

**West Coast Observing System (WCOS)
Project Document
Phase I**

Prepared for the

National Marine Sanctuaries

by:

The National Coastal Data Development Center
With
The Partnership for Interdisciplinary Studies of Coasts and Oceans
And
The National Oceanographic Data Center

March 17, 2006

Table of Contents

1.0	EXECUTIVE SUMMARY	1
2.0	INTRODUCTION	3
3.0	TECHNICAL APPROACH	4
3.1	OVERVIEW	4
3.2	SYSTEM SPECIFICATIONS	5
4.0	PROCESS DESCRIPTIONS	6
4.1	DATA COLLECTION AND INITIAL PROCESSING	6
4.1.1	<i>PISCO</i>	7
4.1.2	<i>Olympic Coast</i>	8
4.2	WCOS-NODC CONDUIT	9
4.3	NCDDC INFORMATION BROKER	10
4.3.1	<i>Metadata Transformation</i>	11
4.3.1.1	Transform EML to FGDC XML	11
4.3.1.2	Post Transformation Clean-up	12
4.3.1.3	Transform FGDC XML to FGDC HTML and FGDC Text	12
4.3.2	<i>Metadata Publishing</i>	12
4.3.2.1	Oracle	12
4.3.2.2	NSDI	12
4.3.2.3	MERMAid	13
4.4	NODC RETRIEVING, PUBLISHING AND ARCHIVING	13
4.4.1	<i>Retrieving and Publishing</i>	13
4.4.2	<i>Archiving</i>	14
4.5	DATA PRODUCTS WEB APPLICATION	14
4.5.1	<i>WCOS-Data Search</i>	14
4.5.2	<i>WCOS-Plots and Statistics</i>	16
4.6	INTERNET MAP SERVICE	21
4.6.1	<i>SIMoN Enhancements and Standards Compliance</i>	21
4.6.2	<i>WCOS Internet Map Service</i>	22
5.0	REFERENCES	25
5.1	NOSP INTERNAL DOCUMENTS	25
5.2	PISCO INTERNAL DOCUMENTS	25
5.3	INTERNET REFERENCES	25
5.3.1	<i>Tool Resources</i>	25
5.3.2	<i>WCOS Specific Internet Resources</i>	26
6.0	APPENDIX A: DATA COLLECTION AND INITIAL PROCESSING	27
6.1	PISCO ADCP DATA FORMAT	27
6.2	PISCO TEMPERATURE DATA FORMAT	31
6.3	PISCO ADCP METADATA FORMAT	33
6.4	PISCO TEMPERATURE METADATA FORMAT	47
6.5	PISCO NAMING SCHEME	55
6.6	PISCO ADCP PROTOCOL	58
6.7	PISCO ADCP DATA PROCESSING	64
6.8	PISCO TEMPERATURE DATA PROCESSING	68
6.9	PISCO TIDBIT TEMPERATURE LOGGER PROTOCOL	70
7.0	APPENDIX B: WCOS-NODC CONDUIT	74
8.0	APPENDIX C: NCDDC INFORMATION BROKER	76
9.0	APPENDIX D: DATA PRODUCTS WEB APPLICATION	79

List of Figures

Figure 1 NMS Locations and Equipment Types..... 2
 Figure 2 Data Flow 4
 Figure 3 PISCO Nearshore Mooring System..... 7
 Figure 4 PISCO Process Flow 8
 Figure 5 WCOS Conduit Process Diagram 10
 Figure 6 NCDDC Information Broker Summary 11
 Figure 7 WCOS Data Search 15
 Figure 8 Queried Datasets..... 16
 Figure 9 WCOS-Plots and Statistics Page 17
 Figure 10 Time Series Plot for Temperature 18
 Figure 11 Stick Plot for ADCP 19
 Figure 12 Box and Whisker Plot for ADCP 20
 Figure 13 Temperature Box and Whisker Plot 20
 Figure 14 Statistics for ADCP 21
 Figure 15 Data Download for ADCP..... 21
 Figure 16 Compliant Map Selection..... 22
 Figure 17 IMS Custom Functionality 23
 Figure 18 WCOS-NODC Conduit Configuration File Example 75
 Figure 19 Webapp Parameters Example File..... 80

List of Tables

Table 1 WCOS System and Software Requirements..... 5
 Table 2 Sanctuary Specifics..... 6
 Table 3 Available Data Types..... 17
 Table 4 Mapping from EML to FGDC..... 76

Tables supporting the PISCO documentation are located in the following sections of Appendix A.

EML field relevant to bottom-mounted ADCPs.....Section 6.3
 EML field relevant to moored temperatures.....Section 6.4
 Fields included in Matlab Structuring for ADCP.....Section 6.7
 Fields included in Matlab Structuring for Temperature.....Section 6.8

Glossary

ADCP	Acoustic Doppler Current Profile
AMS	Archive Management System
API	Application Program Interface
ASCII	American Standard Code for Information Interchange
CeNCOOS	Central & Northern California Ocean Observing System
COARDS	Cooperative Ocean/Atmosphere Research Data Service
CNIDR	Clearinghouse for Networked Information Discovery and Retrieval
cron	Unix tool used to run executables at scheduled intervals (chronologically)
DMAC	Data Management and Communication
EML	Ecological Metadata Language
FGDC	Federal Geographic Data Committee
FTP	File Transfer Protocol
HTTP	Hypertext Transport Protocol
IMS	Internet Map Service
IOOS	Integrated Ocean Observing System
IPT	Integrated Product Team
MERMAid	Metadata Enterprise Resource Management Aid
NCDDC	National Coastal Data Development Center
netCDF	Network Common Data Form
NMS	National Marine Sanctuaries
NMSP	National Marine Sanctuaries Program
NODC	National Oceanographic Data Center
NSDI	National Spatial Data Infrastructure
OPeNDAP	Open-source Project for a Network Data Access Protocol
PISCO	Partnership for Interdisciplinary Studies of Coastal Oceans
RPC	Remote Procedure Call
SDE	Spatial Data Engine
SEA	Sanctuary Environmental Assessment
SIMoN	Sanctuary Integrated Monitoring Network
SWiM	System-Wide Monitoring
WCOS	West Coast Observing System
webapp	Website application
XML	Extensible Markup Language
XSLT	XML Stylesheet Transformation engine

1.0 EXECUTIVE SUMMARY

The National Marine Sanctuaries Program (NMSP) has adopted a system-wide approach to resource monitoring that includes local, regional and national scales of observation. This System-Wide Monitoring Program (SWiM) provides a framework for monitoring in marine sanctuaries. Working in cooperation with key partners, multiple sites have been selected within regions to address questions and concerns relevant to comparatively large areas of the ocean and coasts. At all scales, the primary intent is to benefit from and contribute to existing and proposed observing networks, particularly the Integrated Ocean Observing System (IOOS). Long-range plans are for the NMSP regional networks to contribute critical information and become part of the regional backbone of IOOS.

The West Coast Observation System (WCOS) project was designed to establish an immediate and active presence for the NMSP within the regional structure of the IOOS proposed for the west coast of the United States. During Fiscal Year 2005, twenty-three new Sanctuary Environmental Assessment (SEA) stations were established in SWiM regions to supplement the existing observational array supported by other entities.

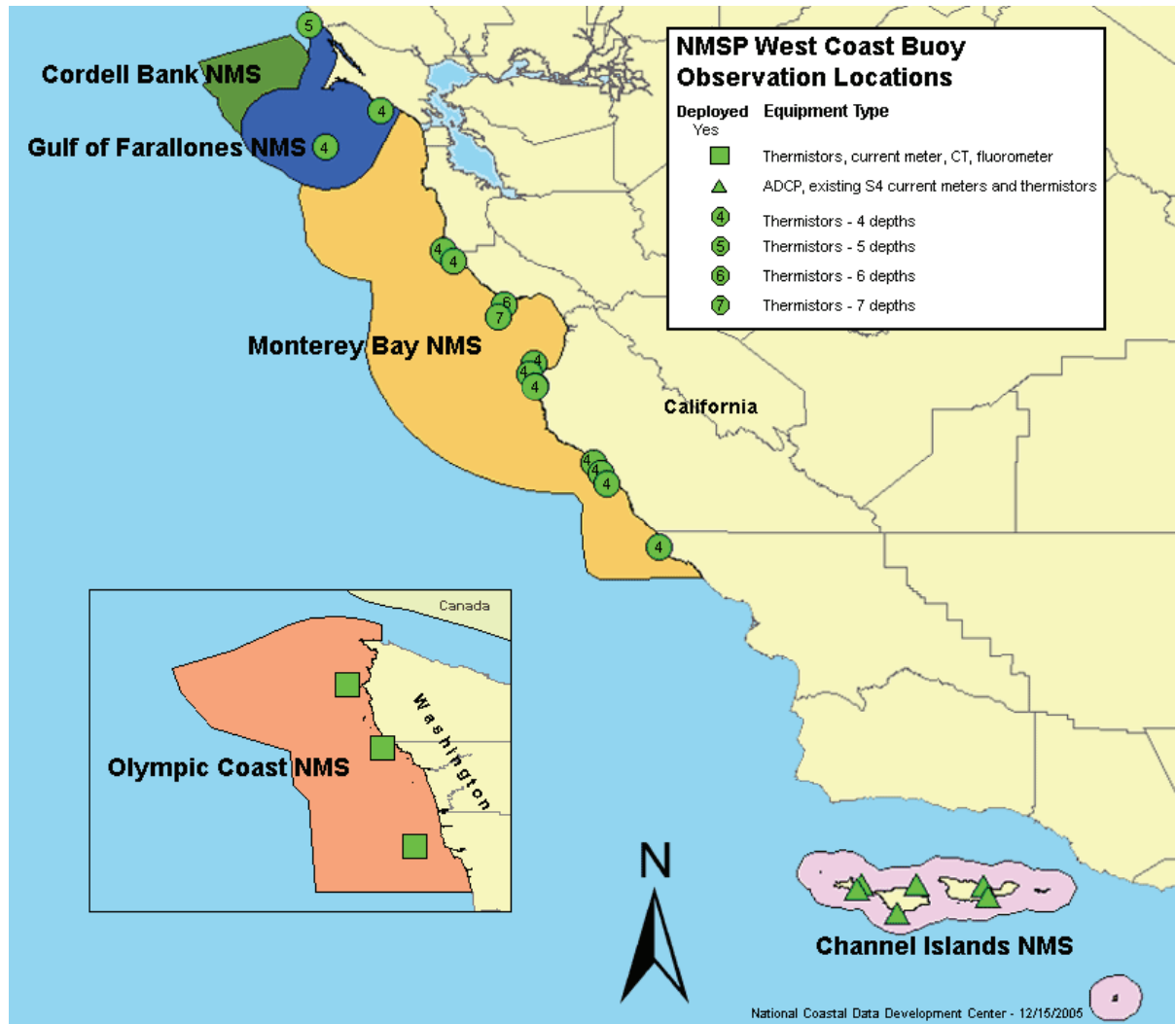


Figure 1 NMS Locations and Equipment Types

The NOAA Satellite and Information Service is providing end-to-end data management services to assist data providers in making this observed information available to constituents in accordance with the IOOS Data Management and Communication (DMAC) Subsystem definitions. Within the scope of the WCOS project, FGDC metadata are created to support data description and discovery; data are accessible from client applications in accordance with OPeNDAP standards; data are visualized and evaluated through common web browsers and map services and data are archived for perpetuity.

This document provides an overview of the WCOS project, as well as detailed technical information to support ongoing execution of project components.

2.0 INTRODUCTION

During July 2004, WCOS project participants met at the Moss Landing Marine Laboratories in Moss Landing, CA. At this meeting, the WCOS project primary objectives were identified as follows:

- To develop a workflow/system to make west coast National Marine Sanctuaries Program (NMSP) monitoring data accessible via the internet in an Integrated Ocean Observing System (IOOS) compatible format, and
- To extend the role and capabilities of the Sanctuary Integrated Monitoring Network (SIMoN).

To achieve the objectives, a project Work Plan was drafted. Subsequently, an Integrated Product Team (IPT) and working groups were formed among the partners to carry out the work plan. WCOS project partners represented at Moss Landing are listed below:

- National Oceanic and Atmospheric Administration (NOAA)
 - National Marine Sanctuaries
 - Monterey Bay NMS
 - Gulf of Farallones NMS
 - Channel Islands NMS
 - Olympic Coast NMS
 - National Oceanographic Data Center
 - National Coastal Data Development Center
 - Northwest Fisheries Science Center
 - Coastal Services Center
 - Office of Response and Restoration
- Sanctuary Integrated Monitoring Network (SIMoN)
- Partnership for Interdisciplinary Studies of Coastal Oceans (PISCO)
- Pacific Coastal Observing System
- Central & Northern California Ocean Observing System (CeNCOOS)

3.0 TECHNICAL APPROACH

3.1 OVERVIEW

The work plan defined a workflow that is designed to include six related processes. The project is initiated through the purchase and deployment of data collection instruments. This event occurs just once during the project lifecycle. After this, a regular workflow has been established to continue beyond the life of this project. Each process is outlined in this section and detailed in Section 4.0. Figure 2 provides an overview of the data flow through each of the six processes.

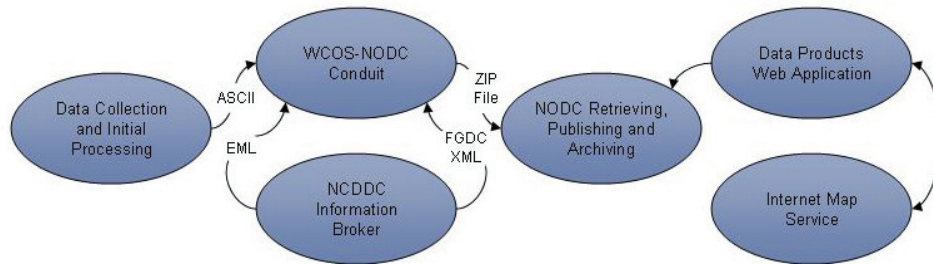


Figure 2 Data Flow

1. Data Collection and Initial Processing: PISCO led the effort to collect and QC data from the project sensors;
2. WCOS-NODC Conduit: NCDDC led the software development effort to transport and transform the data and metadata;
3. NCDDC Information Broker: NCDDC led the software development effort to transform and publish the metadata;
4. NODC Retrieving, Publishing and Archiving: NODC led the effort to publish the data and to archive the data and metadata;
5. Data Products Web Application: NCDDC led the software effort to develop an Internet application to discover, preview and download WCOS data; and
6. Internet Map Service: NCDDC developed an Internet Map Service to make the WCOS data accessible via a GIS application.

3.2 SYSTEM SPECIFICATIONS

Table 1 provides the system and software requirements for the WCOS processes that are described in detail in Section 4.0.

Table 1 WCOS System and Software Requirements

	System	Software
Data Collection and Initial Processing	Windows XP, Linux Server	<ul style="list-style-type: none"> ▪ Boxcar Pro v.4.3.1.1 for thermistors ▪ WinSC Application v.1.29 for ADCPs ▪ Matlab (v.7 or higher)
WCOS-NODC Conduit	UNIX Platform	<ul style="list-style-type: none"> ▪ Java 5.0 (Java 1.5)
NCDDC Information Broker	UNIX Platform	<ul style="list-style-type: none"> ▪ Python 2.3 ▪ cx_Oracle library ▪ libxsltmod library ▪ libxml2 library ▪ libxslt library ▪ Oracle client support libraries
NODC Retrieving, Publishing and Archiving	UNIX Platform	<ul style="list-style-type: none"> ▪ OPeNDAP netCDF server ▪ PostgreSQL ▪ Perl and various Perl modules from CPAN, including Getopt::Std, Digest::MD5, Date::Calc, and File::Temp. ▪ Local Perl modules developed by NODC, including ArMD5, Messenger, and NOAAlib.
Data Products Web Application	UNIX Platform	<ul style="list-style-type: none"> ▪ Java 5.0 (Java 1.5), ▪ Java 5.0 enabled servlet container (Apache Tomcat 5.5 recommended)
NCDDC Internet Map Service	Linux Server	<ul style="list-style-type: none"> ▪ ArcIMS 4.x, customized with HTML and javascript ▪ ArcSDE 8.3, and Oracle 9.2.0.6

4.0 PROCESS DESCRIPTIONS

As outlined in Section 2.0, the WCOS workflow system follows six processes for guiding the data transfer process from the raw data collection stage, through Internet delivery of standardized, IOOS compatible data and metadata, to final archiving at NODC. These steps are described within Sections 4.1 - 4.6 and technical details are provided in the Appendices. Close collaboration between PISCO, NCDDC, and NODC was required to build this system, which is designed to be automated to the fullest extent possible.

1. Data are collected quarterly from each of the WCOS moorings and processed to an ASCII file format. The data collection, initial processing and Quality Control, and metadata creation tasks are performed by the servicing agency.
2. NCDDC acquires the processed data and metadata from the servicing agency, transforms the ASCII data to a COARDS profile netCDF format (IOOS standard format), and transforms the EML metadata to FGDC XML format. During this step, the FGDC metadata are published to a National Spatial Data Infrastructure (NSDI) node. The completed netCDF data and FGDC metadata files are packaged in a ZIP file and placed on the NCDDC FTP server for retrieval by NODC.
3. NODC retrieves the Zip package from NCDDC, publishes the netCDF data files to an OPeNDAP server for public access, and periodically archives all the data and FGDC XML metadata files.
4. The WCOS Data Products Web Application provides a public search engine for locating WCOS data on the Internet. The tool provides the capability to preview data using plots and statistics, and to download the data in ASCII format.
5. The WCOS Internet Map Service displays the locations of the WCOS moorings in conjunction with other observed data. Links in the map display provide direct access to the metadata data files.

4.1 DATA COLLECTION AND INITIAL PROCESSING

During the Data Collection phase of this process, a team of divers service each mooring and collect the data modules. The modules are brought to the office for data downloading and initial processing, and then placed back into the mooring during a future dive.

As shown in Figure 1, 23 new moorings were placed into four of the five west coast NMS within the scope of the WCOS project. Table 2 provides an overview of how data collection and initial processing tasks are handled for moorings within each Sanctuary.

Table 2 Sanctuary Specifics

Sanctuary	Number of Moorings	Servicing Agency	Processing Agency	Data Served By
Channel Islands NMS (CINMS)	6	PISCO	PISCO	PISCO
Gulf of Farallones NMS (GFNMS)	3	GFNMS/Bodega Marine Lab	GFNMS/Bodega Marine Lab	GFNMS/Bodega Marine Lab
Monterey Bay NMS (MBNMS)	11	PISCO	PISCO	PISCO
Olympic Coast NMS (OCNMS)	3	OCNMS	OCNMS	

The raw data are processed to ASCII format and an EML metadata record is created to describe the ASCII formatted data. Format standards have been developed for each type of data. Sections 4.1.1 and 4.1.2 describe the collection and initial processing as it is performed by the servicing agencies.

4.1.1 PISCO

PISCO services the moorings (See Figure 3 for an example of the PISCO Mooring System), gathers the data and performs the initial data processing for the seventeen moorings funded by the WCOS project that fall within the Channel Islands NMS and the Monterey Bay NMS. PISCO's Data Collection and Initial Processing are performed at three month intervals. After processing, the data are submitted to Metacat, PISCO's online data catalog. *Note:* The mooring design and data processing may differ slightly between sanctuaries. See Appendix A for PISCO process details.

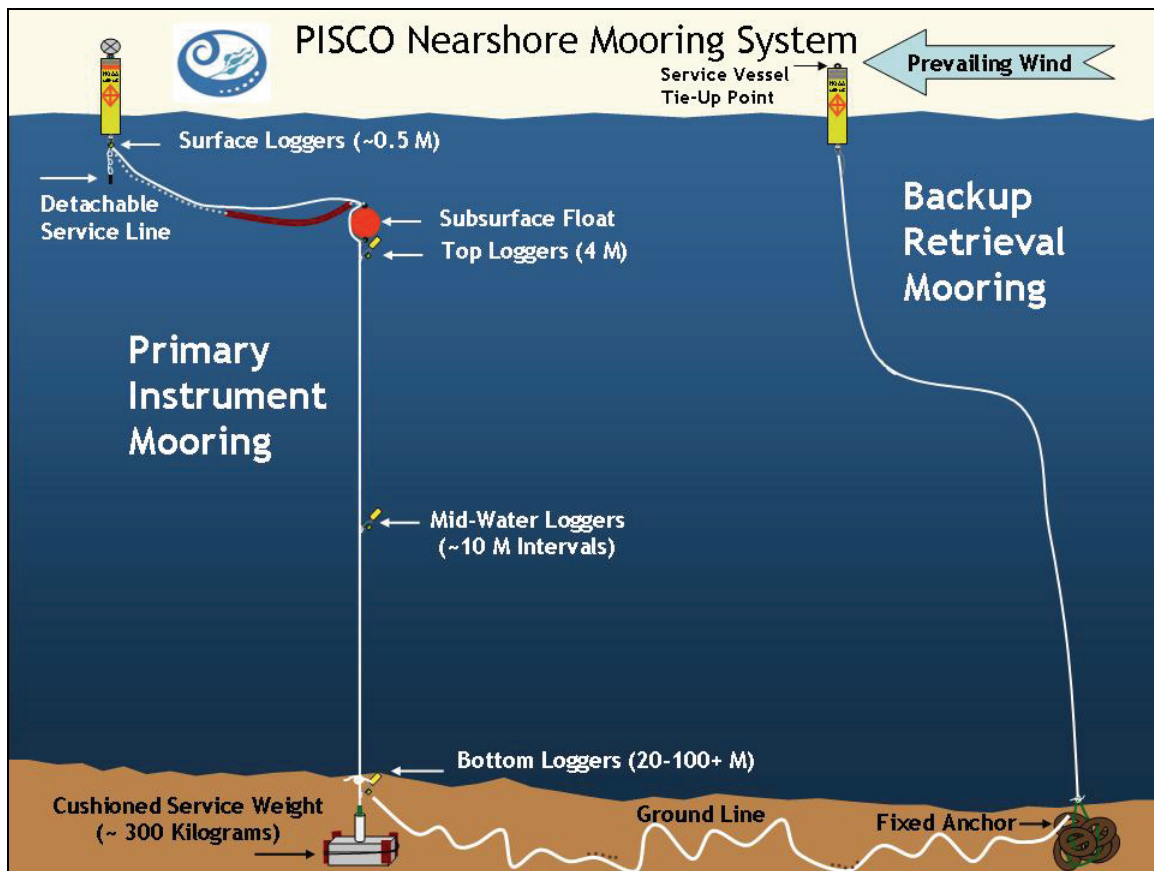


Figure 3 PISCO Nearshore Mooring System

All PISCO/NMSP moorings are in relatively shallow water (100 m or less, with most in approximately 20 or 15 m). All are equipped with a vertical array of thermistors (Onset

Stowaway Tidbits and XTIs). Channel Island NMS moorings are additionally equipped with bottom-mounted ADCPs (RD Instruments 600 kHz Workhorse Sentinels).

