

7.4 - The Python API

- The Python API provides a binding to the TET C API
- All the functions provided by the API are in a single Python module `pytet`

```
from pytet import *
```

Feb-05

TETware training course

THE *Open* GROUP
7-33

Interface to the user-written test code

- The Python binding uses the TET dynamic interfaces, so unlike the Shell and Perl APIs you do not need to setup an iclist for the TCM to communicate with the user code
 - Thereby simplifying the writing of test code

Feb-05

TETware training course

THE *Open* GROUP
7-34

Python Test Startup

- A test list is defined using a Python dictionary and calls are registered using the `pytet.pytet_init()` function. If no startup or cleanup functions are provided, then these are passed as `TET_NULLFP`.

```
testlist = { 1:test1, 2:test2, 3:test3 }
pytet_init(testlist, TET_NULLFP, TET_NULLFP)
```

Feb-05

TETware training course

THE *Open* GROUP
7-35

Python – startup and cleanup functions

- Supports test startup and cleanup functions

```
def startup():
    tet_delete(2, "Marking test 2 as uninitiated")
def cleanup():
    print "tcl: Calling cleanup"
def test1():
    ...
def test2():
    ...
testlist = { 1:test1, 2:test2}
pytet_init(testlist, startup, cleanup)
```

Feb-05

TETware training course

THE *Open* GROUP
7-36

Python API Features

- Uses the same names as the TET C API

For example:

```
myvar = tet_getvar("MY_CONFIG_VARIABLE")
tet_infoline("output text goes here")
tet_result(TET_PASS)
```

Feb-05

TETware training course

THE *Open* GROUP

7-37

Python API Features

- Error handling through exceptions

```
try:
    var = tet_getvar("PYTET_VAR")
except:
    tet_infoline("Failed to get value for
PYTET_VAR")
    tet_result(TET_UNRESOLVED)
return
```

Feb-05

TETware training course

THE *Open* GROUP

7-38

Note

- The pytet API wrapper is generated using the SWIG toolkit and that has some limitations which mean that not all of the TET C API features are presently available in the pytet Python API, e.g. varargs

Feb-05

TETware training course

THE *Open* GROUP

7-39

Building the pytet module

- Need to have installed a supported version of TET and Python
- Set TET_ROOT in the environment
- Check the settings of parameters in the Makefile
 - PYTHON_INC - Python includes directory, e.g. /usr/include/python2.2
- Build the pytet module using the make utility

Feb-05

TETware training course

THE *Open* GROUP

7-40

Building the `pytet` module

```
$ ls
Makefile      pytet_demo  pytet_patches  pytet.py      README
Makefile.swig  pytet.i    pytet_profile  pytet_wrap.c
$ make
/usr/bin/gcc -I/usr/include/python2.2 -c pytet_wrap.c
/usr/bin/gcc -shared -o _pytet.so pytet_wrap.o \
  /opt/lsb-tet3.6/lib/tet3/tcm.o \
  /opt/lsb tet3.6/lib/tet3/libapi.a
$
```

Feb-05

TETware training course

THE *Open* GROUP
7-41

Setup for Using the `pytet` module

- Set `PYTHONPATH` to the location of the shared library `_pytet.so` and module `pytet.py`
- Set `PATH` to include the location of `$TET_ROOT/bin` and location of the Python interpreter.
- See the `README` for more detailed instructions

Feb-05

TETware training course

THE *Open* GROUP
7-42

Using the Python TET module

- Import the Python TET module
- Define the startup and cleanup routines (optional)
- Define the test cases
- Initialize the test list
- Call `pytet_init()` to register the test list, startup and cleanup routines
- Invoke `tcc` to run the test cases

Feb-05

TETware training course

THE *Open* GROUP
7-43

Example Python test case

```
#!/usr/bin/env python
from pytet import *
def startup():
    print "Calling startup"
def cleanup():
    print "Calling cleanup"
def test1():
    try:
        var = tet_getvar("PYTET_VAR")
    except:
        tet_infoline("Failed to get value for PYTET_VAR")
        tet_result(TET_UNRESOLVED)
        return
    tet_infoline("PYTET_VAR is set to " + var)
    tet_result(TET_PASS)
def test2():
    tet_infoline("This is test 2")
    tet_result(TET_PASS)
testlist = { 1:test1, 2:test2 }
pytet_init(testlist, startup, cleanup)
```

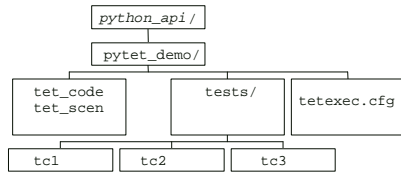
Feb-05

TETware training course

THE *Open* GROUP
7-44

The `pytet_demo` test suite

- The `pytet_demo` is included with the Python API as an example test suite



Feb-05

TETware training course

THE *Open* GROUP

7-45

Invoke `tcc` to run the `pytet_demo` test suite

- Ensure that your environment is set, see `pytet_profile`
- Assuming the `pytet_demo` is installed under `$TET_ROOT`, change to the `$TET_ROOT/pytet_demo` directory and invoke `tcc`:
`tcc -p -e`
- Otherwise set `TET_SUITE_ROOT` to where the `pytet_demo` directory is and invoke `tcc` (see next slide)

Feb-05

TETware training course

THE *Open* GROUP

7-46

Running the `pytet_demo` test suite

```

[python_api]$ ls
Makefile      pytet_demo  pytet_patches  pytet.py  _pytet.so  pytet_wrap.o
Makefile.swig  pytet.i    pytet_profile  pytet.pyc  pytet_wrap.c  README
[python_api]$ export TET_SUITE_ROOT=`pwd`
[python_api]$ cd pytet_demo
[pytet_demo]$ tcc -p -e
tcc: journal file is \
/opt/lab-tet3.6/contrib/python_api/pytet_demo/results/0002e/journal
07:41:28 Execute /tests/tc1
tc1: Calling startup
tc1: Calling cleanup
07:41:29 Execute /tests/tc2
07:41:30 Execute /tests/tc3
[pytet_demo]$
  
```

Feb-05

TETware training course

THE *Open* GROUP

7-47