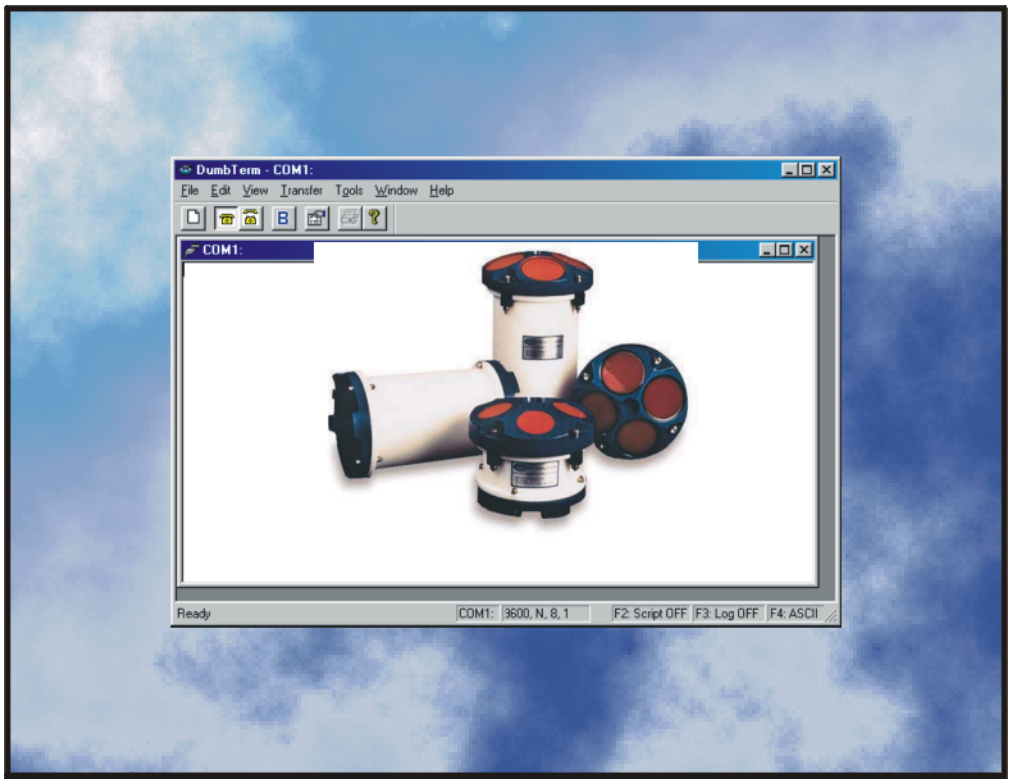
RDI Tools
User's Guide



P/N 957-6157-00 (January 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 *DumbTerm* to communicate with the ADCP. The utility software DOS programs have been provided to supplement features not yet implemented into the Windows environment.

1.1 System Requirements

RDI Tools requires the following:

- Windows 95®, Windows 98®, or Windows NT 4.0® with Service Pack 4 installed
- 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 ***ADCP*** 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 **DumbTerm** shortcut will start *DumbTerm*. **BBLIST** will start the DOS *BBLIST* program.

NOTE. The default directory for DumbTerm installation is C:\program files\RD Instruments\RDI Tools. The utility software will be installed in C:\Program Files\RD Instruments\Utilities. The C_code and Utility shortcuts will display a text file explaining what is included with the utility software.

2 Using DumbTerm

DumbTerm is a dumb terminal emulator program. This Windows compatible program can capture raw data files and help troubleshoot configuration problems. You can use *DumbTerm* 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 DumbTerm Short-Cut Keys

Use the following keys to quickly start tasks.