During servicing, the instruments are replaced with fresh replicates and then returned to the laboratory. Once onshore, data files are downloaded from the instruments using the proprietary software of the instrument manufacturer (Boxcar Pro v.4.3.1.1 for thermistors and WinSC Application v.1.29 for ADCPs). The thermistors are programmed with a two-minute sampling interval (no internal averaging) and eventually redeployed. ADCPs are programmed with continuous two-minute ensembles (45 pings per ensemble) and a vertical resolution of one meter. See the documents “PISCO ADCP Protocol.doc” and “PISCO UCSB Tidbit Temperature Logger Protocol.doc,” located in Sections 6.6 and 6.9 respectively, for more details on using the instruments.

Metadata for both temperature and ADCP datasets are created for the data using EML, allowing great flexibility, but translating the metadata content into a structure and organization appropriate for biological datasets. This structure is hierarchical and requires the metadata to be stored in specific fields, many of which exist nested within other fields. Each metadata file will describe one ASCII data file. The content of the metadata are described in the documents ADCP Metadata Format (Section 6.3) and Temperature Metadata Format (Section 6.4).

ASCII data files and EML metadata files are then uploaded to PISCO’s Metacat data catalog for harvesting by NCDDC. Automated uploading and metadata creation are accomplished with specialty Matlab software developed by PISCO. Figure 4 outlines the PISCO portion of the Data Collection and Initial Processing Process.

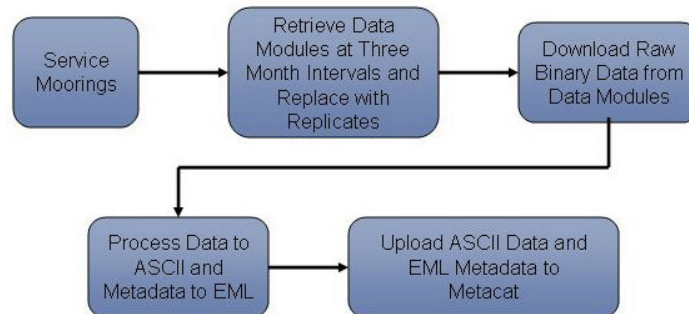


Figure 4 PISCO Process Flow

4.1.2 Olympic Coast

Olympic Coast collects Temperature, ADCP, Dissolved Oxygen, and Chlorophyll data. The datasets are collected on a monthly basis for the six month period of May through October each year. The Olympic Coast processing techniques differ from PISCO’s. However, PISCO and Olympic Coast are using the same ASCII output data formats for both Temperature and ADCP datasets. Project partners have developed a similar ASCII output format for the Dissolved Oxygen and Fluorometry datasets.

Olympic Coast houses the data in a Microsoft Access database, and does not currently have a mechanism for publicly serving their data. Project partners are developing a resolution which will facilitate integration of Olympic Coast collected data into the overall WCOS process flow. The proposed solution is to have PISCO host the data through Metacat; the final outcome and timeline for implementation of this or another resolution is TBD. Details of the OCNMS processing steps will be added at a future date as the details are resolved and the data are added into the system for distribution.

4.2 WCOS-NODC CONDUIT

The WCOS-NODC Conduit is a Java application developed at NCDDC to access the WCOS data and metadata served by Metacat, transform the WCOS data to IOOS standards, transform the metadata from EML to FGDC, package the transformed data and metadata into ZIP files, and serve the files for NODC to retrieve. The list below outlines the steps taken by the Conduit when it is executed (see Figure 5).

1. Check Metacat (against a stored local catalog) for new datasets/metadata.
2. Download each new dataset (via HTTP). The datasets may be located at different sites.
3. Download the associated EML metadata (via HTTP). The metadata may be located at different sites and are not necessarily collocated with the datasets they describe.
4. Convert the new dataset from its native ASCII format to a COARDS profile netCDF file.
5. Call the NCDDC Information Broker to convert and publish the metadata. Section 4.3 explains the process used to convert/publish the metadata. Once the metadata conversion is complete, the file is returned to the Conduit.
6. Create MD5 checksum files for the converted data and metadata files. As a security precaution, MD5 files are used to ensure that the downloaded files match the files that are on the server where they have been downloaded.
7. Package all converted files into ZIP files for transfer. For each instrument deployment, the Conduit creates a ZIP file, which contains the data and metadata files for that deployment and their associated MD5 files.
8. Create an MD5 file for each ZIP file.
9. Transfer all the ZIP files (and their associated MD5 files) to NCDDC's FTP server where they are picked up by NODC.
10. Delete temporary files that are used by the Conduit for processing. Store the ZIP and MD5 files on the FTP server for three weeks. At the end of the three weeks, a cron job deletes them.

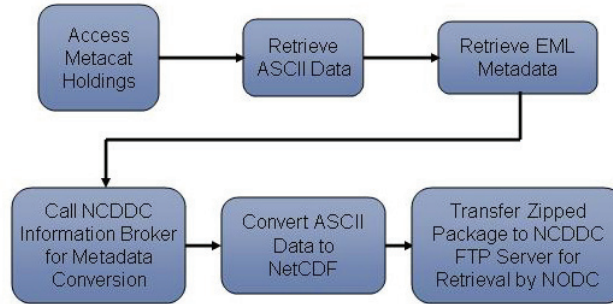


Figure 5 WCOS Conduit Process Diagram

The WCOS-NODC Conduit processes one data type at a time (Temperature or ADCP); therefore, it must be run twice in order to process all new data. To automate the data management process, NCDDC uses a cron job (a scheduling method in Unix) to run the scripts daily. Once the cron job is initiated, the script file runs the application twice to process both the Temperature and ADCP datasets. See Appendix B for details regarding the WCOS-NODC Conduit Process.

4.3 NCDDC INFORMATION BROKER

The NCDDC Information Broker, which is written in the Python programming language and forms a part of NCDDC's data services infrastructure, converts the EML metadata file created by the data collection agency to the standard FGDC Biological profile in XML format. The Broker also publishes the FGDC record to the NCDDC metadata catalog and the NSDI Clearinghouse node. After the metadata have been converted and published, it is returned to the WCOS-NODC Conduit where it is packaged in a ZIP file with the data and posted to the NCDDC FTP site for retrieval by NODC (see Section 4.4). Figure 6 provides a summary of the steps the NCDDC Information Broker takes to transform and publish the metadata.

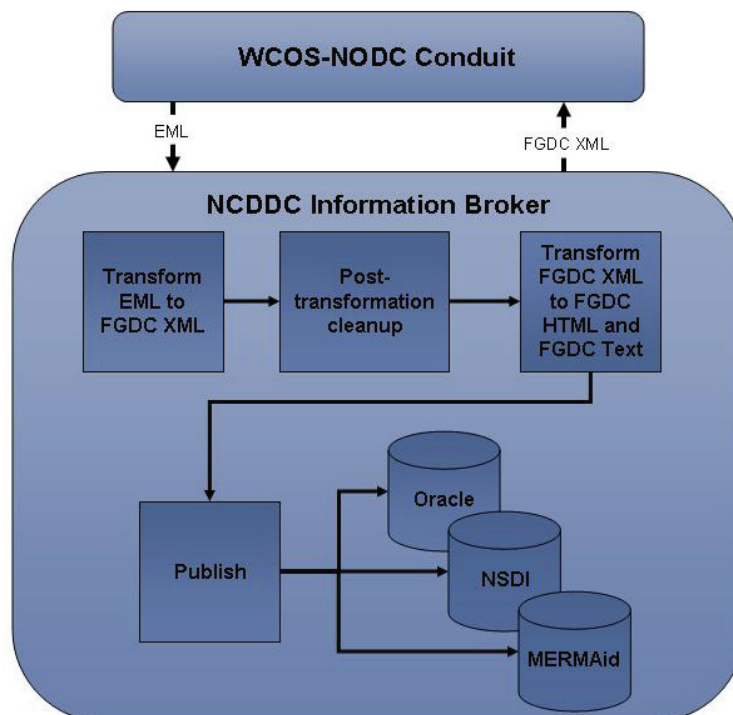


Figure 6 NCDDC Information Broker Summary

4.3.1 Metadata Transformation

The metadata transformation includes the first four steps of the Information Broker portion of Figure 6. Due to the conciseness of the steps, the description of the conversions to FGDC HTML and Text have been combined in Section 4.3.1.3.

4.3.1.1 Transform EML to FGDC XML

Metadata describing the WCOS datasets are originally expressed in Ecological Metadata Language, version 2.0.1. To publish these metadata into the NSDI Clearinghouse node and NCDDC's metadata catalog, they must first be converted to FGDC's metadata format. The FGDC Biological Profile is used to most closely align with the EML contents. This transformation is accomplished by ingesting the EML metadata into an XML Stylesheet Transformation (XSLT) engine, followed by some post transformation cleanup.

XSL transformations describe mappings from one set of metadata components, or terms, to another. For the WCOS metadata, the terms in the EML metadata are mapped to FGDC Biological Profile terms. The mapping of each term from EML to FGDC is described in Table 4 in Appendix C (Section 8.0).

While the majority of the FGDC XML metadata are derivable from the EML content, some modifications are made to reflect that NODC, not PISCO, will be the distributor of the data. Specifically, PISCO's contact information as the data originator is replaced with standard NCDDC contact information, and contact information as the data distributor is replaced with

NODC's standard contact information. Additionally, a Data Quality Information section is added to the FGDC XML metadata to describe both the transformation of the dataset from ASCII to netCDF and the transformation of the metadata from EML to FGDC XML.

4.3.1.2 Post Transformation Clean-up

Although XSLT does provide for some logical operations and control flow, a post transformation step is necessary to complete the transformation from EML to FGDC XML. These operations, summarized in the following list, are not within the current, inherent capabilities of the XSLT engine and are performed after the transformation by custom software code developed and operational within the NCDDC Information Broker:

- Modify EML date and time values to conform to FGDC's accepted formats (PISCO delimits date-time elements with embedded slashes and hyphens which are not valid in FGDC);
- Using the current system time, insert date and time values into the Process Date, Process Time, Metadata Date, and Metadata Future Review Date elements;
- Insert a value for the Transfer Size element (provided by the WCOS-NODC Conduit); and
- Insert values for Online Linkage elements (provided by the WCOS-NODC Conduit).
- Insert an Entity and Attributes Information section to describe the netCDF file variables using values provided by the WCOS-NODC Conduit.

4.3.1.3 Transform FGDC XML to FGDC HTML and FGDC Text

Finally, the complete FGDC XML metadata record is run through the XSLT again to generate metadata records in HTML and FGDC Text formats to be published to NCDDC's metadata catalogs as described in Section 4.3.2.

4.3.2 Metadata Publishing

After transformation, the resulting FGDC XML metadata are "published" to NCDDC's metadata catalog, NCDDC's NSDI Clearinghouse node, and MERMAid (NCDDC's metadata management tool). The purpose of publication is to facilitate public Internet data search and discovery of as well as access to the data through multiple sources.

4.3.2.1 Oracle

NCDDC's metadata catalog is an Oracle relational database that contains a specific, searchable subset of FGDC metadata elements. These elements are extracted from the FGDC XML metadata record and written to Oracle using a Python implementation of the Oracle Application Program Interface. The outcome of this step allows a user to search and discover the metadata/data through the NCDDC data portal.

4.3.2.2 NSDI

The Clearinghouse node is implemented using the Clearinghouse for Networked Information Discovery and Retrieval's (CNIDR) Isite software, which consists, in part, of an indexer application and a server. The indexer, which is run daily, builds an index of the current metadata holdings to facilitate rapid searches of the metadata. This publishing step involves writing the XML, HTML, and Text metadata files to the specific location on the Unix file

system used by the Isite indexer. The indexer builds its index files using the XML metadata files as input and result sets from queries on the NSDI search engine contain the HTML metadata files.

Publishing in this manner supports a wide distribution of the metadata throughout the system supported by the FGDC, in keeping with the Federal Government's eGov initiative.

4.3.2.3 MERMAid

Finally, the FGDC XML metadata are pushed to MERMAid, which ingests the FGDC XML and creates a record in its object database. This allows NCDDC managers to review, retract and/or edit the FGDC metadata record as required. After the transformation and publishing steps are complete, the transformed metadata record (in FGDC XML format) is returned to the WCOS-NODC Conduit application for packaging and delivery to NODC.

4.4 NODC RETRIEVING, PUBLISHING AND ARCHIVING

4.4.1 Retrieving and Publishing

A scheduled task, written in Perl, runs daily at NODC to retrieve the ZIP files from the NCDDC FTP server. The task is comprised of the following steps:

1. Daily, NODC downloads all new (not previously downloaded) ZIP files and their associated MD5 files from the NCDDC FTP server to the local mirror directory. This mirroring is synchronized to run soon after NCDDC's WCOS-NODC Conduit application finishes.
2. For each data and metadata file in the NODC mirror, if it is more than one year old, the file is removed. The files are then accessible through NODC's archive.
3. In order to avoid processing identical data, the MD5 checksum is calculated for each ZIP file. If the MD5 checksum matches the previous MD5 checksum known for the file, no further processing is performed on the file.
4. Otherwise, all files are extracted from the ZIP file. If any errors occur, no further processing is performed on the ZIP file. It is assumed that any errors in ZIP files will be corrected by a subsequent download. If the errors continue to occur over the course of a day or two, then an NODC Data Officer will assess the situation and, if there is a problem, notify NCDDC.
5. The unzipped files are processed. Each data and metadata file in the ZIP file is accompanied by a matching MD5 checksum file. If the MD5 checksum of any data or metadata file does not match the value in its accompanying MD5 checksum file, the entire ZIP file and all of the files it contains are skipped. If the file is in the NODC mirror directory, it will be retried. If a newer version of the file appears at NCDDC, it will be downloaded to the NODC mirror and the newer version of the file will be processed.

The processed netCDF data and FGDC XML metadata files (along with their associated MD5 checksum files) are immediately available via the following three NODC servers:

1. OPeNDAP (<http://data.nodc.noaa.gov/cgi-bin/nph-dods/nmsp/wcos/>; server registered at OPeNDAP.org),
2. HTTP (<http://data.nodc.noaa.gov/nmsp/wcos/>), and
3. FTP (<ftp://data.nodc.noaa.gov/pub/nmsp/wcos/>).

4.4.2 Archiving

The entire WCOS dataset is archived as a new version each calendar year (including off-line and off-site tape backups) with NODC's Archive Management System (AMS). The AMS consists of a PostgreSQL relational database for tracking datasets, a digital file management system, and a public interface to discover and retrieve archival data. These AMS components are bound together with a suite of Perl scripts.

Using the AMS, an NODC data manager assigns an accession number to each collection of data that NODC archives. Accession number 0002039 is assigned to the WCOS dataset. Once a year as a new version under this accession, an NODC Data Officer will archive:

1. The most recent version of all the WCOS netCDF data and FGDC XML metadata files;
2. An ASCII dump of each netCDF data file;
3. Digital documentation relevant to the interpretation of these data (e.g., this document); and
4. MD5 checksums for each file archived.

The archival WCOS data are discoverable online through the Ocean Archive System, the public interface to the AMS. In addition, each archival data file, if it is the latest version of that data file, is available via NODC's OPeNDAP server.

4.5 DATA PRODUCTS WEB APPLICATION

The WCOS Data Products Web Application (webapp) serves two purposes when dealing with the WCOS data served via OPeNDAP at NODC. The first purpose is to provide a search engine that allows the user to locate specific WCOS data using selected search criteria. This provides an enhanced experience for the public end-user, who would normally only see a list of OPeNDAP files in a directory structure format. The second purpose is to provide the user with the ability to preview data via plots and statistics, and to download the selected dataset in entirety or subset by user specified criteria.

The WCOS Data Products webapp consists of two main web pages: WCOS-Data Search and WCOS-Plots and Statistics. The webapp is located at the following URL: <http://portal.ncddc.noaa.gov/wco/>.






4.5.1 WCOS-Data Search



To complete a search for NMSP Temperature or ADCP data using the search functions and tools on the WCOS-Data Search web page, the user may select a mooring site, year, and data type (or use the defaults) and then click on the "Submit Query" button (see Figure 7). This selection returns a list of all corresponding datasets on the NODC OPeNDAP server that


meets these criteria (see Figure 8). The following information is available for each dataset: start date, end date, instrument depth, file name, a link to the OPeNDAP URL, a link to the IMS Map, and a Plots/Statistics icon. For each dataset, the user may create plots and statistics and/or download the dataset or a subset of the dataset by clicking on the Plots/Statistics icon (see Section 4.5.2).

Step 1: Select mooring site(s)
 Check a point on the map below OR select from the list of sites. Your selected to query will appear in a list below.

Clickable Map of Sites

zoom in 
 zoom out 
 pan 
 default view 
 select site 

Mooring Legend
 unselected 
 selected 



List of Sites

(to select multiple shift+click)

- San Miguel North (BAY)
- Santa Rosa North (BEA)
- MOR
- Pelican/ Prisoners Area (PEL)
- PRO
- Southeast Farallon Island (SEF001)
- SHB001
- San Miguel South (SMS)

[Click here](#) to view a static map of all the sites.

Step 2: Select year and data type
 (to select multiple shift+click)

Year: 2001, 2003, 2004, 2005

Data Type: ADCP, Temperature

Submit Query

Figure 7 WCOS Data Search

Query Results					
Site	Depth (m)	Start Date	End Date	Data Type	Available Products
Pelican/ Prisoners Area (PEL)	3	2004/10/08 1810	2004/11/20 1558	Temperature	DODS Map Plots/Stats
Pelican/ Prisoners Area (PEL)	3	2004/11/23 2216	2005/01/07 1558	Temperature	DODS Map Plots/Stats
Pelican/ Prisoners Area (PEL)	9	2004/10/08 1810	2004/11/20 1558	Temperature	DODS Map Plots/Stats
Pelican/ Prisoners Area (PEL)	9	2004/11/23 2210	2005/01/07 1558	Temperature	DODS Map Plots/Stats
Pelican/ Prisoners Area (PEL)	14	2004/10/08 1810	2004/11/20 1558	Temperature	DODS Map Plots/Stats
Pelican/ Prisoners Area (PEL)	14	2004/11/23 2216	2005/01/07 1558	Temperature	DODS Map Plots/Stats
Southeast Farallon Island (SEF001)	4	2004/11/01 1940	2004/11/25 2350	Temperature	DODS Map Plots/Stats
Southeast Farallon Island (SEF001)	13	2004/11/01 1940	2004/11/25 2350	Temperature	DODS Map Plots/Stats
San Miguel South (SMS)	3	2004/10/11 2110	2004/11/22 2000	Temperature	DODS Map Plots/Stats
San Miguel South (SMS)	3	2004/11/22 2028	2005/01/06 1558	Temperature	DODS Map Plots/Stats
San Miguel South (SMS)	9	2004/10/11 2116	2004/11/22 1952	Temperature	DODS Map Plots/Stats
San Miguel South (SMS)	9	2004/11/22 2036	2005/01/06 1558	Temperature	DODS Map Plots/Stats
San Miguel South (SMS)	14	2004/10/11 2116	2004/11/22 1946	Temperature	DODS Map Plots/Stats

Figure 8 Queried Datasets

When the DODS link is selected, the webapp connects the user to a web page on the DODS server that corresponds to the queried dataset. This web page allows the user to view what parameters are available, view the attributes, and download the data in a different format than what is offered by the Plots/Stats page.

When the Map link is selected from the webapp, a new window is opened for the IMS service. The URL sent to open the IMS also includes commands to select a specific SEA Station Mooring and to zoom to that mooring (e.g., if a user queries the webapp for data only related to a mooring named BAY, the map link opens the IMS service and automatically zooms to the BAY mooring on the map). The WCOS IMS viewer graphically displays the location of the moorings deployed within the scope of the WCOS project, in correlation to other CeNCOOS activities and related datasets of interest.

4.5.2 WCOS-Plots and Statistics

By clicking the Plots/Statistics icon, the Plots and Statistics screen is opened (see Figure 9). This page provides a Download Data link for the user to retrieve the subset in ASCII format, and links to the available plots and statistics for the dataset.

The Plots, Statistics, and Download options can be adjusted based on the Start/End Date, Time Intervals, and (for ADCP) Depth. This option allows the user to subset the data for preview or download based upon the user-defined criteria. The data are averaged for hourly and daily Time Intervals. The Native Time Interval option does not average -- it displays the data in the original time interval for the dataset.

