HEASARC: NICER Analysis Guide

It is strongly recommended that, before starting analysis of NICER data, user review the NICER Mission Guide for an overview of the mission, the instrument subsystems, and the data file structure.

NICER Data Reduction using `NICERDAS`

`NICERDAS` is a set of NICER-specific tools distributed with `HEASoft` version 6.24 or later.

`NICERDAS` includes the following tools:

`nicercal`	- Apply standard calibration to NICER observation
`nicerclean`	- Apply standard NICER screening
`nicerl2`	- Apply NICER Level2 standard calibration and filtering
`nicermergeclean`	- Merge NICER MPU data and apply standard screening
`nicerpi`	- Apply gain calibration to NICER event file (calculate Energy/PI)
`nicertimecal`	- Apply clock calibration to NICER event file
`nicerversion`	- Report NICER software version
`niextract-events`	- Fast event filtering, similar to extractor
`nimaketime`	- Create standard NICER screening GTI
`nimpucal`	- Apply standard NICER calibration to MPU data
`nimpumerge`	- Merge NICER event data from multiple MPUs (or observations)
`niprefilter2`	- create augmented NICER-specific filter file
`niprefilter`	- create NICER-specific filter file

`NICERDAS` uses the HEASARC `CALDB` to access calibration data.

NICER Data Reduction using `NICERDAS`

Standard Data Reduction Steps:

1. `nicercal`: Apply NICER calibration
2. `nimaketime`: Apply standard NICER time screening
3. `nicermergeclean`: Merge and clean event data; also used to merge data from multiple observations if desired

`nicerl2`: A script which runs the filtering, calibration and merging of NICER events. This wrapper script calls `nicercal` + `nimaketime` + `nicermergeclean`.

If users are interested in doing pulse searches or other detailed timing studies, the NICER event files produced by `nicerl2` can be corrected to the solar system barycenter using the `barycorr` task. See the discussion below for more information about using `barycorr` with NICER data.

After data have been cleaned and calibrated (using `nicerl2`, for example), science products (time-filtered events, spectra and/or lightcurves) can be extracted using `xselect`.

Using `nicerl2`

The `nicerl2` task applies standard NICER-recommended calibration processes to an entire NICER observation, as well as standard screening.

This is a high level task that calls multiple sub-tasks as a convenience to the user. It performs the following standard "Level2" processing steps.

1. `nicercal` - apply standard NICER calibration
2. `nimaketime` - create standard screening good time intervals
3. `nicermergeclean` - combine per-MPU data and filter/screen

Please see the help files for these individual tasks for more information about what they do and how they can be used.

Main task parameters:

```
   indir [filename]
          Input directory name. The directory should be a single NICER
          observation directory, which in turn contains
          xti/{events_uf,events_cl,hk,auxil} subdirectories.

   ufdir = "xti/event_uf" [filename]
          Subdirectory containing per-MPU "uf" event files, which are the
          input to the task. This directory name is relative to the
          observation directory.

   cldir = "xti/event_cl" [filename]
          Subdirectory where the output files as listed above are placed.
          This directory name is relative to the observation directory.

   ufafile = "$CLDIR/ni$OBSID_0mpu7_ufa.evt" [filename]
          Name of the master "ufa" event file to be created. Use the
          variables "$CLDIR" for the output directory name, and use
          "$OBSID" for the observation ID number.

   clfile = "$CLDIR/ni$OBSID_0mpu7_cl.evt" [filename]
          Name of the master "cl" event file to be created. Use the
          variables "$CLDIR" for the output directory name, and use
          "$OBSID" for the observation ID number.

   mkfile = "auxil/ni$OBSID.mkf" [filename]
          Name of the MKF filter file, to be used as input, relative to
          the observation directory. Use the variables "$CLDIR" for the
          output directory name, and use "$OBSID" for the observation ID
          number.
```

Examples:

1. Apply NICER calibration and screening to data from observation `1706221428`, overwriting previously existing files that are in the `event_cl` directory:

```
prompt> nicerl2 indir=1706221428 clobber=YES
```

The input directory is a NICER observation number `1706221428`. The output files are placed in the observation directory, in the cleaned event directory, i.e.:

```
1706221428/xti/event_cl/
```

Note that you should be in the root directory which contains the NICER data directory `1706221428`.

