INTERNATIONAL STANDARD

ISO/IEC 8825-5

First edition
2004-11-15

Information technology — ASN.1 encoding rules: Mapping W3C XML schema definitions into ASN.1

Technologies de l'information — Règles de codage ASN.1: Codage de définitions de schéma en W3C XML dans ASN.1

Reference number
ISO/IEC 8825-5:2004(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

CONTENTS

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 8825-5 was prepared jointly by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems* in collaboration with ITU-T. The identical text is published as ITU-T Rec. X.694.

ISO/IEC 8825 consists of the following parts, under the general title *Information technology — ASN.1 encoding rules:*

— *Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

— *Part 2: Specification of Packed Encoding Rules (PER)*

— *Part 3: Specification of Encoding Control Notation (ECN)*

— *Part 4: XML Encoding Rules (XER)*

— *Part 5: Mapping W3C XML schema definitions into ASN.1*

Introduction

This Recommendation | International Standard specifies a mapping from a W3C XML Schema definition (an XSD Schema) into an ASN.1 schema. The mapping can be applied to any XSD Schema. It specifies the generation of one or more ASN.1 modules containing type definitions, together with ASN.1 XER encoding instructions. These are jointly described as an ASN.1 schema for XML documents.

This ASN.1 schema, when used with the ASN.1 Extended XML Encoding Rules (EXTENDED-XER), can be used to generate and to validate the same set of W3C XML 1.0 documents as the original XSD Schema. The resulting ASN.1 types and encodings support the same semantic content as the XSD Schema. Thus ASN.1 tools can be used interchangeably with XSD tools for the generation and processing of the specified XML documents.

Other standardized ASN.1 encoding rules, such as the Distinguished Encoding Rules (DER) or the Packed Encoding Rules (PER), can be used in conjunction with this standardized mapping.

The combination of this Recommendation | International Standard with ASN.1 Encoding Rules provides fully-standardized and vendor-independent compact and canonical binary encodings for data defined using an XSD Schema.

The ASN.1 schema provides a clear separation between the specification of the information content of messages (their abstract syntax) and the precise form of the XML document (for example, use of attributes instead of elements). This results in both a clearer and generally a less verbose schema than the original XSD Schema.

Annex A forms an integral part of this Recommendation | International Standard, and is an ASN.1 module containing a set of ASN.1 type assignments that correspond to each of the XSD built-in datatypes. Mappings of XSD Schemas into ASN.1 schemas either import the type reference names of those type assignments or include the type definitions in-line.

Annex B does not form an integral part of this Recommendation | International Standard, and summarizes the object identifier values assigned in this Recommendation | International Standard.

Annex C does not form an integral part of this Recommendation | International Standard, and gives examples of the mapping of XSD Schemas into ASN.1 schemas.

Annex D does not form an integral part of this Recommendation | International Standard, and describes the use of the mapping defined in this Recommendation | International Standard, in conjunction with standardized ASN.1 Encoding Rules, to provide compact and canonical encodings for data defined using an XSD Schema.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – ASN.1 encoding rules: Mapping W3C XML schema definitions into ASN.1

1 Scope

This Recommendation | International Standard specifies a mapping from any XSD Schema into an ASN.1 schema. The ASN.1 schema supports the same semantics and validates the same set of XML documents.

This Recommendation | International Standard specifies the final XER encoding instructions that are to be applied as part of the defined mapping to ASN.1 types, but does not specify which syntactic form is to be used for the specification of those final XER encoding instructions, or the order or manner of their assignment.

NOTE – Implementers of tools generating these mappings may choose any syntactic form or order of assignment that results in the specified final XER encoding instructions being applied. Examples in this Recommendation | International Standard generally use the type prefix form, but use of an XER Encoding Control Section may be preferred for the mapping of a complete XSD Schema, as a matter of style.

There are different ways (syntactically) of assigning XER encoding instructions for use in EXTENDED-XER encodings (for example, use of ASN.1 type prefix encoding instructions or use of an XER encoding control section). The choice of these syntactic forms is a matter of style and is outside the scope of this Recommendation | International Standard.

