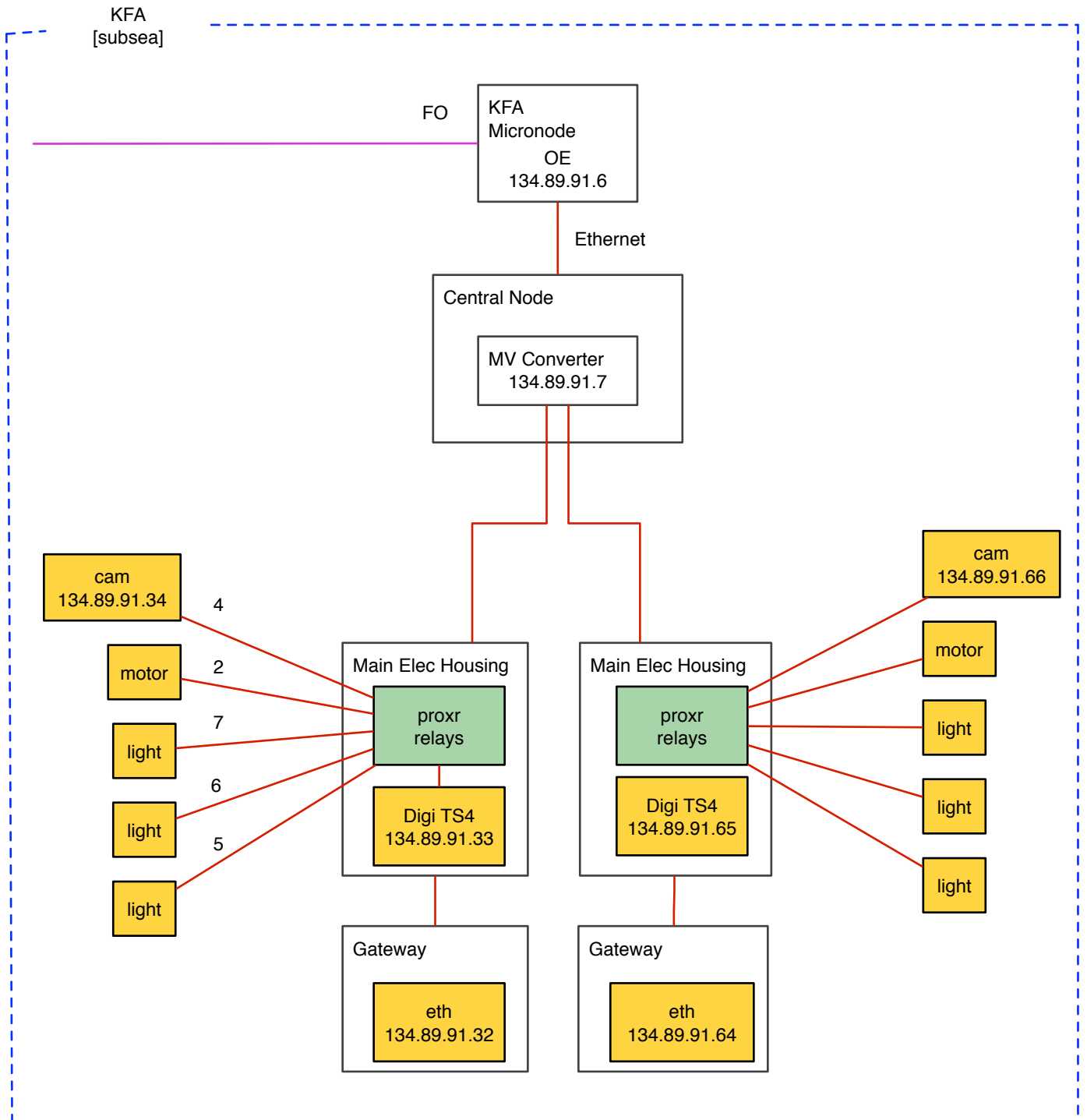


Gateway Software Utility Guide Contents

Topology	swFOCE 2015 configuration
Utility Map	utility reference
Basics	basic gateway concepts
Gateway	system administration common operations
Archive	working with data archives and plots examples

