fhicl-cpp - Support #10613

fhicl-dump should be able to accept absolute paths

10/22/2015 04:00 PM - Christopher Backhouse

Status:	Closed	**Start date:**	10/22/2015
Priority:	Normal	**Due date:**	
Assignee:	Kyle Knoepfel	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	3.06.00	**Spent time:**	3.00 hours

Description

It would be very helpful for fhicl-dump to be able to accept an absolute file path as input (or relative to the pwd). I have a workaround with export MYPATH=/:$FHICL_FILE_PATH and -p MYPATH but it would be nice if this worked automatically.

History

#1 - 10/22/2015 07:07 PM - Kyle Knoepfel

- Tracker changed from Bug to Support

- Status changed from New to Feedback

- Assignee set to Kyle Knoepfel

An important aspect of FHiCL is the concept of the policy--namely, what types of path patterns are accepted. To that end, the feature you request is already present in fhicl-dump. If you type fhicl-dump --help, you will see:

```
  -l [ --lookup-policy ] arg (=1)       lookup policy code:
                                          0 => cet::filepath_maker
                                          1 => cet::filepath_lookup
                                          2 => cet::filepath_lookup_nonabsolute
                                          3 => cet::filepath_lookup_after1
```

The default is 1, which, interestingly, is the only policy that does **not** support the absolute file specification. If you choose any other policy, you will get the behavior you desire (e.g.):

```
[> fhicl-dump -l 0 /home/knoepfel/scratch/build-fhicl-cpp-prof/test/save-restore-ba.d/ba.fcl
Policy is 0; path is "FHICL_FILE_PATH"
a: true
b: true
```

We could do better documenting this. Please let me know if this is not sufficient to address your concern.

#2 - 10/22/2015 08:04 PM - Christopher Backhouse

I wondered if one of the -l options did what I wanted. It would indeed be good to document these with english descriptions rather than what are presumably enums from inside the code.

#3 - 10/27/2015 12:51 PM - Kyle Knoepfel

- Status changed from Feedback to Assigned

- % Done changed from 0 to 70

The printout has been improved to:

```
# Produced from 'fhicl-dump' using:
#   Input  : /home/knoepfel/scratch/build-fhicl-cpp-prof/test/save-restore-ba.d/ba.fcl
#   Policy : cet::filepath_maker
#   Path   : "FHICL_FILE_PATH"

a: true
b: true
```

It is still incumbent on us to document what the particular policy means, but the information is now more explicit.

#4 - 06/22/2020 08:54 AM - Kyle Knoepfel

- % Done changed from 70 to 100

- Target version set to 3.06.00

- Status changed from Assigned to Resolved

Well, it only took 3.5 years to finish the documentation here, but I've finally bit the bullet. The documentation will be provided as part of the fhicl-dump command-line interface:

```
$ fhicl-dump -h
fhicl-dump [-c] <file>
Options:
  -h [ --help ]                          produce this help message
  -c [ --config ] arg                    input file
  ...
  -l [ --lookup-policy ] arg (=permissive)
                                         lookup policy code (see
                                         '--supported-policies')
```

And then providing the --supported-policies command-line option:

```
$ fhicl-dump --supported-policies

The following file-lookup policies are supported:

  'after1'
     No lookup for the first file, and only lookup is performed
     for all subsequent files.

  'all'
     Lookup is performed for all files presented to the filepath-
     making object.

  'nonabsolute'
     Lookup is performed only for files that are not absolute--
     i.e. file names not beginning with the '/' character.

  'none'
     No lookup is done for any files.

  'permissive'
     The first file can be an absolute path, a path relative to '.',
     or a path that can be looked up; all subsequent files must be
     looked up.  This is the policy used by the art framework; it is
     sometimes referred to as the first-absolute-or-lookup-with-dot
     policy.
```

Please let us know if further clarification can be provided.

#5 - 07/14/2020 09:59 AM - Kyle Knoepfel

- Status changed from Resolved to Closed