

DFDL WG Call Minutes

This OPEN document will not be filed. It is being kept active.

Meeting about Meetings\OGF

Project	DFDL 1.0
Meeting Date	29-Nov-18 (Thurs)
Meeting Time	16:00 - 17:00

Created by Steve Hanson on 09-Mar-11
Last Modified by Steve Hanson on 30-Nov-18

OGF DFDL Working Group Call, 29th November 2018

Agenda

Prepare for your meeting by describing the objectives (both immediate and long-term, if appropriate) of the meeting; and planning details.

1. Daffodil Open Source Project

Status update

2. encodingErrorPolicy 'error' behaviour when re-decoding data

The DFDL spec isn't clear on when encodingErrorPolicy 'error' is allowed to cause an error, and when one must be suppressed, if the implementation pre-decodes data into characters.

3. sequence terminator that exists or not depending on expression

To support strings that are terminated only when less than maxLength.

4. New GeoNames schema on DFDLSchemas GitHub

5. Daffodil team have found bugs in IBM DFDL samples

First bug is the SAX writer not handling '&' character correctly.
Second bug is the trace listener not working properly when building a grammar.
Third (possible) bug is schema validation not invoked when building a grammar.

6. AOB

Minutes

Meeting Minutes

Reflect on your meeting as you record all topics and issues discussed, and any tabled conversations. What went well, or what would you do differently next time? Document those so others can take advantage of your learning.

Attendees

Steve Hanson

Mike Beckerle

Apologies

Minutes

1. Daffodil Open Source Project

Aiming for next release by end of year.

2. encodingErrorPolicy 'error' behaviour when re-decoding data

To be discussed on extra call next week.

3. sequence terminator that exists or not depending on expression

To be discussed on extra call next week.

4. New GeoNames schema on DFDSLchemas GitHub

Steve concerned that the schema is effectively doing a parse, a transformation and a serialize all in a single parse, due to clever use of outputValueCalc etc. Suggested that there be two schemas, one that just parses the format (and ideally works with both implementations) and an advanced one that does the transform.

5. Daffodil team have found bugs in IBM DFDL samples

First bug is the SAX writer not handling '&' character correctly.

Second bug is the trace listener not working properly when building a grammar.

Third (possible) bug is schema validation not invoked when building a grammar.

IBM has raised internal issues to fix first and second.

Third needs more investigation, Mike to continue debugging.

IPR Statement

"I acknowledge that participation in this meeting is subject to the OGF Intellectual Property Policy."

Meeting closed

17:00 UK

Next regular call

13th December 2018 @ 16:00 UK but note **extra call on 6th December 2018 @ 16:00**

Create Action Items

Record the to-do's and individuals assigned by entering the appropriate information in the form below. Press the "Create Action Items" button to create specific to do's that can be tracked in the assignee's Work for Me views. " All Action Items will be tracked in the Action Items and Other Meeting Documents tab.

Action Items and Other Meeting Documents

Action Items and Other Meeting Documents

Subject	Document Type	Created	Modified

Next action: **308**

Actions raised at this meeting

No	Action

Current Actions:

No	Action
228	<p>Review set of tutorial lessons (All)</p> <p>17/9/13: Lesson 1 proposes a set of lessons, needs reviewing as over 2 years old. ...</p> <p>22/10: No progress</p> <p>31/10: Becoming a focus for Tresys. Steve to send his 'Modeling Data Formats using DFDL' powerpoint. ...</p> <p>19/11: No further progress</p> <p>26/11: Possibility of help from MITRE high-school student, and from Marisa at IBM. ...</p> <p>11/3/14: No further progress</p> <p>25/3: MITRE have produced a couple of new tutorials under the guidance of James Gariss. Jonathan to forward for review. Mike observed that an html tutorial could be generated from a tdml file using XSLT.</p> <p>11/4: Not discussed</p> <p>15/4: Jonathan will send 4 new mini-tutorials. Need to figure out best way to incorporate into the tutorial structure.</p> <p>29/4: Tutorials received. Mark has taken a quick read. Mark & Steve to review and report back.</p> <p>6/5: Still with Mark and Steve</p> <p>20/5: Mark has reviewed. Will ask IBM information development to recommend a</p>

way to portray the existing and new lessons, preferably web-based. Find somewhere to host them. OGF? GitHub? developerWorks? NCSA?
3/6: Steve has also reviewed.

