
Quilt Documentation

Release 0.1

Ross Delinger Will Dignazio

May 02, 2012

CONTENTS

1	Quilt	3
2	Indices and tables	5
	Python Module Index	7

Contents:

QUILT

```
class quilt.IncomingThread(listen_port, proc_queue)
```

The Incoming thread handles connections from servers and passes the data they send off to the processors

Parameters

- **listen_port** (*int*) – the port to listen for connecting servers on
- **proc_queue** (*Queue*) – The queue to pass messages off to processor with

```
class quilt.OutgoingThread(addr, port, queue)
```

For each outgoing connection one of these threads is created and it will forward the data in its personal Queue to the destination

Parameters

- **addr** (*str*) – Address of the server to connect to
- **port** (*int*) – port number of the server to connect to
- **queue** (*Queue*) – Data queue for outgoing data for this connection

```
class quilt.ProcessorThread(proc_queue, protocol)
```

This thread takes data out of the proc_queue and does things to that data based on the protocol

Parameters

- **proc_queue** (*Queue*) – Queue of data to be processed
- **protocol** (*QuiltProtocol*) – the object used to parse the data recieved

```
class quilt.QuiltServer(addr, incoming_port, max_proc=10)
```

The main server class for Quilt, spins up threads for incoming connections and processing data, and then runs the user interface code

Parameters

- **addr** (*str*) – address of this server
- **incoming_port** (*int*) – port for incoming connects

Type max_proc: int

Param max_proc: maximum number of processor threads to use

```
start()
```

Start the server and display and user interface

```
class quilt.QuiltServer(addr, incoming_port, max_proc=10)
```

The main server class for Quilt, spins up threads for incoming connections and processing data, and then runs the user interface code

Parameters

- **addr** (*str*) – address of this server
- **incoming_port** (*int*) – port for incoming connects

Type max_proc: int

Param max_proc: maximum number of processor threads to use

start ()

Start the server and display and user interface

class quilt.**IncomingThread** (*listen_port, proc_queue*)

The Incoming thread handles connections from servers and passes the data they send off to the processors

Parameters

- **listen_port** (*int*) – the port to listen for connecting servers on
- **proc_queue** (*Queue*) – The queue to pass messages off to processor with

class quilt.**ProcessorThread** (*proc_queue, protocol*)

This thread takes data out of the proc_queue and does things to that data based on the protocol

Parameters

- **proc_queue** (*Queue*) – Queue of data to be processed
- **protocol** (*QuiltProtocol*) – the object used to parse the data recieved

class quilt.**OutgoingThread** (*addr, port, queue*)

For each outgoing connection one of these threads is created and it will forward the data in its personal Queue to the destination

Parameters

- **addr** (*str*) – Address of the server to connect to
- **port** (*int*) – port number of the server to connect to
- **queue** (*Queue*) – Data queue for outgoing data for this connection

QUILT PROTOCOL OBJECT

```
class quilt.protocol.QuiltProtocol(addr, port)
```

```
connect_to_server(server, port)
```

Connect to a server

Parameters

- **server** (*str*) – server address
- **port** (*int*) – port number of the server

```
handle(message)
```

Handler method receives a message and decides how to deal with it. The protocol is split into parts: [destination, cmd, args...]. Destination options are: all or a single server :type message: list :param message: a list of data received from a zeromq recv_multipart

```
handle_ping(args)
```

Handle a ping from an incoming server. Format is [ping, sender] response is [pong, ourname]

Parameters **args** (*list*) – A list of arguments for this command

```
handle_server_connect(args)
```

Handle an incoming server connect, the incoming server sends us server-connect and then we do things like connect back to them. args should look like this: [address, port]

Parameters **args** (*list*) – A list of args that are received from the server

```
ping_server(server)
```

Ping a specific server. Next step in this implementation is to allow for pinging servers we are not directly connected to, this involves routing

Parameters **server** (*str*) – The server to ping

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

q

`quilt, 3`

`quilt.protocol, ??`