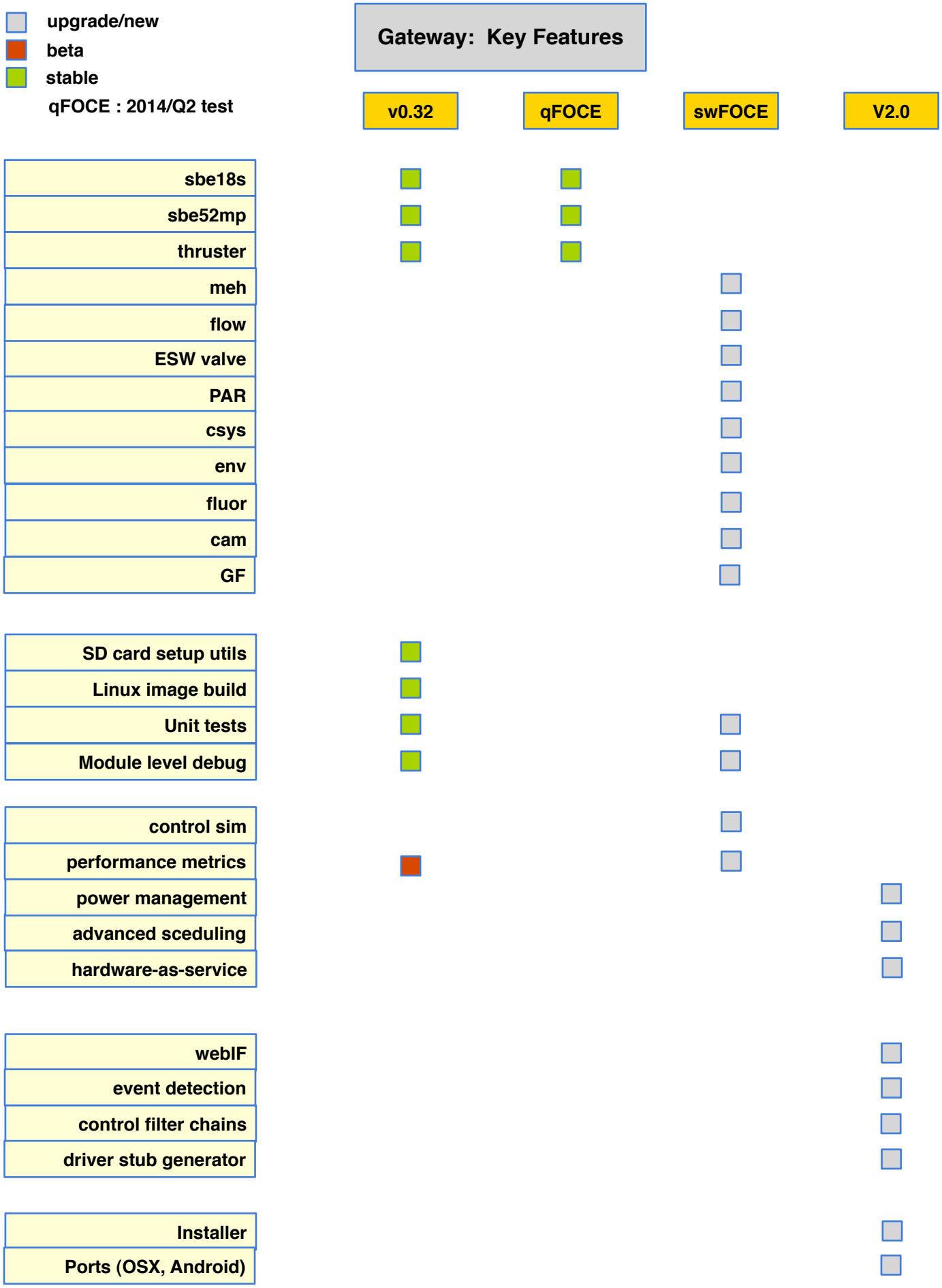
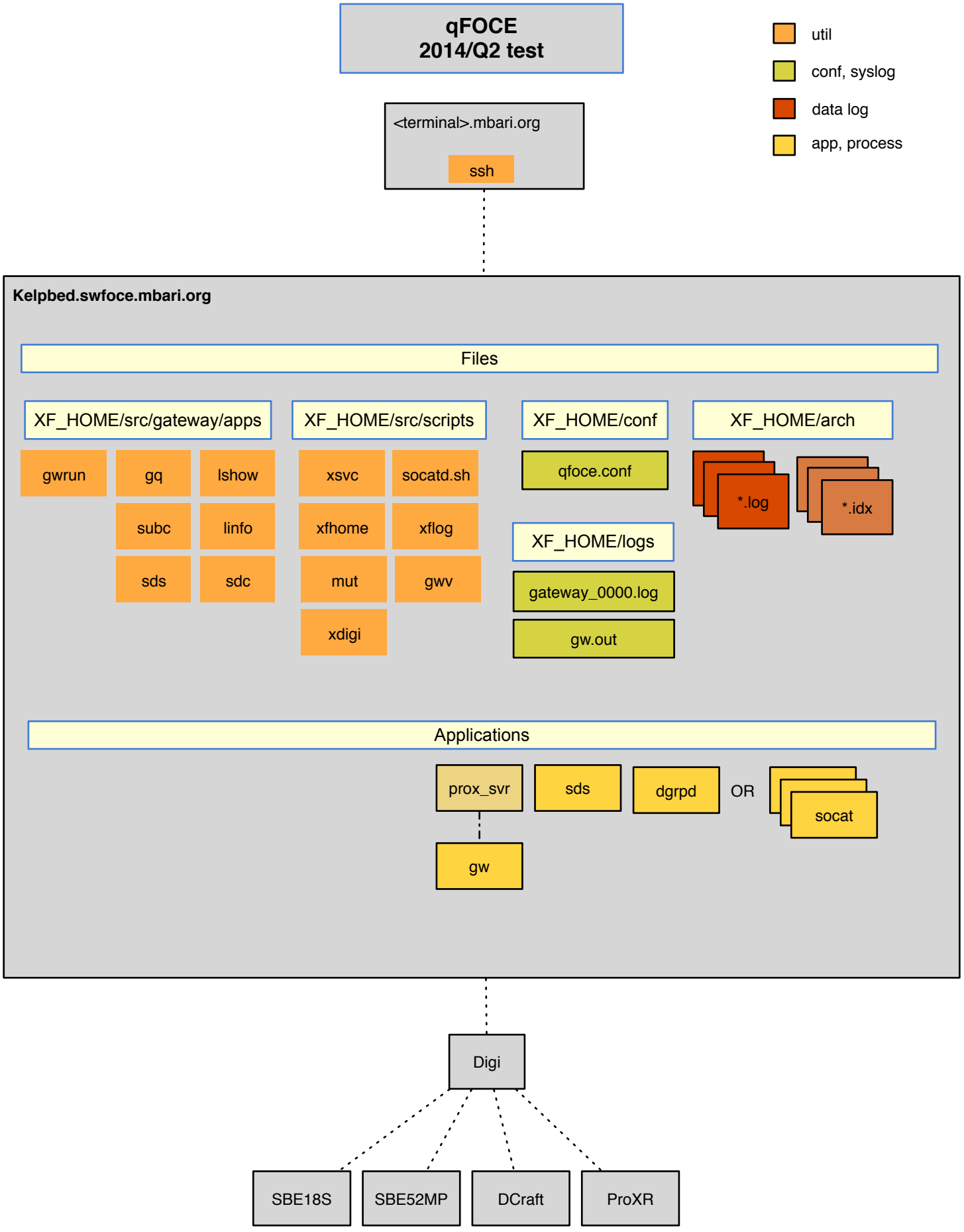
Gateway Software Status Update 21 Mar 2014
Contents

