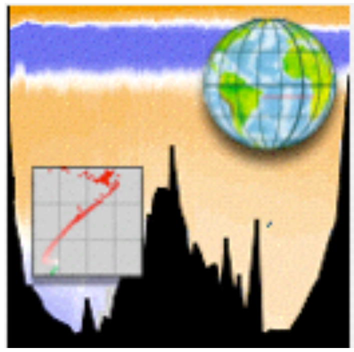
Java OceanAtlas



User Guide

Third Edition

James H. Swift
UCSD Scripps Institution of Oceanography
Mail Code 0214
9500 Gilman Dr.
La Jolla, CA 92093-0214, USA
tel: 858-534-3387
fax: 858-534-7383
email: jswift@ucsd.edu

John R. Osborne
OceanAtlas Software
8010 SW Hawthorne Ln.
Vashon Island, WA 98070
tel: 206-567-0065
email: tooz@oceanatlas.com

Java OceanAtlas development was partly supported by a grant from the US National Science Foundation. Additional Java OceanAtlas development was supported by the National Oceanic and Atmospheric Administration.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

About the User Guide

The Java OceanAtlas User Guide was written by James Swift, who is a research oceanographer and academic administrator at the UCSD Scripps Institution of Oceanography and John "Oz" Osborne, the Java OceanAtlas developer. Considerable assistance was provided by former UCSD student Jeremy Weir.

The User Guide is an attempt to document Java OceanAtlas features in an indexed and consistent manner, a goal which in some ways is nearly out of reach. Java OceanAtlas features are added or evolve over time, and there are many subtle aspects of the application, which may escape description, especially in early editions of the User Guide. The second edition of the User Guide is based on Java OceanAtlas 5.0. New features available in more advanced versions of Java OceanAtlas may or may not be reflected in the edition of the User Guide you have available. We ask your patience.

Java OceanAtlas was designed to adhere to the native look and feel of the various computer operating systems it is installed. All examples in the first edition of the User Guide are taken from the Mac OS X version of Java OceanAtlas 5.0, and so reflect that particular user interface style.

Please email to jswift@ucsd.edu your suggestions for corrections, additions and improvements to the User Guide.

Introduction to Java OceanAtlas

Java OceanAtlas is a Java application - for nearly all computers - which provides an *ad hoc* (rather than pre-set) graphic exploration environment optimized to examine and plot oceanographic profile data, especially profile data organized into what oceanographers call 'sections'. Java OceanAtlas was designed to work well with the Atlas of Ocean Sections data sets (see <http://podaac.jpl.nasa.gov/order/order_displaytools.html>) or the WOCE Hydrographic Program 'WHP-Exchange' format data holdings (see <http://whpo.ucsd.edu> or the WOCE Version 3 DVD data set), functioning as a stand-alone living atlas of oceanographic sections. [OceanAtlas data files for use with Java OceanAtlas can be also be found at <http://odf.ucsd.edu/joa/>.] Additional data of arbitrary parameter composition can be imported at any time, including by direct reading of spreadsheets and WHP-Exchange, NODC, or netCDF format data files (EPIC, WOCE, and ARGO conventions). JOA 5.0 has the capability to access on-line oceanographic data archives and create on-the-fly databases of summary statistics for any profile type that JOA can read whether on your desktop PC or online data archive.

Java OceanAtlas will work with any pressure-indexed data. This includes not only traditional hydrographic parameters such as temperature, salinity, oxygen, and nutrients, but also *any* measured parameter that can be associated, at least for Java

OceanAtlas purposes, with a measurement level (fish egg profiles, pigment profiles, contaminant profiles, etc.).

Java OceanAtlas plots include property-property plots, offset profiles (waterfall), contour plots, map contour plots, line plots of "station" calculations (depth of mixed layer and integrals) and map plots, using color-by-property-value as a plotted variable to aid interpretation. There is also a comprehensive data display window and extensive support for data editing and quality control functions. All Java OceanAtlas plots and the data window are linked and may be ‘browsed’ by sample and/or by station. Plots can be re-scaled, re-sized, or have their colored variable changed. Selected areas of profile, property-property, contour plots, maps, and station value plots can be extracted into new plots. Standard levels, scales, contours, and colors can be changed via user interfaces similar to those used in commercial applications. Java OceanAtlas provides data filtering and exporting. Many different types of calculations can be performed. Double-clicking on Java OceanAtlas plots brings up a dialog to make changes.

Oceanographers familiar with computers but new to Java OceanAtlas can get started without instructions once the application is installed. Start the application and open a data set. A data display window and a menu bar full of possibilities will soon be available. Pull down *Plots* and try something. Bon voyage!

The core ".jar" files used by Java OceanAtlas are updated as needed to fix bugs and add features. See <http://odf.ucsd.edu/joa/> for information on upgrading JOA.

Finally, users should read through the documentation someday. This may turn up something interesting or useful such as a new (to the user) feature or an explanation for some strange-looking plots.

OceanAtlas Web Sites

John 'Oz' Osborne, the Java OceanAtlas developer, operates an OceanAtlas Web site at <http://www.oceanatlas.com/>. This is oriented toward information about the application and his company, OceanAtlas Software, and also contains useful links to other oceanography-oriented Web sites.

James Swift, the scientist who helped develop Java OceanAtlas, operates the official OceanAtlas Web site at <http://odf.ucsd.edu/joa>. This contains information about Java OceanAtlas, all current Java OceanAtlas downloads and updates, and also contains many data files.

Limitations of Java OceanAtlas

Java OceanAtlas is a section-oriented application that works from vertical profile data but also supports horizontal contouring of isosurfaces and station calculation on maps.

Java OceanAtlas can use prodigious amounts of computer memory (RAM), especially when importing full-resolution CTD data; in which case at least ca. 3 MB per CTD profile are needed above normal application requirements (typically ca. 35 MB for a 100-station WOCE bottle data file with a map plot, two property-property plots, and two contour plots open).

Printing plots in Java OceanAtlas is WYSIWYG, the major compromise being placement of plot on page, margins, etc. So while printing directly from Java OceanAtlas may not be the preferred method for obtaining the highest quality output, it is quite functional. The major advantage of internal printing is that the underlying vector resolution of the plot is retained rather using the raster (bitmap) representation which results from most screen capture programs. To enhance printed output, save the plot as PNG file and import it into a vector-oriented drawing program. Or plots can be “printed” directly to Postscript files or PDF files (Mac OS X only).

Steps to Become a Java OceanAtlas ‘Power User’

1. Learn to modify plots. All Java OceanAtlas plots can be modified after they have been created. Try ‘double clicking’ (or right-button clicking if you are using JOA under Windows, UNIX, or Mac OS X) on a plot (or almost any part of any Java OceanAtlas window). This brings up a dialog from which to carry out many customizations.

2. Learn to edit and create Color/Contour bars (called just *colorbars* from now on). This is one of the great feature areas of Java OceanAtlas and not hard to use. Using customized colorbars with contour plots (or zero-amplitude profile plots) is a great way to get more out of Java OceanAtlas. Autoscaled colorbars in many areas allow for more fluid data exploration.

3. Learn to use the Filters. Station, observation, and data range filtering permit the user to focus on the data of greatest interest. Again this is not especially difficult.

4. Learn to export and import data and create custom data files. With Java OceanAtlas directly importing spreadsheet data and WHP-Exchange, NODC, and netCDF format data it is easy to get data into Java OceanAtlas. Hint: If there is trouble with spreadsheet data import, first export a small file in spreadsheet format then use this as a template to guide preparation of spreadsheet data for import.

About the Colorbars Included with Java OceanAtlas

Included with Java OceanAtlas is an assortment of color/contour bars that are useful in examining routine hydrographic data (T, S, O_2, NO_3, PO_4, SiO_3). These mostly follow the coloring schemes used by Prof. Joseph L. Reid of UCSD/SIO. Most are 'two color transitions', e.g., the most extreme low value of the parameter is in the deepest shade of Color 1 to less well saturated shades of Color 1 to less well saturated shades of Color 2 to the deepest shade of Color 2 at the extreme high value of the parameter. The color pairs used for each parameter approximate Reid's choices. This is quite different from the vivid rainbows now in vogue in the computer world. The rainbows print nicely on white but the two color transitions are informative and pleasing to the eye.

In addition to color/contour bars for the most commonly used parameters a few extras are provided. 'NUTR' is designed for the NUTrient Ratio NO3/PO4, which can be calculated using the *Custom Parameter* calculator under the *Calculations* menu. To use this color bar name the NO3/PO4 result 'NUTR'. Similarly 'NOPO' is made for the ratio of NO divided by PO (see *Parameters* under the *Calculations* menu for these.) The 'BVF1' color bar will work with the result of the 'N' calculation in the *Parameters* calculation dialog under the *Calculations* menu.

Other Useful Information

Java OceanAtlas is very selective about file name suffixes in general when reading or importing a data file. Be sure that the correct file name suffixes are used for each data type:

.joa	Java OceanAtlas binary
.jos	Java OceanAtlas spreadsheet ASCII
.sd2	NODC SD2 format ASCII
_hy1.csv	WHP-Exchange format bottle data ASCII
_ct1.csv	WHP-Exchange format CTD data ASCII
_ct1.zip	WHP-Exchange format CTD data zipped directory
.nc	netCDF Profile (JOA accepts netCDF files that adhere to the EPIC (PMEL), ARGO, and WOCE conventions)
.nqdb	JOA database document file
.ptr	EPIC pointer file

When creating a new data file to read into Java OceanAtlas, be certain that one of the parameters is pressure (or depth). (Java OceanAtlas has a built-in parameter name lexicon and will understand many common names for pressure; it's built-in name, which always will work, is 'PRES'.) This is the only profile parameter that Java OceanAtlas requires. If the data are indexed by depth instead of pressure, convert depth to pressure, or, if a ≈1% error is no problem, simply relabel the depths as PRES as an interim measure.

Java applications can use very large amounts of computer memory. Failure in data import, especially for CTD data, can sometimes be traced to low memory.

Bugs

Repairing software problems (‘bugs’) is an ongoing effort, but as the complexity of OceanAtlas has grown, the opportunities for such problems have grown apace. Also, some existing features may, on occasion, develop bugs in newer versions. Through months of testing for each new version, bugs are found and repaired. However some probably slip by. Generally speaking, for greatest stability try to carry out each Java OceanAtlas session in an orderly way, for example opening, adding, and editing data files first, then calculating, then plotting

Technical Support

Technical support is handled via electronic mail and the Internet. Send all technical questions to John (‘Oz’) Osborne <mailto://tooz@oceanatlas.com>. He will make every attempt to resolve your problem as expediently as possible. Include the following information in your email message:

- Brief description of problem
- Repeatability of problem (a list of steps that reproduce a problem is very valuable)
- Version of Java OceanAtlas
- Version of Java Virtual Machine (in the small type above the sponsor logos in the *About Java OceanAtlas* window)
- Operating system version
- Amount of RAM in machine
- Data set specific problem? (In which case be prepared to send the data file to Oz)

Science issues, suggestions for the User Guide, or data file questions may be addressed to Jim Swift <mailto://jswift@ucsd.edu>.

OceanAtlas Software maintains a Web home page at <http://www.oceanatlas.com/>. There is additional OceanAtlas information including a ‘FAQ’ page to answer common OceanAtlas questions. There is also a home page at UCSD/SIO, which includes on-line access to data and application files <http://odf.ucsd.edu/joa/>.

One may obtain any of the Java OceanAtlas application, text, or data files from the Physical Oceanography Distributed Active Archive Center at JPL, only by internet: <http://podaac.jpl.nasa.gov>.

Java OceanAtlas History

Graphic representations have long been the key to examining scientific data. Appropriate plots permit the eye to pick out the salient features of the data. The original inspiration for our development of OceanAtlas (the predecessor to Java OceanAtlas) was the great printed atlases of maps and sections found in oceanographic libraries and in many oceanographers' offices. The 'Atlas of Oceanographic Sections, temperature - salinity - dissolved oxygen - silica, Davis Strait - Labrador Basin - Denmark Strait - Newfoundland Basin, 1965-1967', by A.B. Grant [Report Atlantic Oceanographic Laboratory, 68-5: 80pp, 1968 (unpublished manuscript)] was particularly interesting because the author wisely printed some of the most critical data values on the contoured sections, allowing the reader to compare parameters from the same water sample by paging through the sections. What if this sort of data exploration could proceed in an even more nearly *ad hoc* fashion, despite the ever-increasing volume of data we now have?

The evolution of the microcomputer with color display and the wealth of applications for these computers put onto the desktop the capacity to hold and examine large quantities of ocean data. When we saw the early Apple Macintosh computers with their simple and consistent graphic-oriented operating system we began outlining a proposal to move to the Macintosh the mainframe data holdings, search engine (still known as 'Hydrosearch' to the users of the present nemo.ucsd.edu computer) and calculation and plotting software used by Professor Joseph L. Reid at the Scripps Institution of Oceanography. Our original proposal for *OceanSearch* was thankfully turned down – although with some interest – by the US National Science Foundation and the US Office of Naval Research. 'Thankfully' because in retrospect we could not have achieved our aim, at least on the Macintosh computers of that era and with the time and budget we had proposed.

It was Peter Rhines' demonstration of his IBM-PC application *Atlast* which rekindled our imagination and drive. His inspiration to work with section-oriented data, avoiding the pitfalls inherent in mapping data onto surfaces, was just what we needed, especially when one realizes the wealth of section data to be found in the archives. Literally the same day that we saw *Atlast* we were at work on OceanAtlas for Macintosh. Naively we thought that 'we would simply port Atlast to the Macintosh'. Unfortunately, graphics-oriented applications such as these are all interface and graphics calls – just the sorts of programming issues that are hardware-specific. By the time we completed OceanAtlas 1.0 in early 1991 only the algorithms for potential temperature and density remained! And the applications had already diverged, each taking advantage of the strengths of its native operating system.

Computer-based applications do not replace hard-copy plots. A set of well printed, carefully and similarly prepared plots can be most useful, and if one's own, can be annotated and used profitably over a period of years. But much information can be lost on the way to a good plot and we often yearn to explore other avenues glimpsed in the plots. As we have all learned the computer can enable us to examine data in new, flexible ways, and permits us to take new directions.

The wealth of measured and calculated parameters useful to the study of the origin and circulation of ocean waters suggested the value of a set of simple tools to examine on screen whatever facets and features are of interest to the examiner. Our intent is to provide an educational tool for *ad hoc* real-time visualization of oceanographic data, to permit easy exploration of the oceans at low cost and in new ways. OceanAtlas is not a data presentation program, but is more nearly a data exploration program.

OceanAtlas went public with version 1.0 in early 1991 and evolved through version 2.5.1 in 1994. This was all carried out with bits and pieces of support, mostly from NSF and NOAA. Development support dried up in early 1995 just at a time when we were ready to make the Next Big Step: moving OceanAtlas onto PowerPC Macintosh computers in native mode. Osborne went ahead on his own time with the project, with the intent of selling copies to recoup part of his costs. Fortunately NSF stepped in at the last minute with support for a plan to put the new application, Java OceanAtlas 1.0, into the public domain.

About Ocean Sections

One of the methods oceanographers utilize to collect information about the origin and circulation of ocean waters is to occupy – from a research vessel – a line of ‘stations’ across a region of interest. Such a line is usually called a ‘section’. A ‘station’ is a location where the ship stops and takes observations, in this case of the physical/chemical characteristics of the water column. (There are many different types of oceanographic expeditions and observations. Here we focus on the types which produce the data Java OceanAtlas was originally designed to explore.)

Water column measurements made from the end of the 1800s through the 1970s were typically made by attaching water bottles (with open lids) onto a wire lowered into the ocean. When all the bottles were deployed at the intended levels a ‘messenger’ (brass weight) was dropped down the wire. When a messenger reached a water bottle it activated mechanisms that closed the bottle, caused a deep-sea reversing thermometer to reverse (and thus break the mercury column and preserve its temperature reading), and release another messenger to head down to the next bottle.

Beginning in the early 1970’s a new device, the rosette water sampler, began to replace bottle-on-wire casts. The rosette is frame which holds a CTD (an electronic profiling device which samples Conductivity, Temperature, and Depth), 12-36 water sampling bottles, a central closure-control mechanism referred to as a pylon, and sometimes other equipment. The frame is attached to an armored conducting cable, and can thus be lowered into the ocean. An operator in the ship's laboratory can communicate with the underwater package via the conductor in the cable. Typically the CTD profiles are collected as the rosette descends and the water samples are collected (by closing the bottles by electronic remote control) on the up cast, along with CTD values at the times each bottle is closed. The CTD typically supplies the temperature and pressure (and conductivity) at which a bottle is closed, and laboratory analyses for other parameters are made from the water collected in the closed bottles. Typically – but not always – water samples are analyzed for salinity, oxygen, and the inorganic nutrient salts nitrate (NO3; usually analyzed as 'NO3 + NO2', with NO2 – which is usually very small – being subtracted out after running analyses for it alone), phosphate (PO4), and silicate (SiO3). In recent years increasing attention has been given to analyses of what some oceanographers call 'tracers' (although temperature, salinity, oxygen, and nutrients made excellent tracers, too; it's just habit and nomenclature). These include substances such as CFCs, tritium, helium, radiocarbon, and so on. The list recognized by the WOCE Hydrographic Program Office contains 48 parameters plus 12 ancillary measurements.

Basin-scale studies require basin-spanning data. And so many of the great oceanographic sections cross major basins include full-depth stations whose horizontal and vertical resolution are reasonably well matched to large ocean scales, and include the full suite of routine hydrographic data. These form the backbone of the historical ocean data sets included on the Atlas of Ocean Sections CD-ROM (and also on <http://odf.ucsd.edu/joa>). We tend to focus on the bottle data in OceanAtlas because these are relatively compact, cover the large-scale fields well, and include many parameters of interest.

Ideal data files for an oceanographic atlas of sections would consist of full-depth basin-scale sections of hydrographic data crossing each of the primary ocean regions both meridionally (south-north) and zonally (west-east), with special attention paid to direction and station spacing in the vicinity of bathymetric features and boundaries. Profiles and samples would be spaced horizontally and vertically in such a manner as to resolve the features of interest, and sufficient parameters at appropriate quality would be available to identify and trace the various layers of water masses and their spreading, and to compare with the other sections and regions. We tend to think of the minimum suite of required parameters as temperature, salinity, oxygen, and nutrient measurements, although much can be learned by examining the distributions of other substances, particularly anthropogenic substances or natural isotopes subject to decay over a characteristic time scale appropriate to circulation processes.

Many hydrographic sections are available or can be assembled piecemeal from the historical data files. However, prior to the WOCE Hydrographic Program in the 1990s in most regions there were few, if any, which met the ideal criteria. For example, the North Atlantic sections distributed with the first release version of OceanAtlas were relatively complete, but coverage was much sparser (but still useful) in the South Atlantic, and there were only a few sections each in the North and South Pacific, and two in the Indian Ocean. From some points of view there were more gaps — many in interesting places — than covered regions.

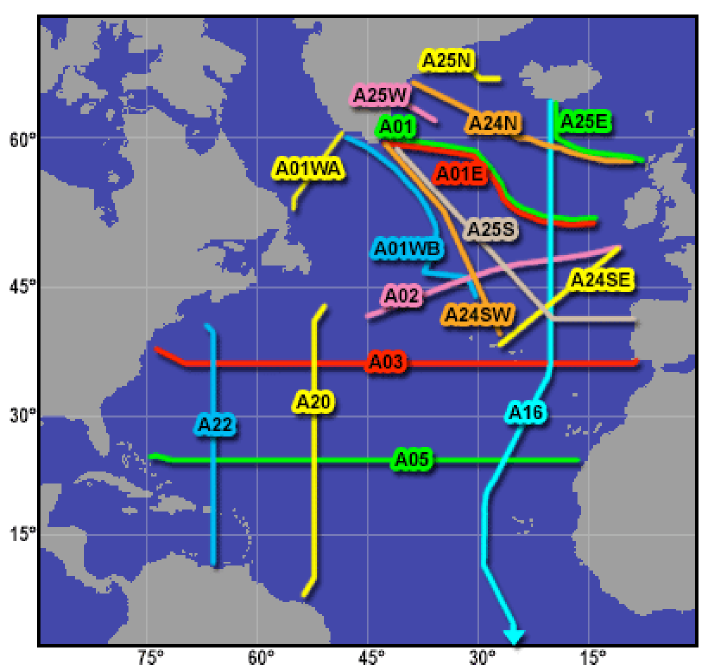
Another issue, and one rarely addressed in the literature, is data quality. One factor is the accuracy and precision of the data. The accuracy and precision feasible for each parameter have improved over time as methodologies have evolved or as new ones are introduced, and can be considered together with the normal oceanographic range for each parameter. For example, seawater temperature has been measurable to ca. ±0.02°C since the introduction of the oceanographic precision reversing thermometer in the late 1800s. Of course the best instruments were rare and expensive, and calibration must be made to recognized international standards, but since the ocean temperature range is about -2 to 30°, oceanographers have had the potential to make temperature measurements with a reasonable signal-to-noise ratio for longer than one century. [Seawater temperature is now routinely measurable at sea to ca. ±0.002°C with the platinum resistance thermometers on modern CTDs, and a properly outfitted calibration facility can achieve about ±0.0002°C accuracy ashore.] But at what level (at what depth) is the thermometric temperature measurement valid? Until the introduction of the CTD, the depth associated with a bottle closure was determined by charting the length of wire deployed between sample bottles along with depths determined by comparing the readings of thermometers protected from pressure with those unprotected from pressure. Not every water bottle on the wire had an unprotected thermometer. The accuracies and precisions - in depth space - of the unprotected thermometers are issues to be considered. And what of the frequency and quality of the calibration of these instruments? Also, one must consider that reading thermometers to hundredths of a degree was an activity itself prone to errors, as were the procedures for correcting the readings for various effects such as the thermal expansion of the thermometer glass via an internal thermometer which read the inside temperature of the protected thermometer itself. And so forth. Just the fact that a certain type of reversing thermometer itself was used is no assurance, on its own, that the data obtained would be of a specific quality. [The same is true for nearly all measurements, of course. The skill and knowledge of the analyst, along with the resources available to the analyst - such as a particular model of salinometer with specified maintenance procedures and specific reference standards - are keys to obtaining the highest feasible quality of measurement.] The best expeditions of the past included a hierarchy of procedures to guard against error and improve data quality, just as the best expeditions of today follow their own proven procedures. Still, measurement is a human activity and prone to human error. How many times does the salinity analyst, having worked a long day on a rolling ship, reach for the next salinity bottle in sequence, and unknowingly grab the wrong bottle, and not notice the mismatch in numbers, and thus produce an erroneous salinity value, however well the original sample was drawn and stored and however well the salinometer is operating?

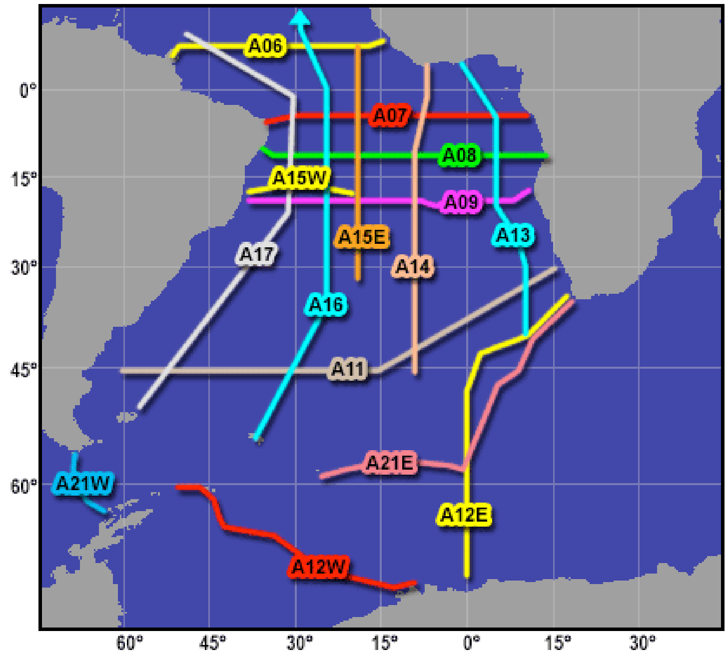
Experience with the water column profile data held by the US National Oceanographic Data Center shows that there are occasional 'flyers' in many ocean data. Sometimes this is a matter of a missing data value being written as a zero, instead of the '-9' value used by NODC. Sometimes the value is a measurement error of some sort. And sometimes the odd values are genuine oceanographic anomalies. It takes years of experience to sort out such matters. The best place to quality control data is at sea, where memories are fresh, the records are at hand, and all personnel involved are together.

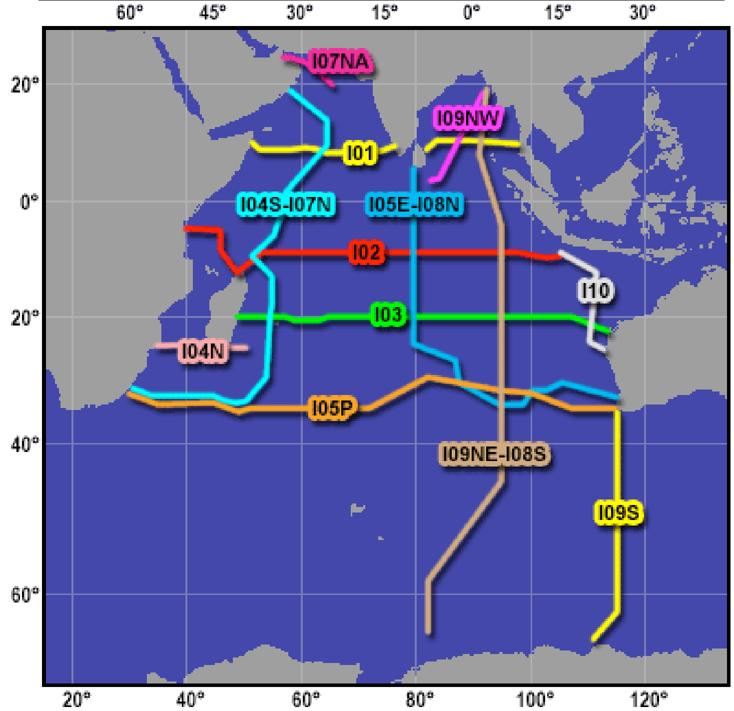
For several decades many oceanographers agreed on the broad outlines of what would constitute a global description of the oceans to a specified set of modern standards, but progress was slow. This was partly because it is tedious and expensive to obtain global oceanographic data, and also because there were many other pressing and interesting oceanographic problems to explore with the resources available. Climate and global change research brought new urgency to understand the oceans' role in the overall climate system. Many years of planning brought forward the World Ocean Circulation Experiment of the 1990s, which was part of the World Climate Research Program. One of the largest elements of WOCE was the WOCE Hydrographic Program, during which basin scale, reference-quality, multi-tracer oceanographic sections were occupied on more than 100 cruises as part of what was called the WHP 'One-Time Survey', and an additional 400 or so cruises were carried out as part of the WHP 'Repeat Hydrography' program. [See the WOCE Version 3 DVD set or the site <http://whpo.ucsd.edu> for information about and data from the WOCE Hydrographic Program.]

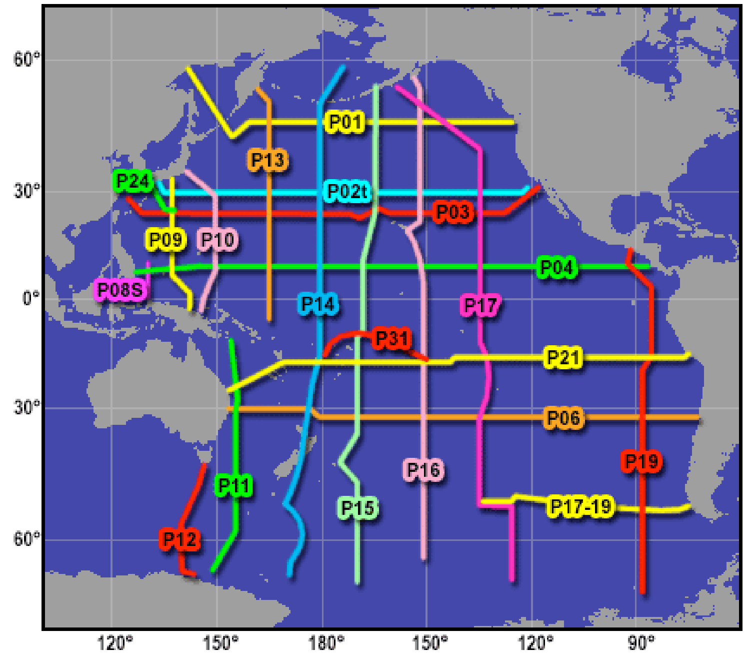
Most WOCE Hydrographic Program One-Time survey sections were sampled at 50km horizontal resolution — closer over bathymetric features so that depth did not change more than 800-1000 meters between adjacent stations — and many utilized a 36-place rosette water sampler for good vertical resolution. For example the 24°N Pacific section comprises approximately 7,000 water samples, and is a delightfully complete zonal view of the subtropical North Pacific.

All the WOCE Hydrographic Program data are available on the WOCE DVDs and CD-ROMs, or from the site <http://whpo.ucsd.edu>. The author has prepared an assembled sub-set of the WOCE Hydrographic Program One-Time Survey sections, joining long lines that were carried out by multiple cruise legs and eliminating overlapping data. Those sections, available from <http://odf.ucsd.edu.joa> or on the Atlas Of Ocean Sections, Version 2, are illustrated in the maps below:




A06
A07
A08
A15W
A09
A15E
A13
A17
A14
A16
A11
A21E
A21W
A12E
A12W
0°
15°
30°
45°
60°
60°
45°
30°
15°
0°
15°
30°


I07NA
I09NW
I01
I04S-I07N
I05E-I08N
I02
I10
I03
I04N
I05P
I09NE-I08S
I09S
20°
0°
20°
40°
60°
20°
40°
60°
80°
100°
120°



Java OceanAtlas 5.0 Features

Java OceanAtlas (JOA) is a software application for viewing and manipulating oceanographic profile data. JOA was designed primarily for oceanographic sections but is useful for looking at data also in the latitude-longitude domain.

Platforms

JOA should run on the following Java-enabled platforms:
- Windows 95, 98, ME, NT, 2000*, XP, Not tested on Vista
- Mac OS X*
- Solaris
- Linux
- Other UNIX OSes with at least JDK 1.4.2 support

Note: JOA has not been tested on a wide variety of UNIX platforms or extensively on all flavors of Windows. JOA has been used extensively with the operating systems marked with an asterisk (*).

Input/Output

The table below lists the current input and output file formats supported by JOA.

Format	Read	Write
EPIC netCDF (bottle and CTD)	X	X
WOCE netCDF (bottle and CTD)	X	
WOCE “Exchange” Bottle	X	
WOCE “Exchange” CTD	X	
Zip files (all supported formats)	X	

EPIC Pointer	X	X
NODC SD2	X	
Tab-Separated Value	X	X
JOA Binary	X	X
PNG (plots only)		X
PostScript (via printer drivers)		X
PDF (Mac OS X only)		X
JOA Database Documents	X	X
NODC Argo Inventory files	X	
NODC GTSPP Inventory files	X	

NdEdit Data Browsing

NdEdit Features

- Read EPIC pointer files generated by JOA and EPIC web data browsers
- Browse gridded netCDF files and extract profiles
- Browse multiple pointer or netCDF files simultaneously
- Up to six views of stations locations (and/or grid points) in space and time:
  latitude/longitude
  latitude/depth
  latitude/time
  longitude/depth
  longitude/time
  depth/time
- Optional bathymetry display in latitude/longitude view
- Zooming to specific regions of interest in location and time
- Interactive filters for latitude, longitude, depth, and time allow narrowing data collection to region/time of interest
- Tools for selecting stations in regions or in user- defined sections
- Optional selection “Inspector” window
- Open selected stations directly into JOA

Dapper Data Browsing

- Dapper wizard for browsing, subsetting, and downloading data from Dapper-enabled OPenDAP online data archives

Data Management

Section Manager

- Reverse stations
- Merge casts at same section

ª Merge all sections into one section

- Sort stations by latitude, longitude, date, station number
- Add ocean bottom from etopo bathymetry files

Section Data Editor

- Edit station metadata
- Edit individual parameter data values

ª Set bottle/observation quality code

- Set sample quality code
- Set selected values to missing

Station Filter
- Include/exclude individual stations/sections
- Include/exclude stations by geographic location using zoomable map
- Exclude stations by absence of measured parameter

Observation Filter
- Up to 4 selection criteria that can be grouped with and/or logic
- Test range of parameter or quality code values
- Results apply to all open property, profile, and contour plots
- Highlight results with contrasting symbols/color or show only observations that meet filter criteria
- Settings savable to settings files
- Interactive plot “enhancement” to filter by the color parameter

Data Quality
- Filter plots by quality codes
- Highlight outliers by quality code or value (contrasting symbol and/or color)
- Display observation quality codes
- Display bottle quality codes in Data Window
- Data Window cross section can show presence/absence of parameter

File Properties
- Edit station metadata: section name, station number, cast number, station location, station date, station maximum depth, station ship code, file comments, map color

Parameter Properties
- Edit parameter name, parameter units, parameter y-axis orientation

WOCE Data Import
- Post process WOCE exchange files:
- Translate WOCE quality codes to IGOSS.
- Convert temperatures from ITS90 to IPTS68.
- Convert from mass to volume units.
- Replace value with missing if quality code criteria met.
- Set all bottle parameters missing if bad bottle quality code.
- Set gas parameters missing when bottle quality is questionable and bottle oxygen is bad.

Spreadsheet (CSV and TSV) Data Import/Export
- Preview window on import
- Choice of field delimiter and missing value indicator
- Support for NODC Argo delimited files
- Station observations are automatically sorted by pressure (as needed) to be in order of increasing pressure.

Export as WOCE Exchange File
- Select station subset
- Date stamp
- Edit/add comments

CTD Decimation
- No decimation
- Decimate to constant interval

- Decimate to JOA interpolation levels
- Decimate to custom, user-defined intervals
- Save/Load custom decimation schemes

Merge Editor

- Merge contents of delimited text file into current dataset

JOA On-the-Fly Databases

Input/Output

- EPIC XML Pointer Files (Input/Output)
- Argo and GTSPP Inventory files (Input)
- Database Documents (Input/Output)
- Dapper Servers (Input)

Database Maintenance

- View contents of any created database
- Drop (delete) selected databases

Calculations

Built-in Calculations

- Minimum value of observed variable
- Maximum value of observed variable
- Depth of minimum value of observed variable
- Depth of maximum value of observed variable
- Maximum depth of non-missing value of observed variable
- Minimum depth of non-missing value of observed variable
- Average of non-missing value of observed variable
- Number of non-missing value of observed variable

Profile Scaler and Integral Calculations (built-in calculations are applied to these calculated parameters and stored in database. For example, theta is calculated as data are ingested but only the results of built-in calculations—for example mean theta—are stored in the created database)

- theta
- sigma-0
- sigma-1
- sigma-2
- sigma-3
- sigma-4
- sigma-n
- specific volume anomaly
- spiciness (Jackett and McDougall)
- spiciness (Flament)
- sound velocity
- O_2 % saturation
- AOU
- NO
- PO
- Brunt-Vaisala frequency
- squared Brunt-Vaisala frequency
- squared Brunt-Vaisala frequency/g
- thermal expansion (alpha)

- alpha * dT/dz
- saline contraction (beta)
- beta * dS/dz
- neutral density (gamma—Jackett and McDougall)
- acoustic travel time
- net heat content
- geopotential anomaly
- potential energy anomaly

"Station" Calculations (since station calculations result in one value per station, the result of the actual calculation is stored in the database)

- Mixed-layer depth based upon parameter of choice and choice of difference method, surface method, or slope method; User-settable tolerance
- Integration of any parameter between any two values of another parameter (options include: use shallowest/deepest observation if surface outcrops or hits bottom if needed, compute weighted mean, interpolation of missing values and interpolation direction: top down, bottom up, or up/down)
- Interpolation of any parameter onto surface value of any other parameter; special interpolation to surface or bottom of ocean

Data Visualization

Data Window

- Current observation: optional display of units and color by quality codes
- Current station metadata
- Section cross section: offset by sequence, distance, latitude, longitude; color symbol by bottle quality code; color symbol by presence/absence of parameter, custom symbol size, zoom y-axis range; reverse station plot order

Features of All JOA Plots

- Optional color legend
- Resizable
- Lockable (plot browsing, colorbar changes, filtering disabled)
- Zoomable (zoom to new window or replace current plot)
- Browsable
- All plots linked by common data "cursor"
- Full control of axes ranges , increments, minor tics
- User-settable background, foreground, grid colors
- Savable as PNG, Postscript, and PDF (Mac OS X only)
- Full control of plot typefaces, type style, type color
- Printable
- User-settable plot name (default names are generated automatically)

Property-Property Plots

- Up to seven x axes: each axis can have unique symbol, symbol size, and connect observation line color
- Change order of x axes
- Symbols colored by third parameter (can be turned off)
- Optional x,y grid
- Isopycnal contour overlay for T-S plots

- User-settable reference pressure for isopycnals
- Plot current station only (optional)

Profile Plots
- Lines colored by third parameter
- Offset by sequence, distance, or time
- Optional y-axis grid
- User-settable plot symbols/symbol sizes
- User-settable amplitude, trace offset, line width

Contour Plots
- Offset by sequence, distance, latitude, longitude, or time
- Autoscale creation of parameter colorbars
- Solid fill contours, contour lines, combination of solid filled and contour lines, skip every nth contour
- Observation or interpolation level overlay
- Residual contour plots with user-defined mean cast
- Contour plots of parameter referenced to user-defined level
- Control of missing value handling (horizontal or vertical interpolation)
- Interpolate up or down cast
- User settable horizontal and vertical browsing panels
- Overlay of contoured field of any other parameter from any open dataset

Contour Gradient Plots
- Geostrophic velocity referenced to standard levels
- Flux
- Uncorrected transport

Map Plots
- User-defined projection: Mercator, Miller, orthographic, stereographic (with polar aspects), Lambert equal area, and Mollweide
- User-defined map region or select from predefined regions
- Optional coastline display
- User-settable station symbol and size
- User-settable station label (offset, angle, and precision)
- Connect station symbols with line option
- User-defined latitude/longitude graticule
- Optional graticule labels
- Retain cartographic aspect or allow map to fill window
- Isolines from built in bathymetry
- Filled bathymetry from ETOPO and Sandwell-Smith
- Color station symbols by parameter interpolated onto isosurface or at surface or bottom
- Color station symbol by value of calculated station parameter
- Color station symbol by station metadata
- Overlay contour plot by parameter interpolated onto isosurface or at surface or bottom
- Overlay contour plot by value of calculated station parameter
- User-settable background, coastline, graticule, graticule labels color
- Map settings savable to files and as default map
- Map tools for creating custom sections

Station Value Plots

- Line plots of mixed layer depth, integration between surfaces, interpolation to surface, station extrema, station statistics, neutral surface.
- Color coded to show where integration outcrops or runs into bottom.

Calculations

Built-in Calculations
- theta
- sigma-0
- sigma-1
- sigma-2
- sigma-3
- sigma-4
- sigma-n
- specific volume anomaly
- spiciness (Jackett and McDougall)
- spiciness (Flament)
- sound velocity
- O_2 % saturation
- AOU
- NO
- PO
- Brunt-Vaisala frequency
- squared Brunt-Vaisala frequency
- squared Brunt-Vaisala frequency/g
- thermal expansion (alpha)
- alpha * dT/dz
- saline contraction (beta)
- beta * dS/dz
- neutral density (gamma—Jackett and McDougall)
- acoustic travel time
- net heat content
- geopotential anomaly
- potential energy anomaly

Calculations are automatically applied when data are added to the current data window.

Custom Parameters
- Arithmetic operations (+, -, /, x) on parameters and user-defined constants
- Derivative and integrals operators
- Multistep arithmetic calculations

Parameter Transformations
- natural log
- log x
- square
- square root
- sine
- cosine
- reciprocal
- e^x
- 10^x
- convert pressure to depth

- convert depth to pressure

Parameter Recoding
- Recode parameter(s) that fall inside/outside a range to missing or other value
- Recode in place or make new parameter
- Recode associated quality code (optional)

Station Calculations
- Mixed-layer depth based upon parameter of choice and choice of difference method, surface method, or slope method; User-settable tolerance
- Integration of any parameter between any two values of another parameter (options include: use shallowest/deepest observation if surface outcrops or hits bottom if needed, compute weighted mean, interpolation of missing values and interpolation direction: top down, bottom up, or up/down)
- Interpolation of any parameter onto surface value of any other parameter; special interpolation to surface or bottom of ocean
- Neutral surface (user-settable neutral density) with error terms
- Station statistics: minimum, maximum, depth of minimum, depth of maximum, average, number of non-missing values of selected parameters
- Station Extrema: minimum or maximum of selected parameters over whole profile or range of another parameter; report values of other parameters at extremum

Section Calculations:
- Difference section between any too sections with control of vertical and horizontal grids and destination grid
- Create mean cast from station subset

•Supporting Resources Management

Color Palettes
- Create/Edit 16 x 16 color palettes
- Blend colors from start to end color
- Save/restore palettes

Colorbars
- Create/edit colorbars up to 128 value/color pairs
- Color bars can be created for four common types of station metadata: date/time, month, latitude, and longitude. Metadata color bars are dynamic, they do not correspond to a specific date range but are scaled according to the metadata ranges of the currently open data files. Adding data to an existing data window will cause all metadata ranges to be recomputed and any plots colored by metadata to be redrawn. Metadata colorbars can be applied to property-property plots, profile plots and map plots.
- Color assignment from color palettes, blending from start to finish color through intermediate color, rainbow, inverse rainbow
- Assign values with autoshape tools (linear, asymptotes, and logistic) or assign user values to color
- Colors individually editable
- Histogram overlay to aid in colorbar creation (measured parameters only)
- Assign new colors without replacing existing values

Interpolation Surfaces

- Create/edit interpolation surfaces of up to 128 levels
- Assign values with autoshape tools (linear, asymptotes, and logistic) or assign user values to levels

Miscellaneous Features

- Windows menu for easy navigation between windows
- Export station calculations to file
- Export interpolated values for contour plot to file

Additional Resources

The official JOA web site at PMEL is <http://www.epic.noaa.gov/epic/software/JavaJOA.htm>.

The official JOA web site at Scripps is <http://odf.ucsd.edu/joa/jsindex.html>.

CHAPTER 2. INSTALLING JAVA OCEANATLAS & SUPPORT FILES

Java OceanAtlas for Mac OS X Installation Notes

1. Download the zipped JOA disk image file.

2. Double click the zip file to unarchive it to a disk image file (.dmg).

3. Double click the .dmg file to mount the disk on the desk top.

4. Open the disk image and drag the Java OceanAtlas 5.0 folder to your Applications folder (or a folder of your choosing).

Mac OS X System Requirements

- OS X version 10.3.5 or higher. Or Panther, Java 1.4.2 Update 2 from Software Update is required. If you do not see an item called Java 1.4.2 Update 2 in Software Update, you already have the latest . Mac OS X 10.4 doesn't require any updates.
- 256 MB RAM (512 MB or more recommended).
- G3 class processor (G4/G5/Intel recommended).
- At least 100 MB hard-disk space.

About Stuffit Expander™

Many of JOA accessory files are stored in .zip files. In addition to the Mac OS X 10.3/10.4 Finder, Stuffit Expander™ is a freeware tool from Aladdin Systems, Inc. that will unzip these files.

Java OceanAtlas for Windows Installation Notes

1. If your machine already has JRE 1.4.2 (or JDK) already installed then download "JOA w/o JVM", otherwise; download an installer that includes JVM called "JOA & JVM".

2. Double click the installation file.

3. Follow the on-screen instructions to complete the installation.

Windows System Requirements

- Microsoft Windows 9X/2000/NT/ME/XP (Vista currently untested).
- 256 MB RAM (512 MB or more recommended).
- At least 100 MB hard-disk space.

About WinZip®

Many JOA support files are stored in .zip files. WinZip® is a shareware tool from WinZip Computing, Inc. that will unzip these files for you to use.

Java OceanAtlas for UNIX Installation Notes

1. Download a JOA UNIX installer for either Linux, Solaris or generic UNIX. These installers do not contain an embedded Java Virtual Machine.

2. Run "chmod +x installer_name" on the command line (make it executable).

3. Start the installer by invoking it's name from a command line (from within an X session).

4. Follow on-screen instructions.

5. If you have more than one JVM installed on your machine, the installer will prompt you to select a one.

Unix System Requirements

- UNIX (Solaris/Linux/BSD etc.).
- 256 MB RAM (512 MB or more recommended).
- At least 100 MB hard-disk space.

About Etopo-5 Add-ins

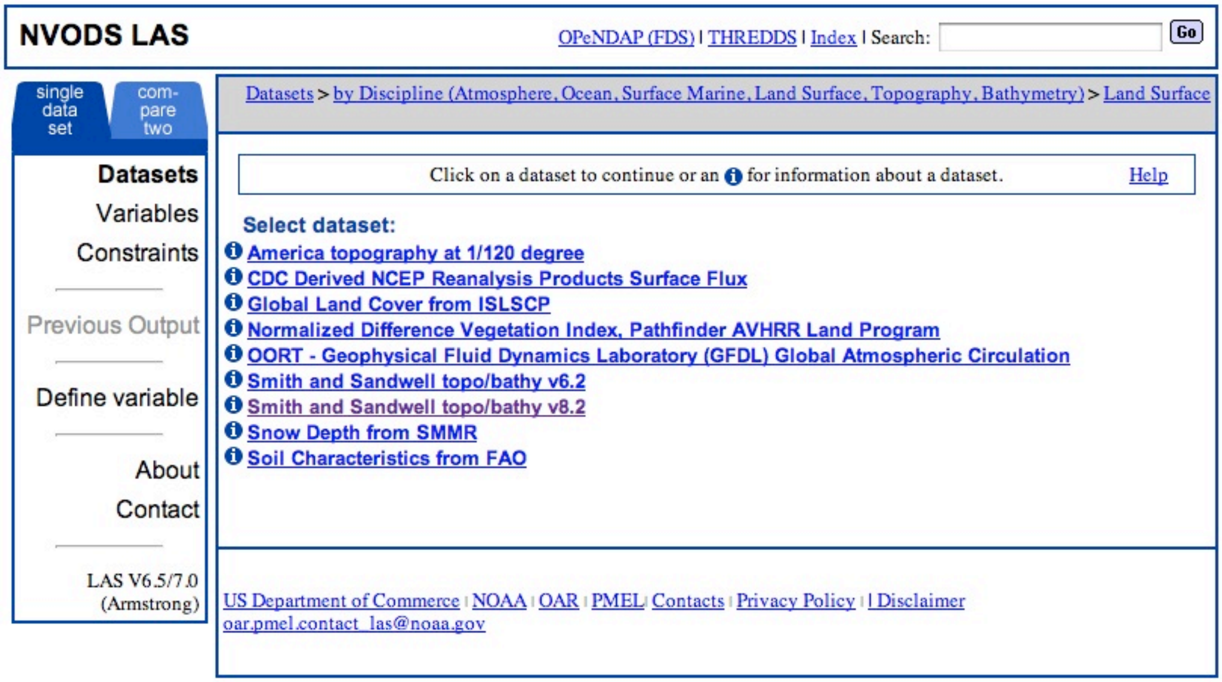
These are binary files extracted from ETOPO-5 with bathymetry and elevations at 5-minute spatial resolution. If you would like to add the higher-resolution ETOPO-5 files download any or all of these files and unzip them into the JOA Support folder/subdirectory on your computer. (Java OceanAtlas 3.1 contains ETOPO-60 and ETOPO-20 files.)

Using the Smith and Sandwell 2-Minute Bathymetry Data with JOA

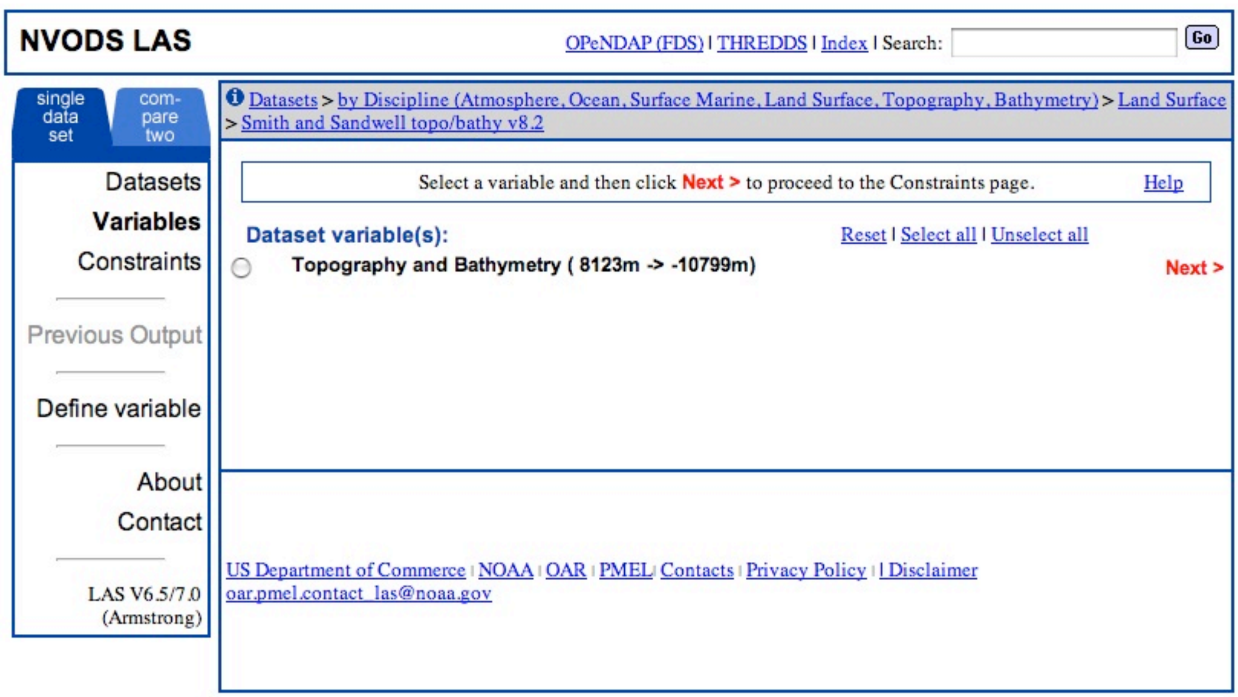
The Smith and Sandwell global bathymetry data set can be used to make a high-resolution bathymetry resource that is compatible with Java OceanAtlas. Due to the size of this data set (around 270MB), we advise you to only use small subsets.

To use the Smith and Sandwell 2 minute bathymetry data with JOA follow these steps:

1. Go to the National Virtual Ocean Data web site at:
   < http://ferret.pmel.noaa.gov/NVODS/servlets/dataset?catitem=16102>:



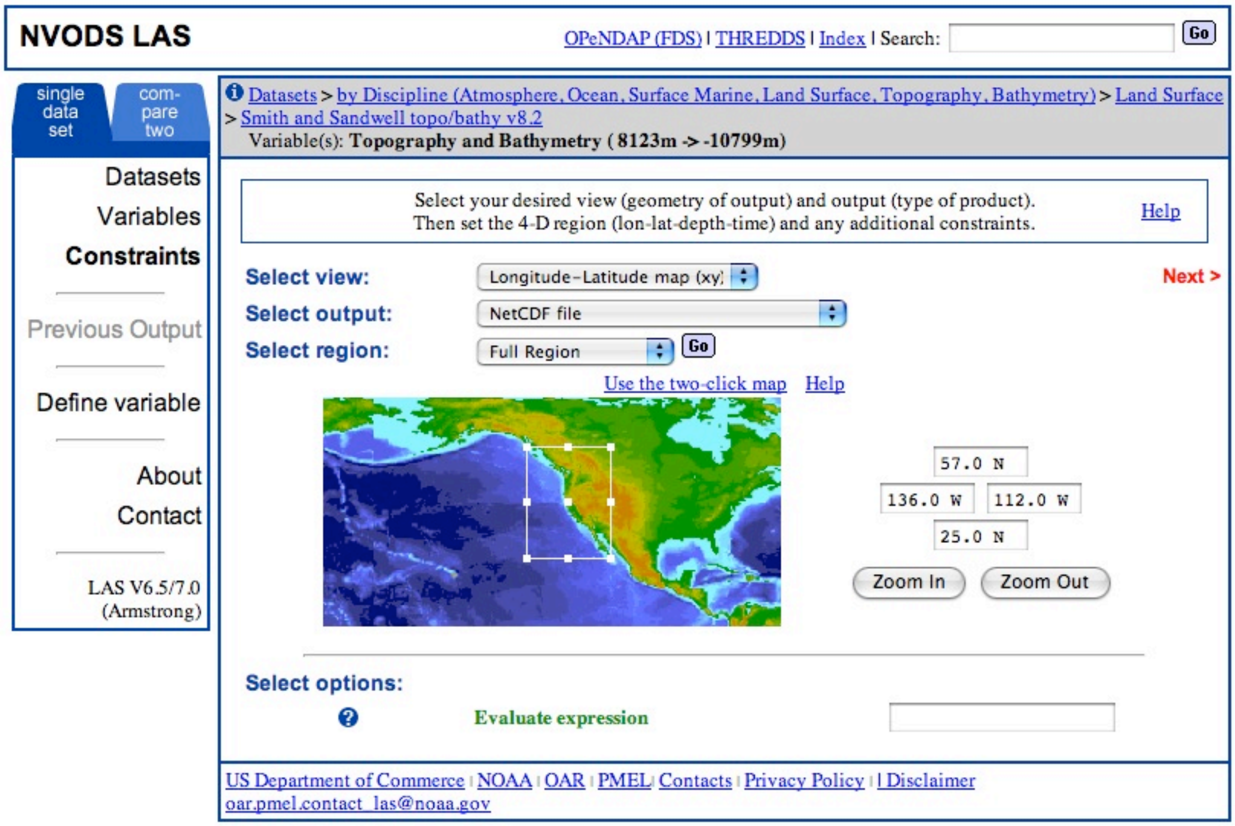
2. In the Data Set list, click on the link for "Smith and Sandwell topo/bathy v8.2 ":



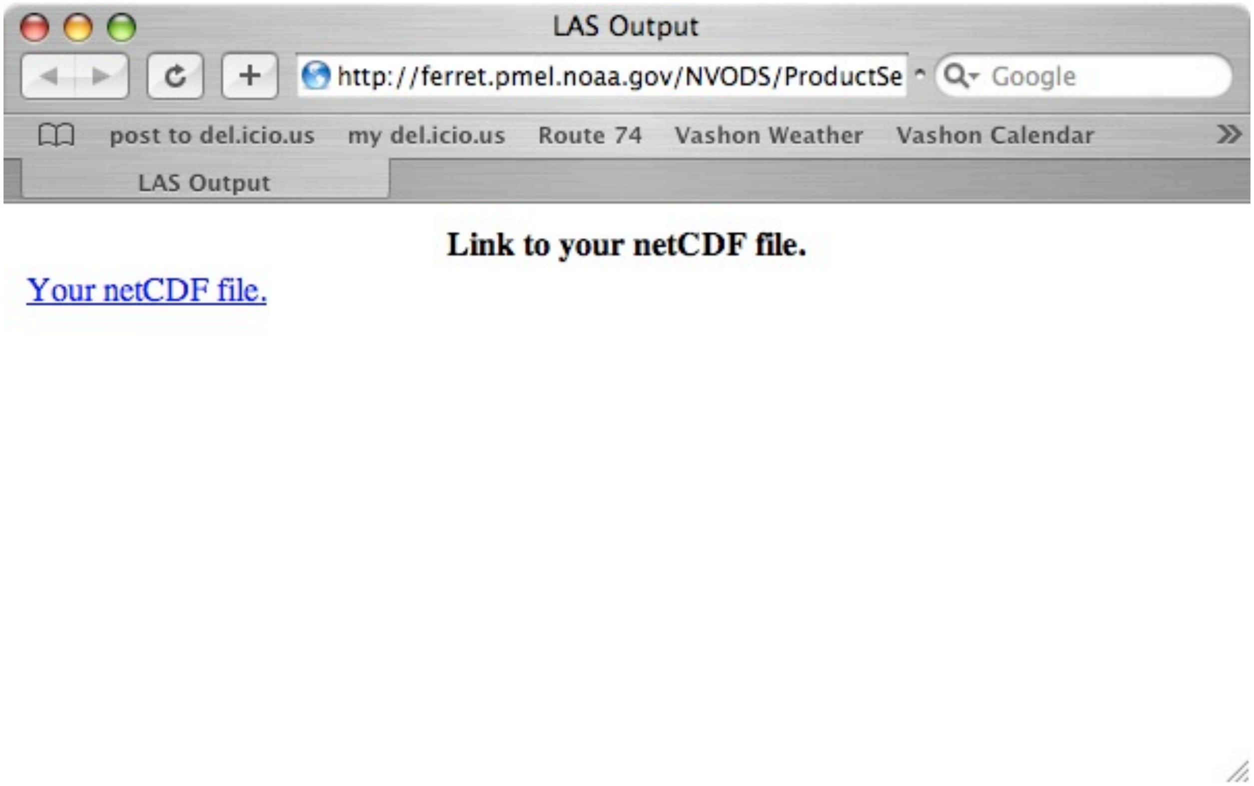
3. Check the *Topography and Batyhmetry* variable and click the *next>* link on the right

4. Click the Link for Use Interactive Map" below the map image. Using the map, zoom buttons, the region pop-up, or the latitude/longitude text fields, define a subarea of the map. Bathymetry/topography data will be returned for this area.

5. Set these options:
   Select view: Longitude-Latitude map (x/y)
   Select output: netCDF



6. Click the red *Next* > link. You will be presented a page with a link to your data:

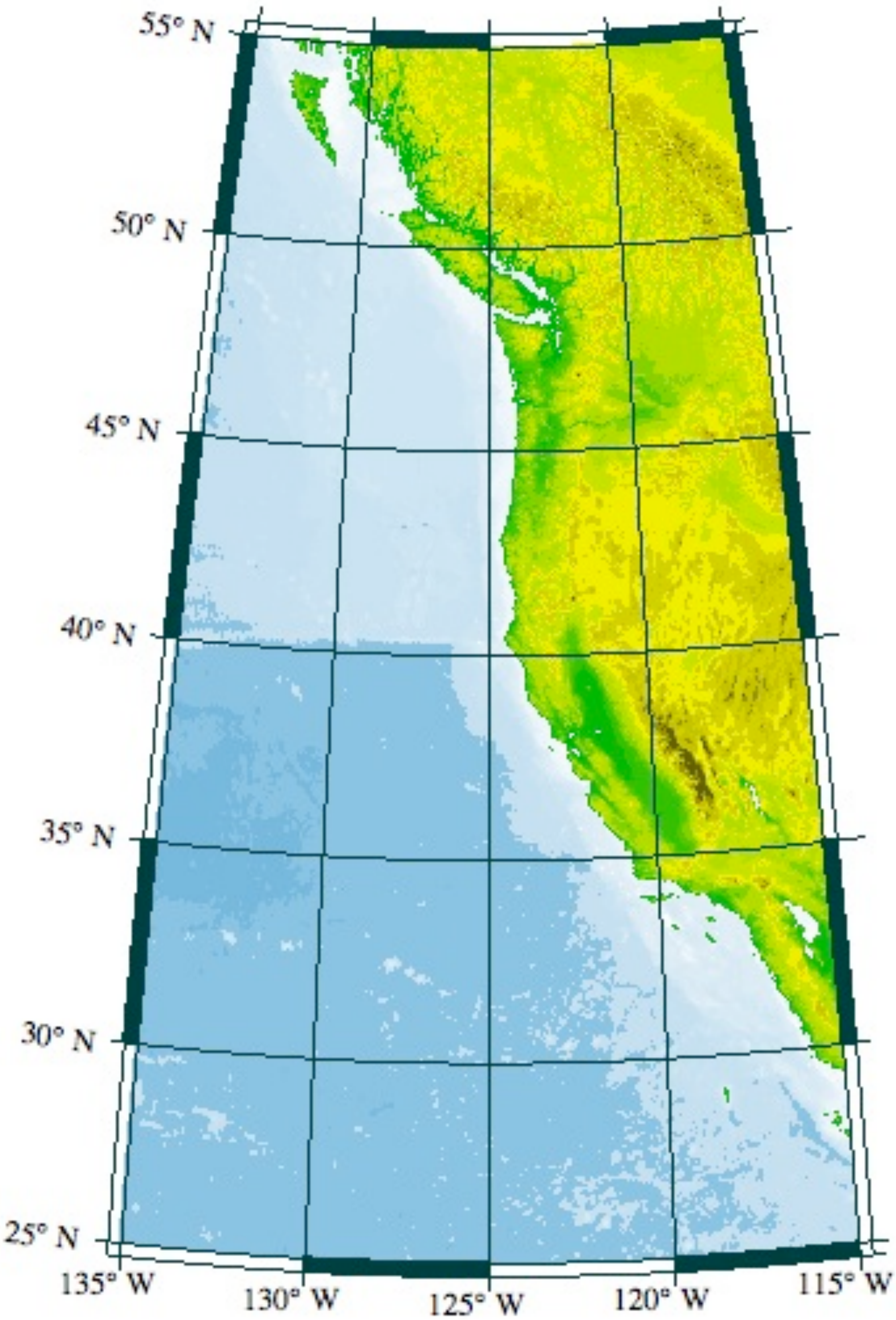


7. Click the link *Your netCDF file*.

8. Your file will be downloaded to the default download directory on your PC or Mac. Change the name to be compatible with JOA—the file I got back from LAS was called 5563CDE002C5CA2695995E278DC0A9AC_ferret_listing.nc. The name you choose must contain the text ”ETOPO2” and end with the .nc extension. Example: ETOPO2_Kodiak.nc, etopo2_Bering_Sea.nc.

8. Downloaded files should be installed in the JOA_Support folder found in your JOA installation folder.

9. You will see the files you downloaded in the *Bathymetry* panel of the map dialog. You can use any of the "ROSE (Relief Of Surface of Earth)" colorbars to color the bathymetry:


55° N
50° N
45° N
40° N
35° N
30° N
25° N
135° W
130° W
125° W
120° W
115° W

CHAPTER 3. INPUT AND OUTPUT

Opening and Adding Data Files

The most basic action necessary to use the Java OceanAtlas application is to open one or more data files. To open a data file:

Start the Java OceanAtlas application. In some operating systems this is done by double-clicking on the JOA icon, or by invoking an *Open* command.

A small window with a control for displaying the NdEdit data browser is displayed in this initial JOA window, The menu bar will be either in this window or across the top of the screen depending on the computer operating system you are using.

From the *File* menu select *Open* (cmd/ctl-O) and navigate the file-selection dialog in whatever means are provided by your computer's operating system, selecting an appropriate file type to open (see below), and ending with the actions necessary to complete the *Open* command.

The file will open, and the JOA Data Window will appear.

In Mac OS X it is possible to double click or ‘drag and drop' .joa and .jos data file icons onto the Java OceanAtlas application icon to open the data file(s). If this technique does not work for a particular data file it is strongly recommended that you try opening the file via the open command instead.

Invoking the *Open* command for more than one data file, or invoking it multiple times on individual data files, will result in one data window being opened for each file opened. In this way multiple data sets can be compared in individual Java OceanAtlas windows.

Invoking the *Add* command (cmd/ctl-A) - which is available only after at least one JOA Data Window is opened, will add the newly selected file to the opened data in the front-most Data Window. In this way it is easy to create concatenated JOA data files. (The resulting files can be saved as a unit.)

Java OceanAtlas is able to read several different file types. Due partly to limitations of the Java language and also to the multi-OS nature of Java OceanAtlas, specific file extensions are required for each file type. See the User Guide section on data files for more information.

Java OceanAtlas Data Files

Java OceanAtlas reads and writes a variety of file types. Due to the multi-platform nature of the application and data files, Java OceanAtlas requires use of a specific file extension for each file type. Below is a list of supported file types:

file extension (suffix)	description
.joa	Java OceanAtlas binary (read, write) Good for storing custom sections as well as files that have been imported from other formats. Java OceanAtlas binary files will usually open significantly faster than importing files in plain text. Java OceanAtlas binaries store parameter units as well as bottle and observation quality codes.
.jos	Spreadsheet (TSV) (read, write) A tab-separated (TSV) format that is compatible with most spreadsheet applications.
_hy1.csv	WOCE Bottle section in WHP-Exchange format (read) This is a comma separated value file (CSV) produced by the WOCE Hydrographic Program. Files in this format include multiple stations and usually correspond to a complete section or cruise. This format is a replacement for the older and more complex .sum and .sea files. Java OceanAtlas preserves all units and quality codes (bottle and observation). Java OceanAtlas provides 'post processing' capabilities to filter observations by quality information as well as perform units conversion. (In a WHP-Exchange data file, Java OceanAtlas recognizes the missing value designators used by that program, hence no alteration of missing values is needed.)
_ct1.csv	WOCE CTD profile in WHP-Exchange format (read) This is a comma separated value file (CSV) produced by the WOCE Hydrographic Program. Files in this format include a single CTD cast. Java OceanAtlas preserves all units and quality codes. Java OceanAtlas provides significant 'post processing' to filter observations by quality information as well as units conversion. CTD files can be decimated to user-selected intervals or Java OceanAtlas standard levels.
_ct1.zip	WOCE CTD section in WHP-Exchange format (read) A zip file that contains multiple individual CTD profiles. (These can take significant computer resources to open!)
.nc	netCDF Profile (read, write) Individual CTD or bottle profiles that adhere to PMEL's EPIC standard. Future versions of Java OceanAtlas will be able to extract sections out of gridded fields.
.ptr	EPIC pointer file (read, write) A text file that contains multiple references to individual netCDF profiles.

.sd2 NODC SD2 (read)
Text files that adhere to NODC's SD2 exchange format. Information on the SD2 exchange format is available from the Java OceanAtlas web site <http://odf.ucsd.edu/joa> and on the Atlas of Ocean Sections CD-ROMs.

.nqdb NQuery Database Document
XML file that describes a connection to an NQuery database. File contains the name of the database, comment, and URL of MySQL database server.

_argoinv.txt Argo inventory
Text file that adheres to NODC Argo inventory format. Each file contains multiple lines that correspond to an individual Argo profile. Each profile is identified by spatial/temporal metadata as well as the URL of the actual netCDF data file.

_gtsppinv.txt GTSPP Inventory
Text file that adheres to NODC GTSPP inventory format. Each file contains multiple lines that correspond to an individual GTSPP profile. Each profile is identified by spatial/temporal metadata as well as the URL of the actual netCDF data file.

Note that Java OceanAtlas writes (exports) data in four different formats (JOA Binary, JOA Spreadsheet, WOCE Exchange, and netCDF Section). You can use Java OceanAtlas to read data in one of the other formats and translate into one of the three. Calculated parameters such as potential temperature and density can be added by Java OceanAtlas between reading and exporting the data, or data subsets or new sections can be made.

You may be interested in having Java OceanAtlas read your own data. Spreadsheet format data are the easiest for most users to generate if they are not already familiar with writing data in one of the other supported formats. Fortunately, Java OceanAtlas spreadsheet format is not too demanding. Java OceanAtlas 'understands' many data headers, and reads data in column (spreadsheet) format. For example, an Excel file saved as comma- or tab- delimited text, set up to meet minimal Java OceanAtlas requirements, will read straight into the application.

<u>Java OceanAtlas requires of each spreadsheet data set at least a station number, a latitude, a longitude, and data at one level (one pressure).</u> Columns must be labeled. Java OceanAtlas will try to recognize heading names from its internal lexicon. [A section name is often wise but is not strictly required.] The words shown below are recognized.

SECTION names are short alphanumeric.
STATION numbers are typically 1-4 digit numbers but can be alphanumeric.
LATITUDEs are expressed in decimal degrees, positive in northern hemisphere and negative in the southern hemisphere.

LONGITUDEs are expressed in decimal degrees, positive in eastern hemisphere and negative in the western hemisphere.

Oceanographers use a left-handed coordinate system, meaning that while the x-axis and y-axis point positive in the usual sense, the z-axis is positive downward. Java OceanAtlas data are indexed by PRESsure, which is the parameter used most often for vertical orientation in physical oceanography. Because the pressure at 10 meters depth in the ocean is quite close to one bar, or one atmosphere, one decibar is nearly the same vertical extent as one meter. Decibars are the pressure unit used throughout oceanography and in Java OceanAtlas.

To make a simple data spreadsheet set, look up the latitude and longitude of your city, convert to decimal degrees, and substitute the name and position for 'YOURCITY', '35.00', and '-120.00' in the example:

STATION<tab>LATITUDE<tab>LONGITUDE<tab>PRES<cr>
YOURCITY<tab>35.00<tab>-120.00<tab>1.0<cr>

The use of '<tab>' in the above denotes a tab character, and '<cr>' denotes a carriage return.

This two-line file, if saved as plain text and with a file name given a '.jos' suffix, can be opened by Java OceanAtlas. If you then do a map plot, you will find a station symbol at the position you provided.

VERY IMPORTANT NOTE: The Java OceanAtlas missing value designator in a .jos spreadsheet data file is '-99'. For example, if you have a column of oxygen data in your spreadsheet data file to import into Java OceanAtlas, but one or more values is missing, use a -99 for each missing data value.

It is only a small extension from this minimum example to see the makings of a small Java OceanAtlas spreadsheet data file holding oceanographic profile data, shown below as a table:

STATION	LATITUDE	LONGITUDE	PRES	TEMP	SALT	O2
312	64.002	-27.833	16	8.0754	35.0699	7.25
312	64.002	-27.833	49	7.6206	35.0744	6.89
312	64.002	-27.833	99	7.4723	35.0872	6.79
312	64.002	-27.833	199	6.8717	35.0855	6.43
312	64.002	-27.833	254	6.6958	35.0683	6.42
312	64.002	-27.833	354	6.3275	35.031	6.34
312	64.002	-27.833	454	5.9039	34.9862	6.44
312	64.002	-27.833	554	5.7533	34.987	6.38
312	64.002	-27.833	844	5.0649	34.9598	6.13
314	64.110	-29.000	51	7.7994	35.0822	6.91
314	64.110	-29.000	403	6.6015	35.0678	6.46
314	64.110	-29.000	707	5.5014	34.976	6.31

314	64.110	-29.000	960	4.633	34.9474	6.1
314	64.110	-29.000	1109	4.1972	34.9435	6.19
314	64.110	-29.000	1417	3.7364	34.9235	6.38
314	64.110	-29.000	1598	3.3992	34.9006	6.51
315	64.132	-29.533	17	8.6936	35.0759	7.12
315	64.132	-29.533	41	7.8779	35.0814	6.44
315	64.132	-29.533	111	7.0148	35.0911	6.51
315	64.132	-29.533	313	6.5638	35.0706	6.34
315	64.132	-29.533	609	5.5097	34.9747	6.14
315	64.132	-29.533	1011	4.4229	34.9566	6.27
315	64.132	-29.533	1215	3.9963	34.9411	6.3
315	64.132	-29.533	1517	3.4827	34.908	6.49
315	64.132	-29.533	1619	3.4704	34.9063	6.5

Other header columns that Java OceanAtlas 'understands' (but does not require) include CAST (an integer number), BOTTOM (depth to bottom in meters), and DATE (dd-mm-yyyy hh:mm.m or dd/mm/yyyy hhmm.m format, where mm.m indicates seconds are encoded as decimal minutes). Note: CAST, in the JOA sense, is one integer number to describe each unique profile gathered at the same station.

Java OceanAtlas will read any column of profile parameter data. For many common parameters, its internal lexicon helps Java OceanAtlas to interpret which of its standard parameters, for example those for which there are predefined color bars, is meant. But Java OceanAtlas will read any parameter, and even assign a name if the column heading (name) is more than four characters, though it is best for new parameters (e.g., concentration of a phytoplankton pigment) for users to supply a four character name in the column heading. Java OceanAtlas has many tools for making new and customized color bars, and for autoscaling contour plots, and so nearly any data will work. A small caveat is that it is best to avoid representing data values as very small (close to zero) numbers, e.g., 0.0003456. Such data often are best shown in columns pre-multiplied by a power of ten to bring their apparent numerical value - to Java OceanAtlas - away from zero. (This has to do with how microcomputers handle storage and arithmetic for very small numbers.)

Saving and Exporting Data Files

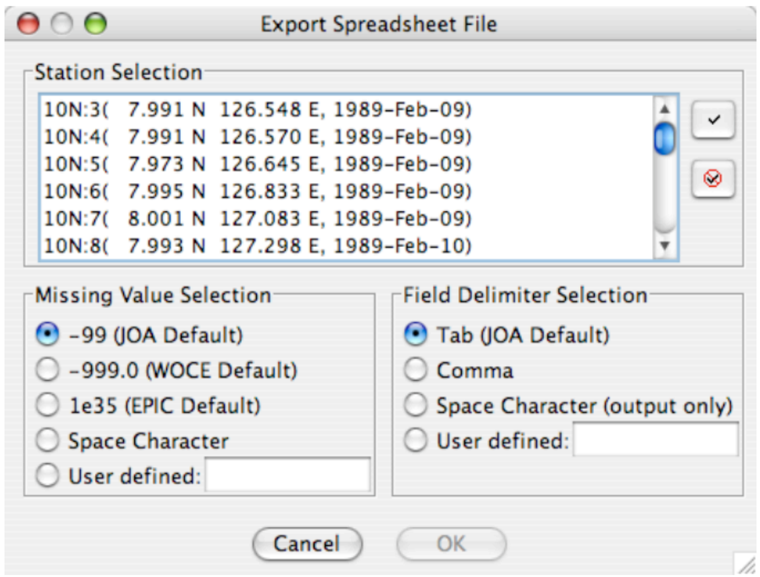
Java OceanAtlas will save files in JOA native binary, readable by Java OceanAtlas on any platform, or export four different file types. The selections are available from the Java OceanAtlas File menu.

Save As Java OceanAtlas native binary

To save the current data file (or multiple 'Added' files) as a JOA native binary file, invoke the *Save As* command from the *File* menu. Java OceanAtlas will suggest a file name with a .joa suffix. It is fine to change the file name, but please remember to retain the .joa suffix. All original and calculated parameters, changes to parameter names and file properties, and so forth, are saved in the resulting file. Hence if you wish a file without added calculated parameters, invoke *Save As* before carrying out those calculations.

Export Spreadsheet ASCII delimited spreadsheet

The *Export Spreadsheet* command, available from the *File* menu, is a powerful tool for exporting Java OceanAtlas data to other applications, including to an editing application to edit one or more data values. Invoking the command brings up the Export Spreadsheet File dialog:



The complete list of all stations in the current data file (including *Added* data files) is presented in this dialog. By clicking, shift-clicking (for contiguous station selection), or control/cmd-clicking (for discontiguous station selection), any of the stations in the list may be selected for export. The *check* button will select all stations, and the *uncheck* button will deselect all stations. The dialog allows you to select a different missing value indicator and field delimiter than the JOA defaults (-99 and tab).

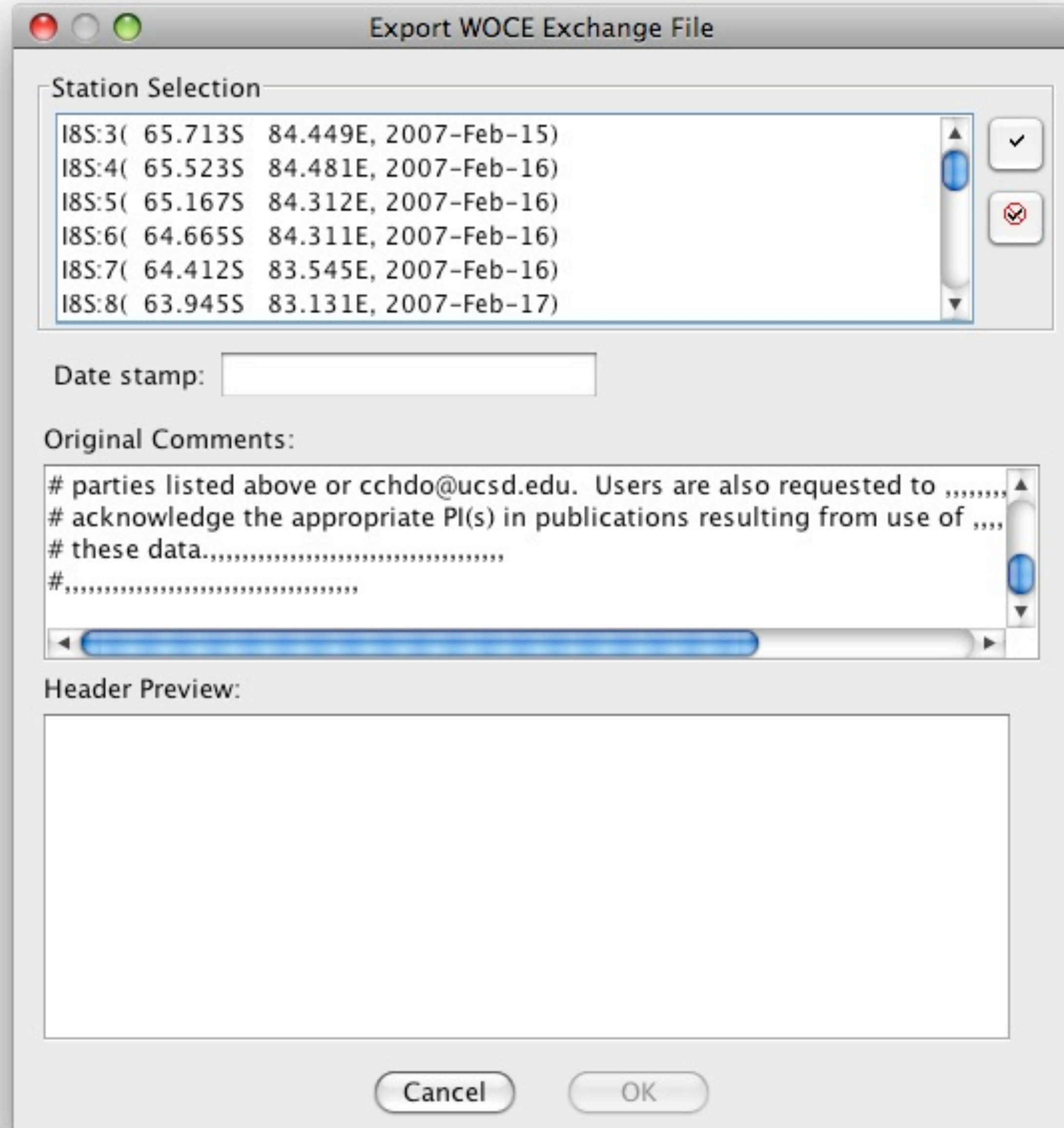
Clicking *OK* after stations have been selected will bring up a standard *Save* dialog, with a suggested name, including the '.jos' file extension. It is fine to change the file name, but please remember to retain the .jos suffix if you plan to read the spreadsheet back into Java OceanAtlas

Exported spreadsheets contain all essential information in the data file(s) selected in delimited ASCII columns. This type of data is easily read into many spreadsheet, plotting, and analysis programs.

Save as WOCE Exchange Format

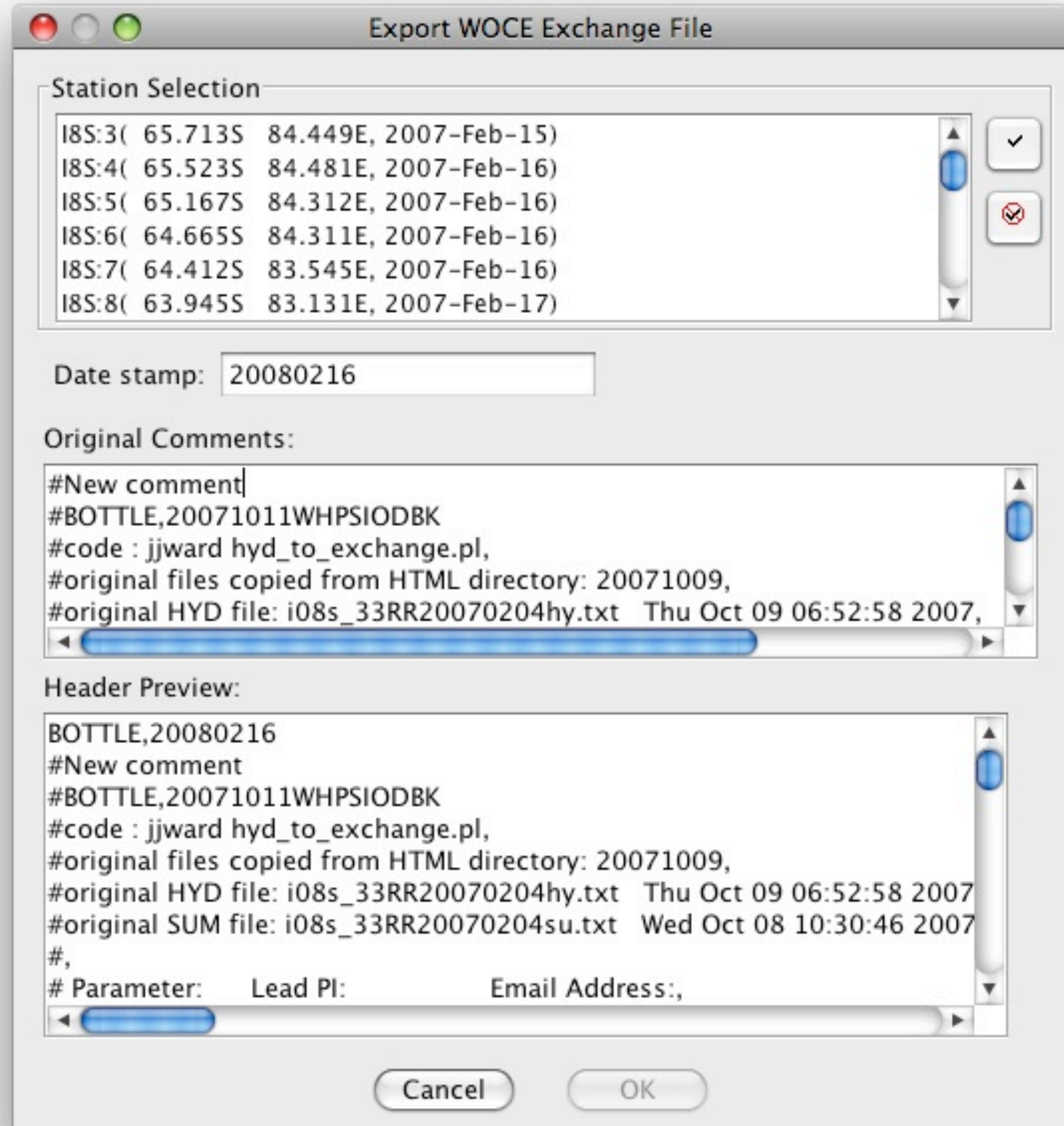
Note: JOA currently only exports the bottle variant of the WOCE Exchange format.

To export the current file set, select *Export WOCE Exchange File* from *File* menu:



The export dialog allows selection of stations, a date stamp comment, and any additional comments you want in the file. JOA also preserves all comments when importing files in WOCE Exchange format. These are presented in the *Original Comments* field and can be freely edited.

Edited comments and the optional date stamp are previewed in the *Header Preview*:



Select any number of stations and click *OK* to produce a file in WOCE Exchange format.

Export netCDF section and Export EPIC Pointer File

The *Export netCDF Section* command in the File menu exports all the stations as individual netCDF files as well as producing an EPIC pointer file. *Export EPIC Pointer File* just produces a pointer file (i.e., no netCDF files).

The pointer file is what ties all the individual netCDF files together. One would produce a pointer file and not export the individual netCDF files if one originally opened a dataset via a pointer file and wanted to produce a pointer file that is a subset of the original data - say after using a station filter or the section tool. Different pointer files can be created that open different views of the same data set without worrying about the actual netCDF files.

Imagine this scenario:

Open 1000 netCDF bottle profiles via a pointer file or the hard way, using the Add Data command 1000 times.

Use the section tool and create custom sections *section1* and *section2*.

Save an EPIC pointer file for *section1* and *section2*.

Quit JOA.

To resume working with *section1* or *section2*, open the pointer files from with JOA.

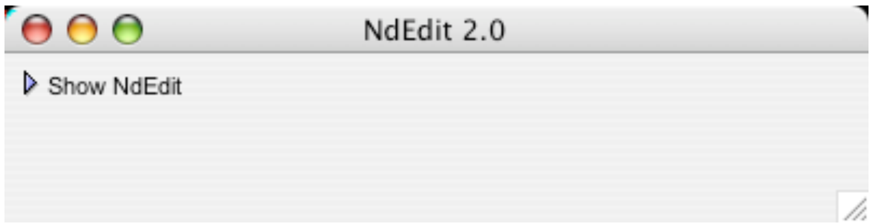
Now you may be asking why not just save the new sections as '.joa' files? You can certainly do this but you would then have the data for *section1* and *section2* in two places - in the original netCDF files and the new .joa files. The pointer file allows you to have many different views of one dataset without storing the data in more than one place.

Java OceanAtlas has available an optional 'front end' to allow browsing large collections of profiles and creating custom JOA sections. This front end is known as NdEdit (for **N** **D**imensional **Edit**or) and is a Java version of Don Denbo's 4DEdit tool. NdEdit was developed mostly for access to the EPIC data archives at NOAA's Pacific Marine Environmental Laboratory. However, NdEdit can be used with data archives built from one's own profile data files. See below for instructions on using NdEdit with your own data.

NdEdit is an interactive Java tool for selecting or subsetting data files from large in-situ data collections. NdEdit makes use of a profile’s metadata (location, depth, time) rather than the actual measured data. The user can graphically view the locations of selected data sets in any 2D view of space and time (e.g., latitude-longitude map view or latitude-time view) and using graphical filtering and selection tools the user can select a desired subset of the data displayed. Actions such as selecting or zooming in one 2D view are duplicated in the other 2D views. NdEdit also features coastline and bathymetry display in the longitude-latitude view, user-settable horizontal and vertical axes, and zooming. Data sets in the filter region can be deleted or retained. Selected profiles (using either a rectangular, polygon, or section line tool) can be opened directly in Java OceanAtlas.

Use of NdEdit with Java OceanAtlas is completely optional - you can choose to ignore this feature of JOA if you desire. NdEdit is available from the main JOA window displayed when Java OceanAtlas is launched. To open the NdEdit browser, click the button entitled *Show NdEdit*. This figure illustrates the main JOA window and the Show NdEdit button:

Initial JOA Window in Windows



Initial JOA Window in Mac OS X

To learn more about NdEdit, including a tutorial in its use, see:

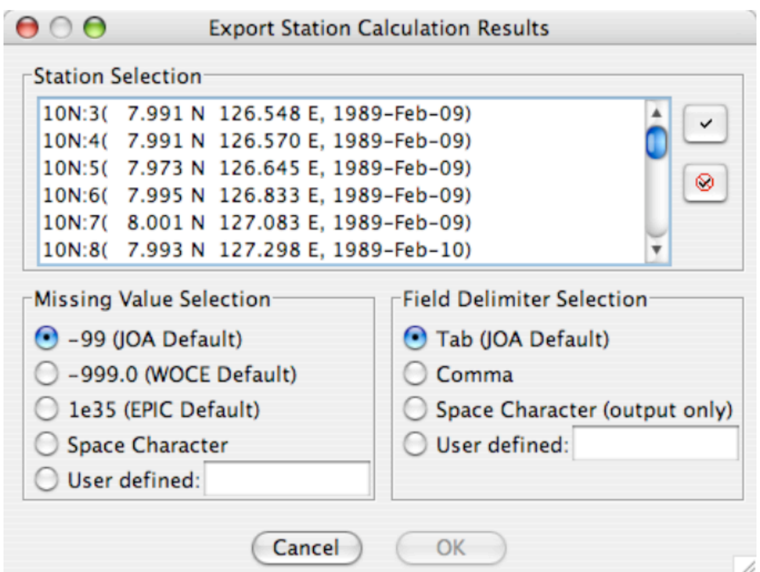
<http://www.pmel.noaa.gov/epic/software/JavaNdedit.htm>.

Export Station Calculations

The results of Java OceanAtlas station calculations (see *Station Calculations* under the *Calculations* menu) are not saved in .joa binary files, .jos spreadsheet files, EPIC .ptr files, or netCDF because these are ”one value per station parameter per station results”. The results of station calculations could, however, be used in other applications. Using the *Export Station Calculations* command under the *File* menu, an ASCII file with these values can be exported with columns for cruise, station, cast, and for each calculation. Here is an example of what the first few lines of a station calculation file looks like exported with JOA’s default settings:

```
Section      Stn. Num    Cast Num.   Lat     Lon    Date   Min of TEMP its-90
P10    1     1      -4.0153     144.8112    1993-Oct-12 17.129610061645508
P10    2     1      -3.9778     144.8333    1993-Oct-12 6.006441116333008
P10    3     1      -3.8888     144.8917    1993-Oct-12 3.7217929363250732
P10    4     1      -3.6167     145.0125    1993-Oct-13 2.546510934829712
```

Choosing *Export Station Calculations* brings up the Export Station Calculation Results dialog:

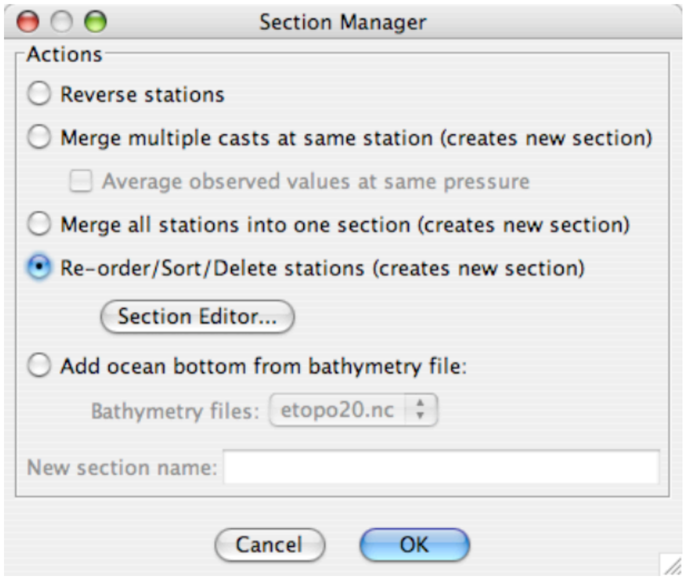


The complete list of all stations in the current data file (including *Added* data files) is presented in this dialog. By clicking, shift-clicking (for contiguous station selection), or control-clicking (for discontiguous station selection), any of the station calculations from the stations in the list may be selected for export. The *check* button will select all stations, and the *uncheck* button will deselect all stations. There is a choice of missing value indicator and field delimiter.

Clicking *OK* after stations have been selected will bring up a standard *Save* dialog, with a suggested name. It is fine to change the file name.

Section Manager

Selecting *Section Manager* from the *Edit* menu brings up the Java OceanAtlas section manager dialog:



The Section Manager can be used to reverse the order of the stations in a data file (particularly useful when the user wishes to have the data plotted in reverse order than found in the file). Using the *Save As* command after reversing the order of stations will result in the reversed-order data file being saved.

Sometimes a data file contains multiple casts at a single station. Java OceanAtlas reads these as individual stations, but use of the *Merge multiple casts at same station* choice will cause Java OceanAtlas to merge each multi-cast station into a single station. This action results in a new Data Window being created

Sometime a data file has been made up from multiple casts and there are multiple samples from the same depth. The Section Manager contains an option to average values at the same depth when merging casts (*Average observed values at same pressure*).

JOA imports files from online sources (EPIC web and Dapper) as single-profile sections. The "Merge all sections into on section" takes all these individual sections and merges then into one logical section. This new section is displayed in a new Data Window and can be saved or exported to a new file.

Sometimes a data file contains unwanted or duplicate stations, or one or more of the stations is out of order in the data file. Using the *Re-order/Sort/Delete stations selection* will highlight bring up the Java OceanAtlas Section Editor dialog. (See separate User Guide entry.)

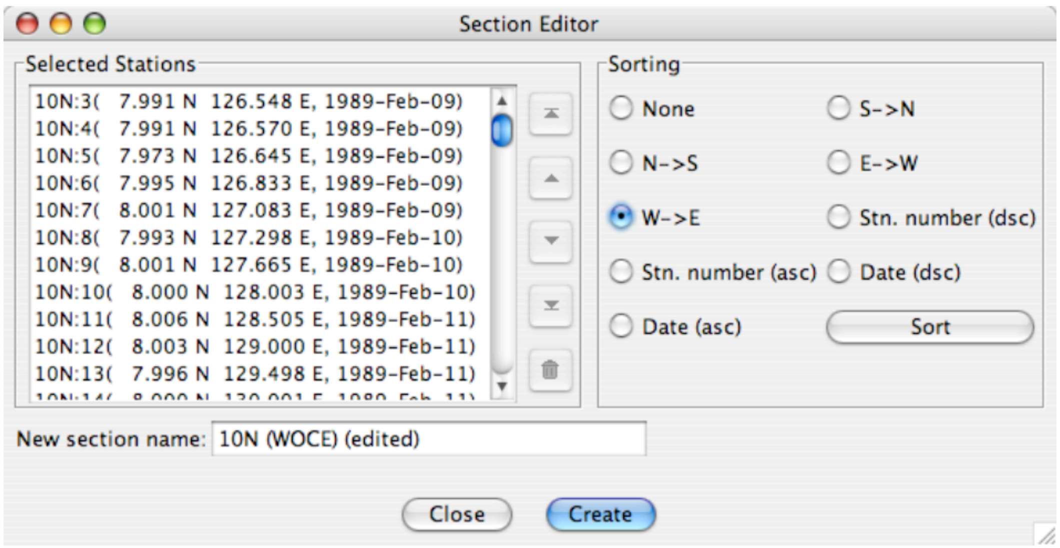
The Section Manager dialog can be exited without action by selecting *Cancel* or the selected choice can be enabled by selecting *OK*.

Data Editor

@TODO

Section Editor

The Java OceanAtlas Section Editor is displayed when a selection of stations is made from a map plot, or when the Section Manager is used to re-order or re-sort stations:



The scrollable list of Selected Stations on the left of the dialog includes all stations currently selected with one of the map plot selection tools, or the entire current data set when invoked from the Section Manager. Any individual station may be selected by clicking on it, any group of stations contiguous in the list can be selected by shift-clicking, and any discontiguous group of stations can be selected by command-clicking (right mouse button). The station(s) selected can then be moved to the top of the list, up in the list, down in the list, to the bottom of the list, or be discarded using the five buttons immediately to the right of the scrolling list.

The entire list of stations (all remaining after discards, if any) can optionally be sorted by position (north to south, west to east, east to west, or south to north), or by ascending or descending station number or date.

Actions of the Section Editor result in a new Data Window being created. Java OceanAtlas will suggest a *new section name* for the new Data Window, but any name of the user's choice can be entered. This does not, however, save the newly created section. The *Save As* command from the file menu for the new section data file must be used to save the newly created section.

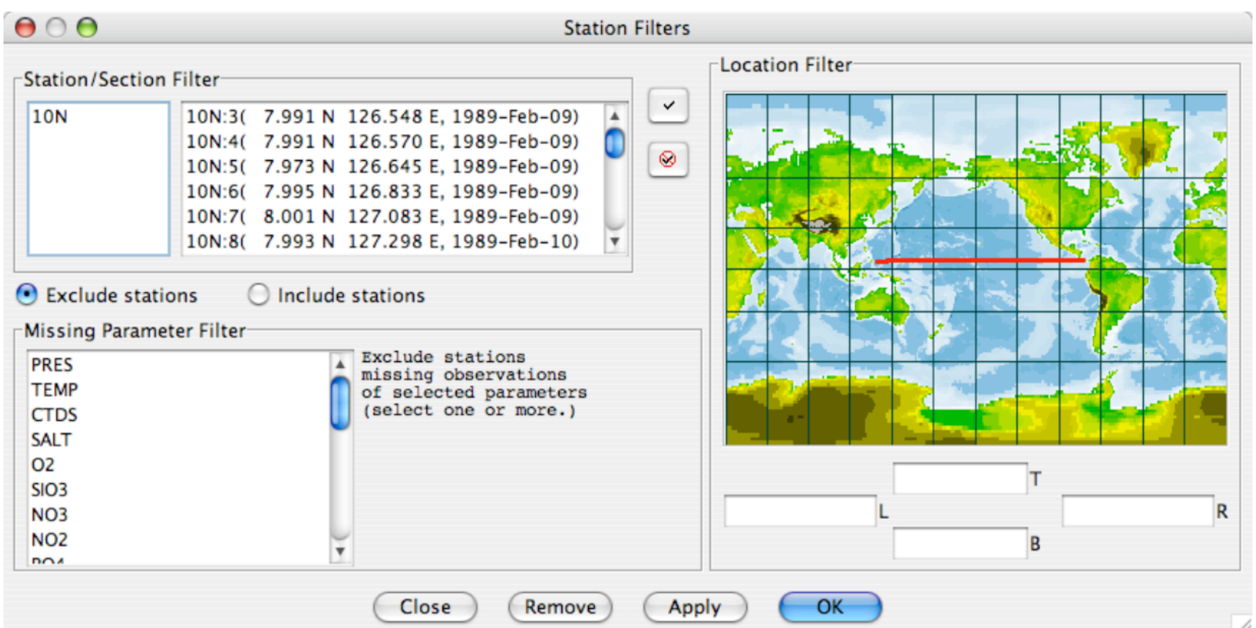
The Section Manager can also add an interpolated value of the bottom of the ocean for files or data sets that don't contain such information. The user has the option of using any of the installed etopo files for extracting this information.

Station Filter

The choices under the Filters menu allow the Java OceanAtlas user to create subsets of stations or observations by means of user-defined filters. This opens up many possibilities, for example creating/choosing a section from a spatially heterogeneous database (such as the sections from the Levitus data set) or determining the geographical and/or property-space location or boundaries of a feature.

