DRAFT International Standard

ISO/IEC DIS 11404

Information technology — General-Purpose Datatypes (GPD)

Technologies de l'information — Types de données à but général (GPD)

ICS: 35.060

ISO/IEC JTC **1**/SC **32**

Secretariat: **ANSI**

Voting begins on:
2024-08-16

Voting terminates on:
2024-11-08

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS 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 11404:2024(en)

© ISO/IEC 2024



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

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

Page

Foreword vi
0 Introduction vii
1 Scope 1
2 Normative references 2
3 Terms and definitions 2
4 Conformance 8
4.1 Direct conformance 8
4.2 Indirect conformance 9
4.3 Conformance of a mapping standard 9
4.4 GPD program conformance 10
5 Conventions used in this International Standard 10
5.1 Formal syntax 10
5.2 Text conventions 11
6 Fundamental notions 11
6.1 Datatype 11
6.2 Value space 12
6.3 Datatype properties 12
6.3.1 Equality 13
6.3.2 Order 13
6.3.3 Bound 13
6.3.4 Cardinality 14
6.3.5 Exact and approximate 14
6.3.6 Numeric 15
6.4 Primitive and non-primitive datatypes 15
6.5 Datatype generator 15
6.6 Characterizing operations 16
6.7 Datatype families 17
6.8 Aggregate datatypes 17
6.8.1 Homogeneity 17
6.8.2 Size 18
6.8.3 Uniqueness 18
6.8.4 Aggregate-imposed identifier uniqueness 18
6.8.5 Aggregate-imposed ordering 18
6.8.6 Access method 18
6.8.7 Recursive structure 19
6.8.8 Structured and unstructured 19
6.8.9 Mandatory and optional components 19
6.9 Provisions associated with datatypes 20
7 Elements of the Datatype Specification Language 21
7.1 IDN character-set 22
7.2 Whitespace 23
7.3 Lexical objects 23
7.3.1 Identifiers 23
7.3.2 Digit-string 24
7.3.3 Character-literal and string-literal 24
7.3.4 Keywords 24
7.4 Annotations 25
7.5 Values 25

© ISO/IEC 2024 – All rights reserved

7.5.1 Independent values 25
7.5.2 Dependent values 26
7.6 GPD program text 27

8 Datatypes 28
8.1 Primitive datatypes 28
8.1.1 Boolean 30
8.1.2 State 30
8.1.3 Enumerated 31
8.1.4 Character 32
8.1.5 Ordinal 34
8.1.6 Date-and-Time 34
8.1.7 Integer 36
8.1.8 Rational 36
8.1.9 Scaled 37
8.1.10 Real 39
8.1.11 Float 41
8.1.12 Complex 43
8.1.13 Void 44
8.2 Subtypes and extended types 45
8.2.1 Range 45
8.2.2 Selecting 46
8.2.3 Excluding 46
8.2.4 Size 47
8.2.5 Explicit subtypes 48
8.2.6 Extended 48
8.3 Generated datatypes 49
8.3.1 Choice 50
8.3.2 Pointer 52
8.3.3 Procedure 53
8.4 Aggregate Datatypes 57
8.4.1 Record 58
8.4.2 Class 60
8.4.3 Set 61
8.4.4 Bag 62
8.4.5 Sequence 64
8.4.6 Array 65
8.4.7 Table 68
8.5 Defined datatypes 69
8.6 Provisions 70
8.6.1 General parameters for provisions 70
8.6.2 Aggregate-specific features 74
8.6.3 Aggregate-component-identifier uniqueness 74
8.6.4 Usage-specific features 74

9 Declarations 75
9.1 Type declarations 76
9.1.1 Renaming declarations 76
9.1.2 New datatype declarations 77
9.1.3 New generator declarations 77
9.2 Value declarations 77
9.3 Termination declarations 77
9.4 Normative datatype declarations 78
9.5 Lexical operations 78
9.5.1 Import 78
9.5.2 Macro 79

