

Space Details

Key:	Rich
Name:	Rich
Description:	
Creator (Creation Date):	rich (Jul 11, 2008)
Last Modifier (Mod. Date):	rich (Jul 11, 2008)

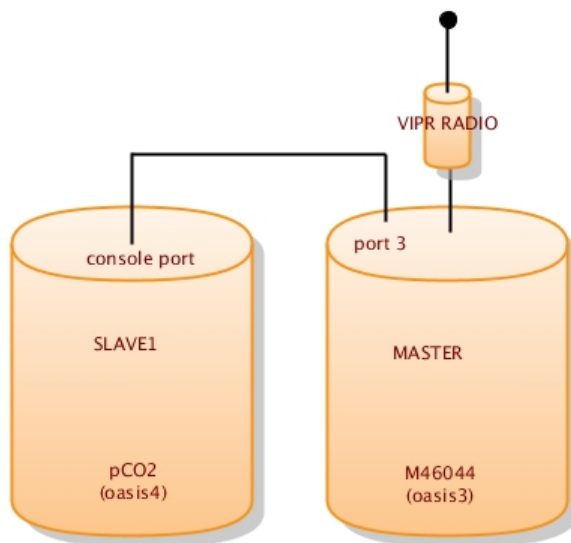
Available Pages

- Home 
 - Data Logging
 - Data Logging for non-ROV cruises
 - Dataprobe
 - DocRicketts Pressure,Depth and Digiquartz
 - Gliffy test
 - GPS Logging Failure of March,2011
 - legacy rovctd der file loading
 - MARS Data Management Research & Requirements
 - Logging
 - MARS SSDS Devices
 - MARS Engineering Data Management
 - MARS Datamanagement Overview
 - mars2ssds
 - Qmonitor classes
 - Navproc
 - new oasis downloads
 - Oasis SSDS Dods Data
 - OasisRB
 - GetOasis
 - Old Oasis Downloads
 - PanTilt and Lapbox
 - pHProbe Workbench v2.x
 - pHProbeWorkbench
 - Poseidon VM Install Notes
 - postgres
 - Python
 - resume 18-Jan-13 pre-django
 - ROV Pilots Dive Log
 - rovctd plots
 - tomcat servlet container
 - database setup
 - Tomcat 5.5 server.xml on anvil
 - web.xml on anvil

- Updating ROVCTD Plots using JFreeChart and Tomcat
 - ROVCTD PLOTTING UPDATE 01-Jun-2011
- Vorne Display Panel - Rackmount Drawing
- WesternFlyer DocRicketts
 - Computers
- XML-RPC Notes
- DATABASE
- pHProbeWorkbench Model Classes

Home

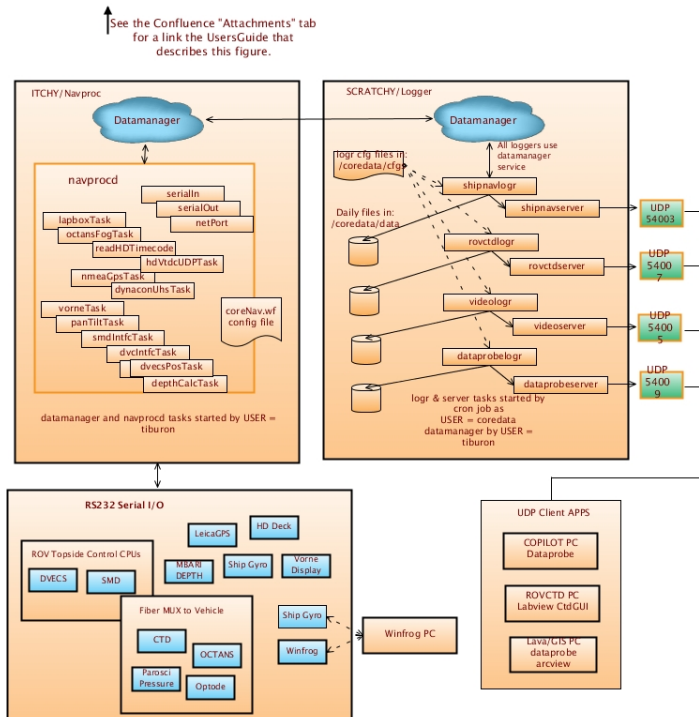
This page last changed on Aug 03, 2012 by [rich](#).



This is the home page for the Rich space.

Data Logging

This page last changed on Jan 23, 2012 by rich.



Data Logging for non-ROV cruises

This page last changed on Feb 02, 2012 by rich.

REFER TO ATTACHED WORD DOC FOR COMPLETE DESCRIPTION

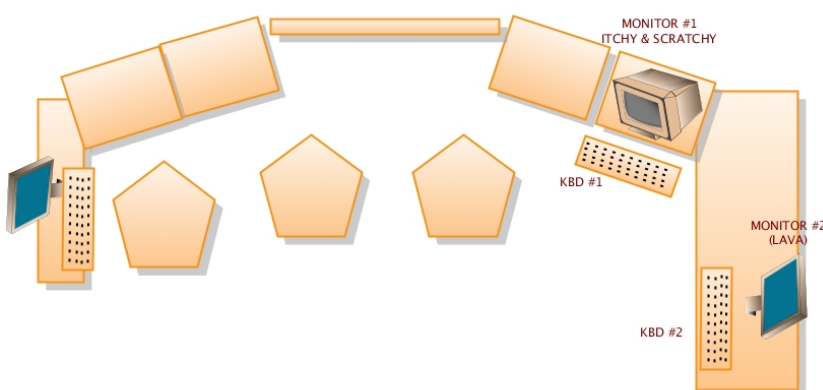
Stopping and Starting Datamanager and Navproc

Linux workstations itchy & scratchy are located below Keyboard #1. Both must be powered ON. There is a third linux box below them but its for backup and stays OFF.

Turn on Monitor #1. You will be directed to switch the Kbd/Monitor between itchy & scratchy a couple of times. To do that, hit scroll-lock key twice to bring up KVM switch software. Then scroll down to ' itchy' or 'scratchy' and select as directed.

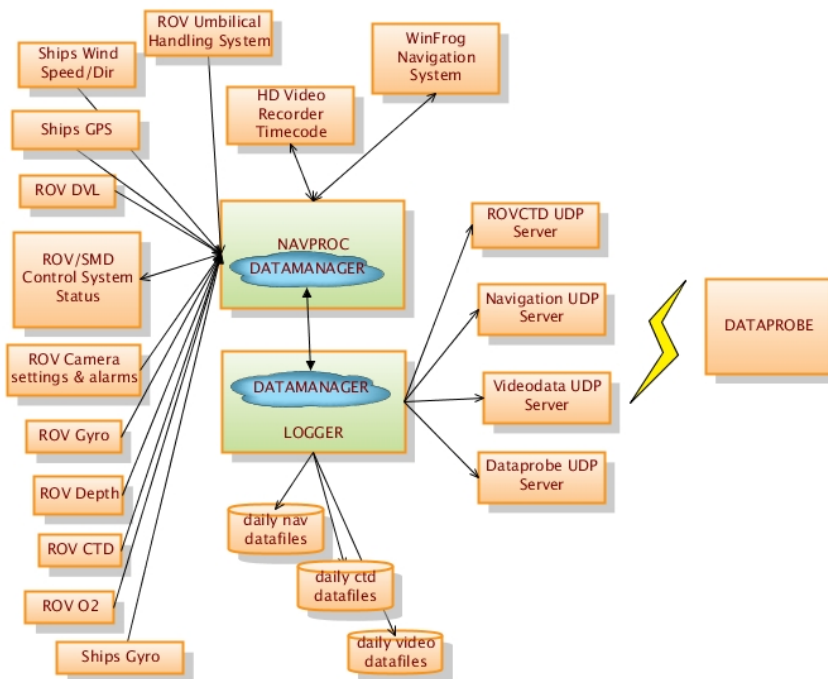
1. Select 'scratchy' on the KVM, select it and hit <cr> a couple of times to get its attention.
2. the login path diverges here depending on how the last user left it.
 - a. if you are presented a command-line login prompt then log in as user = tiburon (pilots or Dan C have pswd)
 - b. if previous user had left it logged in at a cmd line prompt, proceed to step '3'
 - c. If its it was left logged in at the 'modern' 'Graphical UI (red hats red background etc.) then use mouse to do a System>Log Off. And let it close the GUI and revert to the command-line interface. Then go back to (a) above and re-login as tiburon
3. Type dk<cr> to kill any previously running datamanager.
4. Select 'itchy' on the KVM, select it and hit <cr> a couple of times to get its attention.
 - a. Log in as tiburon (same as steps 2a-c above)
 - b. Type nk<cr> then <dk<cr> to kill any previously running navproc or datamanager
 - c. type 'ds' to restart datamanager (may take about 10 seconds to boot - and tell you it couldn't find a boot server, that's OK)
 - d. type 'ns' to restart navproc
5. Re-select 'scratchy' on the KVM
 - a. type ds<cr> to start datamanager (should start almost immediately)
 - b. (note there is no navproc to start on scratchy)

Data logging should start within and Dataprobe over on LAVA should go green two minutes. (It takes about two minutes for the watchdog tasks on scratchy to autostart the data logging. IF NOT, more serious problems are at play. There is a WesternFlyer/DocRicketts Data Logging Users Guide in a white binder above 'LAVA' you will need to start troubleshooting there.



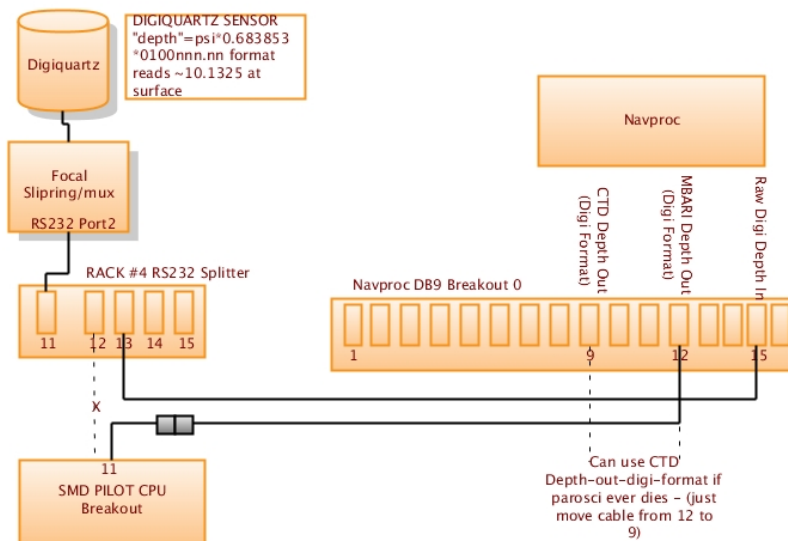
Dataprobe

This page last changed on Mar 30, 2020 by rich.



DocRicketts Pressure,Depth and Digiquartz

This page last changed on Jan 26, 2012 by rich.



As Recieved From SMD

SMD control software with requires a depth input in meters for pilot GUI, nav display etc.

ROV has a Paroscientifics Digiquartz (Model: 8B4000-I s/n 107921) xducer.

ROV Digiquartz configured by SMD to provide depth in meters (nominal/uncorrected)

SMD software allows for pilot input of an offset to account for surface diff (atmospheric plus slight calibration trim) (That offset is sent down to the sensor on startup).

SMD software has no provision to tell us what the offset is for post-processing.

SMD contract specified "pressure" be provided in the 'DVECS' serial string to navproc, however as built was digiquartz depth in meters plus SMD applied pilot-provided depth offset.

SMD contact specified MBARI was to supply our computed rov depth to SMD software via the \$NVP message string for video overlay etc.

Our requirements are for:

- precise pressure>depth conversion (using ships latitude)
- any offsets are known and stable
- nominal 'depth=zero' at surface (psig)
- single source - ie everyone looks at, logs and displays the same value

What we did...

We intercept the data coming from the Digiquartz and run it over to navproc, where a task parses it, converts to pressure, computes depth using ships latitude, applies offsets and then outputs back out a serial port for injection to the SMD system (Refer to FIgure). This also required reprogramming the

Digiquartz to continuously output rs232 on powerup, output (nominal meters of seawater), and no zero-offset tare.

Additionally, we also provide a depth output from the Seabird CTD in Digiquartz format in case the sensor ever dies in the field....

The post-rov-delivery changes to the system to meet MBARI requirements means both the MBARI and SMD software "design" is not clean. There is some data duplication and some seemingly circular data pathways - hopefully they are correctly captured below... although about two years after-the-fact.

