The non-test part of the code is expected to have 2 failures.

googletest-output-test\_.cc:#: Failure

Value of: false

Actual: false

Expected: true

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

2

3

Stack trace: (omitted)

[0;32m[==========] [mRunning 88 tests from 41 test suites.

[0;32m[----------] [mGlobal test environment set-up.

FooEnvironment::SetUp() called.

BarEnvironment::SetUp() called.

[0;32m[----------] [m1 test from ADeathTest

[0;32m[ RUN ] [mADeathTest.ShouldRunFirst

[0;32m[ OK ] [mADeathTest.ShouldRunFirst

[0;32m[----------] [m1 test from ATypedDeathTest/0, where TypeParam = int

[0;32m[ RUN ] [mATypedDeathTest/0.ShouldRunFirst

[0;32m[ OK ] [mATypedDeathTest/0.ShouldRunFirst

[0;32m[----------] [m1 test from ATypedDeathTest/1, where TypeParam = double

[0;32m[ RUN ] [mATypedDeathTest/1.ShouldRunFirst

[0;32m[ OK ] [mATypedDeathTest/1.ShouldRunFirst

[0;32m[----------] [m1 test from My/ATypeParamDeathTest/0, where TypeParam = int

[0;32m[ RUN ] [mMy/ATypeParamDeathTest/0.ShouldRunFirst

[0;32m[ OK ] [mMy/ATypeParamDeathTest/0.ShouldRunFirst

[0;32m[----------] [m1 test from My/ATypeParamDeathTest/1, where TypeParam = double

[0;32m[ RUN ] [mMy/ATypeParamDeathTest/1.ShouldRunFirst

[0;32m[ OK ] [mMy/ATypeParamDeathTest/1.ShouldRunFirst

[0;32m[----------] [m2 tests from PassingTest

[0;32m[ RUN ] [mPassingTest.PassingTest1

[0;32m[ OK ] [mPassingTest.PassingTest1

[0;32m[ RUN ] [mPassingTest.PassingTest2

[0;32m[ OK ] [mPassingTest.PassingTest2

[0;32m[----------] [m2 tests from NonfatalFailureTest

[0;32m[ RUN ] [mNonfatalFailureTest.EscapesStringOperands

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

kGoldenString

Which is: "\"Line"

actual

Which is: "actual \"string\""

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

golden

Which is: "\"Line"

actual

Which is: "actual \"string\""

Stack trace: (omitted)

[0;31m[ FAILED ] [mNonfatalFailureTest.EscapesStringOperands

[0;32m[ RUN ] [mNonfatalFailureTest.DiffForLongStrings

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

golden\_str

Which is: "\"Line\0 1\"\nLine 2"

"Line 2"

With diff:

@@ -1,2 @@

-\"Line\0 1\"

Line 2

Stack trace: (omitted)

[0;31m[ FAILED ] [mNonfatalFailureTest.DiffForLongStrings

[0;32m[----------] [m3 tests from FatalFailureTest

[0;32m[ RUN ] [mFatalFailureTest.FatalFailureInSubroutine

(expecting a failure that x should be 1)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

x

Which is: 2

Stack trace: (omitted)

[0;31m[ FAILED ] [mFatalFailureTest.FatalFailureInSubroutine

[0;32m[ RUN ] [mFatalFailureTest.FatalFailureInNestedSubroutine

(expecting a failure that x should be 1)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

x

Which is: 2

Stack trace: (omitted)

[0;31m[ FAILED ] [mFatalFailureTest.FatalFailureInNestedSubroutine

[0;32m[ RUN ] [mFatalFailureTest.NonfatalFailureInSubroutine

(expecting a failure on false)

googletest-output-test\_.cc:#: Failure

Value of: false

Actual: false

Expected: true

Stack trace: (omitted)

[0;31m[ FAILED ] [mFatalFailureTest.NonfatalFailureInSubroutine

[0;32m[----------] [m1 test from LoggingTest

[0;32m[ RUN ] [mLoggingTest.InterleavingLoggingAndAssertions

(expecting 2 failures on (3) >= (a[i]))

i == 0

i == 1

googletest-output-test\_.cc:#: Failure

Expected: (3) >= (a[i]), actual: 3 vs 9

Stack trace: (omitted)

i == 2

i == 3

googletest-output-test\_.cc:#: Failure

Expected: (3) >= (a[i]), actual: 3 vs 6

Stack trace: (omitted)

[0;31m[ FAILED ] [mLoggingTest.InterleavingLoggingAndAssertions

[0;32m[----------] [m7 tests from SCOPED\_TRACETest

[0;32m[ RUN ] [mSCOPED\_TRACETest.AcceptedValues

googletest-output-test\_.cc:#: Failure

Failed

Just checking that all these values work fine.

Google Test trace:

googletest-output-test\_.cc:#: (null)

googletest-output-test\_.cc:#: 1337

googletest-output-test\_.cc:#: std::string

googletest-output-test\_.cc:#: literal string

Stack trace: (omitted)

[0;31m[ FAILED ] [mSCOPED\_TRACETest.AcceptedValues

[0;32m[ RUN ] [mSCOPED\_TRACETest.ObeysScopes

(expected to fail)

googletest-output-test\_.cc:#: Failure

Failed

This failure is expected, and shouldn't have a trace.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

This failure is expected, and should have a trace.

Google Test trace:

googletest-output-test\_.cc:#: Expected trace

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

This failure is expected, and shouldn't have a trace.

