fhicl-cpp - Feature #23751

fhicl-getpar executable

12/11/2019 12:13 PM - Andrei Gaponenko

Status:	Closed	**Start date:**	12/11/2019
Priority:	Normal	**Due date:**	
Assignee:	Kyle Knoepfel	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	3.06.00	**Spent time:**	24.00 hours

Description

Hello,

Mu2e would like to add one more executable to the fhiclcpp distribution, fhicl-getpar.
The suggested implementation is fully contained in this file
https://cdcvs.fnal.gov/redmine/projects/mu2e-tools/repository/bintools/revisions/master/entry/mu2ebintools/fhicl-getpar.cc

(or "git clone http://cdcvs.fnal.gov//projects/mu2e-tools-bintools/mu2ebintools")

We have been using the fhicl-getpar since 2015 in grid scripts; the commit history shows that the interface of the program is stable.

Andrei

Related issues:

Related to art - Feature #24526: Request that the art team take on support fo...	**Closed**	**06/09/2020**

Associated revisions

Revision 2c927d98 - 06/16/2020 10:24 AM - Kyle Knoepfel

Implement feature #23751: Provide fhicl-get facility.

Issue #23751 requests a command-line facility that mimics the
fhicl::ParameterSet::get<T>(key) C++ interface. The implementation is
based off of Mu2e's fhicl-getpar.

History

#1 - 12/16/2019 10:26 AM - Kyle Knoepfel

- Status changed from New to Accepted

#2 - 06/09/2020 07:56 AM - Kyle Knoepfel

- Target version set to 3.06.00

#3 - 06/09/2020 02:06 PM - Kyle Knoepfel

- Assignee set to Kyle Knoepfel

- Status changed from Accepted to Assigned

#4 - 06/10/2020 08:27 AM - Kyle Knoepfel

- Status changed from Assigned to Feedback

A few questions:

- Would you be opposed to renaming the facility 'fhicl-get-value'?
- The code supports printing only atomic values and sequence-of-atomic values. Is there any reason why the fhiclcpp-provided implementation should not support printing any kind of value corresponding to a key (e.g. a FHiCL table)?
- Is it acceptable for the printout of a sequence to look like:

```
$ fhicl-get-value my.nested.sequence my.fcl
[
  a,
  b,
  c
```

```
    ]
```

#5 - 06/10/2020 09:40 AM - Kyle Knoepfel

- Related to Feature #24526: Request that the art team take on support for fhicl-summary added

#6 - 06/10/2020 01:46 PM - Kyle Knoepfel

- % Done changed from 0 to 70

Upon further analysis and development, I propose that the utility be called **fhicl-inspect**, which will have the following command-line options:

```
$ fhicl-inspect --help
fhicl-inspect (--key|--keys-for) key [-c] <file>
Options:
  -h [ --help ]         Produce this help message
  -c [ --config ] arg   Input configuration file
  --key arg             Fully-qualified parameter key (a.b.c.d)
  --keys-for arg        Print the top-level keys of the fully-qualified 
                        parameter key, which must correspond to a FHiCL table.
```

Printed format

The following configuration:

```
table: {
  nested_value: 4
}
sequence: [a, b, @nil]
```

yields the following printouts:

```
$ fhicl-inspect --key table.nested_value test.fcl
table.nested_value: 4

$ fhicl-inspect --key table test.fcl
table: {
   nested_value: 4
}

$ fhicl-inspect --key sequence test.fcl
sequence: [
   "a",
   "b",
   @nil
]

$ fhicl-inspect --keys-for table test.fcl
nested_value
```

Please let me know if you have any concerns regarding this. I will need to determine how to handle the FHiCL file lookup policy. As FHiCL does not prescribe a specific policy, a user may wish to have a policy that differs from an art context. However, I will explore the possibilities, probably defaulting to the policy art uses.

#7 - 06/10/2020 05:02 PM - Rob Kutschke

Here is my input but you should get Andrei's too. I think that renaming the executable is OK - the number of points of maintenance is small and one-off.   However your proposal modifies the current behaviour in a way that would break how we currently use the tool.   The code that consumes the output of fhicl-getpar expects the undecorated value of the requested parameter:

For the parameter set:

```
process_name : foo 
physics : {
   path : [ a, b, c ]
}
```

We expect:

```
fhicl-getpar -string process_name file.fcl
foo
```

and

```
fhicl-getpar -string physics.path file.fcl
a
b
c
```

We actually use the latter as:

```
foo=`fhicl-getpar --strlist physics.p1  Mu2eG4/fcl/g4test_03.fcl `
echo $foo
a b c
```

I would be happy if we could use a command line argument to switch between your proposal and our current functionality. Again it's a one-off change in a small number places.

I would like to hear from Andrei if he is happy with my proposal (maybe the change needs to be made in more places than I am aware of).

#8 - 06/11/2020 08:01 AM - Kyle Knoepfel

Rob, Andrei, can you point me to the places where the current fhicl-getpar is used?

