RDI Tools
User's Guide

P/N 957-6157-00 (December 2001)

RD Instruments
Acoustic Doppler Solutions

Table of Contents

List of Figures

List of Tables

RD Instruments
Acoustic Doppler Solutions

RDI Tools User's Guide

1 Introduction

This guide is an overview on using the *RDI Tools* utility software provided with your system. Use *BBTalk* to communicate with the ADCP.

NOTE. For previous users of the RDI Tools programs, *BBTalk* replaces the *DumbTerm* program.

1.1 System Requirements

RDI Tools requires the following:

- Windows 95®, Windows 98®, or Windows NT 4.0® with Service Pack 4 installed, Windows 2000®
- Pentium class PC 233 MHz (350 MHz or higher recommended)
- 32 megabytes of RAM (64 MB RAM recommended)
- 6 MB Free Disk Space plus space for data files (A large, fast hard disk is recommended)
- One Serial Port (two or more High Speed UART Serial Port recommended)
- Minimum display resolution of 800 x 600, 256 color (1024 x 768 recommended)
- CD-ROM Drive
- Mouse or other pointing device

1.2 Software Installation

To install the RDI Tools software, do the following.

a. Insert the compact disc into your CD-ROM drive and then follow the browser instructions on your screen. If the browser does not appear, complete Steps "b" through "d."

b. Click the **Start** button, and then click **Run**.

c. Type **<drive>:launch**. For example, if your CD-ROM drive is drive D, type **d:launch**.

d. Follow the browser instructions on your screen.

Once, installed, you will have several shortcuts added to your Windows® **Start** menu. The **BBTalk** shortcut will start *BBTalk*. **BBLIST** will start the DOS *BBLIST* program.

NOTE. The default directory for BBTalk installation is C:\program files\RD Instruments\RDI Tools.

2 Using BBTalk

BBTalk is a dumb terminal emulator program. This Windows compatible program can capture raw data files and help troubleshoot configuration problems. You can use *BBTalk* for serial or parallel communications in either an ASCII or BINARY mode. A binary-to-hexadecimal conversion feature lets you view and record the binary output data in a hexadecimal format. A LOG feature lets you record data to a disk file.

2.1 BBTalk Short-Cut Keys

Use the following keys to quickly start tasks.

Table 1: BBTalk Short-Cut Keys

Shortcut	Description
End	Send wakeup to ADCP
F1	Help menu
F2	Run script file
F3	LOG data
F4	ASCII or Hex display
F5	Communications Properties
F8	Connect
F9	Disconnect
Ctrl+Page Down	Recover recorder data

2.2 Communication Parameters

Before you can establish communications with the ADCP, you must configure *BBTalk*.

a. At the **Connect To** screen, select the ADCP type **(WorkHorse**, **Broad-Band**, or **NarrowBand)** from the list. Select the COM port the ADCP is connected to. Click **Next**.

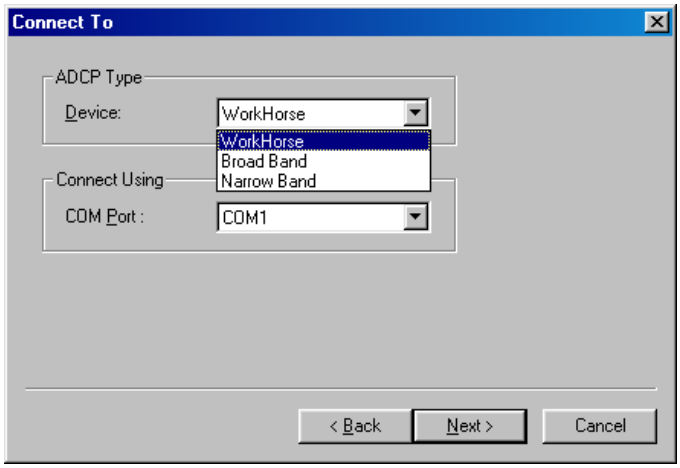


Figure 1. BBTalk Connect To Screen

b. On the **Port Settings** screen, select the baud rate, parity, stop bits, and flow control. Click **Next**.

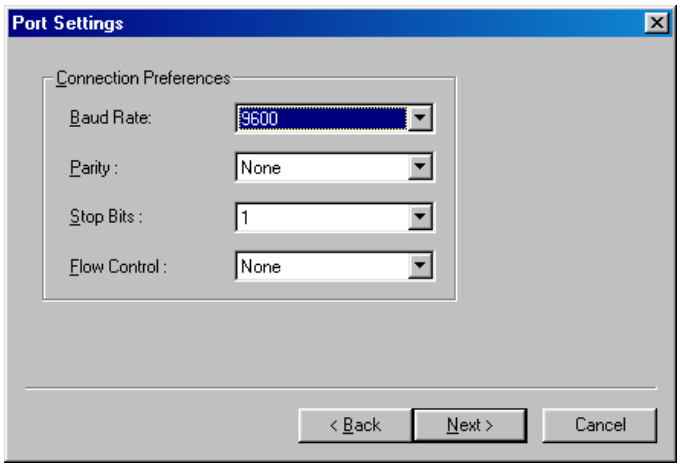


Figure 2. BBTalk Port Settings Screen

NOTE. To return to this screen while running *BBTalk*, click **File**, **Properties**.

c. On the **Options** screen, select the desired settings.
   - **Send Break On New Connection** – As soon as *BBTalk* starts, it will send the ADCP a Break.
   - **Connect to Last Open Port On startup** – *BBTalk* will immediately start without going through the configuration screens.
   - **Overwrite Log Files When Opening** – Log files with the same name will be overwritten.
   - **Error Checking for Script Files** – Check the Script file for errors before running.
   - **Send CK On Baud Rate Change (CB Command) –** Automatically sends a CK command to WorkHorse and BroadBand ADCPs to save the baud rate after the baud rate has been changed using the CB command.

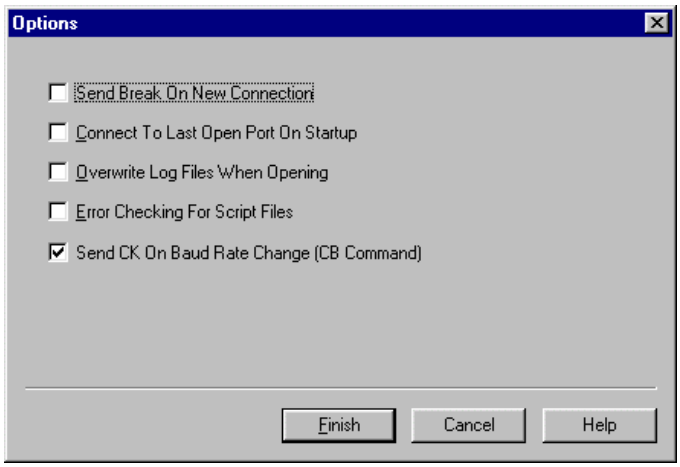


Figure 3. BBTalk Options Screen

NOTE. To return to this screen while running *BBTalk*, click **Tools**, **Options**.

d. Click **Finish**.

e. On the **File** menu, click **Break** (you can also press the **End** key to send a break or use the Toolbar and press the **B** button). You should see the wakeup message appear on the log file window.

```
xxxxxx ADCP
RD INSTRUMENTS (c) 1997-2000
ALL RIGHTS RESERVED
Firmware Version xx.xx
>
```

2.3 Sending Commands to the ADCP

a. Setup the communication parameters between *BBTalk* and the ADCP.

b. Wake-up the ADCP by pressing **End**.

c. At the ">" prompt in the communication window, enter the direct command you wish to send to the ADCP and then press the **Enter** key. Refer to the Command and Output Data Format Guide for a listing of all direct commands and their format.

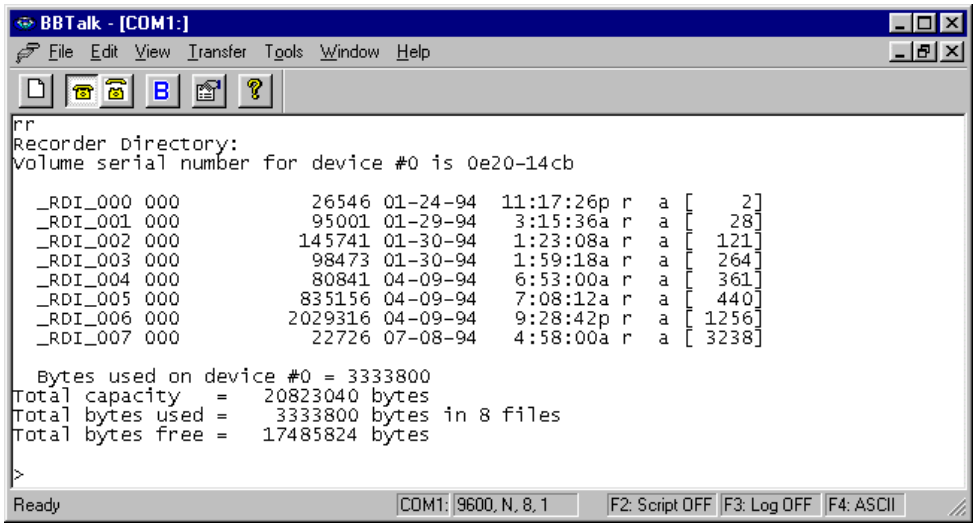


Figure 4. Sending a Command to the ADCP Using BBTalk

You may also send commands or verify if a command was sent using the **Command History** function.

a. On the **Transfer** menu, click **Command History**.

b. Use the scroll bar to view all of the commands sent to the ADCP.

c. To *resend a command*, select a command in the list and click **OK**.

d. To *send a new command*, type the command in the window and click **OK**.

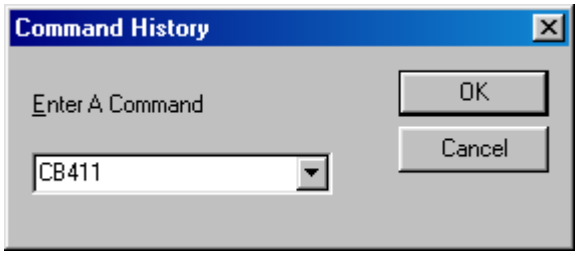


Figure 5. BBTalk Command History Screen

2.4 Recovering Data from the ADCP's Recorder

a. On the **File** menu, click **Recover Recorder**.

b. Enter the directory where the files will be downloaded. Click **OK**.

c. Select the file to recover, or check the **Select All Files** box to recover all of the files. Click **OK**.

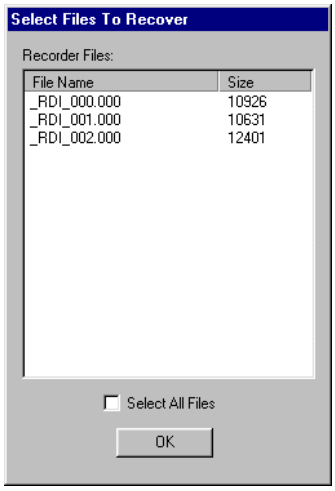


Figure 6. Select Files to Recover Screen

d. *BBTalk* displays current protocol status, filename being received, total size of receiving file and current number of bytes received.

