A System for Shipboard Analysis of Buoy Data

by

Kenneth E. Prada

November 1992

Technical Report

Funding was provided by the Office of Naval Research under Grant No. N000-14-90-J-1490.

Approved for public release; distribution unlimited.
A System for Shipboard Analysis of Buoy Data

by

Kenneth E. Prada

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

November 1992

Technical Report

Funding was provided by the Office of Naval Research
under Grant No. N000-14-90-J-1490.

Reproduction in whole or in part is permitted for any purpose of the United States Government. This report should be cited as Woods Hole Oceanog. Inst. Tech. Rept., WHOI-92-42.

Approved for public release; distribution unlimited.

Approved for Distribution:

George V. Frisk, Chairman
Department of Applied Ocean Physics
and Engineering
6.1 Calibration Table Generation ............................................. 14
6.2 Plot Parameter File Handling ............................................ 14
6.3 Environment Variable Handling ........................................ 14
6.4 Notebook Functions ..................................................... 15

7 Results and Future Work ................................................ 15

8 Acknowledgements ........................................................ 15

A Menu Example .................................................................... 17

B Script, Program and Information Files .................................. 19

C Plot Examples ..................................................................... 21

List of Figures

1 Processing System Block Diagram ....................................... 5
2 Flow Diagram from Instrument to Data Pool ....................... 6
3 Example of X-Windows Menu Display ................................... 7
4 Example of Terminal Menu Display ...................................... 8
5 Processing System Flow Diagram ........................................ 11
6 IMET Plot of Seven Variables for One Day ......................... 22
7 IMET Plot of Seven Variables for Three Days ..................... 23
8 VAWR Plot of Seven Variables for Three Days .................... 24
9 VAWR Plot of Seven Variables for One Month .................... 25
10 VMCM Plot of Three Variables for Three Months .............. 26
11 TPOD Plot of One Variable for Three Months ................. 27
12 VAWR/IMET Overplot of Five Variables for Three Days .... 28
13 Overplot of Air and Sea Temperature from Four Instruments 29

List of Tables

1 System Supported Plot Devices .......................................... 9
2 Some Typical System Environment Variables ..................... 10
ABSTRACT

This report describes a software system for shipboard processing of buoy data. These data are derived from a variety of instruments installed on buoys recovered during turn-around cruises for the SUBDUCTION experiment. The software package, which operates on a network of PCs and SUN workstations, provides an interactive and versatile set of tools for handling these data sets.

The data are extracted from the native instrument storage media and format, converted to a standard storage format in calibrated scientific units. The resulting files are then processed and displayed for evaluation and analysis. These steps occur within a short period of time following buoy recovery.

1 Introduction

The Upper Ocean Processes group of the Department of Physical Oceanography uses a large suite of functions for processing and analysis of buoy data. Typically these functions are performed on the Institution's central computing facilities. These functions are a group of programs created, modified, and refined over the course of many years by many individuals. Their principal advantages involve long-term familiarity and expectation of results. However, their disadvantages include a lack of versatility and adaptation, and a dependence on the central facilities architecture that does not support at-sea use.

The SUBDUCTION project needed the ability to perform processing and analysis of buoy data during recovery and turn-around cruises. The project includes up to five moorings with several eight-month deployments. Data from buoy instruments must be examined during the recovery and re-deployment cruises. Evaluations of these data would determine data integrity and indicate which instruments might be re-deployed after servicing.

Each mooring can contain up to twenty instruments with some instruments using more than eight sensors. For deployments lasting eight months a large volume of data is collected. A substantial portion of these data must be processed, displayed and evaluated in the period between mooring sites.

To support these needs, the buoy processing functions have been collected, revised, expanded, and unified into a comprehensive data processing, display, and analysis system. The system handles data from a large instrument suite, including the VAWR, VMCM, IMET module systems, and Branca temperature pods. Other instruments, such as the XBT and ADCP, may be added in future configurations.

The system design goals are numerous and include:

- reduce dependence on VAX/VMS systems
- implement UNIX to aid long term platform portability
- implement standards for data storage formats
- implement standards for display and print graphics
- support both shipboard and shore-based laboratory needs
- simplify operations
- create and/or improve easy user interaction
- improve annotation and reporting
- increase productivity
This report describes the processing software system written to support shipboard handling, analysis, and display of buoy data from the SUBDUCTION experiment.

2 Data Format - UNIDATA netCDF

Critical to any processing or analysis resource is the storage format. Data from the many buoy instruments exists in diverse forms. These forms need to be converted into a unified standard format for effective development of a data handling system. The netCDF format was selected.

NetCDF, the Unidata network Common Data Form [Rew & Davis, 1990A], [Rew & Davis, 1990B], is an interface for scientific data access and a freely distributed software library that provides an implementation of the interface. The netCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data.

NetCDF data sets have several important features:

- **self-describing** - a netCDF file includes information about the data it contains.
- **network-transparent** - data is represented in a form that can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- **directly accessible** - any subset of the data can be accessed without reading over preceding data.
- **extendible** - data can be appended to a netCDF file without copying the prior data.