...

17/6: No further progress on tutorials. Tim is looking into the creation of some DFDL how-to videos using the IBM Integration Studio.

...

31/3/15: No further progress

14/4: Agreed that the need for better tutorials has become pressing for Daffodil users who aren't using IBM's tools and material. Discussed creating tutorials based on a tdml file with comments that is processed to produce html. Mike to investigate.

28/4: Mike has sent an example tdml file which embeds instances of a new 'tutorial' element in various places. These elements contain html which can be extracted and formatted in a browser. Suggest future DFDL tutorials are created using this technology.

12/5: Not discussed

...

22/9: No further progress

3/11: Daffodil team has someone working on the new 'tutorial' element in tdml files. In time this should result in some new tutorials and re-working of existing tutorials.

5/1/16: Mike has started a bitOrder tutorial using the tdml file approach (uses stylesheets to render html).

16/2: The bitOrder tutorial is available on the web @

<https://opensource.ncsa.illinois.edu/bamboo/artifact/DFDL-MASTER21/JOB1/build-132/Tutorials/bitorder.tutorial.tdml.xml>

1/3: Awaiting review.

...

17/4/18: No further progress

15/5: Daffodil sponsor has requested some progress on tutorials as it will soon become the limiting factor on uptake.

...

29/11: No further progress

242 **Standardise on a single tdml format for DFDL tests (All)**

5/2/14: Steve has requested permission for IBM to view / use the Daffodil tdml files, as a precursor to trying to standardise on a common tdml format. Was formerly part of action 066.

...

18/2: No further progress

11/3: Mike and Steve discussing the best way to share and cooperate on tdml format.

25/3: Discussed the creation of an OGF document that will own and define a standardised tdml format.

11/4: Proposal is for the OGF document to define a tdml format without Tresys or IBM copyright statement.

15/4: Draft document on Redmine

...

6/5: No further progress

20/5: Mark has read through the document. Particularly concerned with how namespaces are handled in the infoset.

...

17/6: No further progress

25/6: Mike has added bit order capability as per action 233.

...

9/12: No further progress

6/1/15: Mike to resurrect this as Tresys would like to run their tdml suite against both Daffodil and IBM DFDL.

...

10/2: No further progress

24/2: Mike updating the Daffodil TDML test runner to handle unparser (ie, serializer) tests

...

14/4: No further progress

28/4: Tresys have enhanced their tdml runner to allow unparser tests and round-trip tests (parser->unparser->parser) as well as the new tutorial tag (see action 228)

12/5: Not discussed

...

3/11: No progress

5/1/16: No progress. Needs more interoperability between implementations to be really useful.

...

25/7/17: No further progress

3/10: No further progress although forthcoming work to add packed/zoned numbers may force progress

...

11/12: Expected to look at this in the next month or so

...

4/9/18: No further progress

16/10: Mike has started work on a TDML runner that can drive a pluggable DFDL implementation, in support of interoperability testing, including IBM DFDL.

1/11: Pluggable TDML runner working. On Github at <https://github.com/OpenDFDL/ibmDFDLCrossTester>. Schema resolution for IBM DFDL achieved using its schema resolver feature and pointing it at Daffodil's resolver. IBM DFDL sample uses mark() on its input stream but IBM believes this is not necessary.

15/11: IBM DFDL and Daffodil have dependencies on different releases of ICU. Forcing changes to the TDML runner to isolate the implementations under test.

29/11: Good progress on the TDML runner, see email from Mike. The [ibmCrossTestRig](#) is not part of Daffodil (because it links against IBM DFDL), but is open source Apache License v2, and is currently in review at <https://github.com/OpenDFDL/ibmDFDLCrossTester/pull/1>. Steve needs to talk to IBM legal to check this is ok as it currently modifies IBM DFDL sample code.