2 Normative references

The following Recommendations | International Standards and W3C specifications contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations, International Standards and W3C specifications are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations, International Standards and W3C specifications listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations. The W3C maintains a list of currently valid W3C specifications. The reference to a document within this Recommendation | International Standard does not give it, as a stand-alone document, the status of a Recommendation or International Standard.

2.1 Identical Recommendations | International Standards

NOTE – The complete set of ASN.1 Recommendations | International Standards are listed below, as they can all be applicable in particular uses of this Recommendation | International Standard. Where these are not directly referenced in the body of this Recommendation | International Standard, a † symbol is added to the reference.

- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.681 (2002) | ISO/IEC 8824-2:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.
- ITU-T Recommendation X.682 (2002) | ISO/IEC 8824-3:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.
- ITU-T Recommendation X.683 (2002) | ISO/IEC 8824-4:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*.
- ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER)*.
- ITU-T Recommendation X.691 (2002) | ISO/IEC 8825-2:2002, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

– ITU-T Recommendation X.692 (2002) | ISO/IEC 8825-3:2002, *Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN)*.

– ITU-T Recommendation X.693 (2001) | ISO/IEC 8825-4:2002, *Information technology – ASN.1 encoding rules: XML Encoding Rules (XER)*.

– ITU-T Recommendation X.693 (2001)/Amd.1 (2003) | ISO/IEC 8825-4:2002/Amd.1:2004, *Information technology – ASN.1 encoding rules: XML Encoding Rules (XER) – Amendment 1: XER Encoding Instructions and EXTENDED-XER*.

2.2 Additional references

– ISO 8601:2000, *Data elements and interchange formats – Information interchange – Representation of dates and times.*

– ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.*

– W3C XML 1.0:2000, *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, Copyright © [6 October 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2000/REC-xml-20001006.*

– W3C XML Namespaces:1999, *Namespaces in XML, W3C Recommendation, Copyright © [14 January 1999] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/1999/REC-xml-names-19990114.*

– W3C XML Information Set:2001, XML *Information Set, W3C Recommendation, Copyright © [24 October 2001] World Wide Web Consortium (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2001/REC-xml-infoset-20011024.*

– W3C XML Schema:2001, *XML Schema Part 1: Structures, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-1-20010502.*

– W3C XML Schema:2001, *XML Schema Part 2: Datatypes, W3C Recommendation, Copyright © [2 May 2001] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2001/REC-xmlschema-2-20010502.*

NOTE – When the reference "W3C XML Schema" is used in this Recommendation | International Standard, it refers to W3C XML Schema Part 1 and W3C XML Schema Part 2.

– IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.

– IETF RFC 1766 (1995), *Tags for the Identification of Languages*.

3 Definitions

3.1 Imported definitions

3.1.1 This Recommendation | International Standard uses the terms defined in ITU-T Rec. X.680 | ISO/IEC 8824-1 and in ITU-T Rec. X.693 | ISO/IEC 8825-4.

NOTE – In particular, the terms "final XER encoding instructions", "type prefix" and "XER encoding control section" are defined in the above-mentioned Recommendations | International Standards.

3.1.2 This Recommendation | International Standard also uses the terms defined in W3C XML Schema and W3C XML Information Set.

NOTE 1 – It is believed that these terms do not conflict with the terms referenced in 3.1.1. If such a conflict occurs, the definition of the term in 3.1.1 applies.

NOTE 2 – In particular, the terms "schema component" and "property (of a schema component)" are defined in W3C XML Schema, and the term "element information item" is defined in W3C XML Information Set.

3.2 Additional definitions

For the purposes of this Recommendation | International Standard, the following additional definitions apply:

3.2.1 **XSD namespace:** A namespace with a URI of "http://www.w3.org/2001/XMLSchema".

3.2.2 **XSI namespace:** A namespace with a URI of "http://www.w3.org/2001/XMLSchema-instance".

3.2.3 **XML namespace:** A namespace with a URI of "http://www.w3.org/XML/1998/namespace".

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
BER	(ASN.1) Basic Encoding Rules
DER	(ASN.1) Distinguished Encoding Rules
PER	(ASN.1) Packed Encoding Rules
URI	(IETF) Uniform Resource Identifier
XER	(ASN.1) XML Encoding Rules
XML	(W3C) Extensible Markup Language
XSD	(W3C) XML Schema

5 Notation

5.1 This Recommendation | International Standard references the notation defined by ITU-T Rec. X.680 | ISO/IEC 8824-1, ITU-T Rec. X.682 | ISO/IEC 8824-3, W3C XML 1.0 and W3C XML Schema.

