INTERNATIONAL STANDARD

ISO/IEC 8824-1

Second edition
1998-12-15

Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation

Technologies de l'information — Notation de syntaxe abstraite numéro un (ASN.1): Spécification de la notation de base

Reference number
ISO/IEC 8824-1:1998(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Published by ISO in 1999

Printed 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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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 part of ISO/IEC 8824 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 8824-1 was prepared 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 Recommendation X.680.

This second edition cancels and replaces the first edition (ISO/IEC 8824-1:1995), which has been technically revised. It also incorporates Amendment 1:1996 and Technical Corrigendum 1:1996.

ISO/IEC 8824 consists of the following parts, under the general title *Information technology — Abstract Syntax Notation One (ASN.1)*:

— *Part 1: Specification of basic notation*

— *Part 2: Information object specification*

— *Part 3: Constraint specification*

— *Part 4: Parameterization of ASN.1 specifications*

Annexes A and B form a normative part of this part of ISO/IEC 8824. Annexes C to G are for information only.

Introduction

This Recommendation | International Standard presents a standard notation for the definition of data types and values. A *data type* (or *type* for short) is a class of information (for example, numeric, textual, still image or video information). A *data value* (or *value* for short) is an instance of such a class. This Recommendation | International Standard defines several basic types and their corresponding values, and rules for combining them into more complex types and values.

Although this standard notation is defined within the OSI framework, it can be used for many other purposes. In the lower layers of the OSI Basic Reference Model (see ITU-T Rec. X.200 | ISO/IEC 7498-1) and in many other protocol architectures, each message is specified as the binary value of a sequence of octets. In the Presentation layer of OSI (see ITU-T Rec. X.216 | ISO/IEC 8822), the nature of user data parameters changes. However, Application layer standards need to define quite complex data types to carry their messages, without concern for their binary representation. In order to specify the data types, they require a notation which does not necessarily determine the representation of each value. Such notation has to be supplemented by the specification of one or more algorithms called **encoding rules** which determine the value of the lower layer octets that carry the Application data (called the **transfer syntax**). The Presentation layer protocol of OSI (see ITU-T Rec. X.226 | ISO/IEC 8823-1) can negotiate which transfer syntaxes (**encodings**) are to be used.

Outside the context of OSI there is increasing recognition of the notion of an abstract value of some class (e.g. a particular 256-colour picture) divorced from the details of any particular encoding where in order to correctly interpret the bit-pattern representation of the value, it is necessary to know (usually from the context), the type (class) of the value being represented, as well as the encoding mechanism being employed. Thus, the identification of a type is an important part of this Recommendation | International Standard.

A very general technique for defining a complicated type at the abstract level is to define a small number of **simple types** by defining all possible values of the simple types, then combining these simple types in various ways. Some of the ways of defining new types are as follows:

a) given an (ordered) list of existing types, a value can be formed as an (ordered) sequence of values, one from each of the existing types; the collection of all possible values obtained in this way is a new type; (if the existing types in the list are all distinct, this mechanism can be extended to allow omission of some values from the list);

b) given an unordered set of (distinct) existing types, a value can be formed as an (unordered) set of values, one from each of the existing types; the collection of all possible unordered sets of values obtained in this way is a new type; (the mechanism can again be extended to allow omission of some values);

c) given a single existing type, a value can be formed as an (ordered) list or (unordered) set of zero, one or more values of the existing type; the collection of all possible lists or sets of values obtained in this way is a new type;

d) given a list of (distinct) types, a value can be chosen from any one of them; the set of all possible values obtained in this way is a new type;

e) given a type, a new type can be formed as a subset of it by using some structure or order relationship among the values.

An important aspect of combining types in this way is that encoding rules should recognize the combining constructs, providing unambiguous encodings of the collection of values of the basic types. Thus, every basic type defined using the notation specified in this Recommendation | International Standard is assigned a **tag** to aid in the unambiguous encoding of values.

Four classes of tag are specified in the notation.

The first is the **universal** class. Universal class tags are only used as specified within this Recommendation | International Standard, and each tag is either:

a) assigned to a single type; or

b) assigned to a construction mechanism.