Stack trace: (omitted)

[0;31m[ FAILED ] [mSCOPED\_TRACETest.ObeysScopes

[0;32m[ RUN ] [mSCOPED\_TRACETest.WorksInLoop

(expected to fail)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

2

n

Which is: 1

Google Test trace:

googletest-output-test\_.cc:#: i = 1

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

n

Which is: 2

Google Test trace:

googletest-output-test\_.cc:#: i = 2

Stack trace: (omitted)

[0;31m[ FAILED ] [mSCOPED\_TRACETest.WorksInLoop

[0;32m[ RUN ] [mSCOPED\_TRACETest.WorksInSubroutine

(expected to fail)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

2

n

Which is: 1

Google Test trace:

googletest-output-test\_.cc:#: n = 1

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

n

Which is: 2

Google Test trace:

googletest-output-test\_.cc:#: n = 2

Stack trace: (omitted)

[0;31m[ FAILED ] [mSCOPED\_TRACETest.WorksInSubroutine

[0;32m[ RUN ] [mSCOPED\_TRACETest.CanBeNested

(expected to fail)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

n

Which is: 2

Google Test trace:

googletest-output-test\_.cc:#: n = 2

googletest-output-test\_.cc:#:

Stack trace: (omitted)

[0;31m[ FAILED ] [mSCOPED\_TRACETest.CanBeNested

[0;32m[ RUN ] [mSCOPED\_TRACETest.CanBeRepeated

(expected to fail)

googletest-output-test\_.cc:#: Failure

Failed

This failure is expected, and should contain trace point A.

Google Test trace:

googletest-output-test\_.cc:#: A

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

This failure is expected, and should contain trace point A and B.

Google Test trace:

googletest-output-test\_.cc:#: B

googletest-output-test\_.cc:#: A

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

This failure is expected, and should contain trace point A, B, and C.

Google Test trace:

googletest-output-test\_.cc:#: C

googletest-output-test\_.cc:#: B

googletest-output-test\_.cc:#: A

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

This failure is expected, and should contain trace point A, B, and D.

Google Test trace:

googletest-output-test\_.cc:#: D

googletest-output-test\_.cc:#: B

googletest-output-test\_.cc:#: A

Stack trace: (omitted)

[0;31m[ FAILED ] [mSCOPED\_TRACETest.CanBeRepeated

[0;32m[ RUN ] [mSCOPED\_TRACETest.WorksConcurrently

(expecting 6 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #1 (in thread B, only trace B alive).

Google Test trace:

googletest-output-test\_.cc:#: Trace B

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #2 (in thread A, trace A & B both alive).

Google Test trace:

googletest-output-test\_.cc:#: Trace A

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #3 (in thread B, trace A & B both alive).

Google Test trace:

googletest-output-test\_.cc:#: Trace B

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #4 (in thread B, only trace A alive).

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #5 (in thread A, only trace A alive).

Google Test trace:

googletest-output-test\_.cc:#: Trace A

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #6 (in thread A, no trace alive).

Stack trace: (omitted)

[0;31m[ FAILED ] [mSCOPED\_TRACETest.WorksConcurrently

[0;32m[----------] [m1 test from ScopedTraceTest

[0;32m[ RUN ] [mScopedTraceTest.WithExplicitFileAndLine

googletest-output-test\_.cc:#: Failure

Failed

Check that the trace is attached to a particular location.

Google Test trace:

explicit\_file.cc:123: expected trace message

Stack trace: (omitted)

[0;31m[ FAILED ] [mScopedTraceTest.WithExplicitFileAndLine

[0;32m[----------] [m1 test from NonFatalFailureInFixtureConstructorTest

[0;32m[ RUN ] [mNonFatalFailureInFixtureConstructorTest.FailureInConstructor

(expecting 5 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #1, in the test fixture c'tor.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #2, in SetUp().

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #3, in the test body.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #4, in TearDown.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #5, in the test fixture d'tor.

Stack trace: (omitted)

[0;31m[ FAILED ] [mNonFatalFailureInFixtureConstructorTest.FailureInConstructor

[0;32m[----------] [m1 test from FatalFailureInFixtureConstructorTest

[0;32m[ RUN ] [mFatalFailureInFixtureConstructorTest.FailureInConstructor

(expecting 2 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #1, in the test fixture c'tor.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #2, in the test fixture d'tor.

Stack trace: (omitted)

[0;31m[ FAILED ] [mFatalFailureInFixtureConstructorTest.FailureInConstructor

[0;32m[----------] [m1 test from NonFatalFailureInSetUpTest

[0;32m[ RUN ] [mNonFatalFailureInSetUpTest.FailureInSetUp

(expecting 4 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #1, in SetUp().

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #2, in the test function.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #3, in TearDown().

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #4, in the test fixture d'tor.

Stack trace: (omitted)

[0;31m[ FAILED ] [mNonFatalFailureInSetUpTest.FailureInSetUp

[0;32m[----------] [m1 test from FatalFailureInSetUpTest

[0;32m[ RUN ] [mFatalFailureInSetUpTest.FailureInSetUp

(expecting 3 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #1, in SetUp().

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #2, in TearDown().

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected failure #3, in the test fixture d'tor.

