DCMTK - Bug #1079

Specifications of DICOM DS and JSON number not fully compatible

2023-06-12 10:17 - Marco Eichelberg

Status:	Closed	**Start date:**	2023-06-12
Priority:	Normal	**Due date:**	
Assignee:	Marco Eichelberg	**% Done:**	100%
Category:	Library and Apps	**Estimated time:**	2:00 hours
Target version:			
Module:	dcmdata	**Compiler:**	
Operating System:			

Description

dcm2json uses the method DcmJsonFormat::normalizeDecimalString() to normalize DICOM DS values before writing them as JSON. However, the method currently does not catch all cases where a legal DICOM DS value is not permitted in JSON.
The JSON type number is defined in https://www.rfc-editor.org/rfc/rfc7159#section-6 as follows:

```
number = [ minus ] int [ frac ] [ exp ]
      decimal-point = %x2E       ; .
      digit1-9 = %x31-39         ; 1-9
      e = %x65 / %x45            ; e E
      exp = e [ minus / plus ] 1*DIGIT
      frac = decimal-point 1*DIGIT
      int = zero / ( digit1-9 *DIGIT )
      minus = %x2D               ; -
      plus = %x2B                ; +
      zero = %x30                ; 0
```

The definition of DICOM DS relies on the ANSI X3.9 standard, i.e. the specification of FORTRAN. The FORTRAN 'signed-real-literal-constant' is defined (in slightly simplified form) as:

```
R709 kind-param is digit-string or scalar-int-constant-name
R710 signed-digit-string is [ sign ] digit-string
R711 digit-string is digit [ digit ] ...
R712 sign is + or -
R713 signed-real-literal-constant is [ sign ] real-literal-constant
R714 real-literal-constant is significand [ exponent-letter exponent ] or digit-string exponent-le
tter exponent
R715 significand is digit-string . [ digit-string ] or . digit-string
R716 exponent-letter is E or D
R717 exponent is signed-digit-string
```

As a regular expression, DICOM DS values can thus be defined as

```
[+-]?([0-9]+.[0-9]*|.?[0-9]+)(E[+-]?[0-9]+)?
```

Differences between JSON number and DICOM DS are thus:

1. JSON does not permit a leading '+' in the significand (i.e. "+1.0" is valid in DICOM but not in JSON)
2. JSON does not permit leading zeroes in the significand (i.e. "0003" is valid in DICOM but not in JSON)
3. JSON does not permit a leading '.' in the significand (i.e. ".5" is valid in DICOM but not in JSON)
4. JSON does not permit a trailing '.' in the significand (i.e. "5." is valid in DICOM but not in JSON)

DcmJsonFormat::normalizeDecimalString() currently accounts for differences 1 and 3, but not for 2 and 4. This should be fixed.

History

#1 - 2023-08-14 18:57 - Marco Eichelberg

- % Done changed from 0 to 100

- Estimated time set to 2:00 h

- Status changed from New to Closed

- Assignee set to Marco Eichelberg

The previous implementation actually did handle cases 1, 2 and 3. Now we also handle case 4 correctly.
Thanks to Mathieu Malaterre for the report.

Closed by commit #862de6320.