Gateway Software Utility Guide

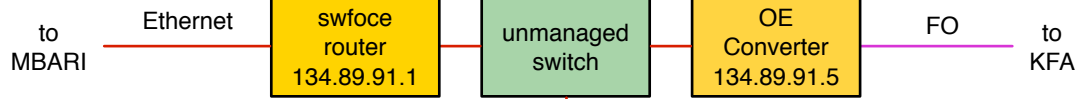
Topology: KFA



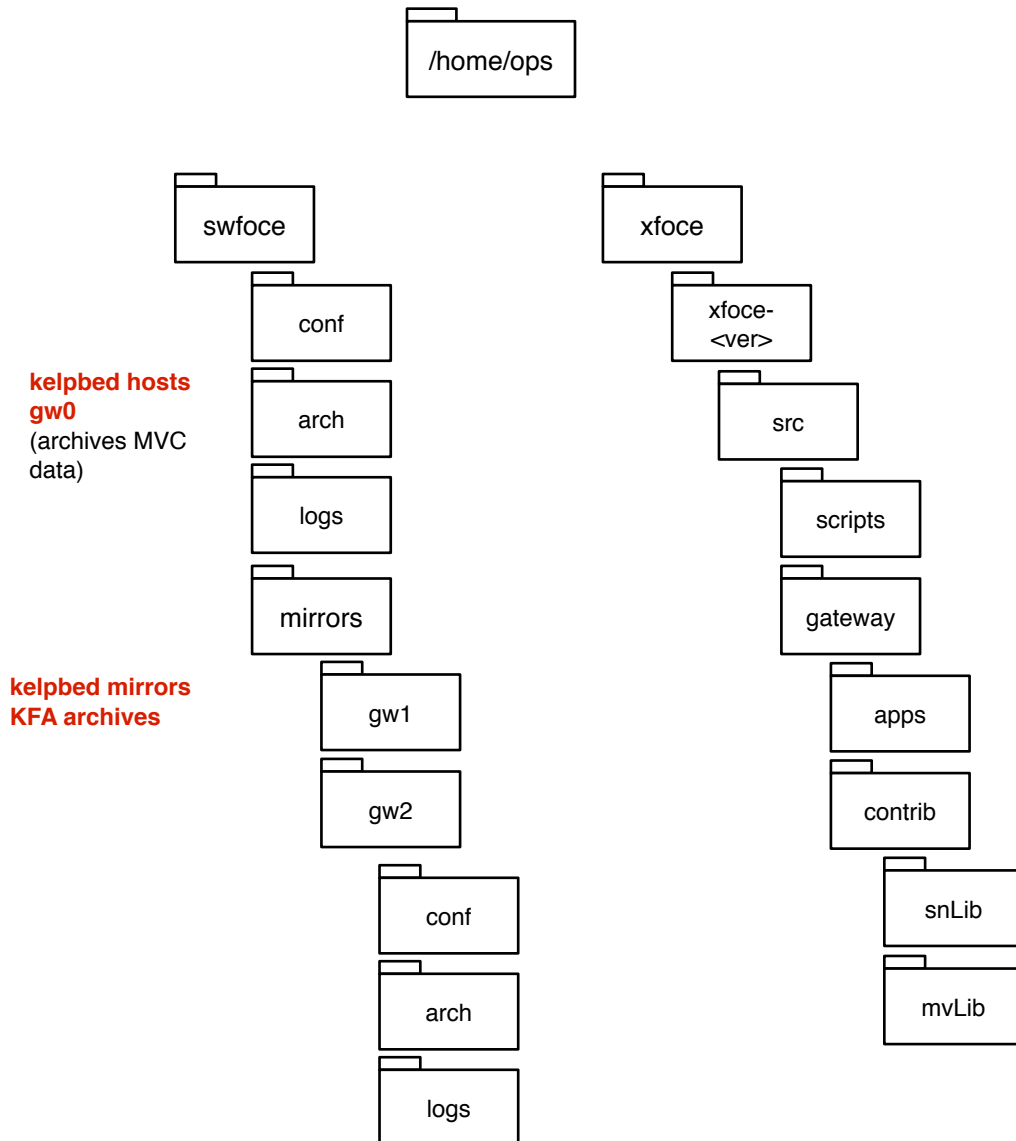
Gateway Software Utility Guide

Topology: HMS

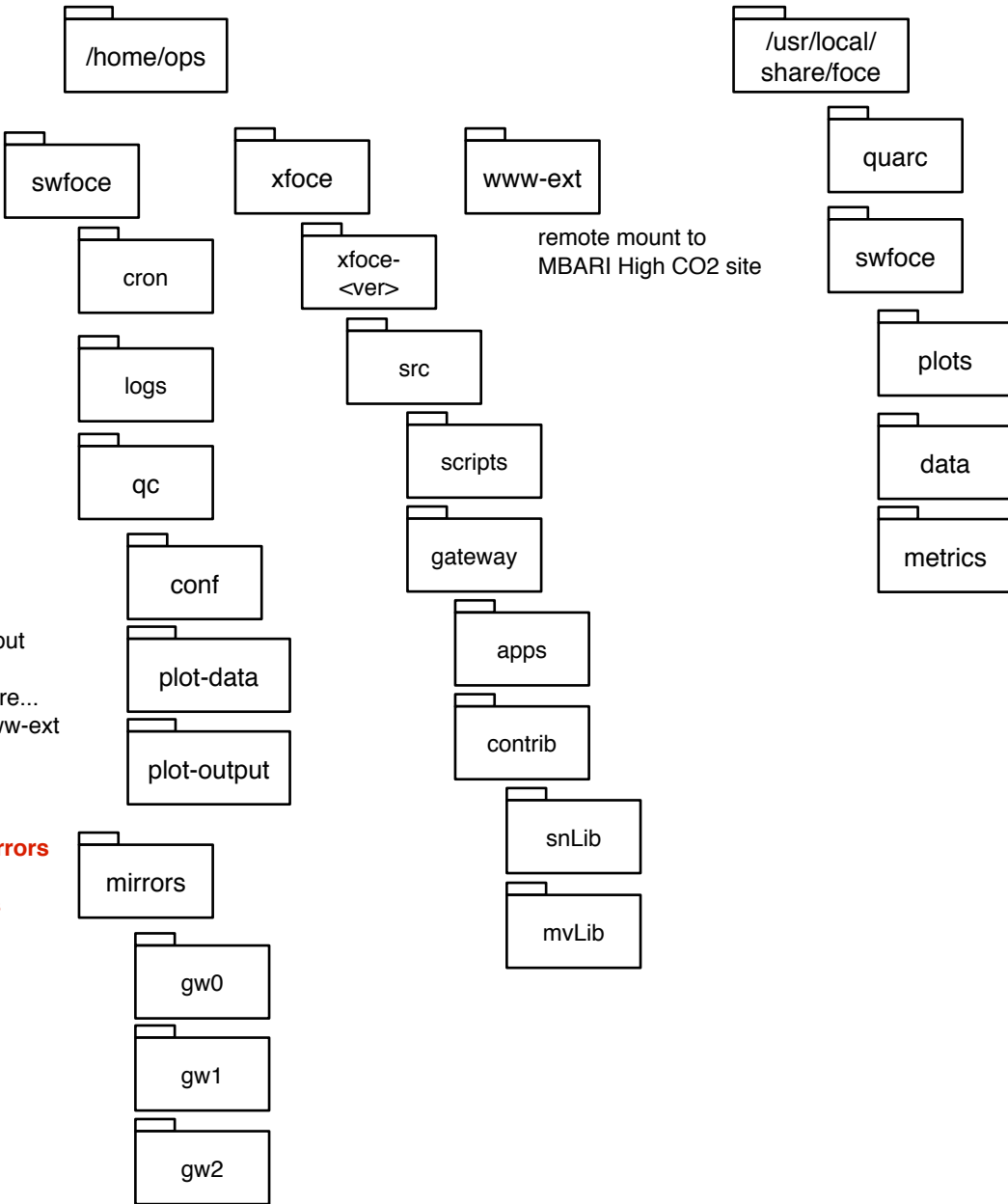
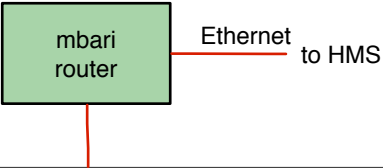
HMS
[pumphouse]



