Network Working Group   S. Legg
Request for Comments: 4914   eB2Bcom
Category: Experimental   July 2007

            Abstract Syntax Notation X (ASN.X) Representation
                       of Encoding Instructions for
                       the XML Encoding Rules (XER)

Status of This Memo

   This memo defines an Experimental Protocol for the Internet
   community. It does not specify an Internet standard of any kind.
   Discussion and suggestions for improvement are requested.
   Distribution of this memo is unlimited.

Copyright Notice

Abstract

   Abstract Syntax Notation X (ASN.X) is an Extensible Markup Language
   (XML) representation for Abstract Syntax Notation One (ASN.1)
   specifications. This document specifies the ASN.X representation of
   encoding instructions for the XML Encoding Rules (XER).

Table of Contents

1. Introduction

    Abstract Syntax Notation X (ASN.X) [ASN.X] is an Extensible Markup
    Language (XML) [XML10] [XML11] representation for Abstract Syntax
    Notation One (ASN.1) [X.680] specifications. The ASN.X
    representation for the ASN.1 basic notation [X.680] [X.680-1] is
    described elsewhere [ASN.X].

    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; however, the notation for
    specific encoding instructions, i.e., the EncodingInstruction and
    EncodingInstructionAssignmentList productions of the notation, are
    defined separately for each set of encoding rules using encoding
    instructions. This document specifies the ASN.X representation for
    EncodingInstructionAssignmentList and EncodingInstruction as they are
    defined for the XML Encoding Rules (XER) [X.693][X.693-1].

    ASN.X is defined in terms of rules for translating from an ASN.1
    specification. This does not preclude an ASN.X module being written
    directly without a pre-existing ASN.1 module; however, such an ASN.X
    module is considered valid if and only if there exists, in principle,
    an ASN.1 module that when translated would yield the ASN.X module.

    By design, an ASN.X module is also the Robust XML Encoding Rules
    (RXER) [RXER] encoding of an ASN.1 value. The ASN.1 type definitions
    for such values, insofar as they pertain to the ASN.1 basic notation,
    are provided elsewhere [ASN.X]; however, this document provides the
    ASN.1 type definitions for representing XER encoding instructions as
    abstract values. These definitions appear in Appendix A.

    The ASN.X translation of the ASN.1 module in Appendix A is presented
    in Appendix B.

    The General Considerations of the specification for ASN.X [ASN.X]
    also apply here.

2. Conventions

    The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
    "SHOULD", "SHOULD NOT", "RECOMMENDED", and "MAY" in this document are
    to be interpreted as described in BCP 14, RFC 2119 [BCP14]. The key
    word "OPTIONAL" is exclusively used with its ASN.1 meaning.

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

The description of the translation of XER encoding instructions into
ASN.X makes use of definitions from the XML Information Set (Infoset)
[INFOSET]. In particular, information item property names follow the
Infoset convention of being shown in square brackets, e.g.,
[local name]. Literal values of Infoset properties are enclosed in
double quotes; however, the double quotes are not part of the
property values. In the sections that follow, "information item"
will be abbreviated to "item", e.g., "element information item" is
abbreviated to "element item". Element items will be referred to by
their [local name] in angle brackets, e.g., "the <name> element item"
means the element item with the [local name] "name". Attribute items
will be referred to by their [local name], e.g., "the name attribute
item" means the attribute item with the [local name] "name".

Code points for characters [UNICODE] are expressed using the Unicode
convention U+n, where n is four to six hexadecimal digits, e.g., the
space character is U+0020.

3. Preprocessing of ImportedTypesIdentification

Interpretation of an ImportedTypesIdentification depends on a
SymbolList in the Imports of a module, but a SymbolList does not have
a direct translation into ASN.X. A module containing at least one
ImportedTypesIdentification must be preprocessed so that each
ImportedTypesIdentification has the intended effect.

If an EncodingInstruction in an XER EncodingInstructionAssignmentList
has a Targets instance that is an ImportedTypesIdentification, then
each Type affected by the ImportedTypesIdentification MUST be
replaced by an EncodingPrefixedType where the Type in the
EncodingPrefixedType is the original Type and the EncodingPrefix
contains the EncodingInstruction minus the TargetList. The
ImportedTypesIdentification SHALL be removed. If the
EncodingInstruction in the EncodingInstructionAssignmentList has no
other Targets instances, then it MUST be removed from the
EncodingInstructionAssignmentList.

4. EncodingInstructionAssignmentList Translation

As described in the specification for ASN.X [ASN.X], the translation
of an EncodingControlSection for XER is an element item with the
[local name] "XER". The translation of the
EncodingInstructionAssignmentList in such an EncodingControlSection
determines the content of the <XER> element item.

An element item with the [local name] "annotation" MAY be added to
the [children] of the <XER> element item.

The translation of each EncodingInstruction nested in the
EncodingInstructionAssignmentList SHALL be appended to the [children]
of the <XER> element item.

5. EncodingInstruction Translation

The translation of an EncodingInstruction for XER can appear in the
translation of an EncodingInstructionAssignmentList (see the previous
section) or an EncodingPrefix. The translation for an EncodingPrefix
is described by the specification for ASN.X [ASN.X].

The translation of an EncodingInstruction in an
EncodingInstructionAssignmentList for XER is an element item with the
[local name] "targettedInstruction". The translation of the
PositiveInstruction or NegatingInstruction in the EncodingInstruction
SHALL be added to the [children] of the <targettedInstruction>
element item. The translation of the TargetList nested in the
encoding instruction SHALL be appended to the [children] of the
<targettedInstruction> element item.

    Aside: The TargetList appears within an EncodingInstruction in an
    EncodingInstructionAssignmentList, but its ASN.X translation
    appears immediately after the translation of the
    EncodingInstruction, rather than within it.

Example

    ENCODING-CONTROL XER
        GLOBAL-DEFAULTS MODIFIED-ENCODINGS
        ATTRIBUTE MyType

    <encodingControls>
     <XER>
      <targettedInstruction>
       <globalDefaults>
        <modifiedEncodings/>
       </globalDefaults>
      </targettedInstruction>
      <targettedInstruction>
       <attribute/>
       <target type="MyType"/>
      </targettedInstruction>
     </XER>
    </encodingControls>

The translation of an EncodingInstruction for XER in an
EncodingPrefix is the translation of the PositiveInstruction or
NegatingInstruction in the EncodingInstruction.

> Aside: The TargetList within an EncodingInstruction in an
> EncodingPrefix is required to be empty and has no ASN.X
> translation in this context.

The translation of a PositiveInstruction is the translation of the
AnyAttributeInstruction, AnyElementInstruction, AttributeInstruction,
Base64Instruction, DecimalInstruction, DefaultForEmptyInstruction,
EmbedValuesInstruction, GlobalDefaultsInstruction, ListInstruction,
NameInstruction, NamespaceInstruction, PIOrCommentInstruction,
TextInstruction, UntaggedInstruction, UseNilInstruction,
UseNumberInstruction, UseOrderInstruction, UseQNameInstruction,
UseTypeInstruction, UseUnionInstruction or WhitespaceInstruction in
the PositiveInstruction.

The translation of a NegatingInstruction that is an
ElementInstruction is the translation of that ElementInstruction.

