

Space Details

Key:	UpperWaterColumnDataVis
Name:	Upper Water Column Data Visualization
Description:	Continue development of infrastructure to enable visualization of MBARI's upper water column observations.
Creator (Creation Date):	mccann (Jan 25, 2010)
Last Modifier (Mod. Date):	mccann (Jan 26, 2010)

Available Pages

- Home 
 - Application Development Environment assessment
 - Conceptual Data Flow Diagram
 - Data Flow diagram with CANON DSS plans for 2011
 - Decision tree for best way to access and work with Upper Water Column Data
 - postgres login via psql with a database acct - not a unix user acct
 - SQLAlchemy, GeoAlchemy, and psycopg2
 - Stoqs OR Mapped Classes
 - Upper Water Column Data holdings inventory
- Postgres Notes - first try

Home

This page last changed on Jan 12, 2012 by [mccann](#).

2010 Project: Upper Water Column Data Visualization (901005)

News flash on 12 January 2012!

The results of this project are now an open source project on Google Code. Please see <http://code.google.com/p/stoqs/> and contribute!

Notes from proposal process in 2009

- [Abstract submitted](#) on 1 July 2009
- [Presentation given](#) to management team on 7 July 2009
- [Original proposal](#) submitted on 4 August 2009
- Phase 1 [Management Team feedback](#) returned on 9 September 2009
- [Response to management team feedback](#) submitted on 17 September 2009
- [Approved Project Proposal 901005](#) (on PCO web site)

Meeting notes and presentations 2010

- [Conceptual Data Flow diagram](#) from discussions between Rich and Mike on 25 January 2010
- CANON Decision Support System (DSS) meeting [presentation on Application Development Environments for working with Upper Water Column Data](#) on 3 February 2010
- [Presentation given at CANON planning meeting](#) on 18 February 2010

Analyses

- [Upper Water Column Data holdings inventory](#): nature and size of the various archives
- [Application Development Environment assessment](#): Refers to the "sandbox" in the [Conceptual Data Flow diagram](#)
- Decision tree for best way to access and work with Upper Water Column Data

Developer Notes

- [Postgres Notes - first try](#) (Notes on installing PostgreSQL on Linux and getting started with a database)
- Additional tools that are needed: [SQLAlchemy](#), [GeoAlchemy](#), [psycpg2](#)
- PostgreSQL Notes: [login via psql with a database account](#) (ala expddba), not a unix user acct.
- [Stoqs OR Mapped Base Classes Diagram](#)
- Graphical User Interfaces for PostgreSQL databases:
 - [pgadmin3 screenshot running on Posiedon Linux](#)
 - [SQL Developer screenshot running on MacOS pointing to the DB running on Linux](#)
 -

Deliverables

- [STOQS project](#) - link to SVN repository of Python stoqs module (ORM to STOQS Database) and example scripts including web scripts for Google Earth Display
- Link to [see data from STOQS in Google Earth](#) (Expand the "Upper Water Column Data" folder and check the "STOQS Data" folder to load the timestamped data.)
- [Data Flow diagram with CANON DSS plans for 2011](#)
- Coming soon: link(s) to videos on YouTube of 4-D fly-thrus of upper water column oceanographic data

Application Development Environment assessment

This page last changed on Jan 28, 2010 by [mccann](#).

Application Development Environment assessment table

Development environment	OPeNDA NetCDF	SQL (CRUD)	SQL (ORM)	Numeric Methods	Visualization	Easy to learn	Freely available	Large user base	Support community
Matlab	 read & write								
Perl	 write								
Python	 read & write								
Java	 read & write								
Ferret	 read & write (grid only)								

