DFDL WG Call Agenda

Meeting about Meetings\OGF

<table>
<thead>
<tr>
<th>Project</th>
<th>DFDL 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting Date</td>
<td>17-Oct-19 (Thurs)</td>
</tr>
<tr>
<td>Meeting Time</td>
<td>16:00 - 17:00</td>
</tr>
</tbody>
</table>

Created by Steve Hanson on 09-Mar-11
Last Modified by Steve Hanson on 16-Oct-19

OGF DFDL Working Group Call, 17th October 2019

Agenda

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

1. Daffodil Open Source Project
   Status update

2. Experiences with calculated values and hidden groups
   Daffodil have a draft on their wiki [https://s.apache.org/daffodil-experience-with-computed-elements](https://s.apache.org/daffodil-experience-with-computed-elements).
   Bradd has some suggestions for inputValueCalc to make it less restrictive.

3. Simple format that is impossible to model
   Mike has encountered a defence industry delimited format that needs to preserve the position of empty array occurrences but is unable to use nil values to do that because there is already a non-empty nil value. This led to an exchange on the possibility of empty occurrences being treated as normal occurrences, which the spec caters for. Does documentFinalTerminatorCanBeMissing need clarification?

4. Recent puzzlers on fn:count and fn:exists
   What do these functions return when used with argument ()?

5. 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
31st October 2019 @ 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

<table>
<thead>
<tr>
<th>Subject</th>
<th>Document Type</th>
<th>Created</th>
<th>Modified</th>
</tr>
</thead>
</table>

Next action: 314

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>
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 unparsers (ie, serializer) tests
...
14/4: No further progress
28/4: Tresys have enhanced their tdml 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 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 250 in
October 2016.

250 Public comment: dfdl:valueLength and dfdl:contentLength descriptions
(Mike)
19/11/14: http://redmine.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
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: Undeffering 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 dfd: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 unparsers 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 dfd: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 dfd:valueLength and dfd:contentLength functions before action can be closed.
...
29/8/19: No further progress
17/10: Incorporated into latest errata document. Noted that this action had its number accidentally swapped with 242 in October 2016.

### 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
...
29/8/19: No further progress
17/10:

### 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 dfd:setVariable, which on unpars is set using dfd: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.
...
29/8/19: No further progress
17/10:

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.
...
29/8/19: No further progress
17/10:

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
effects 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
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 inoset 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 dfdix: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. Steve has reviewed and has several comments.

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.

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. Mike is working through the trackers, adding them to the errata document, and updating the spec if they can simply be added in without further discussion or approval. Several require approval and Steve has reviewed those, and will need discussion.

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 release (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 should probably pin down the ICU release(s) that are eligible for implementations to use.

Closed actions

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

Deferred actions

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

Work items:

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

Experimental Features:

<table>
<thead>
<tr>
<th>No</th>
<th>Title</th>
<th>Origina l Action</th>
<th>Issue Tracker</th>
<th>Implem entor</th>
<th>Experience Document</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://redmine.org/issues/301">https://redmine.org/issues/301</a></td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Elements of type xs:hexBinary' with lengthUnits 'bits'</td>
<td>292</td>
<td><a href="https://redmine.org/issues/344">https://redmine.org/issues/344</a></td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Variable Path Step in DFDL expression</td>
<td>287</td>
<td><a href="https://redmine.org/issues/353">https://redmine.org/issues/353</a></td>
<td>DFDL4S</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLOBs as URIs in infoSet</td>
<td>312</td>
<td><a href="https://redmine.org/issues/335">https://redmine.org/issues/335</a></td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Data streaming layers</td>
<td>304</td>
<td><a href="https://redmine.ogf.org/issues/356">https://redmine.ogf.org/issues/356</a></td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Interoperability</td>
<td>307</td>
<td>N/A</td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Empty element parse policy</td>
<td>306</td>
<td><a href="https://redmine.ogf.org/issues/355">https://redmine.ogf.org/issues/355</a></td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Integer enums as strings in infoset</td>
<td>294</td>
<td><a href="https://redmine.ogf.org/issues/354">https://redmine.ogf.org/issues/354</a></td>
<td>Daffodil</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Bi-directional support (moved out of DFDL 1.0)</td>
<td>241</td>
<td><a href="https://redmine.ogf.org/issues/357">https://redmine.ogf.org/issues/357</a></td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Indirection using pointers</td>
<td>309</td>
<td>TBC</td>
<td>IBM DFDL (z/TPF)</td>
<td></td>
</tr>
</tbody>
</table>