Table 1: DumbTerm 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
Ctrl+N	New connection

2.2 Communication Parameters

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

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

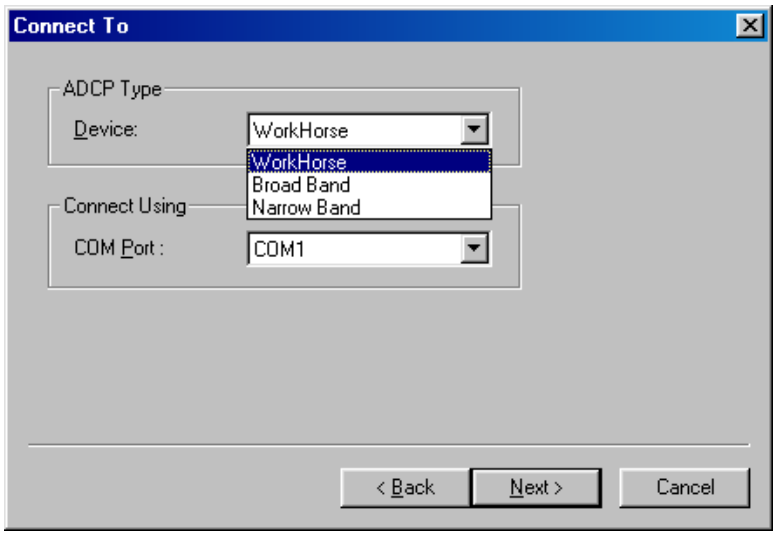


Figure 1. DumbTerm Connect To Screen

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

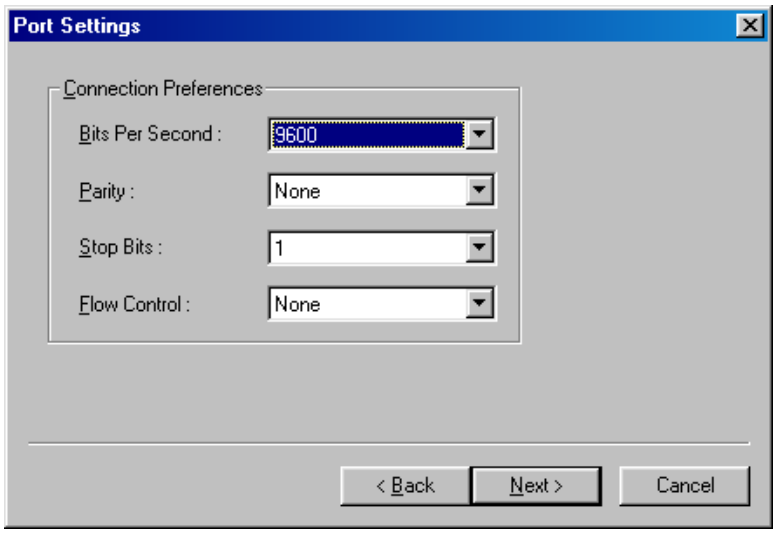


Figure 2. DumbTerm Port Settings Screen

c. On the **Options** screen, select the desired settings.
   - **Send Break On New Connection** – As soon as *DumbTerm* starts, it will send the ADCP a Break.
   - **Connect to Last Open Port On startup** – *DumbTerm* 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.

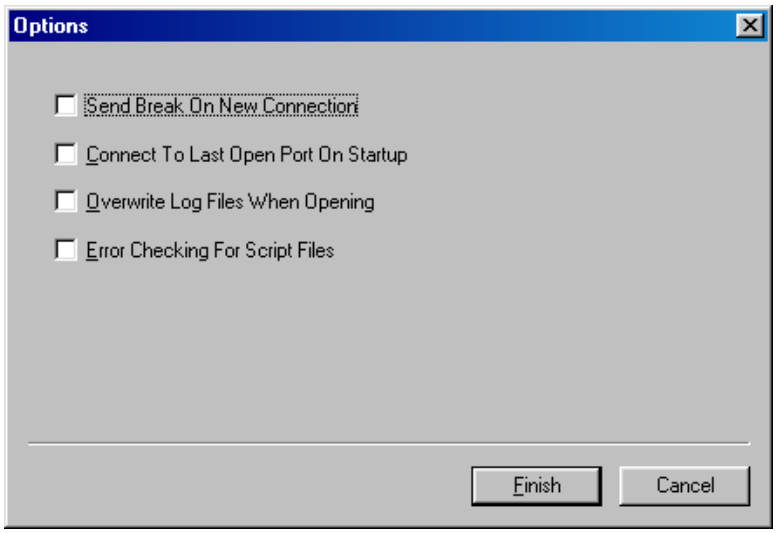


Figure 3. DumbTerm Options Screen

NOTE. To return to this screen while running *DumbTerm*, 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 *DumbTerm* 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.

