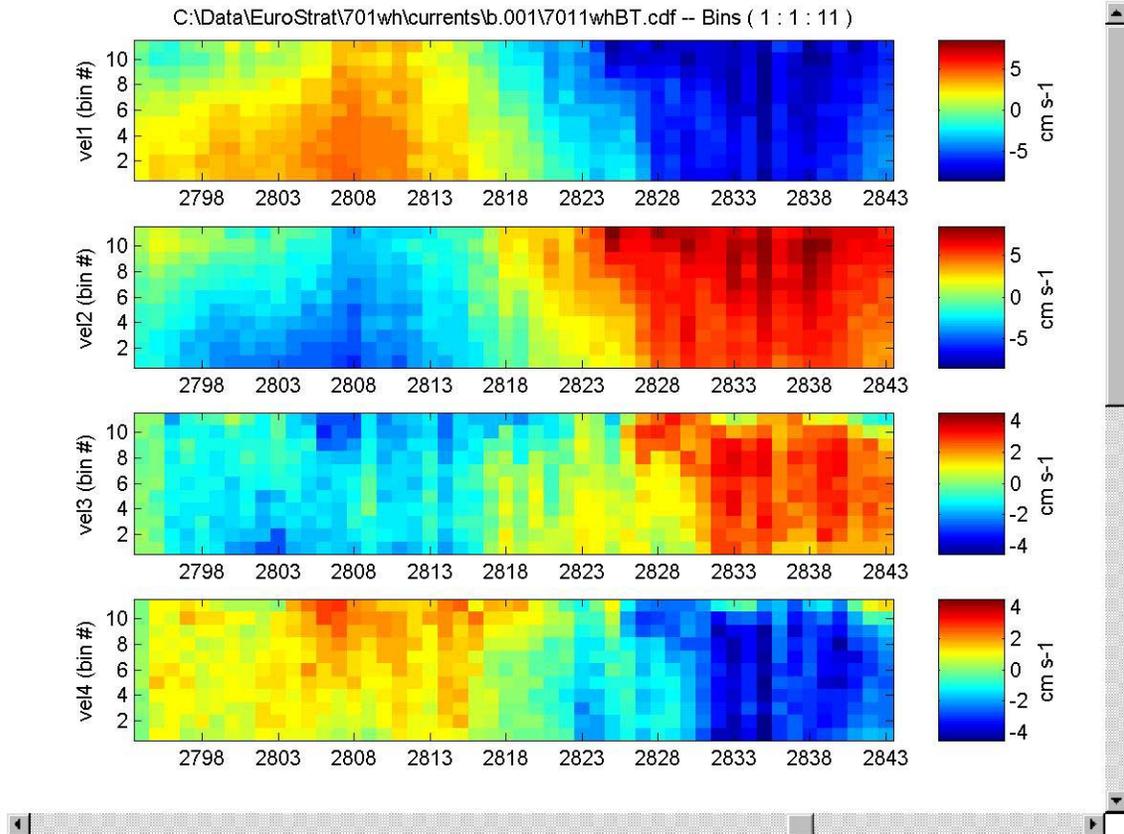


# Acoustic Doppler Current Profiler Data Processing System Manual



Jessica M. Côté, Frances A. Hotchkiss, Marinna Martini, Charles R. Denham  
Revisions by: Andrée L. Ramsey

U.S. Geological Survey Open File Report 00-458 Version 2: May 2003

Sediment Transport Instrumentation Facility  
[Woods Hole Field Center](#), Woods Hole, MA

Section 1. List of Revisions .....	1
Section 2. Overview .....	2
Hardware & Software .....	3
Data Terminology .....	3
Basic Processing .....	4
Program Use.....	5
Special Features .....	5
Deployment log File .....	5
Section 3. Basic Processing .....	7
User Interface.....	7
Setup & Installation .....	8
Naming Conventions .....	9
Getting Started .....	9
Program Summaries.....	10
Program:    runadcp.m.....	11
Program:    fappend.m.....	17
Program:    rdi2cdf.m.....	18
Program:    fixEns.m.....	20
Level:    secondary program.....	20
Program:    runmask.m.....	21
Program:    fillmsk.m.....	23
Program:    StarBare Browser .....	25
Program:    postmask.m.....	28
Program:    goodends.m.....	29
Program:    trimbins.m.....	31
Program:    rdsurface.m.....	33
Program:    adcp2ep.m.....	35
Program:    runbm2g.m.....	37
Section 4. Display and Analysis Programs .....	39
Introduction & Programs .....	39
Program:    StarBeam Browser .....	40
Program:    StarBuck Browser .....	43
Section 5. Tips & Tools .....	47
Instrument setup.....	47
Broad Band Data.....	47
Time Savers .....	47
Runadcp/Rdi2cdf.....	47
Batch Processing.....	48
Example runadcp batch command files .....	49
Example adcp2ep batch command files.....	51
Section 6. Troubleshooting .....	52
Encountering Errors .....	52
Errors in trimBins.m.....	52
Unsuccessful Ensemble trimming.....	53
Section 7. Appendix.....	54
File Inventory.....	54

Startup file.....	57
Diary Log File for Demo file.....	58

## Section 1. List of Revisions

The following is a list of major additions and revisions that have been made to the Acoustic Doppler Current Profiler Data Processing System since its original release in November 2000. Minor revisions include syntax errors in MATLAB version 6.0 and higher. These minor revisions will not be discussed; however, a list of all the functions and their latest revision date can be found in the File Inventory Section of the Appendix.

### Additions:

- Added `fixens.m` function which checks the ADCP data for missing ensemble numbers. If missing ensemble numbers are detected, `fixEns.m` fills in the appropriate ensemble number, time of missing ensemble, and places fill values in for the missing data as place holders.
- Added `fappend.m` function, which concatenates two or more binary files together so that the files may be processed as one file.
- Will now read data that was collected in the downward facing orientation. Note that this will not automatically remove bottom data bins that are below the seabed.
- Added pressure to the list of variables that the ADCP Processing System would read into the file.

### Revisions:

- Formatted the magnetic declination attribute so that it would accept decimal places.
- Revised `adcp2ep.m` so that it shifted the time to the middle of the ensemble. The old `adcp2ep.m` shifted the time so that it was in the middle of the following ensemble.
- Updated batch files to reflect the additional functions.
- The ADCP Processing System calculates depth by first checking for a pressure sensor, and pressure is not available, it will use the `SURFACE.EXE` program for RD Instruments. The `SURFACE.EXE` program is only used for upward oriented ADCPs. Lastly, if the user does not have the `SURFACE.EXE` program, the ADCP Processing System will ask for user input values for the depth and mean tidal range.

## Section 2. Overview

The Sediment Transport Instrumentation Group (STRAIN) at the U.S. Geological Survey (USGS) Woods Hole Field Center has a long-standing commitment to providing scientists high quality oceanographic data. To meet this commitment, STRAIN engineers are vigilant in checking data as well as hardware for signs of instrument malfunction. STRAIN data sets are accompanied by processing histories to detail data processing procedures that may have modified the data signal in the process of removing data noise. The history also enables data to be reprocessed in light of new insight into instrument function and mooring conditions. The USGS ADCP Data Processing System was developed to meet these data quality commitments for Acoustic Doppler Current Profilers (ADCPs) made by RD Instruments (RD Instruments, 1998).

In 1994 the Sediment Transport Instrumentation Group (STRAIN) at the USGS Woods Hole Field Center began deploying RD Instrument ADCPs to sample currents in estuaries, coastal, and continental shelf regions. ADCP data sets are large and complex in structure, so existing STRAIN data processing software was insufficient. In addition, the current ADCP processing programs did not meet STRAIN quality control criteria. The USGS ADCP Data Processing System permits engineers and scientists to monitor data quality by:

- processing data with interactive critical review;
- preserving data quality indicators;
- preserving minimally processed and partially processed versions of data sets.

STRAIN usually deploys ADCPs configured as upward looking and mounted on bottom tripods. When the ADCPs are moored on stable platforms, the velocity profiles are recorded in coordinates parallel to the instrument beams, with minimal internal data processing. ADCP data are recorded in beam coordinates in an effort to adhere to the USGS philosophy to post-process rather than internally process data, preserving the original data quality indicators as well as the initial data set, so that data can be reprocessed if necessary. RDI ADCPs internally screen velocity data and yield an output variable that represents the percentage of pings that are good enough to be incorporated into each ensemble average. The screening is performed on data recorded in either beam or geographic coordinates and includes verification of a minimum acoustic ping correlation value, a beam-to-beam comparison of echo intensity values, and a maximum error velocity filter. If the ADCP is set to record in earth coordinates (geographic coordinates), velocity data are converted into north, east, and vertical components. Data-quality variables, such as percent good, represent averages of all four beams. Recording in beam coordinates retains the values used in the screening for every ensemble of each beam. To increase the accuracy of current measurements, STRAIN ADCP data are recorded in beam coordinates, additional data screening methods are employed during post-processing, and the “cleaned” data are converted into earth coordinates. With each processing step a new data file is created, and the original data file is preserved. This allows the user to be able to step back through the files and identify how data was modified.

This section introduces the USGS ADCP data processing system and provides information is provided about the history of the system's development. The USGS philosophy regarding data quality is discussed to provide an understating of the motivation for creating the system. General information about the following topics will also be discussed: hardware and software required for the system, basic processing steps, limitations of program usage, and features that are unique to the programs.

### Hardware & Software

The ADCP data processing system (ADCP Toolbox) consists of a series of M-files written in the Matlab language by Mathworks and is supported by most computer platforms. The programs have been specifically tested for Unix, Windows 95, 98, 2000, NT, and Macintosh computers. A copy of the ADCP Toolbox can be downloaded from: <http://woodshole.er.usgs.gov/operations/stg/pubs/ADCPtools/index.html>

The current edition of the ADCP Toolbox is Version 5.0. This version has been tested with MATLAB version 6.5, release 13 and should work with MATLAB versions 6.0 and higher. The current release of the ADCP Toolbox has not been tested with MATLAB versions prior to 6.0, and currently we are unable to support the ADCP Toolbox 5.0 on earlier releases of MATLAB.

The data processing system was developed to address the issues associated with data recorded in beam coordinates, but data recorded in earth coordinates may also be processed using these routines. The input data file must conform to RD Instruments' format outlined in Appendix D of the RDI manual (RDI, 1998). This format is binary and specifies the order in which the data appears. During the "deployment" procedure for the RDI ADCP, a deployment log file is created containing the ADCP hardware settings. A similar log file is recorded by the ADCP during data collection. The information in these files is required by the system to transform the data from beam to earth coordinates.

After initial input the data are converted to the NetCDF data format and subsequent processed files are stored in the NetCDF format. The NetCDF toolbox created for Matlab is required to utilize the ADCP data processing system. This toolbox is provided by Dr. Charles Denham and can be downloaded from: [http://woodshole.er.usgs.gov/staffpages/cdenham/public\\_html/MexCDF/nc4ml5.html](http://woodshole.er.usgs.gov/staffpages/cdenham/public_html/MexCDF/nc4ml5.html).

In the future the system may be modified to produce processed data files in \*.mat or ASCII format. The option to extend the system for use on Sontek ADPs is being investigated.

### Data Terminology

The types of variables referenced throughout this document are defined below. They are required components of the input data file.

The RDI data file is comprised of three sections:

*Header data* – information about the data file type, source, structure, and size

*Fixed variable leader data* – the non-dynamic data including the deployment setup parameters and the system configuration.

*Variable leader data* - the dynamic data recorded by the clocks and sensors that change with each ping.

The four major data variables that are recorded for all four beams are:

*Velocity* – velocity measured parallel to the beam in mm/s

*Correlation* – the signal to noise ratio used as a measure of data quality

*Echo intensity* – a measure of the signal strength received by the ADCP

*Percent good* – the percent of pings that passed the data rejection criteria

Once the data are transformed to earth coordinates the velocities' definition changes and an error is inferred:

*Velocity* – the east-west, north-south, and vertical components of velocity

*Error velocity* – difference between the two estimates of vertical velocity

ADCPs produce a time series of vertical velocity profiles. Internal processing resolves the acoustic doppler signal into a set of “bins”, each of which represents a layer of water defined by the acoustic travel time from the head of the transducer. These are interpreted as depth levels, and may include apparent ghost levels above the sea surface that result from reflection. The ADCP is normally configured to record a time series of internally averaged bursts of pings that are averaged and recorded as a single interval. Each of the resulting intervals is given an incremental record number as well as a time stamp. The time is recorded in Julian days.

### Basic Processing

The first stage of the processing is to transfer the data from the instrument to the computer. In the RDI ADCPs the data are recorded on PCMCIA Flash Memory cards. The data can be downloaded using the RDI recover program or copied from the PCMCIA card directly to the desktop.

It is important to establish a naming convention that will be carried throughout the data processing steps. Using either download method the files will be saved with the deployment file name given in the initial setup of the instrument. The USGS uses a naming convention that incorporates the mooring number, followed by a number indicating the instrument position on the mooring, and letters indicating the type of instrument. As of Spring 2003, the USGS STRAIN group indicates an RDI Workhorse ADCP with the letters “wh” and the RDI Broad Band ADCP with the letters “bb”. In most cases the deployment name given to the ADCP will be in this convention. In the event that the filename is not in this form, the raw data file name should be changed to meet this convention. In each subsequent major processing step a new file is created. A letter is added to the end of the original deployment name that indicates the type of

processing that was performed on the data file. In Section 2 the naming conventions used throughout the processing system are demonstrated by example.