250 Public comment: dfdl:valueLength and dfdl:contentLength descriptions (Mike)

19/11/14: <http://redmine.ogf.org/boards/15/topics/63>. Agreed that the function names were ok as per errata 3.18, and that the spec is clear that they refer to the grammar regions. However the grammar regions mentioned do not fully include literal nil values. Discussed what happens when parsing - remember the length or re-parse? What about lengthUnits 'characters' when the data is binary? Also the 'Notes' that follow the table need to be reworked.

26/11: Needs wording to handle all the issues found, assigned to Mike.

...

11/3/15: Still with Mike

25/3: Mike has sent out revised wording, reviewed by Mark and Steve. Noted that the words need to explain the concept of building a complex element from the bottom up, and these words are equally applicable to several places in section 12.3. Mike to revise accordingly.

11/4: More revised wording sent by Mike. Started to review but realised it needed some off-line preparation and thought. Review for next call.

15/4: Review comments from Steve and Tim. The functions need to be clear that they work off the infoset value. The detailed wording is needed but should be removed to a new sub-section of 12.3, probably at end. Most sub-sections of 12.3, and the functions in 23.5.3 will refer to this new sub-section. 23.5.3 should limit

itself to behaviour specific to the functions, such as not potentially represented, the effect of the \$lengthUnits argument. Also discussed what happens if \$path argument returns a nodeset > 1; should be a processing error, can always use a predicate to select one node of an array.

29/4: See various email discussions. Several things noted by Mike, and he recommends a rewrite of some of section 12.3. Then the description of the two functions becomes much simpler. Deferring for now, and will resurrect after current spec revision is finalised.

6/5: Mike is working on a mind map for the length section. Deferring until needed.

....

23/9: Rewrite should be postponed to future 1.1. Still need to answer the original questions about the functions though...

...

25/4/2016: Undeferring action as some of these issues are now impacting Daffodil team as they write their unparser. Steve has sent the email threads on this action to Mike. Mike will combine with his issues and distill into a single thread.

7/6: Thread to include use of a variable with dfdl:outputValueCalc. May be undefined at point of evaluation.

5/7: Mike has been looking at this. Two main points:

1) Computing the content length of a complex element with internal alignment.

2) Computing length in chars of a complex element which is not 100% text

Mike will send out a discussion via email. Noted that rules should also apply to prefixed length calculation.

(Aside: IBM DFDL unparser does not support a prefixed length complex element with length units chars and variable width encoding).

2/8: Mike has sent out several emails.

#1: Proposes that term ComplexValue is added to the grammar to better handle ElementUnused. Knock-on effect on the wording of dfdl:valueLength(). Agreed on the change. **Issue <https://redmine.ogf.org/issues/316> created.**

Spec says escapeCharacter, escapeBlockStart, escapeBlockEnd, escapeEscapeCharacter contribute to the content length of an element. This is not correct, they are part of the value length. **Issue <https://redmine.ogf.org/issues/317> created.**

#2: Gives some options for computing DFDL length functions when target complex element has interior alignment. Agreed that the DFDL processor should detect this and give runtime SDE. This is an example of expression forward reference deadlock. Need new paragraph in section 23 of spec to cover this. **Issue <https://redmine.ogf.org/issues/318> created.**

#3: Argues that DFDL should only encode/decode when it needs to when computing DFDL length functions, to allow for performance. Agreed that this behaviour was ok. **Issue <https://redmine.ogf.org/issues/319> created.**

13/9: Review issues created. Need to come up with the revised descriptions for dfdl:valueLength and dfdl:contentLength functions before action can be closed.

...

[29/11/18: No further progress](#)

279 Improve defaulting description to explicitly cover local groups (Mike)

28/4/15: Only talks about elements, should mention local sequence and choice.

12/5: Not discussed

23/6: Section 15.1.3 needs to say what happens when a choice branch does not contain any elements; such a choice branch is selected (but see action 280 below as minOccurs '0' might change this). Section 9.4 also needs updating to say what happens when local groups are found within a complex type.

11/8: Steve did some tests with IBM DFDL. Just need some words as above.

Action assigned to Mike.

25/8: In progress

...