Programming the Digiquartz

First step was to re-program the Digiquartz to output Depth in meters of seawater (nominal, uncorrected).

Refer to manual and notes below..... (see Confluence Attachments to this page).

Direct connect serially

My notes are sketchy here... refer to digiquartz manual....(Model: 8B4000-I s/n 107921)

Note cmd format is *ddsscc dd=destination(the sensor) ss=source(you, the terminal or PC) cc=cmd. So commands from terminal>digi begin *0100 and data from digi>terminal should begin *0001...

I think below was investigating how it was shipped by SMD:

THIS IS APPROXIMATE... NEEDS VERIFICATION.

```
*0100 MD <cr><lf> MD=2
*0100 PR <cr><lf> PR=238
*0100 MD=0
      UN=8 (metersH2O (freshwater))
      UF=1.00000
      ZS = 0
      ZV = 0
      ZL = 0
```

I think below is how I commanded it - possibly each cmd was preceded by the *0100EW (enable eeprom write cmd) See manual attached...

THIS NEEDS VERIFICATION.

Note *0100EW = (Enable EEPROM write for one command.)

```
*0100EW*0100ZL = 1 <cr><lf> (zero tare locked)
*0100EW*0100UN = 0 <cr><lf> (user defined units based on UF multiplier)
*0100EW*0100UF = 0.683853 <cr><lf> (psi to meters of seawater manual pg 6-9)
*0100EW*0100MD = 2 <cr><lf> (display off, continuous rs232 on)
```

(also assumes PR=238 ZS=0 ZV=0 ZL=0 from prior setup)

The Navproc Code on Itchy

Navproc runs a serialIn task to ingest the Digiquartz message string and publishes it to datamanager as DIGIQ.MSG.STRING (see below, extract from /usr/tiburon2/bin/navproc/coreNav.wf on itchy.)

```
serialPort 0,15 9600 8 1 None CR/LF # ROV Digiquartz Pressure In
```

...	serialIn	ROV.DIGIQ.STRING	CHAR[64] "%s\n"	No	0,15	None
-----	----------	------------------	-----------------	----	------	------

Navproc runs depthCalcTask() (see: /usr/tiburon2/src/navproc/calculateDepth.c)

- consumes ROV.DIGIQ.STRING
- backs out the nominal psi-to-meters correction made in the xducer. ($\text{psi} = \text{digiqDepth} / 0.683853$)
- converts psi to decibars. ($\text{psi} * 0.6894757$)
- removes observed surface pressure offset (10.1325) and publishes as ROV.DIGIQ.SCALED_PRESSURE
- computes depth in meters using standard algorithms and ship's latitude and publishes as ROV.MBARI.DEPTH
- computes depth change rate and publishes as ROV.MBARI.DEPTH_RATE
- formats and outputs ROV.MBARI.DEPTH in Digiquartz format on serial port (back to the SMD system)

What Gets Used

1) Core data logging by shipnavlogr on scratchy logs the following Pressure and Depth items derived from the Digiquartz

ROV.DIGIQ.SCALED_PRESSURE
ROV.MBARI.DEPTH

2) SMD consumes the digiq formatted value of ROV.MBARI.DEPTH as if it had come from the sensor

3) SMD therefore re-outputs the depth via the original DVECS output message string, which ends up back in navproc as ROV.DVECS.DEPTH

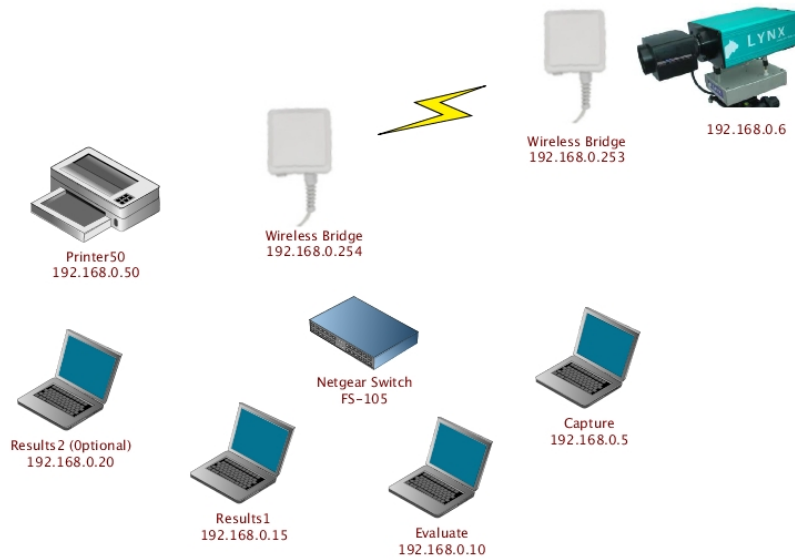
4) Navproc dvcsIntfcTask outputs ROV.MBARI.DEPTH within the NVP string as originally agreed.

5) Navproc overlay.c task consumes ROV.MBARI.DEPTH_RATE , ROV.MBARI.DEPTH, ROV.DVECS.DEPTH (and lots of other items) and outputs on a serial line to the vehicle overlay computer.

6) ROV.MBARI.DEPTH is consumed by the navproc vorneTask and sent serially to the Vorne Display in the control room.

Gliffy test

This page last changed on Jul 24, 2012 by [rich](#).



This is a test

End of test

GPS Logging Failure of March,2011

This page last changed on Feb 02, 2012 by [rich](#).

Ships GPS Lat/Lon were not updating on the WF datamanagement system in the control room. Serial Datastream was making it to the navprocessor (itchy) from the Leica GPS unit. The problem is that the system is coded to look for \$GPGGA messages. (We need \$GGA for datafields like DifferentialMode, HDOP etc). The only data coming down the line were \$ZDA and \$GLL messages, even though \$GGA was enabled on the Leica menu (for serial port 9). Dan Chamberlain tried cycling on/off the messages and discovered that turning off \$DTM messages also turned off \$GGA messages even though the menu reported \$GGA was enabled. (Seems to me to indicate a bug in the Leica code)

Anyway, next time.... Remember that if GPS ever dies in coredata logging, as indicated by dataprobe. This may be the issue.

legacy rovctd der file loading

This page last changed on Oct 05, 2009 by [rich](#).

First: truncate table ctdder

Then, working in small batches within a single year

Open the dts job ROVCtdDerLoader and set the connection1 property to point to the der file you want to load and then run it

Repeat as needed...

Then fix the target table below in two places: the year, and the platform id (989 is tibr, 988 is vnta)

```
INSERT INTO RovctdBinData_1997
```

```
(DatetimeGMT,  
    DeviceID,  
    EpochSecs,  
    LoadID_FK,  
    bin,  
    p,  
    t,  
    c,  
    s,  
    o2,  
    light,  
    oc,  
    ot,  
    trans,  
    rlat,  
    rlon,  
    ptsFlag,  
    o2Flag,  
    lightFlag,  
    latlonFlag,  
    ProcessingID_FK)
```

```
SELECT  
    [dbo].EpochSecondsToDTG(usec) as DatetimeGMT,  
    989,  
    usec,  
    -1997,
```

```
    usec,  
    pres,  
    [temp],  
    cond,  
    dbo.Salinity(cond,temp,pres),  
    oxyg,  
    light,  
    o2curr,  
    o2temp,  
    transvolt,  
    lat,  
    lon,  
    2,  
    2,  
    2,  
    2,  
    0  
FROM      ctdder  
go  
---truncate table ctdder  
---go
```

MARS Data Management Research & Requirements

This page last changed on Sep 27, 2011 by rich.

Log MARS engineering data, publish logged data to SSDS

The physical systems which produce MARS engineering data can be thought of as being comprised of four major components: the shore-side high voltage power controller (PSC), the deployed Medium Voltage Converter (MVC), the deployed Low-Voltage Converter (LVC) and a Onewire HA7 temperature sensor network (which was added within the LVC housing to monitor temperatures).

The sensors and loads of the MVC and LVC are controlled and accessed via the Node Power Controller (NPC) which is at the heart of the wet-side system. The NPC, Onewire and LVC systems are physically contained in the LVC Housing. The MVC is a separate physical housing. As of 2/2011 its sensors are read by the a/d converters of the NPC (a 2nd generation MVC will have its own a/d and controller and be read directly).

The 'PMACS Console' provides the operator user-interface to the 'PMACS Server'. All communications with the NPC, PSC and HA7 systems are through the shore-side 'PMACS Server'

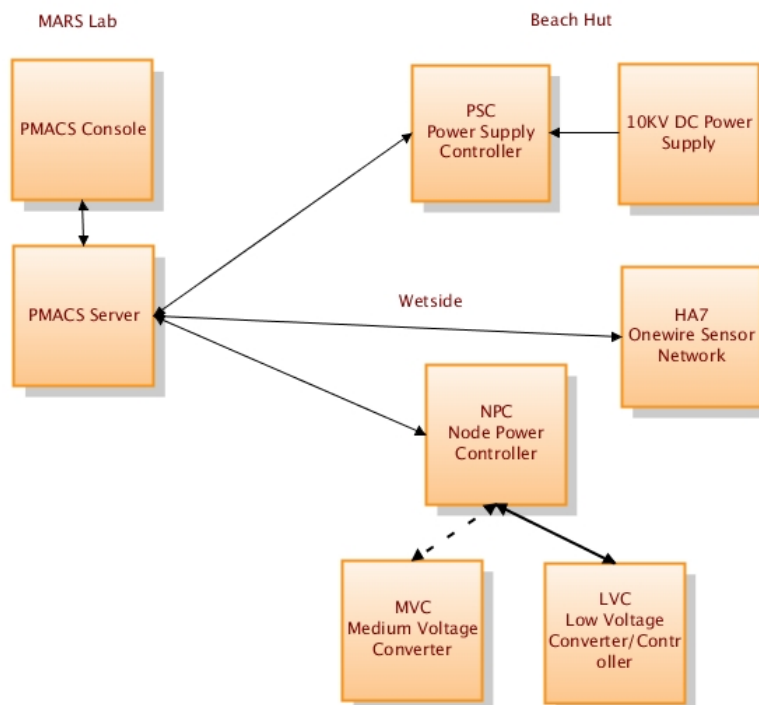


Fig.1 A 'logical' view of the MARS system from a data management perspective.

Besides being the communications gateway to the deployed hardware the PMACS Server polls the NPC subsystems for all of the LVC and MVC engineering data at fixed intervals and logs it to various daily text files. Raw NPC text files are archived to directories accessible via an apache web server. PSC data logging service will need to be created. The PMACS Server also runs an incomplete prototype logger service for the Onewire sensor network, which will need to be reworked.

Peering Deeper into PMACS Server

'PMACS Server' is the both the name given to a hardware platform and to a collection programs written in various languages running there. A great deal of research has gone to understanding how all of the

pieces of the system interact. For our data management purposes, the relevant functionality 'PMACS Server' provides is reduced to:

- Provides the gateway for communications with the PSC and NPC.
- Communication down the cable is via XMLRPC protocol to servers running at those two targets.
- Acts a bridge between XMLRPC and clients (primarily the PMACS Console) via a SOAP service.
- Nothing really prevents clients from communicating via XML-RPC directly however it is probably not wise for the NPC due to loading. In fact, XMLRPC was used directly in my initial data logger for the PSC.
- Periodically polls the NPC for all a/d channels and i/o-bit states and logs as raw counts to daily archive of 'csv' files
- Provides a SOAP service to access the Onewire sensor network and the PSC.
- Hosts an Apache Web Server. Important elements to our understanding are:
 - Exposes the archive of the NPC daily csv files
 - Contains an important xml file that explains mapping of all of the fields of the NPC csv files,(including calibration coefficients).
 - Contains two important 'WSDL' files which enable clients to use the SOAP services to access NPC, PSC and the Onewire server.
 - Hosts the mars wiki.
- The NPC related SOAP Services provide:
 - Mapping of named fields to a/d channels and io port bits and bridge between SOAP and XMLRPC protocols
 - Calibrated engineering values from the NPC a/d data to the PMACS Console (really to any SOAP client)
 - Switching of I/O Ports at the request of PMAC Console to perform the ground fault testing.
 - Switching on/off of both internal loads and external loads etc.

NPC Logging

NPC is a PC104 that resides in LVC. Implements an XMLRPC interface that the PMACS Server wraps with a SOAP server. It controls power through 4 digital IO plus with A/D converter boards. PMACS Server polls the NPC for a full data record that gives the raw counts of each A/D channel and the bit state of each IO port. The record contains info about/from all Internal Loads (cisco routers etc.), External Loads (ports 1-8, 48v and 375v bus), ground fault, MV sensors etc.