kelpbed 134.89.91.2



Gateway Software Utility Guide Topology: MBARI

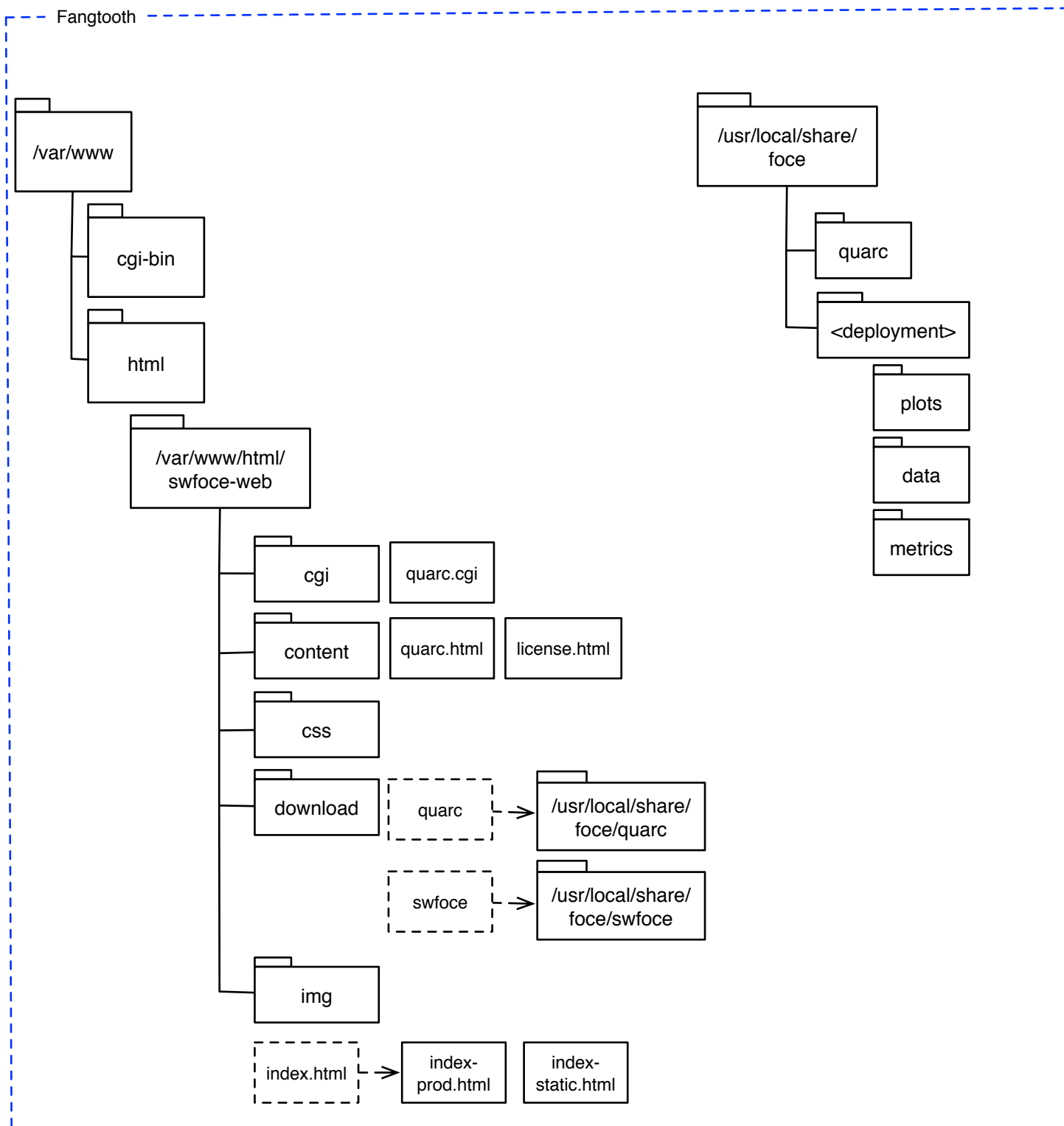
MBARI
SE Lab

```
quick plot output
copied to
/usr/local/share...
/home/ops/www-ext
```

fangtooth mirrors
HMS and
KFA archives

Gateway Software Utility Guide

Topology: swfoce-web, www



Gateway Software Utility Guide

Utility Map

Gateway Management

[xsvc](#)

Start/stop gwrn, sds, sbr
XFHOME/src/scripts

[gwstart](#)

Start gateway and related services
See also: digicfg, gwdio, gwmech, xsvc
XFHOME/src/gateway/apps

[gwstop](#)

Stop gateway and related services
See also: digicfg, gwdio, gwmech, xsvc
XFHOME/src/gateway/apps

[gq](#)

Gateway query client: get/set/do
XFHOME/src/scripts

[digicfg](#)

Configure Digi virtual serial ports
XFHOME/src/scripts

Archive and Data Tools

[xpres](#)

Archive extraction/expression
XFHOME/src/gateway/apps

[skim](#)

Present recent archive data
XFHOME/src/scripts

[gist](#)

Present archive summary metadata
XFHOME/src/gateway/apps

[qplot](#)

Plot CSV data as PNG, PDF
XFHOME/src/scripts

[qsee](#)

Use xpres and plot to generate CSV data and plot
images
XFHOME/src/scripts

Hardware Tools

(use gq when gateway is running)

[gwdio](#)

Read/set gateway DIO
XFHOME/src/gateway/apps

[gwmvc](#)

MV converter control
XFHOME/src/gateway/apps

[gwsn](#)

Sensor Node control
XFHOME/src/gateway/apps

[gwmech](#)

