

# 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	27-Jun-19 (Thurs)
Meeting Time	16:00 - 17:00

Created by Steve Hanson on 09-Mar-11

Last Modified by Steve Hanson on 27-Jun-19

## OGF DFDL Working Group Call, 27th June 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. Clarify choice branch behaviour when occursCountKind 'expression' and 'parsed'.

Mike has revisited the tracker for this (<https://redmine.ogf.org/issues/299>) and emailed the expected behaviour. Steve concurs.

#### 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

#### IPR Statement

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

#### Meeting closed

#### Next regular call

11th July 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

Subject	Document Type	Created	Mod
---------	---------------	---------	-----

Next action: **311**

### Actions raised at this meeting

No	Action

### Current Actions:

No	Action
228	<b>Review set of tutorial lessons (All)</b> 17/9/13: Lesson 1 proposes a set of lessons, needs reviewing as over 2 years old. ... 22/10: No progress 31/10: Becoming a focus for Tresys. Steve to send his 'Modeling Data Formats using DFDL' powerpoint. ... 19/11: No further progress 26/11: Possibility of help from MITRE high-school student, and from Marisa at

IBM.

...

11/3/14: No further progress

25/3: MITRE have produced a couple of new tutorials under the guidance of James Gariss. Jonathan to forward for review.

Mike observed that an html tutorial could be generated from a tdml file using XSLT.

11/4: Not discussed

15/4: Jonathan will send 4 new mini-tutorials. Need to figure out best way to incorporate into the tutorial structure.

29/4: Tutorials received. Mark has taken a quick read. Mark & Steve to review and report back.

6/5: Still with Mark and Steve

20/5: Mark has reviewed. Will ask IBM information development to recommend a way to portray the existing and new lessons, preferably web-based. Find somewhere to host them. OGF? GitHub? developerWorks? NCSA?

3/6: Steve has also reviewed.

...

17/6: No further progress on tutorials. Tim is looking into the creation of some DFDL how-to videos using the IBM Integration Studio.

...

31/3/15: No further progress

14/4: Agreed that the need for better tutorials has become pressing for Daffodil users who aren't using IBM's tools and material. Discussed creating tutorials based on a tdml file with comments that is processed to produce html. Mike to investigate.

28/4: Mike has sent an example tdml file which embeds instances of a new 'tutorial' element in various places. These elements contain html which can be extracted and formatted in a browser. Suggest future DFDL tutorials are created using this technology.

12/5: Not discussed

...

22/9: No further progress

3/11: Daffodil team has someone working on the new 'tutorial' element in tdml files. In time this should result in some new tutorials and re-working of existing tutorials.

5/1/16: Mike has started a bitOrder tutorial using the tdml file approach (uses stylesheets to render html).

16/2: The bitOrder tutorial is available on the web @ <https://opensource.ncsa.illinois.edu/bamboo/artifact/DFDL-MASTER21/JOB1/build-132/Tutorials/bitorder.tutorial.tdml.xml>

1/3: Awaiting review.

...

17/4/18: No further progress

15/5: Daffodil sponsor has requested some progress on tutorials as it will soon become the limiting factor on uptake.

...

29/11: No further progress

10/1/19: Mike has been reviewing a new 4 day tutorial course written by Roger Costello of MITRE. The format is different from existing tutorials.

7/2: Mike waiting to review re-spin of Roger's tutorial.

5/4: Mike needs to review ~400 slides worth of tutorial. Expected to be published in a month or so.

2/5: No update.

31/5: Mike still reviewing, and need publication clearance. Each tutorial needs annotating to make it clear if they are implementation dependent.

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 info set.

...