The netCDF support library and functions offer substantial advantages. The most important of these is platform portability. Most of the popular computing platforms and operating systems are supported by either Unidata or the large number of netCDF users.

NetCDF is useful for supporting access to diverse kinds of scientific data in heterogeneous networking environments and for writing application software that does not depend on application-specific formats.

NetCDF was developed at the Unidata Program Center. It is part of a data processing software development sponsored by the National Science Foundation's Division of Atmospheric Sciences. The program is managed by the University Corporation for Atmospheric Research, which maintains and distributes the netCDF software. It is freely available using ftp procedures on Internet. Current use involves broad ranging scientific applications and disciplines, both national and international.

Some of the important Unidata and netCDF advantages include:

- well documented [Rew 1991]
- multi-dimensional data sets (usually not available from database managers)
- FORTRAN and C language interface.
- UNIX, VMS, MSDOS, and OS/2 operating systems support.
3 Shipboard Data Handling

During the SUBDUCTION turn-around cruises, a specialized data handling and processing system was installed aboard ship. Figure 1 shows a diagram of that system. It contained two SUN IPC workstations and two DOS machines interconnected with a localized ethernet. The DOS machines are used to transfer the data from native instrument recording media or format to the file system on the processing computers.

Shipboard processing involved data from four instrument types, an IMET instrument suite, a Vector Averaging Wind Recorder (VAWR), a Vector Measuring Current Meter (VMCM), and Brancker Temperature Pods (TPOD). In all cases the data must be transferred from the instrument to the processing system. Figure 2 shows a block diagram of the data flow from each instrument to the data pool on the processing system.

The details concerning the transfer operations for each instrument type are described in detail in later sections of this report.
4 System Structure

The system is a series of programs, shell scripts, and information files (see Appendix B). These items are combined in a unique way to allow control and execution of the complex functions that comprise the system processing and analysis. The structural elements of the system are:

1. the primary shell script \texttt{.uop}
2. menu programs \texttt{zuop} and \texttt{tuop}
3. control and interaction shell scripts
4. processing and analysis programs
5. information files

4.1 The Primary Shell Script

The primary shell script \texttt{.uop} is the top level control and execution element in the system structure. It interacts with the user through menu programs. The menu programs accept user requests and return functional names to the \texttt{.uop} script. The \texttt{.uop} script analyzes the functional names and executes either programs or scripts as needed. The programs or scripts return control back to the primary \texttt{.uop} script on completion.

4.2 The UOP Menu

The UOP menu is a dynamic object, defined in a text file \texttt{uop.mnu}. The file contains lines that define operations and name the programs that perform those operations. Lines may be organized within the file to create and name groups of similar or related functions. Appendix A contains a listing of a typical menu file.

The menu file consists of groups of functions. Each group starts with a line describing the group. Additional lines contain descriptions of each function and its terminal and X-window implementations.
As functions are added to the system, the menu file can be revised to accommodate changing needs. Groups can be added, re-organized or removed as needs dictate. Functions can be added to groups. The system uses the menu file as a dynamic user interface to the processing functions.

4.2.1 X-Windows Menu Program \textit{ruop}

A menu program \textit{ruop} supports the X-windows environment. This program provides a point-and-click interface to the system.

The menu definition file \textit{ruop.mnu} is used to define a window and its contents. The window contains groups of pushbuttons that describe the functions available in the system. Each group has a label button and function buttons. The label button (colored grey) contains the group name. The function buttons, all the same color within a group, contain descriptions of the functions they represent. Each group has different color buttons.

Group structure is organized for ease of viewing and comprehension. Groups are arranged in rows and columns. The number of rows and columns increase as the number of menu groups and items increase. The group with the largest number of functions determines the group size for all groups (some groups may have blank buttons). This organizes the groups so that all label buttons are aligned horizontally and groups begin and end uniformly at the top and bottom of the menu window. One button in the window is uniquely colored. This is the QUIT button. It is always colored white.

Figure 3 contains a diagram of a typical X-window system menu (without colors). This diagram should be used as a format example only, since the content may not agree with current menu files.

4.2.2 Generic Terminal Menu Program \textit{tuop}

A program \textit{tuop} exists for system operation with terminals other than X-windows types. The program supports termcap operations for most terminals.
Because of restricted screen size, the system menu is not displayed in its entirety on a single screen. Instead, the primary menu screen contains a list of the functional groups. The user can select a group number which then displays the contents of a that menu group. Figure 4 shows two examples of terminal menu screens.

4.3 Control and Interactive Shell Scripts

Shell scripts are used by the system to handle data flow, provide a batching environment, and to make the user interface simple and effective. For example, graphical output from most programs can be directed to any of several devices. The process would normally involve entering several command lines to accomplish the entire process. Instead, a shell script is interposed between the user and the several programs necessary to accomplish the task.

Information is gathered from the user by the script interactively. This might include instrument number, inclusive dates for processing, output device for graphics, or other information pertinent to the processing operation. This information is used by the script to set environment variables, call the proper programs, select output devices, and determine the order of processing steps.

