Mapping between DFDL 1.0 Infoset and XML Data Model

Status of This Document

This working draft document provides information to the OGF community on the Data Format Description Language (DFDL) standard and its interoperability with XDM. Distribution is unlimited

Abstract

This document defines the mapping from DFDL 1.0 Infoset to W3C XDM, and from W3C XDM to DFDL 1.0 infoset.
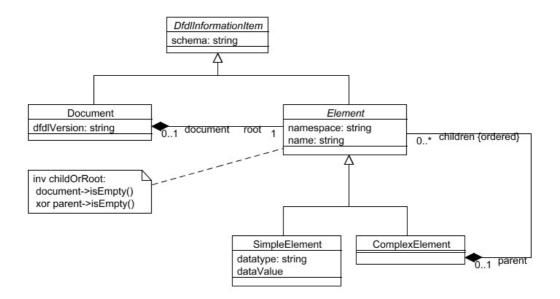
Table of contents

1. Introduction

The W3C XML Data Model [XDM] is the data model for XPath 2.0, XSLT 2.0 and XQuery. This document defines the mapping from DFDL 1.0 [DFDL] Infoset to XDM, and from XDM to DFDL 1.0 Infoset. This enables XML tools that use XDM as a canonical data model to include DFDL processors and thereby enable non-XML data to appear as 'virtual' XML data.

Here is a UML class diagram of the DFDL Infoset, reproduced from the DFDL 1.0 specification.

There are three kinds of concrete DFDL information item, Document, SimpleElement and ComplexElement. There are seven kinds of Nodes in the XDM, but when mapping from and to DFDL only four are used, Document Node, Element Node, Text Node and Namespace Node.

When mapping to DFDL, the mapping fails and produces no Infoset if the XDM contains anything that can not be represented in the DFDL Infoset. This includes, but is not limited to, node kinds other than Document, Element, Text and Namespace, data types other than those supported by DFDL, more than one Document node, multiple children in a Document node, a Text Node that is a sibling of an Element Node.

The following sections define the mapping for each kind of XDM node.

2. Document Node

Document Nodes have the following properties:

o **base-uri**, possibly empty.
o **children**, possibly empty.
o **unparsed-entities**, possibly empty.
o **document-uri**, possibly empty.
o **string-value**
o **typed-value**

Document information items have the following properties:

o **[root]**
o **[dfdlVersion]**
o **[schema]**, possibly empty

2.1. DFDL Infoset to XDM

A Document Node is constructed for a Document information item. Document Node properties are derived from the Document information item as follows.

base-uri

Empty. The DFDL Infoset does not contain document URIs.

children

The sequence of exactly one Element Node constructed from the Element information item found in the **[root]** property.

unparsed-entities

Empty.

string-value

The concatenation of the **string-value** properties of all descendant Text Nodes.

typed-value

The value of **string-value** as xs:untypedAtomic[1].

document-uri

Empty. The DFDL Infoset set does not contain document URIs.

2.2. XDM to DFDL Infoset

A Document information item is constructed for a Document Node. Document information item properties are derived from the Document Node as follows.

---

[1] One of five additional types added by the XML Data Model, see [XDM].

[root]

> The Element information item constructed from the **children** property, which must be a single Element node.

[dfdlVersion]

> The string "dfdl-1.0".

[schema]

> Empty.

3. Element Node

Element Nodes have the following properties:

- o **base-uri**, possibly empty.
- o **node-name**
- o **parent**, possibly empty
- o **type-name**
- o **children**, possibly empty
- o **attributes**, possibly empty
- o **namespaces**
- o **nilled**
- o **string-value**
- o **typed-value**
- o **is-id**
- o **is-idrefs**

SimpleElement information items have the following properties:

- o **[namespace]**
- o **[name]**
- o **[document]**
- o **[parent]**, possibly empty
- o **[children]**, has "no value"
- o **[datatype],** being the name of a built-in XML Schema simple type
- o **[datavalue],** possibly special value *"nil"*
- o **[schema]**, possibly empty

ComplexElement information items have the following properties:

- o **[namespace]**
- o **[name]**
- o **[document]**
- o **[parent]**, possibly empty
- o **[children]**, possibly empty
- o **[datatype]**, has "no value"
- o **[datavalue]**, has "no value"
- o **[schema]**, possibly empty

Note that "no value" is distinct from empty, the empty string or *"nil"*. It is used when a property has no meaning in the context of a ComplexElement or SimpleElement.

3.1. DFDL Infoset to XDM

An Element Node is constructed for each SimpleElement and ComplexElement information item. Element Node properties are derived from the Element information item as follows.

Note: SimpleElement [datavalue] values may contain characters that are illegal in XML, for example, DFDL strings can contain the character code 0 (zero) within them, but XML does not allow this character code in any XML content even if it is represented as a character entity. Nevertheless, a DFDL described string is mapped to an XDM string data value.

base-uri

> Empty. The DFDL Infoset does not contain node URIs.

node-name

xs:QName constructed from the **[name]** and **[namespace]** properties.

parent

If **[parent]** property is empty, then the Document Node, otherwise the Element Node that corresponds to the value of **[parent]**.

type-name

ComplexElement: xs:anyType.

