

Technical Report

April 1994



EPIC Processing Toolbox

Users Guide

by

Melora Park Samelson



Upper Ocean Processes Group
UOP Technical Report 94-2

WHOI-94-11
UOP-94-2

EPIC Processing Toolbox Users Guide

by

Melora Park Samelson

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

April 1994

Technical Report

Funding was provided by the Office of naval Research
under Contract No. N00014-90-J-1495.

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-94-11.

Approved for public release; distribution unlimited.

Approved for Distribution:



Philip Richardson, Chair
Physical Oceanography

Abstract

A set of computer programs for data analysis is described and documented. The programs were developed to aid in the processing of time-series data collected by the Upper Ocean Processes Group. These programs, or tools, utilize a platform independent format known as the Network Common Data Format (netCDF). The format is further defined using a convention known as EPIC which allows easy access to other plotting and manipulation tools available to the UOP group. The general method for implementation of the toolbox is described. A reference section lists the available tools, as well as examples and detailed descriptions of their usage and functionality. The toolbox has been tested and used extensively on both Sun and Silicon Graphics Interface UNIX workstations.

Table of Contents

Abstract	i
I Introduction	1
A. Motivation	1
B. NetCDF	2
C. EPIC	2
D. EPIC Time-series	3
II Building the Tools	3
A. Hardware and Operating Systems	3
B. Libraries	4
III General Notes on Using the Tools	4
A. Time and Depth Dimensions	4
B. Variable Selection	5
C. Start and Stop Times	5
D. Log Files	5
E. NetCDF Attributes	6
F. NetCDF Fill Value	7
IV Alphabetical List of Tools	8
V Tool Description	10
VI Examples	47
A. NetCDF header from a simple time series file	47
B. NetCDF header from a two-dimensional file	49
C. Using the toolbox to put a VAWR file in EPIC	51
D. Create an evenly spaced time series	53
E. Calculate cloud-cover estimate	54
F. Calculate heat fluxes, stress	57
G. Using the toolbox to grid and average a time series file	58
VII Miscellaneous	61
A. The Plot Plus Scientific Graphics System (PPLUS)	61
B. The mexcdf MATLAB interface	61
C. The AVS netCDF interface	61

D. The Explorer netCDF interface	61
References	63
Acknowledgements	64

I Introduction

A. Motivation

Scientific programming and analysis at the Woods Hole Oceanographic Institution has undergone significant changes over the last few years, primarily due to the availability of powerful and inexpensive desktop computers, or "workstations." As at many other academic institutions, computer processing has been done primarily on a large central computer, with the expense of buying, using and maintaining the equipment shared by many users. As individual workstation use increased, the number of people sharing the expense of the central computer decreased. By early 1990, it no longer made sense financially or computationally to continue using the Institution's central computer, and the Upper Ocean Processes (UOP) group in the physical oceanography department began the transition to the workstation environment.

A series of intensive field experiments beginning in 1991 and ending two years later provided another motivation for the transition. Processing capability would be needed at sea (Trask and Brink, 1993) since instruments were being redeployed at the mooring site. This required redundant computer systems that could be shipped away for months at a time. Processing software would have to be developed for the workstations that could extract and display the data from the instruments for at-sea instrument quality control.

Software packages for doing data processing using the Institution's central computer had been developed a number of years ago and were tailored to the central computer, serially connected terminals and Tektronix-style graphical interfaces. Current standards for graphical interfaces and availability of "tool"-based software suggested a different approach for development of the new data processing package.

The initial software development was undertaken by Thomas Danforth (Danforth, 1990) and Kenneth Prada (Prada, 1992). Programs were written to transfer instrument data to disk files, store in a standardized format, and apply calibration constants. Graphical display routines could then be used to view data sets. The data sets could be archived to permanent media. This effort was aimed towards satisfying the goal of a shipboard processing system.

This document describes the development of a set of "tools," or computer programs designed to further the processing and analysis capability of the new processing software. These tools provide the user with the ability to move beyond the

early evaluation required for the shipboard environment and into the area of scientific analysis. Some of the applications include averaging, filtering, Fast Fourier Transform, and heat flux calculations. A list of available tools is provided in section IV below.

B. NetCDF

A platform-independent format (preferably binary, to save space) was needed for data storage. Dr. Richard Signell (USGS, Woods Hole) suggested a public-domain product known as the Network Common Data Format, or NetCDF (Rew *et al.*, 1993), which had been developed by Unidata. Other scientific institutions (National Oceanographic and Atmospheric Administration/Pacific Marine Environmental Laboratory (NOAA/PMEL), United States Geological Survey at Woods Hole) with similar data sets were also selecting this format.

NetCDF defines a binary, platform-independent format. This format is implemented through the use of a collection of subroutines which provide creation and read and write access to data stored in this format. The subroutines are compiled in a library which can be accessed using FORTRAN or C calling conventions. It is capable of handling multi-dimensional data sets. Each NetCDF file contains a self-documenting description of its contents and organization.

C. EPIC

Many kinds of netCDF files can be created in the same sense that many types of formatted ASCII files can be created. For this reason, it is convenient and expedient to further restrict the format of the netCDF file so that programs don't have to account for every possible implementation. Such a restriction had already been documented and implemented by the computing and analysis group at NOAA/PMEL (Denbo and Zhu, 1993), and is widely known as "EPIC."

The EPIC format assumes that each variable in the file is a function of four dimensions: time, depth, latitude and longitude. Often, two or three of these dimensions will only have a size of one. For example, a conductivity-temperature-depth (CTD) profile will have a single dimension for time, latitude and longitude, with the depth dimension being large. A moored temperature recorder will have dimension values greater than one for "time" only.

Selecting a method for storing time information in any file is often a difficult problem. Although the EPIC time format definition is under review, the current primary representation is a two-integer-number format. The first integer is the true Julian Day Number, where the day is taken to begin at midnight. The second integer is the number of milliseconds since midnight. In raw form, this time representation can be difficult to use. However, it has the distinct advantage of being unambiguous and capable of accurately representing both long time periods (years) as well as extremely short time periods. Conversion tools that allow several options for extracting time information and presenting it in an easily readable format are crucial.

D. EPIC Time-series

Most of the data that is collected by the UOP group is from instruments deployed on moorings. This produces time-series data sets at various depths. As a result, our implementation of EPIC almost always assumes the dimensions of latitude and longitude to be fixed. The tools assume that data is dimensioned only in time and possibly in depth as well.

Using a netCDF utility called “ncdump,” it is possible to make a listing of the header from a typical EPIC file in the style that is handled by these tools. Examples of typical EPIC headers that these tools operate on can be seen in section VI.

As different types of data sets become incorporated into our processing system, we expect the capability of the toolbox will increase to meet the demands of the new netCDF styles. As new programs are written to analyze the data sets, they can be incorporated as tools for future analysis.

II Building the Tools

A. Hardware and Operating Systems

The tools were written for use on UNIX workstations and have been successfully compiled, built and tested on:

Silicon Graphics	Indigo	IRIX 5.1
Sun	Sparc2	SunOS 4.1.1
Sun	Sparc10	SunOS 4.1.3
Masscomp	5600	RTU 4.0

The code is written primarily in C, with some of the tools requiring FORTRAN interfaces to selected subroutines. The code should be portable to most platforms, possibly with minor modifications.

B. Libraries

There are two libraries that must be built and installed. First, the netCDF library (libnetcdf.a) is required. The source code has been built and tested on a wide variety of platforms, and is available via anonymous ftp from Unidata. Alternatively, you can copy a library that has already been built from a compatible machine.

The other required library is the EPIC library (libeps.a), which provides a simplified interface for reading and writing EPIC style netCDF files. The toolbox uses the time manipulation subroutines from this library.

III General Notes on Using the Tools

A. Time and Depth Dimensions

As discussed in section I, most of the tools operate on time-series data. In the case of time series at a single depth, there is no ambiguity about how to extract the data for manipulation. However, for time series of data at several or many depths, the user must select which depths, or "bins" to operate on. These two-dimensional files can also be treated as a series of profiles rather than time-series. For example, shear is calculated by operating on all the velocity depth bins at a particular time. Some tools allow operation in either direction. For example, extraction of files to ASCII can be done as a set of time series of data at selected depths, or as a single time series of depth profiles. Tools that have the option of operating in either the time or depth dimension will prompt the user with the question:

Do you want to operate on time-series or profiles (t/p)?

Tools that are designed to run on time series of data at a selected depth can perform the operation repeatedly for different depth bins. Most of the tools will prompt the user to select a range of depth bins to operate on.

```
Number of depth bins= 10  start    0.450 end    10.450
Enter start,end bins # to extract (1 - 10) : 2,7
```

B. Variable Selection

Most tools request the user to select or identify variables in the input list to be operated on. There is usually some flexibility in selecting which variables to save to the output file. The variables and operations that the user selects are recorded in the log file (section II.D below). Variable names are usually preserved from input to output unless the variable has been substantially modified. In that case, the variable names are often modified as well. Use the netCDF utility "ncdump -h" to get a listing of any netCDF file header which includes variable names and attributes.