5.2 When it is necessary in the body of this Recommendation | International Standard to specify, either formally or in examples, the assignment of XER encoding instructions, the type prefix notation is generally used (but see 6.3 and 6.4). In Annex A, an XER encoding control section is used.

5.3 In this Recommendation | International Standard, **`bold Courier`** is used for ASN.1 notation and **bold Arial** is used for XSD notation and for XSD terms and concepts.

5.4 The XSD Schemas used in the examples in this Recommendation | International Standard use the prefix **xsd:** to identify the XSD namespace.

6 Purpose and extent of standardization

6.1 The mapping to ASN.1 that is specified in this Recommendation | International Standard ensures that:

a) any resulting ASN.1 modules generated by tools conforming to this Recommendation | International Standard (from the same XSD Schema) define the same (structured) abstract values;

b) all BASIC-XER, CXER, EXTENDED-XER, and binary encodings of that resulting ASN.1 specification will produce the same encodings (subject to encoder's options); and

c) all XML documents that conform to the source XSD Schema are valid EXTENDED-XER encodings of abstract values of that ASN.1 specification.

6.2 There are many aspects of an ASN.1 definition (such as the use of white-space, or of encoding control sections or type prefixes) that affect neither the abstract values being defined nor the XER or binary encodings of those values. Such aspects of the ASN.1 definition are generally not standardized in this Recommendation | International Standard.

6.3 There are many different ways in ASN.1 of assigning an XER encoding instruction to a type, including:

a) use of a type prefix for every encoding instruction to be assigned; or

b) use of an encoding control section, with a separate encoding instruction for each required assignment; or

c) use of an encoding control section, with a single encoding instruction making a global assignment, possibly supplemented by use of a negating encoding instruction for specific types.

6.4 This Recommendation | International Standard specifies when a final XER encoding instruction shall be present, and uses the syntax of 6.3 a) in most of its examples. However, the use of the different options in 6.3 is not

standardized, and conforming implementations of the mapping may choose any syntactic form (or a mixture of syntactic forms) for the assignment of final XER encoding instructions.

NOTE – The choice among these options does not affect the final binary or XML encodings.

6.5 A formal specification of the required mapping is not provided.

6.6 This Recommendation | International Standard is concerned only with the mapping of XSD Schemas that conform to W3C XML Schema.

NOTE – Such conformance can be either by the provision of one or more W3C XSD schema documents or by other means as specified in W3C XML Schema.

7 Mapping XSD Schemas

7.1 A mapping is based on a source XSD Schema, which is a set of schema components (see W3C XML Schema Part 1, 2.2). No particular representation of schema components or sets of schema components is required or assumed for the mapping, although it is expected that the source XSD Schema will usually be provided as one or more XML schema documents (see W3C XML Schema Part 1, 3.15.2).

NOTE 1 – Since the mapping is defined in terms of schema components (and not in terms of their XML representation), it is not affected by details of the XML representation, such as the use of multiple schema documents linked by **xsd:include** and **xsd:redefine** element information items, the placement of element information items in one or another schema documents, the order of **xsd:attribute** element information items within a **xsd:complexType** element information item, and so on.

NOTE 2 – Two sets of schema documents that differ in many aspects but represent the same set of schema components generate the same set of ASN.1 type assignments, with the same final encoding instructions assigned to them and to their components to any depth.

7.2 The source XSD Schema shall meet all the constraints imposed by the XSD specification. If the source XSD Schema is represented (in part or all) as a set of XML schema documents, each schema document shall be valid according to the XSD Schema for Schemas (see W3C XML Schema Part 1, Appendix A).

7.3 At least one ASN.1 module (see 7.4) shall be generated for each different **target namespace** (whether a namespace name or the keyword **absent**) that is the **target namespace** of one or more schema components in the source XSD Schema. Each ASN.1 module shall contain one or more type assignments corresponding to top-level schema components (see 7.9) that have the same **target namespace**. Each ASN.1 module can also contain one or more special ASN.1 type assignments whose associated ASN.1 type assignments are in the same ASN.1 module (see 7.6).

