TDML - Test Data Markup Language for DFDL

Status of This Document

Grid Working Document (GWD)

Document Change History

2014-04-11 Created.

2014-07-24 Updated with Bit Order Feature Material

Copyright Notice

Copyright © Open Grid Forum (2014). Some Rights Reserved. Distribution is unlimited.

Abstract

Implementation of DFDL (Data Format Description Language) is a test-intensive endeavor. The TDML language, an XML document format, allows creation of tightly specified test cases that can be used across implementations. The XML Schema for TDML is published as part of this document.

This document provides information to the Grid community about DFDL. It addresses testing needs using a language, Test Data Markup Language (TDML), for creating DFDL tests that is intended to operate in an implementation-independent manner. This document does not define any standards or technical recommendations.

Contents

[Abstract 1](#_Toc393986943)

[Contents 1](#_Toc393986944)

[1 Introduction 2](#_Toc393986945)

[2 Limitations 2](#_Toc393986946)

[3 XML Schema 2](#_Toc393986947)

[4 TDML Extension to support bitOrder 6](#_Toc393986948)

[5 References 9](#_Toc393986949)

[6 Security Considerations 10](#_Toc393986950)

[7 Glossary 10](#_Toc393986951)

[8 Contributors 10](#_Toc393986952)

[9 Acknowledgments 10](#_Toc393986953)

[10 Intellectual Property Statement 10](#_Toc393986954)

[11 Disclaimer 10](#_Toc393986955)

[12 Full Copyright Notice 10](#_Toc393986956)

[References 11](#_Toc393986957)

# Introduction

TDML was originated by IBM, which supplied an initial set of 74 tests for DFDL to the community. The Daffodil project has implemented TDML in the programming language Scala, and provided an XML Schema for what has become a Daffodil-specific dialect of TDML. The Daffodil project has thousands of TDML-based tests. IBM has its own dialect of TDML, and also has thousands of tests.

In the interest of convergence and future cross-validation of DFDL implementations, this document publishes a TDML schema.

TDML is intended to be an interchange format for DFDL tests. A TDML test suite can be completely self-contained within a single XML file (standard suffix ".tdml") including both the DFDL schema, the input data and expected result or expected errors/warnings (or both). This is very useful for supporting users who are trying to understand DFDL, and who may have found a bug in an implementation, or may have an incorrect DFDL Schema. A TDML test suite can be written which is almost tutorial in nature by showing basic tests of particular DFDL Schema capabilities, and then exploring variations on them within the same TDML file.

We expect that TDML will be enhanced further by the community. The schema provided here is a starting point.

# Limitations

Error messages - the current TDML allows creation of negative test cases, but the scrutiny given to negative results is limited to keyword search within the diagnostic output messages.

The Daffodil project implementation of TDML has a number of bugs/needed-features. These are tracked in the Project JIRA ticket system. [Daffodil]

# XML Schema

The schema is listed here by way of making its copyright status clear. This schema can also be downloaded from the Daffodil project git repository.

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:sub="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"

xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.ibm.com/xmlns/dfdl/testData"

xmlns:tns="http://www.ibm.com/xmlns/dfdl/testData" elementFormDefault="unqualified"

xmlns:dfdl="http://www.ogf.org/dfdl/dfdl-1.0/"

xmlns:daf="urn:ogf:dfdl:2013:imp:opensource.ncsa.illinois.edu:2012:ext">

<xsd:import namespace="http://www.w3.org/2001/XMLSchema" />

<xsd:import namespace="http://www.ogf.org/dfdl/dfdl-1.0/"/>

<xsd:import namespace="urn:ogf:dfdl:2013:imp:opensource.ncsa.illinois.edu:2012:ext"/>