10 Defined datatypes and generators 79
10.1 Defined datatypes 79
10.1.1 Natural number 80
10.1.2 Modulo 80

© ISO/IEC 2024 – All rights reserved

10.1.3 Bit 81
10.1.4 Bit string 81
10.1.5 Character string 81
10.1.6 Time interval 83
10.1.7 Octet 83
10.1.8 Octet string 83
10.1.9 Private 84
10.1.10 Object identifier 84
10.2 Defined generators 86
10.2.1 Stack 86
10.2.2 Tree 87
10.2.3 Optional 88

11 Mappings 89
11.1 Outward Mappings 89
11.2 Inward Mappings 90
11.3 Reverse Inward Mapping 91
11.4 Support of Datatypes 92
11.4.1 Support of equality 92
11.4.2 Support of order 92
11.4.3 Support of bounds 92
11.4.4 Support of cardinality 92
11.4.5 Support for the exact or approximate property 93
11.4.6 Support for the numeric property 93
11.4.7 Support for the mandatory components 93

Annex A (informative) Character-set standards 94

Annex B (informative) Recommendation for the placement of annotations 96
11.5 B.1 Type-attributes 96
11.6 B.2 Component-attributes 96
11.7 B.3 Procedure-attributes 97
11.8 B.4 Argument-attributes 97

12 Annex C: (informative) Implementation notions of datatypes 98
12.1 C.1 StorageSize 98
12.2 C.2 Mode 98
12.3 C.3 Floating-Point 99
12.4 C.4 Fixed-Point 99
12.5 C.5 Tag 100
12.6 C.6 Discriminant 100
12.7 C.7 StorageSequence 100
12.8 C.8 Packed 100
12.9 C.9 Alignment 100
12.10 C.10 Form 100

© ISO/IEC 2024 – All rights reserved

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 the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member 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 shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 11404 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

This third edition cancels and replaces the second edition (ISO/IEC 11404:2007), which has been technically revised.

© ISO/IEC 2024 – All rights reserved

0 Introduction

0.1 Introduction to the third edition

The "float" datatype has been added to match the computational environments available in modern systems, see 8.1.11. In addition, the float type of various precisions to can support optimized Machine Learning (ML) calculations, classifications, and inferences.

0.2 Introduction to the second edition

This second edition of ISO/IEC 11404 incorporates recent technologies and improvements since the first edition (ISO/IEC 11404:1996). The following improvements have been incorporated into the second edition.

— *Title change to reflect actual usage*. The use of ISO/IEC 11404 is no longer simply a tool for communicating among programming languages (old title: *Language-independent datatypes*). ISO/IEC 11404 is used for formal description of conceptual datatypes in binding (or binding-independent) standards and used as formalization of metadata for data elements, data element concepts, and value domains (see ISO/IEC 11179-3). The old title was potentially misleading because readers might believe that ISO/IEC 11404 is only useful for programming languages. The new title, *General-Purpose Datatypes* captures the essence of ISO/IEC 11404 and its use.

— *Incorporation of latest technologies*. Provide enhancements to the use of ISO/IEC 11404 as a datatype nomenclature reference for current programming languages, interface languages and data representation languages, specifically Java, IDL, Express, and XML.

— *Support for semi-structured and unstructured data aggregates*. Semi-structured data and unstructured data includes aggregates where datatyping and navigation may be unknown or unspecified in advance. For example, some systems permit "discovery" (or "introspection") of data. In some cases, the datatype may be unknown in advance (e.g. at compilation time), but may be discovered and processed at runtime (e.g. via datatype libraries or metadata registries).

— *Support for data longevity, versioning, and migration*. There is a need to support, from a datatyping perspective, obsolete and reserved features, such as data elements and permissible values (enumerations and states). Marking features as "obsolete" allows processing, compilation, and runtime systems to "flag" or diagnose old (deprecated) features, while still maintaining compatibility, so that it is possible to support transitions from past to present. Similarly, marking features as "reserved" allows processing, compilation, and runtime systems to "flag" or diagnose potential incompatibilities with future systems, so that it is possible to support transitions from present to future.