Station filters create an inclusive or exclusive subset of stations based upon position criteria and/or upon station numbers and/or by the presence of data at stations of one or more specified parameters. Use of Station Filters updates the Data Window to show only those stations that match the filter criteria. Station subsets can be saved to new data files with the 'Save' command under the File menu.

Selecting *Station Filters* from the *Filters* menu brings up the Station Filter dialog:

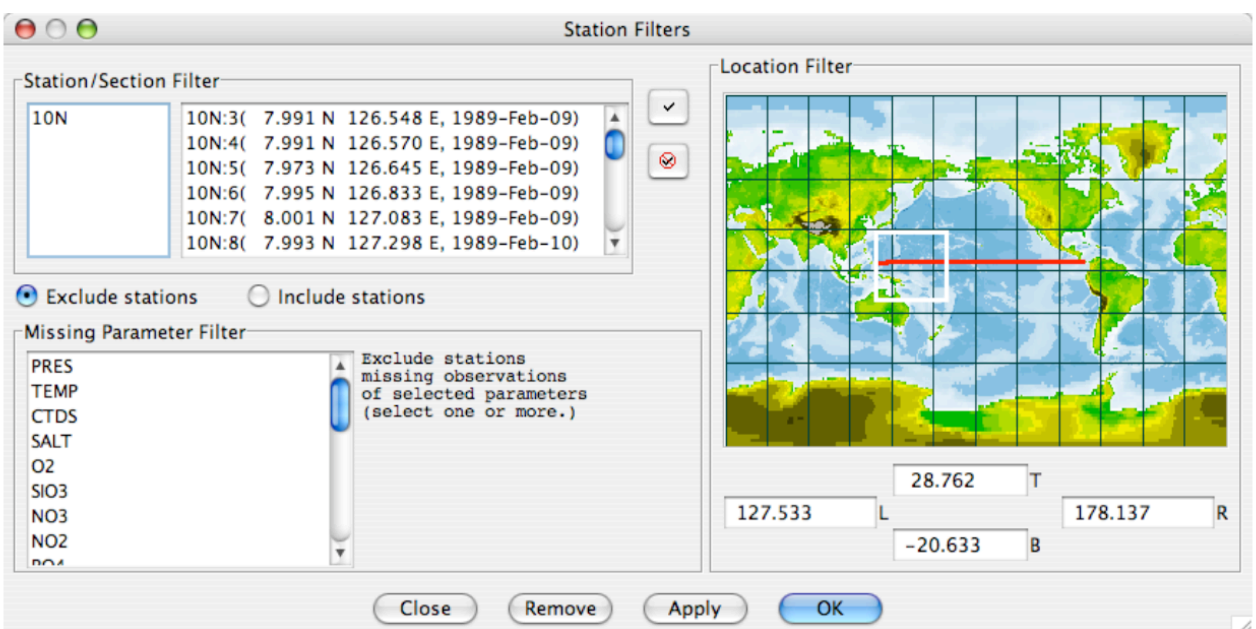


Station/Section Filters - the most commonly used station filters - create a subset of data based upon user-supplied selections from scrollable lists of sections and station numbers. Selections are made by clicking the mouse on a section or station number. Additional contiguous sections/stations are selected by 'shift-clicking' (holding down the 'shift' key while clicking the mouse) and non-contiguous sections/stations are selected by 'control-clicking' (holding down the 'control' key while clicking with a one-button mouse or by using the right mouse button on a multi-button mouse). Check and Uncheck buttons for all or none of the stations are provided for convenience. Station/Section Filters can be used to exclude or include data by selecting either the *Exclude stations* or *Include stations* radio button.

Missing Parameter Filters retain only those stations that have at least one non-missing value of the selected parameters. This will work with any parameter or combination of parameters in the list. *Parameter* selections can be made by clicking, shift-clicking to extend a selection, or control-clicking (right mouse button clicking) for discontiguous selections. This filter is useful to set up contour plots of parameters that are measured at only a portion of the stations within a data set, for example radiocarbon in WOCE data, with the result that the filtered data will exhibit much fewer missing data in the plots.

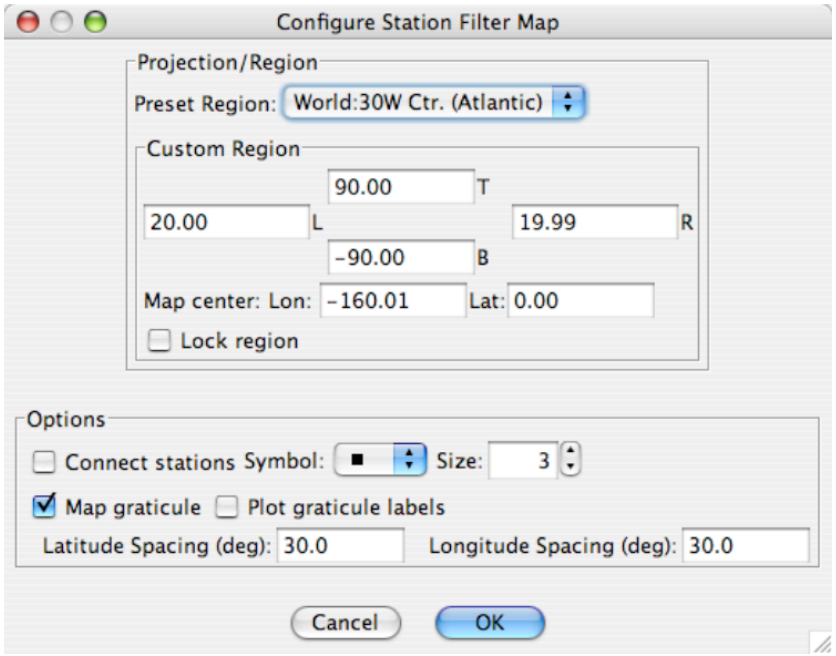
Location Filters create a subset of data based upon user-supplied criteria for latitude and longitude. The simplest way to define a location filter is to drag a selection rectangle on the small map around the data to include (see figure below). Stations falling outside of the selection rectangle will be grayed out after the filter is applied. There are also four text fields for entering more precise position limits, however, only those germane to the intended filter need to be filled in. For example, entering the number '30' in the upper right of the four text fields (and clicking *OK*) is all that is needed to remove stations north of 30°N from a data file. Location Filters are always Inclusive.

> Note: Remember that in Java OceanAtlas west longitudes and south latitudes are negative.



The latitude/longitude limits of the selection are displayed in the four text fields.

Double-clicking on the map in the Station Filters dialog (or control-clicking or clicking with the right mouse button) brings up the Configure Station Filter Map dialog which allows customization of the map display:



Tip: Command/Control clicking on the station filter map will zoom the map while Shift-Command/Control clicking will zoom out (make the map region bigger).

Each of the three types of Station Filters can be used individually or together. For example, one could construct a filter that removed stations south of the equator and also all those that did not contain F11 data.

The Station Filter action is temporary (will not affect the stored data) unless the data are specifically saved to a new file using the *Save* command under the *File* menu. If the data are saved the original data can be either overwritten (by keeping the file name unchanged during the save) or retained (by choosing a new/different file).

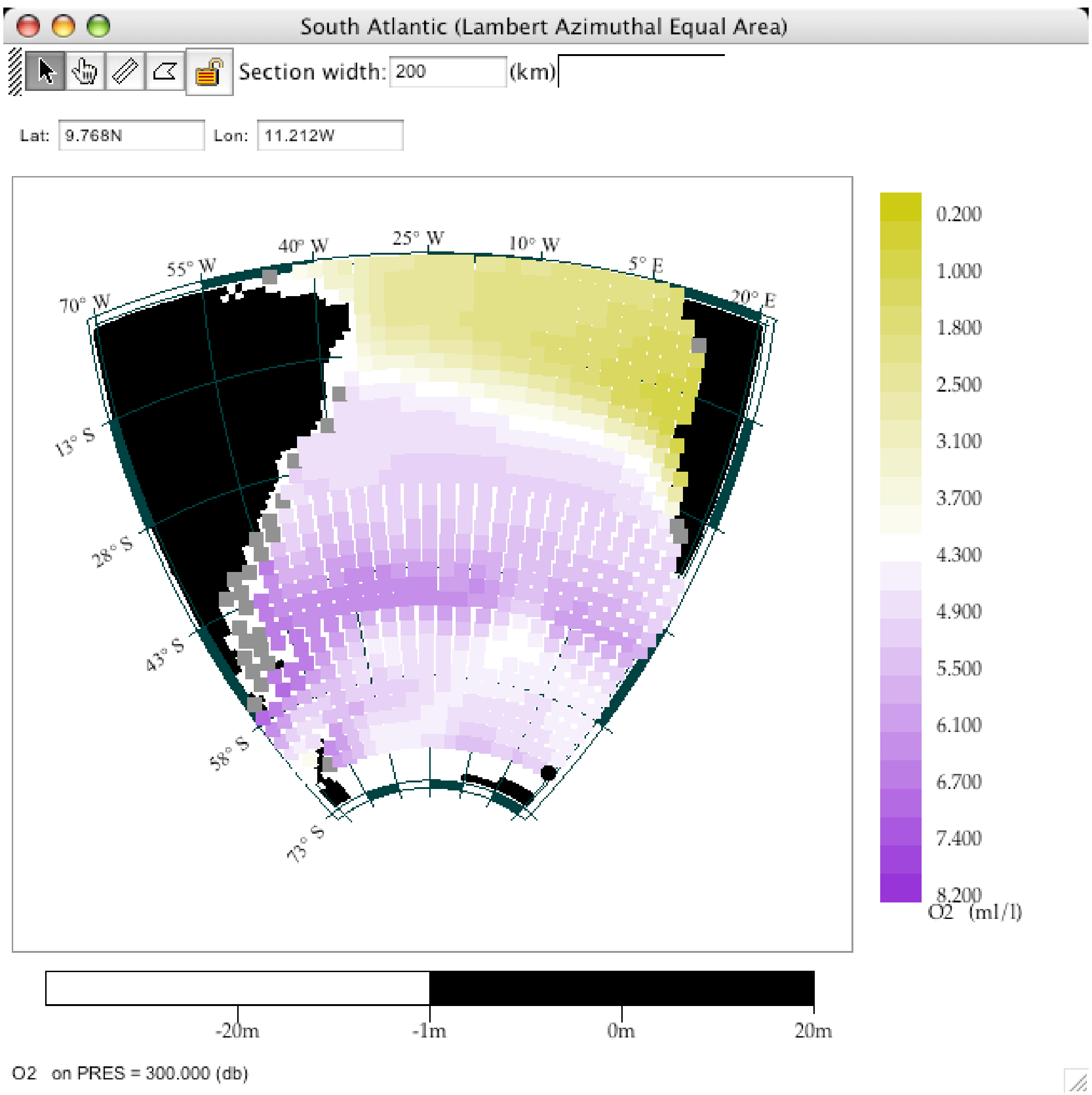
The Station Filter dialog can be closed without applying the filters (if any) by clicking the *Close* button. Clicking on *Remove* will remove the current filters while leaving the dialog open. Clicking on *Apply* applies the current filters to the plots while leaving the dialog open. This allows interactive experimentation with the station filters without needing to recall the dialog. Clicking on *OK* exits the dialog and applies all selected filters to profiles, property-property plots, and contour plots.

There is a certain degree of confusion possible if filters are applied and then additional data sets opened. If you experience difficulty, think through a plan of actions, such as opening files, performing filters, saving files, and then opening and adding files to prepare a data set with the desired contents.

Extracting Data Subsets from Map Plots

Java OceanAtlas map plots contain special features that permit the user to extract station subsets or new sections using a set of graphical tools. These features complement the Station Filters available via the Filters menu.

Note the four icons in the upper left of a Java OceanAtlas map plot (this map plot was made from the Levitus WOA98 Atlantic data set with iso-surface coloring - see *Station Colors* in the *Map* dialog - with O2 selected for *Parameter*, the O2__-global_cbr.xml for *Colors* and PRES-0-6000_srf.xml for *Surface*):



1. The pointer-arrow tool (left-most button) is used to zoom a map to a new range or create a new map window of the selected range. Note: Using this tool to select a sub-area of a Java OceanAtlas map plot will create a subset of the main map, not a filtered subset of data.

2. The pointed-finger tool (2nd button from left) is used to select stations on the map--it does not change the latitude/longitude limits of the map. This tool operates in two modes: (1) Dragging a selection region around a set of stations in a map plot will display that subset in the Section Editor window (see below). By leaving the Section Editor window open and repeating selections, additional sub-areas of stations can be selected. New selections will also be displayed in the Section Editor window. (2) As a single station selection tool it can be clicked on any given station symbol, which will then show up in the Section Editor window (see below). By leaving the Section Editor window open, additional stations can be selected by holding down the shift key when clicking on a station symbol.

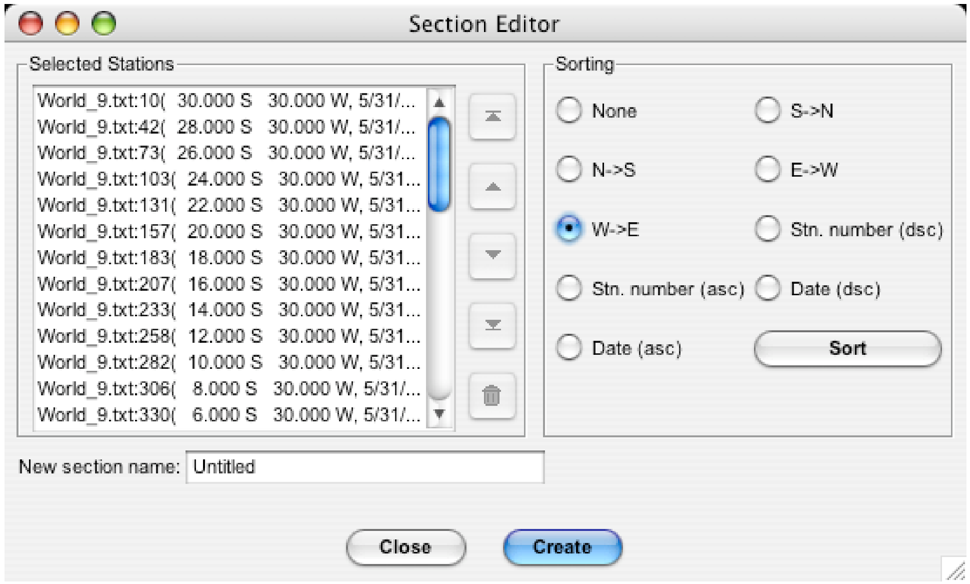
3. The narrow divided-rectangle (3rd button from left) is the section selection tool: First, if needed, type in a new *Section width*. The section width is the swath width of the tool--stations that fall within the tool's boundary will be selected. The default of 200 km is useful mostly for large basin-scale sections. Often a smaller width is desired. Section selections are made using the following steps:

1) Click once at the beginning of the desired section
2) Drag the mouse to enclose the stations to be part of the sections. [Extra stations or duplicates can be discarded later.] Note: the width of the section band on the screen may change according to latitude and the map projection.
3) Click the mouse to end a section segment and begin a new segment
4) Double-click to end section selection.

4. The irregular-area (polygon) selection tool works somewhat like the section tool. Polygon selections are made using the following steps:

1) Click once to start the first side of the polygon
2) Drag the mouse to enclose the stations to be part of the selection. [Extra stations or duplicates can be discarded later.]
3) Click the mouse to end a side and begin a new side
4) Double-click to end polygon selection.

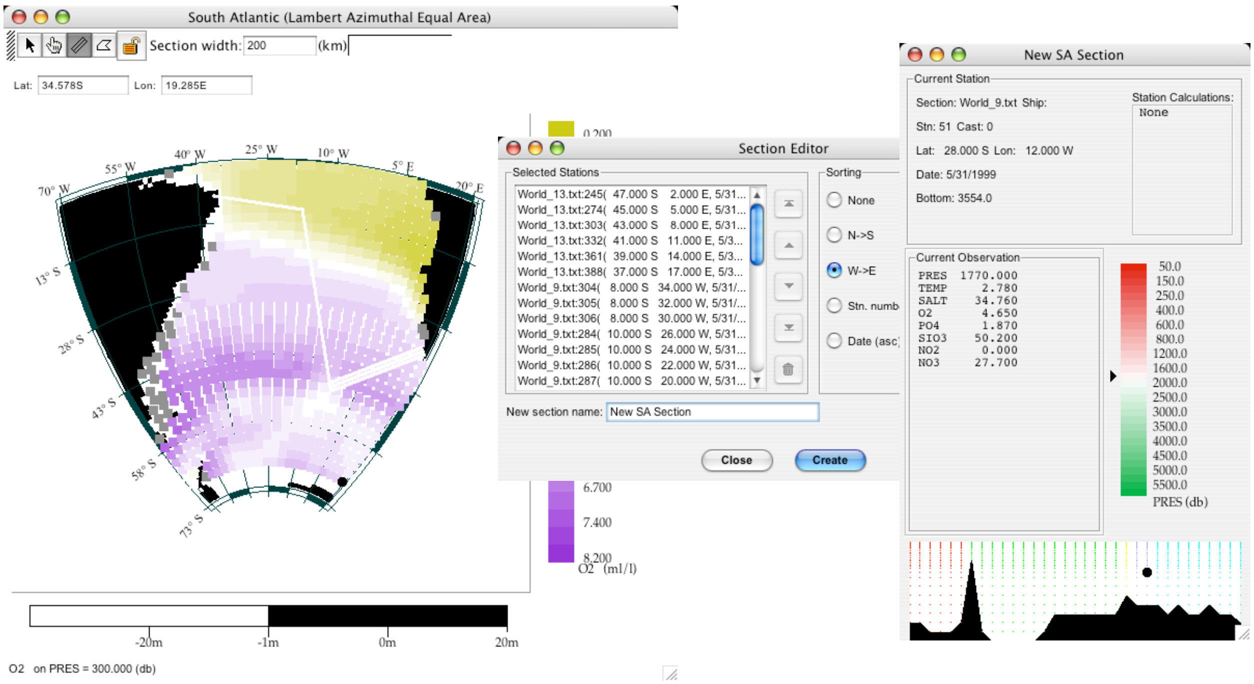
The result of successful applications of choice 2, 3, or 4 above should be the Section Editor window popping up:



In the Section Editor window, Java OceanAtlas presents a list of all the stations it found inside the selection(s) or along the section track. Note: Java OceanAtlas does not pre-sort the selected stations--the list presents the stations as they are found in the data file(s). In addition, the list may include unwanted or duplicate stations. This list can be edited in many ways. The entire list of stations can be sorted by location, station number, or date or individual stations can be moved or deleted in the list. Here, the *W->E Sorting* option was selected and then the *Sort* button was clicked. From the station list in the dialog individual stations or groups of stations (using standard click, shift-click, ctrl/command click techniques) can be selected and, with the five buttons to the immediate right of the station list, moved to the top of the list, moved up one step (per click) in the list, moved down one step (per click) in the list, moved to the bottom of the list, or discarded. When you are ready to create the new section, type in a name for the new section.

Click *Create* and the Data Window for the new section should appear, all ready for plotting. The newly created subset can be saved to a new data file via the *Save As* command under the Java OceanAtlas *File* menu.

Below is an example of creating a new Java OceanAtlas section from a South Atlantic gridded data map plot showing: (left) choice of section tool and result of drag-clicking from South American to Africa along a three segment user-defined path, (center) the Section Editor dialog which came up automatically when the mouse was double-clicked instead of single-clicked at the end of the third segment and with *W->E* sorting chosen and *Sort* button clicked, and (right) the Data Window for the new section which resulted from clicking on *Create* in the Section Editor dialog.



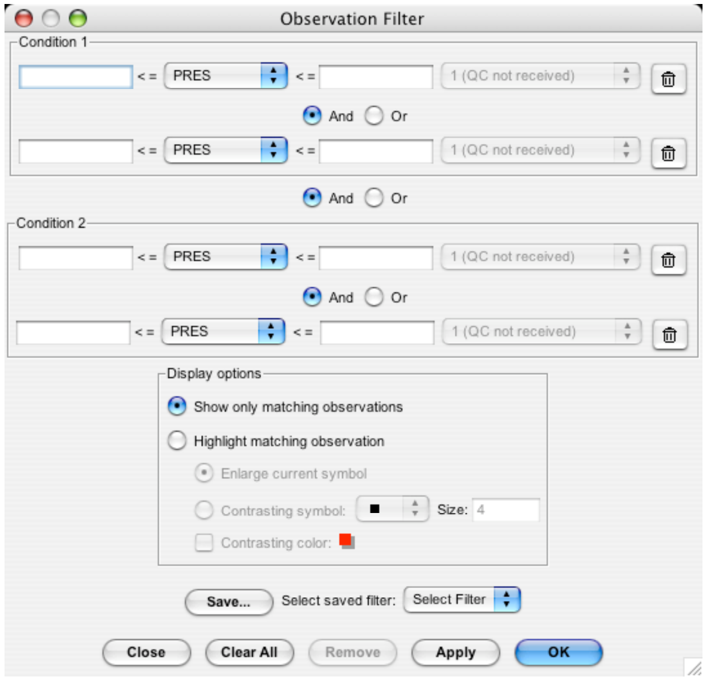
The exact results will depend on the path chosen and the *Section width* setting in effect when the section selection tool was selected on the map plot. Wider section widths will pick up more stations along the path. Remember, Java OceanAtlas is using the center of each station symbol, not the full size of enlarged symbols. The section tool takes a bit of practice to use, the feel of which may be affected by the mouse (or equivalent) settings for your particular computer.

Observation Filter

The choices under the Java OceanAtlas Filters menu allow the user to create subsets of stations or observations by means of user-defined filters. This opens up many possibilities, for example creating/choosing a section from within the open data or determining the geographical and/or property-space location or boundaries of a feature.

Observation filters create a data subset based upon value ranges of one to four selected parameters. Use of Observation Filters updates profile, property-property, and contour plots, either showing only data matching the filter criteria or highlighting them. Observation Filters provide a powerful means to interactively explore a data set. For example, using Observation Filters one can highlight on all profile, property-property, and contour plots all data where 34.7 ≤ SALT ≤ 35.0 AND 40 ≤ SIO3 ≤120, or, alternatively for the same ranges, all data not meeting these nested criteria can be excluded from such plots.

Selecting *Observation Filter* from the *Filters* menu brings up the Observation Filter dialog:



Java OceanAtlas provides up to four observation filters criteria which can be combined in pairs with 'and' or 'or' logic. Each filter provides text fields for entering lower and upper limits for a parameter. Parameters can be selected from a ‘pop-up’ menu/list of all available parameters. If quality codes are contained in the opened data, (as in WOCE Hydrographic Program 'Exchange' _hy1.csv and _ct1.csv data files), you will see an additional pop-up menu on the right of the criteria that allows easy selection of quality code values. Note the quality code values will be displayed in either the WOCE or IGOSS nomenclature. Click on the center popup menu to select which parameter to use in a filter criterion. When you first open the filters dialog, the parameter popups are usually labeled either pressure or ‘PRES’. The menus can be scrolled with the mouse if necessary, and any available parameter can be selected. It is not necessary to supply both lower and upper limits for a parameter.

Note: If the Observation Filters dialog is open when calculated parameters are generated, the new calculated parameters will not be added to the scrollable list until the dialog is closed and re-opened.

Observation Filters affect all pre-existing and new profile, property-property, and contour plots. Two ‘Display options’ are provided:

- *Show Only Matching Observations* excludes from plots all observations that meet the filter criteria.

- *Highlight Matching Observations* leaves all plotted points on the property-property and profile plots, but ‘highlights’ all observations matching the filter criteria. JOA offers choices of enlarging the current plot symbol, providing a contrasting plot symbol of user-chosen size, or a contrasting color of the user's choice. (To choose the color click on the colored square beside that choice.)

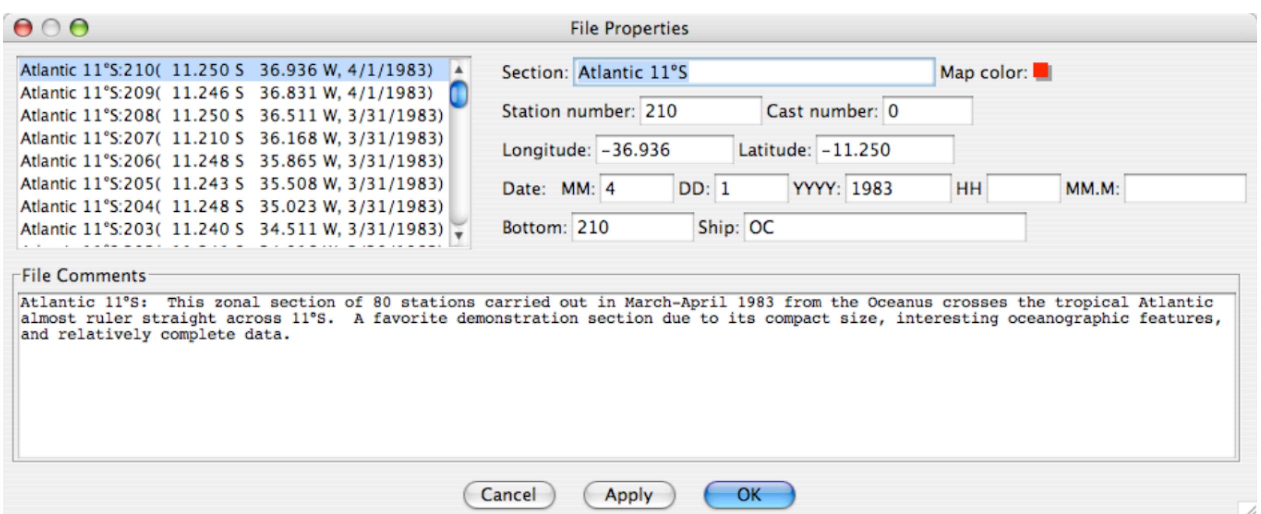
Note: Highlighting matching observations does not work with contour plots—they always use the "show only matching" display option.

In Java OceanAtlas it is possible to save a filter set to a disk file (via the *Save* button) and recall and select a saved filter (from the pop-up menu of saved filter sets).

The Observation Filters dialog can be exited without action of filters (if any) by clicking the *Close* button. Clicking on *Clear All* restores all JOA plots to an unfiltered state. Clicking on *Remove* will remove the current filters while leaving the dialog open. Clicking on *Apply* applies the current filters to the plots while leaving the dialog open. The small trashcan buttons clear individual criteria. This allows interactive experimentation with the observation filters without needing to recall the dialog. Clicking on *OK* exits the dialog and applies all selected filters to profiles, property-property plots, and contour plots.

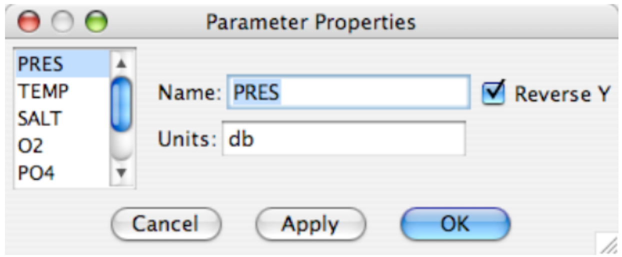
File Properties

The File Properties editor permits editing station header values (metadata), and also permits adding and saving comments regarding the entire data file. The File Properties dialog is where one can also examine previously-saved file comments. The color of the station symbols for each open section can be changed by clicking on the *Map color* swatch, which brings up a color chooser. These changes are not saved unless the edited data file is saved (via the *File* menu) before quitting Java OceanAtlas. Note: File comments are only preserved in JOA's binary files.



Parameter Properties

The *Parameter Properties* dialog, accessed via the Java OceanAtlas *Edit* menu, permits the user to alter the name, units, and plotting direction of all current original and calculated parameters in the opened data file(s):

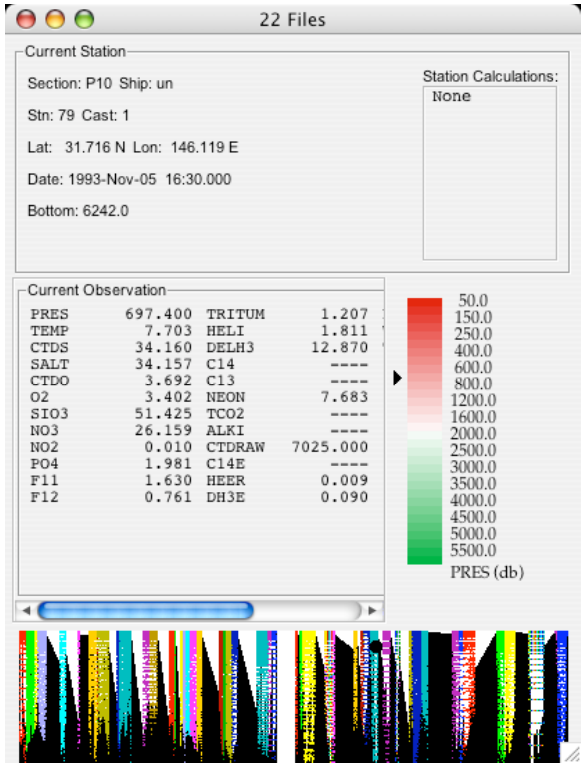


On the left is a scrollable list of all current observed and calculated parameters. Click a parameter in the list to display and optionally edit the parameter's displayed name and units. A check box allows parameters to be plotted in left-handed coordinates, such as pressure and density parameters, to be plotted with a reversed y axis. Any changes made will not be reflected in the actual data file unless explicitly saved.

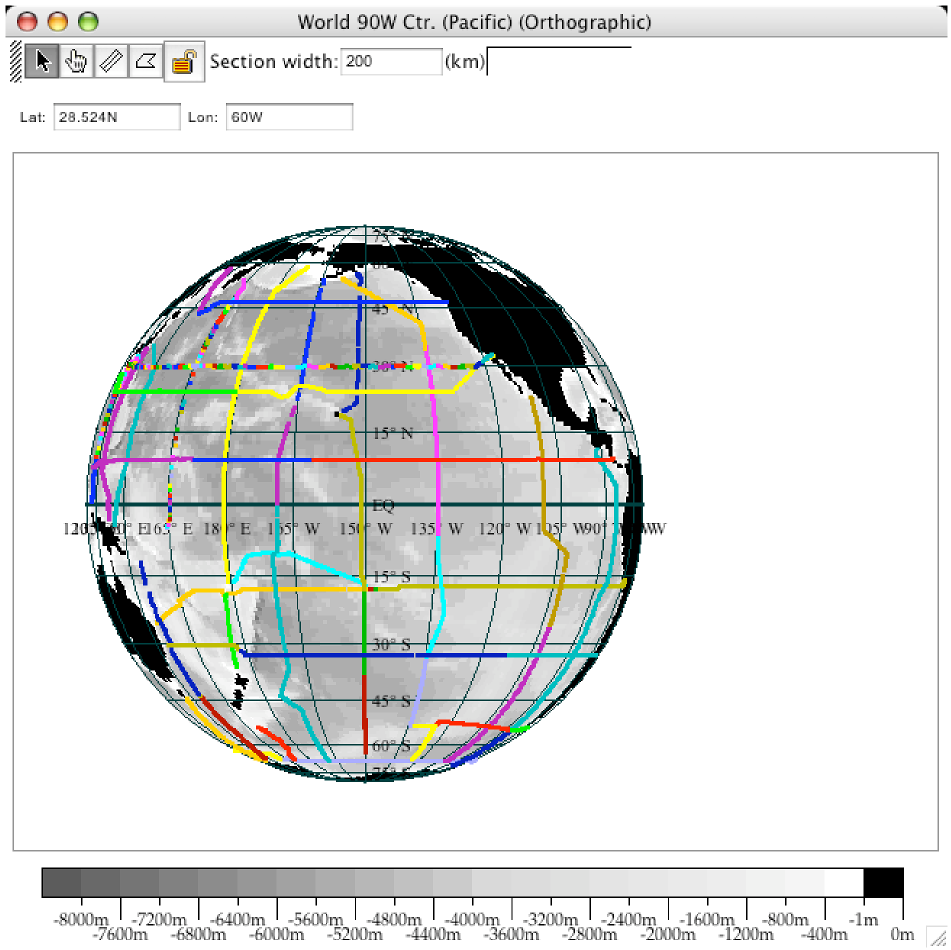
Using NdEdit with Java OceanAtlas

Typically one uses JOA with profiles that are contained in section files created by WOCE or other data management organizations. NdEdit can be useful if one has large numbers of profiles that come from different parts of the World Ocean or from different time periods or would like to synthesize data sets from different sources. The basic unit of data as far as NdEdit is concerned is an individual profile. Thus, to use NdEdit in conjunction with JOA, one may need to take data organized as sections and convert them to single-profile files. The steps to take are as follows:

1) Open up the JOA-compatible data files of interest (you can open as many as can fit in the memory available on the computer) into JOA. Make sure to use the *Add Data* command from the Data Window File menu to concatenate all the sections/profiles of interest into one JOA data window. For example, one could import all the WOCE Pacific cruises into one JOA data window. The following illustration shows the JOA data window with 22 historical sections from the Pacific open at once:

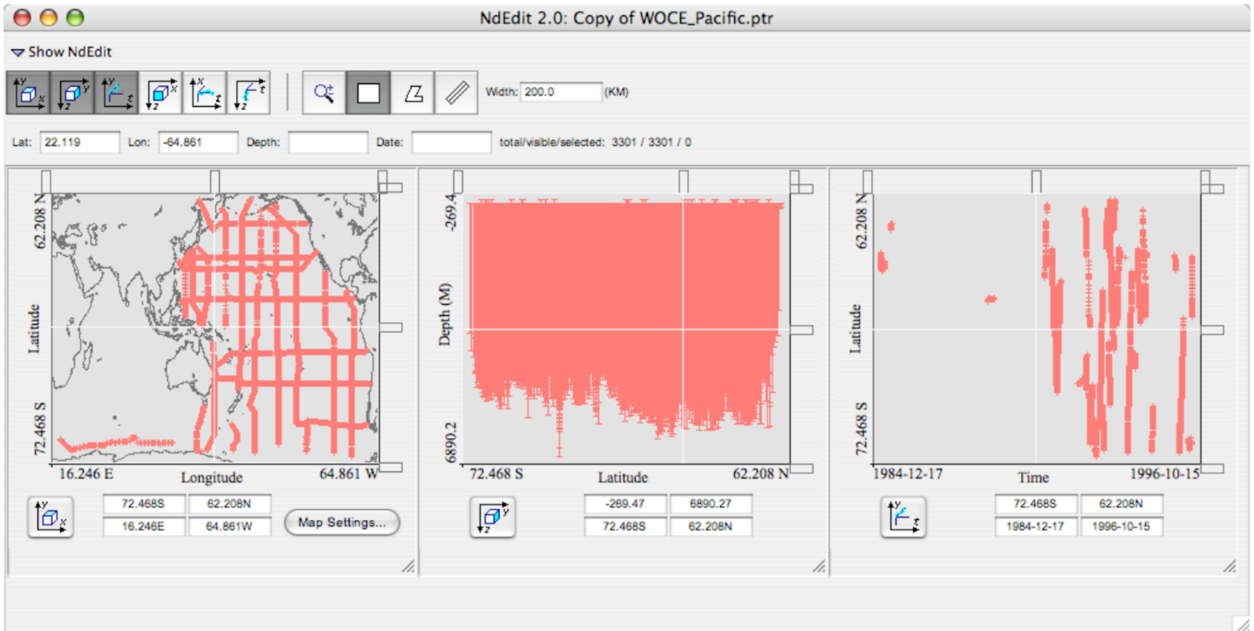


The following illustration shows how the 22 data files look on a JOA map plot:

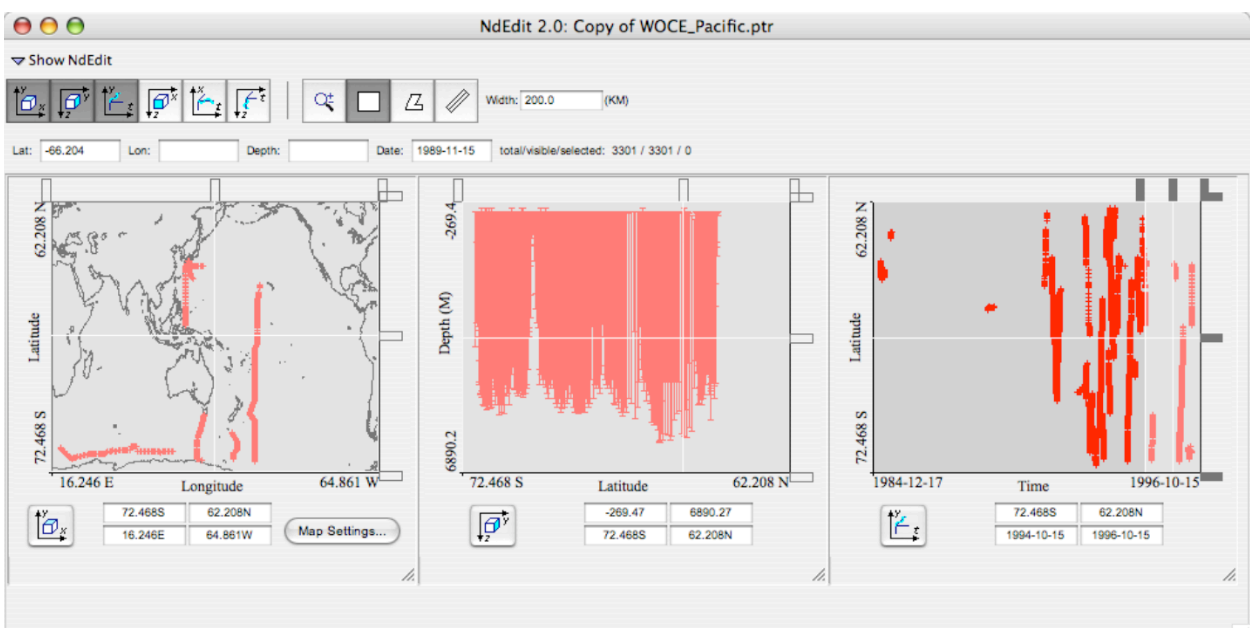


2) Export the collected data as a netCDF Section using the *Export netCDF Section* command in the *File* menu. It is helpful (although not required) to create the netCDF section in a new folder. You will be prompted to save the netCDF section in a new file with a .ptr extension - for example PacificDB.ptr. This file is an EPIC 'pointer' file and is used by NdEdit to display the locations of the individual profiles in its various views. Note: JOA will 'expand' all the sections into the individual profiles and save each station/profile as an individual netCDF file in the folder where the netCDF section is saved.

3) Open the pointer file created in Step #2 from the initial JOA window using the *Browse* command from the *File* menu. If you haven't already done so, click the *Show NdEdit* button to display the NdEdit browser. The PacificDB.ptr file created above looks like this in NdEdit:

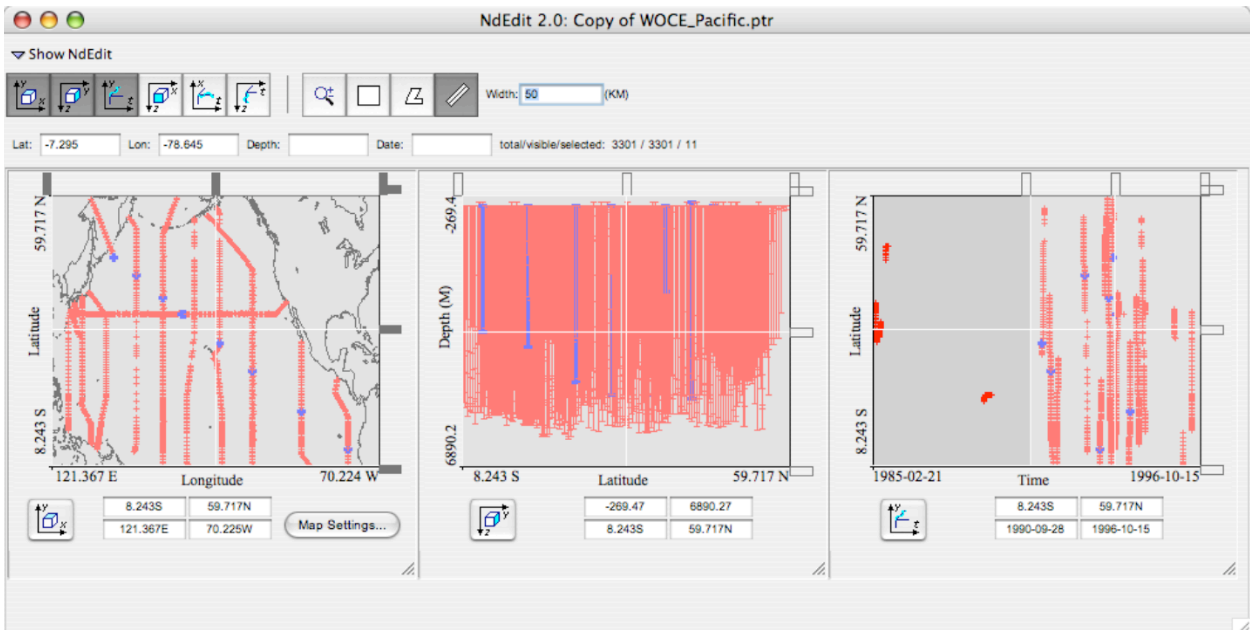


By default, NdEdit displays the three views with latitude as an axis. Click on the appropriate buttons in the toolbar to show the other views based on longitude. The illustration shows that profiles in the pointer file are concentrated in the North and South Pacific, were collected from 1966 to 1996, and include samples from the surface to approximately 6800 meters in depth. At this point one can use NdEdit's powerful interactive filtering to display profiles in a certain time, depth, and/or location domain. For example, using the time filter (dragging the rectangular handles on the horizontal axis) you can show only the profiles gathered since 1994:

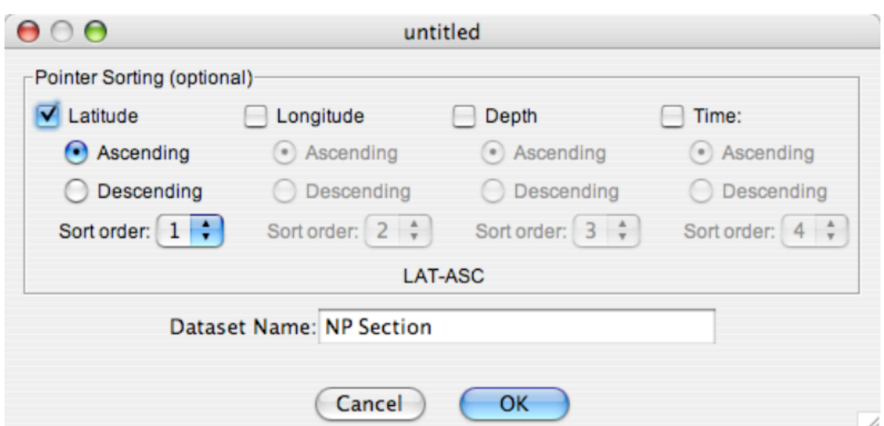


The filter region is shown in white in NdEdit's various views. In the above illustration, the latitude/longitude and the depth/latitude panels are still completely white because the filter region for these dimensions have not been changed. The values of the filter regions can also be set using the text fields at the bottom of each panel. See the NdEdit tutorial for more detail on filters.

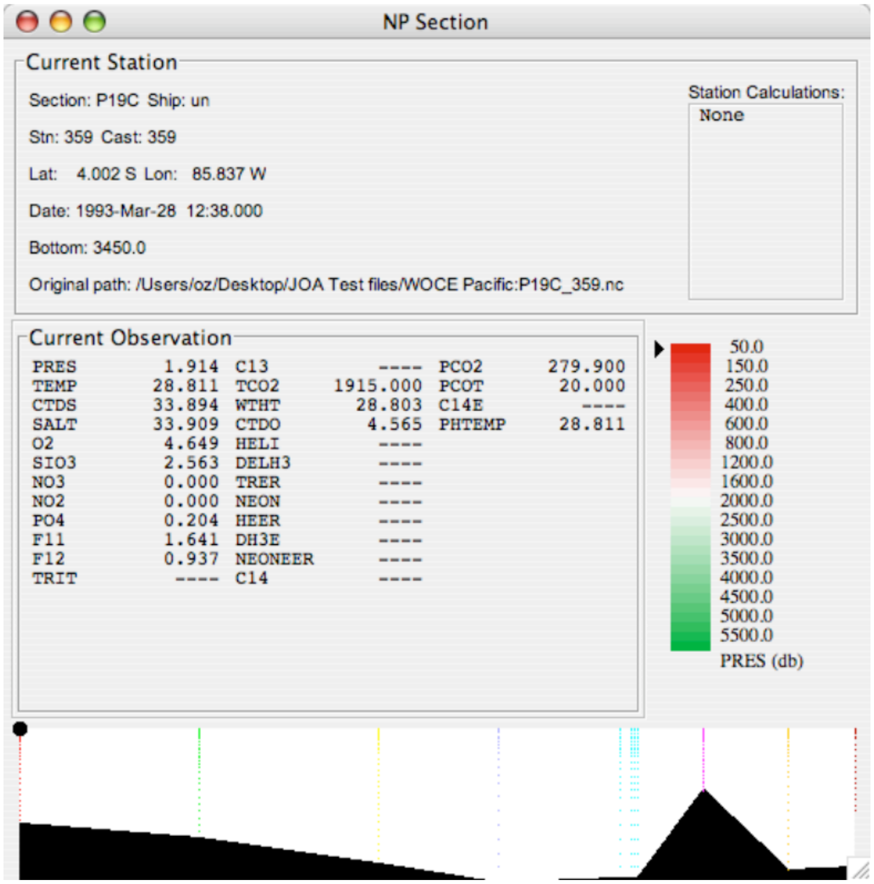
You can also use NdEdit's selection tools to create new sections for display and manipulation in JOA. This example illustrates using the Section Tool to create a new section across the North Pacific. Note: selected profiles are highlighted in blue in the NdEdit panels and the latitude/longitude panel has been zoomed to show the North Pacific:



To open the selected profiles, use the *Open selected files* command in the *File* menu of the main JOA window. NdEdit will present a sorting dialog to specify how you want the profiles ordered in the resulting JOA Data Window. You can also assign a name for the new section:



For the example above, a new JOA data window is displayed with the selected profiles:



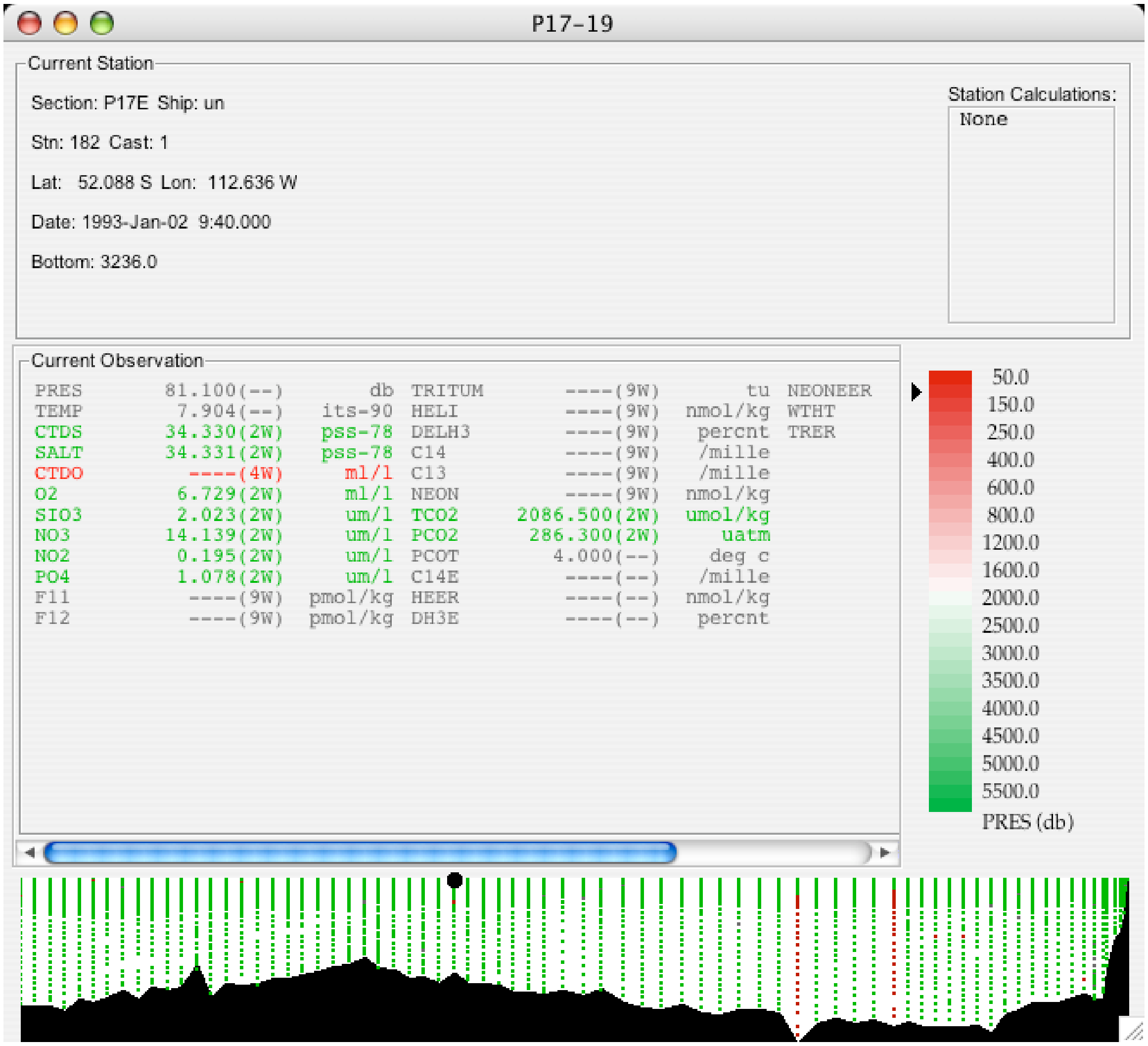
Summary

This example illustrates how one can use JOA to create a database of individual profiles (in this case nearly 3600 individual profiles) and use NdEdit to browse and create new sections from individual profiles. Once created, custom sections can be saved as a JOA binary file or a new pointer file that contains just the metadata for the selected profiles. Note: deep, full-resolution CTD profiles are very memory intensive. One might consider using JOA's decimation features to reduce the size of CTD profiles before creating a large collection of CTD profiles for use in NdEdit.

Maintain Databases

CHAPTER 5. DATA VISUALIZATION

Data Window

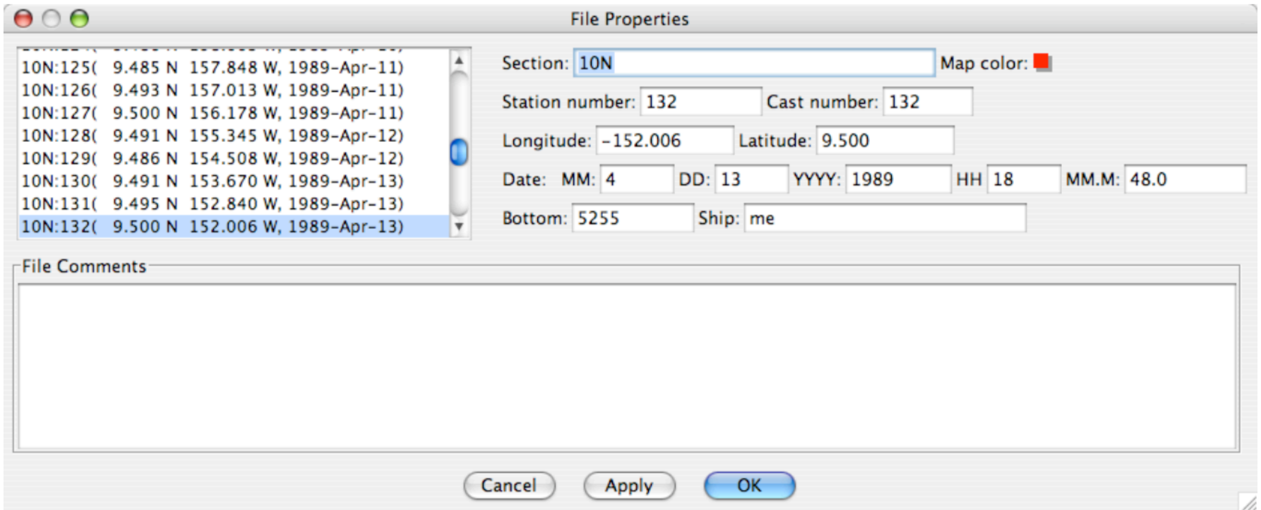


The Java OceanAtlas Data Window provides text and graphics links to open data files, calculated parameters, and the color/contour bar used to color profile and property-property plots. Above is a Data Window copied from a Java OceanAtlas session.

The above is not the default Data Window, but instead several options have been enabled after opening the data file. Each of these options is explained below.

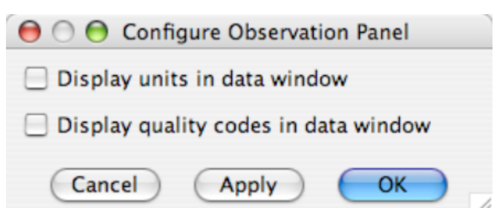
The data file name is listed in the window name ('title') bar. If multiple data files are opened by 'adding', the name in the title bar changes, at first concatenating section names and finally switching to the number of files (e.g., '3 Files').

The panel at the top of the Data Window contains the header information at the station where the browsing dot (cursor) is currently located. Double-clicking (or right-clicking) within the *Current Station* panel brings up the File Properties editor:



The File Properties editor permits editing station header values (meta-data), and also permits adding and saving comments regarding the entire data file. The File Properties dialog is where one can examine previously-saved file comments. The color of the station symbols for each open section can be changed by clicking on the *Map color* swatch, which brings up a color chooser. These changes are not saved unless the edited data file is saved (via the *File* menu) before quitting Java OceanAtlas. These changes are not saved unless the edited data file is saved (via the *File* menu) before quitting Java OceanAtlas. Note: File comments are only preserved when saving data in JOA's internal binary format files.

The center/left panel of the Data Window shows the values of all observed and calculated parameters at the current browser point. In this case a suite of calculated parameters is shown in addition to the original data. Calculated parameters appear in the Data Window in the order they were calculated. Two special features have been enabled in the Data Window shown above. They are accessed by double-clicking (or right-clicking) on this portion of the Data Window, which brings up the Configure Observation Panel dialog:



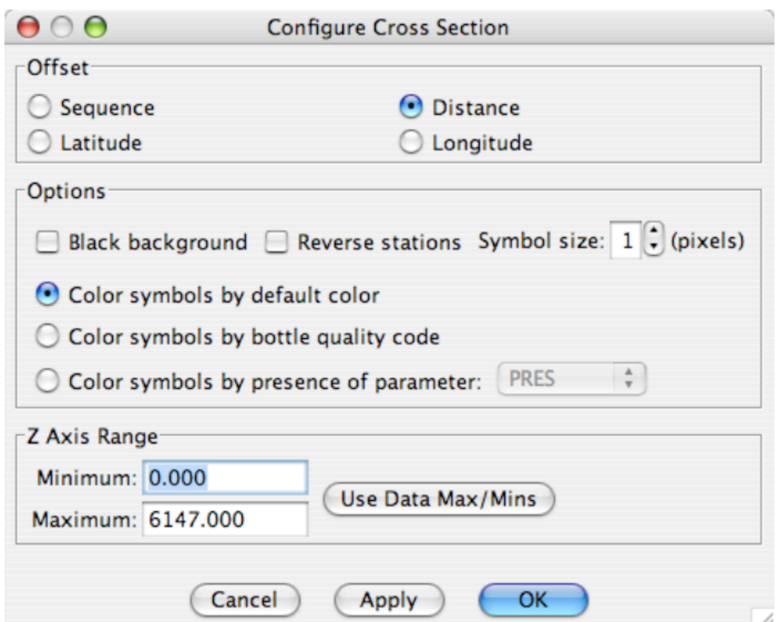
Two options are available which affect the Data Window display: The first, if checked, displays the units of each parameter (if specified) in the Data Window. The second, if checked, displays the parameter quality codes (if available) in the Data Window. An example of data with embedded quality codes understood by Java OceanAtlas is WHP-Exchange format data. Java OceanAtlas can read WOCE ('W') and IGOSS ('I') data quality codes. (See User Guide section on reading various file formats.) Both these options have been enabled in the example Data Window shown above.

Note that with the 'Display quality codes in data window' option checked, in addition to showing the quality codes Java OceanAtlas colors the parameters in the Data Window black for no quality code assigned, green for 'good' quality code, and red for 'bad' quality code. Java OceanAtlas usually uses IGOSS quality codes internally and in those cases - such as the WHP example shown - translates the WOCE quality codes to IGOSS. (The codes and translator are included in the documentation from the WOCE Hydrographic Program Office.)

The current color/contour bar used for profile and property-property plots is displayed on the middle right. The triangular marker on the left side of the color/contour bar shows the specific color appropriate to the observation currently in the Data Window. Double-clicking on the color/contour bar brings up the Color Bar Editor. (See separate User Guide entry for Color Bar Editor.)

The section display at the bottom shows (in default mode) the full suite of observed data points. Clicking any one point in the section display in the Data Window will move the browser to that point (and update the browser position on all plots).

Double-clicking (or right-clicking) on the section display will bring up the Configure Cross Section dialog.



Options are presented to:

change the relative station spacing in the section display from *Distance* (default) to *Sequence*, *Latitude*, or *Longitude*;

change the background color for the section display to black;

reverse the left-right order of the station display;

change the observation symbol size;

color the observation symbols on the section representation by bottle quality code (if present) or presence of selected parameter; and

change the size of the observation symbol on the section representation.

Note that when a data file contains multiple sections, or when multiple data files are open, Java OceanAtlas displays each section in a separate color in the Data Window. That section color - which is also used to color the station symbols on a Java OceanAtlas map plot when the *Station Colors* panel of the tabbed dialog is set to *JOA assigned colors* - can be changed via the *Map color* swatch (just click on it) in the *File Properties* command under the Java OceanAtlas *Edit* menu.

Browsing

Browsing is one of the central features of Java OceanAtlas. Browsing allows one to move an observation point or marker (you can also think of it as a data cursor) through the open data file(s), linked to similar markers in all open plots. The data marker or cursor is shown by a small circle in all plots except for contour plots in "cross-section mode" (see below). The symbol size and style for the data cursor can be changed in the Preferences dialog. The Data Window displays numerical values of all variables at the current browsing location. This capability to interactively link plots and data offers tremendous power in data examination.

The arrow keys on the keyboard are the basic browsing tools: left and right arrows move from one station to the next. Watch the station number change and latitude/longitude change in the Current Station panel of the Data Window, while the data cursor moves on the cross section in the Data Window and on any open JOA plots. This works in all 'directions': clicking on a data point in the cross section display in the Data Window, or on a station dot on a map, or on a data point on a plot, will move the browser on all Java OceanAtlas plots. The up and down arrow keys move methodically 'up' and 'down' the data from a single station (until either the top or bottom of the profile is reached).

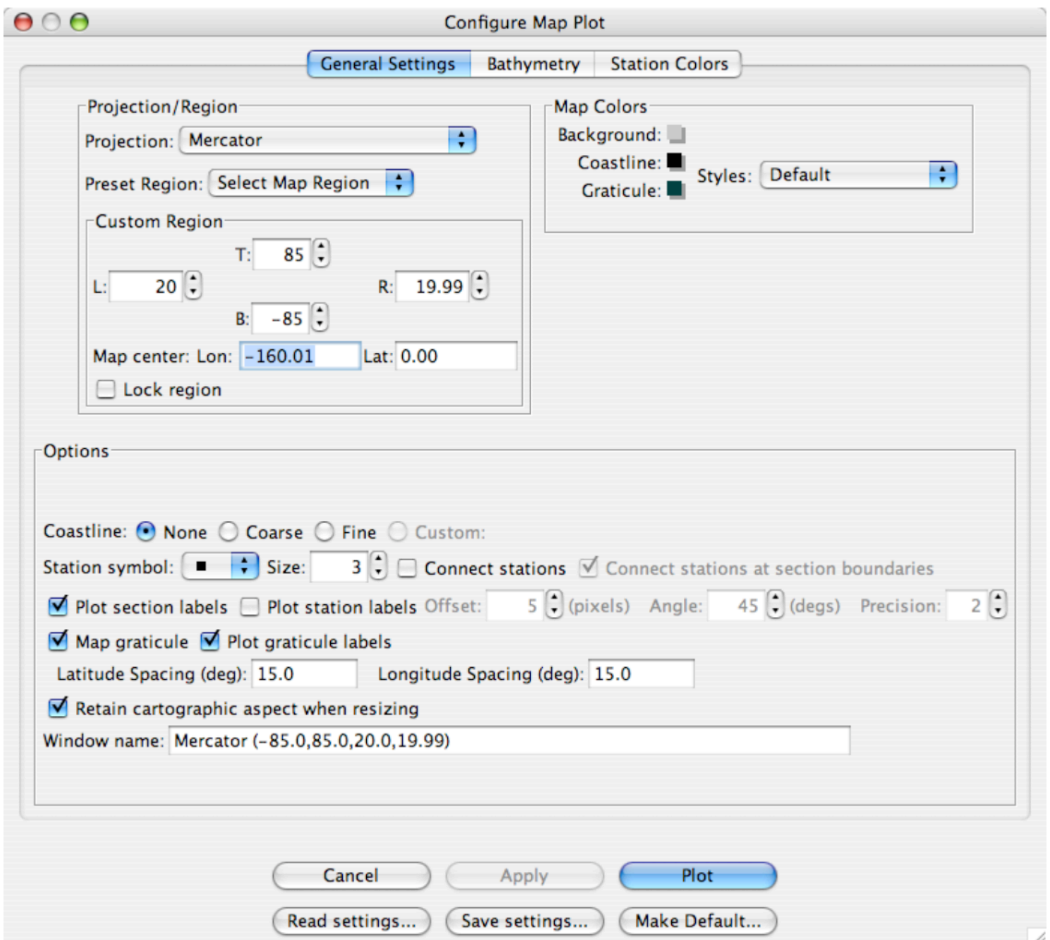
> Tip: Clicking on a data point in the cross section display in the Data Window can be a useful way to find or relocate the browser.

Contour plots attempt to display the current browsing location even though they are derived from interpolated values rather than the original measured values. In general, the browsing spot in contour plots is the closest interpolated data to the observed values at the current browsing position in the data file

On profile plots with pressure as the vertical axis, the horizontal arrows browse along an approximately constant-pressure surface (finding the nearest bottle to that surface). Browsing performance when the profile y axis is sigma-0 or some other variable has a somewhat more spotty appearance because browsing horizontally follows as closely as possible to the current isobar, instead of the current y-axis value for such plots. To examine characteristics along one level of a different surface type more accurately, interpolate the data onto the appropriate surfaces and examine the interpolations via Contour plots.

Map Plots - General Settings Panel

Selecting *Map* from the *Plots* menu (or cmd/ctl-M) brings up the Configure Map Plot dialog:



The dialog will reflect the default map settings if you are creating a new plot or the settings of the current map if editing an existing plot.

From any of the three tabbed pages the buttons at the bottom of the dialog allow the user to

Cancel closes the dialog without producing a map or applying any changes to an existing plot,

Apply the changes without leaving the dialog (*Apply* is only enabled when editing existing maps),

Plot or re-plot according to the specifications in the three tabbed windows,

Read settings from saved map plot configuration files,

Save settings into a map plot configuration file, or

Make Default the current specifications in the three tabbed panels.

A huge range of (optional) customization is possible:

In the *Projection/Region* section of the dialog,

To the right of *Projection:* is a scrollable list of available map projections. [The general subject of map projections is outside the scope of the User Guide. We provide an illustrated comparison of the available Java OceanAtlas map projections in a separate section of the User Guide.]

To the right of *Preset Regions:* is a scrollable list of pre-created map regions, i.e. subsets of the global map.

Custom Region permits the user to type in the Top and Bottom latitudes, and Left (west) and Right (east) longitudes to be used to draw or redraw a map plot (when the Configure Map Plot dialog is opened by double-clicking on any map plot, it shows the latitude and longitude limits of that plot).

Note: Java OceanAtlas uses decimal degrees, with positive values for North latitudes (0 to 90) and East longitudes (0 to 180) and negative values for South latitudes (-90 to -0) and West longitudes (-180 to -0).

Clicking on any of the three color swatches in *Map Colors* brings up a standard color chooser to permit changing the color of the map plot *Background, Coastline, Graticule,* or *Section Labels*. The overall color Style of the map can be changed between the *Default*, or to a white or black map background.

In the *Options* section of the dialog,

Coastline choices are *None, Coarse, Fine*, or a choice of a *Custom* coastline file. Standard coarse and fine global coastline files are included with the Java OceanAtlas installation. (Communicate with John 'Oz' Osborne, the developer, at tooz@oceanatlas.com to learn how to prepare a custom coastline file.)

Note: the option of no coastline is the best choice when one of the filled bathymetry options is chosen, because the fills are based on the ETOPO elevation/depth files, which contain data for land and ocean areas, hence needing no additional coastline information.

Selecting *Connect stations* adds a line joining stations within a section. There is the option to join stations across sections boundaries

Several choices of station *Symbol* are available, and the symbol size can be set as needed. Large sizes (8-10) are useful when the station symbols are colored by water properties.

Plot section labels, when selected, adds a map label for each logical section in all the open data files.

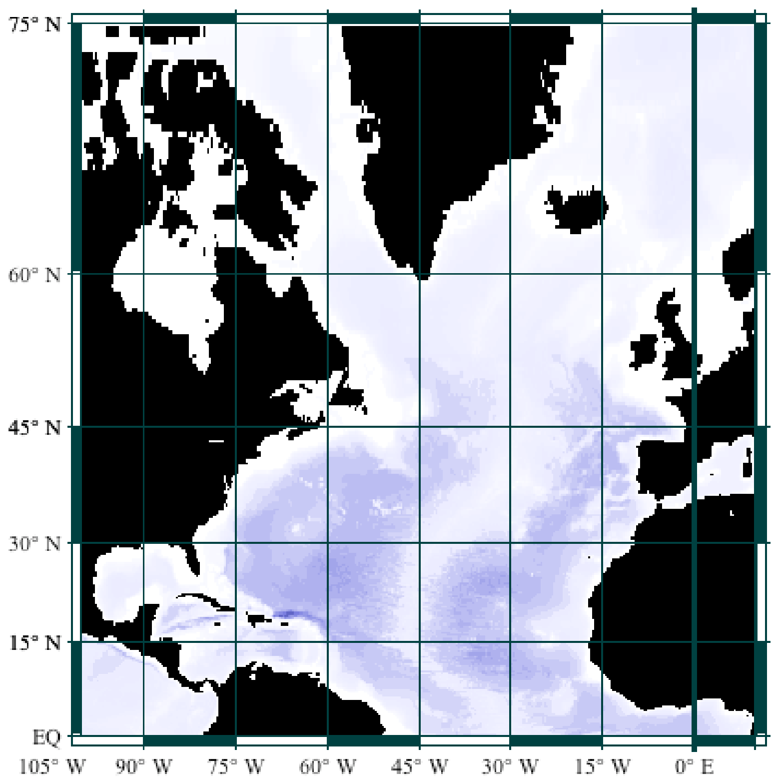
Map graticule, when selected, adds latitude and longitude lines to the map, and *Plot graticule labels*, when selected, adds labels to those lines. The *Latitude Spacing* and *Longitude Spacing* (intervals in degrees) of those lines can also be specified.

The cartographic aspect of a Java OceanAtlas map can be retained when the plot is resized.

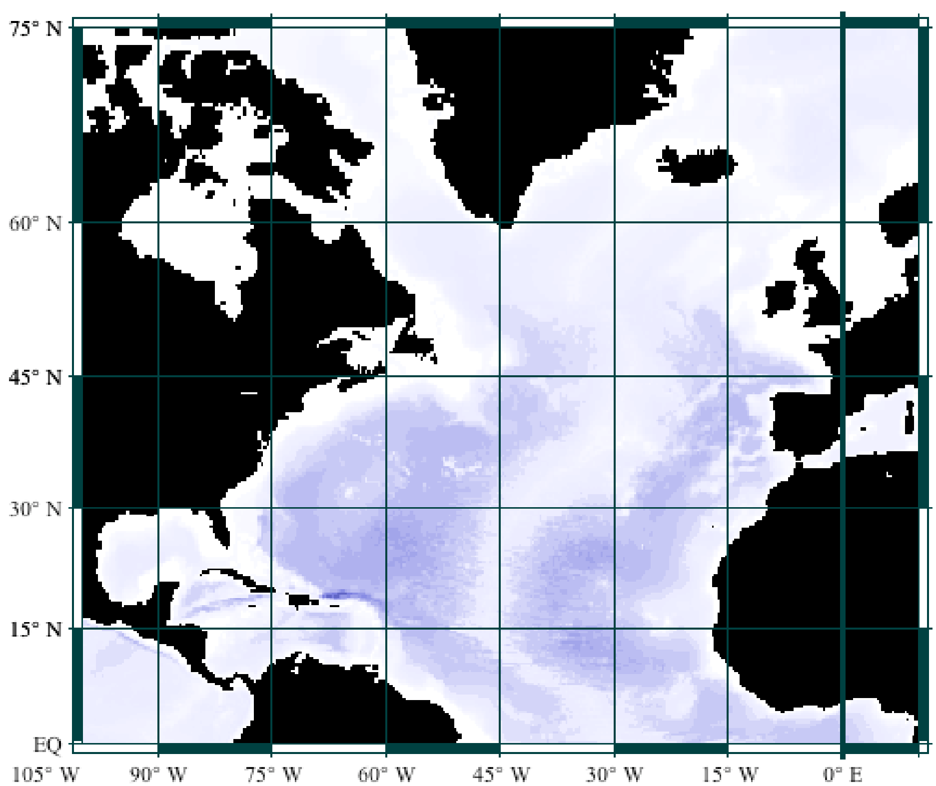
A default *Window name* is suggested but can be changed.

Map Projections

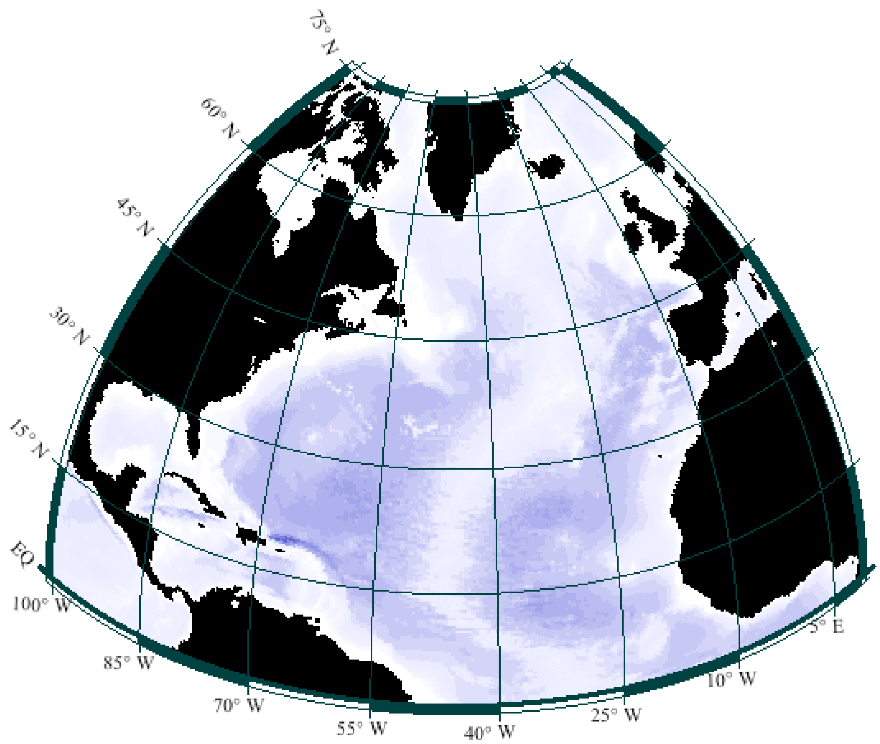
Here we illustrate the available projections, using the same general ocean area and map graticule settings where feasible.



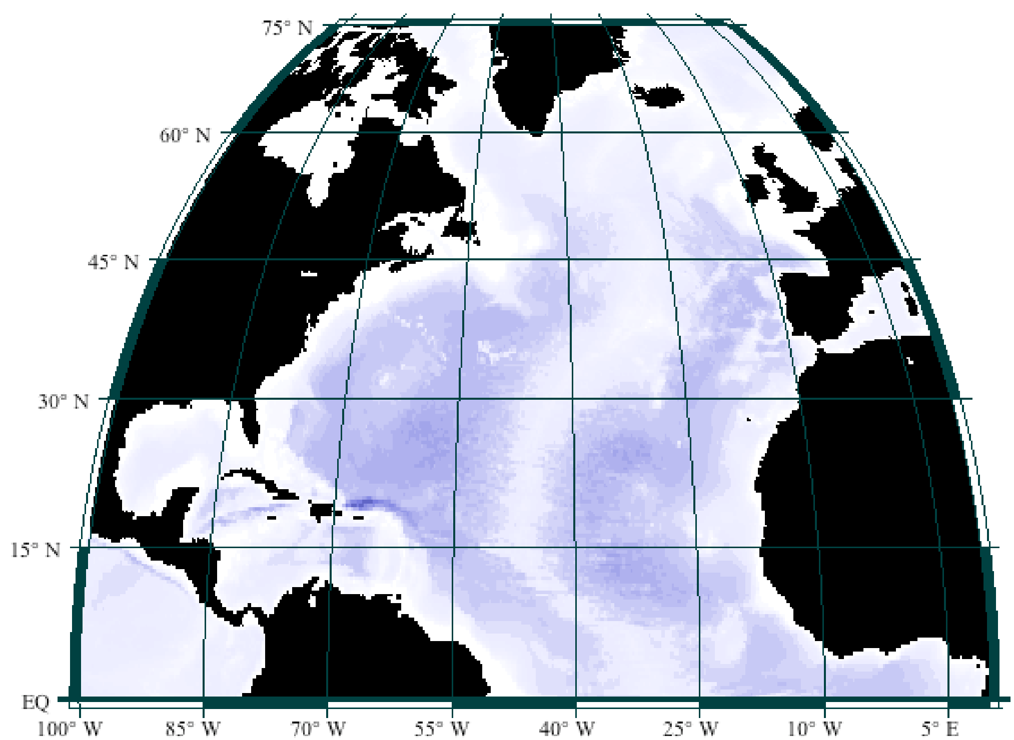
Mercator Projection



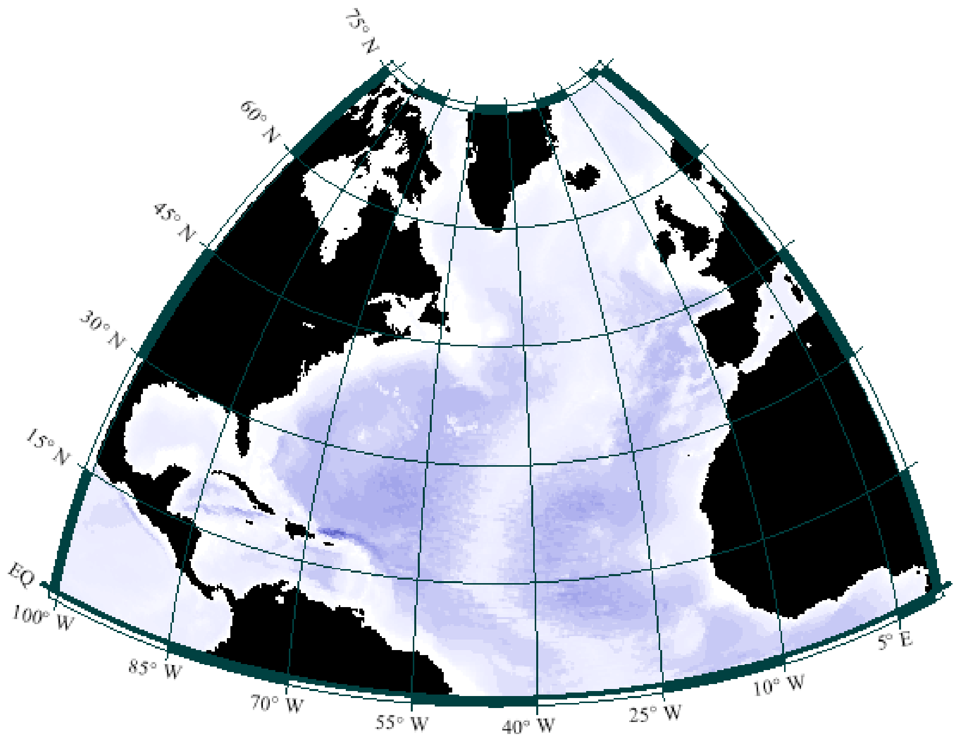
Miller Projection



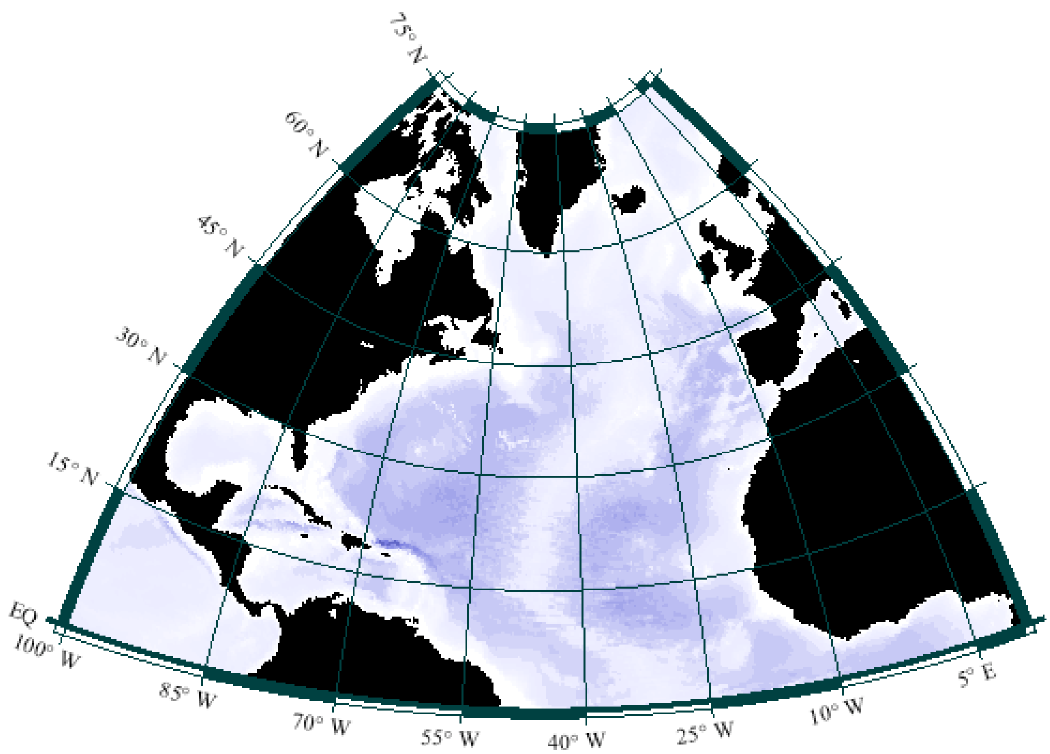
Orthographic Projection



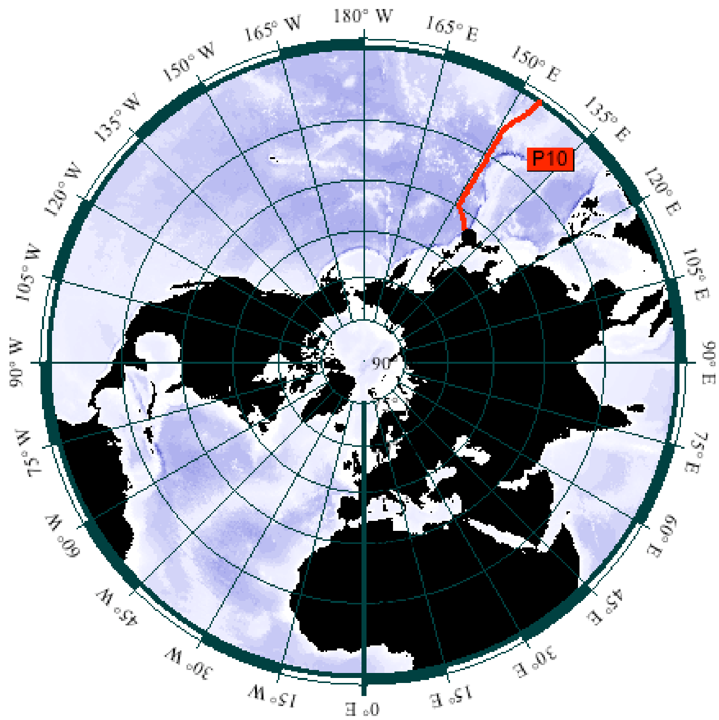
Mollweide



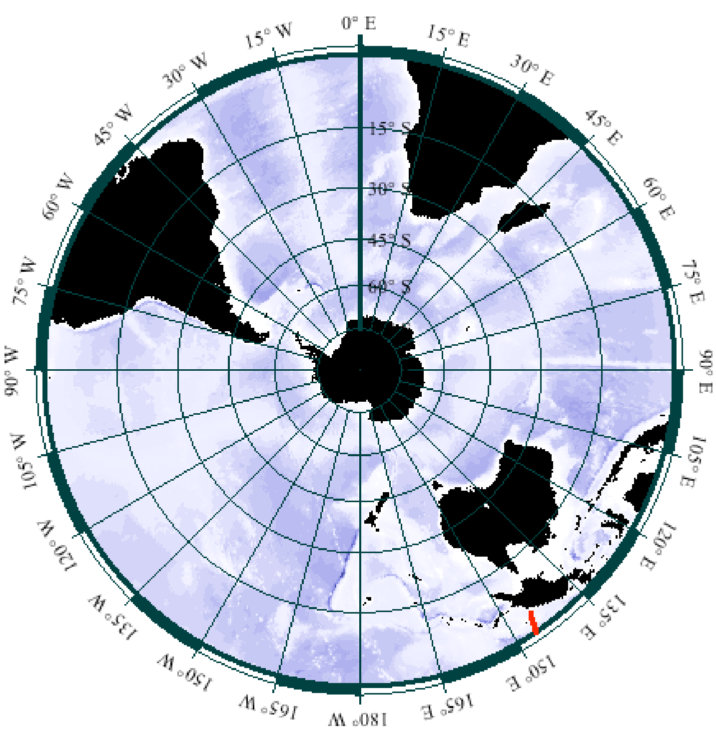
Lambert Equal Area



Stereographic



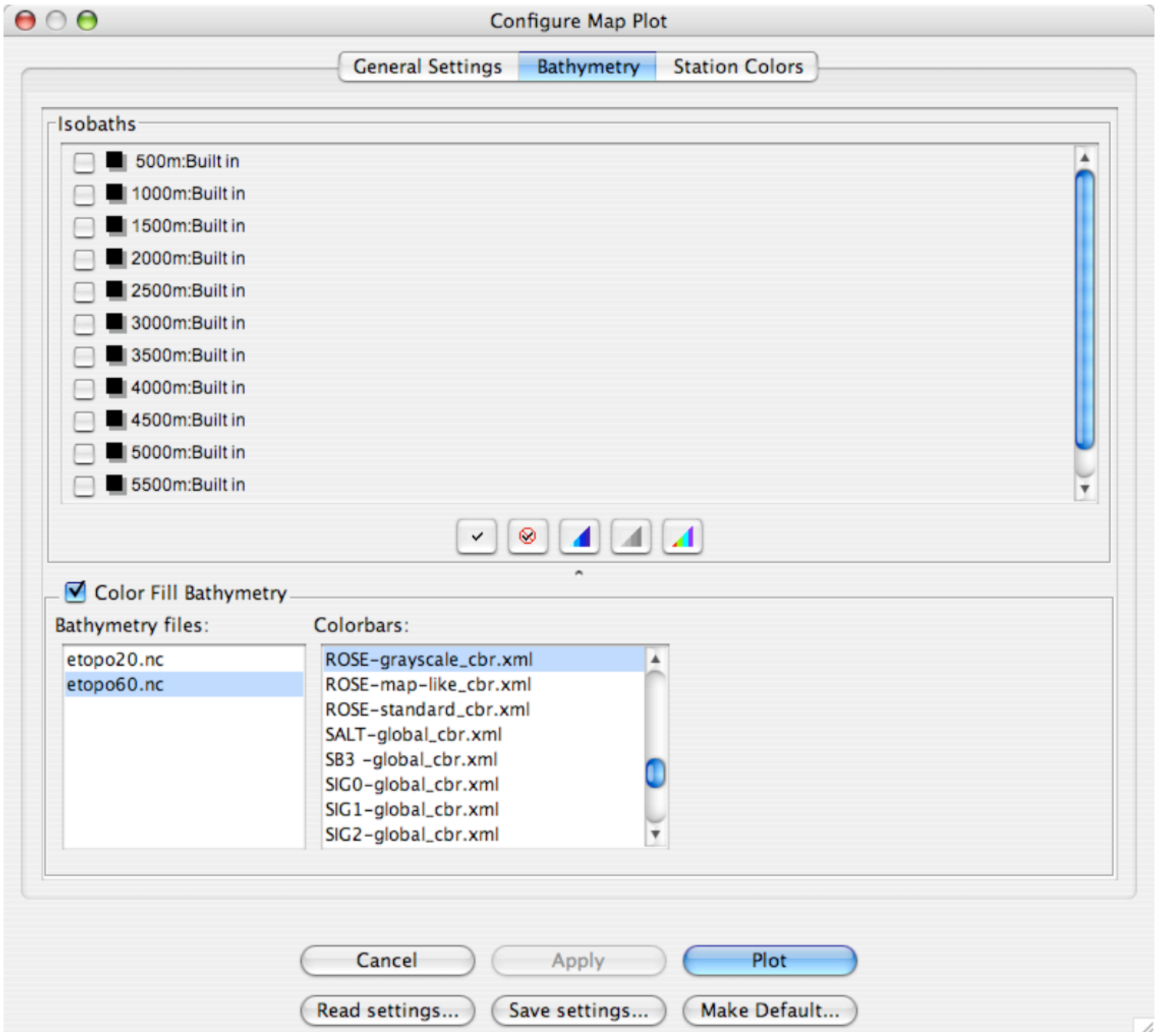
North Pole



South Pole

Configure Map Plot - Bathymetry Panel

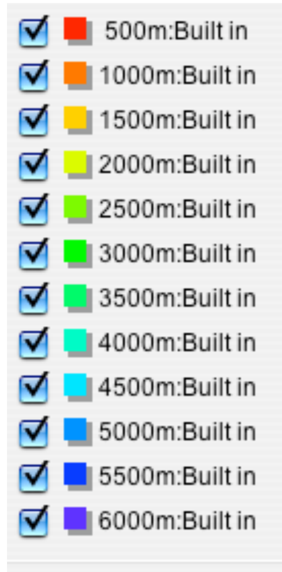
The *Bathymetry* tab on the Configure Map Plot dialog brings up the settings choices for adding bathymetry to Java OceanAtlas map plots:



The settings in the top panel of the dialog allow drawing isobath overlay contours onto map plots:

From the scrollable list of isobath choices click on as many or few of the isobaths values as are needed on the plot.

Click on one of the three color ramp buttons (the three right choices of the five buttons underneath the scrollable list). Choices are a green-blue-purple ramp, an all-gray ramp, and a rainbow ramp. Here is an example of the rainbow color ramp applied to all the default isobaths:



You can also click on the color swatches next to individual isobath values to pick a custom color.

The two left buttons either select or deselect all the isobaths in the list.

If isobaths are added or removed when modifying an existing map plot, be sure to click on the color ramp button again to reset the colors.

Coastlines, available from the *General Settings* page of the Configure Map dialog, are a special kind of isobath. Don't forget to draw in a coastline if you are not using filled bathymetry.

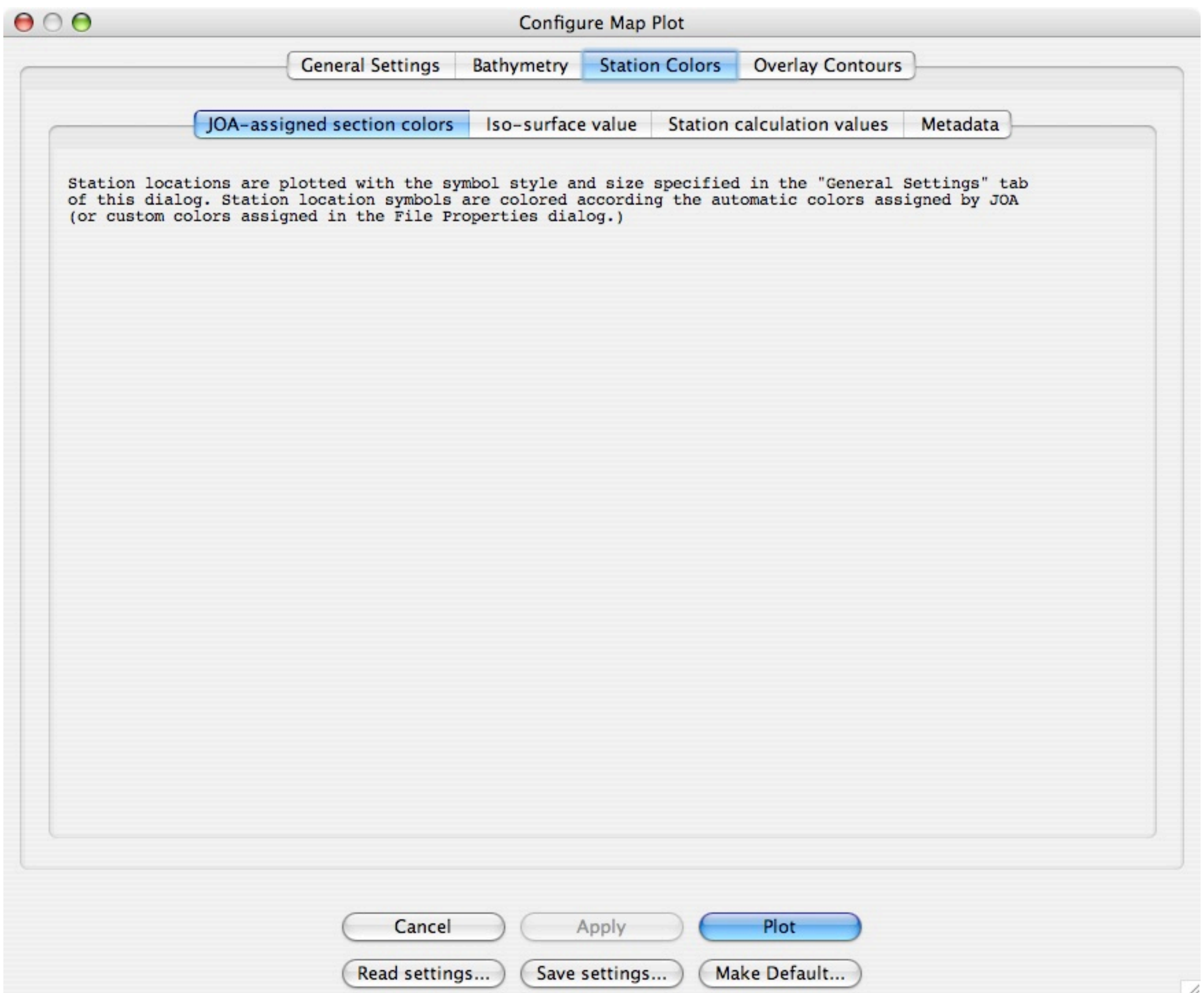
The isobaths included with Java OceanAtlas were provided by Joseph L. Reid of the UCSD Scripps Institution of Oceanography, and are designed for ocean basin scale map plots. Small-scale map plots, for example of Chesapeake Bay, would appear more realistic with higher-resolution bathymetry. Additional isobath files can be added to Java OceanAtlas. Contact John 'Oz' Osborne (tooz@oceanatlas.com) for more information if it is not yet available from the User Guide.

Color fill is the second choice of bathymetry on Java OceanAtlas map plots. This option maps gridded elevations and depths onto the map plot and colors them in accord with a chosen color bar. Java OceanAtlas uses the 'ETOPO' series of gridded bathymetry. The finest resolution supplied, via supplemental files which can be accessed from the same location (CD or website) as the Java OceanAtlas installers, is the 'ETOPO-5' bathymetry, which has a resolution of five minutes of latitude and longitude. These are large files, and so are usually not used on global scale plots. For basin or global scale map plots with color fill bathymetry on most computers, the 'ETOPO-20' bathymetry provides a reasonable balance between bathymetric visualization and computer resources and plotting speed. An 'ETOPO-60' file is also provided. Instructions may be provided on the Java OceanAtlas web site for creating and transferring ETOPO-2 (Sandwell-Smith) ultra-high resolution bathymetry/elevation files. These may be impractically large for ocean basin work, but for a smaller region such as a bay, this could be quite useful.

In addition to choosing a bathymetry data file, it is also necessary to choose a color bar to color the bathymetric fill data. The 'ROSE' (Relief Of Surface of Earth) series of color bars was designed for this purpose, and a set of choices is included with Java OceanAtlas. It is also possible to edit, make and save new 'ROSE' color bars with the Contour Manager (under the *Resources* menu).

Configure Map Plot - Station Colors Panel

The Station Colors tab on the Configure Map Plot dialog brings up the settings choices for coloring the station symbols:



JOA provides four ways to color station symbols:

JOA assigned colors uses a different color choice for each section it detects in the open data file(s). This makes it easier to discern individual sections on a map plot.

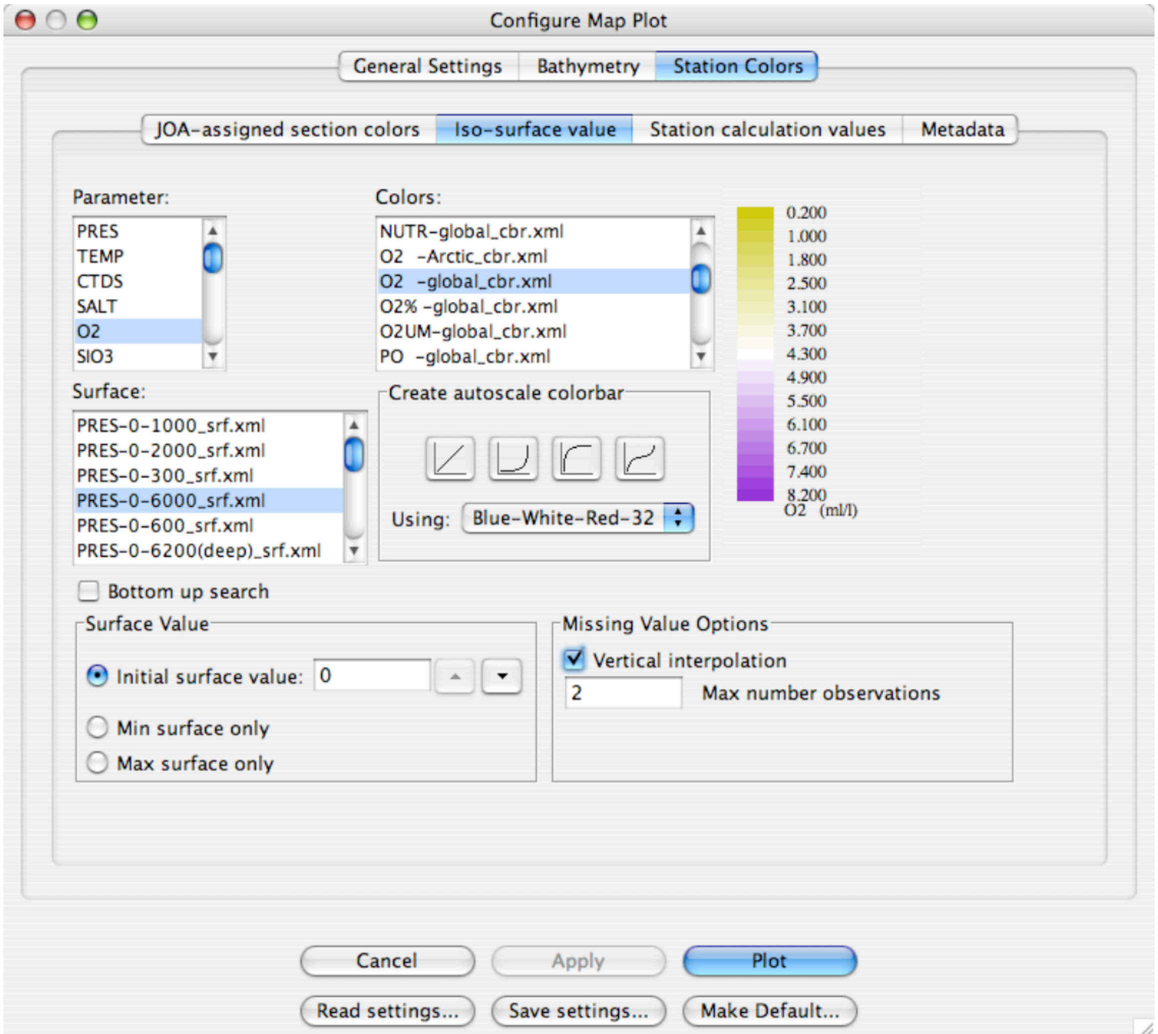
Iso-surface value assigns colors by interpolating the value of a observed (or calculated) parameter onto the value of a surface parameter. For example, salinity can be interpolated onto values of depth (i.e., pressure). After iso-surface colors are assigned to a map's station symbols, using the up and down arrows keys allows recoloring the symbols at the next/previous level of the surface parameter.

Station calculation values assigns colors according to the value of a computed station variable (e.g., depth of mixed layer).