Stack trace: (omitted)

[0;31m[ FAILED ] [mFatalFailureInSetUpTest.FailureInSetUp

[0;32m[----------] [m1 test from AddFailureAtTest

[0;32m[ RUN ] [mAddFailureAtTest.MessageContainsSpecifiedFileAndLineNumber

foo.cc:42: Failure

Failed

Expected nonfatal failure in foo.cc

Stack trace: (omitted)

[0;31m[ FAILED ] [mAddFailureAtTest.MessageContainsSpecifiedFileAndLineNumber

[0;32m[----------] [m1 test from GtestFailAtTest

[0;32m[ RUN ] [mGtestFailAtTest.MessageContainsSpecifiedFileAndLineNumber

foo.cc:42: Failure

Failed

Expected fatal failure in foo.cc

Stack trace: (omitted)

[0;31m[ FAILED ] [mGtestFailAtTest.MessageContainsSpecifiedFileAndLineNumber

[0;32m[----------] [m4 tests from MixedUpTestSuiteTest

[0;32m[ RUN ] [mMixedUpTestSuiteTest.FirstTestFromNamespaceFoo

[0;32m[ OK ] [mMixedUpTestSuiteTest.FirstTestFromNamespaceFoo

[0;32m[ RUN ] [mMixedUpTestSuiteTest.SecondTestFromNamespaceFoo

[0;32m[ OK ] [mMixedUpTestSuiteTest.SecondTestFromNamespaceFoo

[0;32m[ RUN ] [mMixedUpTestSuiteTest.ThisShouldFail

gtest.cc:#: Failure

Failed

All tests in the same test suite must use the same test fixture

class. However, in test suite MixedUpTestSuiteTest,

you defined test FirstTestFromNamespaceFoo and test ThisShouldFail

using two different test fixture classes. This can happen if

the two classes are from different namespaces or translation

units and have the same name. You should probably rename one

of the classes to put the tests into different test suites.

Stack trace: (omitted)

[0;31m[ FAILED ] [mMixedUpTestSuiteTest.ThisShouldFail

[0;32m[ RUN ] [mMixedUpTestSuiteTest.ThisShouldFailToo

gtest.cc:#: Failure

Failed

All tests in the same test suite must use the same test fixture

class. However, in test suite MixedUpTestSuiteTest,

you defined test FirstTestFromNamespaceFoo and test ThisShouldFailToo

using two different test fixture classes. This can happen if

the two classes are from different namespaces or translation

units and have the same name. You should probably rename one

of the classes to put the tests into different test suites.

Stack trace: (omitted)

[0;31m[ FAILED ] [mMixedUpTestSuiteTest.ThisShouldFailToo

[0;32m[----------] [m2 tests from MixedUpTestSuiteWithSameTestNameTest

[0;32m[ RUN ] [mMixedUpTestSuiteWithSameTestNameTest.TheSecondTestWithThisNameShouldFail

[0;32m[ OK ] [mMixedUpTestSuiteWithSameTestNameTest.TheSecondTestWithThisNameShouldFail

[0;32m[ RUN ] [mMixedUpTestSuiteWithSameTestNameTest.TheSecondTestWithThisNameShouldFail

gtest.cc:#: Failure

Failed

All tests in the same test suite must use the same test fixture

class. However, in test suite MixedUpTestSuiteWithSameTestNameTest,

you defined test TheSecondTestWithThisNameShouldFail and test TheSecondTestWithThisNameShouldFail

using two different test fixture classes. This can happen if

the two classes are from different namespaces or translation

units and have the same name. You should probably rename one

of the classes to put the tests into different test suites.

Stack trace: (omitted)

[0;31m[ FAILED ] [mMixedUpTestSuiteWithSameTestNameTest.TheSecondTestWithThisNameShouldFail

[0;32m[----------] [m2 tests from TEST\_F\_before\_TEST\_in\_same\_test\_case

[0;32m[ RUN ] [mTEST\_F\_before\_TEST\_in\_same\_test\_case.DefinedUsingTEST\_F

[0;32m[ OK ] [mTEST\_F\_before\_TEST\_in\_same\_test\_case.DefinedUsingTEST\_F

[0;32m[ RUN ] [mTEST\_F\_before\_TEST\_in\_same\_test\_case.DefinedUsingTESTAndShouldFail

gtest.cc:#: Failure

Failed

All tests in the same test suite must use the same test fixture

class, so mixing TEST\_F and TEST in the same test suite is

illegal. In test suite TEST\_F\_before\_TEST\_in\_same\_test\_case,

test DefinedUsingTEST\_F is defined using TEST\_F but

test DefinedUsingTESTAndShouldFail is defined using TEST. You probably

want to change the TEST to TEST\_F or move it to another test

case.

Stack trace: (omitted)

[0;31m[ FAILED ] [mTEST\_F\_before\_TEST\_in\_same\_test\_case.DefinedUsingTESTAndShouldFail

[0;32m[----------] [m2 tests from TEST\_before\_TEST\_F\_in\_same\_test\_case

[0;32m[ RUN ] [mTEST\_before\_TEST\_F\_in\_same\_test\_case.DefinedUsingTEST

[0;32m[ OK ] [mTEST\_before\_TEST\_F\_in\_same\_test\_case.DefinedUsingTEST

[0;32m[ RUN ] [mTEST\_before\_TEST\_F\_in\_same\_test\_case.DefinedUsingTEST\_FAndShouldFail

gtest.cc:#: Failure

Failed

All tests in the same test suite must use the same test fixture

class, so mixing TEST\_F and TEST in the same test suite is

illegal. In test suite TEST\_before\_TEST\_F\_in\_same\_test\_case,

test DefinedUsingTEST\_FAndShouldFail is defined using TEST\_F but

test DefinedUsingTEST is defined using TEST. You probably

want to change the TEST to TEST\_F or move it to another test

case.