C. Start and Stop Times

Most tools allow the user to select a start and stop time. The user can enter this information by record number or in calendar format.

```
Input file number of time records = 14464
  start 10/05/1992 06:55:00
  end 03/04/1993 22:40:00
Do you want to truncate the output (y,n)?: y

How do you want to enter time (by record=1, calendar=2): 1
  enter starting record [ 1 ] : 100
  enter ending record [14464] : 1999

How do you want to enter time (by record=1, calendar=2): 2
  enter start date (yr mo dy hr mn ): 92 10 22 0 0
  enter end date (yr mo dy hr mn ): 93 3 4 0 0
```

D. Log Files

Most of the tools produce an ASCII file log file for information purposes. The log file name is created by appending the suffix ".log" to the output file name. The log

file contains general information about the input file such as the number and name of variables, number of depth and time records and their minimum and maximum values, and any information needed to recreate the output file. Other products resulting from the execution of the tool, such as statistics and listings of outliers, are saved to the log file as well.

On platforms with UNIX System V file naming conventions, file names have a 14 character limit. The log file name (output file name plus the “.log” extension) will be truncated to the 14 character limit, overwriting the output file, if the output file name has 14 characters. System V users should be careful to select output file names that are not excessive in length.

E. NetCDF Attributes

Careful documentation of any processing is crucial for archiving purposes. NetCDF is self-documenting to the extent that variable dimensions and abbreviated names are part of the file. The use of “attributes” greatly enhances the documentation of the file. An attribute is simply a descriptive character string that is tagged with a name. For instance, “instrument” could be an attribute name, and the description could be “TOGA VAWR #0723”. NetCDF files are capable of having attributes associated with a particular variable and also attributes that are known as “global,” which means they are not associated with any variable.

A record of the processing steps is kept by the netCDF toolbox. A series of global attributes are contained at the end of each netCDF file which show the user the entire processing history of the data file, including the tools that were used, and the file names involved. Each tool appends a global attribute to the new output file header. This global attribute name is the UNIX system date and time, followed by the input file name(s), the output file name, and any easy to document and relevant parameters. Using the netCDF utility, “ncdump -h,” this listing of processing steps is easily displayed.

Other useful global attributes that are commonly found in the toolbox files are:

ATTRIBUTE NAME	EXAMPLE
compliance	EPIC
experiment	TOGA COARE
platform	Mooring #933
instrument_type	VAWR
instrument_number	0723
comments	creating 1 hour averages

F. NetCDF Fill Value

The NetCDF library predefines a “fill” value, or value for which no data is present. In the previous release of the netCDF library, this value was set to be just larger than a valid floating point number, which is typically labeled by a computer as “infinity,” or “inf”. Since different platforms had different thresholds for reaching this invalid floating point value, manipulation of the files might vary between platforms. To correct this problem, the latest version of the netCDF library has defined a new fill value which is the same number on all of its tested platforms. It is a large positive number ($\sim 10e+36$).

Most of the tools will recognize this value and either skip it when processing, or interpolate over the gap. In some cases, the tool will warn the user and exit the program when it cannot manage a file with gaps. Many of the tools have a “bad value” option which can be specified. The default value in all cases is the netCDF fill value. When creating an ASCII file, the user can opt to replace any incidence of the fill value with a more easily readable result, -999.0 for example.

IV Alphabetical List of Tools

- adcptoepic - Converts ADCP ASCII files from program “unpack” to EPIC netCDF.
- addepicvar - Combine variables from two EPIC netCDF files.
- asctoepic - Generic ASCII to EPIC netCDF converter.
- atc_cloud - Calculates theoretical cloud cover from meteorological data stored in EPIC netCDF (part 1 of the flux processing routine using the Clark method).
- diffepic - Calculates differences between selected variables in two separate files.
- dmeanepic - Removes the mean from a time-series.
- editepic - Edits bad data points using linear interpolation allowing variable selection, time range and limits to be set by the user.
- fftepic - Computes an FFT from either scalar or vector data.
- fltepic - Filtering and decimation tool.
- fluxepic - Computes total heat flux using the Clark method.
- gridepic - Grids time-series data using linear interpolation.
- mergepic - Merges two EPIC netCDF files from separate time domains.
- outlier - Outlier detection and linear interpolation.
- provecepic - Calculates progressive vector components from velocity and produces a PPLUS command file with annotation and wind sticks.
- putinepic - Converts a netCDF file from first step processing to EPIC format.
- putinepicmet - Converts an IMET netCDF file from first step processing to EPIC format.

- readepic - Reads an EPIC netCDF file and produces an ASCII column-format listing.
- scattoepic - Converts ASCII SEACAT file to EPIC netCDF.
- shearepic - Compute shear components from velocity.
- statepic - Computes basic statistics.
- timecheck - Creates an evenly sampled time series, filling gaps using a linear interpolation scheme.
- timeedit - Time base correction using a simple offset.
- timeedit2 - Time base correction allowing stretch or shrink.
- transform - Performs several type of algebraic transformations on selected variables:
 1. Quadratic transformation as a function of a variable x .
 2. Vector rotation.
 3. Quadratic transformation as a function of time.
 4. Speed and direction computation from velocity components.
 5. Progressive vector from velocity components.
- unescoepic - Calculates properties of sea water (salinity, density, etc).
- zmerge - Creates a 2-dimensional depth and time file from separate time-series files that have been uniformly gridded in time.

V Tool Description

adcptoepic

usage:

adcptoepic inputpath outputpath

inputpath = path of input ASCII files reside (default "./").
outputpath = path of output EPIC files reside (default "./").

subroutines:

epicutils EPIC utility routines

This routine converts the ASCII output from the routine "unpack" and places the resulting files in EPIC format. The user can select which type of files to convert and which variables to select within the files. The user is asked to provide depth and position information as well as some general instrument information. Truncation of the file can be done by selecting a start and stop time.

The input file names are predefined in the program "unpack" as follows:

header.dat				header info
velb1.dat,	velb2.dat,	velb3.dat,	velb4.dat	velocity data
eab1.dat,	eab2.dat,	eab3.dat,	eab4.dat	echo amplitude data
swb1.dat,	swb2.dat,	swb3.dat,	swb4.dat	spectral width data
gdb1.dat,	gdb2.dat,	gdb3.dat,	gdb4.dat	percent good data
stb1.dat,	stb2.dat,	stb3.dat,	stb4.dat	status data

The output file names predefined in this program are:

adcpheader.epic
adcpvelocity.epic
adcpechoamp.epic
adcpspecwid.epic
adcppergood.epic
adcpstatus.epic

addepicvar

usage:

addepicvar inputfile1 inputfile2 outputfile

inputfile1 = an EPIC netCDF file.
inputfile2 = an EPIC netCDF file.
outputfile = an output EPIC netCDF file.

subroutines:

epicutils EPIC utility routines

This tool allows the user to add variables from inputfile2 to inputfile1, and places the results in outputfile. The sample rates of the two files must match, but the time spans can differ. If the time base of the second file is bigger than the first file, it is truncated to match the first file. If the time base from the second file is shorter than the first file, it places the new records in the appropriate places, and fills the gaps at either end with the netCDF fill value. If the time bases have no overlap, the program exits with a message to that effect. The files must be evenly sampled in time for this program to work properly.

Variable name duplication between the two input files is detected and the user is prompted to determine if the variables should be overwritten. If not, then the user must supply a new name for the duplicate variable in the second input file. The user can choose to overwrite a variable in the first file with a variable from the second file.

The two input files can have a different number of depth bins. The user is asked which depth bins to extract from the second file, and where to insert them in the output file. The number of depth bins that are extracted from the input file must match the number of bins for insertion into the output file. For example, if the user selects bins 1 through 4 from the input file and the output file has 6 bins, then the user could select bins 1 through 4, 2 through 5 or 3 through 6 in the output file.

asctoepic

usage:

```
asctoepic inputfile outputfile
```

```
inputfile = an ASCII file in column format.
```

```
outputfile = an EPIC netCDF file.
```

subroutines:

```
epicutils    EPIC utility routines
```

This tool allows a user to convert time series data in an ASCII tabular format to an EPIC netCDF file. Variables should be listed in separate columns and may be delimited by tabs, spaces, commas or other characters. The time should increase with each successive record. The time need not be explicitly present. Information from the user on many aspects of the file are needed in order to properly convert it. The user is prompted for general information about the experiment, platform, instrument, and for comments.

The user is asked for the number of columns of data in the input file. This number must include all columns, even if some columns will not be saved to the output file. The user is asked to provide the "short name" for each variable. No white space or special characters are allowed for this response. If a column is not desired in the output file, enter "skip". Enter "time" if a column is related to time (see the discussion below about time variables). For each variable also enter a "long name", which is a more descriptive name, and the "units" for this column of data. These entries can contain white space and special characters.

You will be asked for the calendar time for the starting sample as well as the sample rate (enter a nominal value if the data is not evenly sampled in time). Also provide the latitude, longitude and depth of the data set.

The program then needs to know the format of the file to be read. First, it asks for the number of header lines to skip in the file. Then, for each column, enter a C-style format describing this number. In general, all answers should be a floating point format, since the variables are all assumed to be of type "float". This means selecting one of the following:

```
%f    read a number (float or integer) value and store as a float
%*f   skip a number
```

delimiters should be included in the format response:

```
%f, read a number followed by a comma
%f; read a number followed by a semi-column
%*f, skip a number that is followed by a comma
```

Tabs look like white space in C so no delimiter is required.

