

art - Bug #22407

art 3.02.04 constructs modules that are not included in paths

04/18/2019 01:53 PM - Kyle Knoepfel

Status:	Closed	Start date:	04/18/2019
Priority:	Normal	Due date:	
Assignee:	Kyle Knoepfel	% Done:	100%
Category:	Infrastructure	Estimated time:	16.00 hours
Target version:	3.02.05	Spent time:	12.00 hours
Occurs In:	3.02.04	Experiment:	-
Scope:	Internal	SSI Package:	
Description			
Although the workaround includes using the --prune-config program option, this error should be fixed.			

History

#1 - 04/22/2019 10:30 AM - Kyle Knoepfel

- Estimated time set to 16.00 h
- Assignee set to Kyle Knoepfel
- Status changed from New to Assigned
- Category set to Infrastructure

#2 - 04/24/2019 09:48 AM - Kyle Knoepfel

- % Done changed from 0 to 100
- Status changed from Assigned to Resolved

The fix was implemented with commit [art:f58a81](#). It involved transferring some of the information from the configuration-pruning system to the path manager. By doing this, the framework is actually more permissive about the allowed configuration. For example, the framework now allows a user to specify (e.g.)

```
outputs.BadModuleConfig: {}
```

as long as BadModuleConfig does not appear on a trigger or end path. This varies wrt. previous behavior, where a module_type parameter, at minimum, was required, even if the configuration was ignored. Additionally, any libraries corresponding to modules that are not used by the framework are no longer loaded, which also varies wrt. previous behavior.

The fix unfortunately introduces an undesirable coupling in the PathManager test between the PathManager class and the configuration-pruning facilities. This coupling should be removed, which is described in issue [#22439](#).

#3 - 04/29/2019 11:57 AM - Kyle Knoepfel

- Target version set to 3.02.05

#4 - 04/29/2019 11:58 AM - Kyle Knoepfel

- Status changed from Resolved to Closed