Users of this notation are not allowed to explicitly specify universal class tags in their ASN.1 specifications, for these tags are built-in and can be specified explicitly only in this Recommendation | International Standard.

The other three classes of tag are called **application** class tags, **private** class tags, and **context-specific** class tags. There is no formal difference between use of tags from these three classes. Where application class tags are employed, a private or context-specific class tag could generally be applied instead, as a matter of user choice and style. The presence of the three classes is largely for historical reasons, but guidance is given in C.2.12 on the way in which the classes are usually employed.

Tags are mainly intended for machine use, and are not essential for the human notation defined in this Recommendation | International Standard. Where, however, it is necessary to require that certain types be distinct, this is expressed by requiring that they have distinct tags. The allocation of tags is therefore an important part of the use of this notation.

NOTE – Within this Recommendation | International Standard, tag values are assigned to all simple types and construction mechanisms. The restrictions placed on the use of the notation ensure that tags can be used in transfer for unambiguous identification of values.

An ASN.1 specification will initially be produced with a set of fully defined ASN.1 types. At a later stage, however, it may be necessary to change those types (usually by the addition of extra components in a sequence or set type). If this is to be possible in such a way that implementations using the old type definitions can interwork with implementations using the new type definitions in a defined way, encoding rules need to provide appropriate support. The ASN.1 notation supports the inclusion of an **extension marker** on a number of types. This signals to encoding rules the intention of the designer that this type is one of a series of related types (i.e. versions of the same initial type) called an **extension series**, and that the encoding rules are required to enable information transfer between implementations using different types that are related by being part of the same extension series.

Clauses 10 to 31 (inclusive) define the simple types supported by ASN.1, and specify the notation to be used for referencing simple types and for defining new types using them. Clauses 10 to 31 also specify the notation to be used for specifying values of types defined using ASN.1.

Clauses 32 to 33 (inclusive) define the types supported by ASN.1 for carrying within them the complete encoding of ASN.1 types.

Clauses 34 to 39 (inclusive) define the character string types.

Clauses 40 to 43 (inclusive) define certain types which are considered to be of general utility, but which require no additional encoding rules.

Clauses 44 and 48 define a notation which enables subtypes to be defined from the values of a parent type.

Annex A forms an integral part of this Recommendation | International Standard, and gives guidance on how users of this Recommendation | International Standard can refer to ASN.1 types and values defined using CCITT Rec. X.208 | ISO/IEC 8824.

Annex B forms an integral part of this Recommendation | International Standard, and records object identifier and object descriptor values assigned in this Recommendation | International Standard.

Annex C does not form an integral part of this Recommendation | International Standard, and provides examples and hints on the use of the ASN.1 notation.

Annex D does not form an integral part of this Recommendation | International Standard, and provides a tutorial on ASN.1 character strings.

Annex E does not form an integral part of this Recommendation | International Standard, and describes features of the previous version of ASN.1 that have been superseded.

Annex F does not form an integral part of this Recommendation | International Standard, and provides a tutorial on the ASN.1 model of type extension.

Annex G does not form an integral part of this Recommendation | International Standard, and provides a summary of ASN.1 using the notation of clause 5.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – ABSTRACT SYNTAX NOTATION ONE (ASN.1): SPECIFICATION OF BASIC NOTATION

1 Scope

This Recommendation | International Standard provides a standard notation called Abstract Syntax Notation One (ASN.1) that is used for the definition of data types, values, and constraints on data types.

This Recommendation | International Standard

- defines a number of simple types, with their tags, and specifies a notation for referencing these types and for specifying values of these types;
- defines mechanisms for constructing new types from more basic types, and specifies a notation for defining such types and assigning them tags, and for specifying values of these types;
- defines character sets (by reference to other Recommendations and/or International Standards) for use within ASN.1;
- defines a number of useful types (using ASN.1), which can be referenced by users of ASN.1;

The ASN.1 notation can be applied whenever it is necessary to define the abstract syntax of information. It is particularly, but not exclusively, applicable to application protocols.

The ASN.1 notation is referenced by other standards which define encoding rules for the ASN.1 types.