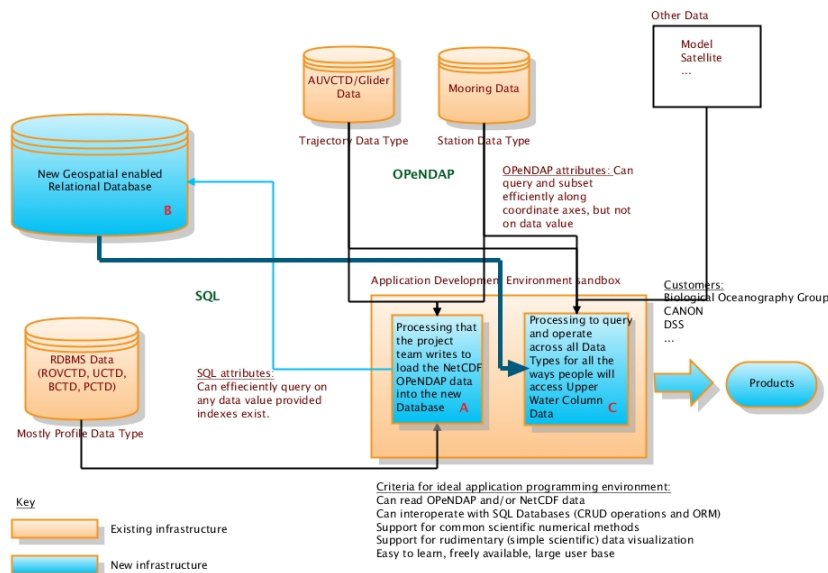
Assessment criteria

- OPeNDAP/NetCDF: For write there needs to be a library available to create NetCDF files. For read there needs to be a capability (library or toolkit) to read both grid and structure DAPs.
- SQL (CRUD): Ability to perform Create, Read, Update, Delete operations in a relational database.
- SQL (ORM): The language as libraries to permit Object Relational Mapping, which makes it easier to use relational data in an object oriented environment.
- Numerical methods: Common numerical methods are available such as those found in the [Numerical Recipes](#) book.
- Visualization: Capability to produce common scientific graphics (multi-axis XY plots, histograms, scatter plots, color-filled contour plots, vector time series plots, geographic plots, etc.)
- Easy to learn: This is rather subjective, but a discriminating parameter is whether professional software engineering skills are required, or if the language can be "picked up" by someone with scientific technical skills.
- Freely available: Yes, this means free - i.e. you don't have to pay for a license.
- Large user base: This means that more than perhaps 100 people use it from around the world.
- Support community: Is there a user group where you can post questions and get timely answers back.

Conceptual Data Flow Diagram

This page last changed on Jan 26, 2010 by [mccann](#).

This diagram shows the existing data archive infrastructure in sand-colored shapes and the new infrastructure to be developed by this project and its customers in blue-colored shapes.



Box A and repository B will be developed as part of the Upper Water Column Data Visualization project. There will be some development done in box C as part of this project, but mainly to perform engineering tests of the system and to demonstrate access by delivering representations of the data to Google Earth. For more in depth analysis and use for the data (e.g. for decision support, model forcing/verification) we expect those users will work in the same application development environment that the developers of the infrastructure are using.

Data Flow diagram with CANON DSS plans for 2011

This page last changed on Dec 05, 2010 by [mccann](#).

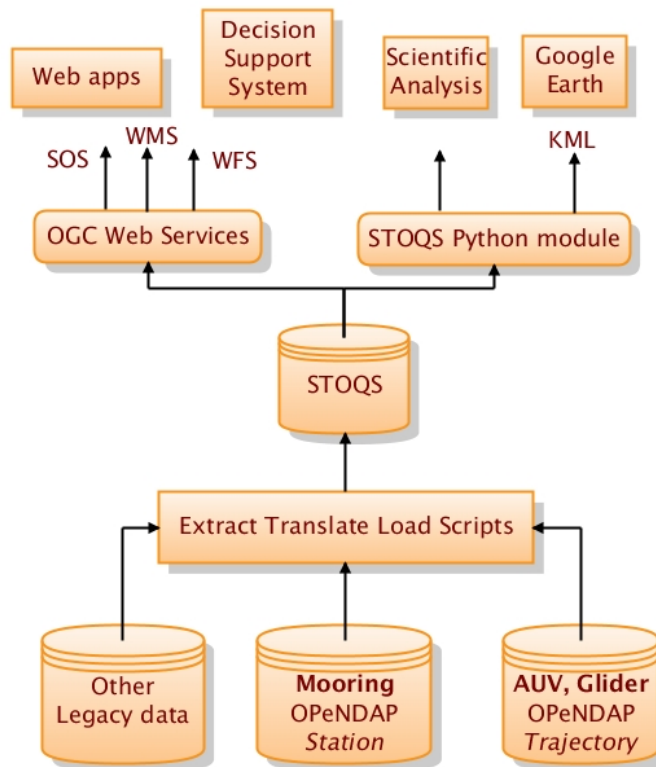


Diagram modified a bit from the original one created at the start of this project

Decision tree for best way to access and work with Upper Water Column Data

This page last changed on Feb 04, 2010 by [mccann](#).