Main electronics housing control
XFHOME/src/gateway/apps

[gwspi](#)

Read SPI devices
XFHOME/src/gateway/apps

[moco](#)

Thruster motor control
XFHOME/src/gateway/apps

[vaco](#)

ESW valve control
XFHOME/src/gateway/apps

Gateway Software Utility Guide

Utility Map

Scripting

[xcycle-start](#)

Run/log a specified profile
XFHOME/src/scripts

[xcycle-stop](#)

Stop a specified profile
XFHOME/src/scripts

[xprofile](#)

List/run a specified profile
XFHOME/src/scripts

[qprofile](#)

Define/run a motor profile
XFHOME/src/scripts

[wuntil](#)

Wait until specified date/time
XFHOME/src/scripts

Configuration Info

[gww](#)

Show gateway software version
XFHOME/src/gateway/apps

[bbsn](#)

Show Beaglebone hardware serial number
XFHOME/src/gateway/apps

[catgpio](#)

Show Beaglebone GPIO configuration
XFHOME/src/scripts

[catsys](#)

Show results of df, ls (arch, logs) , top
XFHOME/src/scripts

Test Tools

[snTerm](#)

Sensor node terminal
XFHOME/src/gateway/apps

[snTest](#)

Sensor node control/test
XFHOME/src/gateway/apps

[mvTest](#)

MV converter control/test
XFHOME/src/gateway/apps

[subc](#)

Gateway pub/sub client
XFHOME/src/gateway/apps

[subs](#)

Pub/sub test server
XFHOME/src/gateway/apps

[vtest.sh](#)

Valgrind wrapper (Linux VM)
XFHOME/src/scripts

Software Tools

[buildenv](#)

Configure gateway software build environment
XFHOME/src/scripts

[xfenv](#)

Configure gateway environment variables
XFHOME/src/scripts

[gwsetup](#)

Configure gateway SD card
XFHOME/src/scripts

[gwcard](#)

Partition SD card , install gateway rootfs
XFHOME/src/scripts

Gateway Software Utility Guide

Utility Map

Expert Tools

[gwrn](#)

gateway main executable
XFHOME/src/gateway/apps

[sds](#)

Shared data server
XFHOME/src/gateway/apps

[sbr](#)

Gateway message handler/service broker
XFHOME/src/gateway/apps

[xrun](#)

xsvc script runner
XFHOME/src/scripts

[testdrv](#)

Gateway driver unit tests
XFHOME/src/gateway/test

[gwgpio](#)

Gateway Beaglebone GPIO control
XFHOME/src/gateway/apps

[cbit](#), [pbit](#) [rbit](#), [sbit](#), [tbit](#),

clear, pulse, read, set, toggle a Beaglebone GPIO bit
See also [gwgpio](#)
XFHOME/src/gateway/apps

[xfoce-init-digi](#)

systemd Digi virtual serial port initialization
XFHOME/src/scripts

[xfoce-init-gpio](#)

systemd GPIO initialization
XFHOME/src/scripts

[xfoce-init-autostart](#)

systemd gateway autostart service
XFHOME/src/scripts

[socat-cn](#)

SystemV socat virtual serial port initialization for central node
[/etc/init.d](#) ([XFHOME/src/gateway/rootfs/etc/init.d](#))

Utilities

[etx](#)

Timestamp converter
XFHOME/src/scripts

[undup](#)

Remove duplicate CSV records, sort file
XFHOME/src/scripts

[repotar](#)

Generate repository snapshot
XFHOME/src/scripts

[sdc](#)

Shared data client
XFHOME/src/gateway/apps

[elmostat](#)

Elmo register interpreter
XFHOME/src/gateway/apps

[archi](#)

Archive utility
XFHOME/src/gateway/apps

Gateway Basics

Gateway Environment

- XF_HOME must be set
- The following directories must be in the PATH variable
XF_HOME/src/scripts
XF_HOME/src/gateway/apps
- The environment and PATH are set up automatically when you log in as ops
- The environment should be set/changed using
source XF_HOME/src/scripts/xfhome [/path/to/new/XF_HOME]*

* default is current directory

Gateway Utility/Script Conventions

- CLI (command line interface) syntax is lower case
- get help using '-h' option
- most components support verbose output with -V (sometimes -v)

Gateway Clients

Services used to query or change the gateway are presented by the service broker (proxy). Gateway clients request operations through a service API; the operations supported include variations of:

get : request information about configuration, state, status
set : change configuration
do : request some action be taken (e.g. set actuation values)

Operations use **topics** to specify what is to be get, set or done; a topic represents a single configuration or state value, and is generally associated with one data primitive (a number or string). Semantically, a topic has a path, name, type and value; some topics may be changed (via set or do, e.g. driver properties, actuation values), and some are read-only.

Topics

Topics are organized hierarchically, and their syntax resembles that used for file system paths, e.g.

/platform/config/id
/service/status/errors.sbel8s.0

By requesting more or less of a topic path, it is possible to access groups of related information, or a single item.

In general, **topic paths are used for GET operations, while only topic names (e.g. act.dcraft.0.cspeed) are needed for SET and ACT operations.**

The primary gateway clients are **gq**, **subc** and **sdsc**. Data archive clients include **xpres** and **gist**. These and other utilities are described in below.

Gateway Utility Summary System Startup

Login

```
ssh ops@kelpbed.swfoce.mbari.org
ops:brynnner
```

for all xfoce utils, the -h option provides help

Start gateway and related services

```
gwstart -h (show all options)
```

List IDs

```
gwstart -L
```

Start up system after reboot:

```
${XFHOME}/src/scripts/gwstart -P <cfg>
```

Move (gw/sds/sbr) logs aside

(leaves logs of running components):

```
gwstart -G
```

Set sensor node power

(not allowed if gw is running):

```
gwstart -N <spec> [see also gwdio -h]
```

Start virtual serial IO:

```
${XFHOME}/src/scripts/gwstart -Ii <id>
```

gwstart - simple gateway start

sets DIO, starts virtual serial ports, sds, sbr and gw.