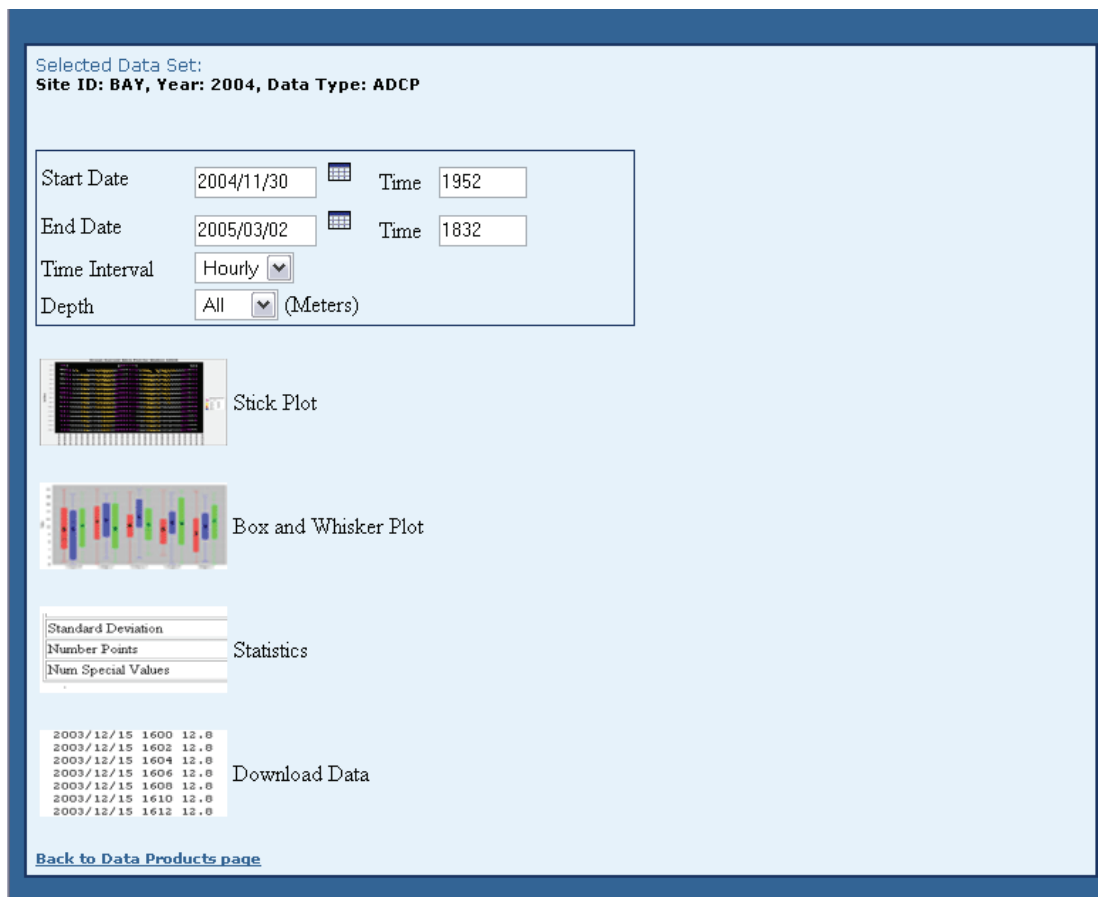


Figure 9 WCOS-Plots and Statistics Page

The variety of plot formats available is dependent on the data type chosen for the search. The plotting capability will be enhanced with the addition of Dissolved Oxygen and Chlorophyll data once OCNMS data are integrated into the system. Table 3 shows the current formats available for each data type.

Table 3 Available Data Types

	Statistics Page	Box and Whiskers Plot	Time Series Plot	Stick Plot
Temperature	X	X	X	
ADCP	X	X		X
Dissolved Oxygen				
Chlorophyll				

Figure 10 - Figure 15 provide examples of the products available through the webapp, using data taken from the Bay Station for the year 2005 with the default times and data types accepted. The Time Series plots are displayed with Temperature in the Y-axis and Time in the X-axis (Figure 10).

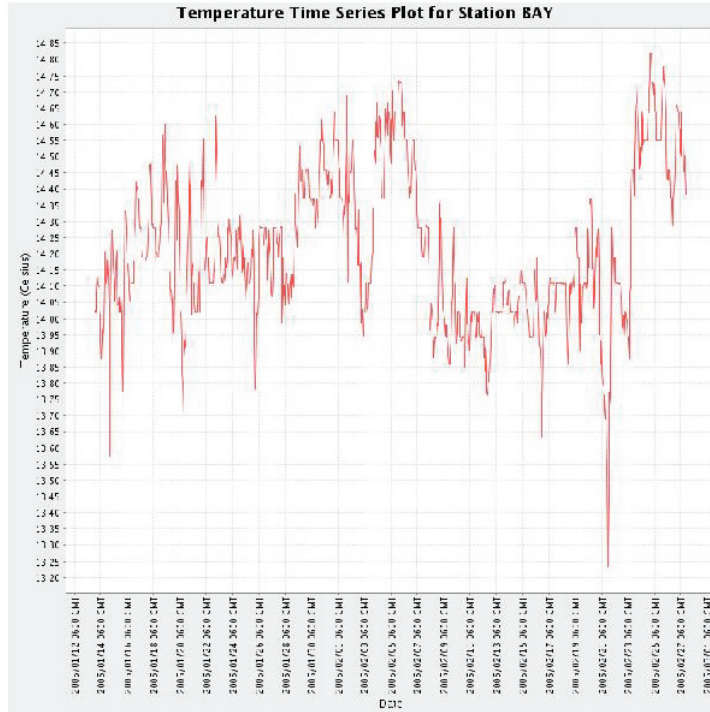


Figure 10 Time Series Plot for Temperature

The Stick plots display the current speed (color-coded) and direction at varying depths along the Y-axis and the time along the X-axis. Figure 11 provides an example of the Stick plot for ADCP.

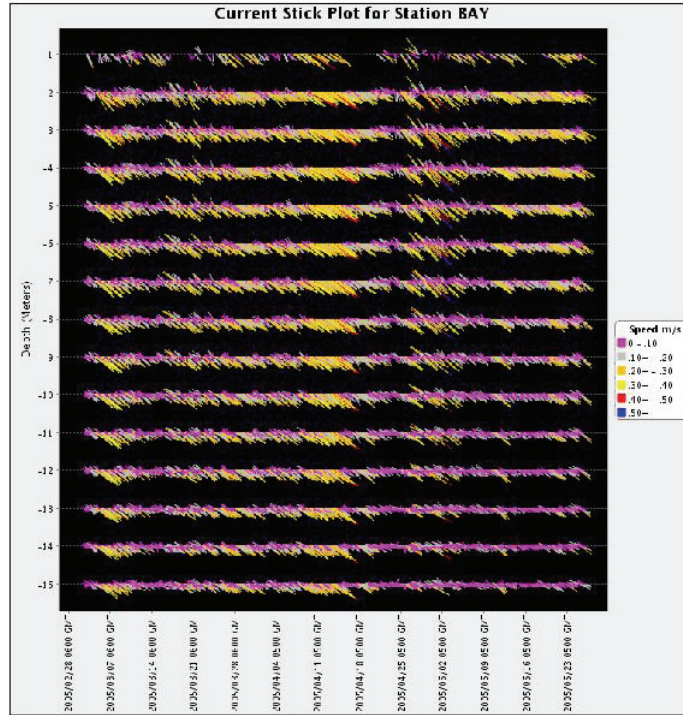


Figure 11 Stick Plot for ADCP

Box and Whisker Plots for ADCP are color-coded based on U, V, and Speed in the X-axis and m/sec in the Y-axis (Figure 12). Temperature Box and Whisker Plots provide the temperature in the Y-axis (Figure 13). The horizontal bar and dot in the middle of the box represent the median and the arithmetic mean, respectively. The symbols outside the plot (top and bottom) represent the outliers.

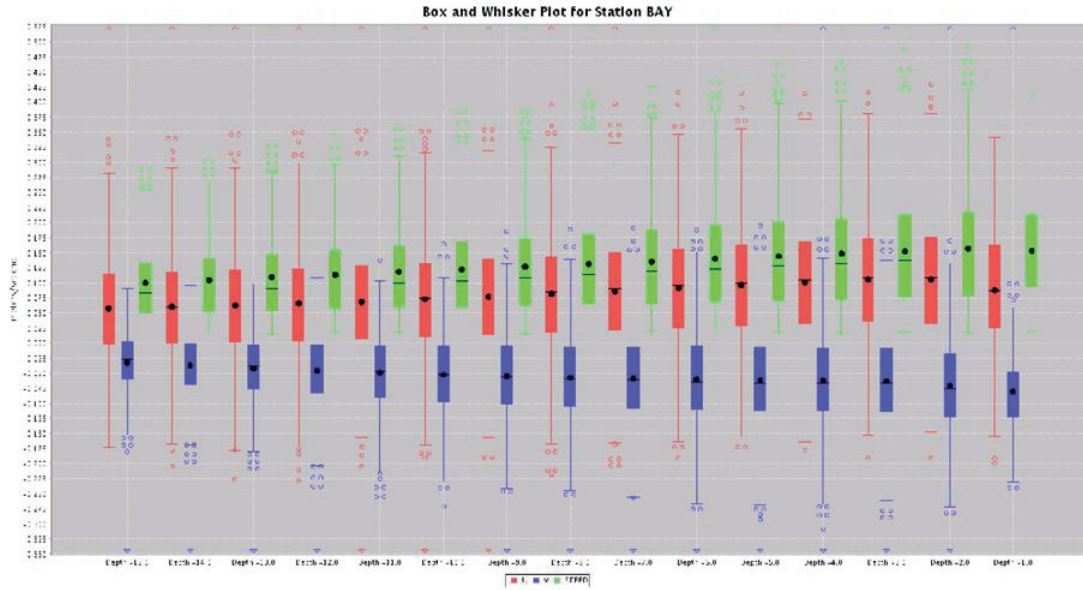


Figure 12 Box and Whisker Plot for ADCP

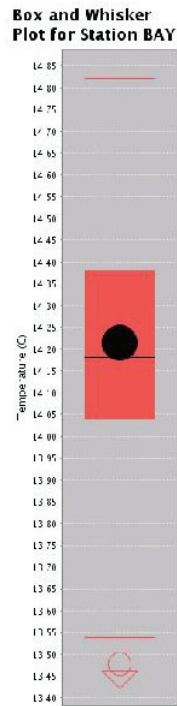


Figure 13 Temperature Box and Whisker Plot

Figure 14 and Figure 15 display examples of the statistics and downloadable data that are provided with each data search.

U Mean	0.08066012442998792
U Median	0.08126666666666668
U Standard Deviation	0.09325385811992776
V Mean	-0.05361872157777731
V Median	-0.051899999999999995
V Standard Deviation	0.06579946166973354
Number of Points	28265
No data points	307219

Figure 14 Statistics for ADCP

Date	Time	Depth	U	V	Speed	Direction
2005/03/02	1900	-15.00000	-0.00213	0.02707	0.02715	355.49339
2005/03/02	2000	-15.00000	0.03540	0.00720	0.03612	78.50344
2005/03/02	2100	-15.00000	0.06847	-0.01197	0.06950	99.91407
2005/03/02	2200	-15.00000	0.05443	-0.02560	0.06015	115.18760
2005/03/02	2300	-15.00000	0.04213	-0.04653	0.06277	137.84092
2005/03/03	0000	-15.00000	0.06603	-0.00853	0.06658	97.36339
2005/03/03	0100	-15.00000	0.01843	-0.05503	0.05804	161.48180
2005/03/03	0200	-15.00000	0.05083	0.00863	0.05156	80.36108
2005/03/03	0300	-15.00000	0.05657	-0.00563	0.05685	95.68719
2005/03/03	0400	-15.00000	0.03853	-0.02160	0.04417	119.27297
2005/03/03	0500	-15.00000	0.05367	-0.01097	0.05478	101.54926
2005/03/03	0600	-15.00000	0.04023	-0.00500	0.04054	97.08412
2005/03/03	0700	-15.00000	0.02917	-0.00240	0.02927	94.70403
2005/03/03	0800	-15.00000	0.04700	0.01017	0.04809	77.79429
2005/03/03	0900	-15.00000	-0.01027	0.00470	0.01129	294.59794
2005/03/03	1000	-15.00000	-0.06110	0.00723	0.06153	276.75155
2005/03/03	1100	-15.00000	-0.03767	0.04827	0.06122	322.03199
2005/03/03	1200	-15.00000	-0.02163	0.02347	0.03192	317.32782
2005/03/03	1300	-15.00000	0.01497	0.02650	0.03043	29.45689
2005/03/03	1400	-15.00000	0.03980	0.00140	0.03982	87.98540
2005/03/03	1500	-15.00000	0.05357	-0.01273	0.05506	103.37161
2005/03/03	1600	-15.00000	0.06923	-0.00993	0.06994	98.16486
2005/03/03	1700	-15.00000	0.08880	-0.03070	0.09396	109.07133
2005/03/03	1800	-15.00000	0.09160	-0.04543	0.10225	116.38128
2005/03/03	1900	-15.00000	0.12020	-0.04317	0.12772	109.75442
2005/03/03	2000	-15.00000	0.10110	-0.04987	0.11273	116.25441
2005/03/03	2100	-15.00000	0.10630	-0.07283	0.12886	124.41769
2005/03/03	2200	-15.00000	0.08333	-0.02593	0.08728	107.28614
2005/03/03	2300	-15.00000	0.07757	-0.06817	0.10326	131.30946
2005/03/04	0000	-15.00000	0.03910	-0.08787	0.09617	156.01130
2005/03/04	0100	-15.00000	0.05137	-0.09047	0.10403	150.41224
2005/03/04	0200	-15.00000	0.06310	-0.07510	0.09809	139.96258
2005/03/04	0300	-15.00000	0.06170	-0.07917	0.10037	142.06827
2005/03/04	0400	-15.00000	0.08320	-0.04063	0.09259	116.03002
2005/03/04	0500	-15.00000	0.08857	-0.01070	0.08921	96.88869
2005/03/04	0600	-15.00000	0.06987	0.03630	0.07873	62.54537

Figure 15 Data Download for ADCP

4.6 INTERNET MAP SERVICE

One of the two primary project objectives was to extend the roles and capabilities of the Sanctuary Integrated Monitoring Network (SIMoN) website. This section describes general functionality that was enhanced as well as specific WCOS related capabilities that were added to SIMoN.

4.6.1 SIMoN Enhancements and Standards Compliance

Partner efforts, led by NCDDC working closely with SIMoN staff, resulted in standardization of SIMoN’s data display capabilities, compliance with Federal standards for Internet accessibility, and cost-savings for NOAA as outlined below:


- a. Enhancement of two pre-existing SIMoN IMS viewers to ensure multi-browser compatibility as well as 508 Compliance;

- i. Habitat viewer (list URL)
- ii. Water Quality viewer (list URL)
- b. Integration of SIMoN geospatial data into NCDDC Oracle/SDE;
- c. Relocation of all IMS viewers from a commercially purchased server to the NCDDC-hosted server;
- d. Design, implementation and incorporation of the WCOS project web site pages into SIMoN; and
- e. Creation of a new IMS viewer for WCOS project Sanctuary Environmental Assessment (SEA) Station moorings (further described in Section 4.6.2). The new viewer provides the capability for integrated display.

All IMS Sites implemented 508 Compliancy Standards. To meet the 508 Compliancy Standards, two buttons were added to the HTML page linking to the IMS. The first allows the user to view a file that describes each data layer and the second enables the user to launch the map (see Figure 16).

Interactive Maps

WCOS Viewer



[MAP DESCRIPTION](#) [LAUNCH MAP VIEWER](#)

This Internet map application provides a common, geospatially referenced view of coastal and ocean observations in and around the West Coast National Marine Sanctuaries. Observation data are generally grouped into two categories: 1) near-real time observations; 2) long term observations. Near real time observations include marine and land surface weather data acquired from the Meteorological Assimilation Data Ingest System (MADIS). Data are assimilated into a geospatially enabled database on an hourly basis and stored for 48 hours. Long term observations include imagery products created for a three month period.

This map application requires Microsoft Internet Explorer 5.5 or greater, Netscape 6.0 or greater, or Mozilla Firefox. The client's browser must have Javascripting enabled, and be able to handle graphics and frames. This site does not request any internet downloads or plug-ins. Best viewed with a screen resolution of 1280 x 1024.

Figure 16 Compliant Map Selection

4.6.2 WCOS Internet Map Service

The purpose of the WCOS IMS viewer is to graphically display the location of the moorings deployed within the scope of the West Coast Observing System project, in correlation to other CeNCOOS activities and related datasets of interest. The map service itself is static, meaning that there is little that requires change once it is set up; however, there are some non-WCOS datasets within the map service that are setup on a real-time or near real-time update schedule. The Surface Marine Measurements layer, for example, updates data every hour.

The mooring locations shapefile was created from a spreadsheet of the locations, listing each mooring's latitude and longitude along with other information relating to the type of data collected and the servicing schedules. The latitude and longitude values serve to georeference the points on a map. Within the IMS, these mooring location points show the location of the moorings in reference to other observation sites and west coast communities, and provide the user with additional information about the sites. Direct access to the WCOS data files is available via a direct link from the IMS to the Data Products Web Application described in Section 4.5.

For the WCOS moorings, the data are collected and made available on a more long-term schedule, monthly to quarterly depending upon the collection schedule for the particular mooring. In place of having the actual data displayed on the map, which would be difficult for the user to interpret, a data layer showing the locations of the moorings was created. This data layer has a hyperlink to the webapp described in Section 4.5 of this document. These links are set once and are not changed unless the webapp URL changes.

Figure 17 is a screen capture of an example of the custom functionality of the IMS. In this example, a user moves the mouse over a mooring location on the map to display some of the information related to that mooring in the white display box to the left of the map. This capability is enabled for the SEA moorings as well as for the Surface Measurements. The surface measurements data can be displayed in both English and Metric measurements.

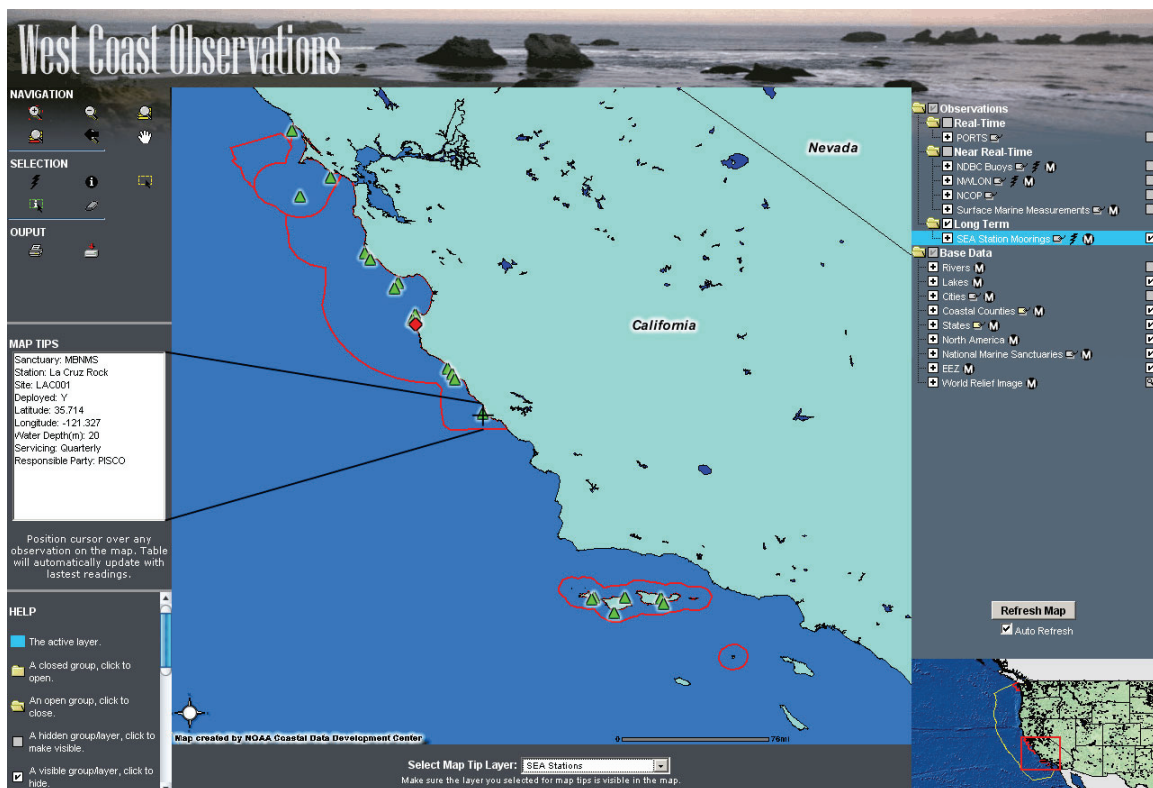


Figure 17 IMS Custom Functionality

The WCOS IMS is hosted by NCDDC and is accessible through the SIMoN website (see Section 5.3.2).

5.0 REFERENCES

5.1 NMSP INTERNAL DOCUMENTS

National Marine Sanctuaries Program (NMSP). *West Coast Observation Project – Definition/Workplan*. 2004.

National Marine Sanctuaries Program (NMSP). *Establishing a Regional Observing Network for National Marine Sancturaries Along the U.S. West Coast – DRAFT*. February 20, 2004.

5.2 PISCO INTERNAL DOCUMENTS

PISCO ADCP Protocol.doc

PISCO UCSB Tidbit Temperature Logger Protocol.doc

Final PISCO Physical Oceanographic Moored Temperature ASCII File Format Specifications, August 2005.

PISCO-Wide Physical Oceanographic Proposed Data and Metadata File Naming Scheme, September 2005.

Proposed PISCO Physical Oceanography Moored Temperature Data Processing Protocol, September 2005.

PISCO Physical Oceanographic Moored ADCP ASCII File Format Specifications, November 2005.

Proposed PISCO Physical Oceanography Moored ADCP Data Processing Protocol, August 2005.