There are special cases for the "time" variable. If "time" was not selected as a variable name, then the program automatically assumes that the file is evenly sampled in time, and it uses the starting time and sample rate to calculate the "time" variable. In many ways, this is the easiest method to use, even if you have time information present in the file. However, if your file is not evenly sampled in time, then you must tell the program how to read the "time". The choices are:

```
1=record, 2=calendar, 3=elapsedsec, 4=yearday 5=continuous yearday
```

```
1=record
```

```
Time is stored as a record number.
```

```
2=calendar
```

```
Time is stored in a calendar format.
```

```
3=elapsedsec
```

```
Time is stored in elapsed seconds.
```

```
4=yearday
```

```
Time is stored in a decimal yearday format.
```

```
Jan 1,1992 at 0000 = 1.0
```

```
Jan 1,1992 at 1200 = 1.5
```

```
5=continuous yearday
```

```
Time is stored in a continuous decimal yearday format
(unimplemented).
```

```
Jan 1,1991 at 0000 = 1.0
```

```
Jan 1,1991 at 1200 = 1.5
```

```
Jan 1,1992 at 0000 = 366.0
```

```
Jan 1,1992 at 1200 = 366.5
```

The program prints out the first line of data in the input file using the format instructions entered by the user. If this line does not look correct, then the opportunity to reenter the format instructions is provided.

Example:

```
-----  
This is the first header line  
this is another header line,  
  sec,  rh ,   lw,   sw,tension,   u,   v,  
45680, 65.0, 567.0, 1113.0, 32573, 10.341, 0.823,  
45690, 63.0, 439.0, 1002.0, 32570, 12.484, 1.583,  
45700, 65.0, 698.0, 1018.0, 32400,  8.262, 5.492,  
45710, 79.0, 494.0, 1021.0, 32500,  4.282, 9.383,  
-----
```

Enter number of columns (variables in input file) : 7
Enter variable information (enter "time" for time or "skip")

```
variable short name for column 1 : skip  
variable short name for column 2 : rh  
variable long name for column 2 : relative humidity  
variable units for column 2 : %  
variable short name for column 3 : lw  
variable long name for column 3 : long wave radiation  
variable units for column 3 : W/m**2  
variable short name for column 4 : sw  
variable long name for column 4 : short wave radiation  
variable units for column 4 : W/m**2  
variable short name for column 5 : tens  
variable long name for column 5 : buoy tension  
variable units for column 5 : counts  
variable short name for column 6 : skip  
variable short name for column 7 : v  
variable long name for column 7 : wind velocity  
variable units for column 7 : m/s
```

Enter the starting date,time from the file (yr mo dy hr mn sc) :
92 10 22 0 0 0

Enter nominal sample rate of data (seconds): 900

Enter latitude in degrees (+ = N) : 1.7554
Enter longitude in degrees (+ = E) : 155.9945
Enter Depth of instrument (meters) : 0.0

Enter number of lines to skip at the beginning of the input file : 3

```
Enter format for variable " skip" column 1 (C style float) : %*f,  
Enter format for variable " rh" column 2 (C style float) : %f,  
Enter format for variable " lw" column 3 (C style float) : %f,  
Enter format for variable " sw" column 4 (C style float) : %f,  
Enter format for variable " tens" column 5 (C style float) : %f,  
Enter format for variable " skip" column 6 (C style float) : %*f,  
Enter format for variable " v" column 7 (C style float) : %f,
```

0.000000, 65.000000, 567.000000, 1113.000000, 32573.000000,0.000000, 0.823000,

Does this look OK for line # 1 (y,n) : y

atc_cloud

usage:

atc_cloud inputfile outputfile

inputfile = a VAWR EPIC netCDF file.
outputfile = an output EPIC netCDF file.

subroutines:

epicutils	EPIC utility routines
cloudest	cloud-cover estimate routine

This is the first of several programs that must be run in order to compute the heat flux using the Clark method (Clark *et al.*, 1974). It is designed to run on VAWR files, and expects particular variable names. Its primary function is to compute the cloud-cover estimate, and corrected short wave radiation. These variables are output to an EPIC file, along with the theoretical clearsky radiation (adjusted by applying the atmospheric transmission coefficient).

This program will compute the following constants that are required as part of the flux calculations if desired:

Atmospheric transmission coefficient (atc)

The user can choose to enter this value directly. Alternatively, the user will be asked to enter dates upon which a clear sky was observed. The routine then calculates the atc for the selected dates and prints the results to the screen. The user is then asked to select a value for atc to use in the calculation, based on the atc values printed to the screen. This value can be checked and tuned by plotting the results of the output file using the PPLUS command file, atc.ppc, which overplots the shortwave radiation, and the atc adjusted clear-sky radiation on selected dates.

Short Wave night-time bias

A night-time bias in the VAWR short wave radiation signal should be removed before calculation of cloud cover occurs. The user can either enter a pre-determined value, or the program will calculate a mean bias for each local solar night as well as the mean bias and variance for the entire data set. The daily mean can be used to detect a trend in the bias, although, at this time, only an offset can be applied. The adjusted short wave signal in the output file has the night-time bias removed from the entire time series. In addition, it has been forced to zero during the night. This adjusted short wave time series, along with the cloud cover, will be used in the flux calculation.

The user is asked to enter a value for the "sina cutoff" (default 0.15). This value is used to adjust the cloud-cover estimate near dawn and dusk when the shape of the theoretical short wave radiation curve and the observed short wave radiation are changing rapidly. In order to avoid large spikes in the cloud-cover estimate around dawn and dusk of every day, the theoretical short wave is forced to zero early causing the cloud-cover estimate to zero as well. The adjusted observed short wave radiation and the theoretical clear-sky radiation that are saved to the output file do not have this cutoff value applied. It is only applied when calculating the cloud-cover estimate.

The cloud cover cannot be calculated for night-time using this method. Values for the cloud-cover estimate are set to the netCDF fill value at night. The user must filter the resulting cloud cover from this program, using a 30-hr running mean. "Fltepic" will produce the desired results including skipping netCDF fill values at night. This filtered file should be used when calculating fluxes using "fluxepic."

diffepic

usage:

```
diffepic inputfile1 inputfile2 outputfile
```

```
inputfile1 = an EPIC netCDF file
```

```
inputfile2 = an EPIC netCDF file
```

```
outputfile = an EPIC netCDF file
```

subroutines:

```
epicutils    EPIC utility routines
```

This tool calculates differences between variables in two files. The two input files do NOT need to be on the same time grids. They can differ in their sample rates and sample times. Differences are calculated by matching the nearest times from the two files. The times must be within a sample interval based on the sample rate of the first file or an error message is generated.

The variables in each of the two input files are listed to the screen. The user must select which variables to difference. New variable names are created by attaching the two variable names using an underscore (“_”). For example, differencing a variable called “temp” from the first file, and a variable called “temp” from the second file produces a variable name called “temp_temp”. The user can calculate differences between many combinations of variables as long as the resulting variable names remain unique.

Difference statistics are listed at the conclusion of the program and in the log file for informational purposes. These include the mean difference, the number of data points used, the standard deviation of the difference, and the root-mean-square difference.

dmeanepic

usage:

```
dmeanepic inputfile outputfile [badvalue]
```

```
inputfile = an input EPIC netCDF file.
```

```
outputfile = an output EPIC netCDF file
```

```
badvalue   = default is netCDF fill value (FILL_FLOAT)
```

subroutines:

```
epicutils   EPIC utility routines
```

This tool is used to remove the mean from a time series. The user selects which variables to operate on. For each variable, a sum is computed over the selected time range from which the mean is calculated. Bad values are skipped. This mean is then subtracted from each point in the time series. Multiple depth bins are allowed and the range is selected by the user. A summary listing of the mean value for each variable is listed at the end of the calculation, and is repeated for each successive depth bin.

editepic

usage:

```
editepic inputfile outputfile [badvalue]
```

```
inputfile   = an EPIC netCDF file.  
outputfile  = an output EPIC netCDF file.  
badvalue    = default is netCDF fill value (FILL_FLOAT)
```

subroutines:

```
epicutils   EPIC utility routines
```

This tool allows the user to apply an outlier window to selected variables over a specified time interval. Outliers found outside the selected window are replaced with a value created by linearly interpolating the surrounding good points. The program checks ahead a predefined number of points (NAHEAD) for the next good data value before doing the interpolation. If it cannot find a good value within this window, it sets the bad point to the netCDF fill value and continues.

This tool operates on a copy of the input file which is created when the program issues the UNIX copy command to a shell:

```
cp inputfile outputfile
```

This means that the number of records and variables in the output file are identical to the input file. Thus, no file truncation or variable selection options are allowed.

The user is prompted to select a variable name, the minimum and maximum values for that variable, and the starting and ending times to check for values outside the specified range. If the user selected zero for both the minimum and maximum, this blanks the values out in the selected time range by placing the netCDF fill value into these records. Any outliers that are detected in the selected record and variable range are listed to the screen and to the log file.

fftepic

usage:

```
fftepic inputfile outputfile
```

```
inputfile = an EPIC netCDF file
```

```
outputfile = an output ASCII file with FFT coefficients
```

subroutines:

```
epicutils
```

```
EPIC utility routines
```

```
fftone
```

```
FORTRAN FFT subroutine
```

```
fftsize
```

```
FORTRAN FFT array sizing subroutine
```

This tool allows the user to select variables from a netCDF file for computation of a Fast Fourier Transform, or FFT. The coefficients are saved to an ASCII output file (typically named `____.coef`). A log file is created from the output file name and the extension `“.log”` containing relevant parameters selected for the FFT.

