Guidelines to Writing Conformance Requirements for OASIS Specifications Version 0.5

7 August 2007

Guideline URIs:

Editor(s):

Martin Chapman

Abstract:

This document provides guidelines on how to write conformance statements for OASIS specifications. The target audience is primarily specification writers and TC members.

Status:

This document is not yet approved. The template is only being used as an editing and review convenience, and will not be used when publishing to the wiki.

Interested parties should send comments on this specification to the TAB by using the “Send A Comment” button on the TAB’s web page at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tab.

Table of Contents

1 Introduction

Effective from 1st June 2007, the OASIS TC Process requires that each specification contain a separate conformance section listing the conformance clauses that need to be observed by implementers or users of the specification in order to claim successful use of a specification.

This document provides guidelines on how to write conformance statements for OASIS specifications. While it is not a requirement to follow these guidelines, it is recommended that TC adopt the advice herein in order to achieve consistency across OASIS specifications.

The target audience is primarily specification writers and TC members.

This document describes the purpose and scope of a conformance clauses, associated issues that a conformance clause shall address as well as issues that a conformance clause may address. Wherever possible, sample text and examples will be given.

The information contained is produced as the result of extensive experience in the writing of specifications and draws upon guidelines requirements from ISO/IEC, IEEE, W3C, WS-I and OASIS.

IEEE http://www.ietf.org/rfc/rfc2119.txt

ISO/IEC Guide 2:2004 Standardization and related activities – General vocabulary (not free)

ISO/IEC Directives Part 2: Rules for the structure and drafting of International Standards

OASIS http://www.oasis-open.org/apps/org/workgroup/ioc/download.php/305/conformance_requirements-v1.pdf

W3C http://www.w3.org/TR/qaframe-spec/

WS-I http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html#conformance

2 Terms and Definitions

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

Conformance – the fulfillment of a product, document, process, or service of specified requirements.

Conformance Claim – a declaration that a product or artifact meets the requirements of one or more conformance clauses. A Conformance claim SHOULD accompany a statement of use declaration when a Committee Specification is being advanced to OASIS Standard.

Conformance Clause – A statement in the Conformance section of a specification that provides a high-level description of what is required for an atrifact to conform. It, in turn, refers to other parts of the specification for details...   A conformance clause must reference one or more normative statements, directly or indirectly, and may refer to another conformance clause.

Conformance Target – an artifact such as a protocol, document, platform, process or service, which is the subject of conformance clauses and normative statements. There may be several conformance targets defined within a specification, and these targets may be diverse so as to reflect different aspects of a specification. For example, a protocol message and a protocol engine may be different targets.

Conformance Testing – a method of verifying implementations of a specification to determine whether or not deviations from the specification exist. *Conformance testing is not covered in these guidelines.*

Normative Statement – a statement made in the body of a specification that defines prescriptive requirements on a conformance target.

3 Conformance Keywords

When writing normative statements and conformance clauses, specific keywords must be used throughout the specification to denote whether or not requirements are mandatory, optional, or suggested. Using a standard set of key word helps to easily identify the normative statements and conformance clauses.

OASIS specifications SHOULD use the following keywords from IETF RFC 2119. This is the default terminology to be used in all OASIS specifications:

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 different 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.

RECOMMENDED – 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 that does not include a particular option MUST be prepared to interoperate with another implementation that 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).

While RFC2119 permits the use of synonyms, to achieve consistency across specifications it is recommended that MUST is used instead of SHALL, and MUST NOT instead of SHALL NOT.

RFC2119 allows both uppercase and lowercase to be used for a keyword, however to enable easy identification of the keywords and consistency across specifications uppercase must be used for keywords at all times.

Alternative keywords:

Some OASIS specifications have ambitions to be advanced to other bodies such as ISO/IEC and ITU-T. In those cases it is permissible to use the ISO keywords instead of the default RFC 2119 ones. A specification that makes se of ISO keywords must explicitly declare this in the specification.

Under no circumstances should the RFC2119 or ISO styles be used in the same documents.

The ISO keywords are:

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.

4 Normative Statements

A specification broadly consist of descriptive text and normative statements. The normative statements define what a conformance target must do to adhere to that part of the specification, and the descriptive text provides background information, descriptions and examples. Descriptive text is not normative and is used to provide contextual information. Normative statements are those that use the RFC2119 keywords (or the ISO keywords if these have been chosen instead), descriptive text does not use these reserved words as keywords.

