

---

---

**Information technology — Web Services  
Interoperability — WS-I Basic Profile  
Version 1.1**

*Technologies de l'information — Interopérabilité des services  
du Web — Profil de base WS-I, version 1.1*

iTeh Standards  
(<https://standards.itih.ai>)  
Document Preview

[ISO/IEC 29361:2008](https://standards.itih.ai/catalog/standards/iso/ecbef956-56f7-415b-86f6-6a05f79423ac/iso-iec-29361-2008)

<https://standards.itih.ai/catalog/standards/iso/ecbef956-56f7-415b-86f6-6a05f79423ac/iso-iec-29361-2008>

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

**iTeh Standards**  
**(<https://standards.iteh.ai>)**  
**Document Preview**

[ISO/IEC 29361:2008](https://standards.iteh.ai/catalog/standards/iso/ecbef956-56f7-415b-86f6-6a05f79423ac/iso-iec-29361-2008)

<https://standards.iteh.ai/catalog/standards/iso/ecbef956-56f7-415b-86f6-6a05f79423ac/iso-iec-29361-2008>



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

## Contents

Foreword .....	viii
1 Scope and introduction .....	1
1.1 Scope .....	1
1.2 Relationships to Other Profiles .....	1
1.3 Changes from Basic Profile Version 1.0 .....	2
1.4 Guiding Principles .....	2
1.5 Notational Conventions .....	3
1.6 Profile Identification and Versioning .....	4
2 Profile Conformance .....	5
2.1 Conformance Requirements .....	5
2.2 Conformance Targets .....	6
2.3 Conformance Scope .....	6
2.4 Claiming Conformance .....	7
3 Messaging .....	7
3.1 SOAP Envelopes .....	8
3.1.1 SOAP Envelope Structure .....	8
3.1.2 SOAP Envelope Namespace .....	9
3.1.3 SOAP Body Namespace Qualification .....	9
3.1.4 Disallowed Constructs .....	9
3.1.5 SOAP Trailers .....	9
3.1.6 SOAP encodingStyle Attribute .....	10
3.1.7 SOAP mustUnderstand Attribute .....	10
3.1.8 xsi:type Attributes .....	10
3.1.9 SOAP1.1 attributes on SOAP1.1 elements .....	11

3.2 SOAP Processing Model..... 11

    3.2.1 Mandatory Headers ..... 11

    3.2.2 Generating mustUnderstand Faults ..... 11

    3.2.3 SOAP Fault Processing ..... 11

3.3 SOAP Faults ..... 12

    3.3.1 Identifying SOAP Faults..... 12

    3.3.2 SOAP Fault Structure ..... 12

    3.3.3 SOAP Fault Namespace Qualification ..... 13

    3.3.4 SOAP Fault Extensibility ..... 14

    3.3.5 SOAP Fault Language ..... 14

    3.3.6 SOAP Custom Fault Codes ..... 14

3.4 Use of SOAP in HTTP ..... 15

    3.4.1 HTTP Protocol Binding ..... 16

    3.4.2 HTTP Methods and Extensions ..... 16

    3.4.3 SOAPAction HTTP Header..... 16

    3.4.4 HTTP Success Status Codes ..... 17

    3.4.5 HTTP Redirect Status Codes ..... 17

    3.4.6 HTTP Client Error Status Codes..... 18

    3.4.7 HTTP Server Error Status Codes ..... 18

    3.4.8 HTTP Cookies ..... 18

4 Service Description..... 19

    4.1 Required Description..... 20

    4.2 Document Structure ..... 20

        4.2.1 WSDL Schema Definitions ..... 20

iteh Standards  
(<https://standards.iteh.ai>)  
Document Preview

<https://standards.iteh.ai/catalog/standards/iso/29361-2008>  
<https://standards.iteh.ai/catalog/standards/iso/29361-2008>

