

Speechresearch/gyropy

API Documentation

August 31, 2011

Contents

Contents	1
1 Package gyropy	4
1.1 Modules	4
1.2 Variables	4
2 Module gyropy.debugging_locks	5
2.1 Functions	5
2.2 Variables	5
2.3 Class State	5
2.3.1 Methods	5
2.3.2 Properties	5
2.4 Class Lock	6
2.4.1 Methods	6
2.4.2 Properties	6
2.4.3 Class Variables	6
2.5 Class Condition	7
2.5.1 Methods	7
2.5.2 Properties	7
2.5.3 Class Variables	8
3 Module gyropy.g_mailbox	9
3.1 Functions	9
3.2 Variables	9
3.3 Class EOF	9
3.3.1 Methods	9
3.3.2 Properties	10
3.4 Class mailbox	10
3.4.1 Methods	10
3.4.2 Properties	12
3.4.3 Class Variables	12
3.5 Class waitbox	12
3.5.1 Methods	13
3.5.2 Properties	13
3.6 Class repbox	14
3.6.1 Methods	14
3.6.2 Properties	14

3.7	Class <code>listbox</code>	15
3.7.1	Methods	15
3.7.2	Properties	17
3.7.3	Class Variables	17
3.8	Class <code>listbox</code>	17
3.8.1	Methods	17
3.8.2	Properties	19
3.8.3	Class Variables	19
4	Module <code>gyropy.g_pipe</code>	20
4.1	Functions	20
4.2	Variables	20
4.3	Class <code>pdf</code>	20
4.3.1	Methods	20
4.3.2	Properties	21
4.3.3	Class Variables	21
5	Module <code>gyropy.g_proxy</code>	22
5.1	Functions	22
5.2	Variables	24
5.3	Class <code>ProtocolError</code>	24
5.3.1	Methods	24
5.3.2	Properties	25
5.4	Class <code>test0</code>	25
5.4.1	Methods	25
5.4.2	Properties	25
5.5	Class <code>test0classic</code>	26
5.6	Class <code>test1</code>	26
5.6.1	Methods	26
5.6.2	Properties	26
5.6.3	Class Variables	26
5.7	Class <code>test1classic</code>	27
5.7.1	Methods	27
5.7.2	Class Variables	27
5.8	Class <code>proxy_object</code>	27
5.8.1	Methods	27
5.8.2	Properties	27
5.9	Class <code>object_store</code>	28
5.9.1	Methods	28
5.9.2	Properties	28
5.10	Class <code>C_threadstate</code>	29
5.10.1	Methods	29
5.10.2	Properties	30
5.10.3	Class Variables	30
5.11	Class <code>io_grabber</code>	30
5.11.1	Methods	30
5.11.2	Properties	31
5.12	Class <code>io_labeler</code>	31
5.12.1	Methods	31
5.12.2	Properties	31
5.13	Class <code>dropsite</code>	32
5.13.1	Methods	32

5.13.2	Properties	32
5.14	Class fancyprint	33
5.14.1	Methods	33
5.14.2	Properties	33
5.15	Class pickler	33
5.15.1	Methods	33
5.15.2	Properties	34
5.16	Class Connection	34
5.16.1	Methods	34
5.16.2	Properties	36
5.16.3	Class Variables	36
5.17	Class proxy_to_a_class	36
5.17.1	Methods	36
5.17.2	Properties	37
5.18	Class remote_exec	37
5.18.1	Methods	37
5.18.2	Properties	38
6	Module gyropy.g.remote_exec	39
6.1	Functions	39
6.2	Variables	39
6.3	Class RemoteException	39
6.3.1	Methods	40
6.3.2	Properties	40
6.4	Class io_grabber	40
6.4.1	Methods	41
6.4.2	Properties	41
6.5	Class response	41
6.5.1	Methods	42
6.5.2	Properties	42
6.6	Class request	42
6.6.1	Methods	42
6.6.2	Properties	42
6.7	Class remote_fcn	43
6.7.1	Methods	43
6.7.2	Properties	43
6.8	Class connection	43
6.8.1	Methods	44
6.8.2	Properties	44
7	Module gyropy.setup	45
	Index	46

1 Package gyropy

1.1 Modules

- **debugging_locks** (*Section 2, p. 5*)
- **g_mailbox** (*Section 3, p. 9*)
- **g_pipe**: A multithreaded version of `os.popen2()`. (*Section 4, p. 20*)
- **g_proxy**: This will behave badly under `isinstance()`. (*Section 5, p. 22*)
- **g_remote_exec** (*Section 6, p. 39*)
- **setup** (*Section 7, p. 45*)

1.2 Variables

Name	Description
<code>--package--</code>	Value: None

2 Module `gyropy.debugging_locks`

2.1 Functions

```
prinfo(*s)
```

2.2 Variables

Name	Description
DEBUG	Value: False
Lockstate	Value: <gyropy.debugging_locks.State object at 0x2caa990>
__package__	Value: 'gyropy'

