} brackets are used to denote options or file names F## is the flight number (i.e. F02, F12), J### is for the Julian day (i.e. J055, J132)
Any qualitative notes about what you find in this process go in: $ICE/note/xped/KRT1/notebook/BOPnotes
BOP tends to be IO-limited, and archiving goes much more slowly if it isn't fed data quickly enough. Use "sudo iotop" to determine which processes are IO-limited vs. CPU-limited. Suggested order: download ELSA, run glnk, download RAD, check for PST renaming, run grad for VHF, then start archiving VHF. Everything else can happen in parallel with the main VHF flow.
Linux tips and useful commands are at the end of this document

Download

Preserve the time stamps of the data (i.e. $ cp -p)
Mark the media envelope after downloaded with your initials
When there are multiple GPS units of the same type, check envelope for ID

Media	Instrument	Instructions
SSD drive	VHF (HERA)	• Make folder in hierarchy: $mkdir $ICE/orig/xped/KRT1/acqn/HERA2/F## • Copy files from media: (The radar drive will mount to /media/qc/RADP##) $cp -p /media/qc/RADP##/jddd/radar* $ICE/orig/xped/KRT1/acqn/HERA2/F##
	ELSA	• Make folder in hierarchy: $mkdir $ICE/orig/xped/KRT1/acqn/ELSA/F## • Copy files from media: $cp -p /media/qc/RADP##/jddd/serial* $ICE/orig/xped/KRT1/acqn/ELSA/F##
GPS Naming Conventions		**Airborne:** **##: First number is the location of the gps;** **Second number is the current version/iteration of receiver used on the particular antenna** **Base Station:** **Prefix: Use the 3 letter acronym of the base (i.e. JBS for Jang Bogo Station)** **##: First number is the antenna number** **Second number is the current version/iteration of the receiver used on the particular antenna**
USB	TRM (IBH12)	• Make folder in hierarchy: $mkdir $ICE/orig/xped/KRT1/acqn/TRM/F## • Copy files from media: $cp -p /media/qc/GPS##/## $ICE/orig/xped/KRT1/acqn/TRM/F## • Extract the content of the zip file (from the TRM folder): $dtrx filename.zip • Rename .##N and .##O files to be IBH12.##N and IBH12.##O
Online	Base GPS (JBS11)	• Make a folder in the hierarchy: $mkdir $ICE/orig/xped/KRT1/acqn/GPS_JBS/F## • Download data from website to BOP USB stick: http://monitor.kopri.re.kr/gps • Copy files from BOP USB stick: $cp -p /media/qc/CORSAIR/###l* $ICE/orig/xped/KRT1/acqn/GPS_JBS/F##

PST RENAMING

NOTE: The need for renaming should be noted in flight log. Also check the results of ELSA breakout against expected PST names

Create file: modify_pst.rc with lines “old_pst new_pst” using the format: P/S/T P/S/T (one PST change per line). An example file can be found at $ICE/code/xped/KRT1/breakout/ELSA/src/ex_modpst.rc.

Copy/place file:
$ $ICE/orig/xped/KRT1/acqn/{HERA2, ELSA}/F##/

Run respective breakout scripts with the appropriate modify pst flag shown below.

BREAKOUT

Scripts in media/ICE/code/xped/KRT1/all breakout data AND generate files for printing
To save a log file, run the breakout script using ./glog. (i.e. $ ./glog ./grad -flags …)
Note: There are warnings that will be displayed (along with other verbose statements). To check if the script worked check the contents of targ and the corresponding print out folders

AQN (ELSA)	$ cd $ICE/code/xped/KRT1/all $ ./glnk --season KRT1 --platform GCX --flight F## [-overwrite] [-modpst] Check that all expected PSTs are there $ du -hcs $ICE/targ/xped/KRT1/breakout/ELSA/F##/*/*/*
VHF	$ cd $ICE/code/xped/KRT1/all $ ./grad -proj KRT1 -plat GCX {-f forces overwrite of breakout} {-p causes pdfs to appear on Desktop} {-modpst to change PST} F##
GPS (do this after ELSA)	$ cd $ICE/code/xped/KRT1/all $ ./ggps [-ignoremissing] F## Jddd yyyy NOTE: Use -ignoremissing flag if missing data NOTE: All GPS can be manually broken out separately if there's a problem with above script $ cd $ICE/code/xped/KRT1/breakout/{GPS} $ ./bo_pos_ash -proj KRT1 -flight F## $ ./bo_pos_jvd F## {or} JDAY NOTE: Do the following three steps if re-running to integrate other sensors into the QC sheets $ cd $ICE/code/xped/KRT1/quality/xlob/teqc $ ./doteqc F## $ cd $ICE/code/xped/KRT1/quality/flight $ ./run_qc.sh F## $ cd $ICE/code/xped/KRT1/print $ ./ggps -proj KRT1 -year 2017 F## ~/Desktop/F##_gps_printing
GQC (not an instrument, but do after GPS & ELSA)	Produces Flight QC, Geology QC, Ice QC, and Review pdfs (this will generate the qc sheets)$ cd $ICE/code/xped/KRT1/all$ ./gqc F##
CAM (not dependent on good PST)	$ cd .$ICE/code/xped/KRT1/breakout/CAM $ ./bo_cam -flight F##