4.2.2 WSDL and Schema Import.....	21
4.2.3 WSDL Import location Attribute Structure.....	22
4.2.4 WSDL Import location Attribute Semantics.....	22
4.2.5 Placement of WSDL import Elements .....	23
4.2.6 XML Version Requirements.....	24
4.2.7 XML Namespace declarations.....	24
4.2.8 WSDL and the Unicode BOM.....	24
4.2.9 Acceptable WSDL Character Encodings.....	24
4.2.10 Namespace Coercion .....	24
4.2.11 WSDL documentation Element.....	25
4.2.12 WSDL Extensions.....	25
4.3 Types .....	26
4.3.1 QName References.....	26
4.3.2 Schema targetNamespace Structure.....	26
4.3.3 soapenc:Array .....	26
4.3.4 WSDL and Schema Definition Target Namespaces .....	28
4.4 Messages.....	28
4.4.1 Bindings and Parts .....	29
4.4.2 Bindings and Faults.....	30
4.4.3 Declaration of part Elements .....	31
4.5 Port Types.....	31
4.5.1 Ordering of part Elements .....	31
4.5.2 Allowed Operations .....	32
4.5.3 Distinctive Operations.....	32
4.5.4 parameterOrder Attribute Construction.....	32
4.5.5 Exclusivity of type and element Attributes .....	32

4.6 Bindings .....33

    4.6.1 Use of SOAP Binding .....33

4.7 SOAP Binding .....33

    4.7.1 Specifying the transport Attribute.....33

    4.7.2 HTTP Transport.....33

    4.7.3 Consistency of style Attribute .....34

    4.7.4 Encodings and the use Attribute.....34

    4.7.5 Multiple Bindings for portType Elements .....34

    4.7.6 Operation Signatures.....34

    4.7.7 Multiple Ports on an Endpoint.....35

    4.7.8 Child Element for Document-Literal Bindings .....35

    4.7.9 One-Way Operations .....35

    4.7.10 Namespaces for soapbind Elements .....36

    4.7.11 Consistency of portType and binding Elements.....36

    4.7.12 Describing headerfault Elements .....36

    4.7.13 Enumeration of Faults.....37

    4.7.14 Type and Name of SOAP Binding Elements .....37

    4.7.15 name Attribute on Faults.....38

    4.7.16 Omission of the use Attribute.....38

    4.7.17 Default for use Attribute .....38

    4.7.18 Consistency of Envelopes with Descriptions .....38

    4.7.19 Response Wrappers .....39

    4.7.20 Part Accessors .....39

    4.7.21 Namespaces for Children of Part Accessors .....39

4.7.22 Required Headers .....	41
4.7.23 Allowing Undescribed Headers.....	41
4.7.24 Ordering Headers .....	41
4.7.25 Describing SOAPAction.....	42
4.7.26 SOAP Binding Extensions .....	43
4.8 Use of XML Schema .....	43
5 Service Publication and Discovery .....	44
5.1 bindingTemplates.....	44
5.2 tModels .....	45
6 Security.....	46
6.1 Use of HTTPS.....	47
Appendix A: Referenced Specifications.....	49
Appendix B: Extensibility Points .....	50
Appendix C: Normative References .....	52
Appendix D: Defined Terms.....	53
Appendix E: Acknowledgements .....	55

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

ISO/IEC 29361 was prepared by the Web Services Interoperability Organization (WS-I) and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

ITeH Standards  
(<https://standards.iteh.ai>)  
Document Preview

[ISO/IEC 29361:2008](https://standards.iteh.ai/catalog/standards/iso/ecbef956-56f7-415b-86f6-6a05f79423ac/iso-iec-29361-2008)

<https://standards.iteh.ai/catalog/standards/iso/ecbef956-56f7-415b-86f6-6a05f79423ac/iso-iec-29361-2008>



# Information technology — Web Services Interoperability — WS-I Basic Profile Version 1.1

## 1 Scope and introduction

### 1.1 Scope

This International Standard defines the WS-I Basic Profile 1.1 (hereafter, "Profile"), consisting of a set of non-proprietary Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability.