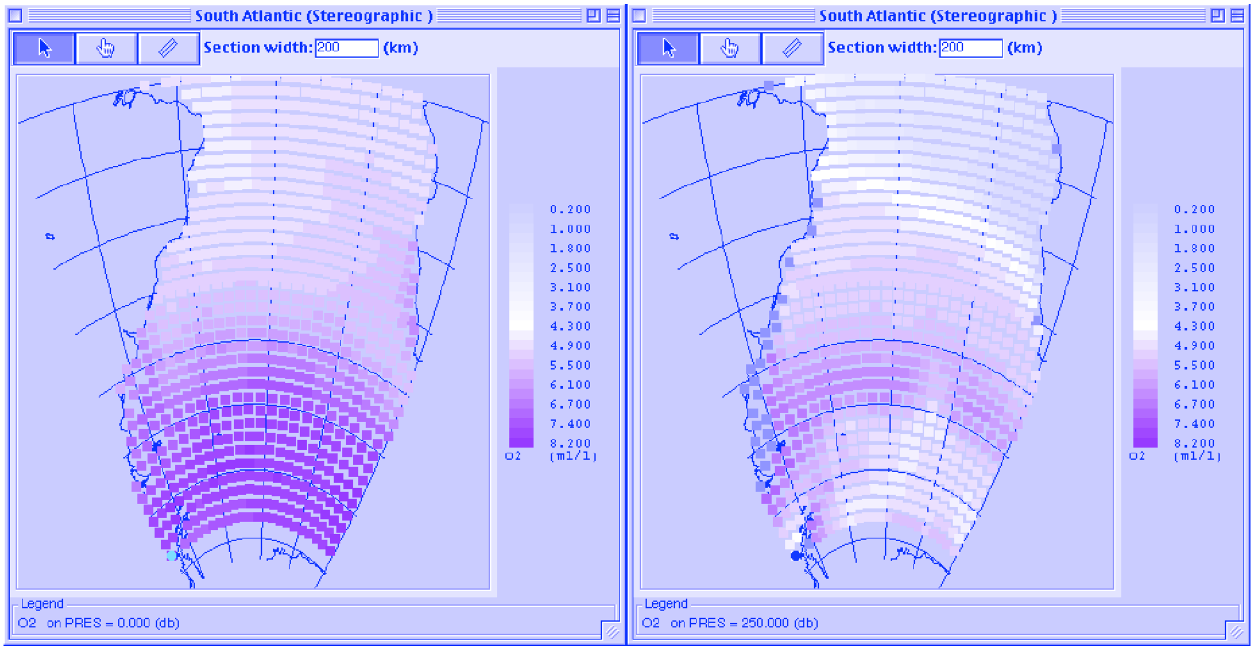
Metadata assigns colors according the values of station metadata values; latitude, longitude, date/time, or month.

Iso-surface Values

Iso-surface/station value coloring colors the station values by the color associated with a chosen parameter & color combination, with the parameter interpolated onto the surface - the set of standard levels chosen:



For example, the map plot below was made from the Levitus WOA98 South Atlantic Ocean data set available from the JOA web site or CDs, with *Parameter*: set to 'O2 ', 'Colors:' set to 'O2 -global_cbr.xml', and with the data interpolated onto the 'Surface:' 'PRES-0-6000_srf.xml' (square station dots were enlarged to provide near-continuous map color):



OceanAtlas South Atlantic map plot set up as described above (left) as originally drawn and (right) after browsing to 250 decibars using the 'down' arrow key.

The example shows that coloring a JOA map by iso-surface or station values can be a powerful tool, and with gridded data, such as the Levitus WOA98, makes a reasonable substitute for contoured data maps.

Iso-surface coloring options include:

Initial surface value: The iso-surface colors are chosen from the user-selected level of the surface. The user can optionally select a starting value from the list of values for the selected surface.

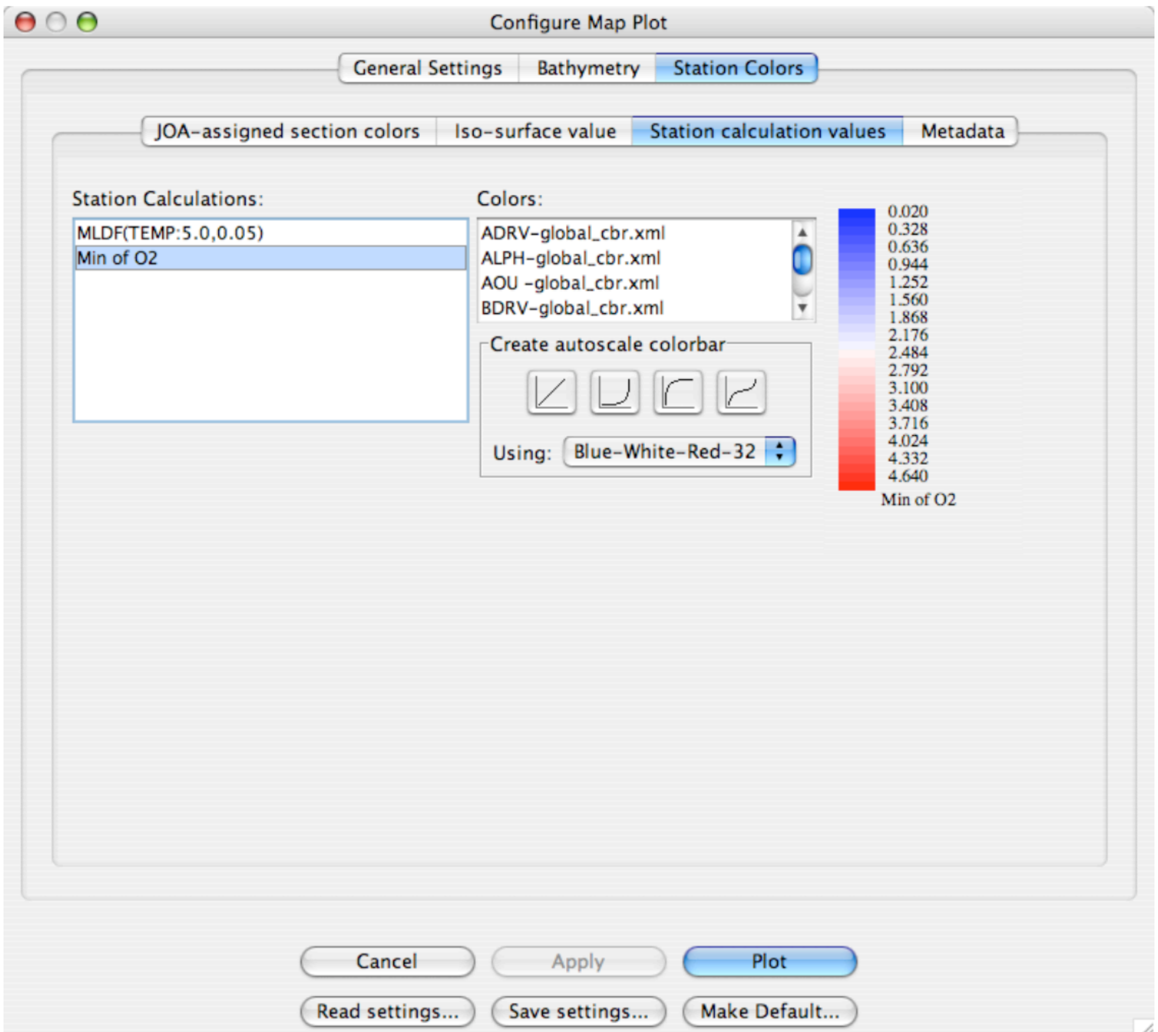
Min surface only/Max surface only: The iso-surface colors are chosen from the minimum or maximum value of the surface parameter found at the station. For example, if pressure is chosen as the surface, *Max* will color by the iso-surface value found at the maximum pressure of the cast. *Min* will return the iso-surface value at the minimum value of the surface. Use these to plot, for example, the 'surface' or 'bottom' salinities of all open casts. This is a better way to plot the surface value of some iso-surface than selecting the say the 5 db iso-surface. Some casts might have shallower observations which will be used if the *Min* option is used instead.

Bottom up search: Java OceanAtlas searches the data for the first match to an iso-surface level from the shallowest observation to the deepest observation, and vice versa for the *Bottom up* choice. When one is plotting iso-surfaces of a parameter onto pressure (i.e., pressure is the y axis), this is not an issue. But imagine plotting silicate onto standard surfaces of salinity in an ocean region where there is a mid-depth salinity extremum. In that case a different result would likely be obtained, depending on this choice.

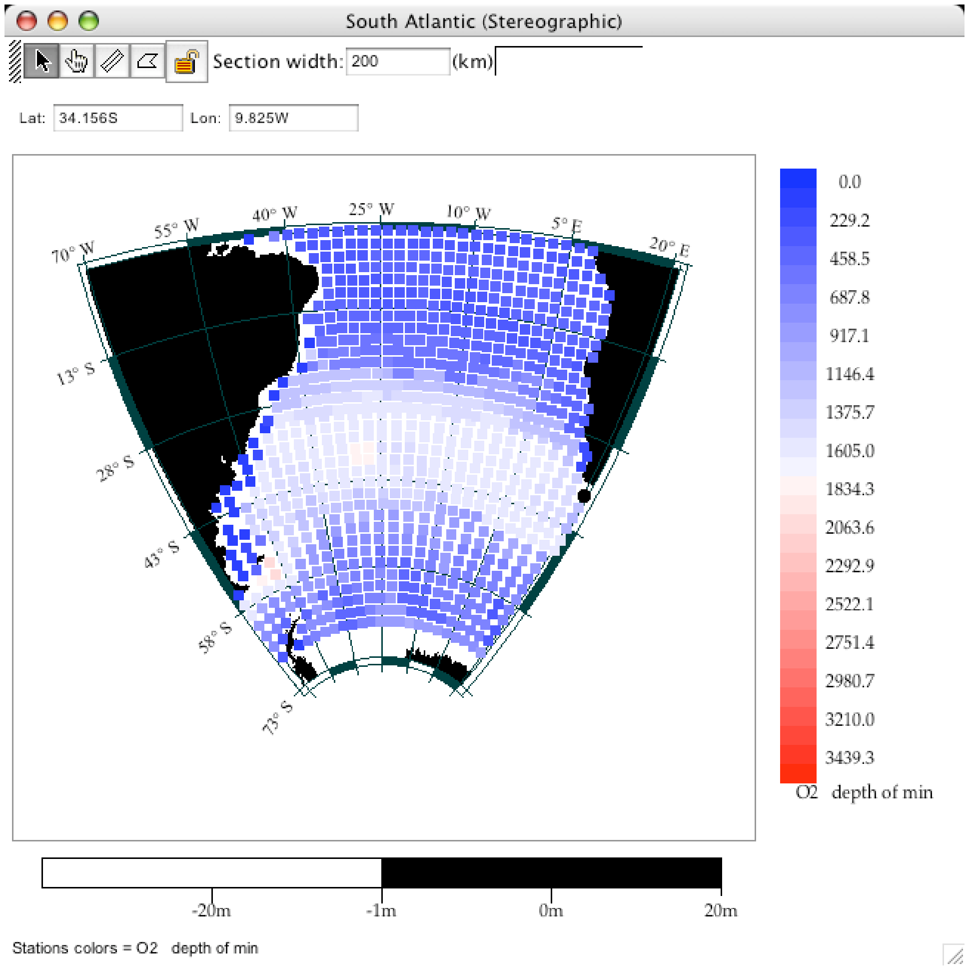
Missing value options: JOA interpolates a value to an iso-surface by finding the two values that bound the iso-surface value. If one of the bounding observations is missing then JOA will report a missing value. If *Vertical interpolation* is checked, JOA will attempt to find a non-missing value up to n observations away from the missing observation.

Station Calculation Values

JOA assigns the color for station symbols based on the value of a calculated station variable and using either a pre-existing or autoscaled colorbar:

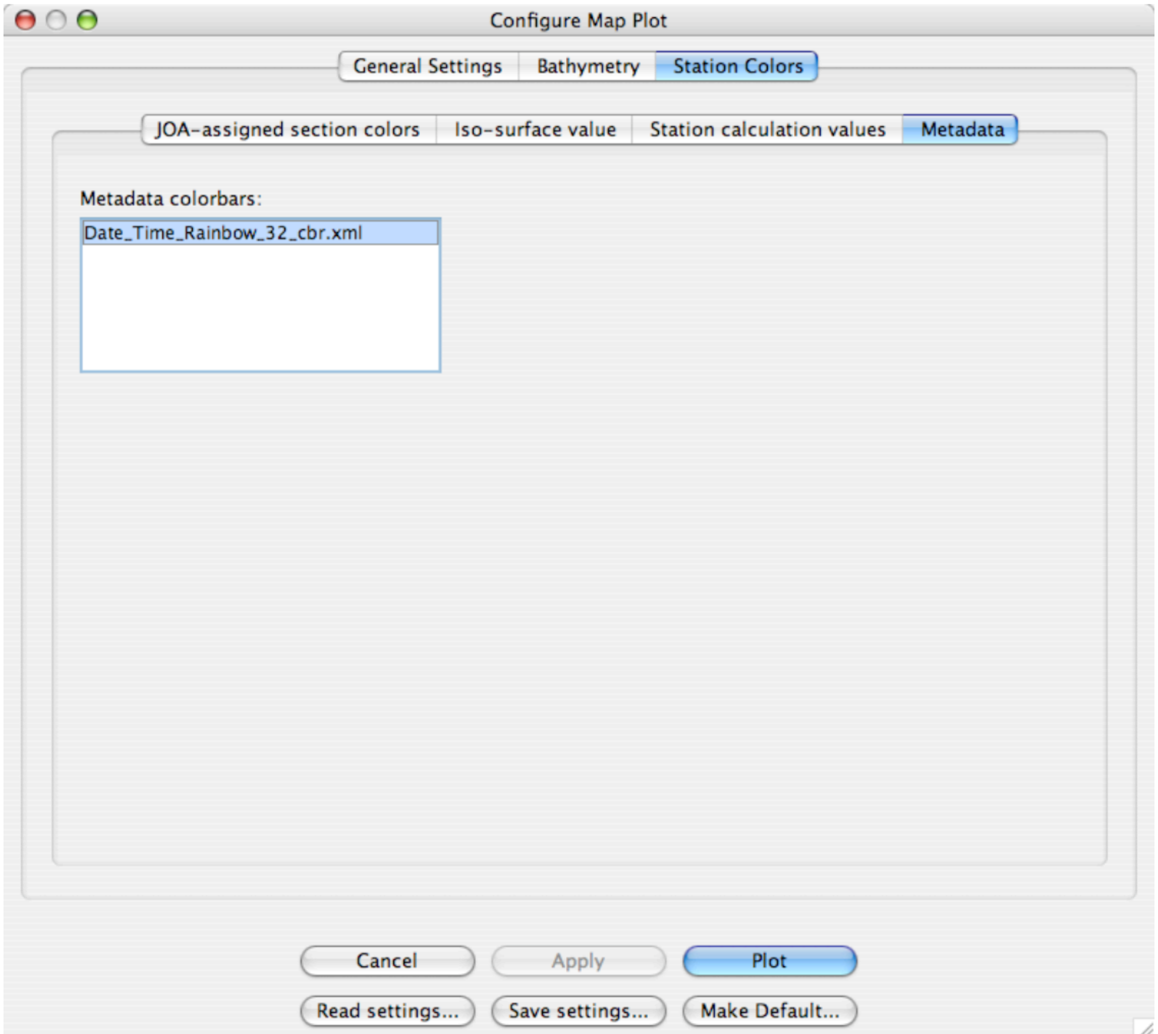


This figure shows the depth of the O2 minimum for the South Atlantic Levitus dataset:

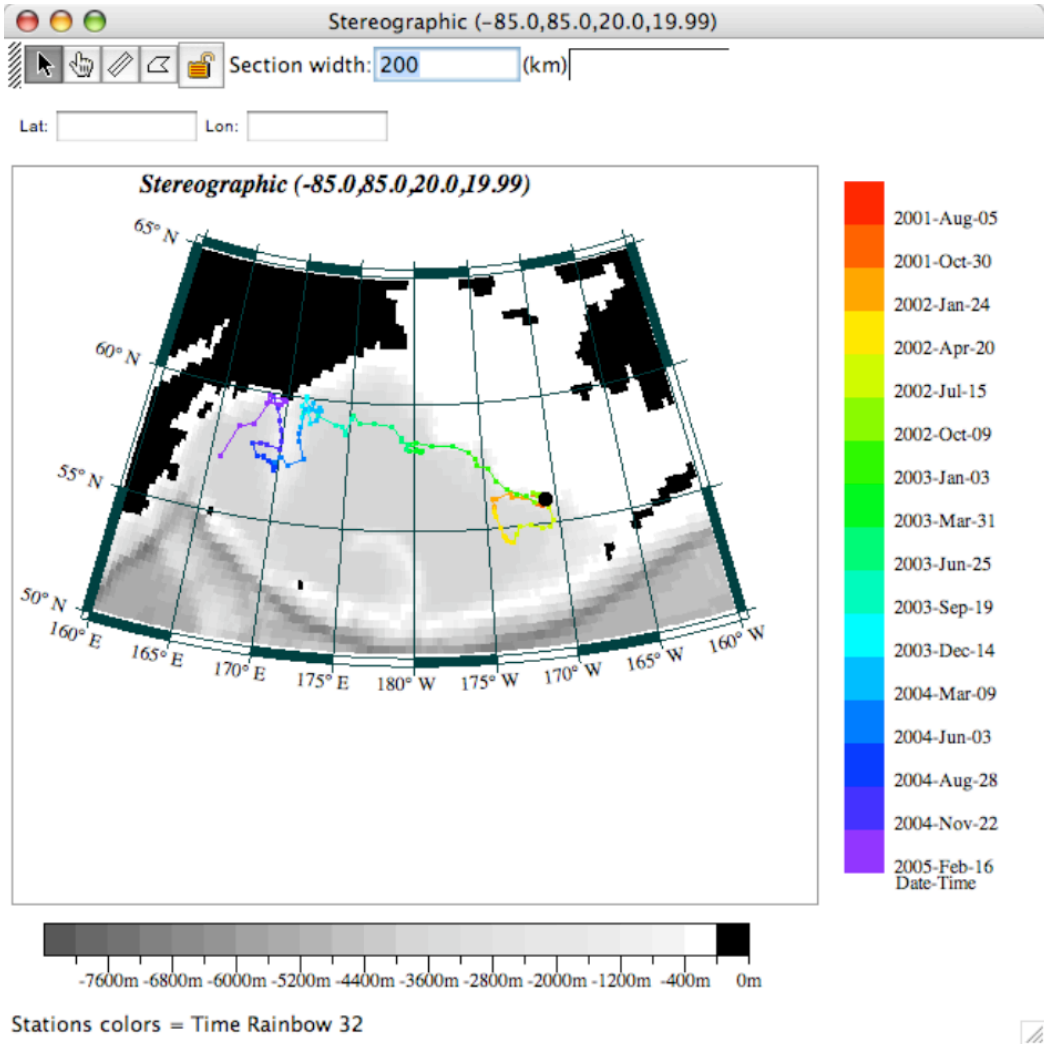


Metadata

This option allows JOA to color station symbols by station metadata including latitude, longitude, date/time, or month (although latitude and longitude aren't particularly useful on a map plot):

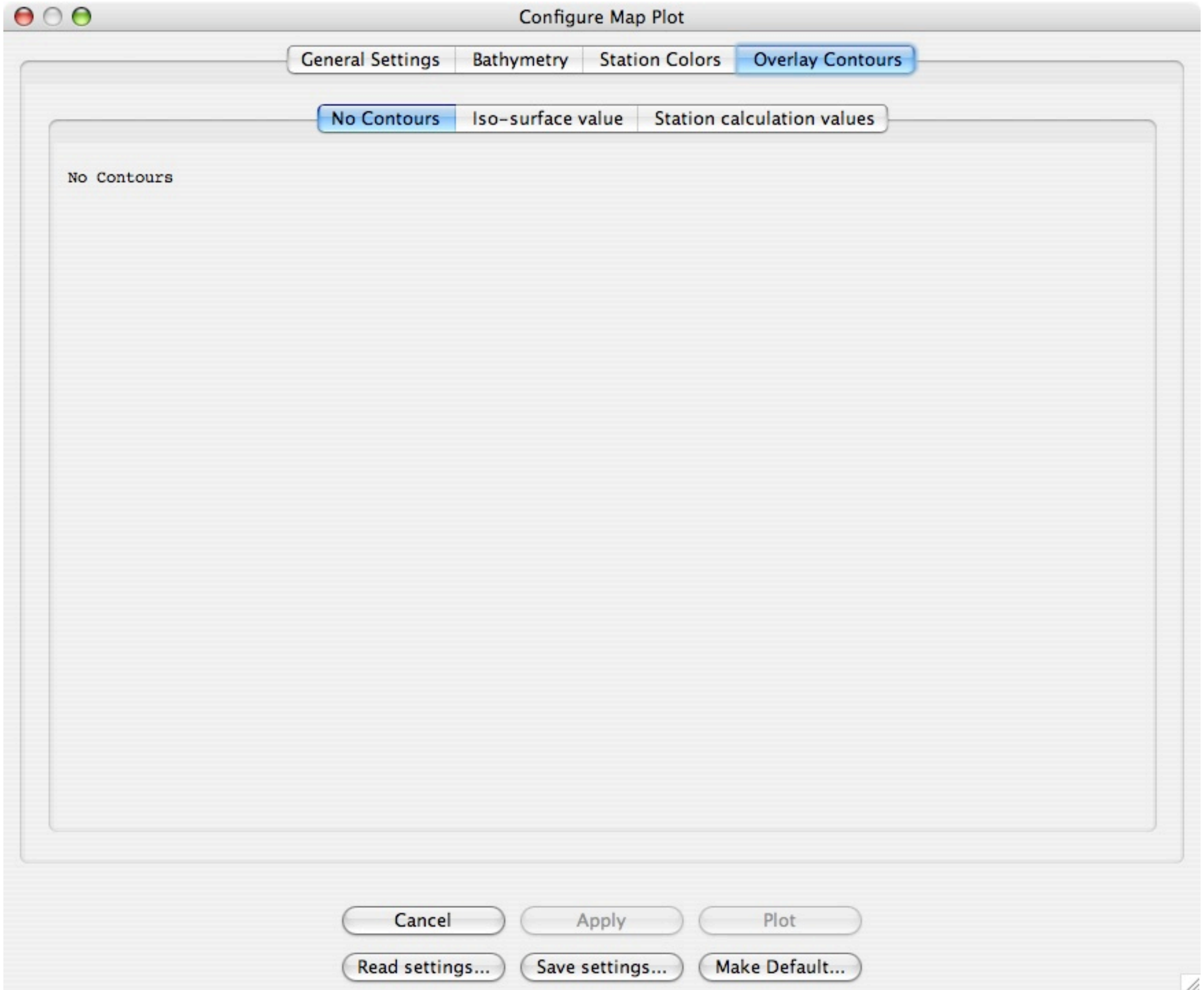


This option is useful when plotting drifter tracks:



Configure Map Plot – Overlay Contours Panel

JOA 5.0 includes the capability to present parameters interpolated onto a surface or a station calculation (as described above) for any number of profiles as a contoured overlay on a map plot. The *Overlay Contours* tab on the *Configure Map Plot* dialog brings up the settings choices for overlaying a contoured parameter field onto a map plot:

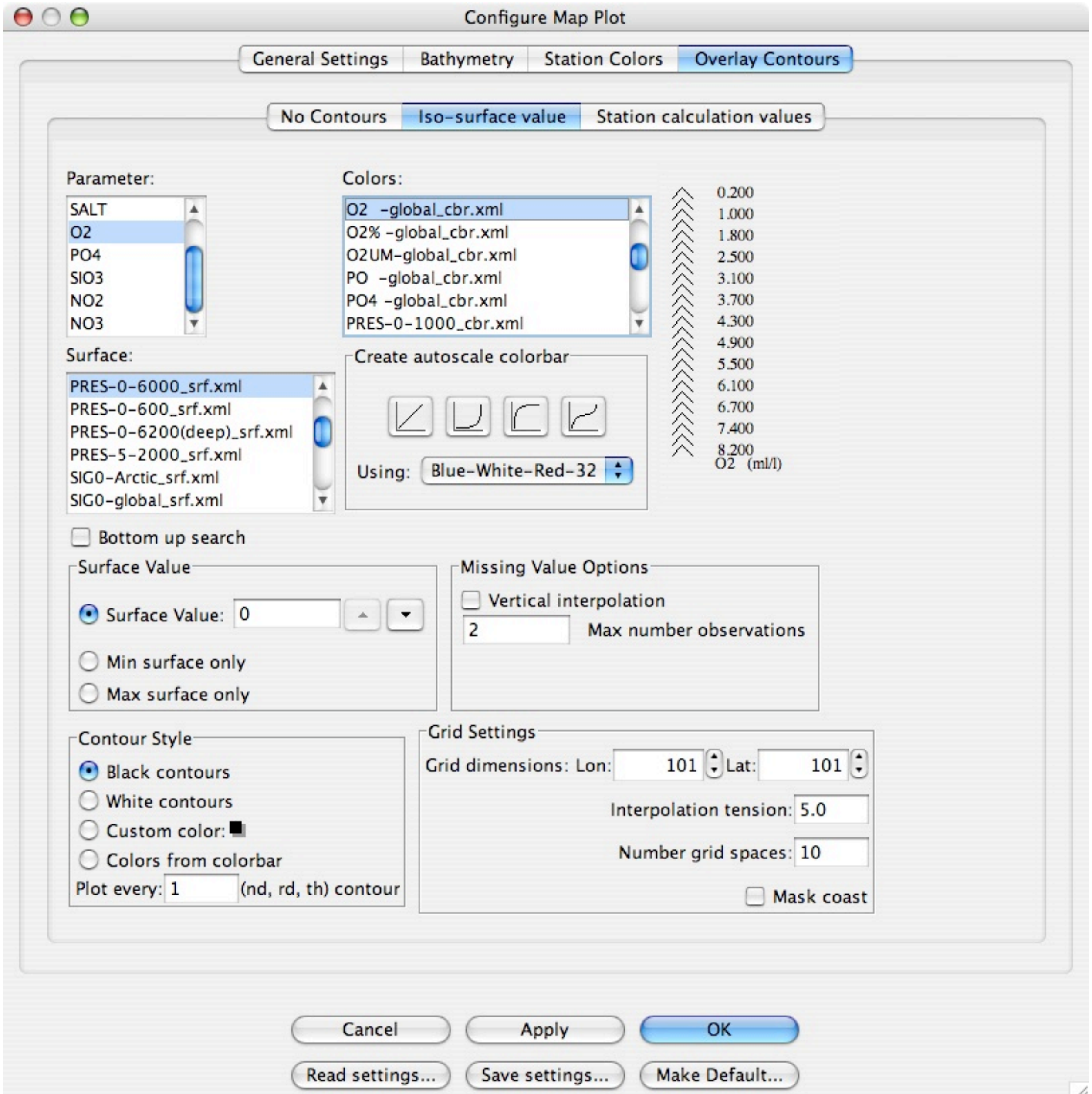


JOA provides two ways to add overlay contours onto a map:

> *Iso-surface value* computes a gridded field by interpolating the value of a observed (or calculated) parameter onto the value of a surface parameter. For example, salinity can be interpolated onto values of depth pressure.
>
> *Station calculation values* computes a gridded field using the values of a computed station variable (e.g., depth of mixed layer).

Iso-surface Values

Iso-surface/station value computes a gridded field of the parameter interpolated onto the selected **Iso-surface Values** surface - the set of standard levels chosen:



The options are similar to station symbol coloring with the addition of new options for contour style and grid settings. Note that selecting a colorbar or creating an autoscale colorbar is required as it defines the number of contours and contour line values plotted on the map.

Contour Style allows you to specify what color to use when drawing the overlay contour colors. The default color is black but you can also chose white contours, a custom color, or colors from the selected (or autoscale) colorbar. Contour labels are also drawn in the selected color.

Grid Settings control how unevenly spaced values in latitude-longitude space are assigned to a rectangular grid (binning) and how interpolation of missing values proceeds. JOA uses the zGrid algorithm developed in the late 1960’s as incorporated into SGT (see Appendix @todo):

> “This gridding routine applies a thin plate spline interpolation that converts irregularly spaced observations into a regular grid for contouring and display. The routine can yield a pure Laplacian solution (minimum curvature) or a pure spline solution, depending on the tension applied.” (http://web.science.oregonstate.edu/ecophysiology/Ocean%20Color%20Iron%20paper%202006%20Suppl.pdf)

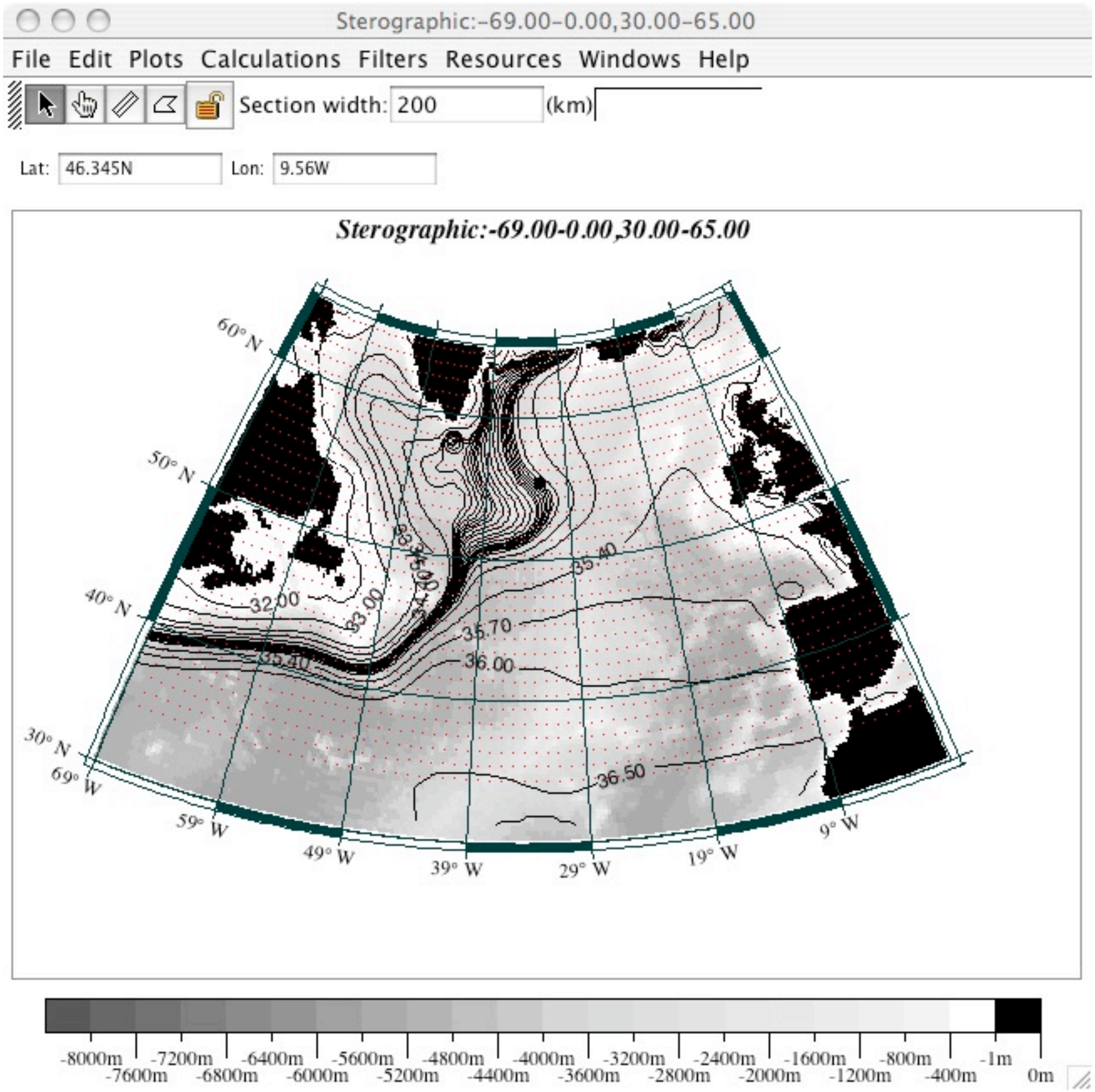
The interpolation tension can range from 0 to positive infinity. The value determines whether the solution is pure Laplacian or pure spline based.

When filling the grid, grid points more than “Number of spaces” grid spaces from the nearest known data point are set to undefined.

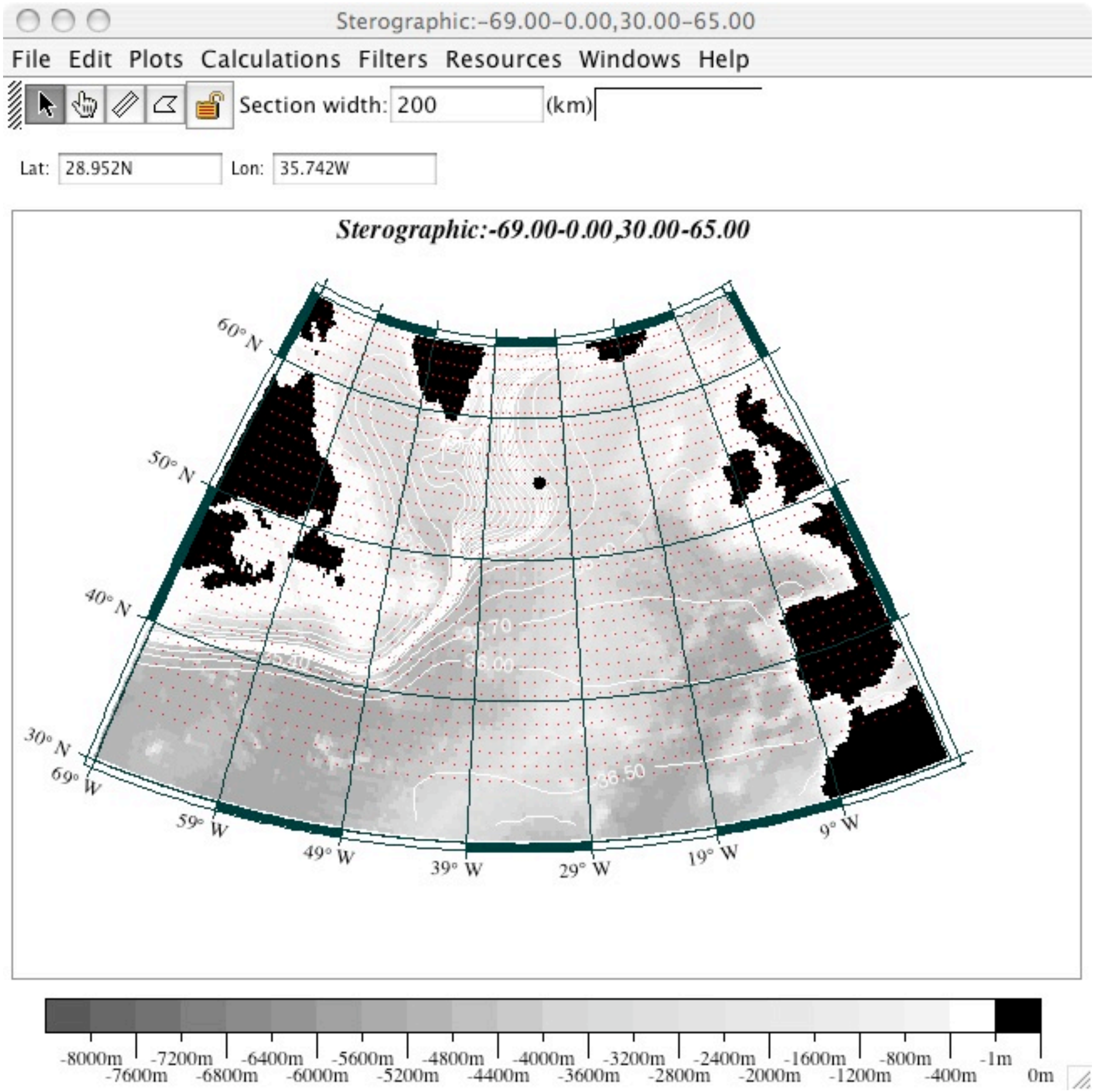
Mask Coast will compare grid points with the etopo20.nc bathymetry file and set grid points on land to a special value (NaN). The contouring algorithms will not attempt to assign values or contour through these points.

Note: The success and interpretability of zGrid-based overlay contours plots are highly dependent upon how sparse or evenly distributed the observed data are. Sparse clumpy data will produce a hard-to-interpolate contour plot whereas evenly distributed data will produce a “reasonable” contour plot.

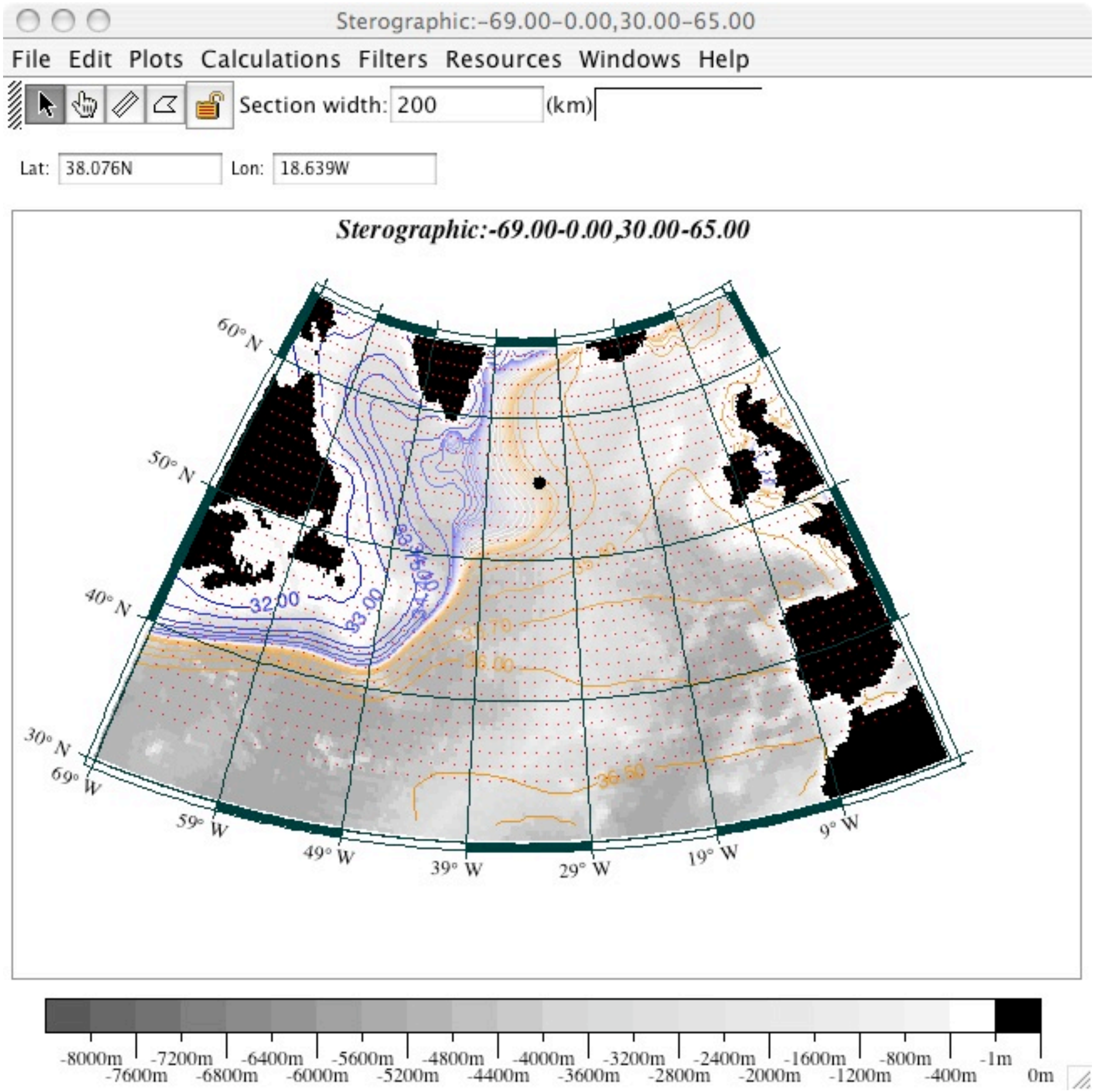
The examples shown here are based upon well-behaved pre-gridded data from World Ocean Atas 2005 climatology and show a contoured map of surface salinity with various contour color options. The size of the contour symbols has been reduced to better show the contours. Mask Coast is turned on.



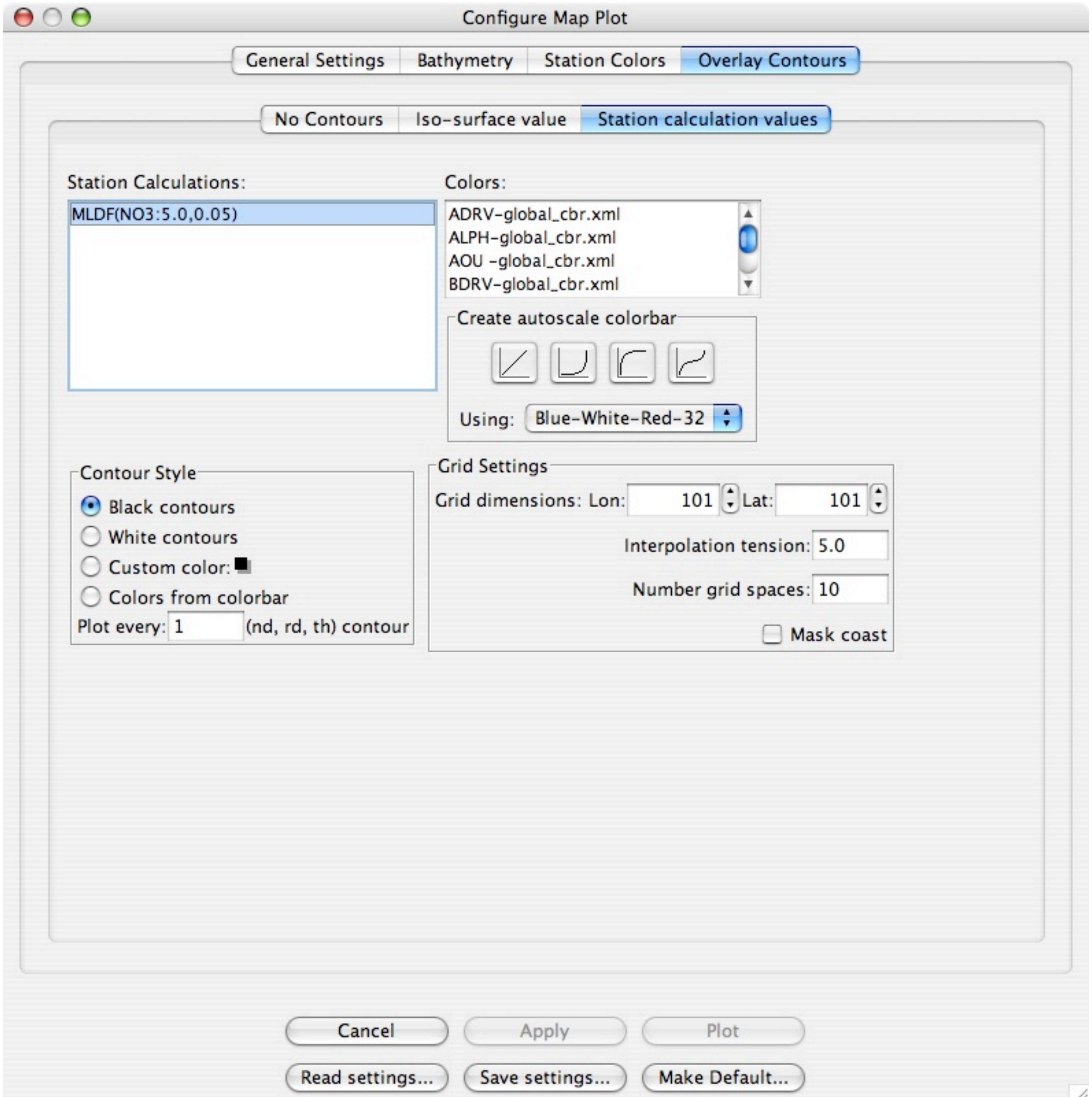
Black (default) contour scheme using default gridding settings.



Sterographic:-69.00-0.00,30.00-65.00
File Edit Plots Calculations Filters Resources Windows Help
Section width: 200 (km)
Lat: 28.952N Lon: 35.742W
Sterographic:-69.00-0.00,30.00-65.00
60° N
50° N
40° N
30° N
69° W
59° W
49° W
39° W
29° W
19° W
9° W
-8000m -7200m -6400m -5600m -4800m -4000m -3200m -2400m -1600m -800m -1m
-7600m -6800m -6000m -5200m -4400m -3600m -2800m -2000m -1200m -400m 0m



White contours and color contours from selected colorbar using default gridding settings.

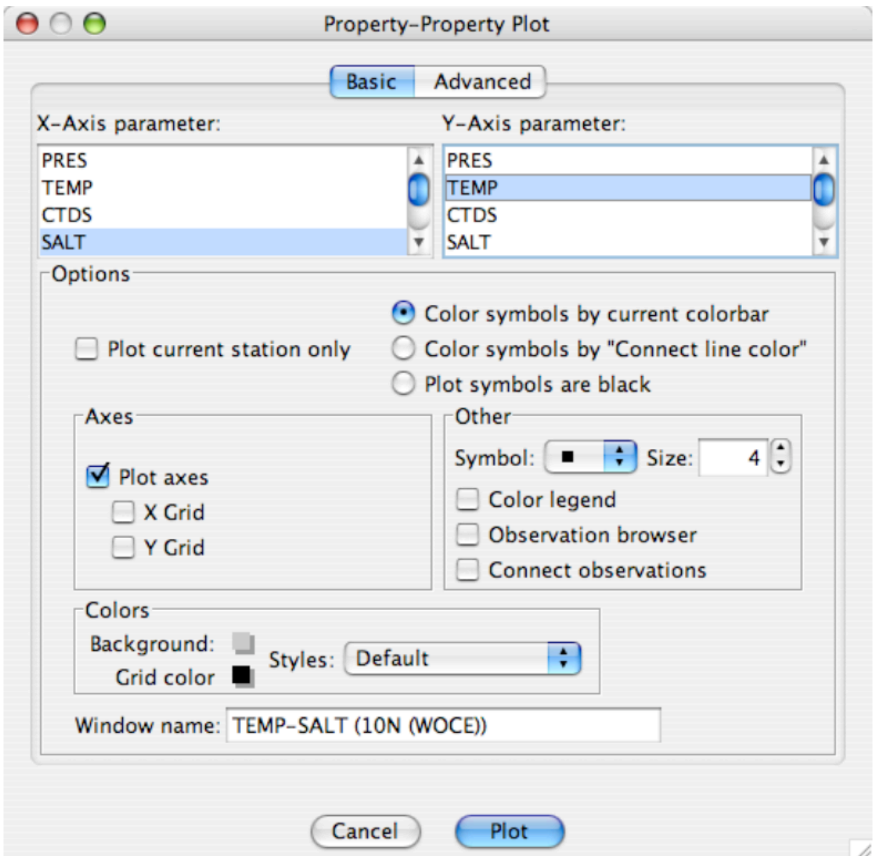


@todo

Property-Property Plots

Java OceanAtlas *Property-Property* plots [cmd/ctl-Y] plots are traditional x-y plots, but with a few added twists. In basic form any original or calculated parameter can be plotted against any other, with any third original or calculated variable for which a color/contour bar exists used to color the plotted points. Actually, on a single property-property plot up to seven different, independently scaled x axes can be plotted against a single y axis. Property-property plots can be browsed point by point, by clicking with the mouse or by using the arrow keys on the keyboard. As with nearly all Java OceanAtlas plots, a usable plot is generated by simply selecting an x parameter and a y parameter and clicking on the 'plot' button. But a substantial range of options is available, such as having a line drawn sequentially connecting the values from the currently-selected station, showing data from only one station at a time, and full plot customization and range control features. Default axis ranges are fit to the data range, but can be modified before or after the plot is drawn. Axes and labels are optional. Selected areas of property-property plots can be zoomed into new plot windows.

Selecting *Property-Property* displays the following dialog:



As noted above, selecting x and y-axis variables and clicking on *Plot* is all that is necessary.

> Names for calculated variables appear in parameter lists only after they have been calculated.

Options include:

Plot current station only, when selected, results in a plot showing only the station currently shown in the *Current Station* panel of the *Data Window*. The left-right arrow keys will browse between the stations.

Color symbols by current color bar will color the data points by the colors corresponding to the color bar in the Data Window. [That color bar can be changed using the *Contour Manager* under the *Resources* menu.]

Color symbols by "Connect line color" will color the plot symbols by the color of the connect line color. For a single x-axis plot, the connect line color defaults to black. For multiple x-axis plots, JOA assigns contrasting colors for each x variable. In either case, these colors are editable from the 'Advanced' panel of the property-property dialog.

Plot symbols are black overrides any coloring by colorbar or connect line color and just draws the symbols in black.

The *Axes* choices include the option to draw or not draw x and y axes, and the option to add a grid to the x and/or y axes.

Other choices include

- a pop-up menu of plot symbols,
- editable symbol size,
- the option to add a color legend (for the color bar),
- the option to add a display of data at the current browsing point (*Observation Browser*), and
- the option to add a line connecting in sequence the data points from the current station. The width of the connect-stations line can be set in the 'Preferences…' dialog under the Edit menu.

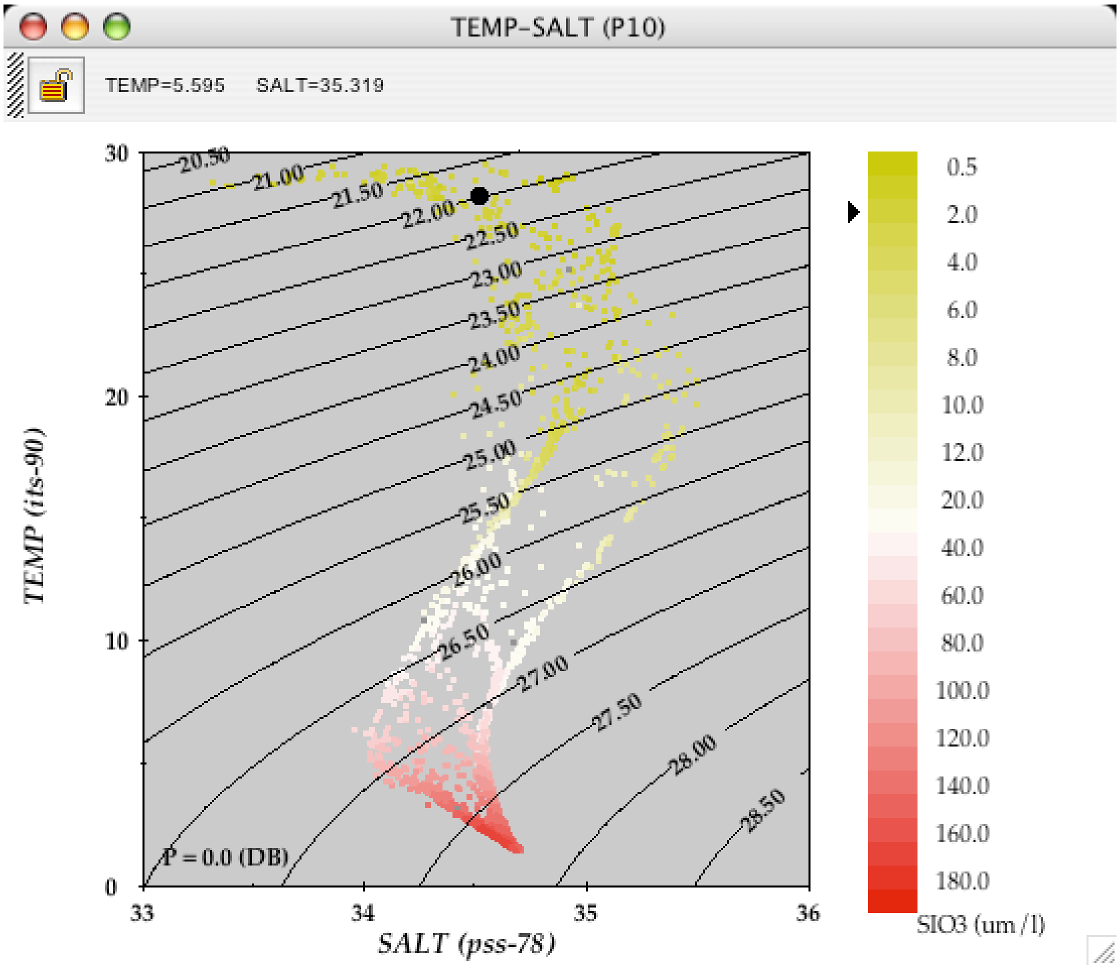
With larger symbol sizes it possible to more vividly 'paint' a property-property plot with the current color parameter shown in the Data Window.

The Colors of the background and grid can be changed (by clicking on the color swatches), and choices of overall plot color scheme are also available from a scrollable list.

Java OceanAtlas will automatically name a plot window. The *Window Name* field provides the option to enter a different name.

If the parameters selected for the plot are temperature and bottle or CTD salinity or potential temperature and bottle or CTD salinity, Java OceanAtlas allows the user to add isopycnals (lines of constant density) at any chosen reference pressure between 0 and 5000 decibars.

The potential temperature versus salinity plot below illustrates the appearance of the isopycnals. The data points are colored by the silicate concentrations.



The *Advanced* tabbed pane of the property-property plot dialog chiefly provides access to axis scaling options. Java OceanAtlas chooses default axis ranges based upon the full scale of each parameter plus an effort to make pleasing ranges. The ranges can often be improved upon via this interface, or by extracting a selection from the property-property plot.

Any existing property-property plot can be edited after it has been created by double-clicking (right-clicking) inside the plot. The axis ranges displayed in the dialog are those of the active plot, not just the original defaults, hence greatly facilitating modifications.

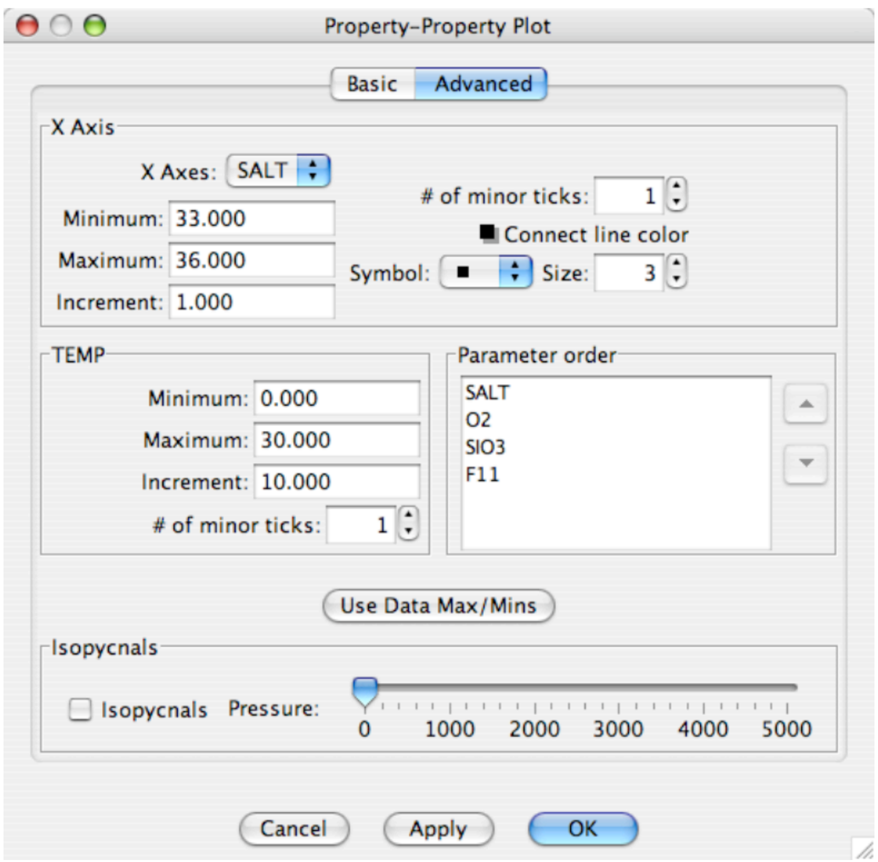
Property-property plots can be re-sized and distorted with the size box, zoomed to full screen or back with the zoom box, moved by dragging the title bar, and so forth. Most importantly, clicking and holding the mouse button in an active property-property plot window defines one corner of a 'selection rectangle' which is completed by dragging the mouse over the diagonal of the intended selection area, and releasing the mouse button to define the opposite corner of the selection rectangle. Upon mouse button release a new property-property plot will be drawn using a 'smart' version of the axis limits of the selection rectangle. The extracted plot can be modified using the same procedures as used for the original plot.

Occasionally Java OceanAtlas' 'smart' property-property plot axis choices interfere with obtaining the exact axis specifications desired by the user. In those cases you can either begin a new Java OceanAtlas property-property plot entering the axis specifications in the *Advanced* panel or edit the existing plot.

Property-property Plots with Multiple X-Axes

Java OceanAtlas offers the capability to select up to seven independently scaled x axes on a single property-property plot.

To invoke multiple x axes when setting up a property-property plot, hold down the control/cmd key while you select (click the mouse on) up to seven different measured or calculated parameters from the scrollable list of choices for x-axis parameters. At this point, except for the multiple selections for the x axes, the *Basic* pane of the property-property plot dialog looks for the same as a single axis x-y plot. But the *Advanced* tabbed pane now has additional features which are related to multi-x-axis property-property plots:



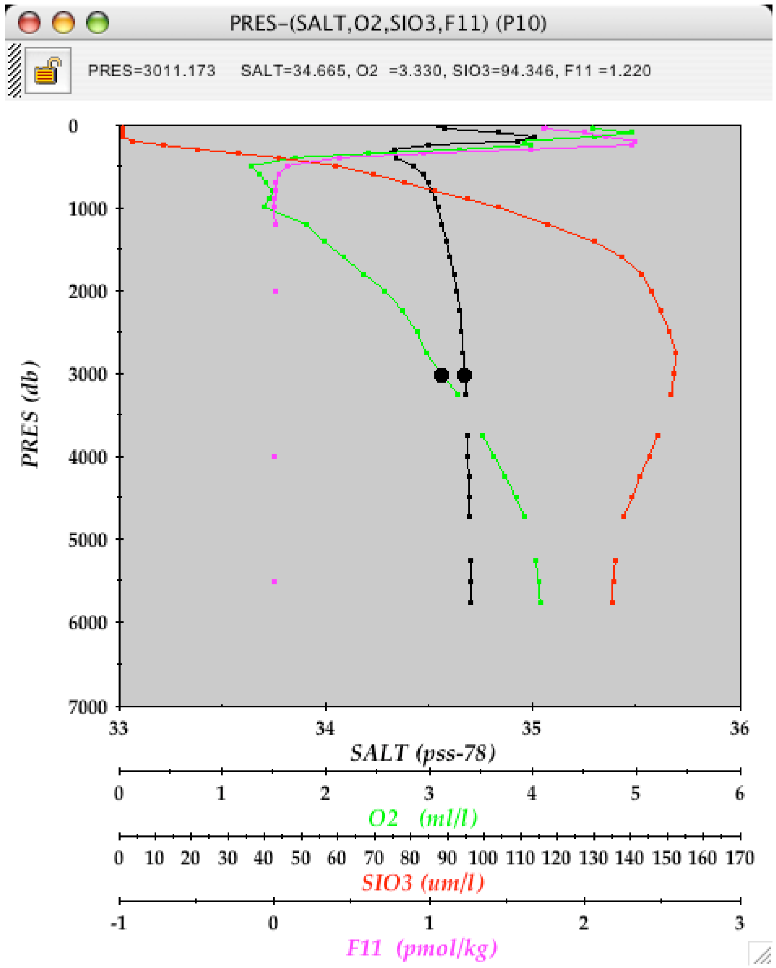
The x-axis pane of the *Advanced* pane includes a pop-up menu of all of the x axes parameters selected in the 'Basic' pane. As each one is selected, the x-axis section of the dialog shows the axis characteristics for the selected parameter. The axis ranges can be changed simply by selecting them and typing over them. The Tab key cycles between the text fields, including the minimum and maximum range, so that new choices can be entered. Java OceanAtlas assigns ‘smart’ axis ranges to property-property plots, but one often wishes to narrow or otherwise specify the axis ranges.

As with other Java OceanAtlas plots, the *Increment* is the increment of major tic marks, each of which will noted on the plot with a label. The *# of minor ticks* can be specified. Please remember that the 'increment' itself does not automatically default to a 'smart' value after the user has manually changed axis range. For example, if the default range (minimum and maximum) for a salinity axis were 33.000-36.000 and the default increment were 1.000, but then the user changed the axis range to 34.400-34.750, but did not change the increment, the increment would stay at 1.000, resulting in an absence of axis labels. (A logical choice for increment in that case would have been 0.05.) In other words, when an axis range is changed, be certain that the increment is appropriate to the range.

For each x-axis parameter the number of tic marks between each increment, the symbol, symbol size, and the color of the line connecting the observation symbols (*Connect line color*) can be adjusted via the provisions on the *Advanced* pane of the property-property plot dialog. The width of the line connecting the observations is set via *Preferences* under the Java OceanAtlas *Edit* menu.

Note: Using the advanced pane, you can set different symbols and sizes for each x axis. The symbol and symbol size settings in the *Basic* panel will change those settings for all the x axes.

Although any settings of the *Basic* panel of the property-property plot dialog are supported in multiple-x-axis mode, for more than one or two x axes it is suggested that users check/select the 'Plot current station only' and the *Connect observations* choices, and uncheck/deselect the *Color plot by current colorbar* choice in order to produce the most intelligible plots. An example of such a plot is shown below:

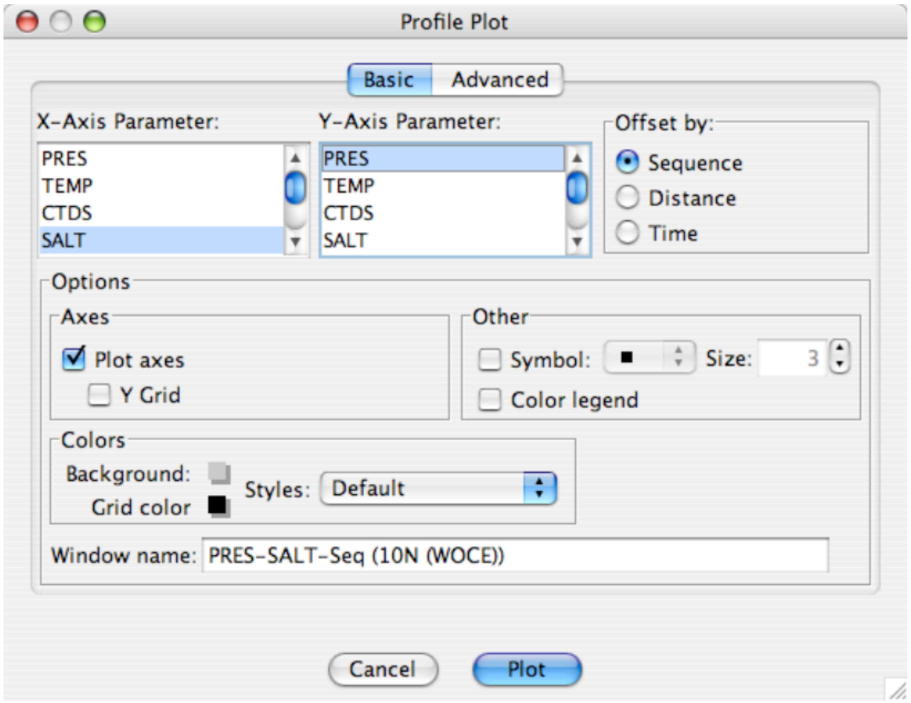


Browsing on a multiple-x-axis property-property plot is based on the top-most x axis in the list and the y axis. The *Parameter order* list in the *Advanced* pane of the property-property plot dialog permits adjustment of the order of appearance of multiple x axes. To move a parameter in the list, select it with the mouse, then use the up and down arrow buttons in the dialog to move its position in the list. In this manner the parameters can be ordered as one wishes.

Profile Plots

Java OceanAtlas profile plots - or 'waterfall' plots - are property-property profiles (connected dots, not interpolated curves) for every station in the data set, each station offset by a pixel separation keyed to sequence (order in the data file – the order cycled through in the Data Window), or by the distance between sequential stations. The profiles are colored by the parameter shown in the Data Window, and for browsing are linked to all other open plots, maps, and the Data Window. Profile plots can be zoomed, re-sized, and distorted Java OceanAtlas is 'smart' about choosing default axis limits. It is possible to chose a vertical-axis subset of the plot by selecting and extracting the desired part of the plot.

Selecting *Profile* [cmd/ctl-F] from the Java OceanAtlas *Plots* menu brings up the profile plot dialog:

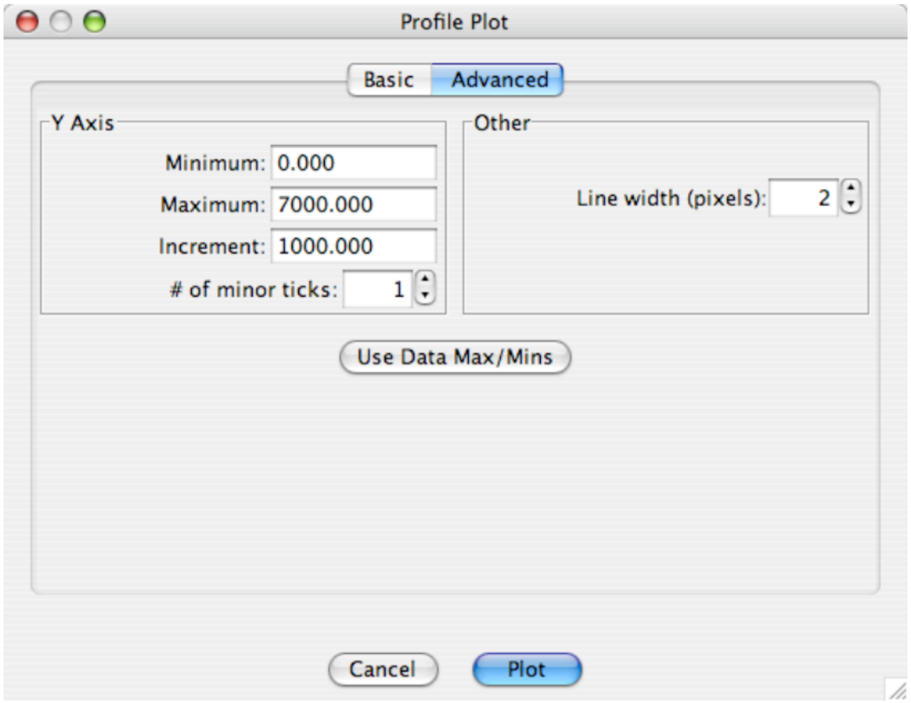


This is a tabbed dialog. The *Basic* panel includes:

Scrollable lists of all original and calculated parameters are provided for selection of the x axis and y axis parameters, and the successive offsets can be made simply sequential, i.e. N pixels offset between each profile, the offsets can be made proportional to the distance between each station, or the time recorded for the station. The y-axis display can be disabled, and a y grid can be added to the plot. A symbol can be added at each data point on the profile plot, and the symbol and its size can be selected. A color legend, for the color bar in the Data Window, can be added. The background and grid colors can be changed (by clicking on the color swatches) and the style of the plot can be changed to white or black background styles. Java OceanAtlas will suggest a window name, but the user can change it.

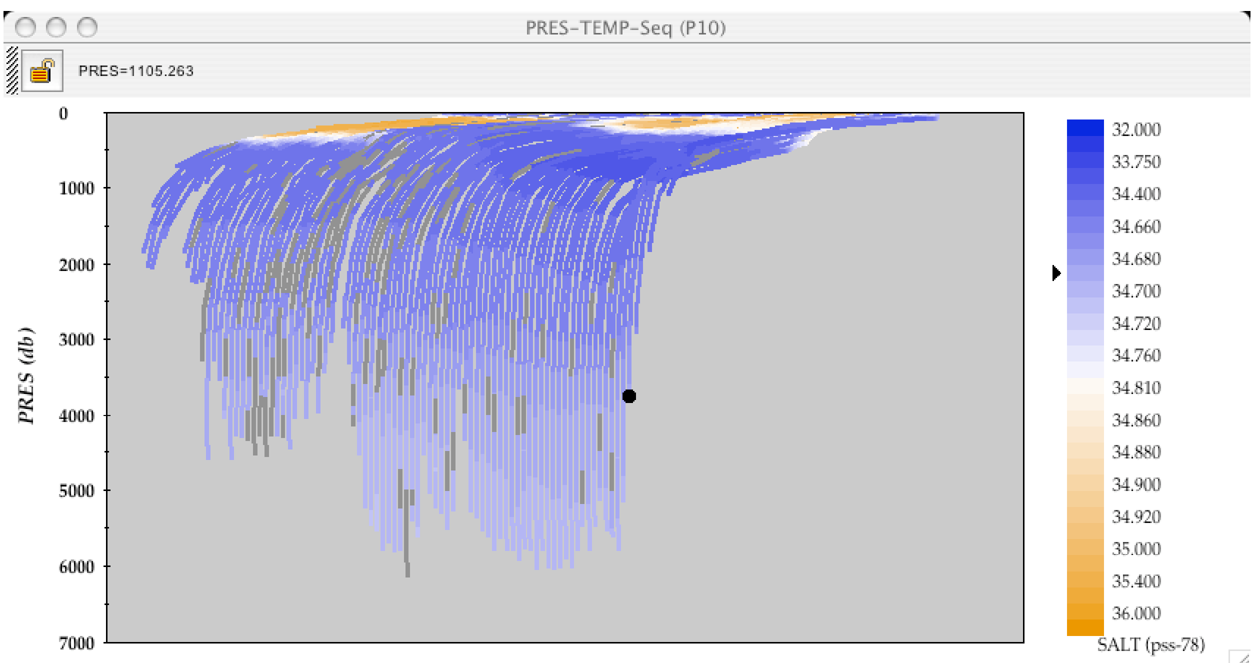
> Offset by distance sets pixel separations between stations keyed to the geographic distance (km) between stations. Offsets by distance can be confusing if the stations are not in positional sequence in the data file, because the offset is not keyed to a fixed starting point but to the previous station in the data file. However, Java OceanAtlas provides tools to sort and reorder data files in various ways. [The data sets included with the Atlas of Ocean Sections, and those available from, http://odf.ucsd.edu/joa) were pre-sorted from west to east and south to north.]

The *Advanced* pane of the profile plot dialog is shown below:



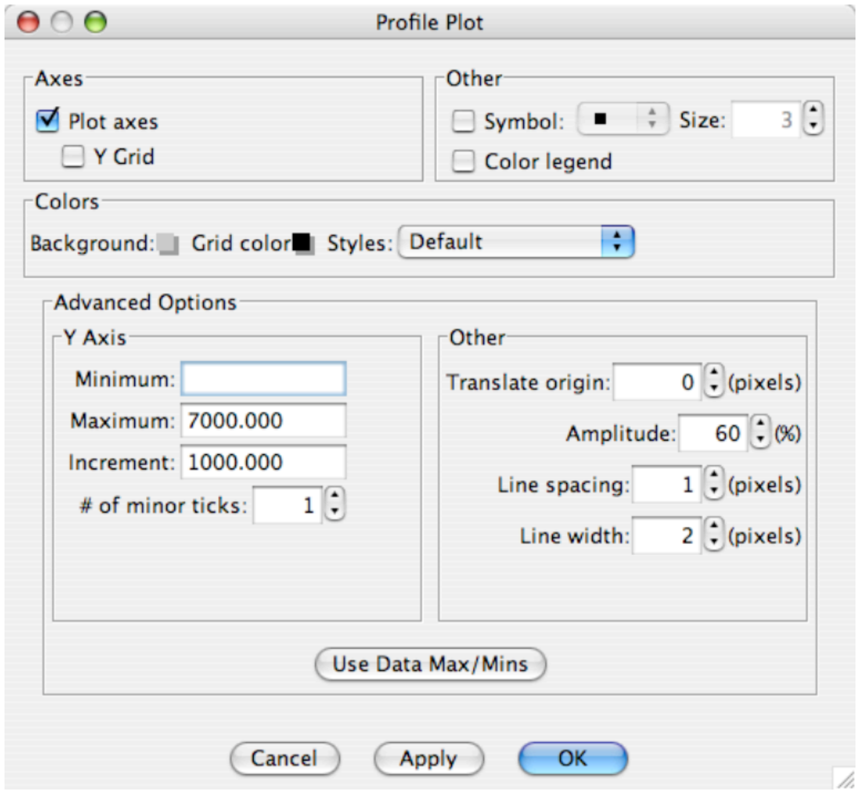
Here, options are presented to adjust the minimum, maximum, major tic increment, the number of minor tics between major tics, and the width of the line connecting observations in each profile.

A basic profile plot is shown below:



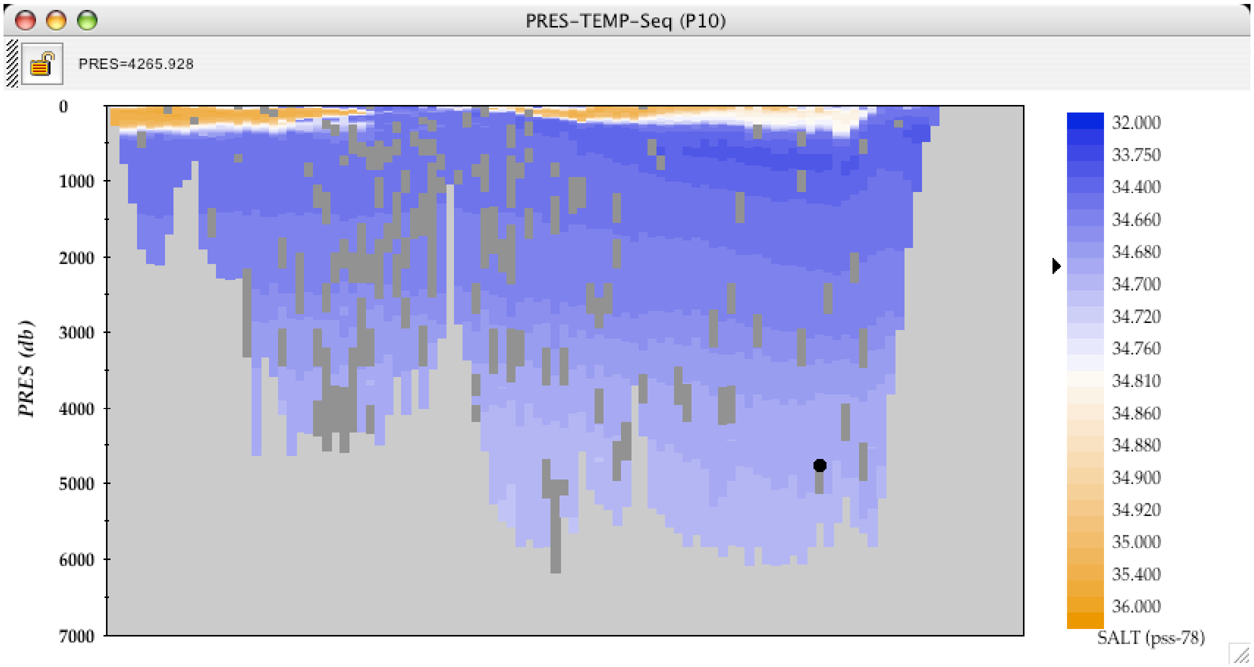
In this case successive temperature profiles are plotted sequentially against pressure, and colored by the 'SALT-global_cbr.xml' color bar. [Note: To make a better-looking plot for display purposes the above plot has already been through the 'options' step noted below. The options settings used to make this plot are shown in the figure below.]

Double-clicking (or right-clicking) on an active Profile plot with a single-button Mac OS mouse brings up the Profile Plot options dialog (see below) with the plot's current settings:



Note that this dialog combines some of the features of the *Basic* and *Advanced* panes of the Profile Plot dialog with some major new features: the capability to adjust the origin, relative amplitude, line spacing and line width of the profiles in the plot. In the case of the example plot, it has already been adjusted to reduce the amplitude (down to 35% from the original 100%), and the line spacing has been increased to 4 pixels. Sometimes profile plots can start too far to the right. The 'spinner' control, 'Translate origin', moves the origin of the plot to the left or right on the plot as needed.

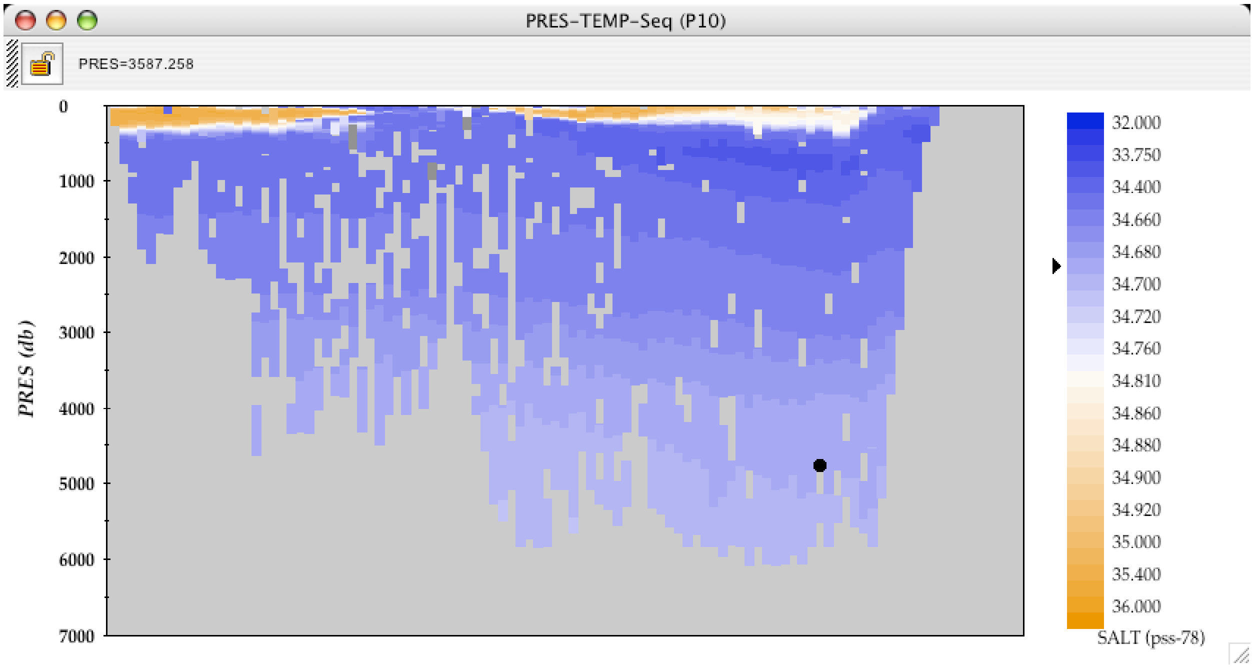
A novel type of profile plot is possible by reducing the amplitude to zero, increasing the line spacing, and increasing the line width to equal or nearly equal the line spacing. The resulting "zero-amplitude" profile plot is shown below:



These zero-amplitude profile plots are quickly-drawn and nice looking alternative to contour plots. Recall that in most oceanographic data temperature is well and reliably sampled, and that in profile plots the coloring is by any third parameter for which there exists a color bar, the Java OceanAtlas zero-amplitude profile plot may be useful for data proofing, because profile plots simply show the data as they are found in the data file. (Because there are almost always temperature data, zero-amplitude profile plots are best made using TEMP as the x axis)

Note the dark gray patches in the plot above. These are locations where values of the colored parameter - SALT, or bottle salinity - were missing. (See the User Guide section on Preferences for how to set the missing value color.)

Note the patches of color that seem to be out of place, such as some of the patches in the deep water that are unusually 'bluer' (fresher) than their neighbors, or the two small patches between 500 and 1000 meters just left of center in the plot which show samples unusually more 'orange' (saltier) than their neighbors. In some cases the ocean does throw surprises at us, but in these cases from the WOCE Hydrographic Program section P10 in the western Pacific the data originators already recognized sampling or analytic problems with these samples and had coded them 'bad' or 'questionable'. If an observation filter is added (see User Guide section on observation filters) to show only those data for which the salinity quality code ('SALT QC') is equal to '2' (good), the plot changes to this:



You can see that clever use of plots and filters can help you explore nearly any facet of the data. In this case we can see that for reasons unknown from the data file itself (but which could be investigated with the documentation on the WOCE V3 DVD or from <http://whpo.ucsd.edu>), the data originators experienced an unusually large number of problems with their salinity analyses, resulting in many 'bad' or 'questionable' observations.

Other notes regarding profile plots:

The browsing cursor may seem to jump up and down somewhat using the left/right arrow keys on Profile plots with vertical axes other than pressure (and usually density), because the browsing is cycling through the data in the vertical order read in from the pressure-sequenced data file.

Roughness in Profile plot coloring - especially noticeable in solid-color (i.e. zero-amplitude) plots - is caused partly by the fact that Java OceanAtlas does a minimum of calculations to create these plots (which makes them fast to draw). The application carries out a vertical linear interpolation with reference only to the current profile to determine where to draw color transitions. The better the vertical resolution of the data, the more nearly accurate the coloring.

Dragging a selection in a profile plot draws a line rather than a rectangle as in property-property plots. A selection in a profile plot, defines the vertical axis limits of a new profile plot or can be used to zoom the current plot (when holding the alt/option key down while making the selection). The extracted plot can be modified using the same procedures as used for the original plot.

Contour Plots

Java OceanAtlas provides contoured section data using a simple contouring algorithm to make a traditional contoured plot.

Java OceanAtlas contour plots are vertical section plots with contours/colors applied to the entire active data set, for example a contoured vertical section of salinity versus pressure. It is not necessary, however, for the z axis on a contour plot to be restricted to pressure. Any observed or calculated parameter can be used as the vertical axis (but an 'interpolation surfaces' must exist or be created for it; see Surface Manager) and any observed or calculated parameter can be plotted on it. For example it is possible to plot a contoured vertical section of phosphate with nitrate as the vertical axis. The horizontal axis on a Java OceanAtlas contour plot is either spatially related, either the horizontal spacing is proportional to the distance between stations (typical), proportional to latitude or longitude, or a fixed sequential offset can be used or temporally related. As with all Java OceanAtlas plots, contour plots share a linked data cursor with other JOA plots, and all contour plots, color/contour bars, and standard surfaces used in preparing contour plots can be customized in numerous ways.

x and y interpolations from the observed data are required before Java OceanAtlas can generate contoured parameter plots. The interpolations are carried out on surfaces (standard values) of the y axis. The methodology and, perhaps, philosophy, underlying interpolations is an important issue. We have deliberately chosen a simple approach. Our goal was to provide useful, essentially correct contour plots which show all features of the data while adding no false features. Hence we chose simple linear interpolation between closest points on the y axis and adjacent profiles (stations) on the x axis (filled contours are actually produced using a bi-linear interpolation algorithm). An alternative approach, and one used often, is some variant of objective analysis, which can produce appealing plots but may miss subscale features and, in some cases, result in a grid diverging significantly from some of the underlying data.

> Technical note on interpolation: When JOA interpolates a parameter onto a grid; the parameter values are assigned to a grid point on a station-by-station basis by linear interpolation. This defines the y grid of a contour plot. The x grid is done somewhat differently. JOA computes the spacing of the grid in the x direction by applying the chosen offset, i.e., distance, latitude, longitude, or sequence. Now JOA has fully defined the grid - the values of the contour parameter interpolated onto the surface levels at each station and the x locations of the stations. Contouring is performed on rectangular regions of the interpolation grid defined by two interpolation levels and two stations. For example the box defined by levels 0 and 1 of the interpolation grid (e.g., pressure values of 0 and 100) and between stations 1 and 2. Contouring proceeds in one of two ways depending upon whether contour lines are chosen or filled contours are chosen:
>
> > Contour lines: JOA will find where contours enter and leave the box by using linear interpolation along the sides of the box and across the corners.

Filled Contours: JOA uses a bi-linear interpolation algorithm to assign the values of individual pixels found in the box.

When one produces a contour plot with both lines and filled contours, there may be some small disagreement because of the different approaches but in general the two techniques produce similar results.

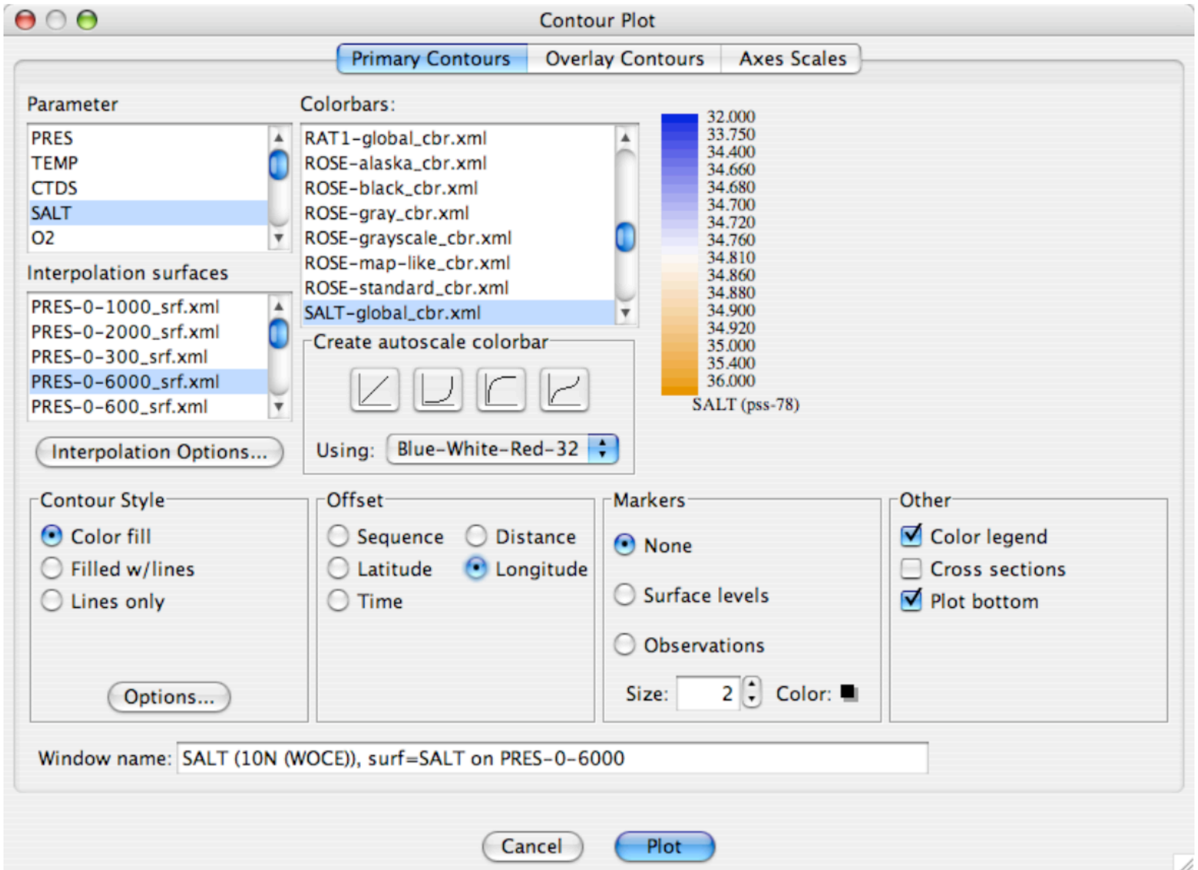
In default mode Java OceanAtlas takes the first instance of a standard level it runs into from the first sample at a station (usually shallowest) to the last (usually deepest). This may cause problems at stations where values of some of the standard levels appear more than once (for example in subpolar regions calculating on standard temperature levels). Hence we also provide in Java OceanAtlas a choice to interpolate 'from the bottom up', which provides the alternative of finding the 'last instance' (usually deepest) instance of a standard surface. In any event, be careful interpreting plots where the standard levels are anything other than monotonic functions. (An example of a monotonically-varying parameter is pressure.) Even density can be multivalued when expressed as density anomaly (sigma-notation), for example as with sigma-0 in many Atlantic data sets and some high latitude ones.

If the possibility of multi-valued surfaces (temperature surfaces, for example) is worrisome, one check when carrying out interpolations is to prepare an interpolation of *pressure* on that variable and plot it, and then examining the plot, or better yet the browsing windows, for unusual features in interpolation.

Some oceanographers find machine contouring the antithesis of the artistry in atlases, even though artistic judgment in hand-contouring can be as deceptive as the computer's approach. Java OceanAtlas uses elementary gridding and contouring algorithms in order to bring out all the features (and problems) of the data.

Java OceanAtlas draws its plot contours (or color divisions on a solid-color contour section) on a 2-dimensional grid of the stations and standard levels. It places a defined contour value by linear interpolation using the data from the two closest standard levels at each station. Java OceanAtlas looks for all instances of a contour value 'fit' at each station from the smallest-valued standard level to the highest-valued standard level. The application then has the interesting task of joining these values to those from adjacent stations. The application does this in a simplistic way, at first not looking 'far' away to join lines. However, interpolation options are available for 'far-field interpolating, which can assist the Java OceanAtlas interpolator to fill in missing values from the interpolation before contouring, in effect looking past a user-provided number of missing-data stations (horizontal) and/or observations (vertical) for contour data. This is useful to fill in contours where data are missing and/or sparse. At the time of this writing, there is no smoothing or 'objective mapping' features.

The Contour Plot dialog is reached by scrolling down the *Plots* menu and selecting *Contour* or via the keyboard shortcut cmd/ctl-1:



This is a complex, tabbed dialog, because many options are available, but, as with other Java OceanAtlas plots, a useful plot can be made by selecting only a bare minimum of options from the *Basic* dialog pane:

Select a Parameter from the scrollable list at the upper left;

Select an Interpolation Surface (typically PRES-0-6000_srf.xml for deep ocean sections) from the scrollable list underneath;

If the *Parameter* selected is already represented by a color bar in the scrollable list of color bars in the top/right section of the dialog, the first matching color bar will be automatically selected. If this is not satisfactory a different color bar can be chosen from the list or you can create a custom autoscaled colorbar; and

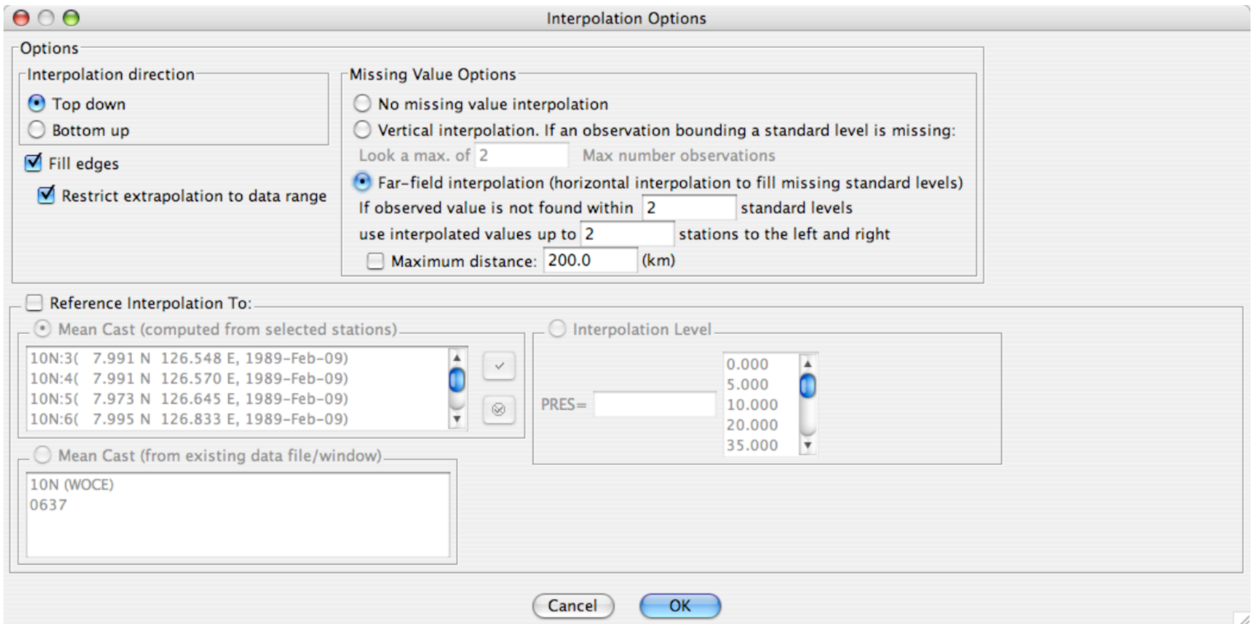
Click the *Plot* button at the bottom of the dialog. A basic contour plot will be drawn.

Basic Contour Plot Options

The *Parameter* list in the upper left contains all original and calculated parameters. Any one parameter can be selected by clicking on its name.

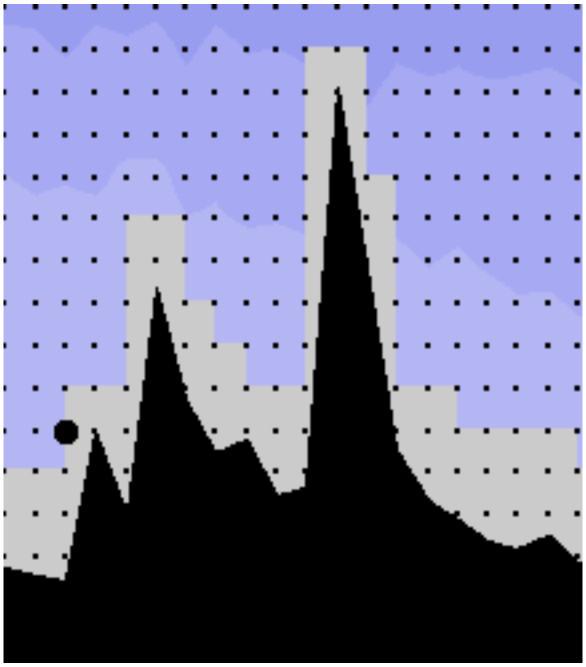
The *Interpolations* list contains all standard level files available to Java OceanAtlas. (More of these standard level files can be made from the *Surface Manager* found under the *Resources* menu.) Any one set of interpolation surfaces can be selected by clicking on its name.

Clicking on the *Interpolation Options* button brings up the Interpolation Options dialog:

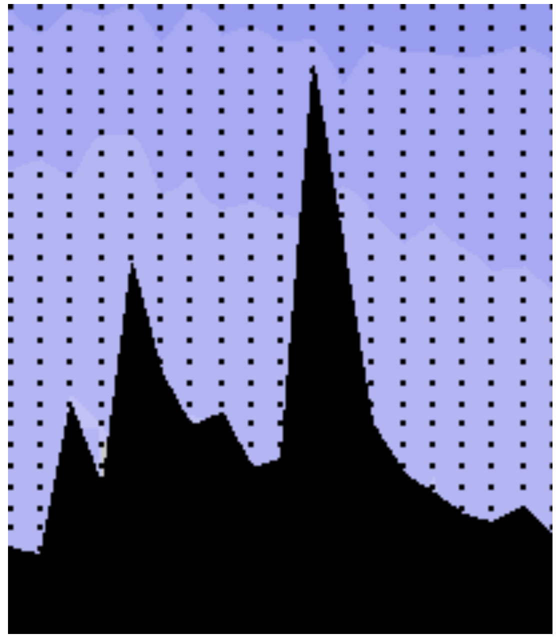


With the default *Top down* interpolation direction choice, Java OceanAtlas searches the data for the first match to the current standard surface from the shallowest observation to the deepest observation, and vice versa for the *Bottom up* choice. When one is plotting a contoured section of a parameter onto pressure (i.e., pressure is the y axis), this is not an issue. But imagine plotting silicate onto standard surfaces of salinity in an ocean region where there is a mid-depth salinity extremum. In that case a different result would likely be obtained, depending on this interpolation direction.

The *Fill edges* options causes JOA to attempt to fill the triangular regions formed by the deepest observation of subsequent profiles and the bottom of the ocean as reported in the station's metadata. It does this by computing a horizontal gradient in the contour parameter and extrapolating it back to the bottom of the ocean. By default extrapolations are limited to the actual range of the contour parameter. These two figures compare a contour plot of salinity on pressure without and with the fill edges option selected.



No Fill Edges



Fill Edges

The *Missing Value Options* permit Java OceanAtlas to look in the horizontal and/or vertical for data to use in preparing the interpolation for a contour plot when there are missing data. Sometimes one prefers to visualize all the 'holes' in the data, but with sparse data, for example dissolved helium data that are not present at every bottle or every station in a hydrographic file, it is sometimes useful to use JOA's missing value options to fill in the holes in the interpolation grid. Note: it is also possible to achieve a useful contour plot by using a station filter to remove all stations totally missing the contour parameter. JOA provides two basic techniques (and associated options) for dealing with missing data:

Vertical interpolation: JOA looks for observations that bound a standard level to interpolate to that level. If one of the bounding observations in missing, this option directs JOA to look n number of observations below or above (depending upon the interpolation direction) in the profile for a non-missing value. A *Maximum number of observations* option is provided to limit the number of observations away the interpolator will look for a non-missing value. This is most useful for bottle profiles where the vertical spacing of bottles increases with depth.

Far-field or horizontal interpolation: If a bounding observation in the current cast is missing, this option directs JOA to use up to n surrounding casts (before and after) to interpolate a value for the missing value. It does this by first calculating the horizontal gradient of the contour parameter near the depth of the missing observation and secondly using the gradient to create "virtual" observations of the contour parameter at the current profile. The closest "virtual" observation to the missing value is then used to interpolate the contour parameter to the surface level. A 'Maximum distance' option is provided so that in cases of uneven station distribution along a track, Java OceanAtlas can be restricted from looking unrealistically far away for data to complete an interpolation.

