

DRAFT INTERNATIONAL STANDARD

ISO/IEC DIS 8824-4

ISO/IEC JTC 1/SC 6

Secretariat: **KATS**

Voting begins on:
2020-09-28

Voting terminates on:
2020-12-21

Information technology — Abstract Syntax Notation One (ASN.1) —

Part 4: Parameterization of ASN.1 specifications

*Technologies de l'information — Notation de syntaxe abstraite numéro un (ASN.1) —
Partie 4: Paramétrage des spécifications de la notation de syntaxe abstraite numéro un*

ICS: 35.100.60

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC DIS 8824-4](#)

<https://standards.iteh.ai/catalog/standards/sist/dc666c64-7425-4dac-b5cd-2055f53a79c8/iso-iec-dis-8824-4>

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.



Reference number
ISO/IEC DIS 8824-4:2020(E)

© ISO/IEC 2020

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC DIS 8824-4

<https://standards.iteh.ai/catalog/standards/sist/dc666c64-7425-4dac-b5cd-2055f53a79c8/iso-iec-dis-8824-4>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

CONTENTS

	<i>Page</i>
Introduction	lvi
1 Scope	56
2 Normative references.....	56
2.1 Identical Recommendations International Standards.....	56
3 Definitions	56
3.1 Specification of basic notation	56
3.2 Information object specification.....	56
3.3 Constraint specification.....	56
3.4 Additional definitions.....	56
4 Abbreviations	56
5 Convention	56
6 Notation	56
6.1 Assignments	56
6.2 Parameterized definitions	56
6.3 Symbols.....	56
7 ASN.1 lexical items.....	56
8 Parameterized assignments.....	56
9 Referencing parameterized definitions.....	56
10 Abstract syntax parameters.....	56
Annex A – Examples	56
A.1 Example of the use of a parameterized type definition.....	56
A.2 Example of use of parameterized definitions together with an information object class.....	56
A.3 Example of parameterized type definition that is finite.....	56
A.4 Example of a parameterized value definition	56
A.5 Example of a parameterized value set definition.....	56
A.6 Example of a parameterized class definition.....	56
A.7 Example of a parameterized object set definition.....	56
A.8 Example of a parameterized object set definition.....	56
Annex B – Summary of the notation.....	56

Introduction

Application designers need to write specifications in which certain aspects are left undefined. Those aspects will later be defined by one or more other groups (each in its own way), to produce a fully defined specification for use in the definition of an abstract syntax (one for each group).

In some cases, aspects of the specification (for example, bounds) may be left undefined even at the time of abstract syntax definition, being completed by the specification of International Standardized Profiles or functional profiles from some other body.

NOTE 1 – It is a requirement imposed by this Recommendation | International Standard that any aspect that is not solely concerned with the application of constraints has to be completed prior to the definition of an abstract syntax.

In the extreme case, some aspects of the specification may be left for the implementer to complete, and would then be specified as part of the Protocol Implementation Conformance Statement.

While the provisions of Rec. ITU-T X.681 | ISO/IEC 8824-2 and Rec. ITU-T X.682 | ISO/IEC 8824-3 provide a framework for the later completion of parts of a specification, they do not of themselves solve the above requirements.

Additionally, a single designer sometimes requires to define many types, or many information object classes, or many information object sets, or many information objects, or many values, which have the same outer level structure, but differ in the types, or information object classes, or information object sets, or information objects, or values, that are used at an inner level. Instead of writing out the outer level structure for every such occurrence, it is useful to be able to write it out once, with parts left to be defined later, then to refer to it and provide the additional information.

All these requirements are met by the provision for parameterized reference names and parameterized assignments by this Recommendation | International Standard.

The syntactic form of a parameterized reference name is the same as that of the corresponding normal reference name, but the following additional considerations apply:

- When it is assigned in a parameterized assignment statement, it is followed by a list of dummy reference names in braces, each possibly accompanied by a governor; these reference names have a scope which is the right-hand side of the assignment statement, and the parameter list itself.
NOTE 2 – This is what causes it to be recognized as a parameterized reference name.
- When it is exported or imported, it is followed by a pair of empty braces to distinguish it as a parameterized reference name.
- When it is used in any construct, it is followed by a list of syntactic constructions, one for each dummy reference name, that provide an assignment to the dummy reference name for the purposes of that use only.