NOTE –The schema components represented in the multiple schema documents become part of the same XSD Schema through the use of the **xsd:include, xsd:redefine,** and **xsd:import** element information items.

7.4 The number of ASN.1 modules generated for each **target namespace** (including the keyword **absent**) may be more than one, but each ASN.1 module shall not contain type assignments corresponding to top-level schema components with different **target namespace**s (including the keyword **absent**).

7.5 When multiple ASN.1 modules are generated for a given **target namespace** (including the keyword **absent**), all the type assignments present in them shall be generated as if they were being added to a single ASN.1 module for the purpose of generating distinct type reference names (see 10.3). The type reference names generated from the **name**s of the top-level schema components with a given **target namespace** shall be the same type reference names regardless of the number of ASN.1 modules generated for that **target namespace** and regardless of the way type assignments are divided among the various ASN.1 modules.

NOTE – This is designed to provide flexibility without compromising interoperability.

7.6 Each special ASN.1 type assignment (see clauses 29, 30, and 31) shall be inserted in the same ASN.1 module as its associated ASN.1 type assignment (see 29.4, 31.4, and 30.4, respectively).

7.7 All ASN.1 modules generated by the mapping shall contain (in the XER encoding control section) a **GLOBAL-DEFAULTS MODIFIED-ENCODINGS** encoding instruction and a **GLOBAL-DEFAULTS CONTROL-NAMESPACE** encoding instruction specifying the XSI namespace.

7.8 A source XSD Schema shall be processed as follows:

a) for each top-level **element declaration**, an ASN.1 type assignment shall be generated by applying clause 14 to the **element declaration**;

b) for each top-level **attribute declaration**, an ASN.1 type assignment shall be generated by applying clause 15 to the **attribute declaration**;

c) for each user-defined top-level **simple type definition**, an ASN.1 type assignment shall be generated by applying clause 13 to the **simple type definition**;

d) for each top-level **complex type definition**, an ASN.1 type assignment shall be generated by applying clause 20 to the **complex type definition**;

e) for each **model group definition** whose **model group** has a **compositor** of **sequence** or **choice**, an ASN.1 type assignment shall be generated by applying clause 17 to the **model group definition**.

NOTE 1 – The remaining schema components of the source XSD schema will be processed as a result of mapping these schema components.

NOTE 2 – The order in which schema components are to be mapped is specified in 10.4. The order of the items of the list above has no significance for the mapping.

7.9 Column 1 of Table 1 lists schema components. Column 2 gives the reference to the clause in W3C XML Schema that defines the schema component. Column 3 lists the clause that defines the mapping of those schema components into ASN.1.

Table 1 – Mapping of XSD schema components

XSD schema component	W3C XML Schema reference	Mapping defined by
attribute declaration	Part 1, 3.2	Clause 15
element declaration	Part 1, 3.3	Clause 14
complex type definition	Part 1, 3.4	Clause 20
attribute use	Part 1, 3.5	Clause 22
attribute group definition	Part 1, 3.6	*not mapped as such*
model group definition	Part 1, 3.7	Clause 17
model group	Part 1, 3.8	Clause 18
particle	Part 1, 3.9	Clause 19
wildcard	Part 1, 3.10	Clause 21
identity-constraint definition	Part 1, 3.11	*ignored by the mapping*
notation declaration	Part 1, 3.12	*ignored by the mapping*
annotation	Part 1, 3.13	*ignored by the mapping*
simple type definition	Part 1, 3.14	Clauses 11, 13
schema	Part 1, 3.15	Clause 9
ordered	Part 2, 4.2.2.1	*ignored by the mapping*
bounded	Part 2, 4.2.3.1	*ignored by the mapping*
cardinality	Part 2, 4.2.4.1	*ignored by the mapping*
numeric	Part 2, 4.2.5.1	*ignored by the mapping*
length	Part 2, 4.3.1.1	Clause 12
minLength	Part 2, 4.3.2.1	Clause 12
maxLength	Part 2, 4.3.3.1	Clause 12
pattern	Part 2, 4.3.4.1	Clause 12
enumeration	Part 2, 4.3.5.1	Clause 12
whiteSpace	Part 2, 4.3.6.1	Clause 12
maxInclusive	Part 2, 4.3.7.1	Clause 12
maxExclusive	Part 2, 4.3.8.1	Clause 12
minExclusive	Part 2, 4.3.9.1	Clause 12
minInclusive	Part 2, 4.3.10.1	Clause 12
totalDigits	Part 2, 4.3.11.1	Clause 12
fractionDigits	Part 2, 4.3.12.1	Clause 12

