DFDL WG Call Agenda

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

Meeting about Meetings\OGF

Project	**DFDL 1.0**
Meeting Date	**11-Nov-21 (Thurs)**
Meeting Time	**16:00 - 17:00**

Created by Steve Hanson on 09-Mar-11
Last Modified by Steve Hanson on 11-Nov-21

OGF DFDL Working Group Call, 11th November 2021

Agenda

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

1. Implementations Update

2. DFDL as an ISO standard

3. 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

Apologies

Minutes

1. Implementations Update

2. DFDL as an ISO standard

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

Meeting closed

Next regular call

9th December 2021 @ 16:00 UK

Action Items and Other Meeting Documents

Subject	Document Type	Created	Mod

Next action: **325**

Actions raised at this meeting

No	Action

Current Actions:

No	Action
289	**Unparsing: expression refers backwards to outputValueCalc which refers beyond it (Mike).** 2/8/16: Need to decide if this is allowed and if so if there are any restrictions. 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. 11/10: Daffodil has implemented this, Mike to provide scenario. 8/11: Mike couldn't find example, will continue to look 10/1/17: Mike has realised that all the examples were reworked to avoid using

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.

...

17/10/19: No further progress

12/12: Daffodil starting to implement dfdl:newVariableInstance

...

16/4/20: No further progress

30/4: Daffodil has implemented newVariableInstance so now in a position to look at this again.

9/7: No update

23/7: Daffodil has implemented newVariableInstance so Mike will look at this again

3/9: Daffodil team looking at this and related issues surrounding expression evaluation when unparsing.

...

29/10: No further progress

12/11: Mike has a scenario in the PCAP schema that uses several outputValueCalc. To simplify the expressions, he would like to use a variable with a value set by newVariableInstance. The problem is that the rules for variables are different from those for outputValueCalc, specifically variables can not refer forwards (because they are evaluated on both parse and unparse) whereas that is the point of outputValueCalc. To ease this would require a change to the spec. WG agreed this is not necessary for DFDL 1.0 but a candidate for 2.0.

10/12: Daffodil to implement as as an experimental feature. Added to table below. GitHub issue https://github.com/OpenGridForum/DFDL/issues/22.

7/1/21: Daffodil have started prototype work on this.

18/2: Work continuing, but it looks like forward reference variables are needed as anticipated.

18/3: Daffodil has implementation of this nearly complete, which Mike will walk through on the next call.

15/4: Mike walked through examples of dfdl:newVariableInstance which refer forward and use 'dfdlx:direction=unparseOnly', using the PCAP schema. Will be part of the experience document.

8/7: Material assembled for experience document.

14/10: Mike to write this up

| 309 | **Create example scenarios to illustrate offset & pointer requirements (Bradd)**

5/4/19: Daffodil have a draft proposal for offset support, TPF have experimental implementation for pointer support. Need examples to show the requirement, especially unparsing.

2/5: Bradd supplied an example of pointers. On parsing the pointer is used as an absolute address to a piece of accessible memory, and the element is parsed from that location. On unparsing memory is allocated and unparsing of the element occurs into that location and the pointer set to the location (memory allocation is implementation-defined). Note the pointer value does *not* appear in the infoset. Looks like a useful and workable addition to DFDL. Could solve the parsing requirements for TIFF image files. Bradd also has extension for offset, which is like pointer but uses relative location instead of absolute. Both are examples of indirection. A further example could be specifying a file to read. Contrast this with what DFDL has used the term 'offset' for in the past, namely as an alternative property to alignment/skip which allows the parser/unparser to jump directly to a |

point in the current buffer. These are orthogonal concepts. Noted that parsing of ZIP files may need both. Secure implementations may need to disallow use of pointers and/or offsets unless they can guarantee to fill everywhere with the fill byte. Implementations should also be deterministic. Agreed that recursion not needed to implement this. Bradd mentioned a further concept 'overflows', an example being an array unparsed into a linked list. Pointers proposal needs to be written up as an experimental feature.

31/5: Bradd to write up pointers proposal as an experimental feature.

...

11/7: No update

8/8: Bradd aiming to get this written up for next time. Also needs issue tracker raising.

29/8: Bradd unable to make the call.

17/10: Written up for review and sent to WG but not as an experimental feature document. Mike also noted http://www.binarydom.com/sdk/doc/bddl.shtml. Mike has reviewed and commented on the write-up, Steve needs to do the same, then send back to Bradd. Main discussion was around unparsing, eg, buffering implications, whether to try and format exactly or canonically. In parallel, Bradd to create an experimental feature document (see table below).

