

Fedora Type Checking

Datastream typechecking for the Content Model Architecture

Warning This page is no longer updated. The project is moved to <http://ecm.sourceforge.net>

Traditionally in Fedora, all data objects was just data objects, there were no types of objects. This changed with the introduction of the Content Model Architecture.

DS-COMPOSITE datastream

Content Models may contain this reserved datastream. It lists the datastreams that must exist in subscribing data objects. Other than declaring the existence of the datastreams, little else can be specified. The MIME type and the format uri are the only specification that can be used by default.

The schema for the datastream can be seen below.

```
<xsd:schema
  targetNamespace="info:fedora/fedora-system:def/dsCompositeModel#"
  xmlns="info:fedora/fedora-system:def/dsCompositeModel#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xsd:element name="dsCompositeModel">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" ref="dsTypeModel"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="dsTypeModel">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" ref="form"/>
      </xsd:sequence>
      <xsd:attribute name="ID" use="required" type="xsd:NCName"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="form">
    <xsd:complexType>
      <xsd:attribute name="FORMAT_URI" use="optional" type="xsd:anyURI"/>
      <xsd:attribute name="MIME" use="optional"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

The content of a DS-COMPOSITE datastream could look like this

```
<dsCompositeModel
  xmlns="info:fedora/fedora-system:def/dsCompositeModel#">
  <dsTypeModel ID="DC">
    <form MIME="text/xml"/>
  </dsTypeModel>
  <dsTypeModel ID="ORIGIN">
    <form MIME="text/xml"/>
  </dsTypeModel>
</dsCompositeModel>
```

Allowing extensions in DS-COMPOSITE

Since Fedora already use DS-COMPOSITE to declare the existence of datastreams, it is the natural location to specify restrictions on the contents of datastreams. Unfortunately, the schema for the DS-COMPOSITE datastream does not allow for any extra content, To that effect, we have made a small change to the schema.

```

<xsd:schema
  targetNamespace="info:fedora/fedora-system:def/dsCompositeModel#"
  xmlns="info:fedora/fedora-system:def/dsCompositeModel#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xsd:element name="dsCompositeModel">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" ref="dsTypeModel"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="dsTypeModel">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="unbounded" ref="form"/>

        <!--Changes begin-->
        <xsd:element minOccurs="0" maxOccurs="1" ref="extensions">
          <!--Changes end-->

        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="ID" use="required" type="xsd:NCName"/>
    </xsd:complexType>
  </xsd:element>

  <!-- Changes begin -->
  <xsd:element name="extensions">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="name" use="optional"/>
    </xsd:complexType>
  </xsd:element>
  <!--Changes end-->

  <xsd:element name="form">
    <xsd:complexType>
      <xsd:attribute name="FORMAT_URI" use="optional" type="xsd:anyURI"/>
      <xsd:attribute name="MIME" use="optional"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

With this changed schema, the contents could look like this:

```

<dsCompositeModel
  xmlns="info:fedora/fedora-system:def/dsCompositeModel#">

  <dsTypeModel ID="DC">
    <form MIME="text/xml"/>
    <extensions name="SCHEMA">

      </extensions>
    </dsTypeModel>
    <dsTypeModel ID="ORIGIN">
      <form MIME="text/xml"/>
      <extensions name="SCHEMA">

        </extensions>
      </dsTypeModel>
    </dsCompositeModel>

```

What is noteworthy here is that the `<dsTypeModel>` and the `<form>` elements are left unchanged. The Fedora code, working with DS-COMPOSITE only looks for these tags, so the new schema will not cause conflicts, and the extensions will be quietly ignored. This is exactly as we want, this change should not make our objects incompatible with an unmodified Fedora.

Schema extensions in DS-COMPOSITE

Now that there is a system in place for extensions to DS-COMPOSITE, looking at extensions become worthwhile. It can be said that there are three kinds of datastreams in a Fedora object:

1. xml embedded in the object
2. bytes embedded in the object
3. external file referenced by URL.

The schema extension will only concern itself with the first option, namely the xml embedded in the datastream. For XML, there already exist a commonly accepted system for specifying the content, i.e. XML Schema. But where to place the schema, then? Embedding it directly in DS-COMPOSITE makes for a very unreadable datastream. Alternatively, you could just specify an URL to the schema, but this approach have problems too. Having the Content Model depend on schemas defined elsewhere, perhaps on remote servers, mean that the content models could break by actions totally unrelated to the repository. The best way, we have found, is to embed the schema in a datastream, either in the content model itself, or in another object. To that purpose we have defined the following extension schema

```
<xsd:schema
  targetNamespace="http://doms.statsbiblioteket.dk/types/dscompositeschema/0/1/#"
  xmlns="http://doms.statsbiblioteket.dk/types/dscompositeschema/0/1/#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xsd:element name="schema">
    <xsd:complexType>
      <xsd:attribute name="type" use="required" type="typetype"/>
      <xsd:attribute name="datastream" use="required" type="idType"/>
      <xsd:attribute name="object" use="optional" type="pidType"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:simpleType name="typetype">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="xsd"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="idType">
    <xsd:restriction base="xsd:ID">
      <xsd:maxLength value="64"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="pidType">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="64"/>
      <xsd:pattern value="([A-Za-z0-9]|-|\.|+)(([A-Za-z0-9])|-|\.|~|_|{0-9A-F}{2}))+"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

Using that extension, the DS-COMPOSITE datastream could look like this

```
<dsCompositeModel
  xmlns="info:fedora/fedora-system:def/dsCompositeModel#"
  xmlns:schema="http://doms.statsbiblioteket.dk/types/dscompositeschema/0/1/#">

  <!-- The DC datastream is declared. It's mime type must be text/xml. It must adhere to the xml schema
residing in the DC_SCHEMA datastream in the "doms: DublinCore_Schema" object. -->
  <dsTypeModel ID="DC">
    <form MIME="text/xml"/>
    <extensions name="SCHEMA">
      <schema:schema type="xsd" datastream="DC_SCHEMA" object="info:fedora/doms: DublinCore_Schema"/>
    </extensions>
  </dsTypeModel>

  <!-- The ORIGIN datastream is declared. It's mime type must be text/xml. It must adhere to the xml schema
residing in the ORIGIN_SCHEMA datastream in this content model-->
  <dsTypeModel ID="ORIGIN">
    <form MIME="text/xml"/>
    <extensions name="SCHEMA">
      <schema:schema type="xsd" datastream="ORIGIN_SCHEMA"/>
    </extensions>
  </dsTypeModel>

</dsCompositeModel>
```