SSL

SSL How to

1. C++ broker (M4 and up)
2. Java Client
3. .Net Client

C++ broker (M4 and up)

- You need to get a certificate signed by a CA, trusted by your client.
- If you require client authentication, the clients certificate needs to be signed by a CA trusted by the broker.
- Setting up the certificates for testing.
  - For testing purposes you could use the following guide to setup your certificates.
  - In summary you need to create a root CA and import it to the brokers certificate data base.
  - Create a certificate for the broker, sign it using the root CA and then import it into the brokers certificate data base.
- Load the acl module using --load-module or if loading more than one module, copy ssl.so to the location pointed by --module-dir

```
Ex if running from source. ./qpidd --load-module /libs/ssl.so
```

- Specify the password file (a plain text file with the password), certificate database and the brokers certificate name using the following options

```
Ex ./qpidd ... --ssl-cert-password-file ~/pfile --ssl-cert-db ~/server_db/ --ssl-cert-name localhost.
localdomain
```

- If you require client authentication you need to add --ssl-require-client-authentication as a command line argument.
- Please note that the default port for SSL connections is 5671, unless specified by --ssl-port

Here is an example of a broker instance that requires SSL client side authenticaiton

```
./qpidd ./qpidd --load-module /libs/ssl.so --ssl-cert-password-file ~/pfile --ssl-cert-db ~/server_db/ --ssl-
cert-name localhost.localdomain --ssl-require-client-authentication
```

Java Client (M4 and up)

- This guide is for connecting with the Qpid c++ broker.
- Setting up the certificates for testing. In summary,
  - You need to import the trusted CA in your trust store and keystore
  - Generate keys for the certificate in your key store
  - Create a certificate request using the generated keys
  - Create a certficate using the request, signed by the trusted CA.
  - Import the signed certificate into your keystore.
- Pass the following JVM arguments to your client.

```
-Djavax.net.ssl.keyStore=/home/bob/ssl_test/keystore.jks
   -Djavax.net.ssl.keyStorePassword=password
   -Djavax.net.ssl.trustStore=/home/bob/ssl_test/certstore.jks
   -Djavax.net.ssl.trustStorePassword=password
```

.Net Client (M4 and up)

- If the Qpid broker requires client authentication then you need to get a certificate signed by a CA, trusted by your client.

Use the connectSSL instead of the standard connect method of the client interface.

connectSSL signature is as follows:

```
public void connectSSL(String host, int port, String virtualHost, String username, String password, String
serverName, String certPath, bool rejectUntrusted)
```

Where

- host: Host name on which a Qpid broker is deployed
- port: Qpid broker port
- virtualHost: Qpid virtual host name
- username: User Name
- password: Password
- serverName: Name of the SSL server

- certPath: Path to the X509 certificate to be used when the broker requires client authentication
- rejectUntrusted: If true connection will not be established if the broker is not trusted (the server certificate must be added in your truststore)

Python & Ruby Client (M4 and up)

Simply use amqps:// in the URL string as defined above