Dummy reference names have the same syntactic form as the corresponding normal reference name, and can be used anywhere on the right-hand side of the assignment statement that the corresponding normal reference name could be used. All such usages are required to be consistent.

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION**Information technology –
Abstract Syntax Notation One (ASN.1):
Parameterization of ASN.1 specifications****1 Scope**

This Recommendation | International Standard is part of Abstract Syntax Notation One (ASN.1) and defines notation for parameterization of ASN.1 specifications.

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

- Recommendation ITU-T X.680 (2020) | ISO/IEC 8824-1:2020, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- Recommendation ITU-T X.681 (2020) | ISO/IEC 8824-2:2020, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- Recommendation ITU-T X.682 (2020) | ISO/IEC 8824-3:2020, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*

3 Definitions

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

3.1 Specification of basic notation

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

3.2 Information object specification

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

3.3 Constraint specification

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

3.4 Additional definitions

3.4.1 normal reference name: A reference name defined, without parameters, by means of an "Assignment" other than a "ParameterizedAssignment". Such a name references a complete definition and is not supplied with actual parameters when used.

3.4.2 parameterized reference name: A reference name defined using a parameterized assignment, which references an incomplete definition and which, therefore, must be supplied with actual parameters when used.

3.4.3 parameterized type: A type defined using a parameterized type assignment and thus whose components are incomplete definitions which must be supplied with actual parameters when the type is used.

3.4.4 parameterized value: A value defined using a parameterized value assignment and thus whose value is incompletely specified and must be supplied with actual parameters when used.

3.4.5 parameterized value set: A value set defined using a parameterized value set assignment and thus whose values are incompletely specified and must be supplied with actual parameters when used.

3.4.6 parameterized object class: An information object class defined using a parameterized object class assignment and thus whose field specifications are incompletely specified and must be supplied with actual parameters when used.

3.4.7 parameterized object: An information object defined using a parameterized object assignment and thus whose components are incompletely specified and must be supplied with actual parameters when used.

3.4.8 parameterized object set: An information object set defined using a parameterized object set assignment and thus whose objects are incompletely specified and must be supplied with actual parameters when used.

3.4.9 variable constraint: A constraint employed in specifying a parameterized abstract syntax, and which depends on some parameter of the abstract syntax.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviation applies:

ASN.1 Abstract Syntax Notation One

5 Convention iTeh STANDARD PREVIEW

This Recommendation | International Standard employs the notational convention defined in Rec. ITU-T X.680 | ISO/IEC 8824-1, clause 5.

[ISO/IEC DIS 8824-4](https://standards.iteh.ai/catalog/standards/sist/dc666c64-7425-4dac-b5cd-2055f53a79c8/iso-iec-dis-8824-4)

6 Notation <https://standards.iteh.ai/catalog/standards/sist/dc666c64-7425-4dac-b5cd-2055f53a79c8/iso-iec-dis-8824-4>

This clause summarizes the notation defined in this Recommendation | International Standard.

6.1 Assignments

The following notation which can be used as an alternative for "Assignment" (see Rec. ITU-T X.680 | ISO/IEC 8824-1, clause 13) is defined in this Recommendation | International Standard:

- ParameterizedAssignment (see 8.1).

6.2 Parameterized definitions

6.2.1 The following notation which can be used as an alternative for "DefinedType" (see Rec. ITU-T X.680 | ISO/IEC 8824-1, 14.1) is defined in this Recommendation | International Standard:

- ParameterizedType (see 9.2).

6.2.2 The following notation which can be used as an alternative for "DefinedValue" (see Rec. ITU-T X.680 | ISO/IEC 8824-1, 14.1) is defined in this Recommendation | International Standard:

- ParameterizedValue (see 9.2).

6.2.3 The following notation which can be used as an alternative for "DefinedType" (see Rec. ITU-T X.680 | ISO/IEC 8824-1, 14.1) is defined in this Recommendation | International Standard:

- ParameterizedValueSetType (see 9.2).

6.2.4 The following notation which can be used as an alternative for "ObjectClass" (see Rec. ITU-T X.681 | ISO/IEC 8824-2, 9.2) is defined in this Recommendation | International Standard:

- ParameterizedObjectClass (see 9.2).

6.2.5 The following notation which can be used as an alternative for "Object" (see Rec. ITU-T X.681 | ISO/IEC 8824-2, 11.3) is defined in this Recommendation | International Standard:

- ParameterizedObject (see 9.2).