The fully processed data files meet the EPIC NetCDF standard developed by the NOAA Pacific Marine Environmental Laboratory (<http://www.pmel.noaa.gov/epic/>). These EPIC standards provide a universal vernacular allowing researchers from different organizations to share oceanographic data without needing translation. The final data file is our Best Basic Version (BBV) indicating that it is clean of erroneous values, converted into earth coordinates, and compliant with the outlined standards. As of Spring 2003, data from the bins closest to the sea surface are included in the BBV, although many are erroneous. The depth of the first good bin changes with changes in the water elevation primarily due to tides. Until more sophisticated methods of cleaning the near-surface data are instituted, scientists will need to exercise judgment in interpreting the layers.

### Program Use

Several people have written the programs collected into the ADCP data processing system, from slightly different approaches. This system was developed to be modular so that this collection of routines has essentially four levels: primary programs, secondary programs, tertiary programs and sub-programs. The primary programs are essentially the interface routines that gather input and output file information, and in some instances write a new file. The secondary and tertiary programs do the processing, which may all be run independently of the primary programs. The sub-programs typically perform data formatting, behind the scenes, and will be listed at the end of a primary or tertiary program description in Section 2.

The system was developed for users of all programming abilities and with the intention of being user friendly. The result is a set of routines that requires a significant amount of interaction. In order to fully benefit from the features of this system it is recommended that it be run in the interactive format. All prompts have a default answer provided and it is essential that the user response follow the format of the default answer explicitly. In Section 4, some tips for the advanced user will be provided to increase the efficiency of processing data in the interactive format. A more advanced technique of batch processing has been developed requiring no user interaction. Batch processing requires prior knowledge about the data set that the interactive programs do not. Batch processing will be discussed in Section 4.

### Special Features

#### *Deployment log File*

When an RDI ADCP is deployed using the DOS “DEPLOY” program a log file is created that contains the codes sent to the ADCP, hardware settings, and the results of each command. The deployment log filename is of the form \*.dlg. A log file is also recorded by the ADCP during data collection. This is the “recovery” log file and is of the form \*.log. These files contain beam configuration information that is required by the

processing routines to transform raw beam coordinate data into earth coordinates. If you cannot locate the \*.dlg file, one can be created using the PS3 command in *DumbTerm* and using F3 to log the output. The essential component required for coordinate transformation has this format:

```
>14 MAR 1999 19:33:07.23 Sent command (PS3) with ADCP response:  
Beam Width: 3.7 degrees
```

Beam	Elevation	Azimuth
1	-70.00	270.00
2	-70.00	90.00
3	-70.00	0.01
4	-70.00	180.00

In the future we hope to provide the option for users to override the request for a deployment log file by specifying azimuths and elevations.

If the data are collected in earth coordinates, the coordinate transformation is not needed; in which case a log file is not requested. Processing data from RDI broadband model ADCPs is another special case that does not require a deployment log file and will be addressed more specifically in Section 4.

## Section 3. Basic Processing

The basic ADCP data processing system consists of a series of programs that converts data into NetCDF format by adding important metadata; editing the data based on quality, time constraints, and depth; transforming the data into geographic coordinates; and by creating an EPIC compatible data file. The final file created is designated as the Best Basic Version (BBV) of the data.

Although this system addresses several data problems, there will always be exceptions. Data sets that have unusual problems or require further adjustment must be handled on a case-by-case basis and are beyond the scope of this manual.

The names of the M-files that comprise this series of basic processing programs are listed below. To distinguish the program levels they are listed in outline form where roman numerals indicate primary programs, capital letters denote secondary programs, numbers signify tertiary programs, and lower case letters are used for sub-programs.

- I. Primary program ~description
  - A. Secondary program ~description
    - 1. Tertiary program ~description
      - a. Sub-program ~description

### User Interface

- I. runadcp.m -Runs the programs that perform the quality checks on the data

### Binary Data to NetCDF

- A. rdi2cdf.m ~To transfer data from the RDI output file to NetCDF format
  - 1. rdhead.m ~ Reads RDI header data format
  - 2. rdflead.m ~ Reads RDI fixed leader data
  - 3. rdvlead.m ~ Reads RDI variable leader data

### Editing and Quality Assessment

- B. fappend.m ~ Concatenates multiple binary files into one file for processing
- C. fixEns.m ~ Checks data for missing ensembles and fills in place holders for any missing ensemble numbers and data
- D. runmask.m ~ Creates a mask file by imposing limits on the data values and applies the mask to the data
  - 1. ncmkmask.m ~ Creates a file identical in structure to the raw data file but filled with 0's
  - 2. fillmsk.m ~Sets the limits of the criteria for masking
    - a. premask.m ~ Scans the raw data file and marks the mask file for data values that exceed masking criteria

3. starbare.m ~Interactive browser allows user to mark additional bad data points.
4. postmask.m ~ Applies the mask file to the data file creating a new file

E. goodends.m ~Removes bad data from the beginning and end of the file

F. trimbins.m ~removes data bins that represents levels above the sea surface

1. rdsurface.m ~runs the RDI DOS surface program to find the mean sea level and half the tidal range

II. adcp2ep.m ~Translates data into variables that are in terms of earth coordinates and creates an EPIC compatible data file

#### Coordinate Transformation

A. runbm2g.m~creates the input and output variables to prepare for coordinate transformation

1. bm2geo.m ~converts beam data to geographic coordinates

#### Setup & Installation

Prior to any ADCP program setup, the NetCDF toolbox must be downloaded and installed as mentioned in Section 1. On page A-1 of the Appendix is a file inventory of the ADCP toolbox that lists the names of all files and their location. The following instructions will take the user through the setup for all of the programs of the ADCP data processing system. The programs are distributed in a folder called ADCP\_tools and should be downloaded in binary format. It is suggested that all of these files be placed in a single folder in the Matlab/toolbox directory on the users computer so the programs may be updated by replacing the entire folder.

To run the system, you will need an additional folder of programs written by Chuck Denham and a couple of programs adopted from Rich Signell. These can be found in the folder ADCP\_tools/AddOns. There are 8 p-code installers that must be downloaded as pure binary. Put the p-codes in your Matlab toolbox directory, at the Matlab prompt change the directory to the toolbox folder (where the p-code files are) and type the name of the p-code, for example "batch\_install". A new folder within the toolbox will be created with each of these p-codes: batch, epic, form, presto, proxy, starbare, starbeam, and starbuck. Many of these toolboxes do behind the scenes work and will never be called directly by the user. Some of these programs and toolboxes will not be discussed until later sections. If you currently have any of these toolboxes please replace them with the current versions. It is recommended that these toolboxes be placed within your Matlab toolbox area.

Once the installers are run and the programs are extracted into folders, you must adjust your Matlab path to reflect these additions. If you have a startup.m file, usually created during the NetCDF toolbox installation process, add the path for the ADCP\_tools and the eight-toolbox folders. If you do not have a startup.m file to set paths upon

Matlab program startup, create one using the example on page 6-4 and place it in the Matlab/bin directory.

### Naming Conventions

In the examples used to illustrate how to run these programs there are certain naming conventions that facilitate the finding and interpretation of data files. As discussed previously, the names are composed of mooring number, position number and instrument specific letters. As an example *9991wh* represents a workhorse ADCP as the uppermost instrument on the mooring #999. A mooring number is a sequential number given to every instrument package that is deployed by the STRAIN group, whether it is a surface buoy, an instrumented mooring, or an instrumented tripod. The ADCP on a tripod will almost always be the uppermost instrument and distinguished by a 1 for position. Continuing with this example for the naming of data files:

9991wh000.000	binary raw data file recorded by the ADCP
9991wh.cdf	NetCDF raw data file (or appended files) translated by rdi2cdf.m
9991whF.cdf	NetCDF raw data file with missing values filled in by fixEns.m
9991wh.msk	the mask file output by runmask.m/fillmsk.m
9991whM.cdf	the new data file after masking output by runmask.m/postmask.m
9991whT.cdf	the trimmed data file output by goodends.m, and modified by trimbins.m/rdsurface.m
	(Note: Downward oriented data will not have a *T.cdf file)
9991wh.nc	EPIC NetCDF file output by adcp2ep.m (BBV)

### Getting Started

At this point all hardware and software requirements should have been met and the processing programs and toolboxes should be installed. As the user, you are almost ready to begin processing data. As a first run, use this system in its interactive format. This requires the user to call only the 2 primary programs. To prepare for processing, the user should gather pertinent data information and files that will be requested during processing (see below). Please review the notes on runadcp.m on pages 2-5 to 2-10 and adcp2ep.m on pages 2-30 to 2-32 to learn what information will be needed. When initially running these programs it is suggested that the user keep a "diary" file (see "help diary" in matlab). The ADCP processing programs output many messages to the Matlab screen and a log of these messages will be helpful to track problems if they arise. Example diary logs can be found in the Appendix, starting on page 6-5.

The following steps can be used to process your data.

1. Place binary data file and log file in a folder that will be used to save subsequent processed data files.
2. Gather information that will be requested.  
For runadcp:

Mooring number  
Deployment date  
Recovery date  
ADCP serial number  
Distance between the ADCP transducers and the sea bed  
Predicted accuracy of the velocity measurements given by PLAN prior to deployment  
Amount of time the ADCP clock was slow  
Magnetic variation at the mooring location in degrees

For `adcp2ep`:

Experiment name  
Project name  
Description of data  
Comments about data  
Longitude in decimal degrees  
Latitude in decimal degrees

3. Start Matlab and change directory to the folder containing the data files.
4. Type “`diary filename`” at the Matlab prompt.
5. Type “`runadcp`” and follow the prompts.
6. At the end of `runadcp`, the starbeam browser will open. Use this opportunity to check over the newly processed data file. If the new data file contains bad values, see *Troubleshooting* in Section 5.
7. Type “`adcp2ep`” and follow the prompts.
8. Type “`starbuck`” and select the appropriate file to view the BBV of the data.

Your data should now be devoid of erroneous values and in a format that is ready to be analyzed further. For an average data file that contains three months of data, these steps could take several hours to process (depending on the size of your computer). Please pay attention to the prompts as well as messages that appear in the Matlab window. The following pages of this section detail this process. The example run for each function lists the prompts as they appear in the Matlab window. A double asterisk (\*\*) indicates an instruction to the user, and will not appear as a Matlab prompt.

### Program Summaries

The following pages list summaries and descriptions of each of the functions used by the ADCP Toolbox.

Program:	runadcp.m
Level:	primary program
Purpose:	To provide the interface for input and output of information required to run the programs that perform the quality checks on the data.
Command:	[theResult] = runadcp ('rawdata',... 'rawcdf', 'theMaskFile', 'theNewADCPFile', 'trimFile')
Description:	<p>This program is used as an organizational tool to gather input file names, build output file names, and check the data types for adherence to the restrictions imposed by this data processing system. All of the secondary programs executed within runadcp.m are stand alone programs and will be discussed individually below. Although they may all be run independently of runADCP.m, by increasing the number of files input to the command line of runadcp.m you may also skip programs that have already been run. More information on running this program at the command line is given in the section entitled "some helpful hints on runadcp". As mentioned, this processing system was designed primarily for data from RDI ADCP Workhorse instruments recorded in beam coordinates. Although it will handle data in Earth coordinates, the runmask and StarBare programs will be skipped.</p>

Most of the prompts described below result from the secondary programs that are doing the processing as indicated in the left column. Therefore, the notes of the prompts are brief and more details can be found in the description of the secondary programs.