2.3 Class State

```
object └─
          gyropy.debugging_locks.State
```

2.3.1 Methods

```
__init__(self, fname=None, llen=79)

x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)
```

```
write(self, name, *s)
```

```
setname(self, name, me=None)
```

```
getName(self)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

2.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

2.4 Class Lock



2.4.1 Methods

```
__init__(self, verbose=0, name=None)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature
 Overrides: `object.__init__` `extit`(inherited documentation)

```
acquire(self)
```

```
release(self)
```

```
locked(self)
```

```
__del__(self)
```

```
__enter__(self)
```

```
__exit__(self, ty, value, traceback)
```

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

2.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

2.4.3 Class Variables

continued on next page

Name	Description
ls	Value: <gyropy.debugging_locks.State object at 0x2caa990>

2.5 Class Condition

object └─
 gyropy.debugging_locks.Condition

2.5.1 Methods

```
__init__(self, lock=None, verbose=0, name=None)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)
```

```
wait(self)
Called from lock.
```

```
acquire(self)
```

```
release(self)
```

```
notify(self)
```

```
notifyAll(self)
```

```
__del__(self)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

2.5.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

2.5.3 Class Variables

Name	Description
ls	Value: <gyropy.debugging_locks.State object at 0x2caa990>

3 Module gyropy.g_mailbox

3.1 Functions

```
prinfo(*s)
```

```
test1(m, n=1000)
```

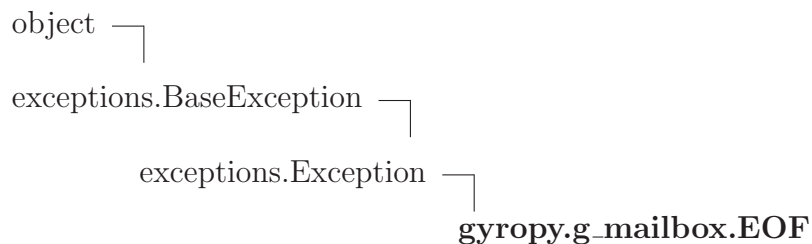
```
test2(m)
```

```
testmt()
```

3.2 Variables

Name	Description
DEBUG	Value: False
testmt_waiting	Value: 0
__package__	Value: 'gyropy'

3.3 Class EOF



3.3.1 Methods

```
__init__(self, *s)

x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)
```

Inherited from exceptions.Exception

```
__new__()
```

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

3.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

3.4 Class mailbox**3.4.1 Methods**

<code>__init__(self)</code> <code>x.__init__(...)</code> initializes x; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> extit(inherited documentation)
<code>close(self, for_put=EOF(), for_get=EOF(), ngx=-1, npx=-1)</code> <hr/> Close the connection. It causes other threads that use the mailbox to raise an exception, unless you give it <code>self.ignore</code> as an argument. <code>Npx</code> and <code>ngx</code> specify how many threads will commit suicide; zero means none; setting either to 1 is a useful way to kill a single thread. Legal values for <code>for_put</code> and <code>for_get</code> are any exception, anything returned from <code>sys.exc_info</code> , <code>self.ignore</code> and <code>None</code> . <ul style="list-style-type: none"> • <code>None</code> => Do not raise an exception; • <code>self.ignore</code> => Do not change whatever exception (or <code>None</code>) is already stored.

__len__(*self*)

__nonzero__(*self*)

put(*self*, *v*)

Put a value into the mailbox. If one of the `close()` functions has been used to set an exception on `for_put`, it will raise the exception instead.

putclose(*self*, *for_get*=`EOF()`, *for_put*=`EOF()`, *ngx*=-1, *npx*=-1, *defer_ex*=`False`)

Close the connection. This is normally called by the producer of data in place of `put()`. It can cause other threads to raise an exception.

This call acts like a `put()` in that it will block until the mailbox is emptied. It does not fill the mailbox, though. Following this call:

- `get()` will not block (it will raise an exception unless `for_get` is `self.ignore` or `npx==0`),
- `put()` will not block (it will raise an exception if `for_put` is not `self.ignore` and `npx!=0`),
- likewise `putclose()` and `getcloses()`.

It will raise an exception (unless `defer_ex` is `True`) if some other thread has closed first and set a `for_put` exception. However, it will set a `for_get` exception (if requested) before expiring.

putraise(*self*)

Raise any residual exceptions. This is normally called by the producer of data

This call acts like a `put()` in that it will block until the mailbox is emptied or until an exception is waiting.

get(*self*)

Get an item from the mailbox. Possibly, raise an exception if `close` or `putclose` has been called on the mailbox.

Notes:

- Suicide has a lower priority than processing data. So, if there is data in the mailbox and `putclose()` has been called, one call to `get()` will succeed then the second call will raise an exception.
- If the mailbox is empty, and if a previous call has specified an exception to be raised by readers, raise it.

