fhicl - Feature #12877

undesirable behavior of @protect_ignore:

06/07/2016 05:13 PM - Andrei Gaponenko

Status:	Closed	**Start date:**	06/07/2016
Priority:	Normal	**Due date:**	
Assignee:	Christopher Green	**% Done:**	100%
Category:		**Estimated time:**	8.00 hours
Target version:	2.02.00	**Spent time:**	4.00 hours

Description

Hello,

I was wrong here:
https://cdcvs.fnal.gov/redmine/issues/8655#note-8

The original use case that lead to @protect_ignore: was to allow
scripts to pre-pend fhicl definitions to an existing config file.
Sometimes a config file needs to be handled by more than one script.
Just like the default binding operator allows to use

```
a : 3
a : 2
a : 1
```

to get a==1 without an error, the "prepending" binding should allow

```
b @protect_ignore: 1
b @protect_ignore: 2
b : 3
```

to get b==1 without an error. The current implementation
throws a "Protection violation" error on such use.

Andrei

Associated revisions

Revision 9ad8052e - 06/15/2016 01:57 PM - Christopher Green

Implement issue #12877: Revise behavior of @protect_ignore.

History

#1 - 06/13/2016 11:35 AM - Kyle Knoepfel

- Tracker changed from Bug to Feature

- Status changed from New to Accepted

- Estimated time set to 8.00 h

#2 - 06/14/2016 02:55 PM - Christopher Green

We propose the following behavior (old behavior, new behavior):

Initial Subsequent	<none>	@protect_ignore	@protect_error
<none>	Replace	Ignore	Error
@protect_ignore	Error	Error Ignore	Error
@protect_error	Error	Error	Error

Please let us know if this is acceptable to you.

#3 - 06/14/2016 03:18 PM - Andrei Gaponenko

Hello,

I'll spell out my reading of the table to make sure we interpret it in the same way. For non-qualified definitions, "replace" in the table means the traditional behavior - "the last definition wins". Any conflicting definitions involving @protect_error give an error - I think that makes sense. If a value is first defined with @protect_ignore, a subsequent non-qualified or @protect_ignore definition should be ignored, so we have "the first definition wins" rule. Yes, this is what I am asking for. Finally, the table suggests to make it en error to define a value with no qualifiers, then attempt to override it downstream with @protect_ignore. I have no strong opinion on this one. I think the suggested behavior is fine.

Andrei

#4 - 06/15/2016 08:24 AM - Christopher Green

- Status changed from Accepted to Assigned

- Assignee set to Christopher Green

#5 - 06/15/2016 01:59 PM - Christopher Green

- Status changed from Assigned to Resolved

- % Done changed from 0 to 100

Implemented with commit:9ad8052.

#6 - 08/02/2016 12:34 PM - Kyle Knoepfel

- Target version set to 2.02.00

#7 - 08/02/2016 12:35 PM - Kyle Knoepfel

- Status changed from Resolved to Closed