Server - Issues #6796

sSrv2 and sSrv3 ordering of the MMS identifiers

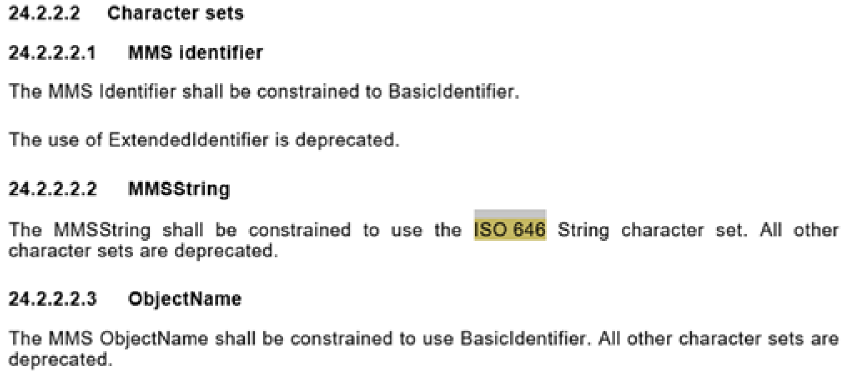
05/28/2024 04:26 AM - Goran Pregrad

Status:	Closed	**Due date:**	07/09/2024
Priority:	Normal		
Assignee:	Goran Pregrad		
Category:			
Target version:			
Discuss in Upcoming Meeting:	No	**Updated Test Document:**	
Clause Reference:		**Test Case ID:**	
61850 Standard:	8-1	**Closed Reason:**	Test Procedure Update
Triggering Tissue:		**Triggering Tissue 2:**	
Final Decision:		**Triggering Tissue 3:**	
Initial Test Document:			

Description

In tests sSrv2 and sSrv3 client requests DUT to return ordered list of "logical nodes" and "data". Return results should be ordered lists of MMS Identifiers. Question that is being raised is what is definition of ordering for returned elements?

Background:
In 8-1 we have following specification:



24.2.2.2 Character sets

24.2.2.2.1 MMS identifier

The MMS Identifier shall be constrained to BasicIdentifier.

The use of ExtendedIdentifier is deprecated.

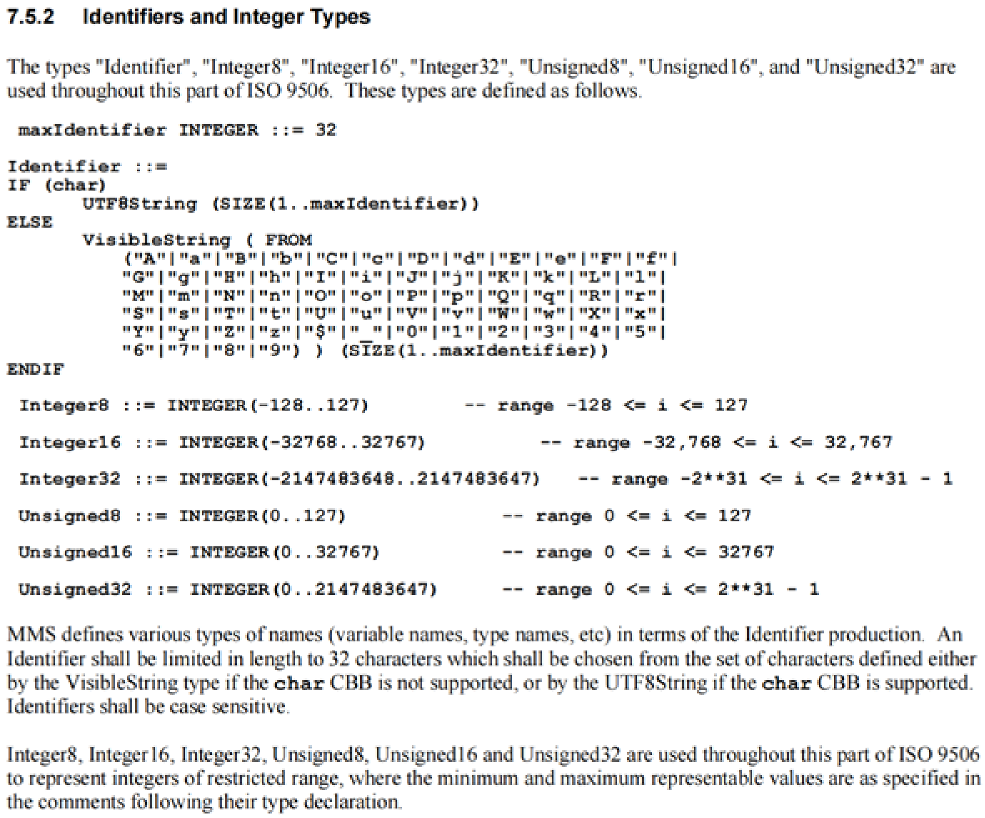
24.2.2.2.2 MMSString

The MMSString shall be constrained to use the ISO 646 String character set. All other character sets are deprecated.

24.2.2.2.3 ObjectName

The MMS ObjectName shall be constrained to use BasicIdentifier. All other character sets are deprecated.

MMS Identifier should be of type BasicIdentifier, however that type does not exist in ISO 9506-2 version 2003 that is referenced. That type is from previous 1999 version of MMS standard. In current valid version it is defined in 7.5.2:



7.5.2 Identifiers and Integer Types

The types "Identifier", "Integer8", "Integer16", "Integer32", "Unsigned8", "Unsigned16", and "Unsigned32" are used throughout this part of ISO 9506. These types are defined as follows.

