DFDL Workgroup Action 191.

2013-01-15

Separated parse and unparse errors. Fixed TBD on NaNs. Added TBDs on pending issue of dfdl:encodingErrorPolicy='error'.

Added SDE for fn:count(....) on a non-array non-optional expression

2012-12-11

Added categories of errors per WG discussion on call last week.

Incorporated errors that exceed implementation limits as a category of processing errors.

2012-12-04

Detail of this sort could be added to the specification, either as an additional section at the end of section 2, or as an appendix.

There is a danger that this section becomes a sweep through the specification to re-list (redundantly), every mention of an error, whether it be processing or SDE. This might add some value in that the errors could then be assigned identifiers, but this is more work and may not be desirable just from the standpoint of completing the next official draft in timely manner.

The following are processing errors:

* Arithmetic Errors
	+ Division by zero
	+ Integer Arithmetic Underflow
	+ Integer Arithmetic Overflow
	+ Note: Floating point math can produce NaN (Not a Number) values. This is not an error, nor are properly typed operations on floating point NaN values.
* Expression Errors
	+ Dynamic Type Error – unable to convert to target type
		- Example: non-digits found in string argument to xs:int(…) constructor.
		- Note: if a DFDL Implementation cannot distinguish Dynamic Type Errors from Static Type Errors, then a Dynamic Type Error should cause a Schema Definition Error
	+ Index out of bounds error – index not < number of occurrences, or is < 1.
		- Note: same error for fn:testBit if bitPos is not 1..8, or for character positions in a string-value
	+ Indexing of non-array non-optional element
		- E.g., x[1] when x is declared and has both minOccurs=”1” and maxOccurs=”1” explicitly, or by not stating either or both of them.
	+ Illegal argument value (correct type, illegal value)
* Parse Errors
	+ Delimiter not found
	+ Data not convertible to type
	+ Assertion failed
	+ Discriminator failed
	+ Required occurrence not found
	+ No choice alternative successfully parsed.
	+ Character Set Encoding - Decode Error
		- E.g., when attempting to parse a delimiter, the data cannot be decoded into characters of the specified encoding. (Similarly when attempting to parse a nil value)
		- TBD: ….and dfdl:encodingErrorPolicy='error'
		- E.g., when attempting to match a regular expression pattern, the data cannot be decoded into characters of the specified encoding.
* Unparsing Errors
	+ Truncation scenarios where truncation is being disallowed
	+ Rounding error – rounding needed but not allowed. (Unparsing)
	+ No choice alternative successfully unparsed.
	+ Character set encoding
		- TBD: … and dfdl:encodingErrorPolicy='error'
		- Not enough room for character's encoding
		- No mapping of Infoset character to an encoding
* Implementation Limit Errors - Implementations can have fixed or adjustable limits that some formats and some data may exceed at processing time. This specification does not further specify what these errors are, but some possible examples are:
	+ Data longer than allowed for representation of a given data type
		- e.g., exceed maximum length of representation of xs:decimal in dfdl:representation=”text”.
		- e.g., exceed maximum number of bytes in a xs:string.
	+ Expression references too far back into infoset (parsing)
	+ Expression references too far forward into infoset (unparsing)
	+ Number of array elements exceeds limit.

The following are Schema Definition Errors regardless of whether they are detected in advance of processing or once processing begins.

* Errors in XML Schema Construction and Structure
	+ See XML Schema Specification Section 5.1
* DFDL as subset: use of XSD constructs outside of DFDL subset
* Implementation Limitations
	+ - Use of DFDL schema constructs not supported by this implementation.
* Example: xs:choice is an optional part of the DFDL specification (see section 21). If not supported, it must be rejected as a Schema Definition Error.
	+ - * Example: use of packed-decimal when it is not supported by the implementation.
			* Example: use of dfdl:assert when it is not supported by the implementation (See Spec section 21 on DFDL Subsets)
		- Note: Unrecognized DFDL properties or property values can produce a Schema Definition Warning and an implementation can attempt to process data despite the warning.
		- Exceeding limits of the implementation for schema size/complexity
			* Example: schema too large – simply a limit on how large the schema can be, how many files, how many top-level constructs, etc.
* Schema Not Valid
	+ See XML Schema Specification Section 5.2
* UPA violation (Unique Particle Attribution)
* Reference to DFDL global definition not found
	+ Format definition (dfdl:defineFormat)
	+ Escape schema definition (dfdl:defineEscapeScheme)
	+ Variable Definition (dfdl:defineVariable)
* DFDL Annotations not well-formed or not valid
* DFDL Annotations Incompatible
	+ E.g., dfdl:assert and dfdl:discriminator at same combined annotation point, or more than one format annotation at an annotation point.
* DFDL Properties and their values
	+ Property not applicable to DFDL annotation
	+ Property value not suitable for property
	+ Property conflict
		- Between Element Reference and Element Declaration
		- Between Element Declaration and Simple Type Definition
		- Between Simple Type Definition and Base Simple Type Definition
		- Between Group Reference and Sequence/Choice of Group Definition
	+ Required property not found
* Expressions
	+ Expression syntax error
	+ Named child element doesn’t exist – E.g., /a/b, and there is no child b in existence.
		- Note: no child *possible* in the schema is a different error, but also a Schema Definition Error, as /a/b would not have a type in that case.
		- Note: This is an SDE, as schema authors are advised to use fn:exists(…) to test for existence of elements when it is possible that they not exist.
	+ Variable read but not defined
	+ Variable assigned after read
	+ Variable assigned more than once
	+ Static Type error – type is incorrect for usage
		- Note: if an implementation is unable to distinguish Static Type Errors from Dynamic Type Errors, then both should cause Schema Definition Errors.
	+ Path step definition not found – e.g., /a/n:b but no definition for n:b as local or global element.
	+ Not enough arguments for function
	+ Expression value is not single node
		- Most DFDL expression contexts require an expression to identify a single node, not an array (aka sequence of nodes). There are a few exceptions such as the fn:count(…) function, where the path expression must be to an array or optional element.
	+ Expression value is not array element or optional element.
		- Some DFDL expression contexts require an array or an optional element.
		- Example: The fn:count(...) function argument must be to an array or optional element. It is an SDE if the argument expression is otherwise.
* Regular Expressions
	+ Syntax error