8 Ignored schema components and properties

8.1 The mapping shall ignore the schema components and properties that are listed in this clause.

8.2 All **annotation**s (see W3C XML Schema Part 1, 3.13) shall be ignored.

NOTE – All attribute information items in a schema document with names qualified with namespaces other than the XSD namespace (see W3C XML Schema Part 1, 3.13.1) are a property of **annotation**s, and are ignored.

8.3 All **identity-constraint definition**s (see W3C XML Schema Part 1, 3.11) shall be ignored.

NOTE – The **identity-constraint definition** provides mechanisms for specifying referential constraints that can be required in a valid instance. ASN.1 currently has no concept of such constraints, and such constraints cannot be mapped into a formal ASN.1 specification, but they may be included as normative comments that are binding on an application implementation.

8.4 All **notation declaration**s (see W3C XML Schema Part 1, 3.12) shall be ignored.

8.5 All schema components that are the **fundamental facet**s (**ordered**, **bounded**, **cardinality**, **numeric**) of **simple type definition**s (see W3C XML Schema Part 2, 4.2) shall be ignored.

8.6 The properties **identity-constraint definition**s, **substitution group exclusion**s and **disallowed substitution**s of **element declaration**s shall be ignored.

8.7 The properties **final**, **abstract**, and **prohibited substitutions** of **complex type definition**s shall be ignored.

8.8 The property **process contents** of **wildcard**s shall be ignored.

NOTE – There is no support in ASN.1 for any action other than **skip**.

8.9 The properties **fundamental facet**s and **final** of **simple type definition**s shall be ignored.

8.10 All **value constraint**s that are present on any **element declaration**s or **attribute declaration**s whose **type definition** is either **xsd:QName** or a **simple type definition** derived from **xsd:QName** or **xsd:NOTATION** shall be ignored.

8.11 All **attribute group definition**s shall be ignored.

NOTE – The **attribute use**s in an **attribute group definition** become part of the **attribute use**s of the **complex type definition**s whose XML representation contains a reference to the **attribute group definition**.

9 The ASN.1 module and namespaces

NOTE – A full description of the relationship between the namespace concept of W3C XML Namespaces and naming in ASN.1 is provided in ITU-T Rec. X.693 | ISO/IEC 8825-4, clause 16. Type reference names and identifiers defined in an ASN.1 module are assigned a namespace by means of a **NAMESPACE** encoding instruction, and otherwise do not have a namespace. The mapping generates **NAMESPACE** encoding instructions as appropriate.

9.1 The mapping generates one or more ASN.1 modules corresponding to all schema components in the Schema that have the same **target namespace**.

9.2 The ASN.1 "ModuleIdentifier" (see ITU-T Rec. X.680 | ISO/IEC 8824-1, clause 12) to be generated by the mapping is not standardized. Where **IMPORTS** statements are used, the ASN.1 module names and module identifiers in the **IMPORTS** statements shall be those generated for the ASN.1 modules generated by the mapping.

NOTE – The choice of "ModuleIdentifier" does not affect the encodings in any of the standard encoding rules.

9.3 The ASN.1 modules shall have a "TagDefault" of **AUTOMATIC TAGS**.

9.4 In each ASN.1 module, there shall be an ASN.1 **IMPORTS** statement importing the ASN.1 type reference names in the module named **XSD** specified in Annex A that are referenced in the ASN.1 module.

9.5 The **IMPORTS** statement shall also import the ASN.1 type reference names of type assignments that have been placed (as a result of the mapping) in other ASN.1 modules but are referenced in this ASN.1 module.

9.6 There shall be no **EXPORTS** statement.

NOTE – This means that all ASN.1 type reference names in the ASN.1 module can be imported into other modules.