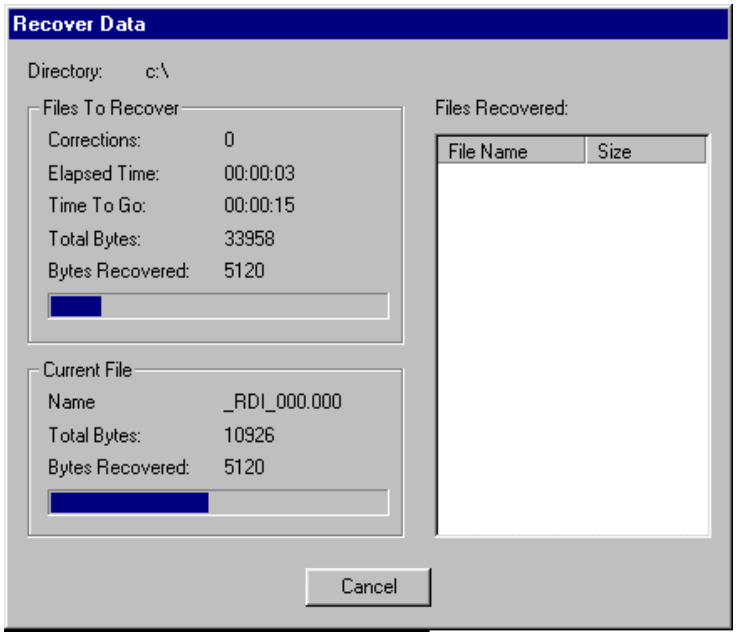


Figure 7. Recovering Data from the ADCP's Recorder

2.5 BBTalk Script Files

You can control what *BBTalk* sends and captures by writing *BBTalk* script files. Script files are simply ASCII files produced by ASCII editors such as MS-DOS EDIT or NotePad (see "Example Script File," page 9). In general, they contain ASCII characters that are sent out through the serial port. In addition, *BBTalk* script files can contain embedded commands that control the behavior of *BBTalk* as it interprets the script file.

- All lines in the script file except those beginning with a '$' or a ';' character are sent out through the serial port delimited with a Carriage Return <CR> (A line feed character <LF> = ASCII 10 decimal is NOT sent).
- Lines containing a semi-colon ';' = ASCII 59. If the first character of a line containing a semi-colon is not a '$' character, then all characters preceding the semi-colon are sent followed by a <CR>. All characters following the semi-colon (including the semi-colon) are ignored. This feature is to provide file comments that the user may insert for script file clarity.
- Lines beginning with a dollar sign '$' = ASCII 36 decimal are script file control commands (see Table 2, page 8).

2.5.1 Running Script Files

To run a script file, press **<F2>**. Select the script file to run from the scroll-down list. If no extension is given for the script file, an extension of *.txt is assumed.

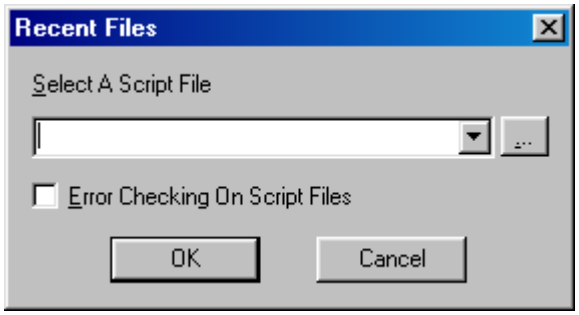


Figure 8. Selecting a Script File to Run

NOTE. When a script file is running, the status bar will change to **F2: Script On**.

2.5.2 Writing Script Files

To write your own script file, do the following.

a. Create an ASCII text file. Use a text editor such as MS-DOS EDIT or NotePad. Use the *.rds extension when saving the file.

NOTE. Double-clicking a *.rds file will automatically start BBTalk and run the script file.

b. To send a command to the ADCP, use one command per line. *BBTalk* will automatically add a carriage return after the command is sent.

c. To add comments to your script file, add a semicolon to the beginning of the line. *BBTalk* will ignore all comments.

d. To use a *BBTalk* script file command (see Table 2), add one command per line.

2.5.3 Script Commands

The following are legal script commands:

Table 2: BBTalk Script Commands

Command	Description
$B	Sends a <BREAK>
$COM:bbbbb,p,d,s	Sets up BBTalk communication parameters: bbbbb = baud rate (1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200) p = parity (N = none, E = even, O = odd) d = number of data bits (7 or 8) s = number of stop bits (1 or 2)
$Dnnnn	Delays nnnn seconds
$H	Toggle HEX/ASCII
$Lfilename.ext	Toggle logging. If the file name is missing and logging is to be turned on, then a window pops up asking for the log file name. Type in the file name and press the enter key to continue. If the file already exists, data will be overwritten.
$lfilename.ext	Same as above, except a lower case "L" is used. If the file already exists, data will be appended.
$Ptext to screen	Prints “[text to screen]” on the screen and also in log file if logging (useful for comments)
$N	Allows you to send Narrowband binary command syntax.
$R	Repeats script file (remainder of script file is ignored
$Wxx,nnn	Wait for character xx (binary representation). If nnn is missing, it waits forever. $W62 means wait for the ">" prompt forever.
$W"c",nnn	The same as above, but "c" is represented as a character. $W">",2 means wait for the prompt for a maximum of two seconds and than continue.
$X	Exits BBTalk

2.5.4 Example Script File

The following example is a printout of the script file TestWH.rds (see "Using a Script File to Test the ADCP," page 11).

```
;---------------------------------------------------------------------------
;                                TESTWH.RDS
;               Script file for testing RD Instruments WorkHorse
;                        ADCP with the BBTalk program
;         Copyright (C) 2001 by RD Instruments - All rights reserved
;       Licensed for exclusive use with RD Instruments products or data
;---------------------------------------------------------------------------
;---------------------------------------------------------------------------

; RDI - WH ADCP testing script file:
; FILE name = "TESTWH.RDS"
; 1 November 2001
; Modified by RDI on:
; none
$LWH_TESTS.TXT
$P ===========================================================================
; Print the following lines.
$P WH ADCP Test
$P *************
$P
$P The following tests are basic tests which will confirm that your system

$P is ready for use.  Some tests will need to be run with the system in
$P water.  You will be prompted when this is necessary.
$P
$P Connect the WH ADCP to power and the PC as described in the manual.
$P Turn on power to the WH ADCP.
$P
$P The results of all tests will be printed to the screen and saved to the
$P log file WH_TESTS.TXT.   A file called WH_TESTS.TXT with the results of
$P this test will be created in the same directory as the BBTalk program
$P is running from.
$P
$P             The following tests will be performed:
$P
$P              PA    Basic Internal System Tests
$P              PC2   Sensor Verification Test
$P              PC1   Beam Continuity Test
$P
$P Program is delaying 20 seconds before continuing.
$P Press <Enter> to continue.
$W62,20
$P ===========================================================================
$P
$P PA -- Basic Internal System Tests
$P The following tests will verify that the internal electronics are
$P performing correctly.  These tests are best run when the transducer
$P face is submerged in water.  A bucket of water deep enough to cover
$P the transducer beams is all that is needed.  If done in air some tests
$P may fail.
$P
$P Program is delaying 10 seconds before continuing.
$P Press <Enter> to continue.
$W62,20
$P ===========================================================================
$P
$P Sending a break to Wake Up the System
$B
$D2
$P ===========================================================================
$P
$P Restoring factory defaults into temporary memory for TEST.
$P
CR1
$D1
$P ===========================================================================
$P
$P Collecting system specific data.
TS?
$D1
PS0
```

```
$D1
PS3
$D1
$P ========================================================================
$P
$P Starting the Automated Tests.
PA
$W62,100
$P
$P All of the above tests should have passed. Review the file WH_TESTS.TXT
$P to verify your tests results.  Remember that some tests will fail
$P unless the transducer is immersed in water.  Consult your Technical
$P Manual for trouble shooting tips if this test did not pass.
$P
$P
$P Program is delaying 15 seconds before continuing.
$P Press <Enter> to continue.
$W62,15
$P ========================================================================
$P
$P PC2 -- Sensor Verification Test
$P The following test will confirm that your heading, pitch, roll,
$P temperature, orientation sensor, and pressure sensor (if installed) are
$P operating.   You should turn and tilt the ADCP and confirm that changes
$P occur in the heading, pitch, and roll.  Verify that the Up/Down setting
$P agrees with the direction of your ADCP transducer.  Verify that the
$P ambient temperature and pressure are reasonable values.
$P
$P This test in itself does not calibrate or confirm the accuracy of the
$P sensors.  However, if you turn and till the ADCP while comparing the
$P output to a known reference then you can confirm the accuracy.
$P
$P The following test is best run when the transducer face is in air and
$P the transducer is pointing the direction you intend to deploy the
$P instrument.
$P
$P This test will continue to run until you stop the test.
$P
$P Program is delaying 20 seconds before continuing.
$P Press <Enter> to continue.
$W62,20
$P ========================================================================
$P
$P Sending a break to Wake Up the System
$B
$D2
$P ========================================================================
$P
PC2
$W62,1000
$P
$P The Sensor test is complete.  The heading, pitch, roll sensors should
$P have changed as you turned and tilted the system.  The Up/Down setting
$P should have agreed with the direction of your ADCP transducer. The
$P ambient temperature and pressure (if installed) should have been
$P reasonable values.
$P
$P All of the above tests should have passed. Review the file WH_TESTS.TXT
$P to verify your tests results.  Consult your Technical Manual for trouble
$P shooting tips if this test did not pass.
$P
$P Program is delaying 10 seconds before continuing.
$P Press <Enter> to continue.
$W62,10
$P
$P ========================================================================
$P
$P PC1 -- Beam Continuity Test
$P The following test will confirm that each of the beams on your
$P transducer is capable of receiving signals.  This test must be run in
$P air and free of external interference to pass.
$P
$P This test will require you to rub each of the beams on the transducer.
$P This is done with quick rubbing movements across each of the urethane
$P faces.
$P
$P Program is delaying 25 seconds before continuing.
$P Press <Enter> to continue.
```

```
$W62,25
$P ========================================================================
$P
$P Sending a break to Wake Up the System
$B
$D2
$P ========================================================================
$P
PC1
$W62,1000
$P
$P The Beam Continuity test is complete.  Each of the beams should have
$P passed.  Review the file WH_TESTS.TXT to verify your tests results.
$P Consult your Technical Manual for trouble-shooting tips if this test
$P did not pass.
$P
$P ========================================================================
$P All tests have been run and if passed your system is ready for
$P deployment.
$L
```

2.5.5 Using a Script File to Test the ADCP

a. Connect the ADCP to the computer as described in the appropriate ADCP User's Guide.

b. Setup communication parameters between *BBTalk* and the ADCP.

c. Click **File**, **Send a Break** to send the wakeup command (BREAK) to the ADCP

d. On the **File** menu, click **Send Script File**. Click the **Browse** button "...".

e. Select the appropriate script file (see Table 3).

Table 3: BBTalk Test Script Files

Script File Name	ADCP Type	Results Saved to
TestBB.rds	Broadband	BB_TESTS.txt
TestOS.rds	Ocean Surveyor	OS_ TESTS.txt
TestWH.rds	Workhorse	WH_ TESTS.txt

NOTE. These script files (*.rds) were copied into the same directory as *BBTalk* when you installed the RDI Tools software CD sent with your system.

Double-clicking a *.rds file will automatically start BBTalk and run the script file.

f. Follow the prompts on the screen (see Figure 9, page 12).

g. To review the test results, open the results log file (*.txt) with any text editor (i.e. NotePad).