— *Extensibility of datatypes and value spaces*. There is a need to support some kind of extensibility concept. For example: (1) a GPD specification of an aggregate contains the elements A and B. (2) An application creates an aggregate with the elements A, B, and C. (3) Are the application's "extensions" of the aggregate acceptable/in conformity with the GPD specification in (1)? The answer to (3) is dependent upon the intent and design of the specification in (1): in some cases extensions are permitted, in some cases extensions are not permitted. The extensibility concept would allow the user of GPD datatypes to describe the kind of extensions permitted. This feature is particularly important in (a) data conformance and (b) application runtime environments that permit "discovery" or "introspection". This feature is available via the "provision()" capability.

Features that are not incorporated within GPD include the following:

— *Namespace capability*. Given the larger number of declarations, a namespace capability is necessary.

© ISO/IEC 2024 – All rights reserved

— *Data representation*. Although these features are a part of GPD annotations, there is no standardization of data representation in these annotations. This step is an important link for data interoperability.

0.3 Introduction to the first edition (ISO/IEC 11404:1996)

Many specifications of software services and applications libraries are, or are in the process of becoming, International Standards. The interfaces to these libraries are often described by defining the form of reference, e.g. the "procedure call", to each of the separate functions or services in the library, as it must appear in a user program written in some standard programming language (Fortran, COBOL, Pascal, etc.). Such an interface specification is commonly referred to as the "<language> binding of <service>", e.g. the "Fortran binding of PHIGS" (ISO/IEC 9593-1:1990, *Information processing systems — Computer graphics — Programmer's Hierarchical Interactive Graphics System (PHIGS) language bindings — Part 1: FORTRAN*).

This approach leads directly to a situation in which the standardization of a new service library immediately requires the standardization of the interface bindings to every standard programming language whose users might reasonably be expected to use the service, and the standardization of a new programming language immediately requires the standardization of the interface binding to every standard service package which users of that language might reasonably be expected to use. To avoid this n-to-m binding problem, ISO/IEC JTC 1, *Information technology* assigned to SC 22 the task of developing an International Standard for language-independent procedure calling and a parallel International Standard for language-independent datatypes, which could be used to describe the parameters to such procedures.

This International Standard provides the specification for the language-independent datatypes. It defines a set of datatypes, independent of any particular programming language specification or implementation, that is rich enough so that any common datatype in a standard programming language or service package can be mapped to some datatype in the set.

The purpose of this International Standard is to facilitate commonality and interchange of datatype notions, at the conceptual level, among different languages and language-related entities. Each datatype specified in this International Standard has a certain basic set of properties sufficient to set it apart from the others and to facilitate identification of the corresponding (or nearest corresponding) datatype to be found in other standards. Hence, this International Standard provides a single common reference model for all standards which use the concept datatype. It is expected that each programming language standard will define a mapping from the datatypes supported by that programming language into the datatypes specified herein, semantically identifying its datatypes with datatypes of the reference model, and thereby with corresponding datatypes in other programming languages.

It is further expected that each programming language standard will define a mapping from those language-independent (LI) datatypes which that language can reasonably support into datatypes which may be specified in the programming language. At the same time, this International Standard will be used, among other applications, to define a "language-independent binding" of the parameters to the procedure calls constituting the principal elements of the standard interface to each of the standard services. The production of such service bindings and language mappings leads, in cooperation with the parallel language-independent procedure calling mechanism, to a situation in which no further "<language> binding of <service>" documents need to be produced: Each service interface, by defining its parameters using LI datatypes, effectively defines the binding of such parameters to any standard programming language; and each language, by its mapping from the LI datatypes into the language datatypes, effectively defines the binding to that language of parameters to any of the standard services.

© ISO/IEC 2024 – All rights reserved

Information technology — General-Purpose Datatypes (GPD)

1 Scope

This International Standard specifies the nomenclature and shared semantics for a collection of datatypes commonly occurring in programming languages and software interfaces, referred to as the General-Purpose Datatypes (GPD). It specifies both primitive datatypes, in the sense of being defined *ab initio* without reference to other datatypes, and non-primitive datatypes, in the sense of being wholly or partly defined in terms of other datatypes. The specification of datatypes in this International Standard is "general-purpose" in the sense that the datatypes specified are classes of datatype of which the actual datatypes used in programming languages and other entities requiring the concept "datatype" are particular instances. These datatypes are general in nature; thus, they serve a wide variety of information processing applications.