Important points:

- Accessed via a SOAP Server or XMLRPC
- Service described by .wsdl file on web server.
- Logger is well established and stable.
- Daily files in web accessible archive location
- Record is raw counts and each value needs to be scaled using calibrations.
- MARS Operators want External Loads 1...8, Groundfaults, Internal Loads broken out as 'logical devices'
- Its not necessary to push NPC raw data into ssds.

PSC Logging

PSC is a PC104 that resides in the beach-house with the 100KV DC supply. Implements an XMLRPC interface that the PMACS Server wraps with SOAP. Initial console application/logger I wrote in early 2009 needs to be turned into a true logger.

Important points:

- Accessed via a SOAP Server or XMLRPC
- Service described by .wsdl file on web server.
- My prototype logger needs following improvements:
 - An errorlog and general hardening
 - Needs to write files in a commonly accessible archive location (exposed thru firewall)
 - Convert from XMLRPC to SOAP protocol (would be very similar to needs of Onewire described later).

Groundfault

No one could really explain how the ground fault detection works. Archeology has now yielded:

- the NPC & PMACS-Server are logging state of all 4 a/d boards 1x per sec.
- the PMACS-Console (vb.net) requests the PMACS-Server (via SOAP) to make a gf() test of a given "name". (ie. 'v400' 'v400r' 'v48' 'v48r' or 'test')
- PMACS-Server maps test names to a digital i/o board, port, and bit and sends a set cmd to the NPC via XMLRPC. (that actually sets the io port bit to 0) that controls which FET gets switched in
- Groundfault test measurements are returned in a single a/d channel of the 1hz NPC raw data stream.
- the PMACS-Server waits ~6 secs, then parses the next NPC record, waits 4 more seconds, clears the io bit and then returns the value scaled to microamps.
- PMACS Console updated GUI and requests the next test in sequence.?(note test='test' never seems to run... I understand that to be a manual process to be done by operator if needed, after switching off the console sequence of tests)..?

Note there appears to be an error on the console in either the test 'A,B,C,D' labeling or with the 375v, 375com, 48v,48com text and boxes.

MARS Operators want all four groundfaults to be broken out of the NPC record to a single timestamped record and treated as a single 'device' so that values can be plotted against a common time axis.

INTERNAL LOADS

Internal loads consist of the LVC internal devices such as the Cisco routers, the HA7 controller, fiber muxes etc that can all be switch on/off via the NPC. Also includes the 'bus monitors' ie. the common bus voltage measurements.

MARS Operators want all internal loads to be broken out of the NPC record to a single timestamped record and treated as a single 'device' so that values can be plotted against a common time axis.

Data need to be scaled in engineering units with individual calibration applied.

EXTERNAL LOADS

External loads are the 8 'science ports'. They are basically switchable on/off 48v and 375v power supplies with associated current sensors and circuit breakers.

MARS Operators want each external load to be broken out of the NPC record to timestamped records and treated as a standalone 'device' so that values can be plotted against a common time axis.

Data need to be scaled in engineering units with individual calibration applied

Onewire Network

The Onewire (or HA7) Network is an array of temperature sensors and a data acquisition box which was added to the LVC housing. Onewire sensors replace the sensors placed strategically on internal devices such as routers, xternal load controllers etc. The original sensors were considered 'too noisy' to be of use.

Important points:

- Accessed via a SOAP Server (written in Ruby)
- Service described by .wsdl file on web server.
- Prototype logger that needs some improvements.
- Currently writes a single massive file (needs log rotation).
- Starts on boot; needs a provision to restart if crashed.
- No errorlog
- Needs to write files in a commonly accessible archive location (exposed thru firewall)

Additional Requirements

MARS software is complex and documentation is 'sparse'. MARS software is consists of many layers and was written in multiple languages. It is an operational system that needs to be kept running 24/7. Therefore a least disruptive path needs to be followed to minimize risk to ongoing science.

Least disruptive to me means:

- Resist modifying the existing applications code as much as possible.
- Avoid adding additional languages and packages.
- Avoid updating existing packages, libraries, servers compilers etc (SOAP, Apache, Python, Ruby VB.Net)

- Avoid increasing number and complexity of processing tasks as much as possible.

Ideas

Much of the data and the information needed to interpret it already are exposed on the pmacs web site. (Though this will need to be secured behind the MARS firewall in the near future).

Although not well documented, a pattern has emerged for how to access and log the data for the missing data pieces via the PMACS and Onewire servers using SOAP calls.

Python is a good choice for it's 'impedance match' to the existing PMACS Server.

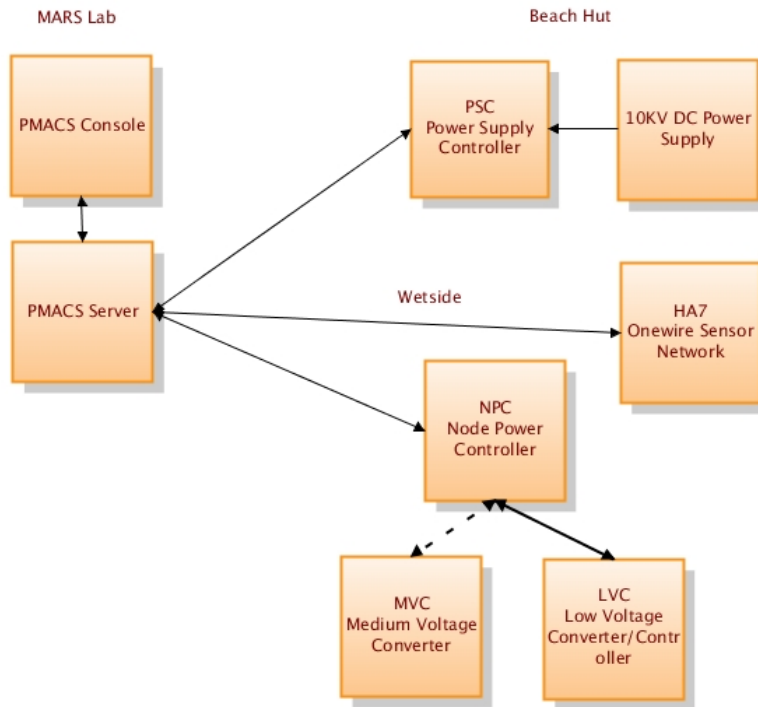
If all data were logged to the archives on pmacs webserver, then the processes needed to ingest that data to ssds could reside on ssds servers on the other side of the MARS firewall and thus minimally disturb MARS.

Ingesting tasks could then pull data from the web as it comes in and use whatever SSDS messaging protocols were needed.

Additional packages, libraries and software layer versions etc need for ingest are kept away from the MARS environment.

Logging

This page last changed on Feb 15, 2011 by [rich](#).



MARS SSDS Devices

This page last changed on Mar 16, 2011 by [rich](#).

Device Name = MARS Device Description = MARS Main Hub

Device ID = 1785 Device UUID = 9a5310bd-4442-11e0-bf45-bd9bff065a7d

Device Name = MARS LOAD1

Device ID = 1772 Device UUID = 6b2e4697-352d-11e0-8cb9-cb191be4ef2b

Device Name = MARS LOAD2

Device ID = 1773 Device UUID = b64dfe9f-352d-11e0-8cb9-cb191be4ef2b

Device Name = MARS LOAD3

Device ID = 1774 Device UUID = cbf3ae50-352d-11e0-8cb9-cb191be4ef2b

Device Name = MARS LOAD4

Device ID = 1775 Device UUID = f05e7f80-352d-11e0-8cb9-cb191be4ef2b

Device Name = MARS LOAD5

Device ID = 1776 Device UUID = fff3e816-352d-11e0-8cb9-cb191be4ef2b

Device Name = MARS LOAD6

Device ID = 1777 Device UUID = 112d3d37-352e-11e0-8cb9-cb191be4ef2b

Device Name = MARS LOAD7

Device ID = 1778 Device UUID = 1eb5e64c-352e-11e0-8cb9-cb191be4ef2b

Device Name = MARS LOAD8

Device ID = 1779 Device UUID = 2e012759-352e-11e0-8cb9-cb191be4ef2b

Device Name = MARS GROUNDFAULT

Device ID = 1780 Device UUID = 7da926f7-352e-11e0-8cb9-cb191be4ef2b

Device Name = MARS ONEWIRE HA7

Device ID = 1781 Device UUID = d6c69b66-352f-11e0-8cb9-cb191be4ef2b

Device Name = MARS PSC

Device ID = 1782 Device UUID = 04acca72-3530-11e0-8cb9-cb191be4ef2b

Device Name = MARS LVC

Device ID = 1783 Device UUID = 4c3bbb62-3530-11e0-8cb9-cb191be4ef2b

Device Name = MARS MVC

Device ID = 1784 Device UUID = 6608cd34-3530-11e0-8cb9-cb191be4ef2b

MARS Engineering Data Management

This page last changed on Nov 16, 2011 by [rich](#).

[Requirements](#)

MARS Datamanagement Overview

This page last changed on Jul 26, 2012 by rich.

Pismo as ssdsadmin

mars to ssds architecture has three major subsystems.

1 Basic data acquisition happens on the mars pmacs server

2 The mars2ssds "stuff" runs on pismo, basically there are three main tasks that scrape data via web requests from the pmacs server archives, break them into pseudo 'ssds devices' and publish to rabbitMQ as ssds protocol buffers. (psc2rabbit, onewire2rabbit, and npc2rabbit tasks in /opt/ssds/mars2ssds)

3 Qmonitor (/opt/ssds/qmonitor) run periodically by cron, (/u/ssds/crontab.pismo) looks at rabbitmq messaging server, whenever it detects an 'active' device queue with no consumer, it starts a consumer for it, that pulls protocol buffer packets and pushes them to sql server dione device base tables in SSDS_Data.

(Note: from home dir /u/ssdsadmin see special files crontab.pismo and .netrc Both mars2ssds and qmonitor projects can be checked out from svn repository in projects of those same names. The cronjob and .netrc are also tucked away for lack of a better place, in the qmonitor/bin dir, although they need to be deployed to /u/ssdsadmin on pismo.)

view on ssds <http://new-ssds.mbari.org:8080/ssds/siamRawDataStep1.jsp>

Mars devices are ssds device 1782 thru 1784

1772-1779 are MARS External Load 1-8

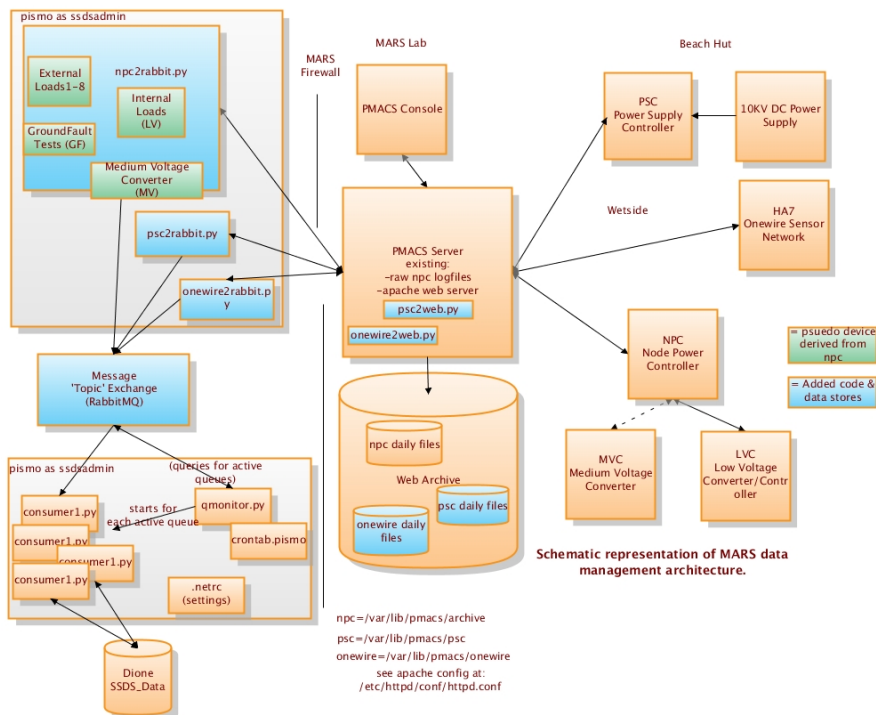
[1780 Groundfault](#)

[1781 Onewire \(HA7\)](#)

[1782 PSC](#)

[1783 LVC](#)

[1784 MVC](#)



On Jul 25, 2012, at 8:19 PM, Rich Schramm wrote:

three separate projects are checked in to svn: mars2ssds, qmonitor and mars2web
mars2web stuff goes to pmacs user acct on pmacs server. (see README.txt in the mars2web/bin dir)
*** everything else below refers to user ssdsadmin on pismo for the mars2ssds and qmonitor stuff

also the cronjob.pismo and the .netrc files are in svn in the qmonitor/bin directory

everything should restart on reboot or if the database is rebooted etc. - needs further testing to verify

run logs are in /opt/ssds/mars2ssds/log and /opt/ssds/qmonitor/log and are set to rotate.