Referenced Interpolations

These options allow you reference an interpolation to the value of standard level or to a mean cast. The mean cast option is how you create a contour plot of residuals in JOA. Options include:

Interpolation level provides the option, to subtract the value of the parameter being interpolated at one standard level from all the other values interpolated at the other standard levels at that station. For example, an interpolation In other words, if in the field to the right of *PRES*= the number '1000' is entered (or if it is selected from the scrollable list of pressures), when the interpolation is carried out, if the calculated geopotential anomaly at 1000 decibars is -1.234 dynamic meters, this value will be subtracted from every interpolated value at that station. Hence, on the contour plot, the surface value will be 1.234 and the value at 1000 decibars will be 0.000.

Mean Cast (computed from selected stations) provides the option to calculate a mean profile onto standard levels for the parameter being contoured. By defining a mean cast, you are directing JOA to produce an anomaly or residual contour plot of the chosen parameter. For a residual plot, the value of the parameter from the mean profile at each standard level is subtracted from the value of the parameter at each station at that standard level. A scrollable list of stations is presented which can be used to select a subset of stations used to define the mean cast. Stations can be selected by clicking, shift-clicking (for contiguous groups of stations) or by cmd/ctl-clicking (for discontiguous stations from the list), or all the stations can be selected by clicking the button with the check symbol, or all can be deselected by clicking the 'uncheck' button underneath it. For example, if one were preparing a contour plot of silicate on pressure, and this option were chosen with all stations selected, the resulting plot would be a silicate anomaly plot on pressure. Any other vertical axis which has a defined set of standard levels is supported by this feature. For example, if one creates a set of standard nitrate surfaces via the Surface Manager, and then sets up a contour plot of phosphate onto nitrate, the resulting anomaly plot will show where the relation of nitrate to phosphate differs from the mean, which can be useful for locating zones of denitrification.

Mean Cast (from existing data file/window) provides the option of creating a residual plot using the first station of any open data window as the reference station. Note: Using the Section Manager you can create a data window that has just a single mean profile. If you have created custom mean casts and saved them as disk file, you will have to open them before they will be visible in the mean casts list.

The *Color Bars* area in the upper right of the dialog offers several choices and features:

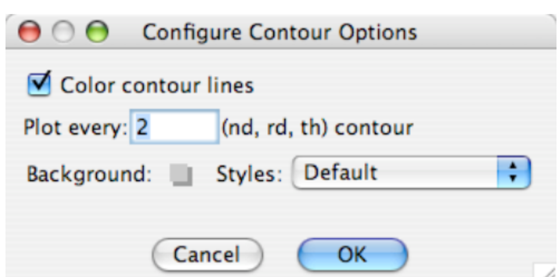
If one or more color bars for the selected parameter already exist, Java OceanAtlas will usually highlight the first matching choice on the list. But ***any*** color bar can be selected - and its numerical ranges and colors will be used on the contour plot - whether the color bar matches the parameter being contoured or not.

An alternative to selecting a color bar is to have Java OceanAtlas autoscale the contours (or color divisions). This is the only option when no color/contour bar exists for the parameter being contoured. Four choices of 'shape' of contour interval over the data range are provided. These distribute the contours linearly through the contoured parameter's data value range, or bunch the contours at the low or high ends or in the middle of the range. The upper and lower parameter value limits of the autoscale range always cover the full range of the parameter being contoured. An autoscaled color bar can be colored by any of the choices in the menu appearing to the immediate right of 'Using:'. These color choices - 16 or 32 colors along a blue-white-red (or red-white-blue) or a rainbow or inverse rainbow color path - are supplied by the application and cannot be added to or modified by the user.

A preview of the selected or autoscaled color bar will appear in the panel at the right of the *Colorbars* area. This space is blank if there is no selected color/contour bar.

Double-clicking on the preview color bar preview, opens the Color Bar Editor, which is described elsewhere.

The *Style* options allow you to select the basic look of a contour plot; solid color filled contours (the default), color fill with overlay contour lines, or just contour lines.
Clicking the *Contour Options* button in the *Style* options section of the dialog brings up the *Configure Contour Options* dialog:



Choices are provided to change from the default coloring of contour lines by the colors of the chosen color/contour bar to black. (This is useful for preparing a contour plot for black & white publication.)

An option is provided to plot only every nth contour on a plot. This is especially useful when the *Color fill w/contours* option is chosen, where the added contour lines make it easier to see subtle features in the contours.

The contour plot background color can be changed by clicking on the color square, which brings up the color picker, or the background color can be changed to white or black by selecting from the menu of choices to the right of *Styles*:.

The 'Offset' options makes the station-to-station separation on the horizontal axis of the contour plot either equal, when the offset by sequence option is chosen, or proportional to either distance, latitude, or longitude between stations.

The *Markers* options provides options for no markers (default) or for either of two kinds of markers: (1) at the levels of the standard surfaces used in the original interpolation, or (2) at the levels of the observed data (*Observations*) at each station - i.e. on a traditional hydrographic section 'plotting the bottles'. The size and color of the markers can be also be customized.

The *Other* option choices add a *Color Legend* to the plot (the selected or autoscaled color/contour bar). This is essential if one wants to know which contour is which (hence the default). The other option is to add a browsing feature (*Show cross sections*). These optional features take up quite a bit of space in the contour plot window, hence reducing the size of the area available for actual contoured data. Hence either, both, or neither of these options may be checked.

Java OceanAtlas will automatically suggest a name for the resulting plot window. The *Window Name* field provides the option to enter a name of your choosing.

Overlay Contour Plot Options

Overlay contour plots are described in Appendix A: Advanced features of Java OceanAtlas 5.0.

Advanced Contour Plot Options

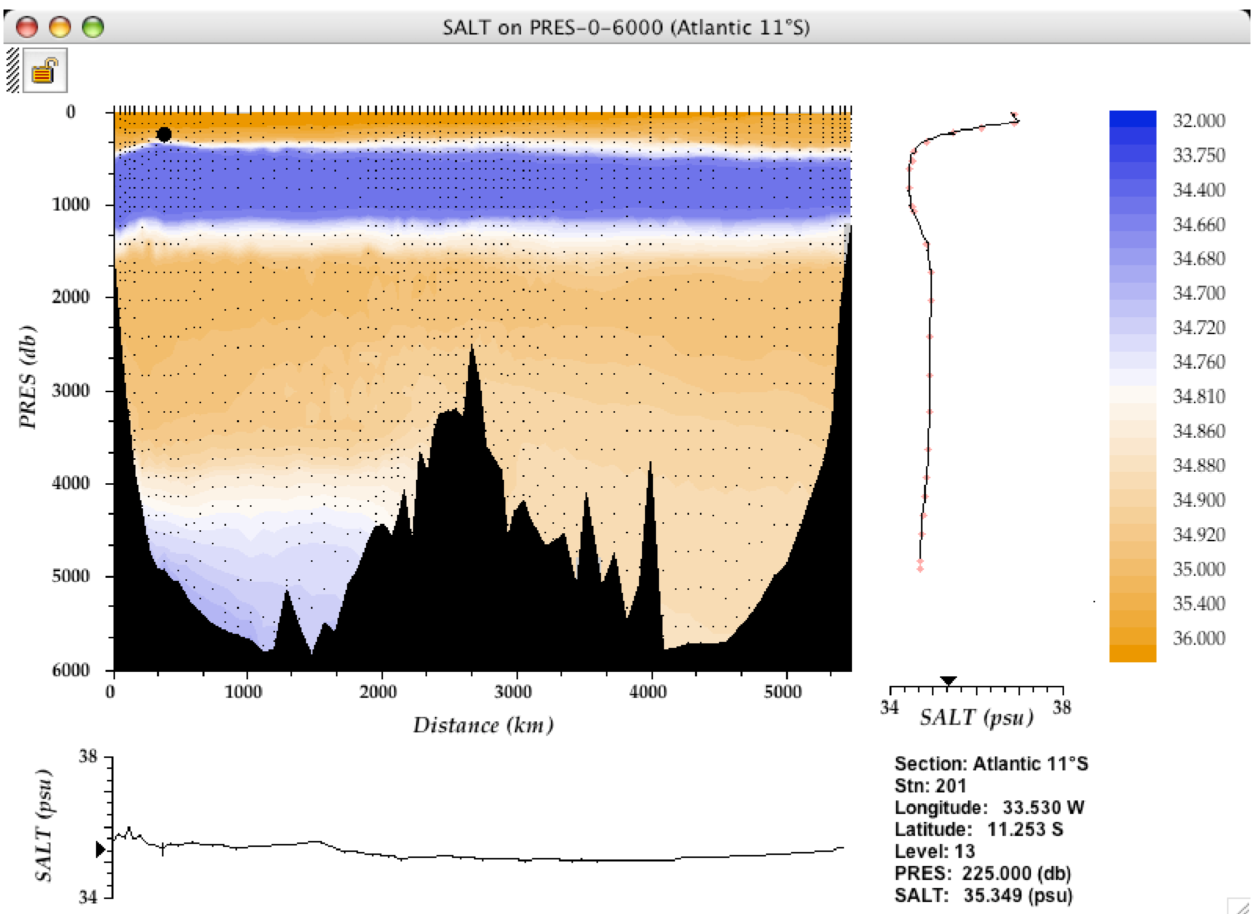
The top two text fields in the *Y Axis Range* options initially show the minimum and maximum y-axis data values in the chosen set of standard interpolation surfaces, i.e. for PRES-0-6000_srf.xml the minimum is 0 and the maximum is 6000. The 'interval' text field is Java OceanAtlas' best attempt at a logical interval between major tics on the Y-axis. To plot only a portion of the range, select the text as necessary and type in the new axis limits to impose on the plot. If this is done, the user should also type in a new interval as/if appropriate. The number of minor tics between major tics can also be specified.

The *X Axis Scale* is usually set to be autoscaled (the default), but the x-axis scale can be made a specified ratio of kilometers per pixel in order to make plots with matching x-axis scales. Note that in some instances with this option not all of the resulting x axis may be shown in the resulting plot, in which case it may be necessary to increase the width of the plot. The number of minor tics between major tics can also be specified.

Clicking on the *Plot* button at the bottom of the Contour Plot dialog will draw the plot as specified. Clicking on the *Done* button acts as a cancel feature, with no action taken.

Below is a solid-color Java OceanAtlas contour plot of salinity versus pressure, with *Observations* markers enabled, and with both the *Color Legend* and *Show cross sections* enabled:

> Java OceanAtlas scales all contour plots to fit into the same size window on the monitor, regardless of the number of stations, y-axis range, or length of the section. However, the contour plot window can be distorted, resized, or zoomed via the size and zoom boxes after the plot is drawn, to better match the aspect ratio 'expected' for the plot.



There are five sections to the contour plot window when both the *Color Legend* and *Show cross sections* options are enabled:

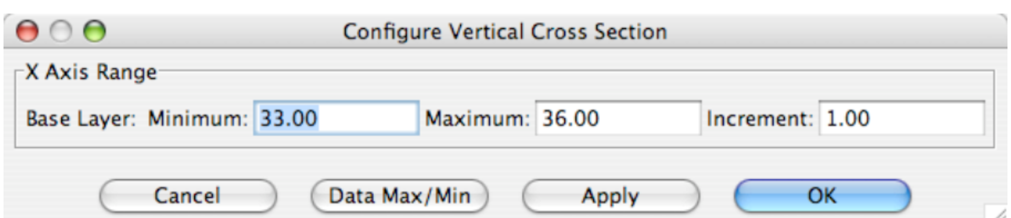
1. The contoured plot begins on the left with the first station in the data set and ends with the last, with the relative station-to-station separation either equal (when the offset by sequence option is chosen) or proportional to the distance, latitude, or longitude between stations. The contours are those corresponding to each transition in the color bar. When contour plots are drawn with pressure as the vertical axis, Java OceanAtlas looks at the data headers for bottom depth, roughly translates it to pressure, and plots it, connecting the lines in a form of bottom representation.

2. The color legend is the selected or autoscaled color bar. The legend is handy for identifying contours on the sections. Custom color schemes with relatively sharply contrasting adjacent colors can make identifying contours easier, whereas the smooth atlas-like color transitions supplied with Power OceanAtlas can be more challenging to identify.

When *Show cross sections* is selected the data cursor correspond to the information shown in the next three segments sections:

3. The right hand section shows a vertical profile of the interpolated data at the station identified by the data cursor. In other words, in the example above this is a profile of salinity. The observed data are shown on the profile as small red symbols so it's easy to compare the interpolated values with the observed values.

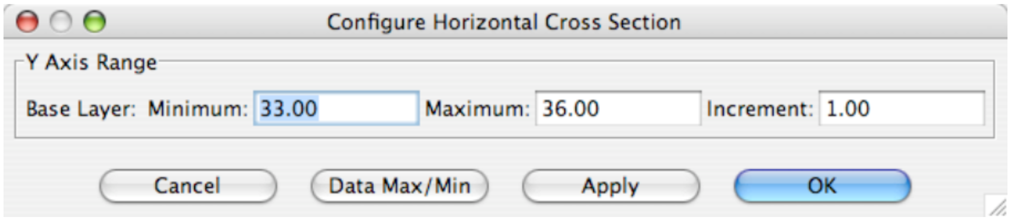
Double-clicking (or right-clicking) on the vertical profile section opens the Configure Vertical Cross Section dialog (shown below) to adjust the *X Axis Range* (useful to examine a particular layer in detail), and major tic increment. The 'Apply' feature allows the plot to be redrawn without closing the dialog, making it easier to experiment with various x-axis ranges. The *Data Max/Min* button sets the x-axis range to the full range of the data in the plot rather than the "pretty" range that JOA computes.



4. The lower left section shows a cross-section profile of the interpolated data at the standard level identified by the browsing cursor. In other words, in the above example this is a profile of salinity along one level of the interpolation grid.

Double-clicking (or right-clicking) on the lower left plot segment brings up the Configure Horizontal Cross Section dialog (shown below) to adjust the *Y Axis*

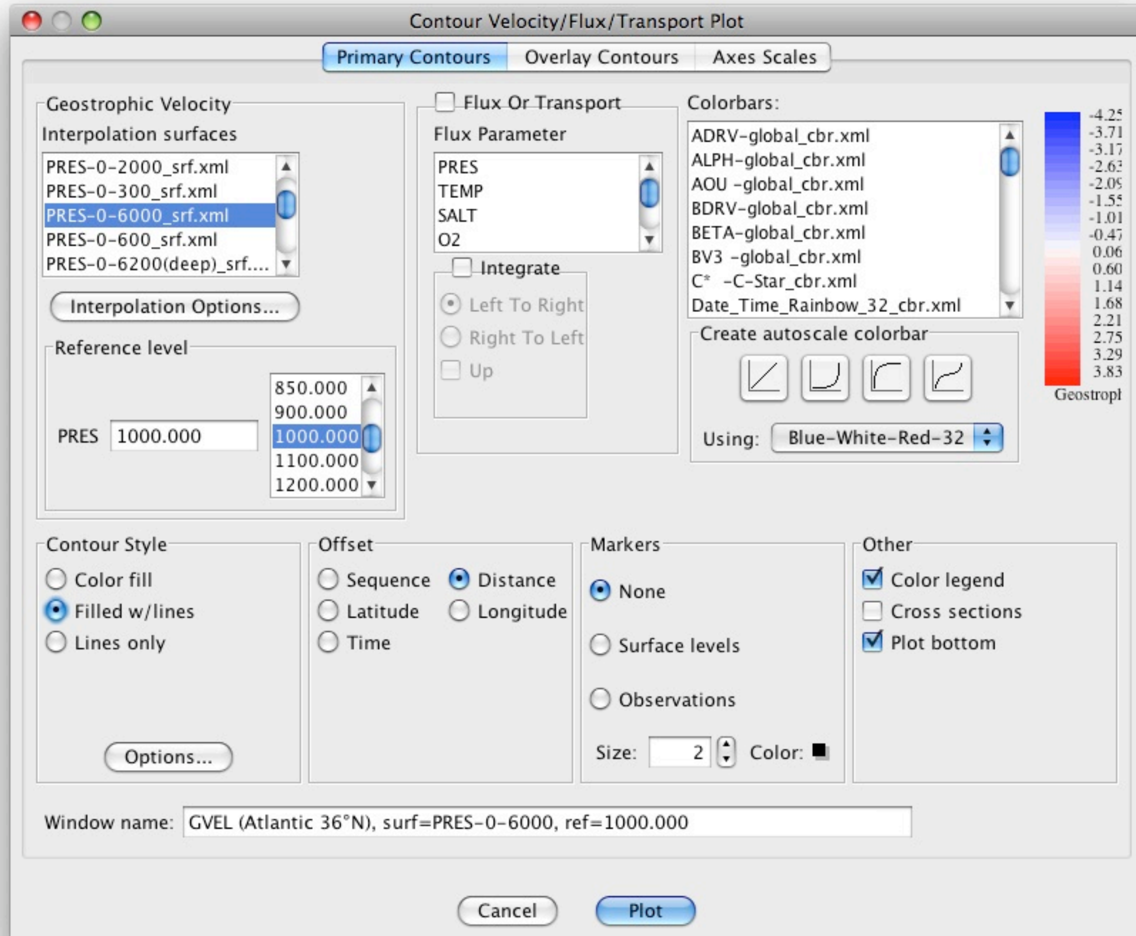
Range (again, useful to examine a particular layer in detail). This works similarly to the Configure Vertical Cross Section dialog.

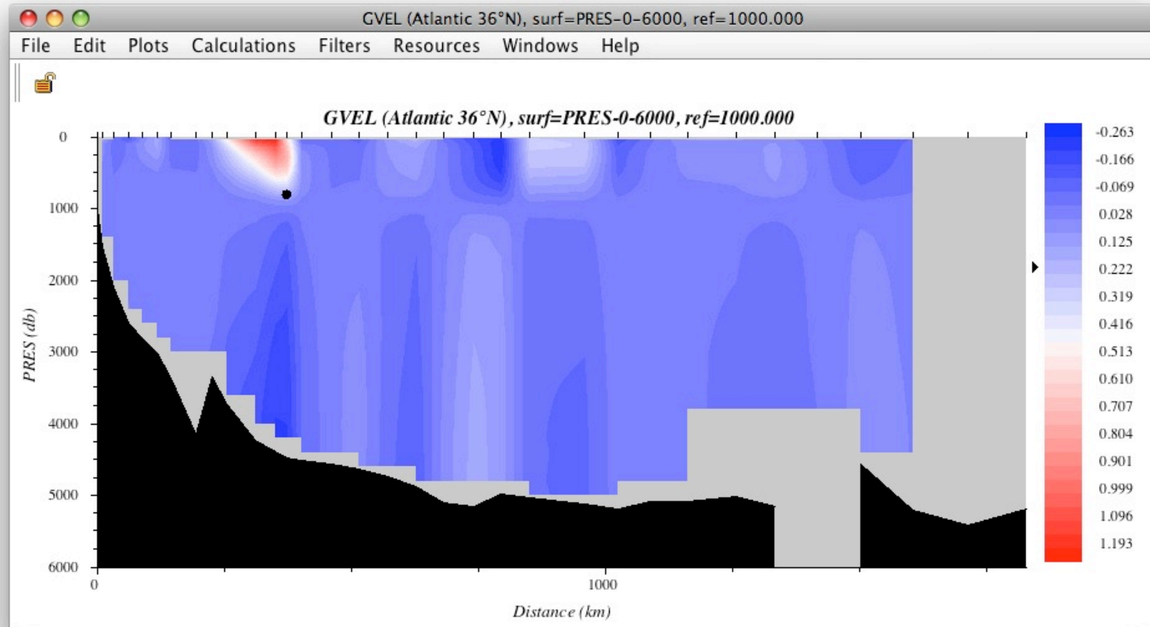


5. The data display in the lower section of a contour plot window provides metadata for the current station and the values of the interpolated parameter and surface value at the data cursor.

Double-clicking on the main section (the large panel on the upper left) of an active contour plot brings up a slightly abbreviated version of the Contour Plot dialog, with current settings shown. This makes it straightforward to modify an existing plot.

Gradient Contour Plots


Contour Velocity/Flux/Transport Plot
Primary Contours
Overlay Contours
Axes Scales
Geostrophic Velocity
Interpolation surfaces
PRES-0-2000_srf.xml
PRES-0-300_srf.xml
PRES-0-6000_srf.xml
PRES-0-600_srf.xml
PRES-0-6200(deep)_srf....
Interpolation Options...
Reference level
PRES
1000.000
850.000
900.000
1000.000
1100.000
1200.000
Flux Or Transport
Flux Parameter
PRES
TEMP
SALT
O2
Integrate
Left To Right
Right To Left
Up
Colorbars:
ADRV-global_cbr.xml
ALPH-global_cbr.xml
AOU -global_cbr.xml
BDRV-global_cbr.xml
BETA-global_cbr.xml
BV3 -global_cbr.xml
C* -C-Star_cbr.xml
Date_Time_Rainbow_32_cbr.xml
Create autoscale colorbar
Using:
Blue-White-Red-32
-3.71
-3.17
-1.01
0.06
0.60
1.14
1.68
2.21
2.75
3.29
3.83
Contour Style
Color fill
Filled w/lines
Lines only
Options...
Offset
Sequence
Distance
Latitude
Longitude
Time
Markers
None
Surface levels
Observations
Size:
2
Color:
Other
Color legend
Cross sections
Plot bottom
Window name:
GVEL (Atlantic 36°N), surf=PRES-0-6000, ref=1000.000
Cancel
Plot

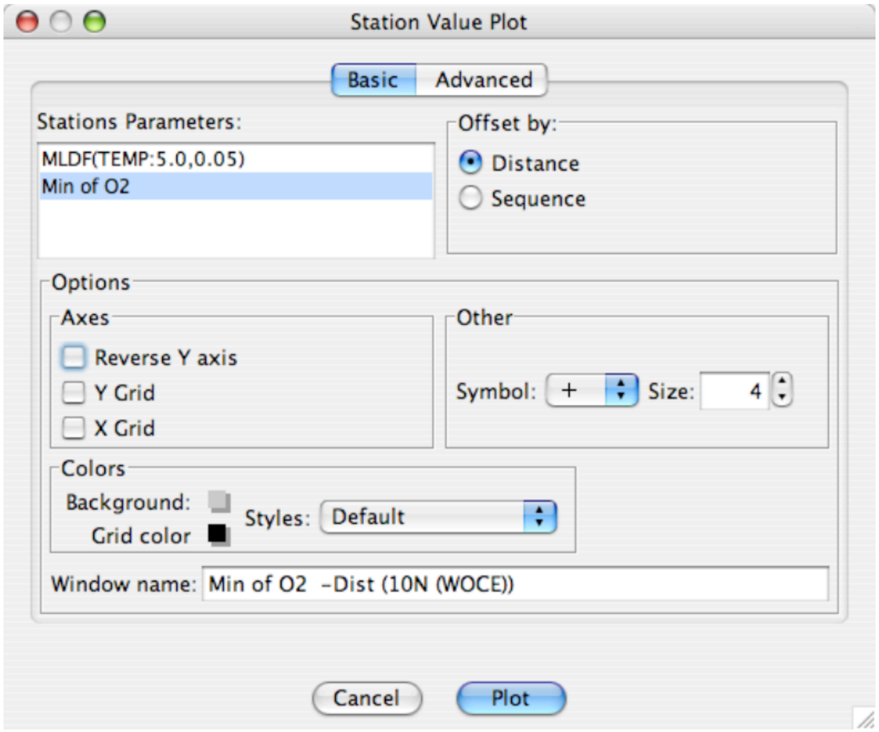


Note: This plot also incorporates a station filter where all stations missing geopotential anomaly have been discarded.

Station Value Plot

Each Java OceanAtlas Station Calculation (available under the *Calculations* menu; see the Station Calculations section of the User Guide) results in a single calculation result per station. The values can be used to color the station symbols on a Java OceanAtlas map plot (see the Map Plot - Station Colors section of the User Guide), or they can be plotted on an x-y plot via *Station Value Plot* under the *Plots* menu.

Selecting *Station Value Plot* brings up the Station Value Plot dialog:



A list of the current station calculations is shown. Any one can be selected.

Options and choices include:

Offset: Plot either in the sequence in which the data files are found in the opened (and 'added' data files), with a constant offset between each station, or with the station spacing along the x axis proportional to the distance (km) between consecutive stations in the data file.

Axes: Reverse the y axis and/or add grids to the x axis and/or y axis.

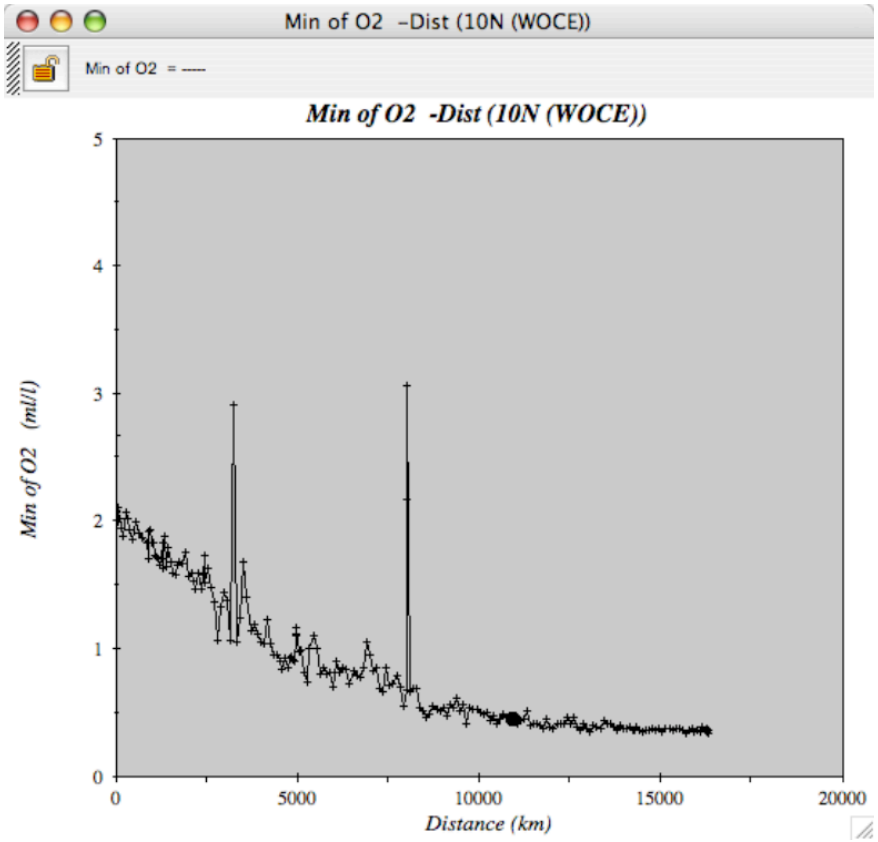
Other: Choose a plot symbol and its size.

Colors: Choose a different background and/or grid color (by clicking on the color swatch), or choose one of the default color styles.

Window name: Change the name of the plot suggested by Java OceanAtlas.

Y Axis and *X Axis*: Specify the minimum and maximum x axis and y axis limits, the increment between labeled tic marks, and the number of minor tics between labeled tic marks.

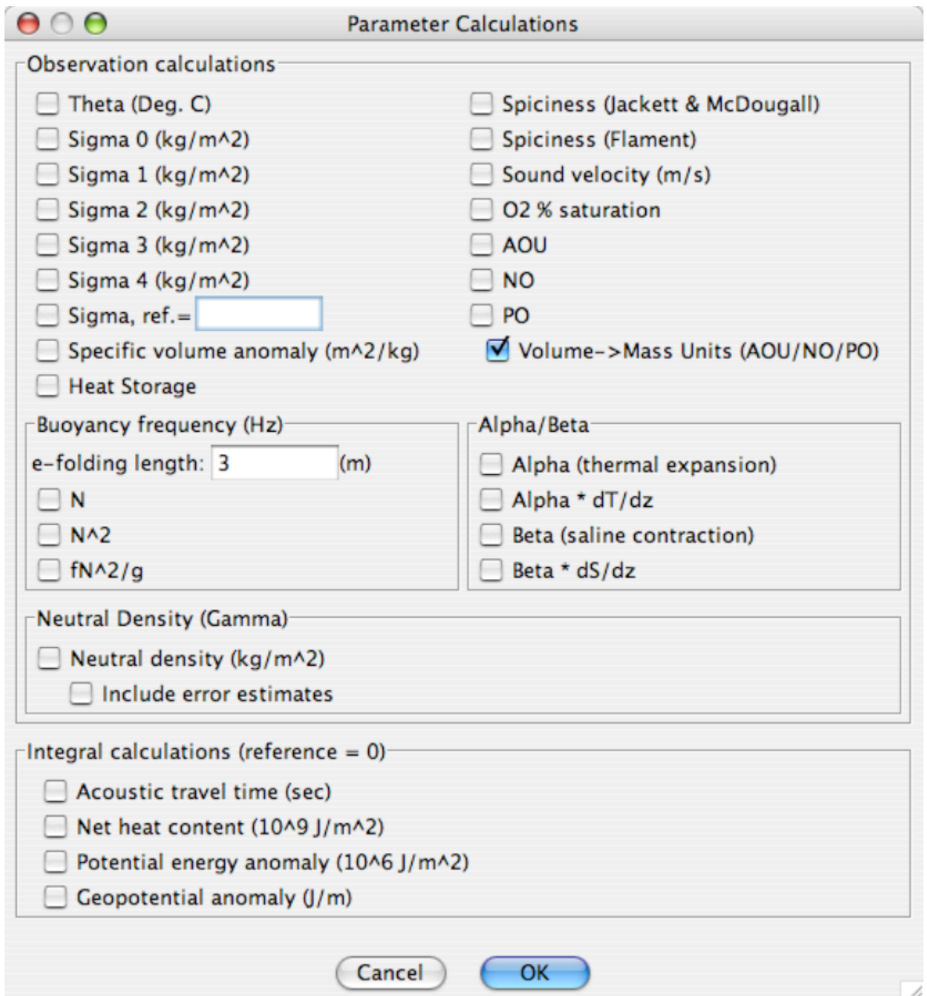
An example of a station value plot, set up in accord with the dialog above, is shown below:


Min of O2 -Dist (10N (WOCE))
Min of O2 = -----
Min of O2 -Dist (10N (WOCE))
Min of O2 (ml/l)
Distance (km)
0
1
2
3
4
5
0
5000
10000
15000
20000

CHAPTER 6. JAVA OCEANATLAS CALCULATIONS

Parameter Calculations

Java OceanAtlas provides a library of standard oceanographic parameter calculations. Microcomputers calculate quickly, and so there is no need to burden data files with extra parameters that can easily be calculated. You are, however, free to include calculated parameters in new data files. The Parameter Calculations dialog is accessed via the *Calculations* menu:



The definitions, algorithms, and references for each of the parameter calculations are provided in a separate User Guide document. Below are brief descriptions. Note that to calculate certain parameters, a data file must contain pressure (PRES, always present), temperature (TEMP), and/or salinity (SALT or CTDS). JOA accepts alternate spellings for these required parameters.

Theta is potential temperature calculated with respect to a reference pressure of 0 decibars. (Oceanographers use potential temperature in order to examine deep temperatures corrected for pressure effects.)

Sigma 0 is density in kg/m^3 minus 1000, calculated with respect to a reference pressure of 0 decibars. This is also known as sigma-theta. (Oceanographers find it useful to examine the distributions of density, or of characteristics plotted against a density parameter as the Y-axis.)

Similarly, *Sigma-1* (etc.) is the density anomaly (as defined immediately above), but calculated with respect to a reference pressure of 1000 decibars (etc.). Generally speaking, oceanographers prefer to examine density with respect to a reference pressure near those of the critical range of pressures in the desired examination, the reason being, essentially, that cold waters are more compressible than warm waters, and so for two parcels of water at equal density at the sea surface, but one warmer (and saltier) than the other, if they are displaced adiabatically to depth, the colder one will be denser at that higher pressure than the warmer one. For an example, prepare and plot sections of Sigma-0 and Sigma-4 for a meridional (north-south) section in the Atlantic Ocean (the long meridional 'Western Basins' composite data set prepared by Arnold Mantyla - available on the Atlas of Ocean Sections CD-ROM or from <http;//odf.ucsd.edu/joa> is a good choice). There are clear differences in the shapes of the distributions of Sigma-0 and Sigma-4.

Java OceanAtlas also provides for calculation of density anomaly at an arbitrary reference pressure. The reference pressure can be entered in several ways; e.g., 1200 or 1.2 for sigma-1.2. Java OceanAtlas will attempt to make sense of whatever is entered.

Specific Volume Anomaly is the anomaly of volume compared to a reference water column at (S, T, P) = (35.0, 0.0, P).

Spiciness is an attempt to characterize the strength of intrusions or, perhaps, the relative degree of mixing of water masses. JOA includes spiciness calculations as described by Jackett and McDougall or Flament.

Sound Velocity is sound speed at (S, T, P) in meters sec^{-1}.

Oxygen % Saturation is calculated from the algorithm for equilibrated seawater at (S, T, 0).

AOU (apparent oxygen utilization) is similar to the above, but is just the difference between the measured and saturation concentrations (in mass units, i.e. μM kg^{-1}).

NO and *PO* are respiration-corrected nitrate and phosphate. These are valid only in mass units (μM kg^{-1}) and so if the original observations are in volume units (ml l^{-1} for O_2 and μM l^{-1} for NO_3 and PO4), the *Volume->Mass Units* button must be checked.

Heat Storage is the local value of temperature times the specific heat.

Buoyancy Frequency, a measure of stability also called Brunt-Vaisala Frequency, is provided in unsquared and squared versions. *Potential Vorticity*, fN^2/g, is a related conservative parameter. Buoyancy frequency calculations from closely-spaced (e.g. CTD) observations must be smoothed. The e-folding length for the filter is user-selectable. (You may want to experiment with higher numerical settings than the default.)

The thermal expansion coefficient, *alpha*, is a measure of the rate of change of density with respect to temperature at constant pressure and salinity. The expansion coefficient for salinity, *beta*, is a measure of the rate of change of density with respect to salinity at constant pressure and temperature. Locally, comparing $\alpha * dT/dz$ with $\beta * ds/dz$ helps to evaluate the differing contributions to stability of the temperature and salinity gradients.

Neutral density (gamma) as developed by Jackett and McDougall. Can optionally compute and report the error terms associated with this calculation.

Java OceanAtlas also provides for *Integral Calculations* where parameters are integrated from the sea surface to the deepest observation. Java OceanAtlas produces negative numbers for these downward-integrated parameters. When they are interpolated, referencing to a deep level reverses the sign:

Acoustic Travel Time is the integrated time in seconds for a sound pulse to travel from the sea surface to a given depth.

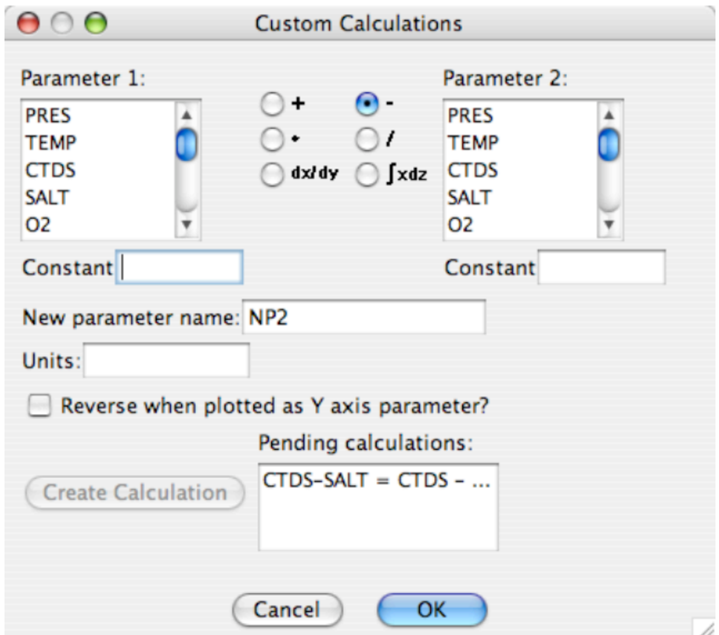
Net Heat Content integrates *Heat Storage* from the surface to the bottom.

Potential Energy Anomaly integrates the product of pressure and volume anomaly to provide a measure of the available potential energy relative to a standard water column.

Geopotential Anomaly is the integrated volume anomaly, i.e. the ‘height’ difference between the observed water column and a standard water column.

Custom Parameters

This command allows construction of new parameters using mathematical operations applied to existing parameters. For example, a difference or a ratio between two parameters can be calculated or a parameter can be divided by a constant. Selecting *Custom Parameter* from the *Calculations* menu presents the Custom Calculations dialog (shown below after the first set-up steps for a calculation of the difference between bottle salinity and CTD salinity in a WOCE Hydrographic Program data file):



There are three ways to specify the operands of a custom calculation:

- selecting a parameter from both lists (e.g., NO3/PO4 or dTemp/dPress);
- selecting a parameter from the left list and entering a numeric constant in the right text field (e.g., NO2 * 10.0)
- selecting a parameter from the right list and entering a numeric constant in the left text field (e.g., 1.0 + SALT)

In all cases an operator must also be selected by clicking one of the center radio buttons, and a name assigned to the new parameter (Java OceanAtlas will suggest a name for the new parameter but this can be changed). Units may also be specified, and will show up in the Data Window if the appropriate option is used.

> Tip: If nitrate/phosphate ratio NO3 /PO4 is calculated and named NUTR the 'NUTR' color/contour bar provided with Java OceanAtlas can be used. Similarly 'NOPO' is made for the ratio of NO divided by PO.

Check the *Reverse when plotted as Y-axis parameter* button if the new parameter should plot with a reversed y axis (as traditionally done in oceanography with pressure and density).

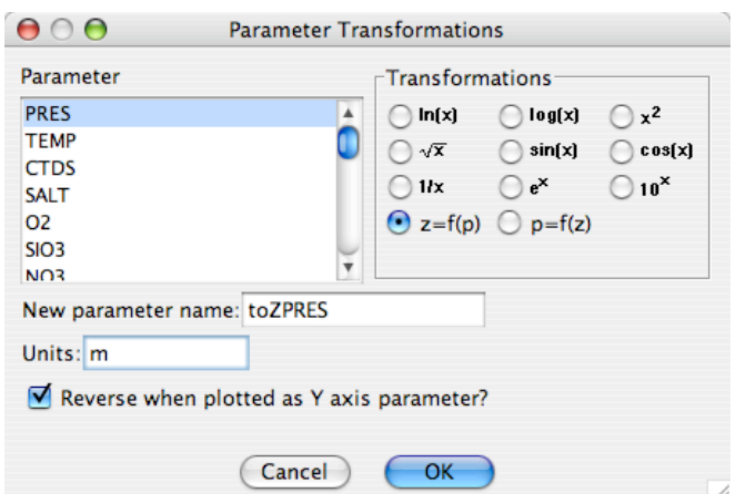
Finally, when the calculation is specified correctly, click on the *Create calculation* button. This does not actually carry out the calculation. Instead, as each new calculation is created via *Create calculation*, Java OceanAtlas adds it to a list of pending calculations. This way a sequence of linked calculations (or simply multiple calculations) can be set up from a single session in the dialog. Note: after you create a pending calculation, the new parameter is available to be used in subsequent custom parameter calculations and is displayed in the parameter lists. The calculations are not actually carried out until *OK* is clicked.

Clicking the *OK* button performs the calculation(s). Calculations are applied to all open observations whether they are visible or not (data can become invisible because of observation filters or station filters). The new parameter(s) can be used in all the standard Java OceanAtlas ways including in other custom calculations.

This is a very simple method to calculate nutrient ratios, CFC ratios, etc. And the capability to use one Custom Parameter to calculate another means that in many cases a string of simple custom calculations can generate a more sophisticated data product.

Parameter Transformations

Java OceanAtlas provides for elementary parameter mathematical transformations. Selecting *Parameter Transformations* from the *Calculations* menu brings up the Parameter Transformations dialog:



The user should select one parameter from the scrolling *Parameter* list, which includes all current original and calculated parameters, and select any of the listed *Transformations*. Java OceanAtlas will suggest a *New parameter name*, but this can be changed. *Units* (optional) may be specified, and if the newly transformed parameter is to be plotted with increasing values down when used as the y axis (as with pressure or density) that option can be checked. Clicking *OK* will carry out the calculation and exit the dialog. In combination with Java OceanAtlas custom calculations, it is feasible to step-wise construct new parameters that are more than one-element calculations, though at this writing Java OceanAtlas does not yet offer a full equation editor/parser.

Note that two of the options permit converting pressure into depth and depth into pressure. These are done via 'close enough for most work' shortcuts (simple quadratic relationships) rather than the oceanographically correct integrations of the equation of state.

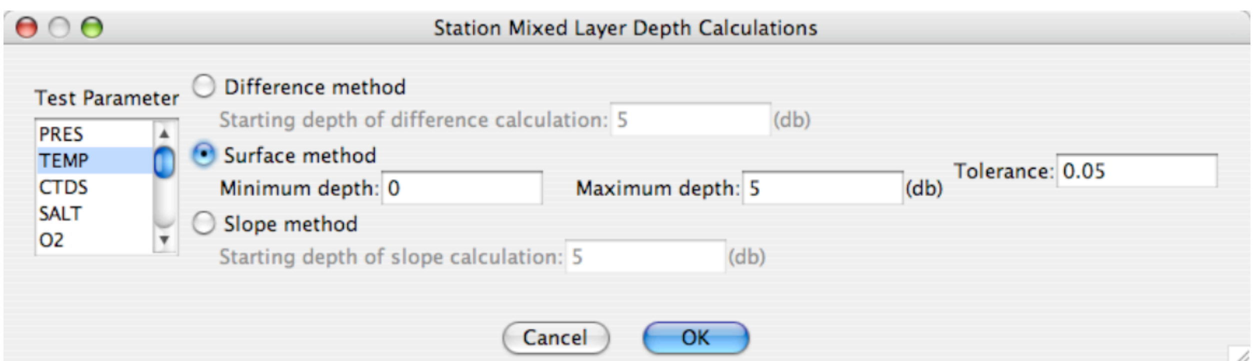
Station Calculations

The term 'station calculations' in Java OceanAtlas refers to calculations with one result per station, rather than one result per pressure at the station, as in all 'parameter calculations'. The results of station calculations are displayed in the Current Station panel of the Java OceanAtlas Data Window.

Java OceanAtlas currently has available six types of station calculations: mixed-layer depth, integration, interpolation, neutral surface, extrema, and station statistics. Each is discussed below.

Mixed-Layer Depth

Java OceanAtlas will calculate the depth of the mixed layer at each station in the opened and added data files in the current Data Window:



Any original or calculated parameter can be used as the test parameter. The *Test parameter* is the parameter whose variability - or lack of it - is used as the indicator of mixed-layer depth.

Three methods of mixed-layer depth calculation are available:

In the *Difference method* Java OceanAtlas, beginning at the *Starting depth of difference calculation*, will look at successively deeper (higher pressure) observations in each data file until it reaches an observation whose value for the test parameter differs from the value of the test parameter at the starting depth by more than a user-chosen *Tolerance*. For example, if 'TEMP' is selected with a starting depth of 5 and a tolerance of 0.5, Java OceanAtlas will report back the thickness between 5 and whatever level contains a temperature 0.5 degrees different than the value at 5.

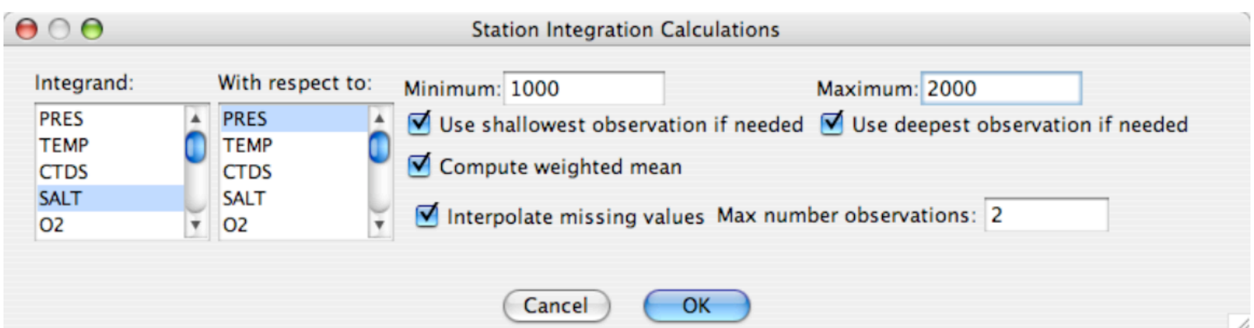
The *Surface method* is similar to the difference method except that instead of comparing the data down the water column to the value at the starting depth, Java OceanAtlas compares the data to a calculated bulk surface value that is created by averaging the test parameter over the range of pressures specified. The depth of the mixed layer is reached when the difference between a observations's value of the test parameter and the bulk surface mean is greater than the tolerance.

The *Slope method* is similar in some respects to the difference method, but each successive measurement pair is used for the examination of whether or not their difference exceeds the tolerance. For example, if 'TEMP' is selected with a starting depth of 10 and a tolerance of 0.05, Java OceanAtlas will compare the next deeper measurement to 10, the one after it to the one after 10, etc., and report back the thickness between 10 and whatever level contains a temperature 0.05 degrees different than the previous value.

Warning: These algorithms are best suited to CTD profiles. The intent is to supply for a typical data file a reasonable approximation of mixed layer calculations.

Integration

Java OceanAtlas will calculate the integral (step-wise) of one original or calculated parameter over a selected range of any other parameter:



Options are presented to

> *Use the shallowest (or deepest) observation if needed* - for example if integrating over a sigma-0 range and the density at the shallowest observation is greater than the starting value should the values at the shallowest observation be substituted or should the calculation result be reported as missing for that station?

Compute weighted mean: weight the result over the limits of integration (otherwise the results are left raw). This results in an integration in the units of the original integrand.

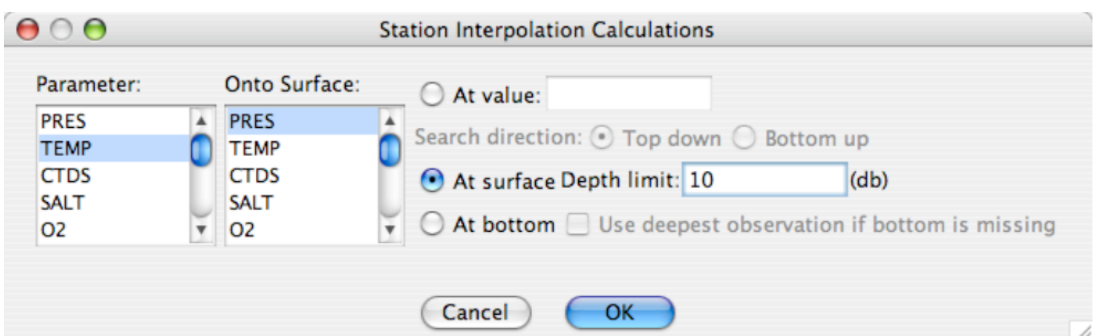
Interpolate missing values: if an observation of the integrand parameter is missing inside the range of integration, should Java OceanAtlas provide a linear interpolation at the level from the closest levels above and below with non-missing observations? Can also set a custom value of *Max number of observations* to limit interpolation of missing values.

The integration direction can be specified to be from shallow to deep, from deep to shallow, or by searching down for minima of the parameter of the axis of integration and up for maxima of the parameter of the axis of integration.

Warning: The integration calculation is not sophisticated. It works reasonably well for ranges of any parameter that does not contain mid-profile extrema (it works well over pressure or density, for example). Despite the logic choices, the results can be misleading when a parameter is integrated over an axis with mid-level extrema. The intent is to supply for a typical data file a reasonable approximation of integration calculations.

Interpolation

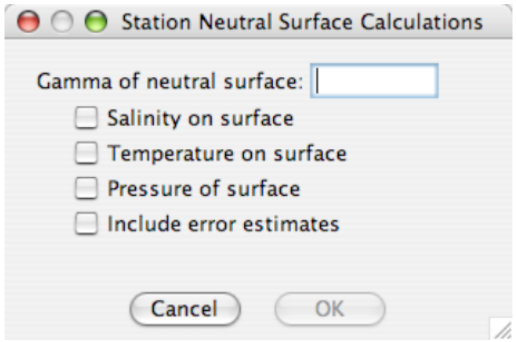
Java OceanAtlas will interpolate one original or calculated parameter to a selected value of any other parameter:



For example, it is possible to interpolate salinity to a value of pressure (depth) or density. It is also possible to compute the depth of an iso-surface of any other observed or calculated parameter by interpolating pressure onto the value of that parameter. For example, interpolating pressure to a value of temperature equal to 10.0 will return the depth of the 10-degree isotherm. If you are interpolating to pressure, JOA has some additional options that allow you to interpolate to the surface or the bottom of the ocean. Interpolations to the surface have the option of defining a depth limit so that the first observation of a deep cast is not considered the surface. When interpolating to the bottom, you can specify whether to use the deepest observation if the depth datum for that station is missing.

Neutral Surface

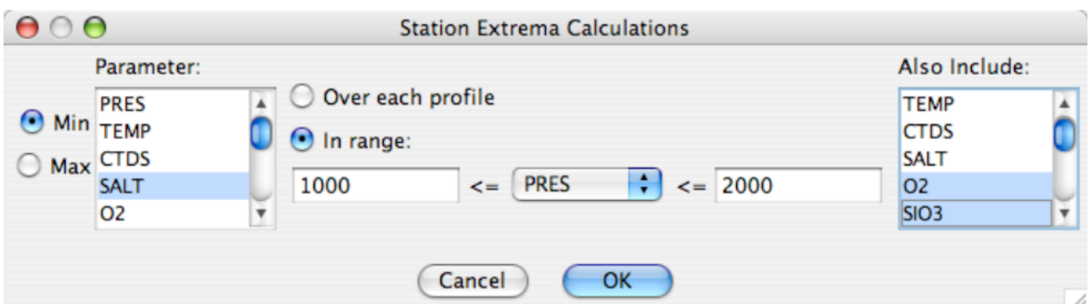
This option calculates the value of salinity, temperature, and depth of a neutral density surface for a given value of neutral density (gamma):



This calculation will automatically calculate gamma if it hasn't already been computed. Users can optionally select to include the error estimates of the observed parameters interpolated onto the neutral surface.

Extrema

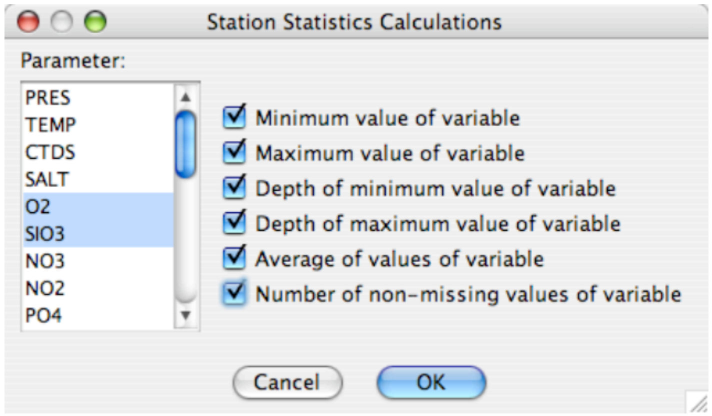
This station calculation option computes the minimum or maximum of a selected parameter for a whole profile or over the range of another parameter:



The figure above shows the Extrema Calculation dialog setup to compute the minimum salinity in a pressure range of 1000 to 2000 decibars for each station. The list on the right allows selection of parameters to report at the extreme value. In this example, O2 and SIO3 will be reported at the salinity minimum found in the 1000-2000 pressure range. Note: each parameter reported at an extremum is a separate station calculation variable.

Station Statistics

This option allows JOA to compute a variety of built-in statistics for selected observed and calculated parameter at each station:



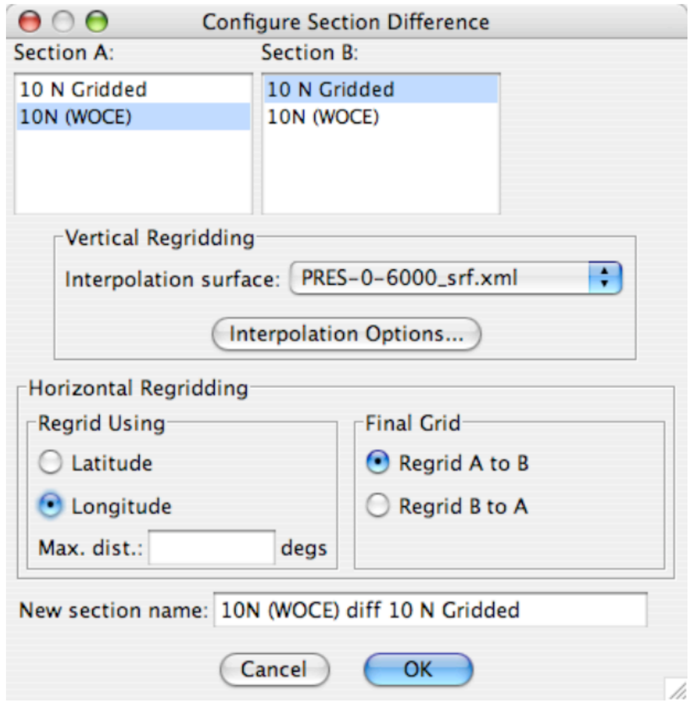
Currently, JOA can compute the minimum, maximum, depth of minimum, depth of maximum, average value, and number of non-missing values for each selected parameter at each station. There is some overlap in the functionality of this calculator with the Extrema calculator -- if you know the minimum value of a parameter you can use the Interpolation calculator to find the associated depth of this minimum.

Section Calculations

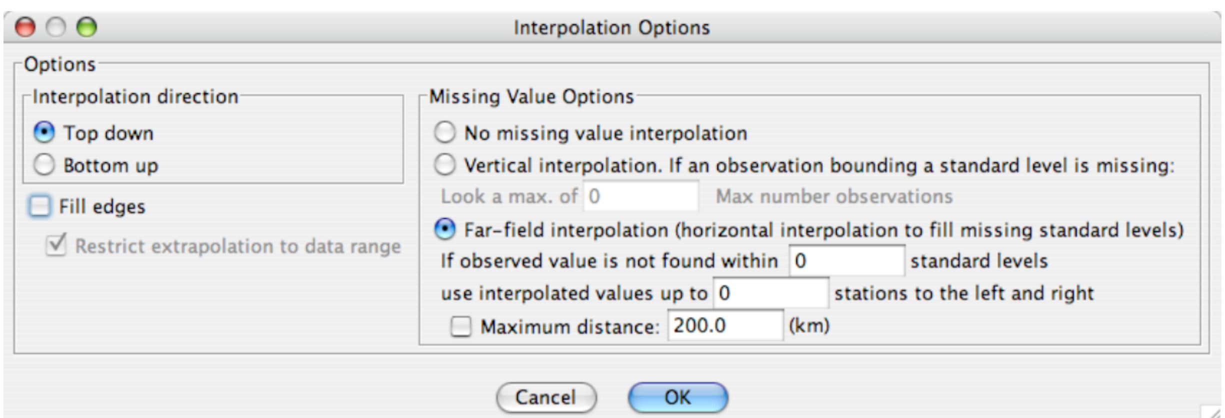
This refers to calculations that are based upon manipulations of complete sections. Section calculations result in the creation of new Java OceanAtlas Data Window. Currently, there are two section-level calculations: section difference and mean cast.

Section Difference

This calculation computes the difference between sections from any two open datasets. Before differencing sections, the two sections must be interpolated to the same horizontal and vertical grid. Thus the two sections have to overlap in pressure and either longitude or latitude. Selecting Section Difference from the Section Calculations menu displays a dialog to identify the sections to difference and regridding options:



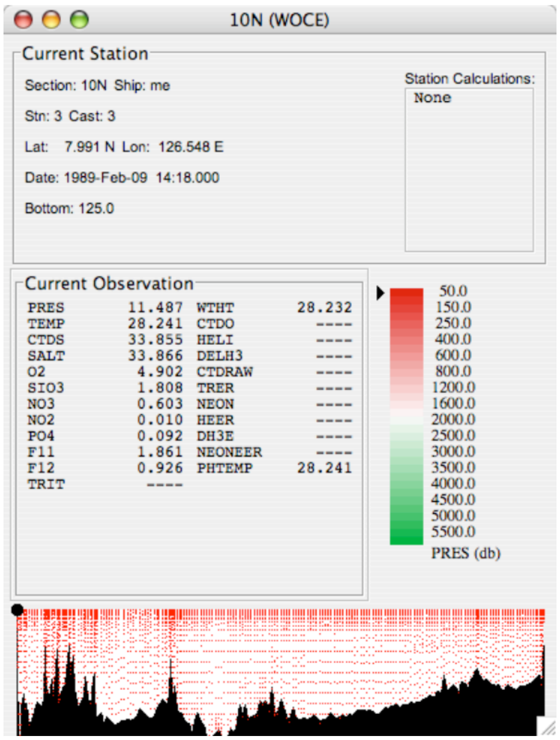
The lists at the top show all open datasets. Here we have setup the difference between a 10N section from WOCE *in-situ* data and an overlapping section from a gridded data source (in this case the gridded data come from the World Ocean Atlas, 1 degree annual average). The vertical grid is selected from the interpolation surface popup—all the original values in each section will be first interpolated to these levels (much like what is done to produce a contour plot). To configure the interpolation settings, click the *Interpolation Options* button:



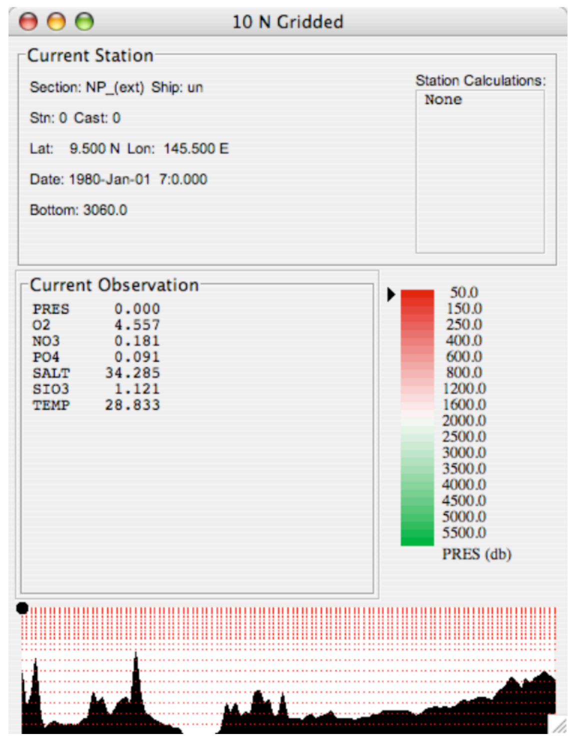
Refer to the section on Contour Plots for a detailed explanation of these options.

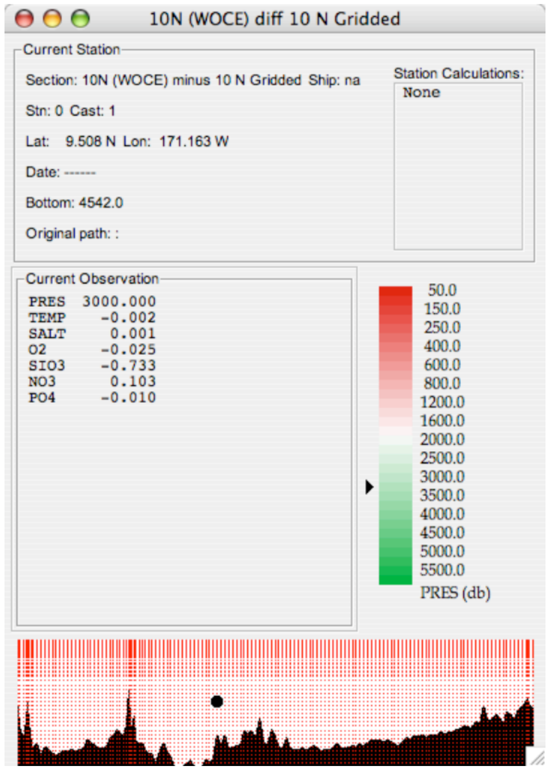
Horizontal gridding is done on the overlap region of the two sections in either the latitude or longitude dimensions. The two sections thus have to overlap in the chosen dimension. The two sections in the example are both along 10° North latitude and overlap in longitude. The final setting is to define the destination grid of the difference section. If *Regrid A to B* is selected, for each location in section B, JOA will attempt to find bounding stations in section A and interpolate the values at each level to the location in B. Stations in section B are, by definition, already on the destination grid. If *Regrid B to A* is selected, for each location in section A, JOA will attempt to find bounding stations in section B and interpolate the values at each level to the location in A. Stations in section A are, by definition, already on the destination grid. Section differencing will only results in stations in the overlap region between the two sections.

The following illustrates the results of the differencing between the two sections in the example above:

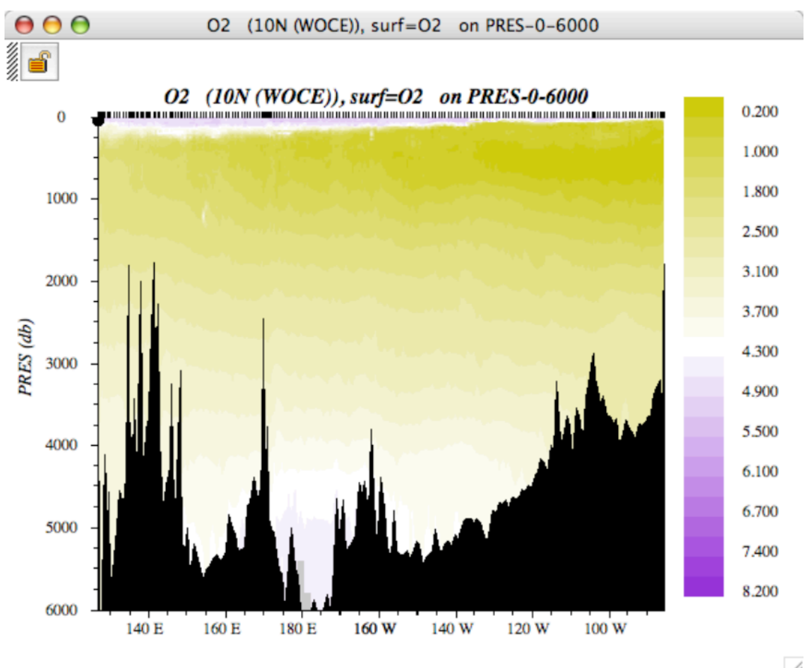


Minus

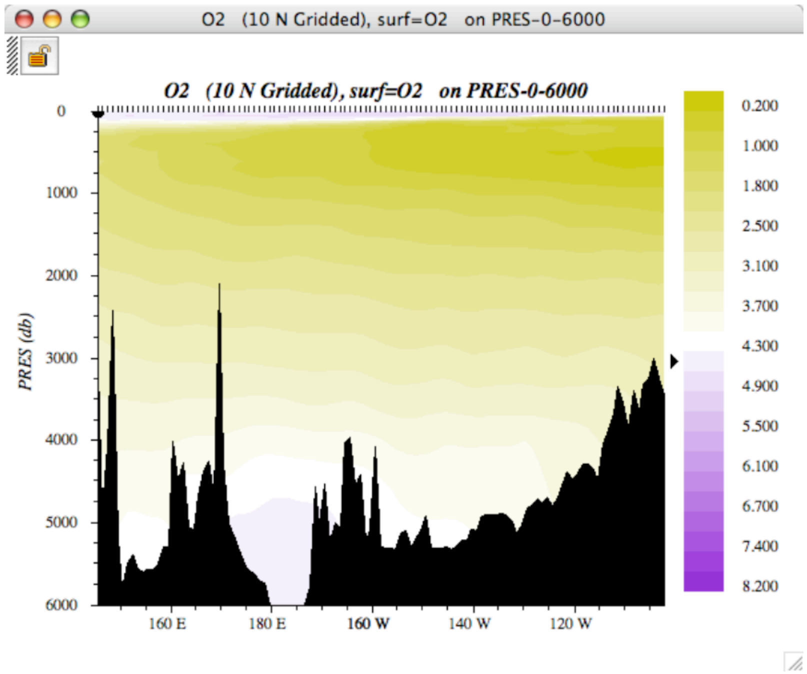


Equals 

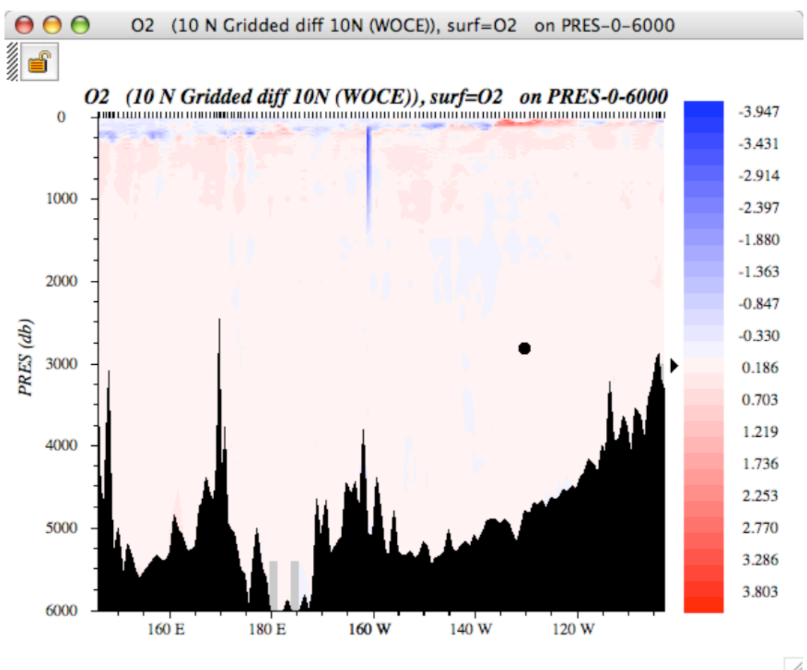
Here is a contour plot of O_2 from the WOCE section:



Here is contour plot of O_2 for the gridded data:



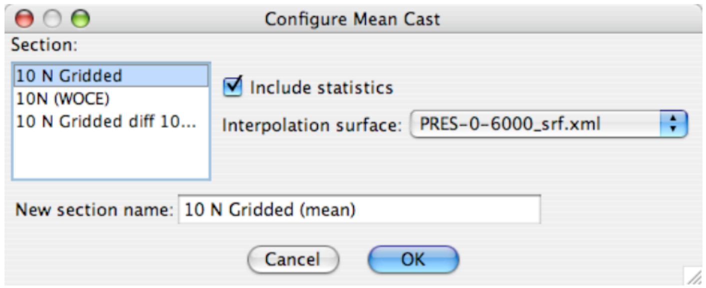
Here is the difference between the *in-situ* data and the gridded data also as a contoured section:



Mean Cast

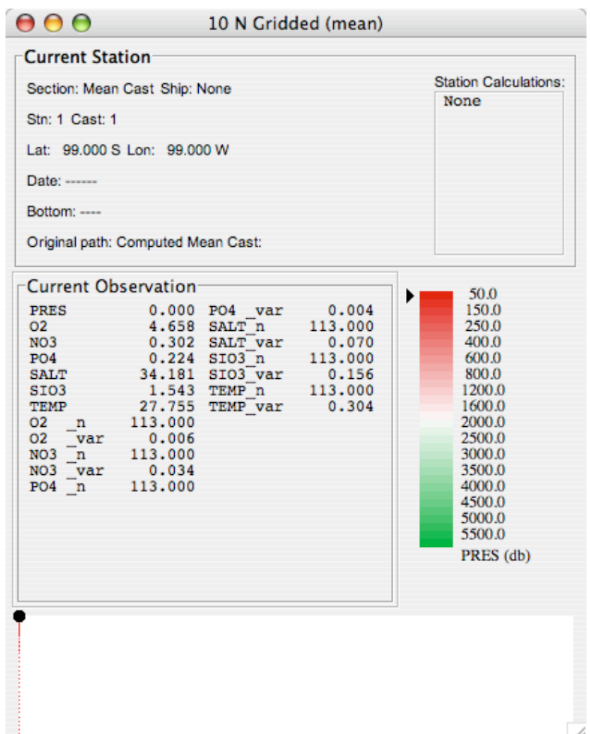
This option computes a single cast that represents the average value of all the observed variables. Before computing the mean cast, all observed values are interpolated to user-selected vertical levels.

Selecting *Mean Cast* from the *Section Calculations* menu displays the configuration dialog for this calculation:



Here I have selected to compute a mean cast of the 10°N gridded section interpolated to the 0-6000 built-in pressure surface.

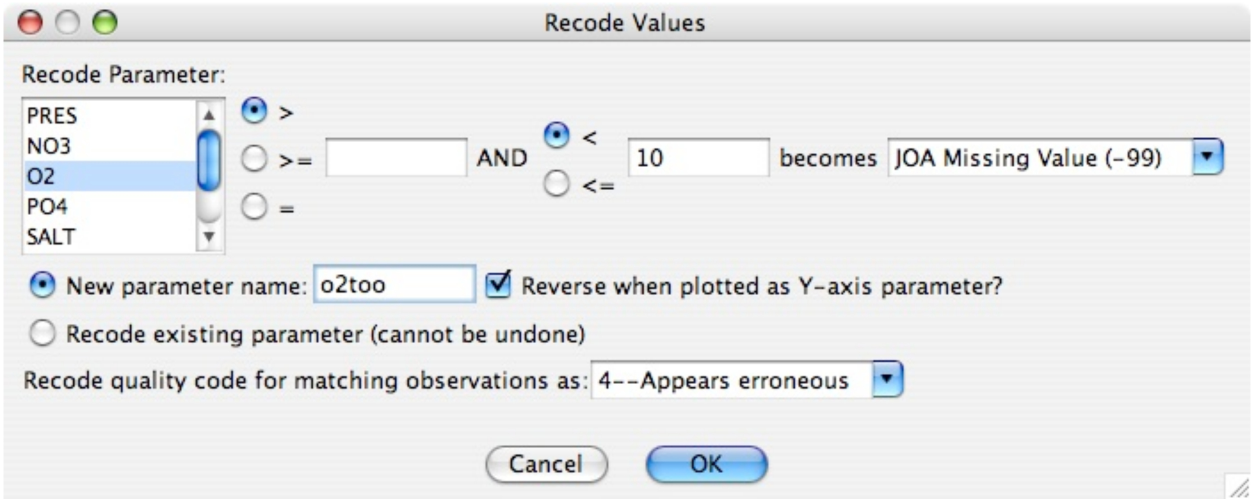
The result of this calculation is always displayed in a new data window:



Notice that this calculation also includes new statistics variables for each observed variable. This mean cast can be saved and used in future JOA sections.

Recode Values

This calculation allows you to filter an existing parameter's values or create a new parameter based upon the values of the parameter. For example, parameter values below a certain level can be recoded as missing. Selecting *Recode Values* from the *Calculations* menu displays the configuration dialog for this calculation:



To recode a parameter, choose a parameter from the list and fill in the range of values. Next, choose relational operators according to whether you want a strict (>, <, or =) or non-strict (≥, ≤) inequality. Finally, choose the new value for parameter values that fall in the selected range. For example, the illustration above is set up to recode all values of O2 below 10 (ml/l) as missing.

Notice that the first logical operation for this recoding is optional, and left blank, since we only care about numbers less than 10.

This interface gives you the option of creating a new parameter or recoding a parameter 'in place.' In the latter, the original values of the parameter are changed according to the recoding specified. This is useful if you want to recode, for example, a nutrient parameter to remove questionable values and then use the recoded parameter in NO, PO, or AOU calculations.

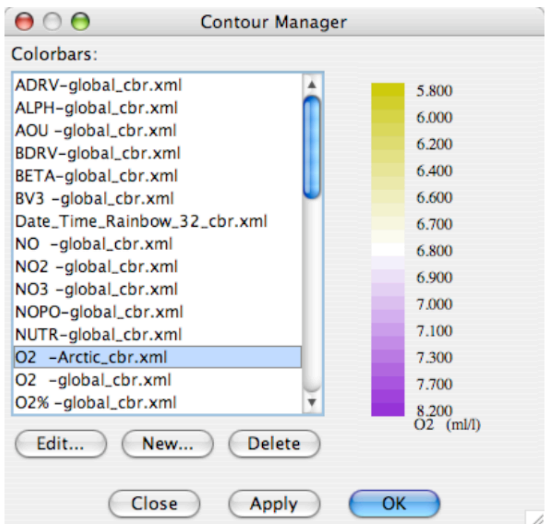
Parameter recoding cannot be undone – if you hopelessly mangle the data, you must close your data file, not saving changes, and then re-open the original data file.

Check the *Reverse when plotted as Y-axis parameter* button if you want the new parameter to plot with a reversed Y axis (as traditionally done in oceanography with pressure and density).

Contour Manager

The top of the four choices in the *Resources* menu is *Contour Manager* (also accessed via cmd/ctl-K), which leads to the interfaces for selecting, editing, creating, and deleting the colorbars used in profile, property-property, and map plots, and the solid colors and/or contour lines which can be chosen for contour plots. (If no color bars/contours are chosen, contour plots are autoscaled.) The Contour Manager is a customization feature that is used often in Java OceanAtlas.

Selecting *Contour Manager* leads to the Color Manager dialog (shown below after a colorbar has been selected):



At first, the Color Manager shows only a scrollable text list of the available color bars, with only the *New* and *Done* buttons active (in bold). Clicking on '*Done*, or clicking on the 'close box' in the upper left, closes the dialog without further action. Selecting the *New* button will open a blank version of the Contour Editor dialog to create a new color bar (see below).

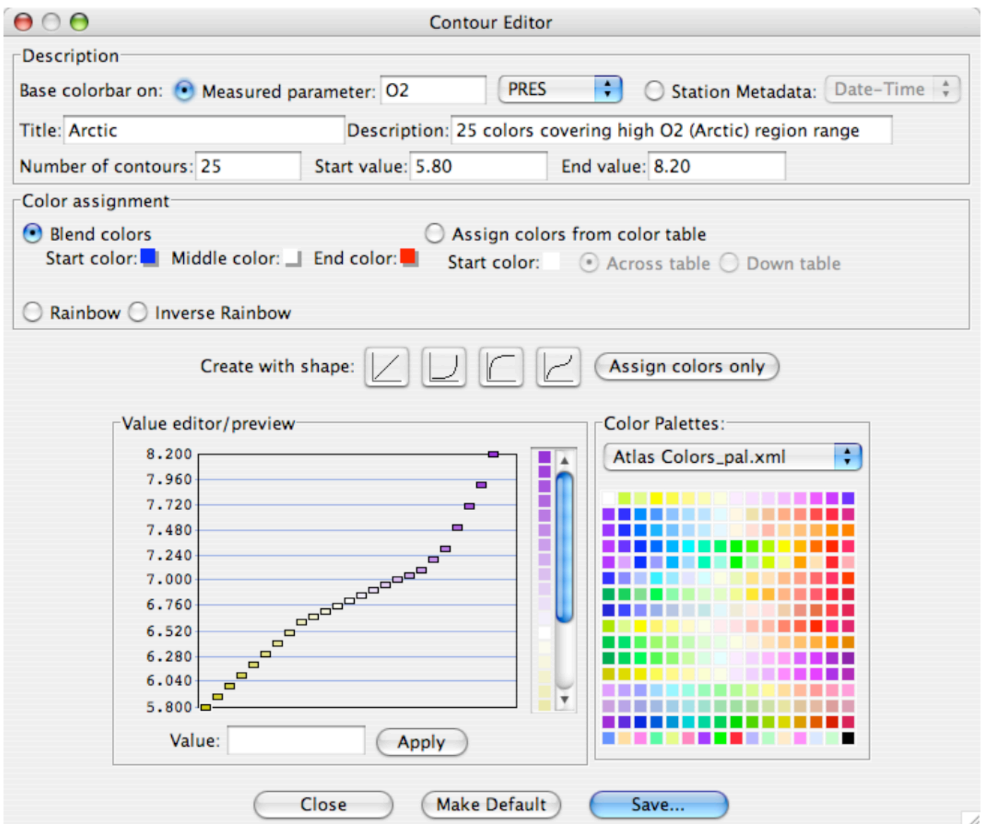
Clicking on one of the choices in the scrollable list will both place a preview of that color bar in the right hand side of the dialog (as shown in the illustration) and also enable the Edit, *Delete*, *Apply*, and *OK* buttons. Edit leads to the same dialog that *New* does, but in this case with the parameters for the selected color bar in place for editing. *Delete* removes the selected color bar file from the disk. *Apply* updates open profile and property-property plots with the selected color bar while the dialog stays open. And *OK* makes the selected color bar the current (not default) color bar in the Data Window and returns you to the active window, meanwhile redrawing the Data Window and all open profile and property-property plots from the current data set, colored now by the newly-chosen color bar.

It is not strictly necessary to create new color bars with the *New* button: a new color bar can be created by editing an existing color bar and either replacing the color bar with the same name or creating a new color bar file on disk. (More about this later.)

Warning: *Delete* deletes the selected color bar file ***permanently*** from the disk, so should be used only as appropriate.

The default color bar is protected by Java OceanAtlas. It cannot be deleted until a new default is set via the Contour Editor dialog.

Selecting *New* or *Edit* brings up the Contour Editor dialog, shown below for the 'O2 - global_cbr.xml' color bar (the main differences when selecting *New* are that the text fields are then blank, the *Preview* area is blank, and different buttons are active.)



A colorbar can be created for any observed or calculated value but colorbars can also be created for station metadata such as date/time, month, latitude, or longitude. The first thing you do when creating a new colorbar is to select whether it is based upon a measured parameter or on station metadata by clicking the appropriate radio button at the top of the dialog. If you create a metadata color, then certain items in the dialog become disabled that are not relevant to metadata such as the range of the colorbar. Metadata colorbars differ from parameter-based colorbars in that they are not based upon a pre-chosen range of values but rather take their values from the actual range of the metadata in the current dataset. That is, metadata colorbars recompute their ranges as data are opened, added, and removed by station filters.

Parameter-based Colorbars

The text in the *Parameter* field can be any parameter, for example one of the parameters from the list of available color bars, any to-be-calculated parameter, or simply *any* user-definable parameter name. There is a popup menu to allow you to easily enter parameters from the current dataset but you can create colorbars for any parameter whether it's been computed or in some other dataset.

> Note: The name must *exactly match a parameter name as spelled* in the Data Window (including blank spaces, such as 'O' '2' 'blank' 'blank') or Java OceanAtlas will not recognize it as a color bar for that parameter.

The text in the *Title* field can be whatever one wishes. Generally a short description is best (e.g., Arctic_1).

> Note: The parameter name will automatically become the prefix for the suggested file name when this color bar is saved on disk, so one need not repeat the variable name in the *Title* text field.

The text in the *Description* field will appear only in this dialog (but is stored on disk in the color bar file). For example it can be whatever text might assist one in recalling the reason the color bar was created. This field can be left empty.

The *Number of Contours* can be whatever one wishes (up to 128). For large numbers of contours there may be difficulty identifying specific colors/contours on plots. The 'global' color bars supplied with JOA are based on subtle color transitions and have up to 32 colors/contours in each. 16 contours is a good choice for more striking color transitions.

> Note: Java OceanAtlas colors are best viewed well with typical monitor bit depth settings equivalent to 'thousands of colors' or 'millions of colors'.

The *Start value* is the lowest numerical value for the desired contour range. It is not necessary that this correspond to the lowest numerical value in the data: all lower-valued data will be colored with the color assigned to the start value.

> Note: The *Start value* will automatically update if contour value changes made during editing (see below) result in a numerically lower value than initially supplied in the Start value field.

Similarly, the *End value* is the highest numerical value for the desired contour range. Again, it is not necessary that this correspond to the highest numerical value in the data: all higher-valued data will be colored with the color assigned to the end value.

Note: The *End value* will automatically update if contour value changes made during editing (see below) result in a numerically higher value than initially supplied in the End value field.

When a colorbar is initially created, one can specify *Color assignment* through several options of two basic techniques, *Blend colors* or *Assign colors from color table*. Just click on the preferred choice.

The *Blend colors* option will create a color bar that transitions in equal color-space intervals between a *Start color* (assigned to the start value contour), through a *Middle color* (assigned to the contour in the numerical midrange between the start value and end value) - white is a good choice for the middle color), to an *End color* (assigned to the end value contour). Clicking once on any of these three color swatches will bring up a dialog permitting one to adjust the color by any of several means. The default blend colors of blue, white, and red produce the smooth color transitions used in the supplied 'TEMP_global.cbr.xml' colorbar. The other choices of the *Blend colors* method are the *Rainbow* and *Inverse Rainbow*. An example of the *Inverse Rainbow* transition is the 'BV3 -global.cbr.xml' colorbar.

The *Assign colors from color table* option for color assignment works by first clicking on the color swatch labeled *Start Color* to select it and then clicking on any one of the 256 colors in the current Color Palette. This will assign that color as the *Start Color* for the *Start value*, and place that color in the swatch. Then, when a new colorbar is created, the colors will be assigned to the N (= Number of contours) contours sequentially across or down the table (whichever option is chosen), moving to the next row (column) down (right) as necessary.

‘The *Create with shape* buttons are used to define the initial values to assign over the range of values, by distributing the contours between the start and end contour values. There are four pre-set distributions: linear, upper asymptote, lower asymptote, and logistic. The linear distribution is often fine, but sometimes it is handy to have the contours bunched toward the low, high, or middle portions of the numerical range. The *Assign colors only* button preserves the existing shape of the color/value assignments (as viewed in the *Value editor/preview* panel), while assigning new colors to all the contours based on the *Color assignment* choices.

When all the elements of the dialog are set appropriately (i.e., parameter, title, number of contours, start value, end value, the color assignment choices, and start color’ and initial shape), clicking on one of the ‘Create with shape’ buttons will create the color bar, which will be shown/previewed in the *Value editor/preview* panel of the dialog.

The *Value editor/preview* panel in the lower left of the Contour Editor dialog provides not only a view of an existing or newly created/edited color bar, but once a color bar has been created or chosen for editing it also provides interfaces for adjusting both the numerical values and individual color/contour assignments:

Clicking on any colored symbol in the graph-like color/contour display (in the center of the *Value editor/preview* panel of the dialog) places the data value associated with that contour in a text field underneath the graph in the field labeled *Value*. This value can be changed in two ways: 1) use the up and down cursor keys to adjust the value or 2) by typing in a new value and clicking on the *Apply* button (or hitting the *Return* key on the keyboard). The next value in the graph can then be auto-selected by hitting the left or right arrows on the keyboard, and so with a little practice one can progressively move through all the values of a colorbar to further customize the values. This technique is the most reliable method for assigning user-specified numerical values to each interval in a colorbar. (It is how nearly all the values in the 'global' color/contour bars supplied with Java OceanAtlas were created.)

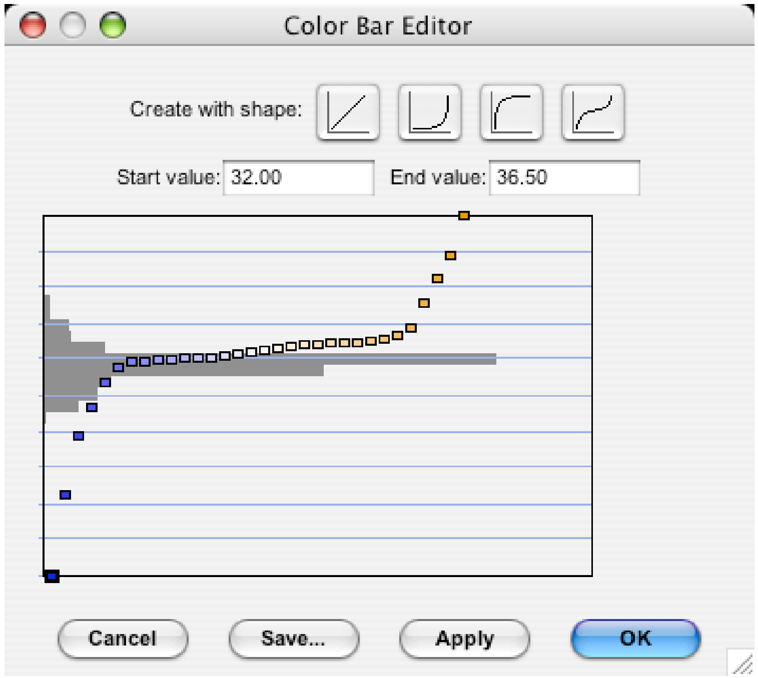
Click-holding on any colored symbol in the graph-like color/contour display (in the center of the *Value editor/preview* segment of the dialog) permits one to change the value associated with that colored symbol by moving the mouse up or down. Moving the mouse left or right automatically selects other colored symbols in this display, permitting one to click-hold-drag with the mouse to reassign values (and shape of values) to all or any selected portion of the color/contour bar.

Clicking on any color swatch in the column of assigned colors (shown to the right of the *Value editor/preview* segment of the dialog) brings up the standard color picker dialog, from which any new color can be assigned to the associated contour. [See the Color Palette Editor description in the User Guide for instruction on using the color picker dialog.]

Clicking on the *Color Palettes* menu will display a pop-up menu of all available color palettes

Color Bar Editor

The Java OceanAtlas Colorbar Editor is opened by double-clicking (or right clicking) on any colorbar legend in Java OceanAtlas:



The graph provides a representation of the distribution of the color intervals over the total range of the colorbar. Underneath the graph, usually in dark gray on a gray background, is a histogram of the occurrence of parameter values in the data file(s) open in the Data Window over the range of the colorbar. [This can be useful information regarding the need to adjust the colorbar.]

The Colorbar Editor offers the opportunity to temporarily or permanently adjust the lower and upper limits and the intervals, i.e. the 'shape' of the colorbar.

The current *Start value* and *End value* (i.e. the lower and upper limits) are displayed. New ones can be entered.

The '*Start value* is the lowest numerical value for the desired contour range. It is not necessary that this correspond to the lowest numerical value in the data: all lower-valued data will be colored with the color assigned to the start value.

> Note: The *Start value* will automatically update if contour value changes made during editing (see below) result in a numerically lower value than initially supplied in the *Start value* field.

Similarly, the *End value* is the highest numerical value for the desired contour range. Again, it is not necessary that this correspond to the highest numerical value in the data: all higher-valued data will be colored with the color assigned to the end value.

> Note: The *End value* will automatically update if contour value changes made during editing (see below) result in a numerically higher value than initially supplied in the *End value* field.

Create with shape buttons are used to assign color/contour numerical values over the range of values, by distributing the contours between the start and end contour values according to four pres-set distributions–linear, upper asymptote, lower asymptote, and logistic. The choice of a linear distribution is often fine, but sometimes it is handy to have the contours bunched toward the low, high, or middle portions of the numerical range. Clicking on one of the *Create with shape* buttons will redraw the colorbar, which will be shown in the editor/preview segment of the dialog underneath.

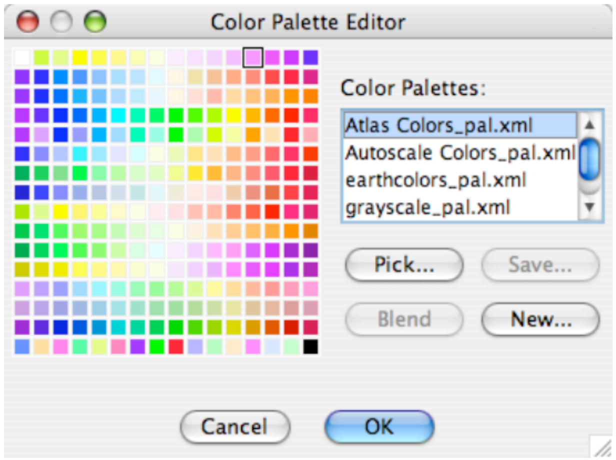
The editor/preview segment of the Color Bar Editor dialog provides not only a view of the color bar, but also provides an interface for adjusting the numerical values of individual color/contour assignments:

> Clicking on any colored symbol in the graph-like color/contour display places the data value associated with that contour in a text field underneath the graph labeled *Value*. This value can be changed in two ways: 1) use the up and down cursor keys to adjust the value or 2) by typing in a new value and clicking on the *Apply* button (or hitting the Return key on the keyboard). The next value in the graph can then be auto-selected by hitting the left or right arrows on the keyboard, and so with a little practice one can progressively move through all the symbols of a color bar set to further customize the values.
>
> Click-holding on any colored symbol in the graph-like color/contour display permits one to change the value associated with that contour/color by moving the mouse up or down. Moving the mouse left or right automatically selects other colored boxes in this display, permitting one to click-hold-drag with the mouse to reassign values (and shape of values) to all or any selected portion of the color/contour bar.

Color Palette Editor

By filling out the desired portions of a color palette with blends and/or picked colors, it is possible to create and save Java OceanAtlas color palettes that can be used to create (or add by editing) color/contour bars with whatever colors one wishes.

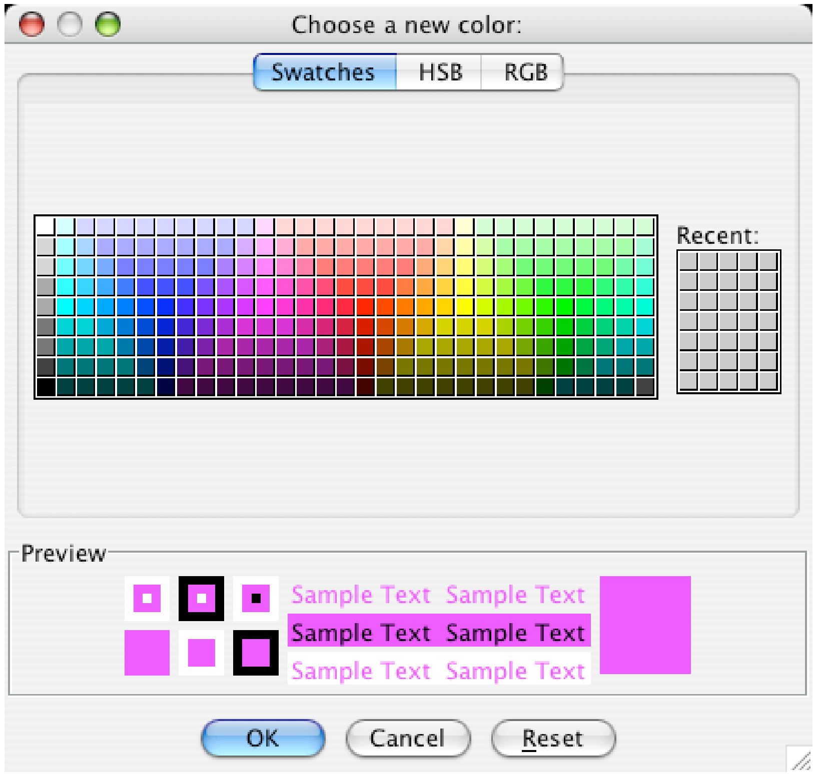
Selecting *Color Table Editor* from the *Resources* menu brings up the Color Table Editor dialog:



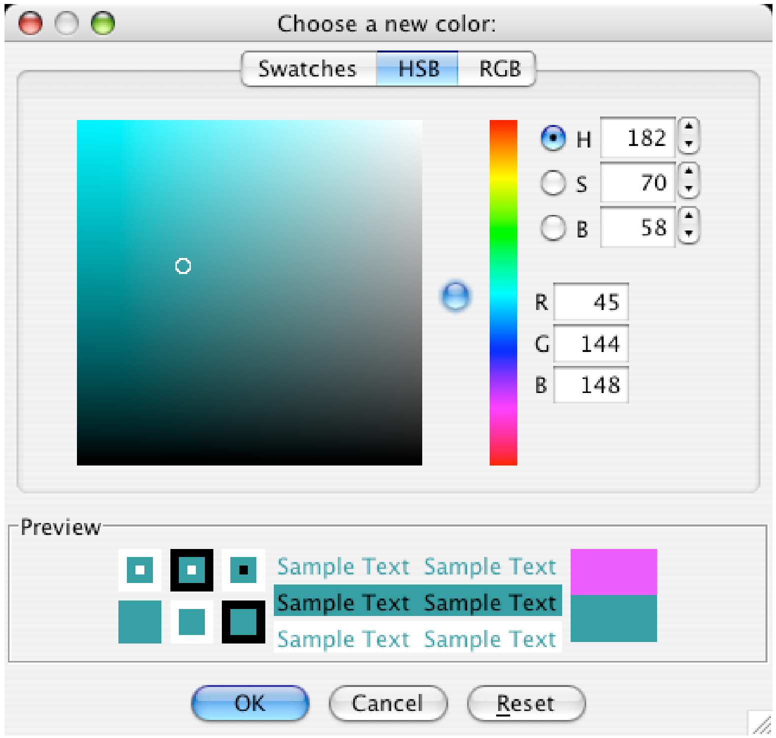
The dialog includes the current 256-color palette (color table), a scrollable list with the names of all current color palettes stored on disk, and buttons for various action choices: *Pick* and *Blend* are explained below; *New* replaces the palette in the dialog with a blank (black) palette; *Save* saves the palette to disk, presenting a dialog to enter in a new name; *Cancel* reverts Java OceanAtlas to its state before opening the dialog; and *OK* closes the Color Palette Editor after one has completed actions.

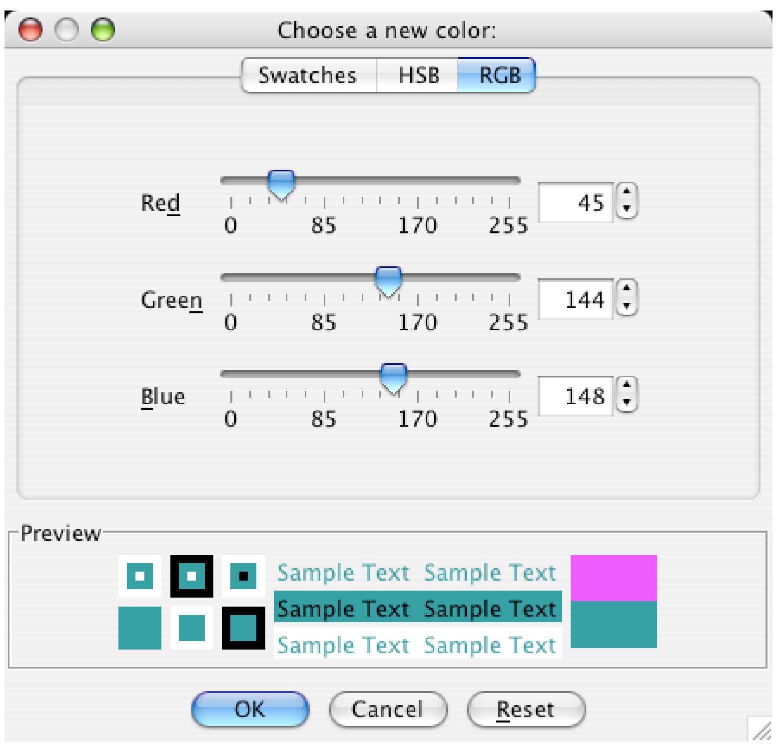
> Note: color palettes cannot be deleted from within the application (a shortcoming that will be fixed in a future version). They can be deleted only by moving them to the trash (or equivalent) from the desktop.

To change a color in a color palette click on any color in the displayed color palette. This will bring up the Color Picker:

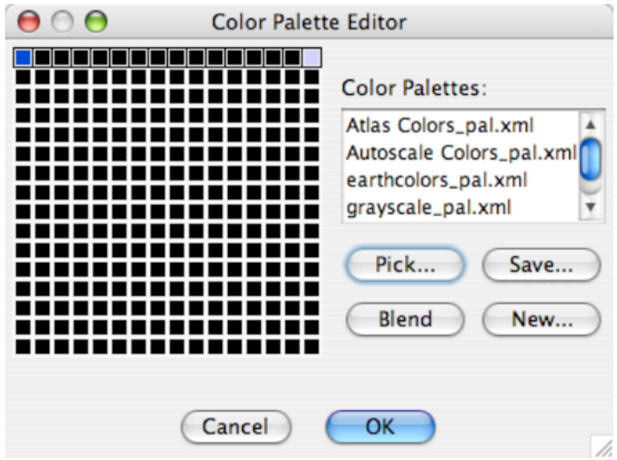


There are three ways to select a new color from the color picker dialog, each selected via a tab near the top: (1) *Swatches* (above) permits clicking on any pre-defined color in the rectangular grid of choices, (2) *HSB* (below) permits choosing a color by adjusting hue angle, saturation, and brightness, and (3) *RGB* (below) permits choosing a color on the basis of red, green, and blue color values.

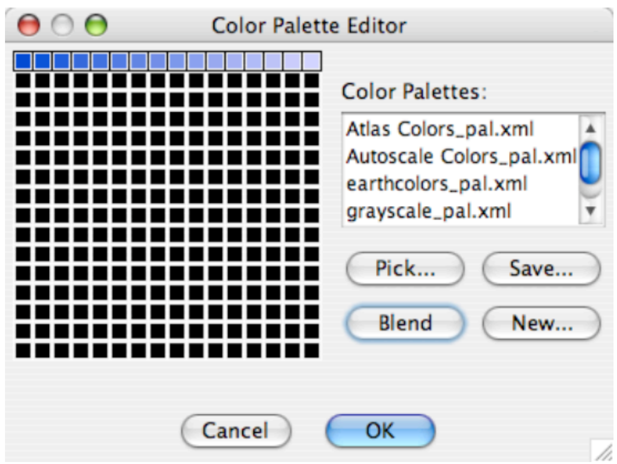




The *Blend* feature is utilized by first clicking on a start color and then shift-clicking on another color to define a color range. You can also click and drag in the palette to select a range of color entries. Either operation will highlight all the colors in the palette between the two selected colors, as in this example:



At this point, clicking on *Blend* will fill the selected color palette space with a uniform blend between the two end-point colors, as in this example:



Surface Editor

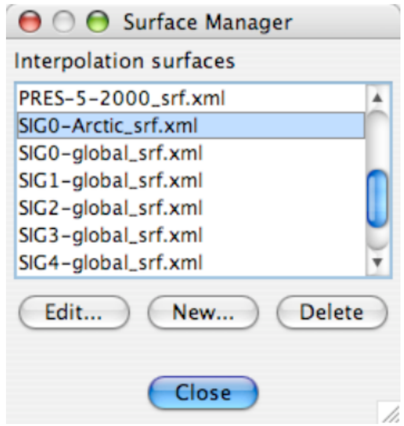
x and y interpolations from the observed data are required before Java OceanAtlas can generate contoured parameter plots. The interpolations are carried out on surfaces (standard values) of the y axis.