Example run:

<u>Program</u>	<u>Prompt</u>	<u>Response</u>
fappend	1. How many binary files used?	1
rdi2cdf	2. Select Binary ADCP File:	9991wh000.000
	3. Save NetCDF ADCP File As:	9991wh000.cdf
	4. Input mooring data for ADCP:	
	A. Enter the mooring number	9991
	B. Enter the deployment date	13-aug-1999
	C. Enter the recovery date	17-aug-1999
	5. Input metadata from the mooring log	
	A. Enter the ADCP's serial number	185

	B. Enter the distance between the ADCP transducers and the sea bed in meters	1.25
	C. Enter the predicted accuracy given by PLAN in cm/s	1.0
	D. Enter the amount of time the ADCP clock was slow by in seconds	35
	E. Enter the magnetic variation at the mooring location in degrees	-16
Starbeam	6. StarBeam Setup	
	**Before answering this dialogue box, read the instructions at the prompt in the Matlab command window.	
	A. Ensemble	<i>defaults</i>
	B. Bin	<i>defaults</i>
	C. Time Axis	'Rec'
	D. Depth Axis	'bin'
	E. Plot Style	'image'
	F. Rotation Angle	0
	G. Color factor	1
	7. Prior to StarBeam, the Matlab prompt read: "In the following figure quickly review the data. Then click "Done" on the Starbeam menu and hit enter"	Done
	In pause mode – execute command anyway? **In the Matlab command window	Yes hit enter
fixEns	8. Save filled ADCP File As: **If no missing ensembles were found, this file will not be used	9991whF.cdf
runmask	9. Save masked ADCP File As: **If data is in earth coordinates skip to prompt 11	9991whM.cdf
StarBare	10. StarBare setup	
	**Before answering this dialogue box, read the instructions at the prompt in the Matlab command window.	
	A. Ensemble	<i>defaults</i>
	B. Bin	<i>defaults</i>
	C. Time Axis	'Rec'
	D. Depth Axis	'bin'
	E. Plot Style	'image'
	F. Rotation Angle	0
	G. Color factor	1
	11. Prior to StarBare, at Matlab prompt:	

	“In the following figure, bad velocity data are displayed in black. Mark any additional data points, then click “Done” on the Starbare window and hit return”	Done
	In pause mode – execute command anyway? **In the Matlab command window	Yes hit enter
goodends	12. Save the Trimmed ADCP File As:	9991whT.cdf
	13. Date of last good ensemble **This prompt will only appear if the date for tilt and recovery are different. Choose Date	Date1 for Tilt Date2 for Recovery
trimbins	14. Defining Water Depth Do you have the RDI surface program?	1. No (mac or Unix) 2. Yes (pc user)
	13-1. User must input the water depth information mean sea level value units half the tidal range value units (skip to prompt 16)	30 meters 1.5 meters
	13-2. Where is the RDI surface program? (proceed to prompt 14)	c:\Programs\RDI\
	15. A figure appears displaying the depths calculated by surface for each ensemble. Check surface program output: Minimum depth in meters: Maximum depth in meters: Are these depths reasonable?	17.35 18.62 N
	16. Input valid range for depth Minimum depth allowed: Maximum depth allowed:	17.8 18.62
Starbeam	17. StarBeam Setup A. Ensemble B. Bin C. Time Axis D. Depth Axis	<i>defaults</i> <i>defaults</i> ‘Rec’ ‘bin’

E. Plot Style	'image'
F. Rotation Angle	0
G. Color factor	1

---

Notes on prompts:

- Prompt 4: It is very important that the deployment and recovery dates be entered in the exact date format as the default! If this information is not available or appropriate, default or obviously bad values should be used.
- Prompt 5: If the response for A, the adcp serial number, is less than 100 or the response to B, the distance above the bed, is greater than 100, the user will be prompted again for the same inputs.
- C. Prior to deployment the RDI PLAN program is used to set the command values given to the ADCP, see the results of the choices, and create a command file. One of the expected values given by PLAN is predicted accuracy of the horizontal velocity measurement in cm/s.
- D. When the instrument is recovered the ADCP clock is checked against the wall clock, and the difference between the two is the amount the ADCP clock is slow by. This number represents the mechanical error of the internal ADCP clock. If there was a mistake setting the time of the instrument to the correct time format, such as the difference between GMT and EST, the time should be post-corrected after rdi2cdf.m is run and not reflected in the slow\_by attribute. If the ADCP clock is faster than the wall clock (very unusual) use a negative sign to indicate that the clock was fast rather than slow.
- E. The magnetic declination is used to post-correct the instrument heading. In the rdi2cdf.m program this adjustment will impact data that is recorded in beam coordinates only. If the data has been recorded in earth coordinates, the heading information has been applied internally in the instrument and the currents will be rotated in adcp2ep.m. Positive numbers are used for degrees clockwise from North and negative degrees are used for degrees counter clockwise from North. In the example run -16 indicates a declination of 16 degrees counter-clockwise from North.
- Prompt 6: For additional information about the StarBeam set up menu and function, the user should refer to the description of StarBeam in Section 3, page 3-1.
- Prompt 7: This statement will be in the Matlab window for 5 seconds prior to the launching of the StarBeam set up window. Once the information for StarBeam has been submitted this prompt will be scrolled out of view. After the user has completed their review of the data in the StarBeam browser, the window maybe closed by clicking "Done" on the StarBeam

menu of the figure window or using the x in the upper right corner of the window. Always respond “yes” to the pause mode question. Be sure to hit the enter key after answering yes or the program will not continue.

- Prompt 10: For additional information about the StarBare set up menu, the user should refer to the StarBare program on page 2-19.
- Prompt 11: Similarly to prompt 7, this message will only be seen prior to the submission of the information for the StarBare browser. After the user has completed their editing of the data in the StarBare browser, the window may be closed by clicking “Done” on the StarBare menu of the figure window or using the x in the upper right corner of the window. Always respond “yes” to the pause mode question. Be sure to hit the enter key after answering yes or the program will not continue.
- Prompt 13: This dialog box is part of the goodends.m program. If there is a large time span between the recovery date for the mooring and the date derived by looking at the tilt record, then the user is requested to choose a date.
- Prompt 14: The RDI surface program is a DOS based routine (i.e. only runs on PCs) that finds the water depth based on echo intensity. If the user does not have this program they must be prepared to provide a depth and half the tidal range for the mooring location.
- Prompt 14-1: Results from answering *No* to prompt 14. After prompt 14-1, the program will skip to prompt 17.
- Prompt 14-2: Results from answering *Yes* to prompt 14. If the path of the surface program cannot be found then a second dialog box will be invoked stating that the path is incorrect and requesting that the user correct the path.
- Prompt 15: In the “check surface program output” dialog box, all three fields can be changed. If the minimum and maximum values are ok, then answer ‘Y’ to “Are these depths reasonable?” and the program will continue. If the depths are not reasonable based on the figure shown, the user may either answer ‘N’ and prompt 16 will appear or the user may change the minimum and maximum depths in the current dialog box and the depth constraints will be modified accordingly.
- Prompt 17: After the user has completed their review of the data in the StarBeam browser, the window maybe closed by clicking “Done” on the StarBeam menu of the figure window or using the x in the upper right corner of the window.

---

Some helpful hints about runADCP.m:

If you want to minimize the user interaction required by this function, give it a full list of filenames when you call the function. Here is an example:

```
[theResult] = runADCP('9991wh000.000','9991wh000.cdf','9991whF.cdf','9991wh000.msk', ...  
'9991whM.cdf', '9991whT.cdf')
```

If 9991wh000.cdf is not an existing file, then you will see two dialog boxes almost immediately asking for the mooring log information. If 9991wh000.cdf already exists then you will proceed directly to masking. If you have an existing NetCDF file, then type in the first two inputs in the command of runadcp and it will skip rdi2cdf, which is the most time intensive function of the program.

Depending on file size this program could take an hour to run. The window will update itself, to indicate the time that has elapsed and the number of ensembles that have been processed. Although runADCP.m does have two pauses, it cannot yet be run with "pause off" due to the star browser functions.

Program: fappend.m  
 Level: secondary program  
 Purpose: fappend.m concatenates two or more ADCP binary files together.  
 Command: fappend('outfile', 'infile\_1', infile\_2', ...)  
 Description: On occasion, the ADCP will record the data into several raw binary files, instead of one. In these cases, the user may want to concatenate the files before processing begins. This function is used within the runadcp.m and trimbins.m functions.

Example Run:

	<u>Prompt</u>	<u>Response</u>
1.	How many binary files used?	1
2.	Select Binary ADCP File?	999wh000.000

OR

1.	How many binary files used?	2
2.	Select 1 <sup>st</sup> Binary ADCP File:	999wh000.000
3.	Select 2 <sup>nd</sup> Binary ADCP File:	999wh000.001

Program: rdi2cdf.m  
 Level: secondary program

Purpose: rdi2cdf.m extracts the raw data from the binary file, converts and catalogs the data into the NetCDF format.

Command: rdi2cdf('rawdata', 'rawcdf', minens, maxens)

Description: The RDI data records have 3 parts; header data, fixed variable leader data, and variable leader data. Each of these data types is handled independently by one of the tertiary programs; rdhead.m, rdfilead.m, or rdvlead.m. These 3 programs were created to read the exact structure of the RDI binary data file as detailed in the RDI manual, Appendix D (1998). It is beyond the scope of this document to provide further details of these tertiary programs. When rdi2cdf is run independently of runadcp, the filenames must be provided such that the call statement for this function is rdi2cdf(rawdata,rawcdf). We suggest using the rawdata file name with a \*.cdf extension for the rawcdf filename. This makes it obvious where the NetCDF file was derived, see example below.  
 theResult = runADCP('9991wh000.000','9991wh000.cdf').

## Example Run:

	<u>Prompt</u>	<u>Response</u>
1.	Input mooring data for ADCP: A. Enter the mooring number B. Enter the deployment date C. Enter the recovery date	999 01-jan-1999 01-jan-2000
2.	Input metadata from the mooring log A. Enter the ADCP's serial number B. Enter the distance between the ADCP transducers and the sea bed in meters C. Enter the predicted accuracy given by PLAN in cm/s D. Enter the amount of time the ADCP clock was slow by in seconds E. Enter the magnetic variation at the mooring location in degrees	185 1.25 1.0 35 -16

---

Notes on prompts:

Prompt 1: It is very important that the deployment and recovery dates be entered in the exact date format as the default! If this mooring information is not

available or useful for the user, default or obviously bad values should be used.

Prompt 2: If the response for A, the adcp serial number, is less than 100 or the response to B, the distance above the bed, is greater than 100, the user will be prompted again for the same inputs.

C. Prior to deployment the RDI PLAN program is used to set the command values given to the ADCP, see the results of the choices, and create a command file. One of the expected values given by PLAN is predicted accuracy of the horizontal velocity measurement in cm/s.

D. This value represents of the error associated with the internal ADCP clock. If there was a mistake setting the time of the instrument to the correct format during set up, such as the difference between GMT and EST, the time should be post-corrected after rdi2cdf.m is run.

E. This information is used to post-correct the instrument heading for local magnetic declination. In the rdi2cdf.m program this adjustment will impact data that is recorded in beam coordinates only. If the data has been recorded in earth coordinates, the heading information has been applied internally in the instrument and the currents will be rotated in adcp2ep.m. Positive numbers are used for degrees clockwise of North and negative degrees are used for degrees counter clockwise of North. In the example run -16 indicates a declination of 16 degrees counter-clockwise of North.

---

Notes on rdi2cdf.m:

Rdi2cdf.m can also be used to create a subset of the full data file by including 2 additional inputs; minens, the minimum ensemble number, and maxens, the maximum ensemble number. As an example the call statement would then read theResult = runADCP('9991wh000.000','9991wh000.cdf',100,1100).

---

Tertiary Programs: rdhead.m, rdflead.m, rdvlead.m  
 Sub-programs: history.m

- Program: fixEns.m
- Level: secondary program
- Purpose: fixEns.m checks the ADCP data for missing ensemble numbers. If missing ensemble numbers are detected, fixEns.m fills in the appropriate ensemble number, time of missing ensemble and places fill values in for the missing data as place holders.
- Command: `[missEns] = fixEns('rawcdf', 'theFilledFile')`
- Description: On occasion, the ADCP will not record a data ensemble because the data that was measured was not accurate. fixEns.m counts through all the record numbers to see if any ensembles are missing. If all ensembles are present, the function stops. If missing ensemble numbers are found, the function will determine how many ensembles are missing, the number of the missing ensemble, and time of the missing ensemble. The function creates a dummy array of fill values for each variable in the dataset, and places these arrays in the dataset at the time of the missing ensemble number as place holders. It also fills in the ensemble number and time of the ensemble. This function does not write over the rawcdf file but creates a new netcdf file.

Example Run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP Data File:	9991wh000.cdf
2.	Select ADCP Filled File	9991whF.cdf

---

Sub-programs: history.m

Program: runmask.m  
 Level: secondary program

Purpose: This program uses the additional variables measured by the ADCP to determine good velocity measurements. The criteria for data quality are imposed on the data to ensure they meet the restrictions, and the data file is then edited.

Command: [theNewADCPFile, theMaskFile] = ...  
 runmask ('rawcdf', 'theMaskFile', 'theNewADCPFile')

Description: Runmask.m is the second secondary program that runadcp calls, and is called with the above command statement. If the raw NetCDF file name, the mask file name or the new ADCP file name are not specified in the call statement the user will be prompted by dialog boxes. Runmask.m creates a mask file which is identical to the raw NetCDF file in size and name with a \*.msk extension, and fills the variables with 0's. The variables in the raw NetCDF file are then scanned to meet the limits set for the velocity, correlation, echo intensity, and percent good criteria (done by a tertiary program fillmsk.m). If these criteria are not met then the velocity for that ensemble and bin is marked with a 1 indicating it is bad. After the data are scanned and the mask is filled, the starbare browser appears to provide the opportunity to mark additional data points as bad. Matlab is in pause mode and will not continue runadcp until after the starbare window is closed and a key is struck. Runmask.m then calls postmask.m to apply the mask file (of logical indices) to the *rawcdf* file producing *theNewADCPFile* which is suggested to have a name like \*M.cdf. If this function was run successfully it will produce *theMaskFile* (\*.msk) file and the *theNewADCPFile* (\*M.cdf).

Note: If the data are in earth coordinates runadcp will skip the masking step. In earth coordinate files the percent good variable represents an average of all four beams, which complicates masking.