	<p>5/1/16: No progress</p> <p>...</p> <p>29/11/18: No further progress</p>
287	<p>Find a way to handle a variable path step in DFDL expression (All)</p> <p>1/3/16: DFDL4S currently using a hack that embeds a regex in a path step.</p> <p>10/5: No progress</p> <p>24/5: Need example from DFDL4S</p> <p>..</p> <p>5/7: Need to ask DFDL4S for example.</p> <p>2/8: DFDL4S sent example. They use dfdl:contentLength() with a path that has a step that contains a regex as a wildcard. Mike has requested the wider set of schemas to be sent, in order to see if there is a viable alternative.</p> <p>13/9: Response received from DFDL4S, not yet analysed.</p> <p>...</p> <p>10/1/17: No further progress</p> <p>7/2: Mike has analysed the schemas and sent a comprehensive reply to DFDL4S. He believes that the variable path step is effectively a way of parameterizing the expression, and has described how this can be done using DFDL variables. DFDL4S have responded and will talk to the contractor that authored the implementation.</p> <p>21/2: No response so far from DFDL4S.</p> <p>4/4: Mike has seen a further example of this. Still no response from DFDL4S.</p> <p>...</p> <p>25/7: No further progress</p> <p>3/10: Mike has seen a further example where an expression needed to look back inside an earlier choice, where there was a common element. Discussed whether XPath 2.0 wildcards could be used (currently not supported in DFDL 1.0). This looks to be a good fit, and would involve only a minimal change to the supported syntax. Steve will email DFDL4S.</p> <p>...</p> <p>11/12: ESA will look into this as part of the next round of changes to DFDL4S.</p> <p>...</p> <p>17/4/18: No update</p> <p>15/5: Steve has emailed DFDL4S asking for a progress update. Also, Mike will put together a concrete proposal.</p> <p>...</p> <p>4/9: No further progress</p> <p>16/10: Steve will send chaser email to DFDL4S team.</p> <p>1/11: Marcus Bento from DFDL4S team responded: "I confirm that Action 287, related to using regex as part of the path in our schemas, is expected to be addressed in the early release of 2019. I've investigated the current implementation, and believe that your suggested approach (based on XPath 2.0) is sufficient for our needs. However, only after the release of December will the contractor analyze the issue further and confirm that the implementation works."</p> <p>...</p> <p>29/11: No progress likely until January at the earliest.</p>
289	<p>Unparsing: expression refers backwards to outputValueCalc which refers beyond it.</p> <p>2/8/16: Need to decide if this is allowed and if so if there are any restrictions.</p> <p>13/9: Motivating scenario is where a variable is being set to a length element using dfdl:setVariable, which on unparse is set using dfdl:outputValueCalc. So although the variable is referring backwards to the length element, it is effectively forward referencing so must block. Mike believes this is unavoidable.</p> <p>11/10: Daffodil has implemented this, Mike to provide scenario.</p> <p>8/11: Mike couldn't find example, will continue to look</p> <p>10/1/17: Mike has realised that all the examples were reworked to avoid using</p>

