

Handling Enum Values

Variables and constants of enum types can be handled with a dedicated set of API macros. The macros can report enum values using enumeration identifiers, rather than plain integral numbers. The following API macros can be used to handle enum values:

- `CPPTTEST_REPORT_ENUM(<scoped enum name>, <txt>, <value>)`
- `CPPTTEST_POST_CONDITION_ENUM(<scoped enum name>, <value string>, <value>)`
- `CPPTTEST_ASSERT_ENUM_EQUAL(<scoped enum name>, <expected>, <actual>)`
- `CPPTTEST_ASSERT_ENUM_EQUAL_MESSAGE(<scoped enum name>, <msg>, <expected>, <actual>)`

where:

- `<scoped enum name>` is full name of enumeration, including all namespace names (for example: `INNER_NS::MyEnumeration`, `INNER_NS::MyClass::MyEnumeration`).
- `<value string>` is a string that describes the reported value.
- `<value>` is an integral value or enumerator.
- `<txt>` is a string that describes the reported value.
- `<expected>` is an integral value or enumerator.



The API macros presented above require the "Enable enum data autogeneration" test configuration option to be enabled. Go to **Test Configuration > Execution tab > General tab > Instrumentation mode > Edit > Instrumentation features > Advanced options** to configure this option.

Using enum values in Parameters Step and Data Sources

You can specify values inside the Test Case Editor's Parameters table using identifiers of enumerators. Press CTRL+SPACE for a list of known enumerators.

By default, only enumerators defined in the project are shown when CTRL+SPACE is pressed. To show enumerators defined outside of the project, open the project properties, go to **Parasoft > C++test > Other Settings** and provide the advanced option(s): `xharness.dumpEnumsFromFile <PATH>`, where `<PATH>` is a disk location of sources that define enumerations. Wildcard `*` can be used while specifying the path. Multiple entries of the `xharness.dumpEnumsFromFile <PATH>` option are allowed.

The following example shows the Parameters step with enumeration identifiers:

| | A | B | C | D | E | F | G |
|-------------|-------|--------|---|---|---|---|---|
| Name | Input | Output | | | | | |
| 1 | MON | TUE | | | | | |
| 2 | TUE | WED | | | | | |
| 3 | WED | THU | | | | | |
| 4 | | | | | | | |

You also can use enumeration identifiers in data sources. The following example shows an extract from a CSV file:

```
"Input", "Output"  
MON, TUE  
TUE, WED  
WED, THU
```



A simple enumerator name or a full name that includes all name spaces can be used. For example, if the `MON` enumerator is defined in `::INNER_NS::MyClass::DaysEnumeration`, you can type either `MON` or `::INNER_NS::MyClass::MON`.

Extracting Enum Value from Data Source or Parameters Step

Use the following helper macro to to extract an enum value from a data source or the Parameters step:

```
CPPTTEST_DS_GET_ENUM(<scoped enum name>, <column name>)
```

The following example shows a simple example of extracting the enum value from a data source:

▼ Variables



| | Type | Name | Value |
|---|-----------|------|---|
| 1 | enum DAYS | _day | CPPTEST_DS_GET_ENUM(enum DAYS, "Input") |