2 Normative references

The following Recommendations and International Standards 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 and Standards 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 and Standards 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.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.*
- ITU-T Recommendation X.216 (1994) | ISO/IEC 8822:1994, *Information technology – Open Systems Interconnection – Presentation service definition*.
- ITU-T Recommendation X.226 (1994) | ISO/IEC 8823-1:1994, *Information technology – Open Systems Interconnection – Connection-oriented presentation protocol: Protocol specification.*
- ITU-T Recommendation X.660 (1992)/Amd.2(1997) | ISO/IEC 9834-1:1993/Amd.2:1998, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures (plus Amendments 1 and 2)*.

- ITU-T Recommendation X.681 (1997) | ISO/IEC 8824-2:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.

- ITU-T Recommendation X.682 (1997) | ISO/IEC 8824-3:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.

- ITU-T Recommendation X.683 (1997) | ISO/IEC 8824-4:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*.

- ITU-T Recommendation X.690 (1997) | ISO/IEC 8825-1:1998, *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 (1997) | ISO/IEC 8825-2:1998, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1)*.

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*.

2.3 Additional references

- CCITT Recommendation T.61 (1988), *Character repertoire and coded character sets for the international teletex service*.

- CCITT Recommendation T.100 (1988), *International information exchange for interactive videotex.*

- ITU-T Recommendation T.101 (1994), *International interworking for videotex services.*

- ISO *International Register of Coded Character Sets to be used with Escape Sequences*.

- ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange*.

- ISO/IEC 2022:1994, *Information technology – Character code structure and extension techniques*.

- ISO 6523:1984, *Data interchange – Structures for the identification of organizations*.

- ISO/IEC 7350:1991, *Information technology – Registration of repertoires of graphic characters from ISO 10367*.

- ISO 8601:1988, *Data elements and interchange formats – Information interchange – Representation of dates and times*.

- ISO/IEC 10646-1:1993, *Information technology – Universal Multiple-Octet Coded Character Set (UCS – Part 1: Architecture and Basic Multilingual Plane*.

- ISO/IEC 10646-1:1993/Amd.2:1996, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane – Amendment 2: UCS Transformation Format 8 (UTF-8)*.

3 Definitions

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

3.1 Information object specification

This Recommendation | International Standard uses the following terms defined in ITU-T Rec. X.681 | ISO/IEC 8824-2:

a) information object;
b) information object class;
c) information object set;
d) instance-of type;
e) object class field type.

3.2 Constraint specification

This Recommendation | International Standard uses the following terms defined in ITU-T Rec. X.682 | ISO/IEC 8824-3:

a) component relation constraint;
b) table constraint.

3.3 Parameterization of ASN.1 specification

This Recommendation | International Standard uses the following terms defined in ITU-T Rec. X.683 | ISO/IEC 8824-4:

a) parameterized type;
b) parameterized value.

3.4 Presentation service definition

This Recommendation | International Standard uses the following terms defined in ITU-T Rec. X.216 | ISO/IEC 8822:

a) (an) abstract syntax;
b) abstract syntax name;
c) defined context set;
d) presentation data value;
e) (a) transfer syntax;
f) transfer syntax name.

3.5 Presentation protocol specification

This Recommendation | International Standard uses the following term defined in ITU-T Rec. X.226 | ISO/IEC 8823-1:

– presentation context identifier.

3.6 Structure for identification of organizations

This Recommendation | International Standard uses the following terms defined in ISO 6523:

a) issuing organization;
b) organization code;
c) International Code Designator.

3.7 Universal Multiple-Octet Coded Character Set (UCS)

This Recommendation | International Standard uses the following terms defined in ISO/IEC 10646-1:

a) Basic Multilingual Plane (BMP);
b) cell;
c) combining character;

d) graphic symbol;
e) group;
f) limited subset;
g) plane;
h) row;
i) selected subset.

3.8 Additional definitions

3.8.1 abstract character: The set of information associated with a cell in a table defining a character repertoire.

NOTE – The information will normally include some or all of the following items:

a) a graphic symbol;

b) a character name; or

c) the definition of functions associated with the character when used in particular environments.