Stack trace: (omitted)

[0;31m[ FAILED ] [mTEST\_before\_TEST\_F\_in\_same\_test\_case.DefinedUsingTEST\_FAndShouldFail

[0;32m[----------] [m8 tests from ExpectNonfatalFailureTest

[0;32m[ RUN ] [mExpectNonfatalFailureTest.CanReferenceGlobalVariables

[0;32m[ OK ] [mExpectNonfatalFailureTest.CanReferenceGlobalVariables

[0;32m[ RUN ] [mExpectNonfatalFailureTest.CanReferenceLocalVariables

[0;32m[ OK ] [mExpectNonfatalFailureTest.CanReferenceLocalVariables

[0;32m[ RUN ] [mExpectNonfatalFailureTest.SucceedsWhenThereIsOneNonfatalFailure

[0;32m[ OK ] [mExpectNonfatalFailureTest.SucceedsWhenThereIsOneNonfatalFailure

[0;32m[ RUN ] [mExpectNonfatalFailureTest.FailsWhenThereIsNoNonfatalFailure

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenThereIsNoNonfatalFailure

[0;32m[ RUN ] [mExpectNonfatalFailureTest.FailsWhenThereAreTwoNonfatalFailures

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual: 2 failures

googletest-output-test\_.cc:#: Non-fatal failure:

Failed

Expected non-fatal failure 1.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Non-fatal failure:

Failed

Expected non-fatal failure 2.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenThereAreTwoNonfatalFailures

[0;32m[ RUN ] [mExpectNonfatalFailureTest.FailsWhenThereIsOneFatalFailure

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual:

googletest-output-test\_.cc:#: Fatal failure:

Failed

Expected fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenThereIsOneFatalFailure

[0;32m[ RUN ] [mExpectNonfatalFailureTest.FailsWhenStatementReturns

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenStatementReturns

[0;32m[ RUN ] [mExpectNonfatalFailureTest.FailsWhenStatementThrows

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenStatementThrows

[0;32m[----------] [m8 tests from ExpectFatalFailureTest

[0;32m[ RUN ] [mExpectFatalFailureTest.CanReferenceGlobalVariables

[0;32m[ OK ] [mExpectFatalFailureTest.CanReferenceGlobalVariables

[0;32m[ RUN ] [mExpectFatalFailureTest.CanReferenceLocalStaticVariables

[0;32m[ OK ] [mExpectFatalFailureTest.CanReferenceLocalStaticVariables

[0;32m[ RUN ] [mExpectFatalFailureTest.SucceedsWhenThereIsOneFatalFailure

[0;32m[ OK ] [mExpectFatalFailureTest.SucceedsWhenThereIsOneFatalFailure

[0;32m[ RUN ] [mExpectFatalFailureTest.FailsWhenThereIsNoFatalFailure

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenThereIsNoFatalFailure

[0;32m[ RUN ] [mExpectFatalFailureTest.FailsWhenThereAreTwoFatalFailures

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual: 2 failures

googletest-output-test\_.cc:#: Fatal failure:

Failed

Expected fatal failure.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Fatal failure:

Failed

Expected fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenThereAreTwoFatalFailures

[0;32m[ RUN ] [mExpectFatalFailureTest.FailsWhenThereIsOneNonfatalFailure

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual:

googletest-output-test\_.cc:#: Non-fatal failure:

Failed

Expected non-fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenThereIsOneNonfatalFailure

[0;32m[ RUN ] [mExpectFatalFailureTest.FailsWhenStatementReturns

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenStatementReturns

[0;32m[ RUN ] [mExpectFatalFailureTest.FailsWhenStatementThrows

(expecting a failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenStatementThrows

[0;32m[----------] [m2 tests from TypedTest/0, where TypeParam = int

[0;32m[ RUN ] [mTypedTest/0.Success

[0;32m[ OK ] [mTypedTest/0.Success

[0;32m[ RUN ] [mTypedTest/0.Failure

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

TypeParam()

Which is: 0

Expected failure

Stack trace: (omitted)

[0;31m[ FAILED ] [mTypedTest/0.Failure, where TypeParam = int

[0;32m[----------] [m2 tests from TypedTestWithNames/char0, where TypeParam = char

[0;32m[ RUN ] [mTypedTestWithNames/char0.Success

[0;32m[ OK ] [mTypedTestWithNames/char0.Success

[0;32m[ RUN ] [mTypedTestWithNames/char0.Failure

googletest-output-test\_.cc:#: Failure

Failed

Stack trace: (omitted)

[0;31m[ FAILED ] [mTypedTestWithNames/char0.Failure, where TypeParam = char

[0;32m[----------] [m2 tests from TypedTestWithNames/int1, where TypeParam = int

[0;32m[ RUN ] [mTypedTestWithNames/int1.Success

[0;32m[ OK ] [mTypedTestWithNames/int1.Success

[0;32m[ RUN ] [mTypedTestWithNames/int1.Failure

googletest-output-test\_.cc:#: Failure

Failed

Stack trace: (omitted)

[0;31m[ FAILED ] [mTypedTestWithNames/int1.Failure, where TypeParam = int

[0;32m[----------] [m2 tests from Unsigned/TypedTestP/0, where TypeParam = unsigned char

[0;32m[ RUN ] [mUnsigned/TypedTestP/0.Success

[0;32m[ OK ] [mUnsigned/TypedTestP/0.Success

[0;32m[ RUN ] [mUnsigned/TypedTestP/0.Failure

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1U

Which is: 1

TypeParam()