Choice of standard levels is an interesting matter. Certainly, too few standard levels and the database – as interpolated – will be distorted. This brings up the rationale behind our empirical choice of standard levels: since the interpolations are linear and 'local', then the ideal list of standard levels, i.e. containing the most information with the least replication, would be those lying on typical data values and bracketing the features of interest.

Fortunately, oceanographers have defined this, in a sense, by their choices of sampling depths on expeditions where bottle sampling with good resolution was a high priority.

Tables of the default standard levels provided with Java OceanAtlas are supplied with the User Guide.

Standard level files can be edited and created with the Java OceanAtlas *Surface Manager*, under the *Resources* menu:

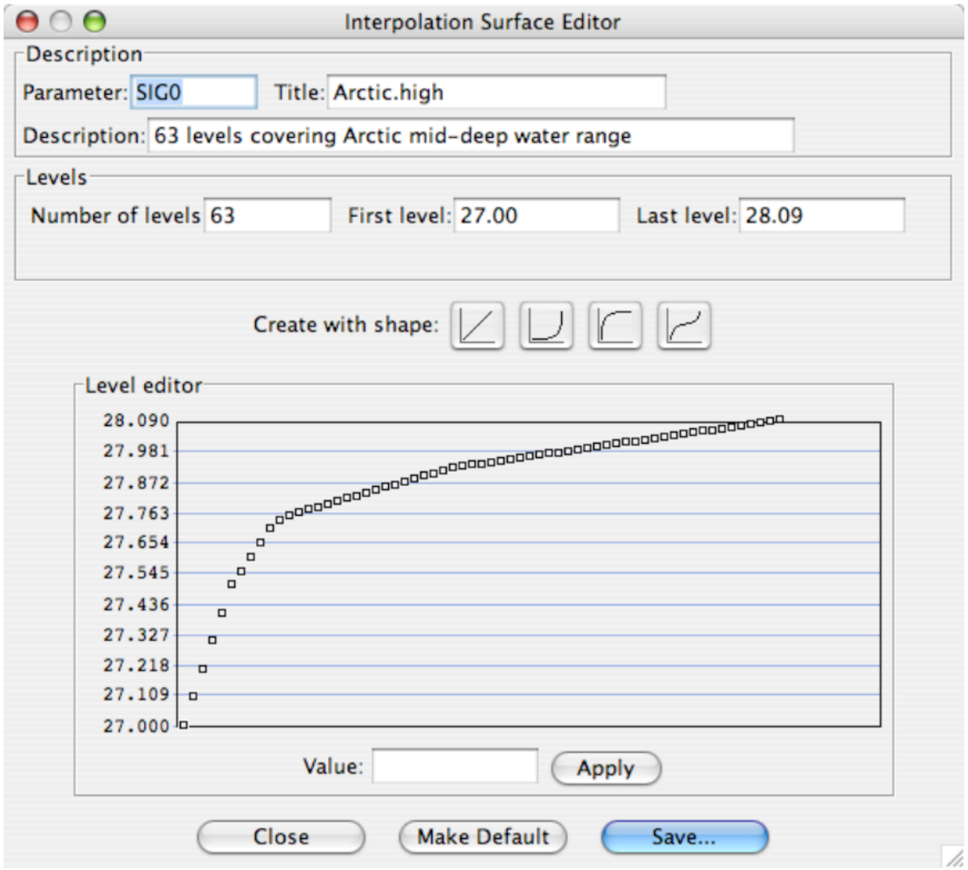


The Surface Manager shows a scrollable list of the available standard surface files, with only the *New* and *Cancel* buttons enabled. Clicking on *Cancel* closes the dialog without further action. Clicking the *New* button will open a blank version of the Surface Editor dialog to create a new set of standard surfaces (see below).

Clicking on one of the choices in the scrollable list will also enable the *Edit* and *Delete* buttons. *Edit* leads to the same dialog that *New* does, but in this case with the parameters for the selected interpolation surfaces file in place for editing. *Delete* removes the selected set of surfaces from the list of available surfaces.

> **Warning**: *Delete* removes the selected surface file ***permanently*** from the disk on your computer, so should be used only as appropriate.

Selecting *New* or *Edit* brings up the Interpolation Surface Editor dialog, shown below for the *Edit* selection of the 'SIG0-global_srf.xml' interpolation surface set (the main differences when selecting *New* are that the text fields are then blank, the level editor area is blank, and different buttons are active.)



The text in the *Parameter* field can be any parameter, for example one of the parameters from the list of available surfaces, any to-be-calculated parameter, or simply *any* user-definable parameter name. The advantage of this over a scrollable list of current parameters is that it lets one create and store interpolation surface sets for parameters not in the current data file.

> Note: The name must *exactly match a parameter name as spelled* in the Data Window (including blank spaces, such as 'O' '2' 'blank' 'blank') or Java OceanAtlas will not recognize it as an interpolation surface for that parameter.

The text in the field labeled *Title* can be whatever one wishes. Generally a short description is best (e.g., Arctic_1).

> Note: The parameter name will automatically become the prefix for the suggested file name when this surface is saved on disk, so one need not repeat the variable name in the *Title* text field.

The text in the field *Description* will appear only in this dialog (but is stored on disk in the surface file). For example it can be whatever text might assist one in recalling the reason the surface set was created. This field can be left empty.

The *Number of Levels* can be whatever one wishes (up to 128).

The *First level* is the lowest numerical value for the desired range. It is not necessary that this correspond to the lowest numerical value in the data.

> Note: The first level value will automatically update if changes made during editing (see below) result in a numerically lower value than initially supplied in the *First level* field.

Similarly, the *Last level* is the highest numerical value for the desired range. Again, it is not necessary that this correspond to the highest numerical value in the data.

> Note: The last level value will automatically update if changes made during editing (see below) result in a numerically higher value than initially supplied in the *Last level* field.

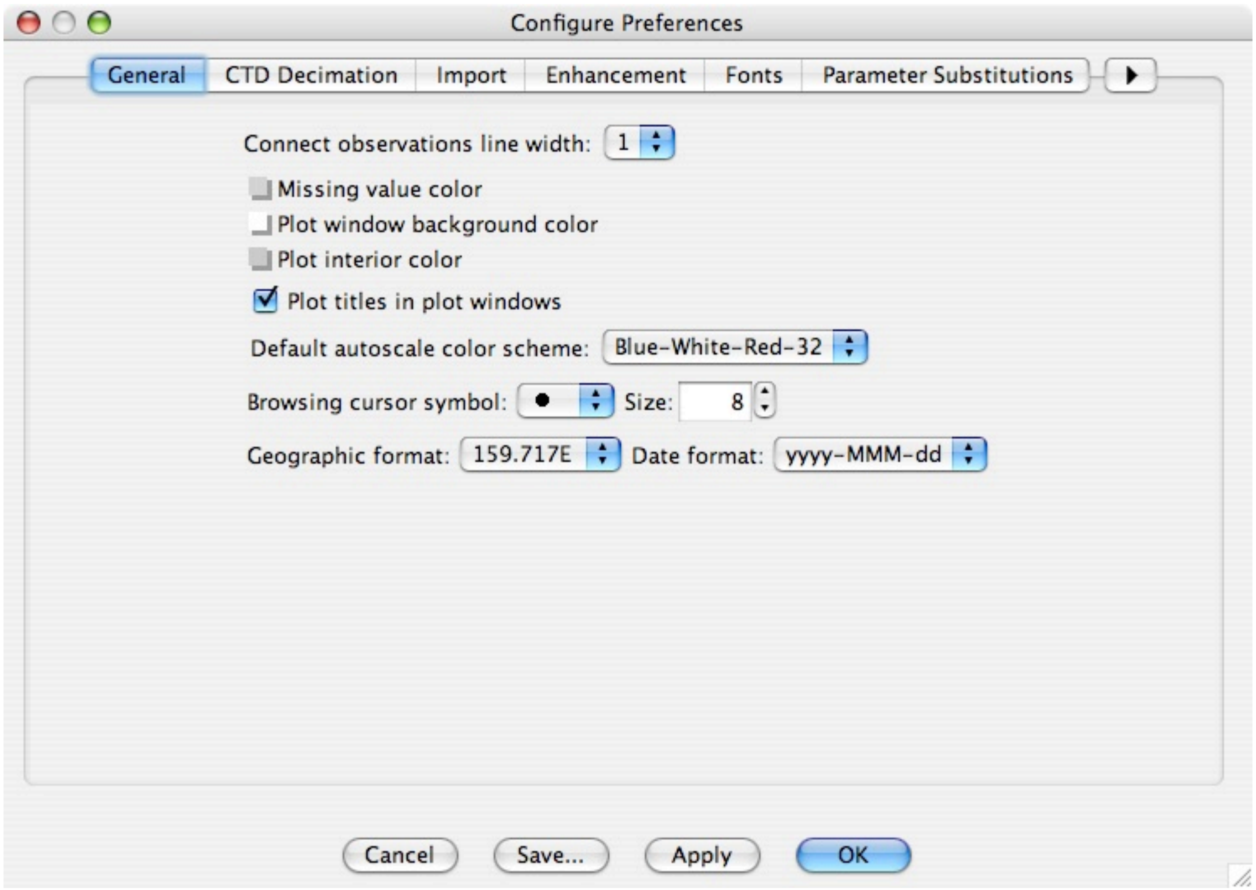
The *Create with shape* buttons are used to define the initial values to assign over the range of values, by distributing the new standard levels between the first and last values according to four pres-set distributions–linear, upper asymptote, lower asymptote, and logistic. The linear distribution is often fine, but sometimes it is handy to have the levels bunched toward the low, high, or middle portions of the numerical range.
When all the elements of the dialog are set appropriately, clicking on one of the *Create with shape* buttons will create the standard surface set, which will be previewed in the 'Level editor' panel of the dialog.

The *Level editor* panel of the dialog provides not only a view of an existing or newly created/edited standard level set, but once a set has been created or chosen for editing it also provides interfaces for adjusting the numerical values of individual level assignments:

> Clicking on any plot symbol in the graph-like level display places the data value associated with that level in a text field underneath the graph labeled *Value*. This value can be changed in two ways: 1) use the up and down cursor keys to adjust the value or 2) by typing in a new value and clicking on the *Apply* button (or hitting the Return key on the keyboard). The next value in the graph can then be auto-selected by hitting the left or right arrows on the keyboard, and so with a little practice one can progressively move through all the levels of a standard level set to further customize the values. This technique is the most reliable method for assigning user-specified numerical values to each interval in a level. (It is how nearly all the values in the surface sets supplied with Java OceanAtlas were created.)
>
> Click-holding on any symbol in the graph-like level display permits one to change the value associated with that level by dragging the mouse up or down. Moving the mouse left or right automatically selects other symbols in this display, permitting one to click-hold-drag with the mouse to reassign values (and shape of values) to all or any selected portion of the level surface set.

Preferences

Java OceanAtlas preferences are set via the *Preferences* dialog under the *Edit* menu:



The current choices for all three panels are only in effect for the current Java OceanAtlas session, unless explicitly saved by clicking the *Save* button. Selecting *Cancel* exits the dialog without putting any changes into effect. *Cancel* is also used to exit the dialog after changes have been saved.

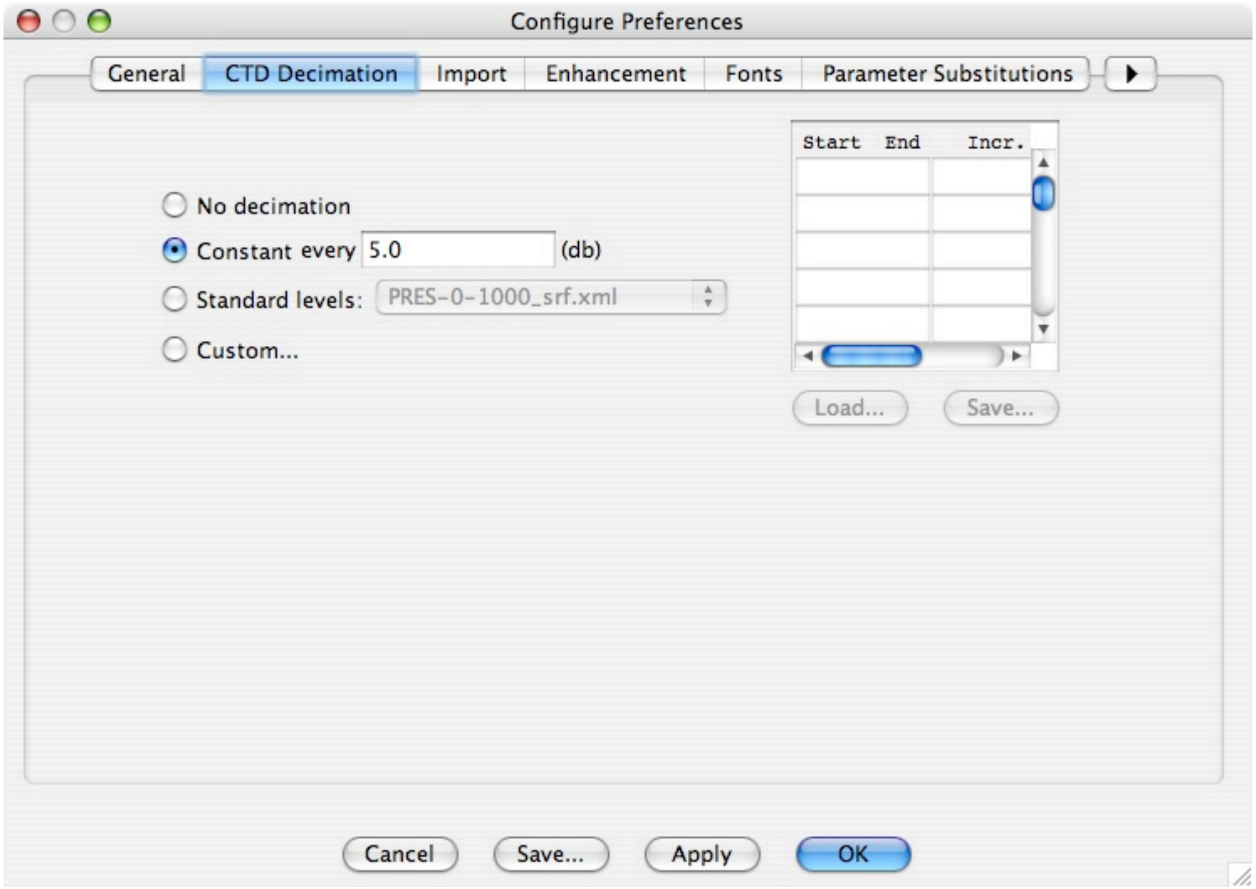
There are 8 tabbed panes displayed in the Preferences dialog.

General Preferences (shown above):

- choose the width of the line used when *connect observations* is chosen on a property-property plot,
- choose the default colors used to represent missing values, plot backgrounds, and plot interiors,
- choose the default color scheme to use for autoscaled color bars and legends,

- Choose the browsing cursor symbol and size
- choose the display formats for dates and geographic locations. For locations, there is a choice of decimal degrees and degrees-minutes-seconds formats. For dates, there is a variety of display formats.

CTD Decimation Preferences:



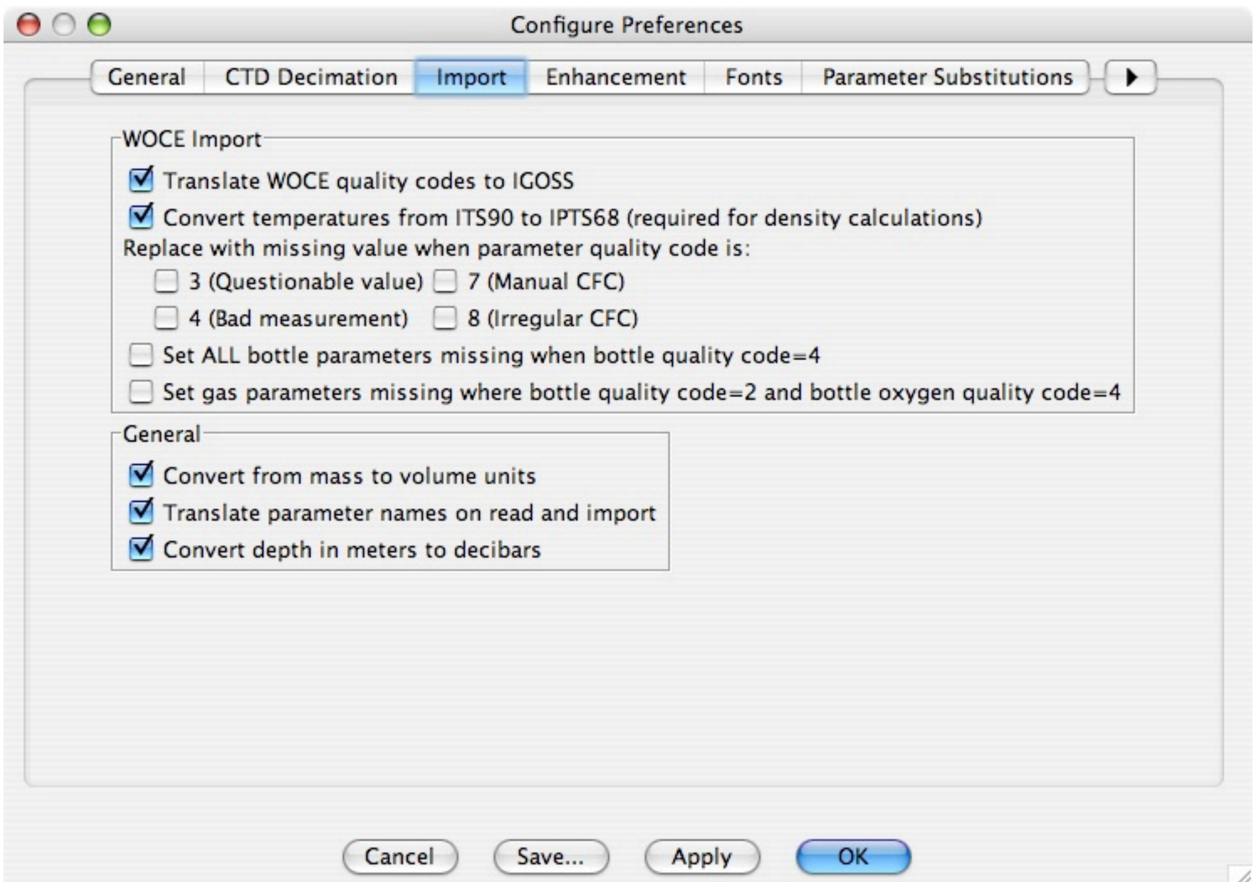
These options control Java OceanAtlas' handling of CTD data when importing from one of the supported CTD data formats. CTD data are bulky and can pose significant memory constraints when using JOA. Consider that a typical deep ocean WOCE Hydrographic Program CTD profile will contain one record every two decibars, or 2500 lines of data. A WOCE Hydrographic Program bottle data file typically has about that many lines of data in an entire 100-station cruise. It is easy to exceed memory limitations for just one cruise of CTD data in native resolution.

- When *No decimation* is selected, Java OceanAtlas imports the CTD data file in its original level of detail.

Java OceanAtlas can reduce the size of imported CTD data files via three different decimation methods:

- Intervals at user-chosen constant pressure intervals can be specified,
- Intervals corresponding to the standard pressure levels on a selected pressure surface set can be specified (the pressure surface sets are those used for interpolating data onto contour plots; see User Guide entry for 'Surface Manager'), or
- User-chosen decimation values for custom pressure intervals can be specified. In the example shown in the figure above, CTD observations are being imported at 2-decibar resolution from 0 to 200 decibars, at 5-decibar resolution from 200 to 500 decibars, and at 10-decibar resolution from 500 to 6500 decibars.

Import Preferences



On the *Import* tabbed pane are options to handle the import of bottle and CTD data from WOCE Hydrographic Program 'WHP-Exchange' data files (_hy1.csv, _ct1.csv, and _ct1.zip files) and general import options.

Translate WOCE quality codes to IGOSS will, if selected, translate the WOCE Hydrographic Program's quality codes to the more commonly used WMO IGOSS quality codes. (See User Guide entry for WOCE and IGOSS quality codes.)

Convert temperatures from ITS90 to IPTS68 permits Java OceanAtlas to use the temperature scale required in the version of the equation of state in use through the 1980s to the time this entry was written (the IPTS68 temperature scale) by converting temperatures defined by the ITS90 temperature scale used to measure temperature and calibrate instruments since 1990 (and used as the temperature definition in WOCE Hydrographic Program data). This is a subtle matter at oceanographic temperatures.

The remaining WOCE Import options have to do with replacing bad or questionable data with the Java OceanAtlas' missing value designator (-99). Some Java OceanAtlas users will prefer to see only 'good' data, but some will wish to be able to view 'questionable' or 'bad' data. [WOCE Hydrographic Program data files contain all measured values, even those proven to be bad measurements. This choice was made by the planners of that program.] Java OceanAtlas provides several means of using the quality code information, for example in the data lists and section display in the Data Window, and WOCE data can be filtered by quality code.

General Import options include:

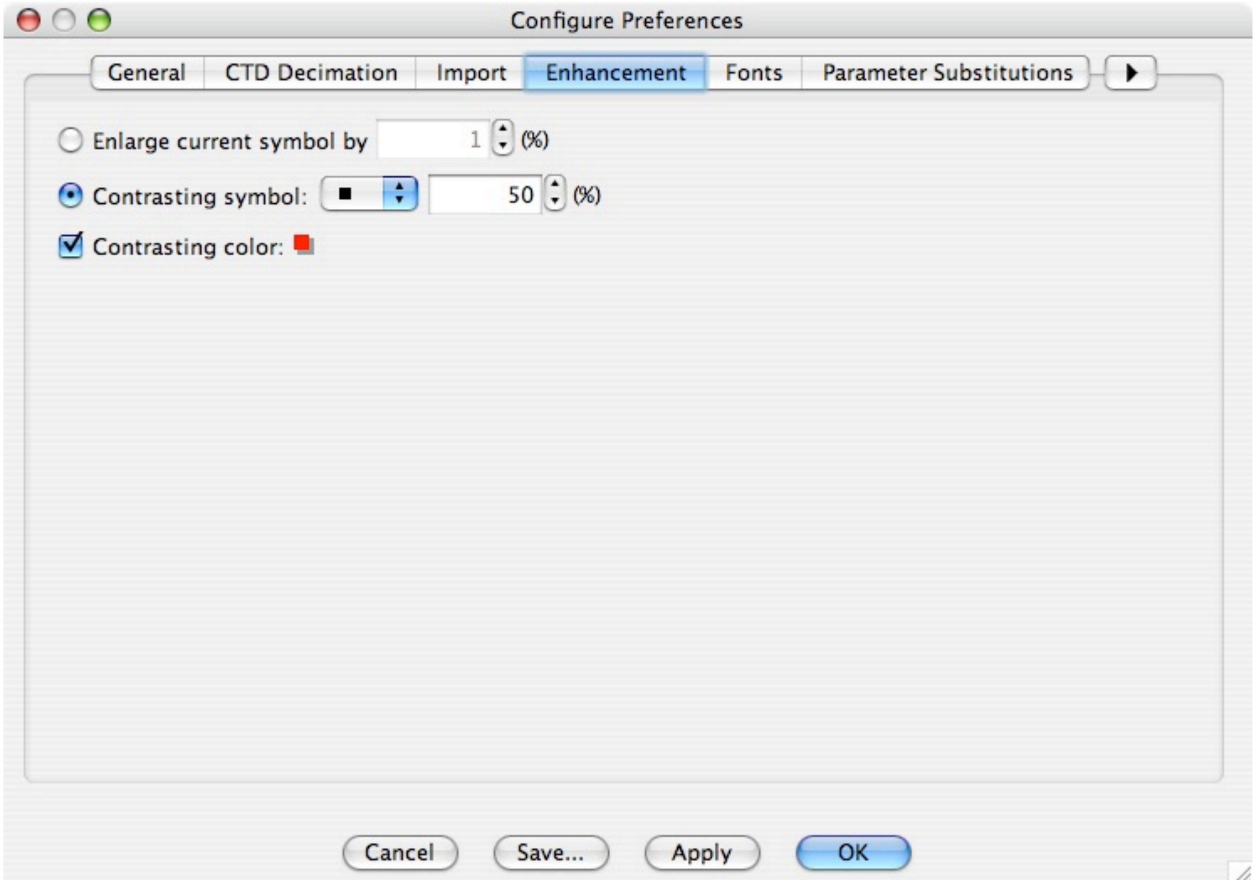
Convert from mass to volume units, when selected, converts the 'mass' units for dissolved oxygen and nutrients used by the WOCE Hydrographic Program (μM/kg) to the traditional 'volume' units (ml/l for O2 and μM/l for nutrients) preferred by some oceanographers, such as the author of this User Guide. For nutrients, the difference factor is only density, which at a typical seawater density of about 1.027 kg/l, is about 3%. [Note: The 3% difference is small enough that it is not feasible to learn the units of nutrient data from examination; mislabeled nutrient data are distressingly common.] But for dissolved oxygen an additional factor enters, making the raw number difference in expressed values different by a factor of approximately a factor of 44.

Check this option to Translate parameter names on input to the JOA parameter name lexicon.

Check *Convert depth in meters to decibars* if the input data are indexed by depth in meters instead of pressure. JOA will convert depth to pressure using a built-in algorithm.

Plot Enhancement Preferences

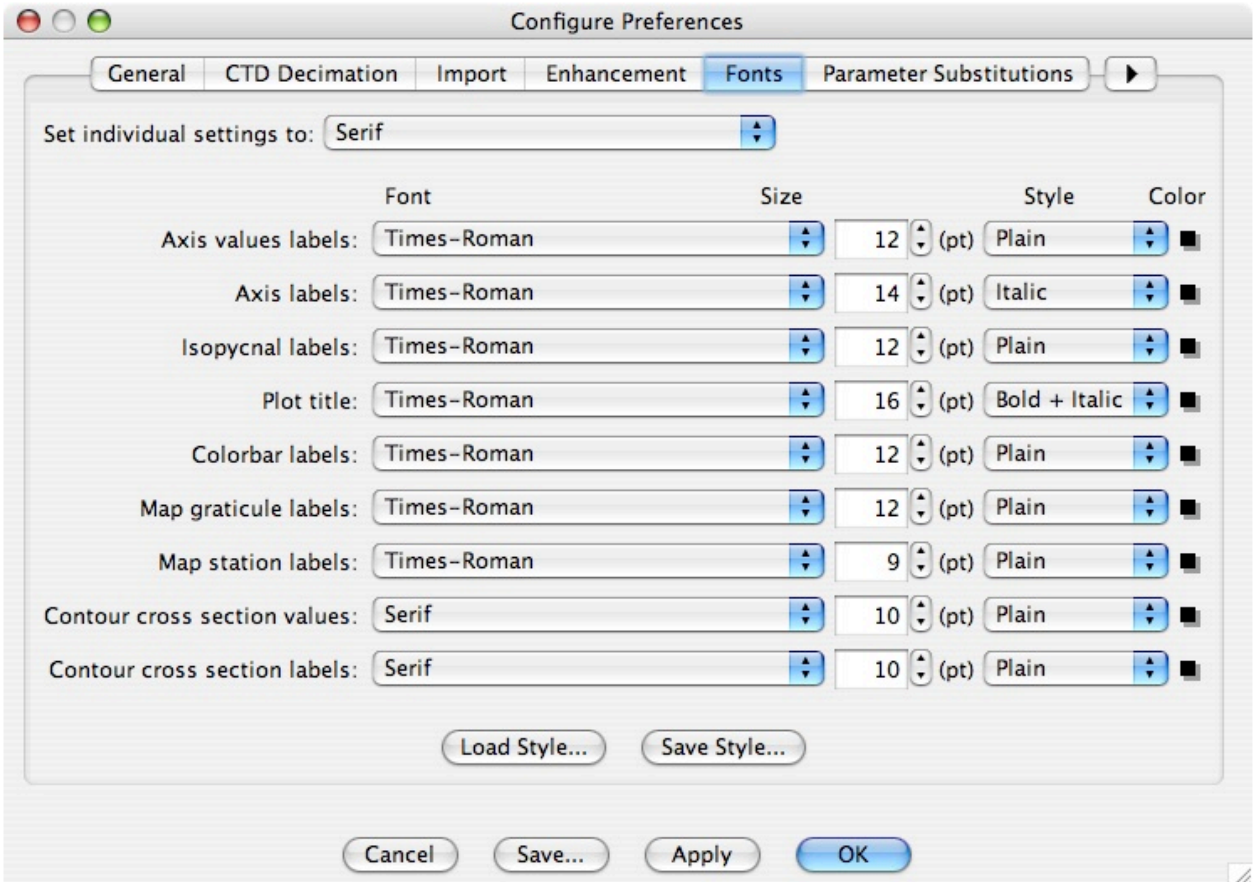
Plot enhancement allows user to highlight points in property-property plots and profile plots by selecting a range of values in a colorbar legend. These preferences control how JOA highlights values that fall in the selected enhancement range:



The default action for enhancement is to enlarge plot symbols by 50%. If this enhancement is subtle you can increase the enlargement value or enhance by choosing a contrasting symbol. In either case, you can also choose to override JOA's choice of symbol color with a contrasting color of your choice.

Font Preferences

Java OceanAtlas uses high-quality typefaces for drawing all plot labels and titles. This panel allows you to set custom type preferences for the various text labels rendered by JOA:

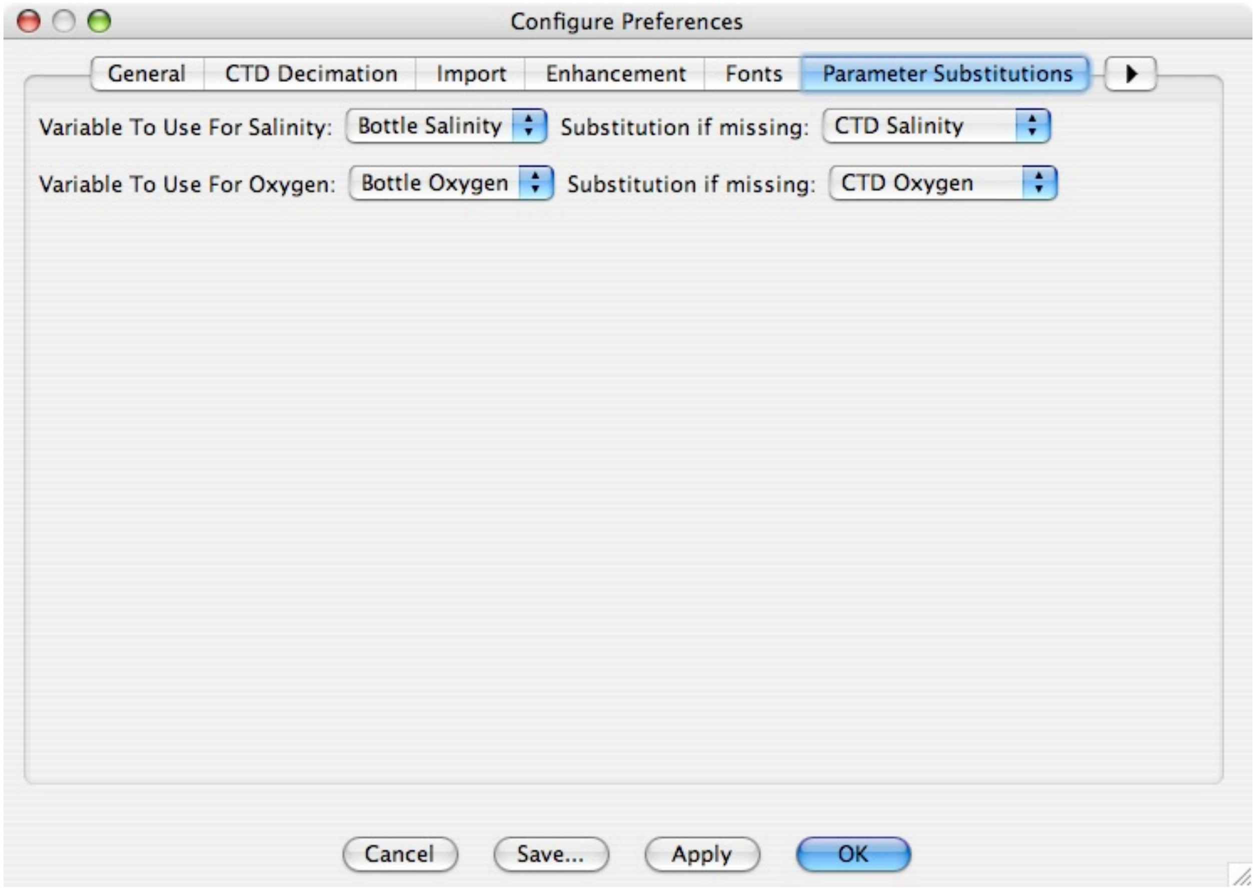


For each type of label, you can choose a font, font size, style (plain, bold, italic, bold+italic), and a color. You can set all the individual labels to the same font by selecting from the *Set individual settings to* menu. Note: this will show you all the fonts installed on your computer—not all fonts can be rendered by JOA in all the possible styles. It is advisable to experiment with obscure or “fancy” fonts.

All of the individual settings can be saved to “style” files on disk. In this way it is possible to have styles for screen use of JOA and styles for printed output from JOA. In addition, the panel settings are saved to the preferences file as the default style. Note: if you change font settings and then click *OK*, all existing plots will be redrawn to reflect the changes.

Parameter Substitutions

This panel allows the user to define standard substitutions for salinity and dissolved oxygen used in calculations on bottle data. Typically, you would choose to use the CTD parameter, if available, when the bottle parameter is reported missing. In addition, this also allows defining the default salinity or oxygen used in calculations. For example, you could choose to use the CTD salinity in place of all bottle salinities when performing calculations because the bottle values might be suspect.



Database Setup and Database Calculations

These options are described in Appendiz xyz.

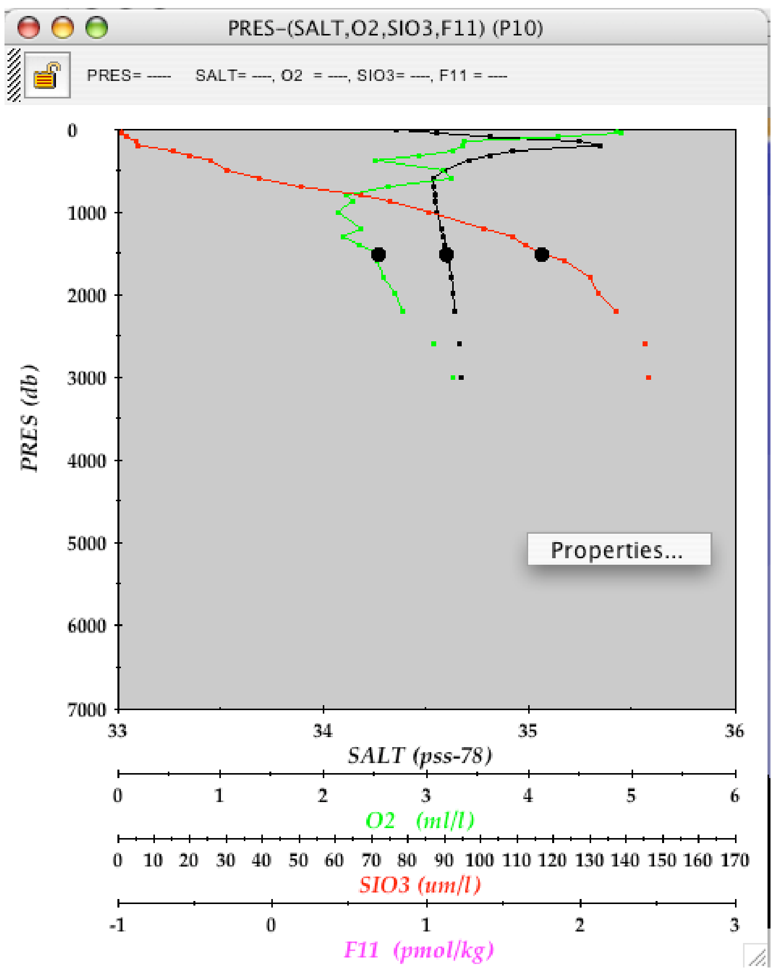
Command key shortcuts

Command/control key shortcuts are implemented for most menu items [press the ‘command’ key on Mac OS X computers and the 'control' key on other computers) and, while still holding that key, the indicated letter key]. These are listed below, or easily viewed by pulling down each menu:

Pressing ‘cmd/ctl’ and this letter	Gives this result
O	Open data file
A	Add data file to open data file(s)
P	Print foreground plot
Q	Quit Java OceanAtlas
M	Map plot
F	Profile plot
Y	Property-property plot

1	Contour Parameter plot
2	Contour Gradient plot
T	Parameter Calculations
K	Contour Manager
R	Edit settings for current plot; this is a good way to edit a plot without moving the data cursor

Also under Mac OS X, pressing the control key while the mouse pointer is over any area of a plot, and then clicking the mouse, will bring up a 'Properties' menu that, if clicked upon, will bring up the relevant plot setup dialog. Under Windows, UNIX and Mac OS installations with a two-button mouse, clicking the right mouse button displays the Properties menu. See example below of a plot showing the 'Properties' menu:



The only way to avoid moving the browsing point when bringing up the plot setup dialog is to use the *Edit Plot Settings* command in the *Edit* menu..

Moving Customized Resources to New Versions of JOA

With the exception of color palettes, all JOA resources (colorbars, interpolation surfaces, map settings, decimation settings, and general preferences) are saved in the XML format. XML has the advantage that XML files are readable (and editable) by humans and are extensible while maintaining backwards compatibility. Your customized resources will still be useable in future versions of JOA even if new settings are added to the resources. If you choose to resave your resources in a future version of JOA then your resources will ”inherit“ any new settings. Currently JOA maintains the following types of user-changeable resources:

Preferences: Master preference settings for the JOA application. Stored here are decimation, WOCE import settings, enhancement settings, the default text style sheet as well as other more generalized settings. Preferences are maintained in the file *joaprefs.xml*. There is only one instance of the JOA preferences file.

Default Map Settings: This file contains the initial settings for any new maps created by JOA. This file stores all the settings manipulated in the Map setup dialog. Map settings are stored in the file *JOADefault_map.xml*. There is only one instance of this file.

Custom Map Settings: These files contain settings for customized maps that you may wish to recreate without remembering the individual settings. These files are named *<settings name>_map.xml*, where you provide the portion of the file name between the angle brackets (for example, *bering_sea_map.xml)*. There can be as many of these resources as required.

Decimation Settings: These files contain settings for customized CTD decimation schemes. These files are named <settings name>_deci.xml, where you provide the portion of the file name between the angle brackets (for example, shelf_deci.xml). There can be as many of these resources as required.

Style Sheets: These files contains settings for typefaces and styles used to draw JOA’s plot labels and titles. These files are named <settings name>_style.xml where you provide the portion of the file name between the angle brackets (for example, *AGU_style.xml)*. There can be as many of these resources as required.

Observation Filters: These files contain settings for customized observation filters. These files are named <settings name>_obsf.xml, where you provide the portion of the file name between the angle brackets (for example, AAIW_obsf.xml). There can be as many of these resources as required.

Colorbars: Colorbars are at the heart of JOA's use of color. They are mappings between values of a measured (or computed) parameter and a color scheme. They are used to color property plots and determine the placement of contours in contour plots. JOA comes with a default set of colorbars designed to cover the ranges of many parameters in the world's oceans. These files are named *<settings name>_cbr.xml*, where you provide the portion of the file name between the angle brackets (for example, *NUTR-global_cbr.xml)*. There can be as many of these resources as required.

Interpolation Surfaces: Interpolation surfaces are used to define a regular grid of an index parameter (typically pressure or density) that observed values are interpolated onto in preparation for contouring. JOA comes with a default set of surfaces for a variety of pressure and density ranges. These files are named *<settings name>_srf.xml*, where you provide the portion of the file name between the angle brackets (for example, *THTA-global_srf.xml)*. There can be as many of these resources as required.

Color Palettes: Color pallets are 16 by 16 grids of colors used primarily in creating color bars. JOA comes with a default set of palettes that store a variety of pre-made color ramps. These files are named *<settings name>_pal.xml*, where you provide the portion of the file name between the angle brackets (for example*, 16-shade blends_pal*.xml). There can be as many of these resources as required

Your custom resources are saved in the *JOA_Support* folder where JOA is installed. To move your custom resources to a new JOA installation, simple copy or move the files to the *JOA_Support* folder in your new JOA folder. If you have modified any of the built-in resources, don't forget to move those as well.

Appendix A: Advanced Features of Java OceanAtlas 5.0

DATA MERGE IN JAVAOCEANATLAS

JOA allows merging arbitrary text data in delimited files to be incorporated into existing data sets.

Data merging typically follows these steps:

1) Open the destination file into JOA.

2) Open the merge source file JOA

3) Preview the parsed merge source file in Merge dialog

4) Optionally edit and recode values found in the source file

5) Create merge criteria and rules to guide merge process

6) Perform merge operation

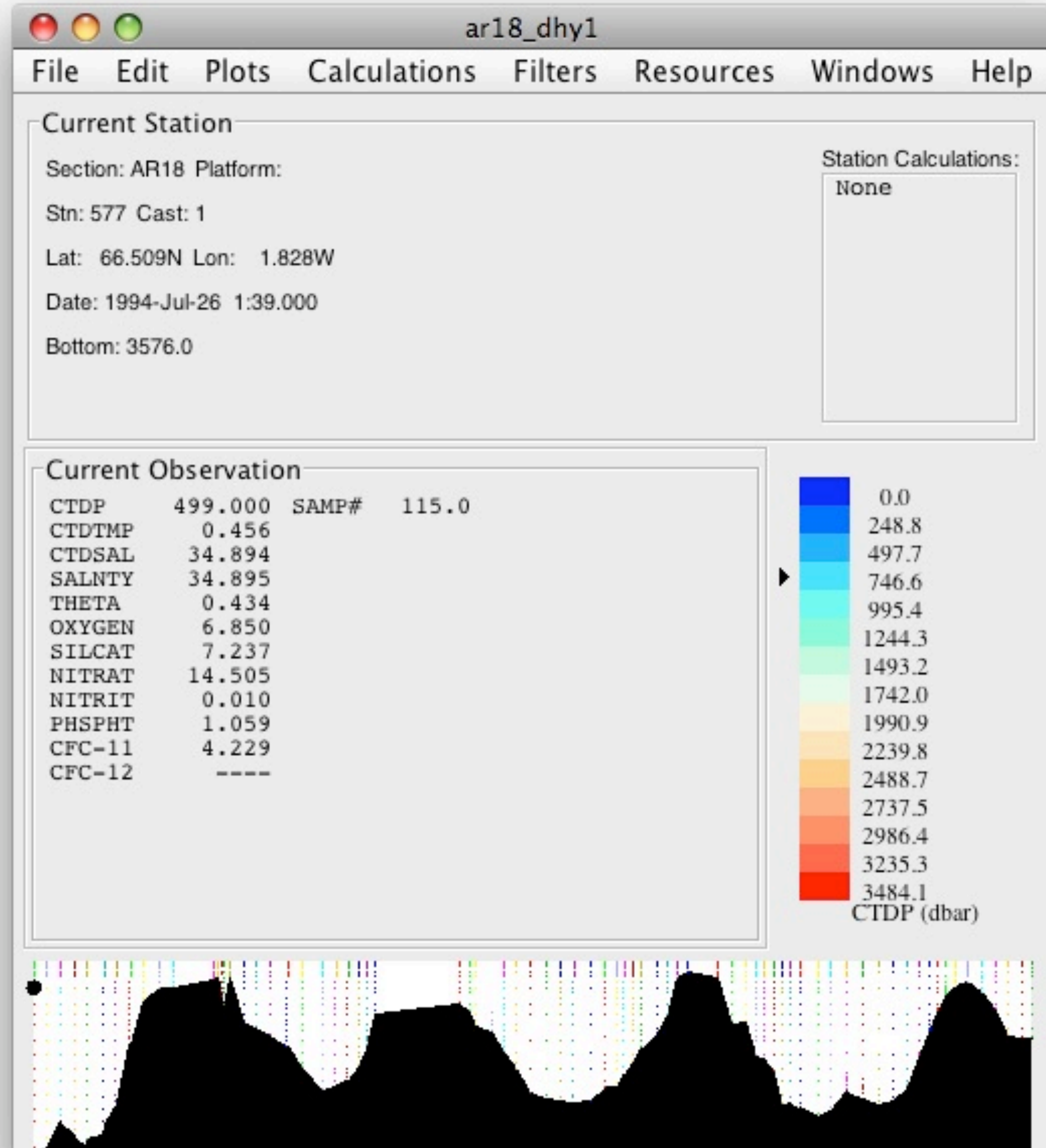
7) View merge log and refine criteria/rules if necessary

8) Save modified file

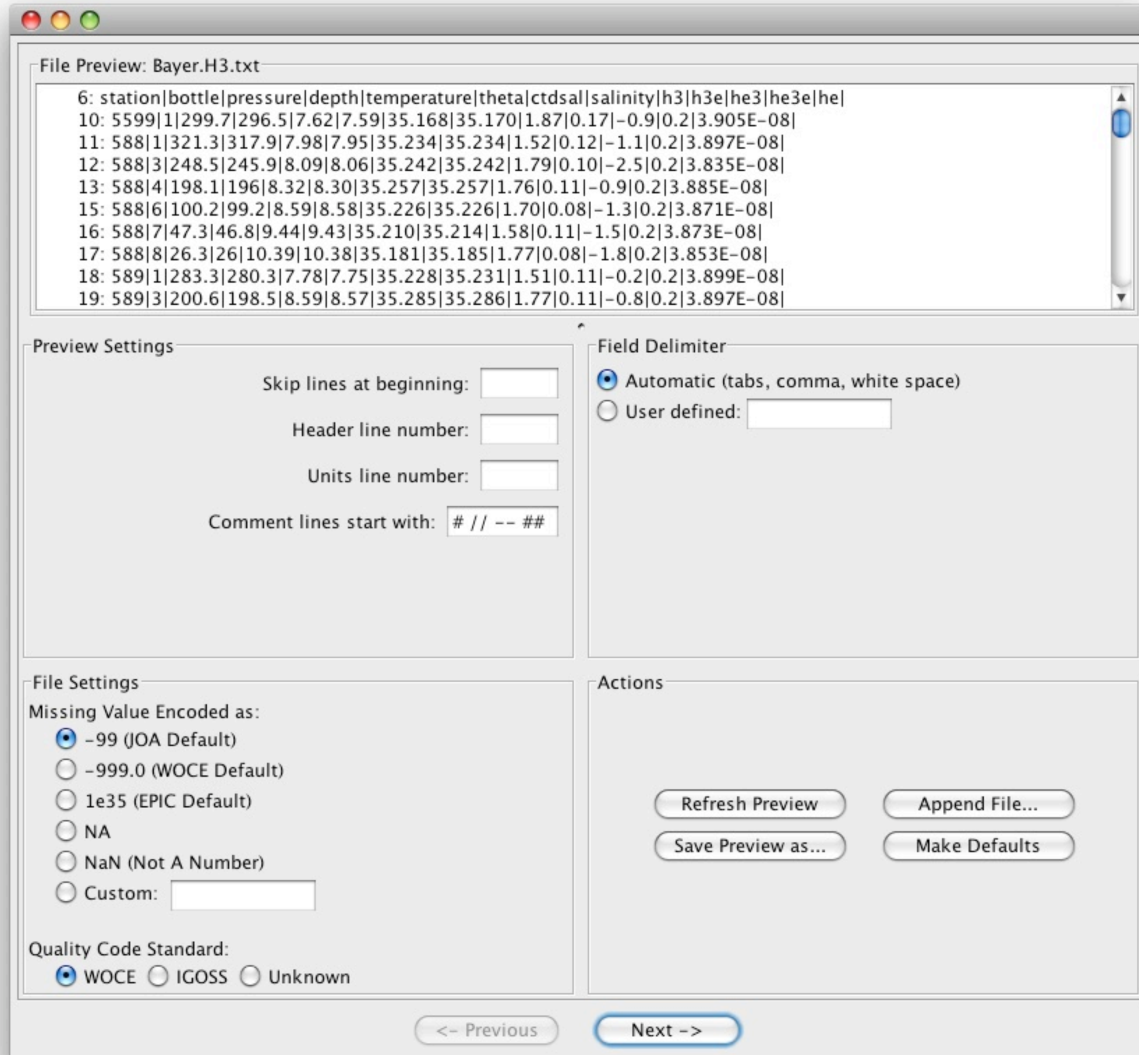
Data Merge Tutorial

This tutorial shows merging a file of updated values into an existing WOCE Exchange file.

Step #1: Open destination file:

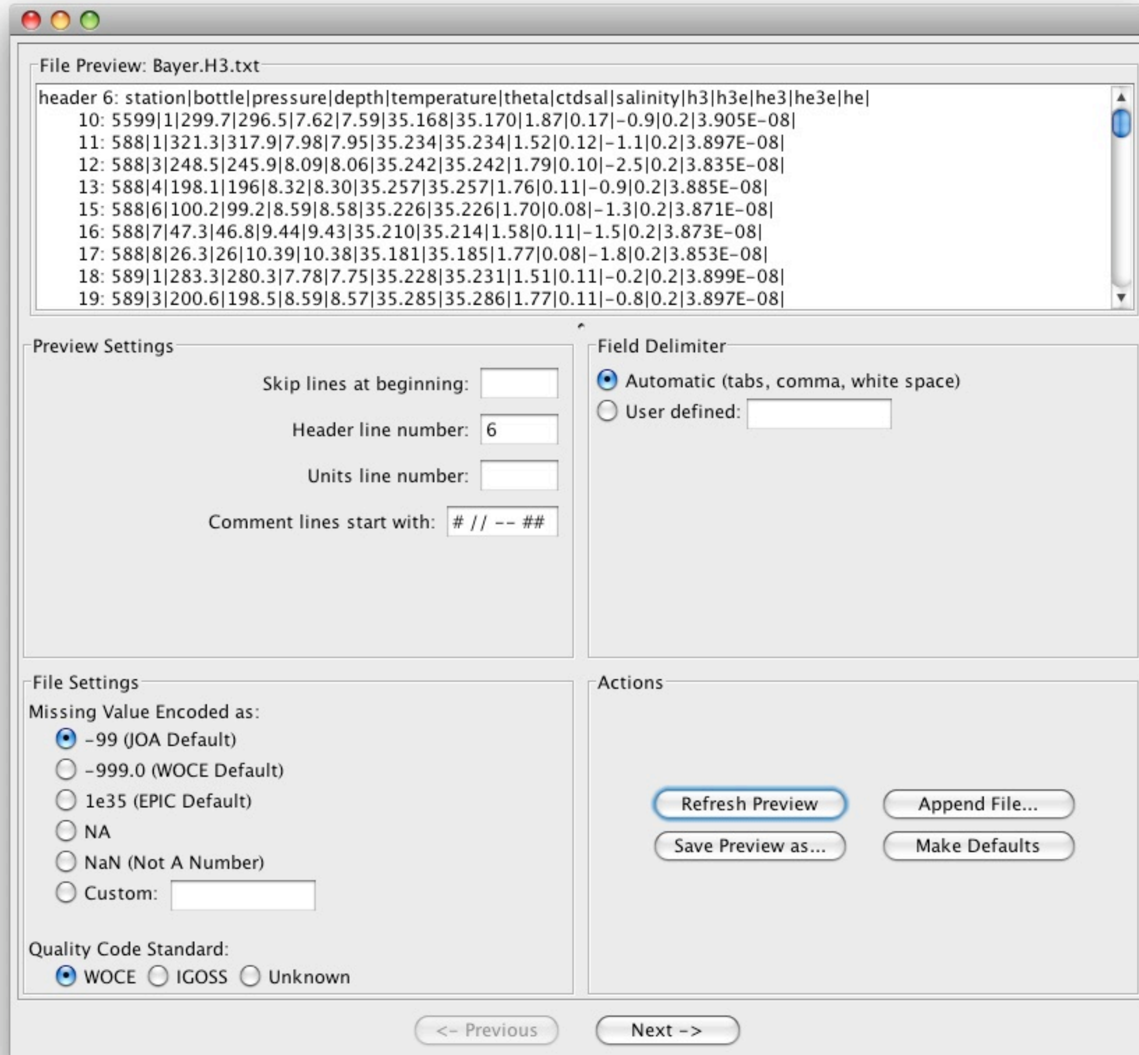


Step 2: Open the merge source by selecting *Merge* from the *File* menu. JOA will open the selected file and parse it for lines and fields. It is assumed that the values in the file are delimited in some way and there is a line that represents the column labels. The column labels are used to define both metadata and parameter values. The metadata are used to find matches in the destination file and the parameters can represent new parameters or parameter values to update in the destination file. Without column headers JOA can not merge the source file into the destination file.



The parsed file is found in the File Preview field. This may look different than the original file because JOA has automatically added a line number, removed blank and comment lines and replaced the delimiter in the file with a vertical bar character ('|'). Usually, using the automatic field delimiter will do a good job breaking the input file into columns. If not, enter a user-defined delimiter and click the *Refresh Preview* button to reparse the input file.

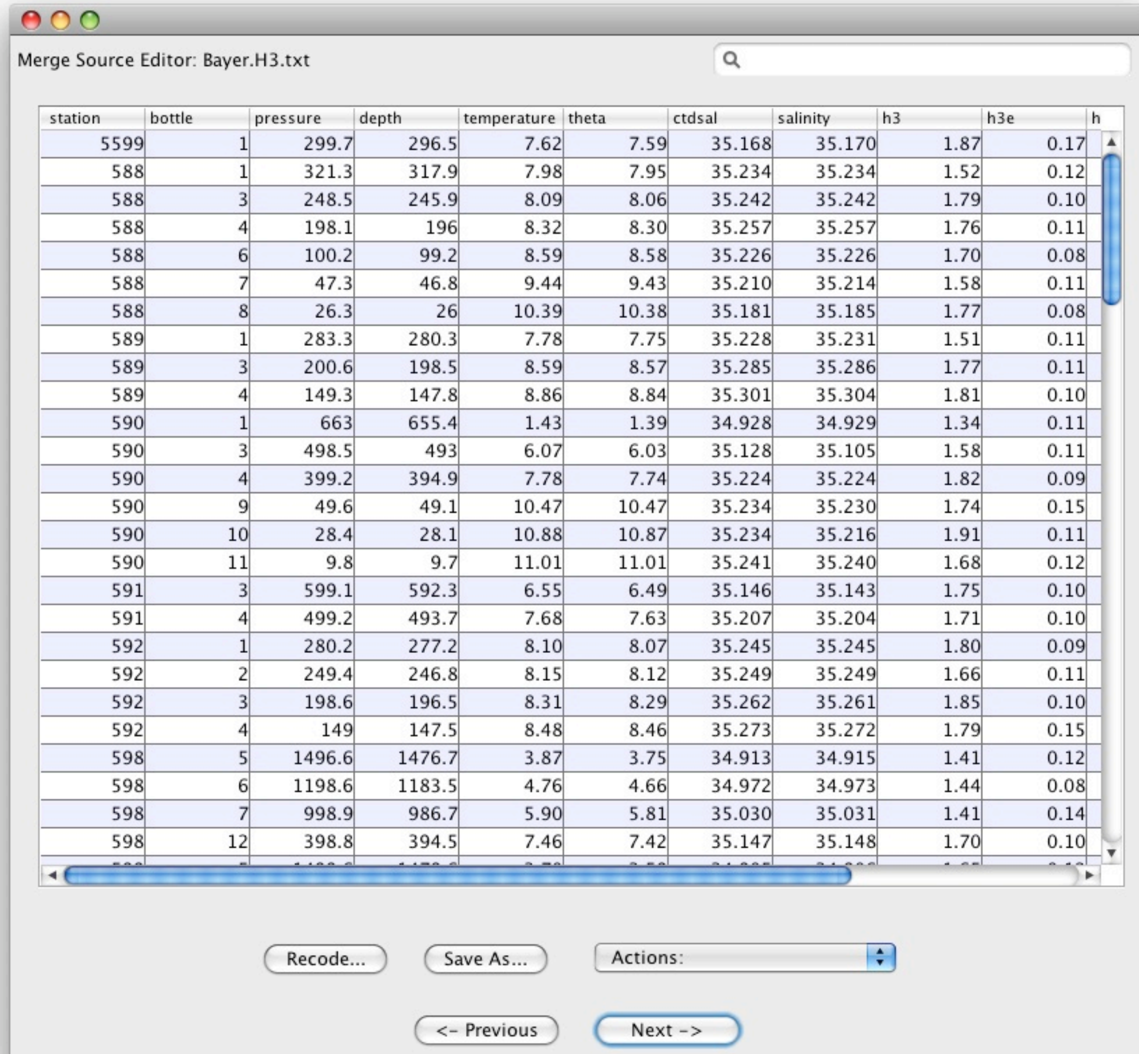
It is important to identify the column header line by entering the line number in the *Header line number* field of *Preview Settings* and clicking *Refresh Preview* to reparse the file:



Do the same if a parameter units line is also included. You can optionally choose to skip any number of lines at the beginning. The *File Settings* for Missing value and quality code convention do not effect parsing the file but inform JOA of these conventions if known.

Other actions available in this panel include appending another source file to the current source file, saving the parsed file in a new file, and making the currently selected options the default settings.

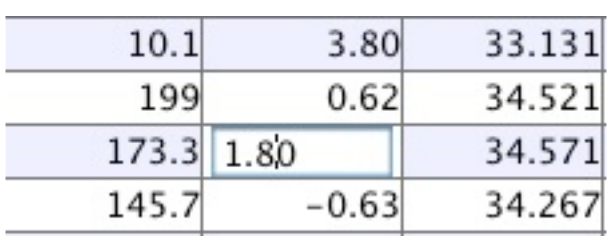
Step #3: Click the *Next->* button to move to the *Merge Source Editor* panel:



station	bottle	pressure	depth	temperature	theta	ctdsal	salinity	h3	h3e	h
5599	1	299.7	296.5	7.62	7.59	35.168	35.170	1.87	0.17	
588	1	321.3	317.9	7.98	7.95	35.234	35.234	1.52	0.12	
588	3	248.5	245.9	8.09	8.06	35.242	35.242	1.79	0.10	
588	4	198.1	196	8.32	8.30	35.257	35.257	1.76	0.11	
588	6	100.2	99.2	8.59	8.58	35.226	35.226	1.70	0.08	
588	7	47.3	46.8	9.44	9.43	35.210	35.214	1.58	0.11	
588	8	26.3	26	10.39	10.38	35.181	35.185	1.77	0.08	
589	1	283.3	280.3	7.78	7.75	35.228	35.231	1.51	0.11	
589	3	200.6	198.5	8.59	8.57	35.285	35.286	1.77	0.11	
589	4	149.3	147.8	8.86	8.84	35.301	35.304	1.81	0.10	
590	1	663	655.4	1.43	1.39	34.928	34.929	1.34	0.11	
590	3	498.5	493	6.07	6.03	35.128	35.105	1.58	0.11	
590	4	399.2	394.9	7.78	7.74	35.224	35.224	1.82	0.09	
590	9	49.6	49.1	10.47	10.47	35.234	35.230	1.74	0.15	
590	10	28.4	28.1	10.88	10.87	35.234	35.216	1.91	0.11	
590	11	9.8	9.7	11.01	11.01	35.241	35.240	1.68	0.12	
591	3	599.1	592.3	6.55	6.49	35.146	35.143	1.75	0.10	
591	4	499.2	493.7	7.68	7.63	35.207	35.204	1.71	0.10	
592	1	280.2	277.2	8.10	8.07	35.245	35.245	1.80	0.09	
592	2	249.4	246.8	8.15	8.12	35.249	35.249	1.66	0.11	
592	3	198.6	196.5	8.31	8.29	35.262	35.261	1.85	0.10	
592	4	149	147.5	8.48	8.46	35.273	35.272	1.79	0.15	
598	5	1496.6	1476.7	3.87	3.75	34.913	34.915	1.41	0.12	
598	6	1198.6	1183.5	4.76	4.66	34.972	34.973	1.44	0.08	
598	7	998.9	986.7	5.90	5.81	35.030	35.031	1.41	0.14	
598	12	398.8	394.5	7.46	7.42	35.147	35.148	1.70	0.10	

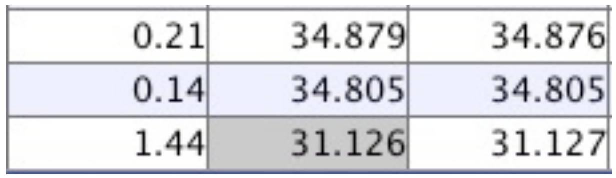
This panel shows the parsed input file in a spreadsheet-like interface that allows editing individual values, saving the edited values to a new file, and performing actions at the column level including adding a new column, renaming column, and deleting a column. In addition values for any parmeter can be recoded according to whether it falls within or outside a user-defined range.

Editing a value: double click a cell to edit it’s value:



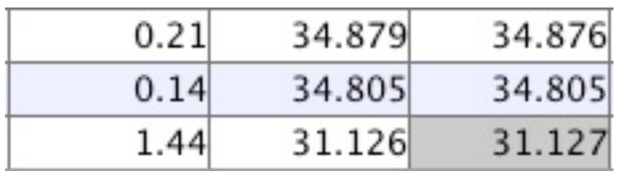
Searching for a value: Click in the search field and enter the text to find:

The matching value (if any) will be highlighted in the display:



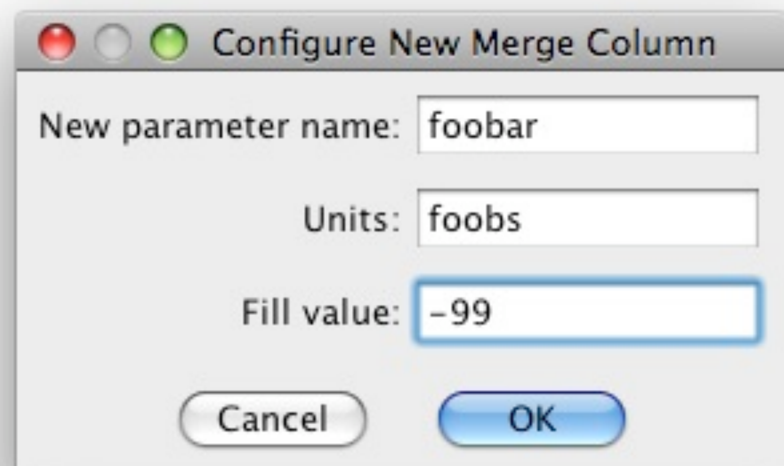
0.21	34.879	34.876
0.14	34.805	34.805
1.44	31.126	31.127

If necessary the spreadsheet will scroll to show you the results. Clicking the tab key will find the next matching field:



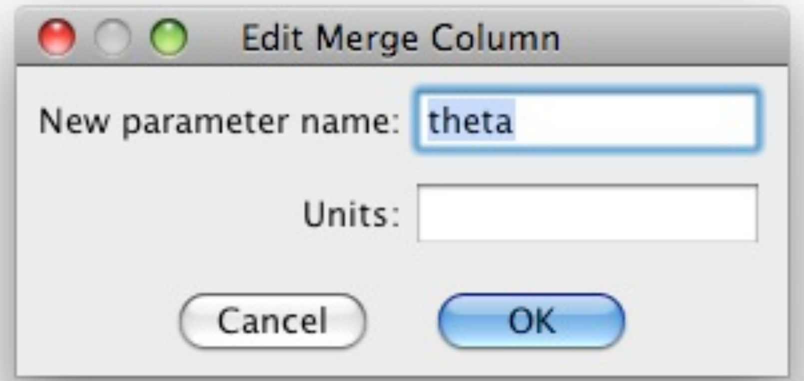
0.21	34.879	34.876
0.14	34.805	34.805
1.44	31.126	31.127

Adding a new column: From the *Actions* popup menu, select *Add Column*:



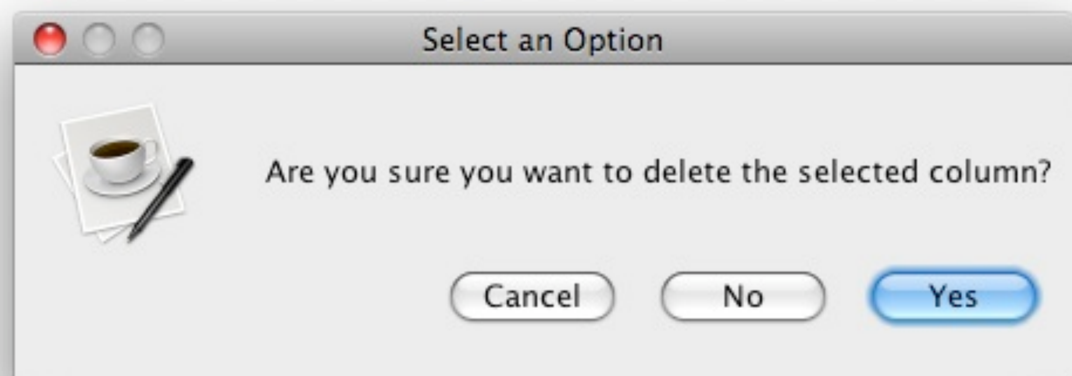
Enter the name of the new column and whether it has units or you want to define a fill value. Click *OK* to add this new column.

Editing and existing column header: From the *Actions* menu, select *Edit Column Names/Units*:



Provide a new column name and optional units and click *OK* to change the titled of the selected column.

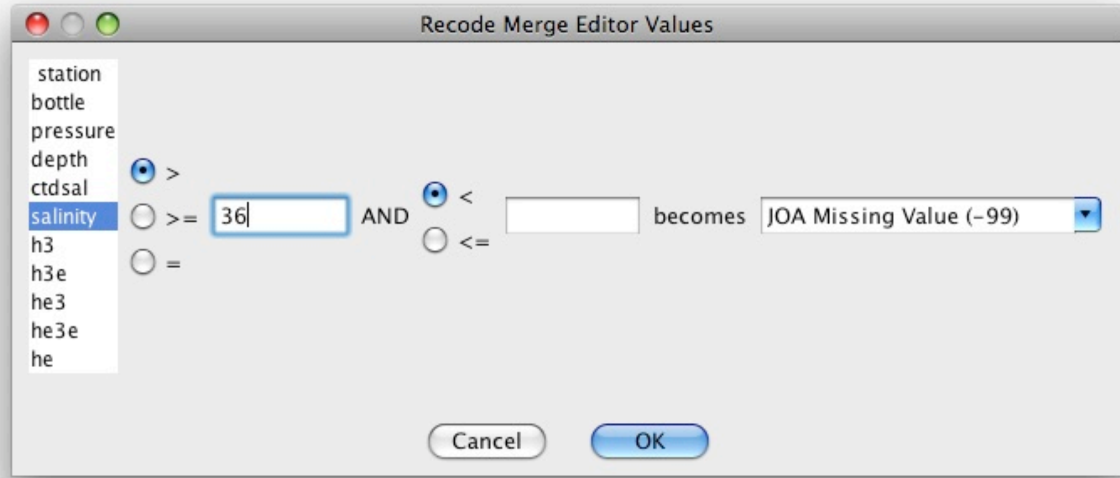
Deleting an existing column: From the *Actions* menu, select *Delete Column*:



Click *Yes* to delete the selected column.

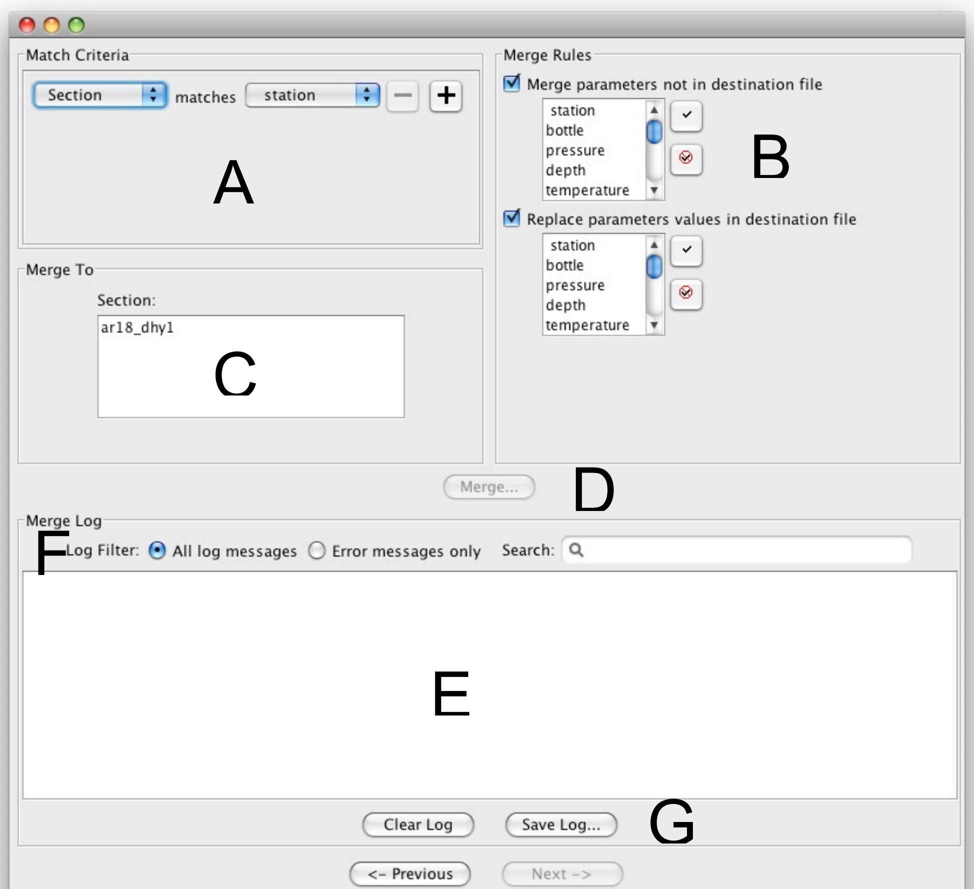
Rearranging the column order: You can rearrange the column ordering by clicking and dragging the column header.

Recoding values: Recoding works on the entire spreadsheet of values. Click the *Recode* button to recode values:



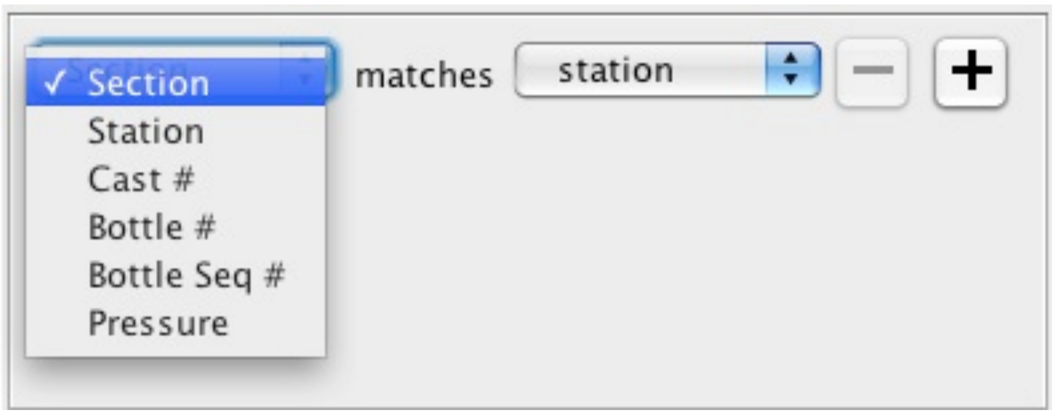
Saving the edited values: Click *Save* to save the current spreadsheet in an external text file.

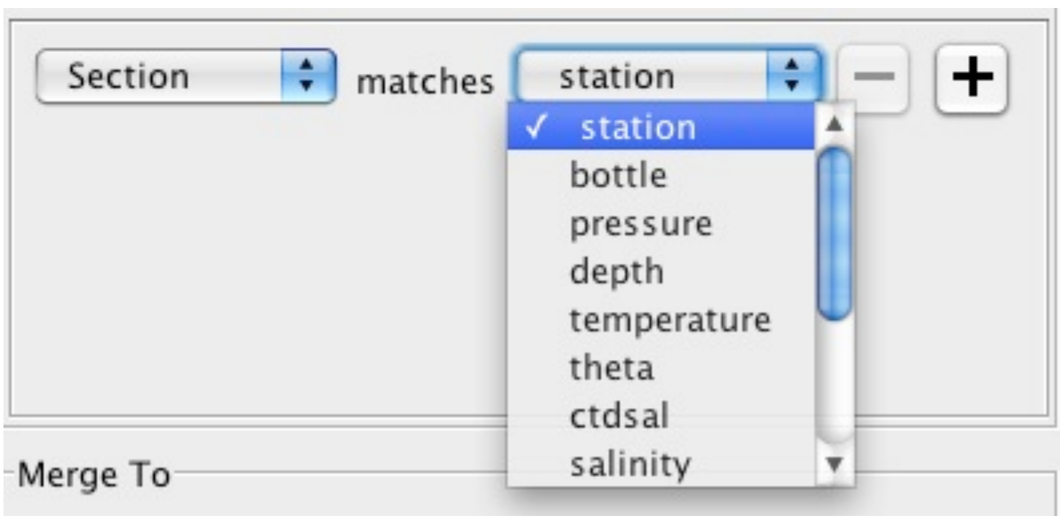
Step #4: Click the *Next->* button to define merge criteria and merge rules:



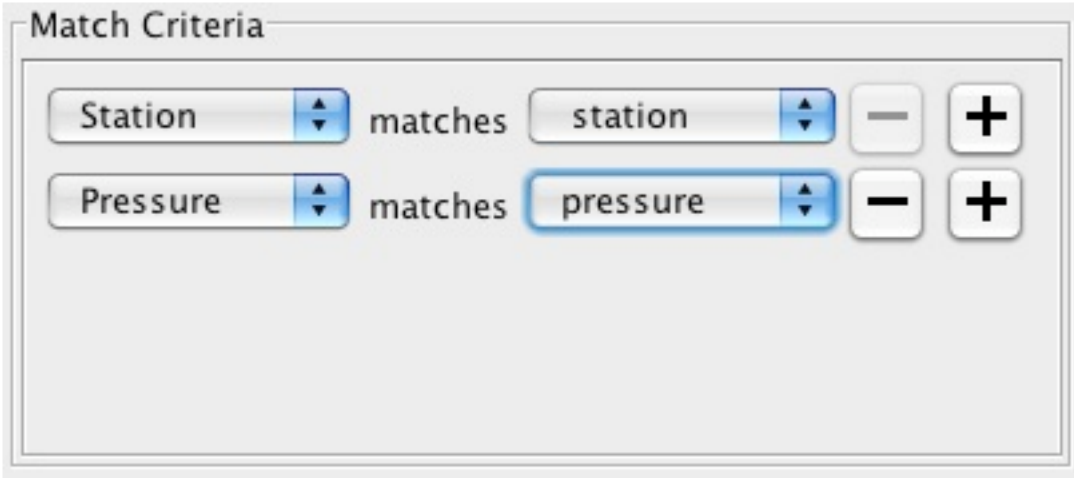
Features of the Merge Panel:

A: Merge Criteria Panel Define the match criteria for matching a row in the source file with an observation in the destination file. There can be up to 6 criteria. Each criteria defines a relationship between a metadata field in the destination file with a column name in the source file:



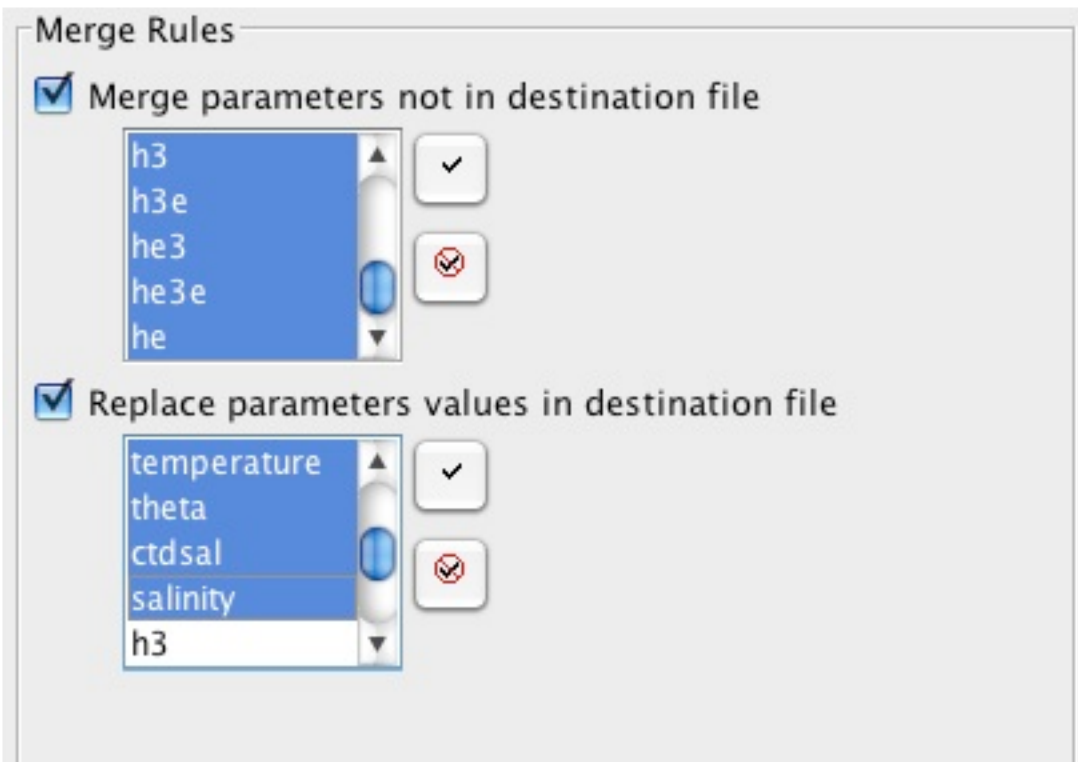


Additional criteria can be added by clicking the large + button:

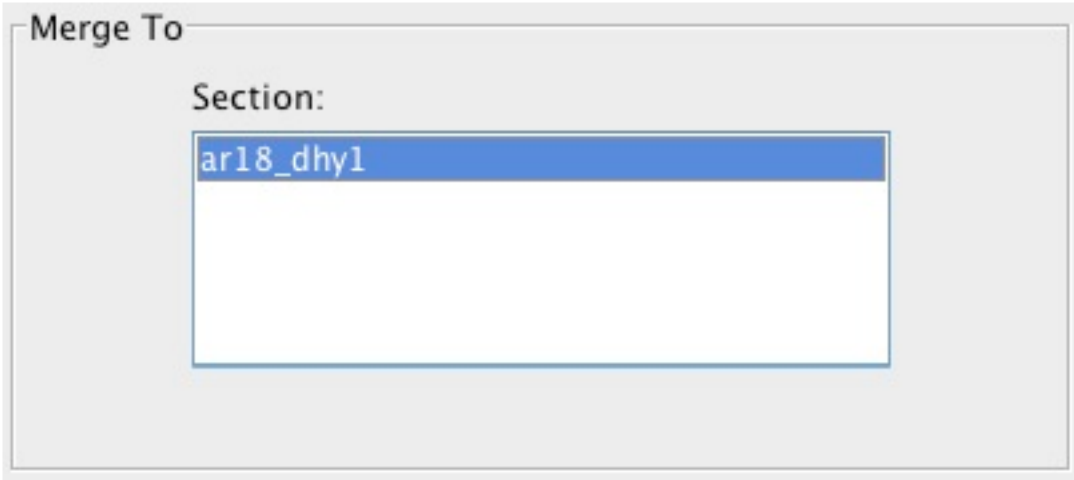


Criteria can be deleted by clicking the large – button.

B: Merge Rules Panel This panel defines whether any measured parameters in the source file will replace existing values in the destination file or parameters not found in the source are added to the destination:



C: Merge Destination Panel Select which of the open data windows to perform the merge on:



D: Merge Button Merge button will be enabled after you have defined merge criteria, merge rules, and a merge destination:

E: Merge Log The merge log shows the progress and results of the merge process. Errors will also be reported in this area.

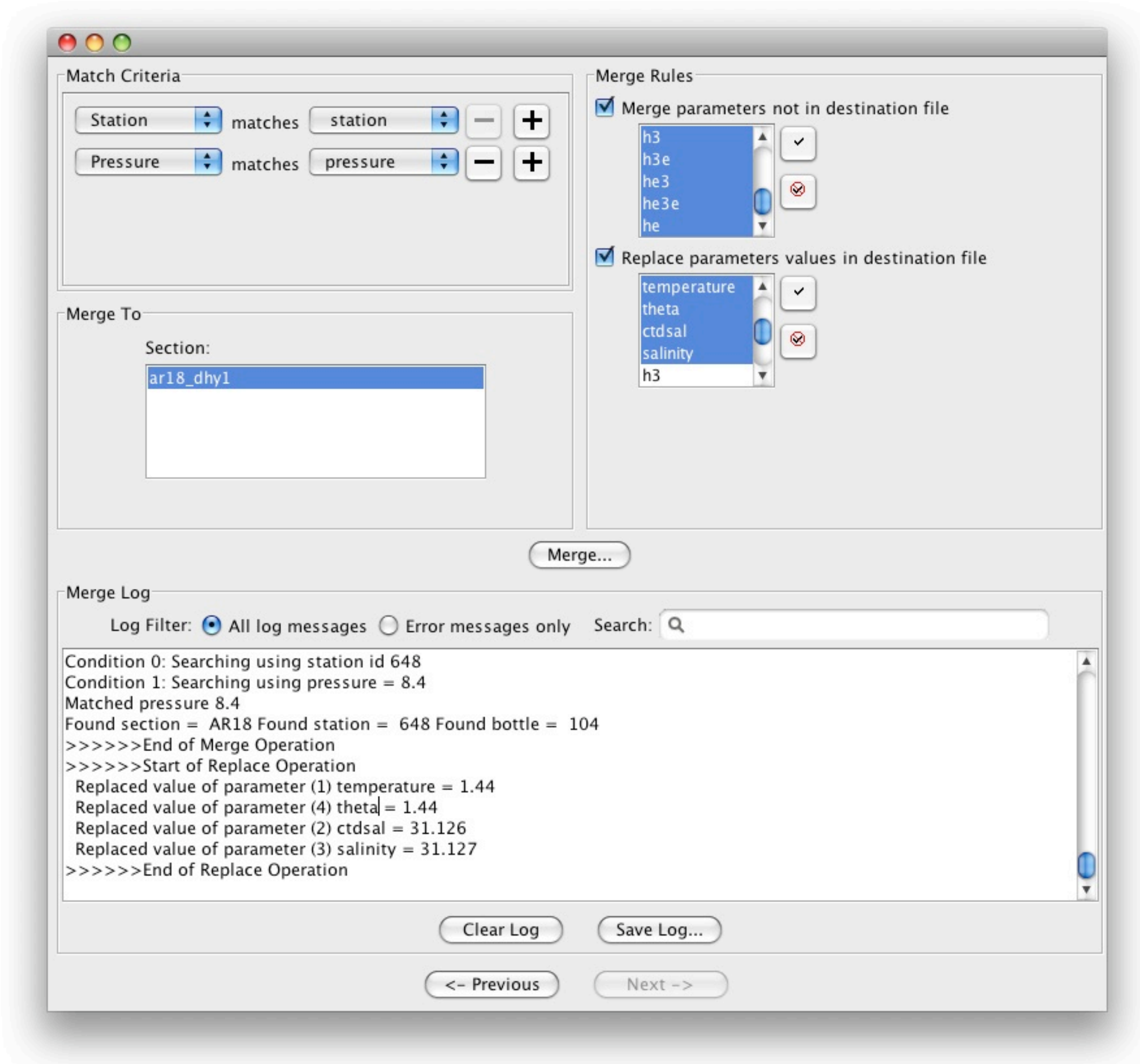
F: Log Filters You can filter the output of the merge log to all output or just error messages by clicking the appropriate button.

G: Log Actions Click *Clear Log* to clear the text in the log area. Click *Save* to save the log text to a text file.

Step #5: Here are settings chosen for this tutorial example. JOA will attempt match source lines by station identifier and pressure. It will attempt to merge new parameters, h3, h3e, he3, he3c, and he and replace values for temperature, theta, ctdsal, and salinity.

Note: Pressure matches are not exact…

Note: All parameters names are mapped to a common lexicon to determine a match.



Click the *Merge* button to begin the merge process. For the example in this tutorial, the results of the merge are:

Condition 0: Searching using station id 5599
Condition 1: Searching using pressure = 299.7
Condition 0: Searching using station id 588
Condition 1: Searching using pressure = 321.3
Matched pressure 321.3
Found section = AR18 Found station = 588 Found bottle = 101
>>>>>>Start of Merge Operation
  Translated parameter name h3 to h3
  Parameter not found in destination, adding parameter = h3
  Translated parameter name h3e to h3e
  Parameter not found in destination, adding parameter = h3e

Translated parameter name he3 to he3
Parameter not found in destination, adding parameter = he3
Translated parameter name he3e to he3e
Parameter not found in destination, adding parameter = he3e
Translated parameter name he to he
Parameter not found in destination, adding parameter = he
Set value of parameter (13) h3 = 1.52
Set value of parameter (14) h3e = 0.12
Set value of parameter (15) he3 = -1.1
Set value of parameter (16) he3e = 0.2
Set value of parameter (17) he = 3.897E-8
>>>>>>End of Merge Operation
>>>>>>Start of Replace Operation
Replaced value of parameter (1) temperature = 7.98
Replaced value of parameter (4) theta = 7.95
Replaced value of parameter (2) ctdsal = 35.234
Replaced value of parameter (3) salinity = 35.234
>>>>>>End of Replace Operation
Condition 0: Searching using station id 588
Condition 1: Searching using pressure = 248.5
Matched pressure 248.5
Found section = AR18 Found station = 588 Found bottle = 103
Set value of parameter (13) h3 = 1.79
Set value of parameter (14) h3e = 0.1
Set value of parameter (15) he3 = -2.5
Set value of parameter (16) he3e = 0.2
Set value of parameter (17) he = 3.835E-8
>>>>>>End of Merge Operation
>>>>>>Start of Replace Operation
Replaced value of parameter (1) temperature = 8.09
Replaced value of parameter (4) theta = 8.06
Replaced value of parameter (2) ctdsal = 35.242
Replaced value of parameter (3) salinity = 35.242
>>>>>>End of Replace Operation
Condition 0: Searching using station id 588
Condition 1: Searching using pressure = 198.1
Matched pressure 198.1
Found section = AR18 Found station = 588 Found bottle = 109
Set value of parameter (13) h3 = 1.76
Set value of parameter (14) h3e = 0.11
Set value of parameter (15) he3 = -0.9
Set value of parameter (16) he3e = 0.2
Set value of parameter (17) he = 3.885E-8
>>>>>>End of Merge Operation
>>>>>>Start of Replace Operation
Replaced value of parameter (1) temperature = 8.32

CREATING RESEARCH DATABASES WITH JAVA OCEANATLAS

Introduction to Research Databases in JOA

JOA incorporates a network-enabled data-based query tool that loads pertinent subsets of multi-disciplinary, earth-science datasets into a temporary, on-the-fly relational database, performs local calculations, and then allows a user to construct sophisticated SQL queries. What distinguishes this capability from other data selection tools is the ability to find a subset of a dataset from the values of measured parameters rather than just the spatial domain. Currently databases can only be built with oceanographic profile data but will be extended to time-series data in future revisions.

JOA computes summary statistics (e.g., average value, depth of maximum value, and depth of minimum value) for a profile from the observed parameters and from user-specified calculations (e.g., theta, sigma, and apparent oxygen utilization.) User-defined calculations such as mixed-layer depth and interpolation of a measured parameter to a standard level can also be specified. A simple two-step process takes the user from pre-selected data (e.g., from local data collections, or distributed data collections available through Dapper/OPeNDAP) to a set of data files selected from the computed summary statistics. First, the user determines what observed and computed variables will be used to construct the database, and therefore are available for subsequent queries. This selection is accomplished via a simple graphical interface that lists all available variables. Additional dialogs allow selection and configuration of a large set of computed variables. A single click then begins the process of ingesting the data files, computing summary statistics for both the observed values and user-specified calculated values, and building and populating a MySQL database. Second, after the database has been populated with the requested data, the user can create a SQL query using a second graphical user interface.

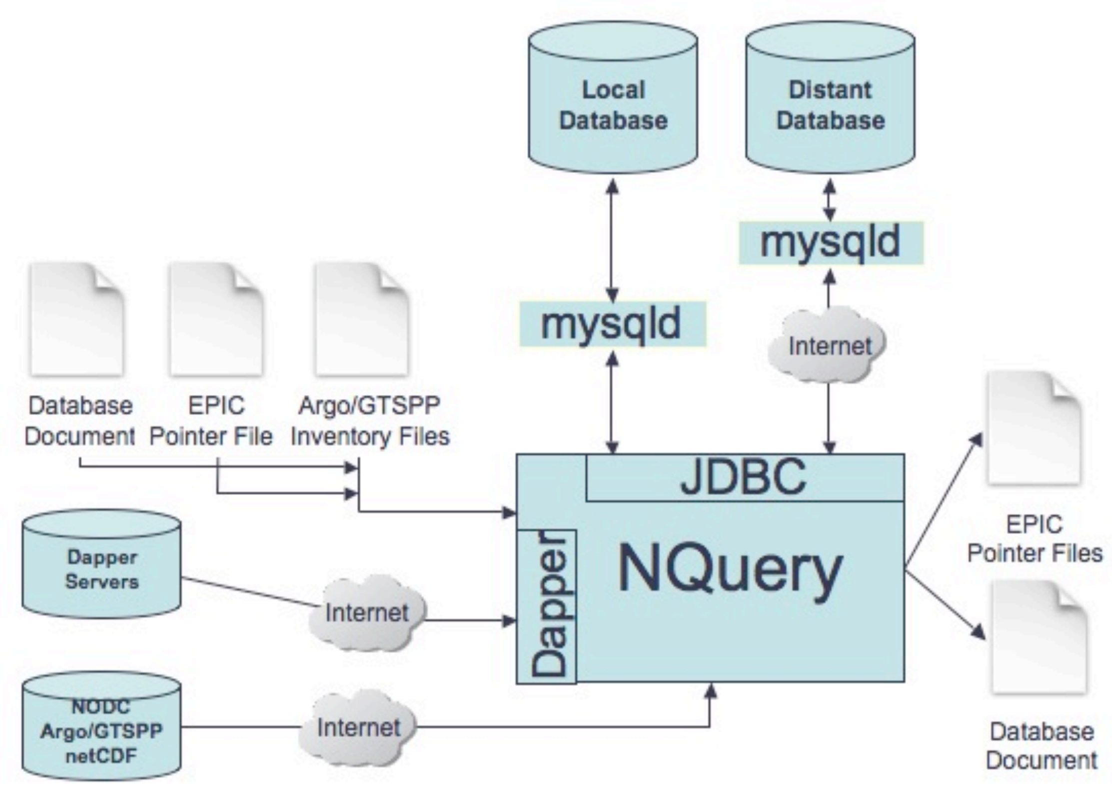
The user can build simple to fairly complex queries by using either the graphical interface, or entering an SQL statement by hand. Once a satisfactory query is constructed, it is executed by the database, resulting in a list of files that satisfies the query. The scientist can then open the original data directly into a JOA data windoew or use these files in other research tools by exporting a “pointer file” that contains the locations of the actual data files.

Limitations of JOA Databases

JOA is currently limited to profile data. JOA can open data from multiple sources (pointer files, Dapper servers, and existing databases) into separate database windows. However, JOA currently does not have the capability to combine datasets from different sources into one JOA document window (and thus into one on-the-fly database).

Browsing Data in JOA

Currently JOA can access profile data from three primary sources: XML/EPIC pointer files, ARGO/GTSPP Inventory files, and Dapper data servers. The way data flows in and out of JOA when creating a database is summarized in this block diagram:



On the left of the JOA symbol are data paths into JOA. They include EPIC XML pointer files that typically describe data on your desktop PC, Argo/GTSPP inventory files that reside on your desktop but refer to actual data files on NODC servers, and Dapper servers that are accessed via the internet. In addition, Database Documents represent previous databases you have created with JOA.

On the right are the output file formats; EPIC traditional and XML pointer files and Database documents.

The connections via the JDBC components are how JOA connects with either local or distant MySQL servers.

EPIC XML Pointer Files: Pointer files typically contain spatial/temporal metadata and a field that "points" to an actual data file located on your desktop computer but can also point to files on the Internet. JOA can read pointer files in the *EPIC XML* convention. These files define datasets of interests in the XML text markup language. Currently the only source of these files is Java OceanAtlas 4.1 or later. Typically, you would use Java OceanAtlas to open data files on your desktop computer (or use online sources) to create a custom dataset that matches your research interest. You would then export a collection of individual netCDF files from JOA with an associated XML pointer file that describes the geospatial domain of the data as well as the location of individual data files. You can browse this pointer file in JOA to build a database from the profiles referred to in the pointer file. In addition to JOA, JOA can produce EPIC XML pointer files from query results that point to files on your local computer or on the Internet. XML pointer files can be read into JOA via the *Open* command in the *File* menu.

Here are the first few lines on an EPIC XML Pointer file:

```
<?xml version="1.0"?>
<epicxml version="1.0" type="profile" uri="file:///Users/oz/Desktop/JOA Test
files/WOCE Pacific Subset">
  <domain>
    <latitude location="south" units="degrees_north">-32.526</latitude>
    <latitude location="north" units="degrees_north">58.4987</latitude>
    <longitude location="west" units="degrees_east">124.9883</longitude>
    <longitude location="east" units="degrees_east">-
71.50170000000003</longitude>
    <vertical location="top" units="db" positive="down">0.0</vertical>
    <vertical location="bottom" units="db" positive="down">8996.0</vertical>
    <date location="start" year="1985" month="3" day="30" hour="5" min="42"
secs="0.0"/>
    <date location="end" year="1996" month="7" day="4" hour="19" min="9"
secs="0.0"/>
  </domain>
  <varlist>
    <variable name="PRES" units="db" lexicon="JOA" algorithm=""/>
    <variable name="TEMP" units="ipts-68" lexicon="JOA" algorithm=""/>
    <variable name="CTDS" units="pss-78" lexicon="JOA" algorithm=""/>
    <variable name="SALT" units="pss-78" lexicon="JOA" algorithm=""/>
    <variable name="O2  " units="ml/l" lexicon="JOA" algorithm=""/>
    <variable name="SIO3" units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="NO3 " units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="NO2 " units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="PO4 " units="um/l" lexicon="JOA" algorithm=""/>
    <variable name="F11 " units="pmol/kg" lexicon="JOA" algorithm=""/>
    <variable name="F12 " units="pmol/kg" lexicon="JOA" algorithm=""/>
    <variable name="TRITUM" units="tu" lexicon="JOA" algorithm=""/>
    <variable name="C13 " units="/mille" lexicon="JOA" algorithm=""/>
    <variable name="O18O16" units="/mille" lexicon="JOA" algorithm=""/>
    <variable name="TCO2" units="umol/kg" lexicon="JOA" algorithm=""/>
    <variable name="ALKI" units="umol/kg" lexicon="JOA" algorithm=""/>
    <variable name="PH  " units="db" lexicon="JOA" algorithm=""/>
    <variable name="WTHT" units="deg" lexicon="JOA" algorithm=""/>
</varlist>
  <fileset id="P01W">
    <varlist>
      <variableref name="PRES"/>
      <variableref name="TEMP"/>
      <variableref name="CTDS"/>
      <variableref name="SALT"/>
      <variableref name="O2  "/>
      <variableref name="SIO3"/>
      <variableref name="NO3 "/>
      <variableref name="NO2 "/>
      <variableref name="PO4 "/>
```

```
      <variableref name="F11 "/>
      <variableref name="F12 "/>
      <variableref name="TRITUM"/>
      <variableref name="C13 "/>
      <variableref name="O18O16"/>
      <variableref name="TCO2"/>
      <variableref name="ALKI"/>
      <variableref name="PH  "/>
      <variableref name="WTHT"/>
    </varlist>
    <station id="30" cast="1" bottom="90.0" reference="P01W_30.nc">
      <date location="point" year="1993" month="9" day="12" hour="13"
min="11" secs="0.0"/>
      <latitude location="point" units="degrees_north">58.4987</latitude>
      <longitude location="point" units="degrees_east">141.8043</longitude>
      <vertical location="top" units="db"
positive="down">9.300000190734863</vertical>
      <vertical location="bottom" units="db" positive="down">49.5</vertical>
    </station>
    <station id="29" cast="1" bottom="148.0" reference="P01W_29.nc">
      <date location="point" year="1993" month="9" day="12" hour="9" min="32"
secs="0.0"/>
      <latitude location="point" units="degrees_north">57.9997</latitude>
      <longitude location="point" units="degrees_east">142.2978</longitude>
      <vertical location="top" units="db"
positive="down">10.199999809265137</vertical>
      <vertical location="bottom" units="db"
positive="down">74.69999694824219</vertical>
    </station>
    <station id="28" cast="1" bottom="190.0" reference="P01W_28.nc">
      <date location="point" year="1993" month="9" day="12" hour="5" min="37"
secs="0.0"/>
      <latitude location="point" units="degrees_north">57.4962</latitude>
      <longitude location="point" units="degrees_east">142.8107</longitude>
      <vertical location="top" units="db"
positive="down">10.600000381469727</vertical>
      <vertical location="bottom" units="db"
positive="down">99.30000305175781</vertical>
    </station>
```

Note: The output has been wrapped to fit the margins of this document. The specification for these files is summarized in Appendix ZZ.