12/12: Bradd sent an updated document. WG will review for next meeting. There was some discussion about the use in the document of the term 'empty' and whether that really meant 'missing'. This led to an in-depth discussion about the different use cases for default values, it is likely that DFDL 2.0 will introduce support for some of these, specifically:

- Item exists in Infoset with default value, so unparse empty rep (the mirror of parsing, as practised by GPB)
- Item exists in data with default value, so remove from Infoset post-validation (the mirror of unparsing, and a requirement from z/TPF who have a post-parse option to do this)

9/1/20: No progress, still needs reviewing.

16/4: Steve & Mike to review latest document dated 2019-12-12 for next call.

30/4: Spent some time discussing Steve's review comments. Conclusion is that the feature is useful and a serious candidate for DFDL 2.0. The properties seem to be the minimum needed to handle the concepts and known use cases. As this is currently an experimental feature we don't have to get it 100% precise now, and can impose restrictions that z/TPF users would be ok with (for example, no initiators or terminators allowed; binary indirection types only). Important though that the properties and their application is driven by the grammar, so next step is for Bradd to see how the grammar is affected. It would be nice if all the behaviour could be handled at the same point in the grammar as 'prefixLength' but that might not be possible. Property name 'indirectionEmptyValue' probably needs a better name, eg 'indirectionUnusedRep'.... other suggestions welcome. Or perhaps the dfdl:fillByte of the indirectionType could be used?

9/7: Bradd has sent an updated document which includes a grammar section. The grammar changes need reviewing by Steve. Property 'indirectionEmptyValue' changed to 'indirectMissingValue', but missing has other connotations. Mike not keen on using fillByte.

23/7: Steve reworked the grammar changes, introducing the concept of a repeating IndirectElement as a sibling of DocumentElement, being the target of an indirection, and using 'indirectionUnusedValue' to dis-ambiguate from nil/empty/missing. Steve explained the concept, Mike and Bradd to review with that in mind. Agreed that the indirection type could have prefixed length.

3/9: Agreed that IndirectElement is useful. Mike tried out the proposed properties to model TIFF format but ran into problems with recursion, as the indirection is really a linked list. Not possible to treat the recursive list as an array. Would also require 'backward' offsets. z/TPF have not implemented offsets but think only 'forward'

	offsets needed for their use cases. TIFF also needs property indirectionBase to be absolute, which is something that can be added to proposal. Bradd to update. (Also discussed 'layering' which is a related subject but was deferred a long time ago). 17/9: Bradd has sent updated document. Need to use correct terminology for 'underlying elements' and 'prior element', and update grammar. Once that done, the proposal looks to be in good shape. Issue tracker https://redmine.ogf.org/issues/363 raised and noted in table below. Next step is for Bradd to create an official OGF experience document. As TIFF could not be modelled, should see if offsets can be used to model zip files, and if possible find other offset examples. ... 29/10: Waiting for Bradd to create experience document. 12/11: Bradd has updated the document to define "underlying elements" and "prior element", and updated the grammar to add indirection as per Steve's proposal. Final step is to create an official OGF experience document. 10/12: No update. GitHub issue is https://github.com/OpenGridForum/DFDL/issues/17. 7/1/21: Not discussed. 18/2: Bradd has a draft experience document which he will add to the GitHub repo. 18/3: Now in GitHub at https://github.com/OpenGridForum/DFDL/blob/master/docs/working-drafts/gwde-dfdl-experience-8-v0.1-experimental-indirection.docx. Bradd will share a real-world worked example on the next call. Steve to think about changing folder structure in GitHub to group experience documents together. 15/4: Bradd not available, postponed 8/7: Bradd walked through example where REST API with JSON payload is converted to internal C mapped memory block, with items of variable length ending up as discontiguous blocks addressed by pointers from the original block, all modelled by DFDL schema. Also showed how the discontiguous memory is converted to a single contiguous block by replacing pointers with offsets and tacking the data onto the end. Noted that the implementation uses the original z/TPF-specific 'tddt' annotations. not 'dfdlx'. Bradd to see if any updates needed to experience document, then action can be closed. 14/10: No update.
321	**Assemble candidate list for DFDL 2.0 (All)** 18/3/21: Discussed some of the candidates (roll in experimental features, tidy up property inconsistencies). Need to create a full list of candidates. ... 14/10: No progress
322	**DFDL 2.0 candidate: Handle embedded XML and JSON in a natural way (All)** 8/7/2021: This was something that was included in the first DFDL 1.0 spec drafts but dropped. 14/10: Mike has created GitHub issue https://github.com/OpenGridForum/DFDL/issues/27.
323	**Erratum: SDE if in-scope delimiter value clashes with an escapeCharacter or escapeEscapeCharacter value (All)** 8/7/2021: Erratum needed to state this. IBM DFDL throws SDE. 14/10: Mike has created GitHub issue https://github.com/OpenGridForum/DFDL/issues/25.
324	**Erratum: Clarify semantic of array element assert containing subexpression " .[1] " (All)** 8/7/2021: Erratum needed to state this is equivalent to "../a[1]" and "../a[dfdl:occursIndex()]" 14/10: Mike has created GitHub issue https://github.com/OpenGridForum/DFDL/issues/26.