PISCO moored temperature metadata content and format, August 2005.

PISCO bottom-mounted ADCP metadata content and format, November 2005.

5.3 INTERNET REFERENCES

5.3.1 Tool Resources

Visit the official web sites for more information about the following software and programs.

ArcIMS – <http://www.esri.com/software/arcgis/arcims/index.html>

EML – <http://knb.ecoinformatics.org/software/eml/>

Oracle – <http://www.oracle.com>

Perl – <http://www.cpan.org/>

PostgreSQL – <http://www.postgresql.org/>

SIMoN – <http://www.mbnms-simon.org>

Tomcat – <http://tomcat.apache.org>

XML – <http://www.w3.org/XML/>

5.3.2 WCOS Specific Internet Resources

CeNCOOS oceanObs - <http://portal.ncddc.noaa.gov/wco/>

IMS - http://www.ncddc.noaa.gov/website/NMSP_WCO

NCDDC Portal - http://portal.ncddc.noaa.gov/portal/jsp/search_options.jsp

NODC FTP server - <ftp://data.nodc.noaa.gov/pub/nmsp/wcos/>

NODC HTTP server - <http://data.nodc.noaa.gov/nmsp/wcos/>

NODC Ocean Archive System –

<http://www.nodc.noaa.gov/cgi-bin/search/prod/accessionsView.pl/details/2039>

NODC OPeNDAP Server - <http://data.nodc.noaa.gov/cgi-bin/nph-dods/nmsp/wcos/>

NSDI/FGDC Clearinghouse Nodes - <http://clearinghouse3.fgdc.gov/>

SIMoN - http://www.mbnms-simon.org/sections/obs/nmsp_wco.php

Webapp - <http://portal.ncddc.noaa.gov/wco/>

6.0 APPENDIX A: DATA COLLECTION AND INITIAL PROCESSING

6.1 PISCO ADCP DATA FORMAT

This section contains a copy of the ADCP_data_format.doc document provided by PISCO, referenced as PISCO Physical Oceanographic Moored ADCP ASCII File Format Specifications, November 2005 in Section 5.0. After the copy of the document, a sample of the ASCII files for ADCP data are provided.

PISCO Physical Oceanographic Moored ADCP ASCII File Format Specifications

(Results of 2004-11-19, 2005-01-26, 2005-06-10 and 2005-07-01 Phys Ocean Conference Calls)

by Patrick Drake, Revised 2005-11-20

General Specifications:

Most ADCP metadata will be stored in a separate file from the ADCP data, except for a one-line header at start of file. (This should make it easier to parse both the data and metadata.)

Each ASCII data file will contain data from a single deployment and a single ADCP only. If a deployment spans multiple raw data files, they should be combined before converting to ASCII.

The number of measurement depths will not change inside a data file, i.e. the number of rows per ensemble will NOT adjust to the instantaneous water depth.

No processing (interpolation, filtering, averaging, etc.) should be done to the data, except to flag or remove inaccurate values. Data known to be grossly inaccurate due to its proximity to the water surface or seabed should be identified with the “bad or missing value code,” 9999. (The precision of the fill value should match the precision of the surrounding data column, e.g. 9999.0, 9999.00, etc.) The location of the water surface will be determined from ADCP pressure measurements, or if pressure is unavailable, the maximum in ADCP echo intensity near the surface.

Data gaps in the interior of the file caused by an instrument or mooring malfunction should be filled with the “bad or missing data value,” 9999.0, 9999.00, etc. The time stamp should still be present. This will yield a regular sampling interval and a “continuous” time series for each file.

Out-of-water periods and data known to be inaccurate (garbage) at the beginning and end of the file should be removed altogether.

Example data are included in the companion file
SWC001_021ADCP020R00_20041012.40.1.TXT.

Specifics:

The data will be organized in columns, with each column representing a different data field. Each row will represent the values of the fields at a different measurement height and time.

Temperature and pressure measurements will be listed on one separate row for each ADCP ensemble, with the depth of the ADCP given as the measurement depth.

field delimiter: single space

record delimiter: new line, no carriage return

The field names will be given inside the file in a one-line, lowercase header, in the order described below:

Column 1

date: This field is a string. Use ISO 8601 standard, YYYY-MM-DD, with digits separated by hyphens.

Column 2

time: Use a separate field for time, because it is more easily recognizable as such when separated from the date. Again use the ISO 8601 standard HH:MM:SS.SSZ, with digits to hundredths of a second. The uppercase Z indicates GMT or universal time and is required. (Most PISCO measurements do not require precision to the hundredths of a second, but some turbulence sensors do, and it's best to incorporate this flexibility now.) If only the date of the measurement is known, include the time column and use 9999 as a fill value for time.

Column 3

yearday: Fractional day of the year, expressed as a decimal, relative to 12 a.m. (GMT) January 1 of the year in which each measurement took place. The yearday value resets to zero every January 1. For files that span multiple years, this will entail using different reference years, and the yearday value will reset to zero (or near-zero) inside the file. Note, noon January 1st is yearday 0.5, *not* yearday 1.5. (This resetting is somewhat awkward, but there is no obvious standard for a "zero-day" other than the first of each year.) The "precision" will be eight decimal places to the right of the decimal point for yeardays. This ensures "precision" to the hundredths of a second, matching the precision of the time field.

Column 4

height: height of velocity, temperature, or pressure measurement, in meters above the sea bottom, expressed to the decimeter

Column 5

depth: depth of velocity, temperature, or pressure measurement, in meters below Mean Sea Level, expressed to the decimeter. Bins above MSL are represented with negative depths.

(Note, all the following columns will use 9999.0, 9999.00, etc, to fill bad or missing values.)

Column 6

waterdepth: fluctuating water depth (sea floor to sea surface distance) at measurement site, in meters, as determined by location of sea surface from ADCP pressure measurements, or if pressure is unavailable, the maximum in ADCP echo intensity. If waterdepth is calculated from pressure, express the value to the millimeter. If calculated from backscatter, express the value to the decimeter.

Column 7

temp_c: ADCP temperature, in degrees Celsius, expressed to the hundredths of a degree

Column 8

pressure: ADCP pressure measurement, if available, in decibars. ADCPs not equipped with pressure sensors should include this column and use the “bad or missing value,” 9999, as a fill.

Column 9

intensity: ADCP echo intensity in units of RDI “counts,” average of all four beams

Column 10

data_quality: A quantitative data quality indicator, specifically RDI’s percent-good #4, in earth coordinates (percentage of successful 4-beam transformations), expressed as a percentage, 50, 80, 95, etc.

Column 11

eastward: True eastward velocity, in meters/second, expressed to the millimeter

Column 12

northward: True northward velocity, in meters/second, expressed to the millimeter

Column 13

upwards: Vertical velocity, in meters/second, expressed to the millimeter

Column 14

flag: A qualitative data quality indicator, or flag. There are only two acceptable values (for the time being), 0 for good, and 9999 for bad. If any of the variables (including temperature or pressure) on a measurement row have valid values, the flag should read 0. If all variables are missing or bad, use 9999. Note bad data should have 9999 in this column as well as a redundant 9999 in another column, such as eastward, etc. Additional codes should be developed.

Column 15

errorvelocity: RDI’s error velocity, in meters/second, expressed to the millimeter. If error velocity is unavailable, use fill value.

Sample file of the ADCP data:

date time yearday height depth waterdepth temp_c pressure intensity data_quality eastward
northward upwards flag

```
2005-08-31 16:43:57Z 242.69719074 001.5 017.2 018.939 9999.00 18.439 9999 9999 9999.000 9999.000 9999.000 9999
2005-08-31 16:43:57Z 242.69719074 002.5 016.2 9999.000 9999.00 9999.000 0131 100 00.037 -0.012 -0.005 100
2005-08-31 16:43:57Z 242.69719074 003.5 015.2 9999.000 9999.00 9999.000 0126 100 00.049 -0.021 00.002 100
2005-08-31 16:43:57Z 242.69719074 004.5 014.2 9999.000 9999.00 9999.000 0120 100 00.020 -0.030 00.007 100
2005-08-31 16:43:57Z 242.69719074 005.5 013.2 9999.000 9999.00 9999.000 0115 100 00.005 -0.031 00.005 100
2005-08-31 16:43:57Z 242.69719074 006.5 012.2 9999.000 9999.00 9999.000 0112 100 00.002 -0.022 00.002 100
2005-08-31 16:43:57Z 242.69719074 007.5 011.2 9999.000 9999.00 9999.000 0109 100 00.009 -0.009 00.001 100
2005-08-31 16:43:57Z 242.69719074 008.5 010.2 9999.000 9999.00 9999.000 0105 100 00.037 -0.028 00.006 100
2005-08-31 16:43:57Z 242.69719074 009.5 009.2 9999.000 9999.00 9999.000 0102 100 00.032 -0.014 00.007 100
2005-08-31 16:43:57Z 242.69719074 010.5 008.2 9999.000 9999.00 9999.000 0099 100 00.008 -0.039 00.007 100
2005-08-31 16:43:57Z 242.69719074 011.5 007.2 9999.000 9999.00 9999.000 0095 100 00.017 -0.037 00.006 100
2005-08-31 16:43:57Z 242.69719074 012.5 006.2 9999.000 9999.00 9999.000 0093 100 00.030 -0.037 -0.002 100
2005-08-31 16:43:57Z 242.69719074 013.5 005.2 9999.000 9999.00 9999.000 0089 097 00.023 -0.060 -0.001 100
```

6.2 PISCO TEMPERATURE DATA FORMAT

This section contains a copy of the Temp_data_format.doc document provided by PISCO, referenced as Final PISCO Physical Oceanographic Moored Temperature ASCII File Format Specifications, August 2005 in Section 5.0. After the copy of the document, a sample of the ASCII files for Temperature data are provided.

Final PISCO Physical Oceanographic Moored Temperature ASCII File Format Specifications

(Results of Nov. 19 Phys Ocean Conference Call, revised 12/10/04)
by Patrick Drake, Revised 2005-08-19

General Specifications:

Store most metadata in a separate file from the temperature data, except for a one-line header at start of file. (This should make it easier to parse both the data and metadata.)

Each data file will contain data from a single deployment and a single temperature sensor only. (Although it may be more convenient to store data from multiple sensors or deployments in one file, a lack of uniform sampling schemes prevents this method from being standardized to all the data.)

No processing (interpolation, filtering, averaging, etc.) should be done to the data, except to flag or remove inaccurate values.

Data gaps in the interior of the file caused by an instrument or mooring malfunction should be filled with the “bad or missing data value,” 9999. The time stamp should still be present. This will yield a regular sampling interval and a “continuous” time series for each file.

Out-of-water periods and data known to be inaccurate (garbage) at the beginning and end of the file should be removed altogether.

Example data are included in the companion file
SWC001_021MXTI004R00_20041012.40.1.TXT.

Specifics:

The data will be organized in columns, with each column representing a different data field, and each row a different data record (measurement). The field names will be given inside the file in a one-line, lowercase header, in the order described in the table below:

date	time	yearday	temp_c	flag
YYYY-MM-DD	HH:MM:SS.SSZ	XXX.XXXXXXXXX	single precision float	0 or 9999

field delimiter: single space

record delimiter: new line

date: This field is a string. Use ISO 8601 standard, YYYY-MM-DD, with digits separated by hyphens. This may not exactly fit the FGDC standard (because of the hyphens), but the form is more recognizable as a date. (This should be our goal, to have an unambiguous, universally recognized time stamp for each record.) And technically, it is only the metadata that must conform to FGDC, not the data.

time: Use a separate field for time, because it is more easily recognizable as such when separated from the date. Again use the ISO 8601 standard, HH:MM:SS.SSZ, with precision out to the hundredths of a second, regardless of the time precision of the actual instrument or measurement. The uppercase Z indicates GMT or universal time and is required. (Most PISCO measurements do not require precision to the hundredths of a second, but some turbulence sensors do, and it's best to incorporate this flexibility now.)

yearday: Fractional day of the year, expressed as a decimal, relative to 12 a.m. (GMT) January 1 of the year in which each measurement took place. The yearday value resets to zero every January 1. For files that span multiple years, this will entail using different reference years, and the yearday value will reset to zero (or near-zero) inside the file. Note, noon January 1st is yearday 0.5, *not* yearday 1.5. (This resetting is somewhat awkward, but there is no obvious standard for a "zero-day" other than the first of each year.) The precision will be eight decimal places to the right of the decimal point for yeardays (see table above). This ensures precision to the hundredths of a second, matching the precision of the time field.

temp_c: sensor temperature in degrees Celsius. Use a single-precision floating-point decimal. Precision of the field should match the precision of the instrument. Missing or bad data should be replaced with the value 9999.

flag: This field is a data-quality indicator. It is an integer and only has two acceptable values (for the time being). These are 0 for good and 9999 for bad. Note bad data should have both a 9999 in the flag column and a 9999 in the temp_c column.

Sample file of the Temperature data (data is space delimited):

```
date time yearday temp_c flag
2004-10-11 22:22:00.00Z 284.93194400 16.18 0
2004-10-11 22:24:00.00Z 284.93333300 16.18 0
2004-10-11 22:26:00.00Z 284.93472200 16.18 0
2004-10-11 22:28:00.00Z 284.93611100 16.18 0
2004-10-11 22:30:00.00Z 284.93750000 16.27 0
2004-10-11 22:32:00.00Z 284.93888900 16.18 0
2004-10-11 22:34:00.00Z 284.94027800 15.48 0
2004-10-11 22:36:00.00Z 284.94166700 15.04 0
2004-10-11 22:38:00.00Z 284.94305600 15.92 0
2004-10-11 22:40:00.00Z 284.94444400 16.09 0
2004-10-11 22:42:00.00Z 284.94583300 15.83 0
2004-10-11 22:44:00.00Z 284.94722200 15.04 0
2004-10-11 22:46:00.00Z 284.94861100 14.78 0
2004-10-11 22:48:00.00Z 284.95000000 14.44 0
```

6.3 PISCO ADCP METADATA FORMAT

This section contains a copy of the ADCP_metadata_content.doc document provided by PISCO, referenced as PISCO bottom-mounted ADCP metadata content and format, November 2005 in Section 5.0.

PISCO bottom-mounted ADCP metadata content and format

(Results of 2005-02-02, 2005-07-01 conference calls)

By Patrick Drake, Revised 2005-11-20

This document is a description of proposed PISCO physical oceanographic bottom-mounted ADCP metadata content and file-format specifications. The metadata described here refer only to standard PISCO ASCII ADCP data files, described in “PISCO Physical Oceanographic Moored ADCP ASCII File Format Specifications” (See “ADCP_data_format.doc.”)

ADCP metadata will be stored as text and written in EML, Ecological Metadata Language (<http://knb.ecoinformatics.org/software/eml/eml-2.0.1/index.html#introduction>), an XML Schema or language that allows for machine readable instructions as well as basic text content.

EML allows great flexibility, but also forces the metadata content into a structure and organization. This structure is basically hierarchical and requires the metadata to be stored in specific fields, many of which exist nested within other fields. Both EML, and the PISCO standards described here, limit the allowable values these fields may take.

Each metadata file will describe one, and only one, ASCII data file.

A complete example of a PISCO ADCP metadata eml file is provided in the companion file ADCP_Template.XML.

Each EML field relevant to bottom-mounted ADCPs is described in Table 1 below, along with its allowed value(s). Changeable values (those that must be adjusted for each data file) are highlighted in red. Unchanging values (those that will be constant from data file to data file and site to site) are highlighted in green.¹ Many fields contain both unchanging and adjustable portions. In Table 1 only, the nested organization of the EML fields is represented by the “>”, which can be interpreted as “contains.” (Note all fields in Table 1 are themselves nested within the EML “dataset” field.)

¹ For printing purposes, the red and green highlights were changed to bold and italics, respectively.

Table 1

EML dataset field name	Meaning / notes	Suggested Value	Required?
shortName	obvious	<i>PISCO bottom-mounted ADCP, SITECODE</i>	no
title	obvious	<i>PISCO: physical oceanography: bottom-mounted ADCP data: Sitename (SITECODE)</i>	yes
1 st creator > organizationName	organization responsible for (or owns) dataset	<i>Partnership for Interdisciplinary Studies of Coastal Oceans (PISCO)</i>	yes
1 st creator > onlineUrl	URL of organization above	<i>www.piscoweb.org</i>	yes
1 st creator > "additional fields"	obvious	do not include additional fields such as "address" or "electronMialAddress" for PISCO here, because there is no PISCO "headquarters"	n/a
2 nd creator > organizationName	organization responsible for (or owns) dataset	National Marine Sanctuary Program (if data are from a joint NMSP/PISCO mooring)	
2 nd creator > onlineURL	URL of organization above	http://www.sanctuaries.nos.noaa.gov (if data are from a joint NMSP/PISCO mooring)	
2 nd creator > "other fields"	obvious	Do not include additional fields for NMSP	
3 rd creator > individualName > salutation, givenName, and surname	PI responsible for (or owner of) dataset	Dr. Principal Investigator No.1	no
3 rd creator > address > deliveryPoint, city, administrativeArea, postalCode, country	street address of PI above	100 Penny Lane, Santa Cruz, California, 95060, USA	no
3 rd creator > phone	phone number of PI above	1-831-555-1212	no
3 rd creator > electronicMailAddress	email of PI above	pinvestigator1@ucsc.edu	no
3 rd creator > onlineUrl	Webpage of PI above	www.principalinvestigator1.org	no
4 th , 5 th , 6 th creator, etc.	other PIs	repeat as desired for responsible PIs	no
1 st associatedParty > individualName > salutation, givenName, surName	other individual associated with the dataset	Mr. John Smith	no
1 st associatedParty > address > deliveryPoint, city, administrativeArea, postalCode, country	street address of associated individual	1000 Dime Street, Pleasantville, New Jersey, 19876, USA	no
1 st associatedParty > phone	obvious	1-202-555-1212	no
1 st associatedParty > electronicMailAddress	obvious	jsmith@ucsc.edu	no
1 st associatedParty > onlineUrl	obvious	www.johnsmith.com	no
1 st associatedParty > role	function performed by associatedParty	Field Technician	no
2 nd , 3 rd , 4 th associatedParty, etc.		repeat as desired for all individuals associated with the dataset	no
language	obvious	<i>English</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
abstract	a summary of the dataset	This metadata record describes bottom-mounted ADCP data collected at Sand Hill Bluff, California, by PISCO. Measurements were collected using an RDI 600kHz Workhorse Sentinel ADCP, beginning 2001-10-26. The instrument depth was 19 meters, in an overall water-depth of 20 meters (both relative to Mean Sea Level, MSL). The instrument was programmed with a sampling interval of 2.0 minutes and a vertical resolution of 1 meter. (Any and all text is allowed here, but the abstract must contain the instrument type, deployment start date, the site name, the instrument depth, and the overall water depth.)	yes
1 st keywordSet > keywords	obvious	<i>EARTH SCIENCE : Oceans : Bathymetry/Seafloor Topography : Continental Margins,</i> <i>EARTH SCIENCE : Oceans : Ocean Circulation : Ocean Currents,</i> <i>EARTH SCIENCE : Oceans : Ocean Temperature : Water Temperature,</i> EARTH SCIENCE : Oceans : Ocean Pressure : Water Pressure (if water pressure was measured)	yes
1 st keywordSet > thesaurus	document where specific keywords are defined	<i>Global Change Master Directory</i>	yes
2 nd keywordSet > keywords	obvious	<i>Oceanographic Sensor Data, Bottom-mounted ADCP Current Profile Data,</i> National Marine Sanctuary Program Data (if data are from a joint NMSP/PISCO mooring)	yes
2 nd keywordSet > thesaurus	document where specific keywords are defined	<i>PISCO Categories</i>	yes
3 rd keywordSet > keywords	obvious	state or province, country, PISCO, continental shelf, ADCP, currents	yes
3 rd keywordSet > thesaurus	Place where specific keywords are defined	not used for 3 rd keywordSet	n/a
4 th , 5 th keywordSet, etc.	obvious	any relevant keywords	no
intellectualRights (for solely-owned PISCO data)	obvious	<p><i>Please cite PISCO in all publications containing this data. The citation should take the form: "This study utilized data collected by the Partnership for Interdisciplinary Studies of Coastal Oceans: a long-term ecological consortium funded by the David and Lucile Packard Foundation and the Gordon and Betty Moore Foundation."</i></p> <p><i>Please send copies of any published manuscript based on these data to the PISCO Program Coordinator (http://piscoweb.org/who/coordinators). Users of these data are encouraged to contact the Principal Investigators. Where appropriate, researchers whose projects are integrally dependent on PISCO data are encouraged to consider collaboration and/or co-authorship with original PISCO investigators.</i></p> <p><i>Extensive efforts are made to ensure that online data are accurate and up to date, but the authors and PISCO will not take responsibility for any errors that may exist in data provided online. Furthermore, the user assumes all responsibility for errors in analysis or judgment resulting from use of the data.</i></p>	

