1.1.1 Registry Filters

Purpose

To identify a subset of the set of all persistent instances of a given registry class.

Definition

```
<!ELEMENT ObjectFilter ( Clause )>

<!ELEMENT RegistryEntryFilter ( Clause )>

<!ELEMENT IntrinsicObjectFilter ( Clause )>

<!ELEMENT ExtrinsicObjectFilter ( Clause )>

<!ELEMENT PackageFilter ( Clause )>

<!ELEMENT OrganizationFilter ( Clause )>

<!ELEMENT ContactFilter ( Clause )>

<!ELEMENT ClassificationNodeFilter ( Clause )>

<!ELEMENT AssociationFilter ( Clause )>

<!ELEMENT ClassificationFilter ( Clause )>

<!ELEMENT ExternalLinkFilter ( Clause )>

<!ELEMENT ExternalIdentifierFilter ( Clause )>

<!ELEMENT SlotFilter ( Clause )>

<!ELEMENT AuditableEventFilter ( Clause )>

<!ELEMENT UserFilter ( Clause )>

<!ELEMENT PathFilter ( Clause )>

<!ELEMENT PathElementFilter ( Clause )>

<!ELEMENT SlotElementFilter ( Clause )>
```

Semantic Rules

1. The Clause element is defined in Section **Error! Reference source not found.**, Clause.
2. For every ObjectFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the RegistryObject UML class defined in [ebRIM]. If not, raise exception: *object attribute error*. The ObjectFilter returns a set of identifiers for RegistryObject instances whose attribute values evaluate to *True* for the Clause predicate.

3. For every RegistryEntryFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the RegistryEntry UML class defined in [ebRIM].
   If not, raise exception: *registry entry attribute error*. The RegistryEntryFilter returns a set of identifiers for RegistryEntry instances whose attribute values evaluate to *True* for the Clause predicate.
4. For every IntrinsicObjectFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the IntrinsicObject UML class defined in [ebRIM]. If not, raise exception: *intrinsic object attribute error*. The IntrinsicObjectFilter returns a set of identifiers for IntrinsicObject instances whose attribute values evaluate to *True* for the Clause predicate.
5. For every ExtrinsicObjectFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the ExtrinsicObject UML class defined in [ebRIM]. If not, raise exception: *extrinsic object attribute error*. The ExtrinsicObjectFilter returns a set of identifiers for ExtrinsicObject instances whose attribute values evaluate to *True* for the Clause predicate.
6. For every PackageFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the Package UML class defined in [ebRIM]. If not, raise exception: *package attribute error*. The PackageFilter returns a set of identifiers for Package instances whose attribute values evaluate to *True* for the Clause predicate.
7. For every OrganizationFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the Organization or PostalAddress UML classes defined in [ebRIM]. If not, raise exception: *organization attribute error*. The OrganizationFilter returns a set of identifiers for Organization instances whose attribute values evaluate to *True* for the Clause predicate.
8. For every ContactFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the Contact or PostalAddress UML class defined in [ebRIM]. If not, raise exception: *contact attribute error*. The ContactFilter returns a set of identifiers for Contact instances whose attribute values evaluate to *True* for the Clause predicate.
9. For every ClassificationNodeFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the ClassificationNode UML class defined in [ebRIM]. If not, raise exception: *classification node attribute error*. The ClassificationNodeFilter returns a set of identifiers for ClassificationNode instances whose attribute values evaluate to *True* for the Clause predicate.
10. For every AssociationFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the Association UML class defined in [ebRIM]. If not, raise exception: *association attribute error*. The AssociationFilter returns a set of identifiers for Association instances whose attribute values evaluate to *True* for the Clause predicate.
11. For every ClassificationFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the Classification UML class defined in [ebRIM]. If not, raise exception: *classification attribute error*. The ClassificationFilter returns a set of identifiers for Classification instances whose attribute values evaluate to *True* for the Clause predicate.