<element name="testSuite">

<complexType>

<sequence>

<choice maxOccurs="unbounded" minOccurs="0">

<element ref="tns:parserTestCase" />

<!-- This is an extension to the IBM TDML language. We allow schemas

to be directly embedded inside the TDML file. A TDML file that contains all

the schemas it needs is a "self contained" TDML file. -->

<element ref="tns:defineSchema" minOccurs="0" />

<element ref="tns:defineConfig" minOccurs="0" />

<element ref="tns:serializerTestCase" />

</choice>

</sequence>

<attribute name="suiteName" type="xs:token" use="optional" />

<attribute name="ID" type="xs:token" use="optional" />

<attribute name="description" type="xs:string" use="optional" />

</complexType>

<unique name="unique-parserTestCase-name">

<selector xpath="parserTestCase" />

<field xpath="@name" />

</unique>

<unique name="unique-serializerTestCase-name">

<selector xpath="serializerTestCase" />

<field xpath="@name" />

</unique>

<unique name="unique-embeddedSchema-name">

<selector xpath="defineSchema" />

<field xpath="@name" />

</unique>

<unique name="unique-embeddedConfig-name">

<selector xpath="defineConfig" />

<field xpath="@name" />

</unique>

</element>

<!-- We want to allow an xsd:schema to be named and directly embedded in

the TDML thereby allowing a fully-self-contained single file test case as

an exchange medium for tests. -->

<element name="defineSchema" type="tns:defineSchemaType" />

<complexType name="defineSchemaType">

<choice maxOccurs='unbounded'>

<element ref='dfdl:format' />

<element ref='dfdl:defineVariable' />

<element ref='dfdl:defineEscapeScheme' />

<element ref='dfdl:defineFormat' />

<element ref='xs:element' />

<element ref='xs:complexType' />

<element ref='xs:group' />

<element ref='xs:simpleType' />

<element ref='xs:include' />

<element ref='xs:import' />

<!-- <any namespace="##other" processContents="lax"/> -->

</choice>

<attribute name="name" type="xs:NCName" use="required" />

</complexType>

<element name="defineConfig" type="tns:defineConfigType"/>

<complexType name="defineConfigType">

<sequence>

<element ref="daf:externalVariableBindings" minOccurs="0" maxOccurs="1"/>

</sequence>

<attribute name="name" type="xs:NCName" use="required" />

</complexType>

<element name="parserTestCase" type="tns:parserTestCaseType" />

<complexType name="parserTestCaseType">

<sequence>

<element ref="tns:document" />

<choice>

<element ref="tns:infoset" />

<element ref="tns:errors" />

</choice>

<element ref="tns:warnings" minOccurs='0' />

<element ref="tns:validationErrors" minOccurs="0" />

</sequence>

<attributeGroup ref="tns:testCaseAttribs" />

</complexType>

<attributeGroup name="testCaseAttribs">

<attribute name="name" type="xs:NCName" use="required" />

<attribute name="ID" type="xs:token" use="optional" />

<attribute name="root" type="xs:NCName" use="required" />

<attribute name="model" type="xs:string" use="optional" /> <!-- is there a type for a path/uri? -->

<attribute name="config" type="xs:string" use="optional" /> <!-- is there a type for a path/uri? -->

<attribute name="description" type="xs:string" use="optional" />

<attribute name="unsupported" type="xs:boolean" use="optional"

default="false" />

<attribute name="validation" type="tns:onOff" use="optional"

default="off" />

</attributeGroup>

<simpleType name="onOff">

<restriction base="xs:token">

<enumeration value="on" />

<enumeration value="limited" />

<enumeration value="off" />

</restriction>

</simpleType>

<element name="document" type="tns:documentType" />

<element name="infoset" type="tns:infosetType" />

<element name="errors" type="tns:errorsType" />

<element name="warnings" type="tns:warningsType" />

<element name="validationErrors" type="tns:validationErrorsType" />

<complexType name="documentType" mixed="true">

<sequence>

<element ref="tns:documentPart" minOccurs="0" maxOccurs="unbounded" />

</sequence>

</complexType>

<element name="documentPart" type="tns:documentPartType" />

<complexType name="documentPartType">

<simpleContent>

<extension base="xs:string">

<attribute name="type" type="tns:documentPartTypeEnum"

use="required" />

<attribute name="replaceDFDLEntities" type="xs:boolean" />

</extension>

</simpleContent>

</complexType>

<simpleType name="documentPartTypeEnum">

<restriction base="xs:string">

<enumeration value="byte" />

<enumeration value="text" />

<enumeration value="bits" />

<enumeration value="file" />

</restriction>

</simpleType>

<complexType name="infosetType">

<sequence>

<element ref="tns:dfdlInfoset" />

</sequence>

</complexType>

<element name="dfdlInfoset" type="tns:dfdlInfosetType" />

<complexType name="dfdlInfosetType" mixed="true">

<sequence>

<any namespace="##any" processContents="lax" minOccurs="0"

