Conformance Requirements Guideline

Version 0.1

Sept 10, 2001

Editors:

Lynne Rosenthal (lynne.rosenthal@nist.gov)
Mark Skall (mark.skall@nist.gov)

Contributors:

Lofton Henderson (lofton@rockynet.com)

Abstract

Document identifying the conformance requirements that need to be included or addressed in OASIS and ebXML specifications. Target audience is primarily specification developers, followed by conformance test suite developers.

Status of this Document

First Draft

Document Version History

22 Aug 2001 updated based on Aug 16 Telecon.
2 Aug 2001 initial draft

Reference Documents

ebXML Technical Architecture Specification, Conformance Clause (http://www.ebxml.org/specs/index.htm).

ISO Guide 2: Standardization and related activities – General vocabulary.

1. Introduction

The objective of this document is to identify the conformance requirements that shall be included or addressed in OASIS specifications. Conformance requirements are the expression, in the form of a statement, which conveys the criteria to be fulfilled [ISO Guide 2]. The conformance requirements are stated in a conformance clause or statement within the specification. This document describes the purpose and scope of a conformance clause as well as the associated issues that a conformance clause should address. Where ever possible, sample text and examples will be given.

The information contained is produced as the result of extensive experience in the development and implementation of conformance clauses and test suites for consensus standards and specifications. It is based on the principles and requirements prescribed by international standards (e.g., ISO/IEC and IEEE) as well as extractions from ebXML, OASIS and W3C specifications.

2. Scope and Audience

This document specifies the general requirements and definitions concerning conformance and related issues. It is intended to contribute fundamentally towards mutual understanding amongst developers of specifications and conformance test suites and tools. It is also intended to provide a suitable source for teaching and for reference, briefly covering basic theoretical and practical principles of conformance.

It is not the aim of this document to define specific conformance requirements for any specific specification – this is the responsibility of committees chartered to develop functional specifications.

This document is intended primarily for the developers of specifications to help enable them to develop a conformance statement within their specification and to create a testable, unambiguous specification. Secondary audiences include, but are not limited to: developers of conformance test suites, software implementers, international standards bodies, and other industry organizations.

3. Conformance (to this document)

[Ed Note: should this section be the last section of this document?]

In order to conform to this document *[Ed note: specification?]*, an OASIS specification shall contain a conformance clause.

4. Normative references

The following normative documents contain provisions, which through reference in this text, constitute provisions of this document. At the time of publications, the editions indicated below were valid. All standards are subject to revision, and parties to

agreements based on this document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ISO/IEC Guide 2: Standardization and related activities – General vocabulary
ISO/IEC Directives Part 3: Rules for the structure and drafting of International Standards.
ebXML Technical Architecture Specification
RFC 2119: Keywords for use in RFC's to Indicate Requirement Levels

5. Terms and definitions

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

Certification – the acknowledgement that a validation has been completed and the criteria established by the certifying organization has been met. Certification has a legal connotation.
Conformance – the fulfillment of a product, process, or service of specified requirements.
Conformance Testing – a method of verifying implementations of a specification to determine whether or not deviations from the specification exist.
Implementation – the realization of a specification – it can be a software product, system, program, protocol, or document instance.
Validation – the process of testing software for conformance to a specific specification.

6. Conformance Clause

The conformance clause is a section of a specification that defines the requirements, criteria, or conditions that must be satisfied to claim conformance. The conformance clause identifies *what* must conform and *how* conformance can be met. Typically it is a high-level description of what is required of implementers and application developers. It may refer to other parts of the standard. It may specify sets of functions, which may take the form of profiles, levels, or other structures. Additionally, it may specify minimal requirements for certain functions and minimal requirements for implementation-dependent values. It may also specify the permissibility of extensions, options, and alternative approaches and how they are to be handled.

Every specification shall contain a conformance clause.

6.1. Rationale for a conformance clause

A conformance clause can:

- Ensure a common understanding of conformance and what is required to claim conformance to a specification,
- Ensure that conformance is consistently addressed within a specification or across related specifications,
- Promote interoperability and open interchange,

- Encourage the use of applicable conformance test suites as well as promote uniformity in the development of conformance test suites.

6.2. Conformance keywords

There are specific words that are used throughout the specification to denote whether or not requirements are mandatory, optional, or suggestions. Using these keywords help to identify the testable statements in a specification. Although the keywords used within the ISO/IEC community differ from the keywords used within the IETF communities, they achieve the same results.

ISO Keywords:

SHALL – to indicate requirements strictly to be followed in order to conform to the standard and in which no deviation is permitted. Equivalent expressions include: is to, is required to, has to, it is necessary. Do not use MUST as an alternative for shall.
SHALL NOT - converse of SHALL.
SHOULD – to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others.
SHOULD NOT – converse of SHOULD.
MAY – to indicate a course of action permissible within the limits of the standard. Equivalent expressions include: is permitted, is allowed.
NEED NOT – to indicate a course of action is not required.CAN – statement of possibility and capability, whether material, physical or causal. Equivalent expressions include: be able to, it is possible to.
CANNOT – converse of CAN.

IETF Keywords (RCF2119)

MUST - the requirement is an absolute requirement of the specification.
MUST NOT – the requirement is an absolute prohibition of the specification.
REQUIRED – see MUST.
SHALL – see MUST.
SHALL NOT – see MUST NOT.
SHOULD – there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a difference course.
SHOULD NOT – there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
REOMMENDED – see SHOULD.
MAY - the item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option. MUST be prepared to interoperate with another implementation which does include the option, though

perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation that does not include the option (except, of course, for the feature the option provides.)

Additionally keywords include:

MANDATORY
OPTIONAL
NORMATIVE
INFORMATIVE

7. Issues to Address in a Conformance Clause

7.1. What needs to conform

The conformance clause identifies the "class of products" that will be developed, where class of product may be an implementation, application, service, and/or protocol. Additionally, the clause specifies the conditions that shall be met in order to claim conformance for that class of product. There may be several classes of products that are identified, each with its own conformance statement or set of conformance criteria.

For example, the OASIS SAML Conformance Clause (draft Aug 17, 2001) provides conformance statements for implementations of SAML (e.g. implementing systems, tools) and applications that execute on SAML implementations.

7.2. Profiles and Levels

Profiles are used as a method for defining subsets of a specification by identifying the functionality, parameters, options, and/or implementation requirements necessary to satisfy the requirements of a particular community of users. Specifications that explicitly recognize profiles should provide rules for profile creation, maintenance, registration and applicability. Appendix A provides additional information on profiles.

Levels are used to indicate nested subsets of functionality, ranging from minimal or core requirements to full or complete functionality. Typically, level 1 is the minimal or core of the specification that must be implemented by all products. Level 2 includes all of level 1 and also additional functionality. This nesting continues until level n, which consists of the entire specification.

It is possible for a specification to have both profiles and levels. If profiles and/or levels are defined, the conformance clause specifies which (if any) of these profiles and/or levels is mandatory. Additionally, any conditions associated with a particular profile, level or combination of these needs to be specified.

7.3. Extensions

An extension to a specification is a mechanism to incorporate functionality beyond what is defined in the specification. An extension may be *private* (often vendor specific) or may be *public* (a full description of the extension is public). Private extensions are usually truly private, i.e., valid for a specific implementation or are only known by prior agreement between implementations. Public extensions are extensions in which the syntax, semantics, identifiers, etc are defined and published allowing anyone to implement the extended functionality.

The presence of extensions can create serious problems in open interchange environments. Clearly, there are two main approaches to handling implementation specific extensions to a specification – to disallow extensions or to allow them.

7.3.1. Disallow Extensions

If extensions are forbidden, each implementation must precisely implement the complete specification. This is called *strict conformance.* This may be a desirable condition to impose on applications of a specification. For example, a software program or XML document instance.