```
getclose(self, for_put=EOF(), for_get=EOF(), npx=-1, ngx=-1,
defer_ex=False)
```

Close the connection. This is normally called by the consumer of data in place of get(). It causes other threads to raise an exception. It will raise an exception if some other thread has closed the mailbox first. Note that if there is something in the mailbox, it will empty and return the value.

Note: If the mailbox is empty, and if a previous call has specified an exception to be raised by readers, raise it.

```
__del__(self)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

3.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

3.4.3 Class Variables

Name	Description
<code>eof</code>	Value: EOF()
<code>ignore</code>	Value: EOF()

3.5 Class waitbox

```
object └─
        gyropy.g_mailbox.waitbox
```

This is a stripped down mailbox that is good for one message between one sender and one reciever. It is an error to call put() or get() or putclose() more than once.

The normal usage is that you create the mailbox, spawn a thread, and hand the mailbox to the new thread. Then, one side calls `get` on the mailbox to wait for an answer. The other side computes something, then calls `put`. The waitbox is then thrown away.

3.5.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`__nonzero__(self)`

`putclose(self, to_be_raised=EOF())`

Close the connection. This is normally called by the producer of data in place of `put()`. It causes other threads to raise an exception.

`put(self, v)`

`get(self)`

Get an object from the mailbox. This will normally block if the mailbox is empty.

Notes:

- Suicide has a lower priority than processing data. So, if there is data in the mailbox and `putclose()` has been called, one call to `get()` will succeed; the second call will raise an exception.
- If the mailbox is empty, and if a previous call has specified an exception to be raised by readers, raise it.

`__del__(self)`

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

3.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

3.6 Class repbox



Like a mailbox, but produces infinite copies of whatever value was put in there last. It is created with a value inside.

Note: This needs revision to more closely follow mailbox's methods.

3.6.1 Methods

```
__init__(self, value=None, suicide=None)
```

x.__init__(...) initializes x; see help(type(x)) for signature
 Overrides: object.__init__ extit(inherited documentation)

```
close(self, e_to_raise=EOF())
```

```
__len__(self)
```

```
closenow(self, e_to_raise=EOF())
```

```
put(self, v)
```

```
get(self)
```

Get an object from the mailbox. This will normally block if the mailbox is empty.

Note: If the mailbox is empty, and if a previous call has specified an exception to be raised by readers, raise it.

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),  

__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

3.6.2 Properties

Name	Description
<i>Inherited from object</i>	

continued on next page

Name	Description
<code>--class--</code>	

3.7 Class *listbox*

object └─ **gyropy.g_mailbox.listbox**

A mailbox with an unbounded queue.

3.7.1 Methods

`--init--`(*self*, *initial=None*)

x.`--init--`(...) initializes *x*; see `help(type(x))` for signature

Overrides: `object.--init--` `extit`(inherited documentation)

`close`(*self*, *for_put=EOF()*, *for_get=EOF()*, *ngx=-1*, *npx=-1*)

Close the connection. It causes other threads to raise an exception, unless you give it `self.ignore` as an argument. *Npx* and *ngx* specify how many threads will commit suicide; zero means none; setting either to 1 is a useful way to kill a single thread.

Legal values for *for_put* and *for_get* are any exception, anything returned from `sys.exc_info`, `self.ignore` and `None`.

- `None` => Do not raise an exception;
- `self.ignore` => Do not change whatever exception (or `None`) is already stored.

`--len--`(*self*)

`put`(*self*, *v*)

Put a value into the mailbox. If one of the `close()` functions has been used to set an exception on *for_put*, it will raise the exception instead.

putclose(*self*, *for_get*=EOF(), *for_put*=EOF(), *ngx*=-1, *npx*=-1, *defer_ex*=False)

Close the connection. This is normally called by the producer of data in place of put(). It can cause other threads to raise an exception.

It does not fill the mailbox. Following this call,

- get() will not block (it will raise an exception unless *for_get* is *self.ignore* or *npx*==0),
- put() will not block (it will raise an exception if *for_put* is not *self.ignore* and *npx*!=0),
- likewise putclose() and getclose().

It will raise an exception (unless *defer_ex* is True) if some other thread has closed first and set a *for_put* exception. However, it will set a *for_get* exception (if requested) before expiring.

putraise(*self*)

Raise any residual exceptions. This is normally called by the producer of data

get(*self*)

Get an object from the mailbox. This will normally block if the mailbox is empty.

Notes:

- Suicide has a lower priority than processing data. So, if there is data in the mailbox and putclose() has been called, one call to get() will succeed; the second call will raise an exception.
- If the mailbox is empty, and if a previous call has specified an exception to be raised by readers, raise it.

getclose(*self*, *for_put*=EOF(), *for_get*=EOF(), *npx*=-1, *ngx*=-1, *defer_ex*=False)

Get an object from the mailbox and close the mailbox. This is normally called by the consumer of data in place of get(). It causes other threads to raise an exception.