The translation of a NegatingInstruction that is not an
ElementInstruction is an element item where the [local name] is the
concatenation of "not-" and the [local name] that would be used by
the translation of the PositiveInstruction in the
NegatingInstruction. No items are added to the [attributes] or
[children] of the element item resulting from the translation of a
NegatingInstruction.

Example

    [XER:NOT ATTRIBUTE] MyType

    <type>
     <prefixed>
      <XER><not-attribute/></XER>
      <type ref="MyType"/>
     </prefixed>
    </type>

5.1. AnyAttributesInstruction Translation

The translation of an AnyAttributesInstruction is an element item
with the [local name] "anyAttributes". If a NamespaceRestriction is
present in the AnyAttributesInstruction, then the translation of the
NamespaceRestriction SHALL be added to the [children] of the
<anyAttributes> element item.

Example

```
ANY-ATTRIBUTES FROM "http://example.com" ABSENT

<anyAttributes>
 <namespace>http://example.com</namespace>
 <local/>
</anyAttributes>
```

5.1.1. NamespaceRestriction Translation

The NamespaceRestriction production is common to the
AnyAttributesInstruction production and the AnyElementInstruction
production.

The translation of a NamespaceRestriction of the "FROM URIList" form
is an element item with the [local name] "from". The translation of
each QuotedURIorAbsent nested in the URIList SHALL be appended to the
[children] of the <from> element item.

The translation of a NamespaceRestriction of the "EXCEPT URIList"
form is an element item with the [local name] "except". The
translation of each QuotedURIorAbsent nested in the URIList SHALL be
appended to the [children] of the <except> element item.

The translation of a QuotedURIorAbsent that is a QuotedURI is an
element item with the [local name] "namespace". The sequence of
character items for the URI in the QuotedURI is added to the
[children] of the <namespace> element item.

The translation of a QuotedURIorAbsent of the "ABSENT" form is an
element item with the [local name] "local".

5.2. AnyElementInstruction Translation

The translation of an AnyElementInstruction is an element item with
the [local name] "anyElement". If a NamespaceRestriction is present
in the AnyElementInstruction, then the translation of the
NamespaceRestriction SHALL be added to the [children] of the
<anyElement> element item.

5.3. AttributeInstruction Translation

The translation of an AttributeInstruction is an element item with
the [local name] "attribute".

   Example

      ATTRIBUTE

      <attribute/>

5.4. Base64Instruction Translation

   The translation of a Base64Instruction is an element item with the
   [local name] "base64".

5.5. DecimalInstruction Translation

   The translation of a DecimalInstruction is an element item with the
   [local name] "decimal".

5.6. DefaultForEmptyInstruction Translation

   The translation of a DefaultForEmptyInstruction is an element item
   with the [local name] "defaultForEmpty". The translation of the
   Value in the DefaultForEmptyInstruction SHALL be added to the
   [children] or [attributes] of the <defaultForEmpty> element item.

   Example

      DEFAULT-FOR-EMPTY AS "unspecified"

      <defaultForEmpty literalValue="unspecified"/>

5.7. ElementInstruction Translation

   The translation of an ElementInstruction is an element item with the
   [local name] "element".

5.8. EmbedValuesInstruction Translation

   The translation of an EmbedValuesInstruction is an element item with
   the [local name] "embedValues".

5.9. GlobalDefaultsInstruction Translation

   The translation of a GlobalDefaultsInstruction is an element item
   with the [local name] "globalDefaults".

   If the DefaultSetting in the GlobalDefaultsInstruction is "MODIFIED-
   ENCODINGS", then an element item with the [local name]
   "modifiedEncodings" SHALL be added to the [children] of the
   <globalDefaults> element item.

If the DefaultSetting in the GlobalDefaultsInstruction is a
ControlNamespace, then the translation of the ControlNamespace SHALL
be added to the [children] of the <globalDefaults> element item.

The translation of a ControlNamespace is an element item with the
[local name] "controlNamespace". An attribute item with the
[local name] "name" SHALL be added to the [attributes] of the
<controlNamespace> element item. The [normalized value] of this
attribute item is the Uniform Resource Identifier (URI) [URI] in the
QuotedURI in the ControlNamespace. If the ControlNamespace has a
Prefix, then an attribute item with the [local name] "prefix" SHALL
be added to the [attributes] of the <controlNamespace> element item.
The [normalized value] of this attribute item is the NCName in the
QuotedNCName in the Prefix.

Example

```
   GLOBAL-DEFAULTS
       CONTROL-NAMESPACE "http://example.com" PREFIX "ex"

   <globalDefaults>
    <controlNamespace name="http://example.com" prefix="ex"/>
   </globalDefaults>
```

5.10. ListInstruction Translation

The translation of a ListInstruction is an element item with the
[local name] "list".

5.11. NameInstruction Translation

The translation of a NameInstruction is an element item with the
[local name] "name".

If the NewNameOrKeyword in the NameInstruction is a NewName, then an
attribute item with the [local name] "newName" SHALL be added to the
[attributes] of the <name> element item. The [normalized value] of
this attribute item is the character string value of the
RestrictedCharacterStringValue in the NewName.

If the NewNameOrKeyword in the NameInstruction is a Keyword, then an
attribute item with the [local name] "conversion" SHALL be added to
the [attributes] of the <name> element item. The [normalized value]
of this attribute item is the word in the Keyword with all letters
downcased, i.e., "capitalized", "uncapitalized", "uppercased", or
"lowercased".

Example

    NAME AS UNCAPITALIZED

    <name conversion="uncapitalized"/>

    NAME AS "category"

    <name newName="category"/>

5.12. NamespaceInstruction Translation

   The translation of a NamespaceInstruction is an element item with the
   [local name] "namespace".

   If a NamespaceSpecification is present in the NamespaceInstruction,
   then an attribute item with the [local name] "name" SHALL be added to
   the [attributes] of the <namespace> element item. The
   [normalized value] of this attribute item is the URI in the QuotedURI
   in the NamespaceSpecification.

   If a Prefix is present in the NamespaceSpecification, then an
   attribute item with the [local name] "prefix" SHALL be added to the
   [attributes] of the <namespace> element item. The [normalized value]
   of this attribute item is the NCName in the QuotedNCName in the
   Prefix.

   Example

    NAMESPACE AS "http://example.com" PREFIX "ex"

    <namespace name="http://example.com" prefix="ex"/>

5.13. PIOrCommentInstruction Translation

   The translation of a PIOrCommentInstruction is an element item with
   the [local name] "piOrComment".

   An attribute item with the [local name] "text" SHALL be added to the
   [attributes] of the <piOrComment> element item. The
   [normalized value] of this attribute item is the character string
   value of the RestrictedCharacterStringValue in the
   PIOrCommentInstruction.

   An attribute item with the [local name] "position" SHALL be added to
   the [attributes] of the <piOrComment> element item. The
   [normalized value] of this attribute item is "beforeTag" if the
   keyword in the Position in the PIOrCommentInstruction is BEFORE-TAG,

"beforeValue" if the keyword in the Position is BEFORE-VALUE,
"afterValue" if the keyword in the Position is AFTER-VALUE, or
"afterTag" if the keyword in the Position is AFTER-TAG.