Here is a flow chart showing a decision tree for the best way to access and work with SQL & OPeNDAP data:

postgres login via psql with a database acct - not a unix user acct

This page last changed on Mar 01, 2010 by rich.

DATE: 3/1/2010 by rschramm

ISSUE: on ubuntu vm, I cant log into postgres with psql as database user expddba (there is no unix expddba)

SOLUTION: there is a postgres file that controls authentication method. Needed to change from IDENT to TRUST. Tis may not be our 'final' best solution, need to know more about it. See ref#2 below for more on that. See ref#1 for the clues that led me to the steps outlined below.

see ref#1: <http://www.cyberciti.biz/faq/psql-fatal-ident-authentication-failed-for-user/>
also ref#2: <http://www.postgresql.org/docs/8.0/static/auth-methods.html>

also, in ref#1, it says to set

```
"host all 127.0.0.1/32 trust"
```

did not work, but i got it to work by:

```
"host all all 127.0.0.1/32 trust"
```

which more correctly match the pattern of the other lines in the file.

```
#####
```

```
rich@ubuntu:/$ sudo gedit /etc/postgresql/8.4/main/pg_hba.conf
```

```
rich@ubuntu:/$ sudo /etc/init.d/postgresql-8.4 restart
```

```
* Restarting PostgreSQL 8.4 database server
```

```
[ OK ]
```

```
rich@ubuntu:/$ psql -d expd -U expddba -W
```

```
Password for user expddba:
```

```
psql (8.4.2)
```

```
Type "help" for help.
```

```
expd=> help
```

```
You are using psql, the command-line interface to PostgreSQL.
```

```
Type: \copyright for distribution terms
```

```
      \h for help with SQL commands
```

```
      ? for help with psql commands
```

```
      \g or terminate with semicolon to execute query
```

```
      \q to quit
```

```
expd=> \q
```

```
rich@ubuntu:/$
```

SQLAlchemy, GeoAlchemy, and psycopg2

This page last changed on Mar 01, 2010 by [rich](#).

If installing on ubuntu first step is to install the python-setuptools to get 'easy_install' (poseidon appears to have it by default out of the box) (ref:<http://www.devpicayune.com/entry/200608221749>)

```
sudo apt-get install python-setuptools
```

To install install [SQLAlchemy](#) on linux for use in Python:

```
sudo easy_install sqlalchemy
```

To install install [GeoAlchemy](#) on linux for use in Python:

```
sudo easy_install geoalchemy
```

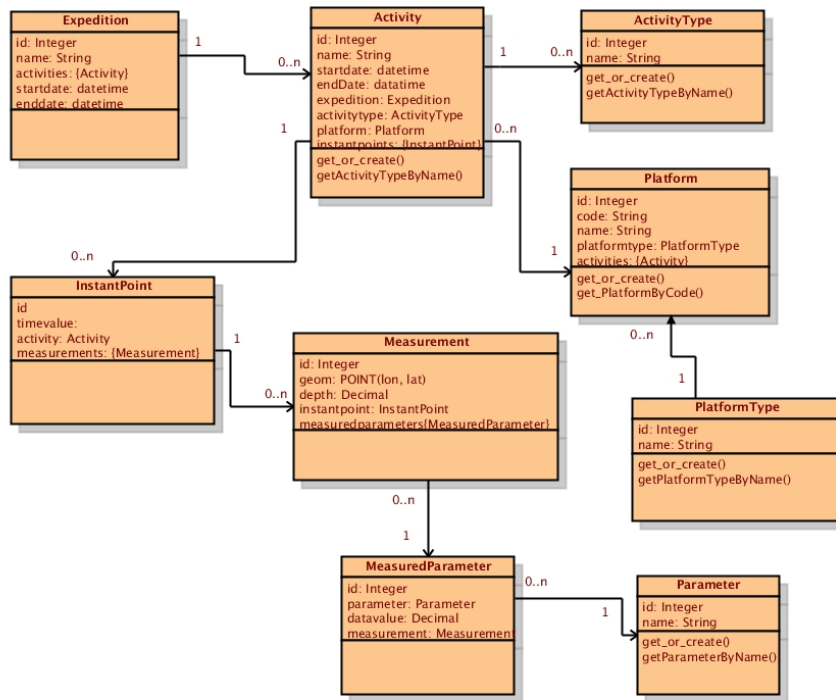
For Python to establish conenction with the PostgreSQL database you need this installed:

```
sudo apt-get install python-psycopg2
```