7.3.2. Allow Extensions

If extensions are allowed, each implementation shall fully support all required functionality of the specification exactly as specified and the extensions shall not contradict nor cause the non-conformance of functionality defined in the specification. This more common approach usually includes some additional, more specific, requirements in the conformance clause, such as:

- Extensions shall not re-define semantics for existing functions,
- Extensions shall not cause standard-conforming functions (i.e., functions that do not use the extensions) to execute incorrectly,
- Extensions shall follow the principles and guidelines of the specification they extend, that is, the specifications must be extended in a standard manner (see section below),
- For implementations and/or applications that contain extensions, extensions shall be clearly described in supporting documentation and the extensions shall be marked as such within the implementation/application,
- For implementations that contain extensions, there shall be a mode under which the implementation can be directed to produce only conformant files (documents) or to operate in a strictly conformant manner.

7.3.3. Mechanism to allow extensions

One mechanism to allow extensions within a specification is to provide a standard way of defining the extension. Basically, this provides a “standard way of being non-standard”. This helps to ensure predictable handling of extensions, that is, its recognition as such and the appropriate action (i.e., to ignore or to implement). The nature of the extension may dictate the method for defining the extension. It may be possible to define a function that indicates external (from the specification) functionality. This external function may take the form of an escape or control character or be an identifier, which

whenever invoked indicates an extension follows. Another method, especially when extending a list of numeric parameters is to use a scheme where positive values represent standardized values and negative values are reserved for private use.

For example, the ISO Computer Graphics Metafile (CGM) is the standard on which the W3C WebCGM Recommendation is based. It provides both a standard function (GDP element) for defining private graphics functionality, as well as the use of negative values to define private values.
Another mechanism that minimizes interoperability problems when extensions are allowed is to have a register for extensions. This document, distinct from the official specification, contains a list of recognized extensions to the standard. See section below.

7.3.4. Registration of implementer extensions or defined values

Registration is a procedure that allows extensions to be acknowledged and made available to the public. The procedures provide for the extension to have some rigor and technical review. Typically, the committee developing the specification is responsible for processing the registration of an extension, thus ensuring adequate quality of a proposed extension and a technical description sufficient to be uniformly implementable. Often registered extensions may migrate into a later version of the specification.

7.4. Implementation defined values

Specifications sometimes need to address:

- Implementation dependent ranges, e.g., minimum or maximum allowed sized,
- Values that may be different for different conforming implementations of the standard (protocols?),
- Features reserved for registration.

7.5. Alternate approaches

Specifications may describe several different ways to accomplish its operation (e.g., a choice of file formats, protocols, or codes). In such a case, the conformance clause shall specify under what conditions an implementation is considered to be conformant. Some possible ways to define conformance include mandating that an implementation shall:

1. implement only one approach (in which case, must it implement a specific approach or any one approach is sufficient),
2. implement every approach,
3. be allowed to implement none of the approaches.

Note: if the specification doesn't describe the different approaches, this becomes an implementation detail irrelevant to conformance.

7.6. International Character codes

[What needs to go here?]

8. What else can be addressed?

Include assertions as part of the specification
Specify a testing program (e.g., validation and certification process)

Appendix A: Profiles
(Informative)

[Ed Note: The following is taken from ISO 8632 Computer Graphics Metafile Standard. Needs to be edited]

A profile of a specification defines the options, elements, and parameters necessary to accomplish a particular function and maximize the probability of interchange between systems implementing the profile. Profiles are defined by application constituencies who agree to adhere to the same subset of the specification. A profile may be a subset of a single specification or may be part of the set of interrelated standards and profiles assembled for the purpose of accomplishing a larger functional purpose. A profile shall not specify any requirement that would contradict or cause non-conformance to its specification.

A profile may:

- Give the meaning of implementation dependent semantics of some elements,
- Enforce common resolution of ambiguous semantics,
- Ensure that identical use of identical elements and parameter values have the same meaning,
- Specify subsets or groupings of publicly defined extensions,
- Prohibit undefined or ill-defined elements or parameter values.

Profiles provide a means to:

- Improve interoperability between implementations by inhibiting the proliferation of private subsets of a specification,
- Provide a foundation for testing and promote uniformity of conformance tests,
- Enhance the availability of consistent implementations of a profile.