TODO:

1)npc2rabbit has the external loads 1-8 turned off. We need to decide on in-row-storage and then run the bulk loads of legacy data (by running npc2server.py), then turn on the real-time loads (uncomment the lines in npc2rabbit.py)

Future enhancement:

1)Would be 'nice' if onewire/psc/npc2rabbit.py would query for the max(timestampseconds) for a device and pickup processing from there.... instead of attempting to reload all records from midnight of the day it is started. very minor issue as the insert just ignores records already loaded

This page last changed on Jul 24, 2012 by [rich](#).

Pismo as ssdsadmin

From home dir /u/ssdsadmin see special files crontab.pismo and .netrc

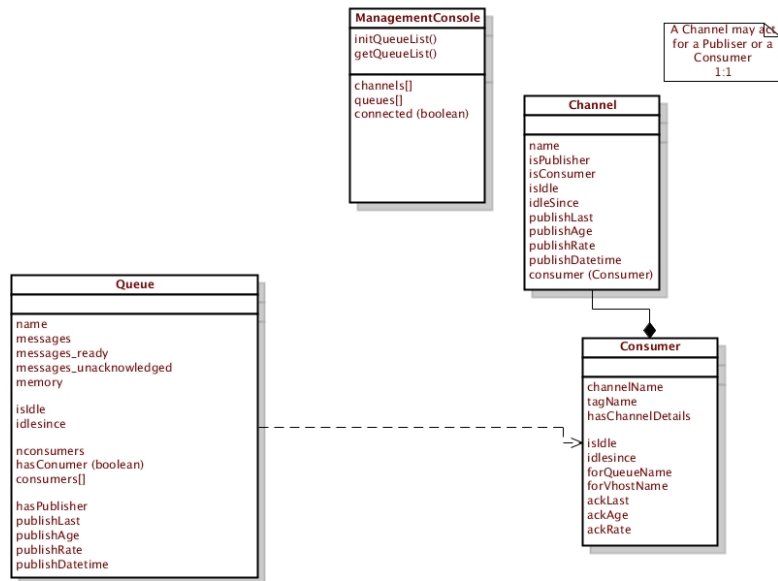
mars to ssds archetecure has three major subsystems.

1. Basic data acquisition happens on the mars pmacs server
2. The mars2ssds stuff runs on pismo, basically three main tasks that scrape data via web requests from the pmacs server archives and publish to rabbitMQ on our messaging server
3. Qmonitor looks at rabbitmq messaging server, whenever it detects an 'active' device queue with no consumer, it starts a consumer that pulls data packets and pushes them to sql server dione device base tables.

Mars devices are 1782 thru 1784

Qmonitor classes

This page last changed on Nov 16, 2011 by rich.

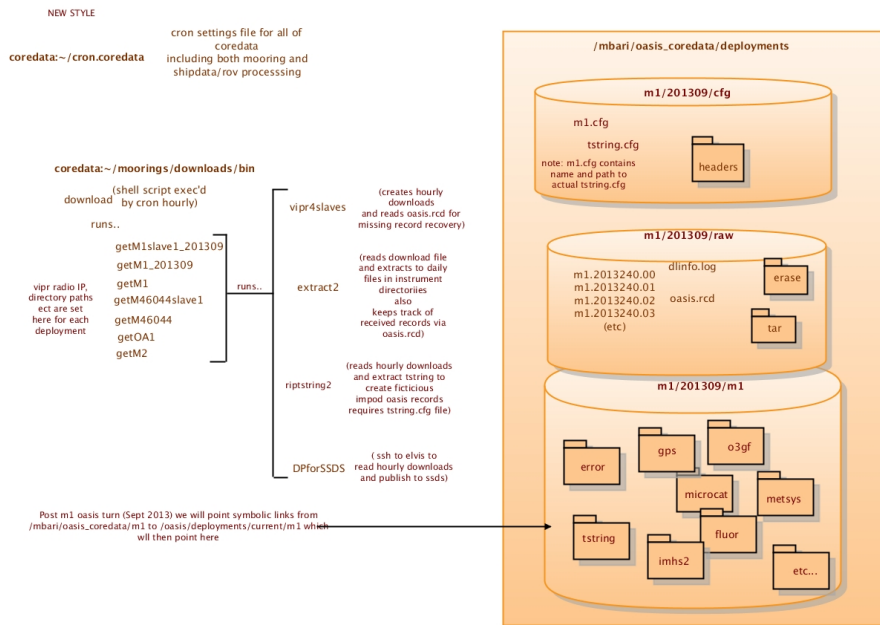


Navproc

This page last changed on Sep 08, 2016 by [rich](#).

new oasis downloads

This page last changed on Aug 28, 2013 by rich.



Oasis SSDS Dods Data

This page last changed on Aug 29, 2013 by [rich](#).

on elvis as ssdsadmin

files are symbolic links in `/var/www/dods_html/data`

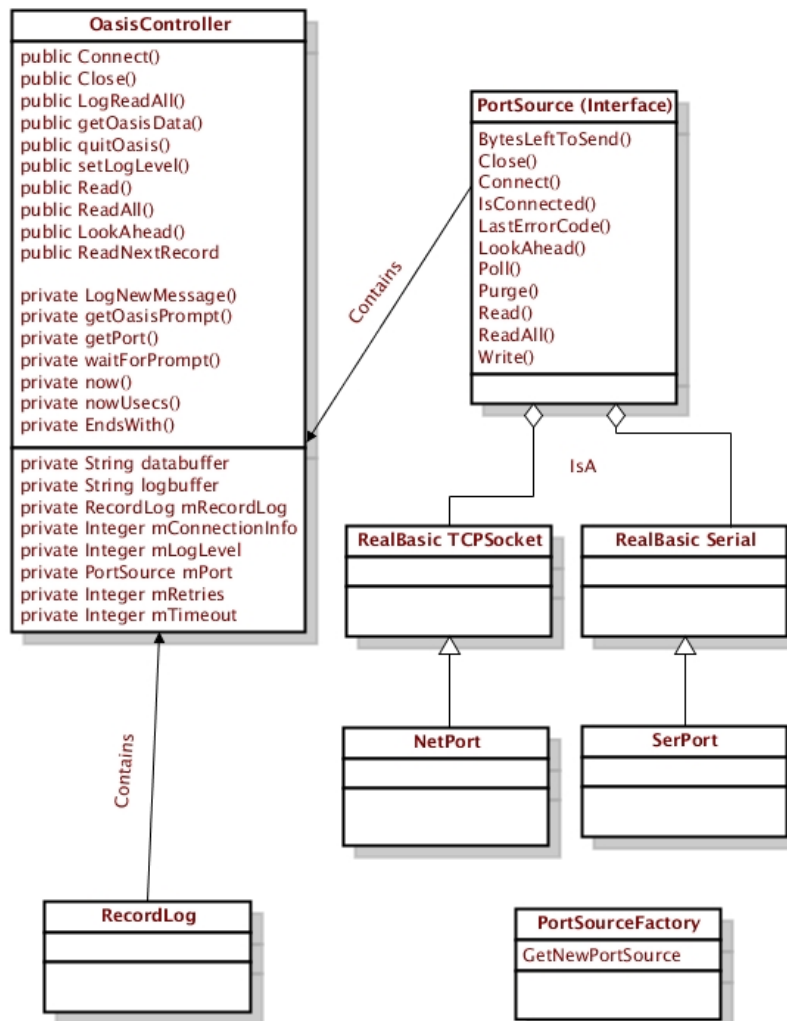
apache config is in `/etc/httpd/conf`

OasisRB

This page last changed on Sep 15, 2009 by [rich](#).

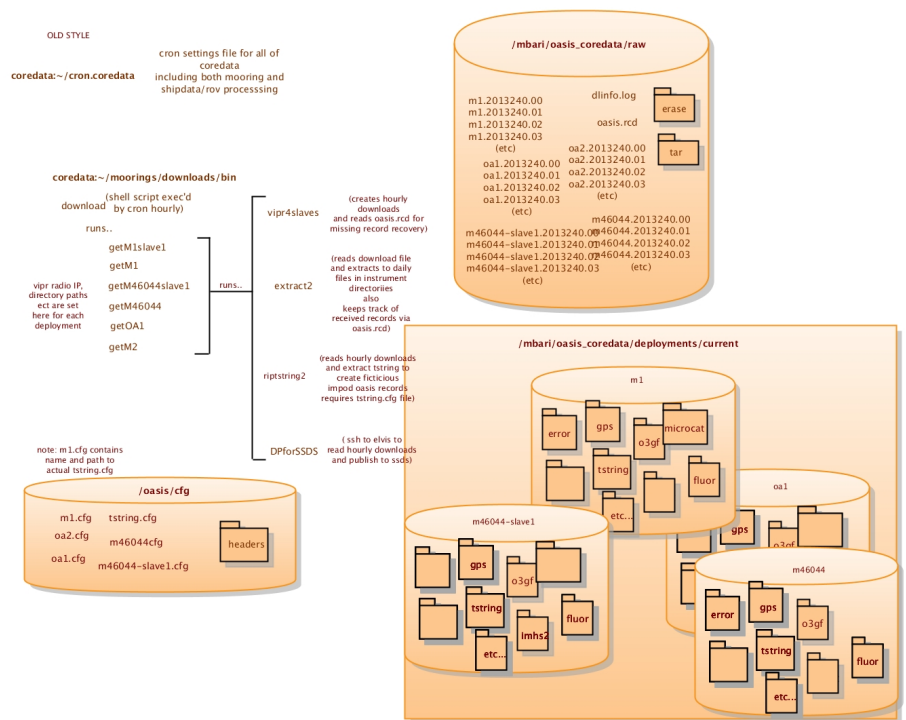
GetOasis

This page last changed on Sep 15, 2009 by rich.



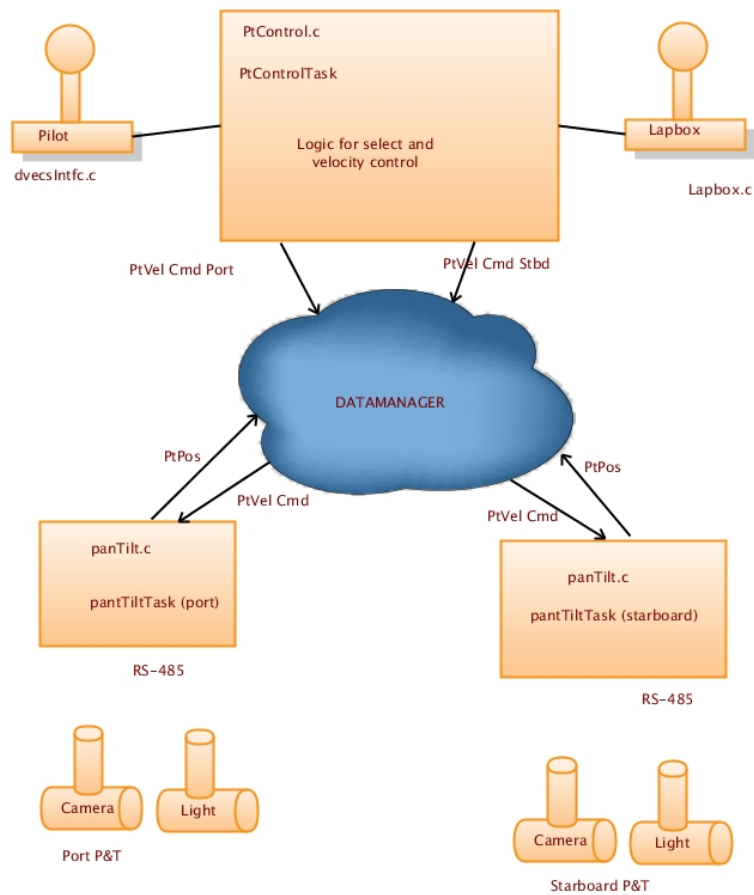
Old Oasis Downloads

This page last changed on Aug 29, 2013 by rich.