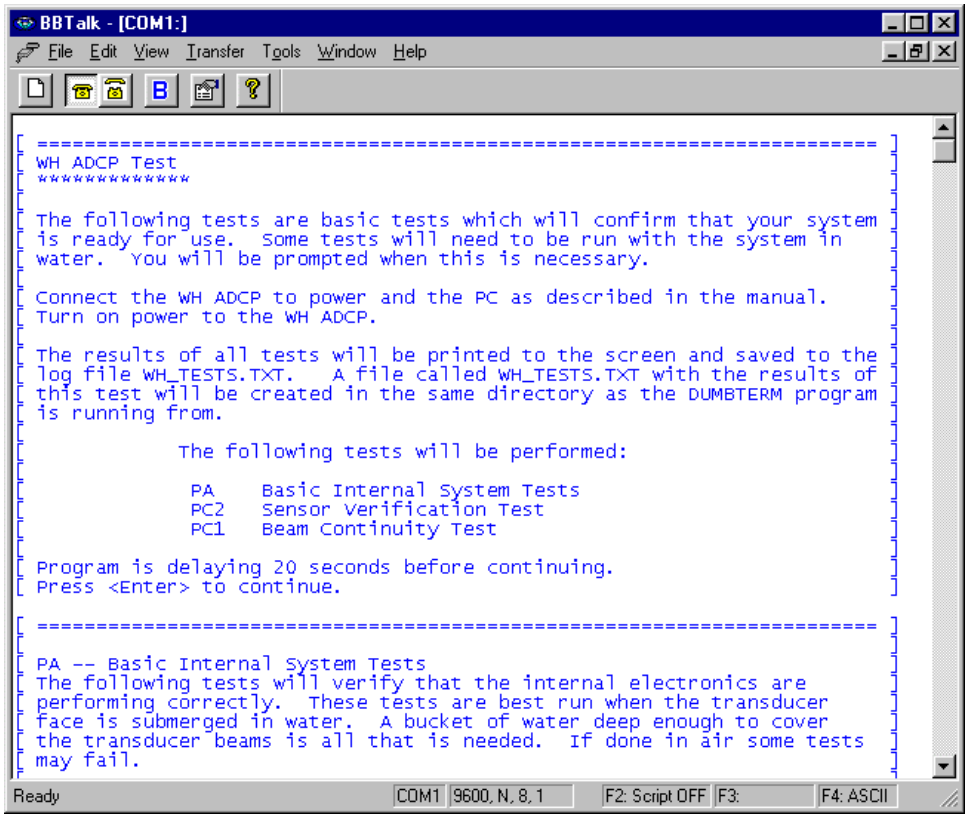


Figure 9. Running a Script File to Test the ADCP

2.6 BBTalk LOG Files

The LOG feature lets you record data to a disk file. You name the file by pressing the **<F3>** key. You can enable logging at any time. The status bar shows the Log status. To use Log, do the following steps.

a. Press **<F3>** to enable the LOG function.

b. Select the Log File from the scroll-down list.

c. If the file already exists, the program asks you if you want to overwrite the existing file unless the **OverWrite Log File** box is selected.

d. All data sent to the screen will now be written to the file you specified. You can enable the Log feature at any time, even if the ADCP is already sending data.

e. To disable Log, press **<F3>**.

3 Using BBLIST

BBLIST takes the binary data files created by the ADCP and lets you convert selected data fields to common units of measurement in an ASCII-text format. You can then use the ASCII files in programs that accept this format. *BBLIST never* alters your original ADCP binary data files. *BBLIST* lets you:

- Set processing parameters.
- Transform velocity data.
- Select the display/conversion limitations (e.g., data fields bin range, ensemble range, output file size).
- Select the ADCP data fields to display or convert.
- Monitor and control the conversion process.
- Look at radial beam data.

3.1 Starting BBLIST

To start *BBLIST*, type the following at the DOS prompt.

- **BBLIST** - Starts *BBLIST* and lets you select the ADCP binary data file from within the program.
- **BBLIST C:\BBDATA\TEST.000 /M** - Starts *BBLIST* and loads the TEST.000 file located on the C: drive in the BBDATA directory. The /M forces *BBLIST* to use monochrome screens.

When you run the program the first time, *BBLIST* displays its introduction and copyright screen. During operation, *BBLIST* creates/updates a pointer file named BBLIST.PTR. This file saves the current working directory, the name of the last binary file used, the name of the format file (*.FMT) last used, and the color selection. This may help save you time by automatically calling up the files you used last. If you need to see the introduction screen again, you must first delete or rename the BBLIST.PTR file.

Extensive help is available while using *BBLIST* by pressing **<F1>**. The help screen lists all of the menus and a description of each option available. You may also print the help screen file by printing the file BBLIST.DOC.

3.2 BBLIST Menus

BBLIST has four main menus (Figure 10) to guide you through the steps needed to convert a ADCP binary raw data file to an ASCII data file.

File Menu. Use this menu to save your menu settings and conversion layout selections to a configuration format file (*.FMT), to load a ADCP binary data file or format file, to display the file information screen, or to exit *BBLIST*.

Process Menu. Use this menu to set the processing parameters for *BBLIST*. You can set the velocity reference, magnetic variation (declination), velocity measurement units, and depth measurement units.

Display Menu. Use this menu to select the type of data to view. Data available for display include ADCP setup, sensors, reference layer, bottom-track, and profile.

Convert Menu. Use this menu to convert the binary ADCP data set into ASCII. Before starting the conversion process, you can define the conversion limitations and the output format.

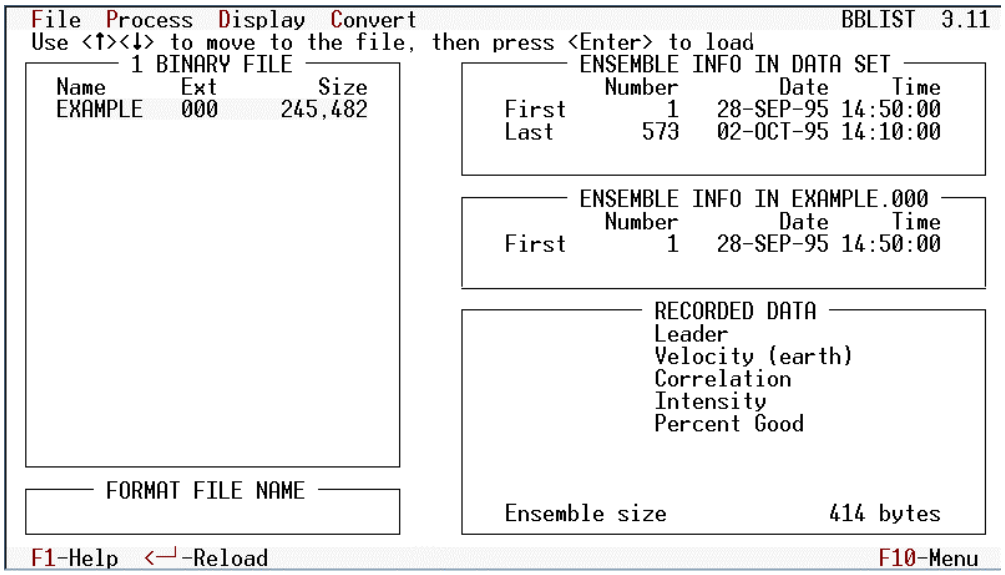


Figure 10. BBLIST Display

3.3 Using BBLIST to Convert Files

This is an example of how to convert the binary file EXAMPLE.000 to ASCII delimited text. For this example, we have chosen to extract only the magnitude, direction, and range data. Remember that *BBLIST* never changes your original data file — you can convert the raw data to as many different ASCII formats as desired.

a. Start BBLIST. To start *BBLIST*, type BBLIST at the DOS prompt.

b. Load EXAMPLE.000 binary file. Press **F3** to load the binary file.

c. Set the processing parameters. Use the **Process** menu (Figure 11, page 17) to set the velocity reference, magnetic variation, velocity measurement units, and depth measurement units. Use the **Space** key to toggle each setting. Make sure that **Mark Below Bottom** is set to **No**.

NOTE. BBLIST uses the direct ADCP commands for presenting depths (ED-command).

d. Set the conversion parameters. Use the **Convert**, **Limits** menu (Figure 12, page 17) to set the conversion limitations and parameters. You may want to increase the Max file size to 1000 kB if you plan to back up the ASCII data to floppy disks.

e. Define the format. Consider the following:

- What data do you want to extract? For this example, we choose *magnitude*, *direction*, and *range*.
- To what program will you export the ASCII data? Does the program support space, tab, or comma-delimited text? We choose space-delimited text for the example. To change, select Field delimiter and press the **Space** key to toggle between **Tab**, **Comma**, and **Space**.
- Do you want vertical or horizontal placed data? For this example, we choose a vertical data format. To change, select Bin layout and press the **Space** key to toggle between **Vertical** and **Horizontal**.

Begin defining the format by doing the following.

1. Select **Convert**, **Define Format**. When you first enter this screen, *BBLIST* displays only a **Format End** marker. To begin entering data fields, press **<End>** to display the **Define Format** selection menu (Figure 13, page 17). If you make a mistake, select delete block and re-enter your choice.
2. Select **Ensemble Info** and press **<end>**. Select **Number** and press **<End>**.
3. Move the cursor to the **Format End** marker and press **<End>**.

4. Select **Profiles**, **Mag and Dir**. Select **M** for magnitude and press **<End>**.
5. Move the cursor to **Line End** and press **<End>**.
6. Select **Profiles**, **Mag and Dir**. Select **D** for direction and press **<End>**.
7. Move the cursor to **Line End** and press **<End>**.
8. Select **Profiles**, **Depth Ref**. Select **R** for range and press **<End>**. Your display should look like Figure 14, page 18.

f. View the format. Press **F9** to view the format before converting (Figure 15, page 18). Use the + and - keys to increase the ensemble number. Press **F9** again to return to the define format screen.

g. Save the format. For future use, save the format to a *.FMT file by pressing **F2** and naming the file (example; MYFORMAT).

h. Begin the conversion. Select **Convert**, **Start Conversion**. Enter a file name for the ASCII file data set. *Be sure to use a file name that differs from any existing file name*. Do not use the name of the binary data set. If *BBLIST* detects that the file name already exists, you are given the option to overwrite the existing files.

3.4 Using a BBLIST Format File to Convert Files

Once you have created a format file, you can use this file to convert other data files.

a. Start *BBLIST*. To start *BBLIST*, type BBLIST.

b. Load EXAMPLE.000 binary file. Press **F3** to load the binary file.

c. Load Format file. Press **F3** to load the format file.

d. Begin the conversion. Select **Convert**, **Start Conversion**. Enter a file name for the ASCII file data set. *Be sure to use a file name that differs from any existing file name*. Do not use the name of the binary data set. If *BBLIST* detects that the file name already exists, you are given the option to overwrite the existing files.