Stoqs OR Mapped Classes

This page last changed on Jun 15, 2010 by rich.

Stoqs OR Mapped Class Diagram



Upper Water Column Data holdings inventory

This page last changed on May 26, 2010 by [mccann](#).

How much data do we have?

Summary

Data Category	CDM dataType	Period of observation	Frequency of observation	Size on disk	# of records	# of parameters	Size of record
ROVCTD	Trajectory	1988 to 2010	15-second	0.40 GB	3 Million time records	16	130 bytes ⁴
Mooring ¹	Station& StationProfile	1989 to 2010	10-minutes and hourly	1.03 GB	4.5 Million time records	20	80 to 4800 bytes ²
AUVCTD	Trajectory	2003 to 2009 (269 surveys)	2-second gridded	0.45 GB	5.6 Million time records	10	80 bytes ³
UCTD	Trajectory						
PCTD	Profile						
BCTD	Profile	1988 to 2010	Nominally fortnightly				

Bottom line: We have just about 2 GB, certainly less than 10 GB. What we are dealing with can fit on a modern laptop.

Notes:

1. See Mooring Data Details table below
2. Record size for mooring data depends on number of profile depths in the record, up to 60 levels for ADCP data
3. Double precision (8-bytes) used for AUVCTD NetCDF data

Mooring Data Details - from THREDDS Data Server (TDS) aggregation of QC'ed data from the netCDF data archive

Mooring	Type	Date Range ³	THREDDS Data Server URL (DODS URL)	#Time	#Depth	#Variables ¹	#MB ⁴
m0	adcp	03-JUN-2004 18:00 to 21-JUN-2007 17:09	http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M0_QC_adcp.html	147246	15	8	70.8
	metsys			144326	1	7	4.0

		03-JUN-2004 17:50 to 20-JUN-2007 23:50	http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M0_QC_metsys.html				
	tstring		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M0_QC_tstring.html	154196	5	4	3.1
m1	adcp	14-JAN-1992 00:30 to 20-OCT-2009 19:00	http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M1_QC_adcp.html	283478	37	8	336.0
	metsys		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M1_QC_metsys.html	885116	1	7	24.8
	tstring		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M1_QC_tstring.html	865193	11	4	152.4
m2	adcp		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M2_QC_adcp.html	108819	60	8	208.8
	metsys		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M2_QC_metsys.html	782999	1	7	22.0
	tstring		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M2_QC_tstring.html	836942	11	4	147.2
M3	adcp	24-MAR-1998 12:00 to 19-DEC-1999 04:00	http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M3_QC_adcp.html	22490	30	8	21.6

	metsys		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M3_QC_metsys.html	89181	1	7	2.4
	tstring		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M3_QC_tstring.html	102748	11	4	18.0
m4	adcp	25-AUG-1999 01:00 to 21-MAY-2000 12:00	http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M4_QC_adcp.html	6469	30	8	6.4
	metsys	24-AUG-1999 21:51 to 07-SEP-2000 15:39	http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M4_QC_metsys.html	53359	1	7	1.6
	tstring		http://elvis.shore.mbari.org:8080/thredds/dodsC/moorings/M4_QC_tstring.html	54704	12	4	10.4
Totals				4537266			1029.5

Notes:

1. Flag variables in the netCDF files are not counted
2. Foreign keys in the database tables are not counted
3. A Date Range is show if Ferret is able to show it (many of these data sets exceed Ferret's 750,000 record limit on irregular axis data)
4. Single precision (4-byte) floating point values for variable storage

Postgres Notes - first try

This page last changed on Feb 22, 2010 by rich.