Running the Tools Separately

If you want more control of the data reduction, you may want to run the `nicercal`, `nimaketime`, and `nicermergeclean` separately, rather than running the `nicerl2` task.

Step 1: run `nicercal`

`nicercal` applies standard NICER-recommended calibration processes to an entire NICER observation (in a standard NICER observation ID directory structure, and thus works on an entire observation directory. The default screening is to remove over-shoot and under-shoot events (since they cannot be calibrated) but to keep X-ray events and forced triggers

An example of using `nicercal` to apply NICER calibration to data to observation 1706221428.

```
prompt> nicercal indir=1706221428 outdir='$INDIR/xti/event_cl' clobber=YES
```

The input directory given above (`indir`) is a NICER observation directory which contains data from observation id 1706221428.

The output files are placed in the observation directory under `1706221428/xti/event_cl/ni1706221428_0mpu*_ufa.evt`. (Note that single quotes are used to prevent `$INDIR` from being expanded by the Unix shell.) The default calibrations files are taken from `CALDB` (Note that the NICER `CALDB` can be installed locally or the HEASARC `CALDB` can be accessed remotely by the user.)

Step 2: run `nimaketime`

NICER has a task named "nimaketime" which applies recommended screening criteria.

```
prompt> nimaketime infile="$obs/auxil/ni${obsroot}.mkf" outfile="$obs/auxil/standard.gti"
```

The output of `nimaketime` is a Good-Time Interval (GTI) file which users can use downstream (here named `standard.gti`); `nimaketime` does not apply the GTIs to the events data.

Note that `$obs` is the observation directory name and `$obsroot` is the observation number. Type `fhelp nimaketime` to see what kind of screening is applied.

`nimaketime` can apply the following screening criteria. The exact filtering string is also given in quotes after each item.

- **nicersaafilt=YES**
  Excludes times within SAA as defined by NICER_SAA. This is a NICER-specific definition of the SAA stored in the NICER CALDB.
  "(`NICER_SAA==0`)"
- **saafilt=NO**
  When YES, excludes times within the generic SAA defined by prefilter. (NOTE that this is set to "no" because normally you do not need to apply both the generic and NICER-specific screening. Use either `nicersaafilt=YES` or `saafilt=YES`, but not both.)
  "(`SAA==0`)"
- **trackfilt=YES**
  Excludes times when NICER is not in target tracking mode.
  "(`ATT_MODE==1` && `ATT_SUBMODE_AZ==2` && `ATT_SUBMODE_EL==2`)"
- **ang_dist=DIST**
  Excludes times when the angular distance between the target and NICER's boresight is greater than DIST (by default, 0.015 degrees). "(`ANG_DIST < 0.015`)"
- **elv=MINELV**. Excludes times when the distance to the earth limb is less than MINELV (by default, 30 degrees).
  "(`ELV > MINELV`)"
- **br_earth=MIN_BR_EARTH**
  The same as ELV for the bright earth. Default value of MIN_BR_EARTH = 40 degrees.
  "(`BR_EARTH > MIN_BR_EARTH`)"
- **cor_range=A-B**
  Specify a range of magnetic cut-off rigidity (COR) to accept (in GeV/c). This can be used for filtering of high background regions. You can use the notation cor_range="A-" to specify a minimum but no maximum, cor_range="-B" to specify a maximum but no minimum, and the default cor_range="-" accepts all CORs.
  "(`COR_SAX > A && COR_SAX < B`)"
- **min_fpm=MIN_FPM**
  Specify a minimum number of enabled detectors. (The default MIN_FPM = 38.)
  "(`NUM_FPM_ON > MIN_FPM`)"

Step 3: run `nicermergeclean`

Up until this point, data are kept on a per-MPU basis. The data must be merged and cleaned using the task `nicermergeclean`.

First, create a list of the individual event lists from all 7 MPUs:

```
prompt> ls $obs/xti/event_cl/ni${obsroot}_0mpu[0-6]_ufa.evt > ufalist.lis
```

Then, run `nicermergeclean` using the list of individual MPU event files as input:

```
prompt> nicermergeclean infiles=@ufalist.lis \
ufafile=$obs/xti/event_cl/ni${obsroot}_0mpu7_ufa.evt \
clfile=1706221428/xti/event_cl/ni1706221428_0mpu7_cl.evt \
gtifile=standard.gti
```

- The first `ls` command lists the calibrated "ufa" files. The second nicermergeclean command merges and cleans the inputs. The GTI (standard.g.) is the result of the previous nimake.me command (see previous slide).
- Here `$obs` is the observation directory name and `$obsroot` is the observation number.
- The GTI file `standard.gti` is the output from `nimaketime`.