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select NetCDF ADCP File:	9991wh000.cdf
2.	Save Masked ADCP File As:	9991whM.cdf
3.	StarBare setup	
	A. Ensemble	<i>defaults</i>
	B. Bin	<i>defaults</i>
	C. Time Axis	'Rec'

D. Depth Axis	'bin'
E. Plot Style	'image'
F. Rotation Angle	0
G. Color factor	1

---

Notes on prompts:

- Prompt 2: The '000' portion of the file name is typically dropped at this stage to allow room for additional letters to be added to the file name indicating the completion of subsequent processing steps.
- Prompt 3: The starbare browser will be described later in this section giving more details on the prompts. Typically the defaults can be submitted and scroll bars will appear to enable viewing of the complete data set. It is suggested that the time axis be set to view by record and the depth axis by bin since these are more meaningful variables at this stage of processing.

---

Notes on runmask:

Each of the four variables is scanned to ensure it falls within certain limits. These limits are extracted from the settings of the ADCP, which are recorded as attributes in the raw NetCDF data file. It is possible for the user to input the limits for masking, but this requires breaking the programs down into several steps. This will be further mentioned in the description of fillmsk, a tertiary program.

Although runmask.m utilizes several tertiary programs, fillmsk.m and the starbare browser are the only 2 that will be described further.

---

Tertiary programs: ncmkmask.m, fillmsk.m, starbare directory (which should include premask.m and postmask.m)

Sub-programs: history.m

Program: fillmsk.m  
 Level: tertiary program

Purpose: Using the minimum and maximum limits for the four ADCP variables, this program scans the data file and then marks the mask file with 1's where the data does not meet the criteria.

Command: [theMaskFile,velR,corT,echI,Pgd] = ...  
 fillmsk('theDataFile','theMaskFile',velR,corT,echI,Pgd);

Description: This program is the main engine of runmask.m and as mentioned earlier masking functions can only be run on beam coordinate data at present. Fillmsk can be run with minimal input as it is called by runmask. If no inputs are provided to the program, then the program extracts the minimum and maximum values allowed for velocity, correlation, echo intensity, and percent good from the raw NetCDF data file. These ranges are listed in the binary data file recorded by the RDI ADCP. Direct use of this program, however, enables the user the ability to input the ranges used for scanning the data for bad values. All inputs must be defined explicitly in order to use the program at the command line. Additionally, the mask file must have been created using ncmkmask.m prior to running fillmsk, if the user intends to specify variable ranges. The four variables, velR, corT, echI, and Pgd, require 2 values each in the form, [min max]. Once the ranges are defined either by the user or by the program, premask.m is invoked to actually perform the scanning of the data and the filling of the mask file. The following prompts are seen when no inputs are given.

Note: Currently fillmsk.m is only used for data recorded in beam coordinates. Future versions will handle the change in the percent good variable in earth coordinate files.

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP Data File:	9991wh000.cdf
2.	Select ADCP Mask File	1. 9991wh000.msk 2. cancel
3-2.	Do you want to create a mask file now?	Yes

---

Notes on prompts:

- Prompt 2: If a mask file has already been created then, at this point it can be selected and submitted. Otherwise, the user should select cancel and the next prompt will appear.
- Prompt 3-2: This prompt is given when a mask file does not yet exist. If 'yes' is entered the program will create a mask file with the same name as the input file but the .msk extension. If 'no' is entered the program will terminate without creating or filling a mask file.

---

Sub-programs: ncmkmask.m, premask.

Program: StarBare Browser  
 Level: tertiary program

Purpose: The StarBare Browser displays a data file overlain by a mask file, and enables the user to manually edit the data.

Command: StarBare('theDataFile', 'theMaskFile')

Description: StarBare requires a NetCDF data file and a corresponding mask file (\*.msk). The mask file is identical in structure and size to the data file, but filled with 1's and 0's. The StarBare interactive browser overlays the data file with the mask file to screen out bad data points. The browser offers slider-controls for scrolling through the data with a point and click capability to mark additional bad data points. The window contains four panels, for showing four data-types for a single beam, or one data-type for four beams. When velocities for earth-coordinates are shown, the panels represent from top to bottom: u (positive east), v (positive north), w (positive up), and the inferred error. The user may remove data points by clicking on a single point or area with the mouse. This action can be similarly undone by highlighting the same area again with the mouse.

The StarBare menubar is for selecting the variables to be shown: velocity, correlation, echo intensity, percent-good, and tilt. The data can be displayed by beam or by type. Other menus such as *Graph*, *TimeAxis*, and *Depth* offer easy ways to change the display without having to go back to the "StarBare Setup" dialog.

StarBare can be run independently of the other masking functions. The mask file must be created by ncmkmask.m prior to running StarBare. Ncmkmask is a command line function, and information on running it can be obtained by typing "help ncmkmask" at the Matlab command prompt.

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP Data File:	9991wh000.cdf
2.	Select ADCP Mask File	9991wh000.msk
3.	StarBare setup	
	A. Ensemble	<i>defaults</i>
	B. Bin	<i>defaults</i>
	C. Time Axis	'Rec'
	D. Depth Axis	'bin'

E. Plot Style	'image'
F. Rotation Angle	0
G. Color factor	1

---

Notes on prompts:

Prompt 1. In the dialog box, the default file type setting in the bottom of the frame is M-files, \*.m. To see all files, use the pull down menu to select all files.

Prompt 2. In the dialog box, the default file type setting in the bottom of the frame is M-files, \*.m. To see all files, use the pull down menu to select all files.

Prompt 3A. The response *defaults* is used to indicate that the user should use the default settings of the setup menu to display all of the data.

The "Averaging" checkbox causes the displays to show ensemble averages if checked. The averaging width is given by the "Step" field. The default is unchecked (no-averaging).

The "Start" field is the first record shown in the plot and may be a record-number, or a valid Matlab date string in quotes. See "help datestr" and "help datenum" for formats. The default is to start at the first record.

The "Count" field is the number of records seen in the window, and the increment used when scrolling through the data. It may be a record-count, or a "dhms" (day-hour-minute-second) time-string in quotes. See "help dhms2d" and "help d2dhms". The default is to show 50 records.

The "Step" field is the number of records in an ensemble average if applicable. It may be a record-count, or a "dhms" time-string in quotes. The default step is 1 record.

The "Ensemble Max" field shows the number of records in the file. Do not change this field.

The "Ensemble Time Sampling" field shows the time sample-interval as a "dhms" time-string. Do not change this field.

Prompt 1B. The "Bin Start, Count, Step, and Max" fields are used for selecting the particular ADCP bins to show, by bin number. The default behavior is to show all bins, thus the response *defaults* to the prompt in the example run.

Prompt 1C. The "Time Axis" field allows the horizontal scaling to be expressed as time or record-number. The default is to show time.

- Prompt 1D. The "Depth Axis" field allows the vertical scaling to be expressed as depth or bin number. The default is to show depth.
- Prompt 1E. The "Plot Style" field offers several choices for the display style to be shown. The default is "image".
- Prompt 1F. The "Rotation" angle for earth-coordinates is given in degrees. A positive angle rotates the horizontal data counter-clockwise, equivalent to rotating the coordinate system clockwise. For example, to correct for a magnetic declination of 16 degrees West, use +16. The default angle is 0. Remember that this data has probably already been corrected for magnetic declination. This option is only appropriate for data recorded in earth coordinates.
- Prompt 1G. The "Color Factor" is for color-scaling the vertical velocity in an image. Zero (0) allows the vertical velocity image to range over the full color-map. A value of one (1) forces the same color scaling for all velocities. A larger number amplifies the vertical velocity scaling by the given factor. The default is 1. This option is only appropriate for data recorded in earth coordinates.

---

Notes on StarBare:

StarBare is a folder of programs or a toolbox as well as a \*.m file that is called to bring up the browser. The folder contains the suite of sub-functions that are required to run this browser.

In the ADCP NetCDF files "ensemble" is a dimension, and "Records" ("Rec") is a variable with the ensemble dimension. Each record contains one ensemble that is typically an average of more than one ping. In the Star Browsers the "averaging" checkbox can be selected to show "ensemble averages". This option displays the average of more than one record averaged over the ensemble dimension.

Program: postmask.m  
 Level: tertiary program

Purpose: This program applies a mask file to a data file in order to remove data that does not meet the quality standards.

Command: postmask('theADCPFile', 'theMaskFile', 'theNewADCPFile')

Description: After the mask has been filled by fillmsk either from runmask or independently, postmask must be run to modify the beam coordinate data file. The mask file is a logical array where 0's indicate good data points and 1's denote bad data points. When applied to the data file, the mask file acts as a screen, to produce a new data file where the bad data points have been replaced with fill values. The fill value used is specified as a variable attribute in the NetCDF file. Similarly to the other programs postmask may be run interactively or at the command line. Postmask requires three input file names, which are self explanatory from the command statement. It is suggested that the 000 portion of the name be dropped for the new ADCP File and a M be added at the end to represent that the data has been masked (see example names in responses below).

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP File	9991wh000.cdf
2.	Select Mask File	9991wh000.msk
3.	Save As ADCP File:	9991whM.cdf

---

Program: goodends.m  
 Level: secondary program

Purpose: This program is used to evaluate the ends of the data record to determine the first and last good ensembles. Any ensembles that fall outside of these limits are then removed.

Command: [minens, maxens, nens, trimFile] = ...  
 goodends('theBeamFile', 'theMaskFile', 'trimFile')

Description: Goodends is the third secondary program to be called by runadcp. Goodends.m uses a 3 step process to determine the first good ensemble (minens) and the last good ensemble (maxens) out of the total number of ensembles (nens).

a) The program first calculates the total number of bins that have a valid velocity data point for each ensemble by looking at the mask file. The first bin where more than 75% of the bins are good is selected as minens. The bin that is closest to the end of the record and has more than 75% of the bins with a velocity data point is maxens.

Note: If the data are in Earth coordinates then no mask file exists and step a) is skipped. Then by default minens starts at the ensemble corresponding to the deployment date, and the first guess for maxens is the ensemble corresponding to the recovery data.

b) Next the pitch and roll records are evaluated between the minens and maxens ensembles to find when the ADCP settled on the bottom and was lifted off the seafloor. This is determined by a pitch and roll that varies by less than 2 degrees. Based on this information a new minens and maxens is selected.

c) Finally the date of the minens and maxens is compared to the deployment and recovery dates. Although the ensembles also have a time, they are only compared to the recovery and deployment by date. If the date of the first good ensemble does not match the deployment date then the program defaults to the first ensemble corresponding to the deployment date. If the date corresponding to the last good ensemble does not match the recovery date a dialog box appears and asks which date you would like to use.

For data that are recorded in earth coordinates this program will get the correct date, but not necessarily the correct time of the good end ensembles. This results in a couple of bad ensembles at the ends of the data set.

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP Netcdf File:	9991whM.cdf
2.	Save the Trimmed ADCP File As:	9991whT.cdf
3.	Select ADCP Mask File:	9991wh000.msk
4.	Date of last good ensemble Choose Date	Date1 for Tilt Date2 for Recovery

---

Notes on prompts:

Prompt 3: You will only be asked to select a mask file if the ADCP Netcdf file given was in beam coordinates.

Prompt 4: If the date derived by looking for a significant change in tilt does not match the recovery date a dialog box appears requesting the user to choose the appropriate date.

---

Notes on goodends:

If the data are in earth coordinates and the user wishes to call goodends from the command line, an empty string must be input for the mask file in the call statement. Example: `goodends('9991wh000.cdf', ' ', '9991whT.cdf')`. Also note that a \*.cdf file will not have been produced since masking is skipped for earth coordinate data, therefore theNewADCPFile is the raw NetCDF file.

---

Sub-programs: gregorian.m, julian.m, history.m

Program: trimbins.m  
 Level: secondary program

Purpose: Trimbins removes data in bins that corresponds to layers above the mean sea level. This program is intended for upward oriented data only.

Command: [MSL, Dstd] = trimBins('rawdata', 'trimFile', MSL, Dstd)

Description: TrimBins.m is used to modify the upward oriented data file such that all the bins fall below the mean sea level (MSL) plus a standard deviation (Dstd). The mean sea level and the standard deviation must be given or are calculated based on the output of the DOS based RDI surface program, which is run on the raw binary data file by rdsurface.m. In runadcp the user is asked if you have the RDI surface program and proceeds accordingly. The ADCP Principles of Operation Manual (pg. 38) states that the top 6% of the data can be contaminated due to the strong echo from the sea surface. Therefore, be aware that this program will cut off any bins that appear to be out of the water, or within the top 6% of the surface. Depth bins that exceed 94% of the mean sea level (MSL) plus half the tidal range (Dstd) will be removed. Trimbins removes bins in every ensemble. If you do not have the RDI surface program the total depth and standard deviation, which can be approximated as half the tidal range, at the mooring location must be provided in the command line in trimbins. Note that the total depth refers to the entire water column, and the transducer height above bottom should be ignored for this value.