It uses fixed configs, referenced by name, defined in the script source.

Stop gateway and related services

```
gwstop -h (show all options)
```

Prepare for power cycle

[WARNING: requires manual power cycle]:

```
${XFHOME}/src/scripts/gwstop -P
```

Stop gateway, leave sensor nodes powered:

```
gwstop -S
```

Reboot gateway:

```
${XFHOME}/src/scripts/gwstop -R
```

Stop virtual serial port:

```
gwstop -Ii <id>
```

Gateway Utility Summary System Startup

Start gateway and related services

```
xsvc show
xsvc -l <logdir> start sds -p
xsvc stop sds
xsvc -l <logdir> start sbr
xsvc stop sbr
xsvc -l <logs> start gw -f <config> -a <arch>
xsvc stop gw
```

Start sds and sbr before starting the gateway

ads, sbr may be left running between gateway restarts, and rarely need to be restarted

Start/Stop Digi Ports

Start Digi port
sudo src/scripts/xfoce-init-digi -I

Stop Digi port
sudo src/scripts/xfoce-init-digi -U

Call Digi config script directly
sudo digicfg -i ME -a <host> -c [init|uninit]

Start/Stop socat Ports

```
sudo /etc/init.d socat-cn [start|stop]
```

id must match name specified in gateway config file (e.g. /dev/tty**ME**00)

Kill an unresponsive app

```
ps -ef|grep <app>
Or
ps -C <app> -o pid= -o args=
```

Then

```
kill -9 <pid>
Or
sudo kill -9 <pid>
```

Gateway Utility Summary Inspect System

Examine gateway status and configuration

```
status          : gq -q svstat
properties      : gq -q svprop
list topics     : gq -q topics
IO properties   : gq -q ioprop
Sampling Properties: gq -q saprop
Debug Settings  : gq -q debug
topics (all)    : gq /
topic (specified: : gq <topic> [<topic>...]
```

Examine gateway system logs

```
less logs/gw.log
tail -f logs/gw.log
less logs/gateway_nnnn.log
```

Examine latest gateway data logs (see xpres)

```
skim -m <t> -a <arch> -s <id>
skimm [-H h] [-M m] [-S s] -a <arch> -s <id>[,<id>...]
gist -a <arch> -s <id>
```

enable/disable module debug (gw.log)

```
gq -s debug_level.<module>=<level>
```

module e.g. SBE18S
DEBUG, INFO, WARN, ERROR, FATAL

Stream pubsub data

```
show all       : subc
show sbe*      : subc -t /data/sbe
show sbe18s.0  : subc -t /data/sbe18s.0
show remote    : subc -a <addr>...
show sds topic : sdc -t /sdata/tref
```

pubsub and sds
topics are defined
in the **streams**
section of the
gateway config file

Check Gateway version

```
gww
```

When the gateway is running, use the gq interface to interact with devices. Otherwise, use the CLI utils.

Gateway Utility Summary Actuate Devices (via gw)

Set MVC port power

```
gq -d act.sysmvc.0.set=<port>:H|L
```

```
0:vic1 1:vic2 2:gw1 3:gw2  
4:fan1 5:fan2
```

Set MVC ground fault

```
gq -d act.sysmvc.0.gfmode=<mode>
```

```
GFC_NORM, GFC_OFF, GFC_TEST,  
GFC_ALL
```

Set pH cal coefficient*

```
gq -s property.sbel8s.0.cal_k=<val>  
gq -s property.sbel8s.0.cal_slo=<val>  
gq -s property.sbel8s.0.cal_off=<val>  
gq -s property.sbel8s.0.cal_cor=<val>
```

* set properties in gateway config file to persist across gateway starts.

Set MEH relay

```
gq -d act.sysmeh.0.set=<rel>:H|L
```

Set sensor node power

```
gq -d act.sysgw.0.set=<node>:H|L
```

Set sensor node port power

```
gq -d act.syssn.0.set=<node>:<port>:H|L
```

Set thruster speed

```
gq -d act.dcraft.0.cspeed=<counts>
```

```
motor speed range: [-1024:1024]
```

Set ESW valve position (abs,rel)

```
gq -d act.eswv.0.set=<valve>:<pos>  
gq -d act.eswv.0.setr=<valve>:<pos>
```

```
pos range: [0.0-1.0]
```

Gateway Utility Summary Actuate Devices (via CLI)

<dev>: device, e.g., /dev/tty01
<term>: terminal e.g., speed[,par[,csize[,stopb]]]
TTY : -T <dev>[,<term>]
SNPORT: -S <0:3>[,<dev>[,<term>]] -p <port>[,<term>]

Set MVC port power

```
gwmvc TTY|SNPORT -H <port,...> -L <port,...>
gwmvc TTY|SNPORT -b
gwmvc TTY|SNPORT -a <n,t> -g <i,n,t>
gwmvc TTY|SNPORT -G <mode>
gwmvc TTY|SNPORT -q -i
```

<ports>:
0:vic1 1:vic2 2:gw1
3:gw2
4:fan1 5:fan2

<mode>: GFC_NORM,
GFC_OFF, GFC_TEST,
GFC_ALL

Set sensor node port power

```
gwsn TTY|SNPORT -H <port,...> -L <port,...>
gwsn TTY|SNPORT -b
gwsn TTY|SNPORT -q -i
```

Set MEH relay port

```
gwmvc TTY|SNPORT -s <port>:[H|L],...
gwmvc TTY|SNPORT -q
```

Set sensor node power

```
gwdio -s <port>|all:[H|L],...
gwdio -r <port,...>|all
```

Control Valves (CLI)

```
vaco TTY|SNPORT -o <valve>,<pos>
vaco TTY|SNPORT -c <valve>,<range=0|1>,<counts>
vaco TTY|SNPORT -q
```

Control Motors (CLI)

```
moco TTY|SNPORT -c <counts>
moco TTY|SNPORT -p <port>[,<term>] -c <counts>
moco TTY|SNPORT -q
```