Gateway Software Status
Executive Summary
21 Mar 2014

```
Gateway Software Status
Executive Summary
25 Mar 2014

- The following configuration exists, and is probably sufficient for qFOCE:
  - gateway v0.32 running on Ubuntu 12.04 VM (analog for CentOS 6 @ Hopkins)
  - Four serial ports via Digi TS4-MEI in RealPort Mode (same as at Hopkins)
  - Devices:
    - pH      : SBE18S
    - CTD     : SBE52MP
    - Thruster: Devicecraft motor controller
    - Relays  : ProXR (not used for qFOCE)
  - Data logged to gateway local disk
    - plan to rsync to MBARI
  - Control and data access via command line interface (CLI)
    - human readable, but maybe arcane for science users
    - can write wrappers and formatters, but need input on what users need/want
  - Shared Data (req/res) and PubSub via messaging clients
    - needs better output formatting
    - needs support for more fields (easy)
  - pH service using shared CTD temp
  - CTD pump runs continuously (shuts down briefly after each sample)
  - Can get(all)/set(many) key configuration parameters
  - No closed loop control
  - Builds cleanly, no known memory leaks or major bugs

- Key features still ahead on the roadmap
  - Closed loop control
    - plan is to integrate control system as another gateway service
    - data streaming potentially enables an alternative distributed control
      implementation: a remote client could use the message bus to receive
      data and send actuation commands
  - Data access tools (GUI, CLI formatters/wrappers)
  - Archive duplication, SSDS ingest
  - Service control (start, stop, restart...)
  - Web clients?

- Top priorities
  - user interfaces
  - concurrency (arbitrating hardware access)
  - closed loop control
  - integration and test
  - driver development
  - any memory leaks (none pending)
  - any bugs (none pending)
```