PanTilt and Lapbox

This page last changed on Jan 24, 2012 by rich.

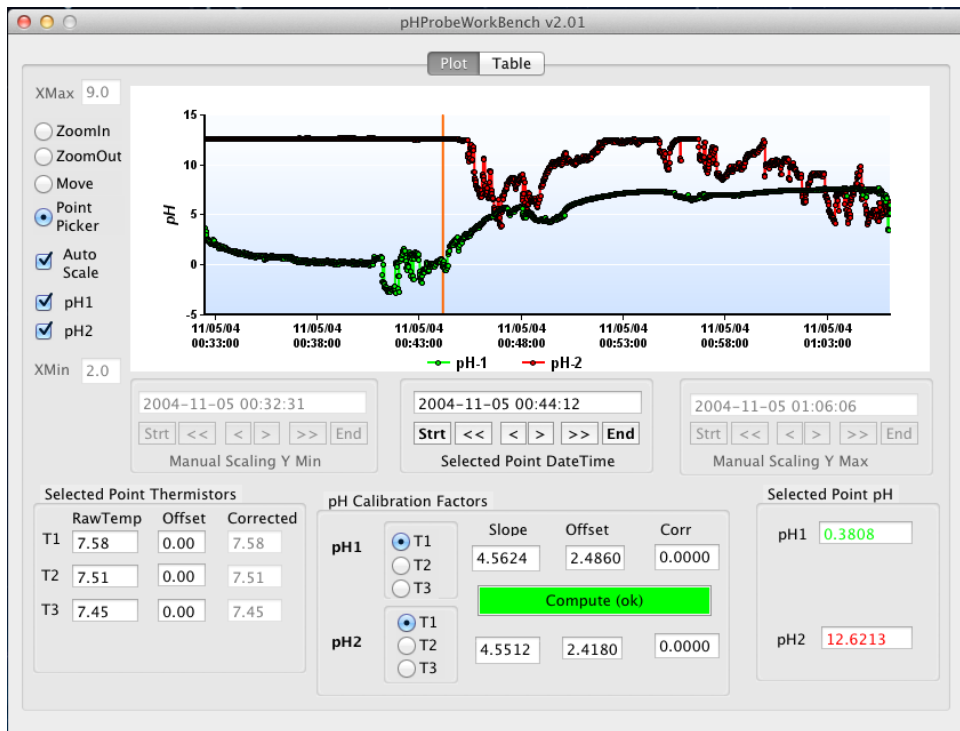


pHProbe Workbench v2.x

This page last changed on Sep 30, 2011 by [rich](#).

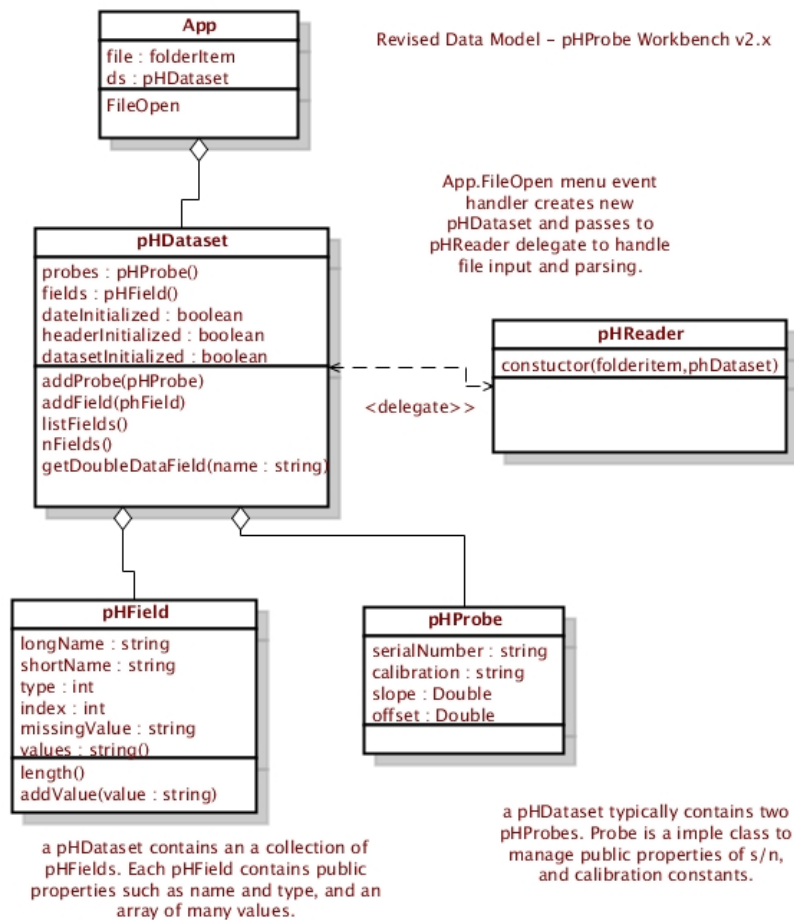
In January 2010, pHProbeWorkbench v1.x was ported from PC only VisualBasic 6.0 to RealStudios RealBasic, to be cross-platform for OSX, Linux and PC. The result is pHProbeWorkbench 2.1. The RealBasic version kept the exact same functionality and user interface, however the code was significantly refactored to take advantage of RealBasics OO Programming features.

Below is a screen capture of the V2.1 user interface.



pHProbeWorkbench

This page last changed on May 12, 2009 by [rich](#).



RealBasic Application

Poseidon VM Install Notes

This page last changed on Feb 25, 2010 by [rich](#).

2/25/2010

Downloaded Poseidon.3.1_32bits.iso to Home/Downloads/Poseidon

Start Fusion, File>New

Continue without disc

Select: "Use operating system installation disc image file" and navigate to .iso file

<s2.png>

Select: Linux and "Other 2.6.x kernel"

<s3.png>

Select: "Customize Settings" and give name (Poseidon.3.1 vm)

Setting Popups... Choose "Hard Drive" bump size from 8 to 20GB, and uncheck both "pre-allocate disk space" and "split to 2gb files" then Apply

Now the Virtual Machine Library should have a powered-off VM, dclick it and power it on

postgres

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$
```

Python

This page last changed on Dec 15, 2010 by [rich](#).

- [Database](#)

resume 18-Jan-13 pre-django

This page last changed on Apr 24, 2013 by [rich](#).

Pick up at 18-Jan-13 with the pre-django snapshot

1) Check for then install EPEL

<http://www.thegeekstuff.com/2012/06/enable-epel-repository/>\\

```
[rich@mbari1548-COS-vm ~]$ yum repolist
Loaded plugins: fastestmirror, refresh-packagekit, security
Determining fastest mirrors
Could not retrieve mirrorlist http://mirrorlist.centos.org/?release=6&arch=i386&repo=os error was
14: PYCURL ERROR 6 - "Couldn't resolve host 'mirrorlist.centos.org'"
repo id                repo name                status
base                   CentOS-6 - Base          0
extras                 CentOS-6 - Extras        0
pgdg91                 PostgreSQL 9.1 6 - i386  0
updates                CentOS-6 - Updates       0
repolist: 0
[rich@mbari1548-COS-vm ~]$
```

shows need stuff..

as root:

```
[rich@mbari1548-COS-vm ~]$ su -
Password:
[root@mbari1548-COS-vm ~]# rpm -Uvh http://ftp.linux.ncsu.edu/pub/epel/6/i386/epel-
release-6-8.noarch.rpm
Retrieving http://ftp.linux.ncsu.edu/pub/epel/6/i386/epel-release-6-8.noarch.rpm
warning: /var/tmp/rpm-tmp.UFfaT0: Header V3 RSA/SHA256 Signature, key ID 0608b895: NOKEY
Preparing...          ##### [100%]
 1:epel-release        ##### [100%]
[root@mbari1548-COS-vm ~]#
```

now redo the repolist

```
[root@mbari1548-COS-vm ~]# yum repolist
```

```
[root@mbari1548-COS-vm ~]# yum repolist
Loaded plugins: fastestmirror, refresh-packagekit, security
Loading mirror speeds from cached hostfile
epel/metalink                12 kB    00:00
* base: mirrors.tummy.co
* epel: linux.mirrors.es.net
* extras: mirrors.finalasp.com
* updates: mirrors.bluehost.com
base                          3.7 kB    00:00
epel                          4.2 kB    00:00
epel/primary_db               4.2 MB    00:00
extras                        3.5 kB    00:00
pgdg91                        2.8 kB    00:00
pgdg91/primary_db             125 kB    00:00
updates                       3.5 kB    00:00
updates/primary_db            1.4 MB    00:00
repo id                      repo                status
name                          status
base                           CentOS-6 -          4,790+12
Base                           Extra Packages for Enterprise Linux 6 - 7,152
epel                           CentOS-6 -          12
i386                           PostgreSQL 9.1 6 - 274
extras
Extras
pgdg91
i386
```

```
updates
Updates
repolist: 12,600
[root@mbari1548-COS-vm ~]#
```

CentOS-6 -

372

apache is working... from centos install earlier. test by getting ip then <http://134.89.11.51:80>

the apache site is at /var/www/

the apache cfg is at /etc/httpd/conf

postgres is there also - verify by becoming su then su - postgres the psql

I want virtualenv for python package management.

As root:

```
[root@mbari1548-COS-vm ~]# easy_install virtualenv
Searching for virtualenv
Reading http://pypi.python.org/simple/virtualenv/
Reading http://www.virtualenv.org
Reading http://virtualenv.openplans.org
Best match: virtualenv 1.9.1
Downloading http://pypi.python.org/packages/source/v/virtualenv/
virtualenv-1.9.1.tar.gz#md5=07e09df0adfca0b2d487e39a4bf2270a
Processing virtualenv-1.9.1.tar.gz
Running virtualenv-1.9.1/setup.py -q bdist_egg --dist-dir /tmp/easy_install-gpmmYD/
virtualenv-1.9.1/egg-dist-tmp-vpWkoe
warning: no previously-included files matching '*' found under directory 'docs/_templates'
warning: no previously-included files matching '*' found under directory 'docs/_build'
Adding virtualenv 1.9.1 to easy-install.pth file
Installing virtualenv script to /usr/bin
Installing virtualenv-2.6 script to /usr/bin

Installed /usr/lib/python2.6/site-packages/virtualenv-1.9.1-py2.6.egg
Processing dependencies for virtualenv
Finished processing dependencies for virtualenv
[root@mbari1548-COS-vm ~]#
```

Also want pip for python package install (yum replacement)

As root:

```
[root@mbari1548-COS-vm ~]# curl https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py | python
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 85929 100 85929    0     0 15544      0  0:00:05  0:00:05 --:--:-- 191k
Downloading/unpacking pip
  Downloading pip-1.3.1.tar.gz (247Kb): 247Kb downloaded
  Running setup.py egg_info for package pip
    warning: no files found matching '*.html' under directory 'docs'
    warning: no previously-included files matching '*.txt' found under directory 'docs/_build'
    no previously-included directories found matching 'docs/_build/_sources'
Installing collected packages: pip
  Running setup.py install for pip
    warning: no files found matching '*.html' under directory 'docs'
    warning: no previously-included files matching '*.txt' found under directory 'docs/_build'
    no previously-included directories found matching 'docs/_build/_sources'
  Installing pip script to /usr/bin
  Installing pip-2.6 script to /usr/bin
Successfully installed pip
Cleaning up...
[root@mbari1548-COS-vm ~]#
```

now as rich, make a sandbox to play in...

```
[rich@mbari1548-COS-vm ~]$ mkdir dev
[rich@mbari1548-COS-vm ~]$ cd dev
```

```
[rich@mbari1548-COS-vm dev]$ mkdir rovctdplots
[rich@mbari1548-COS-vm dev]$ cd rovctdplots
[rich@mbari1548-COS-vm rovctdplots]$ virtualenv rovctdplotsenv
New python executable in rovctdplotsenv/bin/python
Installing setuptools.....done.
Installing pip.....done.
[rich@mbari1548-COS-vm rovctdplots]$ ls
rovctdplotsenv
[rich@mbari1548-COS-vm rovctdplots]$

[rich@mbari1548-COS-vm rovctdplots]$ source ./rovctdplotsenv/bin/activate
```

for matplotlib etc... need numpy

[pip install numpy](#) (takes awhile, at times looks like its hung, just wait...)

[pip install matplotlib](#)

25-Apr-2013

so now I want to enable mod_wsgi

looks like its install into apache at some point (default?) there is a wsgi.conf at /etc/httpd/conf.d that loads the mod_wsgi.so

I needed to edit the /etc/httpd/conf to enable execute perms on /var/www/cgi-bin: <Directory "/var/www/cgi-bin">

```
    AllowOverride None
    Options +ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

also added handler in same file...

