DICOM Correction Proposal

STATUS	Final Text
Date of Last Update	2013/06/13
Person Assigned	David Clunie dclunie@dclunie.com
Submitter Name	John Bush bushjr@battelle.org
Submission Date	2012/08/02

Correction Number CP-1261

Log Summary: Other Double 64 Bit Floating Point Value Representation

Name of Standard

PS 3.5, 3.19 2011

Rationale for Correction:

Sup 49 Enhanced MR introduced the Other Float (OF) Value Representation to encode 32 bit IEEE 754:1985 floating point values in order to support spectroscopy data, but did not provide for a 64 floating point value.

Subsequent experience has shown that there is a need to encode 64 bit floating point bulk data, particularly for use in Raw Data objects, but also potentially for new IODs and SOP Classes, such as are proposed for the DICOS and DICOMDE (and possibly DICOM) CT acquisition data encoding (see, for example, "Regensburger, Mooney, Bush and Schuetter, *Battelle Technical Concept Paper on A Framework for Expanding DICOM, DICONDE, and DICOS Data Formats to the X-ray CT Image Acquisition Interface*, July 2012"). Since it is important to maintain a common infra-structure and toolkit support between DICOM, DICONDE and DICOS, this VR should be added to the base DICOM standard, rather than being a potentially incompatible DICOS or DICONDE extension.

Accordingly, an Other Double (OD) is proposed, identical in all other respects to the existing OF VR, and with the similar changes to PS 3.5 as described in Sup 49.

Since Supplement 14 Unknown VR, toolkit implementers have been made aware of the need to anticipate future addition of new VRs to the standard, and assured that they would all be of the "long" 32-bit Value Length form in Explicit VR Transfer Syntaxes, and so existing toolkits should not fail when encountering this VR. Experience has confirmed that the addition of UT and OF VRs did not cause problems in this respect.

Since Sup 49, the Application Hosting API was added and PS 3.19 created. This includes am XML schema for the representation of the Native Model. Accordingly, the OD VR needs to be added to the XML schema as well.

Correction Wording:

Add OD VR to PS 3.5:

Table 6.2-1
DICOM VALUE REPRESENTATIONS

VR Name	Definition	Character Repertoire	Length of Value
…	…	…	…
OD **Other Double String**	**A string of 64-bit IEEE 754:1985 floating point words. OD is a VR which requires byte swapping within each 64-bit word when changing between Little Endian and Big Endian byte ordering (see Section 7.3).**	**not applicable**	**2^{32}-8 maximum**

…	…	…	…

…

6.4 VALUE MULTIPLICITY (VM) AND DELIMITATION

….

Data Elements with a VR of SQ, OF, **OD,** OW, OB or UN shall always have a Value Multiplicity of one.

…

7.1.2 DATA ELEMENT STRUCTURE WITH EXPLICIT VR

When using the Explicit VR structures, the Data Element shall be constructed of four consecutive fields: Data Element Tag, VR, Value Length, and Value. Depending on the VR of the Data Element, the Data Element will be structured in one of two ways:

- for VRs of OB, OW, OF, **OD,** SQ and UN the 16 bits following the two character VR Field are reserved for use by later versions of the DICOM Standard. These reserved bytes shall be set to 0000H and shall not be used or decoded (Table 7.1-1). The Value Length Field is a 32-bit unsigned integer. If the Value Field has an Explicit Length, then the Value Length Field shall contain a value equal to the length (in bytes) of the Value Field. Otherwise, the Value Field has an Undefined Length and a Sequence Delimitation Item marks the end of the Value Field.
- for VRs of UT the 16 bits following the two character VR Field are reserved for use by later versions of the DICOM Standard. These reserved bytes shall be set to 0000H and shall not be used or decoded. The Value Length Field is a 32-bit unsigned integer. The Value Field is required to have an Explicit Length, that is the Value Length Field shall contain a value equal to the length (in bytes) of the Value Field.

  Note:VRs of UT may not have an Undefined Length, ie. a Value Length of FFFFFFFFH.
- for all other VRs the Value Length Field is the 16-bit unsigned integer following the two character VR Field (Table 7.1-2). The value of the Value Length Field shall equal the length of the Value Field.

Table 7.1-1
DATA ELEMENT WITH EXPLICIT VR OF OB, OW, OF, OD, SQ, UT OR UN

Tag		**VR**		**Value Length**	**Value**
Group Number (16-bit unsigned integer)	Element Number (16-bit unsigned integer)	VR (2 byte character string) of "OB", "OW", “OF”, **“OD”,** “SQ”, “UT” or "UN"	Reserved (2 bytes) set to a value of 0000H	32-bit unsigned integer	Even number of bytes containing the Data Element Value(s) encoded according to the VR and negotiated Transfer Syntax. Delimited with Sequence Delimitation Item if of Undefined Length.
2 bytes	2 bytes	2 bytes	2 bytes	4 bytes	'Value Length' bytes if of Explicit Length

…

7.3 BIG ENDIAN VERSUS LITTLE ENDIAN BYTE ORDERING

…

In the default case of Little Endian encoding, Big Endian Machines interpreting Data Sets shall do 'byte swapping' before interpreting or operating on certain Data Elements. The Data Elements affected are all those having VRs that are multiple byte Values and that are not a character string of 8-bit single byte codes. VRs constructed of a string of characters of 8-bit single byte codes are really constructed of a string of individual bytes, and are therefore not affected by byte ordering. The VRs that are not a string of characters and consist of multiple bytes are:

- 2-byte US, SS, OW and each component of AT
- 4-byte OF, UL, SL, and FL
- 8 byte **OD,** FD

Note: For the above VRs, the multiple bytes are presented in increasing order of significance when in Little Endian format. For example, an 8-byte Data Element with VR of FD, might be written in hexadecimal as 68AF4B2CH, but encoded in Little Endian would be 2C4BAF68H.

Add OD VR to PS 3.19 XML Schema for Native Model:

A.1.6 Schema

…

```
Tag = attribute tag { xsd:string{ minLength="8" maxLength="8" pattern="[0-9A-F]{8}" }
}
VR = attribute vr { "AE" | "AS" | "AT"| "CS" | "DA" | "DS" | "DT" | "FL" | "FD"
                    | "IS" | "LO" | "LT" | "OB" | "OD" | "OF" | "OW" | "PN" | "SH" |
"SL"
                    | "SQ" | "SS" | "ST" | "TM" | "UI" | "UL" | "UN" | "US" | "UT" }
PrivateCreator = attribute privateCreator{ xsd:string }
…
```