12. For every ExternalLinkFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the ExternalLink UML class defined in [ebRIM]. If not, raise exception: *external link attribute error*. The ExternalLinkFilter returns a set of identifiers for ExternalLink instances whose attribute values evaluate to *True* for the Clause predicate.
13. For every ExternalIdentiferFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the ExternalIdentifier UML class defined in [ebRIM]. If not, raise exception: *external identifier attribute error*. The ExternalIdentifierFilter returns a set of identifiers for ExternalIdentifier instances whose attribute values evaluate to *True* for the Clause predicate.
14. For every SlotFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the Slot UML class defined in [ebRIM]. If not, raise exception: *slot attribute error*. The SlotFilter returns a set of identifiers for Slot instances whose attribute values evaluate to *True* for the Clause predicate.
15. For every AuditableEventFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the AuditableEvent UML class defined in [ebRIM]. If not, raise exception: *auditable event attribute error*. The AuditableEventFilter returns a set of identifiers for AuditableEvent instances whose attribute values evaluate to *True* for the Clause predicate.
16. For every UserFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the User UML class defined in [ebRIM]. If not, raise exception: *auditable identity attribute error*. The UserFilter returns a set of identifiers for User instances whose attribute values evaluate to *True* for the Clause predicate.
17. Path is a derived, non-persistent class based on the ClassificationNode and Classification classes from ebRIM. The visible attributes of the Path class are "path" and "pathDepth". Each is derived from the corresponding method defined in ebRIM for a ClassificationNode or Classification instance. The getPath() method acts on a ClassificationNode or Classification instance to produce a character string that can be queried by the predicates of a StringClause element. The getPathDepth() method acts on a ClassificationNode or Classification instance to produce an integer that identifies the level of the referenced node and that can be queried by the predicates of an IntClause element. For an external Classification instance, getPathDepth() may return void since the depth of the node referenced by that classification may not be known if it wasn't supplied by the classifier. For every PathFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the Path class just defined. If not, raise exception: *path attribute error*. The PathFilter returns a set of Path instances whose attribute values evaluate to *True* for the Clause predicate.
18. PathElement is a derived, non-persistent class based on the ClassificationNode and Classification classes from ebRIM. The visible attributes of PathElement are "level" and "value". Each is a character string. The dynamic instances of PathElement are derived from the getPathElements() method defined in ebRIM for a ClassificationNode or Classification instance. This method returns a set of level/value pairs for each ClassificationNode or Classification instance. For an external

Classification instance, getPathElements() may return void since the explicit structure of the node referenced by that classification may not be known if it wasn't supplied by the classifier. For every PathElementFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify a public attribute of the PathElement class just defined. If not, raise exception: *path element attribute error*. The PathElementFilter returns a set of PathElement instances whose attribute values evaluate to *True* for the Clause predicate.

19. SlotElement is a derived, non-persistent class based on the Slot class from ebRIM. The visible attribute of PathElement is"value". It is a character string. The dynamic instances of SlotElement are derived from the "values" attribute defined in ebRIM for a Slot instance. For every SlotElementFilter XML element, the leftArgument attribute of any containing SimpleClause shall identify the "value" attribute of the SlotElement class just defined. If not, raise exception: *slot element attribute error*. The SlotElementFilter returns a set of Slot instances whose "value" attribute evaluates to *True* for the Clause predicate.

Example

The following is a complete example of RegistryEntryQuery combined with Clause expansion of RegistryEntryFilter to return a set of RegistryEntry instances whose objectType attibute is "CPP" and whose status attribute is "Approved".

```
<RegistryEntryQuery>
   <RegistryEntryFilter>
       <Clause>
          <CompoundClause   connectivePredicate="And" >
             <Clause>
                <SimpleClause  leftArgument="objectType" >
           <StringClause  stringPredicate="equal" >CPP</StringClause>
                </SimpleClause>
             </Clause>
             <Clause>
                <SimpleClause  leftArgument="status" >
           <StringClause  stringPredicate="equal"
                                          >Approved</StringClause>
                </SimpleClause>
             </Clause>
          </CompoundClause>
       </Clause>
   </RegistryEntryFilter>
</RegistryEntryQuery>
```