7.5 - The Ruby API

- The Ruby API provides a binding to the TET C API
- All the functions provided by the API are in a single Ruby module `Rbtet`
- This module must be imported at the start of the test code

```
require “Rbtet”
```


Interface to the user-written code

- The Ruby 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 code
  - Thereby simplifying the writing of test code

Defining a Ruby Test Set

- The test set is wrapped in a Ruby class
- The test set class is a subclass of Rbtet::SuperTestSet

```
class TestSet < Rbtet::SuperTestSet
…
end
```


Defining the test list

- At the start of the test set class the test list is defined as a class variable hash list

```
@@testlist = Hash[1, “test1”, 2, “test2”]
```

- This example defines 2 tests for the test set

Defining the startup and cleanup methods

- Optional startup and cleanup methods can be defined in the test set class

```
def startup()
   Rbtet.tet_infoline("Calling startup")
end

def cleanup()
   Rbtet.tet_infoline("Calling cleanup")
end
```


Defining the test methods

- The test methods are defined in the test set class

```
test1()
    Rbtet.tet_infoline("Calling test 1")
end

test2()
    Rbtet.tet_infoline("Calling test 2")
end
```

Ruby API Features - Using the TET API

- Uses the same TET C API names
- The Ruby API names must be qualified by the module namespace

```
myvar = Rbtet.tet_getvar(“MY_CONFIG_VARIABLE”)
Rbtet.tet_infoline(“output text goes here”)
Rbtet.tet_result(Rbtet::TET_PASS)
```


Ruby API Features - Handling Errors

- Errors handling through exceptions

```
begin
   var = Rbtet.tet_getvar(“RBTET_VAR”)
rescue
   Rbtet.tet_infoline(“Failed to get value for
   RBTET_VAR”)
   Rbtet.tet_infoline(Rbtet::TET_UNRESOLVED)
   return
end
```

Note

- The Ruby TET API wrapper is generated using the SWIG toolkit that has some limitations which mean that not all the TET C API features are presently available in RbTET API, e.g. varargs (tet_printf() etc)

Building the Rbtet Module

- Need to have installed a supported version of TET and Ruby
- Set `TET_ROOT` in the environment
- Check the settings of parameters in the `Makefile`
  - RUBY_INC - Ruby include directory,
    e.g /usr/local/lib/ruby/1.8/i686-linux
- Build the `Rbtet` module using the `make` utility

Building the Rbtet Module

```
$ ls
Makefile    README      rbtet.i          rbtet_demo
      rbtet_wrap.c
Makefile.swig     Rbtet.rb    rbtet_profile

$make
/usr/bin/gcc -I/usr/local/lib/ruby/1.8/i686-linux -c
   rbtet_wrap.c \
   /usr/bin/gcc -shared -fPIC -o rbtet_ext.so rbtet_wrap.o \
   /user2/tet/lib/tet3/tcm.o /user2/tet/lib/tet3/libapi.a
$
```


Setup for using the Rbtet module

- Set RUBYPATH to the location of the Rbtet module (Rbtet.rb)
- Set RUBYLIB to the location of the Rbtet shared library (rbtet_exs.so)
- Set PATH to include the location of the $TET_ROOT/bin and the Ruby interpreter
- See the README for more detailed instructions

Using the Ruby TET Module

- Import the Rbtet module
- Define the test set class
- Define the test list class variable
- Define the startup and cleanup methods (optional)
- Define the test methods

Using the Ruby TET Module

- Register the test set instance, startup and cleanup methods with `Rbtet.rbtet_init()`
- Invoke `tcc` to run the test cases

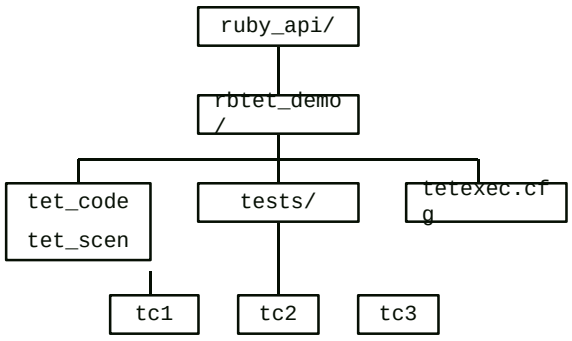
Example Ruby test case

```
require "Rbtet"

class TestSet < Rbtet::SuperTestSet
    @@testlist = Hash[1,"test1", 2,"test2"]

    def startup()
        Rbtet.tet_infoline("Test startup")
    end

    def cleanup()
        Rbtet.tet_infoline("Test cleanup")
    end

    def test1()
        begin
            var = Rbtet.tet_getvar("RBTET_TC1_VAR")
        rescue
            Rbtet.tet_infoline("Failed to get a value for RBTET_TC1_VAR")
            Rbtet.tet_result(Rbtet::TET_UNRESOLVED)
            return
        end
        Rbtet.tet_infoline("RBTET_TC1_VAR is set to " + var)
        Rbtet.tet_result(Rbtet::TET_PASS)
    end

    def test2()
      Rbtet.tet_result(Rbtet::TET_PASS)
    end
end

Rbtet.rbtet_init(TestSet.new, "startup", "cleanup")
```


THE Open GROUP

The rbtet_demo test suite

- The rbtet_demo is included with the Ruby API as an example test suite



THE Open GROUP

Invoke tcc to run the rbtet_demo test suite

- Ensure that you environment is set, see `rbtet_profile`
- Assuming rbtet_demo is installed under `TET_ROOT`, change to the directory and invoke `tcc`:
  - `tcc -p -e`
- Otherwise set `TET_SUITE_ROOT` to where the rbtet_demo directory is and invoke `tcc` (See next slide)

Running the rbtet_demo test suite

```
[rbtet_demo] ls
results  tests  tet_code  tet_scen  tetexec.cfg
[rbtet_demo] export TET_SUITE_ROOT=`pwd`
[rbtet_demo] tcc -p -e .
tcc: journal file is
   /home/neil/work/rbtet/rbtet_demo/results/0001e/journal
11:58:08  Execute /tests/tc1
11:58:09  Execute /tests/tc2
11:58:10  Execute /tests/tc3
[rbtet_demo]
```