Express all/part of data archive

```
xpres -d arch -s 1812 -b 2014/06/01T12:00 -e 2014/06/02T12:00
xpres -d arch -s 1812 -B 1394000000 -B 1394012000
skim -M 30 -d arch -s 1812
etx [-u] 1394000000
date --date="2014/03/05T06:13" +%s
date --date='@1394000000'
```

Examine latest gateway data logs (see xpres)

```
skim [-H h] [-M m] [-S s] -a <arch> -s <id>
skimm [-H h] [-M m] [-S s] -a <arch> -s <id>[,<id>...]
gist -a <arch> -s <id>
```

Examine archive summary

```
gist -a <arch> -s <id>
```

Convert Timestamp

```
etx [-u] 1394000000
date --date="2014/03/05T06:13" +%s
date --date='@1394000000'
```

Get most recent timestamp in archive

```
archi -a <arch> -s <id> -n <device_name>
```

Gateway: Data Archive

Where is the data?

Each instrument or other data source stores its data in a single archive. Archives are kept on the gateway in `XF_HOME/arch` by default.

Each archive consists of one or more files, segmented by size; the segment size may be configured per device in the gateway configuration file. Each log segment consists of a data (.dat) and index (.idx) pair; the index enables fast searches on large archives.

Archive file names reflect the device type, instance, source ID, and archive segment, e.g. `sbcl8s_1_1813_0000.dat`. Do not rename archive files.

In some configurations, the data may also be mirrored to other locations using `rsync` or similar utilities.

How is data stored?

An archive data file (.dat) consists of a binary file header and one or more data records.

Data records consists of a binary record header, plus the record data; record data are stored in their original format (i.e., as originated from the device). The binary headers contains metadata that enable searches, diagnostics, and repairs e.g. `stream_id`, sequence number, data size, etc. Similarly, the index files contain information about segment content.

Since there is one archive per device, the archive may contain multiple streams (record types). Each stream ID corresponds to a specific data format. This is somewhat device-dependent (i.e., up to the driver developer), but best practice is to assign a unique stream ID to records with different number, type and/or order of fields. As a guideline, any data that may be handled transparently by any application may use the same stream ID.

stream ID 0-99 are reserved.

stream ID 0 : metadata record

stream ID 99 : error record

Device data stream ID are defined in `xf-id.h`. Devices may use non-reserved stream IDs as they wish.

Different device **types** (e.g. `sbcl8s` and `sbe52`) may use the same stream ID for different purposes.

Different device **instances** (`sbcl8s.0` and `sbcl8s.1`) must use the same stream ID for the same purpose.

See also:

`src/gateway/core/xf-log.c`

`src/include/xf-log.h`

Gateway: Data Archive

How do I see the data?

Archive contents are viewed using a utility called **xpres**. xpres enables flexible, formatted presentation of all or part of an archive.

For example, it is possible to filter by time and record type. It is also possible to include or exclude various parts of the archive metadata, and offers some control over representation of binary data.

A script named **skim** uses xpres to dump the latest data in the archive. Specify the amount of data in hours, minutes, and/or seconds, and optionally use an xpress config file to format the output.

A utility named **gist** provides an overview of an archive. This can be useful for finding the archive extents (start/end times, number of records...), for example.

XPRES Overview

```
xpres -d arch -s <source ID> ...
```

There are lots of options:

- start, end times (date or epoch time)
- stream ID
- skip modulo n
- formatted output
- Automate complex queries using config file:

```
xpres [options] < myquery.conf
```

See Also: Detailed use info below

xpres Options

Usage: xpres [options...] [< config_file]

options:

-s: source ID
-t: stream ID
-b: begin time (YYYY/MM/DDTHH:MM:SS)
-e: end time (YYYY/MM/DDTHH:MM:SS)
-B: begin time (epoch sec)
-E: end time (epoch sec)
-a: archive directory
-n: skip interval
-o: output format
 e.g., -o <source>,<stream>,"<format specifiers>"[,<mnemonics>...]
 where
 <source>,<stream> : source or stream ID or -1 for all
 <format specifiers> : printf specifiers
 <mnemonics> : mnemonics for data components (see below)

Note:

Quotation marks and escaped characters on the command line must be double escaped:

xpres -a arch -s 300 -o 300,-1,\"%lld,%lld\\n\",ets,seq

-F: output time format (see strftime, strptime)

-D: debug output

-V: verbose output

-h: print help message

Data Component Mnemonics:

ets: epoch time msec	[%llu]	
etm: epoch time sec	[%llu]	
etf: epoch time fraction	[%d]	(3-digit msec)
ftu: local time/date	[%s]	(specified by time format using -f option; see strftime)
ftl: UTC time/date	[%s]	(specified by time format using -f option; see strftime)
utm: UTC time/date	[%s]	(specified by time format using -f option; see strftime)
asc: data buffer ASCII	[%s]	
pas: data buffer ASCII	[%s]	(printable ASCII only, ignore control characters)
hex: data buffer hex	[%s]	
bin: data buffer binary	[%s]	
src: source ID	[%u]	
pid: platform ID	[%u]	
str: stream ID	[%u]	
stn: stream ID name	[%u]	
seq: sequence number	[%llu]	
dln: data length	[%llu]	
dsz: data record size	[%llu]	

xpres Examples

xpres Examples

```
# Dump entire log
xpres -d arch -s 2001
1394576574919,2001,301,-4.1415,277.1500,0.0000**
1394576577169,2001,301,-4.1415,277.1500,0.0000**
...

# Default, specify time window
xpres -d arch -s 2001 -b 2014/03/11T23 -e 2014/03/11T23:01
1394578801253,2001,301,-4.1419,277.1500,0.0000**
1394578803756,2001,301,-4.1417,277.1500,0.0000**
...
```

Using xpres config files

```
# xpres.conf [may use any valid file name/extension]
-d arch
-s 2001
-n 1
-v
-o 2001,-1,"%llu,%d,%s\n",ets,dln,asc
-o 2001,301,"%s,%llu,%llu,%s\n",utm,ets,dln,asc
-o -1,-1,"%c %llu.%llu,%d,%s,%s\n",ets,etm,dln,hex
-o 999,-1,"%b %llu.%llu,%d,%s\n",ets,etm,dln,bin
-f "%Y-%m-%d %H.%M.%S"
```

