Code Samples that Actually Compile

Clare Macrae

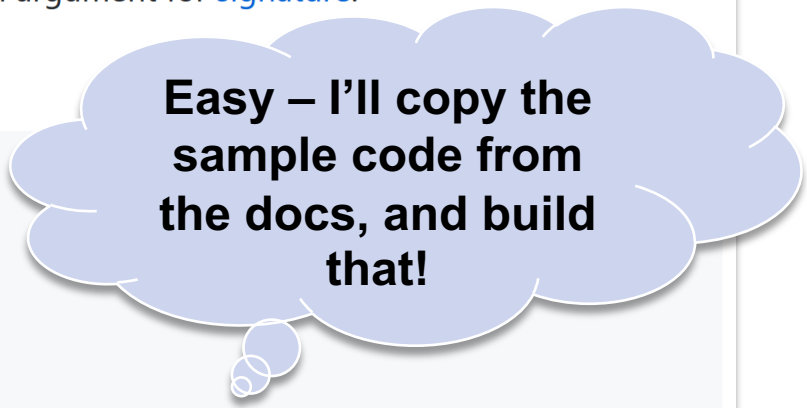
She/Her

Recently…

- I needed to quickly try these Catch2 macros:
  - **`TEMPLATE_TEST_CASE_METHOD_SIG`**
  - **`TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG`**
- The docs had code samples.

Catch2 also provides `TEMPLATE_TEST_CASE_METHOD_SIG` and `TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG` to support fixtures using non-type template parameters. These test cases work similar to `TEMPLATE_TEST_CASE_METHOD` and `TEMPLATE_PRODUCT_TEST_CASE_METHOD`, with additional positional argument for signature.

Example:




```
template <int V>
struct Nttp_Fixture{
    int value = V;
};

TEMPLATE_TEST_CASE_METHOD_SIG(Nttp_Fixture, "A TEMPLATE_TEST_CASE_METHOD_SIG based test run that
    REQUIRE(Nttp_Fixture<V>::value > 0);
}

template< typename T, size_t V>
struct Template_Foo_2 {
    size_t size() { return V; }
};

TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(Template_Fixture_2, "A TEMPLATE_PRODUCT_TEST_CASE_METHOD_SI
{
    REQUIRE(Template_Fixture_2<TestType>{}.m_a.size() >= 2);
}
```

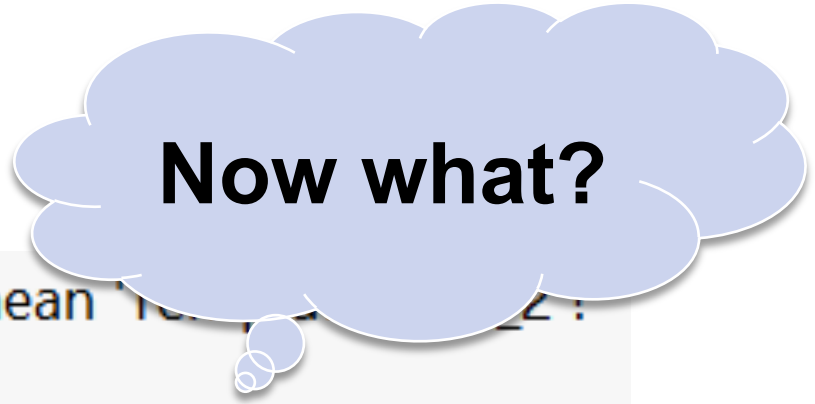
```cpp
#include "catch.h"

// https://github.com/catchorg/Catch2/blob/master/docs/test-fixtures.md

template <int V>
struct Nttp_Fixture{
    int value = V;
};

TEMPLATE_TEST_CASE_METHOD_SIG(Nttp_Fixture, "A TEMPLATE_TEST_CASE_METHOD_SIG based test run that
 succeeds", "[class][template][nttp]",((int V), V), 1, 3, 6) {
    REQUIRE(Nttp_Fixture<V>::value > 0);
}

template< typename T, size_t V>
struct Template_Foo_2 {
    size_t size() { return V; }
};

TEMPLATE_PRODUCT_TEST_CASE_METHOD_SI(Template_Fixture_2, "A TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG based
 test run that succeeds", "[class][template][product][nttp]", ((typename T, size_t S), T, S),(std::array,
  Template_Foo_2), ((int,2), (float,6)))
{
    REQUIR(Template_Fixture_2<TestType>{}.m_a.size() >= 2);
}
```

No template named 'Template_Fixture_2'; did you mean 'Template_Foo_2'?
'Template_Foo_2' declared here

No template named 'Template_Fixture_2'; did you mean 'Template_Foo_2'?
'Template_Foo_2' declared here

Too few template arguments for class template 'Template_Foo_2'
template is declared here

Too few template arguments for class template 'Template_Foo_2'
template is declared here



No template named 'Template_Fixture_2'; did you mean '[illegible]'?
'Template_Foo_2' declared here

No template named 'Template_Fixture_2'; did you mean 'Template_Foo_2'?
'Template_Foo_2' declared here

Too few template arguments for class template 'Template_Foo_2'
template is declared here

Too few template arguments for class template 'Template_Foo_2'
template is declared here

Sounds Familiar?