AddHandler wsgi-script .wsgi

restarted httpd

then

created a small test.wsgi file in /var/www/cgi-bin

```
[root@mbari1548-COS-vm cgi-bin]# more test.wsgi
def application(environ, start_response):
    """Simplest possible application object"""
    output = "Hello World"
    status = '200 OK'
    response_headers = [('Content-type', 'text/plain'),
                        ('Content-Length', str(len(output)))]
    start_response(status, response_headers)
    return [output]
```

make it read and executable..

browse to <http://134.89.11.51/cgi-bin/test.wsgi>

it works

ROV Pilots Dive Log

This page last changed on Oct 05, 2011 by [rich](#).

rovctd plots

This page last changed on Jul 25, 2008 by [rich](#).

test deployment is on anvil

1. use remote desktop to login as rich on anvil
2. need richdatepicker? in c:\rovctd\test.aspx
3. need ant and java installed and in path etc
4. on mando, open dos cmd window,
 - a. cd to c:\iag\projects\RovctdPlots
 - b. type ant run.1 (runs the run.1 target in the build.xml file)
 - i. console text says its hitting <http://anvil:8085/rovctd/servlet/RovctdDataServlet?platform=vnta&raw=0&year=2008&yearday=191&dive=null&domain=epochsecs&r1=p&r2=t&noHTML>

tomcat servlet container

This page last changed on Jul 25, 2008 by [rich](#).

rovctd plots and data servlets are served by tomcat

on anvil, see c:\Program Files\Apache Software Foundation\Tomcat 5.5

the server port number is changed from default 8080 to 8085 in the tomcats Catalina Service

see:conf\server.xml and conf\server-minimal.xml files

```
<!--rschramm, change from default 8080 to 8085 12-Apr-2007-->f(clean);- <!--Define a non-SSL  
HTTP/1.1 Connector on port 8080-->f(clean);  
  <Connector port="8085" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"  
maxSpareThreads="75" enableLookups="false" redirectPort="8443" acceptCount="100"  
connectionTimeout="20000" disableUploadTimeout="true" />
```

[FOR DATABASE SETUP SEE THIS](#)

database setup

This page last changed on Jul 25, 2008 by rich.

See <http://tomcat.apache.org/tomcat-5.0-doc/jndi-datasource-examples-howto.html> and <http://service.baynote.net/tomcat-docs/printer/jndi-datasource-examples-howto.html>

1) Copy the following into CATALINA_HOME/common/lib:

- a) jTDS JDBC Driver (for MS SQL Server)
- b) Apache commons-pool
- c) Apache commons-dbcp
- d) Apache commons-logging
- e) Apache commons-collection

*** these files will be named like: commons-dbcp-1.2.1.jar
and should reside in the common/lib folder, not a subfolder (winzip might extract them to a sub-folder of the name commons-dbcp-1.2.1)

2) Configure the JNDI DataSource in Tomcat by adding a declaration for your resource to \$CATALINA_HOME/conf/server.xml. Add this in between the </Context> tag of the examples context and the </Host> tag closing the localhost definition.

```
<Context path="/rovctd" docBase="rovctd" reloadable="true" crossContext="true">
  <Resource
    driverClassName="net.sourceforge.jtds.jdbc.Driver"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory"
    maxActive="10"
    maxIdle="5"
    maxWait="5000"
    name="jdbc/ROVCTD"
    password="password"
    type="javax.sql.DataSource"
    url="jdbc:jtds:sqlserver://mando.shore.mbari.org:1433/Expd"
    username="expddba"
    removeAbandoned="true"
    removeAbandonedTimeout="60"
    logAbandoned="true"/>
</Context>
```

3) Add the following snippet to your applications WEB-INF/web.xml file. You should ensure that you respect the element ordering defined by the DTD when you create your applications web.xml file.

```
<resource-ref>
<description>
  Resource reference to a factory for java.sql.Connection
  instances that may be used for talking to a particular
  database that is configured in the server.xml file.
</description>
<res-ref-name>jdbc/ROVCTD</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
</resource-ref>
```

4) make sure jTds driver file is in

C:\Program Files\Apache Software Foundation\Tomcat 5.5\common\lib
(also might need it in C:\Program Files\Apache Software Foundation\Tomcat 5.5\shared\lib
but when I was test deploying to anvil it was needed in common)

5) Fetching the connection will be done by:

```
Context initContext = new InitialContext();
Context envContext = (Context)initContext.lookup("java:/comp/env");
DataSource ds = (DataSource)envContext.lookup("jdbc/ROVCTD");
Connection conn = ds.getConnection();
```

LOOK AT THESE EXAMPLES FOR [conf/server.xml](#) and [web.xml](#)

;

Tomcat 5.5 server.xml on anvil

This page last changed on Jul 25, 2008 by [rich](#).

~~<!-- See midway for change to port 8085 and at bottom for the database connection --R.Schramm-->~~

~~<!-- Example Server Configuration File -->~~

~~<!-- Note that component elements are nested corresponding to their parent-child relationships with each other -->~~

~~<!-- A "Server" is a singleton element that represents the entire JVM, which may contain one or more "Service" instances. The Server listens for a shutdown command on the indicated port.~~

~~Note: A "Server" is not itself a "Container", so you may not define subcomponents such as "Valves" or "Loggers" at this level.~~

~~-->~~

~~<Server port="8005" shutdown="SHUTDOWN">~~

~~<!-- Comment these entries out to disable JMX MBeans support used for the administration web application -->~~

~~<Listener className="org.apache.catalina.core.AprLifecycleListener" />~~

~~<Listener className="org.apache.catalina.mbeans.ServerLifecycleListener" />~~

~~<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />~~

~~<Listener className="org.apache.catalina.storeconfig.StoreConfigLifecycleListener"/>~~

~~<!-- Global JNDI resources -->~~

~~<GlobalNamingResources>~~

~~<!-- Test entry for demonstration purposes -->~~

~~<Environment name="simpleValue" type="java.lang.Integer" value="30"/>~~

~~<!-- Editable user database that can also be used by~~

~~UserDatabaseRealm to authenticate users -->~~

~~<Resource name="UserDatabase" auth="Container"~~

~~type="org.apache.catalina.UserDatabase"~~

~~description="User database that can be updated and saved"~~

~~factory="org.apache.catalina.users.MemoryUserDatabaseFactory"~~

~~pathname="conf/tomcat-users.xml" />~~

~~</GlobalNamingResources>~~

~~<!-- A "Service" is a collection of one or more "Connectors" that share a single "Container" (and therefore the web applications visible within that Container). Normally, that Container is an "Engine", but this is not required.~~

~~Note: A "Service" is not itself a "Container", so you may not define subcomponents such as "Valves" or "Loggers" at this level.~~

~~-->~~

~~<!-- Define the Tomcat Stand-Alone Service -->~~

~~<Service name="Catalina">~~

~~<!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Each Connector passes requests on to the associated "Container" (normally an Engine) for processing.~~

By default, a non-SSL HTTP/1.1 Connector is established on port 8080. You can also enable an SSL HTTP/1.1 Connector on port 8443 by following the instructions below and uncommenting the second Connector entry. SSL support requires the following steps (see the SSL Config HOWTO in the Tomcat 5 documentation bundle for more detailed instructions):

- If your JDK version 1.3 or prior, download and install JSSE 1.0.2 or

- later, and put the JAR files into "\$JAVA_HOME/jre/lib/ext".
- Execute:
 %JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA (Windows)
 \$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA (Unix)
 with a password value of "changeit" for both the certificate and the keystore itself.

By default, DNS lookups are enabled when a web application calls `request.getRemoteHost()`. This can have an adverse impact on performance, so you can disable it by setting the "enableLookups" attribute to "false". When DNS lookups are disabled, `request.getRemoteHost()` will return the String version of the IP address of the remote client.

```
-->
<!--rschramm, change from default 8080 to 8085 12-Apr-2007-->
<!--Define a non-SSL HTTP/1.1 Connector on port 8080-->
<Connector
port="8085" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true" />
<!-- Note : To disable connection timeouts, set connectionTimeout value
to 0 -->

<!-- Note : To use gzip compression you could set the following properties :

compression="on"
compressionMinSize="2048"
noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml"
-->

<!--Define a SSL HTTP/1.1 Connector on port 8443-->
<!--
<Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
-->

<!--Define an AJP 1.3 Connector on port 8009-->
<Connector port="8009"
enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />

<!--Define a Proxied HTTP/1.1 Connector on port 8082-->
<!-- See proxy documentation for more information about using this.-->
<!--
<Connector port="8082"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" acceptCount="100" connectionTimeout="20000"
proxyPort="80" disableUploadTimeout="true" />
-->

<!-- An Engine represents the entry point (within Catalina) that processes
every request. The Engine implementation for Tomcat stand alone
analyzes the HTTP headers included with the request, and passes them
on to the appropriate Host (virtual host). -->

<!-- You should set jvmRoute to support load-balancing via AJP ie :
<Engine name="Standalone" defaultHost="localhost" jvmRoute="jvm1">
-->

<!--Define the top-level container in our container hierarchy-->
<Engine name="Catalina" defaultHost="localhost">
```

<!-- The request dumper valve dumps useful debugging information about the request headers and cookies that were received, and the response headers and cookies that were sent, for all requests received by this instance of Tomcat. If you care only about requests to a particular virtual host, or a particular application, nest this element inside the corresponding <Host> or <Context> entry instead.

For a similar mechanism that is portable to all Servlet 2.4 containers, check out the "RequestDumperFilter" Filter in the example application (the source for this filter may be found in "\$CATALINA_HOME/webapps/examples/WEB-INF/classes/filters").

Request dumping is disabled by default. Uncomment the following element to enable it. -->

```
<!--  
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>  
-->
```

<!-- Because this Realm is here, an instance will be shared globally -->

<!-- This Realm uses the UserDatabase configured in the global JNDI resources under the key "UserDatabase". Any edits that are performed against this UserDatabase are immediately available for use by the Realm. -->

```
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"  
resourceName="UserDatabase"/>
```

<!-- Comment out the old realm but leave here for now in case we need to go back quickly -->

```
<!--  
<Realm className="org.apache.catalina.realm.MemoryRealm" />  
-->
```

<!-- Replace the above Realm with one of the following to get a Realm stored in a database and accessed via JDBC -->

```
<!--  
<Realm className="org.apache.catalina.realm.JDBCRealm"  
driverName="org.gjt.mm.mysql.Driver"  
connectionURL="jdbc:mysql://localhost/authority"  
connectionName="test" connectionPassword="test"  
userTable="users" userNameCol="user_name" userCredCol="user_pass"  
userRoleTable="user_roles" roleNameCol="role_name" />  
-->
```

```
<!--  
<Realm className="org.apache.catalina.realm.JDBCRealm"  
driverName="oracle.jdbc.driver.OracleDriver"  
connectionURL="jdbc:oracle:thin:@ntserver:1521:ORCL"  
connectionName="scott" connectionPassword="tiger"  
userTable="users" userNameCol="user_name" userCredCol="user_pass"  
userRoleTable="user_roles" roleNameCol="role_name" />  
-->
```

```
<!--  
<Realm className="org.apache.catalina.realm.JDBCRealm"  
driverName="sun.jdbc.odbc.JdbcOdbcDriver"  
connectionURL="jdbc:odbc:CATALINA"  
userTable="users" userNameCol="user_name" userCredCol="user_pass"  
userRoleTable="user_roles" roleNameCol="role_name" />  
-->
```

<!-- Define the default virtual host

Note: XML Schema validation will not work with Xerces 2.2.

```
-->  
<Host name="localhost" appBase="webapps"
```

```
unpackWARs="true" autoDeploy="true"
xmlValidation="false" xmlNamespaceAware="false">
```

<!-- Defines a cluster for this node,
By defining this element, means that every manager will be changed.
So when running a cluster, only make sure that you have webapps in there
that need to be clustered and remove the other ones.
A cluster has the following parameters:

className = the fully qualified name of the cluster class

name = a descriptive name for your cluster, can be anything

mcastAddr = the multicast address, has to be the same for all the nodes

mcastPort = the multicast port, has to be the same for all the nodes

mcastBindAddr = bind the multicast socket to a specific address

mcastTTL = the multicast TTL if you want to limit your broadcast

mcastSoTimeout = the multicast readtimeout

mcastFrequency = the number of milliseconds in between sending a "I'm alive" heartbeat

mcastDropTime = the number a milliseconds before a node is considered "dead" if no heartbeat is received

tcpThreadCount = the number of threads to handle incoming replication requests, optimal would be the same amount of threads as nodes

tcpListenAddress = the listen address (bind address) for TCP cluster request on this host,
in case of multiple ethernet cards.
auto means that address becomes
InetAddress.getLocalHost().getHostAddress()

tcpListenPort = the tcp listen port

tcpSelectorTimeout = the timeout (ms) for the Selector.select() method in case the OS
has a wakeup bug in java.nio. Set to 0 for no timeout

printToScreen = true means that managers will also print to std.out

expireSessionsOnShutdown = true means that

useDirtyFlag = true means that we only replicate a session after setAttribute,removeAttribute has been called.

false means to replicate the session after each request.

false means that replication would work for the following piece of code: (only for SimpleTcpReplicationManager)

```
<%
HashMap map = (HashMap)session.getAttribute("map");
map.put("key","value");
%>
```

replicationMode = can be either 'pooled', 'synchronous' or 'asynchronous'.