This International Standard expressly distinguishes three notions of datatype:

— the conceptual, or abstract, notion of a datatype, which characterizes the datatype by its nominal values and properties;

— the structural notion of a datatype, which characterizes the datatype as a conceptual organization of specific component datatypes with specific functionalities; and

— the implementation notion of a datatype, which characterizes the datatype by defining the rules for representation of the datatype in a given environment.

This International Standard defines the abstract notions of many commonly used primitive and non-primitive datatypes which possess the structural notion of atomicity. This International Standard does not define all atomic datatypes; it defines only those which are common in programming languages and software interfaces. This International Standard defines structural notions for the specification of other non-primitive datatypes, and provides a means by which datatypes not defined herein can be defined structurally in terms of the GPDs defined herein.

This International Standard defines a partial terminology for implementation notions of datatypes and provides for the use of this terminology in the definition of datatypes. The primary purpose of this terminology is to identify common implementation notions associated with datatypes and to distinguish them from conceptual notions.

This International Standard specifies the required elements of mappings between the GPDs and the datatypes of some other language. This International Standard does not specify the precise form of a mapping, but rather the required information content of a mapping.

© ISO/IEC 2024 – All rights reserved

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 8824 (all parts), *Information technology — Abstract Syntax Notation One (ASN.1)*

ISO/IEC 10646, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 13886, *Information technology — Language Independent Procedure Calling (LIPC)*

ISO/IEC 14977, *Information technology — Syntactic metalanguage — Extended BNF*

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

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE These definitions might not coincide with accepted mathematical or programming language definitions of the same terms.

3.1
actual parametric datatype
datatype appearing as a parametric datatype in a use of a datatype generator, in contrast to the formal-parametric-types appearing in the definition of the datatype generator

3.2
actual parametric value
value appearing as a parametric value in a reference to a datatype family or datatype generator, in contrast to the formal-parametric-values appearing in the corresponding definitions

3.3
aggregate datatype
generated datatype each of whose values is made up of values of the component datatypes, in the sense that operations on all component values are meaningful

3.4
annotation
descriptive information unit attached to a datatype, or a component of a datatype, or a procedure (value), to characterize some aspect of the representations, variables, or operations associated with values of the datatype

3.5
approximate
property of a datatype indicating that there is not a 1-to-1 relationship between values of the conceptual datatype and the values of a valid computational model of the datatype

3.6
bounded
property of a datatype, meaning both bounded above and bounded below

© ISO/IEC 2024 – All rights reserved

3.7
bounded above
property of a datatype indicating that there is a value U in the value space such that, for all values s in the value space, $s \leq U$

3.8
bounded below
property of a datatype indicating that there is a value L in the value space such that, for all values s in the value space, $s \geq L$

3.9
characterizing operations
⟨datatype⟩[1)] collection of operations on, or yielding, values of the datatype that distinguish this datatype from other datatypes with identical value spaces

3.10
characterizing operations
⟨datatype generator⟩ collection of operations on, or yielding, values of any datatype resulting from an application of the datatype generator that distinguish this datatype generator from other datatype generators and produce identical value spaces from identical parametric datatypes

3.11
component datatype
datatype which is a parametric datatype to a datatype generator

NOTE A component datatype is a datatype on which the datatype generator operates.

3.12
datatype
set of distinct values, characterized by properties of those values, and by operations on those values

3.13
datatype declaration
means provided by this International Standard for the definition of a datatype which is not itself defined by this International Standard

3.14
datatype family
collection of datatypes which have equivalent characterizing operations and relationships, but value spaces that differ in the number and identification of the individual values

3.15
datatype generator
generator
operation on datatypes, as objects distinct from their values, that generates new datatypes

3.16
defined datatype
datatype defined by a type-declaration.