Note: It will raise an exception if some other thread has closed the mailbox first.

__del__(*self*)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

3.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

3.7.3 Class Variables

Name	Description
<code>eof</code>	Value: <code>EOF()</code>
<code>ignore</code>	Value: <code>EOF()</code>

3.8 Class *listbox*

object —
gyropy.g_mailbox.listbox

A mailbox with an unbounded queue.

3.8.1 Methods

<code>__init__(self, initial=None)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>help(type(x))</code> for signature
Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)

close(*self*, *for_put*=EOF(), *for_get*=EOF(), *ngx*=-1, *npx*=-1)

Close the connection. It causes other threads to raise an exception, unless you give it *self.ignore* as an argument. *Npx* and *ngx* specify how many threads will commit suicide; zero means none; setting either to 1 is a useful way to kill a single thread.

Legal values for *for_put* and *for_get* are any exception, anything returned from *sys.exc_info*, *self.ignore* and *None*.

- *None* => Do not raise an exception;
- *self.ignore* => Do not change whatever exception (or *None*) is already stored.

__len__(*self*)**put**(*self*, *v*)

Put a value into the mailbox. If one of the *close()* functions has been used to set an exception on *for_put*, it will raise the exception instead.

putclose(*self*, *for_get*=EOF(), *for_put*=EOF(), *ngx*=-1, *npx*=-1, *defer_ex*=False)

Close the connection. This is normally called by the producer of data in place of *put()*. It can cause other threads to raise an exception.

It does not fill the mailbox. Following this call,

- *get()* will not block (it will raise an exception unless *for_get* is *self.ignore* or *npx*==0),
- *put()* will not block (it will raise an exception if *for_put* is not *self.ignore* and *npx*!=0),
- likewise *putclose()* and *getclosen()*.

It will raise an exception (unless *defer_ex* is *True*) if some other thread has closed first and set a *for_put* exception. However, it will set a *for_get* exception (if requested) before expiring.

putraise(*self*)

Raise any residual exceptions. This is normally called by the producer of data

get(self)

Get an object from the mailbox. This will normally block if the mailbox is empty.

Notes:

- Suicide has a lower priority than processing data. So, if there is data in the mailbox and putclose() has been called, one call to get() will succeed; the second call will raise an exception.
- If the mailbox is empty, and if a previous call has specified an exception to be raised by readers, raise it.

getclose(self, for_put=EOF(), for_get=EOF(), npx=-1, ngx=-1, defer_ex=False)

Get an object from the mailbox and close the mailbox. This is normally called by the consumer of data in place of get(). It causes other threads to raise an exception.

Note: It will raise an exception if some other thread has closed the mailbox first.

__del__(self)***Inherited from object***

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

3.8.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

3.8.3 Class Variables

Name	Description
eof	Value: EOF()
ignore	Value: EOF()

4 Module `gyropy.g_pipe`

A multithreaded version of `os.popen2()`. Note that the argument list isn't quite the same.

4.1 Functions

popen2(*path*, *args*, *bufsize*=0)

Forks off a process, and returns the processes input and output file descriptors. The latter is really a *pfd* (defined above).

Path and args are passed directly into `os.execvp()`.

test()

4.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gyropy'</code>

4.3 Class *pfd*

object —
 gyropy.g_pipe.pfd

Pseudo-file descriptor. Just like a FD, except that it waits for the process at the end of the pipe to terminate when you close it.

4.3.1 Methods

__init__(*self*, *p*)

x.**__init__**(...) initializes *x*; see `help(type(x))` for signature

Overrides: *object*.**__init__** *exitit*(inherited documentation)

read(*self*)

readline(*self*)

`readlines(self)``next(self)``__iter__(self)``flush(self)``xreadlines(self, sizehint=-1)``close(self)``__del__(self)`***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

4.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

4.3.3 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "Pseudo-file descriptor. Just like a FD, exc..."

5 Module *gyropy.g_proxy*

This will behave badly under `isinstance()`. if `isinstance(x,y)` probably not `isinstance(remote_x, y)`. Also, dicts are passed by value instead of by reference, so if you remotely modify a dict, you will not see the modification locally.

Version: 0.23

5.1 Functions

Id(*object*)

Return the identity of an object. This is guaranteed to be unique among simultaneously existing objects. (Hint: it's the object's memory address.)