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

	<p>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 <a href="https://github.com/OpenDFDL/ibmDFDLCrossTester/pull/1">https://github.com/OpenDFDL/ibmDFDLCrossTester/pull/1</a>. Steve needs to talk to IBM legal to check this is ok as it currently modifies IBM DFDL sample code.</p> <p>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.</p> <p>7/2: If permitted, the tests from IBM can be used to see how the IBM and Daffodil tdmIs have diverged.</p> <p>5/4: IBM are permitted to send Daffodil some example tdmIs. Steve to send some to Mike for next call.</p> <p>...</p> <p>31/5: No progress</p>
250	<p><b>Public comment: dfdl:valueLength and dfdl:contentLength descriptions (Mike)</b></p> <p>19/11/14: <a href="http://redmine.ogf.org/boards/15/topics/63">http://redmine.ogf.org/boards/15/topics/63</a>. 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.</p> <p>26/11: Needs wording to handle all the issues found, assigned to Mike.</p> <p>...</p> <p>11/3/15: Still with Mike</p> <p>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.</p> <p>11/4: More revised wording sent by Mike. Started to review but realised it needed some off-line preparation and thought. Review for next call.</p> <p>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 &gt; 1; should be a processing error, can always use a predicate to select one node of an array.</p> <p>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.</p> <p>6/5: Mike is working on a mind map for the length section. Deferring until needed.</p> <p>....</p> <p>23/9: Rewrite should be postponed to future 1.1. Still need to answer the original questions about the functions though...</p> <p>...</p> <p>25/4/2016: Undeferring 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.</p> <p>7/6: Thread to include use of a variable with dfdl:outputValueCalc. May be undefined at point of evaluation.</p> <p>5/7: Mike has been looking at this. Two main points:</p> <ol style="list-style-type: none"> <li>1) Computing the content length of a complex element with internal alignment.</li> <li>2) Computing length in chars of a complex element which is not 100% text</li> </ol> <p>Mike will send out a discussion via email. Noted that rules should also apply to</p>

	<p>prefixed length calculation.          (Aside: IBM DFDL unparser 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 dfdl:valueLength(). Agreed on the change. <b>Issue <a href="https://redmine.ogf.org/issues/316">https://redmine.ogf.org/issues/316</a> created.</b>          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. <b>Issue <a href="https://redmine.ogf.org/issues/317">https://redmine.ogf.org/issues/317</a> created.</b>          #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. <b>Issue <a href="https://redmine.ogf.org/issues/318">https://redmine.ogf.org/issues/318</a> created.</b>          #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. <b>Issue <a href="https://redmine.ogf.org/issues/319">https://redmine.ogf.org/issues/319</a> created.</b>          13/9: Review issues created. Need to come up with the revised descriptions for dfdl:valueLength and dfdl:contentLength functions before action can be closed.          ...  <a href="#">31/5/19: No further progress</a></p>
279	<p><b>Improve defaulting description to explicitly cover local groups (Mike)</b>          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          ...  <a href="#">31/5/19: No further progress</a></p>
287	<p><b>Find a way to handle a variable path step in DFDL expression (All)</b>          1/3/16: DFDL4S currently using a hack that embeds a regex in a path step.          10/5: No progress          24/5: Need example from DFDL4S          ..          5/7: Need to ask DFDL4S for example.          2/8: DFDL4S sent example. They use dfdl:contentLength() with a path that has a step that contains a regex as a wildcard. Mike has requested the wider set of schemas to be sent, in order to see if there is a viable alternative.          13/9: Response received from DFDL4S, not yet analysed.          ...          10/1/17: No further progress          7/2: Mike has analysed the schemas and sent a comprehensive reply to DFDL4S. He believes that the variable path step is effectively a way of parameterizing the expression, and has described how this can be done using DFDL variables.          DFDL4S have responded and will talk to the contractor that authored the implementation.          21/2: No response so far from DFDL4S.          4/4: Mike has seen a further example of this. Still no response from DFDL4S.</p>