upgrade/new
beta
stable
qFOCE : 2014/Q2 test
Gateway: Key Features
v0.32
qFOCE
swFOCE
V2.0
periodic sampling
segmented binary logs
gateway logs
log formatter/converter
stream log/pub/parse/post
CLI data streamer
pubsub cli/svr
shared data cli/svr
CLI get/set service props
File configuration
CLI actuation
virtual devices (e.g. csys)
S2S data/control
CLI service status util
Power switching
rsync logs
SSDS ingest
improved CLI wrappers
GUI actuation
GUI data streamer
GUI get/set service props
OSDT streamer
log check/repair util
log splitter util
termio
snlib
TCP/Serial (fixed term)
dio
aio
spi
http


upgrade/new
beta
stable
qFOCE : 2014/Q2 test
Gateway: Key Features
v0.32
qFOCE
swFOCE
V2.0
sbe18s
sbe52mp
thruster
meh
flow
ESW valve
PAR
csys
env
fluor
cam
GF
SD card setup utils
Linux image build
Unit tests
Module level debug
control sim
performance metrics
power management
advanced sceduling
hardware-as-service
webIF
event detection
control filter chains
driver stub generator
Installer
Ports (OSX, Android)


qFOCE
2014/Q2 test
util
conf, syslog
data log
app, process
<terminal>.mbari.org
ssh
Kelpbed.swfoce.mbari.org
Files
XF_HOME/src/gateway/apps
gwrun
gq
lshow
subc
linfo
sds
sdc
XF_HOME/src/scripts
xsvc
socatd.sh
xfhome
xflog
mut
gwv
xdigi
XF_HOME/conf
qfoce.conf
XF_HOME/logs
gateway_0000.log
gw.out
XF_HOME/arch
*.log
*.idx
Applications
prox_svr
gw
sds
dgrpd
OR
socat
Digi
SBE18S
SBE52MP
DCraft
ProXR

Appendices

Using FOCE gateway: data and operations

Good To Know	basic Gateway concepts
Sys HOWTO	system administration
Inf HOWTO	data stream operations
Log HOWTO	data log operations
Data Samples	data formats

Good To Know

```
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
```

```
Gateway Utility/Script Conventions

- CLI syntax is lower case
- get help using '-h' option
```

```
Gateway Proxy

Each gateway has a proxy that is used to query or change the gateway.
A proxy supports three basic operations:

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

These functions operate on topics; a topic represents a single configuration or state
value. Semantically, a topic has a path, name, type and value; some topics may be set
(e.g. driver properties, actuation values), and some are read-only.

Topics are organized much like a file system. They are referenced using a syntax much
like that used for files e.g.

  /platform/config/id
  /service/status/errors.sbe18s.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 are
needed for SET and DO operations.

The gq utility is a CLI gateway proxy client. Use

  gq -q topics,/

to see a list of all topics published by the gateway
```

Gateway: System HOWTO

Login

```
ssh ops@kelpbed.swfoce.mbari.org
ops:brynner!
goxf
```

See what's running

```
xsvc show
socatd.sh -c show
```

Examine gateway system logs

```
use 'less' or 'xflog'
xflog logs/gw.out
xflog logs/gateway_nnnn.out
:q! to exit xflog
```

Start/Stop SDS

```
xsvc start sds -p
xsvc stop sds
```

Start/Stop Digi Ports

```
sudo socatd.sh -H <host> -P <port> -l <ptyN> -b -c start
sudo socatd.sh -l <ptyN> -c stop
```

Start/Stop Gateway

```
xsvc start gw [-f <config_file>]
xsvc stop gw
```

Start the sds server before starting the gateway

If socat is being used (instead of RealPort) to control Digi devices, start socat instances before starting the gateway