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	How many binary files used?	1
2.	Select Binary ADCP File:	9991wh000.000
3.	Select Ensemble trimmed ADCP File:	9991whT.cdf
4.	Where is the RDI surface program?	c:\Programs\RDI\
5.	A figure appears displaying the depths calculated by surface for each ensemble. Check surface program output:	
	Minimum depth in meters:	17.35
	Maximum depth in meters:	18.62
	Are these depths reasonable?	N
6.	Input valid range for depth	

Minimum depth allowed:	17.8
Maximum depth allowed:	18.62

- |                   |                 |
|-------------------|-----------------|
| 7. StarBeam setup |                 |
| A. Ensemble       | <i>defaults</i> |
| B. Bin            | <i>defaults</i> |
| C. Time Axis      | 'Rec'           |
| D. Depth Axis     | 'bin'           |
| E. Plot Style     | 'image'         |
| F. Rotation Angle | 0               |
| G. Color factor   | 1               |

---

Notes on prompts:

- Prompt 1. This refers to the number of binary files that were converted to netCDF format in rdi2cdf.m.
- Prompt 2. The binary file is the input for the RDI DOS surface program. However, the depths will only be displayed for the ensemble numbers in the trimmed data file.
- Prompt 3. This file is simply modified and no new file is created after bin trimming has occurred.
- Prompt 4. If the path of the surface program can not be found then a second dialog box will be invoked stating that the path is incorrect and requesting that the user correct the path.
- Prompt 5. In the "check surface program output" dialog box, all three fields can be changed. If the minimum and maximum values are ok, then answer 'Y' to "Are these depths reasonable?" and the program will continue. If the depths are not reasonable, the user may either answer 'N' and prompt 5 will appear or the user may change the minimum and maximum depths in the current dialog box and the depth constraints will be modified accordingly.

---

Notes on trimbins:

In this program a new file is not created, but the trimmed ADCP file is simply modified. Therefore, just because a \*T.cdf file exists does not mean that trimbins has been run. If trimbins is run successfully then it will be reflected in the *history* attribute of the NetCDF file.

---

Tertiary programs: rdsurface.m  
 Sub-programs: history.m

Program: rdsurface.m  
 Level: tertiary program

Purpose: This program runs the RDI surface program to estimate an approximate mean sea level for upward oriented data files.

Command: [MSL, Dstd, Dout] =...  
 rdsurface('rawdata', ADCP\_offset, ensembles, progPath, 'DepthFile')

Description: Rdsurface.m is called by trimbins to calculate the water depth from the binary data. In order to run this program the user must be working on a PC that has the RDI surface program installed. The surface program looks at the echo intensity in the bins of each beam for each ensemble to determine the locations of the water surface. Surface is run on the entire data set as given in the raw data file. Rdsurface.m sub-samples the output of the surface program to extract the depths only for the ensembles that have been determined to be of good quality. However, some of the depth estimates from surface may still be inaccurate and the user is given the opportunity to check these depths during the program. Rdsurface outputs all of the good depth estimates (Dout), the average of the good depths as the mean sea level (MSL), and the half the tidal range (Dstd).

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	How many binary files used?	1
2.	Select Binary ADCP File:	9991wh000.000
3.	Select NetCDF ADCP File to find ADCP offset:	9991wh000.cdf
4.	Where is the RDI surface program?	c:\Programs\RDI\
5.	A figure appears displaying the depths calculated by surface for each ensemble. Check surface program output:	
	Minimum depth in meters:	17.35
	Maximum depth in meters:	18.62
	Are these depths reasonable?	N
6.	Input valid range for depth	
	Minimum depth allowed:	17.8
	Maximum depth allowed:	18.62

---

Notes on prompts:

- Prompt 1. This refers to the number of binary files that were converted to netCDF format in rdi2cdf.m.
- Prompt 2. The binary file is the input for the RDI DOS surface program. However, the depths will only be displayed for the ensemble numbers given by “ensembles”.
- Prompt 3. In order to determine the mean sea level, the program needs to know the height of the instrument off the bottom, which is represented as the ADCP offset in the NetCDF files. The ADCP offset can be given in the command line or extracted from a NetCDF file.
- Prompt 4. If the path of the surface program can not be found then a second dialog box will be invoked stating that the path is incorrect and requesting that the user correct the path.
- Prompt 5. In the “check surface program output” dialog box, all three fields can be changed. If the minimum and maximum values are ok, then answer ‘Y’ to “Are these depths reasonable?” and the program will continue. If the depths are not reasonable, the user may either answer ‘N’ and prompt 5 will appear or the user may change the minimum and maximum depths in the current dialog box and the depth constraints will be modified accordingly.

---

Sub-programs: surface.exe (RDI, 1997)

Program: adcp2ep.m  
 Level: primary program

Purpose: This program is used to translate ADCP data into variables that are in terms of earth coordinates and create an EPIC NetCDF data file.

Command: `epDataFile = adcp2ep('adcpFile', 'epDataFile', 'ADCPTtype', 'dlgFile')`.

Description: Adcp2ep is the next program in the ADCP data processing system to be run after the ADCP data are processed by runadcp.m (or the individual programs that comprise the editing quality assessment). This program can be run on both beam and earth coordinate data files. Beam coordinate data are transformed into earth coordinates by the secondary program, runbm2g.m that actually calls bm2geo.m to do the real work. Earth coordinate data are simply read directly and the variables are renamed from numbered velocities to u, v, w etc. If a magnetic declination was given to the rdi2cdf.m program (through runadcp.m) it will be stored in the *heading bias* attribute. The Earth coordinate data will be rotated to account for magnetic declination in adcp2ep.m.

In the EPIC NetCDF data file the echo intensity, correlation, and percent good variables are averaged for all 4 beams producing a single variable for each parameter. Additional things such as converting “TIM” to time and time2, shifting the time to correspond to the center of the ensemble, and adjusting the depth to be the true water depth of the bin are handled in this program. Likewise, the units of velocity are converted from mm/s to cm/s.

If adcp2ep.m is being run in sequence after runadcp.m then adcpFile = trimFile in the command line. The ‘ADCPTtype’ input is used to distinguish data from an ADCP “workhorse” from an ADCP “broad band” instrument. This is discussed further in Section 4, page 4-1.

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP File:	9991whT.cdf
2.	Save data in geographic coordinates as:	9991wh.nc
3.	Select ADCP Deployment Log File:	9991wh.dlg OR 9991wh.log

4.	Data Collection Information	
	A. Mooring	9991
	B. Experiment	'Mass Bay Longterm'
	C. Project	'MWRA'
	D. Descript	'Boston Buoy B ADCP'
	E. Comments	<i>additional data info</i>
	F. Longitude	
	Value	-70.5529
	Format	decimal degrees
	Units	degree east
	G. Latitude	
	Value	42.2470
	Units	degree north

---

Notes on prompts:

- Prompt 1. The ADCP file provided should have been processed using runadcp.m, and bad data points removed. According to the format followed in runadcp.m the file will have a name in the format \*T.cdf (the trim file).
- Prompt 2. The Best Basic Version (BBV) file resulting from this program is typically given the \*.nc extension to distinguish it from the other NetCDF data files.
- Prompt 3. The deployment log file is a product of the RDI deployment program. When an ADCP is programmed and “deployed” by the computer, the beam elevations and azimuths specific to that instrument will be recorded in this file.
- Prompt 4.
- A. All instruments deployed by STRAIN are given a sequential mooring number and is recorded in the initial NetCDF file created by rid2cdf.m
  - B. This is simply the common name of the experiment.
  - C. The Project usually describes the main funding agency or field office.
  - D. Descript is a description of the type of data that is in the NetCDF file. In this case it will always be *something* ADCP.
  - E. Any other comments about the data should be included here.
  - F. The longitude should always be given in decimal degrees, please do not change the format of the longitude. Values should be entered and the units selected according to the recorded position of the mooring.
  - G. See longitude, same applies for latitude.

---

Notes on adcp2ep.m:

The most time consuming portion of this program is the bm2geo conversion where the data are rotated into Earth coordinates. A data file cur.mat is created each time this portion of the program is run. In the event that the program crashes before the epDataFile is produced but after the data are transformed, cur.mat can be read in at line 129 of adcp2ep.m.

---

Secondary programs: runbm2g.m, ep\_time.m, gregorian.m, uv\_rotate.m, history.m

Program: runbm2g.m  
Level: secondary program

Purpose: This program gathers the information and reformats the structure of the data to prepare for the transformation from beam to earth coordinates.

Command: cur = runbm2g('BeamFile', 'ADCPtype', 'dlgFile')

Description: Runbm2g.m provides the interface between adcp2ep.m and bm2geo.m for data in beam coordinates. This program extracts the compass heading, the pitch, and the roll information from the input beam coordinate file to apply corrections for orientation of the instrument. It also reads the elevations and azimuths for the ADCP instrument from the deployment log file to be used in the transformation matrix. The orientation of the instrument (up or down) as well as the blanking distance is also entered from the attributes of the input NetCDF file. After gathering this information the runbm2g.m calls bm2geo.m to convert the data into geographic coordinates ensemble by ensemble. A cell matrix of currents is output by this program as well as saved in a file called cur.mat in the working directory.

If running this program at the command line the ADCP type must be specified as BB (broad band) or WH (Workhorse). In interactive format the program will assume that the data are from the Workhorse instrument. More information is given in Section 4 on processing data from an RDI broad band model ADCP. See below in notes on runbm2g for list of optional command line inputs.

## Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP File in Beam coordinates:	9991whT.cdf
2.	Select ADCP Deployment Log File:	9991wh.dlg OR 9991wh.log

---

Notes on prompts:

- Prompt 1. The ADCP file given should be devoid of bad data points as a result of being passed through the filtering routines of runadcp.m. According to the format followed in runadcp.m, the file will have a name in the format \*T.cdf (the trim file).
- Prompt 2. The deployment log file is a product of the RDI deployment program. When an ADCP is programmed and “deployed” by the computer, the beam elevations and azimuths specific to that instrument will be recorded in this file.

---

Notes on runbm2g.m:

The primary purpose of runbm2g.m is to read the data from the file into bm2geo.m in the proper format. bm2geo.m can be run independently, but it is not recommended (see help bm2geo.m in Matlab for more information). There are other optional inputs for runbm2g.m that are used in running bm2geo.m seen in bold in the following command line.

```
cur = runbm2g(BeamFile, ADCPtype, dlgFile, theElevations, theAzimuths,  
theHeading, thePitch, theRoll, theOrientation, theBlankingDistance)
```

There are four beam elevations and azimuths hard-wired into the ADCP instrument. The heading, pitch, and roll data are collected by the ADCP in a time series throughout the deployment. The orientation can be upward looking (Up) or downward (down). The blanking distance is the distance above that head of the ADCP where the acoustic signal is contaminated by ringing. This parameter is dictated by the firmware loaded into the instrument.

---

Tertiary program: bm2geo.m

## Section 4. Display and Analysis Programs

### Introduction & Programs

Several tools have been developed that allow the user to display and discriminate the data in NetCDF format. These viewers are part of the Star\* suite of toolboxes developed by the USGS for ADCP data. StarBeam has been mentioned and is used in the runadcp.m program to view data in non-EPIC format. StarBare was fully described in Section 2 for use in further editing bad data points. The last Star\* program is Starbuck which displays the BBV of the data with EPIC variables. Each Star\* program is not only a function, but also a folder of programs (or toolbox) that are required to run the viewers. The Star\* toolboxes are included with the ADCP Toolbox download.

- Program: StarBeam Browser
- Purpose: The StarBeam Browser is used to display an ADCP data file with variables created by rdi2cdf.m and a time base in Julian days.
- Command: StarBeam('theDataFile', 'thePermission')
- Description: Starbeam is both a directory and a function. StarBeam opens 'theDataFile' of ADCP measurements for browsing, with 'thePermission': 'write' or 'nowrite' (default). If no filename is given, or if a wildcard is given a filename will be requested through a dialog box. The "starbeam" object is returned or placed in the workspace as "ans". The "StarBeam Browser" window offers slider-controls for scrolling through the data. The window contains four panels, for showing four data-types for a single beam, or one data-type for four beams. When velocities for earth-coordinates are shown, the panels represent from top to bottom: u (positive east), v (positive north), w (positive up), and the inferred error.
- The StarBeam menu bar is for selecting the variables to be shown: velocity, correlation, echo intensity, percent-good, and tilt. The data can be displayed by beam or by type. Other menus offer easy ways to change the display without having to interact with the "StarBeam Setup" dialog. Details on the menus in the browser window are provided in the Notes on StarBeam below.

Example run:

	<u>Prompt</u>	<u>Response</u>
1.	Select ADCP File	9991wh.cdf
2.	StarBeam setup	
	A. Ensemble	<i>defaults</i>
	B. Bin	<i>defaults</i>
	C. Time Axis	'Rec'
	D. Depth Axis	'bin'
	E. Plot Style	'image'
	F. Rotation Angle	0
	G. Color factor	1

---

Notes on prompts:

- Prompt 1. In the dialog box, the default file type setting in the bottom of the frame is M-files, \*.m. To select the \*cdf files, use the pull down menu.

