DCMTK - Bug #888

storescp fails parsing UN VR sequence items

2019-06-24 17:54 - Marco Eichelberg

Status:	Closed	**Start date:**	2019-06-24
Priority:	Normal	**Due date:**	
Assignee:	Marco Eichelberg	**% Done:**	100%
Category:	Library and Apps	**Estimated time:**	0:00 hour
Target version:			
Module:	dcmdata	**Compiler:**	
Operating System:			

Description

How to reproduce:

- Run a storescp configured to convert VR=UN encoded attributes to real VR.
- Send a Implicit VR Little Endian encoded image that contains a VR=UN sequence of a known attribute.
- The storescp will fail to receive the image with error messages indicating that it encountered an unknown VR.
- Interestingly dcmconv doesn't display the same issue

Patch to enable recover-VR for storescp:

```
--- a/dcmnet/apps/storescp.cc
+++ b/dcmnet/apps/storescp.cc
@@ -342,7 +342,10 @@ int main(int argc, char *argv[])
     cmd.addSubGroup("data set trailing padding (not with --write-dataset or --bit-preserving):");
       cmd.addOption("--padding-off",            "-p",      "no padding (default)");
       cmd.addOption("--padding-create",         "+p",   2, "[f]ile-pad [i]tem-pad: integer",
-                                                "align file on multiple of f bytes and items\non multiple of i bytes");
+                                                 "align file on multiple of f bytes and items\n
on multiple of i bytes");
+    cmd.addSubGroup("handling of defined length UN elements:");
+      cmd.addOption("--retain-un",           "-uc",    "retain elements as UN (default)");
+      cmd.addOption("--convert-un",          "+uc",    "convert to real VR if known");
 #ifdef WITH_ZLIB
     cmd.addSubGroup("deflate compression level (only with --write-xfer-deflated/same):");
       cmd.addOption("--compression-level",      "+cl",  1, "[l]evel: integer (default: 6)",
@@ -789,6 +792,17 @@ int main(int argc, char *argv[])
     }
     cmd.endOptionBlock();

+    cmd.beginOptionBlock();
+    if (cmd.findOption("--retain-un"))
+    {
+      dcmEnableUnknownVRConversion.set(OFFalse);
+    }
+    if (cmd.findOption("--convert-un"))
+    {
+      dcmEnableUnknownVRConversion.set(OFTrue);
+    }
+    cmd.endOptionBlock();
+
 #ifdef WITH_ZLIB
     if (cmd.findOption("--compression-level"))
     {
```

A patch that seems to fix the bug:

```
--- a/dcmdata/libsrc/dcitem.cc
```

```
+++ b/dcmdata/libsrc/dcitem.cc
@@ -4306,7 +4306,7 @@ OFCondition DcmItem::newDicomElement(DcmElement *&newElement,

         // sequences and items:
         case EVR_SQ :
-            newElement = new DcmSequenceOfItems(tag, length);
+         newElement = new DcmSequenceOfItems(tag, length, readAsUN);
             break;
         case EVR_na :
             if (tag.getXTag() == DCM_Item)
```

Reported 2019-06-21 by Jesper Bojesen <jbojesen@vitalimages.com>

History

#1 - 2019-10-11 10:39 - Jan Schlamelcher

- Target version changed from 3.6.5 to 3.6.6

#2 - 2020-01-05 14:31 - Marco Eichelberg

- File dcmtk_issue_888.zip added

Apparently the issue only occurs when DICOM data is read in multiple "chunks", i.e. when the Sequence that has UN value representation does not fit into a single PDU. Attached a sample file that can be used to demonstrate the issue.

#3 - 2020-01-05 14:47 - Marco Eichelberg

- Category set to Library and Apps

- Status changed from New to Closed

- Assignee set to Marco Eichelberg

- % Done changed from 0 to 100

Closed by commit #4e24811ec.

#4 - 2020-05-25 13:29 - Michael Onken

- Target version deleted (3.6.6)

Files

dcmtk_issue_888.zip	12.9 KB	2020-01-05	Marco Eichelberg