

# XML Schema for Job Definition Format

Graham Mann

Internet Printing Group, Adobe Systems Inc

[gmann@adobe.com](mailto:gmann@adobe.com)

# Agenda

- Why use XML schema
- Summary of XML schema capability
- Limitations of XML schema
- JDF schema and namespaces
- Validation
- Conclusion

# Job Definition Format

- Defined by the JDF CIP4 Specification
  - <http://www.cip4.org/>
- XML Schema
  - <http://www.w3.org/XML/Schema>

# Why use XML schema

- **Why schema at all**
  - Provides a 'Contract' between parties exchanging JDF
  - Validation or conformance
- **Constrain XML documents to JDF**
  - Content and datatypes
- **Why choose XML schema (and not DTD's)**
  - Namespace support - JDF specification requirement
  - Stronger data typing

# Other advantages of XML schema

- **Take advantage of other work**
  - Tools, editors validators etc
  - XSLT, XPath etc
- **Better understanding of JDF**
  - Enable a 'visualization' of JDF
  - Creation of compliant examples

# Summary of XML schema capability

- **Definition of Content**
  - JDF nodes, child elements and attributes
- **Datatypes**
  - Primitives, derived types, enumerations etc
- **Reusability**
  - Complex and simple types
  - Create new types by extending existing ones
  - Types from other namespaces

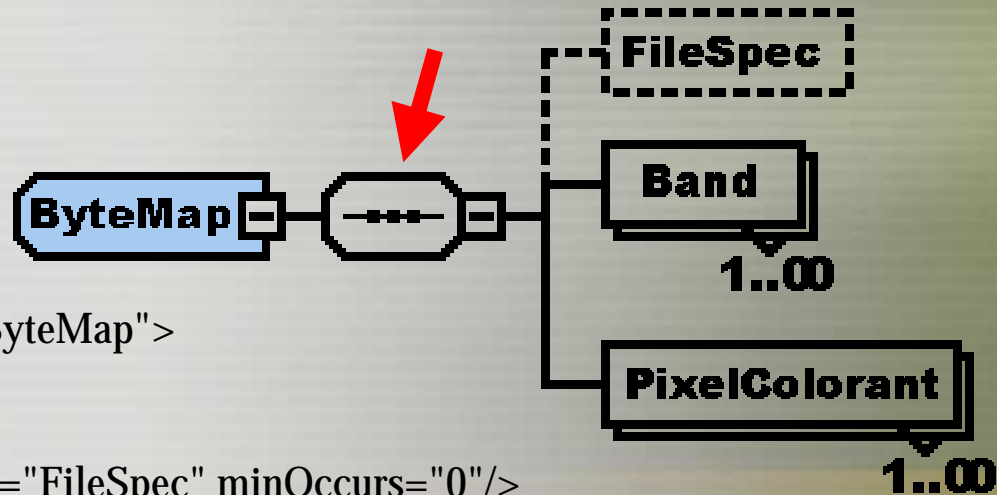
# XML schema datatypes

- **Simple Type**
  - string, integer, boolean etc
  - Used in JDF attributes
- **Complex Type**
  - Contain child elements and attributes
  - eg JDF type ParameterResource
- **Derive new simple and complex types**
  - eg JDF type XYPair
  - eg JDF type ByteMap

# Complex Type

```
<element name="ByteMap">  
  <complexType>  
    <sequence>  
      <element name="FileSpec" minOccurs="0"/>  
      <element name="Band" maxOccurs="unbounded"/>  
      <element name="PixelColorant" maxOccurs="unbounded"/>  
    </sequence>  
    <attribute name="BandOrdering"  
              type="NMTOKEN" use="optional"/>  
    <attribute name="Resolution"  
              type="jdf:XYPair" use="required"/>  
  </complexType>  
</element>
```