Return Value

integer

log(**args*)

Prinfo(*s*)

Oid(*o*)

diO(*s*)

prinfo(*s*)

S(*s*)

deal_with_arbitrary_exception(*c*, *boxid*)

thread_within_listen(*inbox*, *threadstate*, *c*)

thread_within_listen_wrapper(*mbox*, *threadstate*, *conn*)

import_mod(*name*, *path*)

Import a module from the specified path, or, failing that, look in `sys.path` then for a builtin. If *path* is `None`, only look in `sys.path` and builtins. If *path* is an array containing `None`, replace the `None` with `sys.path`.

s_listen(*c*)

This reads answers and puts them in the correct mailboxes.

s_listen_wrapper(*c*, *s_listen_done*)**get_raise**(*mbox*, *conn*)**transform_callable**(*onm*)**create_property**(*onm*)**class_under_proxied_instance**(*ocid*, *conn*)**enh_exc_class_maker**(*ex_nm*, *ex_mod*)**construct_enhanced_exception**(*conn*, *ex_nm*, *ex_mod*, *ex_clid*, *ex_args*, *trb*)**test_ee**()**cached_class_u_p_i**(*ocid*, *conn*)**proxy_factory**(*oid*, *cloid*, *conn*)**test_minimal**()**test_simpleclass**()**test_minimalclass**()**test_EQ**()**test_sys**()**test_builtins**()**test_exit**()**test_globals**()

```
test_file()
```

```
test_numpy()
```

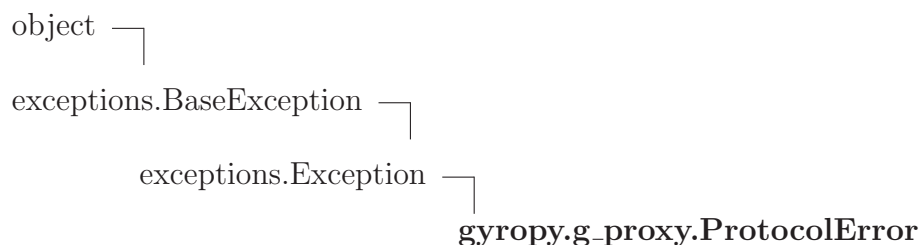
```
test_exception()
```

```
server(name)
```

5.2 Variables

Name	Description
DEBUG	Value: 2
LOGFILE	Value: './g_proxy.client'
TRANSPORTABLE_TYPES	Value: int, float, complex, tuple, str, long, types.NoneType, ty...
TRANSPORTABLE_TYPES_O	Value: frozenset([Id(q) for q in TRANSPORTABLE_TYPES]+ [Id(objec...
NO_CLATTR	Value: frozenset(['__dict__', '__class__', '__weakref__', '__ini...
TEST	Value: tuple([(None, False, 1, 2.0, frozenset([33, 11]), (6,)), '...

5.3 Class ProtocolError



5.3.1 Methods

```
__init__(self, *a)
```

x.__init__(...) initializes x; see help(type(x)) for signature

Overrides: object.__init__ extit(inherited documentation)

Inherited from exceptions.Exception

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

5.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

5.4 Class test0

object —
 gyropy.g_proxy.test0

5.4.1 Methods

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`, `__new__()`, `__reduce__()`,
`__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

5.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

5.5 Class test0classic

5.6 Class test1



5.6.1 Methods

```
__init__(self, x)
```

x.**__init__**(...) initializes *x*; see help(type(*x*)) for signature

Overrides: object.**__init__** extit(inherited documentation)

```
__eq__(self, other)
```

```
w(self)
```

Inherited from object

__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(), **__repr__**(), **__setattr__**(), **__sizeof__**(), **__str__**(), **__subclasshook__**()

5.6.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

5.6.3 Class Variables

Name	Description
whatami	Value: 'I am the class'

5.7 Class test1classic

5.7.1 Methods

```
--init--(self, x)
```

```
--eq--(self, other)
```

```
w(self)
```

5.7.2 Class Variables

Name	Description
whatami	Value: 'I am the class'

5.8 Class proxy_object

```

object └─
          gyropy.g_proxy.proxy_object

```

5.8.1 Methods

Inherited from object

```
--delattr--(), --format--(), --getattr__(), --hash--(), --init--(), --new--(), --reduce--(),
--reduce_ex--(), --repr--(), --setattr--(), --sizeof--(), --str--(), --subclasshook--()
```

5.8.2 Properties

Name	Description
<i>Inherited from object</i>	
--class--	

5.9 Class `object_store`

`object` └─
 `gyropy.g_proxy.object_store`

5.9.1 Methods

```
__init__(self, storetype=dict)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

```
put(self, xid, x)
```

```
putid(self, x)
```

```
rm(self, oid)
```

```
get(self, xid)
```

```
disconnect(self)
```

This is only called for the rowlp case, and the object store is a WeakDictionary, so if we encounter an object, we know that something else references it. Consequently, we cannot depend on it being deleted and calling its `__del__` item any time soon.

