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: 01-Oct-20 (Thurs)
Meeting Time: 16:00 - 17:00

Created by Steve Hanson on 09-Mar-11
Last Modified by Steve Hanson on 30-Sep-20

OGF DFDL Working Group Call, 1st October 2020

Agenda

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

1. Implementations 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
15th October 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.
Next action: **321**

**Actions raised at this meeting**

<table>
<thead>
<tr>
<th>No</th>
<th>Action</th>
</tr>
</thead>
</table>

**Current Actions:**

<table>
<thead>
<tr>
<th>No</th>
<th>Action</th>
</tr>
</thead>
</table>
| 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 unparsing 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, unpars 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.
17/9: No further progress

1/10:

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 DF DL. 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 DF DL 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](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 DF DL 2.0 will introduce support for some of these, specifically:
- Item exists in Infoset with default value, so unparsl 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 unparsling, 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 DF DL 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.org/ogf/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.

1/10:

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 Daffodi. 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 Sil 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.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
Investigate dir/textNumberOfCheckpoints=10 and plus sign behaviour (All)

11/10: Completed draft sent out by Mike for review.

Mike has made good progress on the 10/10 draft, and I'm confident he'll have it ready for review by the end of the month. I've reviewed the draft and found it to be well-written and comprehensive. Mike has done an excellent job of summarizing the key points and providing examples to illustrate the concepts.

I've also suggested a few minor changes to improve the clarity and flow of the text. Mike has agreed to make these changes and I look forward to reviewing the updated draft.

Thank you, Mike, for your hard work on this project. I'm confident that the final product will be a valuable resource for our team.

Best,
[Signature]

11/10: Mike has updated the draft based on my feedback. It's now ready for review by the team. I encourage everyone to take a look and provide your input.

Mike has also suggested a few additional changes that I think are worth considering. I'll send these changes to the team for review as well.

Thank you, Mike, for your dedication to this project. I'm confident that the final product will be a valuable resource for our team.

Best,
[Signature]
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 dfd:binaryNumberCheckPolicy and for dfd:textNumberCheckPolicy (zoned) and is well defined for both as not dependent on ICU. So, need to decide whether dfd:textNumberCheckPolicy (standard) and dfd:calendarCheckPolicy should be implementation-dependent. More thought needed.

17/9: Discussed whether we need to fork ICU to obtain a level where the behaviour is fixed.
1/10:

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.
17/9: No further progress.
1/10:

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.
17/9: No further progress.

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.
1/10:

320 Move DFDL WG documents from OGF Redmine to OGF GitHub (All)
17/9: Agreed that it made sense to re-base DFDL WG documents into OGF GitHub. Steve believes Redmine does not support TLS 1.2 and is prone to being hacked (advert-like comments appear sometimes in issue trackers). Would need to move documents, WG call minutes, issue trackers, news items and introductory material. DFDLSchemas should remain as a separate GitHub organization, but linked to.
1/10:

Closed actions
### Deferred actions

<table>
<thead>
<tr>
<th>No</th>
<th>Action</th>
</tr>
</thead>
</table>
| 250 | **Standardise on a single tdmI format for DFDL tests (All)**  
 5/2/14: Steve has requested permission for IBM to view / use the Daffodil tdmI files, as a precursor to trying to standardise on a common tdmI 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 tdmI format.  
25/3: Discussed the creation of an OGF document that will own and define a standardised tdmI format.  
11/4: Proposal is for the OGF document to define a tdmI 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 tdmI suite against both Daffodil and IBM DFDL.  
...  
10/2: No further progress  
24/2: Mike updating the Daffodil TDML test runner to handle unparsers (ie, serializer) tests  
...  
14/4: No further progress  
28/4: Tresys have enhanced their tdmI runner to allow unparsers 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](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](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. Bradl 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:

<table>
<thead>
<tr>
<th>No</th>
<th>Item</th>
<th>Owner</th>
<th>Target</th>
<th>Status</th>
</tr>
</thead>
</table>

Experimental Features:

<table>
<thead>
<tr>
<th>No</th>
<th>Title</th>
<th>Original Action</th>
<th>Issue Tracker</th>
<th>Implementor</th>
<th>Experience Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Elements of type xs:hexBinary 'with lengthUnits 'bits'</td>
<td>292</td>
<td><a href="https://redmine.ogf.org/issues/344">https://redmine.ogf.org/issues/344</a></td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feature Description</td>
<td>Status</td>
<td>Link</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Variable Path Step in DFDL expression</td>
<td>287</td>
<td><img src="https://redmine.org/issues/353" alt="Redmine" /> DFDL4S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>BLOBs as URIs in infoset</td>
<td>312</td>
<td><img src="https://redmine.org/issues/356" alt="Redmine" /> Daftodil In progress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Data streaming layers</td>
<td>304</td>
<td><img src="https://redmine.org/issues/341" alt="Redmine" /> Daftodil</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Interoperability</td>
<td>307</td>
<td>N/A <img src="https://redmine.org/dmsf_files/13589?download=" alt="Redmine" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Empty element parse policy</td>
<td>306</td>
<td><img src="https://redmine.org/dmsf_files/13596?download=" alt="Redmine" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Integer enums as strings in infoset</td>
<td>294</td>
<td><img src="https://redmine.org/issues/354" alt="Redmine" /> Daftodil</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Bi-directional support moved out of DFDL 1.0</td>
<td>241</td>
<td><img src="https://redmine.org/issues/357" alt="Redmine" /> TBD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Indirection using pointers</td>
<td>309</td>
<td><img src="https://redmine.org/issues/363" alt="Redmine" /> IBM DFDL (z/TPF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>User-defined functions</td>
<td>316</td>
<td>TBC Daftodil</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>