Now what?

Search for missing symbol in docs – No...

Search the whole repo for similar code

Found it – so, how does it differ from the code in the docs?

Paste in the relevant missing bit…

Test it – it works!

Fix the docs and submit a pull request – of course

A Better Way…

Class.tests.cpp

```
94  // begin-snippet: template_test_case_and_product_methods
95  TEMPLATE_TEST_CASE_METHOD_SIG(Nttp_Fixture, "A TEMPLATE_TEST_CASE_METHOD_SIG based test run that succeeds", "[cla
96      REQUIRE(Nttp_Fixture<V>::value > 0);
97  }
98
99  TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(Template_Fixture_2, "A TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG based test rur
100 {
101     REQUIRE(Template_Fixture_2<TestType>{}.m_a.size() >= 2);
102 }
103 // end-snippet
```

Class.tests.cpp

```
55  // begin-snippet: class_test_helpers
56  template <int V>
57  struct Nttp_Fixture{
58      int value = V;
59  };
60
61  template<typename T>
62  struct Template_Fixture_2 {
63      Template_Fixture_2() {}
64
65      T m_a;
66  };
67
68  template< typename T, size_t V>
69  struct Template_Foo_2 {
70      size_t size() { return V; }
71  };
72  // end-snippet
73  #endif
```

test-fixtures.source.md

```
89  ## Signature-based parametrised test fixtures
90
91  > [Introduced](https://github.com/catchorg/Catch2/issues/1609) in Catch 2.8.0.
92
93  Catch2 also provides `TEMPLATE_TEST_CASE_METHOD_SIG` and `TEMPLATE_PRODUCT_TEST_CA
94  fixtures using non-type template parameters. These test cases work similar to `TEM
95  with additional positional argument for [signature](test-cases-and-sections.md#sig
96
97  Example:
98
99  snippet: class_test_helpers
100
101 snippet: template_test_case_and_product_methods
102
```

Console window

```
...>mdsnippets
Config:
...
Added 1 .source.md files
Searching 473 files for snippets
Added 2 snippets
Added 0 files for snippets
Added 0 snippets
Processing C:\Users\Clare\Documents\Programming\GitHub\Catch2-claremacrae
 \docs\test-fixtures.source.md
```

On github: test-fixtures.md

Signature-based parametrised test fixtures

> Introduced in Catch 2.8.0.

Catch2 also provides `TEMPLATE_TEST_CASE_METHOD_SIG` and `TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG` to support fixtures using non-type template parameters. These test cases work similar to `TEMPLATE_TEST_CASE_METHOD` and `TEMPLATE_PRODUCT_TEST_CASE_METHOD`, with additional positional argument for signature.

Example:

```
template <int V>
struct Nttp_Fixture{
    int value = V;
};

template<typename T>
struct Template_Fixture_2 {
    Template_Fixture_2() {}

    T m_a;
};

template< typename T, size_t V>
struct Template_Foo_2 {
    size_t size() { return V; }
};
```

snippet source / anchor

```
TEMPLATE_TEST_CASE_METHOD_SIG(Nttp_Fixture, "A TEMPLATE_TEST_CASE_METHOD_SIG based test run that
    REQUIRE(Nttp_Fixture<V>::value > 0);
}

TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(Template_Fixture_2, "A TEMPLATE_PRODUCT_TEST_CASE_METHOD_SI
{
    REQUIRE(Template_Fixture_2<TestType>{}.m_a.size() >= 2);
}
```