3.8.2 abstract value: A value whose definition is based only on the type, independent of how it is represented in any encoding rule.

NOTE – Use of the term "abstract value" is frequently an assertion that what is being said probably varies based upon the encoding rules used.

3.8.3 ASN.1 character set: The set of characters, specified in clause 10, used in the ASN.1 notation.

3.8.4 ASN.1 specification: A collection of one or more ASN.1 modules.

3.8.5 associated type: A type which is used only for defining the value and subtype notation for a type.

NOTE – Associated types are defined in this Recommendation | International Standard when it is necessary to make it clear that there may be a significant difference between how the type is defined in ASN.1 and how it is encoded. Associated types do not appear in user specifications.

3.8.6 bitstring type: A simple type whose distinguished values are an ordered sequence of zero, one or more bits.

NOTE – Where there is a need to carry embedded encodings of an abstract value, the use of the embedded-pdv type will in general provide a more flexible mechanism for announcement or agreement on the nature of the encodings than the bitstring type.

3.8.7 boolean type: A simple type with two distinguished values.

3.8.8 character: A member of a set of elements used for the organization, control or representation of data.

NOTE – For example, this implies that an accent combining character and lower case 'e' are two characters in the ISO 646 French Version, and not the single character é.

3.8.9 character abstract syntax: Any abstract syntax whose values are specified as the set of character strings of zero, one or more characters from some specified collection of characters.

3.8.10 character repertoire: The characters in a character set without any implication on how such characters are encoded.

3.8.11 character string types: Simple types whose values are strings of characters from some defined character set.

3.8.12 character transfer syntax: Any transfer syntax for a character abstract syntax.

NOTE – ASN.1 does not support character transfer syntaxes which do not encode all character strings as an integral multiple of 8 bits.

3.8.13 choice types: Types defined by referencing a list of distinct types; each value of the choice type is derived from the value of one of the component types.

3.8.14 component type: One of the types referenced when defining a CHOICE, SET, SEQUENCE, SET OF, or SEQUENCE OF.

3.8.15 constraint: A notation which can be used in association with a type, to define a subtype of that type.

3.8.16 control characters: Characters appearing in some character repertoires that have been given a name (and perhaps a defined function in relation to certain environments) but which have not been assigned a graphic symbol, and which are not spacing characters.

NOTE – NEWLINE and TAB are examples of control characters that have been assigned a formatting function in a printing environment. DLE is an example of a control character that has been assigned a function in a communication environment.

3.8.17 Coordinated Universal Time (UTC): The time scale maintained by the Bureau International de l'Heure (International Time Bureau) that forms the basis of a coordinated dissemination of standard frequencies and time signals.

NOTE 1 – The source of this definition is Recommendation 460-2 of the Consultative Committee on International Radio (CCIR). CCIR has also defined the acronym for Coordinated Universal Time as UTC.

NOTE 2 – UTC and Greenwich Mean Time are two alternative time standards which for most practical purposes determine the same time.

3.8.18 element: A member of an element class, distinguishable from all other elements of the same class.

3.8.19 element class: A type (whose elements are its values) or information object class (whose elements are all possible objects of that class).

3.8.20 element set: One or more elements of the same element class.

3.8.21 embedded-pdv type: A type whose set of values is the union of the sets of values in all possible abstract syntaxes. This type is a part of an ASN.1 specification that carries a value whose type may be defined externally to that ASN.1 specification. It also carries an identification of the type of the value being carried as well as an identification of the encoding rule used to encode the value.

3.8.22 encoding: The bit-pattern resulting from the application of a set of encoding rules to a value of a specific abstract syntax.

3.8.23 (ASN.1) encoding rules: Rules which specify the representation during transfer of the values of ASN.1 types. Encoding rules also enable the values to be recovered from the representation, given knowledge of the type.

NOTE – For the purpose of specifying encoding rules, the various referenced type (and value) notations, which can provide alternative notations for built-in types (and values), are not relevant.

3.8.24 enumerated types: Simple types whose values are given distinct identifiers as part of the type notation.