Usefull web site: <https://help.ubuntu.com/community/PostgreSql>

1) assumed postgres installed.....try to see if pgadmin3 is too....

```
rich@rich-desktop:/$ pgadmin3
```

The program 'pgadmin3' is currently not installed. You can install it by typing:

```
sudo apt-get install pgadmin3
```

```
bash: pgadmin3: command not found
```

2) installed pgadmin3

```
rich@rich-desktop:/$ sudo apt-get install pgadmin3
```

```
[sudo] password for rich:
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

The following packages were automatically installed and are no longer required:

```
linux-backports-modules-2.6.24-21-generic libdns35 octave3.0-htmldoc
```

Use 'apt-get autoremove' to remove them.

The following extra packages will be installed:

```
pgadmin3-data
```

Recommended packages:

```
pgagent
```

The following NEW packages will be installed:

```
pgadmin3 pgadmin3-data
```

0 upgraded, 2 newly installed, 0 to remove and 69 not upgraded.

Need to get 4206kB of archives.

After this operation, 12.2MB of additional disk space will be used.

Do you want to continue [Y/n]? Y

```
Get:1 http://de.archive.ubuntu.com hardy/universe pgadmin3-data 1.8.2-0ubuntu1 [1070kB]
```

```
Get:2 http://de.archive.ubuntu.com hardy/universe pgadmin3 1.8.2-0ubuntu1 [3136kB]
```

```
Fetch: 4206kB in 6s (680kB/s)
```

Selecting previously deselected package pgadmin3-data.

```
(Reading database ...
```

```
dpkg: serious warning: files list file for package `gprolog' missing, assuming package has no files currently installed.
```

```
229768 files and directories currently installed.)
```

```
Unpacking pgadmin3-data (from .../pgadmin3-data_1.8.2-0ubuntu1_all.deb) ...
```

Selecting previously deselected package pgadmin3.

```
Unpacking pgadmin3 (from .../pgadmin3_1.8.2-0ubuntu1_i386.deb) ...
```

```
Setting up pgadmin3-data (1.8.2-0ubuntu1) ...
```

```
Setting up pgadmin3 (1.8.2-0ubuntu1) ...
```

3) run pgadmin3 ... showed no databases....and threw multiple gtkPizza messages.....

```
rich@rich-desktop:/$ pgadmin3
```

```
(pgadmin3:7980): Gtk-WARNING **: /build/builddd/gtk+2.0-2.12.9/gtk/gtkwidget.c:8547: widget class 'GtkPizza' has no property named 'row-ending-details'
```

4) change password

```
rich@rich-desktop:/$ sudo -u postgres psql postgres
```

Welcome to psql 8.3.9, the PostgreSQL interactive terminal.

```
Type: \copyright for distribution terms
```

```
\h for help with SQL commands
```

```
? for help with psql commands
```

```
\g or terminate with semicolon to execute query
```

```
\q to quit
```

```
postgres=# \password postgres
```

```
Enter new password:
```

```
Enter it again:
```

```
postgres=# quit
postgres-# exit
postgres-#
postgres=# sudo -u postgres createdb mydb
postgres-# \q
rich@rich-desktop:/$
```

5) created a database.....

```
rich@rich-desktop:/$ sudo -u postgres createdb mydb
rich@rich-desktop:/$
```

6) re-ran pgadmin3,

```
rich@rich-desktop:/$ pgadmin3
```

in gui, I "added server": File > Add Server and entered "localhost"

back in another terminal.....

7) create an expdba user and expd database:

```
rich@rich-desktop:/$
rich@rich-desktop:~/Desktop$ sudo -u postgres createuser -D -A -P expddba
Enter password for new role:
Enter it again:
Shall the new role be allowed to create more new roles? (y/n) y
```

8) create an expd database:

```
rich@rich-desktop:~/Desktop$ sudo -u postgres createdb -O expddba expd
rich@rich-desktop:~/Desktop$
```