4.4 Information Files

Several types of information files are used to enhance the processing flow. These files provide information to configure program operation thus eliminating the need for a user to enter this information at each execution instance of a program. There are also programs in the system suite specifically to create and edit these information files.
### Table 1: System Supported Plot Devices

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>SIZE</th>
<th>PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostScript Device</td>
<td>8.5&quot; X 11.0&quot; (A)</td>
<td>post11</td>
</tr>
<tr>
<td>PostScript Device</td>
<td>8.5&quot; X 14.0&quot; (LEGAL)</td>
<td>post14</td>
</tr>
<tr>
<td>Tektronix 4010/4014</td>
<td>11.0&quot; X 17.0&quot; (B)</td>
<td>xplot</td>
</tr>
<tr>
<td>HP 7475</td>
<td>8.5&quot; X 11.0&quot; (A)</td>
<td>hp7475b</td>
</tr>
<tr>
<td>HP 7470</td>
<td>8.5&quot; X 11.0&quot; (A)</td>
<td>hp7475a</td>
</tr>
<tr>
<td>HP DRAFTPRO (7570A)</td>
<td>17.0&quot; X 22.0&quot; (C)</td>
<td>draftLc</td>
</tr>
<tr>
<td>HP DRAFTPRO (7570A)</td>
<td>22.0&quot; X 34.0&quot; (D)</td>
<td>draftLd</td>
</tr>
<tr>
<td>ZETA (HPGL)</td>
<td>11.0&quot; X 17.0&quot; (B)</td>
<td>hp7475b</td>
</tr>
<tr>
<td>ZETA (HPGL)</td>
<td>8.5&quot; X 11.0&quot; (A)</td>
<td>hp7475a</td>
</tr>
</tbody>
</table>

4.4.1 Calibration Tables

Several instruments must have the recorded values converted into calibrated scientific units. The calibration factors are instrument and sensor specific. A table file for each instrument is created by a special program and contains calibration and scaling values for each sensor that the instrument contains. The values are derived from sensor calibration procedures performed prior to instrument use and deployment.

4.4.2 Plot Parameters

There is a wide range of visualization needs for each instrument and data type. These needs must be provided for without maintaining an immense suite of specifically tailored plot programs. They must also be available without requiring the user to repeatedly enter information and parameters with each execution of a program.

Plot parameter files are used to make the visualization process a simple one. Each plot parameter file contains information concerning the desired variable(s) and scaling for each type of plot operation. The visualization programs read these files. The files may be edited either with the programs provided within the system package or with any text editor.

4.5 The System Plot Package

System plotting uses a generic UNIX plot method. The system is simple and lacks frills, but it is easy to use and supports a large variety of output devices (see Table 1). Plot program output is in device independent form and is directed to the stdout device. This allows re-direction to files for intermediate storage or piping to conversion filters.

Typical plots are scaled with a 0.75:1.0 aspect ratio. This allows the use of standard paper sizes, which have approximately that ratio. Each of the device drivers scales the plot data to fit correctly with the individual device limits.
| VAWRSOURCE | VAWR data input directory path |
| VAWRDEST   | VAWR data output directory path |
| VAWRNOTES  | VAWR notebook directory path    |
| VAWRTABLE  | VAWR calibration table directory path |
| VAWRNUM    | VAWR instrument number for processing |
| IMETSOURCE | IMET data input directory path |
| IMETYEAR   | year for IMET processing start |
| IMETMONTH  | month for IMET processing start |
| IMETDAY    | day for IMET processing start |

Table 2: Some Typical System Environment Variables

4.5.1 PostScript Files

PostScript output from the system plot functions can be handled in several ways. Output may be directed to the laser printer in two sizes, standard letter size (8.5" X 11.0") or legal size (8.5" X 14.0"). Alternatively, PostScript files may be previewed on the screen using the SUN supplied pageview function or the GhostScript resource from the Free Software Foundation.

4.6 User Shell Environment

A user shell has an environment which consists of a number of variables (strings). These variables can convey pre-defined information to programs and scripts. An example is the environment variable HOME which is set by the system when the shell is initiated. HOME contains the name of the user's home directory (for example /usr/ken).

The system uses environment variables to simplify its operations. They are used to signify directory paths, instrument numbers, and other information. Repetitive queries to the user are eliminated when certain variables are set by the user. The programs use the information provided in the environment variable rather than query the user.

Table 2 defines some of the environment variables used by the system. There are menu functions included for the display, setting and editing of environment variables.

The names of the system environment variables can be found in files specific to each instrument type:

- imet.env  IMET variable names
- vawr.env  VAWR variable names
- vmcm.env  VMCM variable names
- tpod.env  TPOD variable names

These are text files and contain single line entries each of which contains the name of a variable. Additions and deletions can be made to accommodate system changes and growth. This can be done with a text editor.

1 Laser printer must be capable of and configured for legal size paper.
5 Processing Steps

There are several steps in the shipboard processing flow. Figure 5 shows an overall flow diagram for all instruments in the SUBDUCTION suite. The following sections describe the steps related to the individual instruments.

The shipboard system primarily performs format conversion, calibration and visualization operations. Processing of a more complex nature is not intended for the shipboard environment at this time. This report does not include information about these operations.

