Status	Final Text
Date of Last Update	2016/09/08
Person Assigned	David Clunie mailto:dclunie@dclunie.com
Submitter Name	Mathieu Malaterre mailto:mathieu.malaterre@gmail.com
Submission Date	2015/12/18

Correction Number CP-1564
Log Summary: Copying unrecognized VRs
Name of Standard PS3.5 2016c
Rationale for Correction: The standard anticipates an existing implementation encountering unrecognized (new standard) Value Representations and describes how they may be ignored since the parsing rules are defined. However, the standard is silent on how such data elements with unrecognized VRs should be copied, and though this may be deduced from first principles, it would be helpful to describe it.
Correction Wording:

Amend DICOM PS3.5 as follows:

6.2 Value Representation (VR)

The Value Representation of a Data Element describes the data type and format of that Data Element's Value(s). ???? lists the VR of each Data Element by Data Element Tag.

Values with VRs constructed of character strings, except in the case of the VR UI, shall be padded with SPACE characters (20H, in the Default Character Repertoire) when necessary to achieve even length. Values with a VR of UI shall be padded with a single trailing NULL (00H) character when necessary to achieve even length. Values with a VR of OB shall be padded with a single trailing NULL byte value (00H) when necessary to achieve even length.

All new VRs defined in future versions of DICOM shall be of the same Data Element Structure as defined in ??? (i.e., following the format for VRs such as OB, OD, OF, OL, OW, SQ and UN).

Note

1. Since all new VRs will be defined as specified in ???, an implementation may choose to ignore VRs not recognized by applying the rules stated in ???.

2. **When converting a Data Set from an Explicit VR Transfer Syntax to a different Transfer Syntax, an implementation may copy Data Elements with unrecognized VRs in the following manner:**

    - **If the endianness of the Transfer Syntaxes is the same, the Value of the Data Element may be copied unchanged and if the target Transfer Syntax is Explicit VR, the VR bytes copied unchanged. In practice this only applies to Little Endian Transfer Syntaxes, since there was only one Big Endian Transfer Syntax defined.**

    - **If the source Transfer Syntax is Little Endian and the target Transfer Syntax is the (retired) Big Endian Explicit VR Transfer Syntax, then the Value of the Data Element may be copied unchanged and the VR changed to UN, since being unrecognized, whether or not byte swapping is required is unknown. If the VR were copied unchanged, the byte order of the value might or might not be incorrect.**

    - **If the source Transfer Syntax is the (retired) Big Endian Explicit VR Transfer Syntax, then the Data Element cannot be copied, because whether or not byte swapping is required is unknown, and there is no equivalent of the UN VR to use when the value is big endian rather than little endian.**

    The issues of whether or not the element may be copied, and what VR to use if copying, do not arise when converting a Data Set from Implicit VR Little Endian Transfer Syntax, since the VR would not be present to be unrecognized, and if the data element VR is not known from a data dictionary, then UN would be used.

An individual Value, including padding, shall not exceed the Length of Value...