EML dataset field name	Meaning / notes	Suggested Value	Required?
intellectualRights (if data are from a joint NMSP/PISCO mooring)	obvious	<p>PISCO and NOAA request that end-users acknowledge both entities in any publications using these PISCO/Sanctuary data. The acknowledgement should read: "This study utilized data collected by the Partnership for Interdisciplinary Studies of Coastal Oceans: a long-term ecological consortium funded by the David and Lucile Packard Foundation and the Gordon and Betty Moore Foundation. Some PISCO data are also provided by the National Oceanic and Atmospheric Administration." PISCO also requests that end-users provide PISCO with copies of any published manuscripts using the PISCO/Sanctuary data. Manuscripts should be sent to the PISCO Program Coordinator (contact information available at www.piscoweb.org.) NOAA, NESDIS and NODC would appreciate recognition as the resource from which these data were obtained in any publications and/or other representations of these data.</p> <p>Extensive efforts are made to ensure that online data are accurate and up to date, but the authors and PISCO will not take responsibility for any errors that may exist in data provided online. Furthermore, the user assumes all responsibility for errors in analysis or judgment resulting from use of the data.</p> <p>NOAA makes no warranty regarding these data, expressed or implied, nor does the fact of distribution constitute such a warranty. NOAA, NESDIS and NODC cannot assume liability for any damages caused by any errors or omissions in these data, nor as a result of the failure of these data to function on a particular system.</p>	yes
distribution > online > url	Web address where data can be downloaded online	www.piscoweb.org	yes
coverage > geographicCoverage > geographicDescription	refers to PISCO site where data was collected	<p>Sand Hill Bluff: SHB001: This nearshore mooring is located on the northwestern edge of Monterey Bay, California, USA, approximately 5 km West of the city of Santa Cruz. The mooring lies in an overall water depth of 020 meters (referenced to Mean Sea Level, MSL). The altitudeMinimum and altitudeMaximum tags in this initial coverage section refer to the ADCP measurement range and are also referenced to MSL. They do not represent the overall water depth. Note the nominal range of the ADCP may extend from near-bottom (a depth expressed as a negative altitude) to slightly above MSL (a height expressed as a positive altitude).</p>	yes
coverage > geographicCoverage > boundingCoordinates > westBoundingCoordinate		longitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > eastBoundingCoordinate		longitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > northBoundingCoordinate		latitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > southBoundingCoordinate		latitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMinimum		-019 (minimum altitude of sensor range below MSL, expressed as a negative altitude)	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMaximum		+004 (maximum altitude of sensor range below MSL, expressed as a positive or negative altitude)	yes
coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeUnits	obvious	<i>meters</i>	yes
coverage > temporalCoverage > rangeOfDates > beginningDate > calendarDate		first date in dataset, expressed as YYYY-MM-DD	yes
coverage > temporalCoverage > rangeOfDates > beginningDate > time		time of first measurement in dataset, expressed as HH:MM:SS.SSZ	yes
coverage > temporalCoverage > rangeOfDates > endDate > calendarDate		last date in dataset, expressed as YYYY-MM-DD	yes
coverage > temporalCoverage > rangeOfDates > endDate > time		time of last measurement in dataset, expressed as HH:MM:SS.SSZ	yes
purpose	obvious	<i>This data was collected by PISCO to understand the physical processes of the inner continental shelf and their potential effects on marine ecology.</i>	yes
contact > positionName	Title of primary contact for information about this dataset	<i>Physical Oceanographic Data Manager</i>	yes
contact > organizationName	obvious	<i>PISCO</i>	yes
contact > electronicMailAddress	obvious	<i>phys-ocean-data-manager@piscoweb.org</i>	yes
contact > onlineUrl	obvious	<i>www.piscoweb.org</i>	yes
methods > methodStep > description	description of a method used to collect or process data that is applicable to entire dataset	<i>Methods for PISCO bottom-mounted ADCP collection and quality-control are available online, see the protocol citation.</i>	yes
methods > methodStep > protocol > title	a mandatory title for the method above	MY CAMPUS bottom-mounted ADCP data collection and quality-control methods #1	no
methods > methodStep > protocol > creator > organizationName		<i>PISCO</i>	no
methods > methodStep > protocol > distribution > online > url	online address for temperature methods used by a particular campus that are applicable to an entire dataset	<i>www.piscoweb.org/data/protocols/YOUR_ADCP_METHODS_#1_FILE_NAME</i>	no
methods > methodStep > instrumentation	Instrument used to collect data	RDI 600 kHz Workhorse Sentinel ADCP	yes
methods > sampling > studyExtent > coverage > geographicCoverage > geographicDescription	description of site, especially overall water depth	<i>In this sampling section the altitudeMaximum tag refers to the altitude of Mean Sea Level (MSL), and altitudeMinimum tag refers to the nominal altitude of the sea floor. In other words, the altitudeMinimum represents the overall water depth, expressed as a negative altitude.</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > westBoundingCoordinate		longitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > eastBoundingCoordinate		longitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > northBoundingCoordinate		latitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > southBoundingCoordinate		latitude of ADCP, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMinimum		-020 (altitude of the sea floor, relative to MSL)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMaximum		000 (altitude of Mean Sea Level)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeUnits	obvious	<i>meter</i>	yes
methods > sampling > samplingDescription	description of sampling	<i>Methods for PISCO bottom-mounted ADCP data collection and quality-control are available online, see the protocol citation. Data were collected using an RDI 600kHz Workhorse Sentinel ADCP set to 45 pings per 2.0-minute ensemble, with a vertical resolution of 1 meter.</i>	yes
project > title	obvious	<i>Partnership for Interdisciplinary Studies of Coastal Oceans (PISCO)</i>	yes
project > abstract	obvious	<i>PISCO is a large-scale marine research program that focuses on understanding the near-shore ecosystems of the U.S. West Coast. An interdisciplinary collaboration of scientists from four universities, PISCO integrates long-term monitoring of ecological and oceanographic processes at dozens of coastal sites with experimental work in the lab and field. We explore how individual organisms, populations, and ecological communities vary over space and time. Findings are applied to issues of ocean conservation and management, and are shared through our public outreach and student training programs.</i>	yes
project > funding	obvious	<i>PISCO is funded by the David and Lucile Packard Foundation and the Gordon and Betty Moore Foundation.</i>	yes
access > allow > principal	A group who can read and write to the data set	<i>PISCO staff</i> (see metadata example file for specifics)	yes
access > allow > permission (for PISCO staff)	obvious	<i>read and write</i>	yes
access > allow > principal	A group who can read and write to the data set	<i>general public</i>	yes
access > allow > permission (for general public)	obvious	<i>read only</i>	yes
dataTable > entityName	Name of 1st ASCII data file	DATA_FILENAME.TXT	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > physical > objectName	same as above	DATA_FILENAME.TXT	yes
dataTable > physical > size	obvious	File size, in bytes	no
dataTable > physical > characterEncoding	obvious	<i>ASCII</i>	yes
dataTable > physical > dataFormat > textFormat > numHeaderLines	obvious	<i>1</i>	yes
dataTable > physical > dataFormat > textFormat > recordDelimiter	obvious	<i>new line, no carriage return (#x0A)</i>	yes
dataTable > physical > dataFormat > textFormat > attributeOrientation	Are fields in data table the rows or columns?	<i>column</i>	yes
dataTable > physical > dataFormat > textFormat > simpleDelimited > fieldDelimiter	obvious	<i>single space (#x20)</i>	yes
dataTable > physical > distribution > online > url	Web address specific to 1 st dataTable	<i>www.piscoweb.org/data/ucsc/DATA_FILENAME.TXT</i>	yes
dataTable > attributeList > attribute > attributeName (for date)	name of field in column 1 of data table	<i>date</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for date)	obvious	<i>GMT calendar date of each ADCP measurement record, in ISO 8601 standard format, YYYY-MM-DD</i>	yes
dataTable > attributeList > attribute > storageType (for date)		<i>date</i>	yes
dataTable > attributeList > attribute > measurementScale > dateTime > formatString (for date)	date format	<i>YYYY-MM-DD</i>	yes
dataTable > attributeList > attribute > measurementScale > dateTime > dateTimePrecision (for date)	Precision of date (for our uses this will be the effective resolution of the instrument)	<i>1 day</i>	yes
dataTable > attributeList > attribute > attributeName (for time)	name of field in column 2 of data table	<i>time</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for time)	obvious	<i>Greenwich Mean Time of each ADCP measurement record, to the one-hundredth of a second, in ISO 8601 standard format, HH:MM:SS.SSZ. The Z indicates "Zero Meridian" or Greenwich Mean Time and must be included</i>	yes
dataTable > attributeList > attribute > measurementScale > dateTime > formatString (for time)	time format	<i>HH:MM:SS.SSZ,</i>	yes
dataTable > attributeList > attribute > measurementScale > dateTime > precision (for time)	Precision of time (for our uses this will be the effective resolution of the instrument)	<i>1 second</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for time)	Info on accuracy of time	-10 to +30 ppm, or -5.26 to 15.76 minutes per year, from RDI manuals	no
dataTable > attributeList > attribute > attributeName (for yearday)	name of field in column 3 of data table	yearday	yes
dataTable > attributeList > attribute > attributeDefinition (for yearday)	obvious	Greenwich Mean Time of each ADCP measurement record, expressed as fractional decimal days since Jan. 1 of the year measurement was made. For example, 12 noon GMT on Jan. 2 is represented by yearday 1.5, NOT yearday 2.5.	yes
dataTable > attributeList > attribute > measurementScale > interval > unit > standardUnit (for yearday)	units of field in column 3 of data table	nominalDay	yes
dataTable > attributeList > attribute > measurementScale > interval > precision (for yearday)	Precision of yearday (for our uses this will be the effective resolution of the instrument)	0.0000116 (Indicates 0.0000116 days, or 1 second)	yes
dataTable > attributeList > attribute > measurementScale > interval > numericDomain > numberType (for yearday)		real	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for yearday)	Info on accuracy of yearday	-10 to +30 ppm, or -5.26 to 15.76 minutes per year	no
dataTable > attributeList > attribute > attributeName (for height)	name of field in column 4 of data table	height	yes
dataTable > attributeList > attribute > attributeDefinition (for height)	obvious	height of velocity, temperature, or pressure measurement, in meters above the sea bottom	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for height)	units of field in column 4 of data table	meter	yes
dataTable > attributeList > attribute > measurementScale > ratio > precision (for height)	Precision of height	0.1 meters	yes
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for height)		real	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for height)	Info on accuracy of height	0.5 meters (an estimate of the accuracy of the height measurement)	yes
dataTable > attributeList > attribute > attributeName (for depth)	name of field in column 5 of data table	depth	yes
dataTable > attributeList > attribute > attributeDefinition (for depth)	obvious	depth of velocity, temperature, or pressure measurement, in positive meters below Mean Sea Level. Bins above MSL are represented with negative depths.	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for depth)	units of field in column 5 of data table	<i>meter</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > precision (for depth)	Precision of depth (for our uses this will be the effective resolution of the instrument)	<i>0.1 meters</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > number Type (for depth)		<i>real</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for depth)	Info on accuracy of depth	<i>0.5 meters (an estimate of the accuracy of the depth measurement)</i>	yes
dataTable > attributeList > attribute > attributeName (for waterdepth)	name of field in column 6 of data table	<i>waterdepth</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for waterdepth)	obvious	<i>fluctuating water depth (sea floor to sea surface distance) at measurement site, in meters, as determined by location of sea surface from ADCP pressure measurements, or if pressure is unavailable, the maximum in ADCP echo intensity.</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for waterdepth)	units of field in column 6 of data table	<i>meters</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > precision (for waterdepth)	Precision of waterdepth (for our uses this will be the effective resolution of the instrument)	0.1 or 0.001 meters If waterdepth is calculated from pressure, express the value to the millimeter. If calculated from backscatter, express the value to the decimeter.	yes
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for waterdepth)	???	<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for waterdepth)	Missing value code	9999.0 or 9999.000	yes
dataTable > attributeList > attribute > missingValueCode > definition (for waterdepth)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for waterdepth)	Info on accuracy of waterdepth	1 meter (if calculated from backscatter)	yes
dataTable > attributeList > attribute > attributeName (for temp_c)	name of field in column 7 of data table	<i>temp_c</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for temp_c)	obvious	<i>seawater temperature from ADCP sensor</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > interval > unit > standardUnit (for temp_c)	units of field in column 7 of data table	<i>Celsius</i>	yes
dataTable > attributeList > attribute > measurementScale > interval > precision (for temp_c)	Precision of temp_c (for our uses this will be the effective resolution of the instrument)	0.01	yes
dataTable > attributeList > attribute > measurementScale > interval > numericDomain > numberType (for temp_c)	obvious	<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for temp_c)	Missing value code	<i>9999.00</i>	yes
dataTable > attributeList > attribute > missingValueCode > definition (for temp_c)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for temp_c)	Info on accuracy of temp_c	<i>plus or minus 0.4 degrees Celsius, from RDI Manual</i>	no
dataTable > attributeList > attribute > attributeName (for pressure)	name of field in column 8 of data table	<i>pressure</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for pressure)	obvious	<i>pressure at ADCP</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > customUnit (for pressure)	units of field in column 8 of data table	<i>decibar</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > precision (for pressure)	Precision of pressure (for our uses this will be the effective resolution of the instrument)	???	yes
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for pressure)		<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for pressure)	Missing value code	<i>9999.0000</i>	yes
dataTable > attributeList > attribute > missingValueCode > definition (for pressure)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for pressure)	Info on accuracy of pressure	???	no
dataTable > attributeList > attribute > attributeName (for intensity)	name of field in column 9 of data table	<i>intensity</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for intensity)	obvious	<i>ADCP echo intensity (or backscatter), in RDI counts. This value represents the average of all 4 beams, rounded to the nearest whole number</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > interval > unit > standardUnit (for intensity)	units of field in column 9 of data table	<i>number</i>	yes
dataTable > attributeList > attribute > measurementScale > interval > precision (for intensity)	Precision of intensity (for our uses this will be the effective resolution of the instrument)	<i>1</i>	yes
dataTable > attributeList > attribute > measurementScale > interval > numericDomain > numberType (for intensity)		<i>integer</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for intensity)	Missing value code	<i>9999</i>	yes
dataTable > attributeList > attribute > missingValueCode > definition (for intensity)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > attributeName (for data_quality)	name of field in column 10 of data table	<i>data_quality</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for data_quality)	obvious	<i>A post-processing quantitative data quality indicator. Specifically, RDI percent-good #4, in earth coordinates (percentage of successful 4-beam transformations), expressed as a percentage, 50, 80, 95, etc.</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for data_quality)	units of field in column 10 of data table	<i>number</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > precision (for data_quality)	Precision of invalid (for our uses this will be the effective resolution of the instrument)	<i>1</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for data_quality)		<i>real</i>	???
dataTable > attributeList > attribute > attributeName (for eastward)	name of field in column 11 of data table	<i>eastward</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for eastward)	obvious	<i>True eastward water velocity (negative values represent westward velocities)</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for eastward)	units of field in column 11 of data table	<i>meterPerSecond</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > ratio > precision (for eastward)	Precision of eastward (for our uses this will be the effective resolution of the instrument)	0.02 meterPerSecond (for precision, use the standard deviation given by RDI's software)	yes
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for eastward)		<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for eastward)	Missing value code	9999.000	yes
dataTable > attributeList > attribute > missingValueCode > definition (for eastward)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for eastward)	Info on accuracy of eastward	0.02 meterPerSecond, from RDI Software (for accuracy, use the standard deviation given by RDI's software (this will be the same value as the precision))	yes
dataTable > attributeList > attribute > attributeName (for northward)	name of field in column 12 of data table	<i>northward</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for northward)	obvious	<i>True northward water velocity (negative values represent southward velocities)</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for northward)	units of field in column 12 of data table	<i>meterPerSecond</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > precision (for northward)	Precision of northward (for our uses this will be the effective resolution of the instrument)	0.02 meterPerSecond (for precision, use the standard deviation given by RDI's software)	yes
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for northward)		<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for northward)	Missing value code	9999.000	yes
dataTable > attributeList > attribute > missingValueCode > definition (for northward)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for northward)	Info on accuracy of northward	02 meterPerSecond, from RDI Software (for accuracy, use the standard deviation given by RDI's software (this will be the same value as the precision))	yes
dataTable > attributeList > attribute > attributeName (for upwards)	name of field in column 13 of data table	<i>upwards</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for upwards)	obvious	<i>True vertical water velocity (positive toward outer space)</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for upwards)	units of field in column 13 of data table	<i>meterPerSecond</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > ratio > precision (for upwards)	Precision of upwards (for our uses this will be the effective resolution of the instrument)	do not include for now	no
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for upwards)		<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for upwards)	Missing value code	9999.000	yes
dataTable > attributeList > attribute > missingValueCode > definition (for upwards)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for upwards)	Info on accuracy of upwards	do not include for now	no
dataTable > attributeList > attribute > attributeName (for flag)	name of field in column 14 of data table	<i>flag</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for flag)		<i>Various codes that describe the data, see codeDefinitions below</i>	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefinition > code (for flag)	a flag code	0	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefinition > definition (for flag code "0")	definition of flag code above	<i>good data</i>	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefinition > code (for flag)	a flag code	9999	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefinition > definition (for flag code "9999")	definition of flag code above	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefinition > code (for flag codes "100-199")	a set of flag codes	100-199 (a set of flag codes to be defined by UCSB)	no
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefinition > code (for flag codes "200-299")	a set of flag codes	200-299 (a set of flag codes to be defined by UCSC)	no

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefintion > code (for flag codes "300-999")	a set of flag codes	300-399 (a set of flag codes to be defined by OSU)	no
dataTable > attributeList > attribute > attributeName (for errorvelocity)	name of field in column 15 of data table	<i>errorvelocity</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for errorvelocity)	obvious	<i>The difference of two vertical velocities, each measured by an opposing pair of ADCP beams</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > unit > standardUnit (for errorvelocity)	units of field in column 15 of data table	<i>meterPerSecond</i>	yes
dataTable > attributeList > attribute > measurementScale > ratio > precision (for errorvelocity)	Precision of errorvelocity (for our uses this will be the effective resolution of the instrument)	do not include for now	no
dataTable > attributeList > attribute > measurementScale > ratio > numericDomain > numberType (for errorvelocity)		<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for errorvelocity)	Missing value code	<i>9999.000</i>	yes
dataTable > attributeList > attribute > missingValueCode > definition (for errorvelocity)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for errorvelocity)	Info on accuracy of errorvelocity	do not include for now	no

6.4 PISCO TEMPERATURE METADATA FORMAT

This section contains a copy of the Temp_metadata_content.doc document provided by PISCO, referenced as PISCO moored temperature metadata content and format, August 2005 in Section 5.0.

PISCO moored temperature metadata content and format

Results of 2004-12-15 and 2005-01-19 conference calls

By Patrick Drake, Revised 2005-08-19

This document is a description of PISCO physical oceanographic moored temperature metadata content and file-format specifications. The metadata described here refer only to standard PISCO ASCII temperature data files, described in Final PISCO Physical Oceanographic Moored Temperature ASCII File Format Specifications (See “Temp_data_format.doc.”)

Temperature metadata will be stored as text and written in EML, Ecological Metadata Language (<http://knb.ecoinformatics.org/software/eml/eml-2.0.1/index.html#introduction>), an XML Schema or language that allows for machine readable instructions as well as basic text content.

A complete example of a PISCO temperature metadata EML file is provided in the companion file MXTI_template.xml.

EML allows great flexibility, but also forces the metadata content into a structure and organization. This structure is basically hierarchical and requires the metadata to be stored in specific fields, many of which exist nested within other fields. Both EML, and the PISCO standards described here, limit the allowable values these fields may take.

Each metadata file will describe one, and only one, ASCII data file.

Each EML field relevant to moored temperature is described in Table 1 below, along with its allowed value(s). Changeable values (those that must be adjusted for each data file) are highlighted in red. Unchanging values (those that will be constant from data file to data file and site to site) are highlighted in green.² Many fields contain both unchanging and adjustable portions. In Table 1 only, the nested organization of the EML fields is represented by the “>”, which can be interpreted as “contains.” (Note all fields in Table 1 are themselves nested within the EML “dataset” field.)

² For printing purposes, the red and green highlights were changed to bold and italics, respectively.