	<p>...</p> <p>25/7: No further progress</p> <p>3/10: Mike has seen a further example where an expression needed to look back inside an earlier choice, where there was a common element. Discussed whether XPath 2.0 wildcards could be used (currently not supported in DFDL 1.0). This looks to be a good fit, and would involve only a minimal change to the supported syntax. Steve will email DFDL4S.</p> <p>...</p> <p>11/12: ESA will look into this as part of the next round of changes to DFDL4S.</p> <p>...</p> <p>17/4/18: No update</p> <p>15/5: Steve has emailed DFDL4S asking for a progress update. Also, Mike will put together a concrete proposal.</p> <p>...</p> <p>4/9: No further progress</p> <p>16/10: Steve will send chaser email to DFDL4S team.</p> <p>1/11: Marcus Bento from DFDL4S team responded: "I confirm that Action 287, related to using regex as part of the path in our schemas, is expected to be addressed in the early release of 2019. I've investigated the current implementation, and believe that your suggested approach (based on XPath 2.0) is sufficient for our needs. However, only after the release of December will the contractor analyze the issue further and confirm that the implementation works."</p> <p>...</p> <p>10/1/19: No progress likely until January at the earliest.</p> <p>7/2: Steve will reach out to DFDL4S team again.</p> <p>5/4: Not addressed in the early release of 2019, expected in a later release of 2019. Action can be closed once a tracker issue has been created to specify formally the behaviour of DFDL 1.0 XPathS with wildcards.</p> <p>...</p> <p><a href="#">31/5: No progress on tracker</a></p>
289	<p><b>Unparsing: expression refers backwards to outputValueCalc which refers beyond it.</b></p> <p>2/8/16: Need to decide if this is allowed and if so if there are any restrictions.</p> <p>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.</p> <p>11/10: Daffodil has implemented this, Mike to provide scenario.</p> <p>8/11: Mike couldn't find example, will continue to look</p> <p>10/1/17: Mike has realised that all the examples were reworked to avoid using variables, hence why can't be found.</p> <p>7/2: Daffodil will soon be implementing dfdl:newVariableInstance which will bring this up again.</p> <p>...</p> <p>17/4/18: Waiting for Daffodil to implement dfdl:newVariableInstance</p> <p>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.</p> <p>...</p> <p><a href="#">31/5/19: No further progress</a></p>
292	<p><b>Write up proposal for allowing hexBinary elements to have dfdl:lengthUnits='bits' (Mike)</b></p> <p>7/2/17: Mike will create a proposal for evaluation.</p> <p>21/2: No progress</p> <p>4/4: Daffodil has experimental implementation, will be evaluated and written up.</p> <p>...</p>



	<p>15/5/18: Daffodil to write up.  7/8: Reviewed by Steve.  4/9: No further progress</p> <p>...</p> <p>15/11: Mike to revise after Steve's comments  29/11: Mike has sent out a second proposal due to problems with the first, but Steve not comfortable with the second proposal as hexBinary starts to get integer properties, even if only when a new property is set. Mike will take this back to drawing board. Agreed that bitOrder does come into play here.  10/1/19: Mike has done some thinking via email to which Steve has responded. Further discussion concluded that a) need to honour bitOrder but not byteOrder; b) minLength &amp; maxLength facets are compatible with 'bits' as long as lengthKind is not 'implicit'; c) need a switch to turn on lengthUnits for hexBinary as it has not been relevant before.  7/2: Daffodil has implemented a), Mike to check if b) and c) done too. Also see action 301 - will be experimental feature.  5/4: Needs to be made into an experimental feature. b) will not be implemented until DFDL 2.0. Mike to raise tracker issue, target DFDL 2.0, design to be informed by experience document. Action can be closed when tracker in place.</p> <p>...</p> <p>31/5: No progress on tracker</p>
293	<p><b>Investigate solutions to enabling choices in hidden groups to be unparsed (All)</b></p> <p>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.</p> <p>...</p> <p>17/4/18: On hold.  15/5: Daffodil now looking at this and will write up a proposal. Potential commonality with action 289.</p> <p>...</p> <p>31/5/19: No further progress</p>
294	<p><b>Converting integer enumerations to meaningful strings in infoSet (Mike)</b></p> <p>18/4/17: Requirement from Daffodil user for parser to convert an integer enum to a meaningful string value in infoSet.  Daffodil has put forward a proposal but it relies on [unionMemberSchema] which is a validation-only property.  <a href="https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Enumerations+and+Range+Tables+via+Simple+Type+Unions">https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Enumerations+and+Range+Tables+via+Simple+Type+Unions</a>  Mike to re-think the approach, and also consider whether this kind of transformation is really a post-DFDL step.  Steve to check how XQuery would approach the same problem.</p> <p>...</p> <p>25/7: No progress  3/10: Also received same request from a product team at IBM.  21/11: Consider whether any additional annotation is not DFDL, for possible wider applicability.  11/12: On Daffodil priority list to investigate</p> <p>...</p> <p>6/3/18: No progress  17/4/18: Latest proposal is at  <a href="https://cwiki.apache.org/confluence/display/DAFFODIL/Proposal%3A+Features+to+Support+Table-Lookup">https://cwiki.apache.org/confluence/display/DAFFODIL/Proposal%3A+Features+to+Support+Table-Lookup</a>. Feedback requested. Review for next call.  15/5/18: Review for next call.  7/8: No further progress  4/9: Steve to review for next call</p>