5.1 Extraction from Instrument

IMET files derived from SUBDUCTION buoys are in the original IMET format [Prada 1990]. They are typically contained on an NHance/ISI optical disk cartridge. There are 24 one-hour files per day. These can be directly transferred to the UNIX system.

VAWR and VMCM data is recovered from the instruments on a digital tape cassette. These cassettes must be processed on a DOS system. The program used to read a SEADATA cassette and create a DOS disk file is seadata. [Danforth 1990] This program reads the cassette and produces two files, a data file and a comment file. These files can be transferred to the UNIX system.

TPOD data is stored using internal instrument memory. A program operating on the DOS system extracts these data and creates an ASCII file with annotation and descriptive information. This file can be transferred to the UNIX system.
5.2 Transfer to the UNIX System

There are two transfer methods available. The first method involves the use of the PC-NFS protocol. PC-NFS is a systems protocol that mounts a disk directory on the SUN computer as though it were a disk actually resident on the DOS system. For example, the /pu/data directory on the UNIX system might be linked as the F: drive on the DOS system. When this method is used, transfer is accomplished by copying the desired files between the optical disk drive (typically E:) and the UNIX PC-NFS mounted drive (typically F:).

The alternative method for file transfer uses the ftp protocol. This method is more user intensive and would only be used if PC-NFS fails.

The IMET copy operation is done in multiple steps, since it is not possible to mount all the needed files on the DOS optical drive at the same time. There is a DOS procedure that creates a DOS script file to accomplish this task without operator intervention. The user runs a DOS program imetzfer that requests drive designations for the DOS optical drive and the PC-NFS drive along with start and end dates. The program then generates a DOS script file to mount and copy the required files in reasonably sized groups. This method can be implemented by typing uop at the DOS prompt. This executes a script which runs the imetzfer program and then executes the script file produced by imetzfer.

Copy procedures for VAWR, VMCM, and TPOD are much simpler. Since there are few files (typically two per instrument), they can be copied in a single step. These operations are typically carried out by a DOS script file.

5.3 Conversion to netCDF

The next step in processing for each data type is conversion to netCDF format. The IMET process request starts and end dates. One netCDF file is created for each day using 24 of the original IMET format one-hour files.

Since original IMET recording formats differ slightly from experiment to experiment, there are separate conversion programs for different experiments (some differences also exist between the deployments in SUBDUCTION). The conversion script handles the experiment differences through user queries.

File names for IMET data sets follow a chronological pattern. The original format files are named yymmd-dhh.met and the netCDF format files are named yymmdd.met.

VAWR and VMCM conversion to netCDF form is more complex. Since the binary files derived from the cassettes often contain data and clock errors, the conversion program must handle these issues. In most cases the error records are deleted from the resulting netCDF file. Transcripts of the error messages are displayed to the user and written to a log file (see subsequent section on notebook functions). At this stage the netCDF files only contain raw instrument counts and must undergo further processing before proper visualization can be performed.

TPOD data also is converted to raw format netCDF files. This conversion process is less demanding than the VAWR/VMCM process, since the instrument recording retrieval methods are substantially more reliable.

File names for VAWR, VMCM or TPOD data conform to a prescribed standard. That standard is: vawrnnnn, vmcmnnnn, or tpodnnnn, where nnnn is a four-digit integer representing the instrument number. This format is used throughout the system.

\footnote{2pu11 is a SUN-IPC machine network name for one of the shipboard systems}
5.4 Conversion to Calibrated Form

Data values from IMET instruments are already in calibrated scientific units. One calibration step is performed that corrects for magnetic compass variation at the site. This variation affects IMET wind values.

The data values acquired from the VAWR, VMCM, and TPOD instruments are not in calibrated scientific units. There are procedures for each data type to convert from raw to calibrated form. These functions read calibration information from files specific to each individual instrument and convert the raw counts to scientific units. The results are stored in second stage netCDF files that contain both the derived scientific values and the calibration information used for the derivations.

5.5 Visualization Functions

Visualization functions involve the display of data values in graphic form to one of several devices. Typically a series of plots will be displayed to the console screen. This can be an iterative process until the data, scaling, and annotation are acceptable. Then the resulting plot is output to a hard copy device. This is typically a PostScript device but may also be a plotter.

Appendix C contains several example plots from the SUBDUCTION instrument suite. Refer to these examples for the following sections.

5.5.1 IMET Plots

There are two choices for plotting IMET data. The first produces sheets or screen displays that contain a single day of measurements using the one-minute sample interval typical for IMET (figure 6). The second method plots multiple days of IMET measurements on a single sheet and may include from several days to several months (figure 7). To match the lower sampling rate for comparison to the VAWR, the data may be averaged at up to 15 minute intervals. In each case the plot contains from one to twelve trace panels within a single display frame. Each panel is labeled with minimum and maximum values and the variable name.

IMET measurements may also be plotted with other instrument measurements in overplot functions described in subsequent sections.

5.5.2 VAWR and VMCM Plots

VAWR and VMCM plots are variable length time series. The plots can contain up to twelve trace panels within the display frame. A VAWR plot typically has seven or eight trace panels. A VMCM plot will usually have three trace panels. The contents of each panel is user selectable (by configuration file). Minimum and maximum y-axis values may be derived from user specification or auto scaling. Figures 8 and 9 show short and long duration VAWR plots. The short duration plot is for the same period as that shown in the IMET multiple day plot. Figure 10 shows a three month VMCM plot.