3.17
defined generator
datatype generator defined by a type-declaration

1) Angle brackets indicate the subject field to which the concept belongs, in accordance with ISO 10241.

© ISO/IEC 2024 – All rights reserved

3.18
exact
property of a datatype indicating that every value of the conceptual datatype is distinct from all others in any valid computational model of the datatype

3.19
formal-parametric-type
identifier, appearing in the definition of a datatype generator, for which a datatype will be substituted in any reference to a (defined) datatype resulting from the generator

3.20
formal-parametric-value
identifier, appearing in the definition of a datatype family or datatype generator, for which a value will be substituted in any reference to a (defined) datatype in the family or resulting from the generator

3.21
general-purpose datatype
GPD
datatype defined by this International Standard

3.22
GPD-generated datatype
GPD datatype
datatype defined by the means of datatype definition provided by this International Standard

NOTE Although "GPD datatype" expands to "general-purpose datatype datatype" and might appear redundant, it is to be read as "general-purpose-datatype datatype", where GPD is an adjective and datatype (standalone) is a noun.

3.23
generated datatype
datatype defined by the application of a datatype generator to one or more previously-defined datatypes

3.24
generated internal datatype
datatype defined by the application of a datatype generator defined in a particular programming language to one or more previously-defined internal datatypes

3.25
generator declaration
means provided by this International Standard for the definition of a datatype generator which is not itself defined by this International Standard

3.26
instruction
provision that conveys an action to be performed

[ISO/IEC Guide 2]

3.27
internal datatype
datatype whose syntax and semantics are defined by some other standard, specification, language, product, service or other information processing entity

3.28
inward mapping
conceptual association between the internal datatypes of a language and the general-purpose datatypes which assigns to each general-purpose datatype either a single semantically equivalent internal datatype or no equivalent internal datatype

© ISO/IEC 2024 – All rights reserved

3.29
lower bound
value L such that, for all values s in the value space in a datatype which is bounded below, $L \leq s$

3.30
mandatory requirement
requirement of a normative document that must necessarily be fulfilled in order to comply with that document [adapted from ISO/IEC Guide 2]

NOTE A "mandatory requirement" is also known as an "exclusive requirement".

3.31
mapping
⟨datatypes⟩ formal specification of the relationship between the internal datatypes that are notions of, and specifiable in, a particular programming language and the general-purpose datatypes specified in this International Standard

3.32
mapping
⟨values⟩ corresponding specification of the relationships between values of the internal datatypes and values of the general-purpose datatypes

3.33
meta-identifier
name of a non-terminal symbol

[ISO/IEC 14977]

NOTE See note in 5.1 concerning the context of the specialized usage of this term for describing the syntax of ISO/IEC 11404 program text.

3.34
non-terminal symbol
⟨EBNF⟩ syntactic part of the language being defined

[ISO/IEC 14977]

NOTE See note in 5.1 concerning the context of the specialized usage of this term.

3.35
normative datatype
collection of specifications for datatype properties that may be simultaneously satisfied by more than one actual datatype

3.36
normative document
document that provides rules, guidelines or characteristics for activities or their results

[ISO/IEC Guide 2]

NOTE 1 The term "normative document" is a generic term that covers such documents as standards and technical specifications.

NOTE 2 A "document" is to be understood as any medium with information recorded on or in it, such as a paper document or program code.

3.37
optional requirement
requirement of a normative document that must be fulfilled in order to comply with a particular option permitted by that document

© ISO/IEC 2024 – All rights reserved

[ISO/IEC Guide 2]

NOTE An optional requirement may be either (1) one of two or more alternative requirements; or (2) an additional requirement that must be fulfilled only if applicable and that may otherwise be disregarded.

3.38
order
mathematical relationship among values

NOTE See 6.3.2.

3.39
ordered
property of a datatype which is determined by the existence and specification of an order relationship on its value space

3.40
outward mapping
conceptual association between the internal datatypes of a language and the general-purpose datatypes that identifies each internal datatype with a single semantically equivalent general-purpose datatype

3.41
parametric datatype
datatype on which a datatype generator operates to produce a generated datatype

3.42
parametric value (1)
value which distinguishes one member of a datatype family from another

3.43
parametric value (2)
value which is a parameter of a datatype or datatype generator defined by a type-declaration

NOTE See 9.1.