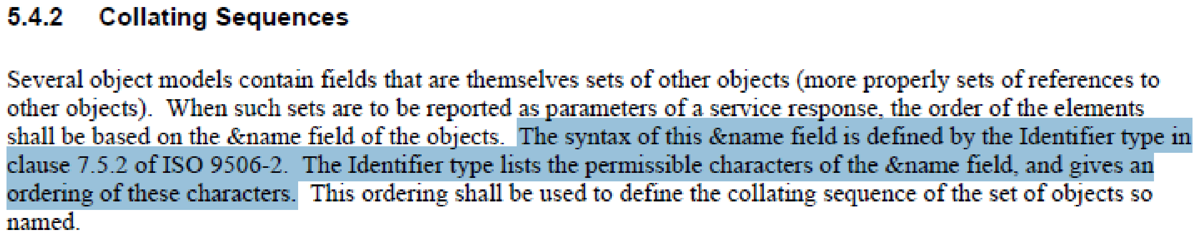
```
 maxIdentifier INTEGER ::= 32

Identifier ::=
IF (char)
       UTF8String (SIZE(1..maxIdentifier))
ELSE
       VisibleString ( FROM
           ("A"|"a"|"B"|"b"|"C"|"c"|"D"|"d"|"E"|"e"|"F"|"f"|
           "G"|"g"|"H"|"h"|"I"|"i"|"J"|"j"|"K"|"k"|"L"|"l"|
           "M"|"m"|"N"|"n"|"O"|"o"|"P"|"p"|"Q"|"q"|"R"|"r"|
           "S"|"s"|"T"|"t"|"U"|"u"|"V"|"v"|"W"|"w"|"X"|"x"|
           "Y"|"y"|"Z"|"z"|"$"|"_"|"0"|"1"|"2"|"3"|"4"|"5"|
           "6"|"7"|"8"|"9") ) (SIZE(1..maxIdentifier))
ENDIF

 Integer8 ::= INTEGER(-128..127)         -- range -128 <= i <= 127

 Integer16 ::= INTEGER(-32768..32767)          -- range -32,768 <= i <= 32,767

 Integer32 ::= INTEGER(-2147483648..2147483647)    -- range -2**31 <= i <= 2**31 - 1

 Unsigned8 ::= INTEGER(0..127)             -- range 0 <= i <= 127

 Unsigned16 ::= INTEGER(0..32767)          -- range 0 <= i <= 32767

 Unsigned32 ::= INTEGER(0..2147483647)     -- range 0 <= i <= 2**31 - 1
```

MMS defines various types of names (variable names, type names, etc) in terms of the Identifier production. An Identifier shall be limited in length to 32 characters which shall be chosen from the set of characters defined either by the VisibleString type if the **char** CBB is not supported, or by the UTF8String if the **char** CBB is supported. Identifiers shall be case sensitive.

Integer8, Integer16, Integer32, Unsigned8, Unsigned16 and Unsigned32 are used throughout this part of ISO 9506 to represent integers of restricted range, where the minimum and maximum representable values are as specified in the comments following their type declaration.

In IEC 61850 8-1 (Ed2Amd1 only one with explicit declaration) CBB char is not used, so based on that Identifier should be of type VisibleString. This VisibleString is same as in 1999 version BasicIdentifier.

Now in ISO 9506-1 there is sentence below in blue:



5.4.2 Collating Sequences

Several object models contain fields that are themselves sets of other objects (more properly sets of references to other objects). When such sets are to be reported as parameters of a service response, the order of the elements shall be based on the &name field of the objects. The syntax of this &name field is defined by the Identifier type in clause 7.5.2 of ISO 9506-2. The Identifier type lists the permissible characters of the &name field, and gives an ordering of these characters. This ordering shall be used to define the collating sequence of the set of objects so named.

Base on it I would think that if UTF-8 is used than it is using its ASCII alike ordering. And if VisibleString is used than ordering per 7.5.2 should be used.
Difference would be following:
List = “Aa”, “AH”, “1A”
OrderedListUTF8 = “1A”, “AH”, “Aa”
OrderedListVisibleString = “Aa”, “AH”, “1A”

We found different vendors implementing different ordering rules. In my opinion ordering should be done per provided order of characters in VisibleString of Identifiers. Would you agree?
Additionally, I think 8-1 reference of BasicIdentifier should be changed to Identifier:VisibleString.

History

#1 - 05/28/2024 08:54 AM - IEC 61850 TPWG

- Due date set to 06/11/2024

- Status changed from New to In Progress

- Assignee set to Goran Pregrad

A server has to have a specific method of ordering, but that method is not standardized.

Remove "order" from the test procedure.

#2 - 06/11/2024 08:05 AM - IEC 61850 TPWG

- Due date changed from 06/11/2024 to 07/09/2024

#3 - 06/11/2024 08:15 AM - Goran Pregrad

- File Solution to redmine 6796.docx added

Proposed solution is attached.

#4 - 06/11/2024 08:36 AM - IEC 61850 TPWG

- Status changed from In Progress to Resolved

#5 - 06/11/2024 09:30 AM - Richard Schimmel

- Status changed from Resolved to Closed

- Closed Reason Test Procedure Update added

- Closed Reason deleted (--Not Set---)

Files

MMSIdentifier.png	187 KB	05/28/2024	Goran Pregrad
CollatingSequences.png	23 KB	05/28/2024	Goran Pregrad
CharacterSets.png	33.2 KB	05/28/2024	Goran Pregrad
Solution to redmine 6796.docx	21.7 KB	06/11/2024	Goran Pregrad