5.5.3 TPOD Plots

TPOD plots typically contain a single trace for temperature. Each plot is a time series and may range from several days to several months. Figure 11 is a three month TPOD plot.
5.5.4 Other Plot Functions

IMET/VAWR comparison plots are available. These plots contain several trace panels that display the same variable from each instrument for the identical period of time. The IMET data is usually averaged over the operational sample interval of the VAWR. The plots aid in evaluation of sensor and calibration differences during companion deployments of both instrument types. Figure 12 is an example of this plot type with five variables compared for a three day period.

Temperature overplots are also possible. This function allows multiple trace plotting in a single frame. Comparisons of air and sea temperatures from various instruments can be accomplished. The plot can handle up to twelve traces. Figure 13 shows a temperature overplot.

Comparison of various dis-similar measurements is possible with a generic overplot procedure. Similar to the temperature overplot, several traces are plotted in the same frame.

6 Other System Functions

The system has some basic housekeeping and maintenance functions. These provide:

- Calibration table generation
- Plot parameter creation and editing
- Environment Variable Handling
- Notebook functions

6.1 Calibration Table Generation

Each instrument that requires a data calibration step (VAWR, VMCM, and TPOD) have functions that create calibration information files for individual instruments. The functions are interactive, requesting value input from the user in response to specific queries. The table files thus created are used by the raw-to-calibrated conversion programs.

6.2 Plot Parameter File Handling

Visualization functions are controlled through the use of plot parameter files. These files contain lists of variable names and scaling values. The individual plot programs read these files and use the information to format the content and annotation of each plot frame.

6.3 Environment Variable Handling

Since environment variables are convenient to use, they must be readily available for change. There are functions contained in the system menu structure specifically for manipulating environment variables pertaining to each instrument type.
6.4 Notebook Functions

Record keeping throughout the handling and processing operations is important. The system contains functions that allow automatic or manual notebook entries.

Notebook files are created and allocated to individual instruments. Records pertaining to an instrument are recorded in the file assigned to that particular instrument. Notebook files are named with reference to the instrument. For example, VAWR number 707 would have a notebook file named vawr0707.log. The file would typically reside in the same directory as the data files for that instrument.

The notebook process generally begins early in the data handling sequence. For example, when a cassette from a VAWR or VMCM is read, a log file is created that contains comments from the operator and a record of errors during the read operation. When the data file is subsequently converted, calibrated and processed, the programs involved in these operations add entries to the log file. These entries may contain additional error information, program names, user names, and execution times.

The user can add entries to a log file at any time. These entries might contain any information pertinent to the data, its handling or display. The user may display or print the contents of any notebook file.

7 Results and Future Work

The system has been used aboard ship on two SUBDUCTION turn-around cruises and during the intervening time ashore. Production of quality control data plots during the cruise was effective and quick. These plots proved valuable in assessing individual instrument performance. It also provided an excellent evaluation of the entire data set for each mooring site and the whole experiment area.

The system needs further enhancements and additions. These should include a more generic approach to some of the features and functions, particularly visualization. Also, additional processing functions should be added to the package.

The success of this processing system has prompted a look at how similar features might be implemented for handling other oceanographic data types. A working group is considering standards and conventions for applying netCDF to a variety of data classes in oceanographic research. An Oceanographic Analysis Resource System (OARS) has been proposed, supported by a Basic Oceanographic Analysis Toolkit (BOAT). The objective will be to simplify the use and exchange of oceanographic data and provide generic tools for manipulation, processing, and visualization.

Other systems based on the netCDF data access method provide an incentive for further development of the concept. [Denbo 1991] [Wessel & Smith, 1991]

8 Acknowledgements

This work was supported by the Office of Naval Research under grant N000-14-90-J-1490. Maintenance and enhancement programming support has been provided by Nancy Galbraith, who with Nancy Penninton, gave valuable feedback and suggestions on operational and user interface details. Melora Park-Samelson assisted in coding most of the data calibration routines.
References


A Menu Example

This appendix contains an example of a menu file. This is an example of format only, so content may not agree with the current menu.

#IMET PROCESSING
CONVERT IMET to netCDF: _imetcdf: _imetcdf
PLOT IMET Data: _imetplt: _imetplt
IMET Plot Parameters: _imet_var: _imet_var
DISPLAY IMET Single Sheets: _imetshs: _imetshs
DISPLAY IMET Long Plots: _plotout: _plotout
IMET Environment: _imetenv: _imetenv
$

#VAWR PROCESSING
CONVERT VAWR to netCDF: _vawr_cdf: _vawr_cdf
CONVERT VAWR netCDF RAW to CAL: _vawr_cal: _vawr_cal
PLOT VAWR Data: _vawrplot: _vawrplot
VAWR Plot Parameters: _vawr_var: _vawr_var
DISPLAY VAWR Plot File: _plotout: _plotout
CREATE VAWR Instrument Table: _vawrtabl: _vawrtabl
VIEW/EDIT VAWR Table: _vawredit: _vawredit
VAWR Environment: _vawrenv: _vawrenv
$