    Example

       PI-OR-COMMENT AS "<!-- This is a comment. -->" BEFORE-TAG

       <piOrComment text="<!-- This is a comment. -->"
                    position="beforeTag"/>

5.14. TextInstruction Translation

   The translation of a TextInstruction is an element item with the
   [local name] "text".

   If a TextToBeUsed is present in the TextInstruction and the
   NewNameOrKeyword in the TextToBeUsed is a NewName, then an attribute
   item with the [local name] "newName" SHALL be added to the
   [attributes] of the <text> element item. The [normalized value] of
   this attribute item is the character string value of the
   RestrictedCharacterStringValue in the NewName.

   If a TextToBeUsed is present in the TextInstruction and the
   NewNameOrKeyword in the TextToBeUsed is a Keyword, then an attribute
   item with the [local name] "conversion" SHALL be added to the
   [attributes] of the <name> element item. The [normalized value] of
   this attribute item is the word in the Keyword with all letters
   downcased, i.e., "capitalized", "uncapitalized", "uppercased", or
   "lowercased".

    Examples

       TEXT AS UPPERCASED

       <text conversion="uppercased"/>

       TEXT AS "A4"

       <text newName="A4"/>

5.15. UntaggedInstruction Translation

   The translation of an UntaggedInstruction is an element item with the
   [local name] "untagged".

5.16. UseNilInstruction Translation

   The translation of a UseNilInstruction is an element item with the
   [local name] "useNil".

5.17. UseNumberInstruction Translation

   The translation of a UseNumberInstruction is an element item with the
   [local name] "useNumber".

5.18. UseOrderInstruction Translation

   The translation of a UseOrderInstruction is an element item with the
   [local name] "useOrder".

5.19. UseQNameInstruction Translation

   The translation of a UseQNameInstruction is an element item with the
   [local name] "useQName".

5.20. UseTypeInstruction Translation

   The translation of a UseTypeInstruction is an element item with the
   [local name] "useType".

5.21. UseUnionInstruction Translation

   The translation of a UseUnionInstruction is an element item with the
   [local name] "useUnion".

5.22. WhiteSpaceInstruction Translation

   The translation of a WhiteSpaceInstruction is an element item with
   the [local name] "whiteSpace".

   An attribute item with the [local name] "action" SHALL be added to
   the [attributes] of the <whiteSpace> element item. The
   [normalized value] of this attribute item is the word in the
   WhiteSpaceAction in the WhiteSpaceInstruction with all letters
   downcased, i.e., "replace" or "collapse".

   Example

      WHITESPACE COLLAPSE

      <whiteSpace action="collapse"/>

6. TargetList Translation

   The TargetList production is common to all the encoding instructions.

   The translation of a TargetList is the concatenation of the
   translations of each Targets instance in the TargetList.

   The translation of a Targets instance is an element item with the
   [local name] "target". The translation of the TypeIdentification,
   BuiltInTypeIdentification or IdentifiersInContext in the Targets
   instance SHALL be added to the [children] and/or [attributes] of the
   <target> element item.

      Aside: A Targets instance that is an ImportedTypesIdentification
      will have been removed by the preprocessing described in
      Section 3.

6.1. TypeIdentification Translation

   The translation of a TypeIdentification of the "ALL" form is an
   element item with the [local name] "allTypes"; otherwise, the
   translation of a TypeIdentification is the translation of its
   ModuleAndTypeReference followed by the translation of its
   ComponentReference, if present, followed by the translation of its
   QualifyingInformationPart, if present.

   The translation of a ModuleAndTypeReference is an attribute item with
   the [local name] "type". The [normalized value] of this attribute
   item is a qualified name [XMLNS10] for the expanded name
   corresponding to the typereference in the ModuleAndTypeReference (see
   Section 5.1 of the specification for ASN.X [ASN.X]).

   The translation of a ComponentReference is an element item with the
   [local name] "component". The [children] property of the <component>
   element item is set to the sequence of character items for a solidus
   ('/', U+002F) separated list of the translations of the ComponentId
   instances in the ComponentIdList in the ComponentReference, excluding
   "ALL" if present. Leading and/or trailing white space character
   items [ASN.X] MAY be added to the [children] of the <component>
   element item. White space character items MAY be added immediately
   before and/or after any character item for the solidus character
   ('/', U+002F). If the final ComponentId in the ComponentIdList in
   the ComponentReference is "ALL", then an element item with the
   [local name] "allTextuallyPresent" SHALL follow the <component>
   element item.

If a ComponentId identifies the Type in a NamedType, then the
translation of the ComponentId is a qualified name for the expanded
name [XMLNS10] of the NamedType [RXEREI]. If the NamedType is
subject to an RXER ATTRIBUTE or ATTRIBUTE-REF encoding instruction
[RXEREI], or subject to an RXER COMPONENT-REF encoding instruction
that references a top-level NamedType that is subject to an RXER
ATTRIBUTE encoding instruction, then the qualified name is prefixed
with the commercial at character ('@', U+0040).

If a ComponentId identifies a Type that is not in a NamedType (i.e.,
the component type for a SEQUENCE OF Type or SET OF Type), then the
translation of the ComponentId is the character string "item".

If a ComponentId does not identify a Type and the ComponentId is an
identifier, then the translation of the ComponentId is the
identifier.

If a ComponentId does not identify a Type and the ComponentId is "*",
then the translation of the ComponentId is the character string
"item".

The <component> element item is required to be self-contained [RXER].

    Aside: An element item is self-contained if all namespace prefixes
    used by the element item and its contents are declared within the
    element item.

The translation of a QualifyingInformationPart is the translation of
the QualifyingInformation instance in the QualifyingInformationPart.

The translation of a QualifyingInformation instance of the
"identifier" form is an element item with the [local name]
"identifier". An attribute item with the [local name] "name" SHALL
be added to the [attributes] of the <identifier> element item. If
the ModuleTypeAndReference and ComponentReference (if present) in the
TypeIdentification identify a BIT STRING, ENUMERATED, or INTEGER type
where the associated BitStringType, EnumeratedType, or IntegerType
(respectively) is subject to an RXER VALUES encoding instruction,
then the [normalized value] of the attribute item is the replacement
name [RXEREI] for the identifier; otherwise, the [normalized value]
is the identifier.

The translation of a QualifyingInformation instance of the "ALL" form
is an element item with the [local name] "allIdentifiers".

Examples

    Assume this type assignment:

        MyType ::= SEQUENCE {
            a   SEQUENCE OF CHOICE {
                b   ENUMERATED { red, green, blue },
                c   INTEGER
            }
        }

    MyType.a.*.ALL

    <target type="MyType">
     <component> a / item </component>
     <allTextuallyPresent/>
    </target>

    MyType.a.*.b:ALL

    <target type="MyType">
     <component> a / item / b </component>
     <allIdentifiers/>
    </target>

6.2. BuiltInTypeIdentification Translation

   The translation of a BuiltInTypeIdentification is the translation of
   its BuiltInTypeName followed by the translation of its
   BuiltInTypeQualifyingInformationPart, if present.