3.8.25 extension addition: One of the added notations in an extension series. For set, sequence and choice types, each extension addition is the addition of either a single extension addition group or a single component type. For enumerated types it is the addition of a single further enumeration. For a constraint it is the addition of a subtype element.

NOTE – Extension additions are both textually ordered (following the extension marker) and logically ordered (having increasing enumeration values, and, in the case of CHOICE alternatives, increasing tags.)

3.8.26 extension addition group: One or more components of a set, sequence or choice type grouped within version brackets. An extension addition group is used to clearly identify the components of a set, sequence or choice type that were added in a particular version of an ASN.1 module.

3.8.27 extension addition type: A type contained within an extension addition group or a single component type that is itself an extension addition (in such case it is not contained within an extension addition group).

3.8.28 extensible constraint: A subtype constraint with an extension marker.

3.8.29 extension insertion point: The location within a type definition where extension additions are inserted. This location is the end of the type notation of the immediately preceding type in the extension series if there is a single ellipsis in the type definition, or immediately before the second ellipsis if there is an extension marker pair in the definition of the type.

3.8.30 extension marker: A syntactic flag (an ellipsis) that is included in all types that form part of an extension series.

3.8.31 extension marker pair: A pair of extension markers between which extension additions are inserted.

3.8.32 extension-related: Two types that have the same extension root, where one was created by adding zero or more extension additions to the other.

3.8.33 extension root: An extensible type that is the first type in an extension series. It carries either the extension marker with no additional notation other than comments and white-space between the extension marker and the matching "}" or ")", or an extension marker pair with no additional notation other than a single comma, comments and white-space between the extension markers.

NOTE – Only an extension root can be the first type in an extension series.

3.8.34 extension series: A series of ASN.1 types which can be ordered in such a way that each successive type in the series is formed by the addition of text at the extension insertion point.

NOTE – Both nested and unnested types can be extended.

3.8.35 extensible type: A type with an extension marker.

3.8.36 external reference: A type reference, value reference, information object, etc., that is defined in some other module than the one in which it is being referenced, and which is being referred to by prefixing the module name to the referenced item.

EXAMPLE – ModuleName.TypeReference

3.8.37 external type: A type which is a part of an ASN.1 specification that carries a value whose type may be defined externally to that ASN.1 specification. It also carries an identification of the type of the value being carried.

3.8.38 false: One of the distinguished values of the boolean type (see "true").

3.8.39 governing; governor: Relative to some object, object set, value set, value or subtype, the information object class or type which controls its interpretation by restricting the items(s) involved to be value notation of that class or type, respectively.

3.8.40 integer type: A simple type with distinguished values which are the positive and negative whole numbers, including zero (as a single value).

NOTE – Particular encoding rules limit the range of an integer, but such limitations are chosen so as not to affect any user of ASN.1.

3.8.41 items: Named sequences of characters from the ASN.1 character set, specified in clause 11, which are used to form the ASN.1 notation.

3.8.42 module: One or more instances of the use of the ASN.1 notation for type, value, etc., encapsulated using the ASN.1 module notation (see clause 12).

3.8.43 null type: A simple type consisting of a single value, also called null.

3.8.44 object: A well-defined piece of information, definition, or specification which requires a name in order to identify its use in an instance of communication.

3.8.45 object descriptor type: A type whose distinguished values are human-readable text providing a brief description of an object.

NOTE – An object descriptor value is usually associated with a single object. Only an object identifier value unambiguously identifies an object.

3.8.46 object identifier: A value (distinguishable from all other such values) which is associated with an object.

3.8.47 object identifier type: A simple type whose distinguished values are the set of all object identifiers allocated in accordance with the rules of ITU-T Rec. X.660 | ISO/IEC 9834-1.

NOTE – The rules of ITU-T Rec. X.660 | ISO/IEC 9834-1 permit a wide range of authorities to independently associate object identifiers with objects.

3.8.48 octetstring type: A simple type whose distinguished values are an ordered sequence of zero, one or more octets, each octet being an ordered sequence of eight bits.

3.8.49 open type notation: An ASN.1 notation used to denote a set of values from more than one ASN.1 type.

