

distribution.url

The additionalAttribute Name OPeNDAPServer was originally classed as a distributionInformation type additionalAttribute. However, as we examined the other additionalAttributes in that group, it became clear that OPeNDAPServer was unique in that it was a simple URL.

Examples

/Granule/AdditionalAttributes/AdditionalAttribute

```
<AdditionalAttribute>
  <Name>OPeNDAPServer</Name>
  <Values>
    <Value>http://acdisc.gsfc.nasa.gov/opendap/EarthProbe_TOMS_Level3/TOMSEPL3.008/contents.html</Value>
  </Values>
</AdditionalAttribute>
```

/gmi:MI_Metadata/gmd:distributionInfo/gmd:MD_Distribution/gmd:distributor/gmd:MD_Distributor/gmd:distributorTransferOptions/gmd:MD_DigitalTransferOptions

```
<gmd:MD_DigitalTransferOptions>
  <gmd:onLine>
    <gmd:CI_OnlineResource>
      <gmd:linkage>
        <gmd:URL>http://acdisc.gsfc.nasa.gov/opendap/EarthProbe_TOMS_Level3/TOMSEPL3.008/contents.html</gmd:URL>
      </gmd:linkage>
      <gmd:description>
        <gco:CharacterString>OPeNDAPServer</gco:CharacterString>
      </gmd:description>
      <gmd:function>
        <gmd:CI_OnLineFunctionCode codeList="http://www.ngdc.noaa.gov/metadata/published/xsd/schema/resources/Codelist/gmxCodeLists.xml#CI_OnLineFunctionCode"
          codeListValue="download">download</gmd:CI_OnLineFunctionCode>
      </gmd:function>
    </gmd:CI_OnlineResource>
  </gmd:onLine>
</gmd:MD_DigitalTransferOptions>
```

Standard Content vs. Additional Attributes

Facilitating direct access to data is an important capability of any metadata dialect, so most dialects include mechanisms for describing access URLs. The ECHO dialect is no exception. [The ECHO Data User's Guide](#) describes two types of online resources in ECHO metadata:

```
<OnlineAccessURLs>
  <OnlineAccessURL>
    <URL>ftp://daac.nasa.gov/granule_1234.zip</URL>
    <URLDescription>Compressed data granule</URLDescription>
    <MimeType>application/zip</MimeType>
  </OnlineAccessURL>
</OnlineAccessURLs>

<OnlineResources>
  <OnlineResource>
    <URL>http://daac.nasa.gov/products/product_A.html</URL>
    <Description>Main product overview page.</Description>
    <Type>Product Overview</Type>
    <MimeType>text/html</MimeType>
  </OnlineResource>
</OnlineResources>
```

Either of these approach could be used to include an OPeNDAPServer in ECHO metadata without defining a new additionalAttribute. This is an interesting example of the difference between "soft" and "hard" typing in metadata dialects. Hard typing is an approach that uses very specific names to define elements in metadata. In this case, completely hard typing would be done using an element named OPeNDAPServer which would look like:

```
<OPeNDAPServer>http://acdisc.gsfc.nasa.gov/opendap/EarthProbe_TOMS_Level3/TOMSEPL3.008/contents.html</OPeNDAPServer>
```

This approach unambiguously describes the semantics of the element but it places difficult requirements on clients that must know a priori the names of all links in the metadata dialect in order to find them. For example, the element <OGCWMSServer> would give the address of an OGC Web Map Service or <WCS> would give the address of a WCS server.

The approach used by ECHO (illustrated above) is a good example of soft typing. This approach uses a general structure that includes an element that gives the type of the structure. In the ECHO case, the general structures are the OnlineAccessURL and OnlineResource and the MimeType or Description /URLDescription elements in these structures define the types. The OnlineAccessURL and OnlineResource structures are said to be "soft typed".

This approach makes it say for clients to ask the common question: "what links are available for this resource?" and to get an answer that includes helpful descriptive information. In the hard typing case, this question can not be asked without knowing all of the element names that contain links. In addition, the client would need to know information about how to use each of those link types.

One of the benefits of the ECHO approach to defining AdditionalAttributes is that they are, in fact, soft typed in that they are all in AdditionalAttribute elements with the name elements providing information about which Additional Attribute each is. The original design, i.e. without an AdditionalAttributeType, has the same hard typed problem. A client needs to know the names of all AdditionalAttributes that include links in order to answer the critical and simple "what links" question.