	<p>variables, hence why can't be found. 7/2: Daffodil will soon be implementing dfdl:newVariableInstance which will bring this up again. ... 17/4/18: Waiting for Daffodil to implement dfdl:newVariableInstance 15/5: Daffodil team have supplied an example of this from the PCAP schema. Likely to require a flag on newVariableInstance (or maybe variable declaration) to indicate whether needed on parse, unparse or both. ... 29/11: No further progress</p>
292	<p>Write up proposal for allowing hexBinary elements to have dfdl:lengthUnits='bits' (Mike) 7/2/17: Mike will create a proposal for evaluation. 21/2: No progress 4/4: Daffodil has experimental implementation, will be evaluated and written up. ... 15/5/18: Daffodil to write up. 7/8: Reviewed by Steve. 4/9: No further progress ... 15/11: Mike to revise after Steve's comments 29/11: Mike has sent out a second proposal due to problems with the first, but Steve not comfortable with the second proposal as hexBinary starts to get integer properties, even if only when a new property is set. Mike will take this back to drawing board. Agreed that bitOrder does come into play here.</p>
293	<p>Investigate solutions to enabling choices in hidden groups to be unparsed (All) 7/2/17: Study of problem needed in order to best evaluate any proposals. 21/2: Mike has circulated a proposal internally within Daffodil. 4/4: No progress but immediate need has gone away. On hold for now. ... 17/4/18: On hold. 15/5: Daffodil now looking at this and will write up a proposal. Potential commonality with action 289. ... 29/11: No further progress</p>
294	<p>Converting integer enumerations to meaningful strings in infoset (Mike) 18/4/17: Requirement from Daffodil user for parser to convert an integer enum to a meaningful string value in infoset. Daffodil has put forward a proposal but it relies on [unionMemberSchema] which is a validation-only property. https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Enumerations+and+Range+Tables+via+Simple+Type+Unions Mike to re-think the approach, and also consider whether this kind of transformation is really a post-DFDL step. Steve to check how XQuery would approach the same problem. ... 25/7: No progress 3/10: Also received same request from a product team at IBM. 21/11: Consider whether any additional annotation is not DFDL, for possible wider applicability. 11/12: On Daffodil priority list to investigate ... 6/3/18: No progress 17/4/18: Latest proposal is at https://cwiki.apache.org/confluence/display/DAFFODIL/Proposal%3A+Features+to</p>

	<p>+Support+Table-Lookup. Feedback requested. Review for next call. 15/5/18: Review for next call. 7/8: No further progress 4/9: Steve to review for next call 16/10: Steve thinks there is a major problem with the carrying of the rep properties, as the types are different. Mike to respond to Steve's comments. Need something that is a cross between IVC / OVC and prefixLengthType! 1/11: Mike to think some more. Need to be exact as to which properties can no longer be carried on the element / string type. Steve wondered if the mechanism could be generic, allowing any type to appear as a string in the infoset via a default toString style mapping. ... 29/11: No further progress</p>
<p>301</p>	<p>How to indicate DFDL v2 in schemas (All) 6/2/18: Agree on best way to do this. The DFDL namespace was originally intended to be used for this, but XSDL for example uses a separate 'XMLSchema-versioning' namespace and min/max attributes which allows schemas to be authored that may be processed by both XSDL 1.0 and XSDL 1.1. More investigation required. ... 17/4/18: No progress 15/5/18: Steve to familiarise himself with how XSDL does this. 7/8: No further progress 4/9: Need concrete statement of the requirements to be solved by a versioning mechanism. ... 29/11: No further progress</p>
<p>303</p>	<p>Investigate why the concept of 'potentially trailing group' is needed when suppressing separators (Steve) 17/4/2018: Steve thinks that this coped with a particular scenario but needs to do some research. 15/5/18: Steve has a scenario where this is useful (using a choice of simple types in CSV). Problem then is that sections 14.2.2 & 14.2.3 only talk about elements. Perhaps a 'source-to-source' transformation could be added that converted potentially trailing groups to potentially trailing elements, so that 14.2.2 & 14.2.3 required no updates? 7/8: No further progress 4/9: Steve believes this may have been to handle the TLog format, where xs:groups were added to records to define commonly used groups of fields, without adding an extra level to the infoset, while retaining suppression of optional elements. At least one anomaly remains, however, which is use of anyEmpty with a zero-length sequence - this does not have its separator suppressed. Steve to test whether IBM DFDL has this anomaly. This is kind of allowing optional sequences by the back door; maybe DFDL v2 should allow (0,1) for a sequence that is not the content model of a complex type? 16/10: Tests needed using IBM DFDL to see what it does with empty sequences and potentially trailing groups 1/11: Steve thought of two more examples why potentially trailing group is needed. 1) A column in a CSV-style record could be either string or integer; modelled as a choice of a string element and an integer element. If the column is optional and empty, this should not break suppression of the record's separators. 2) EDI segments can contain repeating fields which use a different 'repeat' separator; modelled as a sequence to carry the repeat separator with a single child repeating element. If the repeating field is optional and empty, this should not break suppression of the segment's separators. Steve to send example schemas. 15/11: Steve has sent schema showing example of 1) and 2) and ran some tests. Turns out that 1) is not potentially trailing, unless extra sequences added around</p>