3.44
primitive datatype
identifiable datatype that cannot be decomposed into other identifiable datatypes without loss of all semantics associated with the datatype

3.45
primitive internal datatype
datatype in a particular programming language whose values, conceptually, are not constructed in any way from values of other datatypes in the language

3.46
provision
expression of normative wording that takes the form of a statement, an instruction, a recommendation or a requirement

NOTE 1 Adapted from ISO/IEC Guide 2.

NOTE 2 These types of provision are distinguished by the form of wording they employ; e.g. instructions are expressed in the imperative mood, recommendations by the use of the auxiliary "should" and requirements by the use of the auxiliary "shall".

3.47
recommendation
provision that conveys advice or guidance

© ISO/IEC 2024 – All rights reserved

[ISO/IEC Guide 2]

3.48
regular value
element of a value space that is consistent with a datatype's properties and characterizing operations

3.49
representation
⟨general-purpose datatype⟩ mapping from the value space of the general-purpose datatype to the value space of some internal datatype of a computer system, file system or communications environment

3.50
representation
⟨value⟩ sign(s) of that value in the representation of the datatype

NOTE In this context, the term "sign" is used in its terminological sense (i.e. a symbol) and not in its mathematical sense (i.e. positive or negative).

3.51
requirement
provision that conveys criteria to be fulfilled

[ISO/IEC Guide 2]

3.52
sentence
⟨EBNF⟩ sequence of symbols that represents the start symbol

[ISO/IEC 14977]

NOTE See note in 5.1 concerning the context of the specialized usage of this term.

3.53
sentinel value
element of a value space that is not completely consistent with a datatype's properties and characterizing operations

NOTE A numeric datatype, which includes characterizing operations such as **Equal** and **InOrder**, may include sentinel values such as **not-a-number**, **indeterminate**, **not-applicable**, **+infinity**, **-infinity** and so on. These characterizing operations are not defined for sentinel values.

3.54
sequence
⟨EBNF⟩ ordered list of zero or more items

[ISO/IEC 14977]

NOTE See note in 5.1 concerning the context of the specialized usage of this term.

3.55
start symbol
⟨EBNF⟩ non-terminal symbol that is defined by one or more syntax rules but does not occur in any other syntax rule

[ISO/IEC 14977]

NOTE See note in 5.1 concerning the context of the specialized usage of this term.

© ISO/IEC 2024 – All rights reserved

3.56
statement
provision that conveys information

[ISO/IEC Guide 2]

3.57
subsequence
⟨EBNF⟩ sequence within a sequence

[ISO/IEC 14977]

NOTE See note in 5.1 concerning the context of the specialized usage of this term.

3.58
subtype
datatype derived from another datatype by restricting the value space to a subset whilst maintaining all characterizing operations

3.59
terminal symbol
⟨EBNF⟩ sequence of one or more characters forming an irreducible element of a language

[ISO/IEC 14977]

NOTE See note in 5.1 on the context of the specialized usage of this term.

3.60
upper bound
value U such that, for all values s in the value space in a datatype which is bounded above, $s \leq U$

3.61
value space
set of values for a given datatype

3.62
variable
computational object to which a value of a particular datatype is associated at any given time; and to which different values of the same datatype may be associated at different times

4 Conformance

An information processing product, system, element or other entity may conform to this International Standard either directly, by utilizing datatypes specified in this International Standard in a conforming manner (4.1), or indirectly, by means of mappings between internal datatypes used by the entity and the datatypes specified in this International Standard (4.2).

NOTE The general term *information processing entity* is used in this clause to include anything which processes information and contains the concept of *datatype*. Information processing entities for which conformance to this International Standard may be appropriate include other standards (e.g. standards for programming languages or language-related facilities), specifications, data handling facilities and services, etc.