Which is: '\0'

Expected failure

Stack trace: (omitted)

[0;31m[ FAILED ] [mUnsigned/TypedTestP/0.Failure, where TypeParam = unsigned char

[0;32m[----------] [m2 tests from Unsigned/TypedTestP/1, where TypeParam = unsigned int

[0;32m[ RUN ] [mUnsigned/TypedTestP/1.Success

[0;32m[ OK ] [mUnsigned/TypedTestP/1.Success

[0;32m[ RUN ] [mUnsigned/TypedTestP/1.Failure

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1U

Which is: 1

TypeParam()

Which is: 0

Expected failure

Stack trace: (omitted)

[0;31m[ FAILED ] [mUnsigned/TypedTestP/1.Failure, where TypeParam = unsigned int

[0;32m[----------] [m2 tests from UnsignedCustomName/TypedTestP/unsignedChar0, where TypeParam = unsigned char

[0;32m[ RUN ] [mUnsignedCustomName/TypedTestP/unsignedChar0.Success

[0;32m[ OK ] [mUnsignedCustomName/TypedTestP/unsignedChar0.Success

[0;32m[ RUN ] [mUnsignedCustomName/TypedTestP/unsignedChar0.Failure

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1U

Which is: 1

TypeParam()

Which is: '\0'

Expected failure

Stack trace: (omitted)

[0;31m[ FAILED ] [mUnsignedCustomName/TypedTestP/unsignedChar0.Failure, where TypeParam = unsigned char

[0;32m[----------] [m2 tests from UnsignedCustomName/TypedTestP/unsignedInt1, where TypeParam = unsigned int

[0;32m[ RUN ] [mUnsignedCustomName/TypedTestP/unsignedInt1.Success

[0;32m[ OK ] [mUnsignedCustomName/TypedTestP/unsignedInt1.Success

[0;32m[ RUN ] [mUnsignedCustomName/TypedTestP/unsignedInt1.Failure

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1U

Which is: 1

TypeParam()

Which is: 0

Expected failure

Stack trace: (omitted)

[0;31m[ FAILED ] [mUnsignedCustomName/TypedTestP/unsignedInt1.Failure, where TypeParam = unsigned int

[0;32m[----------] [m4 tests from ExpectFailureTest

[0;32m[ RUN ] [mExpectFailureTest.ExpectFatalFailure

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual:

googletest-output-test\_.cc:#: Success:

Succeeded

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual:

googletest-output-test\_.cc:#: Non-fatal failure:

Failed

Expected non-fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 fatal failure containing "Some other fatal failure expected."

Actual:

googletest-output-test\_.cc:#: Fatal failure:

Failed

Expected fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFailureTest.ExpectFatalFailure

[0;32m[ RUN ] [mExpectFailureTest.ExpectNonFatalFailure

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual:

googletest-output-test\_.cc:#: Success:

Succeeded

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual:

googletest-output-test\_.cc:#: Fatal failure:

Failed

Expected fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure containing "Some other non-fatal failure."

Actual:

googletest-output-test\_.cc:#: Non-fatal failure:

Failed

Expected non-fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFailureTest.ExpectNonFatalFailure

[0;32m[ RUN ] [mExpectFailureTest.ExpectFatalFailureOnAllThreads

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual:

googletest-output-test\_.cc:#: Success:

Succeeded

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual:

googletest-output-test\_.cc:#: Non-fatal failure:

Failed

Expected non-fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 fatal failure containing "Some other fatal failure expected."

Actual:

googletest-output-test\_.cc:#: Fatal failure:

Failed

Expected fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFailureTest.ExpectFatalFailureOnAllThreads

[0;32m[ RUN ] [mExpectFailureTest.ExpectNonFatalFailureOnAllThreads

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual:

googletest-output-test\_.cc:#: Success:

Succeeded

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual:

googletest-output-test\_.cc:#: Fatal failure:

Failed

Expected fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

(expecting 1 failure)

gtest.cc:#: Failure

Expected: 1 non-fatal failure containing "Some other non-fatal failure."

Actual:

googletest-output-test\_.cc:#: Non-fatal failure:

Failed

Expected non-fatal failure.

Stack trace: (omitted)

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFailureTest.ExpectNonFatalFailureOnAllThreads

[0;32m[----------] [m2 tests from ExpectFailureWithThreadsTest

[0;32m[ RUN ] [mExpectFailureWithThreadsTest.ExpectFatalFailure

(expecting 2 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected fatal failure.

Stack trace: (omitted)

gtest.cc:#: Failure

Expected: 1 fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFailureWithThreadsTest.ExpectFatalFailure

[0;32m[ RUN ] [mExpectFailureWithThreadsTest.ExpectNonFatalFailure

(expecting 2 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected non-fatal failure.

Stack trace: (omitted)

gtest.cc:#: Failure

Expected: 1 non-fatal failure

Actual: 0 failures

Stack trace: (omitted)

[0;31m[ FAILED ] [mExpectFailureWithThreadsTest.ExpectNonFatalFailure

[0;32m[----------] [m1 test from ScopedFakeTestPartResultReporterTest

[0;32m[ RUN ] [mScopedFakeTestPartResultReporterTest.InterceptOnlyCurrentThread

(expecting 2 failures)

googletest-output-test\_.cc:#: Failure

Failed

Expected fatal failure.

Stack trace: (omitted)

googletest-output-test\_.cc:#: Failure

Failed

Expected non-fatal failure.