   If the BuiltInTypeName is "BIT STRING", "BOOLEAN",
   "CHARACTER STRING", "EMBEDDED PDV", "EXTERNAL", "GeneralizedTime",
   "INTEGER", "NULL", "ObjectDescriptor", "OBJECT IDENTIFIER",
   "OCTET STRING", "REAL", "RELATIVE-OID", or "UTCTime", then the
   translation of the BuiltInTypeName is an attribute item with the
   [local name] "type". The [normalized value] of this attribute item
   is a qualified name with the namespace name being
   "urn:ietf:params:xml:ns:asnx" and the local part being the
   BuiltInTypeName with any white space separators replaced by a single
   hyphen character ('-', U+002D), e.g., BIT STRING becomes BIT-STRING.

   If the BuiltInTypeName is a RestrictedCharacterStringType, then the
   translation of the BuiltInTypeName is an attribute item with the
   [local name] "type". The [normalized value] of this attribute item
   is a qualified name with the namespace name being
   "urn:ietf:params:xml:ns:asnx" and the local part being the
   RestrictedCharacterStringType, i.e., BMPString, GeneralString,

GraphicString, IA5String, ISO646String, NumericString,
PrintableString, TeletexString, T61String, UniversalString,
UTF8String, VideotexString, or VisibleString.

If the BuiltInTypeName is "CHOICE", then the translation of the
BuiltInTypeName is an element item with the [local name] "choice".

If the BuiltInTypeName is "ENUMERATED", then the translation of the
BuiltInTypeName is an element item with the [local name]
"enumerated".

If the BuiltInTypeName is "INSTANCE OF", then the translation of the
BuiltInTypeName is an element item with the [local name]
"instanceOf".

If the BuiltInTypeName is "SEQUENCE", then the translation of the
BuiltInTypeName is an element item with the [local name] "sequence".

If the BuiltInTypeName is "SEQUENCE OF", then the translation of the
BuiltInTypeName is an element item with the [local name]
"sequenceOf".

If the BuiltInTypeName is "SET", then the translation of the
BuiltInTypeName is an element item with the [local name] "set".

If the BuiltInTypeName is "SET OF", then the translation of the
BuiltInTypeName is an element item with the [local name] "setOf".

The translation of a BuiltInTypeQualifyingInformationPart is the
translation of the BuiltInTypeQualifyingInformation instance in the
BuiltInTypeQualifyingInformationPart.

The translation of a BuiltInTypeQualifyingInformation instance of the
"identifier" form is an element item with the [local name]
"identifier". An attribute item with the [local name] "name" SHALL
be added to the [attributes] of the <identifier> element item. The
[normalized value] of this attribute item is the identifier in the
BuiltInTypeQualifyingInformation instance.

The translation of a BuiltInTypeQualifyingInformation instance of the
"ALL" form is an element item with the [local name] "allIdentifiers".

    Examples

       BOOLEAN:true

       <target xmlns:asnx="urn:ietf:params:xml:ns:asnx"
               type="asnx:BOOLEAN">
        <identifier name="true"/>
       </target>

       ENUMERATED:ALL

       <target>
        <enumerated/>
        <allIdentifiers/>
       </target>

6.3. IdentifiersInContext Translation

   The translation of an IdentifiersInContext instance is an element
   item with the [local name] "components".

   If the IdentifierList in the IdentifiersInContext instance is not of
   the "ALL" or "COMPONENTS" form, then for each identifier in the
   IdentifierList, an element item with the same [local name] (i.e.,
   "attribute", "element", "component", "group", "item", "member", or
   "simpleContent") as the translation of the NamedType corresponding to
   the identifier SHALL be appended to the [children] of the
   <components> element item. An attribute item with the [local name]
   "name" SHALL be added to the [attributes] of each <attribute>,
   <element>, <component>, <group>, <item>, <member>, or <simpleContent>
   element item. The [normalized value] of each of these attribute
   items is a qualified name for the expanded name [XMLNS10] of the
   NamedType [RXEREI] corresponding to the identifier.

   If the IdentifierList in the IdentifiersInContext instance is of the
   "ALL" form, then an element item with the [local name]
   "allTextuallyPresent" SHALL be appended to the [children] of the
   <components> element item.

   If the IdentifierList in the IdentifiersInContext instance is of the
   "COMPONENTS" form, then an element item with the [local name]
   "allFirstLevel" SHALL be appended to the [children] of the
   <components> element item.

   An element item with the [local name] "in" SHALL be appended to the
   [children] of the <components> element item. The translation of the
   TypeIdentification in the IdentifiersInContext instance SHALL be
   added to the [children] and/or [attributes] of the <in> element item.

Example

   Assume this type assignment:

      MyType ::= SEQUENCE {
           field   INTEGER,
           fieldAtt [RXER:NAME AS "field"][RXER:ATTRIBUTE] BOOLEAN
      }

   field, fieldAtt IN MyType

   <target>
    <components>
     <element name="field"/>
     <attribute name="field"/>
     <in type="MyType"/>
    </components>
   </target>

7. Security Considerations

   The ASN.X translation of an XER encoding instruction is semantically
   equivalent to the original XER encoding instruction. The security
   considerations that apply to an application built from an original
   ASN.1 specification with XER encoding instructions apply equally to
   an application built from the ASN.X translation of the ASN.1
   specification.

   See the main specification for ASN.X [ASN.X] for security
   considerations related to ASN.X modules.

8. References

8.1. Normative References

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

   [URI]   Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform
              Resource Identifiers (URI): Generic Syntax", STD 66, RFC
              3986, January 2005.

   [RXER]   Legg, S. and D. Prager, "Robust XML Encoding Rules (RXER)
              for Abstract Syntax Notation One (ASN.1)", RFC 4910, July
              2007.

   [RXEREI]   Legg, S., "Encoding Instructions for the Robust XML
              Encoding Rules (RXER)", RFC 4911, July 2007.

   [ASN.X]   Legg, S., "Abstract Syntax Notation X (ASN.X)", RFC 4912,
                July 2007.

   [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.

   [X.693-1]   Amendment 1: (to ITU-T Rec. X.693 | ISO/IEC 8825-4) XER
                encoding instructions and EXTENDED-XER.

   [XML10]   Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E. and
                F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fourth
                Edition)", W3C Recommendation,
                http://www.w3.org/TR/2006/REC-xml-20060816, August 2006.

   [XML11]   Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E.,
                Yergeau, F., and J. Cowan, "Extensible Markup Language
                (XML) 1.1 (Second Edition)", W3C Recommendation,
                http://www.w3.org/TR/2006/REC-xml11-20060816, August 2006.

   [XMLNS10]   Bray, T., Hollander, D., Layman, A., and R. Tobin,
                "Namespaces in XML 1.0 (Second Edition)", W3C
                Recommendation,
                http://www.w3.org/TR/2006/REC-xml-names-20060816, August
                2006.

   [INFOSET]   Cowan, J. and R. Tobin, "XML Information Set (Second
                Edition)", W3C Recommendation,
                http://www.w3.org/TR/2004/REC-xml-infoset-20040204,
                February 2004.

   [UNICODE]   The Unicode Consortium, "The Unicode Standard, Version
                4.0", Boston, MA, Addison-Wesley Developers Press, 2003.
                ISBN 0-321-18578-1.