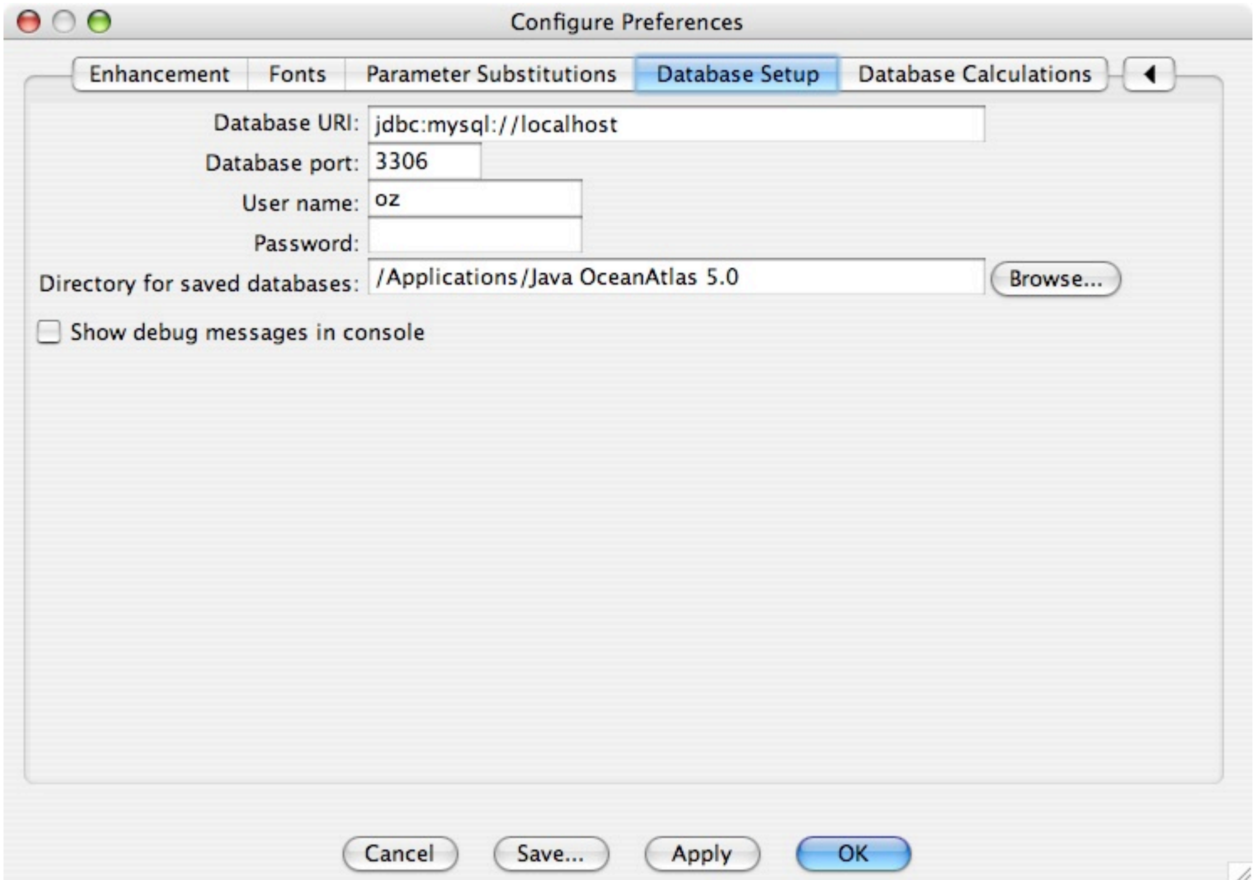
Connecting to a Database Server

JOA can connect to MySQL database servers running on your local machine or servers reachable via the Internet. Follow the instructions in Appendix YY for instructions on installing MySQL on your desktop computer. The examples in this user guide assume that MySQL is installed and running on your desktop computer.

By default, JOA tries to access a MySQL server at the URL: jdbc:mysql://localhost on port 3306. The URL for a distant server might look like: jdbc:mysql://dbserver.oceanatlas.com (a fictitious server). Port 3306 is the default for

MySQL installations—you should check with the MySQL server administrator at your institution to verify the MySQL port number in use.

To change the MySQL settings, open the **Preferences** menu and click on the Database Setup tab:



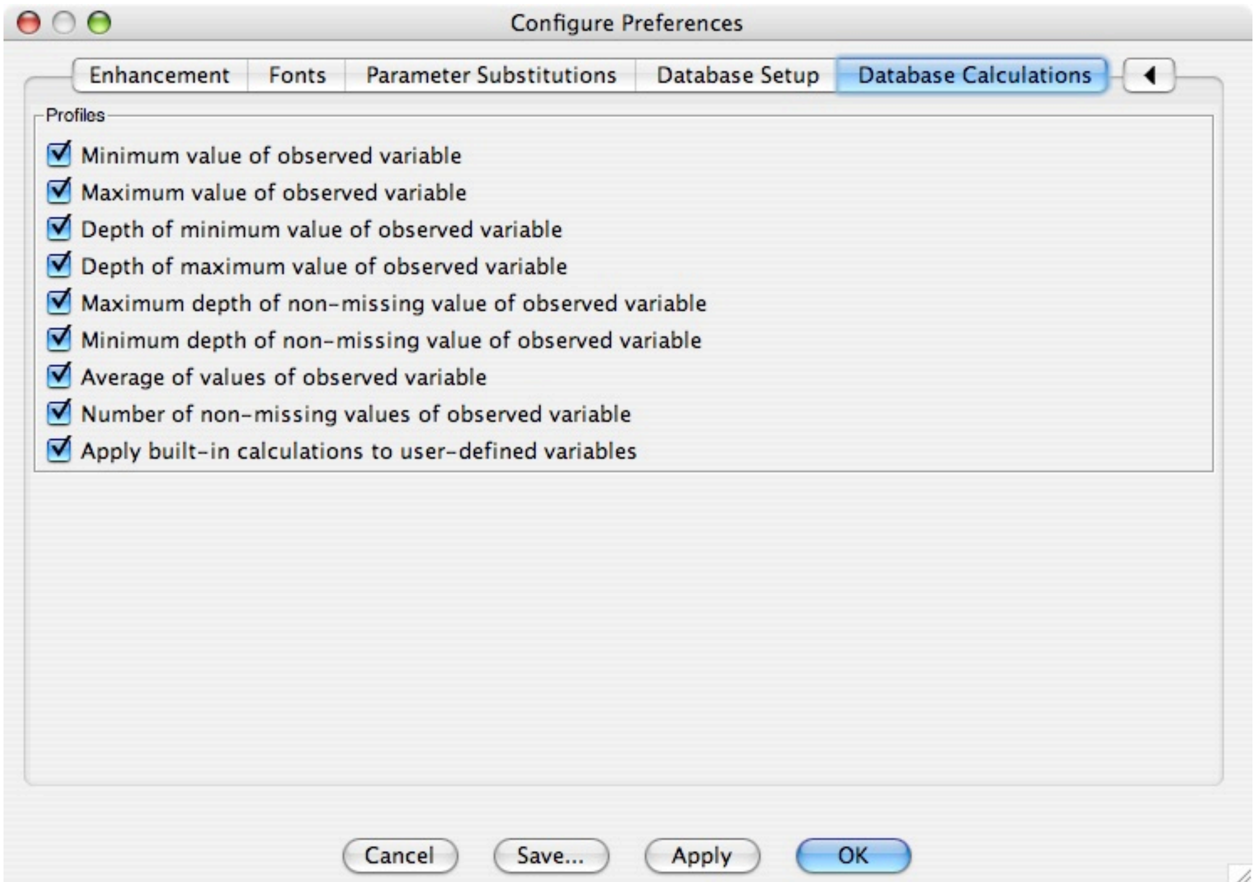
Here you can change the URL of the database server, the port number of the server, assign a default user name on the database server, and set a default directory to store JOA's database documents. You can enter a password for the specified user name but it is only saved for the current JOA session—it is not saved to the preferences file.

If you are having trouble building databases in JOA, you can turn on a debug mode for this database functionality. Click **Show debug messages in console** to have diagnostic messages added the the console output. These messages will often help you figure out what's going wrong. In addition, the console output is useful to the developer when fixing bugs. Note: this setting is not saved to the preferences file.

Clicking **OK** will allow any changes to persist for the current session. Click **Save**, to write your database settings to the JOA preferences file (with the exception of password). Click **Browse** to set a different default location for saved database documents.

Specifying Built-In Calculations for New Databases

JOA includes 8 built-in summary calculations. By default all these calculations are selected. To change which built-in calculations are computed; choose the **Preferences** command and click on the **Database Calculations** tab:



The selected calculations will be performed on all variables in the ingested data files plus any variables that JOA computes from the observed data. Each calculation will result in a corresponding column in the database for each observed or calculated variable encountered. For example, if there are 5 measured variables in the ingested data, JOA will build a table with 40 columns for the calculation results.

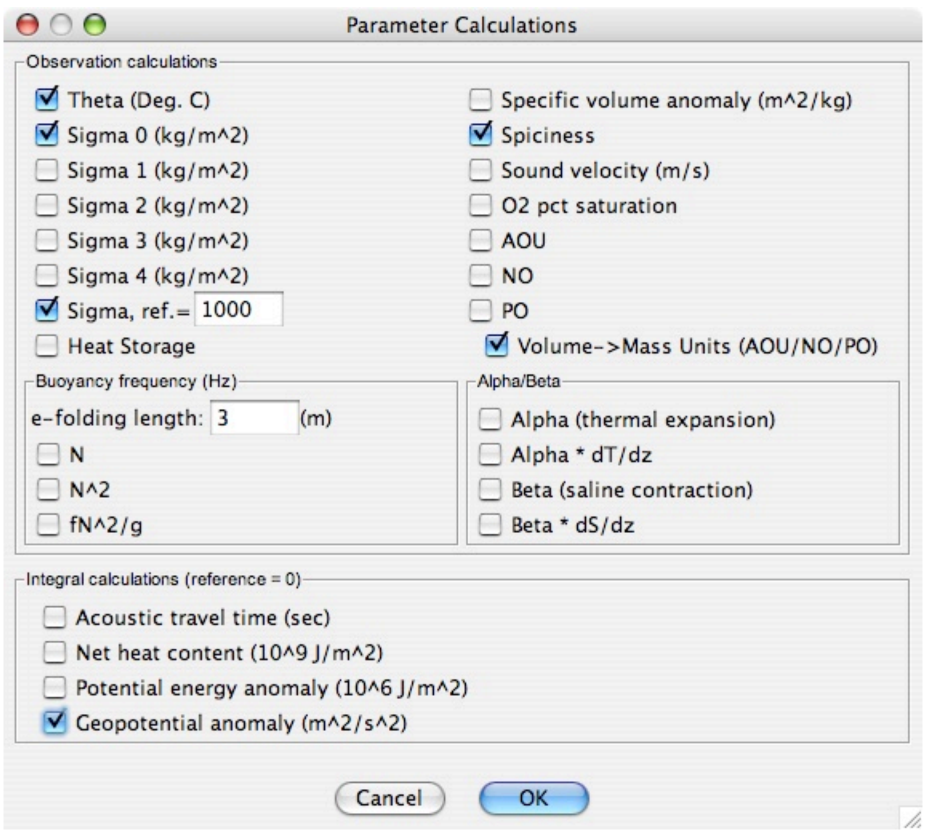
Calculating New Database Variables with JOA

JOA can calculate two types of new, user-defined, variables from the ingested data files; parameter calculations and “station” calculations. Parameter calculations result in a new value for each observation in the input data file—a new value at each observation or “bottle.” Examples are theta, sigma-theta, AOU, NO, spiciness, buoyancy frequency, spiciness, and geopotential anomaly. After JOA reads a profile, it calculates any observation calculations and then calculates the summary statistics for each observed variable and calculated variable.

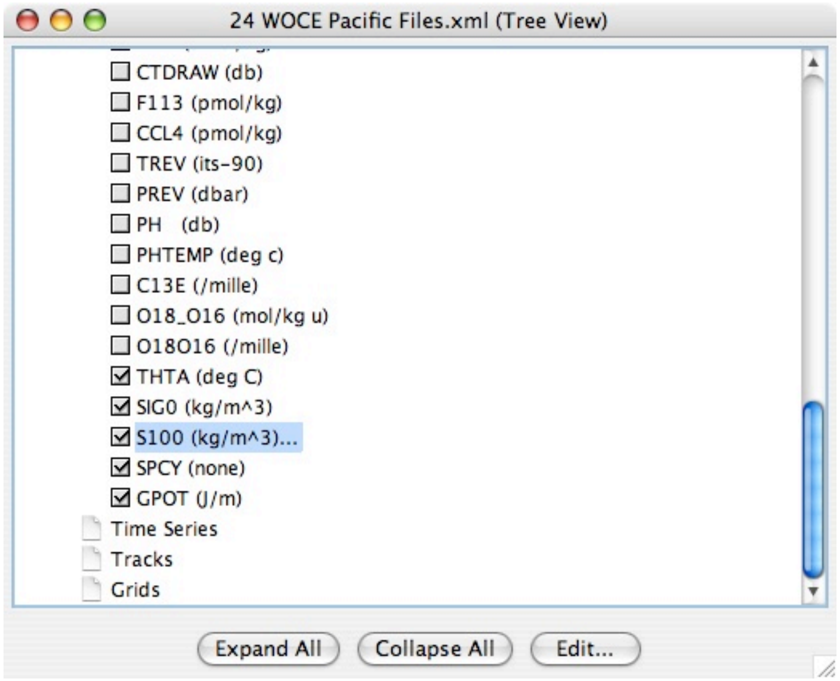
Station calculations represent a single value for each ingested profile. Examples include mixed layer depth, interpolation of an observed value to a surface of another observed variable (e.g., temperature interpolated to the surface), and integration of any observed variable in a range of another observed variable (e.g., average salinity in the range of 1000-2000 decibars). You can include as many station calculations as desired in the database.

Parameter Calculations

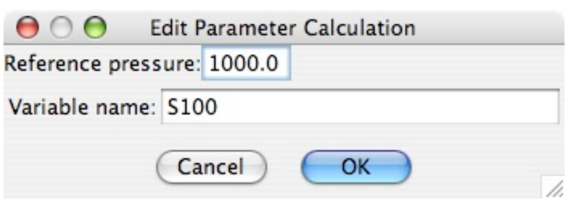
New parameter variables are added to the variable inspector with the Parameter Calculations command in the Profile Calculations menu. The following illustration shows which calculations are currently available:



New parameter calculations are added to the variable inspector window at the end of the variable list:



Calculations that have an argument (e.g., the reference pressure for the density calculation) can be edited by selecting them in the list and clicking the Edit button (can also just double-click the calculation in the list). In the above illustration, editing the S100 calculation presents the following dialog:

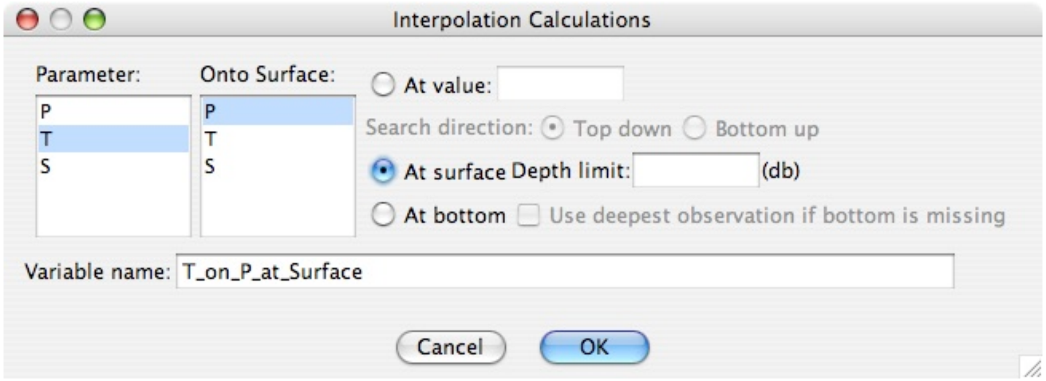


This dialog allows you to change the reference pressure and rename the calculation. Similar dialogs will be presented when editing other parameter calculations that require an argument.

Station Calculations

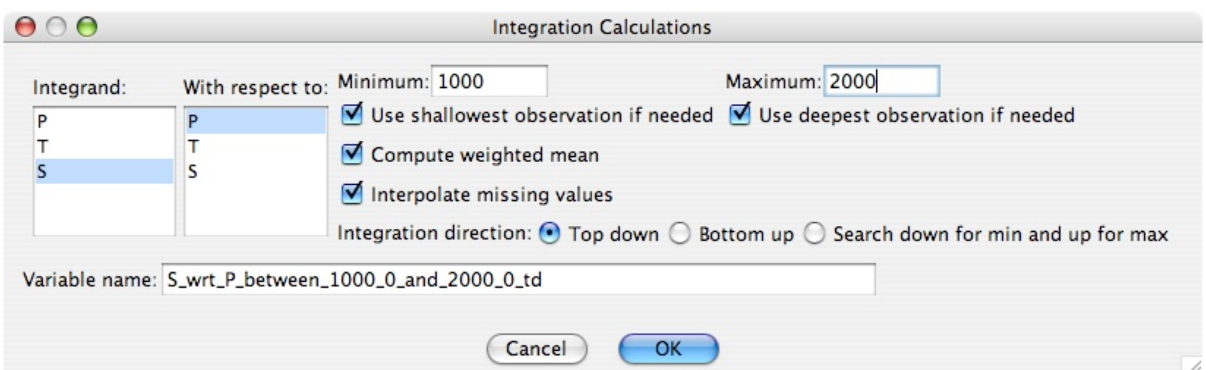
There are three types of station calculations, mixed-layer depth, interpolations, and integrations. These calculations are specified from the appropriate Profile Calculations submenu from the Calculations menu.

Interpolation: JOA will interpolate one observed parameter to a selected value of any other parameter:



For example, it is possible to interpolate salinity to a value of pressure (depth) or density. It is also possible to compute the depth of an iso-surface of any other observed or calculated parameter by interpolating pressure onto the value of that parameter. For example, interpolating pressure to a value of temperature equal to 10.0 will return the depth of the 10-degree isotherm. If you are interpolating to pressure, JOA has some additional options that allow you to interpolate to the surface or the bottom of the ocean. Interpolations to the surface have the option of defining a depth limit so that the first observation of a deep cast is not considered the surface. When interpolating to the bottom, you can specify whether to use the deepest observation if the depth datum for that station is missing.

Integration: JOA will calculate the integral (step-wise) of one original or calculated parameter over a selected range of any other parameter:



Options are presented to

> *Use the shallowest (or deepest) observation if needed* - for example if integrating over a sigma-0 range and the density at the shallowest observation is greater than the starting value should the values at the shallowest observation be substituted or should the calculation result be reported as missing for that station?

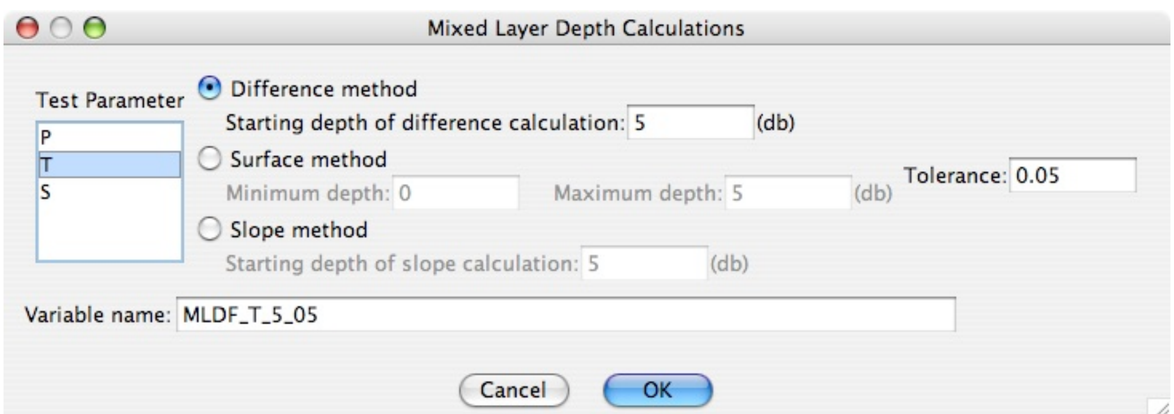
Compute weighted mean: weight the result over the limits of integration (otherwise the results are left raw). This results in an integration result in the units of the original integrand.

Interpolate missing values: if an observation of the integrand parameter is missing inside the range of integration, should JOA provide a linear interpolation at the level from the closest levels above and below with non-missing observations? Can also set a custom value of *Max number of observations* to limit interpolation of missing values.

The integration direction can be specified to be from shallow to deep, from deep to shallow, or by searching down for minima of the parameter of the axis of integration and up for maxima of the parameter of the axis of integration.

Warning: The integration calculation is not sophisticated. It works reasonably well for ranges of any parameter, which does not contain mid-profile extrema (it works well over pressure or density, for example). Despite the logic choices, the results can be misleading when a parameter is integrated over an axis with mid-level extrema. The intent is to supply for a typical data file a reasonable approximation of integration calculations.

Mixed-Layer Depth: JOA will calculate the depth of the mixed layer for each ingested profile and add to the database:



Any original parameter can be used as the test parameter. The Test Parameter is the parameter whose variability - or lack of it - is used as the indicator of mixed-layer depth.

Three methods of mixed-layer depth calculation are available:

In the *Difference method* JOA, beginning at the *Starting depth of difference calculation*, will look at successively deeper (higher pressure) observations in each data file until it reaches an observation whose value for the test parameter differs from the value of the test parameter at the starting depth by more than a user-chosen *Tolerance*.

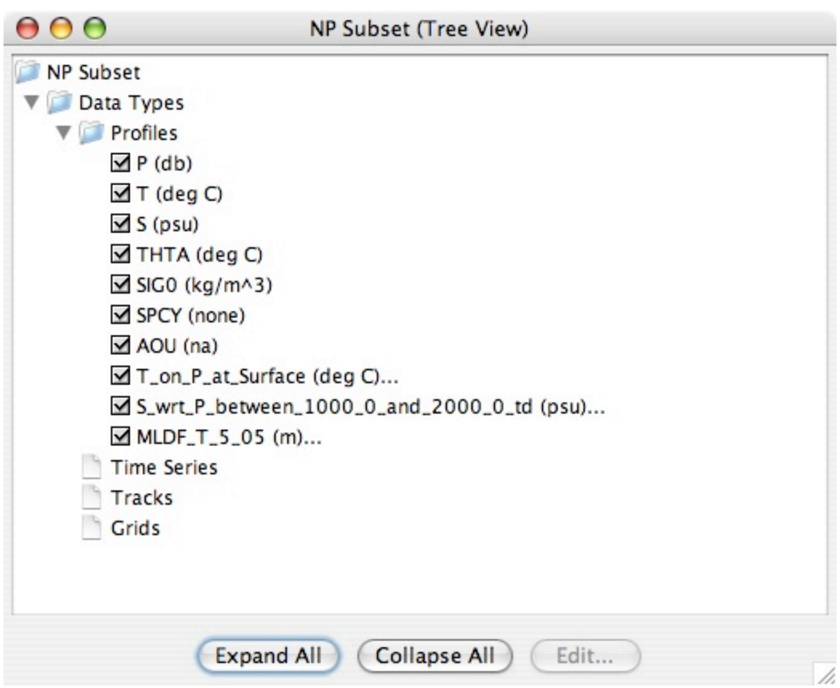
For example, if 'T' is selected with a starting depth of 5 and a tolerance of 0.5, JOA will report back the thickness between 5 and whatever level contains a temperature 0.5 degrees different than the value at 5.

The *Surface method* is similar to the difference method except that instead of comparing the data down the water column to the value at the starting depth, NQuery compares the data to a calculated bulk surface value that is created by averaging the test parameter over the range of pressures specified. The depth of the mixed layer is reached when the difference between a observation's value of the test parameter and the bulk surface mean is greater than the tolerance.

The *Slope method* is similar in some respects to the difference method, but each successive measurement pair is used for the examination of whether or not their difference exceeds the tolerance. For example, if 'T' is selected with a starting depth of 10 and a tolerance of 0.05, JOA will compare the next deeper measurement to 10, the one after it to the one after 10, etc., and report back the thickness between 10 and whatever level contains a temperature 0.05 degrees different than the previous value.

Warning: These algorithms are best suited to CTD profiles. The intent is to supply for a typical data file a reasonable approximation of mixed layer calculations.

Editing Station Calculations: After configuring a station calculation, it is added to the variable inspector at the end of the variable list:



To edit an existing station calculation, select the calculation in the list and click the Edit button (or double click the calculation.) The appropriate dialog will open that will allow you to change the settings of the calculation as well as it's name.

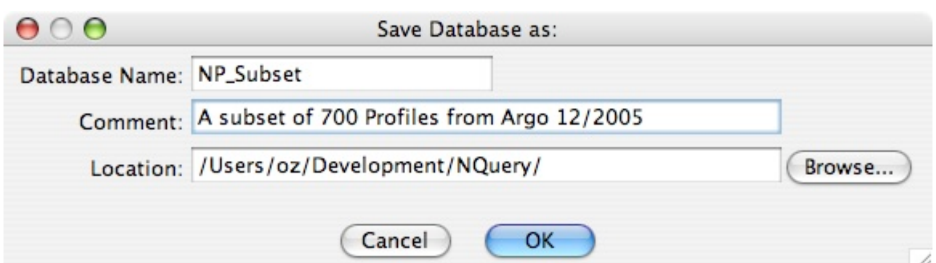
Currently, there is no way to remove a station or parameter calculation from the variable list. If you change you mind about whether a calculation is needed simply uncheck the box next to it in the list.

Creating Databases in JOA

Creating a New Database

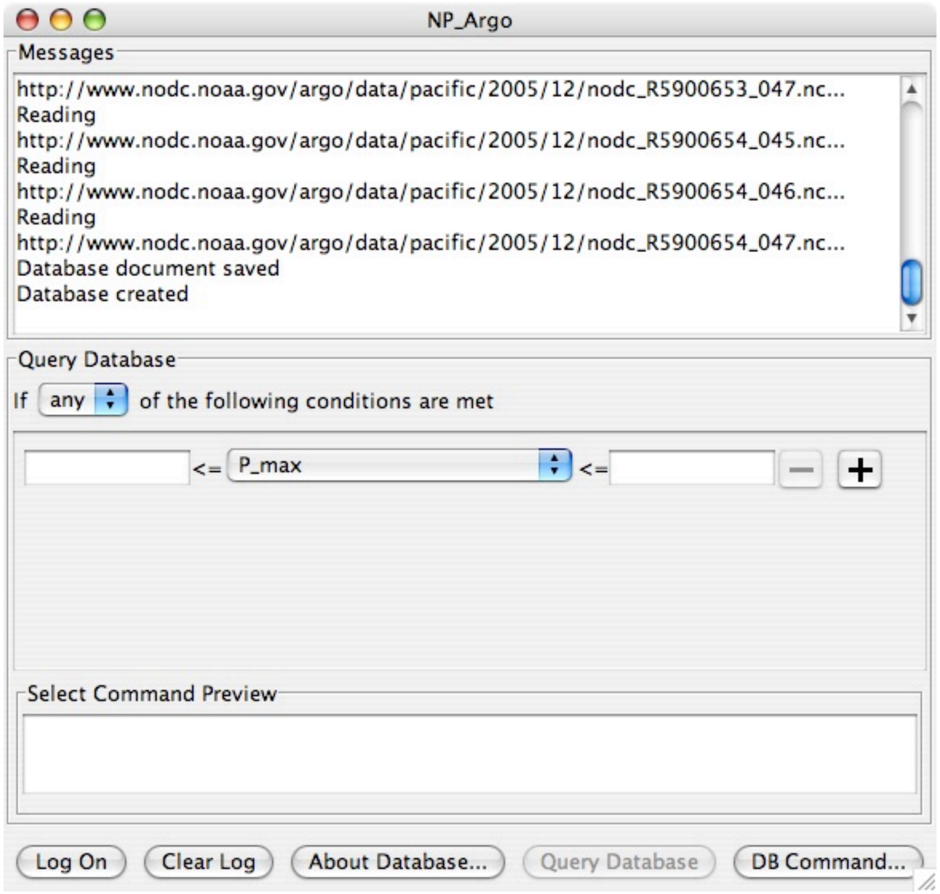
After you have selected variables and defined custom calculations, you are ready to create a new MySQL database from the selected profiles. With the Variable Inspector window selected, issue the **Create Database** command in the **File** menu. If you haven't entered a password yet, you will be prompted to do so:

You will next be prompted to name your new database and provide an optional comment. The database name will be used to name the actual MySQL database. The database name and comment will also be stored in the Database Document created with every database so existing databases can be reopened. You can chose to save the Database Document in the specified location or chose a new location by clicking the **Browse** button.



Note: the database name should only contain standard upper and lowercase letters and numbers. Special characters should be limited to the underscore and dash characters. Using other punctuation characters and/or spaces will cause an error creating the database.

Click **OK** to build the database structure, ingest the selected profiles, calculate summary statistics, calculate user-defined variables, and populate the database tables. A new Database Document will open and the progress of data ingest will appear in the Messages are of the window. If there are no errors, you will see *Database document* *saved* and *Database created* messages in the console:



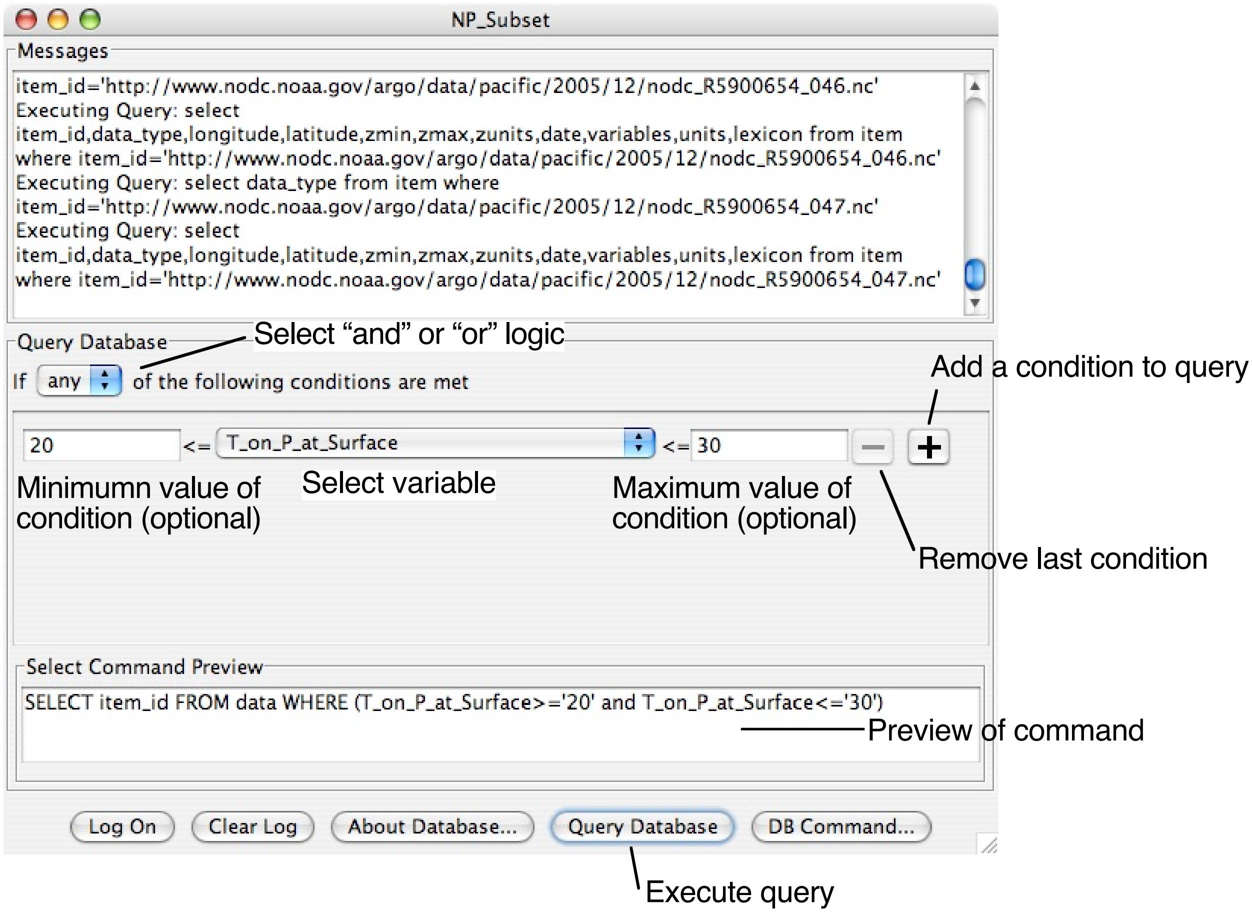
Opening Existing Databases

When JOA builds an on-the-fly database, it also creates a separate document that allows you to reopen the database. This is called a “database document”. These files have an .nqdb extension and are stored in the directory specified in the JOA Preferences dialog. After a database document has been created, it can be reopened via the **Open** menu in the **File** menu.

Searching JOA Databases

The primary purpose of a JOA database is to identify data of interest by its characteristics using the summary statistics computed from the observed values and any user-defined calculations such as mixed layer depth. NQuery provides two ways to query a database; graphical and command line.

The graphical user interface for building queries is located in the middle panel of the database documethe query builder:



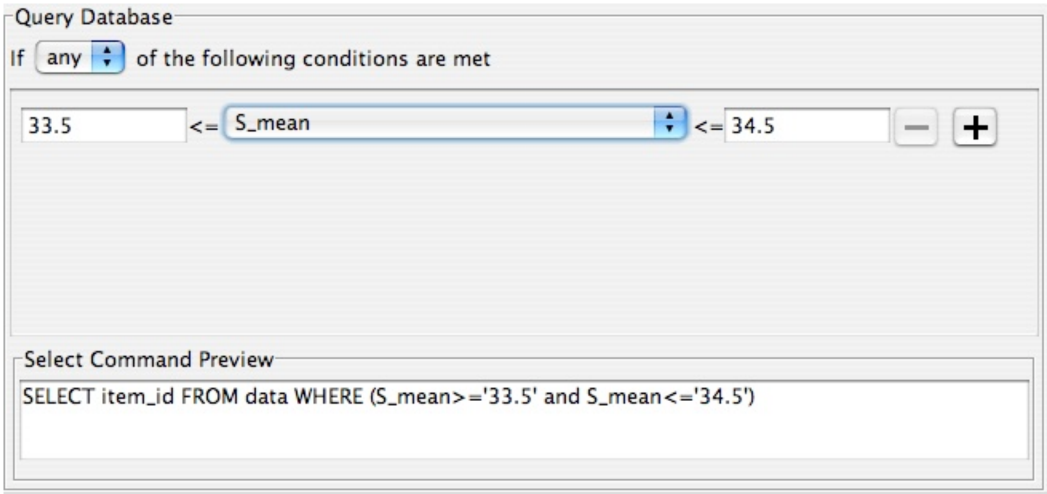
Search logic: A search can be constructed with either “or” or “and” logic. Selecting “any” from the popup menu means that individual conditions will be connected with “or” logic and will broaden the results of the query. Selecting “all” from the popup menu means that individual conditions will be connected with an “and” statement and will narrow the results of the query.

Query Builder: Each condition is composed of a variable from the database (actually a column in a table) and a range of values to test that variable’s values against. Select a variable from the center popup menu and provide at least a minimum value or a maximum value or both. To add a new condition, click the large “plus” button. To remove the last condition, click the large “minus” button. Note you can not remove the first condition.

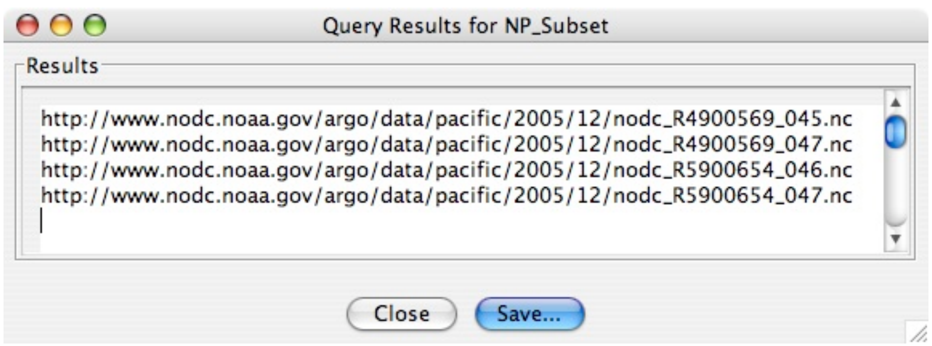
Query Preview: The exact SQL query is constructed as you use the query builder and is shown in the Select Command Preview area of the window.

A query can have as many conditions as you want but note that the graphical query builder can not group conditions with parentheses or include a “not” operator. If you wish to construct a query out of the scope of the graphical query builder, click the **DB Command** button to enter your query expression.

Example of a Simple Query: In this example we are looking for all the profiles in a certain range of salinities:

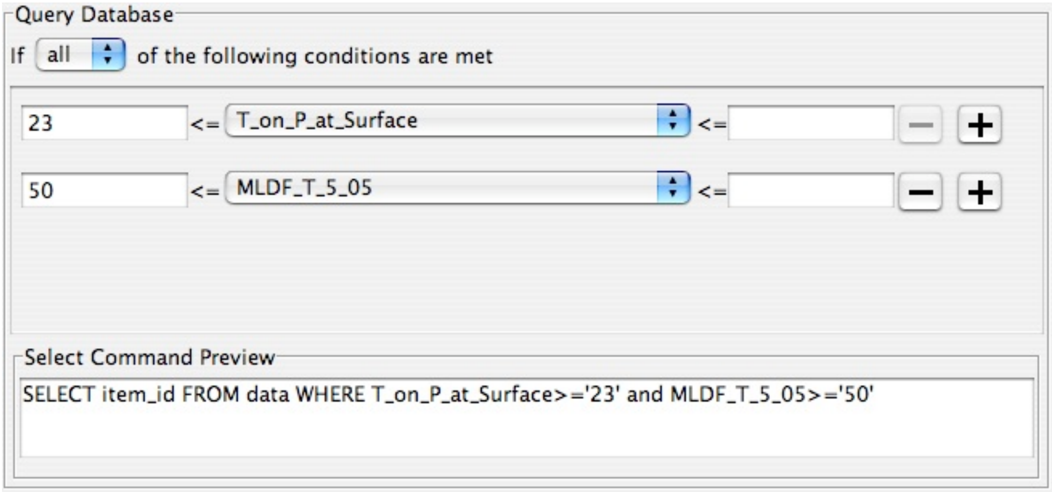


Which returns the following results:



Clicking **Save** would allow you to save the results as an EPIC pointer.

Example of a Complex Query: A complex query allows you to test the values in the database against two or more conditions. This example tests whether the interpolated sea-surface temperature is greater than 23° AND the computed mixed layer depth is greater than 50m:



Notice that the all option is selected in the logic popup to use "and" logic in the query. This query finds only 2 matching profiles:

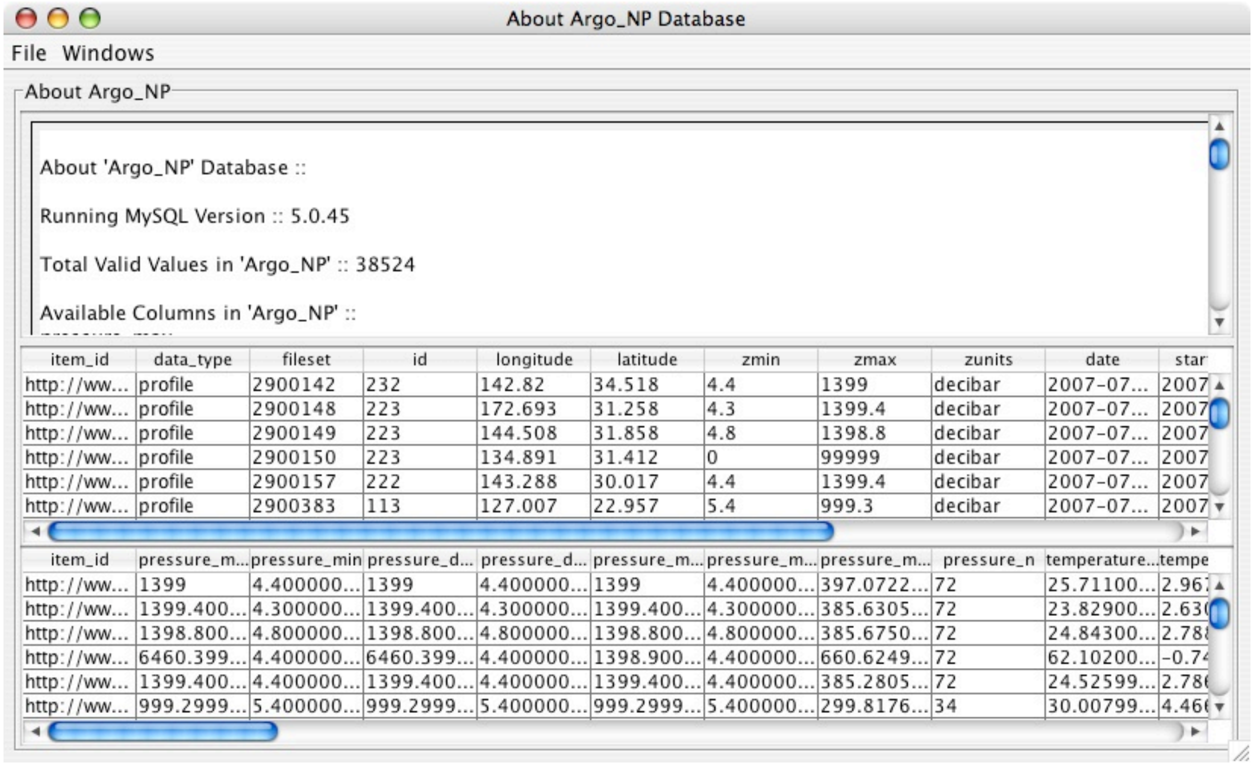
Saving and Exporting Query Results

Query results can be written to either a traditional EPIC pointer file or a new XML pointer file. Currently, only Java OceanAtlas 4.1 or greater can directly read these files and only for profile files that can be located on your local machine or files reachable with an HTTP URL like the Argo and GTSPP archives at NODC.

MAINTAINING NQUERY DATABASES

Getting information about the Current Database

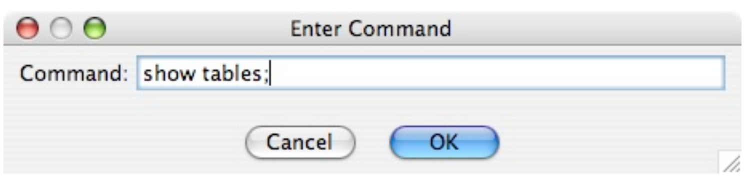
To find out general information about the current database (the one open in the active database document window), click the **About Database** button in the database document window. A window with a scrolling list of the total number of values and column names in the current database and tables of the *item* and *data* tables will be opened:



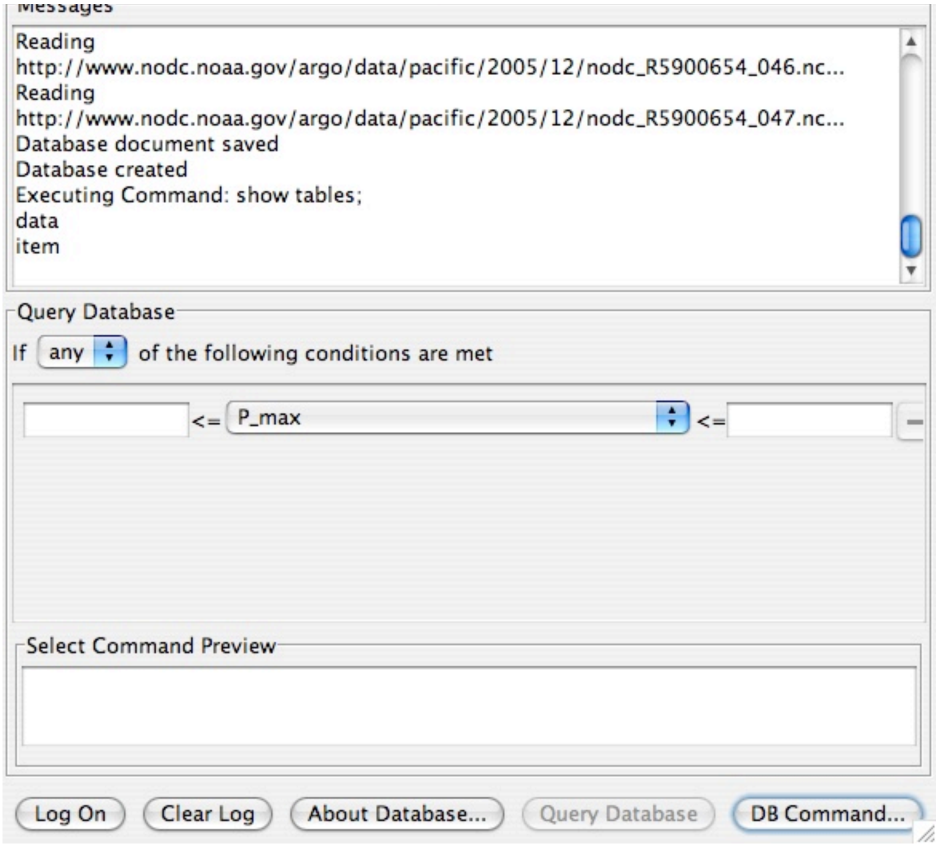
The *item* table (middle table) list the metadata for every profile summarized in the database and the *data* table (lower table) shows the actual computed values stored in the database.

Issuing MySQL Commands From Within JOA

Any MySQL command can be executed directly from JOA by clicking the **DB Command** button in the database document window. This will display the Enter Command window. Type your command and click the B button to send you command to the MySQL database server:



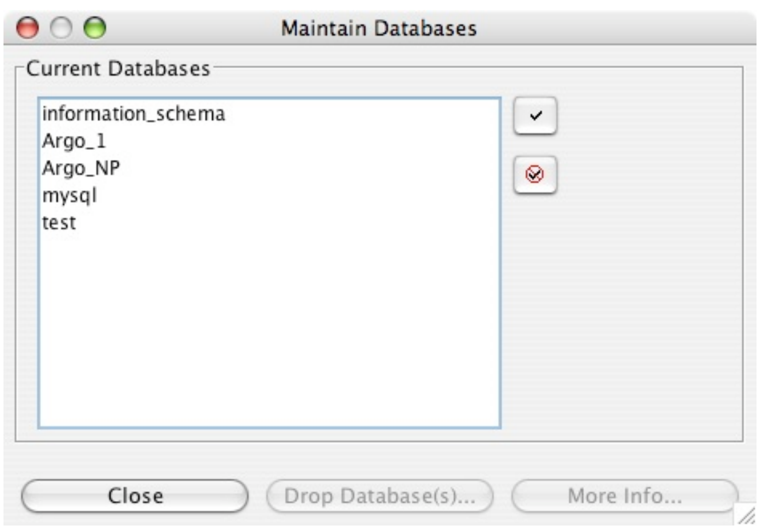
The output is shown in the message area of the database document window:



Note: All MySQL commands must be terminated by a semicolon.

Maintaining Your Databases

To display a list of all the databases you have created with JOA, use the **Maintain Databases** command in the **File** menu:



Deleting a database document (a file with a .nqdb extension) at the operating system level does not delete the actual database stored in the MySQL server. To delete a database, you need to drop it either through the MySQL command line interface or by selecting the database(s) to drop and click the **Drop Database** button. Make sure you delete the associated database document file after you drop its associated database.

Click the **More Info** button command described above to display a summary of the database represented by this document

Other Database Document Functions

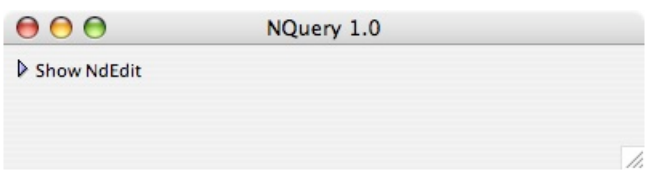
Log Off: Disconnects a Database Document from the database server. To use the database, you will have to reconnect using the Log On button and providing a valid password.

Clear Log: This button clears the text from Messages area of the Database Document window.

Example #1: Browsing an EPIC XML Pointer File

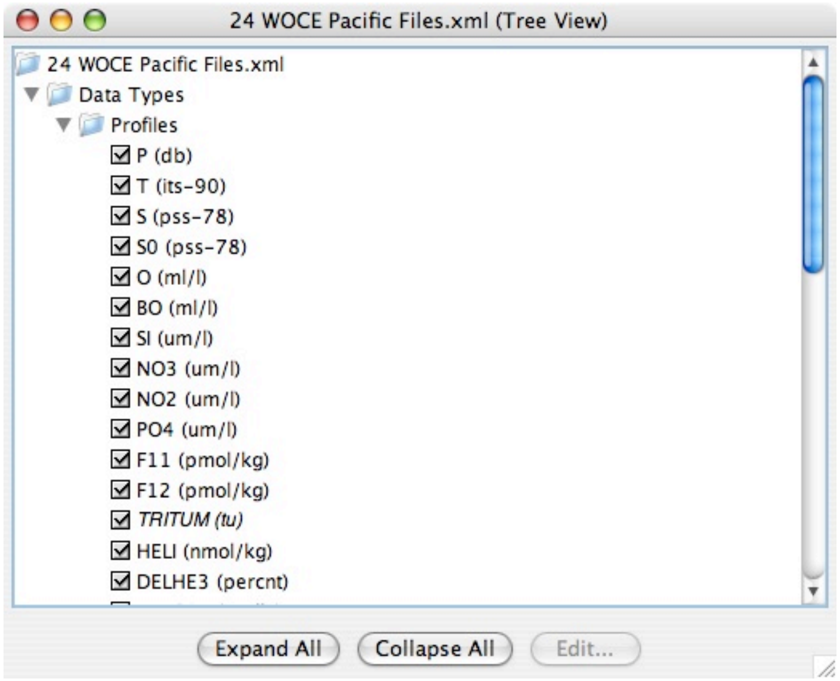
In this example, we will create a database by opening a pointer file that references data on your desktop computer. In this case, the data are all the WOCE profiles in the Pacific Ocean. To create this collection of 3,590 profiles, I opened all the individual section files in Java OceanAtlas and exported a folder of individual netCDF files tied together with a pointer file.

0) Launch the JOA application. You will see the JOA application window:

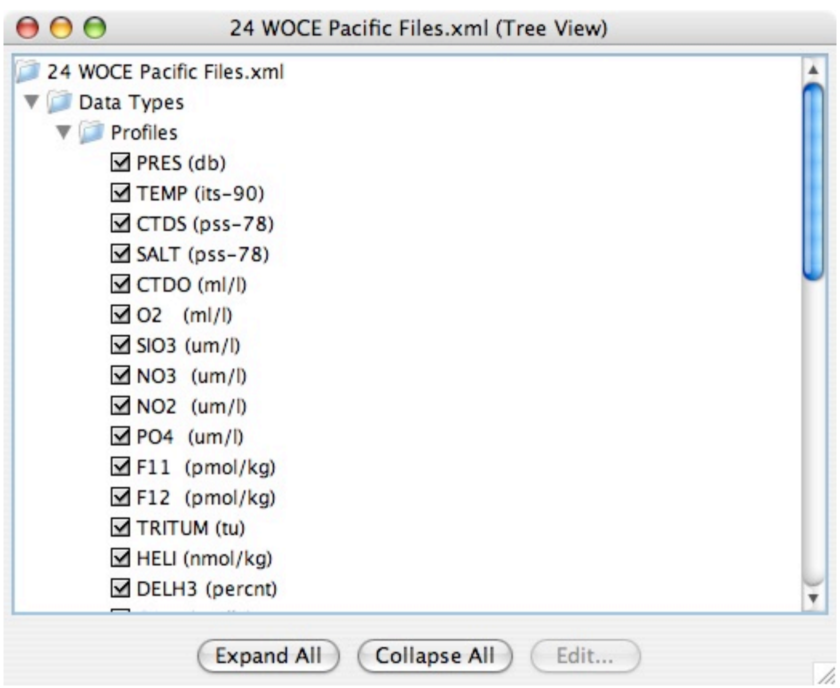


Note: This illustrates the Mac OS X look and feel; Windows and Linux will place the menubar in this window.

1) Open the XML pointer file in JOA using the **Open** command in the **File** menu. You will see the JOA *Variable Inspector* window. Here you will select the variables to compute the built-in summary statistics on and add custom calculations:



Note: in the above illustration, variables names have been expressed in the EPIC lexicon. Notice that the variable TRITUM is rendered in italics. This is because that variable name does not exist in the EPIC variable lexicon so the original name is used. The next illustration shows what the variable inspector would look like if you didn't translate the variable names:



See later in this chapter for a complete discussion of JOA calculations.

Example #2: Browsing an Argo/GTSPP Inventory Files

The latest inventory files can be downloaded from the Argo and GTSPP Web sites at:

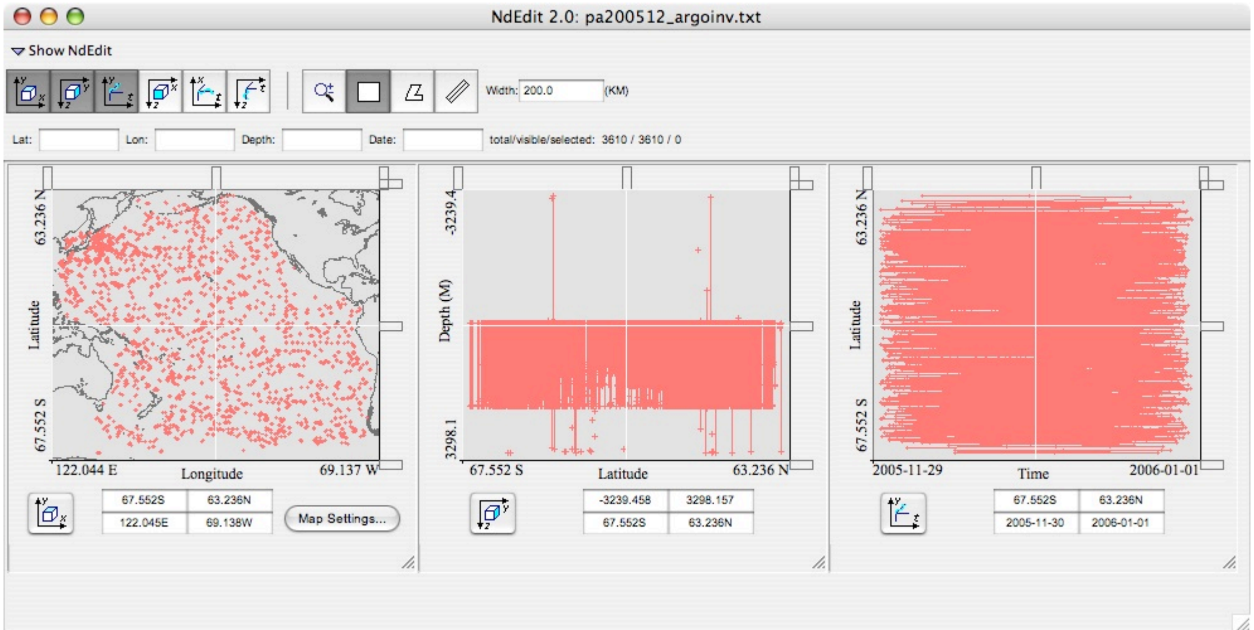
http://www.nodc.noaa.gov/argo
http://www.nodc.noaa.gov/GTSPP

These files contain spatial/temporal metatdata and a URL to a netCDF file that contains the actual profile data. Here is an example of the first few lines of an Argo or GTSPP inventory file:

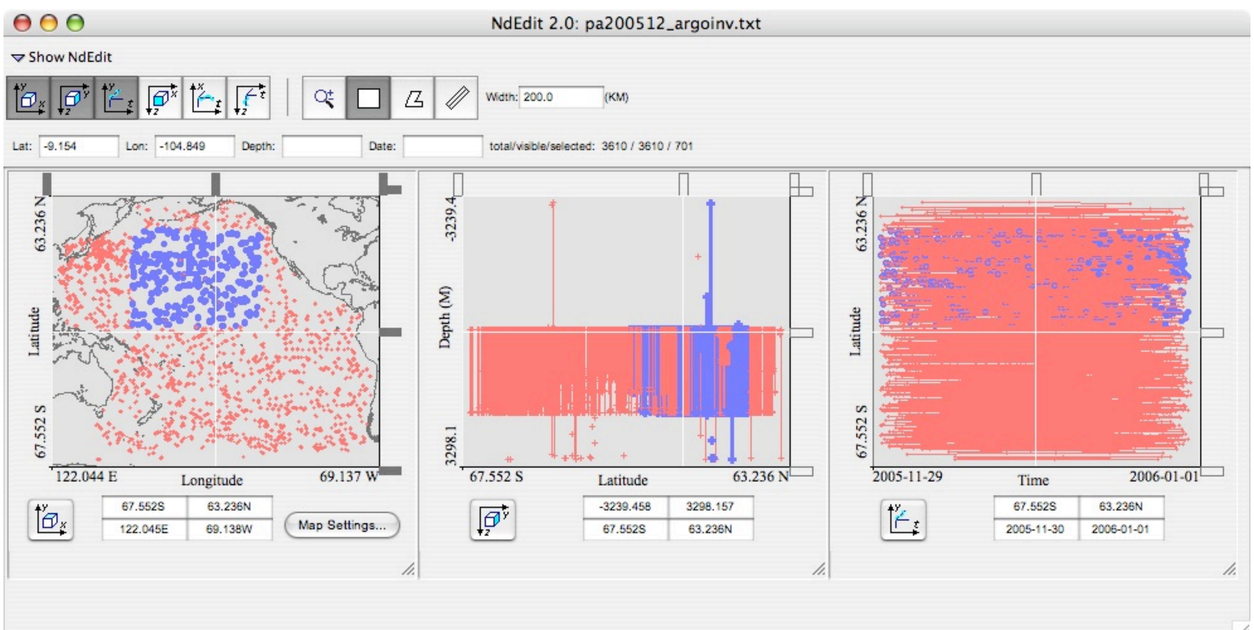
```
callSign,data_URL,file,ocean,date,time,time_qc,latitude,longitude,position_qc
,data_center,data_mode,num_of_levels,min_D_P,max_D_P,num_of_param,param1,para
m2,param3,param4,param5
1900148,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900148_104.
nc,pacific/2005/12/nodc_R1900148_104.nc,pacific,2005-12-04,17:23,1,-
53.363998,151.05499,1,AO,R,49,10.5000,949.000,3,PRES,TEMP,PSAL
1900148,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900148_105.
nc,pacific/2005/12/nodc_R1900148_105.nc,pacific,2005-12-15,06:22,1,-
53.508999,150.86099,1,AO,R,70,10.8000,1999.20,3,PRES,TEMP,PSAL
1900148,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900148_106.
nc,pacific/2005/12/nodc_R1900148_106.nc,pacific,2005-12-25,20:18,1,-
53.705002,151.01601,1,AO,R,50,11.0000,999.200,3,PRES,TEMP,PSAL
1900153,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900153_104.
nc,pacific/2005/12/nodc_R1900153_104.nc,pacific,2005-12-04,11:46,1,-
55.549000,161.98000,1,AO,R,50,10.7000,999.600,3,PRES,TEMP,PSAL
1900153,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900153_105.
nc,pacific/2005/12/nodc_R1900153_105.nc,pacific,2005-12-15,03:32,1,-
54.407001,163.89700,1,AO,R,69,11.0000,1949.00,3,PRES,TEMP,PSAL
1900153,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R1900153_106.
nc,pacific/2005/12/nodc_R1900153_106.nc,pacific,2005-12-25,13:44,1,-
56.813999,164.53799,1,AO,R,50,10.9000,998.900,3,PRES,TEMP,PSAL
2900139,http://www.nodc.noaa.gov/argo/data/pacific/2005/12/nodc_R2900139_117.
nc,pacific/2005/12/nodc_R2900139_117.nc,pacific,2005-12-
04,22:52,1,36.577000,153.28300,1,AO,R,72,3.80000,1398.80,3,PRES,TEMP,PSAL
```

Note: The lines of this file have been wrapped to the margins of this document.

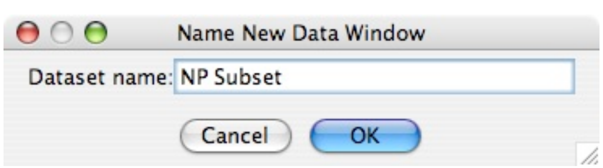
You use the *NdEdit* component of JOA to filter the inventory files, select a subset of profiles, and ingest the selected profiles into JOA. Inventory files can be read into JOA via the **Browse** command in the **File** menu:



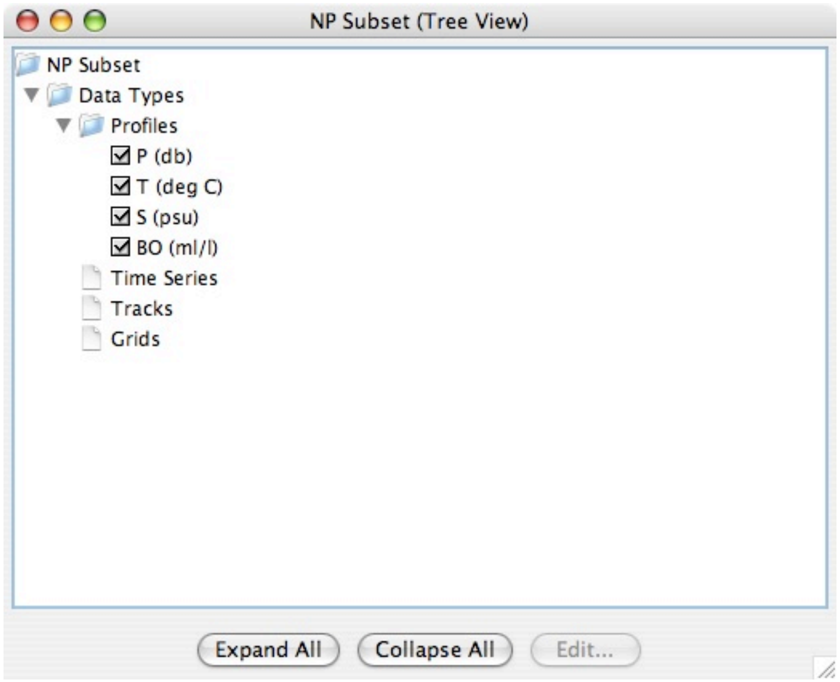
Select any number of profiles to create your database:



To open the variable inspector for this selection use the **Create New Database from Selection** command in the **File** menu. You will be prompted to name the new data selection:



You will see JOA's Variable Inspector window:

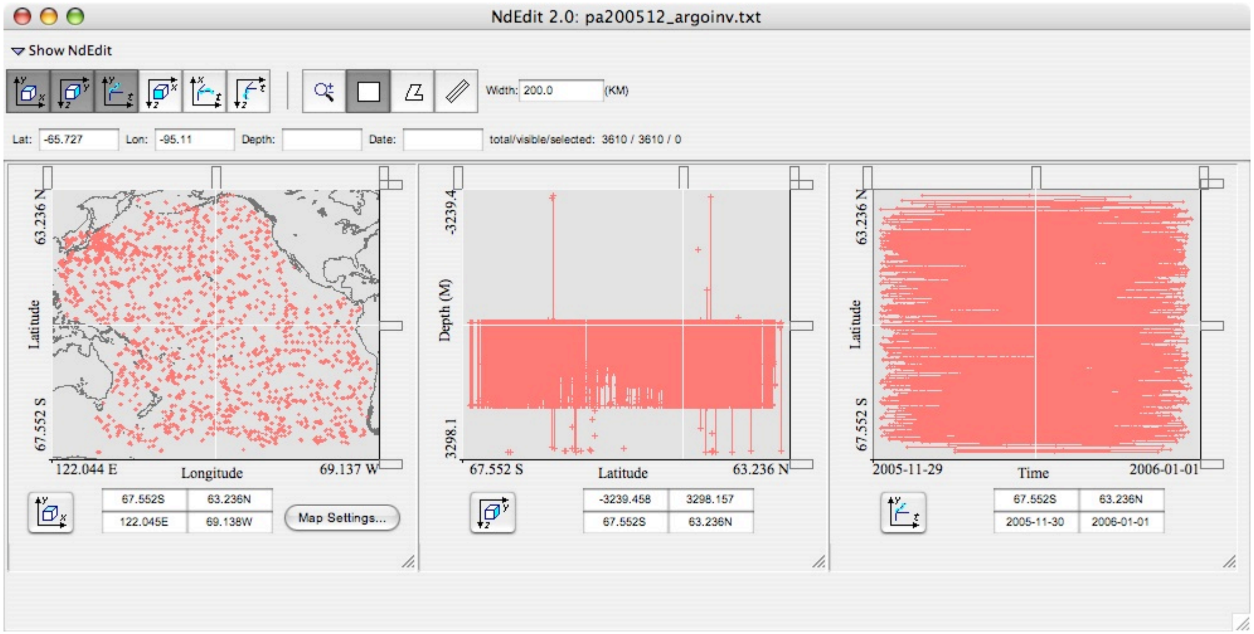


After you have selected a subset of Argo or GTSPP profiles, JOA will use the URL in the inventory files to ingest the actual data file.

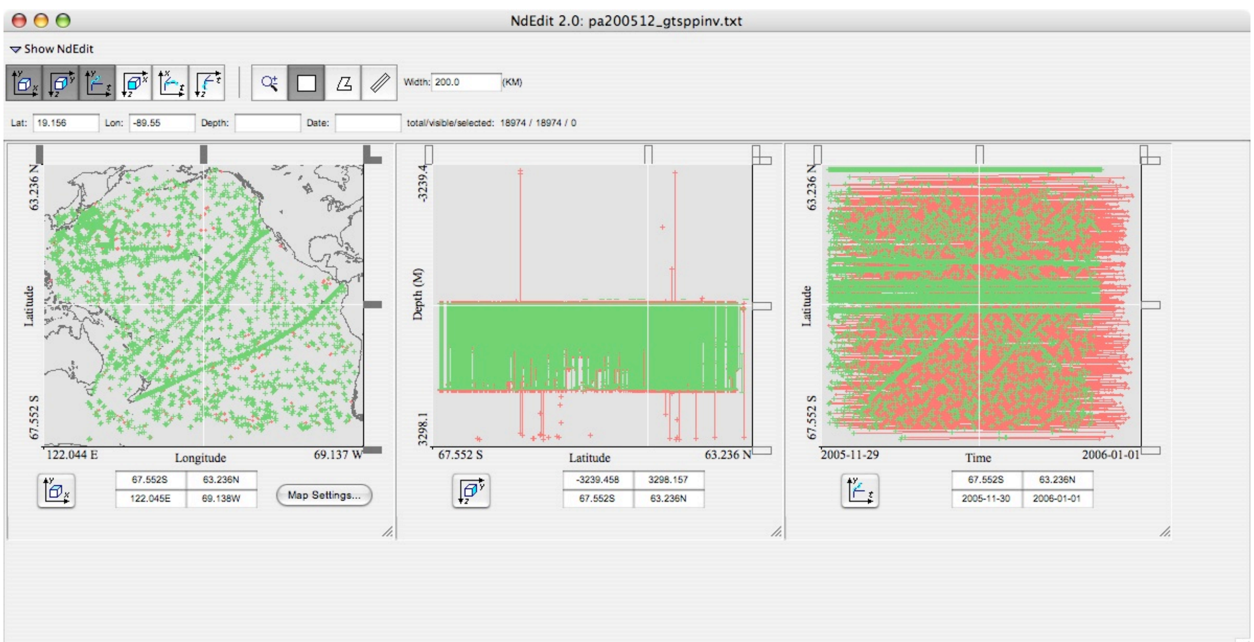
Example #3: Working With More Than One Inventory File

The NdEdit component can work with more than inventory file at a time. This allows comparison between different data sets and can make selecting synoptic datasets easier.

Open an Argo inventory file using the **Browse** command in the **File** menu:



Now use the same command and open a GTSPP pointer file:



The Argo profiles will be displayed in red while the GTSPP profiles will be displayed in green. The two pointer files are in separate virtual *layers* in NdEdit. You can now easily see how the datasets match up in time and space. To select data from NdEdit that has multiple data layers, select a data layer by right clicking (ctrl-clicking on Mac OS X without a two-button mouse) on any of the space-time panels. A popup menu appears that has the current layer checked:

Choose which layer you would like to select from by clicking its entry in the menu.

Limitation: NdEdit allows you to select from only one pointer file layer at a time and thus you are forced to create separate NQuery databases from multi-layer inventory files. A future version of NdEdit will allow selection from both inventories at the same time and thus will allow you to create an NQuery database that contains data from multiple sources.

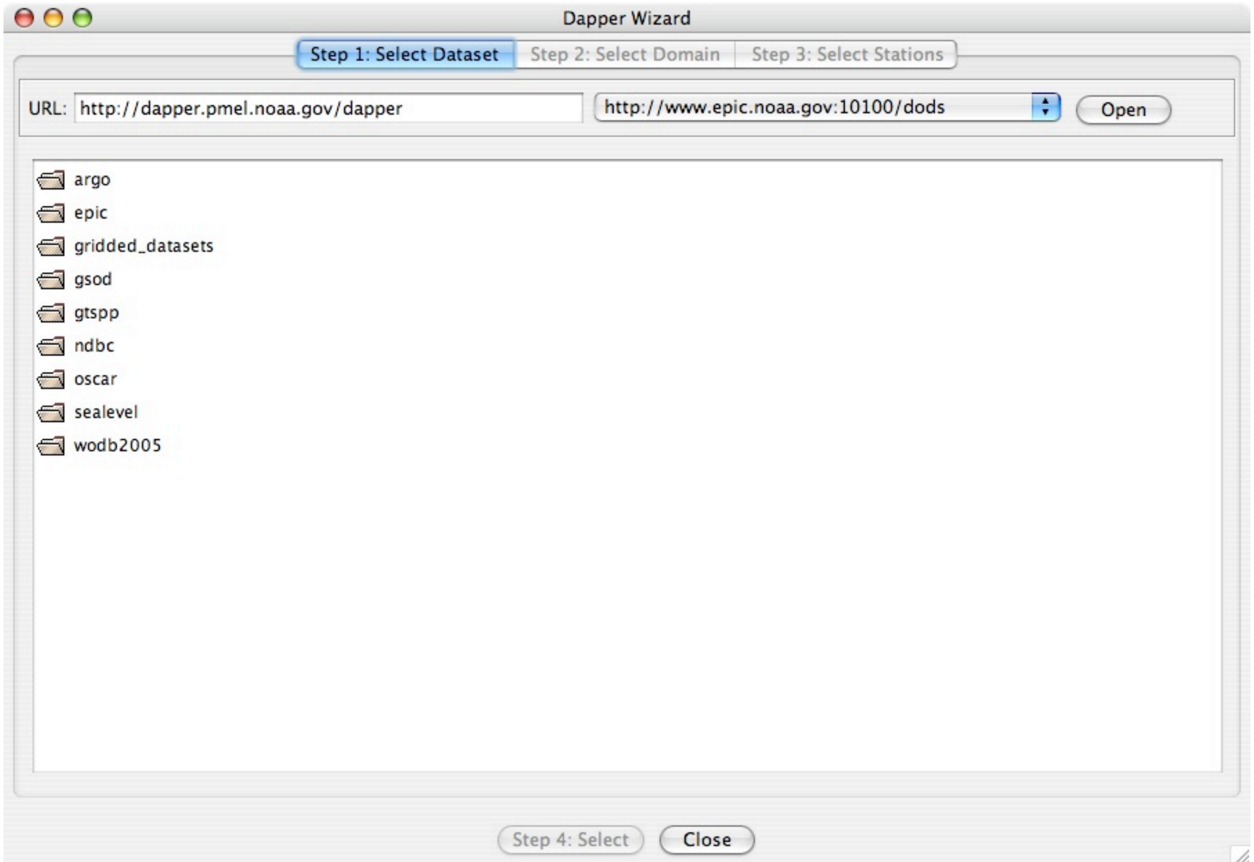
Example #4: Building a Database with Data from Dapper Servers

This example illustrates using JOA 5.0 to browse and extract data from a *Dapper* server. Dapper is a technology developed at NOAA/PMEL to allow access to large collections of profile data stored on central data servers. See http://www.epic.noaa.gov/epic/software/dapper/ for a more complete description of how Dapper works. This example will illustrate extracting data from the NODC World Ocean Database 2001 on a server at NOAA/PMEL. Note: you must have access to the internet to use the Dapper capabilities of JOA.

1) Launch JOA 5.0

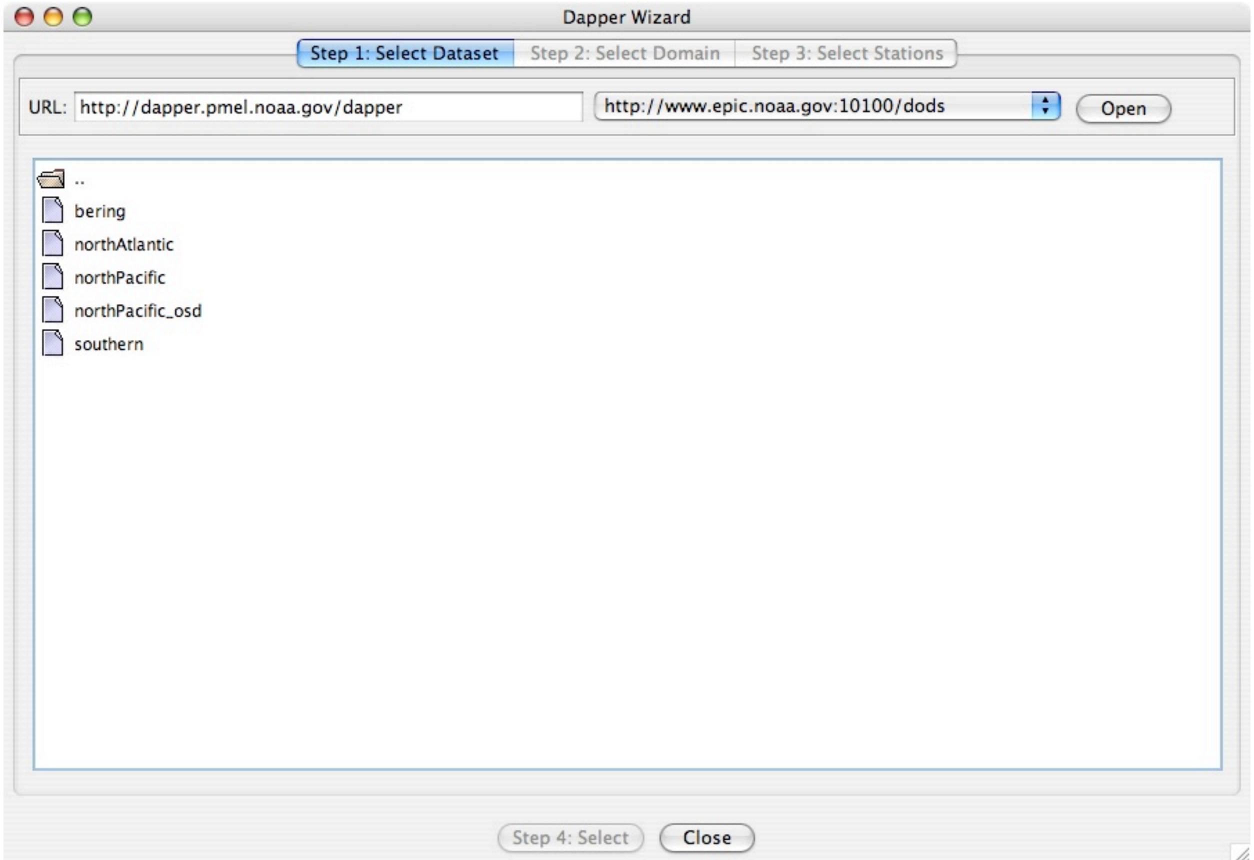
2) From the **File** menu select **Browse Dapper**. You will be presented with the *Dapper Wizard* that allows you to select a data collection, describe a geospatial/temporal domain in which to search for data, refine your selection using space and time filters, select profiles of interest, and open selected profiles in JOA.

Here is the first step of the four-part Dapper Wizard:



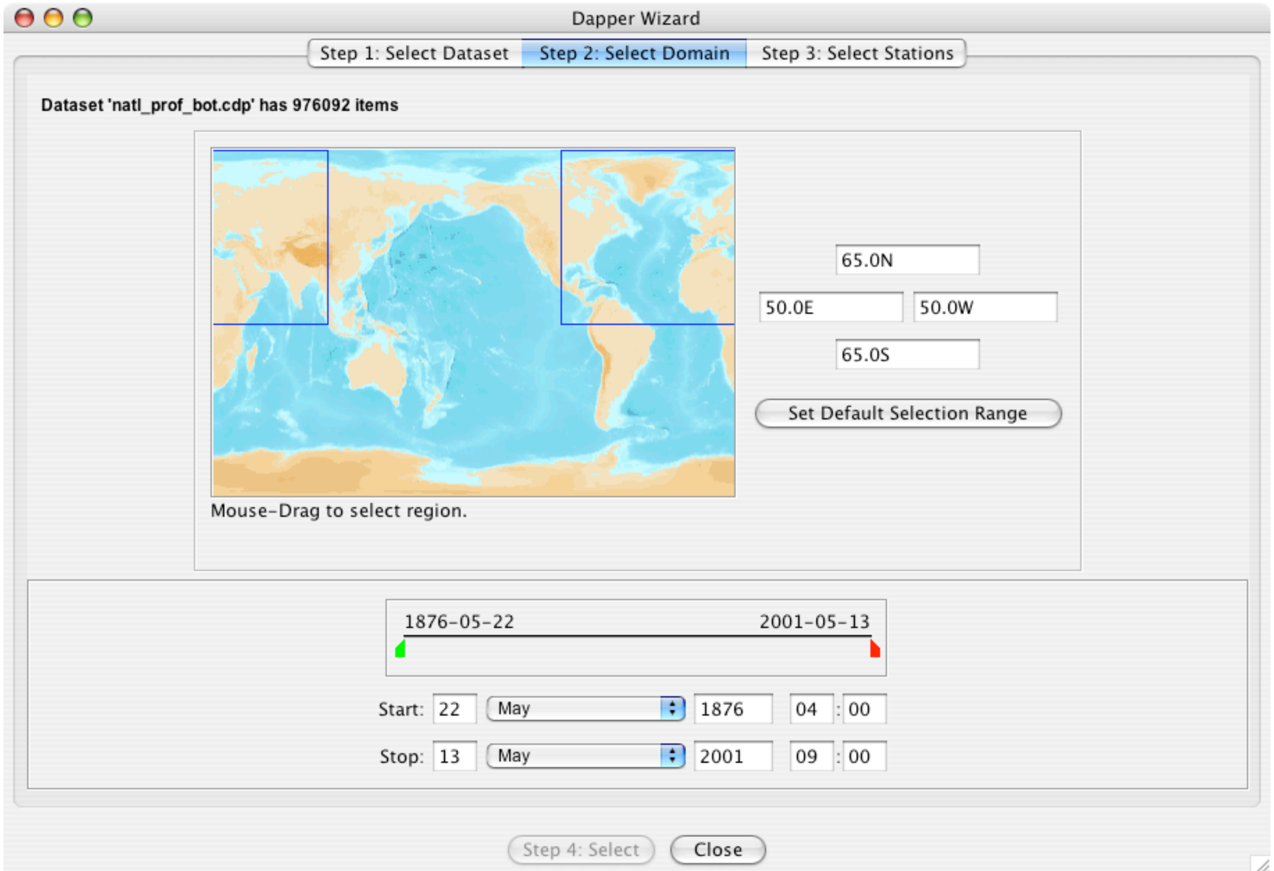
In this panel of the wizard, you enter the URL of a Dapper server (defaults to www.epic.noaa.gov:10100/dods), and select a data collection served at that address. In this example, there are 9 data collections available including *epic, argo, gtspp* and *wod2005*.

3) Double click the *wod2005*entry to browse datasets available in this collection:



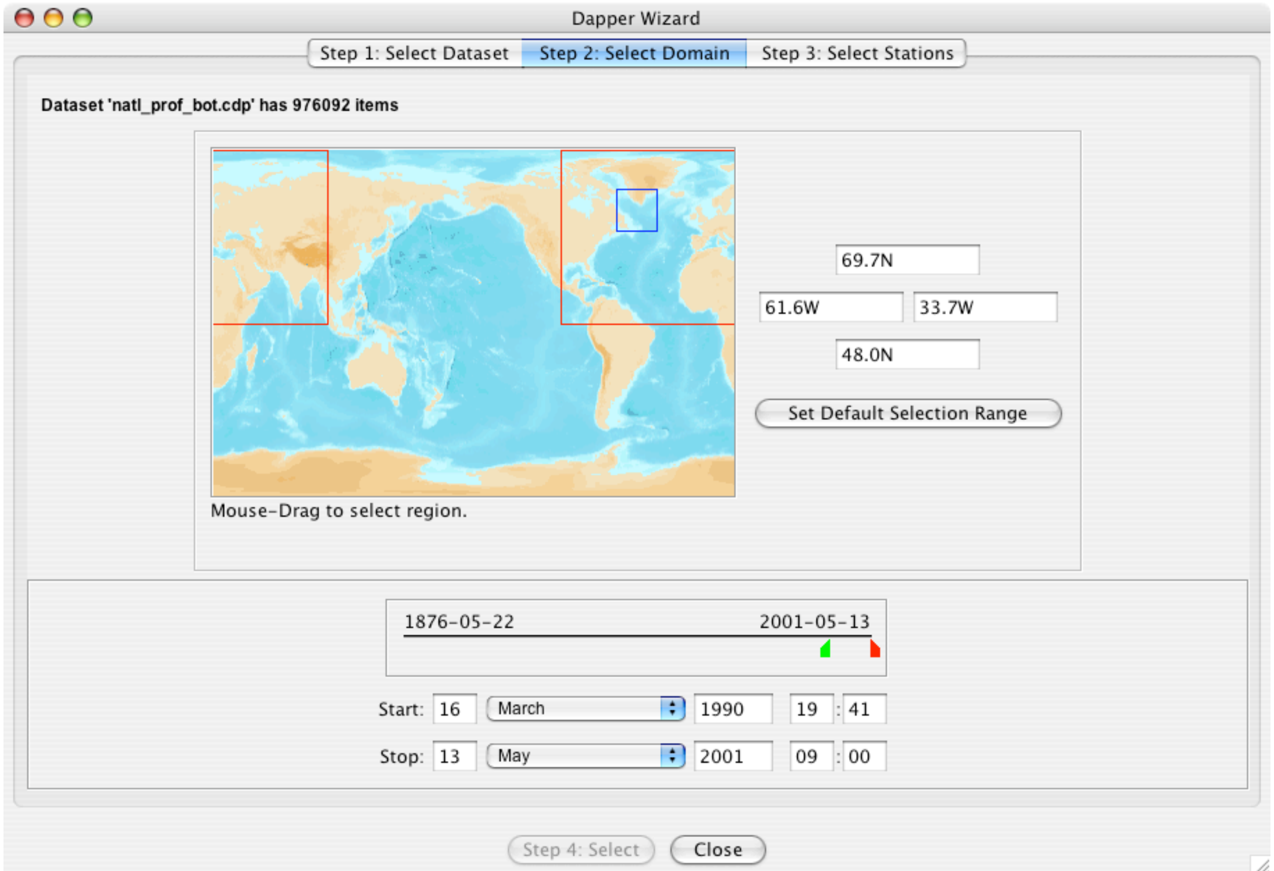
In the World Ocean Database 2005 collection, there are 5 datasets corresponding to ocean regions.

4) To go to the next step in the wizard, select a dataset and click the **Step 2: Select Domain** tab or simply double click the dataset of interest. To go back to the list of data collections, double click the top entry in the list (a folder icon with two dots). After a (sometimes lengthy) progress bar goes away, you will see the domain selector panel for the chosen dataset:

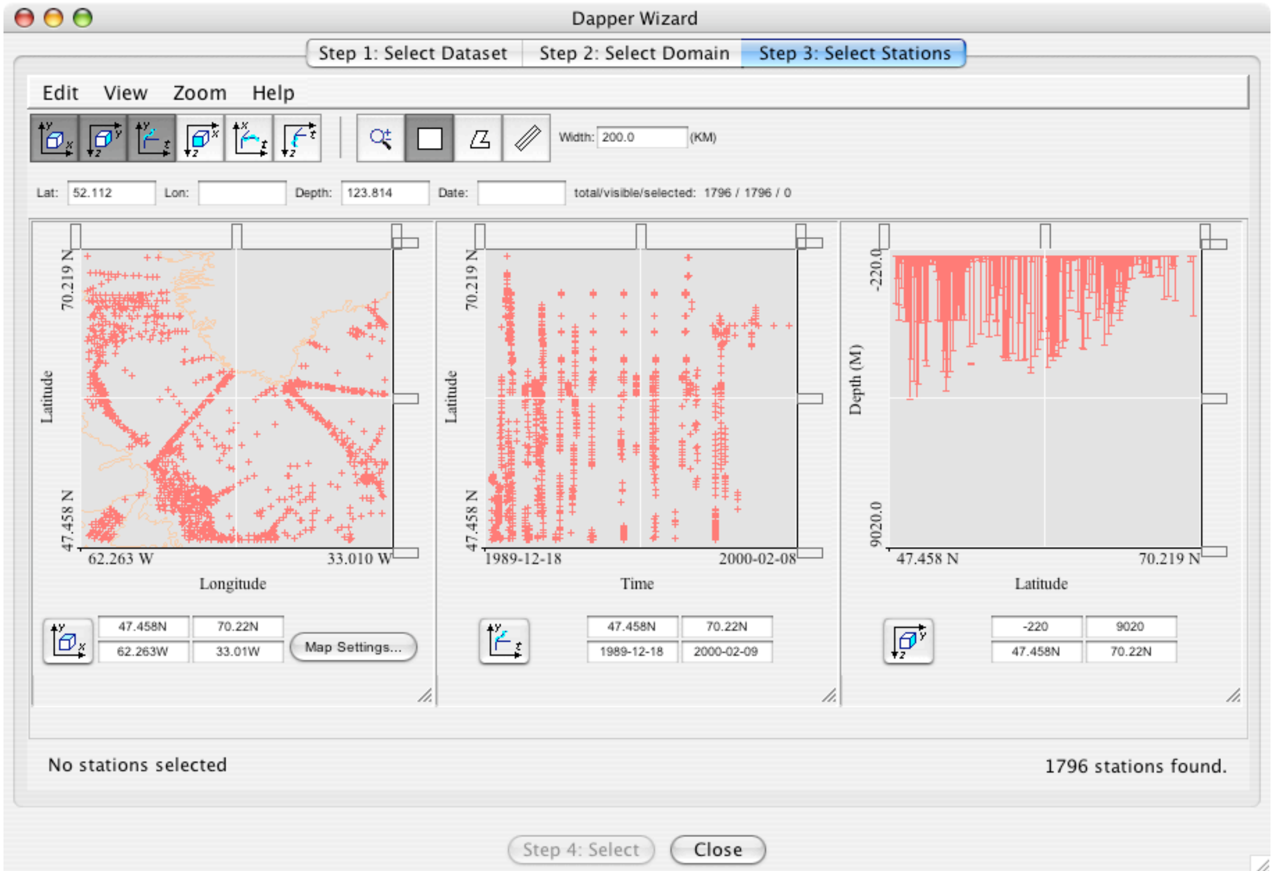


The map shows the spatial domain of the dataset and the time selector shows the temporal domain. Above the map is a label that lists how many profiles are in the dataset–in this dataset close to 1 million.

Because this is a large dataset, you will want to restrict the spatial and temporal domains. Dapper currently can return a maximum of 10,000 individual profiles. You can define a more restricted domain by dragging a rectangle onto the map (or entering explicit longitude/latitude coordinates in the fields to the right of the map) and using the handles on the time line control (or use the popup menus and text fields to enter an explicit date/time). For this example, select an area in the North Atlantic between Canada and Greenland and greatly reduce the time range:

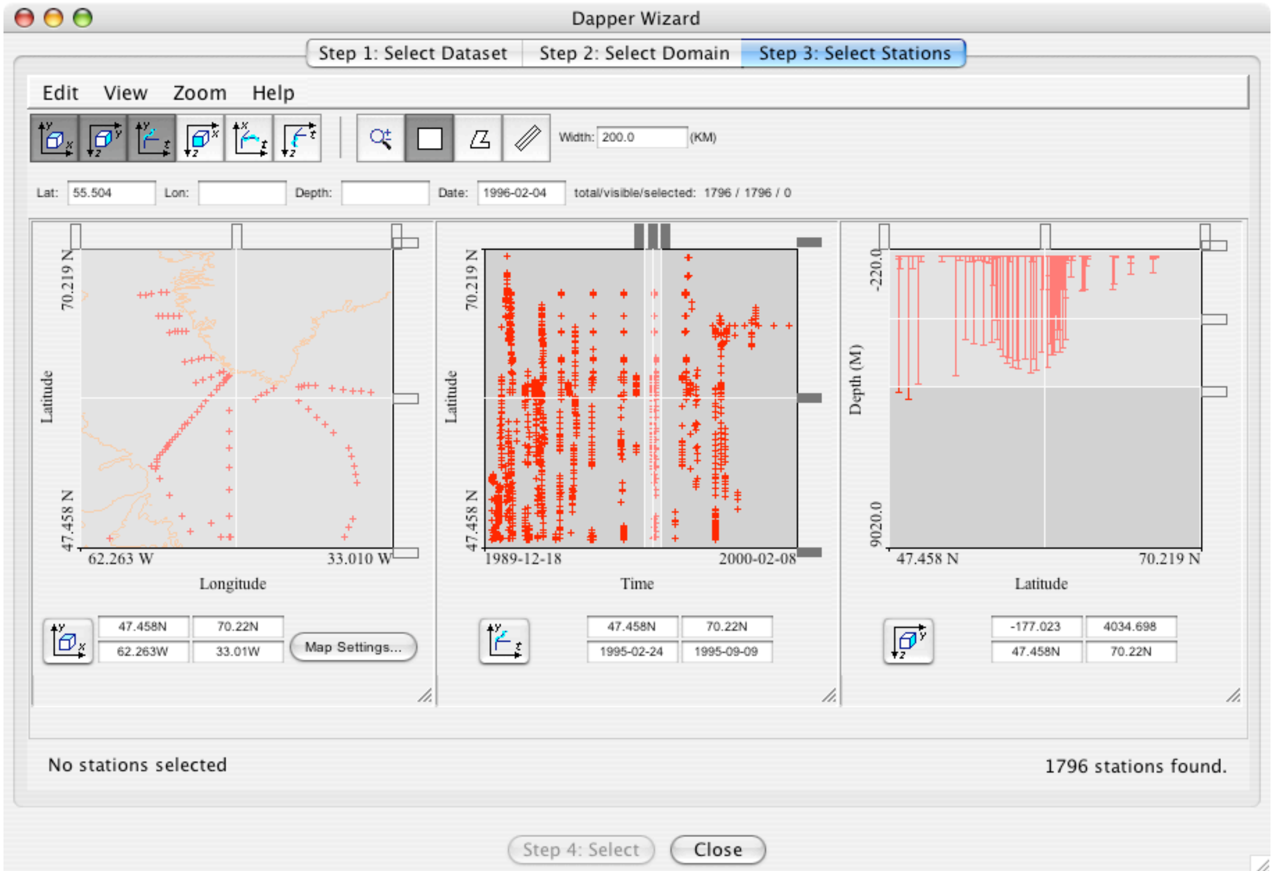


5) Click the **Step 3: Select Stations** tab to go to the fine selection panel of the wizard and select individual profiles for further analysis. At this point you will wait for the server to locate all the profiles in the space/time domain you specified. After the progress bar goes away you will see the individual profile locations displayed in the NdEdit data browser:

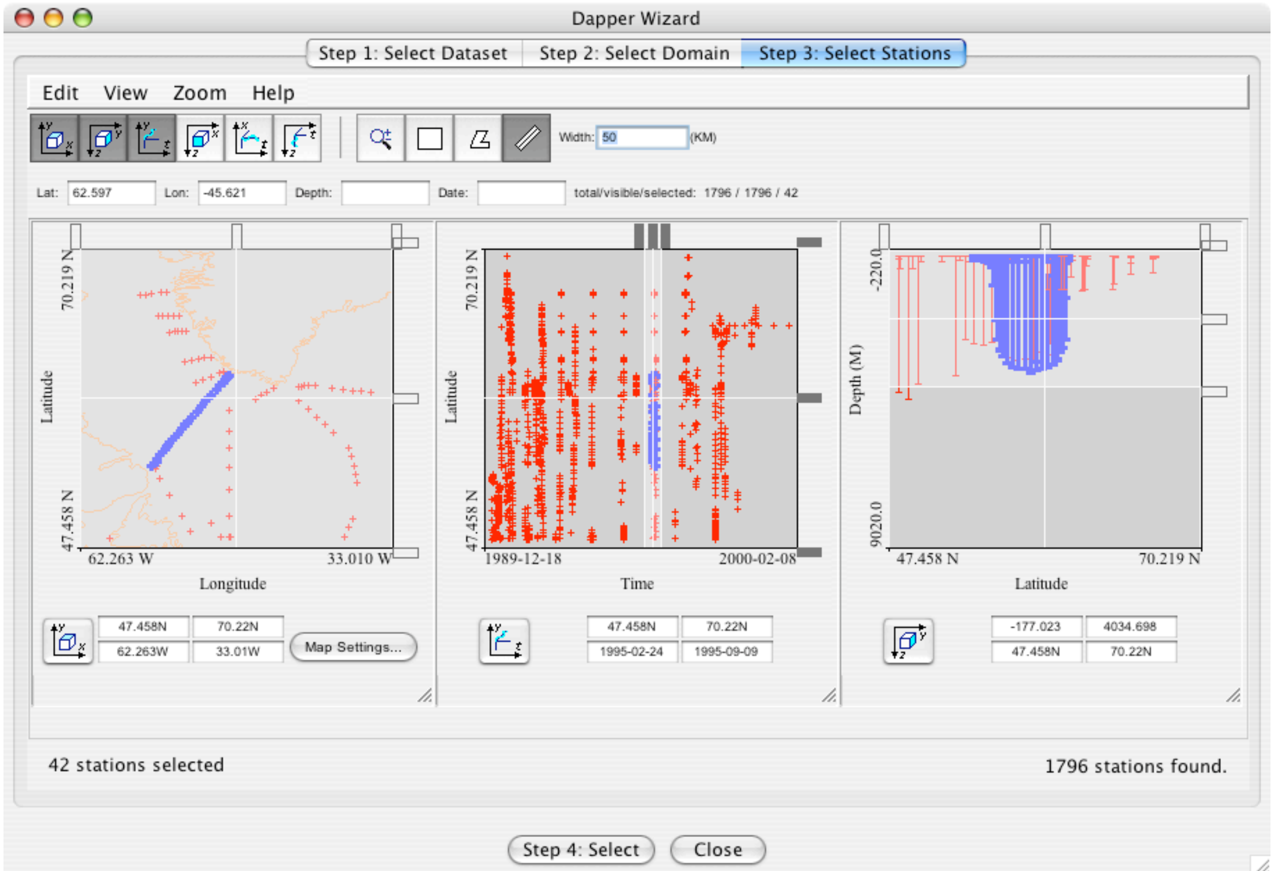


This search located 1,796 individual profiles. The default NdEdit view displays three panels for longitude/latitude, latitude/time, and latitude/depth slices of the dataset. If you have never used NdEdit, visit the NdEdit help and tutorials at http://www.epic.noaa.gov/epic/software/JavaNdedit.htm.

6) You can use NdEdit's interactive filters to reduce the number of visible profiles. Notice how many profiles have disappeared after narrowing the time domain:

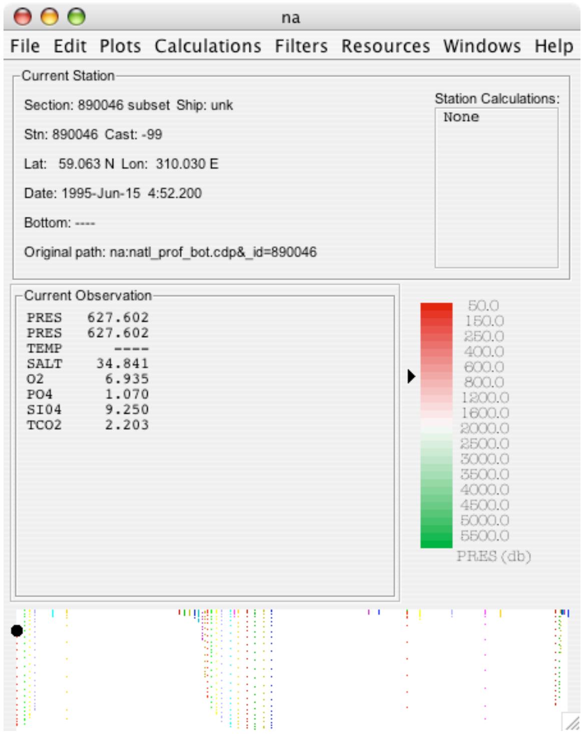


7) Use the section tool to select a line of profiles across the Labrador Sea:



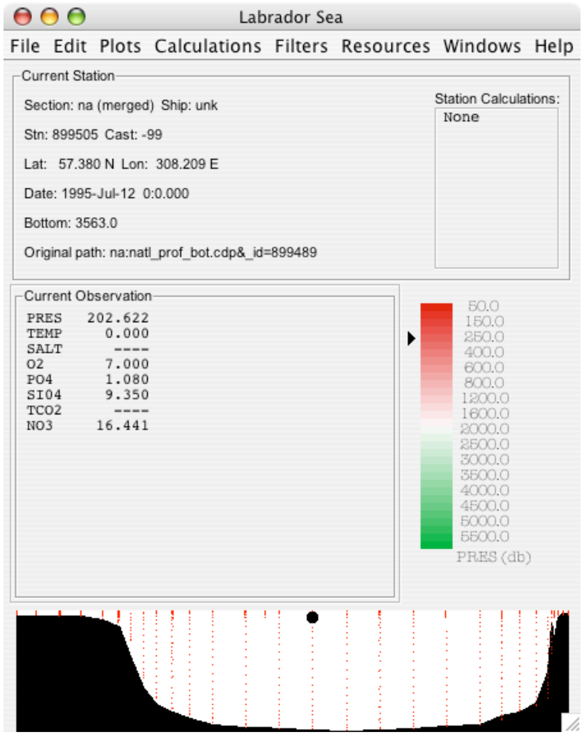
The 42 selected profiles are displayed in blue.