	<p>the branches, in which it case it is. 2) is potentially trailing. Add syntax to the sequence and it is no longer potentially trailing. 2) is used in all the IBM EDIFACT and X12 schemas, for example. This is behaving according to the spec, and crucially when it is not actually trailing a separator is output for the sequence to keep position in the parent sequence. So back to the first bullet of this action - sections 14.2.2 & 14.2.3 should be updated to include potentially trailing groups. This does not help Mike's scenario though, where he would like an empty, syntax free sequence containing just an assert or discriminator NOT to cause a separator to be output, in the same way that an inputValueCalc element does not cause a separator to be output. This is not possible today. To handle this in DFDL 1.0 we would need an errata to add a new property to a sequence to say it was for logical purposes only.</p> <p>29/11: Issue raised in Redmine to cover the updates to section 14.2 - https://redmine.ogf.org/issues/338. Mike to think more on whether his requirement for empty, syntax free sequence is essential for DFDL 1.0.</p>
304	<p>Proposal for Data Streaming for layered transforms (All)</p> <p>17/4/2018: Daffodil team to provide feedback on the proposal as prototype gets implemented. Others to review.</p> <p>15/5/18: Mike to send link to updated proposal, which is working fine in prototype.</p> <p>7/8: Reviewed by Steve</p> <p>4/9: Steve to revisit the proposal, one idea was to group the properties into a new DFDL annotation.</p> <p>16/10: Discussed pros and cons of having a separate annotation. Mike to decide which way to jump based on real examples.</p> <p>1/11: No further progress</p> <p>15/11: Not discussed</p> <p>29/11: No further progress</p>
305	<p>Create tracker issues in Redmine for spec clarifications from 7th August (Mike)</p> <p>7/8: Several clarifications to text in section 9 and property descriptions in sections 12, 13, 14.</p> <p>4/9: Mike to do this</p> <p>16/10: No progress</p> <p>1/11: Mike will create the issues as tests are created. OGF switching to GitHub so that is likely the way forward for trackers.</p> <p>15/11: Not discussed</p> <p>29/11: No further progress, Mike wants to do this before year end.</p>
306	<p>Confirm IBM DFDL behaviour when parsing empty strings (Steve)</p> <p>7/8: IBM DFDL has not fully implemented the behaviour changes arising from action 140 with respect to empty string elements. Daffodil is about to do so. IBM DFDL users have complained about lack of defaults when parsing but other than that appear happy. Are the rules in the spec for empty strings over complicated? Steve to document the behaviour for IBM DFDL to inform the discussion.</p> <p>...</p> <p>1/11: In progress - there are a lot of subtle scenarios</p> <p>15/11: Not discussed</p> <p>29/11: No further progress</p>
307	<p>Demonstrate implementation interoperability (Steve, Mike)</p> <p>4/9: Need to make sure that DFDL spec section 21 lists a correct set of optional features, the implication being that Daffodil and IBM DFDL (and any other minimally conforming implementation) correctly implement the remaining required features. First step - see if there are any obvious omissions.</p> <p>16/10: Steve sent email stating IBM DFDL's missing core features and non-compliant behaviour, and Mike responded. Discussion continuing via two separate email threads. Part 1 for core features. Part 2 for optional features. For</p>

the core features, agreed that the following needs to happen:

- 1) IBM adds encodingErrorPolicy='replace'
 - 2) Daffodil adds encodingErrorPolicy='error'
 - 3) Daffodil ensures that, if not implementing default/fixed when parsing, it gives an SDE if a required occurrence has empty rep and element has default/fixed set.
 - 4) A position is agreed on BOM handling - ongoing via email.
- 1/11: Just BOM to conclude on from the above list
15/11: Not discussed
29/11: No further progress.

Closed actions

No	Action

Deferred actions

No	Action
241	Public comment: Bi-di properties placement in precedence section (All) 7/11: This looks deliberate but the asymmetry between parsing and unparsing is unclear. Really needs Daffodil or IBM DFDL to implement these properties, which has not happened yet. Deferring this action. ... 23/9: Candidate to be moved out to 1.1 ?

Work items:

No	Item	Owner	Target	Status