8.2. Informative References

   [X.693]   ITU-T Recommendation X.693 (12/01) | ISO/IEC 8825-4:2002,
                Information technology - ASN.1 encoding rules: XML
                encoding rules (XER).

Appendix A. ASN.1 for XER Encoding Instruction Notation

   This appendix is normative.

   XER-EncodingInstructionNotation
       { iso(1) identified-organization(3) dod(6)
         internet(1) private(4) enterprise(1)
         xmled(21472) asnx(1) module(0) xer-ei-notation(3) }

   -- Copyright (C) The IETF Trust (2007). This version of
   -- this ASN.1 module is part of RFC 4914; see the RFC itself
   -- for full legal notices.
   --
   -- Regarding this ASN.1 module or any portion of it, the author
   -- makes no guarantees and is not responsible for any damage
   -- resulting from its use. The author grants irrevocable permission
   -- to anyone to use, modify, and distribute it in any way that does
   -- not diminish the rights of anyone else to use, modify, and
   -- distribute it, provided that redistributed derivative works do
   -- not contain misleading author or version information.
   -- Derivative works need not be licensed under similar terms.

   DEFINITIONS
   RXER INSTRUCTIONS
   AUTOMATIC TAGS
   EXTENSIBILITY IMPLIED ::= BEGIN

   IMPORTS
       AnyURI,
       NCName
           FROM AdditionalBasicDefinitions
               { iso(1) identified-organization(3) dod(6)
                 internet(1) private(4) enterprise(1)
                 xmled(21472) asnx(1) module(0) basic(0) }
       Annotation,
       Value
           FROM AbstractSyntaxNotation-X
               { iso(1) identified-organization(3) dod(6)
                 internet(1) private(4) enterprise(1)
                 xmled(21472) asnx(1) module(0) notation(1) }
       TargetList
           FROM TargetListNotation
               { iso(1) identified-organization(3) dod(6)
                 internet(1) private(4) enterprise(1)
                 xmled(21472) asnx(1) module(0)
                 target-list-notation(4) }
       ;

```
XER-EncodingInstructionAssignmentList ::= SEQUENCE {
    annotation    Annotation OPTIONAL,
    instructions  [GROUP] XER-TargettedEncodingInstructions
}

XER-TargettedEncodingInstructions ::= SEQUENCE SIZE (1..MAX) OF
    targettedInstruction XER-TargettedEncodingInstruction

XER-TargettedEncodingInstruction ::= SEQUENCE {
    instruction  [GROUP] XER-GeneralEncodingInstruction,
    targetList   [GROUP] TargetList OPTIONAL
}

XER-EncodingInstruction ::= XER-GeneralEncodingInstruction
    (WITH COMPONENTS { ..., globalDefaults ABSENT })

XER-GeneralEncodingInstruction ::= [SINGULAR-INSERTIONS] CHOICE {
    anyAttributes        XER-WildcardInstruction,
    not-anyAttributes    XER-NegatingInstruction,
    anyElement           XER-WildcardInstruction,
    not-anyElement       XER-NegatingInstruction,
    attribute            XER-SimpleInstruction,
    not-attribute        XER-NegatingInstruction,
    base64               XER-SimpleInstruction,
    not-base64           XER-NegatingInstruction,
    decimal              XER-SimpleInstruction,
    not-decimal          XER-NegatingInstruction,
    defaultForEmpty      XER-DefaultForEmptyInstruction,
    not-defaultForEmpty  XER-NegatingInstruction,
    embedValues          XER-SimpleInstruction,
    not-embedValues      XER-NegatingInstruction,
    globalDefaults       XER-GlobalDefaultsInstruction,
    list                 XER-SimpleInstruction,
    not-list             XER-NegatingInstruction,
    name                 XER-NameInstruction,
    not-name             XER-NegatingInstruction,
    namespace            XER-NamespaceInstruction,
    not-namespace        XER-NegatingInstruction,
    piOrComment          XER-PIOrCommentInstruction,
    not-piOrComment      XER-NegatingInstruction,
    text                 XER-TextInstruction,
    not-text             XER-NegatingInstruction,
    untagged             XER-SimpleInstruction,
    not-untagged         XER-NegatingInstruction,
    element              XER-NegatingInstruction,
    useNil               XER-SimpleInstruction,
    not-useNil           XER-NegatingInstruction,
    useNumber            XER-SimpleInstruction,
```

```
      not-useNumber          XER-NegatingInstruction,
      useOrder               XER-SimpleInstruction,
      not-useOrder           XER-NegatingInstruction,
      useQName               XER-SimpleInstruction,
      not-useQName           XER-NegatingInstruction,
      useType                XER-SimpleInstruction,
      not-useType            XER-NegatingInstruction,
      useUnion               XER-SimpleInstruction,
      not-useUnion           XER-NegatingInstruction,
      whiteSpace             XER-WhiteSpaceInstruction,
      not-whiteSpace         XER-NegatingInstruction
   }

   XER-SimpleInstruction ::= SEQUENCE { }

   XER-NegatingInstruction ::= XER-SimpleInstruction

   XER-WildcardInstruction ::= SEQUENCE {
      namespaceRestriction  [GROUP] XER-NamespaceRestriction OPTIONAL
   }

   XER-NamespaceRestriction ::= [NO-INSERTIONS] CHOICE {
      from    XER-URIList,
      except  XER-URIList
   }

   XER-URIList ::= SEQUENCE SIZE(1..MAX) OF
      uriOrAbsent [GROUP] XER-QuotedURIorAbsent

   XER-QuotedURIorAbsent ::= [SINGULAR-INSERTIONS] CHOICE {
      namespace  AnyURI,
      local      NULL
   }

   XER-DefaultForEmptyInstruction ::= SEQUENCE {
      value  [GROUP] Value
   }

   XER-GlobalDefaultsInstruction ::= SEQUENCE {
      defaultSetting  [GROUP] [SINGULAR-INSERTIONS] CHOICE {
         modifiedEncodings  NULL,
         controlNamespace   XER-ControlNamespace
      }
   }

   XER-ControlNamespace ::= SEQUENCE {
      name    [ATTRIBUTE] AnyURI,
      prefix  [ATTRIBUTE] NCName OPTIONAL
```

```
   }

   XER-NameInstruction ::= SEQUENCE {
       newNameOrKeyword  [GROUP] XER-NewNameOrKeyword
   }

   XER-NewNameOrKeyword ::= [NO-INSERTIONS] CHOICE {
       newName     [ATTRIBUTE] UTF8String,
       conversion  [ATTRIBUTE] XER-Conversion
   }

   XER-Conversion ::= ENUMERATED {
       capitalized   (0),
       uncapitalized (1),
       uppercased    (2),
       lowercased    (3)
   }

   XER-NamespaceInstruction ::= SEQUENCE {
       namespace  [GROUP] XER-NamespaceSpecification OPTIONAL
   }

   XER-NamespaceSpecification ::= [HOLLOW-INSERTIONS] SEQUENCE {
       name    [ATTRIBUTE] AnyURI,
       prefix  [ATTRIBUTE] NCName OPTIONAL
   }

   XER-PIOrCommentInstruction ::= SEQUENCE {
       text      [ATTRIBUTE] UTF8String,
       position  [ATTRIBUTE] XER-Position
   }

   XER-Position ::= ENUMERATED {
       beforeTag   (0),
       beforeValue (1),
       afterValue  (2),
       afterTag    (3)
   }

   XER-TextInstruction ::= SEQUENCE {
       newNameOrKeyword  [GROUP] XER-NewNameOrKeyword OPTIONAL
   }

   XER-WhiteSpaceInstruction ::= SEQUENCE {
       action  [ATTRIBUTE] XER-WhiteSpaceAction
   }

   XER-WhiteSpaceAction ::= ENUMERATED {
```

```
        replace  (0),
        collapse (1)
    }

    ENCODING-CONTROL RXER

        SCHEMA-IDENTITY   "urn:oid:1.3.6.1.4.1.21472.1.0.3"
        TARGET-NAMESPACE "urn:ietf:params:xml:ns:asnx" PREFIX "asnx"

    END
```