#VMCM PROCESSING
CONVERT VMCM to netCDF: _vmcm_cdf: _vmcm_cdf
CONVERT VMCM netCDF RAW to CAL: _vmcm_cal: _vmcm_cal
PLOT VMCM Data: _vmcmplot: _vmcmplot
VMCM Plot Parameters: _vmcm_var: _vmcm_var
DISPLAY VMCM Plot File: _plotout: _plotout
CREATE VMCM Instrument Table: _vmcmtabl: _vmcmtabl
VIEW/EDIT VMCM Table: _vmcmedit: _vmcmedit
VMCM Environment: _vmcmenv: _vmcmenv
$

#TPOD Processing
CONVERT TPOD to netCDF: _tpod_cdf: _tpod_cdf
CONVERT TPOD netCDF RAW to CAL: _tpod_cal: _tpod_cal
PLOT TPOD Data: _tpodplot: _tpodplot
TPOD Plot Parameters: _tpod_var: _tpod_var
DISPLAY TPOD Plot File: _plotout: _plotout
CREATE TPOD Instrument Table: _tpodtabl: _tpodtabl
VIEW/EDIT TPOD Table: _tpodedit: _tpodedit
TPOD Environment: _tpodenv: _tpodenv
$

#netCDF Functions
DISPLAY netCDF File Parameters: _cdffshow: _cdffshow
DISPLAY netCDF File: _cdffdump: _cdffdump
EDIT netCDF file: _cdfedit: _cdfedit
PRINT netCDF File Parameters: _cdfflist: _cdfflist
CONVERT netCDF to ASCII: _cdfasc: _cdfasc
CONVERT netCDF to Binary: _cdfbin: _cdfbin
$ #OTHER PLOT FUNCTIONS
OVERPLOT Temperatures: \_tempplot: \_tempplot
OVERPLOT Generic Variables: \_overplot: \_overplot
SET/EDIT VAWR vs IMET Parameters: \_vaim\_var: \_vaim\_var
OVERPLOT VAWR vs IMET: \_vawr\_imet: \_vawr\_imet
DISPLAY Plot File: \_plotout: \_plotout
$
#NOTEBOOK FUNCTIONS
CREATE Notebook Entries: \_makenotes: \_makenotes
DISPLAY Notebook: \_shownotes: \_shownotes
PRINT Notebook: \_listnotes: \_listnotes
$
#MISCELLANEOUS FUNCTIONS
QUIT - EXIT - FINIS: \texttt{quit: quit}
OPEN Shell Window: \_dounix: \_dounix
SET Viewer Name: \_setview: \_setview
SET Editor Name: \_setedit: \_setedit
VIEW A File: \_view: \_view
EDIT A File: \_edit: \_edit
OPEN UNIGRAPH Window: \_unigraph: \_unigraph
$
$
B  Script, Program and Information Files

This appendix contains lists of the system shell scripts and programs along with their functions.

