7. Registry Services

7.1 GetRegisteredObject DTD's

Purpose

To obtain one or more registered objects, and some associated metadata, by submitting a RegistryEntryQuery to the registry/repository that holds the desired objects.

NOTE: Initially, the RegistryEntryQuery is a single AssignedURN!

Definitions

Get Request DTD

```
<!ELEMENT GetRegisteredObject
   (  RegistryEntryQuery,
      RecursiveAssocOption?,
      WithShortDescription?   )>

<!ELEMENT RecursiveAssocOption ( AssociationRole+ )>
<!ATTLIST RecursiveAssocOption
   depthLimit  CDATA    #IMPLIED >

<!ELEMENT AssociationRole EMPTY >
<!ATTLIST AssociationRole
   assocRole      (%assocRoleList)      #REQUIRED  >

<!ELEMENT WithShortDescription EMPTY >
```

Get Result DTD

```
<!ELEMENT GetRegisteredObjectResult
   ( RegisteredObject*,  StatusResult  )>

<!ELEMENT RegisteredObject
   (     ClassificationScheme
      |  RegistryPackage
      |  UnknownObject
      |  WithdrawnObject
      |  RemoteObject      )>
<!ATTLIST RegisteredObject
   assignedURN          CDATA   #REQUIRED
   commonName           CDATA   #REQUIRED
   objectURL            CDATA   #REQUIRED
   objectType           CDATA   #REQUIRED
   fileType             CDATA   #IMPLIED
   registrationStatus   CDATA   #REQUIRED
   stability            CDATA   #REQUIRED
   shortDescription     CDATA   #IMPLIED  >

<!ELEMENT UnknownObject  (#PCDATA) >
<!ATTLIST UnknownObject
   byteEncoding    CDATA    #IMPLIED  >

<!ELEMENT WithdrawnObject EMPTY >

<!ELEMENT RemoteObject EMPTY >
```

Semantic Rules

1. If the RecursiveOption element is not present , then set Limit=0. If the RecursiveOption element is present, interpret its depthLimit attribute as an integer literal. If depthLimit cannot be interpreted as a positive integer, then stop execution and raise the exception: *invalid depth limit*; otherwise, set Limit=N, where N is that positive integer.

2. Set Depth=0. Let Result denote the set of RegisteredObject elements to be returned as part of the GetRegisteredObjectResult. Initially Result is empty. Let RegEnt be a set of registry entry instances. Initially RegEnt is empty.

3. If the WithShortDescription element is present, then set WSD="yes"; otherwise, set WSD="no".

4. Execute the RegistryEntryQuery according to the Semantic Rules of RegistryEntryQuery specified in Section 7.4.1. Let R be the set of RegistryEntryReference elements returned by the RegistryEntryQResult and let S be the set of status elements returned in the StatusResult. If any status element in S is an exception condition, then stop execution and return the same StatusResult element in the GetRegisteredObjectResult.

5. Execute Semantic Rules 6 and 7 with X as the set of RegistryEntry instances referenced by R. Then set Depth=Depth+1. If Depth is now equal to Limit, then return the content of Result as the set of RegisteredObject elements in the GetRegisteredObjectResult element; otherwise, continue with Semantic Rule 8.

6. Let X be a set of RegistryEntry instances. For each registry entry E in X, do the following:

   a) If E.objectURL references a registered object in this registry/repository, then create a new RegisteredObject element, with values for its attributes derived as specified in Semantic Rule 7.

      1) If E.objectType="scheme", then put the referenced ClassificationScheme DTD as the subelement of this RegisteredObject.

      2) If E.objectType="rpkg", then put the referenced RegistryPackage DTD as the subelement of this RegisteredObject.

      3) Otherwise, i.e., if the object referenced by E has an unknown internal structure, then put the content of the registered object as the #PCDATA of a new UnknownObject subelement of this RegisteredObject.

   b) If E.objectURL references a registered object in some other registry/repository, then create a new RegisteredObject element, with values for its attributes derived as specified in Semantic Rule 7, and create a new RemoteObject element as the subelement of this RegisteredObject.

   c) If E.objectURL is void, i.e. the object it would have referenced has been withdrawn, then create a new RegisteredObject element, with values for its attributes derived as specified in Semantic Rule 7, and create a new WithdrawnObject element as the subelement of this RegisteredObject.

7. Let E be a registry entry and let RO be the RegisteredObject element created in Semantic Rule 6. Set the attributes of RO to the values derived from the corresponding attributes of E. If WSD="yes", include the value of the shortDescription attribute; otherwise, do not include it. Insert this new RegisteredObject element into the Result set.

8. Let R be defined as in Semantic Rule 4. Execute Semantic Rule 9 with Y as the set of RegistryEntry instances referenced by R. Then continue with Semantic rule 10.

9. Let Y be a set of RegistryEntry instances. Let NextLevel be an empty set of RegistryEntry instances. For each registry entry E in Y, and for each AssociationRole A of the RecursiveAssocOption, do the following:

   a) Let Z be the set of associated items E' linked to E under association instances having E as the given item, E' as the associated item, and A as the association role.

   b) Add the elements of Z to NextLevel.

10. Execute Semantic Rules 6 and 7 with X as the set of new registry entries that are in NextLevel but are not yet represented in the Result set. Then add the elements of X to Y and set Depth=Depth+1. If Depth is now equal to Limit, then return the content of Result as the set of RegisteredObject elements in the GetRegisteredObjectResult element; otherwise, repeat Semantic Rules 9 and 10 with the new set Y of registry entries.

11. If any exception, warning , or other status condition results during the execution of the above, then return appropriate status elements as the StatusResult of the GetRegisteredObjectResult element created in Semantic Rule 5 or Semantic Rule 10.