UHDAS+CODAS Documentation

An **ADCP** (Acoustic Doppler Current Profiler) is a device made by Teledyne RD Instruments that uses the Doppler frequency shift of an acoustic ping to infer water velocity. It is a registered trade name. ADCPs are attached to moorings, lowered on CTD rosette packages, and mounted on ship hulls. This web site deals with shipboard ADCPs.

If you are unfamilar with how and ADCP works, please visit the Teledyne R.D.Instruments website and download the ADCP (BroadBand) Practical Primer. They will ask you to create an account, but there is no cost. This outlines the underlying concepts behind how an ADCP works.

An overview of shipboard ADCP systems was written for the GO_SHIP Repeat Hydrography Manual. For convenience, an html version of the shipboard ADCP contribution is included here.*

A **CODAS** (Common Ocean Data Access System) database is a way to store and access oceanographic data. CODAS was developed in the late 1980's as a portable, self-describing format for oceanographic data, with emphasis on processed ADCP data.

"**CODAS processing**" refers to the ADCP data processing software and procedures that were developed around the CODAS format. The processing steps have become increasingly automated, but human judgement is still required for the final product, and the software is highly flexible in allowing manual configuration and execution of individual steps.

"**UHDAS**" (University of Hawaii Data Acquisition System) acquires data from RDI ADCPs and ancillary sensors (eg. gps, gyrocompass, gps and inertial attitude sensors...) and uses CODAS processing to incrementally build a dataset of averaged, edited ocean velocities for each ADCP and ping type specified. Processed data and plots are served on the shipboard network, and daily status summaries are emailed.

Note

UHDAS is not a program, but a complex set of codes and system configurations. Setting up a new UHDAS installation requires Linux system administation skills, an understanding of UHDAS, and detailed knowledge of the particular suite of instruments and network environment on the ship. We are working to make the installation process easier and more automated, but at present it is not sufficiently documented to be done without specific training.

CODAS documentation has accumulated over the years as acquisition programs, instruments, and processing programs have changed. Some of the old (but highly detailed) documentation contains references that are obsolete, but still relevant to old data sets; and many of the problems described in the old documentation still occur.

UHDAS+CODAS Documentation is evolving, as we gradually include more of it under the unifying framework facilitated by Sphinx. The full documentation is available at various entry points:

- http://currents.soest.hawaii.edu/docs/adcp_doc/index.html
- at sea on vessels with UHDAS acqusition systems, at

  http://currents/adcp
- on your computer, if you install the CODAS suite of ADCP processing software

*http://currents.soest.hawaii.edu/docs/adcp_doc/best_practices.html

Data for Scientists

There are three categories of data, all located in the logging directory, /home/data/CRUISEID. The cruise distribution should contain the entire /home/data/CRUISEID directory. The **png_archive** directory in each instrument processing directory is supposed to contain a summary collection of PNG files from the cruise.

This table is a short description of the contents:

subdirectory	contents	importance	back up for ...
raw	all raw data	critical	archiving
rbin	intermediate files	nice to have	anyone who gets 'raw'
gbin	intermediate files	nice to have	anyone who gets 'raw'
proc	processed data • codas database • underway figure archive • matlab files	final at-sea product	science CD after cruise

---

If the cruise name is **"vg0810"** :

- Data are logged under **/home/data/vg0810** on the ADCP linux computer "currents". The active UHDAS cruise directory contains the following subdirectories:
- **raw**: logging goes on here in directories named by instrument
  - ADCP (wh300, nb150, os150, os75, os38)
  - ancillary (gpsnav, gyro, ashtech, posmv, seapath, mahrs, phins...)
  - config (configuration directory)
  - log (diagnostics directory)
- **rbin**: intermediate version of ancillary ascii data in "raw" subdirectory, stored as binary.
  - ancillary only (gpsnav, gyro, ashtech, posmv, seapath, mahrs, phins...)
- **gbin**: time-matched ADCP, navigation, attitude; time in UTC
  - enhancements added for Python processing make the Python-generated gbin directory slightly different from the Matlab-generated gbin directory.
- **proc**: one CODAS processing directory for each instrument+pingtype possible;
  - standard CODAS processing subdirectories
  - config (instrument configurations used at sea)
  - png_archive (figures from the web site)