#9 - 06/11/2020 02:09 PM - Kyle Knoepfel

From Andrei:

> We use it from the worker node script in the mu2egrid package,
> https://cdcvs.fnal.gov/redmine/projects/mu2egrid/repository/revisions/master/entry/bin/impl/mu2eprodsys.sh
> or "git clone http://cdcvs.fnal.gov/projects/mu2egrid"
>
> The script is plain shell, so adding decorations to the printout is
> undesirable. What you propose is really an "inspect" script, but our
> use case is more of a "getpar" variety, where we know exactly what
> information we want to obtain, and just need to get that integer or
> string parameter from fhicl.
>
> ```
> mu2egrid (master)$ ack fhicl-getpar
> bin/impl/mu2eprodsys.sh
> 137:    local subrun=$(fhicl-getpar --int mu2emetadata.firstSubRun $localFCL)
> 141:    local run=$(fhicl-getpar --int mu2emetadata.firstRun $localFCL)
> 144:    local NSHOW=$(fhicl-getpar --int mu2emetadata.maxEvents $localFCL)
> 147:    local SEED=$(fhicl-getpar --int services.SeedService.baseSeed $localFCL)
> 288:    if fhicl-getpar --keys mu2emetadata $localFCL | grep -q ignoreSource; then
> 289:        ignoreSource="$(fhicl-getpar --int mu2emetadata.ignoreSource $localFCL)"
> 325:    # invoke fhicl-getpar as a separate command outside of for...done so that errors are trapped
> 326:    keys=( $(fhicl-getpar --strlist mu2emetadata.fcl.inkeys $localFCL) )
> 329:        rfns=( $(fhicl-getpar --strlist $key $localFCL ) )
> 353:    metalist=( $(fhicl-getpar --keys mu2emetadata.fcl $localFCL) )
> 357:            prologkeys) keys=( $(fhicl-getpar --strlist mu2emetadata.fcl.prologkeys $localFCL) )
> 364:        rfns=( $(fhicl-getpar --strlist "mu2emetadata.fcl.prolog_values.$key" $localFCL ) )
> 415:    keys=( $(fhicl-getpar --strlist mu2emetadata.fcl.outkeys $localFCL ) )
> 417:        oldname=$(fhicl-getpar --string $key $localFCL)
> ```

#10 - 06/11/2020 02:13 PM - Kyle Knoepfel

Thanks for the extra information, Andrei. Now that I understand the context a bit better, I am rethinking some of the functionality. I will update you when I have something closer to what you intended.

#11 - 06/16/2020 10:54 AM - Kyle Knoepfel

- % Done changed from 70 to 100

I have completed a first implementation (fhicl-cpp:2c927d), which I believe conforms to your request. Upon further thought, it became clear that this feature is intended to mimic the C++ interface of fhicl::ParameterSet::get<T>(key) and fhicl::ParameterSet::get_names().

Executable name and program options

The name of the executable is **fhicl-get**. Just like ParameterSet::get, the purpose of fhicl-get is to return the *value* corresponding to a parameter key. The supported command-line options are:

```
$ fhicl-get --help
```

```
Usage: fhicl-get [options] <key> <file>

Required parameters:
  <key>       A fully-qualified parameter key of the form 'a.b.c.d'
  <file>      A valid FHiCL document that contains the parameter with
              the name as specified for <key>

Supported options:
  -h [ --help ]                        Produce this help message
  --atom-as arg                        Return value for the supplied key as an
                                       atom with the provided C++ type.
  --sequence-of arg                    Return value for the supplied key as a
                                       sequence of the provided C++ type.
  --names-in                           Print the top-level names of the
                                       supplied key, which must correspond to a
                                       FHiCL table.
  --lookup-policy arg (=permissive)    File-lookup policy (see --supported-policies)
  --lookup-path arg (=FHICL_FILE_PATH) path or environment variable to be used
                                       by lookup-policy
  --supported-types                    list the C++ types supported for by the
                                       --atom-as and --sequence-of options.
  --supported-policies                 list the supported file lookup policies
```

Note that bash completions of the program options will be enabled upon setting up the fhiclcpp UPS product.

Supported C++ types

To determine which types can be used for the --atom-as or --sequence-of program options, the user may type:

```
$ fhicl-get --supported-types

For the following command line:

  fhicl-get --atom-as=T <key> <file>

the <file> is queried for the <key>, and an attempt is made to
interpret the corresponding value as an object of type T.

If instead the command line were specified as:

  fhicl-get --sequence-of=T <key> <file>

then the value corresponding to <key> would be interpreted as an
std::vector<T> object.

For either the --atom-as or --sequence-of program options, an
exception will be thrown if the <key> parameter does not exist in the
<file>, or if the parameter does not correspond to a value that can be
interpreted according to the user-specified command-line.

Supported types include:
  'double'
  'int'
  'string'
```

Use cases

For the configuration:

```
# test.fcl
globe_facts: {
  NumContinents: 7
  Bulgaria: country
  London: city
}
explorers: ["da Gama", Columbus, Magellan]
```

we get the following queries:

```
$ fhicl-get --atom-as=string globe_facts.Bulgaria test.fcl
country

$ fhicl-get --atom-as=int globe_facts.NumContinents test.fcl
7
```

```
$ fhicl-get --names-in globe_facts test.fcl
Bulgaria
London
NumContinents
```

```
$ fhicl-get --sequence-of=string explorers test.fcl
da Gama
Columbus
Magellan
```

Please let us know if this interface does not meet your needs.

#12 - 06/16/2020 01:16 PM - Kyle Knoepfel

- Status changed from Feedback to Resolved

#13 - 07/14/2020 10:37 AM - Kyle Knoepfel

- Status changed from Resolved to Closed