<table>
<thead>
<tr>
<th>SCRIPT NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>_cdflist</td>
<td>list netCDF file parameters to print device</td>
</tr>
<tr>
<td>_dounI</td>
<td>start a shell script</td>
</tr>
<tr>
<td>_edit</td>
<td>execute the assigned text editor</td>
</tr>
<tr>
<td>_imetcdf</td>
<td>IMET to netCDF conversion for multiple days</td>
</tr>
<tr>
<td>_imetenv</td>
<td>set, edit, delete IMET environment variables</td>
</tr>
<tr>
<td>_imetplt</td>
<td>IMET plot control, all variations</td>
</tr>
<tr>
<td>_imetshs</td>
<td>display control for IMET plots for multiple days</td>
</tr>
<tr>
<td>_listnotes</td>
<td>list notebook to print device</td>
</tr>
<tr>
<td>_makenotes</td>
<td>make notebook entries</td>
</tr>
<tr>
<td>_overplot</td>
<td>overplot control</td>
</tr>
<tr>
<td>_plotout</td>
<td>output control of plot files</td>
</tr>
<tr>
<td>_print</td>
<td>execute assigned file print process</td>
</tr>
<tr>
<td>_setedit</td>
<td>assign text editor process</td>
</tr>
<tr>
<td>_setview</td>
<td>assign text file viewer</td>
</tr>
<tr>
<td>_shownotes</td>
<td>screen display notebook entries</td>
</tr>
<tr>
<td>_tempplot</td>
<td>temperature plot generation control</td>
</tr>
<tr>
<td>_tpodcdf</td>
<td>TPOD to netCDF conversion control</td>
</tr>
<tr>
<td>_tpodedit</td>
<td>edit TPOD table file</td>
</tr>
<tr>
<td>_tpodenv</td>
<td>set, edit, delete TPOD environment variables</td>
</tr>
<tr>
<td>_tpodplot</td>
<td>TPOD plot control</td>
</tr>
<tr>
<td>_unigraph</td>
<td>setup and execute Unigraph process</td>
</tr>
<tr>
<td>_uop</td>
<td>primary UOP control script</td>
</tr>
<tr>
<td>_vawredit</td>
<td>edit VAWR table files</td>
</tr>
<tr>
<td>_vawrenv</td>
<td>set, edit, delete VAWR environment variables</td>
</tr>
<tr>
<td>_vawrimet</td>
<td>VAWR/IMET overplot control</td>
</tr>
<tr>
<td>_vawrplot</td>
<td>VAWR plot control</td>
</tr>
<tr>
<td>_vawrraw</td>
<td>VAWR raw data plot control</td>
</tr>
<tr>
<td>_view</td>
<td>execute assigned file view process</td>
</tr>
<tr>
<td>_vmcmedit</td>
<td>edit VMCM table files</td>
</tr>
<tr>
<td>_vmcmenv</td>
<td>set, edit, delete VMCM environment variables</td>
</tr>
<tr>
<td>_vmcmplot</td>
<td>VMCM plot control</td>
</tr>
<tr>
<td>_vmcmraw</td>
<td>VMCM raw data plot control</td>
</tr>
<tr>
<td>cdf.asc</td>
<td>convert netCDF files to ASCII</td>
</tr>
<tr>
<td>cdf.bin</td>
<td>convert netCDF files to binary</td>
</tr>
<tr>
<td>cdf.dump</td>
<td>screen display netCDF file records</td>
</tr>
<tr>
<td>cdf.edit</td>
<td>edit netCDF file records</td>
</tr>
<tr>
<td>cdf.list</td>
<td>list cdf file records to print device</td>
</tr>
<tr>
<td>cdf.scan</td>
<td>scan netCDF file for minimum and maximum values</td>
</tr>
<tr>
<td>cdf.show</td>
<td>display netCDF file header and parameters</td>
</tr>
<tr>
<td>disp.env</td>
<td>display environment names from selected file</td>
</tr>
<tr>
<td>echo2</td>
<td>output to stderr for script</td>
</tr>
<tr>
<td>gotoxy</td>
<td>screen cursor control from script</td>
</tr>
<tr>
<td>PROGRAM NAME</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>imet_env</td>
<td>IMET environment variable handling</td>
</tr>
<tr>
<td>imet.log</td>
<td>IMET notebook entries</td>
</tr>
<tr>
<td>imet.long</td>
<td>IMET long plot function</td>
</tr>
<tr>
<td>imet_name</td>
<td>create IMET file name from environment variables</td>
</tr>
<tr>
<td>imet.list</td>
<td>display contents of IMET notebook</td>
</tr>
<tr>
<td>imet.notes</td>
<td>generate a one-day IMET plot</td>
</tr>
<tr>
<td>imet.plot</td>
<td>edit imet plot variables file</td>
</tr>
<tr>
<td>next_day</td>
<td>generate environment variables for next day</td>
</tr>
<tr>
<td>over.plot</td>
<td>generic overplot function</td>
</tr>
<tr>
<td>prev_day</td>
<td>generate environment variable for previous day</td>
</tr>
<tr>
<td>subd1.cdf</td>
<td>convert SUBDUCTION 1 data to netCDF</td>
</tr>
<tr>
<td>subd2.cdf</td>
<td>convert SUBDUCTION 2 data to netCDF</td>
</tr>
<tr>
<td>temp.plot</td>
<td>plot temperatures from various instruments</td>
</tr>
<tr>
<td>title</td>
<td>display a title line on the console screen</td>
</tr>
<tr>
<td>tpod_cal</td>
<td>convert TPOD raw data to calibrated netCDF</td>
</tr>
<tr>
<td>tpod.cdf</td>
<td>convert TPOD data to netCDF</td>
</tr>
<tr>
<td>tpod.env</td>
<td>manipulate TPOD environment variables</td>
</tr>
<tr>
<td>tpod.log</td>
<td>create TPOD notebook entries</td>
</tr>
<tr>
<td>tpod.list</td>
<td>print TPOD notebook entries</td>
</tr>
<tr>
<td>tpod_notes</td>
<td>display TPOD notebook entries</td>
</tr>
<tr>
<td>tpod_plot</td>
<td>plot TPOD data</td>
</tr>
<tr>
<td>tpod.tabl</td>
<td>generate table file for TPOD data</td>
</tr>
<tr>
<td>tpod_var</td>
<td>edit TPOD plot variables</td>
</tr>
<tr>
<td>tuop</td>
<td>terminal menu program</td>
</tr>
<tr>
<td>uopxplot</td>
<td>X-windows plot output with mouse interaction</td>
</tr>
<tr>
<td>vaim_var</td>
<td>edit VAWR/IMET plot variables</td>
</tr>
<tr>
<td>vawr.cal</td>
<td>convert raw VAWR to calibrated</td>
</tr>
<tr>
<td>vawr.cdf</td>
<td>convert VAWR data to netCDF</td>
</tr>
<tr>
<td>vawr.env</td>
<td>manipulate VAWR environment variables</td>
</tr>
<tr>
<td>vawr.imet</td>
<td>generate VAWR/IMET comparison plots</td>
</tr>
<tr>
<td>vawr.log</td>
<td>create VAWR notebook entries</td>
</tr>
<tr>
<td>vawr.list</td>
<td>print VAWR notebook entries</td>
</tr>
<tr>
<td>vawr.notes</td>
<td>display VAWR notebook entries</td>
</tr>
<tr>
<td>vawr.plot</td>
<td>plot VAWR data</td>
</tr>
<tr>
<td>vawr.raw</td>
<td>plot VAWR raw data</td>
</tr>
<tr>
<td>vawr.tabl</td>
<td>generate table file for VAWR data</td>
</tr>
<tr>
<td>vawr.var</td>
<td>edit VAWR plot variables</td>
</tr>
<tr>
<td>vmcm_cal</td>
<td>convert raw VMCM to calibrated</td>
</tr>
<tr>
<td>vmcm.cdf</td>
<td>convert VMCM data to netCDF</td>
</tr>
<tr>
<td>vmcm.env</td>
<td>manipulate VMCM environment variables</td>
</tr>
<tr>
<td>vmcm.log</td>
<td>create VMCM notebook entries</td>
</tr>
<tr>
<td>vmcm.list</td>
<td>print VMCM notebook entries</td>
</tr>
<tr>
<td>vmcm.notes</td>
<td>display VMCM notebook entries</td>
</tr>
<tr>
<td>vmcm.plot</td>
<td>plot VMCM data</td>
</tr>
<tr>
<td>vmcm.raw</td>
<td>plot VMCM raw data</td>
</tr>
<tr>
<td>vmcm.tabl</td>
<td>generate table file for VMCM data</td>
</tr>
<tr>
<td>vmcm.var</td>
<td>edit VMCM plot variables</td>
</tr>
<tr>
<td>xplot</td>
<td>X-windows plot output</td>
</tr>
<tr>
<td>xuop</td>
<td>X-windows menu program</td>
</tr>
</tbody>
</table>
C Plot Examples