snippet source / anchor

On github: test-fixtures.md

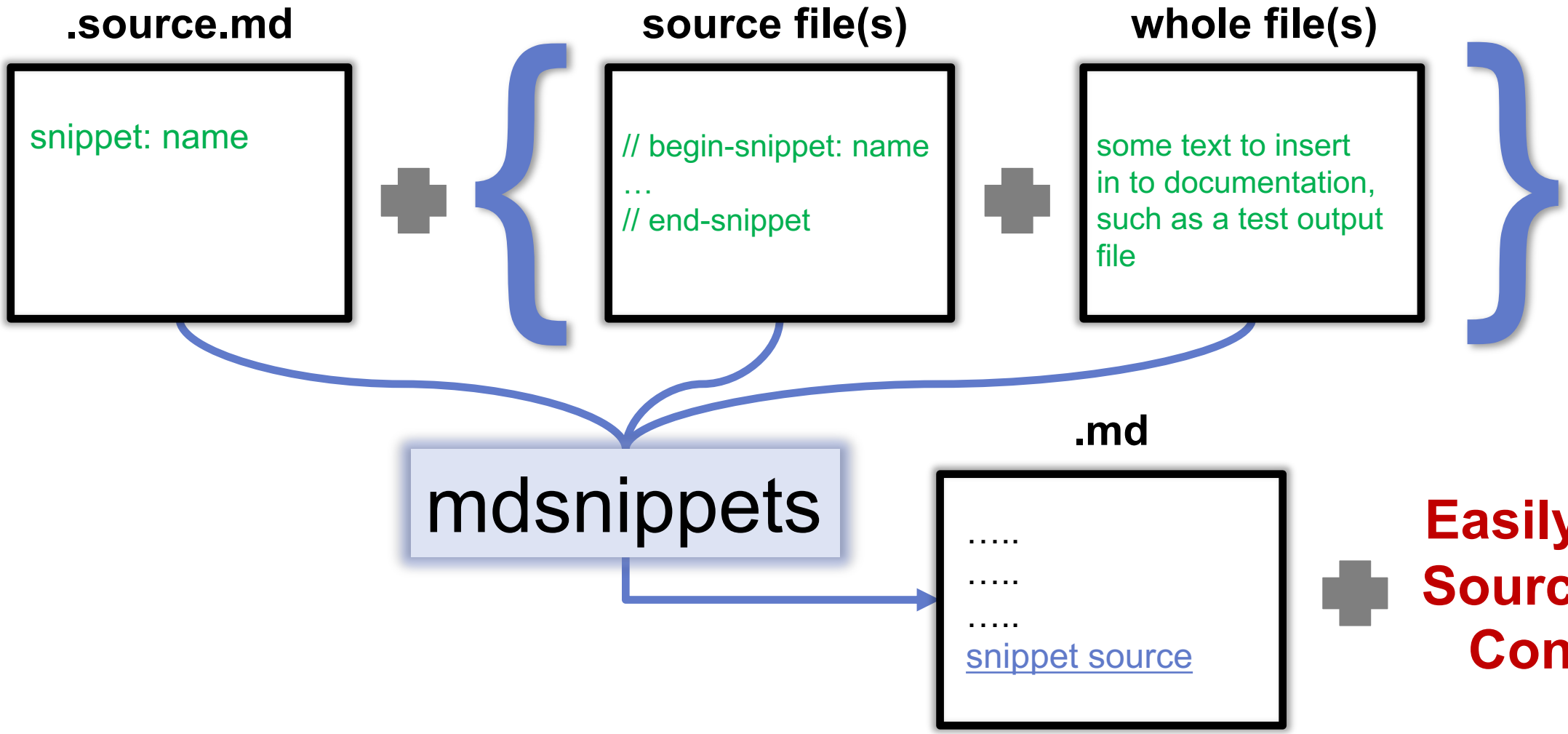
```
TEMPLATE_TEST_CASE_METHOD_SIG(Nttp_Fixture, "A TEMPLATE_TEST_CASE_METHOD_SIG based test run that
    REQUIRE(Nttp_Fixture<V>::value > 0);
}

TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(Template_Fixture_2, "A TEMPLATE_PRODUCT_TEST_CASE_METHOD_SI
{
    REQUIRE(Template_Fixture_2<TestType>{}.m_a.size() >= 2);
}
```

snippet source | anchor

https://github.../Class.tests.cpp#L94-L103

```
89  TEMPLATE_PRODUCT_TEST_CASE_METHOD(Template_Fixture_2, "A TEMPLATE_PRODUCT_TEST_CASE_METHOD based test ru
90  {
91      REQUIRE( Template_Fixture_2<TestType>::m_a.size() == 0 );
92  }
93
94  // begin-snippet: template_test_case_and_product_methods
95  TEMPLATE_TEST_CASE_METHOD_SIG(Nttp_Fixture, "A TEMPLATE_TEST_CASE_METHOD_SIG based test run that succeed
96      REQUIRE(Nttp_Fixture<V>::value > 0);
97  }
98
99  TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(Template_Fixture_2, "A TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG based
100 {
101     REQUIRE(Template_Fixture_2<TestType>{}.m_a.size() >= 2);
102 }
103 // end-snippet
104
105 using MyTypes = std::tuple<int, char, double>;
106 TEMPLATE_LIST_TEST_CASE_METHOD(Template_Fixture, "Template test case method with test types specified in
107 {
108     REQUIRE( Template_Fixture<TestType>::m_a == 1 );
109 }
110
```

What's going on here?



Tips for your Markdown

- Insert table of contents
  - `toc`
- Use include to pull in files without the [source] link:
  - `include: include_using_test_frameworks_list.include.md`
  - `include: https://raw.githubusercontent.com/.../inc_conan.include.md`

Tips for Running

- `mdsnippets -readonly true`

Guaranteed-working code samples

… and a way to see their context.

mdsnippets.com

Thanks to @SimonCropp for this excellent tool!