

RASC

Building the C++ Broker and Client Libraries

The root directory for the C++ distribution is named `qpidd-0.4`. The README file in that directory gives instructions for building the broker and client libraries. In most cases you will do the following:

```
[qpidd-0.4]$ ./configure}}
[qpidd-0.4]$ make
```

Running the C++ Broker

Once you have built the broker and client libraries, you can start the broker from the command line:

```
[qpidd-0.4]$ src/qpidd
```

Use the `--daemon` option to run the broker as a daemon process:

```
[qpidd-0.4]$ src/qpidd --daemon
```

You can stop a running daemon with the `--quit` option:

```
[qpidd-0.4]$ src/qpidd --quit
```

You can see all available options with the `--help` option

```
[qpidd-0.4]$ src/qpidd --help
```

Most common questions getting qpidd running

Error when starting broker: "no data directory"

The qpidd broker requires you to set a data directory or specify `--no-data-dir` (see help for more details). The data directory is used for the journal, so it is important when reliability counts. Make sure your process has write permission to the data directory.

The default location is

```
/lib/var/qpidd
```

An alternate location can be set with `--data-dir`

Error when starting broker: "that process is locked"

Note that when qpidd starts it creates a lock file in data directory are being used. If you have a un-controlled exit, please mail the trace from the core to the dev@qpidd.apache.org mailing list. To clear the lock run

```
./qpidd -q
```

It should also be noted that multiple brokers can be run on the same host. To do so set alternate data directories for each qpidd instance.

Using a configuration file

Each option that can be specified on the command line can also be specified in a configuration file. To see available options, use `--help` on the command line:

```
./qpidd --help
```

A configuration file uses name/value pairs, one on each line. To convert a command line option to a configuration file entry:

- remove the '--' from the beginning of the option.
- place a '=' between the option and the value (use *yes* or *true* to enable options that take no value when specified on the command line).
- place one option per line.

For instance, the `--daemon` option takes no value, the `--log-to-syslog` option takes the values *yes* or *no*. The following configuration file sets these two options:

```
daemon=yes
log-to-syslog=yes
```

Can I use any Language client with the C++ Broker?

Yes, all the clients work with the C++ broker; it is written in C++, but uses the AMQP wire protocol. Any broker can be used with any client that uses the same AMQP version. When running the C++ broker, it is highly recommended to run AMQP 0-10.

Note that JMS also works with the C++ broker. For more details on using the Java client refer to these pages:

- [How to Use JNDI](#)
- [URL Formats for Qpid](#)
- [Example Classes](#)

Authentication

Linux

The PLAIN authentication is done on a username+password, which is stored in the `sasldb_path` file. Usernames and passwords can be added to the file using the command:

```
saslpasswd2 -f /var/lib/qpidd/qpidd.sasldb -u <REALM> <USER>
```

The REALM is important and should be the same as the `--auth-realm` option to the broker. This lets the broker properly find the user in the `sasldb` file.

Existing user accounts may be listed with:

```
sasldblistusers2 -f /var/lib/qpidd/qpidd.sasldb
```

NOTE: The `sasldb` file must be readable by the user running the `qpidd` daemon, and should be readable only by that user.

Windows

On Windows, the users are authenticated against the local machine. You should add the appropriate users using the standard Windows tools (Control Panel->User Accounts). To run many of the examples, you will need to create a user "guest" with password "guest".

If you cannot or do not want to create new users, you can run without authentication by specifying the `no-auth` option to the broker.

Slightly more complex configuration

The easiest way to get a full listing of the broker's options are to use the `--help` command, run it locally for the latest set of options. These options can then be set in the conf file for convenience (see above)

```

./qpidd --help

Usage: qpidd OPTIONS
Options:
  -h [ --help ]           Displays the help message
  -v [ --version ]        Displays version information
  --config FILE (/etc/qpidd.conf) Reads configuration from FILE

Module options:
  --module-dir DIR (/usr/lib/qpidd) Load all .so modules in this directory
  --load-module FILE        Specifies additional module(s) to be loaded
  --no-module-dir          Don't load modules from module directory

Broker Options:
  --data-dir DIR (/var/lib/qpidd) Directory to contain persistent data generated by the broker
  --no-data-dir            Don't use a data directory. No persistent
                           configuration will be loaded or stored
  -p [ --port ] PORT (5672) Tells the broker to listen on PORT
  --worker-threads N (3)    Sets the broker thread pool size
  --max-connections N (500) Sets the maximum allowed connections
  --connection-backlog N (10) Sets the connection backlog limit for the
                             server socket
  --staging-threshold N (5000000) Stages messages over N bytes to disk
  -m [ --mgmt-enable ] yes|no (1) Enable Management
  --mgmt-pub-interval SECONDS (10) Management Publish Interval
  --ack N (0)               Send session.ack/solicit-ack at least every
                             N frames. 0 disables voluntary ack/solicit
                             -ack

Daemon options:
  -d [ --daemon ]          Run as a daemon.
  -w [ --wait ] SECONDS (10) Sets the maximum wait time to initialize the
                              daemon. If the daemon fails to initialize, prints
                              an error and returns 1
  -c [ --check ]           Prints the daemon's process ID to stdout and
                              returns 0 if the daemon is running, otherwise
                              returns 1
  -q [ --quit ]            Tells the daemon to shut down

Logging options:
  --log-output FILE (stderr) Send log output to FILE. FILE can be a file name
                              or one of the special values:
                              stderr, stdout, syslog
  -t [ --trace ]           Enables all logging
  --log-enable RULE (error+) Enables logging for selected levels and component
                              s. RULE is in the form 'LEVEL+:PATTERN'
                              Levels are one of:
                              trace debug info notice warning error critical
                              For example:
                              '--log-enable warning+' logs all warning, error
                              and critical messages.
                              '--log-enable debug:framing' logs debug messages
                              from the framing namespace. This option can be
                              used multiple times
  --log-time yes|no (1)     Include time in log messages
  --log-level yes|no (1)    Include severity level in log messages
  --log-source yes|no (0)   Include source file:line in log messages
  --log-thread yes|no (0)   Include thread ID in log messages
  --log-function yes|no (0) Include function signature in log messages

```

Loading extra modules

By default the broker will load all the modules in the module directory, however it will NOT display options for modules that are not loaded. So to see the options for extra modules loaded you need to load the module and then add the help command like this:

