Committee Draft ISO/IEC CD	
Date: **2005-12-28**	Reference number: ISO/JTC 1/SC 32**N1390**
Supersedes document SC 32N1163	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/IEC JTC 1/SC 32 Data Management and Interchange Secretariat: USA (ANSI)	Circulated to P- and O-members, and to technical committees and organizations in liaison for voting (P-members only) by: **2006-03-27** Please return all votes and comments in electronic form directly to the SC 32 Secretariat by the due date indicated.

ISO/IEC CD 20944-03:200x(E)

Title: Information technology - Metadata Registry Interoperability & Bindings (MDR-IB) Part 3: Common Provisions for Conformance

Project: 1.32.17.01.03.00

Introductory note: The attached document is hereby submitted for a three-month letter ballot to the National Bodies of ISO/IEC JTC 1/SC 32. The ballot starts 2005-12-28.

Medium: E

No. of pages: 19

Address Reply to: SC 32 Secretary, ISO/IEC JTC 1/SC 32, Farance Inc, Island Box 256, New York, NY 10044-0205, United States of America

Telephone: +1 212 486-4700; E-mail: SC32-Sec@JTC1SC32.org

Reference number of working document: **ISO/IEC JTC1 SC32 N1390**

Date: 2005-12-25

Reference number of document: **ISO/IEC CD2 20944-03**
[Release Sequence #8]

Committee identification: ISO/IEC JTC1 SC32 WG2

SC32 Secretariat: US

Information technology — Metadata Registries Interoperability and Bindings (MDRIB) — Part 03: Common provisions for conformance

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

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.

Document type: International standard
Document subtype: if applicable
Document stage: (30) Committee
Document language: E

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees 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 20944-03 was prepared by Technical Committee ISO/IEC JTC1, *Information Technology*, Subcommittee SC32, *Data Management and Interchange*.

ISO/IEC 20944 consists of the following parts, under the general title *Information technology — Metadata Registries Interoperability and Bindings (MDRIB)*:

— *Part 01: Framework*

— *Part 02: Common vocabulary*

— *Part 03: Common provisions for conformance*

— *Part 04: Generic usage*

— *Part 05: Common data structures and services*

— *Part 06: Semi-structured aggregation*

— *Part 20: Common provisions for coding bindings*

— *Part 21: XML coding binding*

— *Part 22: DIVP coding binding*

— *Part 23: ASN.1 coding binding*

— *Part 40: Common provisions for application programming interface (API) bindings*