PRINT

NOTE: you may do breakout and printing in parallel

VHF	• The grad script at the breakout step should generate a folder on your desktop • Print its content: ~/Desktop/F##_HERA2_printing/* • Classify the sheets in the RADAR binder NOTE: the VHF radargrams are stored at: $ICE/targ/xped/KRT1/quality/xtra/plots/pyk1.RADnh3
AQN	• The glnk script at the breakout step should have generated a folder on your desktop • Print its content : ~/Desktop/F##_pcor_printing/* • Classify the sheets in the PCOR binder
GPS	• The ggps script at the breakout step should have generated a folder on your desktop • Print its content: ~/Desktop/F##_GPS_for_printing/* • Classify the sheets in the PCOR binder. Group instruments together, in the order:GCX12, GCX11, GCX21, GCX31, GCX41, ZGN11, ZGN21, ZHS11, CAS11
GQC	• Print the the pdf files and classify the sheets in the FLIGHT NOTES binder: $ICE/targ/xped/KRT1/quality/flight/plots/F##/*.pdf

QC

Instructions	On the flight based QC sheet, highlight quality of each sub-section of transects using the following color code: 1 = blue 2 = yellow 3 = orange pink = missing data or sections where the instrument could not be expected to perform normally Note any error messages, synchronization problems, or times missing in each dataset
Flight Notes	Review the flight notes to check any PST renaming, FOP passes to BOP, and events that could explain data singularities

GPS	Each reciever has at least 7 satellites, with few l's (indicates loss of lock). See print-out or `teqc -help` for symbol list explanation Ideally we want observations/cycle slip to be >1000 ORIENTATION (dPOS): 1. At least a combination of 1 tail + 2 wings at any time across full section of transect 2. No combinaison of 1 tail + 2 wings for < 10% of the transect 3. No combinaison of 1 tail + 2 wings for > 30% of the transect CENTER OF GRAVITY (CG): 1. At least 7 satellite at any time across full section of transect 2. No satellites for < 10% of the transect 3. No satellites for > 30% of the transect
ALT	LASER: 1. Dropouts for <10% of the transect 2. Dropouts for between 10% and 30% of the transect 3. Dropouts for > 30% of the transect
IMU	Roll/Pitch:1. Roll < 1 degree2. Roll < 10 degrees3. Roll > 10 degrees
RADAR	It is recommended that VHF QC is done from pyeva or digital copies of the radargrams. The paper printouts do not have sufficient dynamic range to see all features. $ cd $ICE/syst/linux/src/pyeva $ python pyeva.py $PST SURFACE (Low gain, channel 1): 1. Surface traceable without interruption across entire section 2. One or more locations where the surface is present but poorly defined or indistinct 3. One or more location where the surface is not present in the radargram Note: Traces should not saturate at the surface LAYERS (Low/High gain): 1. Layers visible to a depth of ~8 μs above the bed for >80% of the section 2. Layers visible to a depth of ~8 μs above the bed for 40-80% of the section 3. Layers visible to a depth of ~8 μs above the bed for <40% of the section BEDROCK (High gain, channel 2): 1. Bed is clear and traceable across full section of transect 2. Bed is faint or partially obscured by noise at one or more points along the section 3. Bed is not present or completely obscured by noise at one or more points along the section
Media	On the flight based QC sheet, track each media ID that has been used for this flight

Scans

FLIGHT NOTES	Scan the flight notes into a pdf/jpg format and copy them to: $ICE/note/xped/KRT1/acqn/Flight_Logs/F## Flight notes hard copy is stored in the FLIGHT NOTES binder
QC SHEETS	Scan the flight based QC sheet into a pdf/jpg format and copy them to: $ICE/note/xped/KRT1/acqn/Flight_Logs/F## QC hard copy is stored in the FLIGHT NOTES binder