The `nicermergeclean` task is a high-level task that merges data from multiple NICER MPU slices, and applies standard screening. It is equivalent to running the tasks `nimpumerge` (for merging MPU data) and `nicerclean` (for applying screening).

An input to the task is a set of several MPU event files, which have been calibrated but not finely screened. These are typically named `niNNNNNNNNNN_0mpuN_ufa.evt`, where the "ufa" indicates unscreened but calibrated data. The output of the task is a single merged and screened event file, which is typically named `niNNNNNNNNNN_0mpu7_cl.evt`, where the "cl" indicates a cleaned event file. Here "mpu7" means all MPUs, numbered 0-6, have been merged into one file designated as "**mpu7**."

Another input to the task is a GTI (good time interval) file which contains the desired screening good time intervals. Note that `nicermergeclean` does not by itself create a GTI file, but rather uses one created externally by the user. The NICER team recommends to use `nimaketime` first to create a GTI file using standard screening criteria before running `nicermergeclean`.

This task runs the following lower-level NICER tasks to complete its work.

1. `nimpumerge` - to merge per-MPU data into a single file.
2. `nicerclean` - to apply screening criteria.

In addition to applying good time screening, the `nicerclean` task also applies per-event screening based on event type and pulse height range. See the documentation for nicerclean for more information. By default, events with pulse invariant (PI) pulse heights above 20 (i.e. 200 eV and above). This can be changed using the pirange parameter. For example, use `pirange=35:1000` to select events with nominal pulse heights in the range 0.35 - 10.00 keV. When `trumpetfilt=YES`, `nicermergeclean` performs standard filtering to exclude background events using the event PI_RATIO column. This is known as "**trumpet filtering**" since the PI vs PI_RATIO cloud looks like a trumpet. See the help for `nicerclean` for more information:

```
prompt> fhelp nicerclean
```

Specific Examples: merge per-MPU data from multiple observations

Create a list of the `UFA` files from obsid 1234567890, using these commands on the Unix command line:

```
prompt> ls 1234567890/xti/event_cl/ni1234567890_0mpu[0-6]_ufa.evt > ufalist.lis
```

This creates a file, `ufalist.lis`, which contains a list of all the `UFA` files from all the MPUs from OBSID 1234567890 Then append the `UFA` files from OBSIDS 1111111111:

```
prompt> ls 1111111111/xti/event_cl/ni1111111111_0mpu[0-6]_ufa.evt >> ufalist.lis
```

Then use the `ufalist.lis` as input to `nicermergeclean` to merge the data from the 2 OBSIDS:

```
prompt> nicermergeclean infiles=@ufalist.lis \
ufafile=merged_ufa.evt \
clfile=merged_cl.evt
```

If you want to apply time screening, then you will need either:

- Merge the filter files (i.e. the `.mkf` files) from 1234567890 & 1111111111, using the general HEASoft tool "ftmerge", then run `nimaketime` on result, OR
- Make individual observation GTI files using `nimaketime` on 1234567890 & 1111111111, then merge the resulting GTI files using the general HEASoft tool "ftmgtime" (combining the data using "OR" logic to accept times that are either of the GTIs). For example for the two NICER observations above, suppose you've run `nicermaketime` on both, outputting the GTI files as `<obsN>/auxil/standard.gti` (where `<obsN>` is either 1234567890 or 1111111111. Then to create the merged gti file, you should run the following commands on the command line. These commands create a list of GTIs from the first observation, appends the list of the GTIs from the second observation, then merges the GTIs using `OR` logic with `ftmgtime`, overwriting ("clobbering") previous output:

  ```
  prompt> ls 1234567890/auxil/standard.gti > combined.gti
  prompt> ls 1111111111/auxil/standard.gti >> combined.gti
  prompt> ftmgtime @combined.gti merged.gti OR clobber=YES
  ```

Merging MPU-Merged Data (`MPU7` files) from Different Observations

If you have multiple OBSIDs for a given target, it's possible to merge the `mpu7_ufa` files from each OBSID to create a combined event file for all the observations of the target.