Table 1

EML dataset field name	Meaning / notes	Suggested Value	Required?
shortName	obvious	<i>PISCO moored temperature</i> , SITECODE	no
title	obvious	<i>PISCO: physical oceanography: moored temperature data: Sitename (SITECODE)</i>	yes
1 st creator > organizationName	organization responsible for (or owns) dataset	<i>Partnership for Interdisciplinary Studies of Coastal Oceans (PISCO)</i>	yes
1 st creator > onlineUrl	URL of organization above	www.piscoweb.org	yes
1 st creator > "other fields"	obvious	do not include other fields such as "address" or "electronMialAddress" for PISCO here, because there is no PISCO "headquarters"	n/a
2 nd creator > organizationName	organization responsible for (or owns) dataset	National Marine Sanctuary Program (if data are from a joint NMSP/PISCO mooring)	
2 nd creator > onlineURL	URL of organization above	http://www.sanctuaries.nos.noaa.gov (if data are from a joint NMSP/PISCO mooring)	
2 nd creator > "other fields"	obvious	Do not include additional fields for NMSP	
3 rd creator > individualName > salutation, givenName, and surname	PI responsible for (or owner of) dataset	Dr. Principal Investigator No.1	no
3 rd creator > address > deliveryPoint, city, administrativeArea, postalCode, country	street address of PI above	100 Penny Lane, Santa Cruz, California, 95060, USA	no
3 rd creator > phone	phone number of PI above	1-831-555-1212	no
3 rd creator > electronicMailAddress	email of PI above	pinvestigator1@ucsc.edu	no
3 rd creator > onlineUrl	Webpage of PI above	www.principalinvestigator1.org	no
4 th , 5 th , 6 th creator, etc.	other PIs	repeat as desired for responsible PIs	no
1 st associatedParty > individualName > salutation, givenName, surName	other individual associated with the dataset	Mr. John Smith	no
1 st associatedParty > address > deliveryPoint, city, administrativeArea, postalCode, country	street address of associated individual	1000 Dime Street, Pleasantville, New Jersey, 19876, USA	no
1 st associatedParty > phone	obvious	1-202-555-1212	no
1 st associatedParty > electronicMailAddress	obvious	jsmith@ucsc.edu	no
1 st associatedParty > onlineUrl	obvious	www.johnsmith.com	no
1 st associatedParty > role	function performed by associatedParty	Field Technician	no
2 nd , 3 rd , 4 th associatedParty, etc.		repeat as desired for all individuals associated with the dataset	no
language	obvious	<i>English</i>	yes
abstract	a summary of the dataset	This metadata record describes moored seawater temperature data collected at Terrace Point, California, by PISCO. Measurements were collected using an XTI Stowaway thermistor beginning 2003-09-6. The instrument depth was 016 meters, in an overall water-depth of 018 meters (both relative to Mean Sea Level). The sampling interval was 2.0 minutes. (Any and all text is allowed here, but the abstract must contain the deployment start date, the site name, the instrument depth, and the overall water depth.)	yes
1 st keywordSet > keywords	obvious	<i>EARTH SCIENCE: Oceans: Bathymetry/Seafloor Topography: Continental Margins, EARTH SCIENCE: Oceans: Ocean Temperature: Water Temperature</i>	yes
1 st keywordSet > thesaurus		<i>Global Change Master Directory</i>	yes
2 nd keywordSet > keywords	obvious	<i>Oceanographic Sensor Data, Moored Temperature Data, National Marine Sanctuary Program</i> (if data are from a joint NMSP/PISCO mooring)	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
2 nd keywordSet > thesaurus	document where specific keywords are defined	PISCO Categories	yes
3 rd keywordSet > keywords	obvious	continental shelf, seawater, temperature, state or province, country, PISCO	yes
3 rd keywordSet > thesaurus	Place where specific keywords are defined	not used for 3 rd keywordSet	n/a
4 th , 5 th keywordSet, etc.	obvious	relevant keywords	no
intellectualRights (for solely-owned PISCO data)	obvious	<p><i>Please cite PISCO in all publications containing this data. The citation should take the form: "This study utilized data collected by the Partnership for Interdisciplinary Studies of Coastal Oceans: a long-term ecological consortium funded by the David and Lucile Packard Foundation and the Gordon and Betty Moore Foundation."</i></p> <p><i>Please send copies of any published manuscript based on these data to the PISCO Program Coordinator (http://piscoweb.org/who/coordinators). Users of these data are encouraged to contact the Principal Investigators. Where appropriate, researchers whose projects are integrally dependent on PISCO data are encouraged to consider collaboration and/or co-authorship with original PISCO investigators.</i></p> <p><i>Extensive efforts are made to ensure that online data are accurate and up to date, but the authors and PISCO will not take responsibility for any errors that may exist in data provided online. Furthermore, the user assumes all responsibility for errors in analysis or judgment resulting from use of the data.</i></p>	
intellectualRights (if data are from a joint NMSP/PISCO mooring)	obvious	<p><i>PISCO and NOAA request that end-users acknowledge both entities in any publications using these PISCO/Sanctuary data. The acknowledgement should read: "This study utilized data collected by the Partnership for Interdisciplinary Studies of Coastal Oceans: a long-term ecological consortium funded by the David and Lucile Packard Foundation and the Gordon and Betty Moore Foundation. Some PISCO data are also provided by the National Oceanic and Atmospheric Administration." PISCO also requests that end-users provide PISCO with copies of any published manuscripts using the PISCO/Sanctuary data. Manuscripts should be sent to the PISCO Program Coordinator (contact information available at www.piscoweb.org.) NOAA, NESDIS and NODC would appreciate recognition as the resource from which these data were obtained in any publications and/or other representations of these data.</i></p> <p><i>Extensive efforts are made to ensure that online data are accurate and up to date, but the authors and PISCO will not take responsibility for any errors that may exist in data provided online. Furthermore, the user assumes all responsibility for errors in analysis or judgment resulting from use of the data.</i></p> <p><i>NOAA makes no warranty regarding these data, expressed or implied, nor does the fact of distribution constitute such a warranty. NOAA, NESDIS and NODC cannot assume liability for any damages caused by any errors or omissions in these data, nor as a result of the failure of these data to function on a particular system.</i></p>	yes
distribution > online > url	Web address where data can be downloaded online	www.piscoweb.org	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
coverage > geographicCoverage > geographicDescription	refers to PISCO site where data was collected	Terrace Point: TPT001: This nearshore mooring is located offshore of the city of Santa Cruz, California, USA. The mooring lies in an overall water depth of 018 meters (referenced to Mean Sea Level, MSL). The altitudeMinimum and altitudeMaximum tags in this initial coverage section refer to the logger measurement range (with depths expressed as negative altitudes) and are also referenced to MSL. They do NOT represent the overall water depth.	yes
coverage > geographicCoverage > boundingCoordinates > westBoundingCoordinate		longitude of mooring, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > eastBoundingCoordinate		longitude of mooring, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > northBoundingCoordinate		latitude of mooring, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > southBoundingCoordinate		latitude of mooring, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMinimum		-016 (minimum depth of sensor below MSL, expressed as a negative altitude)	yes
coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMaximum		-016 (maximum depth of sensor below MSL, expressed as a negative altitude)	yes
coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeUnits	obvious	<i>meters</i>	yes
coverage > temporalCoverage > rangeOfDates > beginningDate > calendarDate		first date in dataset, expressed as YYYY-MM-DD	yes
coverage > temporalCoverage > rangeOfDates > beginningDate > time		time of first measurement in dataset, expressed as HH:MM:SS.SSZ	yes
coverage > temporalCoverage > rangeOfDates > endDate > calendarDate		last date in dataset, expressed as YYYY-MM-DD	yes
coverage > temporalCoverage > rangeOfDates > endDate > time		time of last measurement in dataset, expressed as HH:MM:SS.SSZ	yes
purpose	obvious	<i>This data was collected by PISCO to understand the physical processes of the inner continental shelf and their potential effects on marine ecology.</i>	yes
contact > positionName	Title of primary contact for information about this dataset	<i>Physical Oceanographic Data Manager</i>	yes
contact > organizationName	obvious	<i>PISCO</i>	yes
contact > electronicMailAddress	obvious	<i>phys-ocean-data-manager@piscoweb.org</i>	yes
contact > onlineUrl	obvious	<i>www.piscoweb.org</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
methods > methodStep > description	description of a method used to collect or process data that is applicable to entire dataset	<i>Methods for PISCO moored temperature collection and quality-control are available online, see citation below.</i>	no
methods > methodStep > protocol > title	a mandatory title for the method above	MY CAMPUS moored temperature collection and quality-control methods #1	no
methods > methodStep > protocol > creator > organizationName		<i>PISCO</i>	
methods > methodStep > protocol > distribution > online > url	online address for temperature methods used by a particular campus that are applicable to an entire dataset	<i>www.piscoweb.org/data/protocols/YOUR_TEMPERATURE_METHODS_#1_FILE_NAME</i>	no
methods > methodStep > instrumentation	Instrument used to collect data	Onset StowAway XTI Temperature Logger (Onset Computer Corp. XTI32-05+37)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > geographicDescription	description of site, especially overall water depth	<i>In this sampling section the altitudeMaximum tag refers to the altitude of Mean Sea Level (MSL), and altitudeMinimum tag refers to the nominal altitude of the sea floor. In other words, the altitudeMinimum represents the overall water depth, expressed as a negative altitude.</i>	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > westBoundingCoordinate		longitude of logger, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > eastBoundingCoordinate		longitude of logger, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > northBoundingCoordinate		latitude of logger, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > southBoundingCoordinate		latitude of logger, expressed as ±XXX.XXXX degrees (Be as accurate as possible.)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMinimum		-018 (altitude of the sea floor, relative to MSL)	yes
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeMaximum		000 (altitude of Mean Sea Level)	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
methods > sampling > studyExtent > coverage > geographicCoverage > boundingCoordinates > boundingAltitudes > altitudeUnits	obvious	<i>meter</i>	yes
methods > sampling > samplingDescription	description of sampling	<i>Methods for PISCO moored temperature data collection and quality-control are available online, see the protocol citation. Data were collected using an Onset StowAway XTI Temperature Logger set to a 2.0-minute sampling interval</i>	yes
project > title	obvious	<i>Partnership for Interdisciplinary Studies of Coastal Oceans (PISCO)</i>	yes
project > abstract	obvious	<i>PISCO is a large-scale marine research program that focuses on understanding the near-shore ecosystems of the U.S. West Coast. An interdisciplinary collaboration of scientists from four universities, PISCO integrates long-term monitoring of ecological and oceanographic processes at dozens of coastal sites with experimental work in the lab and field. We explore how individual organisms, populations, and ecological communities vary over space and time. Findings are applied to issues of ocean conservation and management, and are shared through our public outreach and student training programs.</i>	yes
project > funding	obvious	<i>PISCO is funded by the David and Lucile Packard Foundation and the Gordon and Betty Moore Foundation.</i>	yes
access > allow > principal	A group who can read and write to the data set	<i>PISCO staff (see metadata xml example file for specifics)</i>	yes
access > allow > permission (for PISCO staff)	obvious	<i>read and write</i>	yes
access > allow > principal	A group who can read and write to the data set	<i>general public</i>	yes
access > allow > permission (for general public)	obvious	<i>read only</i>	yes
dataTable > entityName	Name of 1st ASCII data file	DATA_FILE_NAME.TXT	yes
dataTable > physical > objectName	same as above	DATA_FILE_NAME.TXT	yes
dataTable > physical > size	obvious	File size, in bytes	no
dataTable > physical > characterEncoding	obvious	<i>ASCII</i>	yes
dataTable > physical > dataFormat > textFormat > numHeaderLines	obvious	<i>1</i>	yes
dataTable > physical > dataFormat > textFormat > recordDelimiter	obvious	<i>new line, no carriage return (#x0A)</i>	yes
dataTable > physical > dataFormat > textFormat > attributeOrientation	Are fields in data table the rows or columns?	<i>column</i>	yes
dataTable > physical > dataFormat > textFormat > simpleDelimited > fieldDelimiter	obvious	<i>single space (#x20)</i>	yes
dataTable > physical > distribution > online > url	Web address specific to 1 st dataTable	<i>www.piscoweb.org/data/ucsc/DATA_FILE_NAME.TXT</i>	yes
dataTable > attributeList > attribute > attributeName (for date)	name of field in column 1 of data table	<i>date</i>	yes
dataTable > attributeList > attribute > attributeDefinition (for date)	obvious	<i>GMT calendar date of each temperature measurement record, in ISO 8601 standard format, YYYY-MM-DD</i>	yes
dataTable > attributeList > attribute > storageType (for date)		<i>date</i>	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > dateTime > formatString (for date)	date format	YYYY-MM-DD	yes
dataTable > attributeList > attribute > measurementScale > dateTime > dateTimePrecision (for date)	Precision of date (for our uses this will be the effective resolution of the instrument)	1 day	yes
dataTable > attributeList > attribute > attributeName (for time)	name of field in column 2 of data table	time	yes
dataTable > attributeList > attribute > attributeDefinition (for time)	obvious	Greenwich Mean Time of each temperature measurement record, to the one-hundredth of a second, in ISO 8601 standard format HH:MM:SS.SSZ. The Z indicates "Zero Meridian" or Greenwich Mean Time and must be included	yes
dataTable > attributeList > attribute > measurementScale > dateTime > formatString (for time)	time format	HH:MM:SS.SSZ,	yes
dataTable > attributeList > attribute > measurementScale > dateTime > precision (for time)	Precision of time (for our uses this will be the effective resolution of the instrument)	1 second	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for time)	Info on accuracy of time	100 ppm, or 52.6 minutes/year, from Onset Manuals	no
dataTable > attributeList > attribute > attributeName (for yearday)	name of field in column 3 of data table	yearday	yes
dataTable > attributeList > attribute > attributeDefinition (for yearday)	obvious	Greenwich Mean Time of each temperature measurement record, expressed as fractional decimal days since Jan. 1 of the year measurement was made. For example, 12 noon GMT on Jan. 2 is represented by yearday 1.5, NOT yearday 2.5.	yes
dataTable > attributeList > attribute > measurementScale > interval > unit > standardUnit (for yearday)	units of field in column 3 of data table	nominalDay	yes
dataTable > attributeList > attribute > measurementScale > interval > precision (for yearday)	Precision of yearday (for our uses this will be the effective resolution of the instrument)	0.0000116 (Indicates 0.0000116 days.)	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for yearday)	Info on accuracy of yearday	100 ppm, or 52.6 minutes/year, from Onset Manuals	yes
dataTable > attributeList > attribute > attributeName (for temp_c)	name of field in column 4 of data table	temp_c	yes
dataTable > attributeList > attribute > attributeDefinition (for temp_c)	obvious	seawater temperature	yes
dataTable > attributeList > attribute > measurementScale > interval > unit > standardUnit (for temp_c)	units of field in column 4 of data table	Celsius	yes

EML dataset field name	Meaning / notes	Suggested Value	Required?
dataTable > attributeList > attribute > measurementScale > interval > precision (for temp_c)	Precision of temp_c (for our uses this will be the effective resolution of the instrument)	0.17	yes
dataTable > attributeList > attribute > measurementScale > interval > numericDomain > numberType (for temp_c)	obvious	<i>real</i>	yes
dataTable > attributeList > attribute > missingValueCode > code (for temp_c)	Missing value code	9999.00	yes
dataTable > attributeList > attribute > missingValueCode > definition (for temp_c)	Meaning of missing value code	<i>bad or missing data</i>	yes
dataTable > attributeList > attribute > accuracy > attributeAccuracyReport (for temp_c)	Info on accuracy of temp_c	0.33 degrees celcius, from Onset Manuals	no
dataTable > attributeList > attribute > attributeName (for flag)	name of field in column 5 of data table	<i>flag</i>	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefintion > code (for flag)	a flag code	0	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefintion > definition (for flag)	definition of flag code above	<i>good data</i>	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefintion > code (for flag)	a flag code	9999	yes
dataTable > attributeList > attribute > measurementScale > nominal > nonNumericDomain > enumeratedDomain > codeDefintion > definition (for flag)	definition of flag code above	<i>bad or missing data</i>	yes

6.5 PISCO NAMING SCHEME

This section contains a copy of the NamingScheme.doc document provided by PISCO, referenced as PISCO-Wide Physical Oceanographic Proposed Data and Metadata File Naming Scheme, September 2005 in Section 5.0.

PISCO-Wide Physical Oceanographic Proposed Data and Metadata File Naming Scheme

(Result of 2005-01-26 and 2005-02-02 conference calls and update of naming scheme in “Physical Oceanographic Data Handling” document of 2002-04-17)
Revised 2005-09-013, by Patrick Drake

To facilitate data exchange between campuses and the archiving of data in PISCO’s MetaCat catalogue, PISCO will adopt a standardized naming scheme for its physical oceanographic data and metadata files. This scheme will apply to all data uploaded to MetaCat, but it need not be used by those working and archiving the data locally. MetaCat data package identifiers will incorporate these names (see below).

File Naming Scheme

The file naming scheme has been designed to provide unique names, maximum information at a glance, and help reduce the likelihood of overwriting similarly named files. Except for the extension, revision number and Document ID(see below), data files and metadata files describing the same data will have the exact same name (i.e., the first 30 characters will be identical). Files names will consist of eight parts or fields, plus an extension. The parts, in order from left to right, are:

- 1. Site abbreviation:** 6 uppercase characters followed by an underscore. Choose site names only from PISCO-wide list (use uppercase “X” for unused characters)
- 2. Overall water depth at mooring or bottom-mounted ADCP, in meters below Mean Sea Level (MSL):** 3 digits (use 000 for onshore temperature loggers or other onshore instruments)
- 2. Instrument type:** 4 uppercase characters – choose only instrument abbreviations from list below
- 3. Depth of instrument, in meters below Mea Sea Level (MSL):** 3 digits (for onshore instruments, use estimated height in meters, e.g. 001)
- 4. Replicate instrument identifier:** Uppercase “R” (for replicate) followed by a two digit replicate number and an underscore. For the first instrument, use “R00_”. For a second (replicate) instrument, use “R01_”, etc.

5. Date of first measurement in file (for processed QA/QC data): eight digits in “YYYYMMDD” format. For raw files, this will be the date of the first “good” data in the file

6. Document ID: This part is required for consistency with MetaCat’s ID system. For XML metadata files, use “.50.” For processed text data files, this field should always be a “.40.” For processed Matlab structure files, use “.30.” For unprocessed Matlab structure files and raw text temperature files, use “.20.” For raw RDI and Onset binary files, use “.10.”

7. Revision Number: This field is also required for consistency with MetaCat. It represents the version number of the data or metadata file. It should increment by one, from one, for every revision. Data and metadata files need not have the same revision number.

8. Extension: period followed by three uppercase characters. Data files should be in ASCII format and have a “.TXT” extension. Metadata files should be in Ecological Metadata Format and carry an “.XML” extension. Raw binary RDI files should have a “.000” extension. Raw Onset files should have a “.dtf” extension. Matlab data files should carry a “.MAT” extension.

Example:

Data file:

FCXXXX_015MXTI008R00_20001015.40.1.TXT

Metadata file:

FCXXXX_015MXTI008R00_20001015.50.3.XML

Site - FCXXXX_ (Fogarty Creek)

Water depth at Mooring or ADCP - 015 (meters)

Instrument type - MXTI (Mooring XTI temperature logger)

Instrument depth - 008

Replicate number - R00_

First file date - 20001015 (Oct. 15, 2000)

Data file revision number - 1

Metadata file revision number - 3

Instrument Abbreviation Codes

ADCP - Acoustic Doppler Current Profiler

VACM - Vector averaging current meter (OSU only)

CTDX - CTD

CTXX - CT

CTWF - CT with fluorometer

MRFL - Mooring fluorometer

INFL - Intertidal fluorometer

MXTI - Mooring (Onset) XTI temperature logger

MTBD - Mooring (Onset) Tidbit temperature logger
OTBD - Onshore (Onset) Tidbit temperature logger
OOPT - Onshore (Onset) Optic Stowaway temperature logger
IPAR - Intertidal photosynthetically active radiation (PAR) sensor
APAR - Atmospheric PAR sensor (onshore or on a mooring met station for example)

MetaCat ID

MetaCat IDs are required by MetaCat and are stored in MetaCat and within metadata files. Each “data package” will have a unique ID. For the purposes of PISCO physical oceanographic data, a data package consists of one and only one ASCII data file and its associated EML metadata file.

The MetaCat ID will be identical to the metadata file name, minus the extension (last four characters).

Example:

Metadata file name:
FCXXXX_015MXTI008R00_20001015.50.3.XML

MetaCat package ID:
FCXXXX_015MXTI008R00_20001015.50.3

6.6 PISCO ADCP PROTOCOL

This section contains a copy of the PISCO ADCP Protocol.doc document provided by PISCO, referenced as PISCO UCSB ADCP Protocol in Section 5.0.

PISCO UCSB ADCP Protocol

Site Selection and Mounting Frame Installations

Selecting a site for an ADCP deployment takes into account a number of factors. Data quality requires the ADCP be located in an area exposed to the prevailing currents, usually off of headlands or along segments of fairly straight coastline, and unaffected by obstructions such as submerged ledges, isolated rocks, or barrier kelp beds. ADCP sites at PISCO UCSB consist of the bottom mounted (upward looking) ADCP and a vertical subsurface buoyed thermistor string located at least 10m away. The thermistor string is positioned perpendicular to the principal axes current experienced at the site to avoid the possibility of drift kelp caught on the mooring intersecting the ADCP beams. We also try to target deep sand substrate allowing us to easily install the ADCP frame by jetting 10' fiberglass poles into the sand and mounting the frame on the exposed ends. The deep sand substrate also reduces the increasing problem of kelp plants growing up around the ADCP. In areas where only rocky substrate exists the frames are weighted to the bottom with blocks made from a conglomerate of concrete and lead. Considerations for the safety of both personnel and instrumentation require the installations be located in areas that are regularly accessible by divers and in areas unlikely to be affected by boat anchors or fishing traps. Sites are not marked with any sort of surface float except for areas VERY rarely transited by recreational or fishing vessels.

ADCP Deployment and Service

These instructions are to be used in addition to Chapter 4 in the Workhorse Sentinel Technical Manual available at <http://www.rdinstruments.com/>. Topic headings will be given to direct you to the proper paragraph. Since, most likely, the ADCP you're going to be working on has just been recovered from a deployment, we'll start from there.

Recovering ADCP from the field:

Remember to take an instrument case, a yellow transducer protector cap and a garbage bag when embarking on an ADCP recovery. Divers should put the yellow protector cap on before unstrapping from the frame. Large barnacles or other abrasive fouling can be carefully knocked off underwater to reduce the chances of scratching the transducer coating. Check the condition of the frame, clear any drift fouling and check the condition of the zinc every time you visit a site. Once on deck the typically biofouled unit can be placed inside the garbage bag and then into the padded case. Transport the ADCP either vertically or horizontally, NOT upside down, and put it toward the back of the boat to reduce bouncing around. It is very important to record the time the unit was recovered and the unique serial number of the unit in the field notes.

Downloading ADCP data:

Before you do anything else get the data out of the instrument. Refer to the individual units notebook to determine when, where and how the unit was deployed. Setup information will reside in a sequentially numbered folder in the C:\adcp\ directory (nap13 for example). This is where the stored recorder data should be dumped. Connect the serial cable to the COM port of the computer, to the ADCP, and to the DC power supply. The DC power connection ensures that the downloading process continues even if internal battery power is depleted. Start WinSC and click on <file>, <recover recorder data>. Follow the prompts being sure to select the proper folder from the C:\adcp\ directory. Downloading 30Mb of data takes about 1.5 hours.

Check ADCP clock

Before disconnecting the serial connection check the ADCP clock for drift during the deployment. Using WinSC click on [file] [terminal] and then press the [end] key to wake up the instrument. At the prompt> type TS? and hit return. Compare this time to the host computer time (of course set exactly to GMT) and note any differences in the field notes. You cannot unplug the ADCP serial cable and continue.

Check recovered ADCP data:

Once downloading is complete, start the program WinADCP. Click on <file>, <open> and choose the file you wish to look at. Compare the start time and end time with your field notes. Look at the data and see if they look reasonable. Can you see the surface layer? Do the magnitude and direction of the currents look ok? If all's well use WinADCP to determine by inspection the first and last 'good' ensemble number in the deployment. These will be used later to clip the out-of-water data. Now, send the data off to PISCO file server data entry area before you do anything else. Confirm it got there. Now you can move on and service the instrument.