maxOccurs="unbounded" />

</sequence>

</complexType>

<complexType name="errorsType">

<sequence>

<element ref="tns:error" maxOccurs="unbounded" minOccurs="1" />

</sequence>

<attributeGroup ref="tns:errorWarnAttribs" />

</complexType>

<complexType name="warningsType">

<sequence>

<element ref="tns:warning" maxOccurs="unbounded" minOccurs="0" />

<!-- use <warnings/> to indicate no warnings should occur. -->

</sequence>

<attributeGroup ref="tns:errorWarnAttribs" />

</complexType>

<complexType name="validationErrorsType">

<sequence>

<element ref="tns:error" maxOccurs="unbounded" minOccurs="0" />

<!-- use <validationErrors/> to indicate no validation should occur. -->

</sequence>

<attributeGroup ref="tns:errorWarnAttribs" />

</complexType>

<element name="error" type="xs:string" />

<element name="warning" type="xs:string" />

<xs:attributeGroup name="errorWarnAttribs">

<xs:attribute name="match" use="optional" default="all">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="all" />

<xs:enumeration value="any" />

</xs:restriction>

</xs:simpleType>

</xs:attribute>

</xs:attributeGroup>

<!-- IBM chose to call unparsing 'serializing' in TDML -->

<element name="serializerTestCase" type="tns:serializerTestCaseType" />

<complexType name="serializerTestCaseType">

<sequence>

<element ref="tns:infoset" />

<element ref="tns:document" minOccurs="0" /> <!-- must have either document, or errors, or both -->

<element ref="tns:errors" minOccurs="0" />

<element ref="tns:warnings" minOccurs='0' />

</sequence>

</complexType>

</xsd:schema>

# TDML Extension to support bitOrder

The TDML test data markup language is extended to support testing of data that uses the dfdl:bitOrder property. Two new capabilities were added:

document parts with bytes written in right-to-left increasing order

document parts with least-significant-bit-first bit order.

The combination of these two capabilities allow direct translation of examples from the MIL-STD-2045 specification into TDML test cases.

For example, the TDML fragment below illustrates the first part of the example in Table B-I of the MIL-STD-2045 specification

|  |
| --- |
| <document bitOrder="LSBFirst">  <documentPart type="bits" byteOrder="RTL">Version XXXX 0011</documentPart>  <documentPart type="bits" byteOrder="RTL">FPI XXX0 XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL">GPI for Originator Address XX1X XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL">FPI for URN X1XX XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL">URN X0000000 00000000 01100111 1XXX XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL">FPI for Unit Name 1XXX XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL">Unit Name (UNITA) U X101 0101</documentPart>  <documentPart type="bits" byteOrder="RTL"> 0XXX XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL"> N XX10 0111</documentPart>  <documentPart type="bits" byteOrder="RTL"> 01XX XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL"> I XXX1 0010</documentPart>  <documentPart type="bits" byteOrder="RTL"> 100X XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL"> T XXXX 1010</documentPart>  <documentPart type="bits" byteOrder="RTL"> 0001 XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL"> A XXXX X100</documentPart>  <documentPart type="bits" byteOrder="RTL"> 1111 1XXX</documentPart>  <documentPart type="bits" byteOrder="RTL"> DEL XXXX XX11</documentPart>  <documentPart type="bits" byteOrder="RTL">GPI for Recip. Addr Group XXXX X1XX</documentPart>  <documentPart type="bits" byteOrder="RTL">GRI for R\_ONE XXXX 0XXX</documentPart>  <documentPart type="bits" byteOrder="RTL">FPI for URN XXX1 XXXX</documentPart>  <documentPart type="bits" byteOrder="RTL">URN XXXX00000 00000000 00000000 011X XXXX</documentPart>  </document> |