Appendix B. ASN.1 for Target List Notation

   This appendix is normative.

```
   TargetListNotation
       { iso(1) identified-organization(3) dod(6)
         internet(1) private(4) enterprise(1)
         xmled(21472) asnx(1) module(0) target-list-notation(4) }

   -- Copyright (C) The IETF Trust (2007).  This version of
   -- this ASN.1 module is part of RFC 4914; see the RFC itself
   -- for full legal notices.
   --
   -- Regarding this ASN.1 module or any portion of it, the author
   -- makes no guarantees and is not responsible for any damage
   -- resulting from its use.  The author grants irrevocable permission
   -- to anyone to use, modify, and distribute it in any way that does
   -- not diminish the rights of anyone else to use, modify, and
   -- distribute it, provided that redistributed derivative works do
   -- not contain misleading author or version information.
   -- Derivative works need not be licensed under similar terms.

   DEFINITIONS
   RXER INSTRUCTIONS
   AUTOMATIC TAGS
   EXTENSIBILITY IMPLIED ::= BEGIN

   IMPORTS
       Markup,
       NCName,
       QName
           FROM AdditionalBasicDefinitions
               { iso(1) identified-organization(3) dod(6)
                 internet(1) private(4) enterprise(1)
                 xmled(21472) asnx(1) module(0) basic(0) }
       ;
```

```
    TargetList ::= SEQUENCE SIZE (1..MAX) OF target Targets

    Targets ::= [NO-INSERTIONS] CHOICE {
        allTypes         NULL,
        identifiedTypes  [GROUP] QualifiedTypeIdentification,
        components        IdentifiersInContext,
        allImportsFrom   SEQUENCE { }
            -- allImportsFrom is not used in this version
    }

    -- TypeIdentification and BuiltInTypeIdentification
    QualifiedTypeIdentification ::= [HOLLOW-INSERTIONS] SEQUENCE {
        types          [GROUP] [SINGULAR-INSERTIONS] CHOICE {
            specificType  [GROUP] SpecificTypeIdentification,
            choice        NULL,
            enumerated    NULL,
            instanceOf    NULL,
            sequence      NULL,
            sequenceOf    NULL,
            set           NULL,
            setOf         NULL
        },
        qualification  [GROUP] QualifyingInformationPart OPTIONAL
    }

    SpecificTypeIdentification ::= [HOLLOW-INSERTIONS] SEQUENCE {
        type               [ATTRIBUTE] QName,
        component          Markup OPTIONAL,
        allTextuallyPresent  NULL OPTIONAL
    }

    QualifyingInformationPart ::= [NO-INSERTIONS] CHOICE {
        allIdentifiers  NULL,
        identifier      SEQUENCE {
            name            [ATTRIBUTE] NCName
        }
    }

    IdentifiersInContext ::= SEQUENCE {
        identifiers  [GROUP] IdentifierList,
        in           CHOICE {
            allTypes      NULL,
            specificType  [GROUP] SpecificTypeIdentification
        }
    }

    IdentifierList ::= [NO-INSERTIONS] CHOICE {
        list               [GROUP] IdentifiedComponents,
```

```
       allTextuallyPresent  NULL,
       allFirstLevel        NULL
   }

   IdentifiedComponents ::= SEQUENCE SIZE (1..MAX) OF
       identifiedComponent [GROUP] IdentifiedComponent

   IdentifiedComponent ::= [SINGULAR-INSERTIONS] CHOICE {
       component       GenericIdentifiedComponent,
       element         GenericIdentifiedComponent,
       attribute       GenericIdentifiedComponent,
       group           GenericIdentifiedComponent,
       member          GenericIdentifiedComponent,
       item            GenericIdentifiedComponent,
       simpleContent   GenericIdentifiedComponent
   }

   GenericIdentifiedComponent ::= SEQUENCE {
       name  [ATTRIBUTE] QName
   }

   ENCODING-CONTROL RXER

       SCHEMA-IDENTITY   "urn:oid:1.3.6.1.4.1.21472.1.0.4"
       TARGET-NAMESPACE "urn:ietf:params:xml:ns:asnx" PREFIX "tln"

   END
```