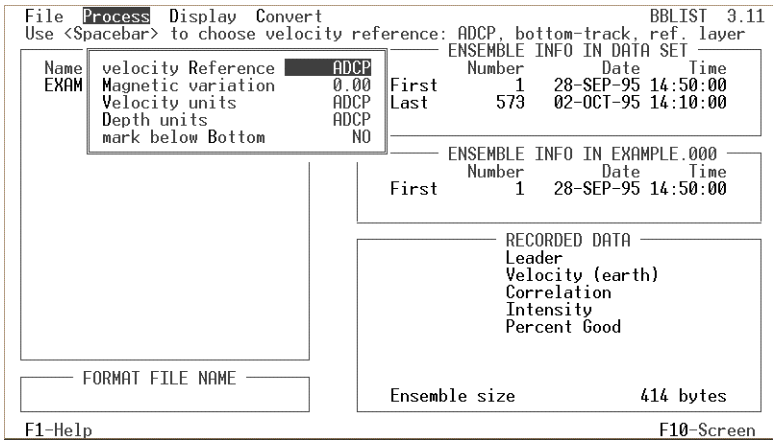


Figure 11. Set the Processing Parameters

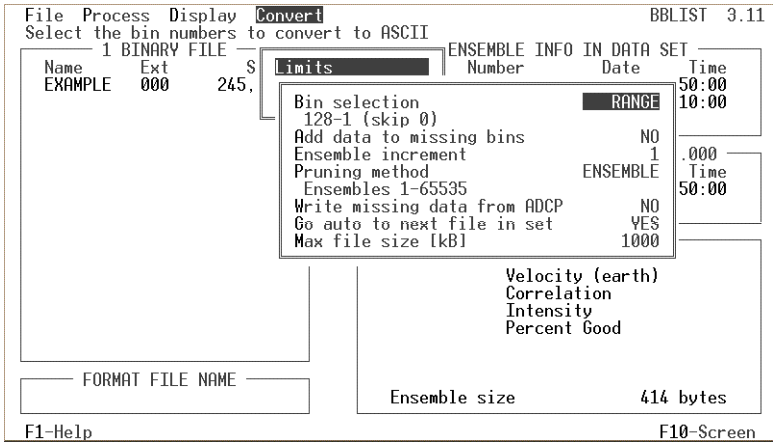


Figure 12. Set the Conversion Limitations and Parameters

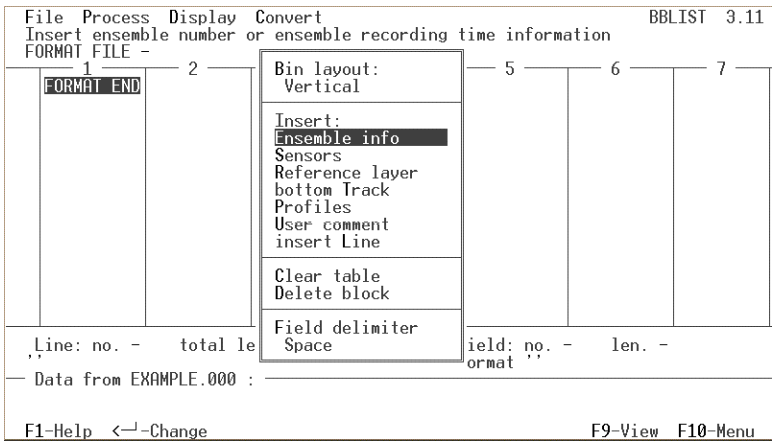


Figure 13. Define Format Selection Menu

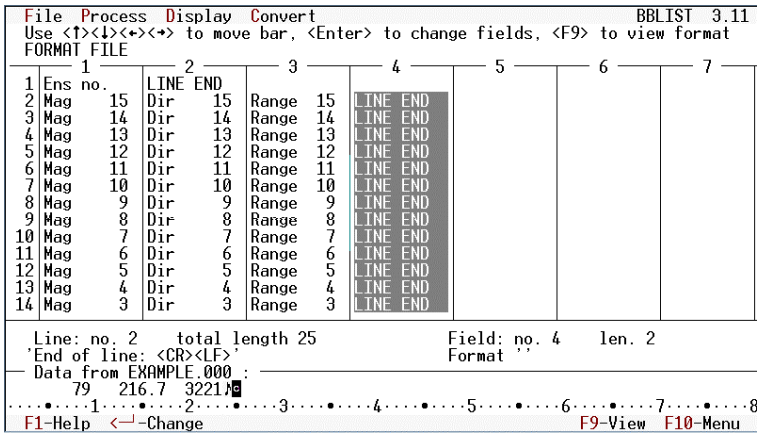


Figure 14. Defining the Format

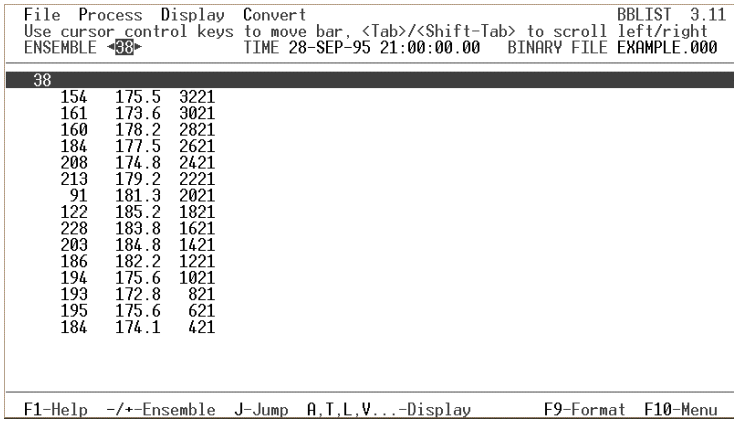


Figure 15. View the Format before Converting

3.5 Report File

When the conversion process is complete, *BBLIST* creates an ASCII report file (*.RPT). You can view this file with any text editor. This file contains the following information about the settings and data in the converted files.

- ***ADCP information*** - system frequency, beam angle, number of profiling beams, transducer orientation (up/down), transducer pattern (concave/convex), transducer connection (connected/disconnected), and CPU firmware version number.
- ***ADCP setup*** - number of bins, bin length, blank after transmit length, pings per ensemble, time per ping, and profiling mode.
- ***ASCII file data format*** - a description on the contents of each line in the converted data file (for one ensemble).
- ***Processing parameters*** - velocity units, velocity reference, depth units, bin sequence, magnetic variation.

Sample BBLIST report file:

```
REPORT FOR ASCII DATA CONVERSION
--------------------------------

1. ADCP INFORMATION:

   Frequency 300 kHz
   Beam angle 20 deg
   4 beam system
   Up-looking orientation
   Convex beam pattern
   Transducer head connected
   CPU firmware 8.01

2. ADCP SETUP:
   Number of bins 15
   Bin length 200 cm
   Blank after transmit 200 cm
   Distance to first bin 421 cm
   Transmit length 207 cm
   Pings per ensemble 300
   Time per ping 1.99 s
   Profiling mode 1

3. ASCII FILE DATA FORMAT:

   Line 1: Ensemble number
   Line 2-16: Magnitude, Direction, Bin range

4. PROCESSING PARAMETERS:

   Velocity units: ADCP
   Velocity reference: BT
   Depth units: ADCP
   Bins: From 128 to 1 skip 0 bin
   Magnetic variation 0.00 deg
   Do not mark data below bottom

END OF REPORT
-------------
```

3.6 BBBATCH Program

When you have mastered the *BBLIST* program, you can use *BBBATCH* to convert binary data sets to ASCII data sets in a DOS batch mode. This comes in handy when you have several data sets to convert or have a large data set that you want to convert overnight. Here is the syntax for *BBBATCH*:

BBBATCH BinaryFileName FormatFileName AsciiFileName

BinaryFileName	= name of the binary data set to convert (no extension needed)
FormatFileName	= name of the format file to use
AsciiFileName	= name of the ASCII data set (unique name)

Running *BBBATCH* without any command line parameters displays the syntax.

4 Utility Software

The following Windows® programs have been provided to supplement data processing. These programs have been installed to the directory C:\Program Files\Rd Instruments\RDI Tools.

4.1 Using BBSub

BBSub is a raw data file subsectioning utility. This allows you to save a portion of a large raw data file as a separate raw data file. *BBSub* starts copying ensembles from the input file to the output file starting with the ensemble specified by the **Start Ensemble Number**. If the **Start Ensemble Number** is specified to zero, *BBSub* starts copying from the first read ensemble. *BBSub* will continue copying ensembles until it reaches ensemble number **End Ensemble Number**. If the **End Ensemble Number** is specified to zero, it copies until the end of the file. Only ensembles with valid checksum are copied.

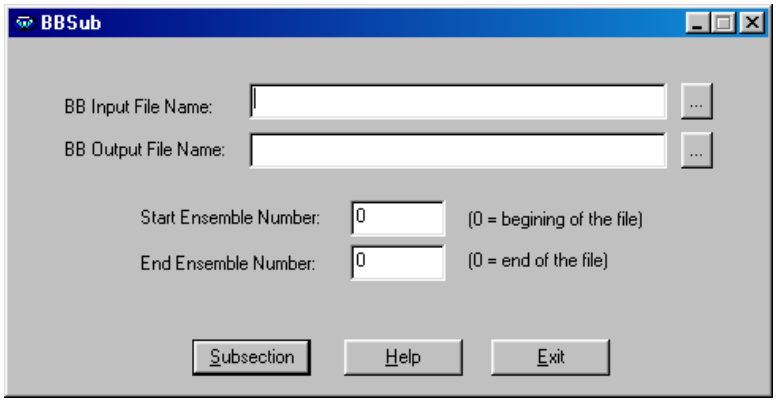


Figure 16. BBSub Display

To subsection a file, do the following.

a. Enter the name and path of the BroadBand ADCP raw data in the **BB Input File Name** box or use the browse button "..." to locate the file. *BBSub* "remembers" the input and output directory if the **Browse** button "..." is selected.

b. Enter the name and path of the subsectioned file in the **BB Output File Name** box or use the **Browse** button "..." to locate the file.

c. Enter the **Start Ensemble Number**.

d. Enter the **End Ensemble Number**.

e. Press the **Subsection** button to begin.

f. Press the **Exit** button to exit *BBSub*.

You can run *BBSub* in the batch mode by writing a simple DOS *.bat file that contains command lines with the files that you want to process. For each file to be processed, add a line to the batch file specifying the name of the raw data file to be subsectioned, the name of the output file, and the starting and ending ensemble numbers.

NOTE. For unattended operation, make sure that output file does not exist; otherwise the batch file will pause to confirm overwriting the file.

Sample Batch file

```
start /w "C:\Program Files\RD Instruments\Utilities\BBSub.exe" -in:InputFile
-out:OutputFile -start:StartEns -end:EndEns
```

4.2 Using *BBConv*

BBConv is a BroadBand raw data to ASCII conversion program. It uses a decoder file (.DEC) to determine how to convert the data (see "Using Decoder Files," page 28). A decoder file contains interpreted instructions for converting to ASCII.

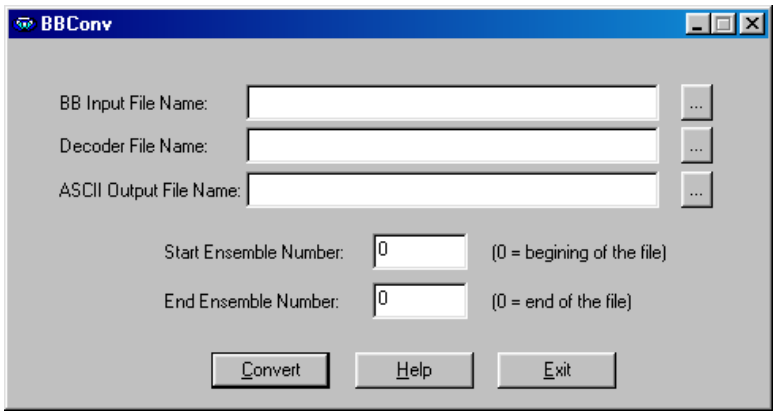


Figure 17. *BBConv* Display

To convert a file, do the following.

a. Enter the name and path of the BroadBand ADCP raw data in the **BB Input File Name** box or use the **Browse** button "..." to locate the file. *BBConv* "remembers" the input and output directory if the **Browse** button "..." is selected.

b. Enter the name and path of the decoder file in the **Decoder File Name** box or use the browse button "..." to locate the file.

c. Enter the name and path of the ASCII file in the **ASCII Output File Name** box or use the **Browse** button "..." to locate the file.

d. Enter the **Start Ensemble Number**.

e. Enter the **End Ensemble Number**.

f. Press the **Convert** button to begin.

g. Press the **Exit** button to exit *BBConv*.