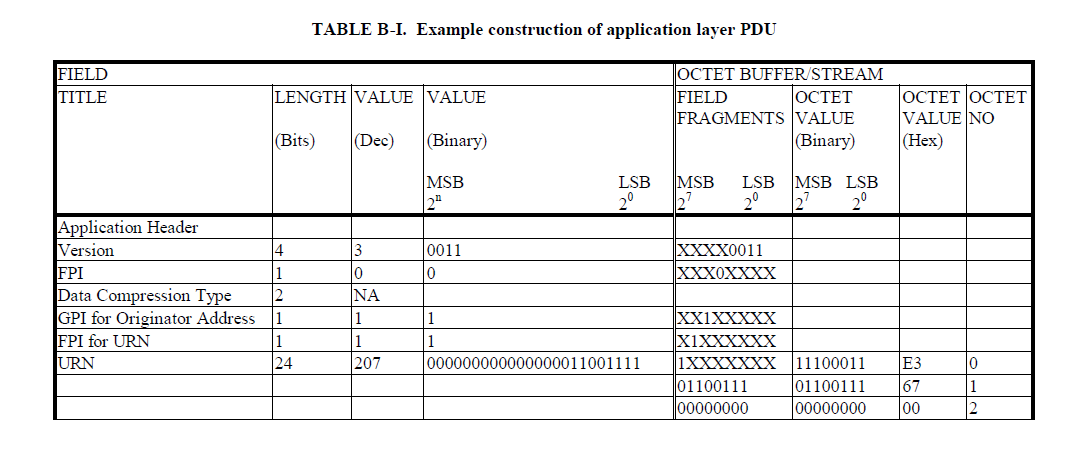
**Table : TDML for part of MIL-STD-2045 header. Specifies the bytes E3 67 00 80 55 67 92 1A FC 77 00 00 00.**

In the above, the tdml:document element has an attribute named 'bitOrder' which has values 'MSBFirst' (the default), or 'LSBFirst'. This attribute appearing on the tdml:document element is just a convenience. It is equivalent to specifying the same attribute on each tdml:documentPart element that does not have its own 'bitOrder' attribute.

The tdml:documentPart elements also each carry an attribute named 'byteOrder' with values 'LTR' (the default) meaning “left to right”, and 'RTL' meaning “right to left”. With value 'RTL' the data is interpreted as bytes numbered starting from the right.

This data example uses type='bits', and so any character other than 0 or 1 is ignored. Hence, we are able to conveniently insert labels (remembering to avoid characters 1 and 0 in the text) and we use X to represent a bit placeholder that will be supplied in a subsequent tdml:documentPart.

This same convention is used in the MIL-STD-2045 specification to illustrate example data:



The combination of properties of bitOrder 'LSBFirst' and byteOrder 'LTR' is disallowed.

The byteOrder attribute can also be used on tdml:documentPart elements of type 'byte'.

# References

|  |  |
| --- | --- |
| [Daffodil] | <https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Daffodil%3A+Open+Source+DFDL> |
| [DFDL] | Michael J Beckerle, Steven M Hanson, Alan W Powell. Data Format Description Language (DFDL) v1.0 Specification. Open Grid Forum. (<http://redmine.ogf.org/dmsf/dfdl-wg>)  Forthcoming Update: GFD-P-R.207 (2014)  Obsolete: GFD-P-R.174. January 2011. |
| [DFDLCharset] | Michael J Beckerle, DFDL-Specific Character Set Encodings, Open Grid Forum 2014  Forthcoming |
| [MILSTD2045] | CONNECTIONLESS DATA TRANSFER APPLICATION LAYER STANDARD, MIL-STD-2045-47001D w/CHANGE 1, 23 June 2008 (available publicly from US Dept. of Defense at <http://assistdocs.com/>) |

# Security Considerations

TDML does not address security issues directly, but correctness of a DFDL implementation is critical to system security of any system using DFDL when accessing data reading or writing. Some security considerations for DFDL are described in [DFDL].

# Glossary

DFDL - Data Format Description Language

TDML - Test Data Markup Language

# Contributors

Michael J. Beckerle (Corresponding Author)

Tresys Technology

Columbia, MD

USA

Email: mbeckerle@tresys.com

# Acknowledgments

We would like to acknowledge the contribution by IBM of an initial suite of 75 TDML test cases for use by the DFDL community.

# 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.

# 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.

# Full Copyright Notice

Copyright (C) Open Grid Forum (2014). Some Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included as references to the derived portions on all such copies and derivative works. The published OGF document from which such works are derived, however, may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing new or updated OGF documents in conformance with the procedures defined in the OGF Document Process, or as required to translate it into languages other than English. OGF, with the approval of its board, may remove this restriction for inclusion of OGF document content for the purpose of producing standards in cooperation with other international standards bodies.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

# References

[Daffodil] <https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Daffodil%3A+Open+Source+DFDL>

[DFDL] Michael J Beckerle, Steven M Hanson, Alan W Powell. Data Format Description Language (DFDL) v1.0 Specification. Open Grid Forum. (<http://redmine.ogf.org/dmsf/dfdl-wg>)

Forthcoming Update: GFD-P-R.207 (2014)

Obsolete: GFD-P-R.174. January 2011.