	<p>16/10: Steve thinks there is a major problem with the carrying of the rep properties, as the types are different. Mike to respond to Steve's comments. Need something that is a cross between IVC / OVC and prefixLengthType!</p> <p>1/11: Mike to think some more. Need to be exact as to which properties can no longer be carried on the element / string type. Steve wondered if the mechanism could be generic, allowing any type to appear as a string in the infoset via a default toString style mapping.</p> <p>...</p> <p>10/1/19: No further progress</p> <p>7/2: Commercial demand means Daffodil need to implement this soon. Also see action 301 - will be experimental feature.</p> <p>5/4: Needs to be made into an experimental feature. Daffodil in process of implementing. Becoming more elaborate!</p> <p>2/5: No progress</p> <p><a href="#">31/5: Daffodil working on this as experimental feature, and evaluating it, prior to final proposal.</a></p>
<b>301</b>	<p><b>How to indicate DFDL v2 in schemas (All)</b></p> <p>6/2/18: Agree on best way to do this. The DFDL namespace was originally intended to be used for this, but XSDL for example uses a separate 'XMLSchema-versioning' namespace and min/max attributes which allows schemas to be authored that may be processed by both XSDL 1.0 and XSDL 1.1. More investigation required.</p> <p>...</p> <p>17/4/18: No progress</p> <p>15/5/18: Steve to familiarise himself with how XSDL does this.</p> <p>7/8: No further progress</p> <p>4/9: Need concrete statement of the requirements to be solved by a versioning mechanism.</p> <p>...</p> <p>10/1/19: No further progress</p> <p>7/2: The motivation behind this action was to be able to accommodate anticipated future DFDL 2.0 features in DFDL 1.0 schemas. A better approach is to call these features 'experimental' and to use a separate namespace from the DFDL one. See email thread. This primarily affects Daffodil so Mike will look at their '2.0' additions and see how to convert these to experimental. The z/TPF product team in IBM have done the equivalent of this to support non-contiguous data via 'pointers' - see minutes. Need to decide on the URL for the experimental namespace and a matching prefix. Agreed that the namespace declaration must be on the schema root, and scoping of properties is supported. Not decided whether element form property support needed nor what the syntax would be.</p> <p>5/4: Steve has a sent a proposal for a minimum set of changes for an experimental facility. Mike has responded and there is no need to be that restrictive. It makes syntactic sense to allow attribute and element forms using a QName. New features should always be enabled by a new experimental property, so that they don't trip up other implementations. A new OGF experience document should result from each experimental feature. Need somewhere to document how experimental features work - a separate OGF document?</p> <p>2/5: Mike sent proposal that this should be documented by an erratum to the DFDL 1.0 spec. Agreed by WG. Needs tracker raising.</p> <p><a href="#">31/5: No progress on tracker</a></p>
<b>304</b>	<p><b>Proposal for Data Streaming for layered transforms (All)</b></p> <p>17/4/2018: Daffodil team to provide feedback on the proposal as prototype gets implemented. Others to review.</p> <p>15/5/18: Mike to send link to updated proposal, which is working fine in prototype.</p> <p>7/8: Reviewed by Steve</p> <p>4/9: Steve to revisit the proposal, one idea was to group the properties into a new</p>