Closed actions

No	Action

Deferred actions

No	Action
250	**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 https://redmine.ogf.org/issues/339 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. 10/1/19: Daffodil have removed the modified IBM DFDL sample code. Steve to ask whether IBM can donate tests from the existing IBM DFDL test suite. 7/2: If permitted, the tests from IBM can be used to see how the IBM and Daffodil tdmls have diverged. 5/4: IBM are permitted to send Daffodil some example tdmls. Steve to send some to Mike for next call. ... 31/5: No progress 27/6: Steve has sent example TDML files to Mike, under the existing IBM Grant of Copyright agreement with Apache. 11/7: Daffodil to adapt their test runner so that it works with the IBM TDML file variant. Action can be closed when this is done. ... 29/8: No further progress 17/10: Noted that this action had its number accidentally swapped with 242 in October 2016. Renumbered back to 250. ... 23/7/20: No further progress 3/9: Nothing likely to happen for this action in medium term, so moving to deferred actions
316	**Proposed new experimental feature 'User defined functions' (Mike)** 9/1/20: Version 2.5.0. of Daffodil will include this, needs writing up. Does XPath 2.0 provide for this? 16/4: Nothing in core XPath 2.0 that allows user-defined functions, that is left to XSL etc. A write-up is here: https://cwiki.apache.org/confluence/display/DAFFODIL/Proposal%3A+Feature+to+Support+User+Defined+Functions. Mike has an example where values need to be normalised for later comparison. Bradd has an example where floating point precision needs adjustment. Steve is concerned that DFDL is straying too far into the transformation layer, maybe "user-defined-functions-for-DFDL" should be a separate spec? ... 3/9: No further progress. 17/9: Nothing likely to happen for this action in medium term, so moving to deferred actions.

Work items:

No	Item	Owner	Target	Status

Experimental Features:

N	Title	Act	GitHub Tracker	Impleme	Experience Document	Release

o		ion		ntor		
1	Experimental feature mechanism	301	N/A	Apache Daffodil	https://github.com/OpenGridForum/DFDL/blob/master/docs/current/gwdi-experimental-features-conventions.docx	1.0
2	Elements of type 'xs:hexBinary' with lengthUnits 'bits'	292	https://github.com/OpenGridForum/DFDL/issues/12	Apache Daffodil		2.0 candidate
3	Variable Path Step in DFDL expression	287	https://github.com/OpenGridForum/DFDL/issues/13	ESA DFDL4S		2.0 candidate
4	BLOBs as URIs in infoset	312	https://github.com/OpenGridForum/DFDL/issues/15	Apache Daffodil		2.0 candidate
5	Data streaming layers	304	https://github.com/OpenGridForum/DFDL/issues/11	Apache Daffodil		2.0 candidate
6	Interoperability	307	N/A	Apache Daffodil	https://github.com/OpenGridForum/DFDL/blob/master/docs/current/gwde-dfdl-experience-6-v0.1-interoperability.docx	1.0
7	Empty element parse policy	306	N/A	Apache Daffodil	https://github.com/OpenGridForum/DFDL/blob/master/docs/current/gwde-dfdl-experience-7-emptyElementParsePolicy.docx	1.0
8	Integer enums as strings in infoset	294	https://github.com/OpenGridForum/DFDL/issues/14	Apache Daffodil		2.0 candidate
10	Indirection using pointers	309	https://github.com/OpenGridForum/DFDL/issues/17	z/TPF DFDL	https://github.com/OpenGridForum/DFDL/blob/master/docs/working-drafts/gwde-dfdl-experience-8-v0.1-experimental-indirection.docx	2.0 candidate
11	User-defined functions	316	https://github.com/OpenGridForum/DFDL/issues/21	Apache Daffodil		2.0 candidate
12	Unparser forward reference variables	289	https://github.com/OpenGridForum/DFDL/issues/22	Apache Daffodil		2.0 candidate