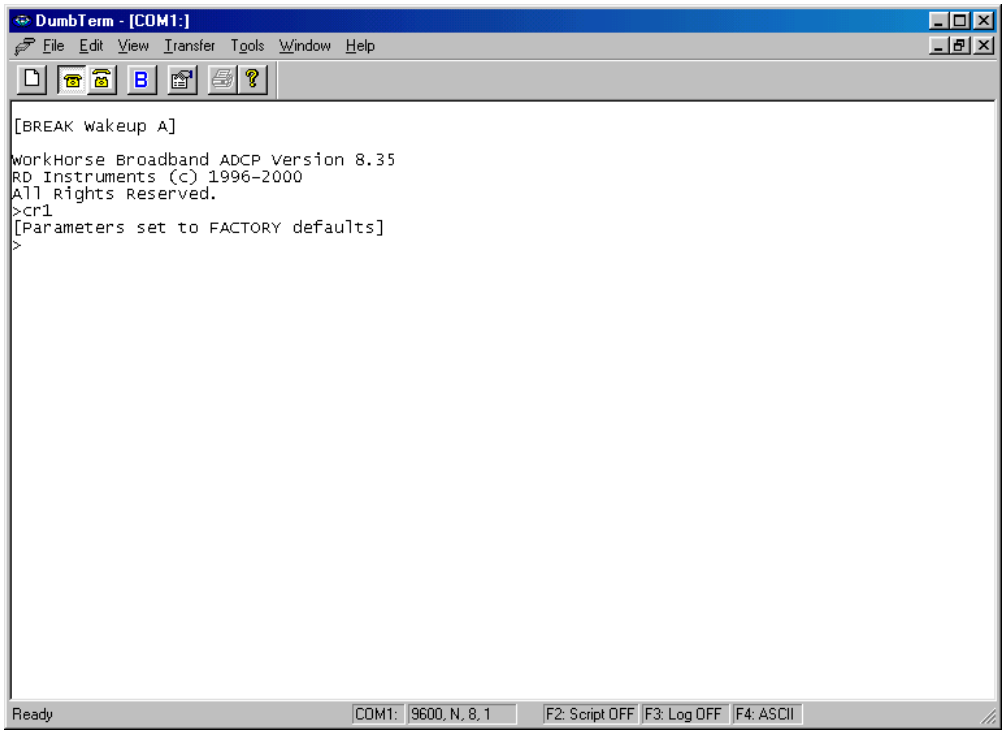


Figure 4. Sending a Command to the ADCP Using DumbTerm

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**.

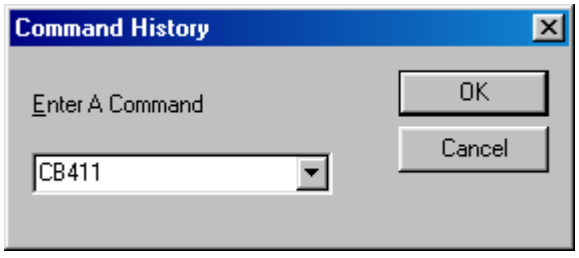


Figure 5. DumbTerm Command History Screen

2.4 DumbTerm Script Files

You can control what *DumbTerm* sends and captures by writing *DumbTerm* 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, *DumbTerm* script files can contain embedded commands that control the behavior of *DumbTerm* 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).

2.4.1 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.

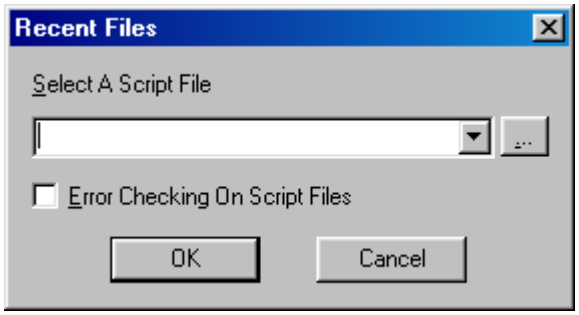


Figure 6. Selecting a Script File to Run

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

2.4.2 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 *.txt extension when saving the file.

b. To send a command to the ADCP, use one command per line. *DumbTerm* 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. *DumbTerm* will ignore all comments.