You can run *BBConv* in the batch mode by writing a simple DOS batch file (*.bat) that contains command lines with the files that you want to process. For each file to be processed, add a line to the batch file specifying the name of the raw data file to be converted, the name of the decoder file, the name of the output file, and the starting and ending ensemble numbers.

NOTE. For unattended operation, make sure that output file does not exist; otherwise the batch file will pause to confirm overwriting the file.

Sample Batch file

```
start /w "C:\Program Files\RD Instruments\Utilities\BBConv.exe" -
in:InputFile -dec:DecoderFile -out:OutputFile
```

4.3 Using *BBMerge*

BBMerge merges the ASCII comma delimited format data (created by using *BBConv*), back into the raw data file format, resulting in a new raw data file.

NOTE. The ASCII text file and the BroadBand data input file should have the same number of records. If the text file contains fewer records, then zero values will be merged into the remaining BroadBand data file records. If the BroadBand data file has fewer records, then the remaining ASCII text file will be ignored.

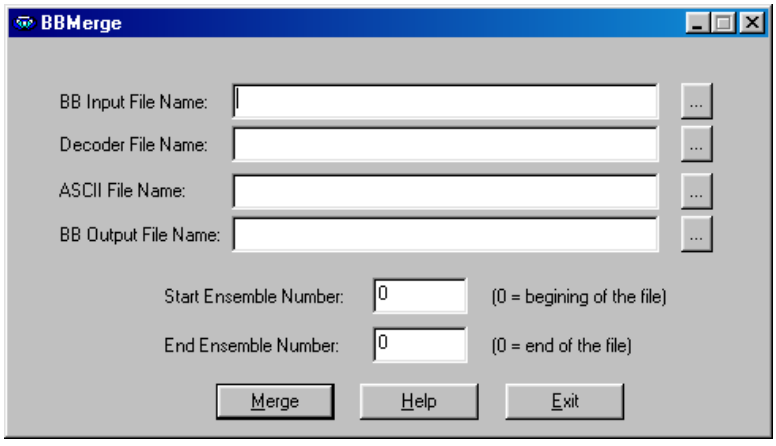


Figure 18. *BBMerge* Display

To merge a file, do the following.

a. Enter the name and path of the BroadBand ADCP raw data in the **BB Input File Name** box or use the **Browse** button "..." to locate the file. *BBMerge* "remembers" the input and output directory if the **Browse** button "..." is selected.

b. Enter the name and path of the decoder file in the **Decoder File Name** box or use the **Browse** button "..." to locate the file.

c. Enter the name and path of the ASCII file in the **ASCII File Name** box or use the **Browse** button "..." to locate the file.

d. Enter the name and path of the new BroadBand ADCP raw data file in the **BB Output File Name** box or use the **Browse** button "..." to locate the file.

e. Enter the **Start Ensemble Number**.

f. Enter the **End Ensemble Number**.

g. Press the **Merge** button to begin.

h. Press the **Exit** button to exit *BBMerge*.

You can run *BBMerge* in the batch mode by writing a simple DOS batch file (*.bat) that contains command lines with the files that you want to process. For each file to be processed, add a line to the batch file specifying the name of the raw data file to be merged, the name of the decoder file, the name of the ASCII file, the name of the new output file, and the starting and ending ensemble numbers.

NOTE. For unattended operation, make sure that output file does not exist; otherwise the batch file will pause to confirm overwriting the file.

Sample Batch File

```
start /w "C:\Program Files\RD Instruments\Utilities\BBMerge.exe" -
in:InputFile -dec:DecoderFile –ascii:ASCIIFile -out:OutputFile
```

4.4 Using BBCheck

BBCheck checks your data files for integrity and quality. It does some simple error and problem checking. You should use *BBCheck* on your data files before you erase the original data files on your recorder.

BBCheck performs the following tests.

- Valid checksum
- Sequential ADCP ensemble number
- Built in test code (BIT) is zero
- Configuration byte remains constant
- All data types identified

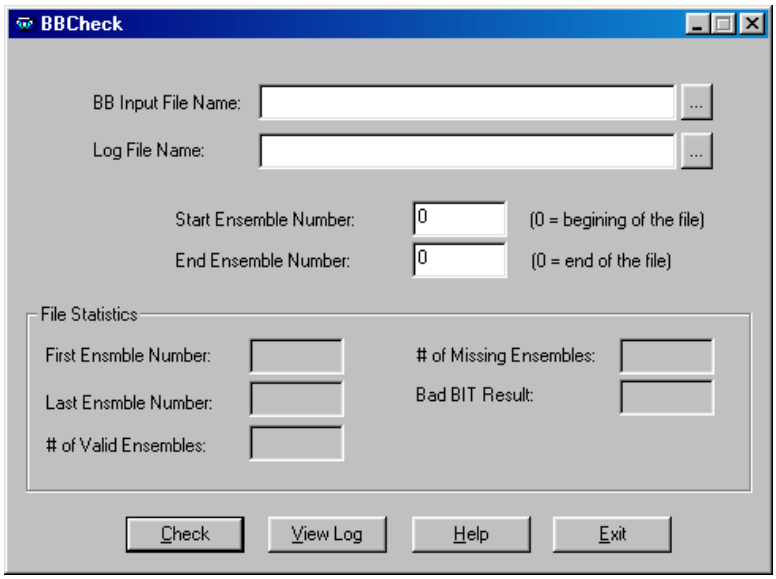


Figure 19. BBCheck Display

To check a file for errors, do the following.

a. Enter the name and path of the BroadBand ADCP raw data in the **BB Input File Name** box or use the **Browse** button "..." to locate the file. *BBCheck* "remembers" the input and output directory if the **Browse** button "..." is selected.

b. Enter the name and path of the log file in the **Log File Name** box or use the **Browse** button "..." to locate the file. *BBCheck* will create the log file if it does not exist.

c. Enter the **Start Ensemble Number**.

d. Enter the **End Ensemble Number**.

e. Press the **Check** button to begin.

f. Press the **View Log** button to view detailed information about the file and the data ID numbers detected by *BBCheck*.

g. Press the **Exit** button to exit *BBCheck*.

You can run *BBCheck* in the batch mode by writing a simple DOS *.bat file that contains command lines with the files that you want to check. For each file to be checked, add a line to the batch file specifying the name of the raw data file to be checked, the name of the output log file, and the starting and ending ensemble numbers.

NOTE. For unattended operation, make sure that the output log file does not exist; otherwise the batch file will pause to confirm overwriting the file.

Sample Batch file

```
start /w "C:\Program Files\RD Instruments\Utilities\BBCheck.exe" -
in:InputFile  -out:OutputLogFile -start:StartEns -end:EndEns
```

4.5 Using BBSlice

BBSlice copies ensembles from the input file to an output file checking each ensemble number. If the ensemble number is not sequential, a new deployment file is opened, and the ensembles are copied into it. The user supplies an **Output File Prefix** and the deployment file name is created by adding "nnn.mmm", where "nnn" is deployment number starting with "000". "mmm" is a file extension that starts with "000" and increases if file reaches a maximum file size supplied by the user. If the maximum file size is 0 there is no limitation on the file size of the output file.

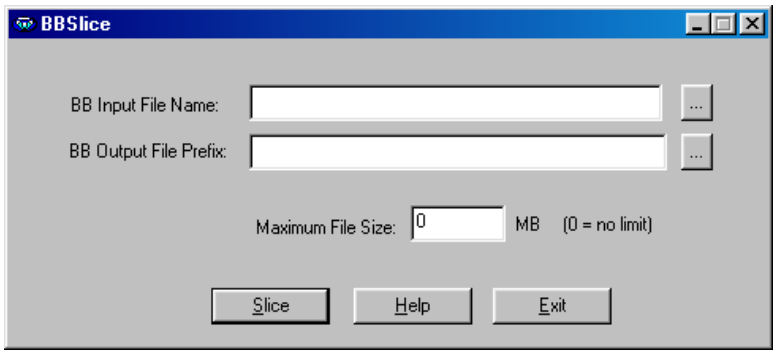


Figure 20. BBSlice Display

To slice a file, do the following.

a. Enter the name and path of the BroadBand ADCP raw data in the **BB Input File Name** box or use the **Browse** button "..." to locate the file. *BBSlice* "remembers" the input and output directory if the Browse button "..." is selected.

b. Enter the file name prefix in the **BB Output File Prefix** box or use the browse button "..." to locate the file.

c. Enter the **Maximum File Size**.

d. Press the **Slice** button to begin.

e. Press the **Exit** button to exit *BBSlice*.

You can run *BBSlice* in the batch mode by writing a simple DOS *.bat file that contains command lines with the files that you want to process. For each file to be processed, add a line to the batch file specifying the name of the raw data file to be subsectioned, the name of the output file, and the starting and ending ensemble numbers.

NOTE. For unattended operation, make sure that output file does not exist; otherwise the batch file will pause to confirm overwriting the file.

Sample Batch file

```
start /w "C:\Program Files\RD Instruments\Utilities\BBSlice.exe" -
in:InputFile  -out:OutputFile -size:MaxFileSize
```

4.6 Using BBss Speed of Sound Calculator

BBss allows you to quickly calculate the speed of sound in the water. Enter values for the water temperature, salinity, and depth. The Sound Speed and Coefficient will be calculated based on the parameters entered.

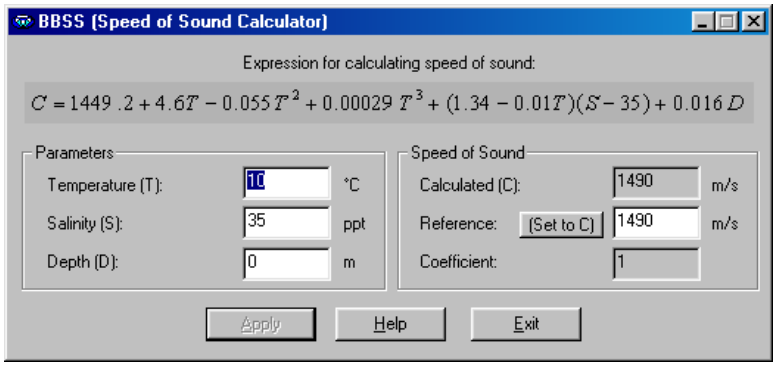


Figure 21. BBss Display

To calculate the speed of sound, do the following.

a. Enter the water temperature in the **Temperature** box.

b. Enter the water salinity in the **Salinity** box.

c. Enter the water depth in the **Depth** box.

d. Enter a reference speed of sound in the **Reference** box. The default is 1490 m/s.

e. Click **Apply**. The calculated speed of sound and coefficient will appear in the Calculated (C) and Coefficient boxes.

f. Click **Exit** to exit *BBss*.

5 Using Decoder Files (*.DEC)

RDI provides example DEC files that help remove user-selected data from binary data files and stores the information into ASCII comma delimited format.

A typical decoder file may contain the following:

```
; Temperature Data Decoder File
0080,3,2
0080,27,-2,0.01,2 ; Temperature (ET) in degrees
$L
```

- Lines beginning with a semicolon (;) are ignored and are for user comments.
- In addition, decoder lines may be followed by semi-colons for additional comments.
- Blank lines are ignored.
- "$L" signifies a CR/LF should be output before outputting more data

Lines containing decoding instructions are broken down as follows:

id,offset,size,multiplier,format

where:

id	Data Structure identifier in hex (e.g. 0080). This information is presented MSB-LSB. Be careful, since the data stream outputs LSB followed by MSB.
offset	The byte # from the first byte of the data structure (e.g. The first data byte in the structure following; The ID is byte #3 . Byte #1 is the LSB of the ID Byte #2 is the MSB of the ID Byte #3 is the first data byte of the structure.
size	The number of bytes contiguous bytes contained in the data (LSB to MSB). A negative number indicates signed data; a positive number indicates unsigned data.
multiplier	The conversion factor (optional)
format	The format specifies the number of decimal places to output (optional).