4.1 Direct conformance

An information processing entity which *conforms directly* to this International Standard shall:

© ISO/IEC 2024 – All rights reserved

1. specify which of the datatypes and datatype generators specified in Clause 8 and Clause 10 are provided by the entity and which are not, and which, if any, of the declaration mechanisms in Clause 9 it provides; and

2. define the value spaces of the general-purpose datatypes used by the entity to be identical to the value spaces specified by this International Standard; and

3. use the notation prescribed by Clause 7 through Clause 10 of this International Standard to refer to those datatypes and to no others; and

4. to the extent that the entity provides operations other than movement or transformation of values, define operations on the general-purpose datatypes which can be derived from, or are otherwise consistent with, the characterizing operations specified by this International Standard.

NOTE 1 This International Standard defines a syntax for the denotation of values of each datatype it defines, but, in general, requirement 3 does not require conformance to that syntax. Conformance to the value-syntax for a datatype is required only in those cases in which the value appears in a type-specifier, that is, only where the value is part of the identification of a datatype.

NOTE 2 The requirements above prohibit the use of a type-specifier defined in this International Standard to designate any other datatype. They make no other limitation on the definition of additional datatypes in a conforming entity, although it is recommended that either the form in Clause 8 or the form in Clause 10 be used.

NOTE 3 Requirement 4 does not require all characterizing operations to be supported and permits additional operations to be provided. The intention is to permit addition of semantic interpretation to the general-purpose datatypes and generators, as long as it does not conflict with the interpretations given in this International Standard. A conflict arises only when a given characterizing operation could not be implemented or would not be meaningful, given the entity-provided operations on the datatype.

NOTE 4 Examples of entities which could conform directly are language definitions or interface specifications whose datatypes, and the notation for them, are those defined herein. In addition, the verbatim support by a software tool or application package of the datatype syntax and definition facilities herein should not be precluded.

4.2 Indirect conformance

An information processing entity which conforms indirectly to this International Standard shall:

1. provide mappings between its internal datatypes and the general-purpose datatypes conforming to the specifications of Clause 11 of this International Standard; and

2. specify for which of the datatypes in Clause 8 and Clause 10 an inward mapping is provided, for which an outward mapping is provided, and for which no mapping is provided.

NOTE 1 Standards for existing programming languages are expected to provide for indirect conformance rather than direct conformance.

NOTE 2 Examples of entities which could conform indirectly are language definitions and implementations, information exchange specifications and tools, software engineering tools and interface specifications, and many other entities which have a concept of datatype and an existing notation for it.

4.3 Conformance of a mapping standard

In order to conform to this International Standard, a standard for a mapping shall include in its conformance requirements the requirement to conform to this International Standard.

NOTE 1 It is envisaged that this International Standard will be accompanied by other standards specifying mappings between the internal datatypes specified in language and language-related standards and the general-purpose datatypes. Such mapping standards are required to comply with this International Standard.

NOTE 2 Such mapping standards may define "generic" mappings, in the sense that for a given internal datatype the standard specifies a parameterized general-purpose datatype in which the parametric values are not derived from

© ISO/IEC 2024 – All rights reserved

parametric values of the internal datatype nor specified by the standard itself, but rather are required to be specified by a "user" or "implementor" of the mapping standard. That is, instead of specifying a particular general-purpose datatype, the mapping specifies a family of general-purpose datatypes and requires a further user or implementor to specify which member of the family applies to a particular use of the mapping standard. This is always necessary when the internal datatypes themselves are, in the intention of the language standard, either explicitly or implicitly parameterized. For example, a programming language standard may define a datatype INTEGER with the provision that a conforming processor will implement some range of Integer; hence the mapping standard may map the internal datatype INTEGER to the general-purpose datatype:

```
integer range (min..max)
```

and require a conforming processor to provide values for "min" and "max".

4.4 GPD program conformance

A GDP conforming program[2] is a specification that uses datatypes and datatype values and their syntaxes as specified in this International Standard. Such a specification may be self-contained: there is no requirement for the existence of a GDP conformant implementation that can produce or operate on the specification. The requirements of this International Standard to which the specification conforms shall be clearly identified, either in the specification itself, or in documentation that is unambiguously identified in the specification.