```
clear(self)
```

Inherited from `object`

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),  

__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

5.9.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

5.10 Class `C_threadstate`



5.10.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` extit(inherited documentation)

`to_idle(self, me)`

`to_busy(self, me)`

`to_stopped(self, me)`

`to_starting(self, me)`

`anticipate_stop(self)`

`n_busy(self)`

`n_idle(self)`

`list(self)`

`busylist(self)`

`wait_until_all_done(self)`

Called from main thread.

`reap_stopped_threads(self)`

Called from main thread.

Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,

`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

5.10.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

5.10.3 Class Variables

Name	Description
IDLE	Value: 0
BUSY	Value: 1
STARTING	Value: 2

5.11 Class `io_grabber`

object └─ **gyropy.g_proxy.io_grabber**

This buffers text, transmitting it in units of lines. That's important at the other end if there are multiple connections, because `fancyprint` adds a prefix at the beginning of each write, assuming it is the beginning of each line.

5.11.1 Methods

`__init__(self, name)`

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`connect(self, c)`

`write(self, s)`

`writelines(self, x)`

`flush(self)`

close(*self*)

Inherited from object

`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

5.11.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

5.12 Class `io.labeler`



5.12.1 Methods

__init__(*self*, *name*)

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

writelines(*self*, *x*)

write(*self*, *s*)

flush(*self*)

Inherited from object

`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

5.12.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

5.13 Class dropsite



5.13.1 Methods

__init__ (<i>self</i>)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)

make (<i>self</i> , <i>dbg</i> =' ')
--

find (<i>self</i> , <i>boxid</i> , <i>dbg</i> =None)
For internal use only.

count (<i>self</i>)

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
 __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

5.13.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

5.14 Class fancyprint



5.14.1 Methods

```
__init__(self, fd, prefix='', lock=None)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)
```

```
fprint(self, s)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

5.14.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

5.15 Class pickler



5.15.1 Methods

```
__init__(self, conn)
x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)
```

```
dump(self, obj, proxy=NEVER)
```

```
persistent_id(self, obj)
```

```
oid(self, obj)
```

```
load(self, s)
```

```
persistent_load(self, oidstr)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),  
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

5.15.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

5.16 Class Connection

```
object └─ gyropy.g-proxy.Connection
```

You need to call `close()`.

5.16.1 Methods

```
__init__(self, wfd, rfd, stderr, stdout)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit` (inherited documentation)

```
send(self, op, boxid, args, proxy=MAYBE)
```

`close_wfd(self)`

Called as part of the QUIT sequence.

`remote_remove_from_store(self, oid)`**`remote_test(self, i)`****`remote_setattr(self, objid, k, v)`****`remote_delattr(self, objid, k)`****`remote_getattr(self, objid, k)`****`remote_call(self, objid, a, kw)`****`remote_call_method(self, objid, aname, a, kw)`****`remote_import_mod(self, name, path)`**

Import a module from the specified path, or, failing that, look in `sys.path` then for a builtin. If path is `None`, only look in the remote machine's `sys.path` and builtins. If path is an array containing `None`, replace the `None` with the remote machine's `sys.path`.

`remote_globals(self)`**`remote_get_classinfo(self, cloid)`**Gets information about `obj.__class__`.**`serve(self)`****`close(self)`****`terminate(self)`****`__del__(self)`*****Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

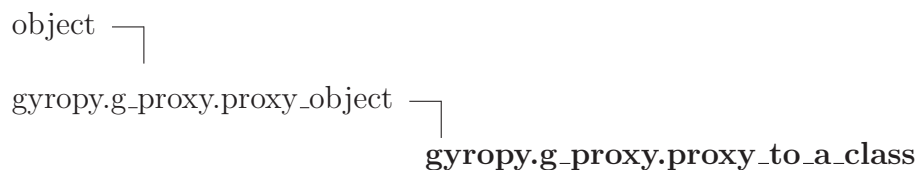
5.16.2 Properties

Name	Description
<i>Inherited from object</i> <code>__class__</code>	

5.16.3 Class Variables

Name	Description
<code>sys</code>	Value: <code>sys</code>

5.17 Class `proxy_to_a_class`



5.17.1 Methods

`__init__`(*self*, *oid*, *conn*)
`x.__init__(...)` initializes `x`; see `help(type(x))` for signature
 Overrides: `object.__init__` extit(inherited documentation)

`__setattr__`(*self*, *k*, *v*)
`x.__setattr__('name', value) <==> x.name = value`
 Overrides: `object.__setattr__` extit(inherited documentation)

`__delattr__`(*self*, *k*)
`x.__delattr__('name') <==> del x.name`
 Overrides: `object.__delattr__` extit(inherited documentation)

```
__getattr__(self, k)
```

```
x.__getattr__('name') <==> x.name
```

```
Overrides: object.__getattr__ extit(inherited documentation)
```

```
__call__(self, *a, **kw)
```

