

## fhicl-cpp - Feature #22574

### FHiCL C++ interface for multiple choice values

05/09/2019 06:09 PM - Gianluca Petrillo

<b>Status:</b>	Under Discussion	<b>Start date:</b>	05/09/2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour

#### Description

I would like to request support in FHiCL C++ interface for multiple choice values.  
I am going with an...

#### Example

Here unit is a configuration parameter of integral type (int) that may represent three choices: GeV (1), MeV (2) and megajoule (3).

#### Configuration data structure

```
struct Config {  
  
    using Name      = fhicl::Name;  
    using Comment   = fhicl::Comment;  
    using Choices   = fhicl::Choices;  
  
    fhicl::Atom<double> energy {  
        Name      { "energy" },  
        Comment { "energy (unit is specified by `unit` parameter)" }  
    }; // energy  
  
    fhicl::Choice<int> unit {  
        Name      { "unit" },  
        Comment { "energy unit" },  
        Choices {  
            {  
                Name { "GeV" }, // the name of the choice  
                Comment{ "energies are in GeV" }, // and its description  
                1 // the value of `unit` if this choice is picked  
            },  
            {  
                Name { "MeV" }, // the name of the choice  
                Comment{ "energies are in MeV" }, // and its description  
                2 // the value of `unit` if this choice is picked  
            },  
            {  
                Name { "MJ" }, // the name of the choice  
                Comment{ "energies are in megajoule" }, // and its description  
                3 // the value of `unit` if this choice is picked  
            },  
        },  
        1 // default: GeV  
    }; // unit  
  
}; // Config
```

If the default value is omitted, the parameter is mandatory. Otherwise, its value must match a supported choice, or else a `logic_error` type of exception will be thrown at run time.

#### Configuration FHiCL table

```
energy: 10
```

```
unit: "MeV"
```

## Description on console screen

```
## energy (unit is specified by `unit` parameter)
```

```
energy: <double>
```

```
## energy unit
```

```
## Choose one among:
```

```
## "GeV" : energies are in GeV
```

```
## "MeV" : energies are in MeV
```

```
## "MJ" : energies are in megajoule
```

```
unit: "GeV" # default
```

## User code to access the values

```
double energy = config().energy();
int unit = config().unit();
switch (unit) {
    case 1: break;
    case 2: energy *= 1e-3; break;
    case 3: energy *= JouleToGeV * 1e-6; break;
} // switch
```

Throws a `art::error::Configuration` type of `art::Exception` (or equivalent) if the value is not acceptable.

## Bonus points

Additional candies:

- works with enum and/or enum class types
- can be used in a sequence (`fhicl::ChoiceSequence`)
- choice value converted from name by default (would not work in the example, but if the type is `std::string` instead of `int` it will save replicating the Name into the value on each entry)
- different choices may have the same value (e.g. "MJ" and "megajoule" both supported and associated to the same value 3; if 3 is the default value, it will be written in the description as the first of the matching choices)

## History

#1 - 05/13/2019 10:26 AM - Kyle Knoepfel

- Status changed from New to Under Discussion

This feature requires a design discussion. Although it seems to be a valuable feature to add, it is not clear to what extent the [fhicl-cpp](#) package can easily support this. We will get back to you.