DFDL WG Call Minutes

Meeting about Meetings\OGF

Project	**DFDL 1.0**
Meeting Date	**03-Sep-20 (Thurs)**
Meeting Time	**16:00 - 17:00**

Created by Steve Hanson on 09-Mar-11
Last Modified by Steve Hanson on 15-Sep-20

OGF DFDL Working Group Call, 3rd Sept 2020

Agenda

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

1. Implementations Update

2. dfdl:textNumberCheckPolicy="strict"
Spec might not be 100% correct in its description of difference between lax and strict.
Further, documenting this as a delta over lax probably not a good idea after **action 313**.

3. dfdl:calendarCheckPolicy="lax"
Suggestion is to make the behaviour of calendarCheckPolicy="lax" implementation dependent, due to potential var
in behaviour of ICU libraries from release to release. See also **action 313** for dfdl:textNumberCheckPolicy.

4. 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
Bradd Kadlecik

Apologies

Minutes

1. Implementations Update
No updates other than noted in actions

2. dfdl:textNumberCheckPolicy="strict"
Spec might not be 100% correct in its description of strict for standard numbers. Fold into **action 313**.

3. dfdl:calendarCheckPolicy="lax"
Suggestion is to make the behaviour of calendarCheckPolicy="lax" implementation dependent, due to potential variations in behaviour of ICU libraries from release to release, as noted for text numbers. Fold into **action 313**.

4. Action 250 (Standardise on a single tdml format for DFDL tests)
Moved to 'Deferred actions' as no prospect of further work in medium term.

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

Meeting closed
17:10 UK

Next regular call
17th September 2020 @ 16:00 UK

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

Subject	Document Type	Created	Modified

Next action: **320**

Actions raised at this meeting

No	Action

Current Actions:

No	Action
289	**Unparsing: expression refers backwards to outputValueCalc which refers beyond it.** 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.
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).

| 311 | **Move DFDL 1.0 spec to Grid Recommendation as per GFD.152 (All)**
27/6/19: Steve to create experience document for IBM DFDL usage. Mike to do same for Daffodil. Steve to reach out to the DFDL4S team at ESA for them to do the same. Mike to send email to OGF to inform them of the WG's intent and to establish who to deal with going forward.
11/7: Mike has sent email to Steve for review. |

	8/8: Mike emailed OGF and Alan Sill has replied. He has forwarded the email to the Grid Forum Steering Committee with a request for responses and asked Jens Jensen as VP for Standards to help the WG work through the process. Agreed that a draft should be created that fixes all the known typographical bugs (as noted in tracker 233) - Mike will do this. Mike will follow-up with Jens Jensen also. 29/8: DFDL WG needs to find an External Reviewer as there is no Data Area chair currently. Needs to be independent from IBM and Tresys. Also, need to establish whether the existing document can be edited or a new document is needed - the latter would entail a 6 month delay. Need to get the current errata experience document updated with the set of trackers that we need in 1.0. 17/10: External reviewer is to be Martin Westhead, who was a founding member of the DFDL WG. Assumption is that we can use the existing spec and add the errata to it, although the number of errata is large. Mike is working through the issue trackers, adding them to the errata document at https://redmine.ogf.org/dmsf_files/13384?download=, and updating the spec if they can simply be added in without further discussion or approval. Several require approval and Steve has reviewed those. Mike has responded to Steve's comments with further changes, ready for another review cycle. One erratum requiring more thought is 5.39 (issue 299) as it introduces the new concept of a choice branch with zero occurrences. 12/12: Steve has reviewed latest errata, no changes required. Note 5.39 is correct as written, no more discussion needed. Mike has incorporated the errata into the DFDL spec. Steve has reviewed the spec, and Mike has further updated it. Current version at https://redmine.ogf.org/dmsf_files/13601?download=. Martin Westhead thinks he can review by end of year. Mike will create an html version of the current version, marked as a draft, for users for convenience, and will investigate best place to host it. 9/1/20: The html version of the updated spec is available, and Martin Westhead has started to review the spec and is providing comments. He has concerns about the first few sections of the spec, particularly with regard to concepts being mentioned before explanation, so requiring multiple passes to understand. Mike considering the best way to fix this. Martin is much happier with the later sections. Mike has found a few more minor typographical errors and is fixing. 16/4: Martin has reviewed up to page 123 but had to stop due to other commitments. Mike has integrated errata 5.60 - 5.63 into a draft spec which is in Redmine as gwdrp-dfdl-v1.0.5-r12.docx, with change tracking on. Mike then created another version, accepted changes, and has started to update with some of Martin's suggestions, got as far as page 50. This is likely to result in structural changes to the spec. Mike to contact Martin again. 30/4: Mike has now incorporated up to page 123, so all of Martin's comments to-date. Mike has contacted Martin, waiting to hear back. As part of this work, Mike has prototyped the property precedence section as tables. Reviewed and some minor comments, but agreed it was an improvement over simple bullets. Mike had some questions about the precedence which were answered. At the end of the call, Steve wondered whether an alternative approach might be better - relate properties to grammar constructs. Then everything is driven from the grammar. 9/7: No further reviews from Martin Westhead. Looks like Grid Recommendation will be pushed out to 2021, as 6 month Public Comment needed. 23/7: Agreed to look for additional reviewer to help speed things up. 3/9: Martin Westhead has reviewed the entire spec. No need for an additional reviewer. Mike to carry on with incorporating Martin's comments as a high priority. Roger Costello observed a property/enum index would be useful.
313	**Investigate dfdl:textNumberCheckPolicy='lax' and plus sign behaviour (All)** 29/8/19: IBM DFDL and Daffodil look like they behave differently. Spec suggests IBM DFDL is correct. Yet both use ICU under the covers! More investigation needed. 17/10: Looks like ICU changed behaviour in a recent release, thought to be 61 around Dec 2017. Daffodil is picking up a later release. IBM DFDL is back on 51.2. We can't have this moving around, so the spec needs to pin down the ICU release(s) that are eligible for implementations to use. Daffodil will change to pick up an earlier release.

