

Variables (access)

```
Set VsUtils.variables['<name>'] =
    v
```

```
Get v =
    VsUtils.variables['<name>']
```

Tip: use `<from VsUtils import variables as vs>` to access variables easily with `<vs['...']>`

Assertion

```
assertTrue(expr)
```

```
assertFalse(expr)
```

```
assertEqual(fisrt, second)
```

```
assertNotEqual(fisrt, second)
```

```
assertAlmostEqual(fisrt, second,
    places=7)
```

```
assertNotAlmostEqual(fisrt, second,
    d, places=7)
```

```
assertRaises(excClass, callObj,
    *args, **kwargs)
```

```
fail()
```

all assertions with additional parameters:
 (... , msg=None, continueTest=True)
 msg could use format string: "...%i" %
 vs[...]

Async assertion

```
assertTrueInterval(expr)
```

```
assertFalseInterval(expr)
```

```
assertAlmostEqualInterval(first, se
    c, places)
```

```
assertNotAlmostEqualInterval(first
    , sec, places)
```

```
assertTrueAtLeastInterval(expr)
```

```
assertAtLeastOnceTrue(expr)
```

all assertions with additional parameters:
 (... , msg=None)

Async usage

```
async = ATLASyncTest(self)
async.assert...
(lambda:vs[varname]...,msg)
async.assert...
('vs[varname]...',msg)
async.assert...(function,msg)
async.execute(timeout)
```

expr and msg could be lamda, string or function
 expr should contains only one variable.
 expr is evaluated one time to get the variable

Advanced tips

How to disable warning import msg

```
sys.modules['ihooks'].current_i
    mporter.addOkModule('xxx')
```

In custom assert, how to get correct file/line

Use `tbskip` param in `self.result.addFailure`

How to interact with operator ?

Use `wx` python binding

Interact with Logbook

```
rtc.setLogLevel(level)
rtc.startLogSession('SessionName'
    , 'comment')
rtc.logMessage(RTC.Logger.level,
    'UserModule', 'msg')
...
rtc.stopLogSession()
```

Levels are: *Debug, Info, Warning, Error, Critical*

RTC services

Creation

```
rtc = RTC.RTCManager(ip [,
    port])
```

Connection

```
rtc.connect(ip [, port])
```

Configure if not done by mmi

```
rtc.configure(client name,
    dbPath)
```

Usercode control

```
rtc.[start/stop]Code(codeName)
```

Monitoring control

```
rtc.
    [start/stop]Monitor(taskName)
```

Archiver control

```
rtc.
    [start/stop]Recordset(rsName)
```

Manage errors

```
try:... except: SystemError ,
    detail:
```

VS callback

```
class notifier(Vs.Callback):
    def variableDidChange(*args):
        print 'callback called'
callback = notifier()
notif = VsUtils.vc.
    registerCallbackValueChanged(nam
    e, callback)
...
del notif
```

Callback class should be created outside of
register method.

Other callback are available.



By **Vicnet**
cheatography.com/vicnet/

Published 30th November, 2016.
 Last updated 30th November, 2016.
 Page 1 of 2.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopad.com>

Variables (creation)

```
VsUtils.vc.createVariable('<name>'  
, Vs.<type>, [capacity=1])
```

```
VsUtils.vc.deleteVariable('<name>'  
)
```

Types are: *INTEGER, DOUBLE, OPAQUE*

Tips

```
import time  
time.sleep(seconds)
```

Testcase

```
class <TestCase name>(ATLTestCase):  
    def initialize(self):  
        # no exception inside this  
method  
    def finalize(self):  
        # no exception inside this  
method  
    def execute(self):  
        # test steps here, see  
Assertions
```

Test structure (tsdef)

TestSuite (*tsdef*)

Root *TestGroup*

TestGroup

TestCase

TestCase

Groups are optionals

Running tests

```
$PYTHONPATH /opt/gc/lib/python  
/opt/vs/lib
```

```
python utest/ATL/ATLTestRunner.py
```

-c *config.xml*

-r *result.xml*

-v *verbose in console*

[--server] *xmlrpc server*

Test results

Passed tests within limits

Failed test not in limits

Aborted not completed (*exception*)



By **Vicnet**
cheatography.com/vicnet/

Published 30th November, 2016.
Last updated 30th November, 2016.
Page 2 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopad.com>