Run using options file

```
xpres < conf/lshow_example.conf
2014-03-11 22.22.54,1394576574,25,-4.1415,277.1500,0.0000**
2014-03-11 22.22.57,1394576577,25,-4.1415,277.1500,0.0000**
...
```

override the source

```
# note that it uses the default format, since no format
# for the specified source is defined in the config file
```

```
xpres -s 5001 < conf/lshow_example.conf
1395095518326,5001,701, 0.7024, 22.2956, 0.08, 5.260****
1395095523436,5001,701, 0.7025, 22.2965, 0.09, 4.519****
...
```

skim Examples

skim Examples

Show help message

\$ skim -h

usage: skim

-H <hour> : hours [0]

-M <min> : minutes [0]

-S <sec> : seconds [0]

-s <id> : stream id

-d <arch> : arch directory [/home/headley/host/hg/xfoce-31mar2014/xfoce/arch]

-c <cfg> : config file

-v : verbose output [FALSE]

-h : print this help message

Show last 123 h 48 m of data of source 1812

\$ skim -H 123 -M 48 -s 1812

1401843568881,1812,301,0.0000,2.6751,0.0000,0.0000**

1401843574900,1812,301,0.0000,2.6862,0.0000,0.0000**

1401843580925,1812,301,0.0000,2.6969,0.0000,0.0000**

...

Show last 24 h of data of source 1812, use config file

\$ skim -H 24 -s 1812 -c mycfg

2014/07/21T12:34 0.0000,2.6751,0.0000,0.0000

2014/07/21T12:34 0.0000,2.6862,0.0000,0.0000

2014/07/21T12:34 0.0000,2.6969,0.0000,0.0000

...

gist Examples

gist Examples

Show help message

\$ gist -h

Usage: gist [-ahst]

options:

-a: archive directory

-h: print help message

-s: source ID

-t: stream ID

Show log summary

\$ gist -a arch/ -s 1649

source ID:[1649] segments:[1] first:[0000] last:[0000]

[path : arch/sbe52mp_0_1649_0000.idx]

[platform : 1001]

[source : 1649]

[min_key : 1401843751183]

[max_key : 1401843939336]

[records : 002]

[start date : 2014/06/04T01:02:31]

[end date : 2014/06/04T01:05:39]

Plot Overview

A pair of utility scripts, **qplot** and **qsee**, are provided to generate simple PDF plots from delimited data:

qplot

generates a plot from one or more data files

qsee

automates plot generation, creating data files using **skim**, and invoking **qplot**

Features

- output in PDF and Postscript formats
- use command line and/or config file to specify options
- useful for data quick-look data and performance metrics plots

qplot exposes only a small fraction of the gnuplot API; it is intended for simple diagnostic applications. **qplot** and **qsee** are written in bash, and may be freely modified.

Requirements

bash

GNU plotutils package (<http://www.gnu.org/software/plotutils/>)

There are potentially many, long command line options for **qplot** and **qsee**; place command line options in a text file and read using file redirection or pipes:

```
qplot < myplot.conf
cat myplot.conf | qplot
```

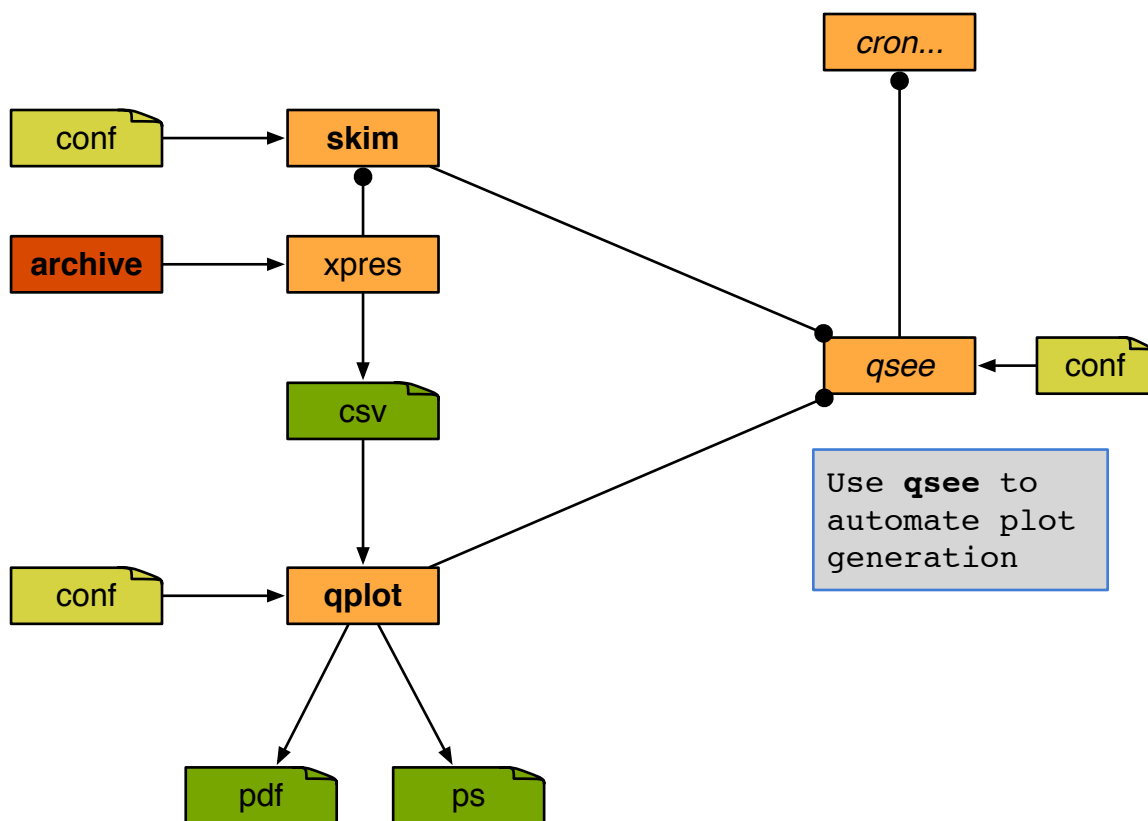
- The files may include one option/value pair per line
- Command line options override config file
- Lines beginning with '#' are ignored, making it possible to include comments or disable options without deleting them

See Also

sample config files in src/scripts/qsee

Plot Process

Use **skim**/xpres and **qplot** to manually generate plots



qplot

```
$ qplot -h
```

qplot: Generate plots from one or more delimited text files

usage: qplot [options] file...