NOTE A GDP conforming program is a special case of directly conforming entity.

5 Conventions used in this International Standard

5.1 Formal syntax

This International Standard defines a formal datatype specification language. The notation defined in ISO/IEC 14977, Extended Backus-Naur Form (EBNF), is used in defining that language. Table 5-1 summarizes the ISO/IEC 14977 EBNF syntactic metalanguage.

NOTE The terms meta-identifier, non-terminal symbol, sentence, sequence, start symbol, subsequence, and terminal symbol have special meaning in the context of EBNF notation (see Clause 3, Definitions).

Table 5-1 — Summary of ISO/IEC 14977 EBNF Syntactic Metalanguage Notation

Representation	ISO/IEC 10646 Character Names	Metalanguage Symbol
' '	apostrophe	first quote symbol
" "	quotation mark	second quote symbol
(* *)	left parenthesis with asterisk, asterisk with right parenthesis	start/end comment symbols
( )	left parenthesis, right parenthesis	start/end group symbols
[ ]	left square bracket, right square bracket	start/end option symbols
{ }	left curly bracket, right curly bracket	start/end repeat symbols
? ?	question mark	special sequence symbol
-	hyphen-minus	except symbol
,	comma	concatenate symbol
=	equals sign	defining symbol
|	vertical line	definition separator symbol
*	asterisk	repetition symbol
;	semicolon	terminator symbol

[2] A GPD conforming program might be an 11404 GPD datatype definition or a data declaration based upon an 11404 GPD datatype.

© ISO/IEC 2024 – All rights reserved

EXAMPLE 1 The following syntax rules illustrate repetition (asterisk and curly brackets) and option square brackets:

```
aa = "A" ;
bb = 3 * aa, "B" ;
cc = 3 * [aa], "C" ;
dd = {aa}, "D" ;
ee = aa, {aa}, "E" ;
ff = 3 * aa, 3 * [aa], "F" ;
```

Terminal strings defined by these rules are as follows:

```
aa: A
bb: AAAB
cc: C AC AAC AAAC
dd: D AD AAD AAAD AAAAD  etc.
ee: AE AAE AAAE AAAAE AAAAAE  etc.
ff: AAAF AAAAF AAAAAF AAAAAAF
```

EXAMPLE 2 The following syntax rules illustrate a definitions list (vertical line), an exception (hyphen-minus), and comments (parentheses and asterisks):

```
letter = "A" | "B" | "C" | "D" | "E" | "F"
| "G" | "H" | "I" | "J" | "K" | "L" | "M"
| "N" | "O" | "P" | "Q" | "R" | "S" | "T"
| "U" | "V" | "W" | "X" | "Y" | "Z" ;
vowel = "A" | "E" | "I" | "O" | "U" ;
consonant = letter - vowel ;        (* These examples are from ISO/IEC 14977 *)
```

Terminal strings defined by these rules are as follows:

```
letter:    A B C D E F G H I J  etc.
vowel:     A E I O U
consonant: B C D F G H J K L M  etc.
```

5.2 Text conventions

Within the text:

— A reference to a terminal symbol syntactic object consists of the terminal symbol in fixed width courier, e.g. `type`.

— A reference to a non-terminal symbol syntactic object consists of the non-terminal-symbol in fixed width italic courier, e.g. *`type-declaration`*.

— Mathematical notation, properties, and characterizing operations are in bold, e.g., **InOrder(x,y)**.

— Non-italicized words which are identical or nearly identical in spelling to a non-terminal-symbol refer to the conceptual object represented by the syntactic object. In particular, *`xxx-type`* refers to the syntactic representation of an "xxx datatype" in all occurrences.

6 Fundamental notions

6.1 Datatype

A datatype is a set of distinct values, characterized by properties of those values and by operations on those values. Characterizing operations are included in this International Standard solely in order to identify the datatype. In this International Standard, characterizing operations are purely informative and have no normative impact.

© ISO/IEC 2024 – All rights reserved