This tool will calculate real or imaginary FFTs (auto-spectra or rotary auto-spectra). The user is prompted to select either velocity component pairs for rotary auto-spectra, or a single variable such as temperature, for auto-spectra. Variables selected are listed in the log file.

The user is asked to select starting and ending calendar times or records for the FFT computation. The tool tests the number of records to determine that it is factorable. If it is not, the selected time period is truncated to the next nearest factorable size. This record selection is saved to the log file.

If there is no sample-rate attribute (`“rec_period”`) in the netCDF file the user is prompted to enter the sample rate in seconds. This tool assumes that the file is evenly spaced in time, and that any gaps have been interpolated over. It tests for any occurrence of the netCDF fill value and exits the program warning the user if any are found.

The output of `fftepic` should be processed using the band-averaging program `“bndavgn”`. This tool asks for input ASCII files produced from `fftepic`, and produces an ASCII file suitable for plotting. Auto-spectra or cross-spectra can be produced at this stage.

ftepic

usage:

```
ftepic inputfile outputfile [badvalue]
```

```
inputfile   = an EPIC netCDF file.  
outputfile  = an output EPIC netCDF file.  
badvalue    = default is netCDF fill value (FILL_FLOAT)
```

subroutines:

```
epicutils   EPIC utility routines
```

This tool performs filtering and decimation on an evenly sampled input file. The averaging method is a simple box car filter which performs a running mean. The filter length is the number of points used to compute the mean. The decimation parameter allows the user to select subsets of the running mean for output. If the decimation selected is "1", then every computed mean is saved. If the decimation is equivalent to the filter length, then this produces a conventional box-car average.

The time series may be truncated, with the start and stop times entered as record number or calendar date and time. The user can also select which variables to operate on. This filter can be performed on either time series data at selected depths, or on profiles at selected time values.

Any "bad values" are skipped during averaging. If there are no good data values in the averaging interval, the averaged value is set to the selected "bad value."

fluxepic

usage:

```
fluxepic inputfile outputfile
epicutils    EPIC utility routines
```

```
inputfile    = a VAWR EPIC netCDF file.
outputfile   = an output EPIC netCDF file.
```

subroutines:

```
epicutils    EPIC utility routines
bulk.f       FORTRAN bulk dynamics flux routine
vapor.f      FORTRAN vapor routine
longwave.f   FORTRAN longwave flux routine
```

This tool calculates heat flux using an input VAWR file, as well as an input cloud-cover estimate file, and an adjusted short wave radiation file if desired. It saves any significant screen information to a log file with the extension “.log” applied to the output file name. This tool asks the user to input relevant information about the following items:

- Truncation for new start and stop times (record number or calendar)
- Name of the EPIC file that contains the cloud-cover estimate. This file does not need to be on the same time base as the input file (it has probably been filtered and is not on the same time base).
- Whether the user wants to use the short wave radiation from the input file, or from another EPIC file generated by running the program “atc.cloud”. Generally you should pick the adjusted short wave radiation from the output of “atc.cloud”. This file does need to be on the same time base as the input file.
- The atmospheric transmission coefficient. This should be the same atc that was used to generate the cloud-cover estimate.
- The short wave night-time bias. If you are using the adjusted short-wave radiation from the output of “atc.cloud”, the night-time bias should be entered as “0.0” because it should have already been removed from the time series. If

you choose to use the short wave from the input file, then you may want to enter a non-zero value. If you do this, your night-time values for short wave may not be exactly zero.

- Height of the wind, air, and relative humidity sensors in meters.
- Fluxes can be plotted using the PPLUS command file, flux.ppc. Wind stress can be plotting using the PPLUS command file, stress.ppc.

gridepic

usage:

```
gridepic inputfile outputfile [badvalue]
```

inputfile = an EPIC netCDF file.

outputfile = an output EPIC netCDF file.

badvalue = default is netCDF fill value (FILL_FLOAT)

subroutines:

epicutils EPIC utility routines

This tool grids data to a new time base or a new depth grid using a linear interpolation scheme. The user enters the start and stop time and new sample interval for regridding or the new start and stop depths and depth bin size. The user can select which variables to process. Multiple depth bins are allowed.

If bad values are present in the input file, and they surround a gridpoint, the gridpoint will be set to the selected bad value.

mergepic

usage:

```
mergepic inputfile1 inputfile2 outputfile
```

```
inputfile1 = an EPIC netCDF file.  
inputfile2 = an EPIC netCDF file.  
outputfile = an output EPIC netCDF file.
```

subroutines:

```
epicutils    EPIC utility routines
```

This tool merges (in the time dimension) two input EPIC files producing one outputfile. Both input files must have the same number of variables in the same order. Inputfile2 is essentially appended in time to inputfile1. Global and variable attributes (header information) is saved only from the first input file. No file truncation or variable selection is allowed. The files are joined without regard to the time values so an unevenly sampled time series may result. Use the tool "timecheck" to interpolate over the gap.

outlier

usage:

```
outlier inputfile outputfile [badvalue]
```

```
inputfile   = an EPIC netCDF file.  
outputfile  = an output EPIC netCDF file.  
badvalue    = default is netCDF fill value (FILL_FLOAT)
```

subroutines:

```
epicutils   EPIC utility routines
```

This tool does outlier detection and linear interpolation. The user can select which variables to process and save to the output file as well as a start and stop time interval. The user is prompted for the allowable window for each variable (minimum and maximum values). The default range is plus or minus the netCDF fill value. Any values found outside the specified window are replaced with a linearly interpolated value computed from the nearest good values. The program looks ahead a predefined (NAHEAD) number of records before deciding to give up. In that case the value is set to the netCDF fill value and the window checking process continues. The same window for the selected variable is applied throughout the entire record. See "editepic" for an alternative to removing outliers from subsets of the file.

A bad value can be entered at the command line and any data points equivalent to that bad value will be considered an outlier. This value is global in the sense that it is applied uniformly to all variables and records.

When outliers are found, they are listed to the screen and to the log file. Outliers greater than or equal to the netCDF fill value are not listed to the screen. A summary table showing the total number of outliers (including any netCDF fill values) for each variable is generated at the end.

provecepic

usage:

```
provecepic inputfile outputfile
```

```
inputfile = an EPIC netCDF file
```

```
outputfile = an EPIC netCDF file
```

subroutines:

```
epicutils      EPIC utility routines
```

This tool calculates progressive vector data from velocity components and stores the results in an output file. It also generates a PPLUS command file which can be used to produce a plot. Wind vector sticks can also be included in the output command file for plotting. This PPLUS command file name is the outputfile name with the extension “.ppc”.

The tool prompts the user to select the east and north components of velocity from the variable list. The user is asked whether to include wind sticks. If selected, the user is asked for the name of the EPIC netCDF file that contains the wind components, and to identify the variable names that correspond to the wind components.

The user also selects the starting and ending times, and depth bin values from the input file.

New variable names are created for the EPIC output file by prepending “prov_” to the original variable name. For example, “east” becomes “prov_east”. The units for the velocity and the sample rate must be determined in order to convert to kilometers properly. The tool attempts to determine the units from the input file information, but if it is unable to find the proper information it will prompt the user to enter a conversion factor. Several options are provided.

The user is prompted for the annotation interval for the progressive vector. The annotation variable names are created by prepending “an_” to the velocity component variable names. When an annotation interval has been reached, the values of the progressive vector coordinates are written to the output file. Otherwise, the record is blank. These data are used to mark the provec at the annotation interval with a symbol. The annotation labels (calendar dates) are placed in the PPLUS command

file and may be edited accordingly before plotting. Too many annotation labels creates a confusing plot, and will overflow the PPLUS label buffer.

The user is also prompted for the wind stick interval. This information is calculated and stored only in the PPLUS command file and may be edited before plotting.

A simpler alternative to this program is to use "transform". This calculates the progressive vector components and includes them as variables in the output file. No wind sticks, annotation or labeling is included.

putinepic

usage:

```
putinepic inputfile outputfile [nointeractive]
```

`inputfile` = a netCDF ".cal" file generated by the UOP processing system.
`outputfile` = an output EPIC netCDF file.
`nointeractive` = the user will not be prompted for time base corrections (default is to prompt).

subroutines:

<code>epicutils</code>	EPIC utility routines
<code>dayear</code>	UOP time handling routine (seconds of the year)
<code>cdftime</code>	UOP time handling routine (seconds of the year)

This tool reformats the UOP netCDF calibrated instrument files (VAWR, VMCM, TPOD) into EPIC netCDF format. This is the first step that must be performed before any other processing or plotting can be done using the tools described in this document. Because this program attempts to meet OARS (Oceanographic Analysis Resource System) standards, it asks the user to input information that is already in the input netCDF file, but is buried in such a way that extraction is difficult. It also asks for latitude, longitude and depth information. The position information can be critical for later computations and should be entered accurately.

The user can select which variables to transfer to the new output format. Attributes for the selected variables in the input file are copied to the output file. Global attributes are reworked to meet the OARS compliance, and written to the output file. The file can also be truncated, with start and stop times entered by the user. During run time, relevant information is output to the terminal, as well as to a log file.

This tool performs outlier detection and interpolation. The user is asked to input a minimum and maximum value for each variable selected (default is plus or minus the netCDF fill value), and each point outside this range is linearly interpolated over using the nearest valid data points. An attribute called "valid_range" is added to the output file for each selected variable for which outlier detection is being done. The value for any outlier found is sent to the user terminal as well as to the log file. A

summary of the total number of outliers detected and repaired is printed at the end of the run.