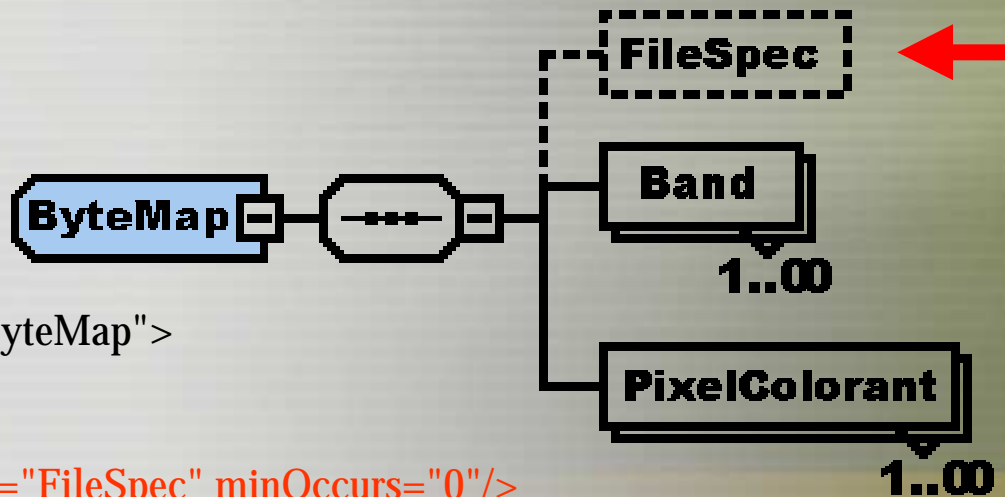
# JDF example Content model



```
<element name="ByteMap">  
  <complexType>  
    <sequence>  
      <element name="FileSpec" minOccurs="0"/>  
      <element name="Band" maxOccurs="unbounded"/>  
      <element name="PixelColorant" maxOccurs="unbounded"/>  
    </sequence>  
    <attribute name="BandOrdering" type="NMTOKEN" use="optional"/>  
    <attribute name="Resolution" type="jdf:XYPair" use="required"/>  
  </complexType>  
</element>
```

# JDF example

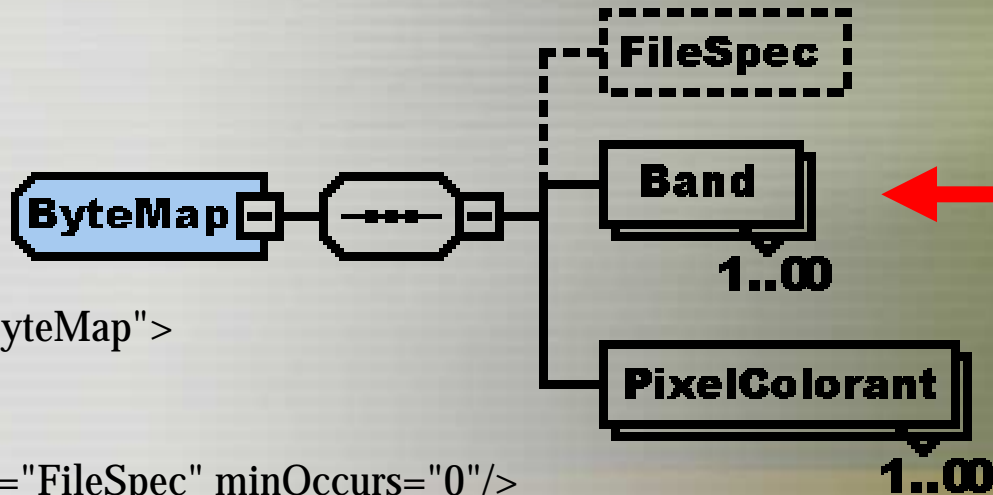
## Optional child element



```
<element name="ByteMap">
  <complexType>
    <sequence>
      <element name="FileSpec" minOccurs="0"/>
      <element name="Band" maxOccurs="unbounded"/>
      <element name="PixelColorant" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="BandOrdering" type="NMTOKEN" use="optional"/>
    <attribute name="Resolution" type="jdf:XYPair" use="required"/>
  </complexType>
</element>
```