Section 1 introduces the Profile, and explains its relationships to other profiles.

Section 2, "Profile Conformance", explains what it means to be conformant to the Profile.

Each subsequent section addresses a component of the Profile, and consists of two parts: an overview detailing the component specifications and their extensibility points, followed by subsections that address individual parts of the component specifications. Note that there is no relationship between the section numbers in this International Standard and those in the referenced specifications.

### 1.2 Relationships to Other Profiles

This Profile is derived from the Basic Profile 1.0 by incorporating any errata to date and separating out those requirements related to the serialization of envelopes and their representation in messages. Such requirements are now part of the Simple SOAP Binding Profile 1.0, identified with a separate conformance claim. This separation is made to facilitate composability of Basic Profile 1.1 with any profile that specifies envelope serialization, including the Simple SOAP Binding Profile 1.0 and the Attachments Profile 1.0. A combined claim of conformance to both the Basic Profile 1.1 and the Simple SOAP Binding Profile 1.0 is roughly equivalent to a claim of conformance to the Basic Profile 1.0 plus published errata.

This Profile, composed with the Simple SOAP Binding Profile 1.0 supercedes the Basic Profile 1.0. The Attachments Profile 1.0 adds support for SOAP with Attachments, and is intended to be used in combination with this Profile.

### 1.3 Changes from Basic Profile Version 1.0

This specification is derived from the [Basic Profile Version 1.0](#), and incorporates published errata against that specification. The most notable changes are:

- MESSAGE conformance target - Some requirements that had a MESSAGE conformance target in BP1.0 now use a new target, ENVELOPE. This facilitates alternate serialisations of the message, such as that described in the Attachments Profile.
- SOAP Binding - Requirements relating to the SOAP binding's serialization of the message have been moved to the Simple SOAP Binding Profile to facilitate other serializations.

### 1.4 Guiding Principles

The Profile was developed according to a set of principles that, together, form the philosophy of the Profile, as it relates to bringing about interoperability. This section documents these guidelines.

#### *No guarantee of interoperability*

It is impossible to completely guarantee the interoperability of a particular service. However, the Profile does address the most common problems that implementation experience has revealed to date.

#### *Application semantics*

Although communication of application semantics can be facilitated by the technologies that comprise the Profile, assuring the common understanding of those semantics is not addressed by it.

#### *Testability*

When possible, the Profile makes statements that are testable. However, such testability is not required. Preferably, testing is achieved in a non-intrusive manner (e.g., examining artifacts "on the wire").

#### *Strength of requirements*

The Profile makes strong requirements (e.g., MUST, MUST NOT) wherever feasible; if there are legitimate cases where such a requirement cannot be met, conditional requirements (e.g., SHOULD, SHOULD NOT) are used. Optional and conditional requirements introduce ambiguity and mismatches between implementations.

#### *Restriction vs. relaxation*

When amplifying the requirements of referenced specifications, the Profile may restrict them, but does not relax them (e.g., change a MUST to a MAY).

#### *Multiple mechanisms*

If a referenced specification allows multiple mechanisms to be used interchangeably, the Profile selects those that are well-understood, widely implemented and useful. Extraneous or underspecified mechanisms and extensions introduce complexity and therefore reduce interoperability.

#### *Future compatibility*

When possible, the Profile aligns its requirements with in-progress revisions to the specifications it references. This aids implementers by enabling a

graceful transition, and assures that WS-I does not 'fork' from these efforts. When the Profile cannot address an issue in a specification it references, this information is communicated to the appropriate body to assure its consideration.

#### *Compatibility with deployed services*

Backwards compatibility with deployed Web services is not a goal for the Profile, but due consideration is given to it; the Profile does not introduce a change to the requirements of a referenced specification unless doing so addresses specific interoperability issues.