- Pooled means that the replication happens using several sockets in a synchronous way. Ie, the data gets replicated, then the request return. This is the same as the 'synchronous' setting except it uses a pool of sockets, hence it is multithreaded. This is the fastest and safest configuration. To use this, also increase the nr of tcp threads that you have dealing with replication.
- Synchronous means that the thread that executes the request, is also the thread the replicates the data to the other nodes, and will not return until all nodes have received the information.
- Asynchronous means that there is a specific 'sender' thread for each cluster node, so the request thread will queue the replication request into a "smart" queue, and then return to the client.
The "smart" queue is a queue where when a session is added to the queue, and the same session

already exists in the queue from a previous request, that session will be replaced in the queue instead of replicating two requests. This almost never happens, unless there is a large network delay.

-->
<!--

When configuring for clustering, you also add in a valve to catch all the requests coming in, at the end of the request, the session may or may not be replicated.

A session is replicated if and only if all the conditions are met:

1. useDirtyFlag is true or setAttribute or removeAttribute has been called AND
2. a session exists (has been created)
3. the request is not trapped by the "filter" attribute

The filter attribute is to filter out requests that could not modify the session, hence we don't replicate the session after the end of this request.

The filter is negative, ie, anything you put in the filter, you mean to filter out, ie, no replication will be done on requests that match one of the filters.

The filter attribute is delimited by ;, so you can't escape out ; even if you wanted to.

filter=". \.gif;.\.js;" means that we will not replicate the session after requests with the URI ending with .gif and .js are intercepted.

The deployer element can be used to deploy apps cluster wide.

Currently the deployment only deploys/undeploys to working members in the cluster so no WARs are copied upon startup of a broken node.

The deployer watches a directory (watchDir) for WAR files when watchEnabled="true"

When a new war file is added the war gets deployed to the local instance, and then deployed to the other instances in the cluster.

When a war file is deleted from the watchDir the war is undeployed locally and cluster wide

-->

<!--

```
<Cluster className="org.apache.catalina.cluster.tcp.SimpleTcpCluster"
managerClassName="org.apache.catalina.cluster.session.DeltaManager"
expireSessionsOnShutdown="false"
useDirtyFlag="true"
notifyListenersOnReplication="true">
```

```
<Membership
className="org.apache.catalina.cluster.mcast.McastService"
mcastAddr="228.0.0.4"
mcastPort="45564"
mcastFrequency="500"
mcastDropTime="3000"/>
```

```
<Receiver
className="org.apache.catalina.cluster.tcp.ReplicationListener"
tcpListenAddress="auto"
tcpListenPort="4001"
tcpSelectorTimeout="100"
tcpThreadCount="6"/>
```

```
<Sender
className="org.apache.catalina.cluster.tcp.ReplicationTransmitter"
replicationMode="pooled"
ackTimeout="15000"/>
```

```
<Valve className="org.apache.catalina.cluster.tcp.ReplicationValve"
filter=". \.gif;.\.js;.\.jpg;.\.htm;.\.html;.\.txt;"/>
```

```
<Deployer className="org.apache.catalina.cluster.deploy.FarmWarDeployer"
tempDir="/tmp/war-temp/"
deployDir="/tmp/war-deploy/"
watchDir="/tmp/war-listen/"
watchEnabled="false"/>
</Cluster>
```

```

-->

<!-- Normally, users must authenticate themselves to each web app
individually. Uncomment the following entry if you would like
a user to be authenticated the first time they encounter a
resource protected by a security constraint, and then have that
user identity maintained across all web applications contained
in this virtual host. -->
<!--
<Valve className="org.apache.catalina.authenticator.SingleSignOn" />
-->

<!-- Access log processes all requests for this virtual host. By
default, log files are created in the "logs" directory relative to
$CATALINA_HOME. If you wish, you can specify a different
directory with the "directory" attribute. Specify either a relative
(to $CATALINA_HOME) or absolute path to the desired directory.
-->
<!--
<Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs" prefix="localhost_access_log." suffix=".txt"
pattern="common" resolveHosts="false"/>
-->

<!-- Access log processes all requests for this virtual host. By
default, log files are created in the "logs" directory relative to
$CATALINA_HOME. If you wish, you can specify a different
directory with the "directory" attribute. Specify either a relative
(to $CATALINA_HOME) or absolute path to the desired directory.
This access log implementation is optimized for maximum performance,
but is hardcoded to support only the "common" and "combined" patterns.
-->
<!--
<Valve className="org.apache.catalina.valves.FastCommonAccessLogValve"
directory="logs" prefix="localhost_access_log." suffix=".txt"
pattern="common" resolveHosts="false"/>
-->

<Context path="/rovctd" docBase="rovctd" reloadable="true" crossContext="true">
<Resource
driverClassName="net.sourceforge.jtds.jdbc.Driver"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="10"
maxIdle="5"
maxWait="5000"
name="jdbc/ROVCTD"
password="password"
type="javax.sql.DataSource"
url="jdbc:jtds:sqlserver://SOLSTICE:1433/Expd"
username="expddba"
removeAbandoned="true"
removeAbandonedTimeout="60"
logAbandoned="true"/>
</Context>

</Host>

</Engine>

</Service>

</Server>

```

web.xml on anvil

This page last changed on Jul 25, 2008 by [rich](#).

sits below webapps\rovctd\WEB-INF\web.xml

declares servletname to java class mapping and a link to the database connection defined in conf\server.xml

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

<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">

<web-app>

<servlet>
<servlet-name>
RovctdDataServlet
</servlet-name>

<servlet-class>
org.mbari.rovctd.datagenerators.RovctdDataServlet
</servlet-class>

<init-param>
<param-name>deployedDate</param-name>
<param-value>15-Aug-2006</param-value>
</init-param>

<init-param>
<param-name>jdbcDriver</param-name>
<param-value>net.sourceforge.jtds.jdbc.Driver</param-value>
</init-param>

</servlet>

<servlet-mapping>
<servlet-name>RovctdDataServlet</servlet-name>
<url-pattern>/servlet/RovctdDataServlet</url-pattern>
</servlet-mapping>

<resource-ref>
<description>
Resource reference to a factory for java.sql.Connection
instances that may be used for talking to a particular
database that is configured in the server.xml file.
</description>
<res-ref-name>jdbc/ROVCTD</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<res-auth>Container</res-auth>
</resource-ref>

</web-app>
```

Updating ROVCTD Plots using JFreeChart and Tomcat

This page last changed on Dec 30, 2010 by [rich](#).

29-Dec-2010, Want to revisit rovctd plots and tomcat servlet container to prepare to bring them up-to-date, they havent been touched for years....

eg. tomcat on anvil is 5.5 and current rev is 7.0, java was 1.4

Im setting it up on macbook in my windows 7 VM

Install java sdk

First I installed java6 jre but that was a mistake... I will need the java sdk to get tools.jar etc. that ANT requires. (jre might be ok though for deployment to anvil or typhoon?)

So get java sdk from oracle... not my vm is 32-bit cpu. so get the right one., download w/firefox and click the resulting exewhich was jdk-6u23-windows-i586.exe

Set environment variables:

JAVA_HOME C:\Program Files\Java\jdk1.6.0_23

JRE_HOME C:\Program Files\Java\jdk1.6.0_23 (for tomcat)

Add to PATH: ;%JAVA_HOME%\bin

Install ant

Install ant 1.8.2, firefox download and unzip to c:\ant <<< my choice....

Set environment variables:

ANT_HOME C:\Program Files\Ant\apache-ant-1.8.2

Add to PATH: ;%ANT_HOME%\bin

Install Tomcat

download only the 32-bit/64-bit Windows Service Installer (it gets the Tomcat binaries)

Actually the link downloads: apache-tomcat-7.0.5.exe5)

cd to it in a cmd window and execute it, or just click it in the firefox downloads-manage

use defaults, or pick a different port # and be sure to pick an admin acct and password to be able to manage it remotely. NOTE Anvil gets port 8085 because there is an apache with 8080...

test: <http://localhost:8080>

see: <http://tomcat.apache.org/download-70.cgi>

see also: <http://www.mulesoft.com/tomcat-windows/>

Clean up java path problems

I had a lot of issues with no class def found when trying to run java to test installation. Turned out something, either tomcat or javasdk put a copy of java.exe in Windows\system32. I deleted those 3 exe's all installed with same timestamp.

Also had to go to tomcat7\bin and run the tomcat7w.exe and check the java tab to get it to point away from the jre6 jvm.dll installed earlier and point to the sdk version down in C:\Program Files\Java\jdk1.6.0_23\jre\bin\client\jvm.dll

Get a bunch of Jar files....

jTDS (1.2 upgraded to 1.2.5)

commons.pool (1.3 > 1.5.5)

commons.dbcp (1.2.1 > 1.4)

commons.logging (1.1.1)

commons-collections (3.2 > 3.2.1)

All of the above, basically download the binary distributions, unzip somewhere, then drag the .jar file to c:/Program Files/Apache Software Foundation/Tomcat 7.0/lib

Get jFreeChart

download distribution of jFreeChart (1.0.2 > 1.0.13)

get the jfreechart-1.0.13.jar and jcommon-1.0.16.jar that is with it and drag to c:/Program Files/Apache Software Foundation/Tomcat 7.0/lib

Get the old rovctd webapps folder

from anvils tomcat 5.5 and drop it into my tomcat 7.0 webapps dir

Checkout RovctdPlots from cvs to C:\Projects

edit all applications.java to point to localhost 8080 or....anvil or typhoon

edit build.xml

configure server.xml etc according to database inst. here and elsewhere...

need the jtids add, also note that is where the connection is set to either solstice or other...

Restart tomcat....

I often had to start twice after stoppin, it throws error when stopping, then have to start twice, once with error.

Fist time after startup, takes a long time to runt the app.

Able to do ant

clean

prepare

compile

deploy.libs

deploy.datagenerators

deploy.plots

run.1 thru 5

Stopped for today... doesnt run over browser. I get errors

http://134.89.11.41:8080/rovctd/pages/stackedtimeseries.jsp?
platform=vnta&dive=3596&domain=epochsecs&noHTMLHeader=0&r1=p&r2=t&r3=s&r4=light&r5=o2\
\ org.apache.jasper.JasperException: /pages/stackedtimeseries.jsp(24,49) Attribute value
request.getParameter("platform") is quoted with " which must be escaped when used within the value
org.apache.jasper.compiler.DefaultErrorHandler.jspError(DefaultErrorHandler.java:40)
org.apache.jasper.compiler.ErrorDispatcher.dispatch(ErrorDispatcher.java:407)

org.apache.jasper.compiler.ErrorDispatcher.jspError(ErrorDispatcher.java:88)
org.apache.jasper.compiler.Parser.parseAttributeValue(Parser.java:280)

This page last changed on Jun 01, 2011 by rich.

Stop tomcat 5.0

1. in services, stop and then disable from starting Tomcat 5.0

Install java 1.6

1. a. get jdk-6u23-windows-i586.exe from oracle (tempbox\rich) and put in c:\download or other temporary directory on anvil
b. open cmd window, navigate to and run the above .exe . (optional: on the wizard that comes up, dont install database and src code)
c. will create c:\Program Files\java\jre6 and c:\Program Files\jdk1.6.0_23
d. Set environment variables:
JAVA_HOME C:\Program Files\Java\jdk1.6.0_23
JRE_HOME C:\Program Files\Java\jdk1.6.0_23 (for tomcat)
e. Add to PATH: ;%JAVA_HOME%\bin
1. a. **open new cmd** window and type java -version
i. should respond java version 1.6.0_23 etc...

Install ant

1. create dir C:\Program Files\Ant
2. Copy apache-ant-1.8.2-bin.zip to above and unzip all
a. creates apache-ant-1.8.2 sub directory
3. Set environment variables:
ANT_HOME C:\Program Files\Ant\apache-ant-1.8.2
4. Add to PATH: ;%ANT_HOME%\bin
5. **open new cmd** window and type ant -version should respond ant version 1.8.2 etc..

Install tomcat

1. Get from tempbox\rich\downloads: apache-tomcat-7.0.5.exe
2. or ... download only the 32-bit/64-bit Windows Service Installer (it gets the Tomcat binaries)
Actually the link downloads: apache-tomcat-7.0.5.exe
3. Put it to some temp dir, open a dos cmd window, navigate to it and execute it
4. Use default port 8080 (typhoon), or pick a different port e.g. 8085 for anvil) admin acct = "admin" and our password to be able to manage it remotely.
5. NOTE Anvil is/was port 8085 because there is already an Apache server with 8080.
6. **Accd to M.McCann, IS has new standards... port 8443, firewall cfg, and ssl stuff to be configured here....**
7. Use services panel to stop old apache server, and start the apache7 service. Note:Had to do this twice...first time failed to start. Also doesnt like to stop cleanly....
8. I was able to go to another machine and <http://134.89.10.35:8085/> or <http://anvil:8085>

Also see:

see:<http://tomcat.apache.org/download-70.cgi>

see also:<http://www.mulesoft.com/tomcat-windows\\>

Install jTDS Drivers, jFreeChart and a bunch of other jar files:

These should all be available in tempbox\rich\downloads\justthejarsplease
jfreechart-1.0.13.jar (from tempbox/rich/downloads)

jcommon-1.0.16.jar

jFastCGI-2.0.jar

commons-logging-1.1.1.jar

commons-pool-1.5.5.jar

commons-collections-3.2.1.jar

commons-dbcp-1.4.jar

jtids-1.2.5.jar

Copy them all to C:\Program Files\Apache Software Foundation\Tomcat 7.0\lib

Or else get them from sourceforge etc like:

1. via <http://jtids.sourceforge.net/>
 - a. jtids-1.2.5-dist.zip copy to C:\downloads\jtids
 - b. extract to there
 - c. move the jtids-1.2.5.jar to C:\Program Files\Tomcat 7.0\lib
 - d. delete everything else including the zip
 - e.
- 2.
- 3.
4. modify C:\Program Files\Tomcat 7.0\conf\server.xml so bottom looks like:

```
<Context path="/rovctd" docBase="rovctd" reloadable="true" crossContext="true">
  <Resource
    driverClassName="net.sourceforge.jtids.jdbc.Driver"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory"
    maxActive="10"
    maxIdle="5"
    maxWait="5000"
    name="jdbc/ROVCTD"
    password="password"
    type="javax.sql.DataSource"
    url="jdbc:jtids:sqlserver://SOLSTICE:1433/Expd"
    username="expddba"
    removeAbandoned="true"
    removeAbandonedTimeout="60"
    logAbandoned="true"/>
  </Context>

</Host>
</Engine>
</Service>
</Server>
```

Install jFreeChart and a bunch of other jar files:

jfreechart-1.0.13.jar (from tempbox/rich/downloads)

jcommon-1.0.16.jar

jFastCGI-2.0.jar

commons-logging-1.1.1.jar

commons-pool-1.5.5.jar

commons-collections-3.2.1.jar

commons-dbcp-1.4.jar

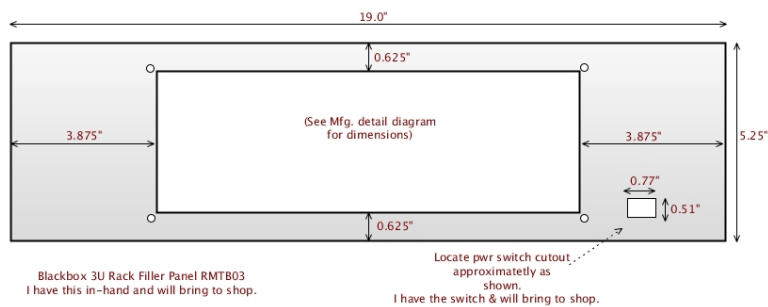
jtids-1.2.5.jar

Install the webapps

1. rovctd
 - a. grab rovctd folder from tempbox\rich\downloads put in C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps
2. jitensha-rougu (Brian Schlining's...) ANVIL ONLY, NOT TYPHOON?
 - a. copy Tomcat 5.5\webapps\jitensha-rogu.war to Tomcat 7.0\webapps
 - b. restart tomcat service (should create cause tomcat to create webapps\jitensh-rogu folder and sub-folders)
3. any other webapps...besides defaults? (manager and ROOT)

Vorne Display Panel - Rackmount Drawing

This page last changed on Mar 07, 2012 by rich.



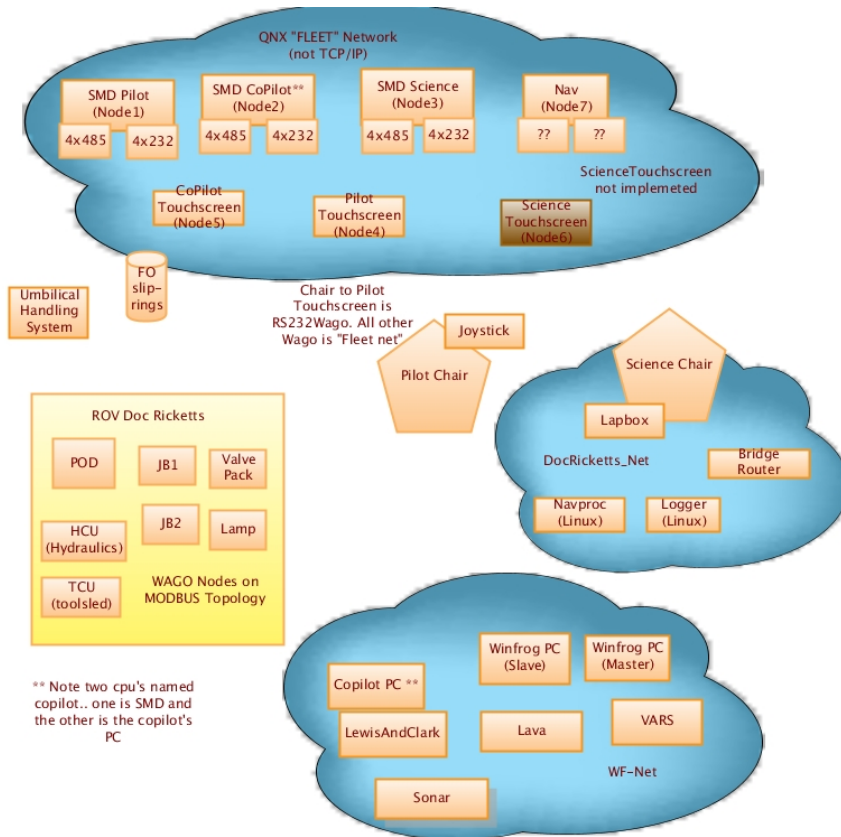
WesternFlyer DocRicketts

This page last changed on Jan 24, 2012 by [rich](#).

System View

Computers

This page last changed on Jan 24, 2012 by rich.



XML-RPC Notes

This page last changed on Jul 11, 2008 by [rich](#).

SEE <http://xmlrpc-c.sourceforge.net/>

DATABASE

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

Add py database stuff

12-Dec-2010 Need something to talk python to mssql from osx.

see: <http://wiki.python.org/moin/SQL%20Server>

also see: <http://meantheory.wordpress.com/2009/10/01/mssql-freetds-pyodbc-snow-leopard>

INSTALL pyodbc

- download and unzip pyodbc-2.1.8.zip
- unzip to ~/Downloads/pyodbc-2.1.8
- follow readme: python setup.py build install

INSTALL freetds:

- follow instructions from: <http://meantheory.wordpress.com/2009/10/01/mssql-freetds-pyodbc-snow-leopard/>
- download and unzip freetds-stable.tgz
- unzip to ~/Downloads/freetds-0.82
- ./configure -prefix=/usr/local/freetds -with-iodbc=/usr -with-tdsver=8.0
- make
- sudo make install

Test freetds:

1) sudo /usr/bin/lipo -info /usr/local/freetds/lib/libtdsodbc.so

should see:

Non-fat file: /usr/local/freetds/lib/libtdsodbc.so is architecture: x86_64

2) /usr/bin/iodbctest "Driver=/usr/local/freetds/lib/libtdsodbc.so;Server=134.89.12.29;Port=1433;TDS_Version=8.0;UID=expddba;pwd=*****;"

3) test with python

```
import pyodbc
driver='/usr/local/freetds/lib/libtdsodbc.so'
uid='expddba'
pwd='expdpasswordhere'
server='134.89.12.29'
port='1433'
cn = "Driver=%s;Server=%s;Port=%s;TDS_Version=8.0;UID=%s;pwd=%s;" %
( driver,server,port,uid,pwd)
cnxn = pyodbc.connect(cn)
c1 = cnxn.cursor()
c1.execute("exec sp_who")
for row in c1:
    print row
```

Note the code above does not require using an ODBC DSN such as you might create with the the iODBC.app....

Though that can be done, but the iODBC.app downloaded from and install it from xcode doesnt work... complains about wrong architecure when trying to test the connection

But you can manually create/use the 3 config files that define the DNS that iODBC.app uses.

see: <http://lists.ibiblio.org/pipermail/freetds/2009q2/024684.html>

The connection string becomes: cn = "Driver=%s;DSN=expd_ftds;TDS_Version=8.0;UID=%s;pwd=%s;" % (driver,uid,pwd)

Below are extracts from the three files that will create the "DSN=expd_ftd" used above.:

Ignore the "Actual" drivers, those were install from ActualTechnologies for my RealBasic ODBC projects.....

****IN** /usr/local/freetds/etc/freetds.conf**

```
[expd_ftds]
    host = solstice.shore.mbari.org
    port = 1433
    tds version = 8.0
```

If it exists: **IN /opt/local/etc/freetds/freetds.conf**

```
[expd_ftds]
    host = solstice.shore.mbari.org
    port = 1433
    tds version = 8.0
```

****IN** /Library/ODBC/odbcinst.ini**

```
[ODBC Drivers]
Actual SQL Server          = Installed
Actual Open Source Databases = Installed
Actual Oracle              = Installed
Actual Access              = Installed
FreeTDS                    = Installed
```

```
[Actual SQL Server]
Driver = /Library/ODBC/Actual SQL Server.bundle/Contents/MacOS/atsqlsrv.so
Setup =
```

```
[Actual Open Source Databases]
Driver = /Library/ODBC/Actual Open Source Databases.bundle/Contents/MacOS/atopnsrvc.so
Setup =
```

```
[Actual Oracle]
Driver = /Library/ODBC/Actual Oracle.bundle/Contents/MacOS/atoradb.so
Setup =
```

```
[Actual Access]
Driver = /Library/ODBC/Actual Access.bundle/Contents/MacOS/ataccess.so
Setup =
```

```
[FreeTDS]
Driver = /opt/local/lib/libtdsodbc.0.s0
Setup =
```

```
[ODBC Connection Pooling]
PerfMon    = 0
Retry Wait =
```

****IN** /Library/ODBC/odbc.ini**

```
[ODBC Data Sources]
Solstice = Actual SQL Server
equinox  = Actual SQL Server
expd_ftds = FreeTDS
```

```
[equinox]
Driver      = /Library/ODBC/Actual SQL Server.bundle/Contents/MacOS/atsqlsrv.so
Description = equinox expddba
Server      = 134.89.12.101
UserID      = expddba
UseKeychain = Yes
ServerName  = equinox
host        = 134.89.12.101
client charset = UTF-8
Port        = 51001
```

```
[solstice]
Driver      = /Library/ODBC/Actual SQL Server.bundle/Contents/MacOS/atsqlsrv.so
Description = solstice flyerstaging
Server      = 134.89.12.29
Database    = expd
UserID      = expddba
UseKeychain = Yes
ServerName  = Solstice
host        = 134.89.12.29
client charset = UTF-8

[expd_ftds]
Driver      = /opt/local/lib/libtdsodbc.0.so
Description = macports version
ServerName  = expd_ftds
```

pHProbeWorkbench Model Classes

This page last changed on Sep 30, 2011 by [rich](#).