1. Create a list called `ufalist.lis` of all the `mpu7_ufa` files. This assumes all the NICER OBSID directories are under the current working directory, and that the standard NICER directory structure is used:

   ```
   prompt> ls */xti/event_cl/ni*_0mpu7_ufa.evt > ufalist.lis
   ```

2. Merge the files with `nimpumerge`:

   ```
   prompt> nimpumerge infiles=@ufalist.lis outfile=merged_ufa.evt mpulist=7
   ```

3. You can then use `nicerclean` to screen this data:

   ```
   prompt> nicerclean infile=merged_ufa.evt outfile=merged_cl.evt
   ```

4. If you want to apply time screening, then you will need either:
   - Merge filter files (.mkf files) from the individual observa.ons using `ftmerge`, then run `nimaketime` on result, OR
   - Make GTI files using `nimaketime` on each `observation`, then merge the resulting GTIs with `ftmgtime` (combining the files using "OR" logic).

Selecting data with XSELECT

You can then use the HEASoft `xselect` package to generate data products from the screened, merged events data. For example:

```
prompt> > xselect
XSEL> set datadir 1234567890/xti/event_cl
XSEL> read event ni1234567890_0mpu7_cl.evt
XSEL> set binsize 1
```

```
XSEL> extract curve
XSEL> plot curve
XSEL> set phaname PI > extract spectrum > plot spectrum
XSEL> save curve
XSEL> extract spectrum
XSEL> save spectrum
XSEL> exit
```

Barycentering Photon Arrival Times

You can apply barycentering to NICER data using the HEASoft `barycorr` command.

```
i prompt> barycorr infile=orig.evt outfile=bary.evt \
orbitfiles="$obs/auxil/ni${obsroot}.orb" \
ra=<RA of target> dec=<Dec of target> \
refframe=ICRS
```

- RA and Dec should be the target coordinates in J2000 in degrees
- Here `$obs` is the observation directory name and `$obsroot` is the observation number
- Use the `.orb` orbit file for input, and not the filter file (.mkf). The `.orb` file has the highest precision. In rare cases, the standard filtered orbit solution may not be functioning. In that case the `SPS_ORBIT` extension can be used `ni${obsroot}.orb[SPS_ORBIT]`
- **HIGHLY RECOMMENDED**: use `refframe=ICRS`, which selects a modern ephemeris. If you don't do this, there can be millisecond-errors! Deepto Chakrabarty translated new updated JPL ephemerides DE421 and DE430, which are now included in the HEASoft 6.22 distribution. Select them with the `ephem=JPLEPH.430` to select `DE430`.

Background Correction

As noted above, NICER events can include various background components in addition to the good science events. These background components are usually minimized through standard event selection and cleaning, but residual background events may remain in even "clean" science data. This residual background should be modeled and subtracted from NICER lightcurves &/or spectra for X-ray source analysis.

Non-cosmic background contamination can be most readily seen at low energies, E < 0.3 keV and at high energies, E > 6.0 keV.

- The low energy background component is dominated by detector noise which produces a "noise peak" near near $E \approx 0.2$ keV. The height of this noise peak may be particularly large in cases of low values of Sun Angle or Bright Earth Angle, when optical impinges on the XRCs and FPMs. Optical light leakage can also produce enhanced low energy background. The noise peak varies from FPM to FPM, and is particularly large in FPM #34. If unusual variations are seen at low energies, users should examine the values of `SUN_ANGLE` and `BRIGHT_EARTH` from the makefilter (`.mkf`) file
- The high energy background component is usually caused by charged particles which are not removed by the standard screening. This background component often manifests itself

The NICER CALDB

The NICER caldb can be accessed using standard HEASoft `caltools`. We can also use the HEASARC `pycaldb` package (experimental) to examine and explore the NICER CALDB.

Using `quzcif` with the NICER CALDB

A standard `quzcif` call would be

```
prompt> quzcif NICER XTI - - MPU_GAIN now now -
Found: ftp://heasarc.gsfc.nasa.gov/caldb/data/nicer/xti/bcf/gain/nixtiflightpi20170601v001.fits   1
```

Last Updated: March 27, 2019