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

2.4.3 Script Commands

The following are legal script commands:

Table 2: DumbTerm Script Commands

Command	Description
$B	Sends a <BREAK>
$COMx:bbbbb,p,d,s	Sets up DumbTerm communication parameters:
	x = com port number (1 through 4)
	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
$L	Toggle logging. If 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.
$Gscript.ext	Chains scripting to “script.ext” (remainder of current script file is ignored.) Settings of current script file are retained.
$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
$T	Display PC time on screen (save to file if logging (e.g. {13:34:23})

2.4.4 Example Script File

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

```
;------------------------------------------------------------------------
; TESTWH.TXT
;                Script file for testing RD Instruments WorkHorse
;                         ADCP with the DUMBTERM program
;         Copyright (C) 2000 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.TXT"
; 17 April 2000
; Modified by RDI on:
; none
$LWH_RSLTS.TXT
;========================================================================
; 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_RSLTS.TXT.   A file called WH_RSLTS.TXT with the results of
$P this test will be created in the same directory as the DUMBTERM 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.
$D20
$P
$P
$P
$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.
$D10
$P
$P=======================================================================
$P
$P Sending a break to Wake Up the System
$B
$P=======================================================================
$P
$P Restoring factory defaults into temporary memory for TEST.
$P
```

```
CR1
$P=========================================================================
$P
$P Collecting system specific data.
TS?
PS0
PS3
$P=========================================================================
$P
$P Starting the Automated Tests.
PA
$P
$P All of the above tests should have passed. Review the file WH_RSLTS.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.
$D15
$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 25 seconds before continuing.
$D25
$P
$P=========================================================================
$P
$P Sending a break to Wake Up the System
$B
$P=========================================================================
$P
PC2
$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_RSLTS.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.
$D10
$P
$P=========================================================================
$P
$P PC1 -- Beam Continuity Test
$P The following test will confirm that each of the beams on your
```

```
$P transducer are 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.
$D25
$P=============================================================================
$P
$P Sending a break to Wake Up the System
$B
$P=============================================================================
$P
PC1
$P
$P The Beam Continuity test is complete.  Each of the beams should have
$P passed.  Review the file WH_RSLTS.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.
$LWH_RSLTS.TXT
```

2.4.5 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 *DumbTerm* 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: DumbTerm Test Script Files

Script File Name	ADCP Type	Results Saved to
BBtest.txt	Broadband	BB_RSLTS.txt
OStest.txt	Ocean Surveyor	OS_RSLTS.txt
WHtest.txt	Workhorse	WH_RSLTS.txt

f. Follow the prompts on the screen.