- Prompt 2A. The response *defaults* is used to indicate that the user should use the default settings of the setup menu to display all of the data.
- The "Averaging" checkbox causes the displays to show ensemble averages if checked. The "Step" field gives the averaging width. The default is unchecked (no-averaging).
- The "Start" field is the first record shown in the plot and may be a record-number, or a valid Matlab date string in quotes. See "help datestr" and "help datenum" for formats. The default is to start at the first record.
- The "Count" field is the number of records seen in the window, and the increment used when scrolling through the data. It may be a record-count, or a "dhms" (day-hour-minute-second) time-string in quotes. See "help dhms2d" and "help d2dhms". The default is to show 50 records.
- The "Step" field is the number of records in an ensemble average if applicable. It may be a record-count, or a "dhms" time-string in quotes. The default step is 1 record.
- The "Ensemble Max" field shows the number of records in the file. Do not change this field.
- The "Ensemble Time Sampling" field shows the time sample-interval as a "dhms" time-string. Do not change this field.
- Prompt 2B. The "Bin Start, Count, Step, and Max" fields are used for selecting the particular ADCP bins to show, by bin number. The default behavior is to show all bins, thus the response *defaults* to the prompt in the example run.
- Prompt 2C. The "Time Axis" field allows the horizontal scaling to be expressed as time or record-number. The default is to show time.
- Prompt 2D. The "Depth Axis" field allows the vertical scaling to be expressed as depth or bin number. The default is to show depth.
- Prompt 2E. The "Plot Style" field offers several choices for the display style to be shown. The default is "image".
- Prompt 2F. The "Rotation" angle for earth-coordinates is given in degrees. A positive angle rotates the horizontal data counter-clockwise, equivalent to rotating the coordinate system clockwise. For example, to correct for a magnetic declination of 16 degrees West, use +16. The default angle is 0. Remember that this data has probably already been corrected for magnetic declination and will be recorded in the history attribute.

Prompt 2G. The "Color Factor" is for color-scaling the vertical velocity in an image. Zero (0) allows the vertical velocity image to range over the full color-map. A value of one (1) forces the same color scaling for all velocities. A larger number amplifies the vertical velocity scaling by the given factor. The default is 1.

---

#### Notes on StarBeam:

Once the StarBeam browser is initiated the <Starbeam> and <colormaps> menu bars can be used to select other image options. A descriptive list of the menu selections is provided below.

<Starbeam> menu "Setup" will bring up the original StarBeam setup dialog box.

The top selections of the "Graph" menu are essentially the same as the "Plot Style" options in the setup dialog described in notes on Prompt 2E. Below the plot styles are selections to change the visual features of the browser window; averaging, colorbars, page setup, and update. The print option is also provided under this menu.

"Time Axis" and "Depth Axis" are described above in Prompts 2C and 2D.

"Beam" can be used to plot all four variables, velocity (vel), correlation (cor), echo intensity (AGC), and percent good (PGd) for each one of the four beams.

A single variable for multiple "Beams" are displayed with the next selection. The default is "velocity" showing vel1, vel2, vel3, and vel4.

"Instrument" allows the user to view tilt and voltage information recorded by the ADCP during deployment.

To close the StarBeam window either select "Done" on the Starbeam menu or click the right hand corner of the figure window.

<colormaps> menu has 16 selections which are different colormaps that the user may select. The default colormap is "jet".

"help" prints information about the colormap menu in the Matlab command window.

The next 2 selections, "remember" and "restore" are respectively used to place a copy of the current color map in the Matlab memory, and to return to the colormap placed in memory. If remember is not used, restore will always return to jet, the default colormap.

The last selection, "done" is used to complete the colormap selections.

- Program: StarBuck Browser
- Purpose: The StarBuck Browser is used to display an ADCP data file that has EPIC time and variables.
- Command: StarBuck('theDataFile', 'thePermission')
- Description: Starbuck is both a directory and a function. StarBuck opens 'theDataFile' of ADCP measurements for browsing, with 'thePermission': 'write' or 'nowrite' (default). If no filename is given, or if a wildcard is given a filename will be requested through a dialog box. The "starbuck" object is returned or placed in the workspace as "ans". The "StarBuck Browser" window offers slider-controls for scrolling through the data. The window contains four panels, for showing four data-types for a single beam, or one data-type for four beams. Velocities for earth-coordinates are shown, and the panels represent from top to bottom: u (positive east), v (positive north), w (positive up), and the inferred error.
- Once the browser is open, the StarBuck menu bar can be used to select the variables to be shown, such as velocity data combinations. Other menus offer easy ways to change the display without having to interact with the "StarBuck Setup" dialog.

Example run:

<u>Prompt</u>	<u>Response</u>
1. Select ADCP EPIC file:	9991wh.nc
2. StarBuck setup	
A. Help	
B. Ensemble	<i>defaults</i>
C. Bin	<i>defaults</i>
D. Time Axis	'Rec'
E. Depth Axis	'bin'
F. Plot Style	'image'
G. Rotation Angle	0
H. Color factor	1

---

Notes on prompts:

- Prompt 1. In the dialog box, the default file type setting in the bottom of the frame is M-files, \*.m. To select the \*cdf files, use the pull down menu.
- Prompt 2A. Help in the setup window gives the same information displayed when "help starbuck" is typed at the Matlab prompt.

- Prompt 2B. The response *defaults* is used to indicate that the user should use the default settings of the setup menu to display all of the data.
- The "Averaging" checkbox causes the displays to show ensemble averages if checked. The "Step" field gives the averaging width. The default is unchecked (no-averaging).
- The "Start" field is the first record shown in the plot and may be a record-number, or a valid Matlab date string in quotes. See "help datestr" and "help datenum" for formats. The default is to start at the first record.
- The "Count" field is the number of records seen in the window, and the increment used when scrolling through the data. It may be a record-count, or a "dhms" (day-hour-minute-second) time-string in quotes. See "help dhms2d" and "help d2dhms". The default is to show 50 records.
- The "Step" field is the number of records in an ensemble average if applicable. It may be a record-count, or a "dhms" time-string in quotes. The default step is 1 record.
- The "Ensemble Max" field shows the number of records in the file. Do not change this field.
- The "Ensemble Time Sampling" field shows the time sample-interval as a "dhms" time-string. Do not change this field.
- Prompt 2C. The "Bin Start, Stop, Count, Step, and Max" fields are used for selecting the particular ADCP bins to show, by bin number. The default behavior is to show all bins, thus the response *defaults* to the prompt in the example run. The "Bin Stop" field is ignored at present.
- Prompt 2D. The "Time Axis" field allows the horizontal scaling to be expressed as time or record-number. The default is to show time.
- Prompt 2E. The "Depth Axis" field allows the vertical scaling to be expressed as depth or bin number. The default is to show depth.
- Prompt 2F. The "Plot Style" field offers several choices for the display style to be shown. The default is "image". The "contour" option superimposes contours on the existing image.
- Prompt 2G. The "Rotation Angle" for earth-coordinates is given in degrees. A positive angle rotates the horizontal data counter-clockwise, equivalent to rotating the coordinate system clockwise. For example, to correct for a magnetic declination of 16 degrees West, use +16. The default angle is 0.

Remember that this data has probably already been corrected for magnetic declination.

Prompt 2H. The "Color Factor" is for color-scaling the vertical velocity in an image. Zero (0) allows the vertical velocity image to range over the full color-map. A value of one (1) forces the same color scaling for all velocities. A larger number amplifies the vertical velocity scaling by the given factor. The default is 1.

---

#### Notes on StarBuck:

Once the starbuck browser is initiated the <Starbuck> and <colormaps> menu bars can be used to select other image options. A descriptive list of the options provided through these menus is below.

<Starbuck> menu "About" gives the program version information.

"Setup" will bring up the original StarBuck setup dialog box.

The "Velocity" can be used to plot different permutations and combinations of the velocity data. The default is four panels with U-V-W-Err, but this can be decreased to two panels with just U-V.

The "Quality" factors are average echo intensity (AGC), average percent good (PGd), and the vertical velocity error (Werr). This option results in a three panel plot showing the quality factors.

The top selections of the "Graph" menu are essentially the same as the "Plot Style" options in the setup dialog described in notes on Prompt 2F. Below the plot styles are selections to change the visual features of the browser window; averaging, colorbars, page setup, and update. The print option is also provided under this menu.

Likewise, "Time Axis" and "Depth Axis" are described above in Prompts 2D and 2E.

To close the StarBuck window, either select "Done" on the Starbuck menu, or click the right hand corner of the figure window.

<colormaps> menu has 16 selections which are different colormaps that the user may select. The default colormap is "jet".

"help" prints information about the colormap menu in the Matlab command window.

The next two selections, "remember" and "restore" are respectively used to place a copy of the current color map in the Matlab memory, and to return to the colormap placed in memory. If remember is not used, restore will always return to the default colormap jet.

The last selection, “done” is used to complete the colormap selections.

## Section 5. Tips & Tools

### Instrument setup

There are a few things the user should remember when setting up the ADCP instrument prior to deployment that will enhance the use of this processing system.

1. If possible the instrument should be setup to record in Beam coordinates. This may not be desirable for instruments not mounted on fixed platforms.
2. The user should note if a heading bias was given to the ADCP prior to deployment. When requested for the magnetic declination during `runadcp.m`, 0 should be entered since the data will be rotated as collected.
3. Save the deployment log file generated by the RDI deploy program. If the deployment log file is not available, the azimuths and elevations for the specific ADCP must be obtained through communication with the instrument using the PS3 command. The default settings for the elevations and azimuths may be used as well, see Broad Band Data below.

### Broad Band Data

`Runadcp.m` will process data from an RDI Broad Band model ADCP without any difficulty. The only modification that needs to be made is in `adcp2ep.m`. For broad band data the ADCP type must be specified to be Broad Band in `adcp2ep.m`, otherwise the program will assume the data are from a workhorse. This is accomplished by using the call statement:

`adcp2ep(adcpFile, epDataFile, ADCPtype)`, specifying the input and output file names and setting `ADCPtype` to `BB`. Since the broad band instrument has a perfect beam configuration, the defaults are used, the elevations are all  $-70$  and the azimuths are exactly 270, 90, 0, and 180.

### Time Savers

The amount of time that it takes to process an ADCP data set can be reduced by the following methods.

#### *Runadcp/Rdi2cdf*

The most time consuming portion of `runadcp` is the initial conversion of the data from binary to NetCDF format. This conversion actually occurs through the `rdi2cdf` function. `Rdi2cdf` can be run independently of `runadcp` and only requires user interaction at the very beginning of the program. See the description of the `Rdi2cdf.m` program in Section 2 for details on running this program. Once the initial NetCDF is produced, the file can be used in `runadcp` to perform the data quality checking steps. The first two input

filenames for runadcp must be specified at the command line in order to use the program in this manner.

### *Batch Processing*

Batch processing was developed so that the ADCP data processing programs could be run with minimal user intervention. Parameters are input directly from a batch file and browsers for inspection of the data are suppressed. Two interactive transactions are retained: the use of StarBare to examine the automatically-generated masking file, and optionally mask additional points; and the user check of water depths derived by the RDI surface program, if it used. For problematic data files with large amounts of bad data points, batch processing is not useful.

The Batch toolbox should have been installed with the other p-code installers as discussed in Section 2. In order to batch process an ADCP data file, a batch command file must first be generated. The batch command file can be saved with the extension \*.bch. This file contains all the pertinent information that would be entered by the user if the programs were run in interactive format. A separate command file is required for runadcp.m and adcp2ep.m. These files must be **exactly** as seen in the following example below, with no empty lines between input statements, appropriate percent signs for comments, and single quotes around the appropriate inputs.

The command line to run a batch file is: batch('demo.bch') where 'demo.bch' is the file name.

### Example runadcp batch command files

- The command file below will allow runadcp to run to completion. This batch file does not use the SURFACE.EXE file.

<u>Example</u>	<u>Input description</u>
runadcp	%program to be run by batch
1	%number of Raw Binary Files
'9991wh000.000'	%first raw binary data file
'	%second raw binary data file is exist
'9991wh000.cdf'	%non-existent raw NetCDF data file
'9991whF.cdf'	%raw Filled data file
'9991wh000.msk'	%the mask file
'9991whM.cdf'	% the Masked data file
'9991whT.cdf'	%the trimmed data file
'999'	%Mooring number
'13-aug-1999'	%Deployment date
'17-aug-2000'	%Recovery date
185	%ADCP serial number
1.25	%height above bottom of ADCP
1.0	%predicted accuracy in cm/s
35	%ADCP clock was slow by in sec
-15	%magnetic declination
'No'	%do you have the surface program
30	%mean sea level in meters
3	%half the tidal range in meters

---

- If the raw NetCDF file already exists the command file appears as below. Note if using the SURFACE.EXE program, see the following batch file description.

<u>Example</u>	<u>Input description</u>
runadcp	%program to be run by batch
1	%number of Raw Binary Files
'9991wh000.000'	%first raw binary data file
'	%second raw binary data file is exist
'9991wh000.cdf'	%existing raw NetCDF data file
'9991whF.cdf'	%raw Filled data file
'9991wh000.msk'	%the mask file
'9991whM.cdf'	% the Masked data file
'9991whT.cdf'	%the trimmed data file
'No'	%do you have the surface program**
30	%mean sea level in meters
3	%half the tidal range in meters

---

3. The command file below will allow runadcp to run to completion and uses the SURFACE.EXE file.

<u>Example</u>	<u>Input description</u>
runadcp	%program to be run by batch
1	%number of Raw Binary Files
'9991wh000.000'	%first raw binary data file
'	%second raw binary data file is exist
'9991wh000.cdf'	%non-existent raw NetCDF data file
'9991whF.cdf'	%raw Filled data file
'9991wh000.msk'	%the mask file
'9991whM.cdf'	% the Masked data file
'9991whT.cdf'	%the trimmed data file
'999'	%Mooring number
'13-aug-1999'	%Deployment date
'17-aug-2000'	%Recovery date
185	%ADCP serial number
1.25	%height above bottom of ADCP
1.0	%predicted accuracy in cm/s
35	%ADCP clock was slow by in sec
-15	%magnetic declination
'Yes'	%do you have the surface program**
'c:\Programs\RDI'	%Surface program path

---

\*\*With these inputs runadcp will pause in the middle of the trimbins program when the figure displaying the depths for each ensemble appears. The program will remain in pause until the user responds to the "Check surface program output" dialog. (See trimbins in Section 2).