10 Name conversion

10.1 General

10.1.1 This Recommendation | International Standard specifies the generation of:

a) ASN.1 type reference names corresponding to the **name**s of **model group definition**s, top-level **element declaration**s, top-level **attribute declaration**s, top-level **complex type definition**s, and user-defined top-level **simple type definition**s;

b) ASN.1 identifiers corresponding to the **name**s of top-level **element declaration**s, top-level **attribute declaration**s, local **element declaration**s, and local **attribute declaration**s;

c) ASN.1 identifiers for the mapping of certain **simple type definition**s with an **enumeration** facet (see 12.4.1 and 12.4.2);

d) ASN.1 type reference names of special type assignments (see clauses 29, 30, and 31); and

e) ASN.1 identifiers of certain sequence components introduced by the mapping (see clause 20).

10.1.2 All of these ASN.1 names are generated by applying 10.3 either to the **name** of the corresponding schema component, or to a member of the **value** of an **enumeration** facet, or to a specified character string, as specified in the relevant clauses of this Recommendation | International Standard.

10.2 Generating ASN.1 type definitions that are references to ASN.1 type assignments

10.2.1 This subclause applies as explicitly invoked by other clauses of this Recommendation | International Standard to generate an ASN.1 type definition that is a reference (a "DefinedType") to an ASN.1 type assignment.

10.2.2 If an ASN.1 type definition (R, say) that is a "DefinedType" is to be inserted in an ASN.1 module (M, say) other than the ASN.1 module where the referenced ASN.1 type assignment (TA, say) is being inserted, and the type reference name of TA is identical to either the type reference name of another ASN.1 type assignment being inserted in module M or to another type reference name being imported into module M, then R shall be an "ExternalTypeReference" (constructed as appropriate for module M) for TA; otherwise it shall be a "typereference" for TA.

10.3 Generating identifiers and type reference names

10.3.1 This subclause applies as explicitly invoked by other clauses of this Recommendation | International Standard to generate an ASN.1 type reference name or identifier.

10.3.2 **Name**s of **attribute declaration**s, **element declaration**s, **model group definition**s, user-defined top-level **simple type definition**s, and top-level **complex type definition**s can be identical to ASN.1 reserved words or can contain characters not allowed in ASN.1 identifiers or in ASN.1 type reference names. In addition, there are cases in which ASN.1 names are required to be distinct where the **name**s of the corresponding XSD schema components (from which the ASN.1 names are mapped) are allowed to be identical.

10.3.3 The following transformations shall be applied, in order, to each character string being mapped to an ASN.1 name, where each transformation (except the first) is applied to the result of the previous transformation:

- the characters " " (SPACE), "**.**" (FULL STOP), and "**_**" (LOW LINE) shall all be replaced by a "-" (HYPHEN-MINUS); and
- any character except "A" to "Z" (LATIN CAPITAL LETTER A to LATIN CAPITAL LETTER Z), "a" to "z" (LATIN SMALL LETTER A to LATIN SMALL LETTER Z), "0" to "9" (DIGIT ZERO to DIGIT NINE), and "-" (HYPHEN-MINUS) shall be removed; and
- a sequence of two or more HYPHEN-MINUS characters shall be replaced with a single HYPHEN-MINUS; and
- HYPHEN-MINUS characters occurring at the beginning or at the end of the name shall be removed; and
- if a character string that is to be used as a type reference name starts with a lower-case letter, the first letter shall be capitalized (converted to upper-case); if it starts with a digit (DIGIT ZERO to DIGIT NINE), it shall be prefixed with an "**X**" (LATIN CAPITAL LETTER X) character; and
- if a character string that is to be used as an identifier starts with an upper-case letter, the first letter shall be uncapitalized (converted to lower-case); if it starts with a digit (DIGIT ZERO to DIGIT NINE), it shall be prefixed with an "**x**" (LATIN SMALL LETTER X) character; and
- if a character string that is to be used as a type reference name is empty, it shall be replaced by "**X**" (LATIN CAPITAL LETTER X); and
- if a character string that is to be used as an identifier is empty, it shall be replaced by "**x**" (LATIN SMALL LETTER X).