	<p>DFDL annotation.</p> <p>16/10: Discussed pros and cons of having a separate annotation. Mike to decide which way to jump based on real examples.</p> <p>1/11: No further progress</p> <p>15/11: Not discussed</p> <p>...</p> <p>10/1/19: No further progress</p> <p>7/2: Daffodil currently not using a separate annotation. Also see action 301 - will be experimental feature.</p> <p>5/4: Needs to be made into an experimental feature. Mike to raise tracker issue, target DFDL 2.0, design to be informed by experience document. Action can be closed when tracker in place.</p> <p>...</p> <p><a href="#">31/5: No progress on tracker</a></p>
<b>305</b>	<p><b>Create tracker issues in Redmine for spec clarifications from 7th August (Mike)</b></p> <p>7/8: Several clarifications to text in section 9 and property descriptions in sections 12, 13, 14.</p> <p>4/9: Mike to do this</p> <p>16/10: No progress</p> <p>1/11: Mike will create the issues as tests are created. OGF switching to GitHub so that is likely the way forward for trackers.</p> <p>15/11: Not discussed</p> <p>29/11: No further progress, Mike wants to do this before year end.</p> <p>...</p> <p><a href="#">31/5/19: No further progress</a></p>
<b>306</b>	<p><b>Confirm IBM DFDL behaviour when parsing empty strings (Steve)</b></p> <p>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.</p> <p>...</p> <p>1/11: In progress - there are a lot of subtle scenarios</p> <p>15/11: Not discussed</p> <p>...</p> <p>7/2/19: No further progress</p> <p>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.</p> <p>2/5: Mike sent proposal for new enum property <code>dfdl:emptyElementPolicy</code> with values 'noOptionalEmptyElements' (matches current IBM DFDL behaviour) and 'optionalEmptyElementsWithSyntax' (matches DFDL 1.0 spec and Daffodil behaviour). Review for next call.</p> <p>31/5: Email discussion on the name and enums for the property. Steve proposed <code>dfdl:emptyElementParsePolicy = 'treatAsMissing'   'treatAsEmpty'</code>. Daffodil has implemented this and it has shown it to work. For positional occurrences where order needs preserving, also need to use <code>nillable='true'</code> and <code>dfdl:nilValue='%ES'</code>. 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 <code>dfdl:nilValue='%ES'</code> behaviour might be taking precedence over 'missing', and needs to investigate further.</p> <p><a href="#">27/6: Email sent by Steve. <code>dfdl:nilValue='%ES'</code> is the 1st zero-length rep check, so</a></p>

