



# UBL Maintenance Governance Procedures Version 1.0

Committee Note Draft 01 Working Draft 02

1 February 2015 22:40z

## Specification URIs

### This version:

<http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/cnd01wd02/UBL-Governance-v1.0-cnd01wd02.xml> (Authoritative)  
<http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/cnd01wd02/UBL-Governance-v1.0-cnd01wd02.html>  
<http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/cnd01wd02/UBL-Governance-v1.0-cnd01wd02.pdf>

### Previous version:

N/A

### Latest version:

<http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/UBL-Governance-v1.0.html>  
<http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/UBL-Governance-v1.0.pdf>

### Technical Committee:

OASIS Universal Business Language TC

### Chairs:

Tim McGrath ([tim.mcgrath@documentengineeringservices.com](mailto:tim.mcgrath@documentengineeringservices.com)), Document Engineering Services  
G. Ken Holman ([gkholman@CraneSoftwrights.com](mailto:gkholman@CraneSoftwrights.com)), Crane Softwrights Ltd.

### Editors:

Ole Madsen ([olema@digst.dk](mailto:olema@digst.dk)), Danish Agency for Digitisation, Ministry of Finance  
G. Ken Holman ([gkholman@CraneSoftwrights.com](mailto:gkholman@CraneSoftwrights.com)), Crane Softwrights Ltd.

### Additional artefacts:

The ZIP containing the complete files of this release is found in the directory:

- <http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/cnd01wd02/>

### Related work:

This note is related to:

*Universal Business Language Version 2.1*. Edited by Jon Bosak, Tim McGrath and G. Ken Holman. 04 November 2013. OASIS Standard. <http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html>.

This is a Non-Standards Track Work Product. The patent provisions of the OASIS IPR Policy do not apply.

**Abstract:**

This document describes the governance of the process to propose, accept and incorporate changes to the OASIS Universal Business Language (UBL) 2 specifications.

**Status:**

This [Working Draft](#) (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or [approved](#) as a Committee Note Draft. The OASIS document [Approval Process](#) begins officially with a TC vote to approve a WD as a Committee Note Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Note Draft.

See [Appendix A, Release Notes](#) for more information regarding this release package.

**Citation format:**

When referencing this note the following citation format should be used:

**[UBL-Governance]** *UBL Maintenance Governance Procedures Version 1.0*. 1 February 2015 22:40z.  
OASIS Committee Note Draft 01 Working Draft 02. <http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/cnd01wd02/UBL-Governance-v1.0.html>.

---

## Notices

Copyright © OASIS Open 2015. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the “OASIS IPR Policy”). The full [Policy](#) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

---

## Table of Contents

1 Introduction .....	5
1.1 Exclusions .....	6
1.2 OASIS Technical Committee Process .....	6
2 UBL governance .....	7
2.1 Committee participants .....	7
2.2 Model collaboration environment .....	7
2.3 Considering proposals to the committee .....	8
2.4 Creating working drafts .....	8
3 Proposing additions to the committee .....	9
3.1 Important terminology and concepts .....	9
3.1.1 Business Information Entity or component types .....	9
3.1.2 Namespaces .....	11
3.1.3 Cardinality .....	14
3.1.4 Text-based components .....	15
3.2 Contributions of user proposals .....	15
3.2.1 Proposing new document types .....	16
3.2.2 Proposing new business objects .....	16
3.3 Demonstrating contributions using schemas .....	16
3.3.1 Using extensions to remain compatible .....	17
3.3.2 Using incompatible demonstration schemas .....	17
3.4 Submitting proposals to the committee .....	17
4 Release cycle .....	18

## Appendixes

A Release Notes .....	21
A.1 Availability .....	21
A.2 Status of this release .....	21
A.3 Package structure .....	21
B Acknowledgements (Non-Normative) .....	22
C Revision History (Non-Normative) .....	23

# 1 Introduction

To many groups around the world the OASIS Universal Business Language (UBL) Standard is a very important and useful XML-based specification of business documents. The more that the standard is adopted, the more important it becomes and the more influence it wields in new areas of business not yet addressed within its current scope. As this specification comes to the attention of new users, or as existing users identify new requirements, the more critical it becomes to streamline the process by which the specification is improved for all.

The OASIS Universal Business Language Technical Committee has long recognized that there are groups using the UBL Standard who have business requirements that are not met by UBL 2.0 or UBL 2.1. New and existing users with identified requirements come to UBL looking for a way to accommodate their needs for expressing their information in the document structures. But the UBL defined by the committee can never be all things to all people, and so from day one the committee created the UBL document extension point as a technical mechanism provided to accommodate this.

Although the extension point provides a technical solution for custom requirements to be met within the standard and still create XML-schema-valid UBL documents, some business requirements are common amongst many users. Interoperability is inhibited if each user uses the extension point syntax differently to express the same semantics. Interoperability is enhanced when UBL grows with new documents and additional components to accommodate more semantics that are shared amongst many users.

But the UBL specification itself is already very large because of the wide-ranging scope of solutions where it plays a role. UBL 2.1 has over 4,000 business objects standardized in support of 65 different document types. Those new to UBL may be unable to find an existing component to support their requirement, or they may create their own extension component unknowingly duplicating an existing component. UBL's size is both a benefit in its coverage and a detriment in its magnitude.