10.3.4 Depending on the kind of name being generated, one of the three following subclauses applies.

10.3.4.1 If the name being generated is the type reference name of an ASN.1 type assignment and the character string generated by 10.3.3 is identical to the type reference name of another ASN.1 type assignment previously generated in the same ASN.1 module or in another ASN.1 module with the same namespace (including absence of a namespace), or is one of the reserved words specified in ITU-T Rec. X.680 | ISO/IEC 8824-1, 11.27, then a suffix shall be appended to the character string generated by 10.3.3. The suffix shall consist of a HYPHEN-MINUS followed by the canonical lexical representation (see W3C XML Schema Part 2, 2.3.1) of an integer. This integer shall be the least positive integer such that the new name is different from the type reference name of any other ASN.1 type assignment previously generated in any of those ASN.1 modules.

10.3.4.2 If the name being generated is the identifier of a component of a sequence, set, or choice type, and the character string generated by 10.3.3 is identical to the identifier of a previously generated component of the same sequence, set, or choice type, then a suffix shall be appended to the character string generated by 10.3.3. The suffix shall consist of a HYPHEN-MINUS followed by the canonical lexical representation (see W3C XML Schema Part 2, 2.3.1) of an integer. This integer shall be the least positive integer such that the new identifier is different from the identifier of any previously generated component of that sequence, set, or choice type.

10.3.4.3 If the name being generated is the "identifier" in an "EnumerationItem" of an enumerated type, and the character string generated by 10.3.3 is identical to the "identifier" in another "EnumerationItem" previously generated in the same enumerated type, then a suffix shall be appended to the character string generated by 10.3.3. The suffix shall consist of a HYPHEN-MINUS followed by the canonical lexical representation (see W3C XML Schema Part 2, 2.3.1) of an integer. This integer shall be the least positive integer such that the new identifier is different from the "identifier" in any other "EnumerationItem" already present in that ASN.1 enumerated type.

10.3.5 For an ASN.1 type reference name (or identifier) that is generated by applying this subclause 10.3 to the **name** of an **element declaration**, **attribute declaration**, top-level **complex type definition** or user-defined top-level **simple type definition**, if the type reference name (or identifier) generated is different from the **name**, a final **NAME** encoding instruction shall be assigned to the ASN.1 type assignment with that type reference name (or to the component with that identifier) as specified in the three following subclauses.

10.3.5.1 If the only difference is the case of the first letter (which is upper case in the type reference name and lower case in the **name**), then the "Keyword" in the **NAME** encoding instruction shall be **UNCAPITALIZED**.

10.3.5.2 If the only difference is the case of the first letter (which is lower case in the identifier and upper case in the **name**), then the "Keyword" in the **NAME** encoding instruction shall be **CAPITALIZED**.

10.3.5.3 Otherwise, the "NewName" in the **NAME** encoding instruction shall be the **name**.

EXAMPLE – The top-level **complex type definition**:

```
<xsd:complexType name="COMPONENTS">
        <xsd:sequence>
                <xsd:element name="Elem" type="xsd:boolean"/>
                <xsd:element name="elem" type="xsd:integer"/>
                <xsd:element name="Elem-1" type="xsd:boolean"/>
                <xsd:element name="elem-1" type="xsd:integer"/>
        </xsd:sequence>
</xsd:complexType>
```


is mapped to the ASN.1 type assignment:

```
COMPONENTS-1 ::= [NAME AS "COMPONENTS"] SEQUENCE {
      elem     [NAME AS CAPITALIZED] BOOLEAN,
      elem-1   [NAME AS "elem"] INTEGER,
      elem-1-1 [NAME AS "Elem-1"] BOOLEAN,
      elem-1-2 [NAME AS "elem-1"] INTEGER }
```

10.3.6 For an ASN.1 type reference name (or identifier) that is generated by applying this subclause 10.3 to the **name** of an **element declaration, attribute declaration,** top-level **complex type definition** or user-defined top-level **simple type definition**, if the **target namespace** of the schema component is not **absent**, then a final **NAMESPACE** encoding instruction shall be assigned to the ASN.1 type assignment with that type reference name (or to the named type with that identifier) and shall specify the **target namespace** of the schema component.

10.3.7 For an ASN.1 identifier that is generated by this subclause 10.3 for the mapping of a **simple type definition** with an **enumeration** facet where the identifier generated is different from the corresponding member of the **value** of the **enumeration** facet, a final **TEXT** encoding instruction shall be assigned to the ASN.1 enumerated type, with qualifying information specifying the "identifier" in the "EnumerationItem" of the enumerated type. One of the two following subclauses applies.