# JDF example

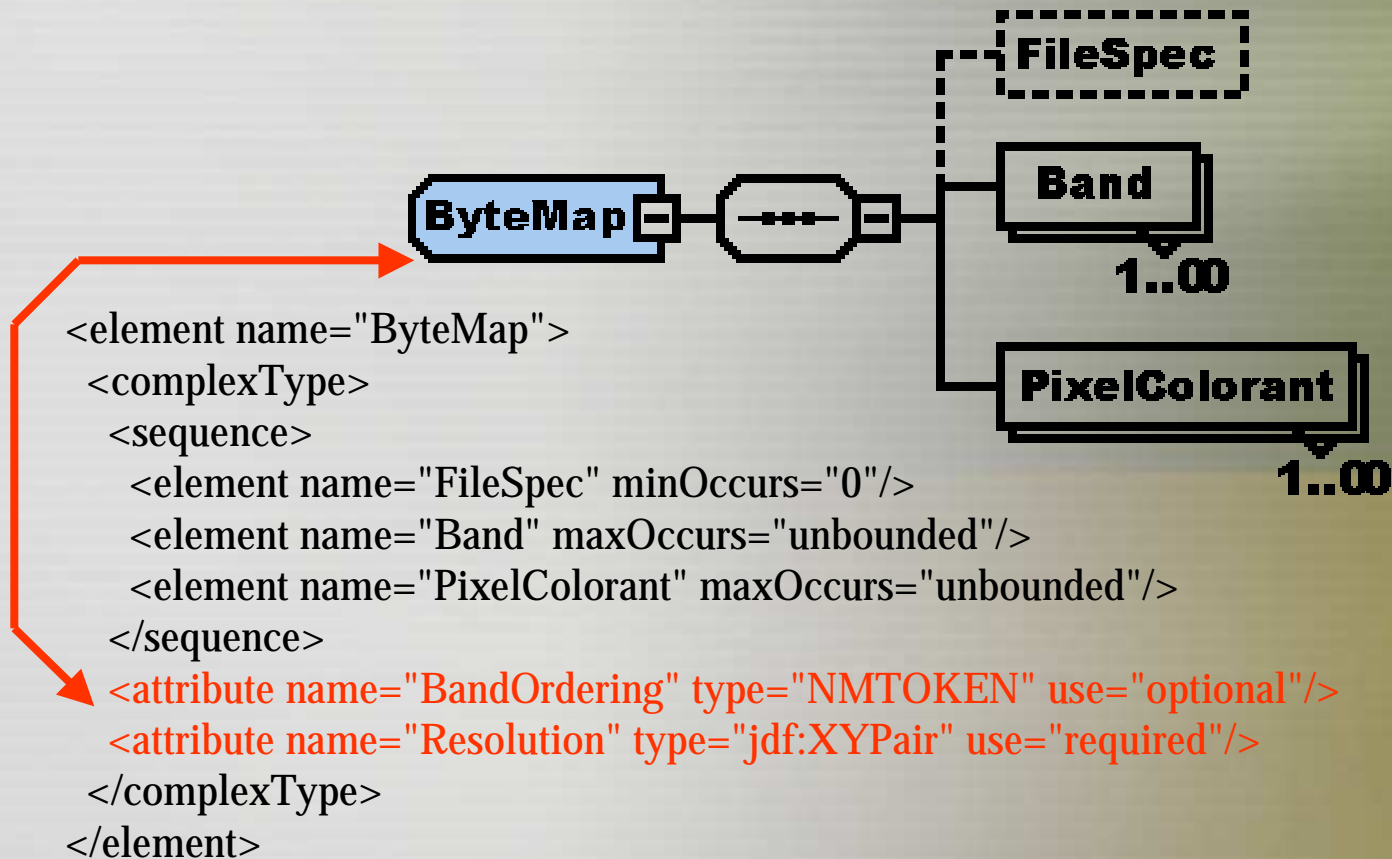
## Unbounded child elements



```
<element name="ByteMap">  
  <complexType>  
    <sequence>  
      <element name="FileSpec" minOccurs="0"/>  
      <element name="Band" maxOccurs="unbounded"/>  
      <element name="PixelColorant" maxOccurs="unbounded"/>  
    </sequence>  
    <attribute name="BandOrdering" type="NMTOKEN" use="optional"/>  
    <attribute name="Resolution" type="jdf:XYPair" use="required"/>  
  </complexType>  
</element>
```