Additionally, if the line has a plus sign (+) at the end of the line, the addition is performed on the data in the line and the line following.

For example, to decode the range of beam 1 through 4 in the bottom track data the following lines should be included.

```
0600,17,2,0.01,0,+          ; Beam #1 Range (cm)
0600,78,1,655.36,2        ; Beam #1 Range MSB
0600,19,2,0.01,0,+          ; Beam #2 Range (cm)
0600,79,1,655.36,2        ; Beam #2 Range MSB
0600,21,2,0.01,0,+          ; Beam #3 Range (cm)
0600,80,1,655.36,2        ; Beam #3 Range MSB
0600,23,2,0.01,0,+          ; Beam #4 Range (cm)
0600,81,1,655.36,2        ; Beam #4 Range MSB
$L
```

In order to simply output of the profile data you can use a combination of two lines. For example, to decode the velocity the following lines should be included.

```
$P idB, offsetB, typeB, offset, bins
idP, offsetP, typeP, multiplier, formatP
```

Where

idB, offsetB, typeB	points to the number that describes the number of bins (normally 000,10,1)
offset	is the number of bytes to advance to the next bins (velocity is 8 bytes, intensity is 4 bytes)
Bins	are the string that describes what bins to output. For example, "1-10" would output bins one through 10, "1,5,7" would output bins one, five and seven, or "1-5,10-20" would output bins one through five and ten through twenty.
idP, offsetP, typeP	points to the data in the first bin for the profile (0100,3,-2 in the example shown below)

```
0080,3,2                        ; Ensemble Number
$P0000,10,1,8,"1-255"           ; Starting Profile
0100,3,-2                       ; Beam 1 Data
$L
$L
```

NOTE. See the included decoder files (*.DEC) for examples. To create your own decoder file, simply cut and paste the appropriate offset information into your own ASCII decoder file. The figures in the Command and Output Data Format guide should aid in creating decoding files.

Table 4: Decoder Files Included with RDI Tools CD

DEC File Name	Description
ADC.DEC	Decoder file for only the ADC channel data
BOTTOM.DEC	Bottom Track Data Decoder File
BTNMACOG.DEC	NMEA Bottom Track Data Decoder Information
BTVEL.DEC	Bottom Track velocity Data Decoder File
DEPTH.DEC	Decoder file for only the Transducer Depth data

Continued Next Page

Table 4: Decoder Files Included with RDI Tools CD (continued)

DEC File Name	Description
DISBIN1.DEC	Decoder file for only the Bin 1 Distance
ENSNO.DEC	Decoder file for only the Ensemble Number data
ENSTIME.DEC	Decoder file for only the Ensemble time Real-Time Clock
ENSTIME2.DEC	Decoder file for Ensemble Number and Time Decoder file when collecting both BB/NB OS Ensembles
EX.DEC	Decoder file for only the Coordinate Transformation (EX)
FIXDLDR2.DEC	Fixed Leader decoder file for OS ensembles with BB/NB pings
FIXEDLDR.DEC	Binary Fixed Leader Decode File
H.DEC	Decoder file for only the Heading data
HDR1.DEC	Header Decoder File ID 7F7F
HDR2.DEC	Header Decoder File ID 0000
HPR.DEC	Binary Fixed & Variable Leader Decode File for Heading, Pitch, and Roll
ID.DEC	Decoder File for decoding the ID numbers
ID2.DEC	Decoder File for decoding the ID numbers from OS Ensembles with both BB/NB Pings
IDSTNDRD.DEC	Standard ID Number Decoder File
LATLONG.DEC	Decoder file used to extract the Lat and Long values from the *VMDAS* ENS, ENX, STA, and LTA data files
LDR&NMEA.DEC	Variable Leader and NMEA Heading Data Decoder Information
LEADER.DEC	Binary Fixed & Variable Leader Decode File
LEADER2.DEC	Binary Fixed & Variable Leader Decoder File for NB Data when Collecting OS ensembles with both BB and NB pings
NGOOD.DEC	Decoder file for only the Number good pings (25 bins only)
NMEA.DEC	Decoder file for removing NMEA data from files created by *VMDAS*
NMEA_ENS.DEC	NMEA Data Decoder Information Ensemble Number
NMEAVEL.DEC	NMEA Data Decoder Information Velocity
SALIN.DEC	Decoder file for only the Salinity data
SUMSQR.DEC	Decoder file for only the sum of (velocity^2) (25 bins only)
SUMVEL.DEC	Decoder file for only the velocity sum (25 bins only)
TIME.DEC	Decoder file for removing Date and Time from Ensemble Data
UPDWN.DEC	Decoder file for only the Orientation data
WBCMD.DEC	Decoder file for only the Fixed Leader WB Command setting Decoder Information
WHCPUsn.dec	Binary Fixed Leader Decode File for WorkHorse CPU Serial Number
WNRIVGPS.DEC	WINRIVER Navigation Decode File

NOTE. The LEADER.DEC is the master ASCII Decoder file for variable and fixed leader data. Use the BOTTOM.DEC file for only the Bottom Track data.

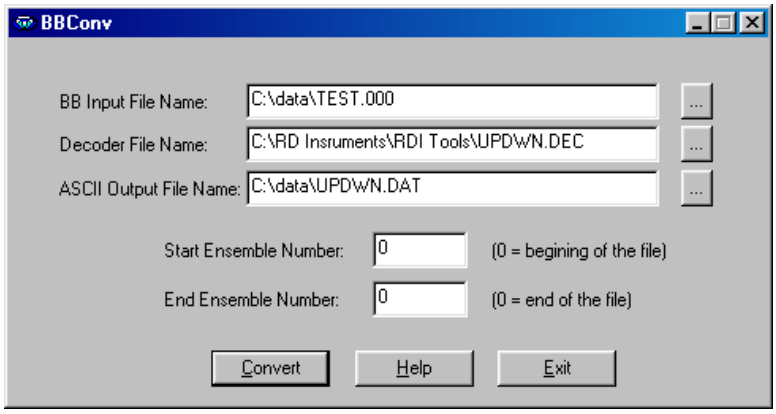
5.1 Example Using UPDWN.DEC

RD Instruments' software only reads the first ensemble to determine the orientation of the ADCP. If your system started pinging and recording data before it was in the correct orientation, then the programs will display the data upside down. If you collected data in beam coordinates, our software will also display the velocity components with the wrong sign. To correct this you must convert the configuration byte in the ADCP leader data containing the orientation setting to the correct value. The following paragraphs explain how to use *BBConv* and *BBMerge* to convert the configuration byte in the ADCP data set to all up or all down.

a. You must first determine which ensemble number has the proper setting of the configuration byte. Use the *BBLIST* program to view your ADCP data. Once your data is loaded press **<ALT+D>** and then **A** to enter the ADCP setup screen in *BBLIST*. Look at the setting in the **ADCP Hardware and Firmware** window for the ADCP orientation.

   Now, press the **+** key until you move to the ensemble that has the correct orientation for your deployment. Record this ensemble number for future use.

b. Use *BBConv* to copy the configuration byte from the ADCP binary data file to an ASCII data file. For this example, we will assume that your raw data file is TEST.000.



Example UPDWN.DEC file:

```
;  Binary Fixed & Variable Leader Decode File
0080,3,2                        ; Ensemble Number
0000,5,2                        ; System Configuration
$L
```

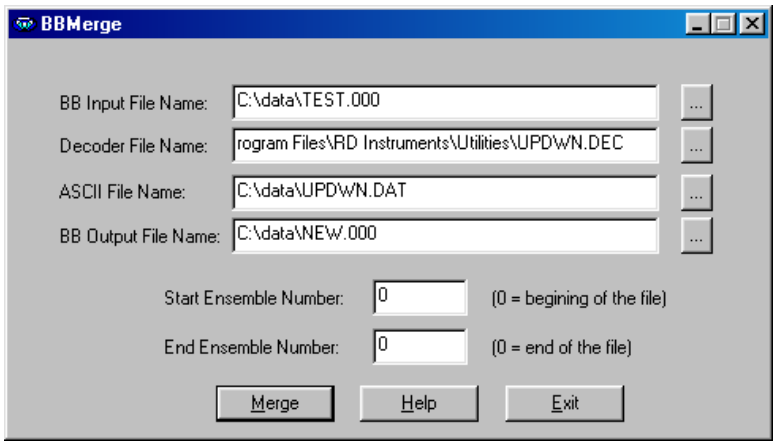
An ASCII text file called UPDWN.DAT will be created. It will have two columns of data. The first column will be the ensemble number and the second column will be the configuration byte.

NOTE. *BBConv* will automatically go to the next extension (i.e. TEST.001) if it exists.

c. Use a text editor to edit the UPDWN.DAT file so that the configuration byte is the same for all ensembles. Locate the ensemble number you recorded in Step "a" and copy the configuration byte setting to all of the other ensembles. Most text editors will allow you to use a "find and replace" option.

   After all of the configuration bytes have been changed, save the file. Be sure to not add any other characters to the file.

d. Now use *BBMerge* to write the new configuration byte back into your data set. *BBMerge* will write the data into a *new* data file. It will not affect the original raw data file.



NOTE. *BBMerge* will automatically go to the next extension (i.e. TEST.001) if it exists.

e. You can now playback your data and the orientation will be correct. If you collected data in beam coordinates, the velocity data will now have the correct signs.

5.2 Example Using ADC.DEC

You can use *BBConv* to obtain the ADC channel information and the heading, pitch, roll, standard deviation information. RD Instruments' *BBLIST* software does not allow you to output the ADC data or the heading, pitch, and roll standard deviation information. Using the *BBConv* program it is possible to output this information into an ASCII text file.

a. You must use the *BBConv* program to copy the ADC, heading, pitch, and roll bytes from the WorkHorse or BroadBand ADCP binary data file to an ASCII data file. To do this, use the following example of what to

enter at the *BBConv* screen. For this example, we will assume that your file name is TEST.000.

BB Input File Name	TEST.000
Decoder File Name	ADC.DEC
ASCII Output File Name	ADC.DAT

An ASCII text file called ADC.DAT will be created. It will have 2 columns of data. The first column will be the ensemble number and the second column will be the heading bytes.

NOTE. *BBCONV* will automatically go to the next extension (i.e. TEST.001) if it exists.

5.3 Example Using DEPTH.DEC

Use DEPTH.DEC to change the depth in the BroadBand, WorkHorse, or Ocean Surveyor ADCP leader data. RD Instruments' software only reads the first ensemble to determine the depth of the ADCP. If you entered the wrong depth during the deployment setup of your system then our software programs will set the depth of the bins incorrectly.

To correct this you must change the depth in the variable leader data for at least the first ensemble. The following explains how to use *BBConv* and *BBMerge* to change the depth bytes in the WorkHorse and BroadBand ADCP data set to a new value.

b. You must use the *BBConv* program to copy the depth bytes from the WorkHorse or BroadBand ADCP binary data file to an ASCII data file. To do this, use the following example of what to enter at the *BBConv* screen. For this example, we will assume that your file name is LEG3009_000000.ENR.