8) To open the selected profiles in JOA, click the **Step 4: Select** button. The Dapper Wizard now will go back to the server and retrieve the data for each profile. JOA will convert the data to its internal format and open a new data window:

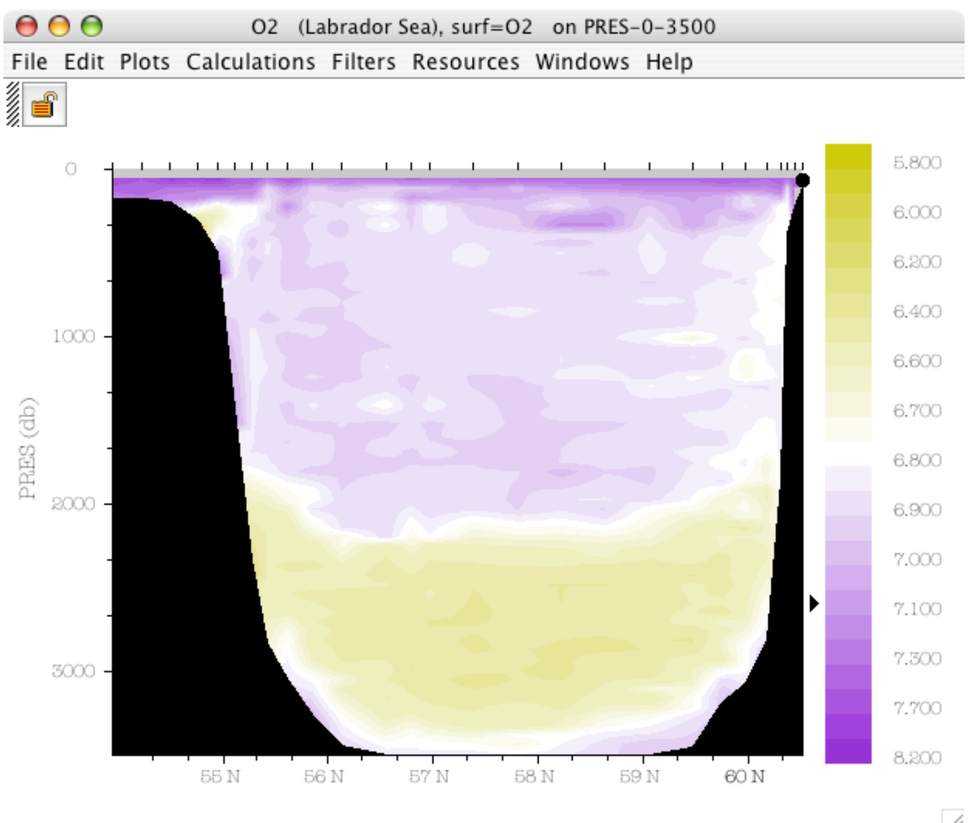


The first thing you may notice is that the profiles are out of section order. This is because the server does not have any concept of a section and returns profiles as they are found in the database.

9) Use the JOA Section Manager to merge the profiles into a single section, sort ascending by latitude, and add an approximate ocean bottom:



10) Visualize the data. Here is a plot of dissolved oxygen across this section:



Note: I further filtered the profiles in JOA to only include profiles from June of 1995 to make a smoother plot. As shown in the following profile plot offset by time, the June profiles (left clump) are very different than the July profiles (right clump):

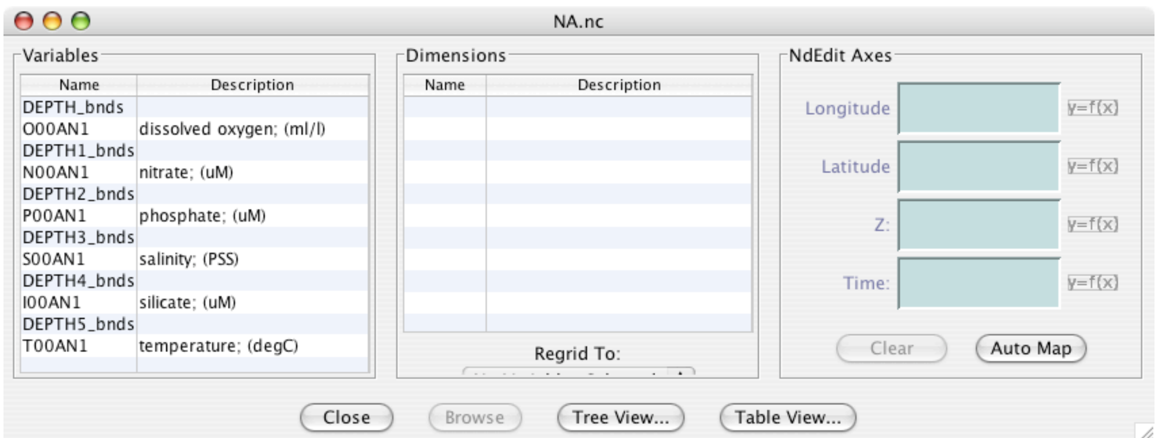
Extracting Data From Gridded NetCDF Files

This example illustrates using JOA 5.0 to extract sections from Levitus 1° annual average netCDF files.

1) Launch JOA 5.0

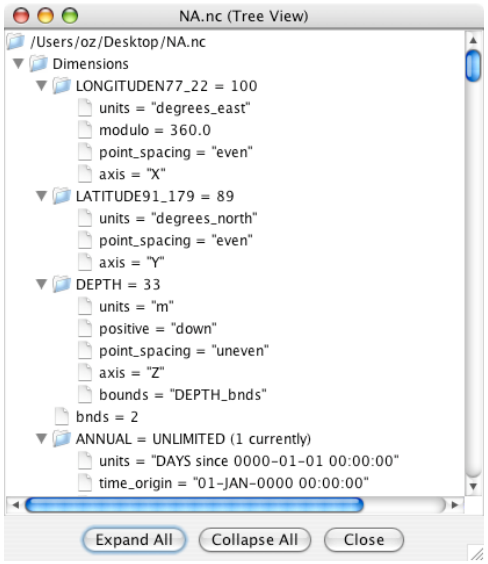
2) From the **File** menu select **Browse**

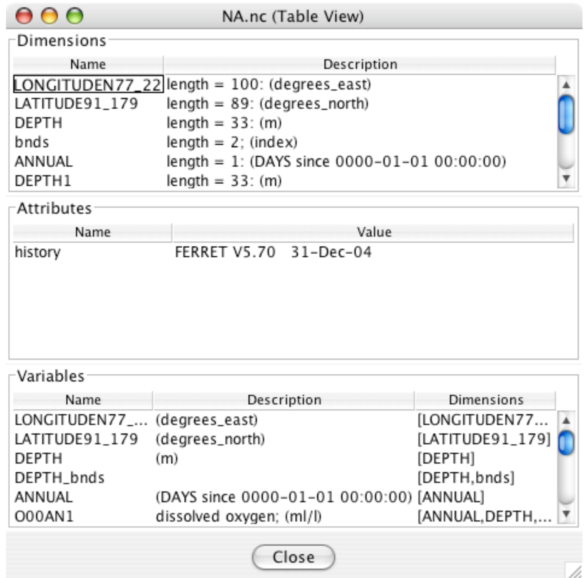
3) In the file chooser dialog, select NA.nc, SP.nc, or NP.nc. You will see the netCDF structure-browsing window:



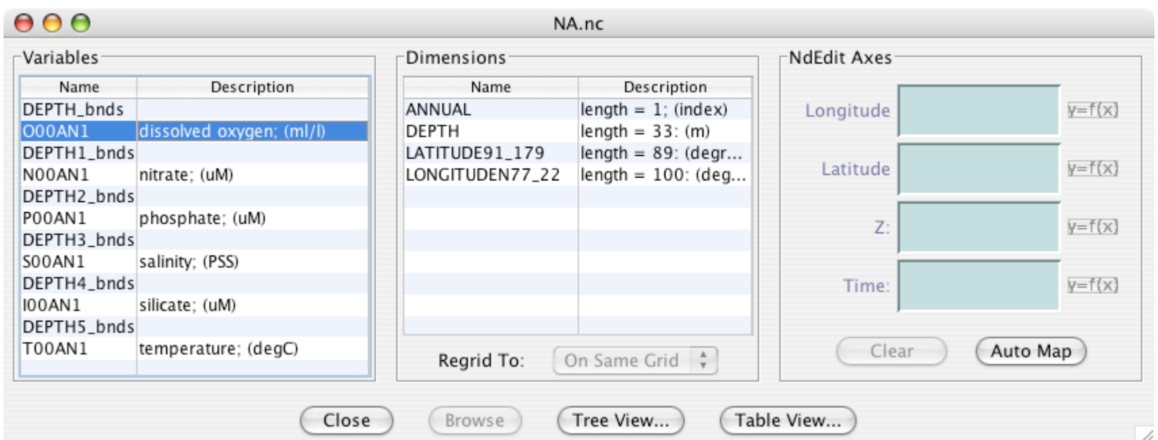
The table on the left shows all the *Variables* in the data file. The middle table will show the *Dimensions* for the selected variables in the Variables table. The right table is used to map the dimensions of the selected variables into the standard geospatial/temporal *axes* (longitude, latitude, depth, and time). This dialog is designed to accommodate generic netCDF files so is not JOA specific. However, you will notice that the goal of this browser is to assign the dimensions of a netCDF file to the standard temporal/geospatial axes used in JOA.

The Table View and Tree View buttons allow you to browse a netCDF file to see more of its structure such as global and variable attributes. For the example data file, here is what the tree and table views look like:



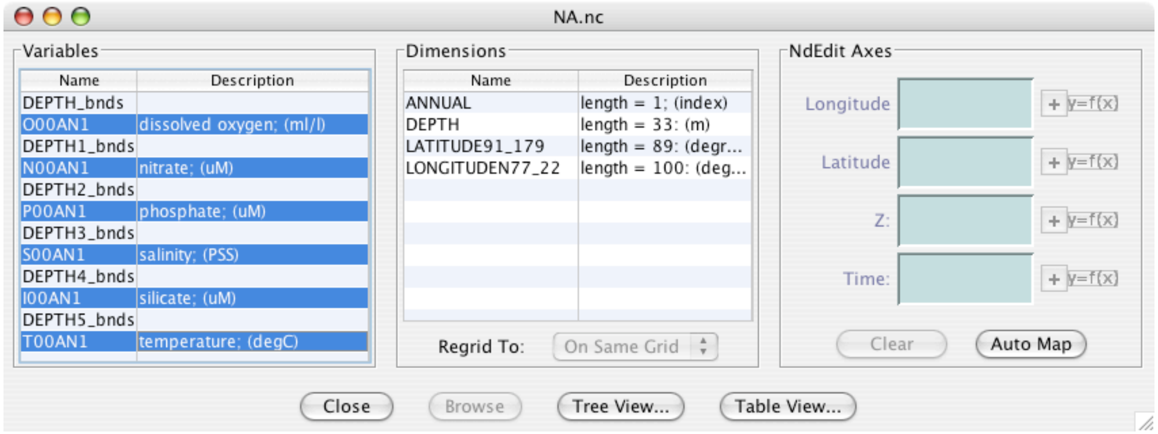


4) Select a measured variable from the Variable table:

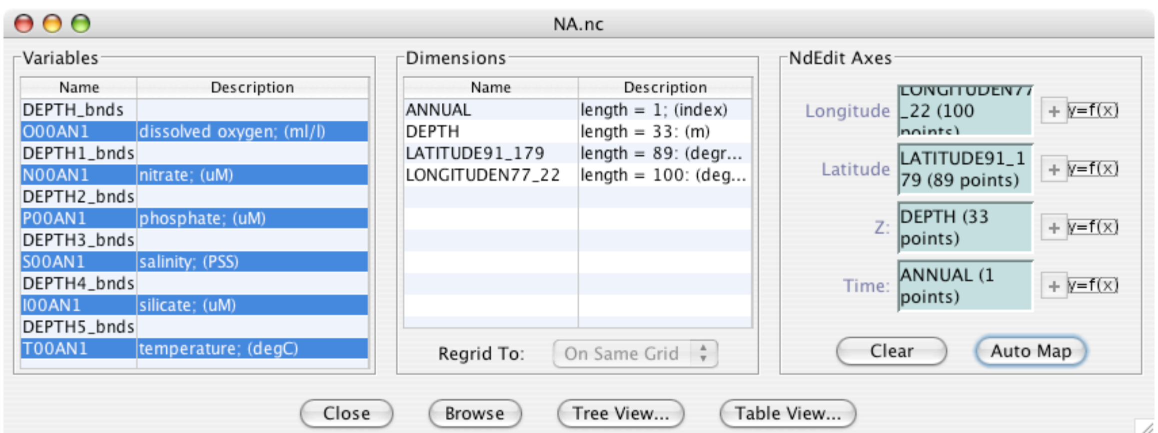


You will notice that the Dimensions table is now filled in with the dimensions of the selected variable. You can tell that the variable shown here has dimensions that are geospatial.

5) Command click to select all the observed variables:

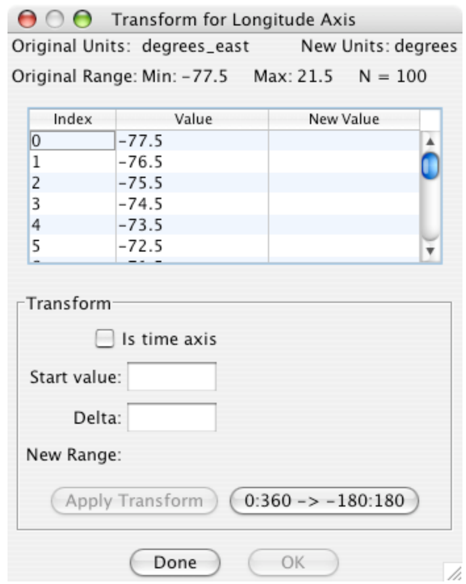


6) Next you need to “map” (or assign) the dimensions of the selected variables to the standard geospatial axes by clicking the **Auto Map** button. Note: you can also manually map dimensions to axes by dragging a dimension from the middle table to the light blue-green drag “targets” in the NdEdit Axes pane. Auto Map tries to match the name of dimensions to the names of the canonical axes, longitude, latitude, z, and time. For the example file shown here, the auto mapping has been able to map the dimensions successfully to the required axes. Here’s what the dialog looks like after **Auto Map**:



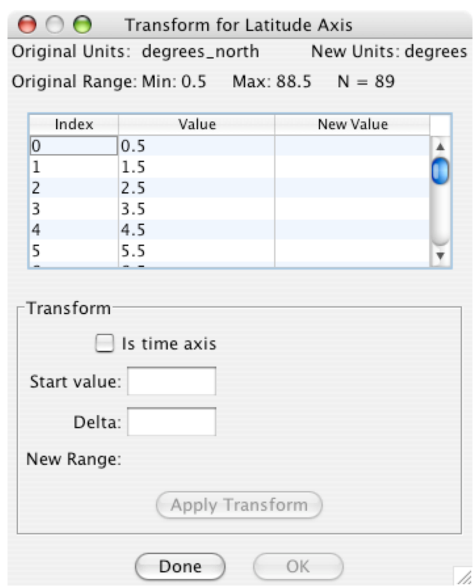
Note: If an axis cannot be assigned by mapping, the small plus (**+**) button next to the axis drag target will become available. Clicking this button allows you to assign an axis if is missing in the source file.

7) Click the **Transform** (y=f(x)) button next to each axis to check out the values for each axis. Clicking the **Transform** button for the Longitude axis presents:



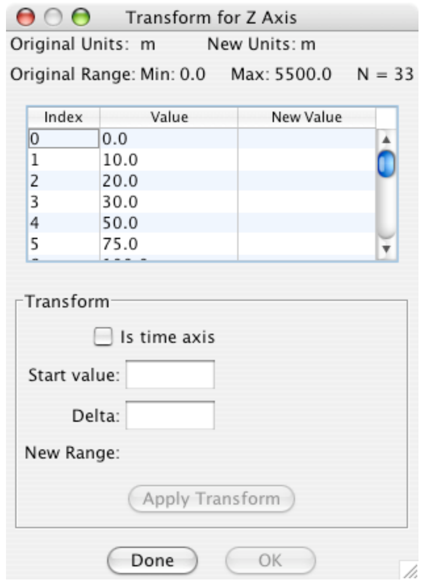
The values for longitudes look pretty reasonable so just close the dialog (Click **Done**).

Click the **Transform** button for the Latitude axis:



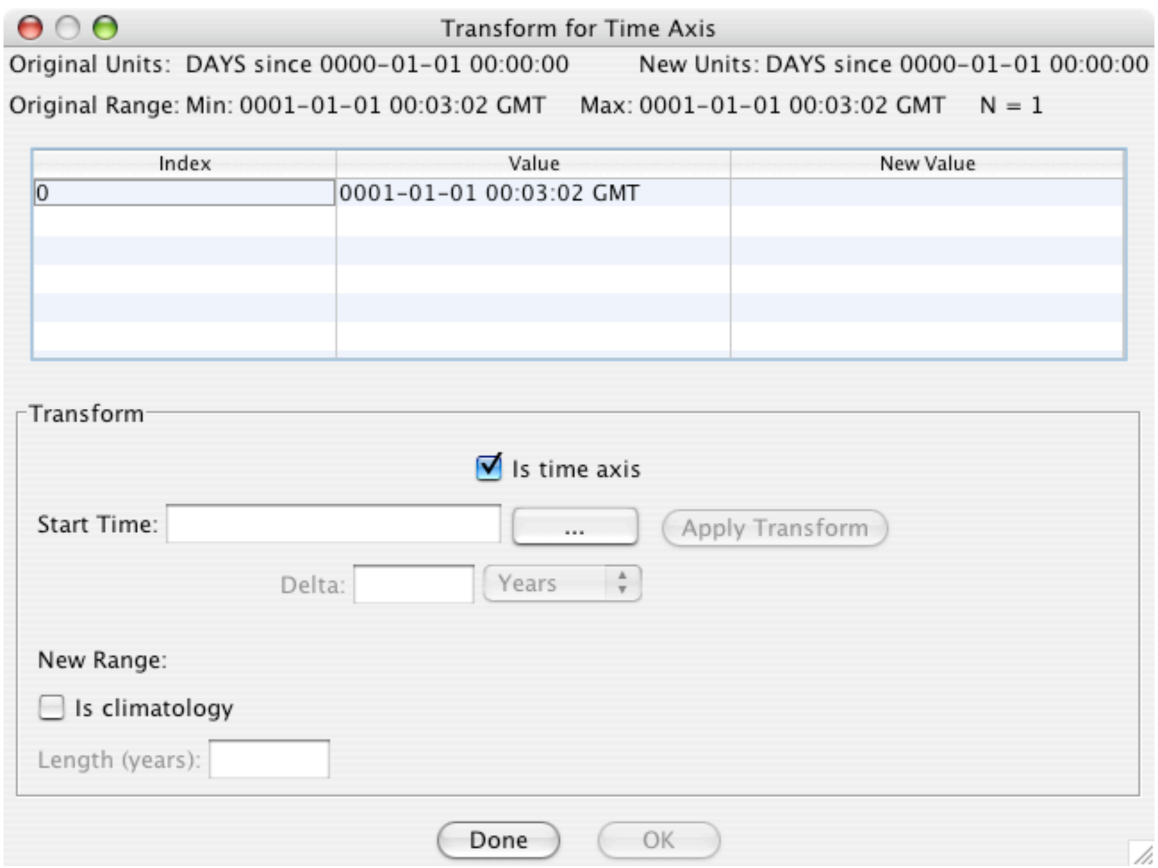
Again, the values for latitude look pretty reasonable so just close the dialog.

Click the **Transform** button for the Z axis:



The values for Z look pretty reasonable so just close the dialog.

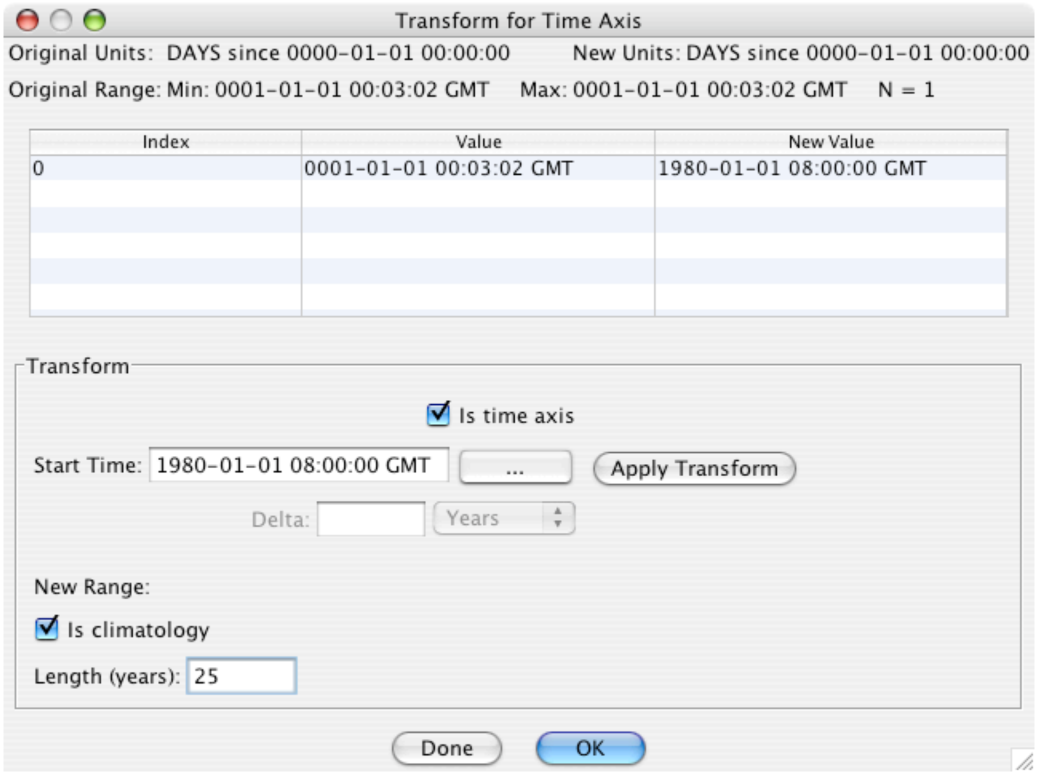
Click the **Transform** for the Time axis:



You may wish to assign a real date to the section you extract from the file. You can either type in a new start time in the format “YYYY-mm-dd hh:mm:ss” or use the “calendar” dialog by clicking on the ellipsis (**…**) button:

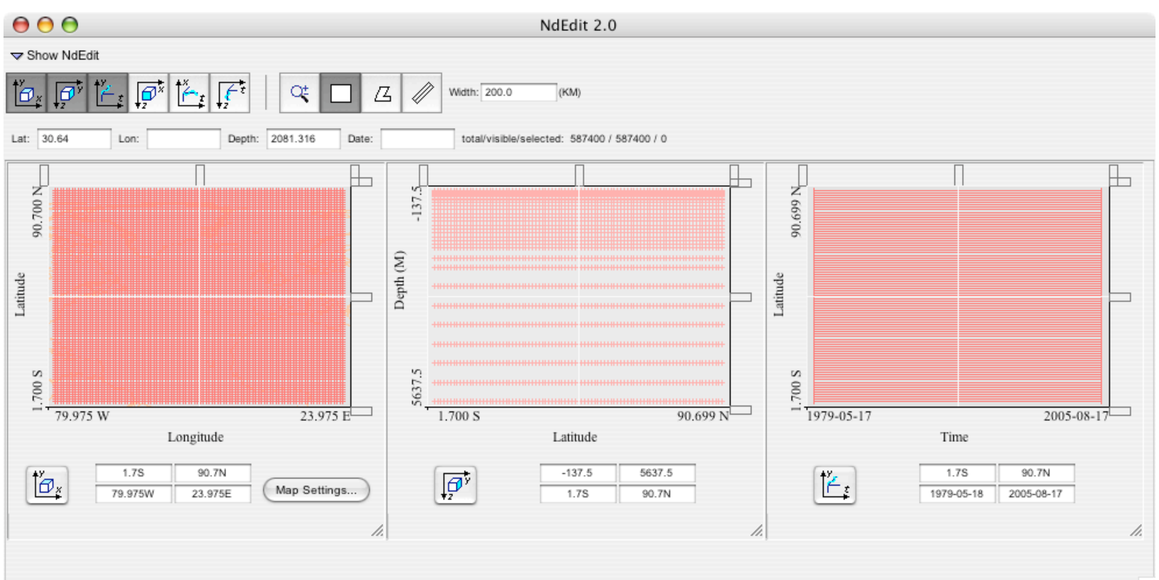
Notice that the initial date is today’s date. To accept today’s date, click the **OK** button or use the dialog to build a different start date. Note: If you are going to be comparing data with other, perhaps observed, data, make the start date before the range of your observed data, When you are finished, click the **OK** button.

You must provide a time delta and time units only if the time axis has more than one point. In this example, there is only a single time, so these buttons are dimmed. After you have a start date, delta, and units (if applicable), click the **Apply Transform** button:

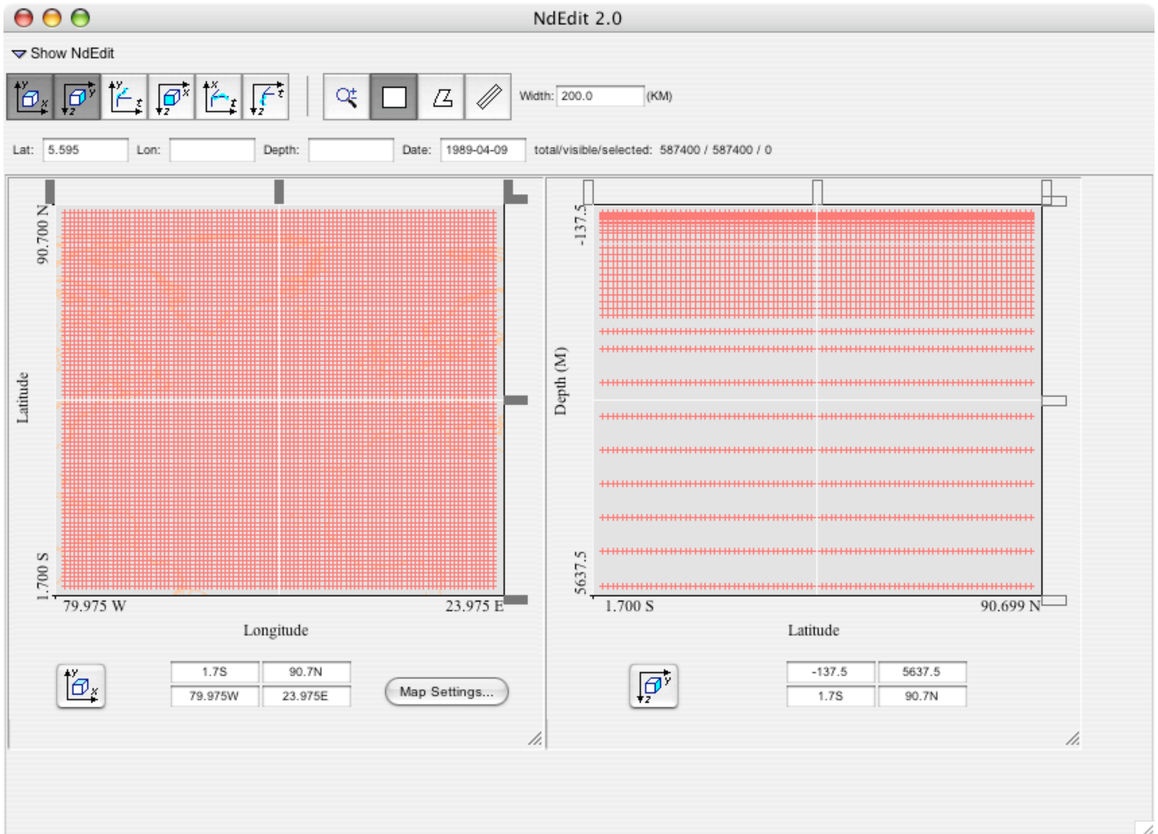


In the table you see the old values next to the new values. To compare data from files with a single-valued time axis with other data, you have to make sure the time axis spans the range of your observed data. To do this, click the **Is climatology** box and provide the length of the extended time axis in years. In the above example, I have extended the (transformed) start date to go from January 1st, 1980 to January 1st, 2005

8) Click the **Browse** button. If you haven't already, click the **Show NdEdit** button in the JOA "starter" window. You will see this display (for the NA.nc file):

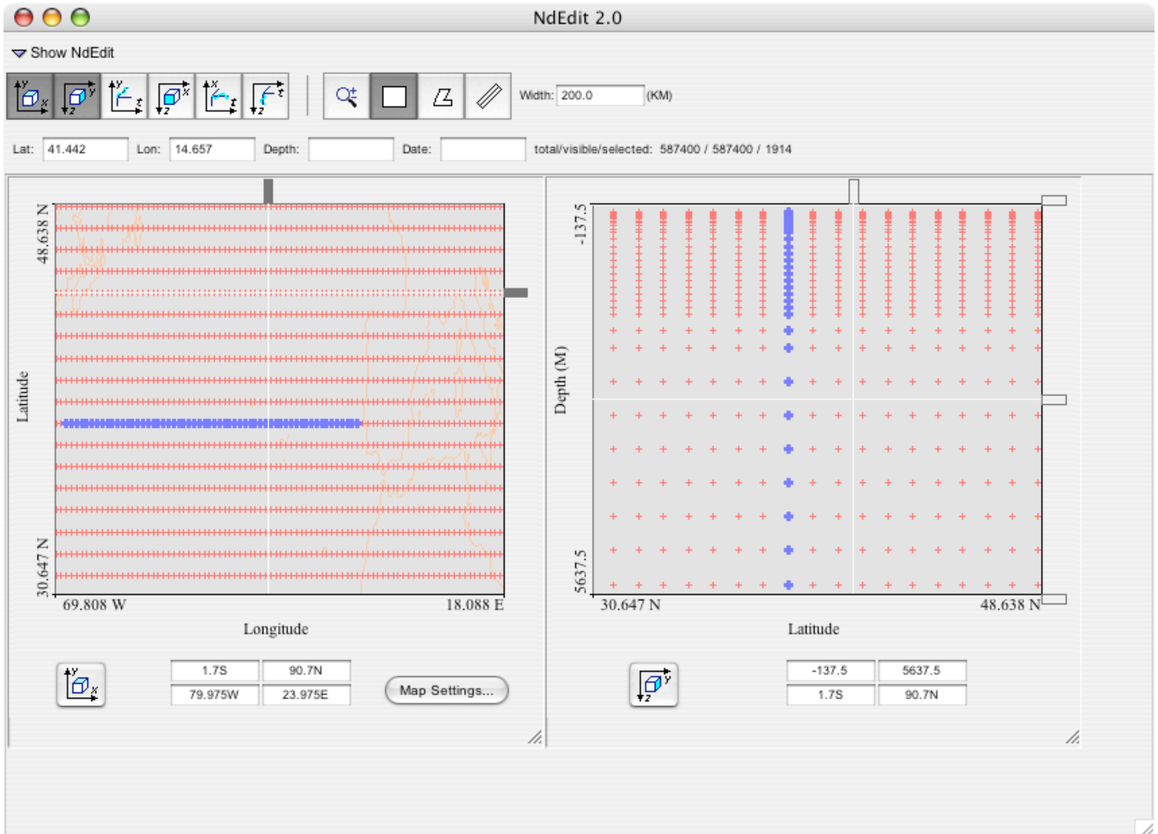


NdEdit now shows the grid structure of the input file. Note in this example, the time/longitude panel shows the single time point expanded to a range of values. You can close this panel and resize the latitude/longitude panel so it's easier to see the underlying coastline:

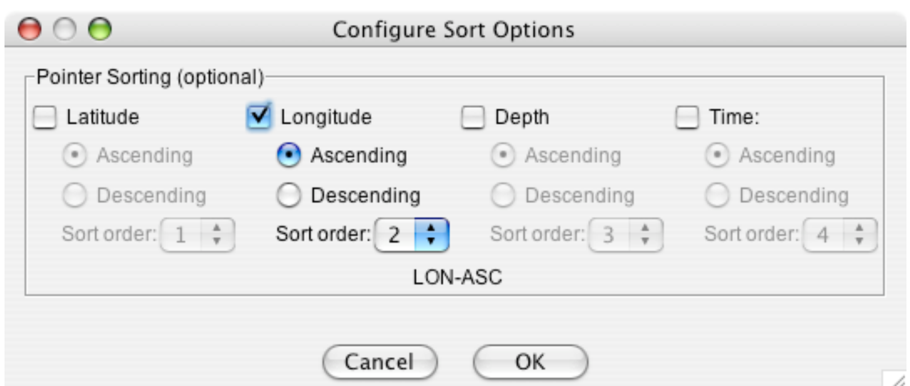


Note: since the grid is evenly spaced in both latitude and longitude, showing the longitude/depth panel doesn't really add any new information.

At this point you can use any of NdEdit's tools to select profiles from the gridded file. Here I zoomed in on the mid-Atlantic Ocean and selected a section across about 38° north latitude:

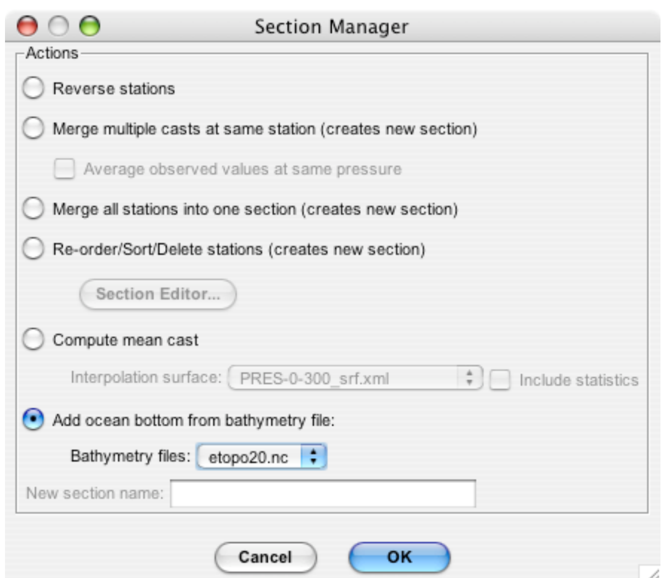


To import this section into JOA, select the **Open selected files** command in the **File** menu: You will be prompted for how you wish the profiles sorted:

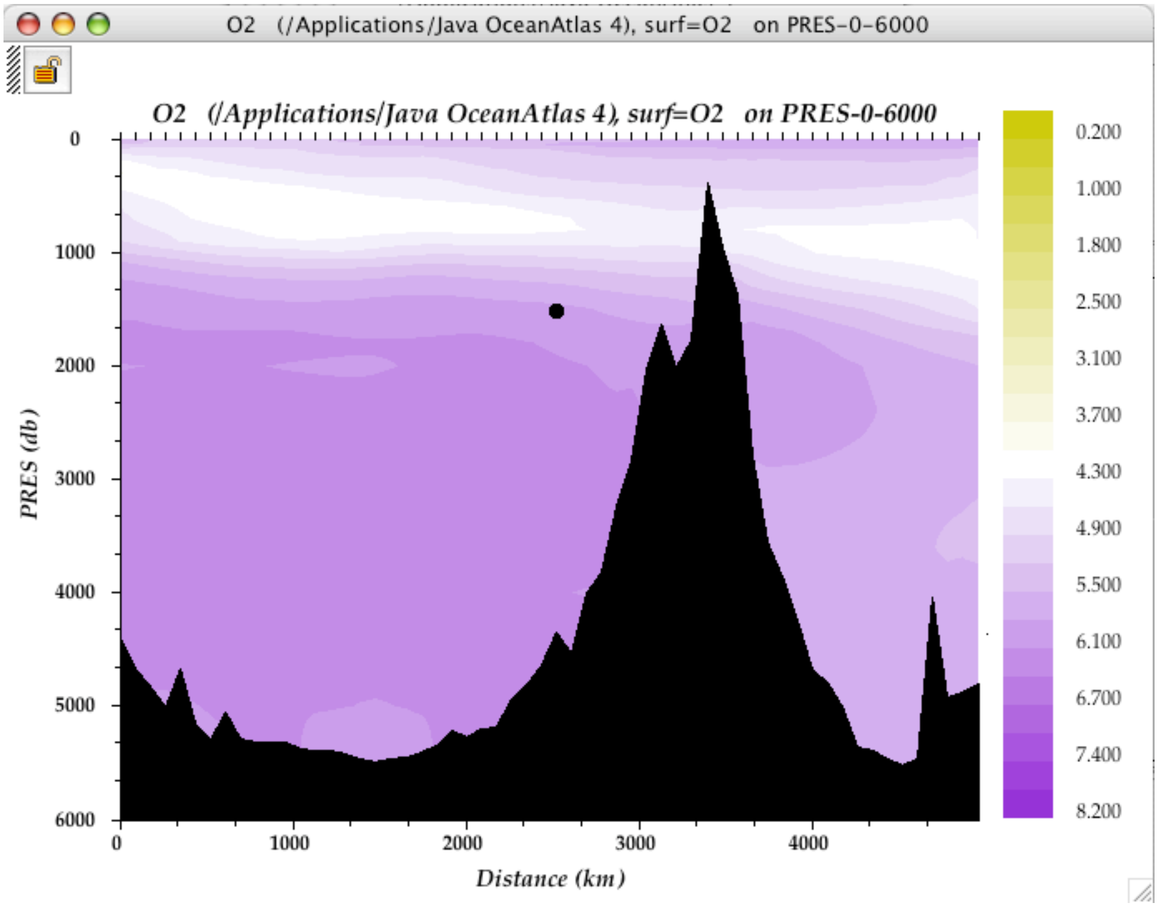


Here I chose to sort profiles by ascending values of longitude. Click the **OK** button to extract selected profiles from the gridded file and open into JOA:

Using the Section Manager I added an approximate ocean bottom from the etopo20.nc bathymetry file:



And finally, here is a contoured section from the derived profiles (Note: I used the station filter to trim off the eastern end of the section):



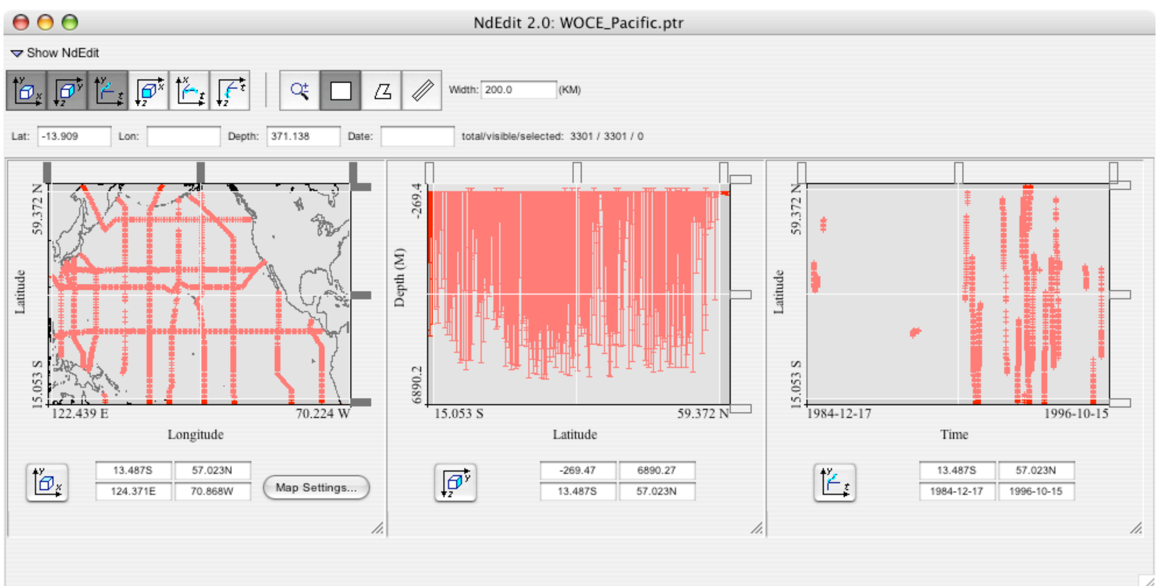
At this point you can save the file as either a JOA binary file or export data as a spreadsheet or netCDF section.

Comparing Gridded Data with Observed Data

Using NdEdit, JOA users can browse and select data from different types of data sets. This example illustrates using JOA to extract and compare data from a gridded data file (World Ocean Atlas 2001 Annual Average) and WOCE observed bottle data.

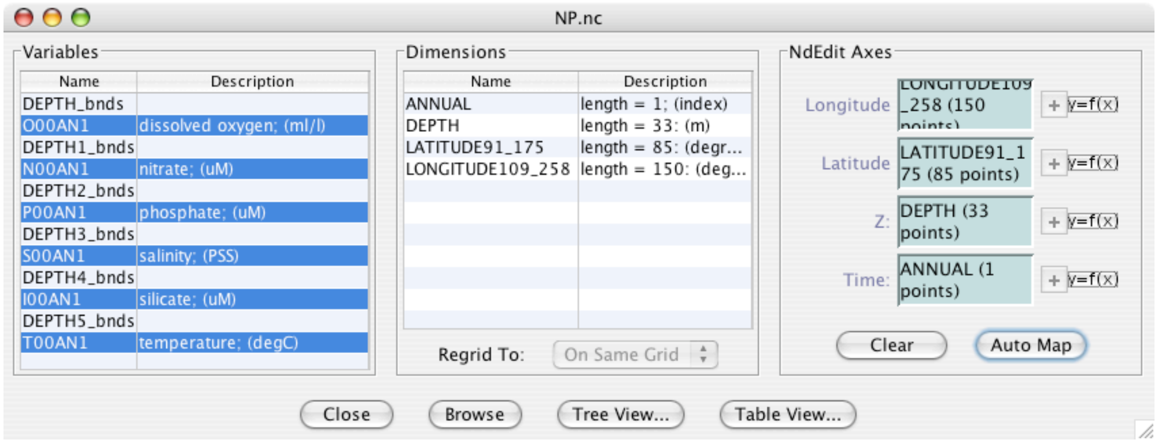
2) Create a “database” of observed data. You can do this in different ways. In this example, the database was created from 24 WOCE Pacific bottle sections downloaded from: http://odf.ucsd.edu/joa/data/whp/whp_pac.html. The individual files were opened in JOA by successive adding to an existing data window. After all 24 files were opened, the individual profiles were exported to a netCDF section by selecting **Export netCDF Section** from the **File** menu. This produces individual netCDF files for each profile along with a “pointer” file. Note: it’s a good idea to create a new folder for your extracted netCDF profiles. In this example, the resulting database has 3,289 individual profiles.

3) Open the pointer file associated with your observed data in NdEdit using the **Browse** command from the **File** menu in the JOA initial window. After a few seconds the pointer file will be displayed in the NdEdit browser:

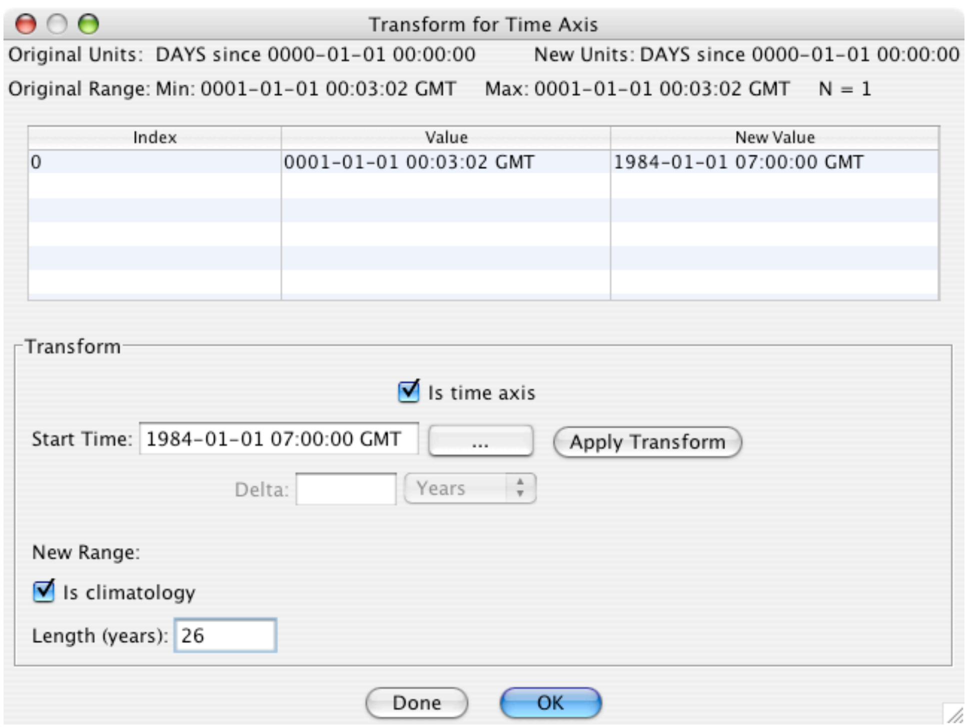


Note: The map panel has been zoomed into the North Pacific.

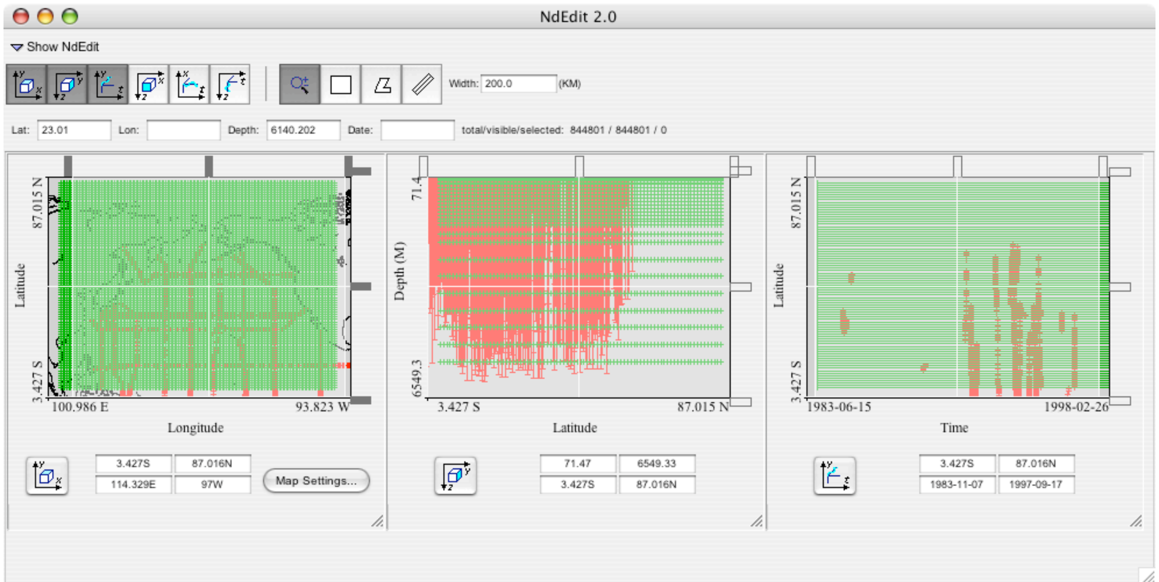
4) Open a gridded file for comparison. In this example the World Ocean Atlas 2001 Annual Average data file is used. Open NP.nc into the NdEdit browser using the **Browse** command in the **File** menu:



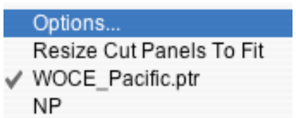
5) Select all the observed variables in the data file and click **Auto Map** to map the dimensions for these variables to the standard NdEdit geospatial/temporal axes. However, the time axis for this file is only a single point, use the Transform dialog to expand it so that it can be compared it with the range of time for the observed data. Here is the Time Axis transform dialog filled in to expand the time axis to span the range of dates in the observed data:



6) Click the **Browse** button to add the gridded data to the NdEdit browser. You will see the grid structure of the netCDF file added as a new layer to NdEdit in a contrasting color:

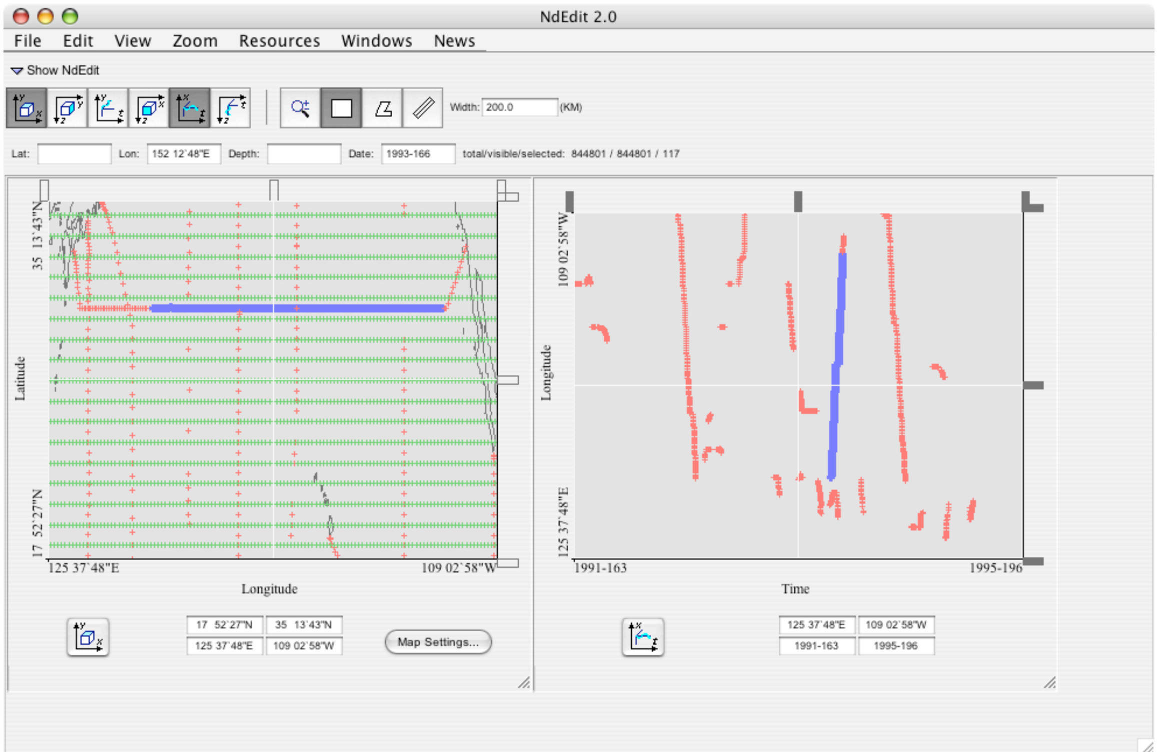


NdEdit allows selections from only one data layer at a time. To choose the selection layer, right click on any of the NdEdit panels to display a contextual menu:

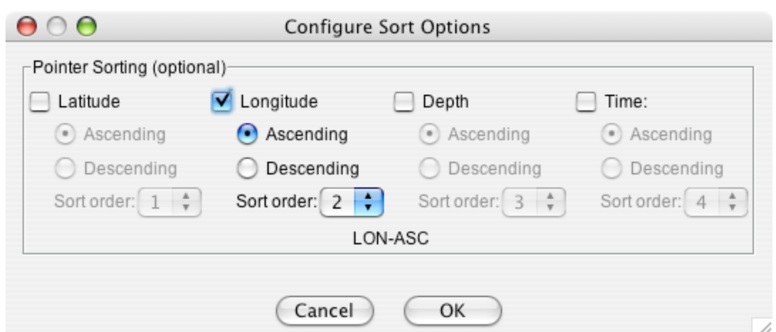


Note: if you don't have a two-button mouse on a Mac OS X computer, ctrl click to display the popup menu. This menu shows that selections are currently set to the WOCE observed data (red in the above illustrations).

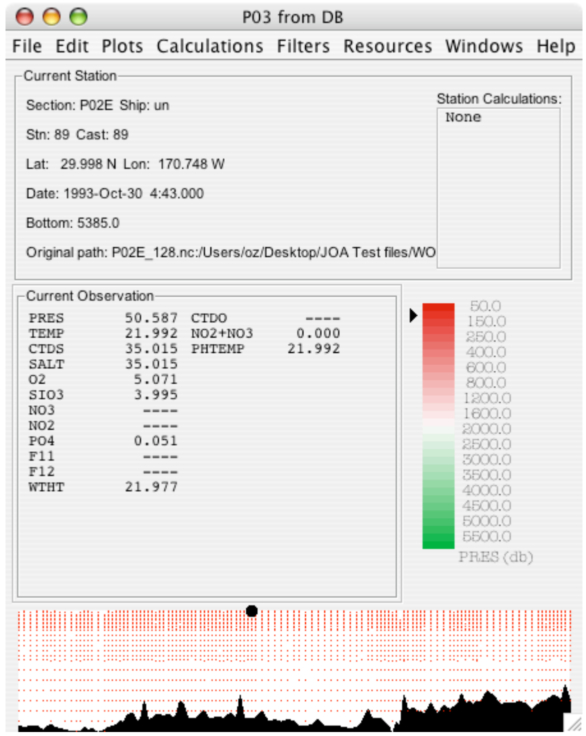
7) Use the NdEdit filters and zoom tools to focus on the area time of interest. Here a combination of zooming and the time filter have been used to focus on a longitudinal section in the WOCE data in the North Pacific. Using the rectangle tool, select a WOCE section from:



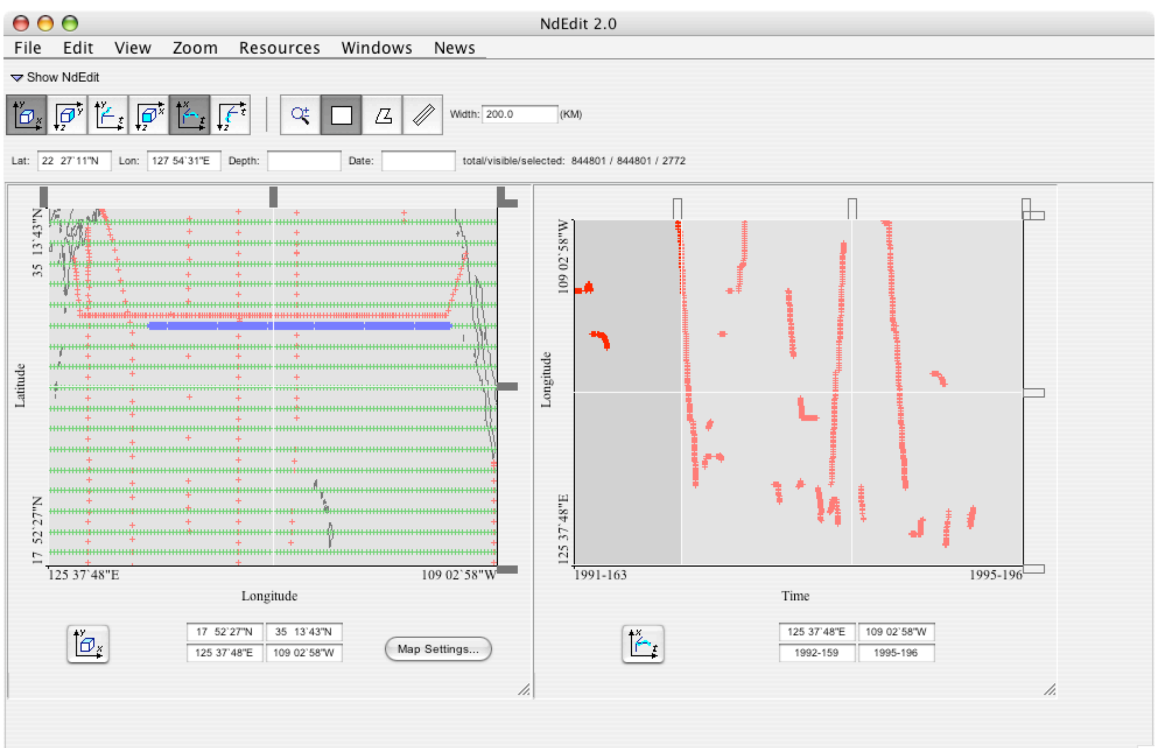
8) Open the selected WOCE data into JOA by selecting the **Open selected files** command in the File menu. You will be prompted for optional pre-sorting of the data:



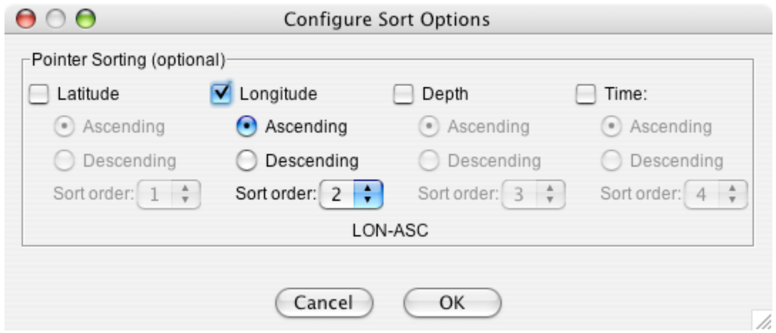
The selected WOCE data opened in JOA:



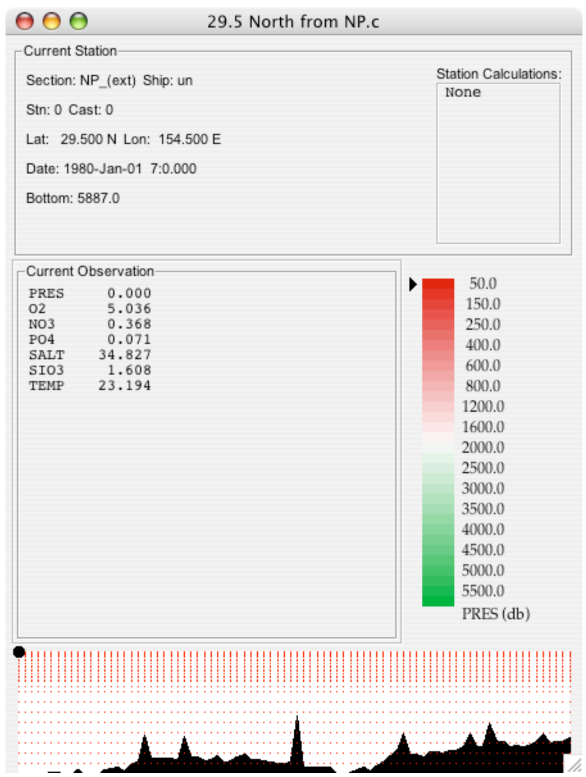
9) Change the selection layer using the contextual popup menu to the gridded data layer (NP) and select a section of interest:



10) Open the selected gridded data into JOA by selecting the **Open selected files** command in the **File** menu. You will be prompted for optional pre-sorting of the data:

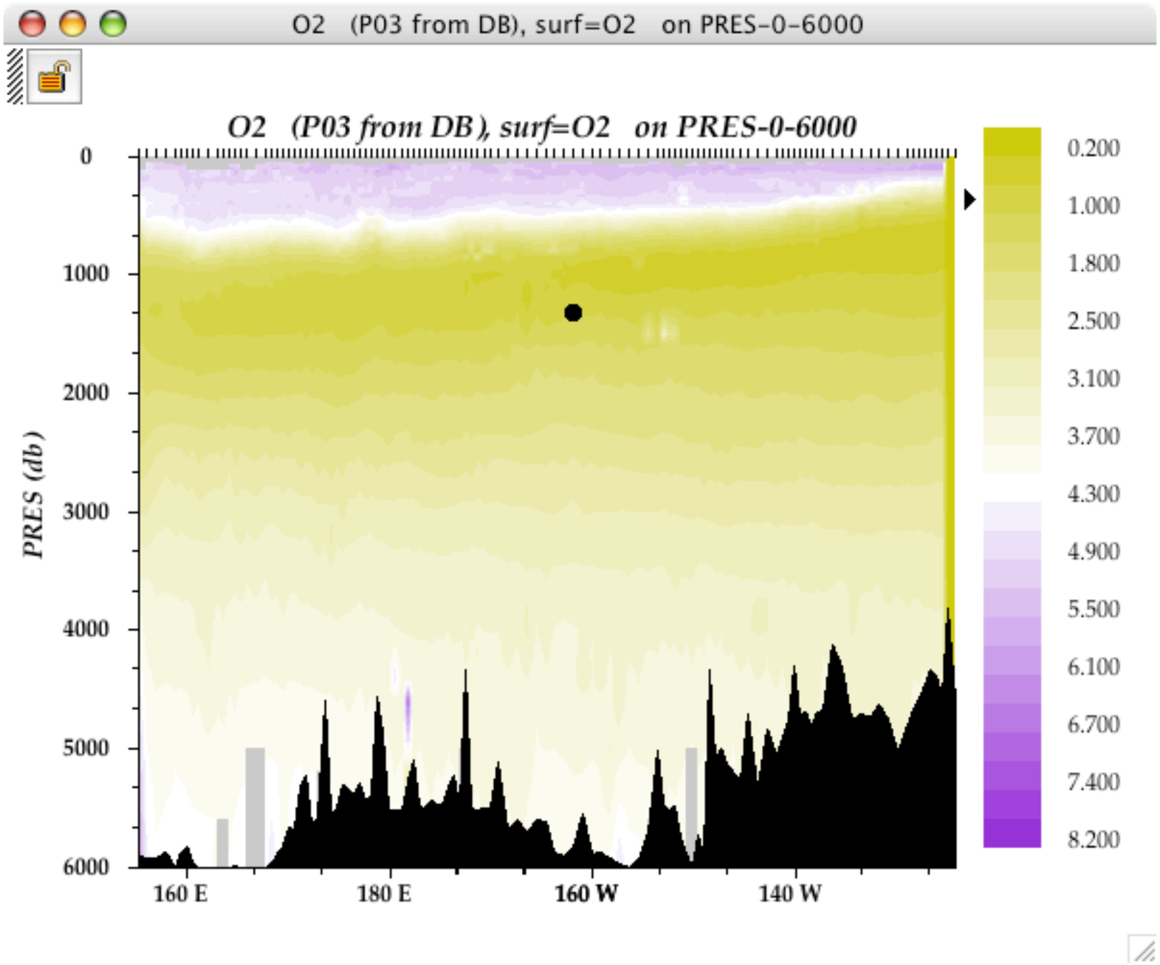


The selected gridded data opened in JOA:

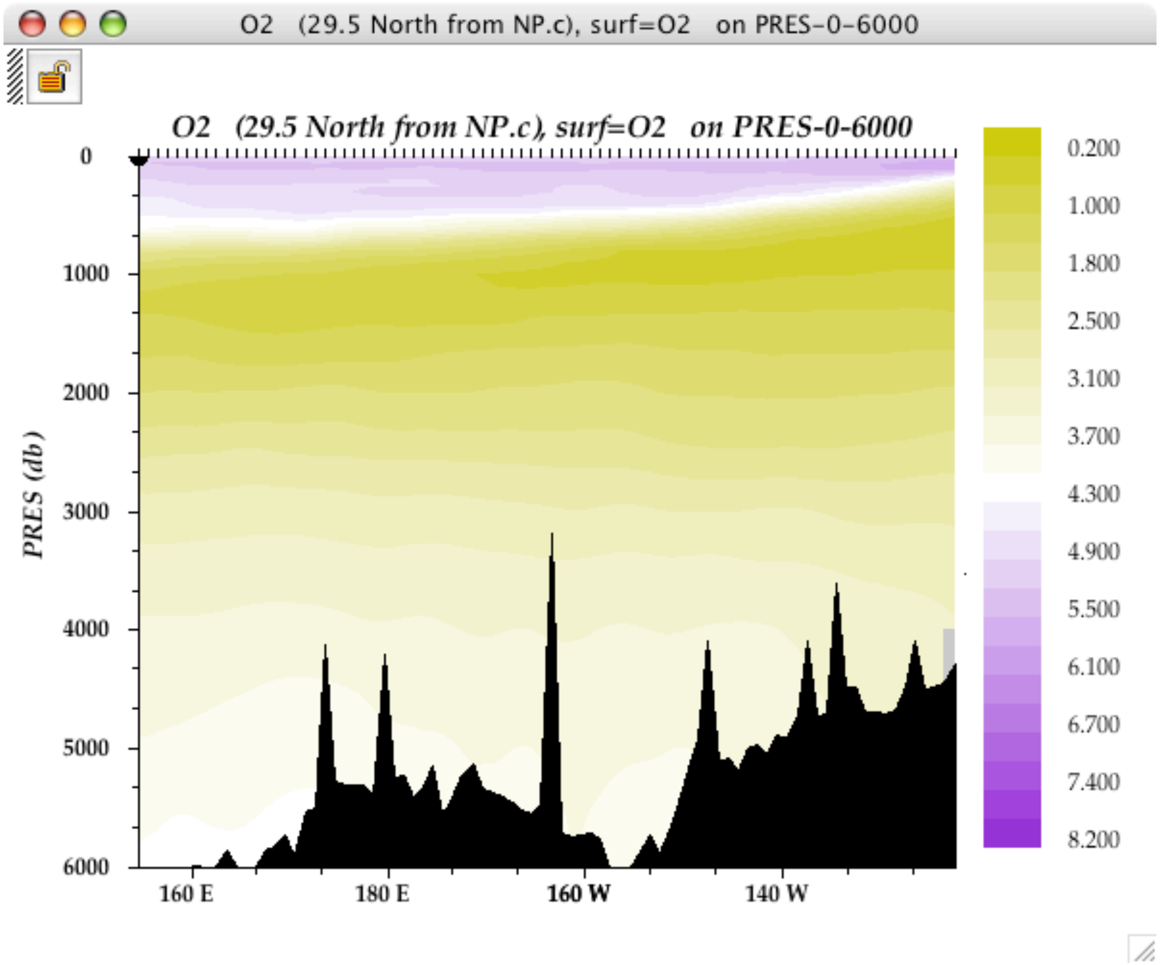


Note: an approximate ocean bottom has been added using the JOA Section Manager.

11) Here is a contour plot of dissolved oxygen from section extracted from the WOCE data:

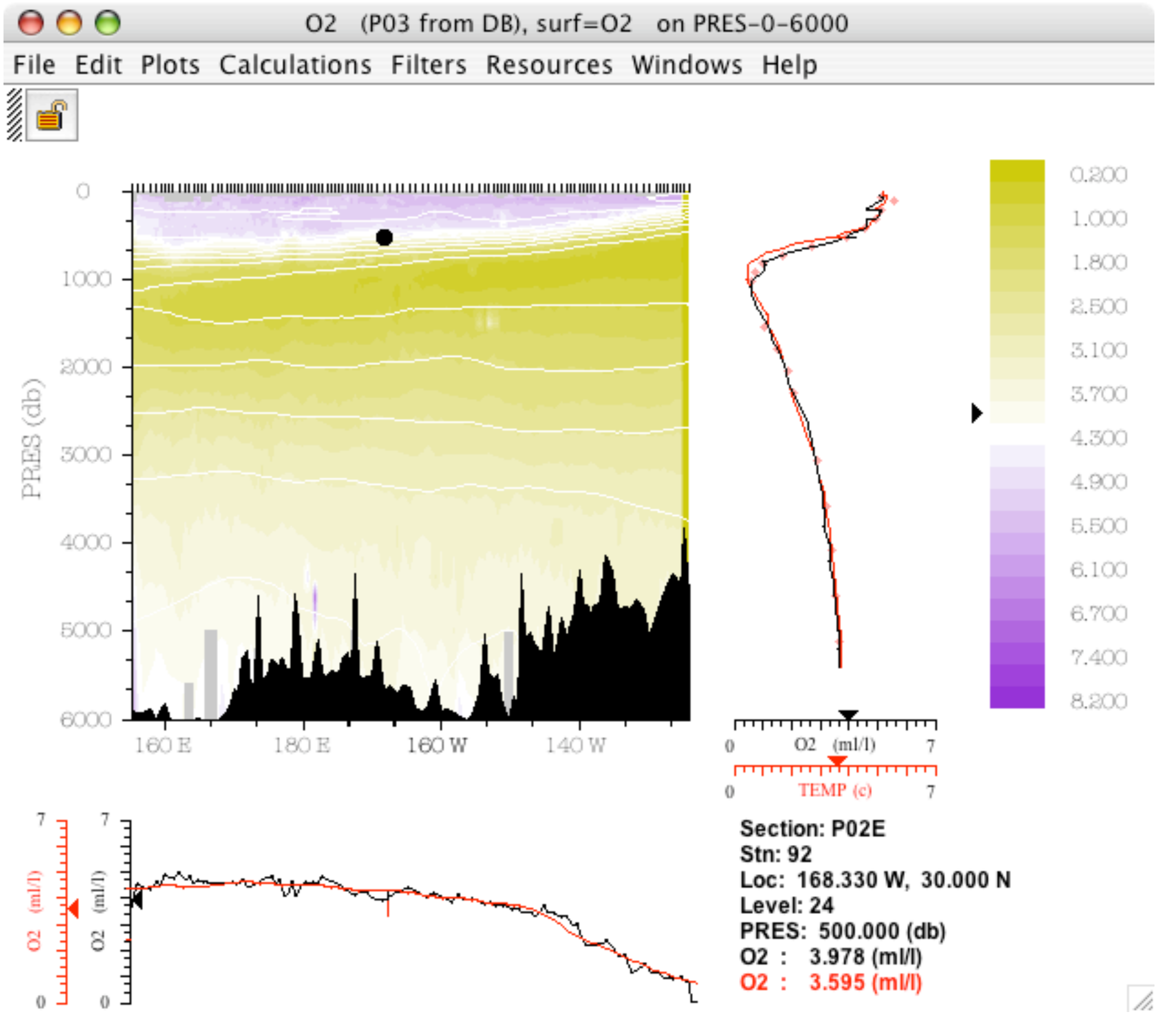


12) And here is a contour plot of dissolved oxygen from the section derived from the gridded data:



13) JOA 5.0 allows for more sophisticated comparisons of sections than just viewing sections side by side. For example, JOA 5.0 allows any parameter to be contoured on top of another parameter, In addition, sections can be differenced and the residuals contoured or overlaid onto another contour plot.

Here is a contour plot of the gridded dissolved oxygen overplotted onto the observed dissolved oxygen:



The gridded oxygen is shown as white contours drawn on top of the solid filled contouring of the base layer from the observed data. Cross sections have been displayed to better compare the two sections. The gridded data are shown as red lines in the cross-section panels and the observed data as a black line. In this case there is a very close correspondence between the observed and gridded data.

APPENDIX B: INSTALLING MYSQL FOR USE WITH NQUERY

Where to get the Installer: As of 11/05, the latest version of MySQL is 5.0. Currently NQuery has only been tested with MYSQL 4.1.5. A good starting point to find the 4.1.x installer is http://dev.mysql.com/downloads/ Follow the link for MySQL 4.1 (currently http://dev.mysql.com/downloads/mysql/4.1.html).

For all platforms, there is both a standard and max version available. NQuery was based on the standard version of the software. The max version has additional capabilities not required by NQuery.

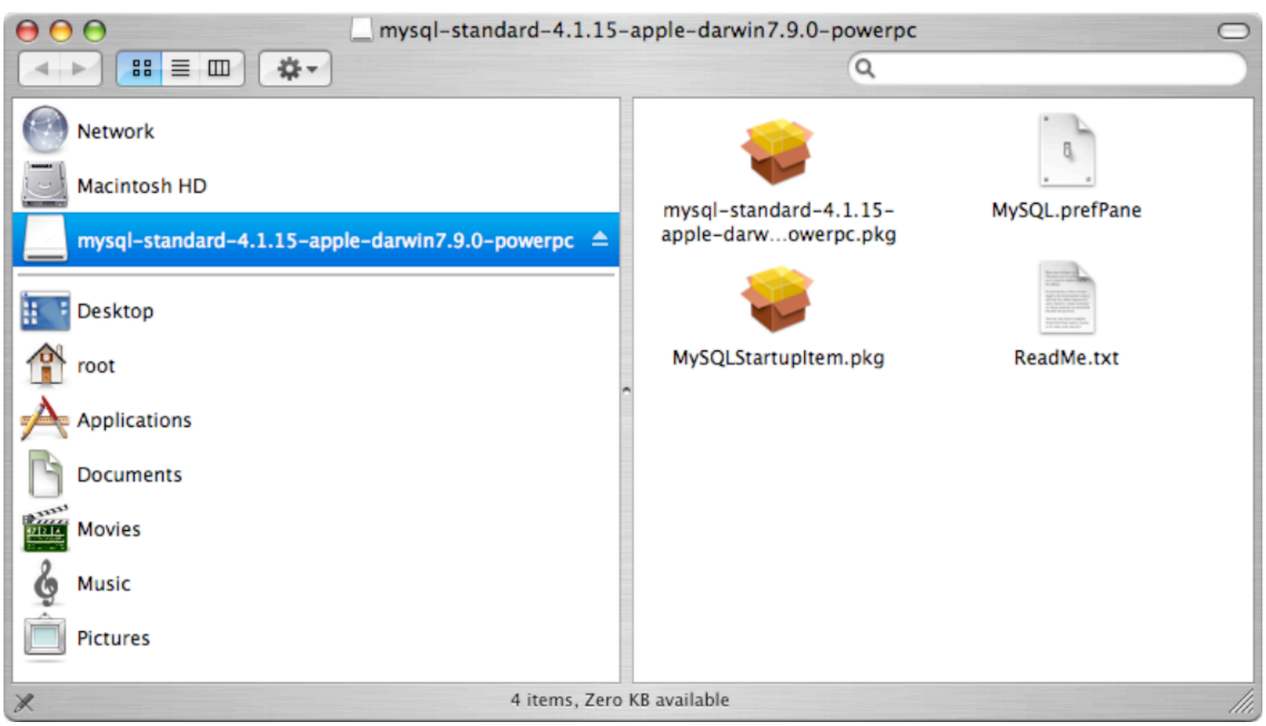
Installation on Mac OS X

0) This section describes installing MySQL through a graphical tool. If you would prefer to install MySQL using the command line interface, see the documentation at:

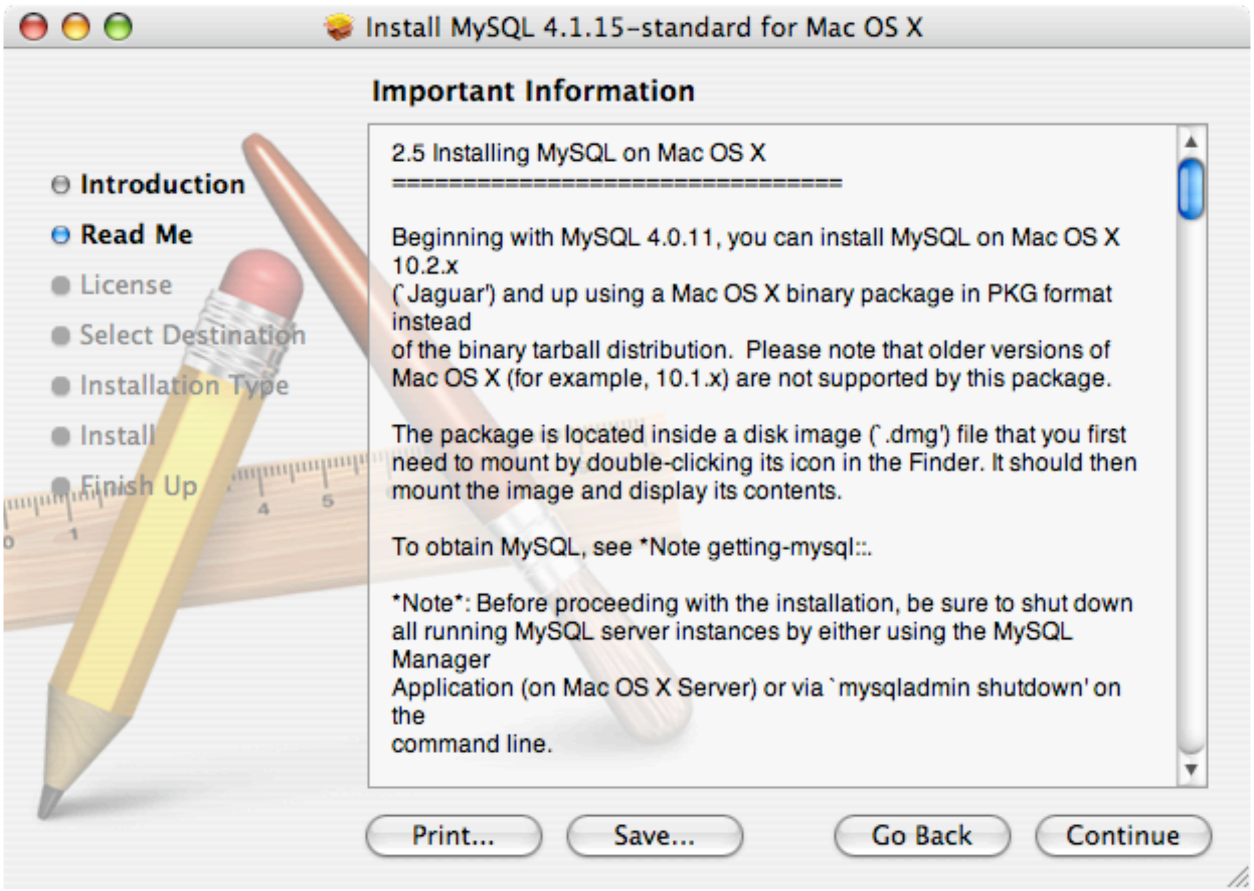
http://dev.mysql.com/doc/refman/4.1/en/mac-os-x-installation.html

1) Download the MySQL installer package for Mac OS X. The installer will be packaged as a “disk image” file with a .dmg file extension. The disk image should automatically open on the desktop with a white external disk icon. If not, double click the .dmg file to mount the disk image.

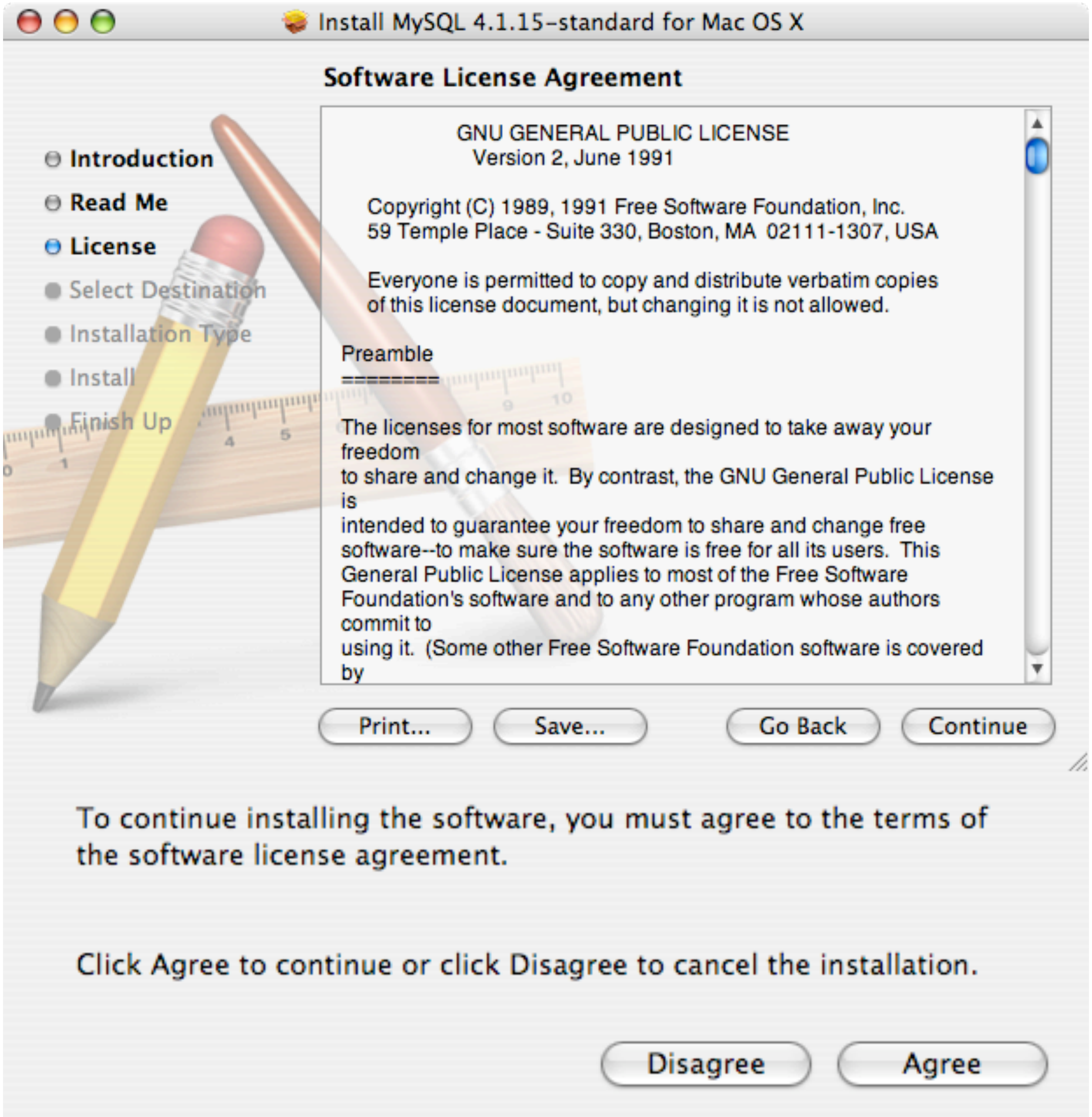
2) Open the disk image to reveal its contents. You can open the disk image into a new window by double-clicking the disk image icon or click on the disk image in the sidebar:



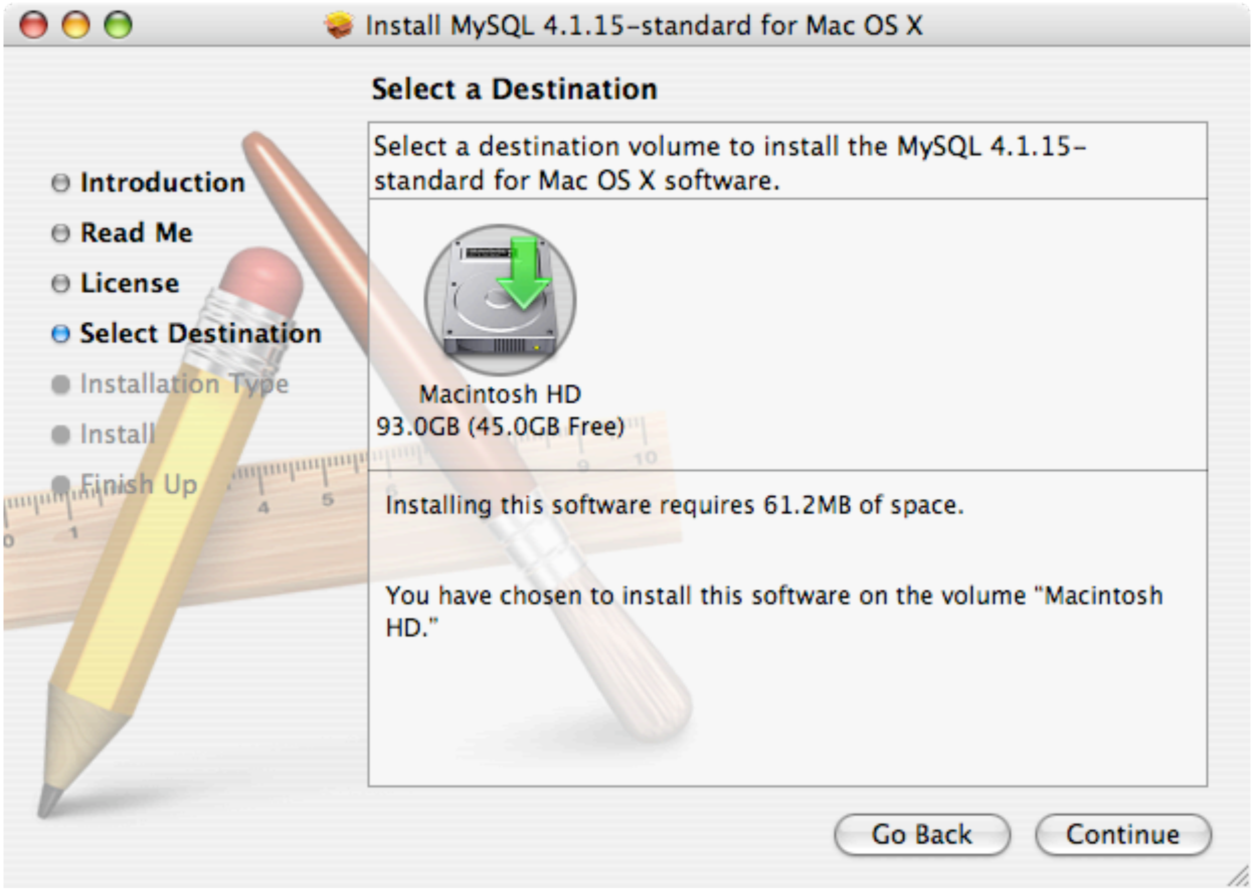
3) Double click on the MySQL installer package (a file with a .pkg extension). In this example, the installer package is called “mysql-standard-4.1.5-apple-darwin-powerpc.pkg.” This will launch the MySQL installation assistant (or wizard):



4) Click the “Continue” button to proceed through the installation process. Click the “Agree” button to accept the GPL agreement:

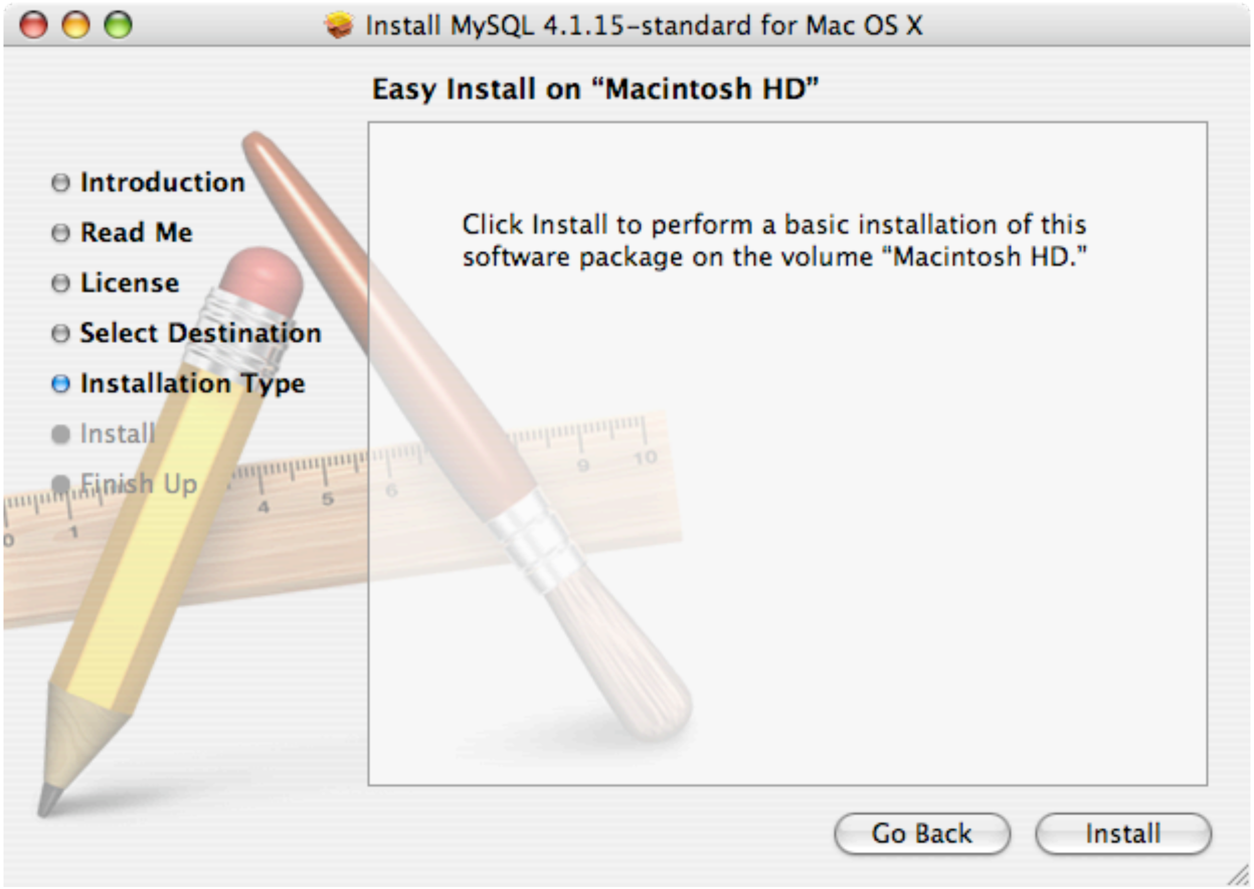


5) Select the destination volume to install MySQL:

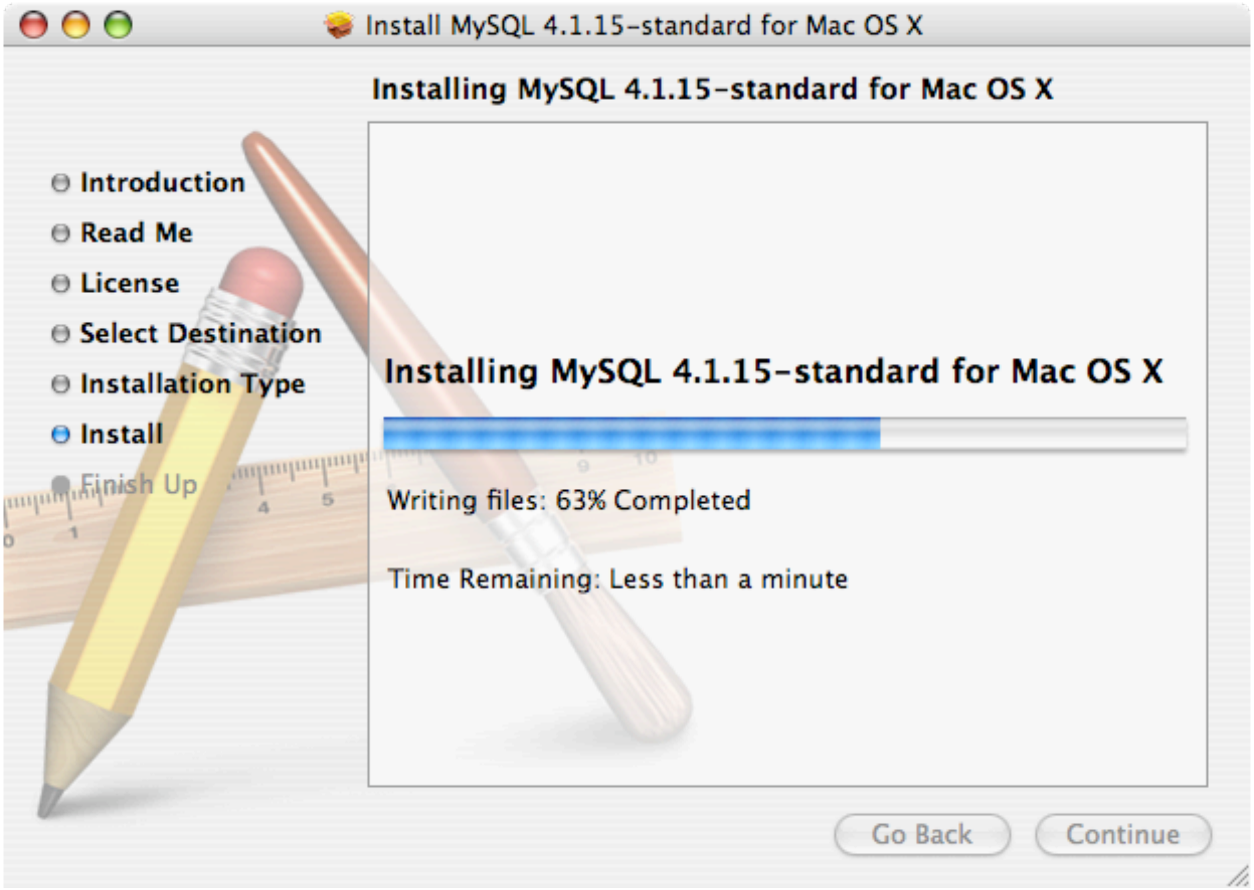


6) Enter an administrator's name and password:

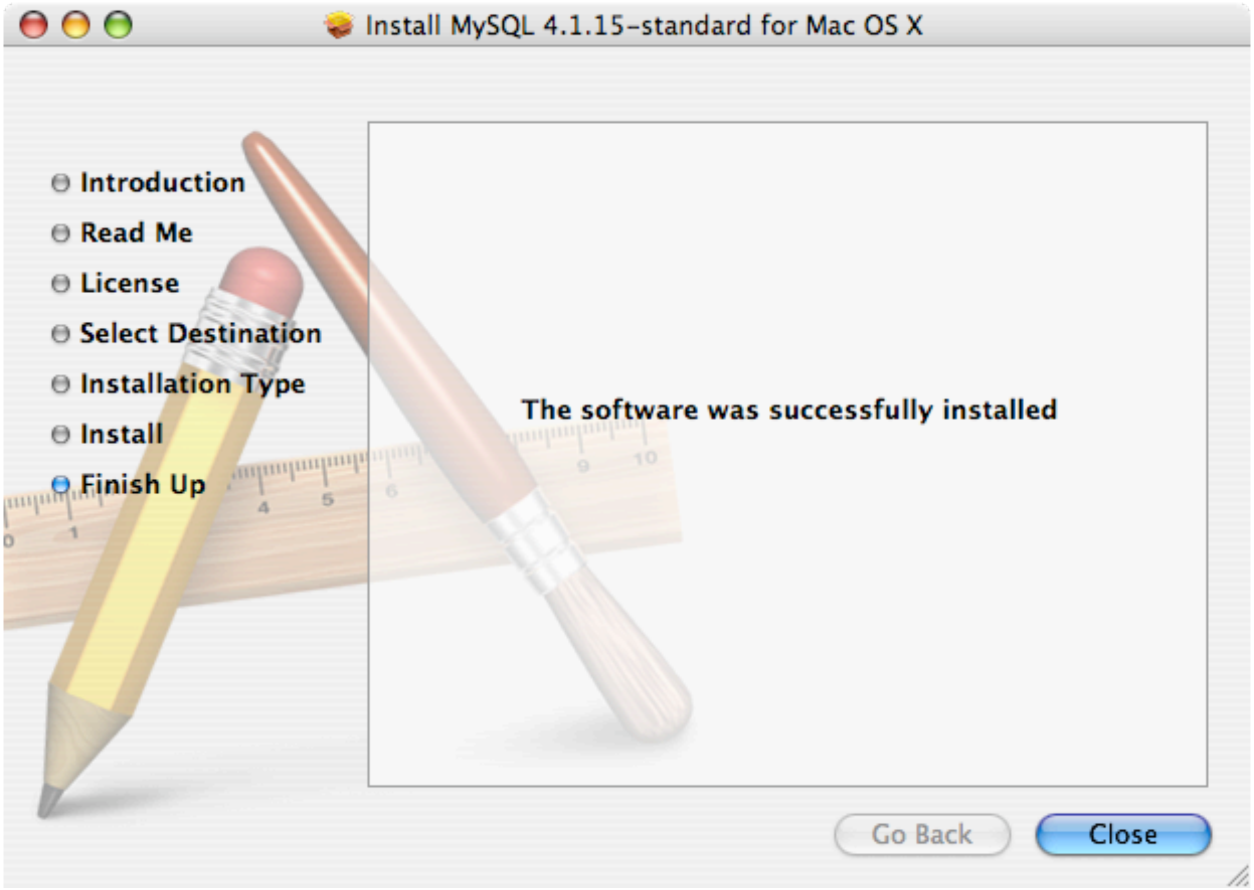
7) There is only one installation type for Mac OS X. Click the “Install” button to start installing MySQL on your Mac:



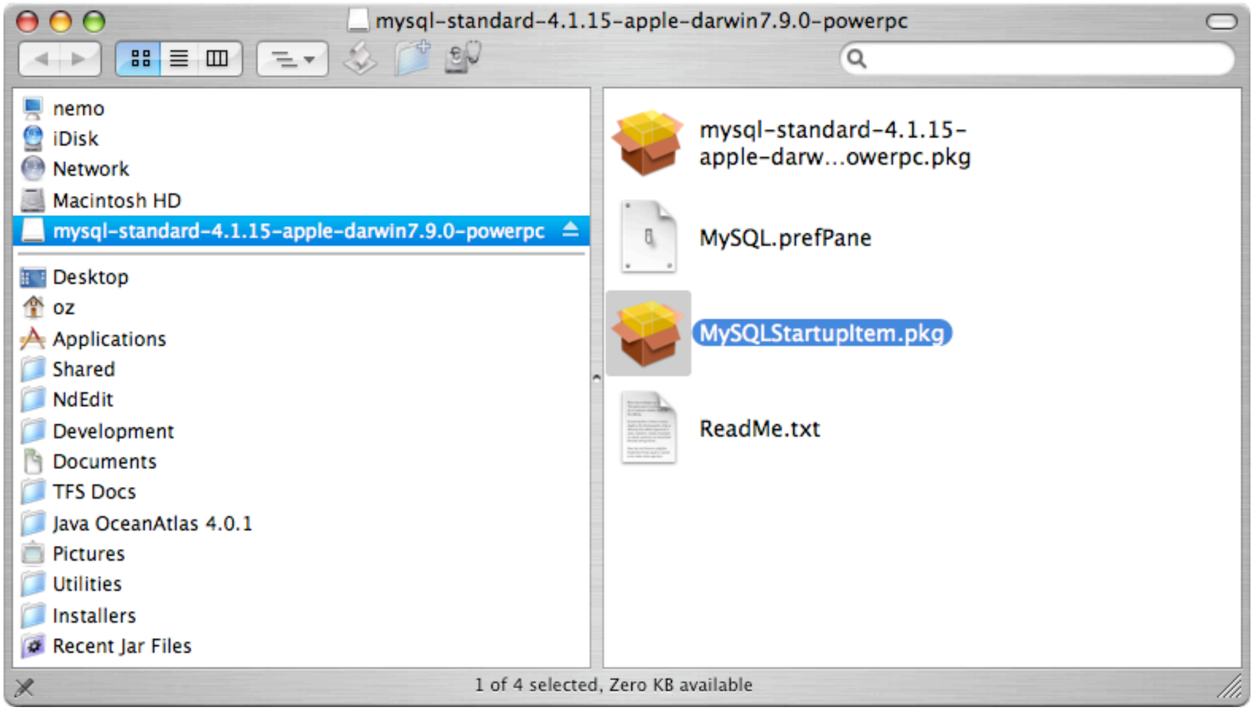
8) You will be informed of the installation progress:



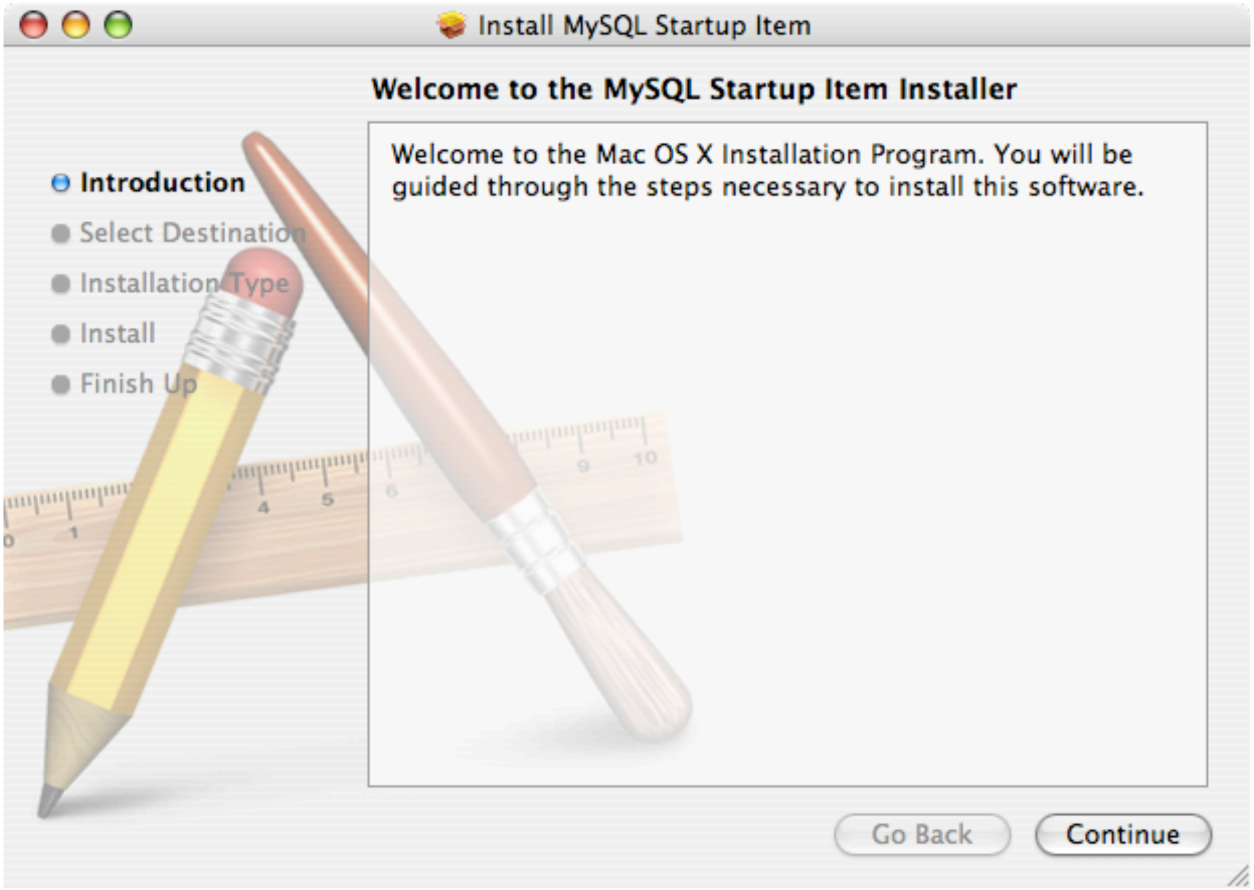
9) You will be notified when the installation is complete. Click the “Close” button to quit the installer application:



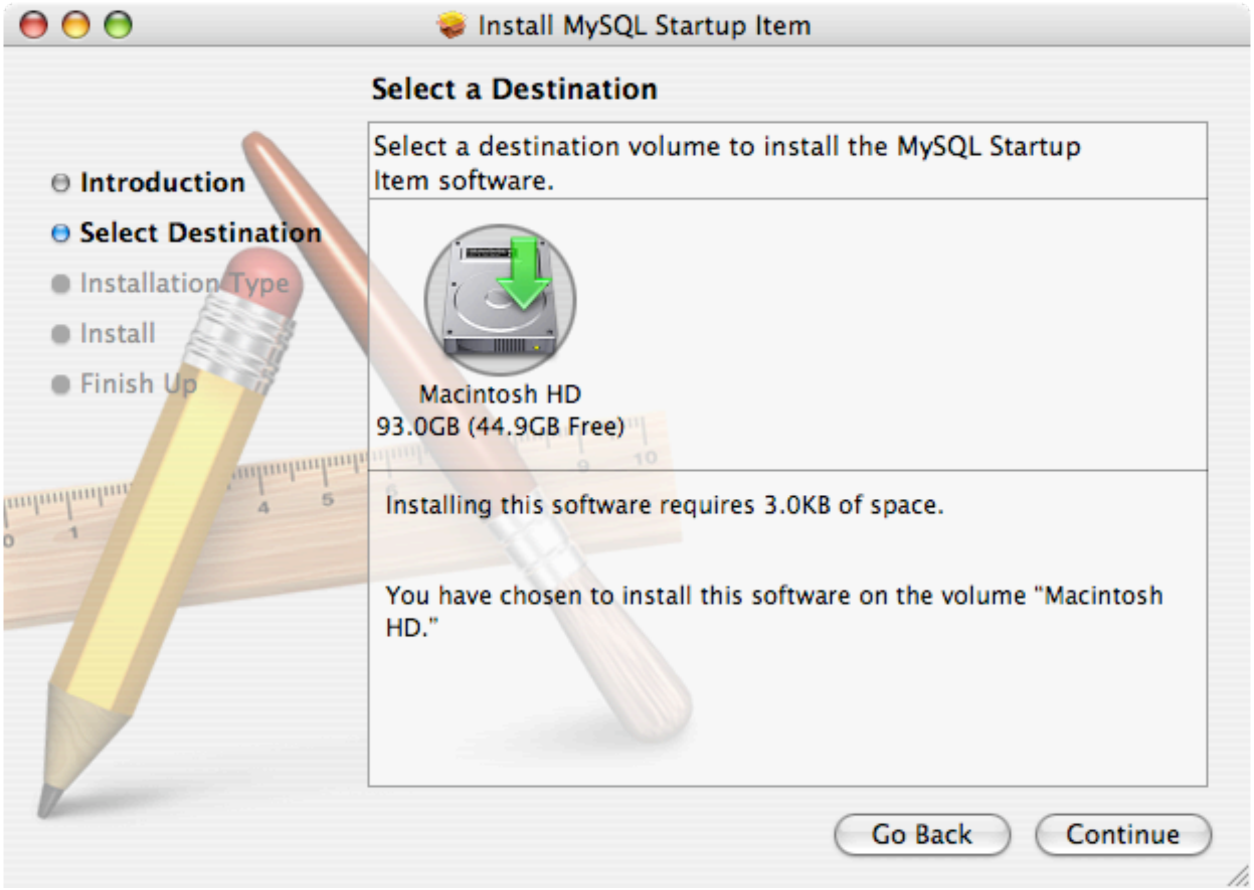
10) At this point MySQL is installed on your Mac but not actually running. MySQL can be started manually through the command line interface but installing the MySQL startup item is recommended for most users. In the MySQL installer disk image, double click the MySQLStartupItem.pkg:



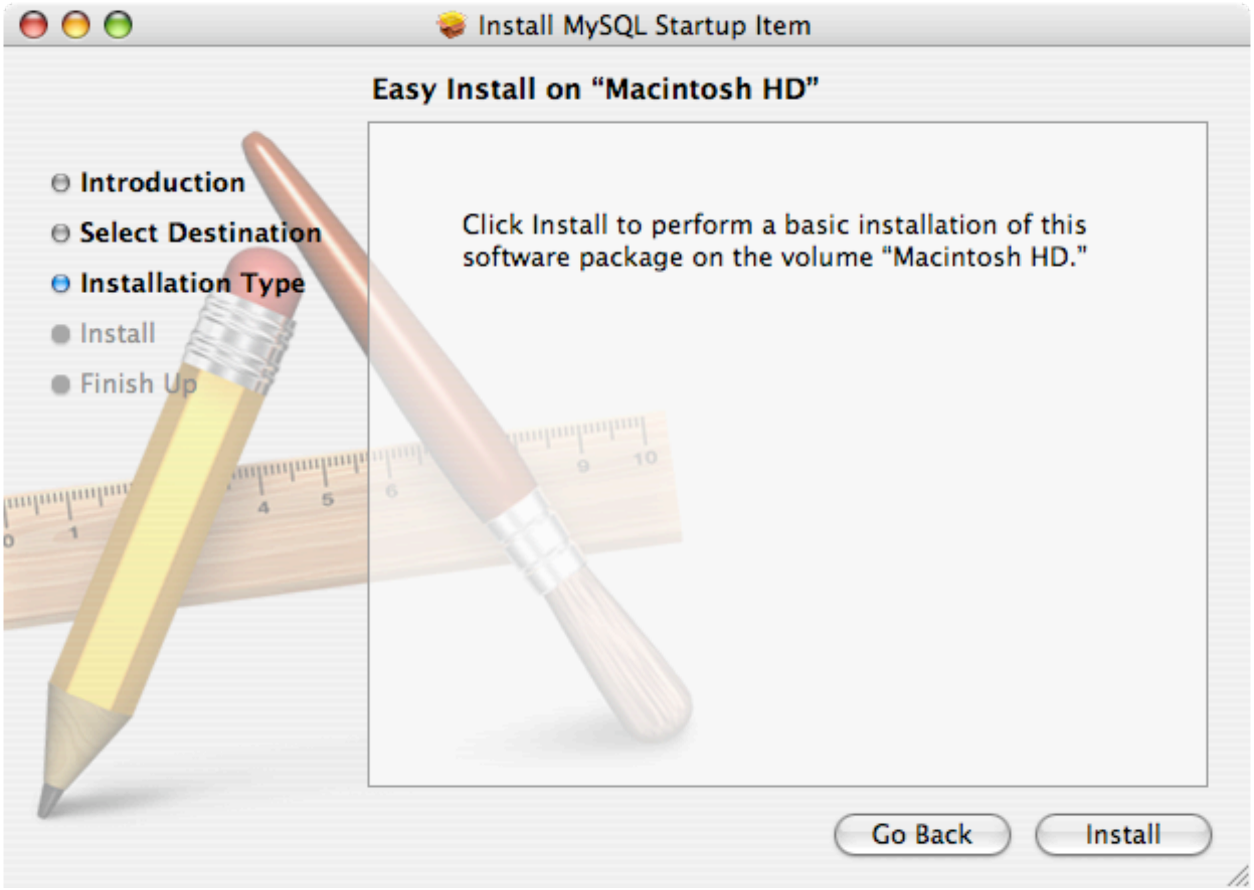
11) This installer will install a MySQL startup control panel in the System Preferences application. This makes it easy to start/stop the database server as well as view its current status. Follow the installation wizard::



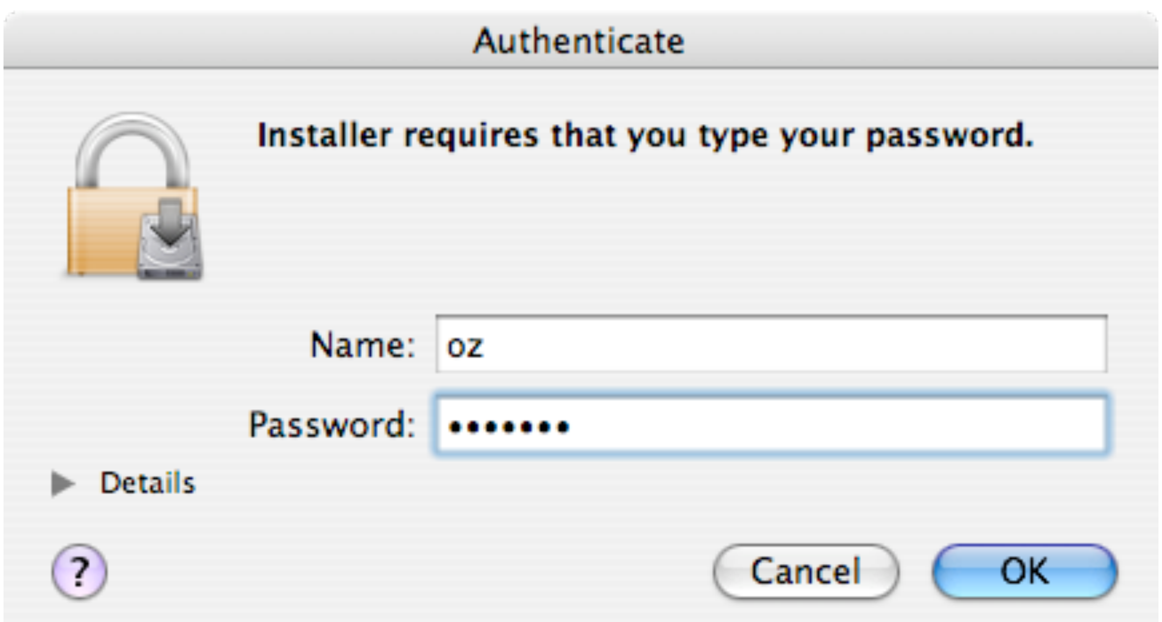
12) Select the same destination volume you installed MySQL on:



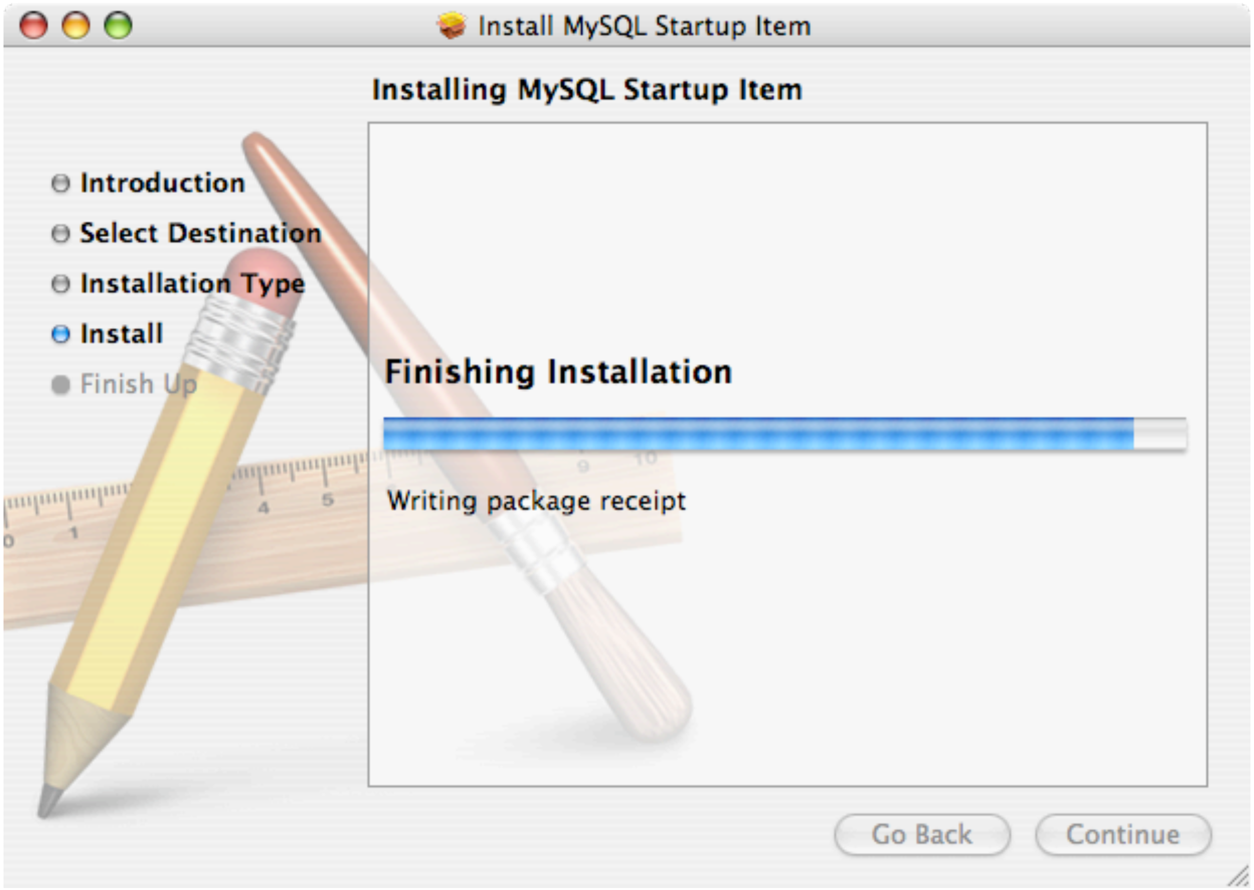
13) Click the “Continue” button:



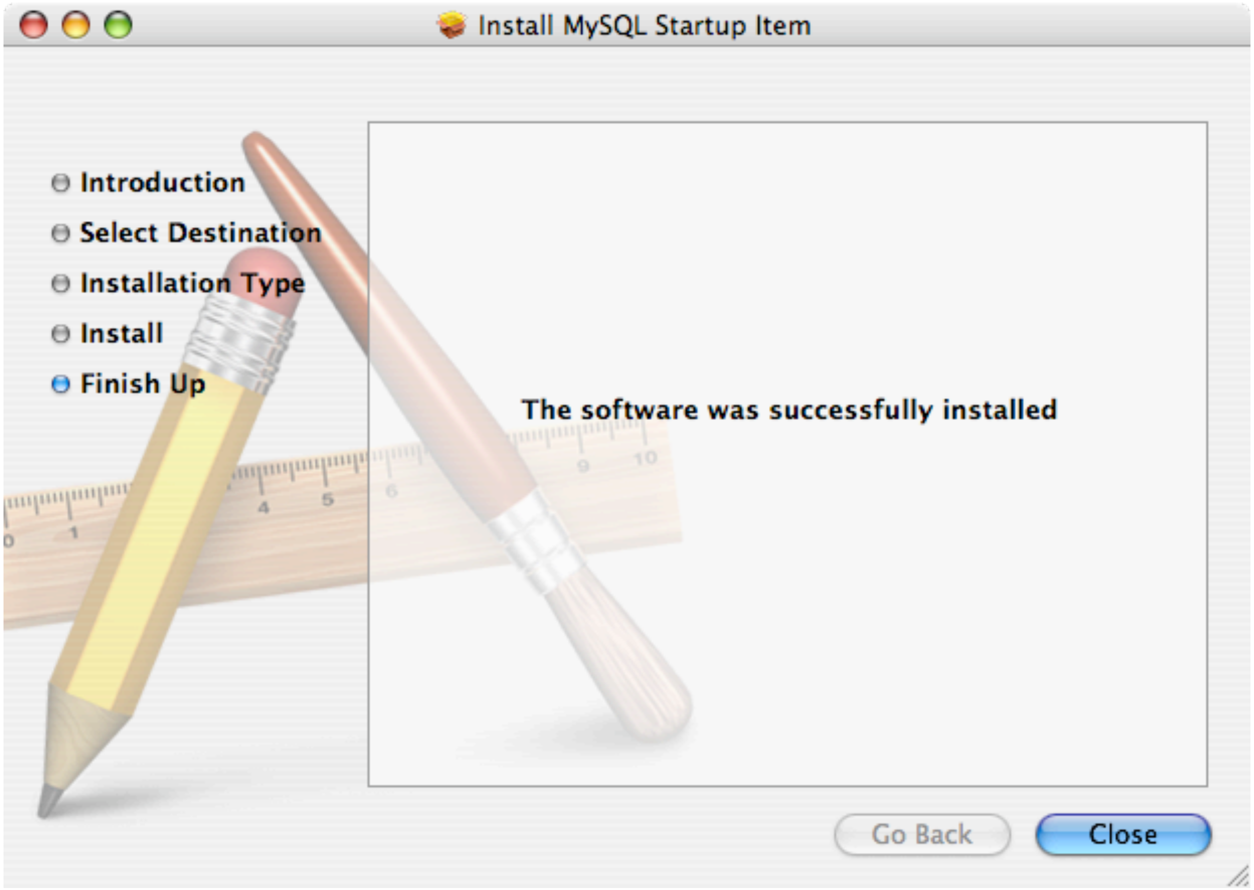
14) Provide an administrator's user name and password:



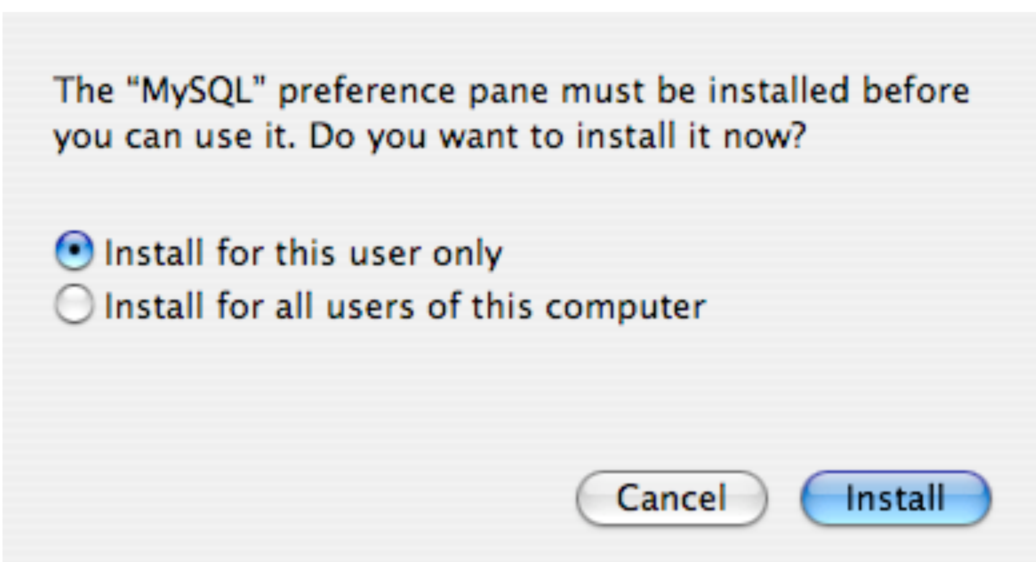
15) You will be informed of the installation progress:



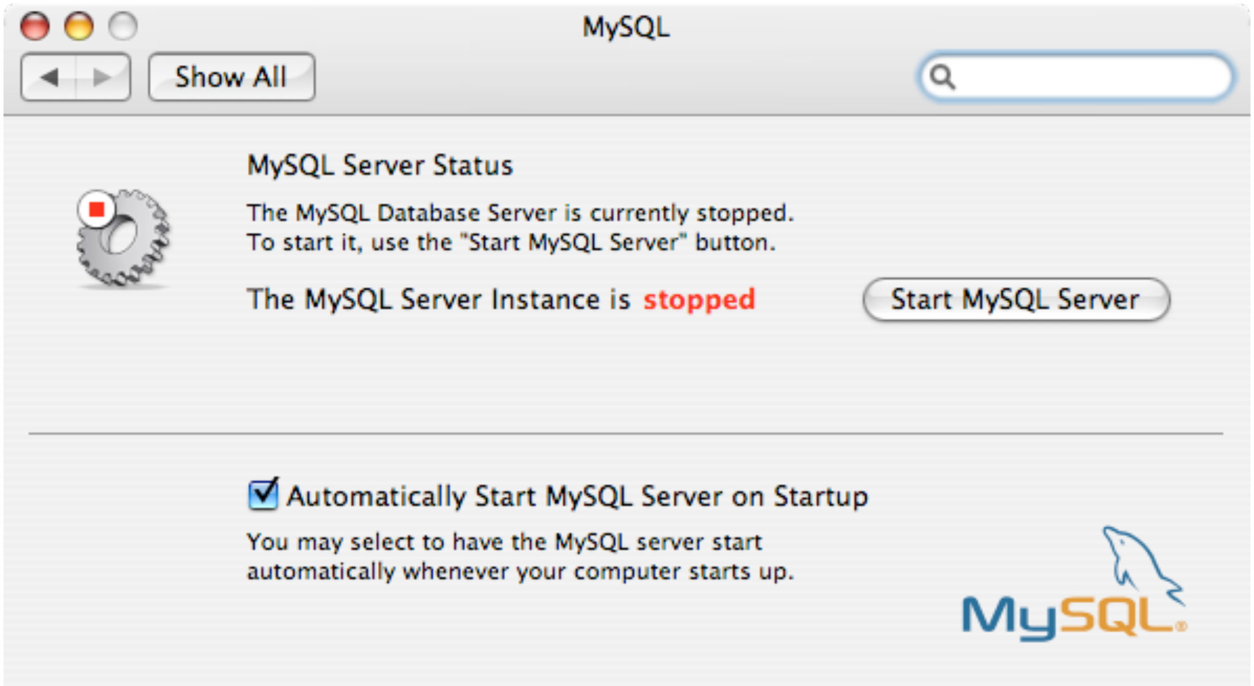
16) Click the “Close” button to quit the installer:



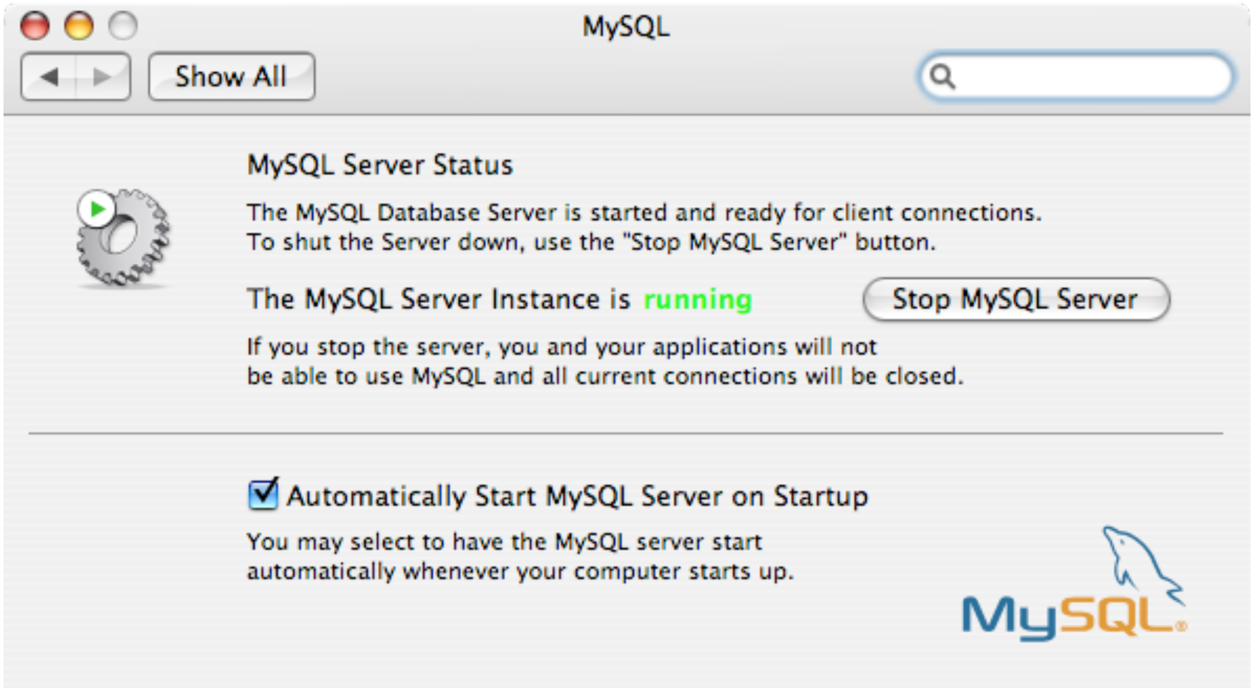
17) At this point, the startup item is installed on the disk but not activated as a System Preference. Double click the file: MySQL.prefPane found on the MySQL disk image.. You may chose to restrict this item to the current user or make available to all users of your Mac:



18) Open System Preferences from the Apple menu or the dock. Click on the MySQL item in the “Other” panel of the System Preferences window. Initailly, the MySQL server is “stopped:”



19) Click the “Start MySQL Server” button. Status should changed to “running.” It is recommended that you check “Automatically Start MySQL Server on Startup:”



Installation on Windows

Complete instructions can be found at:

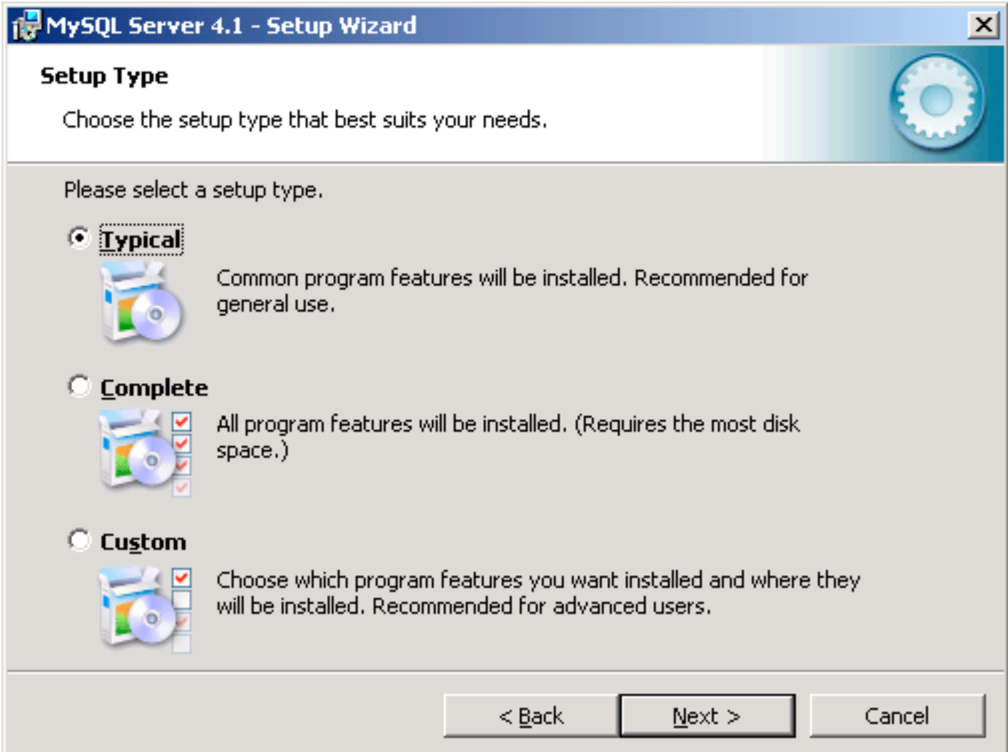
http://dev.mysql.com/doc/refman/4.1/en/windows-installation.html

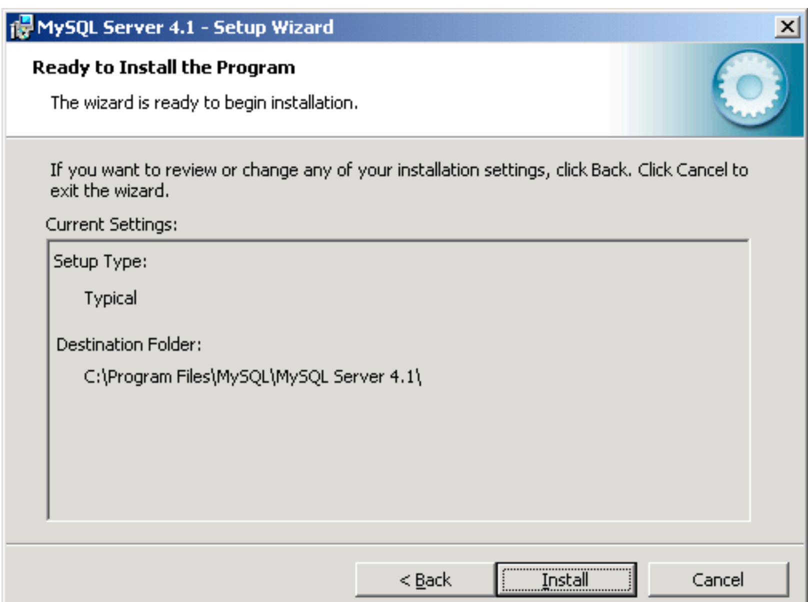
Download installer package. For Windows, the installer comes in a zip file. Open the zip file and extract the actual installer to your disk.

Launch the installer wizard and follow the instructions. Use the following screen images to set MySQL to a recommended configuration:

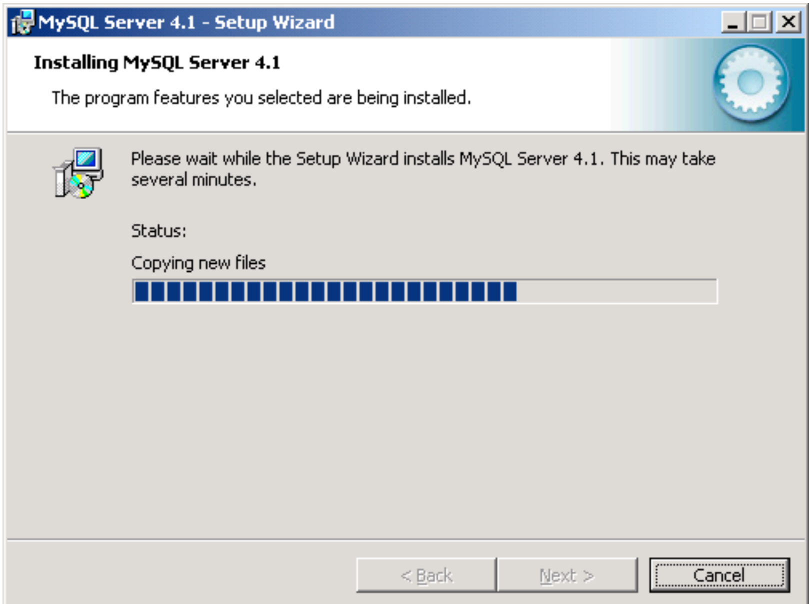
Click Next to continue.

The Typical setup type installs a useable MySQL system for JOA. Other setup types are beyond the scope of this document.

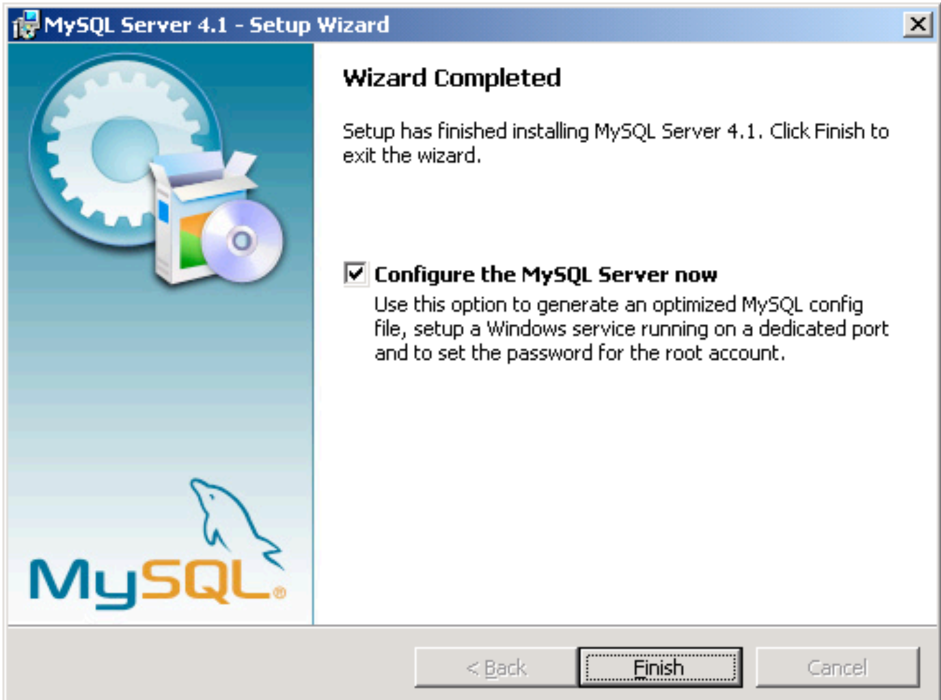




The wizard is now ready to install MySQL. Click the Install button to begin installation. You will see this progress screen as the installation proceeds:

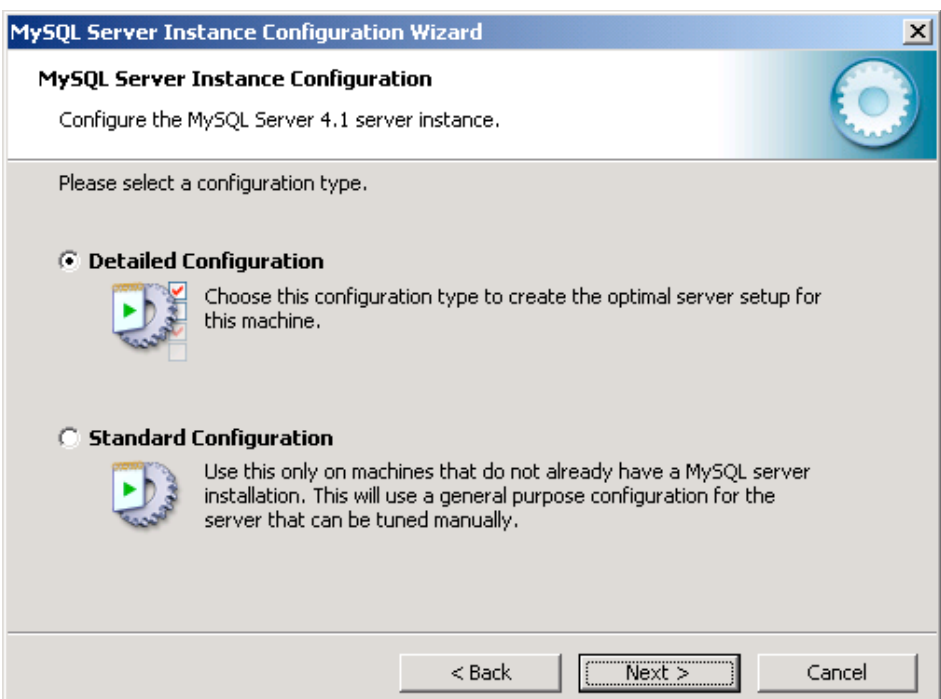


You can optionally set up an account at MySQL.com:

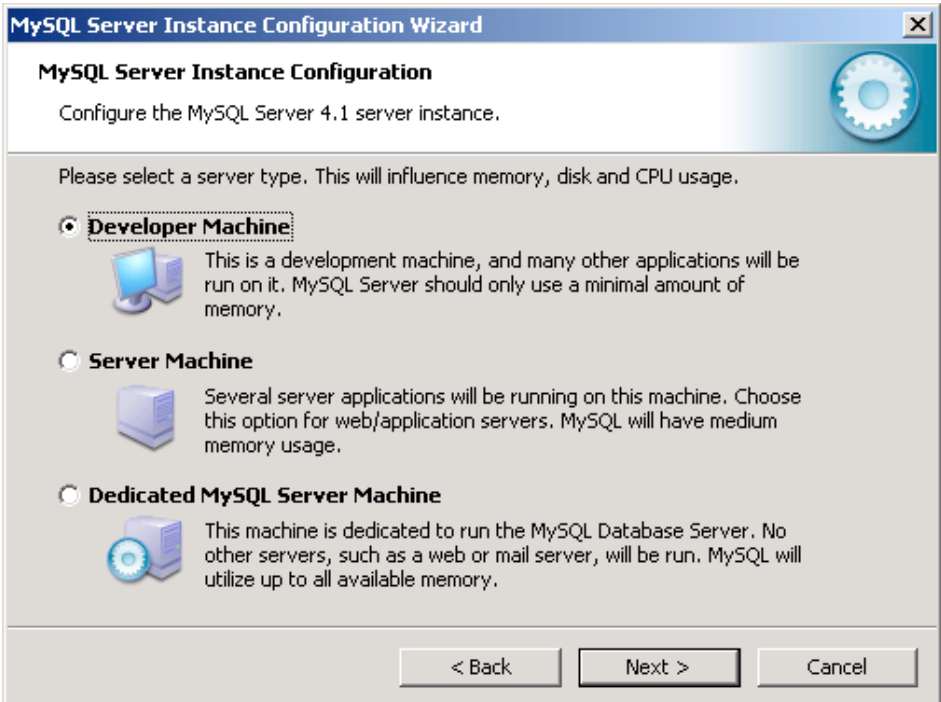


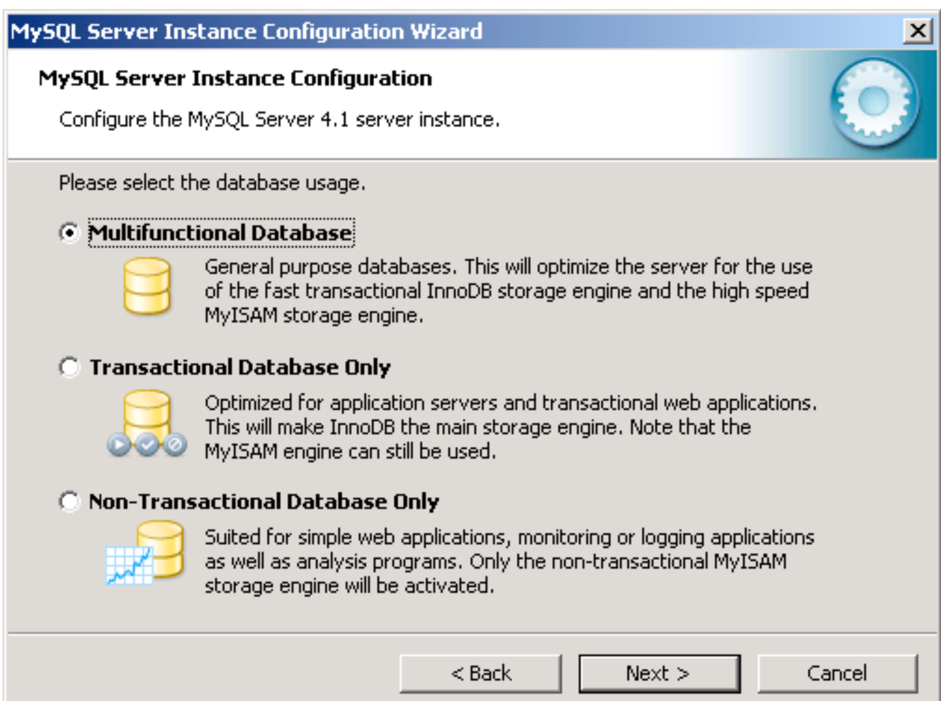
Configuring the MySQL Server on Windows

After installation has completed, you will be prompted to run the MySQL Instant Configuration Wizard. Unless you are comfortable with the command line interface of your operating system (see the Mac OS X installation instructions), we recommend using this wizard in the “Detailed Configuration” mode for initial configuration.

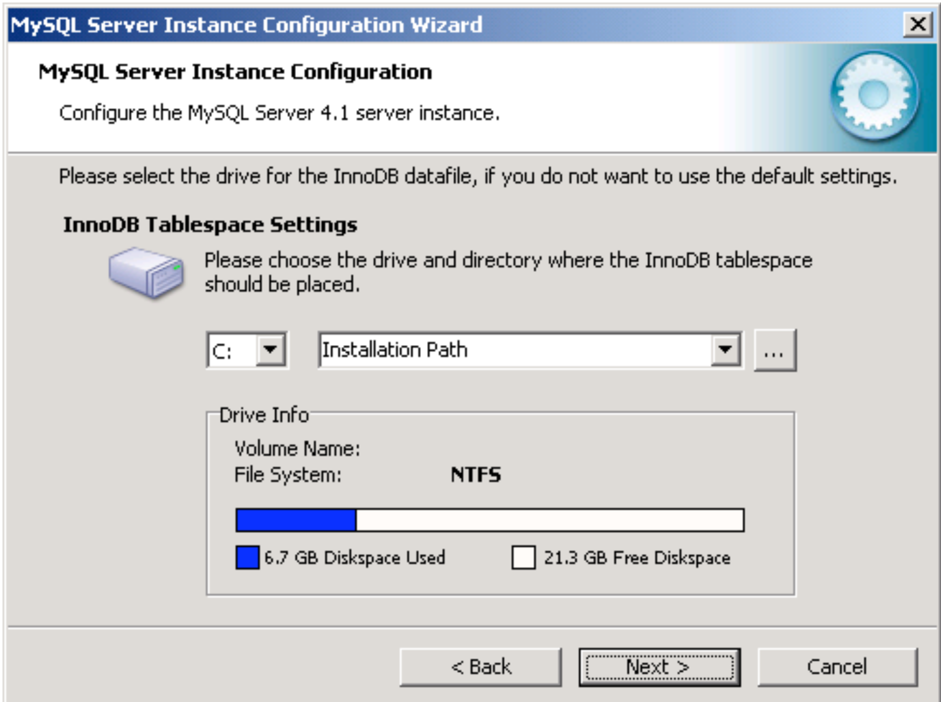


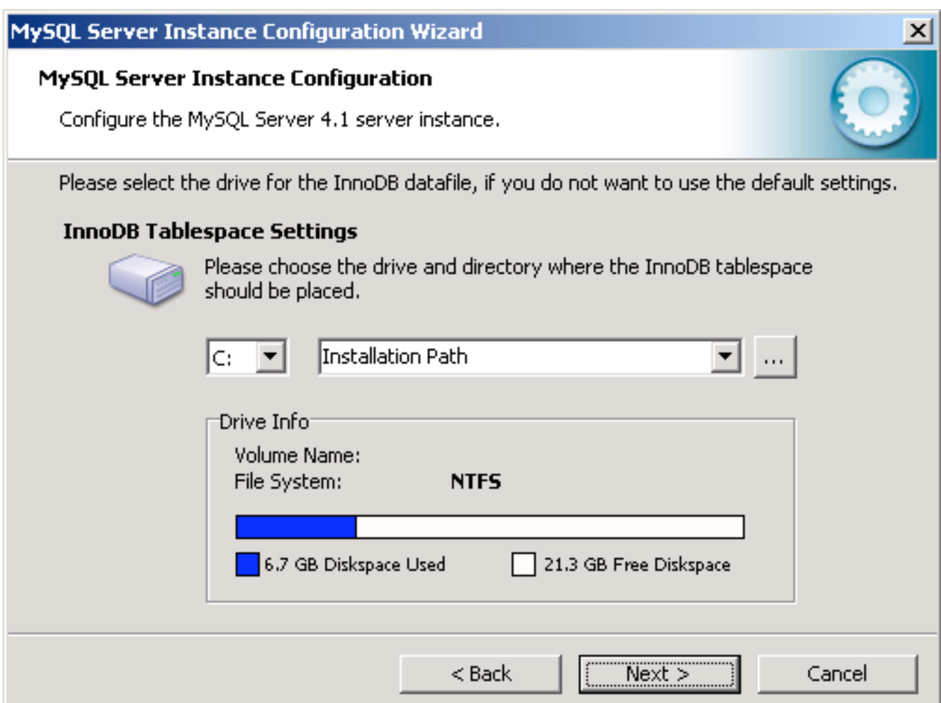
The developer machine option is best for a general purpose desktop computer:



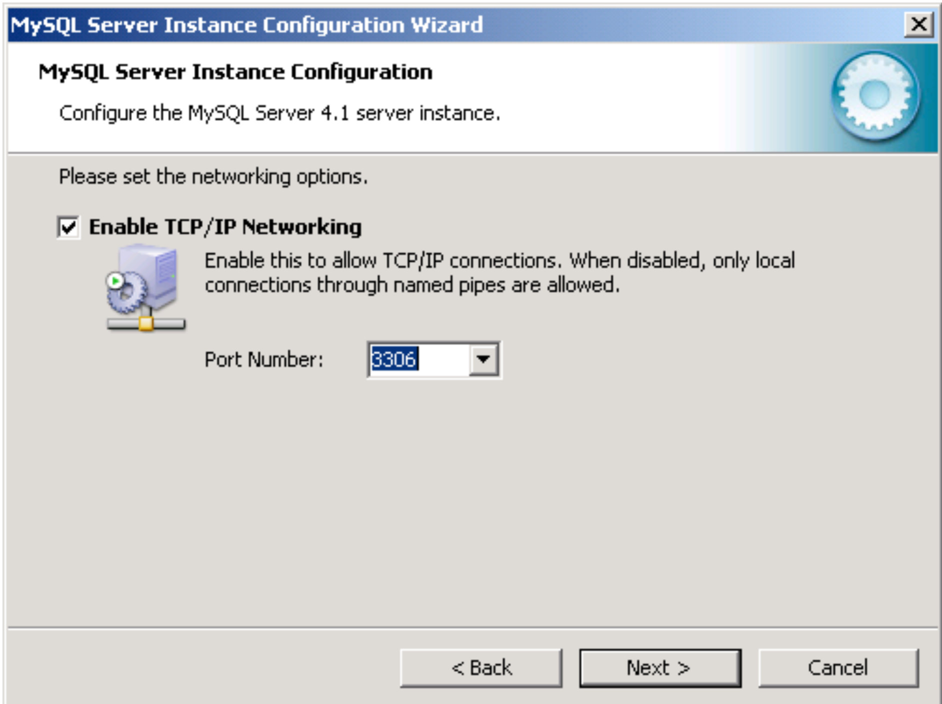


If disk space is tight on your PC, you can choose to store the database files on another disk or partition. Here we accept the default of using the main volume of the PC:

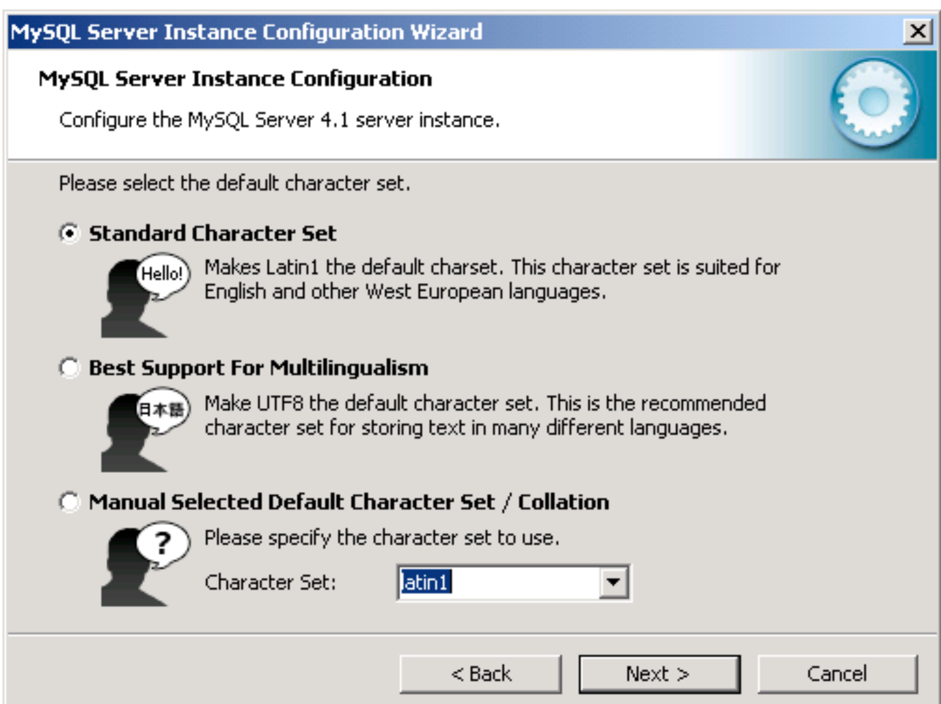




Unless you know, how to use named pipes, enable TCP/IP networking on the default port number:



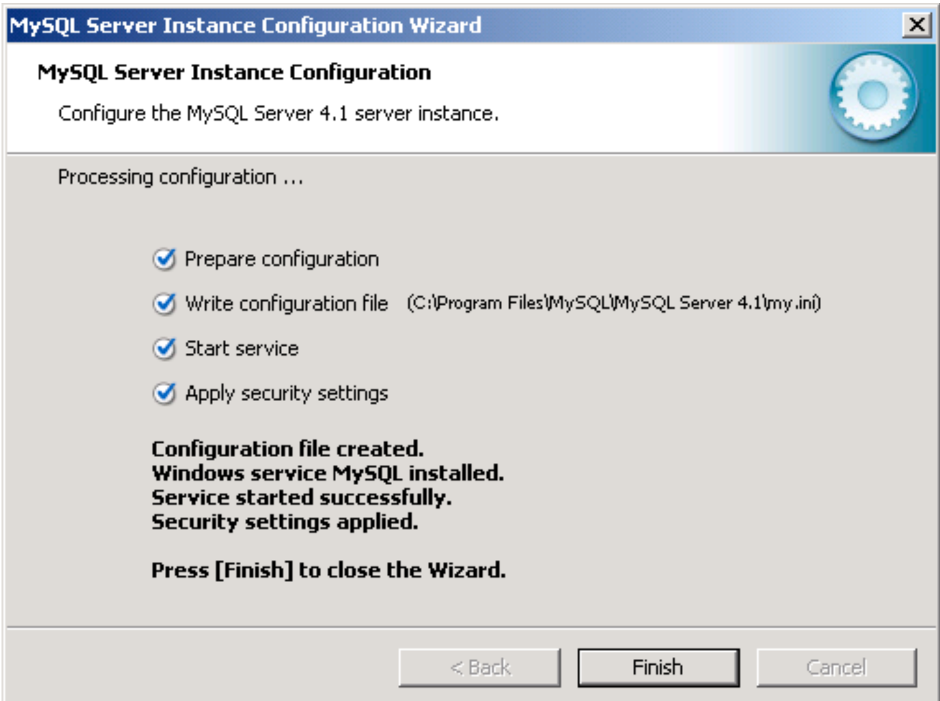
You may wish to choose an alternative character set:



We highly recommend installing MySQL startup as a Windows Service. Otherwise, you will be responsible for starting MySQL from the command line:

MySQL comes with one defined user with the name “root”. You may choose to use this user or you may setup other user names. If you do not have a separate MySQL management application, then new users will have to be added via the command line. We recommend that you assign a password to the root account whether you define other users or not:

If you see the screen below, then MySQL was successfully configured and you can now created databases in NQuery.



Installation on Linux UNIX

There are many different flavors of Linux and this document can not describe them all. Complete instructions can be found at:

http://dev.mysql.com/doc/refman/4.1/en/linux-rpm.html

MySQL Initial Setup

Adding Users to MySQL Database Server

After MySQL is first installed it can be accessed by anyone on your system. The MySQL installation process typically adds two root accounts as well as two anonymous user accounts. The root and user accounts have full control over the server so it's important that the first thing you do is secure the root accounts by adding passwords. Note: the Windows installer, allows you to secure MySQL in the installation wizard. This section is devoted to Mac OS X and UNIX users.

Securing MySQL and Adding Users on Mac OS X

User’s can be added to the MySQL server either through the command line interface or through third-party, free or commercial graphical programs. One recommended commercial tool is MySQL4X Manager described at:

http://www.macosguru.com/macosguru3/products/mysql4xmanagergeneric.html

This manual will describe how to do this from the command line—it’s beyond the scope of this manual to describe all the available graphical tools.

To see this process in excruciating detail, go to:

http://dev.mysql.com/doc/refman/4.1/en/default-privileges.html

First it’s useful to know a little about where MySQL is installed on your Mac. MySQL is installed into /usr/local/mysql. However, mysql is actually a link to a different directory so the real installation directory will look more like: /usr/local/mysql-standard-4.1.15-apple-darwin7.9.0-powerpc. This directory contains the following items:

```
drwxr-xr-x   19 root   wheel    646 Oct 10 03:59 .
drwxr-xr-x   12 root   wheel    408 Jan 18 04:00 ..
-rw-r--r--    1 root   wheel  19071 Oct  9 15:46 COPYING
-rw-r--r--    1 root   wheel   5712 Oct 10 03:54 EXCEPTIONS-CLIENT
-rw-r--r--    1 root   wheel   8307 Oct 10 03:54 INSTALL-BINARY
-rw-r--r--    1 root   wheel   1379 Oct  9 15:46 README
drwxr-xr-x   54 root   wheel   1836 Oct 10 03:59 bin
-rwxr-xr-x    1 root   wheel    801 Oct 10 03:58 configure
drwxr-x---   23 mysql  wheel    782 Jan 21 17:03 data
drwxr-xr-x    4 root   wheel    136 Oct 10 03:58 docs
drwxr-xr-x   61 root   wheel   2074 Oct 10 03:59 include
drwxr-xr-x   10 root   wheel    340 Oct 10 03:59 lib
drwxr-xr-x    3 root   wheel    102 Oct 10 03:58 man
drwxr-xr-x   12 root   wheel    408 Oct 10 03:59 mysql-test
drwxr-xr-x    3 root   wheel    102 Oct 10 03:59 scripts
drwxr-xr-x    5 root   wheel    170 Oct 10 03:59 share
drwxr-xr-x   31 root   wheel   1054 Oct 10 03:59 sql-bench
drwxr-xr-x   14 root   wheel    476 Oct 10 03:59 support-files
drwxr-xr-x   21 root   wheel    714 Oct 10 03:59 tests
```

The most useful directory her is the bin directory. It has these important files:

mysql: the command line interface to the actual database server
mysqladmin: a
mysqld:
mysqlshow

You will need to be an administrator to run the commands below. You will note that the commands are prefaced with “sudo”. This stands for “super user do”. You will be prompted for an administrative password to complete the command.

1) Verify that the MySQL server is running:

```
>sudo /usr/local/mysql/bin/mysqlshow
Password:

+-----------+
| Databases |
+-----------+
| mysql     |
| test      |
+-----------+
```

or:

```
> sudo ./mysqladmin version
Password:
./mysqladmin  Ver 8.41 Distrib 4.1.15, for apple-darwin7.9.0 on powerpc
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX DataKonsult AB
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license

Server version          4.1.15-standard
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /tmp/mysql.sock
Uptime:                 3 days 22 hours 38 min 17 sec

Threads: 1  Questions: 14647  Slow queries: 0  Opens: 940  Flush tables: 1  Open
tables: 51  Queries per second avg: 0.043
```

Note in the above example we have moved into the mysql directory with the cd command. The “./” in subsequent examples means “in the current directory”.

2) Assign passwords to the root accounts.

```
> sudo ./mysql –u root
> mysql> use mysql;
> mysql> SET PASSWORD FOR ''@'localhost' = PASSWORD('newpwd');
```

3) Assign passwords to the anonymous user accounts. Launch the mysql interpreter with this command:

./mysql -u root –password mysql

Here you are logging in as root, will be prompted for a password, and you'll be using the mysql database. You should see a display like this:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.1.15-standard

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Note: you can actually log into the server with any user name (or none) but you will have access to only a "test" database.

Note: All SQL commands are terminated with the semicolon character ';'. To continue a long command on a new line, simple press return—you'll be prompted for the rest of the command with an arrow "->". When you are finished composing a command type the semicolon character and press return.

4) Issue the GRANT command to create a new user:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'oz'@'localhost'
    -> IDENTIFIED BY 'oz1234' WITH GRANT OPTION;
Query OK, 0 rows affected (0.35 sec)
```

This example illustrates creating a user named 'oz' with all privileges and password = 'oz1234'.

APPENDIX C: ADDITIONAL INFORMATION

Algorithms and Documentation

(adapted from the original document written by E. Flinchem for the User Manual for the Mac OS application Power OceanAtlas)

1. Overview

Java OceanAtlas provides considerable internal facilities for computing new results from input data. Calculations fall into five broad categories:

• Scalar functions
• Scalar functions smoothed in z
• Scalar functions integrated in z
• Interpolations onto fixed z-levels, i.e. grids

Results are reported in SI units except where otherwise noted.

2. Scalars

A cast consists of a series of levels ordered by increasing depth at a latitude, longitude coordinate pair. At each level (also known as a ‘bottle’) there exists a set of observed quantities, such as salinity (s), temperature (t), and pressure (p). Polynomial expansions in s, t, and p are used to compute the following derived quantities:

• density
• adiabatic lapse rate
• heat capacity
• sound velocity
• NO (respiration corrected nitrate)
• PO (respiration corrected phosphate)
• apparent oxygen utilization

From the elementary polynomials, four more derived quantities are constructed:

• potential temperature at one atmosphere
• potential density - 1000, i.e. sigma-theta, at five reference pressures
• specific volume anomaly
• heat content

The polynomials are taken from the literature and are reproduced here for reference.

NO and PO

JGR, 90, 6925-6939, 1985. Quantities must be in micromoles/liter.

$$NO(NO_3, O_2) = 8.7864 \times NO_3 + O_2$$
$$PO(PO_4, O_2) = 170.8467 \times PO_4 + O_2$$

Apparent Oxygen Utilization

Solubility Data Series v.7.

$$AOU = e^{-1268.9782 + \frac{36063.19}{T} + 220.1832 \times \log(T) - 0.351299\,T + S \times (0.006299 - \frac{3.5912}{T}) + 0.00000344 \times S^2} - O_2$$

Heat Capacity

JGR, 78, 4499-4507, 1973.

$$Cp(T) = 4.2174 - 3.720283 \times 10^{-3}T + 1.412855 \times 10^{-4}T^2 - 2.654387 \times 10^{-6}T^3 + 2.093236 \times 10^{-8}T^4$$
$$A = -1.381 \times 10^{-2} + 1.938 \times 10^{-4}T - 2.5 \times 10^{-6}T^2$$
$$B = 4.3 \times 10^{-4} - 9.9 \times 10^{-6}T + 1.3 \times 10^{-7}T^2$$
$$Cl‰ = \frac{S‰}{1.80655}$$
$$Cp(T, Cl‰) = Cp(T) + A \times Cl + B \times Cl^{3/2}$$

Adiabatic Temperature Gradient

Bryden, DSR, 20, 401-408, 1973. P is in decibars.

$$\Gamma(S, T, P) = \sum_i \sum_j \sum_k A_{ijk} P^i (S - 35)^j T^k$$

$$A_{000} = 0.35803 \times 10^{-1} \qquad A_{102} = 0.87330 \times 10^{-8}$$
$$A_{001} = 0.85258 \times 10^{-2} \qquad A_{103} = -0.54481 \times 10^{-10}$$
$$A_{002} = -0.68360 \times 10^{-4} \qquad A_{110} = -0.11351 \times 10^{-6}$$
$$A_{003} = 0.66228 \times 10^{-6} \qquad A_{111} = 0.27759 \times 10^{-8}$$
$$A_{010} = 0.18932 \times 10^{-2} \qquad A_{200} = -0.46206 \times 10^{-9}$$
$$A_{011} = -0.42393 \times 10^{-4} \qquad A_{201} = 0.18676 \times 10^{-10}$$
$$A_{100} = 0.18741 \times 10^{-4} \qquad A_{202} = -0.21687 \times 10^{-12}$$
$$A_{101} = -0.67795 \times 10^{-6}$$

Sound Velocity

Chen and Millero, J. of the Acoustical Society of America, 62, 1129-1135, 1977. P is in bars.

$$a_0 = 1.389 - 1.262 \times 10^{-2}t + 7.164 \times 10^{-5}t^2 + 2.006 \times 10^{-6}t^3 - 3.21 \times 10^{-8}t^4$$
$$a_1 = 9.4742 \times 10^{-5} - 1.258 \times 10^{-5}t - 6.4885 \times 10^{-8}t^2 + 1.0507 \times 10^{-8}t^3 - 2.0122 \times 10^{-10}t^4$$
$$a_2 = -3.9064 \times 10^{-7} + 9.1041 \times 10^{-9}t - 1.6002 \times 10^{-10}t^2 + 7.988 \times 10^{-12}t^3$$
$$a_3 = 1.1 \times 10^{-10} + 6.649 \times 10^{-12}t - 3.389 \times 10^{-13}t^2$$
$$a = a_0 + pa_1 + p^2a_2 + p^3a_3$$
$$b_0 = -1.922 \times 10^{-2} - 4.42 \times 10^{-5}t$$
$$b_1 = 7.3637 \times 10^{-5} + 1.7945 \times 10^{-7}t$$
$$b = b_0 + b_1p$$
$$c_0 = 1402.388 + 5.03711t - 5.80852 \times 10^{-2}t^2 + 3.342 \times 10^{-4}t^3 - 1.478 \times 10^{-6}t^4 + 3.1464 \times 10^{-9}t^5$$
$$c_1 = 0.153563 + 6.8982 \times 10^{-4}t - 8.1788 \times 10^{-6}t^2 + 1.3621 \times 10^{-7}t^3 - 6.1185 \times 10^{-10}t^4$$
$$c_2 = 3.126 \times 10^{-5} - 1.7107 \times 10^{-6}t + 2.5974 \times 10^{-8}t^2 - 2.5335 \times 10^{-10}t^3 + 1.0405 \times 10^{-12}t^4$$
$$c_3 = -9.7729 \times 10^{-9} + 3.8504 \times 10^{-10}t - 2.3643 \times 10^{-12}t^2$$
$$c = c_0 + pc_1 + p^2c_2 + p^3c_3$$
$$d = 1.727 \times 10^{-3} - 7.8936 \times 10^{-6}p$$
$$c_{sound} = c + as + bs^{3/2} + ds^2$$

Spiciness (Jacket and McDougall)

Jackett and McDougall, DSR, 32A, 1195-1208, 1985.

$$\tau(\theta,S) = \sum_{i=0}^{4} \sum_{j=0}^{4} a_{ij}\, \theta^i S^j$$

$$a = \begin{bmatrix} 1.609705 \times 10^{-1} & 6.542397 \times 10^{-1} & 5.222258 \times 10^{-4} & -2.586742 \times 10^{-5} & 7.565157 \times 10^{-7} \\ -8.007345 \times 10^{-2} & 5.309506 \times 10^{-3} & -9.612388 \times 10^{-5} & 3.211527 \times 10^{-6} & -4.610513 \times 10^{-8} \\ 1.081912 \times 10^{-2} & -1.561608 \times 10^{-4} & 3.774240 \times 10^{-6} & -1.150394 \times 10^{-7} & 1.146084 \times 10^{-9} \\ -1.451748 \times 10^{-4} & 3.485063 \times 10^{-6} & -1.387056 \times 10^{-7} & 3.737360 \times 10^{-9} & -2.967108 \times 10^{-11} \\ 1.219904 \times 10^{-6} & -3.591075 \times 10^{-8} & 1.953475 \times 10^{-9} & -5.279546 \times 10^{-11} & 4.227375 \times 10^{-13} \end{bmatrix}$$

Spiciness (Flament)

Flament, P., Progress in Oceanography Volume 54, 2002, Pages 493-501

$$\pi(\theta,s) = \sum_{i=0}^{5} \sum_{j=0}^{4} b_{ij}\theta^{i}(s-35)^{j}$$

where:

i	j	b[i,j]	
0	0	0.0	
0	1	7.7442	(-1)
0	2	-5.85	(-3)
0	3	-9.84	(-4)
0	4	-2.06	(-4)
1	0	5.1655	(-2)
1	1	2.034	(-3)
1	2	-2.742	(-4)
1	3	-8.5	(-6)
1	4	1.36	(-5)
2	0	6.64783	(-3)
2	1	-2.4681	(-4)
2	2	-1.428	(-5)
2	3	3.337	(-5)
2	4	7.894	(-6)
3	0	-5.4023	(-5)
3	1	7.326	(-6)
3	2	7.0036	(-6)
3	3	-3.0412	(-6)
3	4	-1.0853	(-6)
4	0	3.949	(-7)
4	1	-3.029	(-8)
4	2	-3.8209	(-7)
4	3	1.0012	(-7)
4	4	4.7133	(-8)
5	0	-6.36	(-10)
5	1	-1.309	(-9)
5	2	6.048	(-9)
5	3	-1.1409	(-9)
5	4	-6.676	(-10)

Density

Millero, et al., DSR, 27A, 255-264, 1980.
Millero, and Poisson DSR, 28A, 625-629, 1981. P is in bars.

$$\rho_w^0 = 999.842594 + 6.793952 \times 10^{-2}T - 9.095290 \times 10^{-3}T^2 + 1.001685 \times 10^{-4}T^3 - 1.120083 \times 10^{-6}T^4 + 6.536332 \times 10^{-9}T^5$$

$$A = .824493 - 4.0899 \times 10^{-3}T + 7.6438 \times 10^{-5}T^2 - 8.2467 \times 10^{-7}T^3 + 5.3875 \times 10^{-9}T^4$$

$$B = -5.72466 \times 10^{-3} + 1.0227 \times 10^{-4}T - 1.6546 \times 10^{-6}T^2$$

$$C = 4.8314 \times 10^{-4}$$

$$a = 54.6746 - 0.603459T + 1.09987 \times 10^{-2}T^2 - 6.1670 \times 10^{-5}T^3$$

$$b = 7.944 \times 10^{-2} + 1.6483 \times 10^{-2}T - 5.3009 \times 10^{-4}T^2$$

$$c = 2.2838 \times 10^{-3} - 1.0981 \times 10^{-5}T - 1.6078 \times 10^{-6}T^2$$

$$d = 1.9107 \times 10^{-4}$$

$$e = -9.9348 \times 10^{-7} + 2.0816 \times 10^{-8}T + 9.1697 \times 10^{-10}T^2$$

$$A_w = 3.239908 + 1.43713 \times 10^{-3}T + 1.16092 \times 10^{-4}T^2 - 5.77905 \times 10^{-4}T^3$$

$$B_w = 8.50935 \times 10^{-5} - 6.12293 \times 10^{-6}T + 5.2787 \times 10^{-8}T^2$$

$$K_w^0 = 19652.21 + 148.4206T - 2.327105T^2 + 1.360477 \times 10^{-2}T^3 - 5.155288 \times 10^{-5}T^4$$

$$D = A_w + cS + dS^{3/2}$$

$$E = B_w + eS$$

$$K^0 = K_w^0 + aS + bS^{3/2}$$

$$K = K^0 + DP + EP^2$$

$$\rho^0 = \rho_w^0 + AS + BS^{3/2} + CS^2$$

$$\rho = \frac{\rho^0}{1 - \frac{P}{K}}$$

Neutral Density (Gamma)

Neutral Density: A description of neutral density can be found at http://oceanworld.tamu.edu/resources/ocng_textbook/chapter06/chapter06_05.htm and http://sam.ucsd.edu/sio210/lect_2/lecture_2.html. The original article is: Jackett, D., R., and T. J. McDougall, 1997: A neutral density variable for the world's oceans. Journal of Physical Oceanography, 27, 237-263.

Other Scalars

Theta is generated from the adiabatic temperature gradient by a single step of Runge-Kutta integration, Fofonoff, DSR, 24, 489-491, 1977.

Specific volume anomaly follows directly from density.

Sigma-Theta is formed by substitution of theta for t in the density polynomial.

Heat Content is the product of the temperature and the specific heat.

3. Scalar functions smoothed in z

Three quantities related to the buoyancy frequency are available, n, n2, and fn2/g. The approximation used is:

$$n^2 = \frac{g\frac{d\rho}{dz}}{\rho}$$

The raw densities are smoothed by the application of a Gaussian weighting function, $e^{-(a/z)^2}$, where a and z are in meters, and then centered finite differences are taken. The result is reported in cycles per hour.

4. Scalar functions integrated in z

The following quantities are computed as running sums of trapezoids down each cast:

• Geopotential Anomaly
• Heat Storage
• Potential Energy Anomaly
• Acoustic Travel Time

Sums are computed from the surface down with a sign convention which gives increasing negative values. When a reference level at depth is subtracted, the values above the reference level are typically positive. If the uppermost data point in a cast is not at the surface, then the layer between the surface and the first data point is assumed to have uniform properties. Note the following definition:

$$\Delta p_i \equiv p_i - p_{i+1}$$
$$\Delta z_i \equiv z_{i+1} - z_i$$

Geopotential Anomaly

$$\Delta\Phi = -\int_{p=0}^{p=P_B} \delta_{s,t,p}\,dp$$
$$\Delta\Phi_n = \frac{1}{2}\sum_{i=1}^{n} \left(\delta_i + \delta_{i+1}\right)\Delta p_i$$

Net Heat Content

$$\text{HEAT} = -\int_{z=0}^{z=Z_B} \rho_\Theta Cp_{s,\Theta} \Theta \, dz$$

$$\text{HEAT}_n = \frac{1}{8}\sum_{i=1}^{n} (Cp_i + Cp_{i+1})(\Theta_i + \Theta_{i+1})(\rho_{\Theta i} + \rho_{\Theta i+1}) \Delta z_i$$

Potential Energy Anomaly

$$\chi = -\frac{1}{g}\int_{p=0}^{p=P_B} p\,\delta_{s,t,p} \, dp$$

$$\chi_n = \frac{1}{4g}\sum_{i=1}^{n} (\delta_i + \delta_{i+1})(p_i + p_{i+1}) \Delta p_i$$

Acoustic Travel Time

$$T = -\int_{z=0}^{z=Z_B} \frac{dz}{c_{s,t,p}}$$

$$T_n = \frac{1}{2}\sum_{i=1}^{n} \frac{\Delta z_i}{(c_i - c_{i+1})}$$

ZGrid Comments

Sets up square grid for contouring , given arbitrarily placed data points. Laplace interpolation is used. The method used here was lifted directly from notes left by Mr Ian Crain formerly with the comp.science div.

Info on relaxation soln of laplace eqn supplied by Dr T Murty. Fortran II oceanography/emr Dec/68 jdt

z = 2-d array of hgts to be set up. points outside region to be contoured should be initialized to 10**35 . the rest should be 0.0
nx,ny = max subscripts of z in x and y directions .
x1,y1 = coordinates of z(1,1)
dx,dy = x and y increments .
xp,yp,zp = arrays giving position and hgt of each data point.
n = size of arrays xp,yp and zp .

Modification feb/69 To get smoother results a portion of the beam eqn was added to the laplace eqn giving delta2x(z)+delta2y(z) - k(delta4x(z)+delta4y(z)) = 0. k=0 gives pure laplace solution. k=inf. gives pure spline solution. cayin = k = amount of spline eqn (between 0 and inf.) nrng...grid points more than nrng grid spaces from the nearest data point are set to undefined.

Modification Dec23/69   Data pts no longer moved to grid pts.

Modification May 5 79 Common blocks work1 and work2 must be dimension at least n points long by the user. Common block work3 must be dimensioned at least ny points long.

Modification June 17,1985 - Handles data values of 1e35. If at least one data value near a grid point is equal to 1e35, the z array is initialized to 1e35 at that grid point - by G.R. Halliwell

Modification March 31, 2005 – Arithmetic ifs converted to block ifs. 1e35 changed to Double.NaN. Ported to Java. - by D.W. Denbo

WOCE and IGOSS Quality Codes

The data files in WHP-exchange format contain WOCE quality flags, where these have been provided. Here is an overview of WOCE Hydrographic Program (WHP) quality flags and their translation to IGOSS quality flags, provided by the WHP Office:

The original WHP data formats contained provisions for up to two quality bytes per measured parameter. The first quality byte, or quality code, was to be assigned by the data provider for the parameter in question, and the second was to be provided on review of the data by external Data Quality Experts (DQEs), or possibly the WHP Office, which is the Data Assembly Center for the WHP.

Meanwhile, there were growing problems with the original WHP data formats (.sum, .sea/.hyd, and .ctd data files - see documentation at <http://whpo.ucsd.edu>). As a result, a shift was made to more nearly exactly described CTD and bottle data formats: the WHP-Exchange formats, or _hy1.csv files for bottle data and _ct1.csv files for CTD data. These include only a single (or no) quality byte per parameter. The QUALT2 byte is used where it exists and otherwise the QUALT1 byte is used, and the quality byte accompanies the data values (immediately follows) in the WHP-Exchange data file.

In order that WHP data can best be used with other data, it may be advantageous for some users to translate the WOCE-specific quality codes into the more widely recognized IGOSS quality codes.

The WHP quality codes for the water bottle itself are:

1	Bottle information unavailable.
2	No problems noted.
3	Leaking.
4	Did not trip correctly.

5 Not reported.
6 Significant discrepancy in measured values between Gerard and Niskin bottles.
7 Unknown problem.
8 Pair did not trip correctly. Note that the Niskin bottle can trip at an unplanned depth while the Gerard trips correctly and vice versa.
9 Samples not drawn from this bottle.

Flags 6, 7, and 8 apply primarily to large volume samplers.

The WHP bottle parameter data quality codes are:

1 Sample for this measurement was drawn from water bottle but analysis not received. Note that if water is drawn for any measurement from a water bottle, the quality flag for that parameter must be set equal to 1 initially to ensure that all water samples are accounted for.
2 Acceptable measurement.
3 Questionable measurement.
4 Bad measurement.
5 Not reported.
6 Mean of replicate measurements (Number of replicates should be specified in the —.DOC file and replicate data tabulated).
7 Manual chromatographic peak measurement.
8 Irregular digital chromatographic peak integration.
9 Sample not drawn for this measurement from this bottle.

The WHP CTD data quality codes are:

1 Not calibrated.
2 Acceptable measurement.
3 Questionable measurement.
4 Bad measurement.
5 Not reported.
6 Interpolated over >2 dbar interval.
7 Despiked.
8 Not assigned for CTD data.
9 Not sampled.

The WMO IGOSS observation quality codes are:

0 No quality control yet assigned to this element
1 The element appears to be correct
2 The element is probably good
3 The element is probably bad

4	The element appears erroneous
5	The element has been changed
6 to 8	Reserved for future use
9	The element is missing

A perfect translation is probably not feasible, but the WHP Office suggests the following WHP-to-IGOSS (not IGOSS-to-WHP) translation rules as reasonable (these translation rules are used by Java OceanAtlas):

WOCE bottle	IGOSS
1	0
2	1
3	3 (see note #1)
4	4
5	0
6	4
7	4
8	4
9	9

water sample	
1	0
2	1
3	2 (see note #2)
4	4
5	0
6	2
7	2
8	2
9	9

ctd	
1	0
2	1
3	2 (see note #2)
4	4
5	0
6	2
7	2
9	9

Note #1: The WHP Office, in the interest of being conservative, has chosen to translate the WOCE bottle quality code 3 into IGOSS quality code 3. A leaking water sample bottle typically results in a discrepancy or error in gas samples, such as oxygen and CFCs, but less often results in data discrepancies for salinity and nutrients. It is suggested that data users who wish to import only "good" data not import any water sample data from bottles with a WOCE code 3 or IGOSS code 3. A data user who is willing to entertain slightly greater risk might choose to import non-gas sample data (e.g., salinity and nutrients) from a WOCE code 3 or IGOSS code 3 water sample bottle, and allow import of gas sample data (e.g. oxygens and CFCs) for bottles with IGOSS Code 2. (The WHP Office is not, however, currently assigning IGOSS code 2 to water sample bottles; but future data originators or data centers may wish to use code 2.)

Note #2: The WHP Office has noted that in general, data originators tend to be conservative and so in DQE reports many WHP code 3 ("questionable") water sample parameter data are deemed WHP code 2 ("good") by the examiners. The IGOSS code 2 ("probably good") seems to be a reasonable interpretation. The WHP Office is not currently assigning IGOSS code 3 ("probably bad") to WHP water sample data values.

Tables showing the parameter values and colors of the Java OceanAtlas color bars:

PRES 0-6000	PRES 0-1000	TEMP_global	THTA_global	SALT_global	SIG0_global	SIG1_global
red	red	blue	blue	blue	red	red
50	20	-1.5	-1.5	33	22	27
100	40	-1.1	-1.1	33.5	22.5	29
150	60	-1	-1	34	23	30
200	80	-0.9	-0.9	34.2	23.5	31
250	100	-0.6	-0.6	34.4	24	31.2
300	120	-0.3	-0.3	34.6	24.5	31.4
400	140	0	0	34.66	25	31.5
500	160	0.3	0.3	34.67	25.5	31.6
600	180	0.6	0.6	34.68	26	31.65
700	200	0.9	0.9	34.69	26.5	31.7
800	225	1.2	1.2	34.7	26.8	31.75
1000	250	1.5	1.5	34.71	27	31.8
1200	275	1.8	1.8	34.72	27.2	31.85
1400	300	2.1	2.1	34.74	27.4	31.9
1600	325	2.4	2.4	34.76	27.5	31.95
1800	350	2.7	2.7	34.785	27.6	32
2000	375	3	3	34.81	27.65	32.05
2250	400	3.3	3.3	34.835	27.7	32.1
2500	425	3.6	3.6	34.86	27.74	32.15
2750	450	3.9	3.9	34.87	27.76	32.2
3000	475	4.4	4.4	34.88	27.78	32.25
3250	500	5	5	34.89	27.8	32.3
3500	550	6	6	34.9	27.82	32.35
3750	600	8	8	34.91	27.84	32.4
4000	650	10	10	34.92	27.86	32.45
4250	700	12	12	34.94	27.88	32.5
4500	750	14	14	35	27.9	32.55
4750	800	16	16	35.1	27.95	32.6
5000	850	19	19	35.4	28	32.65
5250	900	22	22	35.7	28.05	32.7
5500	950	25	25	36	28.08	32.75
5750	1000	28	28	36.5	28.09	32.8
green	green	red	red	orange	green	green

SIG2_global	SIG3_global	SIG4_global	O2_global	O2_Arctic	O2UM_global	O2%
red	red	red	yellow	yellow	yellow	rainbow-red
32	35	41	0.2	5.8	10	4
33	37	43	0.6	5.9	25	7.4
34	38	44	1	6	45	10.8
35	39	45	1.4	6.1	60	14.2
36	40	45.3	1.8	6.2	75	17.5
36.1	40.5	45.5	2.2	6.3	90	20.9
36.2	40.7	45.55	2.5	6.4	105	24.3
36.25	40.8	45.6	2.8	6.5	118	27.7
36.3	40.9	45.65	3.1	6.6	131	31.1
36.35	41	45.7	3.4	6.65	144	34.5
36.4	41.05	45.73	3.7	6.7	157	37.9
36.45	41.1	45.76	4	6.75	170	41.3
36.5	41.15	45.79	4.3	6.8	183	44.6
36.55	41.2	45.82	4.6	6.85	196	48
36.6	41.25	45.85	4.9	6.9	209	51.4
36.65	41.3	45.88	5.2	6.95	222	54.8
36.7	41.34	45.91	5.5	7	235	58.2
36.75	41.38	45.94	5.8	7.05	248	61.6
36.8	41.42	45.97	6.1	7.1	261	65
36.85	41.46	46	6.4	7.2	274	68.4
36.9	41.5	46.03	6.7	7.3	287	71.7
36.95	41.54	46.06	7	7.5	300	75.1
37	41.58	46.09	7.4	7.7	318	78.5
37.05	41.62	46.12	7.8	7.9	336	81.9
37.1	41.66	46.15	8.2	8.2	354	85.3
37.15	41.7	46.18				88.7
37.2	41.75	46.21				92.1
37.25	41.8	46.25				95.5
37.3	41.85	46.29				98.8
37.35	41.9	46.33				102.2
37.4	41.95	46.37				105.6
37.45	42	46.4				109
green	green	green	purple	purple	purple	rainbow-purple

NO3_global	NO2	PO4_global	SIO3_global	NO_global	PO_global	NUTR
green	white	orange	yellow	green	orange	rainbow-red
0.2	0.01	0.08	0.5	170	215	0.5
0.5	0.025	0.18	1	190	235	1.5
0.8	0.05	0.28	2	210	255	2.5
1.1	0.1	0.38	3	230	275	3.5
1.4	0.15	0.48	4	245	290	4.5
1.7	0.2	0.58	5	260	305	5.5
2	0.25	0.68	6	275	320	6.5
2.3	0.3	0.78	7	290	335	7.5
2.6	0.35	0.88	8	305	350	8.5
2.9	0.4	0.98	9	320	365	9.5
3.2	0.45	1.08	10	335	380	10.3
4	0.5	1.18	11	350	395	10.8
6	0.55	1.28	12	365	410	11.2
8	0.6	1.38	13	380	425	11.6
10	0.65	1.48	20	395	440	12
12	0.7	1.58	30	410	455	12.4
14	0.75	1.68	40	425	470	12.8
16	0.8	1.78	50	435	485	13.2
18	0.85	1.88	60	445	500	13.6
20	0.9	1.98	70	455	515	14
22	0.95	2.08	80	465	530	14.4
24	1	2.18	90	475	545	14.8
26	1.1	2.28	100	485	560	15.2
28	1.2	2.38	110	495	575	15.6
30	1.3	2.48	120	510	590	16
32	1.4	2.58	130	525	605	16.4
34	1.6	2.68	140	540	620	16.8
36	1.8	2.78	150	555	635	17.2
38	2	2.88	160	570	655	17.6
40	2.25	2.98	170	590	675	18
42	2.5	3.08	180	610	695	19
44	2.75	3.18	190	630	715	20
		3.28				
purple	purple (blue mid-range)	green	red	purple	green	rainbow-purple NO3/PO4

NOPO	AOU	ALPH	ADRV	BETA	BDRV	RAT1
rainbow-red	blue	blue	blue	blue	blue	blue
0.65	-35	0.07	-5	-3.4	-3	-200
0.661	-20	0.116	-4.832	-3.335	-2.877	-180
0.673	-5	0.162	-4.665	-3.271	-2.755	-160
0.684	5	0.208	-4.497	-3.206	-2.632	-140
0.695	20	0.255	-4.329	-3.142	-2.51	-120
0.706	34	0.301	-4.161	-3.077	-2.387	-100
0.718	45	0.347	-3.994	-3.013	-2.265	-80
0.729	56	0.393	-3.826	-2.948	-2.142	-60
0.74	67	0.439	-3.658	-2.884	-2.019	-40
0.752	78	0.485	-3.49	-2.819	-1.897	-30
0.763	89	0.531	-3.323	-2.755	-1.774	-20
0.774	100	0.577	-3.155	-2.69	-1.652	-15
0.785	111	0.624	-2.987	-2.626	-1.529	-11
0.797	122	0.67	-2.819	-2.561	-1.406	-6
0.808	133	0.716	-2.652	-2.497	-1.284	-3
0.819	144	0.762	-2.484	-2.432	-1.161	0
0.831	155	0.808	-2.316	-2.368	-1.039	3
0.842	166	0.854	-2.148	-2.303	-0.916	6
0.853	177	0.9	-1.981	-2.239	-0.794	11
0.865	188	0.946	-1.813	-2.174	-0.671	15
0.876	199	0.993	-1.645	-2.11	-0.548	20
0.887	210	1.039	-1.477	-2.045	-0.426	30
0.898	221	1.085	-1.31	-1.981	-0.303	40
0.91	232	1.131	-1.142	-1.916	-0.181	50
0.921	243	1.177	-0.974	-1.852	-0.058	60
0.932	254	1.223	-0.806	-1.787	0.065	70
0.944	265	1.269	-0.639	-1.723	0.187	80
0.955	276	1.315	-0.471	-1.658	0.31	90
0.966	287	1.362	-0.303	-1.594	0.432	100
0.977	298	1.408	-0.135	-1.529	0.555	110
0.989	309	1.454	0.032	-1.465	0.677	130
1	320	1.5	0.2	-1.4	0.8	150
rainbow-purple	red	red	red	red	red	red
NO/PO						ADRV/BDRV

BV3	VT3	SB3	SPCY	SVAN	SVEL
rainbow-purple	rainbow-purple	rainbow-purple	rainbow-purple	blue	blue
0.1	0.002	1	21	-6	1440
0.2	0.005	2	23	-2	1443
0.3	0.01	3	25	2	1446
0.4	0.015	4	27	6	1449
0.5	0.02	5	27.08	10	1452
0.6	0.025	7	27.16	15	1455
0.7	0.03	9	27.24	20	1458
0.8	0.04	11	27.32	25	1461
0.9	0.05	15	27.4	30	1464
1	0.075	20	27.48	35	1467
1.1	0.1	25	27.56	40	1470
1.2	0.15	30	27.64	45	1473
1.3	0.2	35	27.72	50	1476
1.4	0.3	40	27.8	55	1479
1.5	0.4	45	27.88	60	1482
1.6	0.5	50	27.96	65	1485
1.7	0.6	55	28.04	70	1488
1.8	0.7	60	28.12	75	1491
1.9	0.8	70	28.2	80	1494
2	0.9	80	28.28	85	1497
2.5	1	90	28.36	90	1500
3	1.1	100	28.44	100	1503
4	1.2	120	28.52	150	1506
5	1.3	140	28.6	200	1509
6	1.4	160	28.68	250	1512
7	1.5	180	28.76	300	1515
8	1.6	200	28.84	350	1518
10	1.7	220	28.92	400	1521
12	1.8	240	29	450	1524
14	2	260	29.5	500	1527
16	2.5	280	30	550	1530
18	3	300	30.5	600	1533
					1536
rainbow-red	rainbow-red	rainbow-red	rainbow-red	red	1539
					1542
					1545
					1548
					1551

red

Table of the Color Values Used for the End Points in the Java OceanAtlas Colorbars:

Color	Hue	Saturation	Brightness
red	0	100	85
green	135	100	65
blue	240	100	85
orange	35	100	90
yellow	60	100	75
purple	275	100	80

Tables showing the standard levels for the interpolation surfaces included with Java OceanAtlas:

surface name:	PRES-0-1000	PRES-0-300	PRES-0-600	PRES-0-6000	PRES-0-6000(deep)
level 1	0	0	0	0	0
2	5	2	5	5	50
3	10	4	10	10	100
4	15	7	15	20	150
5	20	10	20	35	200
6	25	15	30	50	300
7	30	20	40	65	400
8	40	25	50	80	500
9	50	30	60	100	600
10	60	35	70	125	700
11	70	40	80	150	800
12	80	45	90	175	900
13	90	50	100	200	1000
14	100	55	110	225	1100
15	110	60	120	250	1200
16	120	65	130	275	1300
17	130	70	140	300	1400
18	140	75	150	325	1500
19	150	80	160	350	1600
20	160	85	170	375	1700
21	170	90	180	400	1800
22	180	95	190	425	1900
23	200	100	200	450	2000
24	220	105	210	475	2100
25	240	110	220	500	2200
26	260	115	230	550	2300
27	280	120	240	600	2400
28	300	125	250	650	2500
29	320	130	260	700	2600
30	340	135	270	750	2700
31	360	140	280	800	2800
32	380	145	290	850	2900
33	400	150	300	900	3000
34	420	155	310	1000	3100
35	440	160	320	1100	3200
36	460	165	330	1200	3300
37	480	170	340	1300	3400
38	500	175	350	1400	3500

39	520	180	360	1500	3600
40	540	185	370	1600	3700
41	560	190	380	1700	3800

surface name:	PRES-0-1000	PRES-0-300	PRES-0-600	PRES-0-6000	PRES-0-6000(deep)
level 42	580	195	390	1800	3900
43	600	200	400	2000	4000
44	620	205	410	2200	4100
45	640	210	420	2400	4200
46	660	215	430	2600	4300
47	680	220	440	2800	4400
48	700	225	450	3000	4500
49	720	230	460	3200	4600
50	740	235	470	3400	4700
51	760	240	480	3600	4800
52	780	245	490	3800	4900
53	800	250	500	4000	5000
54	820	255	510	4200	5100
55	840	260	520	4400	5200
56	860	265	530	4600	5300
57	880	270	540	4800	5400
58	900	275	550	5000	5500
59	920	280	560	5200	5600
60	940	285	570	5400	5700
61	960	290	580	5600	5800
62	980	295	590	5800	5900
63	1000	300	600	6000	6000

surface name:	SIG0-global	SIG0-Arctic.high	SIG1-global	SIG2-global
level 1	22	27	27	32
2	22.5	27.1	29	33.5
3	23	27.2	31	34.5
4	23.5	27.3	31.05	35
5	24	27.4	31.1	35.2
6	24.2	27.5	31.15	35.4
7	24.4	27.55	31.2	35.5
8	24.6	27.6	31.25	35.6
9	24.8	27.65	31.3	35.7
10	25	27.7	31.35	35.8
11	25.2	27.73	31.4	35.85
12	25.4	27.75	31.45	35.9
13	25.6	27.76	31.5	35.95
14	25.8	27.77	31.55	36
15	26	27.78	31.6	36.05
16	26.2	27.79	31.625	36.1
17	26.4	27.8	31.65	36.15
18	26.6	27.81	31.675	36.2
19	26.8	27.82	31.7	36.25
20	26.9	27.83	31.725	36.3
21	27	27.84	31.75	36.35
22	27.1	27.85	31.775	36.4
23	27.15	27.86	31.8	36.45
24	27.2	27.87	31.825	36.5
25	27.25	27.88	31.85	36.525
26	27.3	27.89	31.875	36.55
27	27.35	27.9	31.9	36.575
28	27.4	27.91	31.925	36.6
29	27.45	27.92	31.95	36.625
30	27.5	27.925	31.975	36.65
31	27.53	27.93	32	36.675
32	27.56	27.935	32.025	36.7
33	27.58	27.94	32.05	36.725
34	27.595	27.945	32.075	36.75
35	27.61	27.95	32.1	36.775
36	27.625	27.955	32.125	36.8
37	27.64	27.96	32.15	36.825
38	27.655	27.965	32.175	36.85
39	27.67	27.97	32.2	36.875
40	27.685	27.975	32.225	36.9

41	27.7	27.98	32.25	36.925

surface name:	SIG0-global	SIG0- Arctic.high	SIG1-global	SIG2-global
level 42	27.715	27.985	32.275	36.95
43	27.73	27.99	32.3	36.975
44	27.745	27.995	32.325	37
45	27.76	28	32.35	37.025
46	27.775	28.005	32.375	37.05
47	27.79	28.01	32.4	37.075
48	27.805	28.015	32.425	37.1
49	27.82	28.02	32.45	37.125
50	27.835	28.025	32.475	37.15
51	27.85	28.03	32.5	37.175
52	27.865	28.035	32.525	37.2
53	27.88	28.04	32.55	37.225
54	27.895	28.045	32.575	37.25
55	27.91	28.05	32.6	37.275
56	27.93	28.055	32.625	37.3
57	27.95	28.06	32.65	37.325
58	27.97	28.065	32.675	37.35
59	27.99	28.07	32.7	37.375
60	28.01	28.075	32.725	37.4
61	28.03	28.08	32.75	37.425
63	28.05	28.085	32.775	37.45
63	28.07	28.09	32.8	37.46

surface name:	SIG3-global	SIG4-global	TEMP-global	THTA-global
level 1	35	40	-1.8	-1.8
2	37	42	-1.6	-1.6
3	39	44	-1.4	-1.4
4	40	44.5	-1.2	-1.2
5	40.25	44.75	-1	-1
6	40.4	45	-0.8	-0.8
7	40.5	45.25	-0.6	-0.6
8	40.6	45.4	-0.4	-0.4
9	40.7	45.45	-0.2	-0.2
10	40.8	45.5	0	0
11	40.9	45.515	0.2	0.2
12	41	45.53	0.4	0.4
13	41.05	45.545	0.6	0.6
14	41.1	45.56	0.8	0.8
15	41.15	45.575	1	1
16	41.175	45.59	1.2	1.2
17	41.2	45.605	1.4	1.4
18	41.215	45.62	1.6	1.6
19	41.23	45.635	1.8	1.8
20	41.245	45.65	2	2
21	41.26	45.665	2.2	2.2
22	41.275	45.68	2.4	2.4
23	41.29	45.695	2.6	2.6
24	41.305	45.71	2.8	2.8
25	41.32	45.725	3	3
26	41.335	45.74	3.2	3.2
27	41.35	45.755	3.4	3.4
28	41.365	45.77	3.6	3.6
29	41.38	45.785	3.8	3.8
30	41.395	45.8	4	4
31	41.41	45.815	4.2	4.2
32	41.425	45.83	4.4	4.4
33	41.44	45.845	4.6	4.6
34	41.455	45.86	4.8	4.8
35	41.47	45.875	5	5
36	41.485	45.89	5.2	5.2
37	41.5	45.905	5.4	5.4
38	41.515	45.92	5.6	5.6
39	41.53	45.935	5.8	5.8
40	41.545	45.95	6	6
41	41.56	45.965	6.2	6.2

surface name:	SIG3-global	SIG4-global	TEMP-global	THTA-global
level 42	41.575	45.98	6.4	6.4
43	41.59	45.995	6.6	6.6
44	41.605	46.01	6.8	6.8
45	41.62	46.025	7	7
46	41.635	46.04	7.5	7.5
47	41.65	46.055	8	8
48	41.665	46.07	8.5	8.5
49	41.68	46.085	9	9
50	41.695	46.1	9.5	9.5
51	41.71	46.115	10	10
52	41.74	46.13	11	11
53	41.8	46.15	12	12
54	41.85	46.2	13	13
55	41.9	46.25	14	14
56	41.92	46.3	16	16
57	41.94	46.34	18	18
58	41.95	46.355	20	20
59	41.96	46.37	22	22
60	41.97	46.385	24	24
61	41.98	46.4	26	26
63	41.99	46.415	28	28
63	42	46.43	30	30