10.3.7.1 If the only difference is the case of the first letter (which is lower case in the identifier and upper case in the member of the **value** of the **enumeration** facet), then the "Keyword" in the **TEXT** encoding instruction shall be **CAPITALIZED.**

10.3.7.2 Otherwise, the "NewName" in the **TEXT** encoding instruction shall be the member of the **value** of the **enumeration** facet.

10.4 Order of the mapping

10.4.1 An order is imposed on the top-level schema components of the source XSD Schema on which the mapping is performed. This applies to **model group definition**s, top-level **complex type definition**s, user-defined top-level **simple type definition**s, top-level **attribute declaration**s, and top-level **element declaration**s.

NOTE – Other top-level schema components are not mapped to ASN.1, and XSD built-in datatypes are mapped in a special way.

10.4.2 The order is specified in the three following subclauses.

10.4.2.1 Top-level schema components shall first be ordered by their **target namespace**, with the **absent** namespace preceding all namespace names in ascending lexicographical order.

10.4.2.2 Within each target namespace, top-level schema components shall be divided into four sets ordered as follows:

a) **element declaration**s;

b) **attribute declaration**s;

c) **complex type definition**s and **simple type definition**s;

d) **model group definition**s.

10.4.2.3 Within each set (see 10.4.2.2), schema components shall be ordered by **name** in ascending lexicographical order.

10.4.3 The mapping generates some ASN.1 type assignments that do not correspond directly to any XSD schema component. These are:

a) choice types (with a final **`USE-TYPE`** encoding instruction) corresponding to a type derivation hierarchy; the type reference names of these types have a "**`-derivations`**" suffix (see clause 29);

b) choice types (with a final **`USE-TYPE`** encoding instruction on the type and a final **`USE-NIL`** encoding instruction on each alternative) corresponding to a type derivation hierarchy where the user-defined top-level **simple type definition** or **complex type definition** that is the root of the derivation hierarchy is used as the **type definition** of one or more **element declaration**s that are **nillable**; the type reference names of these types have a "**`-deriv-nillable`**" suffix (see clause 29);

c) choice types (with a final **`USE-TYPE`** encoding instruction on the type and a final **`DEFAULT-FOR-EMPTY`** encoding instruction on each alternative) corresponding to a type derivation hierarchy where the user-defined top-level **simple type definition** or **complex type definition** that is the root of the derivation hierarchy is used as the **type definition** of one or more **element declaration**s that are not **nillable** and have a **value constraint** that is a **default** value; the type reference names of these types have a "**`-deriv-default-`**" suffix (see clause 29);

d) choice types (with a final **`USE-TYPE`** encoding instruction on the type and a final **`DEFAULT-FOR-EMPTY`** encoding instruction on each alternative) corresponding to a type derivation hierarchy where the user-defined top-level **simple type definition** or **complex type definition** that is the root of the derivation hierarchy is used as the **type definition** of one or more **element declaration**s that are not **nillable** and have a **value constraint** that is a **fixed** value; the type reference names of these types have a "**`-deriv-fixed-`**" suffix (see clause 29);

e) choice types (with a final **`USE-TYPE`** encoding instruction on the type and final **`USE-NIL`** and **`DEFAULT-FOR-EMPTY`** encoding instructions on each alternative) corresponding to a type derivation hierarchy where the user-defined top-level **simple type definition** or **complex type definition** that is the root of the derivation hierarchy is used as the **type definition** of one or more **element declaration**s that are **nillable** and have a **value constraint** that is a **default** value; the type reference names of these types have a "**`-deriv-nillable-default-`**" suffix (see clause 29);

f) choice types (with a final **`USE-TYPE`** encoding instruction on the type and final **`USE-NIL`** and **`DEFAULT-FOR-EMPTY`** encoding instructions on each alternative) corresponding to a type derivation hierarchy where the user-defined top-level **simple type definition** or **complex type definition** that is the root of the derivation hierarchy is used as the **type definition** of one or more **element declaration**s that are **nillable** and have a **value constraint** that is a **fixed** value; the type reference names of these types have a "**`-deriv-nillable-fixed-`**" suffix (see clause 29);