6.2.6 The following notation which can be used as an alternative for "ObjectSet" (see Rec. ITU-T X.681 | ISO/IEC 8824-2, 12.3) is defined in this Recommendation | International Standard:

- ParameterizedObjectSet (see 9.2).

6.3 Symbols

The following notation which can be used as an alternative for "Symbol" (see Rec. ITU-T X.680 | ISO/IEC 8824-1, 13.1) is defined in this Recommendation | International Standard:

- ParameterizedReference (see 9.1).

7 ASN.1 lexical items

This Recommendation | International Standard makes use of the lexical items specified in Rec. ITU-T X.680 | ISO/IEC 8824-1, clause 12.

8 Parameterized assignments

8.1 There are parameterized assignment statements corresponding to each of the assignment statements specified in Rec. ITU-T X.680 | ISO/IEC 8824-1 and Rec. ITU-T X.681 | ISO/IEC 8824-2. The "ParameterizedAssignment" construct is:

ParameterizedAssignment ::=
ParameterizedTypeAssignment
 | **ParameterizedValueAssignment**
 | **ParameterizedValueSetTypeAssignment**
 | **ParameterizedObjectClassAssignment**
 | **ParameterizedObjectAssignment**
 | **ParameterizedObjectSetAssignment**

8.2 Each "Parameterized<X>Assignment" has the same syntax as "<X>Assignment" except that following the initial lexical item there is a "ParameterList". The initial item thereby becomes a parameterized reference name (see 3.4.2):

NOTE 1 – Rec. ITU-T X.680 | ISO/IEC 8824-1 imposes the requirement that all reference names assigned within a module, whether parameterized or not, must be distinct.

NOTE 2 – Where value notation is governed by a parameterized type (or a type that is a parameter) the validity of value notation within the parameterized assignment can only be determined after instantiation of the parameterized type, and may be valid for some instantiations and invalid for others.

ParameterizedTypeAssignment ::=
typerference
ParameterList
 " : := "
Type

ParameterizedValueAssignment ::=
valuereference
ParameterList
Type
 " : := "
Value

ParameterizedValueSetTypeAssignment ::=
typerference
ParameterList
Type
 " : := "
ValueSet

ParameterizedObjectClassAssignment ::=

objectclassreference
ParameterList
 " : :=" "
ObjectClass

ParameterizedObjectAssignment ::=

objectreference
ParameterList
DefinedObjectClass
 " : :=" "
Object

ParameterizedObjectSetAssignment ::=

objectsetreference
ParameterList
DefinedObjectClass
 " : :=" "
ObjectSet

8.3 A "ParameterList" is a list of "Parameter"s between braces:

ParameterList ::= " { " Parameter " , " + " } "

Each "Parameter" consists of a "DummyReference" and possibly a "ParamGovernor":

Parameter ::= ParamGovernor " : " DummyReference | DummyReference

ParamGovernor ::= Governor | DummyGovernor

Governor ::= Type | DefinedObjectClass

DummyGovernor ::= DummyReference

DummyReference ::= Reference

A "DummyReference" in "Parameter" may stand for:

- a "Type" or "DefinedObjectClass", in which case there shall be no "ParamGovernor";
- a "Value" or "ValueSet", in which case the "ParamGovernor" shall be present, and in case "ParamGovernor" is a "Governor" it shall be a "Type", and in case "ParamGovernor" is a "DummyGovernor" the actual parameter for the "ParamGovernor" shall be a "Type";
- an "Object" or "ObjectSet", in which case the "ParamGovernor" shall be present, and in case "ParamGovernor" is a "Governor" it shall be a "DefinedObjectClass", and in case "ParamGovernor" is a "DummyGovernor" the actual parameter for the "ParamGovernor" shall be a "DefinedObjectClass".

A "DummyGovernor" shall be a "DummyReference" that has no "Governor".

8.4 The scope of a "DummyReference" appearing in a "ParameterList" is the "ParameterList" itself, together with that part of the "ParameterizedAssignment" which follows the "ParameterList". The "DummyReference" hides any other "Reference" with the same name in that scope in any given instantiation.

NOTE – This subclause does not apply to "identifier"s defined in "NamedNumberList"s, "Enumeration"s and "NamedBitList"s, since they are not "Reference"s. The "DummyReference" does not hide these "identifier"s (see Rec. ITU-T X.680 | ISO/IEC 8824-1, 19.12 and 20.11).