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

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

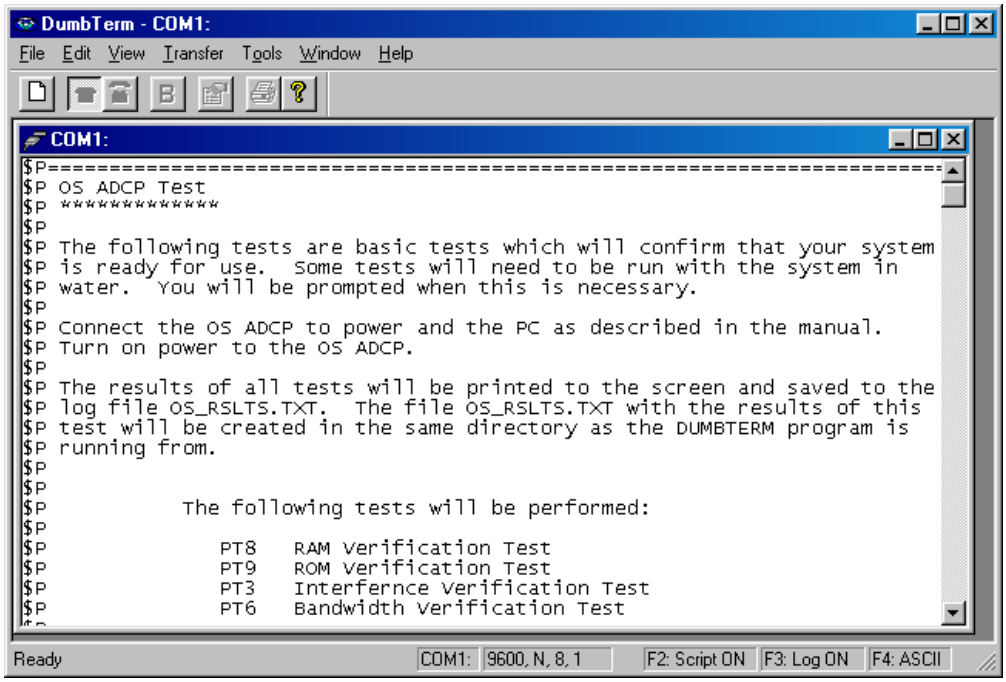


Figure 7. Running a Script File to Test the ADCP

2.5 DumbTerm 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 Utility Software

The following DOS programs have been provided to supplement features not yet implemented into the Windows environment. RDI will incorporate these features in future releases. These programs have been installed to the directory C:\Program Files\Rd Instruments\Utilities.

3.1 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.

3.1.1 Starting BBLIST

To start *BBLIST*, do the following:

- **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.

3.1.2 BBLIST Menus

BBLIST has four main menus (Figure 8) 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.

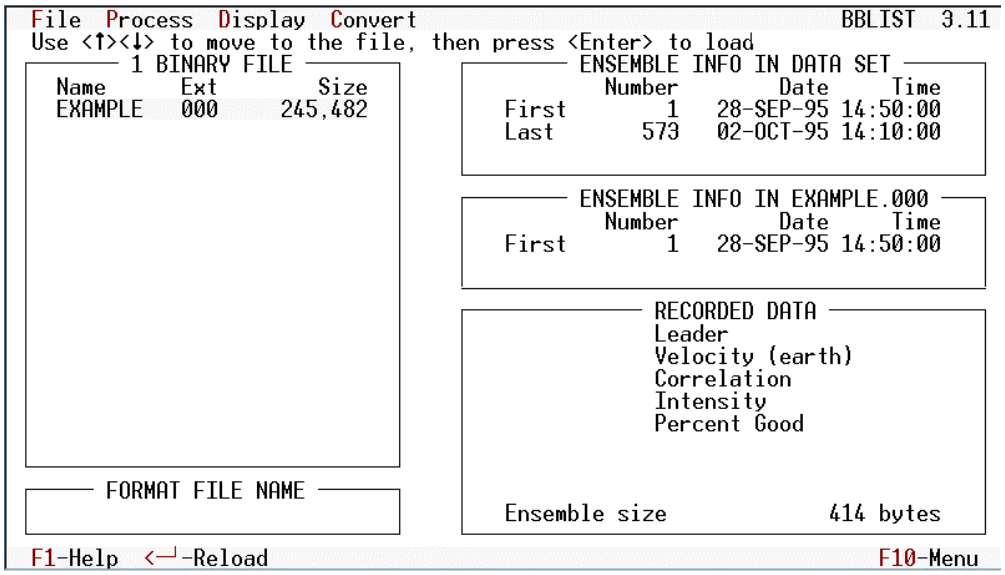


Figure 8. BBLIST Display

3.1.3 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.

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

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

Set the processing parameters. Use the **Process** menu (Figure 9, 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).

Set the conversion parameters. Use the **Convert**, **Limits** menu (Figure 10, 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.

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.

a. 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 11, page 17). If you make a mistake, select delete block and re-enter your choice.

b. Select **Ensemble Info** and press **<end>**. Select **Number** and press **<End>**.

c. Move the cursor to the **Format End** marker and press **<End>**.

d. Select **Profiles**, **Mag and Dir**. Select **M** for magnitude and press **<End>**.

e. Move the cursor to **Line End** and press **<End>**.

f. Select **Profiles**, **Mag and Dir**. Select **D** for direction and press **<End>**.

g. Move the cursor to **Line End** and press **<End>**.

h. Select **Profiles**, **Depth Ref**. Select **R** for range and press **<End>**. Your display should look like Figure 12, page 18.

View the format. Press **F9** to view the format before converting (Figure 13, page 18). Use the + and - keys to increase the ensemble number. Press **F9** again to return to the define format screen.

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

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.

3.1.4 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.

