

Runtime Testing with Lauterbach TRACE32 Support

This section explains how to configure and run runtime testing. It covers:

- [Customizing Built-in Test Configurations](#)
- [Unit Testing](#)
- [Application Monitoring](#)

Customizing Built-in Test Configurations

C++test provides the following Test Configurations to run tests using TRACE32 tools:

- Embedded Systems/Lauterbach TRACE32/Run Tests using TRACE32 (FDX)
- Embedded Systems/Lauterbach TRACE32/Run Application with Memory Monitoring using TRACE32 (FDX)

Test Configurations can be customized using the following test execution flow properties:

- **Core:** The value of this property is used as an argument for "System.CPU" debugger command. Specify your development target's core as the value of this property.
- **Test executable exit point:** The value of this property will be used to set a break point (Break.Set debugger command) that will mark the end of test execution. Typically, `_exit` symbol or similar should be used for this purpose.
- **Path to TRACE32 executable:** Used to run TRACE32 debugger with the automatically-generated test script (`t32xxx.exe -s <test execution script>`). Select a debugger program executable from your development environment.
- **TRACE32 configuration file:** Should be specified only if a non-default configuration file must be used for the TRACE32 executable (`t32xxx.exe -c <config file>`)

Lauterbach TRACE32 test configurations automate:

- Test executable preparation
- Test executable upload to the target or simulator
- Test execution
- Results collection

Test executable uploading, execution and results collection is performed with help of a Lauterbach PRACTICE II script that is generated by C++test. The script is generated based on a template available in `<C++test_install_dir>\engine\etc\templates\for_recipes\t32_fdx.tja`.

You can customize this template as needed. To use a customized version of this script template:

1. Make a copy of the original version of this script.
2. Edit it as desired.
3. Modify the Test Configuration's test flow definition to use the customized version rather than the original script template. The script template path is specified in a flow step with `id="db_scr_gen"`.

Unit Testing

The "Embedded Systems/Lauterbach TRACE32/Run Tests using TRACE32 (FDX)" Test Configuration is provided to facilitate the unit testing process.

To run unit testing using TRACE32:

1. Manually create or automatically generate a set of test cases.
2. Duplicate the "Run Tests using TRACE32 (FDX)" 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](#)

Application Monitoring

To run application monitoring using TRACE32:

1. Duplicate the "Embedded Systems/Lauterbach TRACE32/Run Application with Memory Monitoring using TRACE32 (FDX)" Test Configuration.
2. Modify the Test Configuration as needed. See [Customizing Built-in Test Configurations](#) for details.
3. Select the desired testing context.
4. Run tests with the customized Test Configuration created above.

Learning More

For general information on performing application monitoring and runtime error detection with C++test, see [Runtime Error Detection](#).