# JDF example

## ByteMap attributes



# Summary of complex types

- Can be based upon other complex types
- Content model for child elements
  - Sequence, Choice, All
  - Nested and/or repeated
- Child elements
  - Declared locally
  - Complex types
- Attributes
  - Simple types
  - Groups of attributes

# Limitations of XML schema

- No match for the common JDF content model
- No context validation
- Complex and therefore large schema
- Can't prove JDF makes sense

# Requirements of JDF Specification content model

- Child elements can occur in any order
- Many different child elements
- Some child elements may occur more than once
- JDF comment elements can occur wherever required
- Elements and attributes from other namespaces can be included wherever desired

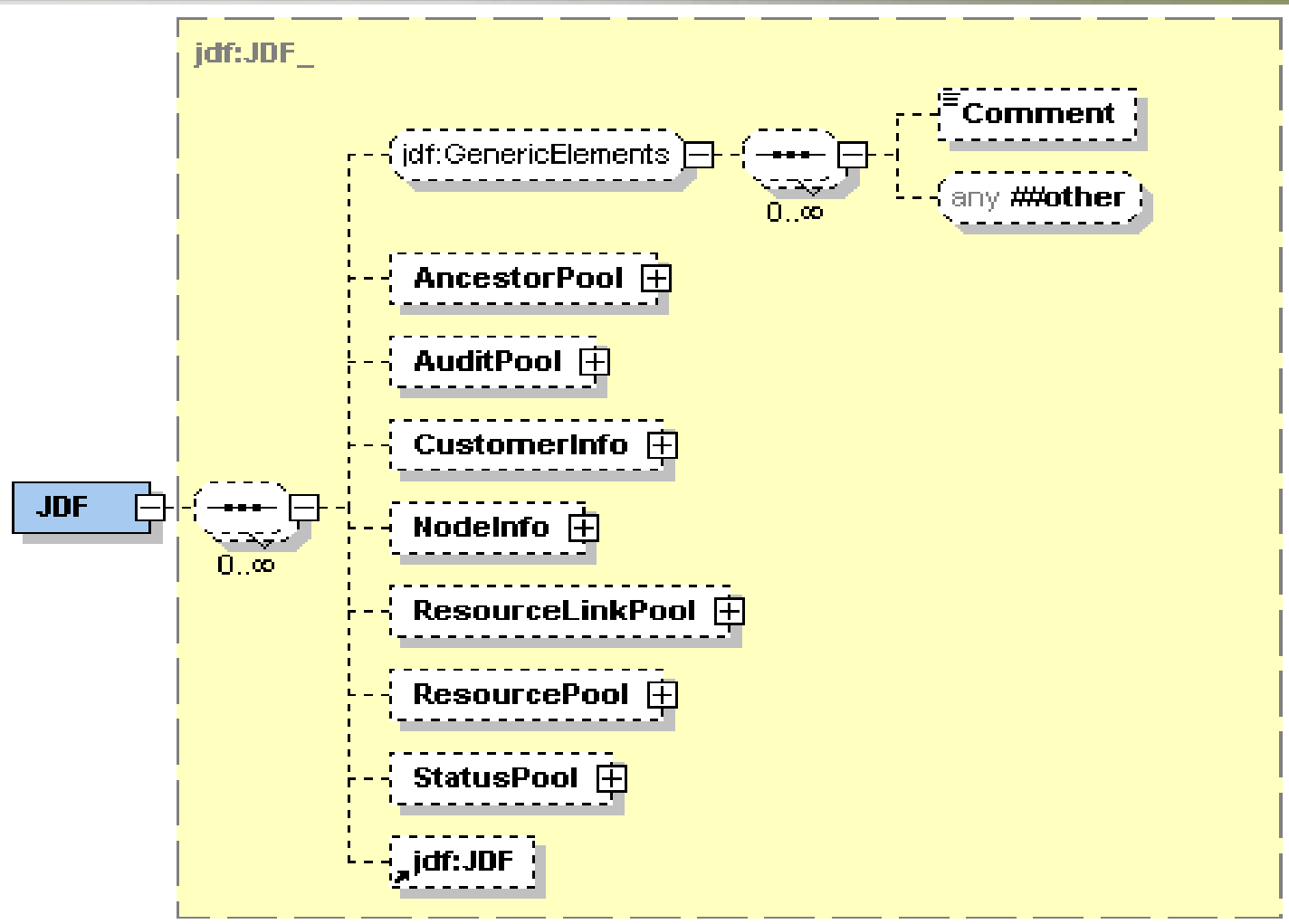
# XML schema content models

- **Sequence**
  - Strict order imposed
- **Choice**
  - Only one child selected
- **All**
  - May appear once or not at all
- **Allow types from other namespaces**
  - `<any namespace="##other" processContents="lax"/>`
- **Extending complex types imposes order**

# Solution for JDF schema

- **Complex type content model**
  - choice - minOccurs="0" maxOccurs="unbounded"
- **For individual element**
  - minOccurs="0" maxOccurs="1"
- **Base types only contain attributes**
  
- **Shortfall with the solution**
  - Effective Occurs becomes 0 - unbounded

# Content model for JDF root element



# JDF requires context validation

- Relating resources to processes
- Mandatory attributes for partitioned resources
- ...

# Process/Resource relationship

- Processes use ResourceLinks to locate resources
- Links only 'point' to the resource
- Schema can ...