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

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

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

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.

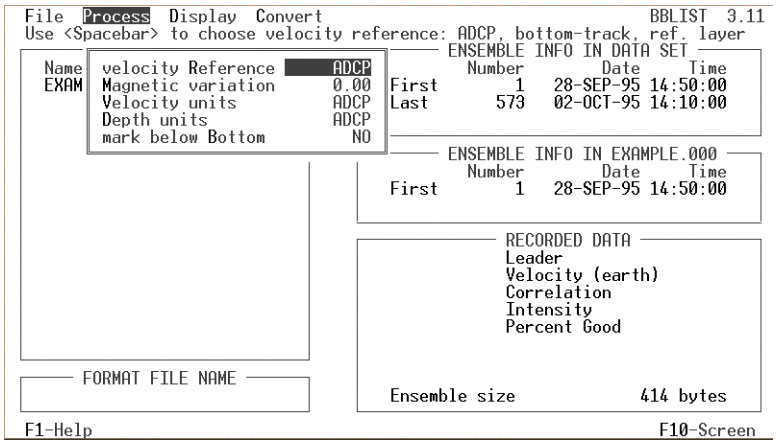


Figure 9. Set the Processing Parameters

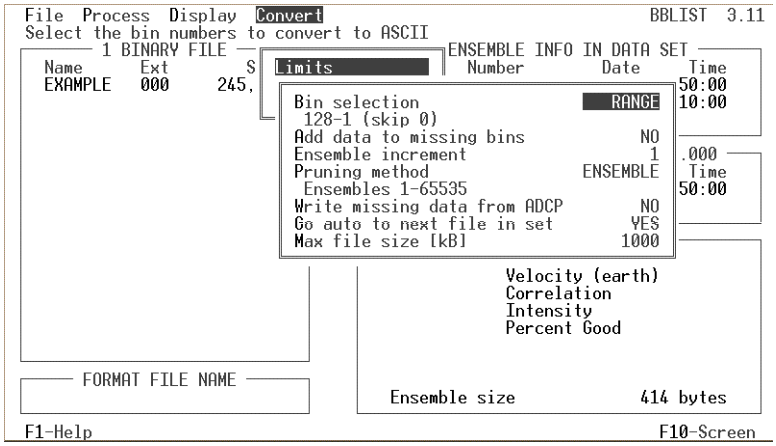


Figure 10. Set the Conversion Limitations and Parameters

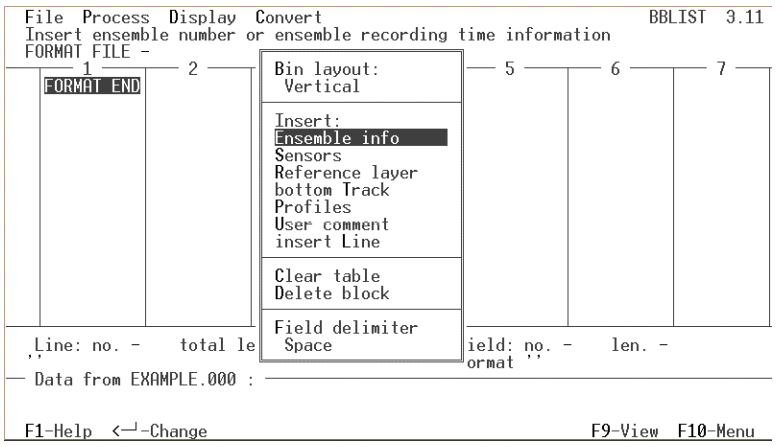


Figure 11. Define Format Selection Menu

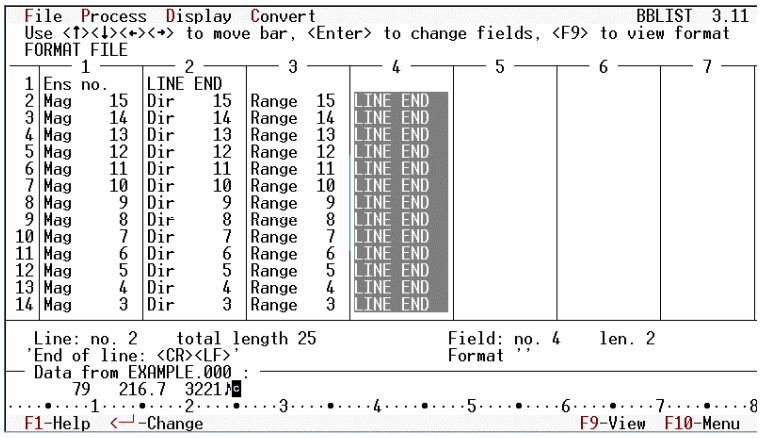


Figure 12. Defining the Format

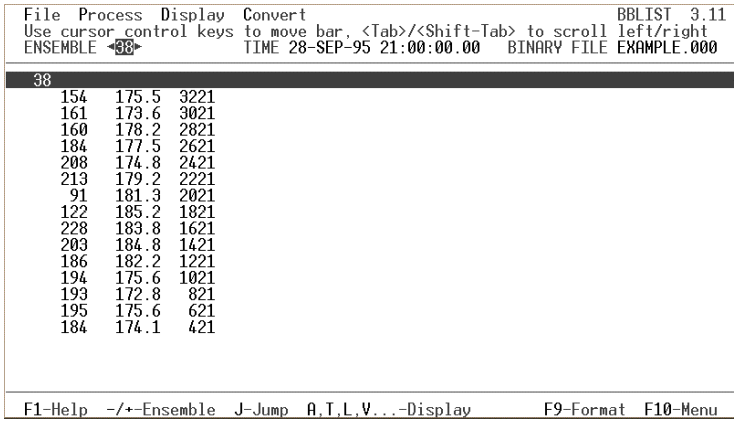


Figure 13. View the Format before Converting

3.1.5 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
-------------
```

3.2 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.

3.3 Using BBSUB

BBSUB is a raw data file subsectioning utility. *BBSUB* starts copying ensembles from “infilename” to “outfilename” starting with the ensemble whose number is “start.” If “start” is specified to zero, *BBSUB* starts copying from the first read ensemble. *BBSUB* will continue copying ensembles until it reaches ensemble number “end.” If “end” is not specified, it copies until the end of the file.

Syntax: BBSUB infilename outfilename start [end]

Where:

infilename	Filename of BroadBand ADCP raw data.
outfilename	Filename of subsectioned file. If this filename contains an extension, only one file will be used for output. If no extension is stated, then sequential extensions will be used (.000, .001, etc.) with each file being about 1 MB long.
start	Starting ensemble number.
end	Ending ensemble number (optional).

Sample BBSUB display:

```
C: \RDI BBSUB data.001 NEW 5 15
BBSUB Version 1.22
Copyright (c) 1994-1995 by RD Instruments.
<ESC> aborts processing
Processing C:\RDI\DATA.001 to NEW.000

Ens #10 Subsectioning ... COMPLETE
BBSUB terminated normally.
```

3.4 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. A decoder file simply contains interpreted instructions for converting to ASCII.

Command syntax: BBCONV decfilename infilename [> outfilename]

Where:

decfilename - Decoder File Name
infilename - Filename of BroadBand ADCP raw data
outfilename - Name of ASCII file to output

A typical decoder file may contain the following:

```
;  Bottom Track Data Decoder File
;  The following indicated SIGNED 2 byte numbers (note the -2 for type)
;
0600,25,-2     ; Beam #1/East/Stbd BT Vel (mm/s) 25th byte of structure 0600h
0600,27,-2     ; Beam #2/North/Fwd BT Vel (mm/s) 27th byte of structure 0600h
```