— *Part 41: C API binding*

— *Part 42: C++ API binding*

— *Part 43: Java API binding*

— *Part 44: ECMAScript API binding*

— *Part 45: Perl API binding*

— *Part 46: LISP API binding*

— *Part 47: PHP API binding*

— *Part 60: Common provisions for protocol bindings*

— *Part 61: SQL/CLI protocol binding*

— *Part 62: WebDAV protocol binding*

— *Part 63: SOAP protocol binding*

— *Part 64: WSDL protocol binding*

— *Part 65: LDAP protocol binding*

— *Part 66: JMS protocol binding*

— *Part 80: Common provisions for profiles*

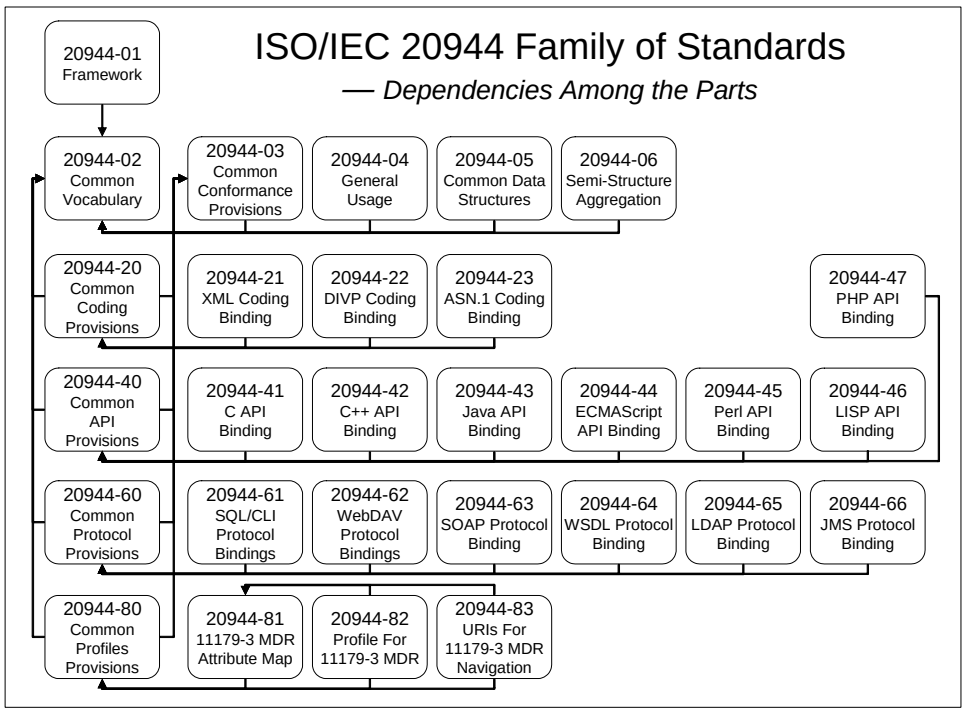
— *Part 81: Attribute mapping for 11179-3 MDR metamodel*

— *Part 82: Profile for 11179-3 MDR metamodel*

— *Part 83: URI suffixes for 11179-3 MDR metamodel navigation*

Introduction

The following diagram shows the organization of the ISO/IEC 20944 family of standards.



Organization of ISO/IEC 20944 family of standards.

This Part provides the common provisions for conformance that are referenced in other parts of this International Standard.

Information technology — Metadata Registries Interoperability and Bindings (MDRIB) — Part 03: Common provisions for conformance

Editor's Note: Each part of 20944 is marked with a common sequence number ("**[Release Sequence #N]**") to indicate they are synchronized and harmonized among themselves. The mark "**[Release Sequence #N]**" does *not* imply that there are a complete set of N-1 prior drafts for any particular Part.

1 Scope

The ISO/IEC 20944 family of standards describe codings, APIs, and protocols for interacting with an ISO/IEC 11179 metadata registry (MDR).

This Part of ISO/IEC 20944 provides provisions that may be used in common, harmonized conformance requirements. This Part addresses the following data interoperability features[1]:

- A common framework for variety control: harmonized concepts for conforming implementations and strictly conforming implementations,
- Harmonized provisions, such as mandatory requirements[2] and optional requirements[3], and their consistent application across all bindings of ISO/IEC 20944.
- Harmonized and consistent treatment of data elements with varying data obligation attributes (e.g., mandatory, conditional, optional, extended) and varying data longevity attributes (e.g., in-use, obsolete, reserved, etc.).

This Part also includes an annex that contains a Rationale that guided the development of this Part. The Rationale also discusses the harmonized use of profiles (e.g., subsets, supersets, changes, etc.) of the data structure and data elements.

---

[1] The concept of *data interoperability* applies to metadata when metadata is treated as data, e.g., metadata item attributes (as specified by ISO/IEC 11179-3) that are transferred or exchanged. The concept of data interoperability is different than *metadata interoperability*: agreement upon the meaning of descriptive data, which is outside the scope of this International Standard.

[2] In the context of this International Standard, the term *mandatory requirement* has the same meaning as it is defined in ISO/IEC Guide 2, subclause 7.5.1: *a requirement of a normative document that must necessarily be fulfilled in order to comply with that document*. There is *no implication* that the aforementioned requirement is compulsory by law or regulation. This kind of *mandatory requirement* is also known as an *exclusive requirement*.

[3] ISO/IEC Guide 2, subclause 7.5.2 defines the term *optional requirement*, which includes the following note: *An optional requirement may be either: a) one of two or more alternative requirements; or b) an additional requirement that must be fulfilled only if applicable and that may otherwise be disregarded*.

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 Guide 2:1996, *Standardization and related activities — General vocabulary*

ISO/IEC TR 10000-1, *Information technology — Framework and taxonomy of International Standardized Profiles — Part 1: General principles and documentation framework*

ISO/IEC 20944-02:—[4], *Information technology — Metadata Registries Interoperability and Bindings (MDRIB) — Common vocabulary*[5]

3 Terms and definitions

For the purposes of this document, the terms and definitions given in Part 02 and the following apply[6].

3.1
base normative document
normative document used for creating derived normative documents

NOTE The term *base data model*, defined in 20944-02, is used throughout the 20944 family of standards to reference the implied application's data model that is being used for the bindings. The *base data model* is tied to the bindings via normative reference, e.g., some other standard defines a data model and uses 20944, via normative reference, to provide some coding, API, or protocol bindings. For Part 82, the "base data model" is the 11179-3 metamodel. Part 04 of this International Standard, explains how other standards and specifications may use or re-use portions of the 20944 family of standards.

3.2
derived normative document
normative document that has one or more provisions in common with another normative document

3.3
base technical specification
normative document used for creating derived technical specifications

3.4
derived technical specification
technical specification that has one or more provisions in common with another normative document

---

[4] To be published.

[5] The international standards, technical reports, and drafts of the 11179, 20943, and 20944 series are available at

http://metadata-standards.org/11179
http://metadata-standards.org/20943
http://metadata-standards.org/20944

[6] Users and implementers of this International Standard may find it useful to reference additional terms and definitions from 20944-02.

3.5
derived standard
standard that has one or more provisions in common with another normative document

3.6
base data model
data model that may be used for derivation

EXAMPLE 1 A base data model, plus extensions, forms a new data model (which is derived from the base data model).

EXAMPLE 2 A profile of a base data model forms a new data model (which is derived from the base data model).

EXAMPLE 3 An API binding may be based upon a data model (i.e., the API is derived from the base data model).

NOTE The term *base data model*, defined in 20944-02, is used throughout the 20944 family of standards to reference the implied application's data model that is being used for the bindings. The *base data model* is tied to the bindings via normative reference, e.g., some other standard defines a data model and uses 20944, via normative reference, to provide some coding, API, or protocol bindings. For Part 82, the "base data model" is the 11179-3 metamodel. Part 04 of this International Standard, explains how other standards and specifications may use or re-use portions of the 20944 family of standards.

4 Conformance

In this International Standard, "shall" is to be interpreted as a requirement upon an implementation; "shall not" is to be interpreted as a prohibition. Undefined behavior is:

1. a "shall" requirement or "shall not" prohibition is violated
2. indicated in this International Standard by the words "undefined behavior"
3. indicated by the omission of any explicit definition of behavior.

There is no difference in emphasis among these three; they all describe "behavior that is undefined".

NOTE 1 Implementations claim conformance to particular features of this International Standard in their Implementation Conformance Statement (ICS).

NOTE 2 See the definition of the acronym SC/C in 20944-02, Clause 3.

4.1 Conformance level

The following subclauses define strictly conforming implementations and conforming implementations. In the context of conformance, the terms "support", "use", "test", "access", and "probe" are defined in each data conformance paradigm that incorporates this Part.

4.2 Profiles, derived standards, subset standards, superset standards, and extensions

Implementations shall indicate which Parts of this International Standard they claim conformity to in their ICS and in their conformance label(s).

NOTE Implementations may use automated techniques to convey their ICS for automated interoperability.

4.3 Strictly conforming implementations

A strictly conforming implementation:

— shall support all mandatory and optional data elements;
— shall not use, test, access, or probe for any extension features;[7]
— shall not exceed limits or smallest permitted maximum values specified by the controlling normative document; and
— shall not interpret or generate data elements that are dependent on any unspecified, undefined, implementation-defined, or locale-specific behavior.

NOTE The use of extensions is undefined behavior.

4.4 Conforming implementations

A conforming implementation shall be at least one of: a conforming coding, a conforming API, a conforming protocol, or a conforming data application.

A conforming implementation:

— shall support all mandatory and optional data elements;
— may use, test, access, or probe for extension features, as permitted by the implementation and data interchange participants, as long as the meaning and behavior of strictly conforming implementations remain unchanged;
— shall not support or use extension features that change the meaning or behavior of strictly conforming implementations;
— may exceed limits or smallest permitted maximum values specified by controlling normative document(s), and to the extent permitted by the implementation; and
— may interpret or generate data elements that are dependent on implementation-defined, locale-specific, or unspecified behavior.

NOTE 1 The use of extensions is undefined behavior.

NOTE 2 All strictly conforming implementations are also conforming implementations.

NOTE 3 An implementation does not conform to this Part if it redefines features via extension methods, and these features change the meaning or behavior of strictly conforming implementations.

4.5 Conformance labels

A conformance label may summarize ICSs. Annex B, Conformance Labels, describes the requirements for human-readable and machine-readable conformance labels.

5 Derived normative documents, profiles, extensions

5.1 Derived normative document (derived standard)

A derived normative document is a normative document that has provisions in common with a base normative document. In comparison to profiles, a derived standard makes no requirements concerned the relationship between conformance to the base normative document and conformance to the derived normative document (or vice versa). Thus an implementation that conforms to the derived normative document is not required to

[7] Extension features include extended data elements, extended value spaces, extended value domains, extended operations, etc..

conform to the base normative document (and vice versa). A derived normative document may contain extensions to the base normative document.

5.2 General principles of a profile

A *profile* is defined in ISO/IEC TR 10000-1 as:

> [a] set of one or more base standards and/or ISPs, and, where applicable, the identification of chosen classes, conforming subsets, options and parameters of those base standards[8], or ISPs necessary to accomplish a particular function

Profiles reference other normative documents. References may be dated or undated.

The general principles of a profile are specified in ISO/IEC TR 10000-1, subclause 6.3.1:

> **6.3.1 General Principles**
>
> A profile makes explicit the relationships within a set of base standards used together (relationships which can be implicit in the definitions of the base standards themselves), and may also specify particular details of each base standard being used. A profile may refer to other International Standardized Profiles[9] in order to make use of the functions and interfaces already defined by them, and thus limit its own direct reference to base standards. It follows that a profile
>
> a) shall restrict the choice of base standard options to the extent necessary to maximise the probability of achieving the objective of the profile; for example to facilitate interworking between IT systems, or porting an application between them, where they have implemented different selections of options of the profile. Thus a profile may retain base standard options as options of the profile provided that they do not affect interworking or portability.
>
> b) shall not specify any requirements that would contradict or cause non-conformance to the base standards to which it refers;
>
> c) may contain conformance requirements which are more specific and limited in scope than those of the base standards to which it refers. Whilst the capabilities and behaviour specified in a profile will always be valid in terms of the base standards, a profile may exclude some valid optional capabilities and optional behaviour permitted in those base standards.
>
> Thus conformance to a profile implies by definition conformance to the set of base standards which it references. However, conformance to that set of base standards does not necessarily imply conformance to the profile.

NOTE While the last paragraph above summarizes one important aspect of interoperability and compatibility (i.e., conformance to the profile implies conformance to the base standard), from the perspective of the developer of profile, a more important interoperability and compatibility issue is item b above: *[a profile] shall not specify any requirements that would contradict or cause non-conformance to the base standards to which it refers*. This requirement has a profound

---

[8] Generically, this International Standard refers to *normative documents* and *technical specifications*; a *standard* is a *normative document* that is a *technical specification* and agreed upon by *consensus*. Thus, the intention of using the terms *normative document* and *technical specification* is to decouple the *consensus process* that is associated with a *standard*.

[9] An International Standardized Profile (ISP) is a *profile* that is also an *international standard*.

effect upon profiles of data interchange standards because profiles inherit certain *implicit* requirements from base standards.

The main elements of a profile definition are specified in ISO/IEC TR 10000-1, subclause 6.3.2:

> **6.3.2 Main elements of a profile definition**
>
> The definition of a profile shall comprise the following elements:
>
> a) a concise definition of the scope of the function for which the profile is defined and the user requirements which it will satisfy, which is capable of being used as an Executive Summary of the profile;
>
> b) an illustration of the scenario within which the profile is applicable, giving, where possible, a diagrammatic representation of the IT systems, applications and interfaces which are relevant;
>
> c) normative reference to a single set of base standards or ISPs, including precise identification of the actual texts of the base standards or ISPs being used; also identification of any approved amendments and technical corrigenda (errata), conformance to which is identified as potentially having an impact on achieving interoperability or portability using the profile;
>
> d) specifications of the application of each referenced base standard or ISP, stating the choice of classes or conforming subsets, and the selection of options, ranges of parameter values, etc, and reference to registered objects;
>
> e) a statement defining the requirements to be observed by IT systems claiming conformance to the profile, including any remaining permitted options of the referenced base standards or ISPs, which thus become options of the profile;
>
> f) if relevant, a reference to the specification of conformance tests for the profile;
>
> g) informative reference to any amendments or technical corrigenda to the base standards referenced in the profile, which have been determined to be not applicable to the profile, and to any other relevant source documents

5.3 Extensions

5.3.1 Kinds of extensions

Extensions are additional provisions that affect the use and implementation of a base normative document (e.g., a standard). Additional provisions that *do not affect* the use and implementation of a base normative document is simply an unrelated normative document.

5.3.1.1 Definition of extension

An extension, in general, is a derived normative document that includes additional provisions with respect to a base normative document. Extensions, in general, may produce more functionality or less functionality. Based on the definition of "extension", there are no implications of any kind of interoperability or compatibility between implementations of the base normative document and implementations of the derived normative document.

NOTE It is a common misunderstanding that an "extension" only provides additional capability. The perception of "additional" is dependent upon the stakeholder's perspective, e.g., one stakeholder's additional capability may be another stakeholder's diminished capability.

5.3.1.2 Extension profiles

An extension profile is a normative document that is an extension and a profile both with respect to a base normative document. For an extension profile, the behavior and meaning of strictly conforming implementations remains unchanged.

NOTE It some contexts, the term *extension* is used to mean *extension profile*.

5.3.2 Uses of extensions

5.3.2.1 Extensions for value spaces

An extended value space for a datatype shall be compatible in characterizing operations and properties. However, values in a value space may be tied to value meanings of a value domain. The interoperability and compatibility of the newly derived value domain is indeterminate because the similarity or change in value meanings (for the extended value space) are specified external to the datatype definition.

5.3.2.2 Extensions for value domains

Isomorphic value domains are equivalent in meaning, but may have different values (e.g., codes).[10]

An overlapping value domain shares some permissible values with another value domain.

An extended value domain is a superset of the base value domain and its permissible values shall retain their meanings with respect to the base value domain.[11]

5.3.2.3 Extensions for data elements

With respect to a base normative document for data interchange, it is possible to create a subsets and supersets for normative documents, technical specifications, and standards (e.g., a "subset technical specification" or a "superset standard"). Both subsets and supersets are considered extensions (i.e., there are additional provisions). A subset data interchange specification implies a subset of conforming data instances; a superset data interchange specification implies a superset of conforming data instances.[12]

---

[10] The creation of isomorphic value domains may be useful for representing code element mappings.

[11] For value domains that represent a (mathematical) partition, i.e., a classification that is comprehensive and not overlapping, it is difficult to extend the value domain without changing the meaning (changing the meaning introduces interoperability and compatibility problems).

[12] Thus, the "subset" and "superset" notions make assertions about interoperability and compatibility between the base normative document and the derived normative document.

6 Annex A: Rationale (informative)

6.1 Conformance level

The distinction between "strictly conforming" and "conforming" implementations is necessary to address the simultaneous needs for interoperability and extensions. This International Standard describes specifications that promote interoperability. Extensions are motivated by needs of users, vendors, institutions, and industries (1) that are not directly specified by the International Standard, (2) that are specified and agreed to outside the International Standard, and (3) that may serve as trial usage for future editions of the International Standard.

6.2 Non-conforming implementations

An implementation that does not conform to this International Standard (either strictly conforming or merely conforming), is a non-conforming implementation.

6.3 Obligation of data elements

There are four kinds of obligation attributes for data elements: mandatory, optional, conditional, and extended. The obligation attribute concerns the validity of the data structure.

6.3.1 Mandatory data elements

Mandatory data elements are always required for the data structure to be valid.

All data instances are required to include these elements. All data applications are required to support these elements.

An implementation that does not support or include one or more mandatory data elements is a non-conforming implementation.

6.3.2 Optional data elements

Optional data elements are permitted, but not required, for the data structure to be valid.

A data instance is permitted, but not required, to include these data elements. Because all data repositories and data readers are required to support all valid (strictly conforming) data instances, effectively, data repositories and data readers are required to support all optional data elements. This might be confusing because "optional" is not optional for data repositories and data readers — the obligation attribute "optional" applies to the validity of the data structure ("optional" is optional for instances of the aggregate datatype). A data writer is required to generate and produce the optional elements of each data instance that is generated and produced.

An implementation that includes or supports an optional data element, but includes or supports it in ways that are inconsistent with a normative document, is a non-conforming implementation. The attribute "optional" does not imply that the implementation has license to implement the data element in any way ("at the option of the implementor"); if the data element is implemented, its requirements are specified in a standard.

NOTE Thus, if optional data element **X** has the datatype **`characterstring`**, then either it doesn't exist or it exists as a **`characterstring`**; if **X** exists, but is not of the **`characterstring`** datatype, then the implementation is not conforming.

6.3.3 Conditional data elements

Conditional data elements are required, but their requirement is dependent upon certain conditions, as defined in a standard. Each conditional data element may individually have a set of conditions. If the conditions are met, the data element is required to be included for the data structure to be valid.

Thus, a data instance is required to include these elements if, individually, each condition is met. By the same reasoning as for optional data elements (above), all data repositories and data readers are required to support all conditional data elements. By the same reasoning above, a data writer is required to support all the conditional elements for each and every data instance generated and produced.

An implementation that includes or supports a conditional data element, but includes or supports it in ways that are inconsistent with a standard, is a non-conforming implementation.

6.3.4 Extended data elements

Extended data elements are not permitted within strictly conforming implementations.

Extended data elements are permitted within conforming implementations to the extent that the implementation individually supports each extended data element, i.e., (1) the implementation allows and uses specific extended data elements, (2) the data interchange participants allow and use specific data elements, and (3) other extended data elements are not used.

For conforming implementations that support extended elements, these elements individually may have their own obligation attributes, e.g., it is possible to have mandatory extended data elements, optional extended data elements, and conditional extended data elements. These obligation attributes determine the validity of the data structure in the context of extended data elements, e.g., an optional extended data element (1) permits but does not require the data element for the data structure to be valid, (2) for conforming implementations that support this extended data element.

NOTE Mandatory extended data elements can cause interoperability problems because a mandatory extended data element (1) requires the data element to exist for the data structure to be valid, (2) for conforming implementations that support this extended data element. In other words, (1) only implementations that support this extended data element are interoperable; and (2) no strictly conforming implementations will interoperate because extended features are required for interoperability.

There are no generic techniques or methods for both supporting extended data elements or extension features and supporting full semantic interoperability; there are only specific techniques and methods for supporting extended data elements (e.g., supported to the extent allowed, as above).

The use of extended data elements outside these circumstances (unsupported environments) causes undefined behavior, which might be:

- appropriate, e.g., ignoring an offending data element if it is an unimportant feature
- inappropriate, e.g., ignoring an offending data element if it is an important feature, such as a security classification
- innocuous, e.g., error messages
- disruptive, e.g., error messages
- predictable, e.g., a program aborting, exiting ungracefully, exiting unexpectedly, or "hanging" indefinitely
- unpredictable, e.g., a program aborting, exiting ungracefully, exiting unexpectedly, or "hanging" indefinitely

There is no correct generalized method for handling undefined behavior. Any particular method for handling undefined behavior can be desirable, undesirable, or both.

Some bindings "relax" the processing of unrecognized extended data elements. Normally, extended data elements create undefined behavior but certain bindings "relax" these requirements to implementation-defined behavior or even ignoring unrecognized extended data elements — both of these "relaxed" processing requirements (implementation-defined behavior; ignoring unrecognized or extended data elements) can be less disruptive.

Extended data elements are both an obligation (data modeling) feature and a conformance level feature (strictly conforming vs. conforming).

6.4 Longevity of data elements

The following longevity attributes indicate intentions for incorporation into past, present, or future editions of a standard.

Longevity attributes are independent of obligation attributes.

6.4.1 Obsolete data elements

Obsolete data elements are defined in the current edition of a standard and may be defined in prior editions. The "obsolete" feature indicates that the definition of the data element is intended to be removed from future editions of a standard.