	Potential issue going forward if we have to move to a later ICU for security or bug-fix reasons. Steve to raise a ticket against ICU to (provide an option to) revert the behaviour. 12/12: ICU issue raised - https://unicode-org.atlassian.net/browse/ICU-20896, Res ponse from ICU is that the change was deliberate, is working as designed, and users should not rely on a precise specification of 'lax' as it is intended to be tolerant for handling user input. Discussed, and the feeling in the WG is to try and weaken the definition of 'lax' so that it becomes implementation-dependent. Steve will look at this further and propose an erratum. ... 23/7/20: No further progress. 3/9: Two related agenda items need folding into this action. 1) We need to make sure that 'strict' is very well defined. 2) Does calendar lax suffer from the same problem? Steve also noted that 'lax' applies for dfdl:binaryNumberCheckPolicy and for dfdl:textNumberCheckPolicy (zoned) and is well defined for both as not dependent on ICU. So, need to decide whether dfdl:textNumberCheckPolicy (standard) and dfdl:calendarCheckPolicy should be implementation-dependent. More thought needed.
315	**Determine the behaviour of fn:count(.) and fn:exists(.) (All)** 17/10/19: Not clear from the spec. Need to consider existing concept of 'knownToExist' and the existing rules about when expressions are evaluated. ... 23/7/20: No further progress. 3/9: Mike has looked at this some more. Also affects fn:exactly-one(.) and fn:empty(.). Noted wording "*(Note that DFDL v1.0 does not support sequences of length > 1.)*" in section 23.5.2.5 which is misleading and should be changed. In Daffodil test suite there are occurrences of fn:exists(.) and also fn:exists(..) - which is an extension of the problem (and by extrapolation ../.. and so on). So, the question is actually what does it mean to apply these functions with an argument that indicates the current element is, or is inside, the element being counted. More thought needed.
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.
317	**Publish some real-world numbers and key use cases for DFDL (All)** 16/4/20: This would help with the standardisation process. ... 23/7: No progress 3/9: Steve has identified that there are ~1600 customers of IBM Integration Bus / IBM App Connect Enterprise, but not clear what percentage use DFDL.
319	**Clarify positive discriminator evaluation in nested choice scenario** 23/7: Section 9.3.3.1 states what happens if a processing errors occurs, should also state what happens if discriminator evaluates to true. 3/9: Daffodil fixed a bug in this area as a result of the discussion.

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

Work items:

No	Item	Owner	Target	Status

Experimental Features:

No	Title	Original Action	Issue Tracker	Implementor	Experience Document
1	Experimental feature mechanism	301	https://redmine.ogf.org/issues/340	Daffodil	https://redmine.ogf.org/dmsf_files/13594?download=
2	Elements of type 'xs:hexBinary' with lengthUnits 'bits'	292	https://redmine.ogf.org/issues/344	Daffodil	
3	Variable Path Step in DFDL expression	287	https://redmine.ogf.org/issues/353	DFDL4S	
4	BLOBs as URIs in infoset	312	https://redmine.ogf.org/issues/356	Daffodil	In progress
5	Data streaming layers	304	https://redmine.ogf.org/issues/341	Daffodil	
6	Interoperability	307	N/A	Daffodil	https://redmine.ogf.org/dmsf_files/13589?download=
7	Empty element parse policy	306	https://redmine.ogf.org/issues/355	Daffodil	https://redmine.ogf.org/dmsf_files/13596?download=
8	Integer enums as strings	294	https://redmine.	Daffodil	

	in infoset		ogf.org/issues/354		
9	Bi-directional support (moved out of DFDL 1.0)	241	https://redmine.ogf.org/issues/357	TBD	
10	Indirection using pointers	309	TBC	IBM DFDL (z/TPF)	In progress
11	User-defined functions	316	TBC	Daffodil	