Stack trace: (omitted)

[0;31m[ FAILED ] [mScopedFakeTestPartResultReporterTest.InterceptOnlyCurrentThread

[0;32m[----------] [m2 tests from DynamicFixture

DynamicFixture::SetUpTestSuite

[0;32m[ RUN ] [mDynamicFixture.DynamicTestPass

DynamicFixture()

DynamicFixture::SetUp

DynamicFixture::TearDown

~DynamicFixture()

[0;32m[ OK ] [mDynamicFixture.DynamicTestPass

[0;32m[ RUN ] [mDynamicFixture.DynamicTestFail

DynamicFixture()

DynamicFixture::SetUp

googletest-output-test\_.cc:#: Failure

Value of: Pass

Actual: false

Expected: true

Stack trace: (omitted)

DynamicFixture::TearDown

~DynamicFixture()

[0;31m[ FAILED ] [mDynamicFixture.DynamicTestFail

DynamicFixture::TearDownTestSuite

[0;32m[----------] [m1 test from DynamicFixtureAnotherName

DynamicFixture::SetUpTestSuite

[0;32m[ RUN ] [mDynamicFixtureAnotherName.DynamicTestPass

DynamicFixture()

DynamicFixture::SetUp

DynamicFixture::TearDown

~DynamicFixture()

[0;32m[ OK ] [mDynamicFixtureAnotherName.DynamicTestPass

DynamicFixture::TearDownTestSuite

[0;32m[----------] [m2 tests from BadDynamicFixture1

DynamicFixture::SetUpTestSuite

[0;32m[ RUN ] [mBadDynamicFixture1.FixtureBase

DynamicFixture()

DynamicFixture::SetUp

DynamicFixture::TearDown

~DynamicFixture()

[0;32m[ OK ] [mBadDynamicFixture1.FixtureBase

[0;32m[ RUN ] [mBadDynamicFixture1.TestBase

DynamicFixture()

gtest.cc:#: Failure

Failed

All tests in the same test suite must use the same test fixture

class, so mixing TEST\_F and TEST in the same test suite is

illegal. In test suite BadDynamicFixture1,

test FixtureBase is defined using TEST\_F but

test TestBase is defined using TEST. You probably

want to change the TEST to TEST\_F or move it to another test

case.

Stack trace: (omitted)

~DynamicFixture()

[0;31m[ FAILED ] [mBadDynamicFixture1.TestBase

DynamicFixture::TearDownTestSuite

[0;32m[----------] [m2 tests from BadDynamicFixture2

DynamicFixture::SetUpTestSuite

[0;32m[ RUN ] [mBadDynamicFixture2.FixtureBase

DynamicFixture()

DynamicFixture::SetUp

DynamicFixture::TearDown

~DynamicFixture()

[0;32m[ OK ] [mBadDynamicFixture2.FixtureBase

[0;32m[ RUN ] [mBadDynamicFixture2.Derived

DynamicFixture()

gtest.cc:#: Failure

Failed

All tests in the same test suite must use the same test fixture

class. However, in test suite BadDynamicFixture2,

you defined test FixtureBase and test Derived

using two different test fixture classes. This can happen if

the two classes are from different namespaces or translation

units and have the same name. You should probably rename one

of the classes to put the tests into different test suites.

Stack trace: (omitted)

~DynamicFixture()

[0;31m[ FAILED ] [mBadDynamicFixture2.Derived

DynamicFixture::TearDownTestSuite

[0;32m[----------] [m1 test from PrintingFailingParams/FailingParamTest

[0;32m[ RUN ] [mPrintingFailingParams/FailingParamTest.Fails/0

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

GetParam()

Which is: 2

Stack trace: (omitted)

[0;31m[ FAILED ] [mPrintingFailingParams/FailingParamTest.Fails/0, where GetParam() = 2

[0;32m[----------] [m1 test from EmptyBasenameParamInst

[0;32m[ RUN ] [mEmptyBasenameParamInst.Passes/0

[0;32m[ OK ] [mEmptyBasenameParamInst.Passes/0

[0;32m[----------] [m2 tests from PrintingStrings/ParamTest

[0;32m[ RUN ] [mPrintingStrings/ParamTest.Success/a

[0;32m[ OK ] [mPrintingStrings/ParamTest.Success/a

[0;32m[ RUN ] [mPrintingStrings/ParamTest.Failure/a

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

"b"

GetParam()

Which is: "a"

Expected failure

Stack trace: (omitted)

[0;31m[ FAILED ] [mPrintingStrings/ParamTest.Failure/a, where GetParam() = "a"

[0;32m[----------] [m3 tests from GoogleTestVerification

[0;32m[ RUN ] [mGoogleTestVerification.UninstantiatedParamaterizedTestSuite<NoTests>

Paramaterized test suite NoTests is instantiated via INSTANTIATE\_TEST\_SUITE\_P, but no tests are defined via TEST\_P . No test cases will run.

Ideally, INSTANTIATE\_TEST\_SUITE\_P should only ever be invoked from code that always depend on code that provides TEST\_P. Failing to do so is often an indication of dead code, e.g. the last TEST\_P was removed but the rest got left behind.