	<p>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. Steve proposing that the property becomes dfdl:emptyElementParsePolicy = 'treatAsAbsent'   'treatAsEmpty'.</p>
307	<p><b>Demonstrate implementation interoperability (Steve, Mike)</b></p> <p>4/9: Need to make sure that DFDL spec section 21 lists a correct set of optional features, the implication being that Daffodil and IBM DFDL (and any other minimally conforming implementation) correctly implement the remaining required features. First step - see if there are any obvious omissions.</p> <p>16/10: Steve sent email stating IBM DFDL's missing core features and non-compliant behaviour, and Mike responded. Discussion continuing via two separate email threads. Part 1 for core features. Part 2 for optional features. For the core features, agreed that the following needs to happen:</p> <ol style="list-style-type: none"> <li>1) IBM adds encodingErrorPolicy='replace'</li> <li>2) Daffodil adds encodingErrorPolicy='error'</li> <li>3) Daffodil ensures that, if not implementing default/fixed when parsing, it gives an SDE if a required occurrence has empty rep and element has default/fixed set.</li> <li>4) A position is agreed on BOM handling - ongoing via email.</li> </ol> <p>1/11: Just BOM to conclude on from the above list</p> <p>15/11: Not discussed</p> <p>29/11: No further progress.</p> <p>10/1/19:</p> <ol style="list-style-type: none"> <li>1) IBM have started the work to add encodingErrorPolicy='replace'.</li> <li>2) Daffodil have a temp setting to tolerate encodingErrorPolicy='error' with a warning.</li> <li>3) Daffodil to investigate whether this is feasible.</li> <li>4) More discussion needed on BOM</li> </ol> <p>7/2: Updates:</p> <ol style="list-style-type: none"> <li>1) In progress</li> <li>2) As above.</li> <li>3) In progress</li> <li>4) No progress</li> </ol> <p>5/4: Updates:</p> <ol style="list-style-type: none"> <li>1) IBM continuing work on encodingErrorPolicy='replace' (and a customer has requested it)</li> <li>2) In Daffodil's backlog but not likely soon</li> <li>3) This will be informed by Daffodil testing for action 306</li> <li>4) Steve has sent proposal for BOMs, preferring to remove from spec. Agreed this was best option, Steve will create tracker issue.</li> </ol> <p>...</p> <p>31/5: No progress on tracker. Good progress elsewhere though. Public schemas CSV, NACHA, ISO8583, TLog all work with IBM DFDL and Daffodil. Plus HL7 and numerous other formats that Daffodil have. IBM DFDL implementing encodingErrorPolicy='replace'.</p> <p>27/6: Mike has created new experience document on Redmine for interoperability. Need to follow the process described by GFD 152.</p>
308	<p><b>Sequence terminator that exists or not depending on expression (Mike, Steve)</b></p> <p>29/11: The motivating example is where a string is either length=X and no terminator or length&lt;X and a terminator, eg :</p> <pre> &lt;xs:element name="value" type="xs:string" dfdl:lengthKind="pattern"     dfdl:lengthPattern="[^\\x7F]{0,49}(?=\\x7F) .{50}" /&gt; &lt;xs:sequence     dfdl:terminator="{if (fn:string-length(/value) eq 50) then '%ES;' else '%#x7F;'}" </pre>

	<p>/&gt;</p> <p>The current rules prevent ES from being used in this manner, but allow WSP*. This does not seem consistent. More research needed to understand the reasoning behind erratum 2.148 which is where the behaviour originates. 10/1/19: Not discussed</p> <p>...</p> <p>2/5: No progress</p> <p>31/5: The stumbling block is the 3rd paragraph of erratum 2.148 in DFDL Experience Document 1: <i>"ES must not appear as the only DFDL string literal in the property. It can only appear as a member of a list."</i> WG can not recall why this extra restriction was applied to ES over and above WSP* , which after all can have zero length. Daffodil have therefore allowed %ES; when the length is not delimited in order to accommodate the example above. Mike to raise a tracker to address the proposed change to the spec.</p> <p><a href="#">27/6:</a></p>
<b>309</b>	<p><b>Create example scenarios to illustrate offset &amp; pointer requirements (Bradd)</b></p> <p>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.</p> <p>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 <i>*not*</i> appear in the info set. 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.</p> <p><a href="#">31/5: Bradd to write up pointers proposal as an experimental feature.</a></p>
<b>310</b>	<p><b>PE or SDE if dfdl:choiceDispatchKey expression evaluates to an empty string? (All)</b></p> <p>31/5/19: Feels like an example of <i>"Dynamic Type Error "</i> - see spec section 2.6</p> <p><a href="#">27/6:</a></p>

## Closed actions

No	Action

## Deferred actions

No	Action
<b>241</b>	<p><b>Public comment: Bi-di properties placement in precedence section (All)</b></p> <p>7/11: This looks deliberate but the asymmetry between parsing and unparsing is unclear. Really needs Daffodil or IBM DFDL to implement these properties, which</p>

	has not happened yet. Deferring this action. ... 23/9: Candidate to be moved out to 1.1 ?
--	---

**Work items:**

No	Item	Owner	Target	Status