

Declaring Content Types for Web Services

Ümit Yalçınalp
Anish Karmarkar
Oracle Corporation
May 19th 2004
WSD F2F Meeting

Goals

- Keep it simple
- Stick to well used conventions
- Address wildcarding

Mechanism to Indicate

- the media type of an *element information item* whose type is *xs:base64Binary*.
- the acceptable media type(s) for an element whose type is *xs:base64Binary* using XML Schema annotation

Document Defines

- `xmlmime` Namespace (to be finalized at publication)
- `mediaType` attribute
 - Value: IANA token based
- `acceptedMediaType` element
 - Schema annotation
 - Value: IANA token based
- Value patterns

Binary Element Information Item

- Base Type is `xs:base64Binary`
- Contains `xmlmime:mediaType` attribute

```
<xs:complexType name="Picture">
  <xs:simpleContent>
    <xs:extension base="xs:base64Binary" >
      <xs:attribute ref="xmlmime:mediaType"
        use="optional" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Schema Annotation

```
<xs:element name="companyPicture" type="tns:Picture"  
  <xs:annotation>  
    <xs:documentation>This element designates the  
    range of values that the declared binary type will  
    accept  
    </xs:documentation>  
    <xs:appInfo>  
    <xlmime:acceptMediaType>image/*</xlmime:acceptedMe  
    diaType>  
    </xs:appInfo>  
  </xs:annotation>  
</xs:element>
```

Task Force Discussion

- Can we use only one attribute?
 - Not possible to use wildcards
 - Need to resolve XMLP issue: 443
- Considered an alternative design
 - Use top level type for each, ie. Image
 - Require binary data to derive from one
 - More complicated
- Schema annotation/Schema extensibility
- Result: Current Document