# STB Tester

Author: Francis Horsman.

## Improvements

### Code-base

* Now OO, in logical packages (src: (Core, Remote, Playback, Record, Api), test)
* Includes a UI, can still be run from the command-line.
* Top-Level-Test api namespace now unique – does not use application global namespace!!
* Top-Level-Tests execute in a sand-boxed namespace optionally with isolation from each other and optionally without access to Python's '***builtin***' methods.
* Program arguments have changed slightly and been extended.
* Camel-case used throughout.
* Non-public class/instance attributes prefixed with '***\_***'.
* Nose can be used programatically to discover tests to execute and can be driven by CI system.
* Top-level-scripts generate xUnit output for easy CI system integration.
* Top-level-scripts can use Nosetests framework i.e.:
  + setUp(),
  + tearDown(),
  + unittest.TestCase,
  + etc.
* Can theoretically be easily integrated into YATES.

### Tests

* Tests can include other tests and optionally any python module available on the host system.
* Included tests match on images in their own folder (change made to findPath).
* Tests can require a particular api name/version (min, max, exact) be available:
  + ***\_\_requires\_\_( [ api descriptions ] )***
* Tests can dynamically import apis from the file-system into the top-level-test's namespace (and all imported tests there-after).
  + ***\_\_loads\_\_( [ api descriptions ] , mode )***
* Tests can dynamically inject an api's methods into an existing apis's namespace.
  + ***\_\_inject\_\_( [ api descriptions ] , mode )***
* Tests have access to their:
  + **\_\_file\_\_** Full path of the test script ***.py***/***.pyc*** file, eg: ***/home/output/scripts/test4.py***
  + **\_\_name\_\_** Full import path for the test, eg: ***output.scripts.test4***
  + **\_\_scrdir\_\_,** Directory of the top-level-test.
  + **\_\_script\_root\_\_** Common root directory for all tests as specified in the config/cmdline (an 'actual' common root directory may not exist).

### Api

* Api now checks for 'aborted' to allow exit from test run, any loops, IO or 3rd party blocking calls will wait until they complete first.
* Api includes safe method for sleeping to allow test aborting.
* Apis will automagically become derived classes of **BaseApi** if not already, see BaseApi docstring for further information.

### UI (Playback only).

* Allows running of scripts.
* Allows stepping through the scripts before and after each api call or call to \_\_requires\_\_, \_\_loads\_\_, \_\_inject\_\_.
* Shows which api and api-methods were made, the time and associated parameters.
* Provides a top-level-script status synopsis (executing, finished, error'd).
* Displays the gstreamer output in the UI rather than a seperate window which grabs focus when it is created (frequently when testing !).
* Contains a program event log (ui actions, api calls, etc.).

## Todo

1. SigInt handling and propagation: test termination.
2. Convert all installation tests to Python Nose/Unittest format.

## Fixes:

### Bugs

1. '**detect**' method '**timeout**' argument didn't work, always returned too quickly.

### Security

1. Tests executed in the application global namespace (highly dangerous and open to errors in the tests or their imported modules) or malicious code.

## Changes required for existing customers.

### Test scripts

1. All api calls prefixed with a namespace which is defined in the api.NAMESPACE attribute of the given api used, this can be overridden using **\_\_loads\_\_**.
2. Scripts can import other scripts.
3. Scripts must use the following format for controlling execution flow (due to imports):

***\_\_requires\_\_({“name”:”YV\_HUM\_ALPHA”, “min”:1.1}) # optional.***

def **test\_something()**:

**...**

***\_\_loads\_\_([“apis/x/b?ns=new\_ns”]) # optional.***

**new\_ns.someMethod(...)**

**....**

if \_\_name\_\_ == *'\_\_main\_\_'*:

test\_something**()**

1. The launch file has changed name and location, unclear if this will work with existing installation script.

### Installation

1. Large proportion of installation tests broken (b\*\*tardised bash/python offspring)

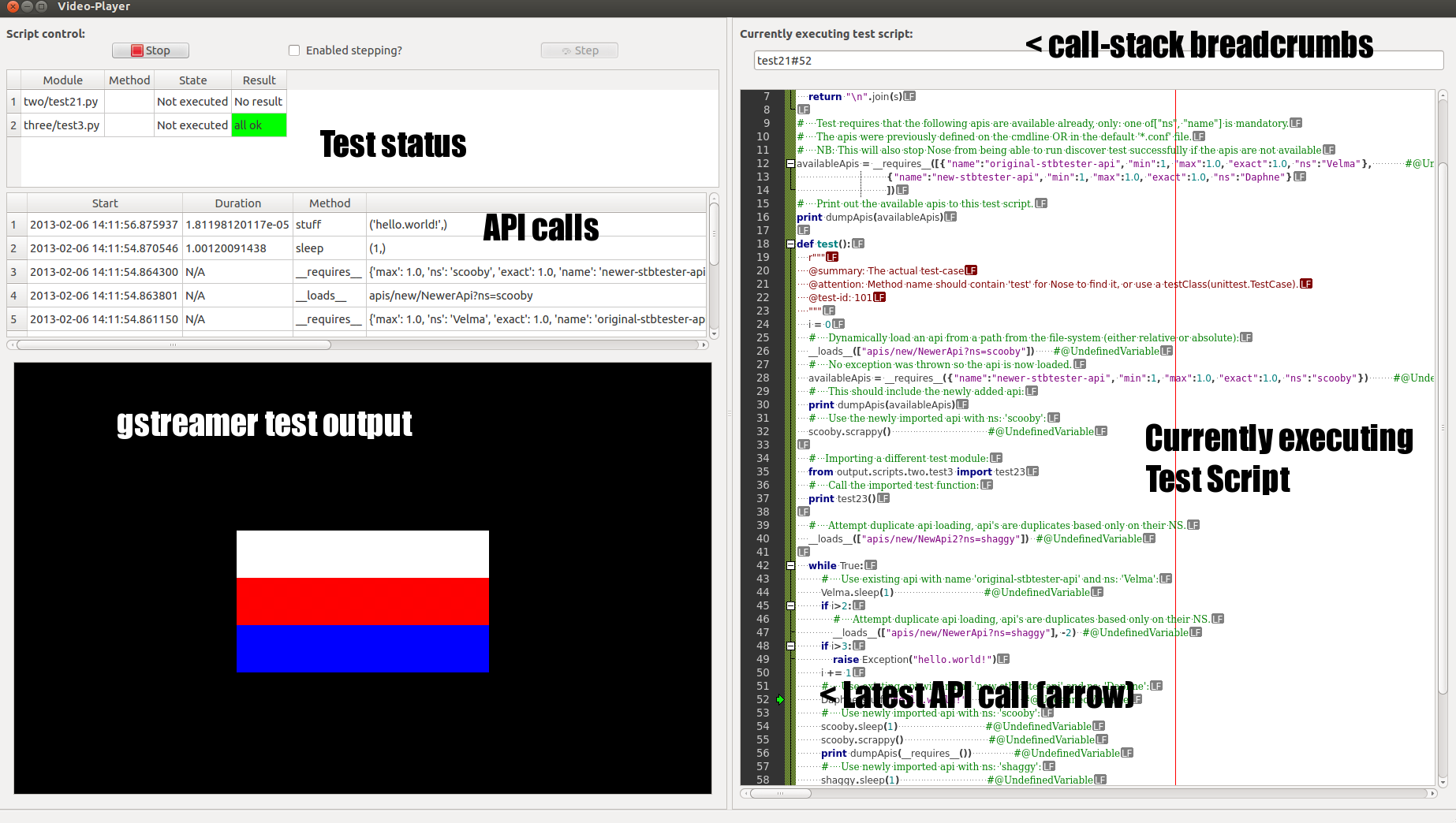
### UI

In order to run the UI, the following dependencies must be satisfied:

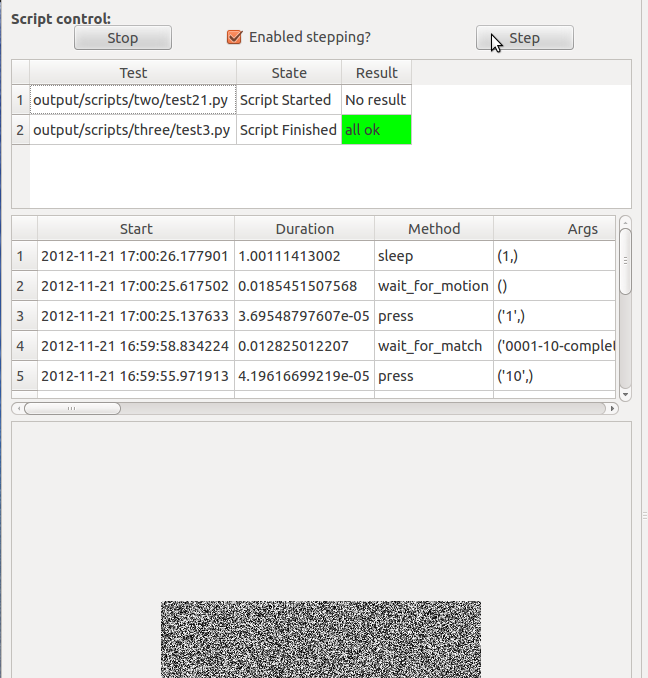
* **PyQt4**
* **PyQt4.Qsci (QsciScintilla)**

### Images

1. Main Gui showing the various panels.



2. Left panels with one test complete and the second test step-paused.



3. Left panel at the end of the test, the second test has raised an exception.

