

Runtime Testing with Spansion FR Softune Support

Testing on the Simulator

The Test Configurations provided with C++test are prepared for testing on the simulator using the semi-hosting feature.

To enable this, add the IOPORT address to the linker command line. For example:

- `"-sc IOPORT=0x00000000"`

This additional option should be specified in the linker options within the C++test Build Settings (right-click that project in the project tree, choose **Properties** from the shortcut menu, then select choose **Parasoft> C++test> Build Settings**).

The INIT section defined in the RAM area is required for unit testing. Refer to the SOFTUNE manual for instructions.

Customizing Built-in Test Configurations

The Test Configurations provided to execute runtime tests for FR Softune projects ("Builtin> Embedded Systems> Spansion FR Softune") may require environment-specific customization. The typical customizations are done with the help of test flow definition properties that are listed for each Test Configuration (in the Execution> General tab, "Execution details" section).

The table below describes the test flow properties for the dedicated FR Softune built-in Test Configuration, which is named "Run Spansion FR Softune Tests - Simulator":

Name	Default Value	Description
Exit Point	<code>__exit</code>	Exit point of the application. This is the point at which the simulator will end the execution.
Directory Path for Test Executable	<code>\${workspace_loc}/ \${project_name}/Debug/ \${project_name}Test.abs</code>	The text executable will be generated in this location. Usually, this should be the same location as the original binary.

Unit Testing

The Run Spansion FR Softune Tests - Simulator Test Configuration is provided to facilitate the unit testing process.

To run unit testing on the simulator:

1. Manually create or automatically generate a set of test cases.
2. Duplicate the FR Softune unit test configuration.
3. Modify the Test Configuration as needed.
 - See [Customizing Built-in Test Configurations](#) for details.
4. Select the desired testing context.
5. Run tests with the customized Test Configuration created above.

Debugging Test Cases

Use External Embedded debugging mode. For more details see:

- [Configuring Debugger Settings](#)
- [Debugging Test Cases](#)