Running Tests from the Command Line

This document explains how to run the Tests generated by Introspect ESP from the command line (or command-oriented environments).

When you use Introspect ESP to develop a Test, you usually run that Test via the “Run” button in the Introspect ESP GUI. But in some cases it is useful to be able to run the Test from a command-line environment (e.g. a DOS window, or from a MATLAB script, or from LabVIEW, etc.). The Python script “runSvtTest.py” (supplied in the “Python” folder of your Introspect ESP installation) can be used to run a Test from the command line.

Python Version

The first requirement to be able to use the “runSvtTest.py” script is that your command-line environment must be using the same version of Python as that supported by Introspect ESP. The current version of Introspect ESP uses Python 2.7 and so your command-line environment must be using some 2.7.x version of Python. If you get an error message about a “bad magic number”, this means that you are using an incompatible version of Python. You can check which version of Python is the default in your command-line environment by executing the command:

```
python –version
```

But if the default version of Python in your command-line environment is not compatible with Introspect ESP, you can work around this by using the full path to the appropriate Python interpreter when you use the “runSvtTest.py” script (e.g. by using “C:\Python27\python” instead of just “python”).

Python Module Search Path

The “runSvtTest.py” script uses the Python modules supplied with Introspect ESP, and in order for this to work, the Python interpreter must be able to find them. Python looks for modules in the folder where the script is, in the folders specified in the PYTHONPATH variable, and in the Python system folders. The search path used when importing modules is in the Python variable ‘sys.path’ – this is initialized from the folders mentioned above when the script is started.

Since the “runSvtTest.py” script is in the same folder as the “dftm” folder which is the root of the Python modules supplied with Introspect ESP, there should be no problem. But if you move the “runSvtTest.py” script to some other folder, you will need to tell the Python interpreter where to find these modules. You can either set the PYTHONPATH environment variable to include the parent folder of the “dftm” folder, or append this folder to ‘sys.path’ in your command-line environment (e.g. an external Python script that is invoking the “runSvtTest.py” script).

For example, if your Introspect ESP installation folder is “C:\MyStuff\IntrospectESP” and the “dftm” folder with the Introspect ESP Python modules is in the “Python” sub-folder of that installation folder, then you could set the PYTHONPATH environment variable with the following command in a DOS window:

```
set PYTHONPATH=C:\MyStuff\IntrospectESP\Python
```

or you could add the following line near the top of your external Python script:

```
sys.path.append(r”C:\MyStuff\IntrospectESP\Python”)
```

Note the use of the ‘r’ prefix to get a raw string – otherwise the backslashes would cause problems.

Command-line arguments for the runSvtTest.py script

The “runSvtTest.py” script expects at least one command-line argument which specifies the path to the Test folder. (Note that a Test folder has sub-folders “Params” (where the “testProcedure.py” file is) and “Results” (where folders of measurement data are kept).) You will usually also supply command-line arguments to specify which options you want. You can see all available command-line options by using the option “--help”. The command-line options all start with a double minus (“--“)

The IESP hardware comes in various form factors. You will usually need to specify the form factor via the “--formFactor” command-line option when you run the “runSvtTest.py” script.

A form factor can have one or more “sub-parts”. For example, the “DV1600” form factor has sub-parts named “moduleA” and “moduleB”. To specify which subparts are to be enabled, you use the “--enabledSubParts” option.

For example, if the Test that you want to run is in the folder "C:\MyStuff\MyTests\TestA", and you are using the DV1600 form factor with both modules enabled, you would run the Test via a command like:

```
python runSvtTest.py --formFactor DV1600 --
enabledSubParts moduleA,moduleB
"C:\MyStuff\MyTests\TestA"
```

The above command should be all on one line (it has wrapped to fit the margins of this document). Note that it is necessary to put the path to the Test folder inside double-quotes if there are any spaces in any folder names in the path.

If your environment has been configured to run ".py" files in Python, you may be able to omit the "python" at the beginning of that command. But if the default python in your command-line environment is not compatible with that required by Introspect ESP (see the "Python Version" section above), you should use the full path to the appropriate Python interpreter (e.g. use "C:\Python27\python" instead of just "python").

Some IESP hardware supports multiple firmware personalities. If you are using a form factor that supports multiple personalities, your script needs to specify which of these firmware personalities to use via the "--defaultPersonality" option.

What the runSvtTest.py script does

After handling the command-line options, the "runSvtTest.py" script does three things:

- Connect to the IESP hardware and initialize the hardware
- Load the Test from the specified folder
- Run the Test

The "runSvtTest.py" script will exit if an error occurs during any of these three operations. If the Test runs to completion without errors, the "runSvtTest.py" script will exit with a zero exit code. This exit code can be queried in most command-line environments to determine success or failure.

If the Test used components that generate measurement data, that data is saved into Result folders under the "Results" sub-folder of the Test folder – the same as happens when you run the Test via the "Run" button in the Introspect ESP GUI. So you can run Tests via the command-line and then later use the Introspect ESP GUI to look at the results.