Cleaning exterior:

Biofouling can be quite variable on the exterior of the ADCP depending on the location of the deployment and the season. Care must be exercised to prevent scratching the 4 transducer heads especially if barnacles have settled. Lime-Away works really well at dissolving the barnacle's basal plates. Alternating treatments of gentle scraping, Lime-Away and rinsing will remove the entire test eventually. When rinsing the Lime-Away off of the instrument be sure to get into the gaps between the end caps and the housing body. Bryozoan fouling can be treated the same way. Remember, be patient and really try not to scratch the transducer faces. Algae and hydroids will wipe off fairly easily requiring only a little encouragement. Hydroids will come off especially easily if rinsed with fresh water and allowed to air dry. They seem to essentially turn to dust. Be careful not to tweak, bend, molest or damage the electrical connector on the bottom of the instrument during the cleaning process.

Servicing the ADCP:

All of this information is in Chapter 4 of the Workhorse Technical Manual. I'll just describe the order of events here and other useful info. This procedure is VERY important and should not be rushed. Care MUST be taken to ensure the integrity of the two o-ring

seals. If for some reason you find yourself in a big hurry to turn the instrument around you can omit servicing the transducer head assembly.

- 1) Remove/service the battery end-cap (4-2.1)
- 2) Replace the battery pack (4-4)
- 3) Install the battery end-cap (4-3.1)
- 4) Remove/service the transducer head assembly (4-2.2)
- 5) Install desiccant pack under rubber band on battery pack
- 6) Hook up both internal electrical connectors
- 7) Replace transducer head assembly (4-3.2)

Testing the ADCP:

NOTE: Remove the DC power supply connector from the serial cable. Failure to remove it will prevent you from detecting a lack of internal battery power in the ADCP. Also, check the computer clock is set accurately to GMT. Write the results of these tests in the notebook.

- 1) Using WinSC click on <file>, <test ADCP>. Follow the instructions and look for any failures. Occasionally the narrow beam test will fail. Try the test again while inverting the unit in a bucket of water. Only the transducer beams need be immersed. This usually solves the problem.
- 2) Set the ADCP clock by clicking on <file>, <new deployment>, check the box to “skip planning step”, <next>, DON'T click the “set clock” box and click <next>. The ADCP clock is set. NOTE: The ADCP clock is set to the computer clock, which should be set to GMT. Make sure.
- 3) Erase the old recorder data by clicking on <file>, <erase recorder data>. Follow the instructions. Remember the input is case sensitive (ErAsE). You can also do this step from the terminal mode by entering >re ErAsE.
- 4) If the unit you're working on has a pressure sensor click on [function] [zero pressure sensor]
- 5) Perform the compass calibration (4-6.3) using the angled wooden stand on the bench somewhere. The unit will spin more easily if you put the stand on a piece of cardboard and do the calibration on a dirty gritty bench. Start the procedure by entering terminal mode (<file>, <terminal>), press the 'end' key, and enter AF at the prompt (>). Select 'B' for 'hard and soft' Follow the instructions. You will do 3 rotations. 2 angled, 1 flat.

Launching the unit for deployment:

You have to determine where the unit is going to go and what the next deployment folder number is going to be (e.g., if the last one was PEL13 and the unit is going out to Pelican again then the next folder will be called... you guessed it... PEL14).

- 1) There's a bug in the WinSC software that makes you do these next couple steps. Click on <file>, <new deployment> then click on cancel. This highlights the <Function> option above. Now click on <Function>, <import planning file>. In the C:\adcp\ directory there should be a whole bunch of .whp files. We're using

40_02.whp for the 600khz units along the mainland and islands and CINMS_40_02.whp for the sanctuary sites. See the notebook to find out the latest as some of these units are about to be moved around.

- 2) Select the proper .whp file and then click on <Function>, <deploy>. NOTE: The (now pinging) unit needs to be kept vertical through one ensemble of pings before it is tipped on its side or inverted (about 1 minute). Remove the serial cable from the instrument now but keep the unit vertical for at least 1 minute. I know this is weird but there's yet another bug in the software that requires the first data record to be recorded in the orientation the unit will be deployed in. If the unit is to be deployed right-side-up you have to launch it right-side-up. If up-side-down, launch it up-side-down.
- 3) REPLACE THE DUMMY PLUG.
- 4) Apply Desitin to the transducer heads and put the yellow protective cap back on.
- 5) Put unit into a padded transport box.
- 6) Deploy

Data Processing

We've recently made a change to our processing routine by using a Matlab script to transform the raw RDI format (.000) to a Matlab structured array (.mat) file. We were using the RDI software WinADCP to export the .mat file but this has proven to be much more convenient for batch processing and the exported structures have some very useful configuration data associated with them.

Available .mat file variables:

```

name: 'adcp'
config: [1x1 struct]
mtime: [ens x1 double]
number: [ens x1 double]
temperature: [ensx1 double]
pressure: [ens x1 double]
east_vel: [ens x18 double]
north_vel: [ens x18 double]
vert_vel: [en x18 double]
error_vel: [ens x18 double]
PG1: [ens x18 double]
PG3: [ens x18 double]
PG4: [ens x18 double]
Intens(1-4): [ens x18x(4) double]
Corr(1-4) : [ens x18x(4) double]
Status(1-4) : [ens x18x(4) double]

```

Configuration variables included:

```

name: 'wh-adcp'
sourceprog: 'instrument'
prog_ver: 16.2100
config: '01000001-11001011'

```

```

    beam_angle: 20
    beam_freq: 600
    beam_pattern: 'convex'
    orientation: 'up'
    simflag: 'real'
    n_beams: 4
    n_cells: 18
    pings_per_ensemble: 40
    cell_size: 1
    blank: 0.8800
    prof_mode: 1
    corr_threshold: 64
    n_codereps: 4
    min_pgood: 0
    evel_threshold: 2000
    time_between_ping_groups: 1.5000
    coord: '00011111'
    coord_sys: 'earth'
    use_pitchroll: 'yes'
    use_3beam: 'yes'
    bin_mapping: 'yes'
    xducer_misalign: 0
    magnetic_var: 0
    sensors_src: '01111111'
    sensors_avail: '00111101'
    bin1_dist: 2
    xmit_pulse: 0.9900
    water_ref_cells: [2x1 double]
    fls_target_threshold: 50
    xmit_lag: 0.2600
    serialnum: [8x1 double]
    sysbandwidth: [2x1 double]
    syspower: 255
    ranges: [18x1 double]

```

Each of these files includes all the above variables and only ‘good’ ensembles. Out of water data have been clipped off.

These .mat files provide a good storage format and are directed down a number of paths depending on the need. All files are reprocessed using another Matlab script into a database importable format. These data are essentially raw with a couple corrections. First, because the time stamp for a particular ensemble is reported at the beginning of the series of pings we add an offset equal to $\frac{1}{2}$ the ensemble length to place the time stamp in the middle of the ensemble. Second, we correct for the magnetic variation at each site transforming the data into ‘true’ earth coordinates. The format for the resulting .txt file looks like:

```

date time yearday height depth waterdepth temp_c pressure intensity data_quality eastward northward upwards flag
2004-11-30 19:52:40Z 334.82824329 001.5 -16.1 018.2 13.87 17.738 9999 9999 9999.000 9999.000 9999.000 9999
2004-11-30 19:52:40Z 334.82824329 002.5 -15.1 9999.0 9999.00 9999.000 0144 0100 00.002 -0.004 00.000 100
2004-11-30 19:52:40Z 334.82824329 003.5 -14.1 9999.0 9999.00 9999.000 0145 0100 -0.010 00.031 00.000 100
2004-11-30 19:52:40Z 334.82824329 004.5 -13.1 9999.0 9999.00 9999.000 0138 0100 -0.037 00.022 00.001 100
2004-11-30 19:52:40Z 334.82824329 005.5 -12.1 9999.0 9999.00 9999.000 0132 0100 -0.042 00.048 00.004 100
2004-11-30 19:52:40Z 334.82824329 006.5 -11.1 9999.0 9999.00 9999.000 0128 0100 -0.051 00.072 00.000 100
2004-11-30 19:52:40Z 334.82824329 007.5 -10.1 9999.0 9999.00 9999.000 0126 0100 -0.049 00.060 00.003 100
2004-11-30 19:52:40Z 334.82824329 008.5 -09.1 9999.0 9999.00 9999.000 0122 0100 -0.053 00.050 -0.003 100
2004-11-30 19:52:40Z 334.82824329 009.5 -08.1 9999.0 9999.00 9999.000 0120 0100 -0.062 00.080 00.001 100
2004-11-30 19:52:40Z 334.82824329 010.5 -07.1 9999.0 9999.00 9999.000 0118 0100 -0.043 00.083 00.003 100
2004-11-30 19:52:40Z 334.82824329 011.5 -06.1 9999.0 9999.00 9999.000 0116 0100 -0.037 00.057 00.004 100
2004-11-30 19:52:40Z 334.82824329 012.5 -05.1 9999.0 9999.00 9999.000 0115 0100 -0.069 00.090 00.006 100
2004-11-30 19:52:40Z 334.82824329 013.5 -04.1 9999.0 9999.00 9999.000 0115 0100 -0.068 00.090 00.005 100
2004-11-30 19:52:40Z 334.82824329 014.5 -03.1 9999.0 9999.00 9999.000 0116 0100 -0.076 00.111 00.002 100
2004-11-30 19:52:40Z 334.82824329 015.5 -02.1 9999.0 9999.00 9999.000 0124 0100 -0.058 00.112 00.000 100
2004-11-30 19:52:40Z 334.82824329 016.5 -01.1 9999.0 9999.00 9999.000 0156 0095 -0.055 00.004 00.000 100
2004-11-30 19:52:40Z 334.82824329 017.5 -00.1 9999.0 9999.00 9999.000 0212 0097 9999.000 9999.000 9999.000 100
2004-11-30 19:52:40Z 334.82824329 018.5 000.9 9999.0 9999.00 9999.000 0176 0070 9999.000 9999.000 9999.000 100
2004-11-30 19:52:40Z 334.82824329 019.5 001.9 9999.0 9999.00 9999.000 0125 0085 9999.000 9999.000 9999.000 100
2004-11-30 19:54:40Z 334.82963218 001.5 -16.1 018.2 13.79 17.722 9999 9999 9999.000 9999.000 9999.000 9999
2004-11-30 19:54:40Z 334.82963218 002.5 -15.1 9999.0 9999.00 9999.000 0147 0100 00.016 00.008 -0.001 100
2004-11-30 19:54:40Z 334.82963218 003.5 -14.1 9999.0 9999.00 9999.000 0146 0100 -0.007 00.009 00.008 100
2004-11-30 19:54:40Z 334.82963218 004.5 -13.1 9999.0 9999.00 9999.000 0140 0100 -0.008 00.018 00.007 100

```

Please note that the above format is still in development. We will probably be adding a few columns of data quality information. We'll generate a separate document describing all the features of the database .txt files.

6.7 PISCO ADCP DATA PROCESSING

This section contains a copy of the PISCO_ADCP_data_processing.doc document provided by PISCO, referenced as Proposed PISCO Physical Oceanography Moored ADCP Data Processing Protocol, August 2005 in Section 5.0.

Proposed PISCO Physical Oceanography Moored ADCP Data Processing Protocol

By Patrick Drake (revised 8/24/05)

I Download raw binary data files from ADCP using RDI Self Contained ADCP Application, other RDI software, or directly from PCIMCIA cards.

II Concatenate (combine) multiple files from the same deployment using DOS or Unix command line instructions.

In DOS, “cd” to data file directory and type:
“copy/b filename1 + filename2 newfilename”

III Translate raw binary data into a Matlab structure file using rradcp.m. Save all available data and don't average ensembles. At the end of this stage, the format of the Matlab structure should be:

```

name: 'ADCP'
config: [1 x 1 struct]
mtime: [1 x ensembles double]
number: [1 x ensembles double]
pitch: [1 x ensembles double]
roll: [1 x ensembles double]
heading: [1 x ensembles double]
pitch_std: [1 x ensembles double]
roll_std: [1 x ensembles double]
heading_std: [1 x ensembles double]
depth: [1 x ensembles double]
temperature: [1 x ensembles double]
salinity: [1 x ensembles double]
pressure: [1 x ensembles double]
pressure_std: [1 x ensembles double]
east_vel: [bins x ensembles double]
north_vel: [bins x ensembles double]
vert_vel: [bins x ensembles double]
error_vel: [bins x ensembles double]
corr: [bins x 4 x ensembles double]
status: [bins x 4 x ensembles double]
intens: [bins x 4 x ensembles double]
percent: [bins x 4 x ensembles double]

```

bt_range: [4 x ensembles double]
 bt_vel: [4 x ensembles double]

IV Process and quality-control Matlab structure file. The step contains the “meat” of the processing. This will consist of rearranging the data in the structure, removing any out of water periods, calculating the location of the water surface, masking the velocity bins near and above the surface with NaNs, compensating for magnetic variation, converting to GMT (if necessary), adding some basic metadata to the structure (such as location, depth and yearday), plotting and visually inspecting the data, and saving the processed data into a new Matlab structure file. All this will be accomplished with Matlab codes customized to each campus. At the end of this stage, the Matlab structure should have at least these fields:

bin_depths: [bins x 1 double]	depth below MSL of bins, in meters (float)
bin_heights: [bins x 1 double]	height above bottom of bins, in meters (float)
config: [1 x 1 struct]	unchanged field produced by rdread.m (structure)
corr1: [ensembles x bins double]	correlation of beam 1, in RDI's units (float)
corr2: [ensembles x bins double]	correlation of beam 2, in RDI's units (float)
corr3: [ensembles x bins double]	correlation of beam 3, in RDI's units (float)
corr4: [ensembles x bins double]	correlation of beam 4, in RDI's units (float)
depth_ADCP: [1 x 1 double]	depth below MSL of ADCP temperature and pressure sensors, in meters (float)
east_vel: [ensembles x bins double]	true eastward velocity, in m/s. The transpose of same field produced by rdread.m, but corrected for magnetic variation (float)
error_vel: [ensembles x bins double]	transpose of same field produced by rdread.m, in m/s (float)
file_name: [char]	standardized PISCO PO file name for this file, example: FCXXX_015ADCP014R00_20001015.30.1.MAT (string)
file_name_original: [char]	original file name of RDI raw binary data file (string)
heading: [ensembles x 1 double]	transpose of same field produced by rdread.m, in degrees (float)
heading_std: [ensembles x 1 double]	transpose of same field produced by rdread.m, in degrees (float)
height: [1 x 1 double]	height above bottom of ADCP temperature and pressure sensors, in meters (float)
intens1: [ensembles x bins double]	intensity of beam 1, in RDI's units (float)
intens2: [ensembles x bins double]	intensity of beam 2, in RDI's units (float)
intens3: [ensembles x bins double]	intensity of beam 3, in RDI's units (float)
intens4: [ensembles x bins double]	intensity of beam 4, in RDI's units (float)
lat: [1 x 1 double]	latitude of site, in decimal degrees (include four places to the right of decimal point) (float)
lon: [1 x 1 double]	longitude of site, in decimal degrees (include four

	places to the right of decimal point) (float)
mintens: [ensembles x bins double]	nan-mean of intens1-4 (float)
mtime: [ensembles x 1 double]	Matlab day number (time) of each ensemble (centered for each ensemble) (float)
name: 'adcp_qc'	name of this structure variable (string)
north_vel: [ensembles x bins double]	true northward velocity, in m/s. The transpose of same field produced by rread.m, but corrected for magnetic variation (float)
notes	any relevant info (string)
number: [ensembles x 1 double]	transpose of same field produced by rread.m (float)
percent1: [ensembles x bins double]	RDI's percent good 1 in earth coordinates (float)
percent2: [ensembles x bins double]	RDI's percent good 2 in earth coordinates (float)
percent3: [ensembles x bins double]	RDI's percent good 3 in earth coordinates (float)
percent4: [ensembles x bins double]	RDI's percent good 4 in earth coordinates (float)
pitch: [ensembles x 1 double]	transpose of same field produced by rread.m, in degrees (float)
pitch_std: [ensembles x 1 double]	transpose of same field produced by rread.m, in degrees (float)
pressure: [ensembles x 1 double]	transpose of same field produced by rread.m, but set to NaNs if no pressure sensor installed (not sure about units) (float)
pressure_std: [ensembles x 1 double]	transpose of same field produced by rread.m, but set to NaNs if no pressure sensor installed (not sure about units) (float)
roll: [ensembles x 1 double]	transpose of same field produced by rread.m, in degrees (float)
roll_std: [ensembles x 1 double]	transpose of same field produced by rread.m, in degrees (float)
salinity: [ensembles x 1 double]	transpose of same field produced by rread.m, but set to NaNs if no salinity sensor installed, (in ppt?)
site_code: [char]	PISCO site code, example: FCXXXX (string)
site_description: [char]	Fox Creek: FCXXXX: The nearshore mooring is located ... (string)
site_name: [char]	PISCO site name, example: Fox Creek, OR (string)
surface: [ensembles x 1 double]	instantaneous water depth, in meters, from echo intensity or pressure (float)
surface_method: [char]	two allowed values: 'intensity' or 'pressure' (this field identifies the method used to calculate the "surface" field above) (float)
temperature: [ensembles x 1 double]	transpose of same field produced by rread.m, in degrees Celsius (float)
vert_vel: [ensembles x bins double]	transpose of same field produced by rread.m, in m/s (float)

water_depth: [1 x 1 double]	MSL water depth at site, in meters (float)
yearday: [ensembles x 1 double]	PISCO yearday of each ensemble (centered in middle of ensemble) (float)

Other customized fields are allowed; these are the minimum. (I'm not sure what the "status" field is, so I removed it from the structure. In my data, this field was all zeros. I also got rid of the "bt_range" and "bt_vel" fields. I guess these are bottom-track range and velocity, which are unnecessary for moored instruments. In my data these fields were also all zeros anyway.)

V Rename processed and raw files using standard PISCP PO naming scheme. Raw file names, including dates, should match the processed names and dates. Both dates should reflect the start of good QA/QC data in the file. For the processed files, this will also be the first date in the file. They couldn't be renamed earlier because the "first date in file" (used in our standard PISCO PO names) was unknown until step IV. Renaming can be done manually per file or automated using scripts customized for each campus' native naming convention. Note, each type of file along the processing "pipeline" should have the same name prefix, but a different document ID and extension (see NamingScheme.doc).

Example:

FCXXXX_015ADCP014R00_20001015.10.1.000 (Raw RDI binary data file)
 FCXXXX_015ADCP014R00_20001015.20.1.MAT (Raw Matlab structure file)
 FCXXXX_015ADCP014R00_20001015.30.1.MAT (Processed Matlab structure file)
 FCXXXX_015ADCP014R00_20001015.40.1.TXT (Processed text data file)
 FCXXXX_015ADCP014R00_20001015.50.1.XML (XML metadata file)

VI Create text data in standard PISCO PO ADCP format using standard PISCO-wide Matlab script, ADCP_text_create.m, to be written based on the structure fields given in step IV above.

VII Create XML metadata file, save metadata file locally, and upload metadata and data files to Metacat using Matlab script eml_batch_process.m. This script can either draw the needed metadata info from the configure.m script and the text data files, or it can get everything directly from the processed Matlab structure files.

VIII Data and metadata safely in Metacat in text formats, yeah!

6.8 PISCO TEMPERATURE DATA PROCESSING

This section contains a copy of the PISCO_temp_data_processing.doc document provided by PISCO, referenced as Proposed PISCO Physical Oceanography Moored Temperature Data Processing Protocol, September 2005 in Section 5.0.

Proposed PISCO Physical Oceanography Moored Temperature Data Processing Protocol

By Patrick Drake (revised 9/13/05)

I Download raw binary data files from Onset Stowaway XTI's and Tidbits using Onset's Boxcar Pro, Greenline, or other software.

III Translate raw binary data into text files using Boxcar Pro's export function. The text files should have the format:

mm/dd/yy hh:mm:ss 12.33
mm/dd/yy hh:mm:ss 12.50, etc.

A few header lines are allowable. Use a single space as the field delimiter.

IV Load text data files into Matlab and process and quality-control the data. This step contains the "meat" of the processing. This will consist of rearranging the data in a new Matlab structure (called "temp_qc"), removing any out of water periods, converting to GMT (if necessary), adding some basic metadata to the structure (such as location, depth and year/day), plotting and visually inspecting the data, and saving the processed data into a new Matlab data file. All this will be accomplished with Matlab codes customized to each campus. At the end of this stage, the Matlab structure "temp_qc" should have the following fields:

depth_logger: [1 x 1 double]	depth below MSL of temperature logger, in meters (float)
file_name: [char]	standardized PISCO PO file name for this file, example: FCXXXX_015MXTI014R00_20001015.30.1.MAT (string)
file_name_original: [char]	original file name of raw Onset dtf binary data file (string)
height: [1 x 1 double]	height above bottom of temperature logger, in meters (float)
instrument [char]	make and model of temperature logger (string)
lat: [1 x 1 double]	latitude of site, in decimal degrees (include four places to the right of decimal point) (float)
lon: [1 x 1 double]	longitude of site, in decimal degrees (include four places to the right of decimal point) (float)
mtime: [ensembles x 1 double]	Matlab day number (time) of each temperature

	measurement, in GMT (float)
name: 'temp_gc'	name of this structure variable (string)
notes [char]	any relevant info (string)
site_code: [char]	PISCO site code, example: FCXXXX (string)
site_description: [char]	Fox Creek: FCXXXX: The nearshore mooring is located ... (string)
site_name: [char]	PISCO site name, example: Fox Creek, OR (string)
temperature: [ensembles x 1 double]	logger temperature reading, in degrees Celsius (float)
water_depth: [1 x 1 double]	MSL water depth at site, in meters (float)
yearday: [ensembles x 1 double]	PISCO yearday of each temperature measurement, in GMT (float)

Other customized fields are allowed; these are the minimum.

