



## lib\_unity: Unity library

Publication Date: 2026/5/22

Document Number: XM-015644-UG v2.7.0

## IN THIS DOCUMENT

1	Overview . . . . .	2
2	Compatibility . . . . .	2
3	How to use . . . . .	2
	3.1 Features . . . . .	2
4	Example . . . . .	2
5	How to build the example . . . . .	3
6	How to run the example . . . . .	3
7	Expected output . . . . .	3

 **Warning**

This documentation is a work in progress.

## 1 Overview

**lib\_unity** is a small utility library that adapts the [Unity unit test framework](#) for use with the **xcommon\_cmake** based XMOS build system. It provides the necessary CMake integration to compile and link Unity into your XCore application, along with helper fixtures for structuring test groups and a test runner.

## 2 Compatibility

- ▶ **Toolchain:** XMOS XTC Tools 15.3.1 or later
- ▶ **Build system:** xcommon\_cmake

## 3 How to use

Add **lib\_unity** to your application's `APP_DEPENDENT_MODULES` list in `CMakeLists.txt`:

```
set(APP_DEPENDENT_MODULES "lib_unity")
```

Pin the dependency to a release tag where possible. If no version is specified, `xcommon_cmake` will use the latest commit on the **develop** branch.

See the [xcommon\\_cmake documentation](#) for dependency management details.

### 3.1 Features

After adding the module, your application gains:

- ▶ The Unity and Unity Fixture headers (**unity.h**, **unity\_fixture.h**)
- ▶ A CMake target that compiles and links the Unity source files automatically
- ▶ No additional configuration is required; Unity's default settings are used

## 4 Example

The repository includes a minimal working example in **examples/basic** that demonstrates a test group and a test runner using the Unity Fixture API.



Test file (`examples/basic/src/test_basic.c`):

```
#include "unity_fixture.h"
#include <math.h>

TEST_GROUP_RUNNER(test_basic) {
    RUN_TEST_CASE(test_basic, sum_test);
    RUN_TEST_CASE(test_basic, div_test);
    RUN_TEST_CASE(test_basic, sinf_test);
}

TEST_GROUP(test_basic);
TEST_SETUP(test_basic) { /*Optional setup*/}
TEST_TEAR_DOWN(test_basic) { /*Optional cleanup*/}

TEST(test_basic, sum_test) {
    int result = 2 + 3;
    TEST_ASSERT_EQUAL_INT(5, result);
}

TEST(test_basic, div_test) {
    int result = 20 / 4;
    TEST_ASSERT_EQUAL_INT(5, result);
}

TEST(test_basic, sinf_test) {
    const float pi = 3.14159265f;
    float angle = pi / 2.0f;
    float result = sinf(angle);
    TEST_ASSERT_FLOAT_WITHIN(0.0001f, 1.0f, result);
}
```

Test runner (`examples/basic/src/main.c`):

```
int main(int argc, const char* argv[])
{
    UnityGetCommandLineOptions(argc, argv);
    UnityBegin(argv[0]);

    printf("\n\n");

    RUN_TEST_GROUP(test_basic);
    return UNITY_END();
}
```

## 5 How to build the example

To build the example, run the following commands from the root of the repository:

```
cd examples/basic
cmake -B build -G "Unix Makefiles"
xmake -C build
```

## 6 How to run the example

Once compiled, to run the example on the simulator, use:

```
xsim bin/test_basic.xe
```

Or on hardware:

```
xrun bin/test_basic.xe
```

## 7 Expected output

A passing test run produces:

```
3 Tests 0 Failures 0 Ignored
OK
```





Copyright © 2026, All Rights Reserved.

---

Xmos Ltd. is the owner or licensee of this design, code, or Information (collectively, the "Information") and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. Xmos Ltd makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

Xmos, Xcore, VocalFusion and the Xmos logo are registered trademarks of Xmos Ltd. in the United Kingdom and other countries and may not be used without written permission. Company and product names mentioned in this document are the trademarks or registered trademarks of their respective owners.