## Example adcp2ep batch command files

1. For data files recorded in beam coordinates.

<u>Example</u>	<u>Input description</u>
adcp2ep	%program to be run by batch
'9991whT.cdf'	%input data file
'9991wh.nc'	%output EPIC data file
'WH'	%ADCP type
'9991wh.dlg'	%deployment log file
'Mass Bay Longterm'	%Experiment
'MWRA'	%Project
'Boston Buoy B ADCP'	%Description of data
-70.5529	%longitude
'degree_east'	%longitude units
42.2470	%latitude
'degree_north'	%latitude units
'additional information'	%Data comment

---

2. For data files recorded in earth coordinates.

<u>Example</u>	<u>Input description</u>
adcp2ep	%program to be run by batch
'9991whT.cdf'	%input data file
'9991wh.nc'	%output EPIC data file
'WH'	%ADCP type
'Mass Bay Longterm'	%Experiment
'MWRA'	%Project
'Boston Buoy B ADCP'	%Description of data
-70.5529	%longitude
'degree_east'	%longitude units
42.2470	%latitude
'degree_north'	%latitude units
'additional information'	%Data comment

---

\*Only difference between the batch command file for beam data and earth data is there is no deployment log file input

## Section 6. Troubleshooting

### Encountering Errors

If `runadcp.m` or `adcp2ep.m` stops before completion as the result of an error, the user may be able to begin processing where the program left off. The History attribute in the NetCDF file contains a log of the programs that have been successfully run on the data file. The comments are in reverse chronological order such that the program most recently completed appears first. If `runadcp` or `adcp2ep` have not been completed then the user may use the history information to run the failed processing programs independently.

The History attribute for the Best Basic Version of ADCP data collected in beam coordinates, with missing ensembles, should appear as follows:

Written to an EPIC standard data file by **adcp2ep.m** (version 1.1);  
Transformed to earth coordinates by **runbm2g.m**;  
Bins were trimmed by **trimBins.m** based on RDI surface output.;  
Ensembles recorded pre and post deployment were trimmed by **goodends.m**;  
The data were filtered using RDI quality control factors in **runmask.m**;  
The missing ensemble numbers that were filled with fill values using **fixEns.m** were:  
32; 467; 1029;  
Converted to NetCDF via MATLAB by **rdi2cdf.m** 3.0 10-Jan-2003

For data collected in earth coordinates with missing ensembles, the History attribute will appear as:

Written to an EPIC standard data file by **adcp2ep.m** (version 1.1);;  
Bins were trimmed by **trimBins.m** based on RDI surface output.;  
ensembles recorded pre and post deployment were trimmed by **goodends.m**;  
The missing ensemble numbers that were filled with fill values using **fixEns.m** were:  
32; 467; 1029;  
Converted to NetCDF via MATLAB by **rdi2cdf.m** 3.0 10-Jan-2003

### *Errors in trimBins.m*

If the history indicates that `rdi2cdf.m`, `runmask.m`, and `goodends.m` were completed for a data file recorded in beam coordinates, then an error must have occurred during `trimBins.m`. `TrimBins.m` is the only program that does not create a new data file. If `trimBins` is not completed then it is recommended that the user re-create the trim file (`*T.cdf`) by re-running `goodends.m` and then re-run `trimBins.m`. For data recorded in earth coordinates this is equivalent to re-running `runadcp.m` with an existing NetCDF file.

### *Unsuccessful Ensemble trimming*

Occasionally, bad ensembles may remain at the ends of the data record. This is a common problem for data recorded in earth coordinates. The **nctrim** utility from the NetCDF toolbox can be easily used to remove ensembles from the ends of the record. Nctrim can be used at any stage of processing. The command line statement is `nctrim('theSrc','theDst',theIndices,'theRecDim')`, where theSrc is the NetCDF file that needs to be trimmed and the resulting file will be theDst. The indices can be given by a range such as 100:3000, which removes everything outside of that range. However, don't confuse the ensemble (record) number, also indicated as the 'Rec' variable in the NetCDF file with the index number. When the data file is displayed in StarBeam the ensemble number is shown on the x-axis. The ensemble numbers start at 1 in the raw NetCDF file. However, as the data are processed bad ensembles will be removed either in the masking procedure or by the goodends program. Therefore, the ensemble number may not match the index number in the data file. It is the index numbers that must be given in nctrim.

## Section 7. Appendix

### File Inventory

The following is a list of the folders and functions that are included in the ADCP Toolbox zip file. The revision dates for the ADCP Toolbox specific programs are included in the right hand column for reference. It is suggested that these files be placed in the Matlabr12\work directory, and kept in the same layout as listed below.

<b>Matlab Path\Folder\Function</b>	<b>Revision Date</b>
D:\Matlabr12\work\ ADCPtools\ adcp2ep.m	30-Apr-2003
bm2dir.m	07-May-1998
bm2geo.m	13-Aug-2001
bm2xyze.m	12-May-1998
datmil.m	March 2000
fappend.m	08-Aug-2001
fillmsk.m	13-Oct-1999
fixEns.m	10-Jan-2003
goodends.m	05-Feb-2003
history.m	28-Dec-2000
mcnote.m	16-Apr-1999
mfriem.m	1999
rdfllead.m	11-Oct-1997
rdhead.m	07-Jan-1995
rdi2cdf.m	03-Apr-2003
rdsurface.m	06-Aug-2001
rdvlead.m	27-Feb-2003
runadcp.m	02-Apr-2003
runbm2g.m	28-Dec-2000
runmask.m	09-Jul-2001
trimbins.m	03-Apr-2003
D:\Matlabr12\work\ ADCPtools\ AddOns Gregorian.m	
Hms2h.m	
Hms2s.m	
Julian.m	
Polar2uv.m	
s2hms.m	
uv_rotate.m	
uv2polar.m	
which1s.m	

```
D:\Matlabr12\work\  
  ADCPtools\  
    AddOns\  
      _installers  
      Batch_install.p  
      Epic_install.p  
      Form_install.p  
      Presto_install.p  
      Proxy_install.p  
      Starbeam_install.p  
      Starbuck_install.p  
      Starbare_install.p
```

```
D:\Matlabr12\work\  
  ADCPtools\  
    Demo  
      Demo000.000
```

```
D:\Matlabr12\work\  
  ADCPtools\  
    extraMs  
      arrowsafe.m  
      create_batch  
      fig2jpeg.m  
      fig2tif.m  
      fixdate.m  
      fixval.m  
      plot_tilt.m  
      run_NOPP  
      scr_off.m  
      testrun  
      tilt2nc.m  
      trimfix.m  
      view_adcp.m
```

(\*The following 8 directories will be created when the p-code installers are run)

```
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                batch\  
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                epic\  
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                form\  
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                presto\  
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                proxy\  
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                starbeam\  
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                starbuck\  
D:\Matlabr12\work\  
    ADCPtools\  
        AddOns\  
            _installers\  
                starbare\  

```

## Startup file

Below is an example startup.m file for a DOS/Windows system that should be located in the Matlab/bin directory on the user's computer. If a startup.m file exists, these paths should be added. The path names may need to be altered if their system is a Mac, Unix, or non-standard PC. The startup file is not required. The same result may be achieved by adding the needed paths to the 'set path' in MATLAB.

---

```
function startup

ncstartup

path(path, 'd:\Matlabr12\toolbox\netcdf')
path(path, 'd:\Matlabr12\work\ADCPTools')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns')

path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\batch')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\epic')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\form')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\proxy')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\starbeam')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\starbare')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\presto')
path(path, 'd:\Matlabr12\work\ADCPTools\AddOns\starbuck')

path(path, 'd:\Matlabr12\work\ADCPTools\ExtraMs')
```

---

## Diary Log File for Demo file

The diary log seen in the following pages was recorded while running the ADCP data processing system for the Demo000.000 data that is provided with the toolbox. This log serves as an example of the type of information that the user will see on the screen while running the ADCP data processing programs. We recommend a diary file be created whenever the user is processing the data using the ADCP data processing system. This file is very helpful if the user encounters problems while running the system, or when trying to recall the parameters that were used to process the data.

---

runadcp

```
%function [theResult] = runadcp(rawdata,rawcdf,theMaskFile,theNewADCPFile,...  
    trimFile);
```

```
%Runs the important ADCP functions in the following order:
```

```
rdi2cdf.m
```

```
runmask.m
```

```
goodends.m
```

```
trimBins.m
```

```
%Occasionally the program will display some results in figures and  
stops to ask if it is ok to continue. If the program seems to be waiting  
and you are unsure what to do, check the matlab command window for  
a message!
```

```
%Type 'pause off' prior to running function if would like to run unattended.
```

```
%If you do not wish to run rdi2cdf.m, to create the first netcdf file  
from this data than, you must have an existing netcdf file and type  
in the first 2 inputs in the command line.
```

```
i.e. theResult = runADCP('999wh.000','999wh.cdf').
```

```
Although you will be prompted for the INPUT files, be sure to have all  
%of them accessible before beginning.
```

```
%INPUTS:
```

```
    rawdata = RDI ADCP binary output file (*.000)
```

```
    rawcdf = the Netcdf file created directly from the binary ADCP file
```

```
    theMaskFile = identical in size and structure to the cdf file
```

```
                containing 0's if the data is good and 1's if it is bad
```

```
    theNewADCPFile = the "masked" data file after initial filtering
```

```
    trimFile = the last file created by trimming the bad ensembles
```

```
If file names are not given, they will be requested.
```

```
%
```

%OUTPUTS:

%           theResult = is the trimFile, the last file created

rawdata =

D:\Data\Demo\Demo000.000

Converting RDI data file to netcdf

You will be asked for some inputs from the mooring log

rdi2cdf.m version 2.2c 10/22/99

Header information from adcp file D:\Data\Demo\Demo000.000

Number of bytes per ensemble 537

Number of data types 6

nbytes =

537

nt =

6

offsets =

18

70

127

289

371

453

CPU Version 8.30

Hardware Configuration for LSB 202

System Frequency = 300 kHz

Convex Beam

Sensor Configuration #1

Transducer head attached

Upward facing beam orientation

Hardware Configuration MSB 65

Beam angle = 20 degrees

4-beam janus configuration

The data is real

Number of beams used to calculate velocity data: 4  
Number of depth cells 20  
Pings per ensemble 300  
Depth cell size 100 cm  
Blank after xmit distance 176 cm  
Profiling mode 1  
Valid range for correlation 64  
Code repetitions 3  
Minimum and maximum percent good for output 0  
Minimum and Maximum Error Velocity values permitted 2000 mm/s  
Time between ping groups   
Coordinate Transformation = 7  
    Data stored coordinates = Beam  
Heading alignment 0  
Heading Bias: 0 deg  
Sensor Source = 127  
    Sound speed computed from ED, ES & ET  
    ED taken from depth sensor  
    EH taken from xducer heading sensor  
    EP taken from xducer pitch sensor  
    ER taken from xducer roll sensor  
    ES derived from conductivity sensor  
    ET taken from temperature sensor  
Sensor Availability = 29  
    heading sensor  
    pitch sensor  
    roll sensor  
    temperature sensor  
Distance to center of bin 1: 326 cm  
Transmit pulse length 148  
Starting water layer 1  
Ending water layer 5  
False target reject range values 50  
Transmit lag distance 51

-----

infilesize =

276692

514 ensembles estimated from file size  
99 ensembles read, 99 converted in 5.873400e+001 sec  
199 ensembles read, 199 converted in 6.210900e+001 sec  
299 ensembles read, 299 converted in 6.548400e+001 sec  
399 ensembles read, 399 converted in 6.883900e+001 sec  
Variable Leader ID not found

velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
499 ensembles read, 499 converted in 7.219300e+001 sec  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found

velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
correlation data ID not found  
intensity data ID not found  
percent good data ID not found  
Variable Leader ID not found  
velocity data ID not found  
Conversion complete  
7.225300e+001 seconds elapsed while processing  
498 was the number of the last ensemble read  
In the following figure quickly review the data  
Then click "Done" on the Starbeam menu, and hit enter

theImageDataLimits =

-151.0000 49.8000  
-78.5000 128.3000  
-69.6000 87.5000  
-28.4000 46.2000

theImageColorLimits =

```
-151.0000 151.0000
-151.0000 151.0000
-87.5000 87.5000
-87.5000 87.5000
```

Warning: Future versions will return empty for empty == scalar comparisons.