If the user has left the default interactive argument on, any large time jumps are detected. The data record on either side of the time jump as well as the current record are displayed to the screen. The user is asked to choose one of three options: 1) continue processing (output the record unchanged), 2) skip this record when processing to the output file, and 3) attempt to fix the time. If there is a true skip in the file, then the user should select option 1. If the time and the data in the record appear to be bad, the user should select option 2. If the time appears to be bad but the data looks reasonable, then the user should select option 3. The time is fixed by adding the sample interval to the previous time to create the new time for this record. Although several methods for correcting the time are possible, this one was chosen to keep the clock word on the proper sample rate time base. If the time gap is any bigger than one sample interval, the assignment of time to this record is arbitrary within the limits of the surrounding time words.

In general, the user should run this program with the interactive portion on unless the time word in the input file is known to be well behaved. An input file with a large erroneous time error can wreak havoc with the output file time word causing, among other things, the year to increment improperly. If "noninteractive" is selected, the file is processed as in option 1.

putinepicmet

usage:

```
putinepicmet inputpath outputfile
```

```
inputfile = a path where the daily IMET files reside  
outputfile = an output EPIC netCDF file.
```

subroutines:

```
epicutils    EPIC utility routines  
dayyear      UOP time handling routine (seconds of the year)  
cdftime      UOP time handling routine (seconds of the year)
```

This tool reformats the UOP netCDF calibrated IMET instrument files into another netCDF format called EPIC in much the same way that program “putinepic” functions. However, since there are multiple IMET files for one experiment (one file per day), special handling is required. The output file is intended to be one large file instead of multiple files, however it is possible to process these files in smaller pieces and merge them later when a single input file is required. Once the output file has been created, any of the standard EPIC tools can be used to further process the data.

Because of the nature of the input files, the output file will always be evenly sampled in time. If there are any missing files, the user is asked whether to continue, and if the response is “yes”, then the missing gap is filled with the netCDF fill value. If files are short, the missing records are filled with the netCDF value. This saves the extra step of checking the time base with the program “timecheck”.

The user can select start and stop days and variables to process. Outlier detection is not allowed at this stage because of the difficulty of linear interpolation with multiple file input. The program generates an output log file that lists relevant information as the program is running.

The IMET longwave radiation sensor sometimes gets connected improperly requiring a recalculation of the measured longwave radiation from the fundamental variables. If you want to perform this recalculation, answer “yes” when prompted by the program, and also enter the required constants.

readepic

usage:

```
readepic inputfile [outputfile]
```

inputfile = an EPIC netCDF file.

outputfile = an output ASCII file. If none is specified than the output it sent to "stdout" (the screen).

subroutines:

none

This tool reads an EPIC file and creates an ASCII listing of data from the input netCDF file. The input netCDF file can have multiple time values as well as depth value (two dimensional).

The user is asked whether to replace any netCDF fill values with a different value. The netCDF fill value is a large number ($\sim 10e+36$) and, in the default floating point format, will produce a 36 digit number. If you anticipate the presence of fill values, then it is best to replace it with a bad value indicator of your choice (-999.0 for example).

The user is asked to select whether the listing will be of "time series" or "profiles". A selection of "time series" data lists all the time records at the starting depth bin, followed by a list of all the time records at successive depths bins. The "profile" option lists data at all the selected depth bins for the first time record, followed by a list of data at all the depth bins for successive time records.

The user can select which variables to list, and the starting and ending time interval (by record or calendar time). The tool also asks for the starting and ending depth bins. "Bins" are not the actual depth values, but are an integer counter from "1" to the number of total depths in the file (much like a time record number).

The user can also specify the style of the time word (none, record number, calendar, EPIC decimal, yearday, continuous yearday), and the output format for the variables being listed. The format is entered using C formats. The default is:

"%8.3f,". You can enter C format control characters as well. Some examples:

```
%8.3f\t  tab character after float variable  
%8.0f\n  line-feed character after float variable  
%10.6f  space separated float variable
```

scattoepic

usage:

scattoepic inputfile outputfile

inputfile = an ASCII seacat instrument file converted to scientific units (*.cnv).

outputfile = an output EPIC netCDF file.

subroutines:

epicutils EPIC utility routines

This tool converts the output of the Seabird Seacat DOS conversion program ("datcnv") to EPIC. It reads the header information to determine the variables in the file. The user is asked to enter some general instrument information that is not readily available in the header. Deployment depth and position are also required.

shearepic

usage:

```
shearepic inputfile outputfile [bad value]
```

```
inputfile = an EPIC netCDF file
```

```
outputfile = an EPIC netCDF file
```

subroutines:

```
epicutils    EPIC utility routines
```

This tool calculates shear from velocity components. The user identifies the velocity component variables. Calculation of dU/dz and dV/dz is done directly without any conversion of velocity units which are typically in cm/s. The resulting shears are therefore in units of cm/s/m.

The tool tests for the presence of data at more than one depth. If it finds multiple depth values, shears are calculated for each time step and saved to a similar two dimensional file. If the tool determines that there is only one depth value, then it prompts the user for a second file name and uses those two files to calculate the shears. If they are not on the same time grid, the program will find the nearest times in the two files for the calculation. Use "gridepic" if you desire matched time bases.

statepic

usage:

statepic inputfile

inputfile = an EPIC netCDF file

subroutines:

none

This tool computes time-series statistics for user selected variables. The user can also select a time range and a depth bin. The output is an ASCII listing to the screen. Statistics that are computed include the mean, the number of points, and the standard deviation. Any netCDF fill values are automatically skipped.

timecheck

usage:

```
timecheck inputfile outputfile
```

```
inputfile   = an EPIC netCDF file.
```

```
outputfile  = an output EPIC netCDF file.
```

subroutines:

```
epicutils   EPIC utility routine
```

This tool checks for time gaps in the input EPIC file, and linearly interpolates over any gaps. It requires the presence of an attribute in the input file called "rec_period" which is the sample rate in seconds. The program assumes that all of the time values were generated from multiples of the sample interval (i.e. clock word * sample rate). It cannot gracefully handle large time spikes, or improper resets in the file that cause the time to jump backwards to the reset time and begin incrementing again. These large time errors must be removed first using either "putinepic" or the UOP processing routines that generate the time word.

Any detected gaps are listed to the screen, and to a log file, and a summary is listed at the end of a run.

The file can be truncated by entering new start and stop time either using record numbers, or calendar date/time. The user can select which variables to write to the output file.

timeedit

usage:

```
timeedit inputfile outputfile
```

```
inputfile    = an EPIC netCDF file.
```

```
outputfile   = an output EPIC netCDF file.
```

subroutines:

```
epicutils    EPIC utility routine
```

This tool allows the user to shift the time base of the input file. The user is prompted to enter the number of seconds to shift the time base. If the number of seconds entered is positive, it is added to each time step in the file. If the number of seconds is negative, it is subtracted from each time step.

The user is asked for the start and stop times and variables to output. Input files with multiple depth values are allowed. The user must select which depth bins to operate on.

timeedit2

usage:

```
timeedit2 inputfile outputfile
```

```
inputfile   = an EPIC netCDF file.  
outputfile  = an output EPIC netCDF file.
```

subroutines:

```
epicutils   EPIC utility routine
```

This tool allows the user to alter the time base of the input file using a linear interpolation scheme. The user is prompted to enter starting and ending times, and the starting and ending number of seconds to shift the time base by. Using these values, the number of seconds to shift each time step is calculated and applied assuming a linear shift. If the number of seconds to shift is positive, it is added to each time step in the file. If the number of seconds is negative, it is subtracted from each time step. This results in a file that is NOT evenly sampled in time. Use the tool "gridepic" afterwards to produce an evenly sampled time series.

The user is asked for the start and stop times and variables to output. Input files with multiple depth values are allowed. The user must select which depth bins to operate on.

transform

usage:

```
transform inputfile outputfile
```

```
inputfile    = an EPIC netCDF file.
```

```
outputfile   = an output EPIC netCDF file.
```

subroutines:

```
epicutils    EPIC utility routines
```

This tool allows the user to perform certain mathematical transformations on selected variables. The file can be truncated by entering start and stop time information using either record numbers, or calendar date/time. The user can select which variables to write to the output file. If any of the variables in the output file are transformed by one of the above methods, a variable attribute is added to the output EPIC file called "transform". This attribute includes relevant coefficients or parameters that were used to accomplish the transformation.

There are five types of manipulations allowed: (1) a quadratic transformation, (2) a vector rotation, (3) a time-varying bias correction, (4) computation of speed and direction from vector components, (5) a progressive vector calculation from velocity components.

1. The first is a linear or first order quadratic transformation to any number of selected variables. The user is asked to enter the coefficients a,b,c for each selected variable. The algorithm applied is of the format:

$$y = a + b*x + c*x*x$$

where x is a variable at time t.

2. The second type of transformation allowed is a vector rotation. You may only transform one vector variable in the file. You are asked to identify the east and north components of the variable and the rotation desired in degrees. If the program can identify a variable in the record that corresponds to direction, it attempts to correct this variable as well. Currently, it looks for variables named "dir" (VAWR) and "wndd" (IMET). If the vector direction is not named "dir" or "wndd", the direction variable will not be corrected properly for the applied rotation. The following algorithm is used:

```
ne = e*cos(m) + n*sin(m)
nn = n*cos(m) - e*sin(m)
dir = arctan(e,n)
```

where:

```
e = east component
n = north component
m = direction in degrees (i.e. 13.5W magnetic variation = -13.5)
ne = new east component
nn = new north component
```

3. The third type of transformation performed is a time varying bias adjustment. The user is asked to enter start and stop times for application of the bias. The record number corresponding to the start and stop times are used to determine the limits of the bias calculation. There are two possible forms for entering the bias.