The following example is taken from the WS-BPEL specification:

http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html#_Toc164738482:

> *WS-BPEL supports extensibility by allowing namespace-qualified attributes to appear on any WS-BPEL element and by allowing elements from other namespaces to appear within WS-BPEL defined elements. This is allowed in the XML Schema specifications for WS-BPEL.*
>
> *Extensions are either mandatory or optional (see section14). … In the case of mandatory extensions not supported by a WS-BPEL implementation, the process definition MUST be rejected. Optional extensions not supported by a WS-BPEL implementation MUST be ignored.*

The first paragraph in the sample is descriptive and provides background information on how to extend the WS-BPEL language. It does not contain any RFC2119 keywords. The second paragraph contains normative statements that directs implementers and users what to do with unknown extensions, and uses the keywords to define what has to be done.

Normative statements form the core of a specification and it is essential that each statement is clear, concise, and unambiguous. It must be clear what conformance target the statement applies to, concise enough to be understood and should be clear what needs to be done.

Before normative statements are made in a specification, it is recommended that the conformance targets are defined. From the above example, a WSPEL implementation is a conformance target. A specification may define one or more conformance targets as appropriate. An example would be

Normative statements must be referenceable so that a statement may be referenced from another part of a specification, but more importantly so they can be referenced from conformance clauses. Should the specification writer want fine grained referencability, each normative statement should be uniquely labeled. This is the approach adopted by some organizations.. If the writer deems this to be too fine grained, then normative statements can appear in there own self contained section, and the section referenced.

Where possible normative statements must not contradict each other, but there are times when this is unavoidable. In these cases, there must be a clear way to separate them so that implementers and users are not required to implement conflicting normative statements. This must be done by correctly writing independent conformance clauses that reference each normative statement respectively.

Examples of Normative Statements

The following example is taken from the Emergency management specification: http://docs.oasis-open.org/emergency/edxl-de/v1.0/EDXL-DE_Spec_v1.0.pdf

In the discussion on representing longitude and latidude the followng normative statement is made:

> *Latitudes north of the equator MAY be specified by a plus sign (+), or by the absence of a minus sign (-), preceding the designating degrees. Latitudes south of the Equator MUST be designated by a minus sign (-) preceding the digits designating degrees. Latitudes on the Equator MUST be*

designated by a latitude value of 0.

This normative statement uses RFC2119 wording, it is clear what the subject is, and provides concise instructions. It is also self contained in that it does not introduce other concepts in the statement not related to latitude.

The following example is made up to protect the innocent and is an example of a badly written normative statement.

When processing a document some features can be ignored and not displayed.

Firstly for a normative statement the recommended keywords are not used; can needs to be replaced with MAY or MUST, and not needs to be qualified. Secondly it is ambiguous in that it is not clear what features can be ignored; this would need to be qualified. Finally, a conformance target has not been defined as it is not clear what processes a documents. A better phrasing would be:

A word processor MAY ignore the following features contained within a documents and SHALL choose NOT to display these features: …list of features..

5 Conformance Section and Clauses

A Conformance section of a specification must contain at least one conformance clause. A specification may define a number of different clauses in the conformance section, where each clause identifies different conformance targets that may wish to conform, such as an implementation, a document, an authoring tool, a protocol etc. Defining more that one conformance clause segments the specification up into different targets that may be conformed to.

A conformance clause identifies what must be conformed to and this is done by reference to normative statements in the specification. A conformance clause therefore identifies a sub-set of the normative statements defined in the body of a specification. Thought should be put into the granularity of references to normative statements. If there are many normative statements referenced by a conformance clause then simply referencing each statement might not be readable or easy to follow. In such cases it may be better to revisit the normative statements and group them into larger referenceable units.

Talk about granularity of references to normative statements e.g WSRP being a bad example.

Conformance clauses must be defined with the separate conformance section of an OASIS specification, and it is recommended that conformance clauses only appear in the conformance section .

A specification must impose no restrictions about who can make a statement of use claiming conformance to one or more conformance clauses (e.g., vendor, user, third party).

There may be more than one conformance clause in a specification, and like normative statements they must be clear, concise, and unambiguous.

Each conformance clause must be uniquely labeled.

When more than one conformance clause exists in a specification the relationship between them must be clearly defined.

There are five types of relationships that are common and should be considered:

Combined – this defines a conformance clause that combines other clauses. For example, clause A, B, and C

Alternates – this defines a distinct conformance clause that exists on it own without reference to another one. For examples say that an implemetor may implement clause A,B or C.