BB Input File Name	LEG3009_000000.ENR
Decoder File Name	DEPTH.DEC
ASCII Output File Name	DEPTH.DAT

c. An ASCII text file called DEPTH.DAT will be created. It will have two columns of data. The first column will be the ensemble number and the second column will be the depth byte.

NOTE. *BBConv* will automatically go to the next file increment (i.e. LEG3009_000001.ENR) if it exists.

d. You must now edit the DEPTH.DAT file so that the depth of at least the first ensemble is the correct depth. You may edit the depth for all en-

sembles but only the first one needs to be edited for our software to scale your depth cells correctly. Locate ensemble number one and change it to match the correct depth.

e. If you wish to change all ensembles then most text editors will allow you to use a find and replacement option. After the depth bytes have been changed save the file. Be sure to not add any other characters to the file.

f. You can now use *BBMerge* to write the new depth bytes back into your data set. *BBMerge* will actually write the data into a new data file. It will not affect the original file. To do this, use the following example of what to enter at the *BBMerge* screen.

BB Input File Name	LEG3009_000000.ENR
Decoder File Name	DEPTH.DEC
ASCII Output File Name	DEPTH.DAT
BroadBand Output File Name	NEW.000

The file NEW.000 will be the corrected data file.

NOTE. *BBMerge* will automatically go to the file increment (i.e. LEG3009_000001.ENR) if it exists.

g. You can now playback your data and the depth of your ADCP and your bins will be correct.

5.4 Example Using ENSEMBLE.DEC

Use *BBConv* and *BBMerge* to change the ensemble number in the WorkHorse or BroadBand ADCP leader data. RD Instruments' software reads the ensemble number in each ensemble to determine the sequence of the ensembles. If you stop and start a recording cycle in the ADCP the ensemble number will re-initialize back to one. Some programs are sensitive to this (RDI's *WinADCP* program for example) and will not combine the files together later.

To correct this you must change the ensemble number in the files following your first file so that they continue the ensemble number sequence. The following explains how to use *BBConv* and *BBMerge* to change the ensemble number bytes in the WorkHorse and BroadBand ADCP data sets to new values.

a. You must use the *BBConv* program to copy the ensemble number bytes from the WorkHorse or BroadBand ADCP binary data file to an ASCII data file. To do this, use the following example of what to enter at the

BBConv screen. For this example, we will assume that your file name is TEST.000.

BB Input File Name	TEST.000
Decoder File Name	ENSNO.DEC
ASCII Output File Name	ENSNO.DAT

b. An ASCII text file called ENSNO.DAT will be created. It will have a single column of data. This column will be the ensemble number in the data set you recorded.

NOTE. *BBConv* will automatically go to the next extension (i.e. TEST.001) if it exists.

c. You must now edit the ENSNO.DAT file so that the ensemble numbers are sequential. Note, the ensemble number must roll over at ensemble 65535. Most spreadsheet programs would allow you to do this editing very easily. After the ensemble numbers have been changed save the file. Be sure to not add any other characters to the file.

d. You can now use *BBMerge* to write the new ensemble number bytes back into your data set. *BBMerge* will actually write the data into a new data file. It will not affect the original file. To do this, use the following example of what to enter at the *BBMerge* screen.

BB Input File Name	TEST.000
Decoder File Name	ENSNO.DEC
ASCII Output File Name	ENSNO.DAT
BroadBand Output File Name	NEW.000

The file NEW.000 will be the corrected data file.

NOTE. *BBMerge* will automatically go to the next extension (i.e. TEST.001) if it exists.

e. You can now playback your data and the ensemble numbers will be sequential.

5.5 Example Using EX.DEC

Use *BBConv* and *BBMerge* to set the EX command setting in the Broad-Band ADCP leader data. RD Instruments' software reads the EX Command setting to determine the coordinate system the ADCP was set in. If your leader data recorded the wrong coordinate system then you will not be able to use RDI's software to correctly display the data in the coordinate system you wish.

To correct this you must convert the EX Command byte in the BroadBand ADCP leader data to the correct value. The following explains how to use *BBConv* and *BBMerge* to convert the configuration byte in the BroadBand ADCP data change the setting to correct value.

a. You must use the *BBConv* program to copy the EX Command byte from the BroadBand ADCP binary data file to an ASCII data file. To do this, use the following example of what to enter at the *BBConv* screen. For this example, we will assume that your file name is TEST.000.

BB Input File Name	TEST.000
Decoder File Name	EX.DEC
ASCII Output File Name	EX1.TXT

b. An ASCII text file called EX1.TXT will be created. It will have two to three columns of data. The first column will be the EX Command setting, the second column will be the ensemble number LSB of the ensemble number and the third column will be the MSB of the ensemble number.

NOTE. *BBConv* will automatically go to the next extension (i.e. TEST.001) if it exists.

c. You must now edit the EX1.TXT file so that the EX Command byte is the same and correct for all ensembles. Change the EX Command byte from its present value to the values shown here for your correct coordinate system:

0	= Beam
15	= Instrument
23	= Ship
31	= Earth

After all of the configuration bytes have been changed save the file to a new file called EX2.TXT. Be sure to not add any other characters to the file.

d. You can now use *BBMerge* to write the new configuration byte back into your data set. *BBMerge* will actually write the data into a new data file. It will not affect the original file. To do this, use the following example of what to enter at the *BBMerge* screen.

BB Input File Name	TEST.000
Decoder File Name	EX.DEC
ASCII Output File Name	EX2.TXT
BroadBand Output File Name	NEW1.000

The file NEW1.000 will be the corrected data file.

NOTE. *BBMerge* will automatically go to the next extension (i.e. TEST.001) if it exists.

e. You can now playback your data and the coordinate system will be correct.

5.6 Example Using H.DEC

Use H.DEC to change the heading in the WorkHorse or BroadBand ADCP leader data. RD Instruments' software reads the heading in each ensemble to determine the direction the ADCP is pointing. If you entered the wrong heading or your heading is incorrect during the deployment of your system then our software programs will calculate the wrong direction of the velocity data.

To correct this you must change the heading in the variable leader data for at each ensemble. The following explains how to use *BBConv* and *BBMerge* to change the heading bytes in the WorkHorse and BroadBand ADCP data set to a new value.

a. You must use the *BBConv* program to copy the heading bytes from the WorkHorse or BroadBand ADCP binary data file to an ASCII data file. To do this, use the following example of what to enter at the *BBConv* screen. For this example, we will assume that your file name is TEST.000.

BB Input File Name	TEST.000
Decoder File Name	H.DEC
ASCII Output File Name	HDG.DAT

b. An ASCII text file called HDG.DAT will be created. It will have two columns of data. The first column will be the ensemble number and the second column will be the heading bytes.

NOTE. *BBConv* will automatically go to the next extension (i.e. TEST.001) if it exists.

c. You must now edit the HDG.DAT file so that the heading of each ensemble is correct. Most text editors will allow you to use a find and replacement option to change a specific value.

d. After the heading bytes have been changed save the file. Be sure to not add any other characters to the file.

e. You can now use *BBMerge* to write the new heading bytes back into your data set. *BBMerge* will actually write the data into a new data file. It will not affect the original file. To do this, use the following example of what to enter at the *BBMerge* screen.

BB Input File Name	TEST.000
Decoder File Name	H.DEC
ASCII Output File Name	HDG.DAT
BroadBand Output File Name	NEW.000

The file NEW.000 will be the corrected data file.

NOTE. *BBMerge* will automatically go to the next extension (i.e. TEST.001) if it exists.

f. You can now playback your data and the heading of your ADCP and the direction of your velocity data will be correct.

5.7 Example Using NBID2BB.DEC

Use NBID2BB.DEC to change the NarrowBand ID number to the BroadBand ID number. The Ocean Surveyor ADCP is capable of collecting both BroadBand and NarrowBand pings in the same ensemble. When this happens it writes the BroadBand data (fixed leader, variable leader, velocity, correlation, echo intensity, and percent good) to the standard ID numbers (0000, 0080, 0100, 0200, 0300, and 0400 respectively). The NarrowBand data (fixed leader, variable leader, velocity, correlation, echo intensity, and percent good) is given new ID numbers (0001, 0081, 0101, 0201, 0301, and 0401 respectively). This is done so that 2 different data types can exist in the same ensemble.

RD Instruments' software reads only the standard the ID numbers when decoding binary data. As a result only the BroadBand ensembles will be decoded. By changing the ID numbers of the both the BroadBand and NarrowBand data types it is possible to have the NarrowBand data be read by either *WinADCP* or *VmDas*.

NOTE. Currently the *BBLIST* program will continue to read the BroadBand data fixed leader type but does appear to display the velocity data from the NarrowBand ensembles once this procedure has been followed.

a. You must use the *BBConv* program to copy the ID numbers from both the BroadBand and NarrowBand data types into a text file. To do this, use the following example of what to enter at the *BBConv* screen. For this example, we will assume that your file name is LEG3009_000000.ENR.

BB Input File Name	LEG3009_000000.ENR
Decoder File Name	ID2.DEC
ASCII Output File Name	1BBNBID.TXT

An ASCII text file called 1BBNBID will be created. It will have 14 columns of data. The first two columns will be the ensemble number LSB and MSB. Columns 3 through 8 will be the ID numbers for the BB data and columns 9 through 14 will be the ID numbers for the NB data.

NOTE. *BBConv* will automatically go to the next file increment (i.e. LEG3009_000001.ENR) if it exists.

b. You must now edit the 1BBNBID.TXT file so that the ID numbers for the BroadBand data are changed from 0, 128, 256, 512, 768, 1024 to 2, 130, 258, 514, 770, 1026 respectively.

NOTE. The values in the file and shown here are the decimal equivalent of the hex values actually found in the binary data files.

You must also change the ID numbers for the NarrowBand data from 1, 129, 257, 513, 769, 1029 to 0, 128, 256, 512, 768, 1024 respectively.

While, most text editors will allow you to use a find and replacement option to change a specific value. It is easier to load this data into a spreadsheet program such EXCEL and then replace the values for the first row of data and then copy that through the rest of the ensembles. If done through EXCEL the changed file can be saved as a MS-DOS CSV file. A CSV file will contain all of the comma delimiters required for use with *BBMerge* in the next step.

NOTE. You will have to change the file name to *.TXT. It is best to save your modified data to a different file name such as 2BBNBID.TXT.

When working with this file be sure to not add any other characters to the file as this cause incorrect data to be merged into your data set.

c. You can now use *BBMerge* to write the new ID Number bytes back into your data set. *BBMerge* will actually write the data into a new data file. It will not affect the original file. To do this, use the following example of what to enter at the *BBMerge* screen.

BB Input File Name	LEG3009_000000.ENR
Decoder File Name	ID2.DEC
ASCII Output File Name	2BBNBID.TXT
BroadBand Output File Name	NEW39_000000. ENR

The file NEW39ENR.000 will be the corrected data file and will contain the ID numbering.

NOTE. *BBMerge* will automatically go to the next extension (i.e. LEG3009_000000.ENR) if it exists.

d. You can now reprocess the raw data with *VmDas* so that you are using the NarrowBand data.

5.8 Example Using NMEA.DEC

Use NMEA.DEC to extract the NMEA data from the *VmDas* STA and LTA files. RD Instruments' software program *VmDas* creates Short Term Average (STA) and Long Term Average (LTA) files. These files have the exact same format as the original output of your ADCP, but contain one more data type. This data type is called NMEA Data. It contains the latitude, longitude, heading, pitch, and roll information that was collected when using *VmDas*.