V Rename processed and raw files using standard PISCP PO naming scheme. Raw file names, including dates, should match the processed names and dates. Both dates should reflect the start of good QA/QC data in the file. For the processed files, this will also be the first date in the file. They couldn't be renamed earlier because the "first date in file" (used in our standard PISCO PO names) was unknown until step IV. Renaming can be done manually per file or automated using scripts customized for each campus' native naming convention. Note, each type of file along the processing "pipeline" should have the same name prefix, but a different document ID and extension (see NamingScheme.doc).

Example:

FCXXXX_015MXTI014R00_20001015.10.1.DTF (Raw Onset binary data file)
 FCXXXX_015MXTI014R00_20001015.20.1.TXT (Raw Boxcar text export file)
 FCXXXX_015MXTI014R00_20001015.30.1.MAT (Processed Matlab structure file)
 FCXXXX_015MXTI014R00_20001015.40.1.TXT (Processed text data file)
 FCXXXX_015MXTI014R00_20001015.50.1.XML (XML metadata file)

VI Create text data in standard PISCO PO temperature format using standard PISCO-wide Matlab script, Temp_text_create.m, to be written based on the structure fields given in step IV above.

VII Create XML metadata file, save metadata file locally, and upload metadata and data files to Metacat using Matlab script eml_batch_process.m. This script can either draw the needed metadata info from the configure.m script and the text data files, or it can get everything directly from the processed Matlab structure files.

VIII Data and metadata safely in Metacat in text formats, yeah!

6.9 PISCO TIDBIT TEMPERATURE LOGGER PROTOCOL

This section contains a copy of the PISCO UCSB Tidbit Temperature Logger Protocol.doc document provided by PISCO, referenced as PISCO UCSB Tidbit Temperature Logger Protocol in Section 5.0.

PISCO UCSB Tidbit Temperature Logger Protocol

Description of the Onset Temperature Loggers:

We are currently using a custom engineered StowAway Tidbit Temp Loggers (Onset Computer Corp. TBIC32+4+27) for our subtidal (mooring) and intertidal temperature measurements. The full temperature range of the off-the-shelf loggers (-5C to +37C) was truncated (+4C to +27C) to improve resolution (0.9 degrees C/bit vs. 0.16 degrees C/bit). After comparing the Tidbit and the XTI models early in our first field season, the XTIs were rejected due to their higher cost, pressure housing maintenance needs, inconvenience in wet environments, and unnecessarily fast temperature response time.

Launching units for deployment:

Note: The Tidbit temperature loggers internal clock is set by the host computer clock as the units are launched. Make sure the host computer clock is set accurately to GMT prior to running the Boxcar software. In addition, the host computer's time zone must be set to an area that doesn't change for daylight savings time. (e.g., Hawaii, Arizona, or Greenwich, England).

- 1) Connect the Optic Base Station to the serial port on the host computer. Run the Boxcar software from the desktop.
- 2) Place a Tidbit Temp. Logger on the Tidbit Coupler. If there's a lot of external light you may have to place a hat or something over the Tidbit.
- 3) There are 3 ways to get to the logger launching dialog box. Click on [logger] then [launch], Click on the little logger icon with the arrow pointing at it, or enter [ctrl][L] from the keyboard.
- 4) The launching parameters should be filled in as follows. The description will be the unit serial number. The sample interval is typically set to at 2 minutes (45 day deployment duration). Always work in Celsius degrees. Don't check the wrap around box or the triggered box. Leave the alarms off and multiple sampling off. Do check the delayed start box and set the date as the future deployment date and the time as 1200GMT. This is done to 1) reduce the time the unit samples out of the water and 2) to synchronize the sampling times. If, for some reason, you need to start the Tidbit logging immediately leave the delayed start box unchecked.
- 5) Click [start] and wait for the unit to launch. On the Tidbit face you will see a little green LED blinking once every 4 seconds in the delayed mode, once every 2 seconds while logging. NOTE: Laptops in power saving mode will sometimes shut off serial communications with the tidbit and appear to have launched it successfully when it really was not. DO check the blinking green light. Onset Computer corp has a patch to fix this problem.
- 6) The Tidbit is ready to launch if the green LED is blinking as described.

- 7) Goto #2 (repeat steps for all necessary Tidbits)

Deploying the Tidbit Temperature Loggers:

The Tidbits can be securely cable tied to just about anything. Ours are attached to vertical subsurface moorings typically deployed in 15m of water. If only tidbits are to be deployed at a site we'll use a 48" sand screw, 5/8 polypropylene truck rope and a couple subsurface 6" lobster floats to create the mooring. If more substantial instrumentation is to be deployed in conjunction with the tidbits (CT, S4s...) we'll sink 500lbs of chain and fabricate a vertical mooring with 1/2" polytron and a 18" soft float.

One notable weakness in the Tidbit design is the tendency for the little optical nipples to break off. We've found that this is usually due to curious crabs with nothing better to do than sit on a Tidbit and pick at it. We now cage the Tidbits in nytex screen to dissuade the pesky crustaceans. Tidbits that have had been damaged cannot be repaired, but the data can be recovered. Call Onset Computer Corp. Technical Support and tell them you want to return a damaged logger for data recovery. They're usually successful and will e-mail you the salvaged data. They charge \$25 for the service.

The delayed start feature can be used to extend the deployment duration of a mooring by having a fresh logger start when another's memory is spent. Do this by setting one unit to start logging on the date (usually 45 days later) a currently logging unit will fill its memory.

Note: It's very important for the field personnel to write down the precise time the loggers are swapped on a mooring and the serial numbers of the units deployed and recovered during that dive. It is easy to mix up the sequence and invert the top and bottom Tidbits. Be careful to keep track of the position of the tidbits and note any uncertainty in the field notes. These data will then be scrutinized during processing. The precise times are also very important, as we need to cut off the portions of the time series where the units are sampling out of the water. This can be done by inspecting each plot around the documented times but during periods where the sea and air temp are similar it's very hard to 'see' when the tidbit was deployed or recovered and deployment times become crucial.

Recovering the Tidbit Temperature Loggers:

- 1) As noted above, write down the precise time (GMT) the loggers are pulled out of the water or moved from their sampling position. Write down the serial numbers of the recovered loggers and their location (T, M, B).
- 2) Clean up the recovered Tidbits with a plastic brush. LimeAway can be used for hard calcareous fouling organisms (barnacles, bryozoans)
- 3) **NOTE:!!!! The host computer clock must be set to a time zone that doesn't recognize daylight savings time. Arizona, Hawaii both work. There's a bug in the Boxcar software and it doesn't handle the time change properly. It'll Spring ahead or Fall back if not set to one of these time zones.**
- 4) Set up the Optic Base Station and start the Boxcar software as before. Place the Tidbit into the Coupler. Cover the Tidbit with a hat or something to minimize external light.

- 8) There are 3 ways to recover the stored data. Click on [logger] then [readout], click on the little logger icon with the arrow pointing away from it, or enter [ctrl][R] from the keyboard.
- 5) Once the readout status bar appears it takes a good 5 minutes to download all the data.
- 6) When complete, you're prompted to enter a file name. File names are sequentially numbered from surface to bottom for each site. Look at the slates or the mooring-field-notes.xls to determine the next available file name/number. Note: It is important to maintain the 3 tidbit sequence (Top, Middle, Bottom) even if data are missing from a particular depth. (e.g., file names xxx001, 004, 007... always designate the top. 002, 005, 008..... always middle). This is important in some processing routines later. After a file name is accepted a graph of the data will appear. If all looks ok go on to the next logger.
- 7) Download all loggers from a particular site in sequence and compare the deployments in boxcar pro by clicking [window] [tile horizontal]. The 3 plots should look VERY similar. Bottom, of course, is typically colder than top and for our sites the bottom data is typically more noisy than the top. Note any suspected inversions in the field notes. When finished, put the data in a folder (named by recovery data) in the data entry area of the PISCO file server.

Data Processing.

Each Tidbit data file (.dtf extension) is converted to an ASCII File (.txt extension) using the export feature in the Boxcar pro software. The export preferences are set to ' Old lotus(pre5.0)/Old excel' and Temp C. The exported files have a one line header over 3 columns of data.

```
Date Time Temperature    (*C)
36800 0.17450 17.96
36800 0.17589 17.47
36800 0.17728 17.96
36800 0.17867 18.12
36800 0.18006 17.96...
```

Date is in some odd excel format (days since Jan 1, 1900). (GMT)

Time in decimal day (GMT)

Temperature Celsius

The .txt files generated above contain the entire time series, even the out-of-water data. These 'bad' data need to be clipped off and this is achieved by running a matlab script that plots the data from an entire deployment, identifies on the ploy the deployment and recovery times as recorded in the field notes. The user must manually click on the desired beginning and end points. The clipped file is then saved out with a .ssv extension.

The data follow a variety of paths from this point depending on the particular need. Files that are going to be processed into the database importable format are loaded into another Matlab script and comes out reformatted like this:

```
date time yearday temp_c flag
2000-10-09 17:01:43.00Z 282.70952000 17.89 0
2000-10-09 17:03:43.00Z 282.71091000 17.89 0
2000-10-09 17:05:43.00Z 282.71230000 18.05 0
2000-10-09 17:07:43.00Z 282.71369000 18.05 0
2000-10-09 17:09:43.00Z 282.71508000 18.05 0
2000-10-09 17:11:43.00Z 282.71647000 18.05 0
```

7.0 APPENDIX B: WCOS-NODC CONDUIT

The WCOS-NODC Conduit is a Java program run via a CSH script file (processPiscoData.csh) that runs on Linux systems. (*Note:* This should extend to other Unix systems, but is currently only deployed on Linux.) The distribution of the files and directories can be deployed in any directory within the system. For the purpose of this document, the deployment directory will be referred to as <ConduitHome>.

The processPiscoData.csh script file is called by a cron job that is set up to run the Conduit once a week. The CSH script file sets up the necessary Java environment and runs the Conduit by executing the program contained in the JAR files. The script accepts one argument, the data type parameter. The value of the data type parameter will either be ADCP or TEMPERATURE. <ConduitHome>/lib contains all necessary JAR files to run the Conduit. The Conduit itself is contained within the file NCDDC.jar, and utilizes a Java properties file (<ConduitHome>/PISCOClientAutoConvert.properties) to establish the general settings that are documented within the properties file. An example of this properties file can be found in Figure 18.

```
#
# URL to the XML RPC server (for metadata translation and publishing).
#
XMLRPCServer
#
# MERMAid directory to put the metadata. (via XML RPC server)
#
MetadataOutputPath
#
# MERMAid catalog ID. (via XML RPC server)
#
MetadataCatalogID
#
# URL to top-level DODS directory.
#
DODSURL
#
# URL to PISCO ADCP catalog.
#
CatalogURLADCP
#
# URL to Temperature PISCO catalog
#
CatalogURLTemp
#
# URL for output. Can be filename or FTP URL.
#
FinalLocationString
#
# File to store catalog of successfully converted data.
#
SuccessFileLocation
#
# Directory to store downloaded data for temporary processing.
#   If a local directory, must include trailing backslash.
#
TempFileLocation
#
# WCO Data Products webapp base URL (protocol, host, and port)
#
WCOWebappBaseURL
#
# Number of threads for processing the datasets simultaneously.
#
ThreadCount
```

Figure 18 WCOS-NODC Conduit Configuration File Example

8.0 APPENDIX C: NCDDC INFORMATION BROKER

The following table describes the mapping of each term from EML to FGDC.

Table 4 Mapping from EML to FGDC

FGDC	EML
idinfo/citation/citeinfo/origin	eml/dataset/creator/individualName eml/dataset/creator/organizationName eml/dataset/creator/positionName
idinfo/citation/citeinfo/pubdate	eml/dataset/pubDate
idinfo/citation/citeinfo/title	eml/dataset/title
idinfo/citation/citeinfo/geoform	'dataTable' if 'eml/dataset/dataTable' is present; 'spatialRaster' if '/eml/dataset/spatialRaster' is present; 'spatialVector' if '/eml/dataset/spatialVector' is present
idinfo/citation/citeinfo/onlink	[value(s) provided by client]
idinfo/descript/abstract	eml/dataset/abstract
idinfo/descript/purpose	eml/dataset/purpose/para
idinfo/timeperd/timeinfo/sngdate/caldate	eml/dataset/coverage/temporalCoverage/singleDateTime/calendarDate
idinfo/timeperd/timeinfo/sngdate/time	eml/dataset/coverage/temporalCoverage/singleDateTime/time
idinfo/timeperd/timeinfo/mdattim/	[not supported]
idinfo/timeperd/timeinfo/rngdates/begdate	eml/dataset/coverage/temporalCoverage/rangeOfDates/beginDate/calendarDate
idinfo/timeperd/timeinfo/rngdates/begtime	eml/dataset/coverage/temporalCoverage/rangeOfDates/beginDate/time
idinfo/timeperd/timeinfo/rngdates/enddate	eml/dataset/coverage/temporalCoverage/rangeOfDates/endDate/calendarDate
idinfo/timeperd/timeinfo/rngdates/endtime	eml/dataset/coverage/temporalCoverage/rangeOfDates/endDate/time
idinfo/timeperd/timeinfo/current	'Unknown'
idinfo/status/progress	'Complete'
idinfo/status/update	eml/dataset/maintenance/maintenanceUpdateFrequency
idinfo/spdom/descgeog	eml/dataset/coverage/geographicCoverage/geographicDescription
idinfo/spdom/bounding/westbc	eml/dataset/coverage/geographicCoverage/boundingCoordinates/westBoundingCoordinate
idinfo/spdom/bounding/eastbc	eml/dataset/coverage/geographicCoverage/boundingCoordinates/eastBoundingCoordinate
idinfo/spdom/bounding/northbc	eml/dataset/coverage/geographicCoverage/boundingCoordinates/northBoundingCoordinate
idinfo/spdom/bounding/southbc	eml/dataset/coverage/geographicCoverage/boundingCoordinates/southBoundingCoordinate
idinfo/keywords/theme/	issued when /eml/dataset/keywordSet@keywordType = 'theme'
idinfo/keywords/theme/themekt	eml/dataset/keywordSet/keywordThesaurus
idinfo/keywords/theme/themekey	eml/dataset/keywordSet/keyword
idinfo/keywords/place/	issued when keywordSet@keywordType = 'place'
idinfo/keywords/place/placekt	eml/dataset/keywordSet/keywordThesaurus

idinfo/keywords/place/placekey	eml/dataset/keywordSet/keyword
idinfo/keywords/stratum/	issued when keywordSet@keywordType = 'stratum'
idinfo/keywords/stratum/stratkt	eml/dataset/keywordSet/keywordThesaurus
idinfo/keywords/stratum/stratkey	eml/dataset/keywordSet/keyword
idinfo/keywords/temporal/	issued when keywordSet@keywordType = 'temporal'
idinfo/keywords/temporal/tempkt	eml/dataset/keywordSet/keywordThesaurus
idinfo/keywords/ptemporal/tempkey	eml/dataset/keywordSet/keyword
idinfo/aconst	'None'
idinfo/useconst	eml/dataset/intellectualRights/para[1,2] if present, otherwise eml/dataset/intellectualRights if present, otherwise 'None'
dataqual/logic	'All data are believed to be logically consistent.'
dataqual/complete	'During the conversion from ASCII to netCDF, no omissions or loss of data occurred.'
dataqual/lineage/procstep/procdesc	'The data has been transformed by NCDDC from its original format, with no loss of information. The transformation was from ASCII tabular format into netCDF (COARDS profile).'
dataqual/lineage/procstep/procd ate	[computed; current system date]
dataqual/lineage/procstep/proctime	[computed; current system time]
dataqual/lineage/procstep/procc ont	[standard NCDDC contact info]
dataqual/lineage/procstep/procd esc	'This metadata record has been derived by NCDDC from its original format, Ecological Metadata Language (EML) version 2.0.1, using the original PISCO EML metadata record as a source. All information contained herein has been derived from the original EML record with the following exceptions: 1) Metadata Contact Information, previously referencing PISCO, has been replaced by NCDDC contact information. 2) Additional Online Linkage elements have been added to the Citation Information element in the Identification Information section. 3) The Distribution Information section has been modified to provide contact information for NOAA/NESDIS/NODC as the data distributor. 4) Two additional Process Step elements (including this one) have been inserted to describe both the conversion of PISCO data from ASCII to netCDF and the creation of this metadata record. 5) Values for Metadata Date and Metadata Future Review Date have been generated using the system clock as there are no analogous elements present in the PISCO EML record.'
dataqual/lineage/procstep/procd ate	[computed; current system date]
dataqual/lineage/procstep/proctime	[computed; current system time]
dataqual/lineage/procstep/procc ont	[standard NCDDC contact info]
eainfo/detailed/enttyp/enttyp1	//entityName
eainfo/detailed/enttyp/enttypd	//entityDescription if present, otherwise 'None provided'
eainfo/detailed/enttyp/enttypds	'Partnership for Interdisciplinary Studies of Coastal Oceans (PISCO)'
eainfo/attr/attrlabl	[computed; value provided by client]
eainfo/attr/attrdef	[computed; value provided by client]
eainfo/attr/attrdefs	'National Coastal Data Development Center (NCDDC)'

eainfo/attr/atrrdomv/edom/edomv	[computed; value provided by client]
eainfo/attr/atrrdomv/edom/edomvd	[computed; value provided by client]
eainfo/attr/atrrdomv/edom/edomvds	'National Coastal Data Development Center (NCDDC)'
eainfo/attr/atrrdomv/rdom/rdommin	[computed; value provided by client; otherwise 'None provided']
eainfo/attr/atrrdomv/rdom/rdommax	[computed; value provided by client; otherwise 'None provided']
eainfo/attr/atrrdomv/rdom/atrrunit	[computed; value provided by client]
eainfo/attr/atrrdomv/rdom/atrrmres	[computed; value provided by client]
eainfo/attr/atrrdomv/udom	[computed; value provided by client]
distinfo/distrib/ctinfo/	[standard NODC contact info]
distinfo/resdesc	eml@packageId
distinfo/distliab	eml/dataset/intellectualRights/para[2,3] if present, otherwise eml/dataset/intellectualRights if present, otherwise 'None'
distinfo/stdorder/digform/diginfo/formname	'netCDF'
distinfo/stdorder/digform/diginfo/formcont	'See Entity/Attribute elements (eainfo)'
distinfo/stdorder/digform/diginfo/transize	[computed; value provided by client]
distinfo/stdorder/digform/digtop/onlinopt/computer/networka/networkr	'http://data.nodc.noaa.gov/nmsp/wcos/'
distinfo/fees	'None'
metainfo/metd	[current system date]
metainfo/metfrd	[current system date + 1 year]
metainfo/metc/ctinfo/	[standard NCDDC contact info]
metainfo/metstdn	'FGDC/NBII Content Standard for Digital Geospatial Metadata (from Ecological Metadata Language 2.0)'
metainfo/metstdv	'1999 Version (from Ecological Metadata Language 2.0)'

9.0 APPENDIX D: DATA PRODUCTS WEB APPLICATION

The webapp has been deployed on Tomcat 5.5.9 and requires Java 5.0 to compile the JSP pages. (*Note:* Java 5.0 is required instead of 1.4 because the JSP pages are not compatible with 1.4; therefore, additional configuration of Tomcat is required to be run on Java 5.0. It is recommended that Tomcat 5.5.12 be used to avoid having to make additional configurations). The webapp may be installed in an arbitrary directory, providing the appropriate “docBase” attribute is set for the “/wco” context in Tomcat’s server.xml file. Webapp parameters are set and described in the web deployment descriptor file – wco/WEB-INF/web.xml.

Upon startup, the webapp acquires a catalog of available data files from NODC’s OPeNDAP server by checking whether or not a local file contains that information. If so it uses it as the catalog. If the file does not exist, the catalog is generated by making a series of HTTP requests to the NODC OPeNDAP server for the information and then saving the catalog to the DODSCatalogFile file. (*Note:* To force the webapp to access the NODC OPeNDAP server, delete the local file and restart the webapp.) The catalog is refreshed on an interval specified by the DODSCatalogUpdateInterval parameter – currently set to one day. The NODC OPeNDAP servers URL is specified with the DODSBaseURL parameter from the web.xml file. Figure 19 provides an example of this web.xml file.

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>

  <context-param>
    <param-name>DODSBaseUrl</param-name>
    <param-value>http://data.nodc.noaa.gov/cgi-bin/nph-dods/nmsp/
wcoc/</param-value>
    <description>
      URL to top level of DODS server.
    </description>
  </context-param>

  <context-param>
    <param-name>DODSCatalogFile</param-name>
    <param-value>C:\tmp\WCO\DODSCatalog.ser</param-value>
    <description>
      Name of file to hold externalized DODSCatalog object file.
    </description>
  </context-param>

  <context-param>
    <param-name>imageType</param-name>
    <param-value>png</param-value>
    <description>
      Image type for JFreeChart.
    </description>
  </context-param>

  <context-param>
    <param-name>DODSCatalogUpdateInterval</param-name>
    <param-value>1</param-value>
    <description>
      Time interval (in days) between DODS Catalog updates.
    </description>
  </context-param>

  <servlet>
    <servlet-name>dataProducts</servlet-name>
    <description>WCO Data Products servlet</description>
    <servlet-
class>gov.noaa.ncddc.WCODataProducts.WCODataProductsServlet</servlet-class>
    <load-on-startup>2</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>dataProducts</servlet-name>
    <url-pattern>servlet/dataProducts/*</url-pattern>
  </servlet-mapping>

</web-app>

```

Figure 19 Webapp Parameters Example File