```
$L
;  This begins the next line of comma delimited data.
0080,3,2     ; Ensemble Number:  3rd byte of structure 0080, unsigned 2 bytes
```

- Lines beginning with a semicolon (;) are ignored and are for user comments.
- In addition, decoder lines may be followed by semi-colons for 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 where:

id — Data Structure identifier in hex (e.g. 0600). 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.

Each ensemble processed will show a dot (.) on the screen. These will not appear in the output file if the output is routed to a file. 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.

3.5 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 called “outfilename.”

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

Syntax: BBMERGE decfilename asciifilename infilename outfilename

Where:

decfilename	Decoder File Name
asciifilename	File name of ASCII text file to merge
infilename	Filename of BroadBand ADCP raw data
outfilename	Name of ASCII file to output

3.6 Example Using BBCONV and BBMERGE

RDI provides example DEC files that help remove user-selected data from binary data files and stores the information into ASCII comma delimited format. For more information on how to use these files, read the included TXT files (default installation directory is C:\Program Files\RD Instruments\Utilities).

Table 4: DEC Files Included with RDI Tools CD

Description	Instructions	DEC File Name
ASCII Decoder file for only the ADC channel data	ADC.TXT	ADC.DEC
ASCII Decoder file for only the Salinity data	SALIN.TXT	SALIN.DEC
ASCII Decoder file for only the Transducer Depth data	DEPTH.TXT	DEPTH.DEC
ASCII Decoder file for only the Heading data	HEADING.TXT	H.DEC
ASCII Decoder file for only the Ensemble Number data	ENSEMBLE.TXT	ENSNO.DEC
ASCII Decoder file for only the Orientation data	UPDWN.TXT	UPDWN.DEC
ASCII Decoder file for only the Number good pings (25 bins only)	SUMSQR.TXT	NGOOD.DEC
ASCII Decoder file for only the sum of (velocity^2) (25 bins only)	SUMSQR.TXT	SUMSQR.DEC
ASCII Decoder file for only the velocity sum (25 bins only)	SUMSQR.TXT	SUMVEL.DEC
ASCII Decoder file for removing NMEA data from files created by VMDAS	NMEA.TXT	NMEA.DEC
ASCII Decoder file for removing Date and Time from Ensemble Data	TIME.TXT	TIME.DEC

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.

Up/Down Example

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 para-

graphs 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. Use the following example of what to enter at the DOS command line. For this example, we will assume that your raw data file name is TEST.000.

   BBCONV UPDWN.DEC TEST.000 >UPDWN.DAT

   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. To do this, use the following example of what to enter at the DOS command line.

   BBMERGE UPDOWN.DEC UPDWN.DAT TEST.000 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 orientation will be correct. If you collected data in beam coordinates, the velocity data will now have the correct signs.

3.7 Using CHECKDAT

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

CHECKDAT 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

Usage: CHECKDAT [drive][path]DataFile [/FIRST#] [/LAST#]

Where:

drive	= Drive containing command file [optional]
path	= DOS directory path to command file [optional]
DataFile	= Raw data file name (wildcards accepted) [Required]
/FIRST#	= Set the first ensemble to check to # [optional]
/LAST#	= Set the last ensemble to check to # [optional]
/?	= Displays a brief help screen

Example 1: To check the raw data file demo011r.000, start *CHECKDAT* by typing **CHECKDAT demo011r.000**.

You should see a message similar to the following.

```
D:\RDI>CHECKDAT demo011r.000

CHECKDAT checking demo011r.000 . . . press ESC to pause

NO ERRORS FOUND:  demo011r.000

DATA FILE INFORMATION: demo011r.000

DATA TYPES RECORDED (Length information in bytes)
ID# (hex)   Offset        Length      Name
0               18            36      Fixed leader
80              54            38      Variable leader
100             92           322      Velocity
200            414           162      Correlation
300            576           162      Intensity
600            738            72      Bottom track
Ensemble size 812 bytes
```

```
CHECKDAT done.
D:\RDI>
```

Example 2: **CHECKDAT ABCDE000.001>REPORT.TXT**. Same as above, but the results are placed in file REPORT.TXT.

3.8 Speed of Sound Calculator

SS allows you to quickly calculate the speed of sound in the water. This program can be operated through a DOS Window. Enter values for the water temperature, salinity, and depth. The Sound Speed and Coefficient will be calculated based on the parameters entered.

3.9 Surface

Surface is an executable program that cannot be operated through the Windows environment. Surface estimates the range from the ADCP to the water surface or bottom from the echo intensity data. This program does not change the original data. It creates a text file with the estimated ranges. This program is intended for customers to estimate where to cut off their data.

3.10 C++ Library Code

The C++ Code has been provided to help you in the creation of your own programs. These files are provided as is and in general are not supported. Use at your own discretion. The files are located in the directory: C:\Program Files\Rd Instruments\Utilities\C_Code.

NOTES