Inherited from object

```
__format__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __sizeof__(),  
__str__(), __subclasshook__()
```

5.17.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

5.18 Class `remote_exec`



5.18.1 Methods

```
__init__(self, path, arglist, cname=None, printlock=None)
```

```
x.__init__(...) initializes x; see help(type(x)) for signature
```

```
Overrides: object.__init__ extit(inherited documentation)
```

```
import_mod(self, name, path=None)
```

```
Import a module from the specified path, or, failing that, look in sys.path then for a builtin. If path is None, only look in the remote machine's sys.path and builtins. If path is an array containing None, replace the None with the remote machine's sys.path. The default is None.
```

```
test(self, i)
```

<code>globals(self)</code>

<code>close(self)</code>

<code>__del__(self)</code>

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

5.18.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

6 Module gyropy.g_remote_exec

6.1 Functions

```
make_remote_exception(name, mod, args, trtext)
```

```
deal_with_arbitrary_exception()
```

```
set_remote_path(pa)
```

```
echo(s)
```

```
log(*s)
```

```
server()
```

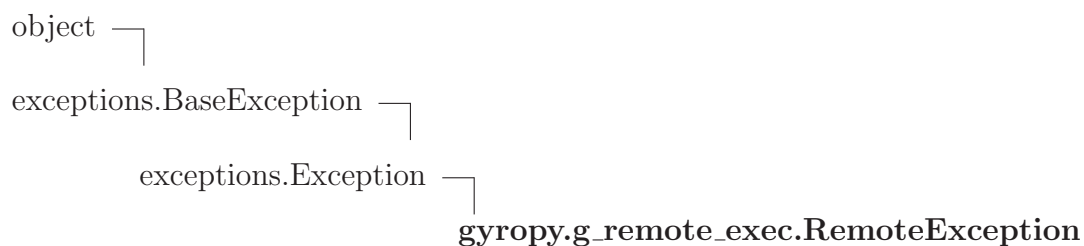
```
test_rf2(i)
```

```
test()
```

6.2 Variables

Name	Description
PROTOCOL	Value: 2
--package--	Value: 'gyropy'

6.3 Class RemoteException



6.3.1 Methods

```
__init__(self, *s)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit` (inherited documentation)

Inherited from `exceptions.Exception`

```
__new__()
```

Inherited from `exceptions.BaseException`

```
__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),  
__setattr__(), __setstate__(), __str__(), __unicode__()
```

Inherited from `object`

```
__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()
```

6.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

6.4 Class `io_grabber`

```

object └─
          gyropy.g_remote_exec.io_grabber

```

This buffers text, transmitting it in units of lines. That's important at the other end if there are multiple connections, because `fancyprint` adds a prefix at the beginning of each write, assuming it is the beginning of each line.

6.4.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`write(self, s)`

`writelines(self, x)`

`flush(self)`

`close(self)`

`get_io(self)`

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

6.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

6.5 Class response

object └─ **`gyropy.g_remote_exec.response`**

6.5.1 Methods

```
__init__(self, ety, val, out, err)

