Network Working Group
Request for Comments: 4792
Updates: 3641
Category: Standards Track

S. Legg
eB2Bcom

January 2007

Encoding Instructions for the Generic String Encoding Rules (GSER)

Status of This Memo

Copyright Notice

Abstract

Abstract Syntax Notation One (ASN.1) defines a general framework for annotating types in an ASN.1 specification with encoding instructions that alter how values of those types are encoded according to ASN.1 encoding rules. This document defines the supporting notation for encoding instructions that apply to the Generic String Encoding Rules (GSER) and, in particular, defines an encoding instruction to provide a machine-processable representation for the declaration of a GSER ChoiceOfStrings type.

Table of Contents

1. Introduction

Abstract Syntax Notation One (ASN.1) [X.680] defines a general framework for annotating types in an ASN.1 specification with encoding instructions [X.680-1] that alter how values of those types are encoded according to ASN.1 encoding rules. This document defines the supporting notation for encoding instructions that apply to the Generic String Encoding Rules (GSER) [GSER], and in particular defines an encoding instruction, the CHOICE-OF-STRINGS encoding instruction, to provide a machine-processable representation for the declaration of a GSER ChoiceOfStrings type.

The CHOICE-OF-STRINGS encoding instruction SHOULD be used instead of simply declaring a ChoiceOfStrings type.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [BCP14].

Throughout this document, "type" shall be taken to mean an ASN.1 type, and "value" shall be taken to mean an ASN.1 abstract value, unless qualified otherwise.

A reference to an ASN.1 production [X.680] (e.g., Type, NamedType) is a reference to text in an ASN.1 specification corresponding to that production.

3. Notation for GSER Encoding Instructions

The grammar of ASN.1 permits the application of encoding instructions [X.680-1], through type prefixes and encoding control sections, that modify how abstract values are encoded by nominated encoding rules.

The generic notation for type prefixes and encoding control sections is defined by the ASN.1 basic notation [X.680] [X.680-1], and includes an encoding reference to identify the specific encoding rules that are affected by the encoding instruction.

The encoding reference that identifies the Generic String Encoding Rules is literally GSER.

The specific notation for an encoding instruction for a particular set of encoding rules is left to the specification of those encoding rules. Consequently, this companion document to the GSER specification [GSER] defines the notation for GSER encoding

instructions. Specifically, it elaborates the EncodingInstruction and EncodingInstructionAssignmentList placeholder productions of the ASN.1 basic notation.

In the context of the GSER encoding reference the EncodingInstruction production is defined as follows, using the conventions of the ASN.1 basic notation:

```
EncodingInstruction ::=
    ChoiceOfStringsInstruction
```

In the context of the GSER encoding reference the EncodingInstructionAssignmentList production (which only appears in an encoding control section) is empty:

```
EncodingInstructionAssignmentList ::= empty
```

4. The CHOICE-OF-STRINGS Encoding Instruction

The CHOICE-OF-STRINGS encoding instruction allows a GSER encoder to encode the alternative of a CHOICE (of restricted string types) without the leading identifier. The optional PrecedenceList also allows a specification writer to alter the order in which a GSER decoder will consider the alternatives of the CHOICE as it determines which alternative has been encoded when the identifier is absent.

The notation for a CHOICE-OF-STRINGS encoding instruction is defined as follows:

```
UnionInstruction ::= "CHOICE-OF-STRINGS" AlternativesPrecedence ?

AlternativesPrecedence ::= "PRECEDENCE" PrecedenceList

PrecedenceList ::= identifier PrecedenceList ?
```

The Type in the EncodingPrefixedType for a CHOICE-OF-STRINGS encoding instruction SHALL be:

(a) a BuiltinType that is a ChoiceType, or

(b) a ConstrainedType that is not a TypeWithConstraint where the Type in the ConstrainedType is one of (a) to (d), or

(c) a BuiltinType that is a PrefixedType that is a TaggedType where the Type in the TaggedType is one of (a) to (d), or

(d) a BuiltinType that is a PrefixedType that is an EncodingPrefixedType where the Type in the EncodingPrefixedType is one of (a) to (d).

The effect of this condition is to force the CHOICE-OF-STRINGS encoding instruction to be textually co-located with the CHOICE type definition to which it applies. This makes it clear to a reader that the encoding instruction applies to every use of the CHOICE type no matter how it might be referenced.

The ChoiceType in case (a) is said to be "subject to" the CHOICE-OF-STRINGS encoding instruction.

The Type of each NamedType of the ChoiceType in case (a) MUST be:

(1) the NumericString, PrintableString, TeletexString (T61String), VideotexString, IA5String, GraphicString, VisibleString (ISO646String), GeneralString, BMPString, UniversalString, or UTF8String type, or

(2) a type notation that references a type that is one of (1) to (4), or

(3) a constrained type where the type that is constrained is one of (1) to (4), or

(4) a prefixed type where the type that is prefixed is one of (1) to (4).

ASIDE: A tagged type is a special case of a prefixed type. An effect of case (4) is that tagging is not significant.

The ASN.1 restricted string type in case (1) MUST be different for each NamedType in the ChoiceType, i.e., no two alternatives have the same restricted string type.

If case (3) applies to any NamedType, then the constraint in case (3) MUST be the same for each NamedType, i.e., either none of the alternatives has a constraint, or all of the alternatives have exactly the same constraint.

Each identifier in the PrecedenceList MUST be the identifier of a NamedType of the ChoiceType.

A particular identifier SHALL NOT appear more than once in the same PrecedenceList.

