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	**12-Dec-19 (Thurs)**
Meeting Time	**16:00 - 17:00**

Created by Steve Hanson on 09-Mar-11
Last Modified by Steve Hanson on 11-Dec-19

OGF DFDL Working Group Call, 12th December 2019

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. 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

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

Meeting closed

Next regular call
9th January 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	Modifie

Next action: **316**

Actions raised at this meeting

No	Action

Current 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. 12/12:
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 ... 5/1/16: No progress ... 17/10/19: No further progress 12/12:
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:
293	**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. ... 17/10/19: No further progress 12/12:
306	**Confirm IBM DFDL behaviour when parsing empty strings (Steve)**

	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. ... 1/11: In progress - there are a lot of subtle scenarios 15/11: Not discussed ... 7/2/19: No further progress 5/4: Steve has documented IBM DFDL behaviour and summarised options. The problem is that IBM DFDL does not handle empty strings correctly when parsing; it errors if required, and throws away if optional (regardless of markup). Mike to test Daffodil (but the only non-compliance is likely to be that a default value is not used if present and required string is empty). Discussed a proposal for a new property that caused Daffodil to implement the IBM rules. More thought needed. 2/5: Mike sent proposal for new enum property dfdl:emptyElementPolicy with values 'noOptionalEmptyElements' (matches current IBM DFDL behaviour) and 'optionalEmptyElementsWithSyntax' (matches DFDL 1.0 spec and Daffodil behaviour). Review for next call. 31/5: Email discussion on the name and enums for the property. Steve proposed dfdl:emptyElementParsePolicy = 'treatAsMissing' | 'treatAsEmpty'. Daffodil has implemented this and it has shown it to work. For positional occurrences where order needs preserving, also need to use nillable='true' and dfdl:nilValue='%ES;'. This shows up a potential flaw in the name, as ES nil processing must take place before the property is applied. Steve has one remaining concern around IBM DFDL's behaviour, and that is whether its dfdl:nilValue='%ES;' behaviour might be taking precedence over 'missing', and needs to investigate further. 27/6: Email sent by Steve. dfdl:nilValue='%ES;' is the 1st zero-length rep check, so IBM DFDL is correct. 2nd check is empty rep check. 3rd check is normal rep - an edge case around EVDP. Failing that it's the absent rep. IBM DFDL does not add anything to infoset for 2 or 3. The currently proposed property name dfdl:emptyElementParsePolicy is therefore not 100% accurate, needs some more thought. 11/7: Daffodil has implemented dfdlx:emptyElementParsePolicy in the experimental namespace and has shown that the IBM DFDL schemas for TLog can be parsed successfully using 'treatAsMissing', thereby demonstrating interoperability. Because it is key to interoperability, it should go into DFDL 1.0, and IBM DFDL needs to at least recognise the property and give an error if 'treatAsEmpty' used. Need to agree on any naming refinements, eg, use 'treatAsAbsent' instead of 'treatAsMissing'. 8/8: Added to experimental features table. New tracker **https://redmine.ogf.org/issues/355**. Keeping action open as there will be an erratum on DFDL 1.0. Likely outcome is that 'treatAsEmpty' is an optional feature, as not implemented by IBM. 29/8: Not closing until erratum written up. 17/10: Experience document 7 created by Mike to describe this (https://redmine.ogf.org/dmsf_files/13596?download=), plus proposed erratum wording. Steve reviewed the document and had several comments. Mike has revised the document. Steve will re-review the document and review the erratum wording. 12/12: Steve is happy with revised document. Needs erratum creating.
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:

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=.
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, Response 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.
314	**Clarify behaviour of documentFinalTerminatorCanBeMissing="yes" (All)** 17/10/19: This property states that is not an error if there is no terminator found, but does not state what the effect is on emptyValueDelimiterPolicy and nilValueDelimiterPolicy. Should the terminator be considered to be logically present? 12/12: Steve believes it should, and would like the following or similar added at the end of the first paragraph in the property description "If not found, the terminator is considered to be logically present for the purposes of establishing representation."
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. 12/12:

Closed actions

No	Action

Deferred actions

No	Action

Work items:

No	Item	Owner	Target	Status

Experimental Features:

No	Title	Original	Issue Tracker	Implem	Experience Document

		l Action		entor	
1	Experimental feature mechanism	301	https://redmine.ogf.org/issues/340	Daffodil	
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	
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 in infoset	294	https://redmine.ogf.org/issues/354	Daffodil	
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)	