ARCHIVE

The LTO-6 tape drive plugs directly into a USB3 port, and usually mounts to /dev/nst0 under Ubuntu. After archiving a flight to a tape, write “F## ##G” on the tape so we know what each contains. Useful commands:* eject tape: $ mt -f /dev/nst0 offline * add tag to new tape: $cd $ICE/code/xped/KRT1/archive; ./add_tad [tag_type] [copy]

RADAR	ORIGINAL DATA (ovhfr x2) • $ cd $ICE/code/xped/KRT1/archive • $ ./find_ovhfr F## • Insert "KRT1 ovhfr copy #1" into the tape drive; make sure you have permission to write to drive:$ sudo chmod a+wr /dev/nst0 • $ python arc_syst.py F## ovhfr {copy} • (Same for copy 2, skipping find_ovhfr) BREAKOUT DATA (tvhfr x1) • $ cd $ICE/code/xped/KRT1/archive/ make sure you have permission to write to drive:$ sudo chmod a+wr /dev/nst0 • $ ./find_tvhfr F## • Insert "KRT1 tvhfr copy #1" into the tape drive • $ python arc_syst.py F## tvhfr 1

ELSA, AQN, GPS, oelpo does not require scanned QC sheets telpo does	ORIGINAL DATA (oelpo x2) • $ cd $ICE/code/xped/KRT1/archive • $ ./find_oelpo F## Jddd • Insert "KRT1 oelpo copy #1" into the tape drive • $python arc_syst.py F## oelpo {copy} • (Same for copy 2) BREAKOUT DATA (telpo x1) • $ cd $ICE/code/xped/KRT1/archive/ • $ ./find_telpo F## Jddd • Insert "KRT1 telpo copy #1" into the tape drive • $ python arc_syst.py F## telpo 1
note, syst, code, PIK	SUNDAY (x2)Whenever you have the chance, we like to archive all of our code, notes and generated picks. This makes recovering from QNAP failure much easier =) Rather than being flight-based, these archives are jday based.• $ cd $ICE/code/xped/KRT1/archive • $ ./find_sunday Jddd • Insert "KRT1 sunday copy #1" into the tape drive • $python arc_syst.py Jddd sunday {copy} • (Same for copy 2)

CLEAN

All	• We keep a minimum of data from the two last flights on media. Before deleting: (1) Check qc sheets to be sure the media was qc-ed and makes sense (2) Create log files of each tape on DIFFERENT LTO than they were written: Look at index written on case, figure out which file you want. To get the 5th file: (use status to check that you're where you expect) $ mt -f /dev/nst0 rewind$ mt -f /dev/nst0 fsf 5$ mt -f /dev/nst0 status $ tar tv -f /dev/nst0 > $ICE/note/xped/KRT1/archive/confirmation/F##.{oelpo,ovhfr}.copy#Then, to get the 6th file, you can either step forward one:$ mt -f /dev/nst0 fsfOr, go back to the start and use `fsf 6` (3) For each piece of media, run: (for rad disks, will have to run for both oelpo and ovhfr)$ cd $ICE/code/xped/KRT1/archive$ python clear_media.py —tagtype {oelpo, ovhfr} —flight F## —media /path/to/folder/with/files [—dryrun]
QNAP	If and when the QNAP fills up, we preferentially delete orig VHF data, after checking that the files were archived. (Keeping targ for flight planning / science purposes) $ cd $ICE/note/xped/KRT1/archive/confirmed <insert tape into different LTO than written from, and for each flight on tape do:> $ as-read F##.tar | tar vt > F##.ovhfr.index $ cd $ICE/code/xped/KRT1/archive/confirmed $ ./confirm_ovhfr_one.sh F##

EXTRA

LINUX	• Mounting media:
	Some media automount – get address by clicking Places (upper left)
	If the media doesn't mount:
	$ sudo fdisk -l (check name of the connected media, should be /dev/sd??)
	$ sudo mkdir -p /media/?? (create the directory where the media will be mounted)
	$ sudo mount /dev/sd?? / media/?? (mount the media)
	$ df -h (check if the media is mounted)
	
	• How to unmount media:
	$ df -h (check the name of the mounted media)
	$ sudo umount /media/?? OR $ sudo umount /dev/sd?? (if unusual chars in /media option) (unmount the media)
	$ df -h (check if the media is unmounted)
	