Options:

-f <path>	: config file path	[]
-j <job>	: job name (select/order jobs)	
-a <dir>	: archive directory	[./plot-data]
-o <ofile>	: output file	[qplot-18591]
-O <dir>	: output directory	[./plot-output]
-r <min Y>	: y-range min	[*]
-R <max Y>	: y-range max	[*]
-i <min X>	: x-range min	[*]
-I <max X>	: x-range max	[*]
-p <ptype>	: point type	[7]
-P <psize>	: point size	[0.3]
-t <fmt>	: time input format	[%s]
-x <fmt>	: x axis (time) format	[%Y-%m-%d %H:%M:%S]
-s <sep>	: data separator	[,]
-D	: debug (keep temporary files)	[N]
-V	: verbose output	[N]
-h	: print this help message	

Examples:

```
qplot -DVf /path/to/my_plot_cfg -j my_job
```

qplot output is written to output directory

qplot Example

Here is skim/xpres configuration.

- specifies formats for 3 devices (1812, 1813, 1649)
- record format: timestamp, data_size, data(printable ascii)
- uses time format "%s" (epoch seconds)

xp-qfoce.conf

```
-a /path/to/archive
-o 1812,-1,"%llu,%d,%s\n",ets,dln,pas
-o 1813,301,"%llu,%d,%s\n",ets,dln,pas
-o 1649,701,"%llu,%d,%s\n",ets,dln,pas
-f "%s"
```

xpres -o option details

source ID stream ID (-1: default) printf format... ...element names (see xpres help)

-o 1812,-1,"%llu,%d,%s\n",ets,dln,pas

Generate CSV data for qplot (last 24 hour) for two devices:

```
cd $XF_HOME/tmp
mkdir qsout
skim -H 24 -s 1812 -t 301 -f xp-qfoce.conf > qsout/1812-301.dat
skim -H 24 -s 1813 -t 301 -f xp-qfoce.conf > qsout/1813-301.dat
```

Here is a qplot configuration:

qp-ph.conf

```
#-d
#-V
-o ph-qplot
-T "pH vs Time"
-X "Timestamp (epoch msec)"
-Y "pH"
-t "%s"
-y qsout/1812-301.dat,1,1,3,"1812-ph"
-y qsout/1813-301.dat,1,1,3,"1813-ph"
```

Use col 1 for X axis data

plot 1 field, col 3 show as "1812-ph" (may use multiple Y columns)

Multiple -y options allowed

qplot Example

Example qplot configuration:

```
#!/bin/bash

# source ID and label mapping
# [defines associative arrays SRC_ID, STR_ID, SRC_LABELS]
. ${QS_CONF_DIR:-"."}/gw-id.conf

# env config
export QP_VERBOSE=${QP_VERBOSE:-"N"}
export QP_DEBUG=${QP_DEBUG:-"N"}
#export QP_OUTPUT_DIR_DFL="./plot-output"
#export QP_ARCH_DIR_DFL="./plot-data"

# job definitions
# - Use QP_JOB_ORDER or -j option to select/order jobs

# - may combine several plots in one job:
#   QP_JOBS[${#QP_JOBS[*]}]="plot,png,par+ctdo"

# - set individual jobs:
QP_JOB_DEFS["ph"]="plot,png,ph"
QP_JOB_DEFS["phv"]="plot,png,phv"
QP_JOB_DEFS["comb-all"]="combine,png,${HOME}/swfoce/qc/plot-output/qp-gw2-sci.pdf,${HOME}/swfoce/qc/plot-output/sc2*png"

# May also define combine jobs in separate file(s)

# use QP_JOB_ORDER to set default job order
# [if not specified on command line w/ -j option]
QP_JOB_ORDER[${#QP_JOB_ORDER[*]}]="ph"
QP_JOB_ORDER[${#QP_JOB_ORDER[*]}]="comb-all"

#QP_KEY_FONT["key"]="arial"
#QP_KEY_SIZE["key"]="9"
#QP_TERM_FONT["key"]="arial"
#QP_TERM_SIZE["key"]="10"
#QP_TERM_OSIZE["key"]="800,600"

# plot job: "ph" definition
QP_OFILE_NAME["ph"]="sc20-ph"
QP_PTITLE["ph"]="pH vs Time"
QP_XTITLE["ph"]="Time (PST)"
QP_YTITLE["ph"]="pH"
QP_YRANGE_MIN["ph"]=6.5
QP_YRANGE_MAX["ph"]=8.5
#QP_XRANGE_MIN["ph"]=0
#QP_XRANGE_MAX["ph"]=10
QP_TFMT["ph"]="%Y-%m-%d %H:%M:%S"
#QP_XTFMT["ph"]="%Y-%m-%d %H:%M:%S"
QP_DSEP["ph"]=","
QP_PLOT_SPECS["ph"]="ph20.csv,1,1,3,${SRC_LABELS[ph20]}"
QP_PLOT_SPECS["ph"]+="+ph21.csv,1,1,3,${SRC_LABELS[ph21]}"

# other job defs...
```

Use col 1 for X axis data

plot 1 field, col 3 label defined by SRC_LABELS[ph20]

Multiple plot spec allowed

Generate the plot:

```
qplot -f qp-ph.conf
```

output:

- plot-output/ph-qplot.pdf
- plot-output/ph-qplot.ps