This appendix contains examples of various plots from system processing output.

figure 6  IMET plot of seven variables for one day
figure 7  IMET plot of seven variables for three days
figure 8  VAWR plot of seven variables for three days
figure 9  VAWR plot of seven variables for one month
figure 10 VMCM plot of three variables for three months
figure 11 TPOD plot of single variable for three months
figure 12 VAWR/IMET overplot of five variables
figure 13 Overplot of air and sea temperatures from four instruments
Figure 6: IMET Plot of Seven Variables for One Day
Figure 8: VAWR Plot of Seven Variables for Three Days
Figure 9: VAWR Plot of Seven Variables for One Month
Figure 10: VMCM Plot of Three Variables for Three Months
Figure 11: TPOD Plot of One Variable for Three Months
Figure 12: VAWR/IMET Overplot of Five Variables for Three Days
Attn: Stella Sanchez-Wade
Documents Section
Scripps Institution of Oceanography
Library, Mail Code C-075C
La Jolla, CA 92093
Hancock Library of Biology &
Oceanography
Alan Hancock Laboratory
University of Southern California
University Park
Los Angeles, CA 90089-0371
Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA
Office of the International
Ice Patrol
c/o Coast Guard R & D Center
Avery Point
Groton, CT 06340
NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149
Library
Skidaway Institute of Oceanography
P.O. Box 13687
Savannah, GA 31416
Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI 96822
Marine Resources Information Center
Building E38-320
MIT
Cambridge, MA 02139
Library
Lamont-Doherty Geological
Observatory
Columbia University
Palisades, NY 10964
Library
Serials Department
Oregon State University
Corvallis, OR 97331
Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI 02882
Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX 77843
Library
Virginia Institute of Marine Science
Gloucester Point, VA 23062
Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA 98195
Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL 33149
Maury Oceanographic Library
Naval Oceanographic Office
Stennis Space Center
NSTL, MS 39522-5001
Marine Sciences Collection
Mayaguez Campus Library
University of Puerto Rico
Mayaguez, Puerto Rico 00708
Library
Institute of Oceanographic Sciences
Deacon Laboratory
Wormley, Godalming
Surrey GU8 5UB
UNITED KINGDOM
The Librarian
CSIRO Marine Laboratories
G.P.O. Box 1538
Hobart, Tasmania
AUSTRALIA 7001
Library
Proudman Oceanographic Laboratory
Bidston Observatory
Wirkenhead
Merseyside L43 7 RA
UNITED KINGDOM
**Title and Subtitle**
A System for Shipboard Analysis of Buoy Data

**Author(s)**
Kenneth E. Prada

**Performing Organization Name and Address**
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

**Type of Report & Period Covered**
Technical Report

**Abstract (Limit: 200 words)**
This report describes a software system for shipboard processing of buoy data. These data are derived from a variety of instruments installed on buoys recovered during turn-around cruises for the SUBDUCTION experiment. The software package, which operates on a network of PCs and SUN workstations, provides an interactive and versatile set of tools for handling these data sets.

The data are extracted from the native instrument storage media and format, converted to a standard storage format in calibrated scientific units. The resulting files are then processed and displayed for evaluation and analysis. These steps occur within a short period of time following buoy recovery.

**Document Analysis**

**Descriptors**
data analysis
buoys
shipboard systems

**Availability Statement**
Approved for public release; distribution unlimited.