To suppress this error for this test suite, insert the following line (in a non-header) in the namespace it is defined in:

GTEST\_ALLOW\_UNINSTANTIATED\_PARAMETERIZED\_TEST(NoTests);

[0;32m[ OK ] [mGoogleTestVerification.UninstantiatedParamaterizedTestSuite<NoTests>

[0;32m[ RUN ] [mGoogleTestVerification.UninstantiatedParamaterizedTestSuite<DetectNotInstantiatedTest>

Paramaterized test suite DetectNotInstantiatedTest is defined via TEST\_P, but never instantiated. None of the test cases will run. Either no INSTANTIATE\_TEST\_SUITE\_P is provided or the only ones provided expand to nothing.

Ideally, TEST\_P definitions should only ever be included as part of binaries that intend to use them. (As opposed to, for example, being placed in a library that may be linked in to get other utilities.)

To suppress this error for this test suite, insert the following line (in a non-header) in the namespace it is defined in:

GTEST\_ALLOW\_UNINSTANTIATED\_PARAMETERIZED\_TEST(DetectNotInstantiatedTest);

[0;32m[ OK ] [mGoogleTestVerification.UninstantiatedParamaterizedTestSuite<DetectNotInstantiatedTest>

[0;32m[ RUN ] [mGoogleTestVerification.UninstantiatedTypeParamaterizedTestSuite<DetectNotInstantiatedTypesTest>

Type paramaterized test suite DetectNotInstantiatedTypesTest is defined via REGISTER\_TYPED\_TEST\_SUITE\_P, but never instantiated via INSTANTIATE\_TYPED\_TEST\_SUITE\_P. None of the test cases will run.

Ideally, TYPED\_TEST\_P definitions should only ever be included as part of binaries that intend to use them. (As opposed to, for example, being placed in a library that may be linked in to get other utilities.)

To suppress this error for this test suite, insert the following line (in a non-header) in the namespace it is definedin in:

GTEST\_ALLOW\_UNINSTANTIATED\_PARAMETERIZED\_TEST(DetectNotInstantiatedTypesTest);

[0;32m[ OK ] [mGoogleTestVerification.UninstantiatedTypeParamaterizedTestSuite<DetectNotInstantiatedTypesTest>

[0;32m[----------] [mGlobal test environment tear-down

BarEnvironment::TearDown() called.

googletest-output-test\_.cc:#: Failure

Failed

Expected non-fatal failure.

Stack trace: (omitted)

FooEnvironment::TearDown() called.

googletest-output-test\_.cc:#: Failure

Failed

Expected fatal failure.

Stack trace: (omitted)

[0;32m[==========] [m88 tests from 41 test suites ran.

[0;32m[ PASSED ] [m34 tests.

[0;31m[ FAILED ] [m54 tests, listed below:

[0;31m[ FAILED ] [mNonfatalFailureTest.EscapesStringOperands

[0;31m[ FAILED ] [mNonfatalFailureTest.DiffForLongStrings

[0;31m[ FAILED ] [mFatalFailureTest.FatalFailureInSubroutine

[0;31m[ FAILED ] [mFatalFailureTest.FatalFailureInNestedSubroutine

[0;31m[ FAILED ] [mFatalFailureTest.NonfatalFailureInSubroutine

[0;31m[ FAILED ] [mLoggingTest.InterleavingLoggingAndAssertions

[0;31m[ FAILED ] [mSCOPED\_TRACETest.AcceptedValues

[0;31m[ FAILED ] [mSCOPED\_TRACETest.ObeysScopes

[0;31m[ FAILED ] [mSCOPED\_TRACETest.WorksInLoop

[0;31m[ FAILED ] [mSCOPED\_TRACETest.WorksInSubroutine

[0;31m[ FAILED ] [mSCOPED\_TRACETest.CanBeNested

[0;31m[ FAILED ] [mSCOPED\_TRACETest.CanBeRepeated

[0;31m[ FAILED ] [mSCOPED\_TRACETest.WorksConcurrently

[0;31m[ FAILED ] [mScopedTraceTest.WithExplicitFileAndLine

[0;31m[ FAILED ] [mNonFatalFailureInFixtureConstructorTest.FailureInConstructor

[0;31m[ FAILED ] [mFatalFailureInFixtureConstructorTest.FailureInConstructor

[0;31m[ FAILED ] [mNonFatalFailureInSetUpTest.FailureInSetUp

[0;31m[ FAILED ] [mFatalFailureInSetUpTest.FailureInSetUp

[0;31m[ FAILED ] [mAddFailureAtTest.MessageContainsSpecifiedFileAndLineNumber

[0;31m[ FAILED ] [mGtestFailAtTest.MessageContainsSpecifiedFileAndLineNumber

[0;31m[ FAILED ] [mMixedUpTestSuiteTest.ThisShouldFail

[0;31m[ FAILED ] [mMixedUpTestSuiteTest.ThisShouldFailToo

[0;31m[ FAILED ] [mMixedUpTestSuiteWithSameTestNameTest.TheSecondTestWithThisNameShouldFail

[0;31m[ FAILED ] [mTEST\_F\_before\_TEST\_in\_same\_test\_case.DefinedUsingTESTAndShouldFail

[0;31m[ FAILED ] [mTEST\_before\_TEST\_F\_in\_same\_test\_case.DefinedUsingTEST\_FAndShouldFail

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenThereIsNoNonfatalFailure

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenThereAreTwoNonfatalFailures

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenThereIsOneFatalFailure

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenStatementReturns

[0;31m[ FAILED ] [mExpectNonfatalFailureTest.FailsWhenStatementThrows

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenThereIsNoFatalFailure

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenThereAreTwoFatalFailures

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenThereIsOneNonfatalFailure

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenStatementReturns

[0;31m[ FAILED ] [mExpectFatalFailureTest.FailsWhenStatementThrows

[0;31m[ FAILED ] [mTypedTest/0.Failure, where TypeParam = int

[0;31m[ FAILED ] [mTypedTestWithNames/char0.Failure, where TypeParam = char

[0;31m[ FAILED ] [mTypedTestWithNames/int1.Failure, where TypeParam = int

[0;31m[ FAILED ] [mUnsigned/TypedTestP/0.Failure, where TypeParam = unsigned char

[0;31m[ FAILED ] [mUnsigned/TypedTestP/1.Failure, where TypeParam = unsigned int

[0;31m[ FAILED ] [mUnsignedCustomName/TypedTestP/unsignedChar0.Failure, where TypeParam = unsigned char

[0;31m[ FAILED ] [mUnsignedCustomName/TypedTestP/unsignedInt1.Failure, where TypeParam = unsigned int

[0;31m[ FAILED ] [mExpectFailureTest.ExpectFatalFailure

[0;31m[ FAILED ] [mExpectFailureTest.ExpectNonFatalFailure

[0;31m[ FAILED ] [mExpectFailureTest.ExpectFatalFailureOnAllThreads

[0;31m[ FAILED ] [mExpectFailureTest.ExpectNonFatalFailureOnAllThreads

[0;31m[ FAILED ] [mExpectFailureWithThreadsTest.ExpectFatalFailure

[0;31m[ FAILED ] [mExpectFailureWithThreadsTest.ExpectNonFatalFailure

[0;31m[ FAILED ] [mScopedFakeTestPartResultReporterTest.InterceptOnlyCurrentThread

[0;31m[ FAILED ] [mDynamicFixture.DynamicTestFail

[0;31m[ FAILED ] [mBadDynamicFixture1.TestBase

[0;31m[ FAILED ] [mBadDynamicFixture2.Derived

[0;31m[ FAILED ] [mPrintingFailingParams/FailingParamTest.Fails/0, where GetParam() = 2

[0;31m[ FAILED ] [mPrintingStrings/ParamTest.Failure/a, where GetParam() = "a"

54 FAILED TESTS

[0;33m YOU HAVE 1 DISABLED TEST

[mNote: Google Test filter = FatalFailureTest.\*:LoggingTest.\*

[==========] Running 4 tests from 2 test suites.

[----------] Global test environment set-up.

[----------] 3 tests from FatalFailureTest

[ RUN ] FatalFailureTest.FatalFailureInSubroutine

(expecting a failure that x should be 1)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

x

Which is: 2

Stack trace: (omitted)

[ FAILED ] FatalFailureTest.FatalFailureInSubroutine (? ms)

[ RUN ] FatalFailureTest.FatalFailureInNestedSubroutine

(expecting a failure that x should be 1)

googletest-output-test\_.cc:#: Failure

Expected equality of these values:

1

x

Which is: 2

Stack trace: (omitted)

[ FAILED ] FatalFailureTest.FatalFailureInNestedSubroutine (? ms)

[ RUN ] FatalFailureTest.NonfatalFailureInSubroutine

(expecting a failure on false)

googletest-output-test\_.cc:#: Failure

Value of: false

Actual: false

Expected: true

Stack trace: (omitted)

[ FAILED ] FatalFailureTest.NonfatalFailureInSubroutine (? ms)

[----------] 3 tests from FatalFailureTest (? ms total)

[----------] 1 test from LoggingTest

[ RUN ] LoggingTest.InterleavingLoggingAndAssertions

(expecting 2 failures on (3) >= (a[i]))

i == 0

i == 1

googletest-output-test\_.cc:#: Failure

Expected: (3) >= (a[i]), actual: 3 vs 9

Stack trace: (omitted)

i == 2

i == 3

googletest-output-test\_.cc:#: Failure

Expected: (3) >= (a[i]), actual: 3 vs 6

Stack trace: (omitted)

[ FAILED ] LoggingTest.InterleavingLoggingAndAssertions (? ms)

[----------] 1 test from LoggingTest (? ms total)

[----------] Global test environment tear-down

[==========] 4 tests from 2 test suites ran. (? ms total)

[ PASSED ] 0 tests.

[ FAILED ] 4 tests, listed below:

[ FAILED ] FatalFailureTest.FatalFailureInSubroutine

[ FAILED ] FatalFailureTest.FatalFailureInNestedSubroutine

[ FAILED ] FatalFailureTest.NonfatalFailureInSubroutine

[ FAILED ] LoggingTest.InterleavingLoggingAndAssertions

4 FAILED TESTS

Note: Google Test filter = \*DISABLED\_\*

[==========] Running 1 test from 1 test suite.

[----------] Global test environment set-up.

[----------] 1 test from DisabledTestsWarningTest

[ RUN ] DisabledTestsWarningTest.DISABLED\_AlsoRunDisabledTestsFlagSuppressesWarning

[ OK ] DisabledTestsWarningTest.DISABLED\_AlsoRunDisabledTestsFlagSuppressesWarning

[----------] Global test environment tear-down

[==========] 1 test from 1 test suite ran.

[ PASSED ] 1 test.

Note: Google Test filter = PassingTest.\*

Note: This is test shard 2 of 2.

[==========] Running 1 test from 1 test suite.

[----------] Global test environment set-up.

[----------] 1 test from PassingTest

[ RUN ] PassingTest.PassingTest2

[ OK ] PassingTest.PassingTest2

[----------] Global test environment tear-down

[==========] 1 test from 1 test suite ran.

[ PASSED ] 1 test.