Cruise Directory Details

UHDAS “raw” (logged) data:

For a ship with an NB150 and an OS38 (for example) in the **raw** subdirectory,

ADCP data

Directories with instrument names (such as nb150 or os38) contain:

- *.raw (binary files with ADCP data)
- *.log (ascii file with time stamps that match each ping)

The raw instrument files are ready to be loaded into Matlab using an overloaded methods read function in your path if you run **radcppath.m** (must specify instrument and logging program). Type **help read** in Matlab for more info.

Serial ascii data:

For example gpsnav, gyro, posmv, ashtech, soundspeed.

All files are ascii, and contain alternating time stamps and messages:

```
$UNIXD, (time stamps)
$NMEA,ascii NMEA message
```

For instruments that send multiple messages, there should be a Unix time stamp for every message, e.g., for the Ashtech there should be:

- $UNIXD ...
- $PASHR,ATT ...
- $UNIXD ...
- $GPGGA ...

Directory tree for raw logging and processing

(example for vg0810), root directory is vg0810, all others below that:

```
vg0810/
      raw/
           config/    (logging configurations:
                          (end-cruise snapshot '/home/adcp/config'))
           log/       (serial data dialogs
                          (end-cruise snapshot of '/home/adcp/log')
           os38/      (raw instrument data)
           nb150/     (raw instrument data)
           gpsnav/    (serial data)
           gyro/      (serial data) ----------------------
           posmv/     (serial data)                       |
           ashtech/   (serial data)                       |
           simrad/    (serial data)                       |
      rbin/                                               |
           gpsnav/    (binary version of serial data)     |
           gyro/      (binary version of serial data) <---
           posmv/     (binary version of serial data)
           ashtech/   (binary version of serial data)
           simrad/    (binary version of serial data)
      gbin/
          ztimefit.txt  (time coefficients for instrument/time) (Python processing)
          os38/
              time/      (time-matching for os38)
              gpsnav/    (time-matched
              gyro/      (            binary
              posmv/     (              version
              ashtech/   (                 of serial
              simrad/    (                       data)
          nb150/
```

```
            time/      (time-matching for nb150)
            gpsnav/    (time-matched
            gyro/      (            binary
            posmv/     (              version
            ashtech/   (                 of serial
            simrad/    (                       data)
        heading/    (all specified headings on one gps timestamp) (Python Processing)
    proc/
        os38bb/        (os38 bb processing directory)
        os38nb/        (os38 nb processing directory)
        nb150/         (nb150 processing directory)
```

where:

- os38 and nb150 are instrument names
- os38bb is for os38 'bb' pings
- os38nb is for os38 'nb' pings
- nb150 is for nb150 pings

UHDAS "rbin" data

As ascii data are logged to serial directories in 'raw/' (e.g. 'raw/ashtech'), a python thread created by UHDAS continually extracts the useful numbers from each NMEA line and appends them to the end of files in a parallel rbin directory (e.g. 'rbin/ashtech').

The raw instrument files are ready to be loaded into matlab using an overloaded methods read function in your path if you run *radcppath.m* (must specify instrument and logging program). Type *"help read"* for more info.

Matlab loadable files for all of the raw ascii nav messages must be created (raw binary or rbin). Each message is parsed and assigned a Unix time stamp. There is a 1:1 correspondence between the raw file name and the rbin file name.

Load into matlab using something like this

```
[rdata, rstruct] = read_bin(file);
```

e.g., for the *$GPGGA* message, the structure looks like

```
rstruct:
rstruct.message
rstruct.rows
{'u_dday'
'dday_frac'
'lon'
'lat'
'qual'
'hdop'
}
```

For an instrument like Ashtech where multiple messages are logged, there is an rbin file created for each type of data, i.e., one for the position (GGA) and one for the attitude (ATT) messages.

