

Windchill Enhancement Request

There are currently different Collection Processors in Windchill which all derive from the base PresentationProcessor class. Need the ability to customize the base PresentationProcessor so that all the derived implementations can inherit those custom collection changes.

The base com.ptc.core.htmlcomp.collection.presentation.PresentationProcessor has been implemented by (extended by) the following classes:

AbstractCollectionProcessor (has its own implementing classes)

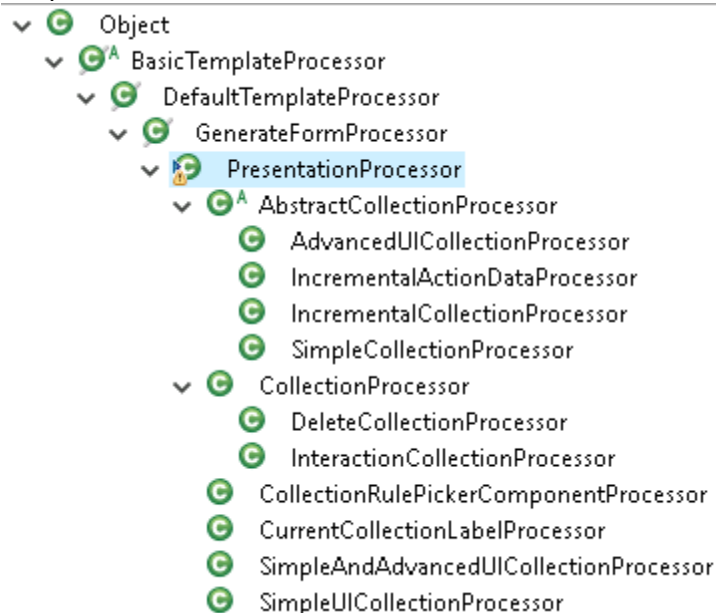
CollectionProcessor (has its own implementing classes)

CollectionRulePickerComponentProcessor

CurrentCollectionLabelProcessor

SimpleAndAdvancedUICollectionProcessor

SimpleUICollectionProcessor



There are a few methods in PresentationProcessor and AbstractCollectionProcessor that are marked private and protected and rightfully so as these classes are being implemented by other classes. So, the implementing classes can be replaced with custom implementations. That can be done by registering the custom implementations in the respective property files like \$WT_HOME/com/ptc/core/htmlcomp/htmlcomp.properties.

However, there are some core changes that we want to make that can be applied across the board. For example, we want to make a change to the getDependentsLabel() method to also check for custom dependents rule:

OOTB this code looks as follows:

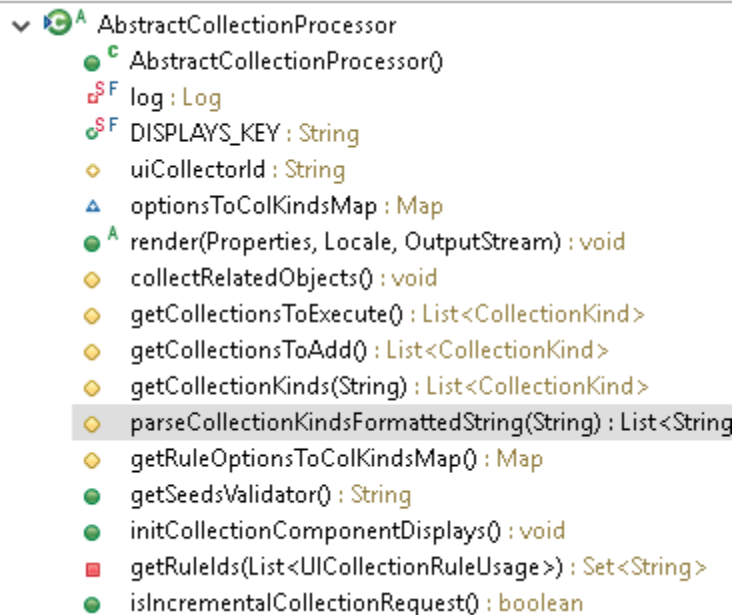
```
if (r.getRuleId().equals("DEPENDENTS"))
```

Customer wants to add additional check:

```
if (r.getRuleId().equals("DEPENDENTS")) || r.getRuleId().equals("CUSTOM_DEPENDENTS"))
```

We need API support so that the base class could be extended and registered so that all the implementing classes would be able to work off of those changes. To ensure that such a change is allowed, there are a few options. The easiest of them is to allow the user to provide a comma separated list of custom Dependent collection rules to be defined as a property or a preference. Then, the OOTB code would check the preference and work on those dependent rules instead of the hardcoding this check.

Similarly, there are some protected methods and a private method in the AbstractCollectionProcessor. Request that this AbstractCollectionProcessor methods be made public and allow custom implementations.



```
AbstractCollectionProcessor
  AbstractCollectionProcessor()
  log : Log
  DISPLAYS_KEY : String
  uiCollectorId : String
  optionsToColKindsMap : Map
  render(Properties, Locale, OutputStream) : void
  collectRelatedObjects() : void
  getCollectionsToExecute() : List<CollectionKind>
  getCollectionsToAdd() : List<CollectionKind>
  getCollectionKinds(String) : List<CollectionKind>
  parseCollectionKindsFormattedString(String) : List<String>
  getRuleOptionsToColKindsMap() : Map
  getSeedsValidator() : String
  initCollectionComponentDisplays() : void
  getRuleIds(List<UICollectionRuleUsage>) : Set<String>
  isIncrementalCollectionRequest() : boolean
```