

Runtime Exceptions

This topic provides details about results in the Runtime Exception category.

Description

A Fix Unit Test Problems> Runtime Exceptions message indicates that a test case causes an unexpected exception (an exception that is not accounted for and handled in the test case code). An exception indicates code that is not sufficiently robust. If exceptions surface in the field, the resulting unexpected flow transfer and potential thread termination could lead to instability, unexpected results, or even crashes or security breaches.

If an exception occurs when a deployed application is running, it can cause system and application instability, security vulnerabilities (such as denial of service attacks), and frequent down time.

If an exception is reported during unit testing, it indicates that the execution of a test case creates conditions under which the code under test throws the exception. After the code under test is integrated into the application, the application might create these same conditions and force the code to throw the exception. As explained in the previous paragraph, any occurrence of this exception in a running deployed application is a critical problem.

Recommended Response

The exceptions that are reported in this category indicate incorrectly behaving code—code that should not throw an exception for the given arguments.

We recommend that you respond to all of the reported runtime exceptions before you work on other code. If the code requires modification, it is faster and easier to fix it as soon you learn about it than to wait until later in the development process. If the code is behaving correctly, you can prevent confusion and the introduction of errors by ensuring that this behavior is clearly documented; when other developers working with the code know exactly how the code is supposed to behave, they will be less likely to introduce errors.

When an exception is reported during unit testing, the recommended response is:

- Use a debugger to investigate the reason for the failure. See [Using a Debugger During Test Execution](#) for details.
- Determine whether the exception indicates a bug in the code under test, or is caused by the test harness that is calling the code under test.
- If the exception is caused by a bug in the code under test, correct the code.
- If the exception is intentional, change the test case registration to use `CPPTTEST_TEST_EXCEPTION` as described in [Test Suite/Test Case Registration Macros](#).
- Often, the exception is not caused by the code under test (which is where the exception manifests itself), but rather caused by the code that calls it. Review the application code that calls the code under test to ensure that the application cannot produce the known exception-causing conditions exposed by the test harness.

Remember that the code under test might later be called by code that is not under your control—and which might create the conditions that could cause this exception. As a result, it's essential to protect the code against the exception-causing inputs.