4.1. Effect on GSER Encodings

A value of a CHOICE type is encoded according to the <ChoiceValue> [GSER] Augmented Backus-Naur Form [ABNF] rule. The ABNF for <ChoiceValue> is reproduced here for convenience:

```
ChoiceValue           = IdentifiedChoiceValue /
                        ChoiceOfStringsValue

IdentifiedChoiceValue = identifier ":" Value
ChoiceOfStringsValue  = StringValue
```

The <IdentifiedChoiceValue> rule MUST be used to encode values of a CHOICE type where the ChoiceType is not subject to a CHOICE-OF-STRINGS encoding instruction.

The chosen alternative of a value of a CHOICE type corresponds to some NamedType in the definition of the type. The <identifier> in the <IdentifiedChoiceValue> is the identifier of this NamedType.

Either the <IdentifiedChoiceValue> rule or the <ChoiceOfStringsValue> rule is used to encode values of a CHOICE type where the ChoiceType is subject to a CHOICE-OF-STRINGS encoding instruction.

If <ChoiceOfStringsValue> has been used, then a GSER decoder MUST determine the chosen alternative by considering the alternatives of the CHOICE in the order prescribed below and accepting the first alternative that allows all of the characters in the <StringValue>.

If the CHOICE-OF-STRINGS encoding instruction has a PrecedenceList, then the alternatives of the ChoiceType referenced by the PrecedenceList are considered in the order identified by that PrecedenceList, and then the remaining alternatives are considered in the order of their definition in the ChoiceType. If the CHOICE-OF-STRINGS encoding instruction does not have a PrecedenceList, then all the alternatives of the ChoiceType are considered in the order of their definition in the ChoiceType.

A GSER encoder MUST use <IdentifiedChoiceValue> if a GSER decoder would determine the chosen alternative to be something other than the chosen alternative of the CHOICE value being encoded; otherwise, <ChoiceOfStringsValue> MAY be used.

Example

Consider this type definition:

```
[GSER:CHOICE-OF-STRINGS PRECEDENCE basicName] CHOICE {
    extendedName  UTF8String,
    basicName     PrintableString
}
```

If a <ChoiceOfStringsValue> has been used, then a GSER decoder would first consider whether the <StringValue> was a valid basicName (a PrintableString) before considering whether it was a valid extendedName (a UTF8String).

4.2. Replacement of Existing ChoiceOfStrings Declarations

In line with the previous declaration [GSER] of the DirectoryString type as a ChoiceOfStrings type, applications using GSER MUST add this encoding instruction:

```
[GSER:CHOICE-OF-STRINGS PRECEDENCE printableString uTF8String]
```

immediately before the "CHOICE" keyword in the definition of the DirectoryString type in the third and every subsequent edition of the SelectedAttributeTypes ASN.1 module of X.520 [X.520-3] [X.520-4] [X.520-5].

For example, this is how the DirectoryString definition would appear in the third, fourth and fifth editions:

```
DirectoryString{INTEGER:maxSize} ::=
[GSER:CHOICE-OF-STRINGS PRECEDENCE printableString uTF8String]
CHOICE {
    teletexString     TeletexString(SIZE (1..maxSize)),
    printableString   PrintableString(SIZE (1..maxSize)),
    universalString   UniversalString(SIZE (1..maxSize)),
    bmpString         BMPString(SIZE (1..maxSize)),
    uTF8String        UTF8String(SIZE (1..maxSize))
}
```

The uTF8String alternative did not appear in the second edition of the SelectedAttributeTypes ASN.1 module of X.520 [X.520-2]. For compatibility, applications using GSER with the second edition of X.520 MUST add this encoding instruction:

```
[GSER:CHOICE-OF-STRINGS PRECEDENCE printableString]
```

immediately before the "CHOICE" keyword in the definition of the DirectoryString type.

For example, this is how the DirectoryString definition would appear in the second edition:

```
DirectoryString{INTEGER:maxSize} ::=
[GSER:CHOICE-OF-STRINGS PRECEDENCE printableString]
CHOICE {
    teletexString     TeletexString(SIZE (1..maxSize)),
    printableString   PrintableString(SIZE (1..maxSize)),
    universalString   UniversalString(SIZE (1..maxSize))
}
```

5. Security Considerations

This specification changes the manner in which ChoiceOfStrings types are declared but does not alter the existing behaviour of GSER implementations. The security considerations for GSER are unchanged (see [GSER]).

6. Normative References

[BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[GSER] Legg, S., "Generic String Encoding Rules (GSER) for ASN.1 Types", RFC 3641, October 2003.

[ABNF] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[X.520-2] ITU-T Recommendation X.520 (1993) | ISO/IEC 9594-6:1994, Information Technology - Open Systems Interconnection - The Directory: Selected attribute types

[X.520-3] ITU-T Recommendation X.520 (08/97) | ISO/IEC 9594-6:1998, Information Technology - Open Systems Interconnection - The Directory: Selected attribute types

[X.520-4] ITU-T Recommendation X.520 (02/01) | ISO/IEC 9594-6:2001, Information technology - Open Systems Interconnection - The Directory: Selected attribute types

[X.520-5] ITU-T Recommendation X.520 (08/05) | ISO/IEC 9594-6:2005, Information technology - Open Systems Interconnection - The Directory: Selected attribute types

[X.680] ITU-T Recommendation X.680 (07/02) | ISO/IEC 8824-1, Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation.

[X.680-1] ITU-T Recommendation X.680 (2002) Amendment 1 (10/03) | ISO/IEC 8824-1:2002/Amd 1:2004, Support for EXTENDED-XER.

Author's Address

Dr. Steven Legg
eB2Bcom
Suite 3, Woodhouse Corporate Centre
935 Station Street
Box Hill North, Victoria 3129
AUSTRALIA

Phone: +61 3 9896 7830
Fax: +61 3 9896 7801
EMail: steven.legg@eb2bcom.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.