  - check that a process has a resource link pool
  - check that resource links are well formed
  - not check they refer to appropriate resources

# Inheriting Status (Partitioned resources)

- Partitioned resources require status information
  - From itself or closest ancestor
  - Cyan Yellow Black - Available
  - Magenta - Unavailable

```
<ExposedMedia Class="Handling" Brand="Gooney" ID="L1" Status="Available" ... >
  <ExposedMedia SheetName="S1">
    <ExposedMedia Side="Front">
      <ExposedMedia Separation="Cyan" ProductID="S1FCPlateJ42"/>
      <ExposedMedia Separation="Magenta" Status="Unavailable"
        ProductID="S1FMPlateJ42"/>
      <ExposedMedia Separation="Yellow" ProductID="S1FYPlateJ42"/>
      <ExposedMedia Separation="Black" ProductID="S1FKPlateJ42"/>
    </ExposedMedia>
  </ExposedMedia>
</ExposedMedia>
```

- Solution - Partition resource attributes optional

# Specification vs Schema rule

Where the schema cannot represent the specification ...

- **Defer to the specification**
- **Schema constrains loosely**
  - Validation always accepts any valid JDF
  - Thus it may accept non valid JDF
- **Resolution is left to any subsequent process**
  - Secondary validation
  - Embedded in application

# Schema and Namespaces

- **Qualify names**
- **JDF namespace**
  - `xmlns:jdf="http://www.CIP4.org/JDFSchema_1"`
- **schemaLocation**
  - Allows parsers to find the schema grammar
- **Any XML content**
  - `<any namespace="##other" processContents="lax"/>`
- **Schema Import**
  - New schemas have access to JDF types

# JDF extension

- **Simple - No additional schema**
  - Supported by the JDF schema
- **Complex - With additional schema**
  - Can be validated
  - Aid general acceptance