In d:\Matlabr11\work\ADCP\_tools\runadcp.m at line 134

fixens is currently running

Yippie...there are no missing ensembles!! :) :) :) :) :) :)

```
## closed: ncid = 3.
```

Running mask functions to remove bad data points

runmask is currently running

```
## Data File: D:\Data\Demo\Demo000.cdf
```

```
## Mask File: D:\Data\Demo\Demo000.msk
```

```
## Defining dimensions ...
```

```
## ensemble ...
```

```
## bin ...
```

```
## Defining variables ...
```

```
## D ...
```

```
## TIM ...
```

```
## Rec ...
```

```
## sv ...
```

```
## vel1 ...
```

```
## vel2 ...
```

```
## vel3 ...
```

```
## vel4 ...
```

```
## cor1 ...
```

```
## cor2 ...
```

```
## cor3 ...
```

```
## cor4 ...
```

```
## AGC1 ...
```

```
## AGC2 ...
```

```
## AGC3 ...
```

```
## AGC4 ...
```

```
## PGd1 ...
```

```
## PGd2 ...
```

```
## PGd3 ...
```

```
## PGd4 ...
```

```
## Hdg ...
```

```
## Ptch ...
```

```
## Roll ...
```

```
## Tx ...
```

```
## xmitc ...
```

```
## xmitv ...
```

```
## dac ...
```

```

## VDD3 ...
## VDD1 ...
## VDC ...
## Filling record-variables ...
## TIM ...
## Rec ...
## sv ...
## vel1 ...
## vel2 ...
## vel3 ...
## vel4 ...
## cor1 ...
## cor2 ...
## cor3 ...
## cor4 ...
## AGC1 ...
## AGC2 ...
## AGC3 ...
## AGC4 ...
## PGd1 ...
## PGd2 ...
## PGd3 ...
## PGd4 ...
## Hdg ...
## Ptch ...
## Roll ...
## Tx ...
## xmitc ...
## xmitv ...
## dac ...
## VDD3 ...
## VDD1 ...
## VDC ...
## Done.
Created mask file,
D:\Data\Demo\Demo000.msk
Pulling out the needed Global attributes from the Data file
This information sets the criteria for masking

vel =

    -1510    1332

## vel1 ...
## vel2 ...
## vel3 ...

```

```
## vel4 ...
## cor1 ...
## cor2 ...
## cor3 ...
## cor4 ...
## AGC1 ...
## AGC2 ...
## AGC3 ...
## AGC4 ...
## PGd1 ...
## PGd2 ...
## PGd3 ...
## PGd4 ...
```

The mask is filled

```
## closed: ncid = 4.
```

```
## closed: ncid = 3.
```

In the following figure bad velocity data are displayed in black

Mark any additional bad data points

Then click "Done" on the Starbare menu, and hit enter

```
theImageDataLimits =
```

```
-154.1873  49.8000
-81.7825  128.3000
-72.0937  87.5000
-29.5841  46.2000
```

```
theImageColorLimits =
```

```
-154.1873  154.1873
-154.1873  154.1873
-87.5000   87.5000
-87.5000   87.5000
```

```
theImageDataLimits =
```

```
-34.5127  23.9000
-20.7016  37.0000
-34.2349  12.8000
-43.3032  46.5000
```

```
theImageColorLimits =
```

-37.0000 37.0000  
-37.0000 37.0000  
-46.5000 46.5000  
-46.5000 46.5000

theImageDataLimits =

-20.5048 24.6000  
-22.5286 36.9000  
-33.8143 11.9000  
-28.4349 69.8000

theImageColorLimits =

-36.9000 36.9000  
-36.9000 36.9000  
-69.8000 69.8000  
-69.8000 69.8000

1 file(s) copied.  
## Masking: D  
## Remaining: 20 records...  
## Masking: TIM  
## Remaining: 498 records...  
## Masking: Rec  
## Remaining: 498 records...  
## Masking: sv  
## Remaining: 498 records...  
## Masking: vel1  
## Remaining: 498 records...  
## Masking: vel2  
## Remaining: 498 records...  
## Masking: vel3  
## Remaining: 498 records...  
## Masking: vel4  
## Remaining: 498 records...  
## Masking: cor1  
## Remaining: 498 records...  
## Masking: cor2  
## Remaining: 498 records...  
## Masking: cor3  
## Remaining: 498 records...  
## Masking: cor4  
## Remaining: 498 records...

```
## Masking: AGC1
## Remaining: 498 records...
## Masking: AGC2
## Remaining: 498 records...
## Masking: AGC3
## Remaining: 498 records...
## Masking: AGC4
## Remaining: 498 records...
## Masking: PGd1
## Remaining: 498 records...
## Masking: PGd2
## Remaining: 498 records...
## Masking: PGd3
## Remaining: 498 records...
## Masking: PGd4
## Remaining: 498 records...
## Masking: Hdg
## Remaining: 498 records...
## Masking: Ptch
## Remaining: 498 records...
## Masking: Roll
## Remaining: 498 records...
## Masking: Tx
## Remaining: 498 records...
## Masking: xmitc
## Remaining: 498 records...
## Masking: xmitv
## Remaining: 498 records...
## Masking: dac
## Remaining: 498 records...
## Masking: VDD3
## Remaining: 498 records...
## Masking: VDD1
## Remaining: 498 records...
## Masking: VDC
## Remaining: 498 records...
Masking is complete based on the read mask file
## ncitem/feval failure while trying: uminus(...)
## closed: ncid = 3.
goodends is currently running
```

ddate =

13-aug-1999

rdate =

17-aug-1999

checking vel1 ...

checking vel2 ...

checking vel3 ...

checking vel4 ...

Based on the # of good bins, 22 is the first good ensemble  
and 382 is the last good, out of 498 total ensembles  
from the file D:\Data\Demo\Demo000.msk

Based on pitch and roll, 22 is the first good ensemble  
and 360 is the last good, out of 498 total ensembles  
in file D:\Data\Demo\DemoM.cdf

minT =

13-Aug-1999

maxT =

17-Aug-1999

The last good ensemble is same as recovery date, 17-Aug-1999

The first good ensemble occurs at 13-Aug-1999  
and the recorded deployment date is 13-aug-1999  
Trimming file from ensemble 22 to 360

src =

NetCDF\_File: 'D:\Data\Demo\DemoM.cdf'  
nDimensions: 2  
nVariables: 30  
nGlobalAttributes: 43  
RecordDimension: 'ensemble'  
nRecords: 498  
Permission: 'nowrite'  
DefineMode: 'data'  
FillMode: 'fill'  
MaxNameLen: 0

dst =

```
NetCDF_File: 'D:\Data\Demo\DemoT.cdf'  
nDimensions: 0  
nVariables: 0  
nGlobalAttributes: 0  
RecordDimension: "  
  nRecords: 0  
Permission: 'clobber'  
DefineMode: 'define'  
FillMode: 'fill'  
MaxNameLen: 0
```

theRecdim =

```
NetCDF_Dimension: 'ensemble'  
  itsLength: 498
```

```
## nctrim: copying "D"  
## nctrim: trimming "TIM"  
## nctrim: trimming "Rec"  
## nctrim: trimming "sv"  
## nctrim: trimming "vel1"  
## nctrim: trimming "vel2"  
## nctrim: trimming "vel3"  
## nctrim: trimming "vel4"  
## nctrim: trimming "cor1"  
## nctrim: trimming "cor2"  
## nctrim: trimming "cor3"  
## nctrim: trimming "cor4"  
## nctrim: trimming "AGC1"  
## nctrim: trimming "AGC2"  
## nctrim: trimming "AGC3"  
## nctrim: trimming "AGC4"  
## nctrim: trimming "PGd1"  
## nctrim: trimming "PGd2"  
## nctrim: trimming "PGd3"  
## nctrim: trimming "PGd4"  
## nctrim: trimming "Hdg"  
## nctrim: trimming "Ptch"  
## nctrim: trimming "Roll"  
## nctrim: trimming "Tx"  
## nctrim: trimming "xmitc"  
## nctrim: trimming "xmitv"  
## nctrim: trimming "dac"
```

```
## nctrim: trimming "VDD3"  
## nctrim: trimming "VDD1"  
## nctrim: trimming "VDC"  
## ncitem/feval failure while trying: uminus(...)  
## closed: ncid = 3.
```

trimbins is currently running

cpfile =

d:\applications\RDI\datatools\Demo000.000

1 file(s) copied.

Running RDI surface program

1 file(s) moved.

RDI surface program found the following depth constraints in meters: min = 17.35 max = 18.62 Mean = 18.1875

Current plot held

Current plot held

User modified depth constraints in meters: min = 17.8 max = 18.62 Mean = 18.1984

ADCP measured 19.4484 m from surface to the seabed (mean sea level)

The tidal range is approximately 0.17305 m

Finding the good bins

Redefining the "Bin" dimension

May take a few minutes...

# in Dsurf 339

# in ensemble 339

f =

```
NetCDF_File: 'D:\Data\Demo\DemoT.cdf'  
nDimensions: 2  
nVariables: 31  
nGlobalAttributes: 43  
RecordDimension: 'ensemble'  
nRecords: 339  
Permission: 'write'  
DefineMode: 'data'  
FillMode: 'fill'  
MaxNameLen: 0
```

File D:\Data\Demo\DemoT.cdf has been modified

## 4 bins were removed from the top of the water column

```
## ncitem/feval failure while trying: uminus(...)
```

```
theImageDataLimits =
```

```
-18.2000 13.7000  
-17.9000 17.4000  
-27.0000 14.8000  
-1.5000 23.5000
```

```
theImageColorLimits =
```

```
-18.2000 18.2000  
-18.2000 18.2000  
-27.0000 27.0000  
-27.0000 27.0000
```

```
ans =
```

```
D:\Data\Demo\DemoT.cdf
```

```
theImageDataLimits =
```

```
-10.3000 14.0000  
-14.9000 18.4000  
-24.3000 8.1000  
-9.1000 15.6000
```

```
theImageColorLimits =
```

```
-18.4000 18.4000  
-18.4000 18.4000  
-24.3000 24.3000  
-24.3000 24.3000
```

```
theImageDataLimits =
```

```
-13.5000 10.3000  
-16.2000 17.0000  
-23.4000 10.2000  
-3.6000 16.2000
```

theImageColorLimits =

```
-17.0000 17.0000  
-17.0000 17.0000  
-23.4000 23.4000  
-23.4000 23.4000
```

theImageDataLimits =

```
-14.4000 8.7000  
-3.7000 15.8000  
-16.4000 9.5000  
-3.5000 16.4000
```

theImageColorLimits =

```
-15.8000 15.8000  
-15.8000 15.8000  
-16.4000 16.4000  
-16.4000 16.4000
```

adcp2ep

adcp2ep is currently running

function epDataFile = adcp2ep(adcpFile, epDataFile, ADCPtype, dlgFile)

This function is used to translate RDI ADCP data into variables that are in terms of earth coordinates and create an epic compatible data file.

If the data is in Beam coordinates it will be transformed by runbm2g.m into Earth coordinates. This transformation can be run on workhorse and broad band data, but the ADCP type must be specified.

Magnetic Declination

If a magnetic declination was provided to the instrument prior to deployment or to rdi2cdf in post-processing it will be applied at this time for both Earth and Beam coordinate data

Inputs:

adcpFile = the ADcp data file in beam coordinates

(Note:if running routines in sequence it should be the trimFile.)

epDataFile = the new Epic compatible file that will be created

ADCPtype = WH or BB; will default to WH if not specified

WH = workhorse, BB = broad band

note: if BB, do not need a dlgFile  
dlgFile = the dialog file that was created when the ADCP was "deployed"

Note: If the names of the files are not given, they will be requested.

Output:

epDataFile = same as input

TIM was corrected by 150 sec = half the ensemble time

Converting "TIM" to "time" and "time2"

The file has 339 ensembles and 16 bins

Averaging echo intensity

Averaging percent good

Data in Beam coordinates is being transformed to Earth

runbm2g is currently running

ADCPtype =

WH

Obtaining Beam configuration information from D:\Data\Demo\999wh.dlg

Beam Width: 3.7 degrees

B =

NetCDF\_File: 'D:\Data\Demo\DemoT.cdf'  
nDimensions: 2  
nVariables: 31  
nGlobalAttributes: 43  
RecordDimension: 'ensemble'  
nRecords: 339  
Permission: 'nowrite'  
DefineMode: 'data'  
FillMode: 'fill'  
MaxNameLen: 0

100 ensembles converted in 1.275800e+001 sec

200 ensembles converted in 2.486600e+001 sec

300 ensembles converted in 3.709300e+001 sec

## ncitem/feval failure while trying: uminus(...)

Modifying the global attributes

Data\_Collection\_Information =

```
Mooring: '9991'  
Experiment: {{7x1 cell} [1]}  
Project: {{5x1 cell} [1]}  
descript: {{8x1 cell} [1]}  
comments: 'test deployment '  
longitude: [1x1 struct]  
latitude: [1x1 struct]
```

Defining variables and their attributes

```
...Defining "time"  
...Defining "time2"  
...Defining "depth"  
...Defining "lon"  
...Defining "lat"  
...Defining "u_1205"  
...Defining "v_1206"  
...Defining "w_1204"  
...Defining "Werr_1201"  
...Defining "AGC_1202"  
...Defining "PGd_1203"  
...Defining "hght_18"  
...Defining "Tx_1211"
```

Copying data to D:\Data\Demo\Demo.nc

```
... time  
... time2  
... lat  
... lon  
... depth  
... u_1205  
... v_1206  
... w_1204  
... Werr_1201  
... AGC_1202  
... PGd_1203  
...hght_18  
... Tx_1211  
## ncitem/feval failure while trying: uminus(...)
```

ans =

D:\Data\Demo\Demo.nc

diary off

---