Appendix C. ASN.X for XER Encoding Instruction Notation

   This appendix is non-normative.

```
   <?xml version="1.0"?>
   <asnx:module xmlns:asnx="urn:ietf:params:xml:ns:asnx"
                xmlns:tln="urn:ietf:params:xml:ns:asnx"
                name="XER-EncodingInstructionNotation"
                identifier="1.3.6.1.4.1.21472.1.0.3"
                schemaIdentity="urn:oid:1.3.6.1.4.1.21472.1.0.3"
                targetNamespace="urn:ietf:params:xml:ns:asnx"
                targetPrefix="asnx"
                extensibilityImplied="true">

    <annotation>
     Copyright (C) The IETF Trust (2007).  This version of
     this ASN.X module is part of RFC 4914; see the RFC itself
     for full legal notices.

     Regarding this ASN.X module or any portion of it, the author
```

```
 makes no guarantees and is not responsible for any damage
 resulting from its use.  The author grants irrevocable permission
 to anyone to use, modify, and distribute it in any way that does
 not diminish the rights of anyone else to use, modify, and
 distribute it, provided that redistributed derivative works do
 not contain misleading author or version information.
 Derivative works need not be licensed under similar terms.
</annotation>

<import name="AbstractSyntaxNotation-X"
        identifier="1.3.6.1.4.1.21472.1.0.1"
        schemaIdentity="urn:oid:1.3.6.1.4.1.21472.1.0.1"
        namespace="urn:ietf:params:xml:ns:asnx"/>

<import name="TargetListNotation"
        identifier="1.3.6.1.4.1.21472.1.0.4"
        schemaIdentity="urn:oid:1.3.6.1.4.1.21472.1.0.4"
        namespace="urn:ietf:params:xml:ns:asnx"/>

<namedType name="XER-EncodingInstructionAssignmentList">
 <type>
  <sequence>
   <optional>
    <element name="annotation" type="asnx:Annotation"/>
   </optional>
   <group name="instructions"
          type="asnx:XER-TargettedEncodingInstructions"/>
  </sequence>
 </type>
</namedType>

<namedType name="XER-TargettedEncodingInstructions">
 <type>
  <sequenceOf minSize="1">
   <element name="targettedInstruction"
            type="asnx:XER-TargettedEncodingInstruction"/>
  </sequenceOf>
 </type>
</namedType>

<namedType name="XER-TargettedEncodingInstruction">
 <type>
  <sequence>
   <group name="instruction"
          type="asnx:XER-GeneralEncodingInstruction"/>
   <optional>
    <group name="targetList" type="tln:TargetList"/>
   </optional>
```

```
     </sequence>
    </type>
   </namedType>

   <namedType name="XER-EncodingInstruction">
    <type>
     <constrained type="asnx:XER-GeneralEncodingInstruction">
      <withComponents partial="true">
       <element name="globalDefaults" use="absent"/>
      </withComponents>
     </constrained>
    </type>
   </namedType>

   <namedType name="XER-GeneralEncodingInstruction">
    <type>
     <choice insertions="singular">
      <element name="anyAttributes"
               type="asnx:XER-WildcardInstruction"/>
      <element name="not-anyAttributes"
               type="asnx:XER-NegatingInstruction"/>
      <element name="anyElement" type="asnx:XER-WildcardInstruction"/>
      <element name="not-anyElement"
               type="asnx:XER-NegatingInstruction"/>
      <element name="attribute" type="asnx:XER-SimpleInstruction"/>
      <element name="not-attribute"
               type="asnx:XER-NegatingInstruction"/>
      <element name="base64" type="asnx:XER-SimpleInstruction"/>
      <element name="not-base64" type="asnx:XER-NegatingInstruction"/>
      <element name="decimal" type="asnx:XER-SimpleInstruction"/>
      <element name="not-decimal" type="asnx:XER-NegatingInstruction"/>
      <element name="defaultForEmpty"
               type="asnx:XER-DefaultForEmptyInstruction"/>
      <element name="not-defaultForEmpty"
               type="asnx:XER-NegatingInstruction"/>
      <element name="embedValues" type="asnx:XER-SimpleInstruction"/>
      <element name="not-embedValues"
               type="asnx:XER-NegatingInstruction"/>
      <element name="globalDefaults"
               type="asnx:XER-GlobalDefaultsInstruction"/>
      <element name="list" type="asnx:XER-SimpleInstruction"/>
      <element name="not-list" type="asnx:XER-NegatingInstruction"/>
      <element name="name" type="asnx:XER-NameInstruction"/>
      <element name="not-name" type="asnx:XER-NegatingInstruction"/>
      <element name="namespace" type="asnx:XER-NamespaceInstruction"/>
      <element name="not-namespace"
               type="asnx:XER-NegatingInstruction"/>
      <element name="piOrComment"
```

```
                    type="asnx:XER-PIOrCommentInstruction"/>
          <element name="not-piOrComment"
                    type="asnx:XER-NegatingInstruction"/>
          <element name="text" type="asnx:XER-TextInstruction"/>
          <element name="not-text" type="asnx:XER-NegatingInstruction"/>
          <element name="untagged" type="asnx:XER-SimpleInstruction"/>
          <element name="not-untagged"
                    type="asnx:XER-NegatingInstruction"/>
          <element name="element" type="asnx:XER-NegatingInstruction"/>
          <element name="useNil" type="asnx:XER-SimpleInstruction"/>
          <element name="not-useNil" type="asnx:XER-NegatingInstruction"/>
          <element name="useNumber" type="asnx:XER-SimpleInstruction"/>
          <element name="not-useNumber"
                    type="asnx:XER-NegatingInstruction"/>
          <element name="useOrder" type="asnx:XER-SimpleInstruction"/>
          <element name="not-useOrder"
                    type="asnx:XER-NegatingInstruction"/>
          <element name="useQName" type="asnx:XER-SimpleInstruction"/>
          <element name="not-useQName"
                    type="asnx:XER-NegatingInstruction"/>
          <element name="useType" type="asnx:XER-SimpleInstruction"/>
          <element name="not-useType" type="asnx:XER-NegatingInstruction"/>
          <element name="useUnion" type="asnx:XER-SimpleInstruction"/>
          <element name="not-useUnion"
                    type="asnx:XER-NegatingInstruction"/>
          <element name="whiteSpace"
                    type="asnx:XER-WhiteSpaceInstruction"/>
          <element name="not-whiteSpace"
                    type="asnx:XER-NegatingInstruction"/>
        </choice>
       </type>
      </namedType>

      <namedType name="XER-SimpleInstruction">
       <type>
        <sequence/>
       </type>
      </namedType>

      <namedType name="XER-NegatingInstruction"
                  type="asnx:XER-SimpleInstruction"/>

      <namedType name="XER-WildcardInstruction">
       <type>
        <sequence>
         <optional>
          <group name="namespaceRestriction"
                  type="asnx:XER-NamespaceRestriction"/>
```

```
      </optional>
     </sequence>
    </type>
   </namedType>

   <namedType name="XER-NamespaceRestriction">
    <type>
     <choice insertions="none">
      <element name="from" type="asnx:XER-URIList"/>
      <element name="except" type="asnx:XER-URIList"/>
     </choice>
    </type>
   </namedType>

   <namedType name="XER-URIList">
    <type>
     <sequenceOf minSize="1">
      <group name="uriOrAbsent" type="asnx:XER-QuotedURIorAbsent"/>
     </sequenceOf>
    </type>
   </namedType>

   <namedType name="XER-QuotedURIorAbsent">
    <type>
     <choice insertions="singular">
      <element name="namespace" type="asnx:AnyURI"/>
      <element name="local" type="asnx:NULL"/>
     </choice>
    </type>
   </namedType>

   <namedType name="XER-DefaultForEmptyInstruction">
    <type>
     <sequence>
      <group name="value" type="asnx:Value"/>
     </sequence>
    </type>
   </namedType>

   <namedType name="XER-GlobalDefaultsInstruction">
    <type>
     <sequence>
      <group name="defaultSetting">
       <type>
        <choice insertions="singular">
         <element name="modifiedEncodings" type="asnx:NULL"/>
         <element name="controlNamespace"
                  type="asnx:XER-ControlNamespace"/>
```

```
      </choice>
     </type>
    </group>
   </sequence>
  </type>
 </namedType>

 <namedType name="XER-ControlNamespace">
  <type>
   <sequence>
    <attribute name="name" type="asnx:AnyURI"/>
    <optional>
     <attribute name="prefix" type="asnx:NCName"/>
    </optional>
   </sequence>
  </type>
 </namedType>

 <namedType name="XER-NameInstruction">
  <type>
   <sequence>
    <group name="newNameOrKeyword" type="asnx:XER-NewNameOrKeyword"/>
   </sequence>
  </type>
 </namedType>

 <namedType name="XER-NewNameOrKeyword">
  <type>
   <choice insertions="none">
    <attribute name="newName" type="asnx:UTF8String"/>
    <attribute name="conversion" type="asnx:XER-Conversion"/>
   </choice>
  </type>
 </namedType>

 <namedType name="XER-Conversion">
  <type>
   <enumerated>
    <enumeration name="capitalized" number="0"/>
    <enumeration name="uncapitalized" number="1"/>
    <enumeration name="uppercased" number="2"/>
    <enumeration name="lowercased" number="3"/>
   </enumerated>
  </type>
 </namedType>

 <namedType name="XER-NamespaceInstruction">
  <type>
```

```
    <sequence>
     <optional>
      <group name="namespace" type="asnx:XER-NamespaceSpecification"/>
     </optional>
    </sequence>
   </type>
  </namedType>

  <namedType name="XER-NamespaceSpecification">
   <type>
    <sequence insertions="hollow">
     <attribute name="name" type="asnx:AnyURI"/>
     <optional>
      <attribute name="prefix" type="asnx:NCName"/>
     </optional>
    </sequence>
   </type>
  </namedType>

  <namedType name="XER-PIOrCommentInstruction">
   <type>
    <sequence>
     <attribute name="text" type="asnx:UTF8String"/>
     <attribute name="position" type="asnx:XER-Position"/>
    </sequence>
   </type>
  </namedType>

  <namedType name="XER-Position">
   <type>
    <enumerated>
     <enumeration name="beforeTag" number="0"/>
     <enumeration name="beforeValue" number="1"/>
     <enumeration name="afterValue" number="2"/>
     <enumeration name="afterTag" number="3"/>
    </enumerated>
   </type>
  </namedType>

  <namedType name="XER-TextInstruction">
   <type>
    <sequence>
     <optional>
      <group name="newNameOrKeyword"
             type="asnx:XER-NewNameOrKeyword"/>
     </optional>
    </sequence>
   </type>
```

```
     </namedType>

     <namedType name="XER-WhiteSpaceInstruction">
      <type>
       <sequence>
        <attribute name="action" type="asnx:XER-WhiteSpaceAction"/>
       </sequence>
      </type>
     </namedType>

     <namedType name="XER-WhiteSpaceAction">
      <type>
       <enumerated>
        <enumeration name="replace" number="0"/>
        <enumeration name="collapse" number="1"/>
       </enumerated>
      </type>
     </namedType>

    </asnx:module>
```