The first is a third order polynomial format:

```
bias = a + b*r + c*r*r + d*r*r*r) :
```

where:

```
r = record number
a,b,c,d are user-supplied constants.
```

The second is a simple linear equation for which the user is asked to enter a starting and ending bias value. The algorithm is a simple linear interpolation of the bias value based on the starting and ending times selected. The calculated bias is then applied to the selected variable.

```
b = (b1-b0)*(r-r0)/(r1-r0) + b0
y = x + b
```

where:

```
r0 = starting record number (corresponding to time t0)
r1 = ending record number (corresponding to time t1)
b0 = starting bias value
b1 = ending bias value
r = a record number in the range r0 to r1
```

b = bias as a function of record number
x = data value to be offset
y = new offset data value

If the record number being processed is outside the range selected, then no transformation is done to the data.

4. The fourth type of transformation performed is a calculation of speed and direction from vector components.

```
spd = sqrt(e*e + n*n)
dir = atan(e/n)
```

The user is asked to identify the vector components and speed and direction variables if they exist. If speed and direction variables are selected, they will be overwritten. If they are not selected, new variables are created. Only one vector calculation of speed and direction is allowed.

5. The fifth choice calculates progressive vector components from the two velocity components, saving the two new variables in the output file. The variable names are created by prepending "prov_" to the original variable names. For example, "east" becomes "prov_east". The units for the velocity and the sample rate must be determined properly in order to convert to kilometers. The program attempts to determine the units from the input file information, but if it is unable to find the proper information, it will prompt the user to enter a conversion factor. For a more detailed progressive vector tool with annotation and wind sticks, see "provecepic".

unescoepic

usage:

```
unescoepic inputfile outputfile [badvalue]
```

inputfile = an EPIC netCDF file.

outputfile = an output EPIC netCDF file

badvalue = default is the netCDF fill value (FILL_FLOAT)

subroutines:

epicutils EPIC utility routines

millard.f Equation of state subroutines

This tool allow the user to calculate various properties of sea water using the equation of state FORTRAN subroutines. A menu of possible calculations is available:

- 1) Calculate salinity from T,C (SAL78)
 - 2) Calculate conductivity from T,S (SAL78)
 - 3) Calculate potential temperature from T,S (or C) (THETA)
 - 4) Calculate sigma-t from T, S (or C) (pressure = 0) (SVAN)
 - 5) Calculate sigmatheta from potential T (or T), S (or C)
(pressure = 0) (SVAN)
 - 6) Calculate density from T,S (or C) ,P (SVAN)
- Enter selection or 'q' to quit :

The user is asked to identify the fundamental variables in the input file used for the calculations:

Identify input variables for calculation

(x=skip,t=temp,s=sal,c=cond,o=potential temp)

All calculations at minimum require the presence of temperature and conductivity. If salinity is present in the input file, it can be selected instead of conductivity.

The menu repeats until the user is done. Variables calculated in previous loops are written to the output file, but are NOT available as input variables for successive calculations. For example, if the input file contains temperature and conductivity, and you want salinity and sigma-t, you must select T and C as the fundamental variables

for both calculations. The salinity calculation will be repeated for the calculation of σ_t .

Special care had to be taken when compiling this tool, since different platforms handled the C-FORTRAN calling conventions differently. Conditional compile code was used to handle the specific cases for SUN and SGI.

zmerge

usage:

zmerge

subroutines:

epicutils EPIC utility routines

This tool reads in multiple time series files at discrete depths, and combines them into a single file that is two dimensional in time and depth. This is a necessary step for producing contour plots using PPLUS. The user is prompted for the output file name and each input file name.

The input files must have the same sample interval and be evenly spaced in time. The tool will select the nearest matching times from the input file. If times are offset between files, use gridepic to create files with samples centered at the same times.

If more than one variable is being processed from each file, then they must appear in the same order in each successive file. For example, if file 1 contains east, north and temperature in that order, then file 2 must also contain east, north and temperature in that order. If file 2 contains north, east and temperature, then the resultant output file will inappropriately mix the east and north components.

The user enters a start and stop time to be used for every file if desired. For each file, the user must enter the input path and file name, the variables to select, and the depth bin, or depth record to extract from the input file (usually "1"). File names must be entered in order of increasing (or decreasing) depth, since the output file will place the first file in depth bin "1," the second file in depth bin "2," etc.

VI Examples

A. NetCDF header from a simple time series file

An example listing of the header portion of an EPIC netCDF VMCM file created using the "ncdump" utility:

```
netcdf vmcm0051t2 {
dimensions:
    time = UNLIMITED ; // (59305 currently)
    depth = 1 ;
    lat = 1 ;
    lon = 1 ;

variables:
    float lon(lon) ;
        lon:units = "degrees" ;
        lon:epic_code = 501 ;
    float lat(lat) ;
        lat:units = "degrees" ;
        lat:epic_code = 500 ;
    float depth(depth) ;
        depth:units = "m" ;
        depth:epic_code = 1 ;
    long time(time) ;
        time:units = "days" ;
        time:epic_code = 624 ;
        time:reset_yr = 1992s ;
        time:reset_mo = 10s ;
        time:reset_da = 1s ;
        time:reset_hr = 19s ;
        time:reset_mn = 15s ;
        time:reset_sc = 0s ;
        time:rec_period = 225.f ;
    long time2(time) ;
        time2:units = "milliseconds" ;
        time2:epic_code = 624 ;
    float north(time, depth, lat, lon) ;
        north:long_name = "north" ;
        north:units = "cm/second" ;
    float east(time, depth, lat, lon) ;
```

```
    east:long_name = "east" ;
    east:units = "cm/second" ;
float dir(time, depth, lat, lon) ;
    dir:long_name = "direction" ;
    dir:units = "degrees" ;
float temp(time, depth, lat, lon) ;
    temp:long_name = "temperature" ;
    temp:units = "degrees C" ;
    temp:therm# = 114s ;
    temp:acal = 0.000909369518 ;
    temp:bcac = 0.000224941802 ;
    temp:ccal = 1.15206759e-07 ;
```

```
// global attributes:
```

```
    :compliance = "EPIC" ;
    :experiment = "TOGA COARE" ;
    :platform = "Mooring 933" ;
    :instrument_type = "vmcm - OSU" ;
    :instrument_number = "0051" ;
```

```
}
```

B. NetCDF header from a two-dimensional file

An example listing of the header portion of an EPIC netCDF 2-dimensional temperature file created using the tool "zmerge":

```
netcdf toga2dtemp {
dimensions:
  time = UNLIMITED ; // (12871 currently)
  depth = 40 ;
  lat = 1 ;
  lon = 1 ;

variables:
  float lon(lon) ;
    lon:units = "degrees" ;
    lon:epic_code = 501 ;
  float lat(lat) ;
    lat:units = "degrees" ;
    lat:epic_code = 500 ;
  float depth(depth) ;
    depth:units = "m" ;
    depth:epic_code = 1 ;
  long time(time) ;
    time:units = "days" ;
    time:epic_code = 624 ;
    time:reset_yr = 1992s ;
    time:reset_mo = 10s ;
    time:reset_da = 5s ;
    time:reset_hr = 7s ;
    time:reset_mn = 0s ;
    time:reset_sc = 0s ;
    time:rec_period = 900.f ;
  long time2(time) ;
    time2:units = "milliseconds" ;
    time2:epic_code = 624 ;
  float temp(time, depth, lat, lon) ;
    temp:long_name = "temperature" ;
    temp:units = "degrees C" ;
    temp:A = 0.00146012233 ;
    temp:B = 0.00023975905 ;
    temp:C = 9.42286434e-08 ;
```

```

temp:D = 3641.19 ;
temp:E = -0.576623 ;
temp:F = 1.53601e-05 ;
temp:G = -4.24191e-10 ;
temp:R25 = 10.f ;
temp:R26 = 2.f ;
temp:R28 = 18.200001f ;
temp:R29 = 2.21f ;
temp:R31 = 47.5f ;
temp:transform = "bias(rec)= A+B*r+C*r*r
                  A = -0.03 B = 0 C = 0 D = 0" ;

// global attributes:
:compliance = "EPIC" ;
:experiment = "TOGA COARE" ;
:platform = "mooring 933" ;
:instrument_type = "tpod" ;
:instrument_number = "3667" ;
:title = "cal TPOD data" ;
:history = " " ;
:comments = "Creating 2D file of surface temperatures" ;
:Thu Apr 1 10:50:27 1993 = "tpod_cdf " ;
:Fri Aug 20 10:52:07 1993 = "putinepic tpod3667.cal tpod3667.epic" ;
:Wed Aug 25 14:12:45 1993 = "timeedit tpod3667.epic tpod3667t.epic" ;
:Wed Jan 19 14:26:29 1994 = "transform tpod3667t.epic tpod3667t2.epic
    badval 9.96921e+36" ;
:Tue Jan 25 11:06:17 1994 = "zmerge ../tpod/tpod3667t2.epic
    toga2dtemp.epic" ;
}