UHDAS "gbin" files:

The gbin file name is based on the raw data.

The gbin is the gridded binary (gbin) file where all fields are gridded in time. The times in the instrument

log file (e.g., raw/nb150/*.log) are the times that determine the grid, i.e., the Unix times of instrument pings. The Ashtech, GPS, seapath, heading, soundspeed are all interpolated to the Unix times of the pings and put into gbin files.

Now, the best time must be determined (decisions made to keep time monotonic increasing, and to track any other time corrections).

The results of this go into the files in the time subdir. *.tim.gbin has 3 rows containing: the Unix pingtimes, the “best” times, and the pc_sec-UTC difference. *.best.gbin is the gridded “best” values of position and heading, given that there are multiple sources for these.

The decision whether to use seapath or Ashtech, for example, or which is the primary navigation device, is NOT made by the program on the fly, but rather is set initially at the start of the cruise in procsetup_onship.py.

Python processing adds two elements to this directory: (1) ztimefit.txt (used to correct ADCP time to UTC for each file) (2) heading (contains an independent directory of binfiles with all the

> specified heading devices gridded onto the same time base. QC has already been applied to these files, so (for example) bad Ashtech or POSMV data will be NaNs

---

UHDAS “proc” directory:

The **proc** directory contains one CODAS processing directory per instrument+pingtype. For the example of a ship with an NB150 and an OS38, the processing directories would be

- nb150
- os38bb
- os38nb

Pingtypes (“bb” and “nb”) are processed separately because the singleping editing criteria are different, and because it is a useful diagnostic to compare the two datasets. Averaging and merging can be done by the user after postprocessing.

Inside each processing directory

CODAS processing is documented in more detail here.

The contents of a CODAS processing directory are (in order of processing stages)

directories: pre-processing and metadata

- `config` (metadata and processing configuration)
- `scan` (charactierize timestamps, time range of data)
- `ping` (default data location for pingdata, LTA, ENX (not UHDAS))
- `load` (stage averages in the form of `*.bin`, `*.cmd` files)
- `adcpdb` (contains `*.blk` (the CODAS files))

directories: CODAS processing

- `nav` (positions : reference-layer smoothing of positions here)
- `edit` (editing of averaged (CODAS) data; plot temperature)
- `cal`
    - `rotate` (apply correction (time-dependent or constant))
    - `watertrk` (watertrack calibration calculation)

- `bottomtrk` (bottomtrack calibration calculation)
- `rotate` (testing effects of rotation; staging area)

directories: output for users

- `grid` (staging lon/lat or time grid for data extraction)
- `vector` (coarsly-gridded data, suitable for vector plots)
- `contour` (finely-gridded data, suitable for contour plots)
- `png_archive` (after "end cruise" on UHDAS installations, archive of png files (one per day) from the at-sea web site)

files: metadata and cruise info

- Cruise ID, such as `km1101` is CRUISEID
- Database name, such as `a_km` is DBNAME
- "decimal day" is floating point days of the year, starting with 0.0

file name	contents
`cruise_info.txt`	for the user: documentation template with metadata and processing (for user)
`dbinfo.txt`	parameters used during processing
`CRUISEID.runlog`	output of most recent quick_adcp.py command
`qpy.cnt`	control file used to process these data
`nav/DBNAME.gps`	3 columns with decimal day, lon, lat) • time is 0-based decimal day
`edit/DBNAME.temp`	transducer temperature
`cal/watertrk/adcpcal.out`	watertrack calibration file
`cal/botmtrk/btcaluv.out`	bottom track calibration file
`cal/rotate/ens_hcorr.asc`	heading correction QC file (if relevant)
`adcpdb/DBNAME.tr`	time range of database (YYYY/MM/DD hh:mm:ss)
`adcpdb/DBNAME.lst`	time ranges for each block file