# No additional schema

- **Declare a namespace in the JDF document**
  - `xmlns:adobe="http://ns.adobe.com/JDF/Extension"`
- **No schemaLocation required**
  - Validation is set to 'lax'

```
<JDF ...
  xmlns:adobe="http://ns.adobe.com/JDF/Extension"
  ...
  <ResourcePool>
    ...
    <adobe:SpecialPrintingParams ID="ID_001" ... />
    ...
  </ResourcePool>
</JDF>
```

# With additional schema

- **Create a schema for the extension**
  - Import JDF schema
  - Create new type(s) - based on jdf types
  - Obey JDF specification rules
- **Document**
  - Declare a namespace
  - Use schemaLocation

# Example extension schema

```
<schema
  targetNamespace="http://ns.adobe.com/JDF/Extension" .../>
  <import namespace="http://www.CIP4.org/JDFSchema_1" .../>

  <element name="SpecialPrintingParams"
            type="ext:SpecialPrintingParamsType"/>
  <complexType name="SpecialPrintingParamsType">
    <sequence minOccurs="0" maxOccurs="unbounded">
      <group ref="jdf:GenericElements" minOccurs="0"/>
      <element ... />
    </sequence>
    <attribute name="ID" type="ID" use="required"/>
  </complexType>
</schema>
```

# Example extension schema

```
<schema
  targetNamespace="http://ns.adobe.com/JDF/Extension" .../>
  <import namespace="http://www.CIP4.org/JDFSchema_1" .../>

  <element name="SpecialPrintingParams"
            type="ext:SpecialPrintingParamsType"/>
  <complexType name="SpecialPrintingParamsType">
    <sequence minOccurs="0" maxOccurs="unbounded">
      <group ref="jdf:GenericElements" minOccurs="0"/>
      <element ... />
    </sequence>
    <attribute name="ID" type="ID" use="required"/>
  </complexType>
</schema>
```

# Example extension schema

```
<schema
  targetNamespace="http://ns.adobe.com/JDF/Extension" .../>
<import namespace="http://www.CIP4.org/JDFSchema_1" .../>

<element name="SpecialPrintingParams"
          type="ext:SpecialPrintingParamsType"/>
<complexType name="SpecialPrintingParamsType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <group ref="jdf:GenericElements" minOccurs="0"/>
    <element ... />
  </sequence>
  <attribute name="ID" type="ID" use="required"/>
</complexType>
</schema>
```

# Example extension schema

```
<schema
  targetNamespace="http://ns.adobe.com/JDF/Extension" .../>
<import namespace="http://www.CIP4.org/JDFSchema_1" .../>

<element name="SpecialPrintingParams"
          type="ext:SpecialPrintingParamsType"/>
<complexType name="SpecialPrintingParamsType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <group ref="jdf:GenericElements" minOccurs="0"/>
    <element ... />
  </sequence>
  <attribute name="ID" type="ID" use="required"/>
</complexType>
</schema>
```

# Example using extension schema

```
<JDF ...  
  xmlns:adobe="http://ns.adobe.com/JDF/Extension"  
  schemaLocation=" ... .."  
  ...  
  <ResourcePool>  
    ...  
    <adobe:SpecialPrintingParams ID="ID_001" ... />  
  </ResourcePool>  
</JDF>
```

# Available XML schema validation tools

- **Editors**
  - <http://www.xmlspy.com>
  - <http://www.extensibility.com>
- **Apache Xerces J**
  - <http://xml.apache.org/>
  - Sample validation code
  - `java sax.SAX2Count -v Example.jdf`
    - [Error] Example.jdf Datatype error:
      - Attribute ID must be declared for element “ByteMap”
- **Others**

# Conclusions

- **XML Schema for JDF is beneficial**
  - Provides contract between participating parties
  - Validation or conformance
- **Schema work is ongoing**
  - Available to CIP4 members in the 'Tools and Infrastructure' working group
- **Get a copy and start work ASAP**