

# Large Data BLOBs in DFDL

# DFDL and Images

- DFDL for Images and Video ??
  - This was originally not in scope for DFDL
- Large demand to use DFDL on the metadata content of image file formats
- How to skip the raw data parts?
  - They have to be preserved, but there's no parsing to be done.

# Modeling Binary BLOBs in DFDL

- Problematic: Large Element of xs:hexBinary type
  - Maximum size restricted by JVM (max Java object is 2G)
  - Text tools (viewers, editors, formatters) are text and line oriented.
    - Tend to crash or freeze on really long lines (megabytes long lines)
  - *A major appeal of using text (XML, JSON) is you can just look at it with basic text tools!*
    - *Giant hexBinary elements make this impractical.*
- Preferred: Use Array of Elements of xs:hexBinary type
  - Each xs:hexBinary element is small (perhaps 64 bytes)
  - JVM objects are all small.
  - Text lines are kept short.

# BLOB as Array of HexBinary Elements

```
<element name="blob" dfdl:lengthKind="explicit" dfdl:length="..." >
<complexType>
<sequence>
<element name="data" minOccurs="0" maxOccurs="1048576" dfdl:occursCountKind="implicit">
<choice>
<element name="line" type="xs:hexBinary" dfdl:lengthKind="explicit" dfdl:length="64"/>
<element name="final" type="pre/blobFinal" dfdl:lengthKind="delimited"/>
</choice>
</element>
</sequence>
</complexType>
</element>

<simpleType name="blobFinal" dfdl:encoding="iso-8859-1">
<annotation><appinfo source="http://www.ogf.org/dfdl/">
<dfdl:assert test="{ dfdl:checkConstraints(.) }" /><!-- length facets below required for parse -->
</appinfo></annotation>
<restriction base="xs:hexBinary">
<minLength value="1"><!-- must be at least 1 byte -->
<maxLength value="63"/> <!-- must be shorter than 64 bytes -->
</restriction>
</simpleType>
```

# BLOB as Array of HexBinary Elements

- Blob looks like this in XML:

```
<blob>
    <data><line>93fbac37 ... bf</line></data>
    <data><line>fc8c3439 ... ab</line></data>
    ...
    <data><final>a93821</final></data>
</blob>
```

- Tool and text editor friendly
- Does not need large JVM objects
- *Note: Parse/Unparse speed - slower than 1 large object*

# Daffodil BLOB Extension for DFDL

- Possible alternative
- Writes blob data out to external file to avoid processing it as text (XML, JSON, etc.)
  - Blob bits are not part of the infoset.
  - A temp filename is present in the infoset instead.
- Caution: may not be supported by daffodil applications
  - requires temp file space, temp file management, etc.
- See Daffodil Extensions Documentation
  - on web: [daffodil.apache.org/binary-large-objects](https://daffodil.apache.org/binary-large-objects)