```

C. Using the toolbox to put a VAWR file in EPIC

```
putinepic vawr0722.cal vawr0722.epic
opening input cdf file vawr0722.cal
opening output cdf file vawr0722.epic
opening output log file vawr0722.epic.log
```

Select variables (time skipped here, handled elsewhere):

```
variable # 0      wnde (y,n) : y
variable # 1      wndn (y,n) : y
variable # 2      spd  (y,n) : y
variable # 3      dir  (y,n) : y
variable # 4      rotor (y,n) : n
variable # 5      compass (y,n) : n
variable # 6      vane  (y,n) : n
variable # 7      interval (y,n) : n
variable # 8      srads (y,n) : y
variable # 9      lrad  (y,n) : y
variable # 10     hrh   (y,n) : y
variable # 11     bpr   (y,n) : y
variable # 12     stmp  (y,n) : y
variable # 13     atmp  (y,n) : y
variable # 14     time      : skipping, not a float
```

The following are global attributes from input netcdf file vawr0722.cal

```
title VAWR cal data
VAWR # VAWR number 0722
history vawr_cdf - Thu Feb 13 15:27:09 1992
vawr_cal - Wed Jul 22 11:31:01 1992
```

Using the global attributes information from above, answer the following questions (for OARS/BOAT compliance):

Enter global attribute for experiment (i.e. subduction 1):

Subduction 1

Enter global attribute for platform (i.e. ship, mooring):

Central

Enter global attribute for instrument type (i.e. vawr, vmcm):

vawr

Enter global attribute for instrument #:

0722

Enter comments you want to appear in the output file:

Putting in EPIC, doing min,max detection

Thanks ... you won't have to enter that stuff again.

Do you want to truncate the time series (y,n)?: y

How do you want to enter time (by record=1, calendar=2, secofyr=3): 2

enter start date (yr mo dy hr mn): 91 6 23 0 26

enter end date (yr mo dy hr mn): 92 2 11 11 30

working ... looking for start and stop records

putinepic ... processing input file records: start = 1586 end = 23963

Reset year from input file 1991 (hope it's the same for start of file)

putinepic ... do you want to detect and repair outliers (y,n)? : y

Enter min,max for wnde (ret for none) : -20,20
min = -20.000000, max = 20.000000

Enter min,max for wndn (ret for none) : -20,20
min = -20.000000, max = 20.000000

Enter min,max for spd (ret for none) : 0,35
min = 0.000000, max = 35.000000

Enter min,max for dir (ret for none) : 0,360
min = 0.000000, max = 360.000000

Enter min,max for srad (ret for none) : 0,1200
min = 0.000000, max = 1200.000000

Enter min,max for lrad (ret for none) :
min = -inf, max = inf

Enter min,max for hrh (ret for none) : 40,100
min = 40.000000, max = 100.000000

Enter min,max for bpr (ret for none) : 1000,1040
min = 1000.000000, max = 1040.000000

Enter min,max for stmp (ret for none) : 20,28
min = 20.000000, max = 28.000000

Enter min,max for atmp (ret for none) : 15,30
min = 15.000000, max = 30.000000

Enter longitude in degrees (+ = E) : -28.9528

Enter latitude in degrees (+ = N) : 25.5317

Enter Depth of instrument (meters) : 0.

working ...

D. Create an evenly spaced time series

```
timecheck vawr0722.epic vawr0722t.epic
```

```
opening input cdf file epictools/vawr0722.epic
```

```
opening output cdf file vawr0722t.epic
```

```
opening output log file vawr0722t.epic.log
```

Select variables to process:

```
variable # 5      wnde (y,n) : y
variable # 6      wndn (y,n) : y
variable # 7      spd  (y,n) : y
variable # 8      dir  (y,n) : y
variable # 9      srad (y,n) : y
variable # 10     lrad (y,n) : y
variable # 11     hrh  (y,n) : y
variable # 12     bpr  (y,n) : y
variable # 13     stmp (y,n) : y
variable # 14     atmp (y,n) : y
```

Sample rate found ... 900.000000

Enter comments you want to appear in the output file:

making evenly spaced time series

Input file number of time records = 22454

```
start 06/22/1991 23:52:30
```

```
end 02/12/1992 05:52:30
```

Do you want to truncate the time series (y,n)?: n

timecheck ... processing input file records:

```
start rec = 1 06/22/1991 23:52:30
```

```
end rec = 22454 02/12/1992 05:52:30
```

working ...

E. Calculate cloud-cover estimate

Before doing this, you should either have an atc value already, or you should already have selected day(s) when the sky is clear for the atc estimate to be made.

Step 1. Calculate the theoretical cloud cover

```
atc_cloud vawr0722t.epic cloud0722.epic
```

```
opening input cdf file epictools/vawr0722t.epic
```

```
opening output epic file cloud0722.epic
```

```
opening output log file cloud0722.epic.log
```

```
Input file number of time records = 22489
```

```
start 06/22/1991 23:52:30
```

```
end 02/12/1992 05:52:30
```

```
Sample rate found ... 900.000000
```

```
Do you want to truncate the time series (y,n)?: n
```

```
Output file processing records
```

```
start rec = 1 06/22/1991 23:52:30
```

```
end rec = 22489 02/12/1992 05:52:30
```

```
Do you want to change the SINA cutoff (default = 0.15) (y,n)? : y
```

```
Enter new cutoff value : 0.10
```

```
Using SINA cutoff = 0.100000
```

```
This program will calculate atc values on days of your choice.
```

```
Do you want to enter a value instead (y,n)? : n
```

```
Enter date when skies are clear for atc calculation (ret when done)
```

```
1 local day (yr mo dy) : 91 7 7
```

```
local solar noon 7/ 7 1991 SWobs=1032.313
```

```
SWtheo(ground)=1286.722 SWobs/SWtheo= 0.802
```

```
2 local day (yr mo dy) : 91 8 18
```

```
local solar noon 8/18 1991 SWobs= 994.206
```

```
SWtheo(ground)=1269.408 SWobs/SWtheo= 0.783
```

```
3 local day (yr mo dy) : 91 9 8
```

```
local solar noon 9/ 8 1991 SWobs= 909.851
```

```
SWtheo(ground)=1231.832 SWobs/SWtheo= 0.739
```

```
4 local day (yr mo dy) : 92 1 12
```

```
local solar noon 1/12 1992 SWobs= 708.736
```

SWtheo(ground)= 932.012 SWobs/SWtheo= 0.760
5 local day (yr mo dy) :
Enter new value for atc : 0.70
Atmospheric transmission coefficient = 0.700000

This program will calculate SW night-time bias.

Do you want to enter a value instead (y,n)? : n

local night 06/23 1991 bias 5.404644
local night 06/24 1991 bias 5.562189
local night 06/25 1991 bias 5.616915
local night 06/26 1991 bias 5.594606
local night 06/27 1991 bias 5.491275

...
...

local night 02/10 1992 bias 6.663414
local night 02/11 1992 bias 6.895904
local night 02/12 1992 bias 7.361335

MEAN night time bias 5.993520 standard dev = 1.772336

Enter value for SW night-time bias : 5.9935

SW night_time bias = 5.993500

Enter comments you want to appear in this output file:

calculating intermediate flux files, constants

working ...

Step 2. Filter the cloud-cover estimate

fltepic cloud0722.epic cloud0722f.epic

opening input cdf file cloud0722.epic
opening output epic file cloud0722f.epic
no bad value specified

Do you want to filter time series or profiles (t/p) : t

Selected filtering time series

Select variables to filter (y/n)

5 - NAME "swadj"	TYPE	float	DIMENSIONS	4	:	n
6 - NAME "clrsky"	TYPE	float	DIMENSIONS	4	:	n
7 - NAME "cloud"	TYPE	float	DIMENSIONS	4	:	y

Sample rate found ... 900.000000

Input file number of time records = 22489

start 06/22/1991 23:52:30

end 02/12/1992 05:52:30

Do you want to truncate the time series (y,n)?: n

Output file processing records

start rec = 1 06/22/1991 23:52:30

end rec = 22489 02/12/1992 05:52:30

Number of depth bins= 1 start 0.000 end 0.000

Enter start,end bins # to extract (0 - 0) :0,0

Starting and ending depths 0.000 0.000

Enter number points to filter over: 119 (should be ~30 hours)

Enter number points to decimate over: 1

Enter comment you want to appear in this output file:

filtering cloud-cover estimate

Number of total in output file = 22370

*****Filter time series*****

F. Calculate heat fluxes, stress

fluxepic vawr0722.epic flux0722.epic

opening input cdf file vawr0722.epic
opening output epic file flux0722.epic
opening output log file flux0722.epic.log

Input file number of time records = 22454
start 06/22/1991 23:52:30
end 02/12/1992 05:52:30

Sample rate found ... 900.000000

Do you want to truncate the time series (y,n)?: n

Output file processing records

start rec = 1 06/22/1991 23:52:30
end rec = 22454 02/12/1992 05:52:30

Enter EPIC file name for corresponding calculated cloud:
cloud0722f.epic

opening input cdf file cloud0722f.epic
Cloud file sample rate found ... 900.000000

Do you want to use SW rad from input file vawr0722.epic (y,n)?