Level/extension – this defines a conformance clause by building on top of another one. For example, clause B requires A and these addition normative statements.

Relaxations – this defines a conformance clause by removing some of the requirements of another conformance clause. For example, clause A but without normative statements x, y and z required by clause A.

It is possible to use a mixture of the above. For example, Clause B extends clause A and requires clauses D or E. Care must be taken though not to over complicate, so that each conformance clause is easy to understand and not open to different interpretations.

If any conformance clause references another one, it is essential that there are no normative statements within the clauses that contradict each other. If there is a contradiction, then the writers should either examine and try to remove the contradiction in the specification text itself or state in the conformance clause what must be done to avoid the contradiction, for example by stating that one overrides the other.

When multiple conformance clauses exist, it must be clear which are the top-level . It is these top-level clauses that relate to the conformance targets that users and vendors can conform to, and are the clauses that should be referenced when claiming conformance to a specification and in making statements of use. For example, a specification may define 5 conformance clauses A, B, C, D and E, where D and E are referenced only by C; A, B and C are the top-level clauses in this case.

Within the conformance section, a clear statement MUST be made as to how optional normative statements (i.e. those using the MAY keywords) are handled. This decision relates to the type of conformance target and the use of the specifications. For example a document that claims conformance to a schema does not have to use any optional features. However, in another scenario, a protocol target should implement optional features in case another party using the protocol makes use of the optional features. In deciding how to dispose of option features, issues that effect interoperability and portability need to be considered.

Examples

The following example is taken from ebXML Registry Services Specification v2.0: http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrs.pdf

5.5 Implementation Conformance

An implementation is a conforming ebXML Registry if the implementation meets the conditions in Section 5.5.1. An implementation is a conforming ebXML Registry Client if the implementation meets the conditions in Section 5.5.2. An implementation is a conforming ebXML Registry and a conforming ebXML Registry Client if the implementation conforms to the conditions of Section 5.5.1 and Section 5.5.2. An implementation shall be a conforming ebXML Registry, a conforming ebXML Registry Client, or a conforming ebXML Registry and Registry Client.

5.5.1 Conformance as an ebXML Registry

An implementation conforms to this specification as an ebXML Registry following conditions:

1. *Conforms to the ebXML Registry Information Model [ebRIM].*
2. *Supports the syntax and semantics of the Registry Interfaces and*
3. *Supports the defined ebXML Registry Schema (Appendix B).*
4. *Optionally supports the syntax and semantics of Section 8.3, SQL*

5.5.2 Conformance as an ebXML Registry Client

An implementation conforms to this specification, as an ebXML Registry if it meets the following conditions:

1. *Supports the ebXML CPA and bootstrapping process.*
2. *Supports the syntax and the semantics of the Registry Client Interfaces.*
3. *Supports the defined ebXML Error Message DTD.*
4. *Supports the defined ebXML Registry Schema (Appendix B).*

This sample, defines three conformance clauses. Section 5.5.1 is a conformance clause for an ebxml Registry conformance target. Section 5.5.2, is a conformance clause for an ebxml Registry Client. Both these clauses reference normative material. The third conformance clause is the introduction paragraph to section 5. This defines three top level conformance clauses, references the clauses containing the details, and defines the relationship between the top-level clauses. In this case it uses a mix of alternative and combined styles: an implementation is either a Registry, or a Client, or a Registry and a Client.

6 Checklist

Are you using the right keywords RFC 2119, and in uppercase.

If you are using ISO keywords have you explicitly stated this in the specifications.

Have you defined your conformance target(s)

Are all normative statements clearly identifiable.

Are all normative statements understandable, clear, and concise.

Are all normative statements referenced directly or indirectly from a conformance clause?

A normative statement that is not related to any conformance clause has no meaning.

Is each normative statement related to a conformance target(s).

Is there a separate section containing the conformance clauses?

Are all conformance clauses clearly identifiable.

Are all conformance clauses understandable, clear, and concise.

Are the top-level conformance clauses clearly identified and related to a conformance target

Is the relationship between all conformance clauses clearly defined using combinations of combined, alternative, level and profile styles.

Are all conformance clauses either top-level or referenced directly or indirectly from a top-level conformance clause?

A conformance clause that is not related to any top-level conformance clause has no meaning.

Are there any contradictions within a conformance clause and any referenced conformance clauses and normative statements. If there are then have these been explicitly noted and have any rules to over-ride the contradictions been made.