The following procedure explains how to use *BBConv* to extract the NMEA data from the STA and LTA files.

a. You must use the *BBConv* program to extract the heading bytes from the STA or LTA *VmDas* binary data files into an ASCII data file. Run *BBConv*. For this example, we will assume that your file name is ADCP007_000000.STA.

BB Input File Name	ADCP007_000000.STA
Decoder File Name	NMEA.DEC
ASCII Output File Name	ADCP007.TXT

NOTE. *BBConv* will automatically go to the next file increment (i.e. ADCP007_000001.STA) if it exists.

An ASCII text file called ADCP007.TXT will be created. It will contain 32 columns of data with each column being comma delimited. The

meaning of each column is explained in the NMEA section of the STA and LTA Data Format description or you can refer to the NMEA.DEC text file.

b. Several of the fields in the ASCII file will have to be converted or scaled into their proper units. To scale the data refer to the NMEA section of the STA and LTA Data Format description.

   Included with this is an example *Excel* spread sheet, NMEA.XLS that has the formulas for scaling all of the data. To use this spreadsheet, open your *BBConv* output text file with *Excel* and then copy this data onto the ***BBConv* DATA IN** tab in the NMEA.XLS spread sheet. You can then click on the **REAL UNITS** tab and you find your data has been converted.

NOTE. An example of data from the file ADCP007.TXT is already included in the NMEA.XLS spreadsheet. You can copy your data right over the data already in the file or copy the NMEA.XLS file to another name. Keep in mind that, the **Real Units** tab is only setup for a limited number of averages. Therefore, you may need to copy the formulas from the final row to additional rows.

5.9 Example Using SALIN.DEC

Use SALIN.DEC to change the salinity in the WorkHorse or BroadBand ADCP leader data. RD Instruments' software reads the salinity in each ensemble to determine the speed of sound for the ADCP velocity measurement. If you entered the wrong salinity during the deployment setup of your system then our software programs will calculate the velocity and depth of each bin incorrectly.

To correct this you must change the salinity in the variable leader data for at each ensemble. The following explains how to use *BBConv* and *BBMerge* to change the salinity bytes in the WorkHorse and BroadBand ADCP data set to a new value.

a. You must use the *BBConv* program to copy the salinity bytes from the WorkHorse or BroadBand ADCP binary data file to an ASCII data file. To do this, use the following example of what to enter at the *BBConv* screen. For this example, we will assume that your file name is TEST.000.

BB Input File Name	TEST.000
Decoder File Name	SALIN.DEC
ASCII Output File Name	SALIN.DAT

b. An ASCII text file called SALIN.DAT will be created. It will have two columns of data. The first column will be the ensemble number and the second column will be the salinity byte.

NOTE. *BBConv* will automatically go to the next extension (i.e. TEST.001) if it exists.

c. You must now edit the SALIN.DAT file so that the salinity of each ensemble is correct. Most text editors will allow you to use a find and replacement option to change a specific value. After the salinity bytes have been changed save the file. Be sure to not add any other characters to the file.

d. You can now use *BBMerge* to write the new salinity bytes back into your data set. *BBMerge* will actually write the data into a new data file. It will not affect the original file. To do this, use the following example of what to enter at the *BBMerge* screen.

BB Input File Name	TEST.000
Decoder File Name	SALIN.DEC
ASCII Output File Name	SALIN.DAT
BroadBand Output File Name	NEW.000

The file NEW.000 will be the corrected data file.

NOTE. *BBMerge* will automatically go to the next extension (i.e. TEST.001) if it exists.

e. You can now playback your data and the salinity of your ADCP and your bins will be correct.

5.10 Example Using SUM.DEC

Use SUM.DEC to pull out the sum of square data. RD Instruments' recently added the following data to its data format: Number of good pings, Sum of Velocity Squared, and Sum of Velocities. With this information it is possible to calculate the standard deviation of the velocity data for each bin in each ADCP ensemble.

Regrettably, the standard RDI software (such as *BBList* and *WinADCP*) does not know how to decode or use this data. In order to gain access to this data we created the *BBConv* program and the Decode files to produce ASCII files. These files can then be loaded into a spreadsheet to calculate the standard deviation.

The following explains how to use *BBConv* and the Decode files to convert the above-mentioned data into ASCII files.

a. You must first modify each of the Decode files (NGOOD, SUMSQR, SUMVEL) that have been provided so that you are decoding all the bins of data you wish. Each of the files already will decode the first 25 bins of data in each ensemble.

   To increase (or decrease) the numbers of bins use a text editor to modify each of the Decode files. You can copy the format that is already there and just change the pointer location value to increase the number of bins being decoded.

b. You will need to run the *BBConv* program three separate times (once for each data type) on each data set that you have.

   For example, if the data set was named RDI001.000 - RDI001.099 and we wanted the Number of Good Pings then the format would be as follows:

BB Input File Name	RDI001.000
Decoder File Name	NGOOD.DEC
ASCII Output File Name	NGOOD.TXT

NOTE. *BBConv* will automatically go to the next extension (i.e. RDI001.001) if it exists.

   This will cause the program to go through the entire data set from extension .000 through .099 and output the number of good pings for each bin in a text file called NGOOD.TXT. You can choose any name for the file you wish. You would then repeat the above for the Sum of Velocity Squared and Sum of Velocities, i.e.:

BB Input File Name	RDI001.000
Decoder File Name	SUMSQR.DEC
ASCII Output File Name	SUMSQR.TXT

and

BB Input File Name	RDI001.000
Decoder File Name	SUMVEL.DEC
ASCII Output File Name	SUMVEL.TXT

5.11 Example Using TIME.DEC

The following procedure explains how to use TIME.DEC to extract the Date and Time data from the BroadBand, WorkHorse, or Ocean Surveyor ADCP data. RD Instruments' software stamps each ensemble with the date and time. If the date or time has been set wrong for your deployment it is possible to change the date or time using the following procedure.

a. You must use the *BBConv* program to extract the date and time bytes from the variable leader data for your data set. Run *BBConv*. For this example, we will assume that your file name is ADCP001_000000.STA.

BB Input File Name	ADCP001_000000.STA
Decoder File Name	TIME.DEC
ASCII Output File Name	ADCP0000.TXT

NOTE. *BBConv* will automatically go to the next file increment (i.e. ADCP001_000001.STA) if it exists.

An ASCII text file called ADCP0000.TXT will be created. It will contain seven columns (fields) of data with each column containing two characters (one byte) of data. Each field is comma delimited. The meaning of each column is as follows: YY, MM, DD, HH, MM, SS, HH

Where: YY = Year, MM = Month, DD = Day, HH = Hour, MM = Minute, SS = Second, HH = Hundreths of Seconds

b. You must now edit the ADCP0000.TXT file so that the Date and Time of each ensemble is correct. Most text editors will allow you to use a find and replacement option to change a specific value.

After the date and time bytes have been changed save the file. Be sure to not add any other characters to the file.

c. You can now use *BBMerge* to write the new date and time bytes back into your data set. *BBMerge* will actually write the data into a new data file. It will not affect the original file. To do this, use the following example of what to enter at the *BBMerge* screen.

BB Input File Name	ADCP001_000000.STA
Decoder File Name	TIME.DEC
ASCII Output File Name	ADCP0000.TXT
BroadBand Output File Name	NEW0000.000

The file NEW0000.000 will be the corrected data file.

NOTE. *BBMerge* will automatically go to the next extension (i.e. NEW0000.001) if it exists.

d. You can now playback your data and the heading of your ADCP and the date and times of your data (NEW0000.*) will be correct.

5.12 Example Using WHCPUsn.DEC

Use WHCPUsn.DEC to obtain the WorkHorse CPU serial number. RD Instruments' (RDI) WorkHorse ADCPs all are given unique serial numbers both for the system and the individual boards within a system. The system serial number is selected by RDI and is provided on the RDI logo and sticker on the pressure case. The individual board serial numbers are electronic serial numbers programmed at RDI. The overall system serial number is not logged in the data collected by the WorkHorse ADCP. However, the WorkHorse CPU electronic serial number is logged in all WorkHorse data sets as part of the fixed leader data of all WorkHorse ADCPs with firmware version 8.20 and higher.

The WorkHorse electronic serial number can also be viewed when the PS0 command is sent to the ADCP. A copy of the PS0 command is stored in all RDI testing files that are run as part of either RDI's DOS based programs or RDI's Windows program *BBTalk* and the associated script file for testing.

Using the WorkHorse CPU electronic serial number recorded in these tests it is possible to cross-reference this to your system serial number. However, RDI recommends that you create your own log of electronic and system serial numbers, because you may update or repair your WorkHorse system in the field and this would change the electronic serial number.

The following procedure explains how to read out the WH CPU electronic serial number from your data set so that you can confirm which system the data set is from. Please note you must have created a cross-reference list first to make this confirmation. If you have not, then it is possible for RDI to perform this cross-reference assuming that you have not changed the WorkHorse CPU board in the field.

a. You must use the *BBConv* program to copy the CPU electronic serial number from the WorkHorse ADCP binary data file to an ASCII data file. Run *BBConv*. For this example, we will assume that your file name is TEST.000.

BB Input File Name	TEST.000
Decoder File Name	WHCPUSN.DEC
ASCII Output File Name	WHCPUSN.TXT

NOTE. *BBConv* will automatically go to the next extension (i.e. TEST.001) if it exists.

b. An ASCII text file called WHCPUSN.TXT will be created. It will have eight columns of data (each column separated by a comma). Each row will repeat the same eight values, as this is the same CPU serial number

recorded in each ensemble. Therefore, only the first row of numbers is required.

To match the values in the first row with the CPU serial number you must convert each comma-separated value from its decimal value to a hexadecimal value.

c. Use the CPU serial number you have obtained and your PS0 results to cross reference to determine your WorkHorse ADCP system serial number. The following example illustrated what checks would be done.

1. Example of PS0 Results from a WorkHorse ADCP s/n 100

```
[BREAK Wakeup A]
WorkHorse Broadband ADCP Version 16.17
RD Instruments (c) 1996-2001
All Rights Reserved.
>ps0
   Instrument S/N:  0
        Frequency:  307200 HZ
    Configuration:  4 BEAM, JANUS
      Match Layer:  10
       Beam Angle:  20 DEGREES
     Beam Pattern:  CONVEX
      Orientation:  DOWN
        Sensor(s):  HEADING  TILT 1  TILT 2  TEMPERATURE
 Temp Sens Offset:  0.00 degrees C

     CPU Firmware:  16.17 [0]
    Boot Code Ver:  Required:  1.13   Actual:  1.13
     DEMOD #1 Ver:  ad48, Type:  1f
     DEMOD #2 Ver:  ad48, Type:  1f
     PWRTIMG  Ver:  85d3, Type:   5

  Board Serial Number Data:
     34  00 00 02 67 AD 9A  09 CPU727-2000-00H  (first 16 values are the
CPU serial number)
     41  00 00 00 09 A0 C6  09 PIO727-3000-00B
     3B  00 00 00 62 72 DD  09 DSP727-2001-04D
     DF  00 00 00 0E 1A 46  09 REC727-1000-04A
>
```

2. Example of the contents of WHCPUSN.TXT (the capture of the output from *BBConv*)

```
52,0,0,2,103,173,154,9
52,0,0,2,103,173,154,9
52,0,0,2,103,173,154,9
52,0,0,2,103,173,154,9
52,0,0,2,103,173,154,9
....repeating
```

3. Converting the first row of values above from decimal to hexadecimal you will obtain

```
52,0,0,2,103,173,154,9 --> 34,0,0,2,67,AD,9A,9
```

4. The converted values match the first 16 digits of the CPU line in the PS0 output.