#### *Focus on interoperability*

Although there are potentially a number of inconsistencies and design flaws in the referenced specifications, the Profile only addresses those that affect interoperability.

#### *Conformance targets*

Where possible, the Profile places requirements on artifacts (e.g., WSDL descriptions, SOAP messages) rather than the producing or consuming software's behaviors or roles. Artifacts are concrete, making them easier to verify and therefore making conformance easier to understand and less error-prone.

#### *Lower-layer interoperability*

The Profile speaks to interoperability at the application layer; it assumes that interoperability of lower-layer protocols (e.g., TCP, IP, Ethernet) is adequate and well-understood. Similarly, statements about application-layer substrate protocols (e.g., SSL/TLS, HTTP) are only made when there is an issue affecting Web services specifically; WS-I does not attempt to assure the interoperability of these protocols as a whole. This assures that WS-I's expertise in and focus on Web services standards is used effectively.

## 1.5 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119](#).

Normative statements of requirements in the Profile (i.e., those impacting conformance, as outlined in "[Conformance Requirements](#)") are presented in the following manner:

**Rnnnn** *Statement text here.*

where "nnnn" is replaced by a number that is unique among the requirements in the Profile, thereby forming a unique requirement identifier.

Requirement identifiers can be considered to be namespace qualified, in such a way as to be compatible with QNames from [Namespaces in XML](#). If there is no explicit namespace prefix on a requirement's identifier (e.g., "R9999" as opposed to "bp10:R9999"), it should be interpreted as being in the namespace identified by the conformance URI of the document section it occurs in. If it is qualified, the prefix should be interpreted according to the namespace mappings in effect, as documented below.

Some requirements clarify the referenced specification(s), but do not place additional constraints upon implementations. For convenience, clarifications are annotated in the following manner: **c**

Some requirements are derived from ongoing standardization work on the referenced specification(s). For convenience, such forward-derived statements are annotated in the following manner: **xxxx**, where "xxxx" is an identifier for the specification (e.g., "WSDL20" for WSDL Version 2.0). Note that because such work was not complete when this document was published, the specification that the requirement is derived from may change; this information is included only as a convenience to implementers.

Extensibility points in underlying specifications (see "[Conformance Scope](#)") are presented in a similar manner:

**Ennnn** *Extensibility Point Name - Description*

where "nnnn" is replaced by a number that is unique among the extensibility points in the Profile. As with requirement statements, extensibility statements can be considered namespace-qualified.

This specification uses a number of namespace prefixes throughout; their associated URIs are listed below. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

- **soap** - "http://schemas.xmlsoap.org/soap/envelope/"
- **xsi** - "http://www.w3.org/2001/XMLSchema-instance"
- **xsd** - "http://www.w3.org/2001/XMLSchema"
- **soapenc** - "http://schemas.xmlsoap.org/soap/encoding/"
- **wSDL** - "http://schemas.xmlsoap.org/wSDL/"
- **soapbind** - "http://schemas.xmlsoap.org/wSDL/soap/"
- **uddi** - "urn:uddi-org:api\_v2"

## 1.6 Profile Identification and Versioning

This document is identified by a name (in this case, Basic Profile) and a version number (here, 1.1). Together, they identify a particular *profile instance*.

Version numbers are composed of a major and minor portion, in the form "major.minor". They can be used to determine the precedence of a profile instance; a higher version number (considering both the major and minor components) indicates that an instance is more recent, and therefore supersedes earlier instances.

Instances of profiles with the same name (e.g., "Example Profile 1.1" and "Example Profile 5.0") address interoperability problems in the same general scope (although some developments may require the exact scope of a profile to change between instances).

One can also use this information to determine whether two instances of a profile are backwards-compatible; that is, whether one can assume that conformance to an earlier profile instance implies conformance to a later one. Profile instances with the same name and major version number (e.g., "Example Profile 1.0" and "Example Profile 1.1") MAY be considered compatible. Note that this does not imply anything about compatibility in the other direction; that is, one cannot assume that conformance with a later profile instance implies conformance to an earlier one.