Implementations should not use obsolete data elements. Implementations that do use obsolete data elements should plan accordingly for future editions of a standard.

An implementation's use of an obsolete data element does not imply that the implementation is non-conforming. Strictly conforming implementations and conforming implementations may still use obsolete data elements for this edition of a standard.

The "obsolete" feature is independent of the obligation attribute, so there might be obsolete mandatory data elements, obsolete optional data elements, obsolete conditional data elements, and obsolete extended data elements.

6.4.2 Reserved data elements

Reserved data elements are not defined in a standard. Data elements may be reserved because (1) they were defined in previous edition(s) of a standard, or (2) they will be defined in some future edition(s) of a standard.

A reserved data element is not permitted in a strictly conforming implementation.

A "reserved data element" might be used in a conforming implementation if (1) the reserved data element were defined, (2) it were defined as an extended data element, and (3) the extended data element were "supported" by implementations and data interchange participants (see below). In other words, a particular implementation extends or overrides (the non-definition of) the "reserved data element" by defining implementation extensions.

Although the "reserved" feature is independent of the obligation attribute, a reserved data element has no definition. Therefore, there are no reserved mandatory data elements, no reserved optional data elements, no reserved conditional data elements, and no reserved extended data elements because mandatory data elements, optional data elements, conditional data elements, and extended data elements all imply a definition of a data element, which conflicts with the undefined nature of "reserved".

Data elements that are defined, but are to be incorporated into future editions of a standard, are extended data elements (i.e., they are not reserved data elements). As these extended data elements become incorporated into a future edition, they will become mandatory data elements, optional data elements, or conditional data elements. Additionally, provisional data elements may be used in transition to a new edition of a standard.

Typically, extended data elements are defined in a specification outside the standard.

Extended data elements are not required for this edition of a standard, i.e., (1) extended data elements are prohibited for strictly conforming systems; (2) extended data elements are not required for conforming systems; and (3) extended data elements, if defined, are not in a standard.

Some bindings "relax" the processing of unrecognized data elements, such as reserved data elements. Normally, reserved data elements create undefined behavior but certain bindings "relax" these requirements to implementation-defined behavior or even ignoring unrecognized or reserved data elements — both of these "relaxed" processing requirements (implementation-defined behavior; ignoring unrecognized or reserved data elements) can be less disruptive.

Conforming implementations may use extended data elements to the extent permitted by the implementation and data interchange participants. See "Extended Data Elements" (above) for further details.

Reserving a data element so that it cannot be overridden by extended data elements is achieved by defining an optional data element with ISO/IEC 11404 datatype **`void`**.

6.5 Recursive and contextual nature of obligation and longevity

An obligation attribute or a longevity attribute of an aggregate data element applies to the aggregate itself, but only indirectly to its components. In the context of the existence of an aggregate and its components, each component individually has its own obligation and longevity attributes (among other attributes). This determination of context and obligation/longevity attributes is applied recursively for all aggregate data elements.

EXAMPLE A data element **X** is optional, and **X** has two subelements: **Y** is mandatory and **Z** is optional. Letting the notation **P.Q** represent the subelement **Q** of **P**, then

— if **X** does not exist, then **X.Y** and **X.Z** cannot exist; stated differently, if **X.Y** or **X.Z** exists, then **X** exists
— if **X** exists, then **X.Y** is required to exist for all conforming implementations
— if **X** exists, then **X.Z** is permitted to exist for all conforming implementations
— if **X** exists and **X.Y** does not exist, then the implementation is non-conforming

Thus, **Y** only becomes mandatory if **X** exists.

7 Annex B: Conformance labels (informative)

A conformance labels may summarize implementation conformance statements (ICSs). Conformance labels should be used to convey ICS information via manual, semi-automated, and automated methods. The methods and techniques for associating or affixing a conforming label are outside the scope of this Standard.

The following is a summary of the possible implementation varieties used in implementation conformance statements (ICS) and their conformance labels:

[EDITOR'S NOTE: TEXT TO BE SUPPLIED.]

NOTE An implementation may claim more than one type of conformance in its implementation conformance statement (ICS).