SimpleElement: xs:QName, the local name being the value of **[datatype]** and the namespace being "http://www.w3.org/2001/XMLSchema".

Optionally, if **[schema]** property is not empty, an implementation may use the value of **[schema]** (a Schema Component Designator) to obtain the actual type definition from the referenced element declaration, and then use its name and namespace to set **type-name**. If so, then for SimpleElement it becomes possible for the value of **[datavalue]** property to be invalid according to the type, in which case **type-name** must be set to xs:anySimpleType.

children

ComplexElement: If **[children]** is empty, then the empty sequence, otherwise the sequence of Element Nodes constructed from the Element information items found in **[children]**.

SimpleElement: Optionally a Text Node constructed from **[datavalue]** property (see section 4)

attributes

Empty sequence.

namespaces

If **[namespace]** property is not empty, then a Namespace Node constructed from the value of **[namespace]** (see section 5), otherwise the empty sequence.

nilled

If **[datavalue]** property has special value *"nil"*, then *"true",* otherwise "*false*".

string-value

ComplexElement: The concatenation of the **string-value** properties of all descendant Text Nodes.

SimpleElement: If the value of **[datavalue]** is special value *"nil"*, then the empty string, otherwise the value of **[datavalue]** converted to its canonical lexical representation.

typed-value

ComplexElement: The value of **string-value** as xs:untypedAtomic.

SimpleElement: If the value of **[datavalue]** is special value *"nil",* then the empty sequence, otherwise the value of **[datavalue]**.

Optionally, if **type-name** has been set from the actual type definition using **[schema]** property, then **typed-value** is set as follows:

ComplexElement: Undefined.

SimpleElement: If the value of **[datavalue]** is special value *"nil",* then the empty sequence, else if **type-name** is xs:anySimpleType, then **string-value** as xs:untypedAtomic, otherwise the value of **[datavalue]**.

If **typed-value** is not undefined, the relationship between the **type-name**, **typed-value**, and **string-value** properties of an Element node is consistent with XML Schema validation.

is-id

> "*false*".

is-idrefs

> "*false*".

3.2. XDM to DFDL Infoset

A SimpleElement or ComplexElement information item is constructed for an Element Node, depending on whether the **type-name** property designates a simple type or a complex type. Element information item properties are derived from the Element Node as follows.

[namespace]

> The namespace name part of the value of **node-name** property.

[name]

> The local name part of the value of **node-name** property.

[document]

> The Document information item constructed from the Document Node.

[parent]

> If **parent** property is the Document Node, then empty, otherwise the Element information item corresponding to the Element Node that is the value of the **parent** property.

[children]

> *ComplexElement:* If **children** property is the empty sequence, then empty, otherwise the sequence of Element information items constructed from the Element Nodes found in **children**.
>
> *SimpleElement:* Has "no value".

[datatype]

> *ComplexElement:* Has "no value".

> *SimpleElement:* If **type-name** refers to a built-in simple type, the local name part of **type-name**, otherwise the local name of the nearest ancestor built-in simple type to **type-name**.

[datavalue]

> *ComplexElement:* Has "no value".

> *SimpleElement:* If **nilled** is *"true",* the special value *"nil",* otherwise the value of **typed-value** property.

[schema]

> Empty.

4. Text Node

Text Nodes have the following properties:

o **content**
o **parent**, possibly empty.

4.1. DFDL Infoset to XDM

If the value of a SimpleElement information item **[datavalue]** property converted to its canonical string representation is not the empty string then a Text Node must be constructed, otherwise no Text Node is constructed. Text Node properties are derived from the SimpleElement information item as follows.

content

> The value of **[datavalue]** property converted to its canonical lexical representation.

parent

> The Element Node constructed from the SimpleElement information item.

4.2. XDM to DFDL Infoset

When a SimpleElement information item is constructed, a Text Node is not used.

5. Namespace Node

Namespace Nodes have the following properties:

o **prefix,** possibly empty
o **uri**
o **parent**, possibly empty.

5.1. DFDL Infoset to XDM

If a SimpleElement or ComplexElement information item **[namespace]** property is not empty, then a Namespace Node must be constructed, otherwise no Namespace Node is constructed. Namespace Node properties are derived from the Element information item as follows.

prefix

> Empty. There are no prefixes for namespaces in DFDL.

uri

> The value of the **[namespace]** property.

parent

> The Element Node constructed from the Element information item.

5.2. XDM to DFDL Infoset

When a SimpleElement or ComplexElement information item is constructed, a Namespace Node is not used.

6. Authors and Contributors

Stephen M. Hanson,
IBM Software Group,
Hursley,
Winchester,UK
smh@uk.ibm.com

We greatly acknowledge the contributions made to this document by the following people.

Sandy Gao, IBM Software Group, Markham, Ontario, Canada

7. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

8. Disclaimer

This document and the information contained herein is provided on an "As Is" basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

9. Full Copyright Notice

10. References

[XDM] XML Data Model http://www.w3.org/TR/2010/REC-xpath-datamodel-20101214/

[DFDL] DFDL 1.0 http://www.ogf.org/documents/GFD.174.pdf/