Appendix D. ASN.X for Target List Notation

   This appendix is non-normative.

```
   <?xml version="1.0"?>
   <asnx:module xmlns:asnx="urn:ietf:params:xml:ns:asnx"
                xmlns:tln="urn:ietf:params:xml:ns:asnx"
                name="TargetListNotation"
                identifier="1.3.6.1.4.1.21472.1.0.4"
                schemaIdentity="urn:oid:1.3.6.1.4.1.21472.1.0.4"
                targetNamespace="urn:ietf:params:xml:ns:asnx"
                targetPrefix="tln"
                extensibilityImplied="true">

    <annotation>
     Copyright (C) The IETF Trust (2007).  This version of
     this ASN.X module is part of RFC 4914; see the RFC itself
     for full legal notices.

     Regarding this ASN.X module or any portion of it, the author
     makes no guarantees and is not responsible for any damage
     resulting from its use.  The author grants irrevocable permission
     to anyone to use, modify, and distribute it in any way that does
     not diminish the rights of anyone else to use, modify, and
     distribute it, provided that redistributed derivative works do
     not contain misleading author or version information.
     Derivative works need not be licensed under similar terms.
```

```
     </annotation>

     <namedType name="TargetList">
      <type>
       <sequenceOf minSize="1">
        <element name="target" type="tln:Targets"/>
       </sequenceOf>
      </type>
     </namedType>

     <namedType name="Targets">
      <type>
       <choice insertions="none">
        <element name="allTypes" type="asnx:NULL"/>
        <group name="identifiedTypes"
               type="tln:QualifiedTypeIdentification"/>
        <element name="components" type="tln:IdentifiersInContext"/>
        <element name="allImportsFrom">
         <annotation>
          allImportsFrom is not used in this version
         </annotation>
         <type>
          <sequence/>
         </type>
        </element>
       </choice>
      </type>
     </namedType>

     <namedType name="QualifiedTypeIdentification">
      <annotation>
       TypeIdentification and BuiltInTypeIdentification
      </annotation>
      <type>
       <sequence insertions="hollow">
        <group name="types">
         <type>
          <choice insertions="singular">
           <group name="specificType"
                  type="tln:SpecificTypeIdentification"/>
           <element name="choice" type="asnx:NULL"/>
           <element name="enumerated" type="asnx:NULL"/>
           <element name="instanceOf" type="asnx:NULL"/>
           <element name="sequence" type="asnx:NULL"/>
           <element name="sequenceOf" type="asnx:NULL"/>
           <element name="set" type="asnx:NULL"/>
           <element name="setOf" type="asnx:NULL"/>
          </choice>
```

```
        </type>
      </group>
      <optional>
       <group name="qualification"
              type="tln:QualifyingInformationPart"/>
      </optional>
     </sequence>
    </type>
   </namedType>

   <namedType name="SpecificTypeIdentification">
    <type>
     <sequence insertions="hollow">
      <attribute name="type" type="asnx:QName"/>
      <optional>
       <element name="component" type="asnx:Markup"/>
      </optional>
      <optional>
       <element name="allTextuallyPresent" type="asnx:NULL"/>
      </optional>
     </sequence>
    </type>
   </namedType>

   <namedType name="QualifyingInformationPart">
    <type>
     <choice insertions="none">
      <element name="allIdentifiers" type="asnx:NULL"/>
      <element name="identifier">
       <type>
        <sequence>
         <attribute name="name" type="asnx:NCName"/>
        </sequence>
       </type>
      </element>
     </choice>
    </type>
   </namedType>

   <namedType name="IdentifiersInContext">
    <type>
     <sequence>
      <group name="identifiers" type="tln:IdentifierList"/>
      <element name="in">
       <type>
        <choice>
         <element name="allTypes" type="asnx:NULL"/>
         <group name="specificType"
```

```
                    type="tln:SpecificTypeIdentification"/>
         </choice>
        </type>
       </element>
      </sequence>
     </type>
    </namedType>


    <namedType name="IdentifierList">
     <type>
      <choice insertions="none">
       <group name="list" type="tln:IdentifiedComponents"/>
       <element name="allTextuallyPresent" type="asnx:NULL"/>
       <element name="allFirstLevel" type="asnx:NULL"/>
      </choice>
     </type>
    </namedType>


    <namedType name="IdentifiedComponents">
     <type>
      <sequenceOf minSize="1">
       <group name="identifiedComponent"
              type="tln:IdentifiedComponent"/>
      </sequenceOf>
     </type>
    </namedType>


    <namedType name="IdentifiedComponent">
     <type>
      <choice insertions="singular">
       <element name="component" type="tln:GenericIdentifiedComponent"/>
       <element name="element" type="tln:GenericIdentifiedComponent"/>
       <element name="attribute" type="tln:GenericIdentifiedComponent"/>
       <element name="group" type="tln:GenericIdentifiedComponent"/>
       <element name="member" type="tln:GenericIdentifiedComponent"/>
       <element name="item" type="tln:GenericIdentifiedComponent"/>
       <element name="simpleContent"
                type="tln:GenericIdentifiedComponent"/>
      </choice>
     </type>
    </namedType>

    <namedType name="GenericIdentifiedComponent">
     <type>
      <sequence>
       <attribute name="name" type="asnx:QName"/>
      </sequence>
     </type>
```

```
      </namedType>

      </asnx:module>
```

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