NIRC2 Reduction Cheat Sheet

I. Requirements:

- >300M RAM (this can be modified)
- the XIDL package
- the IDL coyote package
- the Goddard IDL package
- the Schlegel IDL package (compiled)

II. Setup

(a) Create a Raw/ directory and put all the raw data in it. Move to the directory above Raw.

(b) Create the IDL structure which drives all subsequent programs by typing the following: **xdimg_strct, strct, 'NIRC2', 'Keck', /MKDIR**

(c) If your Raw directory contains irrelevant files, you'll need to edit the structure by setting the FLG_ANLY tag to 0 for those files.
irrel=where(strct.img_root eq 'n0003.fits')
strct[irrel].img_root = 0

(d) Verify that the image types were correctly identified. If not, fix them by changing the TYPE keyword to one of the following: OBJ, STD, TWI, DRK, ZRO, DFT.

(e) Save any changes you made to the structure. Make sure to use the /create flag:
mwrfits, strct, 'struct.fits', /create
Note that at any point during the reductions you can exit out of IDL, and re-start where you left off by reading in the structure as follows: strct=mrdfits('struct.fits',1).

III. CALIBRATIONS ::

Create a bad pixel mask, super-darks, and flats. You will need one set of flats for each filter, and a dark frame for each exposure time of science data. The bad pixel mask is helpful but not necessary.

(a) Create a bad pixel mask. This routine uses the dark frames to pick out hot pixels and it uses well-exposed flat fields to pick out dead pixels. Therefore the structure must include at least one of these types of frames. You can set the DEADPERCENT and HOTPERCENT keywords to change the threshold of what is considered a bad pixel. If multiple flats and darks are present the program goes through each frame looking for dead and hot pixels. In this way it can locate "sometimes" dead and hot pixels. If reducing narrow and wide field data simultaneously, you must run this routine once for the narrow camera and once for the wide camera since these are treated as separate instruments.
xdimg_badpix, strct, CCD='NIRC2W'

Investigate the bad pixel mask like this:
atv, 'PixMask/BadPixMask_w.fits'
If you are unsatisfied, you can always replace the BadPixMask_w.fits file with a mask of your own.

(b) Create a super-dark frame for every type of dark frame that we have. For example, there will be one super-dark frame for each unique combination of exposure time, coadds, sampmode, and nreads. This routine median combines all the dark exposures of the same type.
xdimg_dark, strct

Investigate the super-darks if you like. The darks are saved with the NIRC2 camera name, the exposure time, the number of coadds, the sampmode, and the number of reads (in that order).
xatv, 'Darks/Dark_w10.000_6_3_8.fits'

(c) Create the dome flat. This routine dark subtracts the dome flats with the super-dark frame of the same type, and then median combines, scaling by the median value in a small portion of the image. It then deletes the dark subtracted images unless instructed otherwise. It creates one normalized flat per filter. If reducing narrow and wide field data simultaneously, you must run this routine once for the narrow camera and once for the wide camera since these are treated as separate instruments.
xdimg_dflat, strct, /IR, CCD='NIRC2W', BPM='PixMask/BadPixMask_w.fits'

Investigate the flats as you like. (The "N" is for Normalized.)
xatv, 'Flats/DFlat_w_NKp.fits', min=0, max=1

(d) Create a sky flat. A sky flat is not necessary if you just created a dome flat. Ideally one has > 5 well dithered images per filter with no terribly bright objects. The routine dark subtracts the science images and median combines with clipping. It creates one normalized flat per filter.
xdimg_skyflt, strct, /IR, CCD='NIRC2W', /NOMSK

Investigate the flats as you like:
xatv, 'Flats/SkyFlt_w_NKp.fits'

IV. Reduce the Science Images

(a) Reduce the science images. Specify the name of a bad pixel mask if one exists, and which flat type is preferred for use. (Specify the flat ROOT name only, not including the filter.) If you want to later coadd the final images using Josh's code, include a FILEOUT name to be written with a list of final files.
IMS = where(strct.type EQ 'OBJ' AND $
strct.flg_anly NE 0 and strct.ccd eq 'NIRC2W')

xdimg_proc, strct, IMS, 'Flats/SkyFlt_w_N', /IR, $
BPM='PixMask/BadPixMask_w.fits', FILEOUT='stackFile'

The reduced images are in Final directory, and have a prefix 'f_'. They are in units of counts, i.e. they have not been multiplied by the gain or divided by the exposure time.

(b) Run Josh's coadd code. After a discussion with Hsiao-Wen, we determined that the ideal coadd code would subtract the median from each image, and then coadd the median-subtracted images, weighting by the inverse variance and scaling by the exposure time.