('no' means use adjusted SW from output of the atc_cloud): n

Enter EPIC file name for corresponding adjusted SW rad: cloud0722.epic

opening input cdf file cloud.epic for adjusted short wave rad

Enter atmospheric transmission coefficient: 0.70

Atmospheric Transmission coefficient ATC = 0.700000

Enter SW night time bias: 0.0

Short Wave Radiation night time bias = 0.000000

Enter height of wind measurement (meters): 10

Enter height of air temperature measurement (meters): 10

Enter height of relative humidity measurement (meters): 10

Enter comments you want to appear in this output file:

calculating fluxes

G. Using the toolbox to grid and average a time series file

gridepic vmcm0051t2.epic vmcm0051t2.grid

opening input cdf file vmcm0051t2.epic

opening output cdf file vmcm0051t2.grid

opening output log file vmcm0051t2.grid.log

no bad value specified

Select variables to write to the output file (y/n)

5	- NAME "	north"	TYPE	float	DIMENSIONS	4	:	y
6	- NAME "	east"	TYPE	float	DIMENSIONS	4	:	y
7	- NAME "	spd"	TYPE	float	DIMENSIONS	4	:	n
8	- NAME "	dir"	TYPE	float	DIMENSIONS	4	:	n
9	- NAME "	rotor2"	TYPE	float	DIMENSIONS	4	:	n
10	- NAME "	rotor1"	TYPE	float	DIMENSIONS	4	:	n
11	- NAME "	compass"	TYPE	float	DIMENSIONS	4	:	n
12	- NAME "	temp"	TYPE	float	DIMENSIONS	4	:	y

Sample rate found ... 225.000000

Input file number of time records = 59305

start 10/01/1992 19:28:07

end 03/05/1993 05:58:07

Do you want to truncate the time series (y,n) ?: n

Output file processing records

start rec = 1 10/01/1992 19:28:07

end rec = 59305 03/05/1993 05:58:07

Number of depth bins= 1 start 17.010 end 17.010

Enter start,end bins # to extract (1 - 1) :1,1

Starting and ending records 1 1

Starting and ending depths 17.010 17.010

Enter the new start time, date (yr mo dy hr mn sc): 92 10 20 0 4 22.5

Enter the new end time, date (yr mo dy hr mn sc): 93 3 5 0 15 37.5

Enter the new sample rate (seconds): 225

New start time (mid-point of average) 10/20/1992 00:04:22.5

New end time (mid-point of average) 03/05/1993 00:15:37.5

New sample rate is ... 225.000000 seconds

52227 total samples in gridded file

Enter comment you want to appear in this output file:

Regridding temp so that averaging by 4 matches TPOD sample interval
Reading time from input file ... done
Creating new gridded time ... done
Regridding variable 5 north ... done
Regridding variable 6 east ... done
Regridding variable 12 temp ... done

fltepic vmcm0051t2.grid vmcm0051t2.filt

opening input cdf file vmcm0051t2.grid
opening output epic file vmcm0051t2.filt
no bad value specified

Do you want to filter time series or profiles (t/p) : t

Selected filtering time series

Select variables to filter (y/n)

5 - NAME " north"	TYPE	float	DIMENSIONS	4 : y
6 - NAME " east"	TYPE	float	DIMENSIONS	4 : y
7 - NAME " temp"	TYPE	float	DIMENSIONS	4 : y

Sample rate found ... 225.000000 (seconds)

Input file number of time records = 52227

start 10/20/1992 00:04:22

end 03/05/1993 00:11:52

Do you want to truncate the time series (y,n)?: y

How do you want to enter time (by record=1, calendar=2): 2

enter start date (yr mo dy hr mn): 92 10 22 0 4 30

enter end date (yr mo dy hr mn): 93 3 4 0 30

Output file processing records

start rec = 768 10/22/1992 00:00:37

end rec = 51847 03/04/1993 00:26:52

Number of depth bins= 1 start 17.010 end 17.010

Enter start,end bins # to extract (1 - 1) :1,1

Starting and ending records 1 1

Starting and ending depths 17.010 17.010

Enter number points to filter over: 4

Enter number points to decimate over: 4

New sample rate (seconds) 900.000000

Enter comment you want to appear in this output file:

Averaging to 15 minutes
Number of points in output file = 12769

*****Filter time series*****
Record 0 ... done

VII Miscellaneous

A. The Plot Plus Scientific Graphics System (PPLUS)

Plot Plus (PPLUS) is a scientific plotting package written by Donald W. Denbo. The package is available for Woods Hole Oceanographic Institution employees via a site license arrangement. The current version is 1.2c. The package is designed to be run on UNIX based workstations. It has been run and tested on Sun, Silicon Graphics Incorporated and Masscomp.

The plotting package contains an EPIC netCDF interface which allows files generated from the tools in this package to be read and plotted directly. See the PPLUS manual for instructions on interfacing to EPIC netCDF files, as well as general use instructions.

B. The mexcdf MATLAB interface

A general netCDF interface called "mexcdf" for the MATLAB computation and visualization package was written by Charles Denham under the direction of Richard Signell, USGS, at Woods Hole. They have generously made the interface available to the community.

The mexcdf interface mimics the calls available in the netCDF library for reading and writing data in netCDF format. "M" files are also available that combine these routines to access netCDF EPIC files conveniently. Once the arrays are loaded into memory, the numeric power of MATAB is available for further computations or visualization.

C. The AVS netCDF interface

A netCDF EPIC interface is available from Richard Signell, USGS at Woods Hole for the Advanced Visual Systems (AVS) graphics package. AVS is a powerful 3-D visualization tool which can produce animation and complicated 3-D images.

D. The Explorer netCDF interface

Explorer is a Silicon Graphics Incorporated 3-D visualization packaged. A general netCDF interface developed at the National Center for Supercomputing

Applications at the University of Illinois at Urbana-Champaign and at the NASA-Goddard Space Flight Center, Greenbelt, MD. This interface is able to access the netCDF files produced from this toolbox directly.

References

- Clark, N. E., L. Eber, R. M. Laurs, J. A. Renner, and J. F. T. Saur, 1974. Heat exchange between ocean and atmosphere in the eastern North Pacific for 1961-71. NOAA Tech. Rep. NMFS SSRF-682, National Oceanic and Atmospheric Administration, U. S. Dept. of Commerce, Washington, D.C.
- Danforth, T. W., 1990. The Seadata Program. Woods Hole Oceanographic Institution Technical Report, WHOI-90-44, 73 pp.
- Denbo, D. W., and W. H. Zhu, 1993. EPS Library Users Guide Version 2.1, Pacific Marine Environmental Laboratory, National Oceanic and Atmospheric Administration, 131 pp.
- Prada, K. E., 1992. A System for Shipboard Analysis of Buoy Data. Woods Hole Oceanographic Institution Technical Report, WHOI-92-42, 29 pp.
- Rew, R., G. Davis, and S. Emmerson, 1993. NetCDF Users Guide, An Interface for Data Access, Version 2.3, Unidata Program Center, 186 pp.
- Trask, Richard P. and Nancy J. Brink, 1993. The Subduction Experiment, cruise report, R/V *Oceanus* cruise number 240, leg 3, Subduction 1 mooring deployment cruise 17 June-5 July 1991. *Woods Hole Oceanographic Technical Report*, WHOI-93-12, 77 p.

Acknowledgements

The author wishes to thank members of the Upper Ocean Processes Group for discussions and feedback during the development of the software. Discussions with N. Galbraith and R. Samelson during the preparation of this manuscript are gratefully acknowledged.

DOCUMENT LIBRARY

Distribution List for Technical Report Exchange – May 5, 1994

University of California, San Diego
SIO Library 0175C (TRC)
9500 Gilman Drive
La Jolla, CA 92093-0175

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

Commander
International Ice Patrol
1082 Shennecossett Road
Groton, CT 06340-6095

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149

Library
Skidaway Institute of Oceanography
10 Ocean Science Circle
Savannah, GA 31411

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

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
Building 1003 South
1002 Balch Blvd.
Stennis Space Center, MS 39522-5001

Library
Institute of Ocean Sciences
P.O. Box 6000
Sidney, B.C. V8L 4B2
CANADA

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
Birkenhead
Merseyside L43 7 RA
UNITED KINGDOM

IFREMER
Centre de Brest
Service Documentation - Publications
BP 70 29280 PLOUZANE
FRANCE

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-94-11	2 UOP-94-2	3. Recipient's Accession No.
4. Title and Subtitle EPIC Processing Toolbox Users Guide			5. Report Date April 1994
7. Author(s) Melora Park Samelson			6
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543			8. Performing Organization Rept. No. WHOI-94-11
12. Sponsoring Organization Name and Address Office of Naval Research			10. Project/Task/Work Unit No.
15. Supplementary Notes This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept., WHOI-94-11.			11. Contract(C) or Grant(G) No. (C) N00014-90-J-1495 (G)
16. Abstract (Limit: 200 words) A set of computer programs for data analysis is described and documented. The programs were developed to aid in the processing of time-series data collected by the Upper Ocean Processes Group. These programs, or tools, utilize a platform independent format known as the Network Common Data Format (netCDF). The format is further defined using a convention known as EPIC which allows easy access to other plotting and manipulation tools available to the UOP group. The general method for implementation of the toolbox is described. A reference section lists the available tools, as well as examples and detailed descriptions of their usage and functionality. The toolbox has been tested and used extensively on both Sun and Silicon Graphics Interface UNIX workstations.			13. Type of Report & Period Covered Technical Report
17. Document Analysis			14.
a. Descriptors data analysis time series data NetCDF			
b. Identifiers/Open-Ended Terms			
c. COSATI Field/Group			
18. Availability Statement Approved for public release; distribution unlimited.		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 68
		20. Security Class (This Page)	22. Price