NOTE 1 – The term "open type" is used synonymously with "open type notation" in the body of this Recommendation | International Standard.

NOTE 2 – All ASN.1 encoding rules provide unambiguous encodings for the values of a single ASN.1 type. They do not necessarily provide unambiguous encodings for "open type notation", which carries values from ASN.1 types that are not normally determined at specification time. Knowledge of the type of the value being encoded in the "open type notation" is needed before the abstract value for that field can be unambiguously determined.

NOTE 3 – The only notation in this Recommendation | International Standard which is an open type notation is the "ObjectClassFieldType" specified in ITU-T Rec. X.681 | ISO/IEC 8824-2, where the "FieldName" denotes either a type field or a variable-type value field. The "ANY" notation which was defined in CCITT Rec. X.208 | ISO/IEC 8824 was an open type notation.

3.8.50 parent type (of a subtype): The type that is being constrained when defining a subtype.

NOTE – The parent type may itself be a subtype of some other type.

3.8.51 production: A part of the formal notation used to specify ASN.1.

3.8.52 real type: A simple type whose distinguished values (specified in clause 20) are members of the set of real numbers.

3.8.53 recursive definitions: A set of ASN.1 definitions which cannot be reordered so that all types used in a construction are defined before the definition of the construction.

NOTE – Recursive definitions are allowed in ASN.1: the user of the notation has the responsibility for ensuring that those values (of the resulting types) which are used have a finite representation.

3.8.54 restricted character string type: A character string type whose characters are taken from a fixed character repertoire identified in the type specification.

3.8.55 selection types: Types defined by reference to a component type of a choice type, and whose values are precisely the values of that component type.

3.8.56 sequence types: Types defined by referencing an ordered list of types (some of which may be declared to be optional); each value of the sequence type is an ordered list of values, one from each component type.

NOTE – Where a component type is declared to be optional, a value of the sequence type need not contain a value of that component type.

3.8.57 sequence-of types: Types defined by referencing a single component type; each value in the sequence-of type is an ordered list of zero, one or more values of the component type.

3.8.58 set types: Types defined by referencing a fixed, unordered, list of distinct types (some of which may be declared to be optional); each value in the set type is an unordered list of values, one from each of the component types.

NOTE – Where a component type is declared to be optional, the new type need not contain a value of that component type.

3.8.59 set-of types: Types defined by referencing a single component type; each value in the set-of type is an unordered list of zero, one or more values of the component type.

3.8.60 simple types: Types defined by directly specifying the set of its values.

3.8.61 spacing character: A character in a character repertoire which is intended for inclusion with graphic characters in the printing of a character string but which is represented in the physical rendition by empty space; it is not normally considered to be a control character (see 3.8.16).

NOTE – There may be a single spacing character in the character repertoire, or there may be multiple spacing characters with varying widths.

3.8.62 subtype (of a parent type): A type whose values are a subset (or the complete set) of the values of some other type (the parent type).

3.8.63 tag: A type denotation which is associated with every ASN.1 type.

3.8.64 tagged types: A type defined by referencing a single existing type and a tag; the new type is isomorphic to the existing type, but is distinct from it.

3.8.65 tagging: Replacing the existing (possibly the default) tag of a type by a specified tag.

3.8.66 true: One of the distinguished values of the boolean type (see "false").

3.8.67 type: A named set of values.

3.8.68 type reference name: A name associated uniquely with a type within some context.

NOTE – Reference names are assigned to the types defined in this Recommendation | International Standard; these are universally available within ASN.1. Other reference names are defined in other Recommendations | International Standards, and are applicable only in the context of that Recommendation | International Standard.

3.8.69 unrestricted character string type: A type whose values are values from a character abstract syntax identified separately for each instance of use of that type.

3.8.70 user (of ASN.1): The individual or organization that defines the abstract syntax of a particular piece of information using ASN.1.

3.8.71 value: A distinguished member of a set of values.

3.8.72 value reference name: A name associated uniquely with a value within some context.

3.8.73 value set: A collection of values of a type. Semantically equivalent to a subtype.