x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

6.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

6.6 Class request

```
object └─
          gyropy.g_remote_exec.request
```

6.6.1 Methods

```
__init__(self, path, mname, fname, a, kw)

x.__init__(...) initializes x; see help(type(x)) for signature
Overrides: object.__init__ extit(inherited documentation)
```

```
call(self)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

6.6.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

6.7 Class remote_fcn



6.7.1 Methods

__init__(*self*, *connection*, *pypath*, *mname*, *fname*)
x.__init__(...) initializes *x*; see help(type(*x*)) for signature
 Overrides: object.__init__ extit(inherited documentation)

__call__(*self*, **a*, ***kw*)

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
 __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

6.7.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

6.8 Class connection



6.8.1 Methods

`__init__(self, execpath, arglist)`

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`call(self, pypath, mname, fname, a, kw)`

`close(self)`

`__del__(self)`

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

6.8.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

7 Module gyropy.setup

Index

- gyropy (*package*), 4
 - gyropy.debugging_locks (*module*), 5–8
 - gyropy.debugging_locks.Condition (*class*), 7–8
 - gyropy.debugging_locks.Lock (*class*), 6–7
 - gyropy.debugging_locks.prinfo (*function*), 5
 - gyropy.debugging_locks.State (*class*), 5–6
 - gyropy.g_mailbox (*module*), 9–19
 - gyropy.g_mailbox.EOF (*class*), 9–10
 - gyropy.g_mailbox.listbox (*class*), 15–19
 - gyropy.g_mailbox.mailbox (*class*), 10–12
 - gyropy.g_mailbox.prinfo (*function*), 9
 - gyropy.g_mailbox.repbox (*class*), 13–15
 - gyropy.g_mailbox.test1 (*function*), 9
 - gyropy.g_mailbox.test2 (*function*), 9
 - gyropy.g_mailbox.testmt (*function*), 9
 - gyropy.g_mailbox.waitbox (*class*), 12–13
 - gyropy.g_pipe (*module*), 20–21
 - gyropy.g_pipe.pfd (*class*), 20–21
 - gyropy.g_pipe.popen2 (*function*), 20
 - gyropy.g_pipe.test (*function*), 20
 - gyropy.g_proxy (*module*), 22–38
 - gyropy.g_proxy.C_threadstate (*class*), 28–30
 - gyropy.g_proxy.cached_class_u_p_i (*function*), 23
 - gyropy.g_proxy.class_under_proxied_instance (*function*), 23
 - gyropy.g_proxy.Connection (*class*), 34–36
 - gyropy.g_proxy.construct_enhanced_exception (*function*), 23
 - gyropy.g_proxy.create_property (*function*), 23
 - gyropy.g_proxy.deal_with_arbitrary_exception (*function*), 22
 - gyropy.g_proxy.diO (*function*), 22
 - gyropy.g_proxy.dropsite (*class*), 32
 - gyropy.g_proxy.enh_exc_class_maker (*function*), 23
 - gyropy.g_proxy.fancyprint (*class*), 32–33
 - gyropy.g_proxy.get_raise (*function*), 23
 - gyropy.g_proxy.Id (*function*), 22
 - gyropy.g_proxy.import_mod (*function*), 22
 - gyropy.g_proxy.io_grabber (*class*), 30–31
 - gyropy.g_proxy.io_labeler (*class*), 31–32
 - gyropy.g_proxy.log (*function*), 22
 - gyropy.g_proxy.object_store (*class*), 27–28
 - gyropy.g_proxy.Oid (*function*), 22
 - gyropy.g_proxy.pickler (*class*), 33–34
 - gyropy.g_proxy.Prinfo (*function*), 22
 - gyropy.g_proxy.prinfo (*function*), 22
 - gyropy.g_proxy.ProtocolError (*class*), 24–25
 - gyropy.g_proxy.proxy_factory (*function*), 23
 - gyropy.g_proxy.proxy_object (*class*), 27
 - gyropy.g_proxy.proxy_to_a_class (*class*), 36–37
 - gyropy.g_proxy.remote_exec (*class*), 37–38
 - gyropy.g_proxy.S (*function*), 22
 - gyropy.g_proxy.s.listen (*function*), 22
 - gyropy.g_proxy.s.listen_wrapper (*function*), 23
 - gyropy.g_proxy.server (*function*), 24
 - gyropy.g_proxy.test0 (*class*), 25
 - gyropy.g_proxy.test0classic (*class*), 25–26
 - gyropy.g_proxy.test1 (*class*), 26
 - gyropy.g_proxy.test1classic (*class*), 26–27
 - gyropy.g_proxy.test_builtins (*function*), 23
 - gyropy.g_proxy.test_ee (*function*), 23
 - gyropy.g_proxy.test_EQ (*function*), 23
 - gyropy.g_proxy.test_exception (*function*), 24

- gyropy.g_proxy.test_exit (*function*), 23
- gyropy.g_proxy.test_file (*function*), 23
- gyropy.g_proxy.test_globals (*function*), 23
- gyropy.g_proxy.test_minimal (*function*),
23
- gyropy.g_proxy.test_minimalclass (*func-*
tion), 23
- gyropy.g_proxy.test_numpy (*function*), 24
- gyropy.g_proxy.test_simpleclass (*function*),
23
- gyropy.g_proxy.test_sys (*function*), 23
- gyropy.g_proxy.thread_within_listen (*func-*
tion), 22
- gyropy.g_proxy.thread_within_listen_wrapper
(*function*), 22
- gyropy.g_proxy.transform_callable (*func-*
tion), 23
- gyropy.g_remote_exec (*module*), 39–44
 - gyropy.g_remote_exec.connection (*class*),
43–44
 - gyropy.g_remote_exec.deal_with_arbitrary_exception
(*function*), 39
 - gyropy.g_remote_exec.echo (*function*), 39
 - gyropy.g_remote_exec.io_grabber (*class*),
40–41
 - gyropy.g_remote_exec.log (*function*), 39
 - gyropy.g_remote_exec.make_remote_exception
(*function*), 39
 - gyropy.g_remote_exec.remote_fcn (*class*),
43
 - gyropy.g_remote_exec.RemoteException
(*class*), 39–40
 - gyropy.g_remote_exec.request (*class*), 42–
43
 - gyropy.g_remote_exec.response (*class*), 41–
42
 - gyropy.g_remote_exec.server (*function*),
39
 - gyropy.g_remote_exec.set_remote_path (*func-*
tion), 39
 - gyropy.g_remote_exec.test (*function*), 39
 - gyropy.g_remote_exec.test_rf2 (*function*),
39
- gyropy.setup (*module*), 45