Kill an app that doesn't stop

```
  ps -ef|grep <app>
Or
  ps -C <app> -o pid= -o args=
Then
  kill -9 <pid>
Or
  sudo kill -9 <pid>
```

Gateway: System HOWTO

Control Motors (no gateway)

```
moco -c <counts> -t <dev>[,<speed>,<par>,<dbit>,<sbit>]
```

Check Gateway version

```
gwv
```

Gateway: Info HOWTO

Set thruster speed

```
gq -d act.dcraft.0.cspeed=100
```

Set pH cal coefficient

```
gq -s property.sbe18s.0.cal_k=<val>
gq -s property.sbe18s.0.cal_slo=<val>
gq -s property.sbe18s.0.cal_off=<val>
gq -s property.sbe18s.0.cal_cor=<val>
```

Service Queries

```
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
```

Stream pubsub data

```
subc
subc -t ph
subc -t /sbe18s.0/ctd
```

pubsub and sds topics are defined in the gateway config file

Stream sds data

```
sdc -t /sdata/tref
```

List all topics

```
gq -q topics
```

Show all topic values

```
gq /
```

Show specified topic value(s)

```
gq <topic> [<topic>...]
```

Gateway: Data Log HOWTO

View Logs

lshow -d arch -s <source ID> ...

There are lots of options:
- **start, end times (date or epoch time)**
- **stream ID**
- **skip modulo n**
- **formatted output**
- **Can store complex query in text files (may combine with command line parameters)**

lshow < myquery.txt

```
Usage: lshow [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)
  -d: log 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:
      lshow -d 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:
    etm: epoch time msec        [%llu]
    ets: epoch time sec         [%llu]
    utm: UTC time/date          [%s]   (specified by time format using -f option
    asc: data buffer ASCII      [%s]
    hex: data buffer hex        [%s]
    bin: data buffer binary     [%s]
    uue: data buffer uuencoded  [%s]   (not implemented)
    pda: parsed data            [%s]   (not implemented)
    mdc: metadata cause         [%s]   (not implemented)
    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]
    mdr: metadata ref           [%llu] (not implemented)
    pst: statistics             [%s]   (not implemented)
```

```
Examples

# Show log summary
linf -d arch -s 2001
source ID:[2001] segments:[1] first:[0000] last:[0000]
source ID:[2001] segments:[1] first:[0000] last:[0000]
start date:2014/03/11T22:22:54
end date  :2014/03/18T03:52:59
lchdr[len:010 plt:1000 src:2001 fmt:01 del:0xBEEF path:arch/sbe18s_0_2001_0000.dat]
ihdr[len:036 plt:1000 src:2001 min:1394576574919 max:1395114779924 kmsid:301 del:
0xDEAD n:11739 path:arch/sbe18s_0_2001_0000.idx]

# Dump entire log
lshow -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
lshow -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**
...
```

```
Examples

# Using lshow config file

# lshow.conf
# [may use any name]
-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

lshow < 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

lshow -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****
...
```

Gateway: Data Samples

Pub/Sub (streaming)

```
$ subc -t ph

[key:/data/sbe18s.0/ph][seq:2417]
[source:0/2001/301][size:31]
[TTL=42;]
[time:1395765179]
[body:5.8093,2.3319,296.8978,0.0000x00x00]

[key:/data/sbe18s.0/ph][seq:2419]
[source:0/2001/301][size:31]
[TTL=42;]
[time:1395765181]
[body:5.7965,2.3335,292.1500,0.0000x00x00]

[key:/data/sbe18s.0/ph][seq:2422]
[source:0/2001/301][size:31]
[TTL=42;]
[time:1395765184]
[body:5.8153,2.3334,296.9029,0.0000x00x00]
```

Log (format varies)

```
# Using config file
lshow < 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**
...


# Default, specify time window
lshow -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**
```

Req/Res

```
$ sdc -t /sdata/tref

[sdata:0x8c2fa98]
[platform:0]
[source:5001]
[stream:702]
[time:1395765116483]
[ttl:-1]
[len:9]
[data:0x8c2fab8: 23.7453x00]

[sdata:0x8c2fa98]
[platform:0]
[source:5001]
[stream:702]
[time:1395765124051]
[ttl:-1]
[len:9]
[data:0x8c2fab8: 23.7437x00]

[sdata:0x8c2fa98]
[platform:0]
[source:5001]
[stream:702]
[time:1395765124051]
[ttl:-1]
[len:9]
[data:0x8c2fab8: 23.7437x00]
```