	• Check file systems for errors on a media:
	$ sudo fsck /dev/sd?? (see if there are errors)
	$ sudo fsck -r /dev/sd?? (repair errors)
	$ sudo fsck /dev/sd?? (see if errors have been fixed; run with -r option again if not)
	
	• Copy a disk image from a media to another:
	$ dd if=/dev/sd1 of=~/Desktop/media.img (create an image from sd1 or other media)
	$ dd if=~/Desktop/media.img of=/dev/sd2 (copy the image to sd2 or other media)
	
	• Check size of folder or disk:
	$ du -ch | grep total
	
	• Adjust size of terminal window (without mouse):
	alt+F8 (use arrows to resize, press enter to fix)

ROUTER

• To administer via the webpage:
192.168.1.1
username: (none)
password: default
Setup → DHCP reservations to see what devices are currently connected and how

QNAP

• To manually mount:
$ sudo mount $ICE
{or}
$ sudo mount 192.168.1.106:/ICE $ICE6
$ source ~/.bashrc

• To reprioritize RAID checking:
(The QNAP may slow down drastically when its checking the RAID)
ssh into QNAP
use top to find the PID of the process hogging resources
renice <pid> 20

• To administer via the webpage:
192.168.1.106:8080

LTO DRIVE

• To check files archived on a tape:
tar tv -f /dev/nst0 (I think this only reads back whichever file is the current next one, so will be used with `mt -f /dev/nst0 fsf`
• The LTO-6's are standard tape drives. They are controlled using `mt` and `tar`, both of which have extensive man pages.
• The LTO-3's try to have a filesystem. We have implemented an assortment of as-* scripts that attempt to make them behave like standard tape drives.
• Tape Drive Check (lights):
amber status light: cleaning request
orange error and amber status light blinking fast: cleaning error or media failure
error blinking fast: hardware error
error light stuck on: manual intervention requested
Redoing/rerunning Archive:
delete Index and Log files in $ICE/note/xped/KRT1/archive/{ovhfr, telpo, etc}

TAPES

• Tagging new tapes:
$ cd $ICE/code/xped/KRT1/archive
$ python add_tag.py {tape_type} {1 or 2 or 3}

• Checking tape was recorded properly:

PYEVA

Opening pyeva
• cd $ICE/syst/linux/src/pyeva
• python list_picks.py {look at output, pick pst that needs picking}• python pyeva.py PSTNAME

Initial setup
• If picking the srf, select chan1; for bed, select chan2• Enter {srf,bed}_{lel} in the text box, then click "new picks" (replace lel with your initials)
• Select your picks from the list below

Maneuvering in the gui
• You can left click and drag along the bars below and to the left of the radar gram to select how much is displayed.• Another option for moving the image is to select the next/prev buttons at the bottom of the horizontal bar. This will move the screen just about the same distance that is shown on the bar, with some overlap. This allows for smoother transitions to keep picking.
Picking!!!
• On the display tab, pik1 should be deep blue. This means it is selected. Click along the top of the surface or bed (a little bit above what you want)
• Click pik2 until it is deep blue/ selected. Click along the bottom of the surface or bed (a little bit below what you want).
• If you made a mistake (pik1 points being below the desired line, for instance) and wish to delete one or more of your points, you can hold down the right mouse button and drag across the points to be removed.• If there is an area that you just can't make out (this mostly happens on the bed) select points for pik1 below pik2, this will indicate to the algorithm to ignore this section• As you pick, select "autopick" to see what the algorithm will select for you; go back and make adjustments to your picks if it doesn't look good.
• Be sure to save frequently! There are automatic backups, but recovering data from them is a manual process.

Location: targ/xped/KRT1/breakout/ELSA/P/S/T/<stream>

Stream convention:
AAAaa# : AAA = instrument;
aa = first letter: abbrev for manufacture; second letter: abbrev for what it is measuring;
# = version

ELSA STREAMS

- GPSap3 → GG24
- GPSkc1 → true time
- AUNcp1 → status … IMU
- AUNcp2 → data from ... IMU
- PRSpz2 → current pressure
- AVNiz → radar altimeter (for MKB itself)
- RADnh3 → VHF
- AVNnt1 → OAT
- AVNnt2 → OAT
- CCMei1 → Elphel
- AQNnr{1,2,3,4,5} → internal status packets
- GPSpp1 →
- GPStp3 →

Stream convention:
AAAaa# : AAA = instrument;
aa = first letter: abbrev for manufacture; second letter: abbrev for what it is measuring;
# = version