8.5 The usage of a "DummyReference" within its scope shall be consistent with its syntactic form, and, where applicable, governor, and all usages of the same "DummyReference" shall be consistent with one another.

NOTE – Where the syntactic form of a dummy reference name is ambiguous (for example, between whether it is an "objectclassreference" or "typereference"), the ambiguity can normally be resolved on the first use of the dummy reference name on the right-hand side of the assignment statement. Thereafter, the nature of the dummy reference name is known. The nature of the dummy reference is, however, not determined solely by the right-hand side of the assignment statement when it is in turn used only as an actual parameter in a parameterized reference; in this case, the nature of the dummy reference must be determined by examining the definition of this parameterized reference. Users of the notation are warned that such a practice can make ASN.1 specifications less clear, and it is suggested that adequate comments are provided to explain this for human readers.

Example

Consider the following parameterized object class assignment:

```
PARAMETERIZED-OBJECT-CLASS { TypeParam, INTEGER:valueParam, INTEGER:ValueSetParam } ::=
  CLASS {
    &valueField1      TypeParam,
    &valueField2      INTEGER DEFAULT valueParam,
    &valueField3      INTEGER (ValueSetParam),
    &ValueSetField    INTEGER DEFAULT { ValueSetParam }
  }
```

For the purpose of determining proper usage of the "DummyReference"s in the scope of the "Parameterized Assignment", and for that purpose only, the "DummyReference"s can be regarded to be defined as follows:

TypeParam ::= UnspecifiedType

valueParam INTEGER ::= unspecifiedIntegerValue

ValueSetParam INTEGER ::= { UnspecifiedIntegerValueSet }

where:

- a) **TypeParam** is a "DummyReference" which stands for a "Type". Therefore **TypeParam** can be used wherever a "typereference" can be used, e.g. as a "Type" for the fixed-type value field **valueField1**.
- b) **valueParam** is a "DummyReference" which stands for a value of an integer type. Therefore **valueParam** can be used wherever a "valuereference" of an integer value can be used, e.g. as a default value for the fixed-type value field **valueField2**.
- c) **ValueSetParam** is a "DummyReference" which stands for a value set of an integer type. Therefore **ValueSetParam** can be used wherever a "typereference" of an integer value can be used, e.g. as a "Type" in the "ContainedSubtype" notation for **valueField3** and **ValueSetField**.

8.6 Each "DummyReference" shall be employed at least once within its scope.

NOTE – If the "DummyReference" did not so appear, then the corresponding "ActualParameter" would have no effect on the definition, and would simply be "discarded", while to the user it might seem that some specification was taking place.

"ParameterizedValueAssignment"s, "ParameterizedValueSetTypeAssignment"s, "ParameterizedObjectAssignment"s and "ParameterizedObjectSetAssignment"s that contain either a direct or indirect reference to themselves are illegal.

8.7 Given a set of parameterized assignments (one or more) and a recursive path of parameterized references crossing one or more of those parameterized assignments, all the parameterized references along that path shall satisfy the following condition: Each actual parameter present in the parameterized reference shall either consist solely of a dummy reference (with no lexical items or comments preceding or following the dummy reference) or shall not contain any dummy reference (see A.3).

8.8 In the definition of a "ParameterizedType", "ParameterizedValueSet", or "ParameterizedObjectClass", a circular reference to the item being defined shall not be made, unless such reference is directly or indirectly marked **OPTIONAL** or, in the case of "ParameterizedType" and "ParameterizedValueSet", made through a reference to a choice type, at least one of whose alternatives is non-circular in definition.

8.9 The governor of a "DummyReference" shall not include a reference to another "DummyReference" if that other "DummyReference" also has a governor.

8.10 In a parameterized assignment the right side of the " ::= " shall not consist solely of a "DummyReference".

8.11 The governor of a "DummyReference" shall not require knowledge of either the "DummyReference" or of the parameterized reference name being defined.

8.12 When a value or value set is supplied to a parameterized type as an actual parameter, the type of the actual parameter is required to be compatible with the governor of the corresponding dummy parameter. (See Rec. ITU-T X.680 | ISO/IEC 8824-1, C.6.2 and C.6.3 for details.)

8.13 In defining a parameterized type with a value or a value set dummy parameter, the type used to govern that dummy parameter shall be a type, all of whose values are valid for use in all places to the right of the assignment where the dummy parameter is used. (See Rec. ITU-T X.680 | ISO/IEC 8824-1, C.6.5 for details.)