## 2 Profile Conformance

Conformance to the Profile is defined by adherence to the set of *requirements* defined for a specific *target*, within the *scope* of the Profile. This section explains these terms and describes how conformance is defined and used.

### 2.1 Conformance Requirements

Requirements state the criteria for conformance to the Profile. They typically refer to an existing specification and embody refinements, amplifications, interpretations and clarifications to it in order to improve interoperability. All requirements in the Profile are considered normative, and those in the specifications it references that are in-scope (see "Conformance Scope") should likewise be considered normative. When requirements in the Profile and its referenced specifications contradict each other, the Profile's requirements take precedence for purposes of Profile conformance.

Requirement levels, using [RFC2119](#) language (e.g., MUST, MAY, SHOULD) indicate the nature of the requirement and its impact on conformance. Each requirement is individually identified (e.g., R9999) for convenience.

For example;

**R9999** WIDGETs SHOULD be round in shape.

This requirement is identified by "R9999", applies to the target WIDGET (see below), and places a conditional requirement upon widgets; i.e., although this requirement must be met to maintain conformance in most cases, there are some situations where there may be valid reasons for it not being met (which are explained in the requirement itself, or in its accompanying text).

Each requirement statement contains exactly one requirement level keyword (e.g., "MUST") and one conformance target keyword (e.g., "MESSAGE"). The conformance target keyword appears in bold text (e.g. "**MESSAGE**"). Other conformance targets appearing in non-bold text are being used strictly for their definition and NOT as a conformance target. Additional text may be included to illuminate a requirement or group of requirements (e.g., rationale and examples); however, prose surrounding requirement statements must not be considered in determining conformance.

Definitions of terms in the Profile are considered authoritative for the purposes of determining conformance.

None of the requirements in the Profile, regardless of their conformance level, should be interpreted as limiting the ability of an otherwise conforming implementation to apply security countermeasures in response to a real or perceived threat (e.g., a denial of service attack).

## 2.2 Conformance Targets

Conformance targets identify what artifacts (e.g., SOAP message, WSDL description, UDDI registry data) or parties (e.g., SOAP processor, end user) requirements apply to.

This allows for the definition of conformance in different contexts, to assure unambiguous interpretation of the applicability of requirements, and to allow conformance testing of artifacts (e.g., SOAP messages and WSDL descriptions) and the behavior of various parties to a Web service (e.g., clients and service instances).

Requirements' conformance targets are physical artifacts wherever possible, to simplify testing and avoid ambiguity.

The following conformance targets are used in the Profile:

- **MESSAGE** - protocol elements that transport the ENVELOPE (e.g., SOAP/HTTP messages)
- **ENVELOPE** - the serialization of the soap:Envelope element and its content
- **DESCRIPTION** - descriptions of types, messages, interfaces and their concrete protocol and data format bindings, and the network access points associated with Web services (e.g., WSDL descriptions) (from [Basic Profile 1.0](#))
- **INSTANCE** - software that implements a wsdl:port or a uddi:bindingTemplate (from [Basic Profile 1.0](#))
- **CONSUMER** - software that invokes an INSTANCE (from [Basic Profile 1.0](#))
- **SENDER** - software that generates a message according to the protocol(s) associated with it (from [Basic Profile 1.0](#))
- **RECEIVER** - software that consumes a message according to the protocol(s) associated with it (e.g., SOAP processors) (from [Basic Profile 1.0](#))
- **REGDATA** - registry elements that are involved in the registration and discovery of Web services (e.g. UDDI tModels) (from [Basic Profile 1.0](#))

## 2.3 Conformance Scope

The scope of the Profile delineates the technologies that it addresses; in other words, the Profile only attempts to improve interoperability within its own scope. Generally, the Profile's scope is bounded by the specifications referenced by it.