Nevertheless, UBL has been in production for many years and has been tested and deployed in many organizations. As an example, in Denmark UBL has been in production and required by law on a regional and national level for more than 10 years. For all suppliers to the public sector UBL is mandated for all goods, services and utilities. And yet new developers in Denmark are still addressing what they think are new ideas, when in fact many such ideas are already available in UBL, though perhaps in a slightly different manner than how the developers have conceived.

What is needed is direction for the existing and growing numbers of UBL developers, all with the same goal of using UBL as their business language for their solutions today and in the near future, to contribute to the technical committee their ideas of what may be missing in UBL. They need to know how to contribute to the future of UBL and yet be able to use their ideas immediately within the constraints of UBL within their own production environment.

There has always existed the proper public comment channel with which the OASIS TC process provides a means for outside input, requests and suggestions to be submitted to the committee. This remains true and must always be the means by which non-committee-member contributions are made. However, without guidance it may be unclear to some how best to use the comment channel to put forward proposals of additional documents or additional library business objects for consideration by the committee.

This document details the governance of UBL: the processes of development, maintenance, support and monitoring the specification. For example, what time lines are agreed within the committee for regularly-scheduled minor revisions?

This document also includes a description of the process by which the general community can propose tested additions for the revision to UBL. For examples, what is to be delivered to the committee, how it is to be delivered, how it is made visible as part of the process and how it gets incorporated if accepted. Though it should be remembered that even with these details in place, the UBL technical committee has no obligation to accept and incorporate any proposed additions to the UBL specification.

Finally, this document overviews the anticipated release cycle and target dates for each phase of the OASIS TC process to reach conclusion. This is included to manage the expectations of candidate contributors and guide them in the timeliness of their submissions. Readers are reminded that the more time the UBL technical committee has to review a submission before the committee's established deadlines, the more chance the submission will get full consideration. Any submission received that does not get full consideration may not be incorporated in the next release of UBL and would then be delayed for consideration in a subsequent release. The UBL technical committee accepts no responsibility for considering any particular proposal in any requested release.

## 1.1 Exclusions

The UBL Technical Committee cannot provide a schema generation service to those creating their own document structures or proposing new additions to UBL. Interested parties seeking assistance could use publicly-available software or consult with companies who can provide such a service.

### Note

The UBL community web site has listings of candidate products and services at, respectively, <http://ubl.xml.org/products> and <http://ubl.xml.org/services>.

## 1.2 OASIS Technical Committee Process

The OASIS UBL Technical Committee itself is governed by the OASIS Technical Committee (TC) Process (referred to in this document as the "Process") found at <https://www.oasis-open.org/policies-guidelines/tc-process>. At the time of writing, the items in this Committee Note conform to the Process, but the life cycle of the Process may produce revisions that will need to be accommodated in a revised version of this Committee Note.



## 2 UBL governance

The UBL committee is ultimately responsible for the content of the UBL Specification. The committee has open membership to OASIS members in good standing. Committee members are allowed to actively participate in any of the subcommittees and tasks groups.

### Note

Please see <https://www.oasis-open.org/join> for information on becoming an OASIS member.

Voting on the technical committee is possible only by those committee members who are qualified for voting status at the time of the vote, as described in the Process.

### 2.1 Committee participants

All committee members are welcome to put forward for consideration their suggestions for additions to UBL.

The various roles a committee member may wish to assume responsibility for include:

- subcommittee participants - contributing specifically in an area of expertise focused on by a subcommittee, such as procurement or transportation or some other new subcommittee that may be created, responsible for analyzing contributions and making recommendations to the technical committee
- model editors - responsible for editing the information describing the agreed-upon additions and changes to the document model
- specification editors - responsible for editing the prose content and the packaging of the agreed-upon additions and changes to the specification
- task group participants - contributing time and expertise to the horizontal tasks such as generating artefacts for use by the editors

All committee members are urged to participate in reviewing work products and voting members are compelled to vote on all motions.

### Note

At the time of writing there are two teleconferences that make up a single weekly meeting of UBL technical committee members. This is for the convenience of members from around the world to be able to participate in active discussion. Please see the committee home page for the current meeting details at:

[https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ubl#meetings](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl#meetings)

### 2.2 Model collaboration environment

A Google spreadsheet, termed the "master spreadsheet", contains the latest state of all document types and business objects in UBL. Designated model editors are directed by the technical committee to incorporate subcommittee recommendations in the models by editing the master spreadsheet. The spreadsheet location, instructions and colour legend are found in the summary document at:

<https://docs.google.com/document/d/1RSvewwdS-lp4sl-gYHI5gjNxrgXXqCc5wuA6UI5hh-WA/view>.

Each of the subcommittees can use their own means, including their own copy of the Google spreadsheet, in order to draft their recommendations to the Technical Committee.

The Schema Generation Task Group (SGTG) is available to create schemas for the subcommittee from a Google spreadsheet should the subcommittee not wish to create the schemas themselves. The tools used by the SGTG are publicly available and can be used by the subcommittees directly if they wish.



## 2.3 Considering proposals to the committee

Members of the technical committee propose their own additions to the subcommittees for consideration.

Per the Process, the only proposals from the community that are acceptable for consideration by the subcommittees are those received through the public comment list.

Every proposal, internal or from the community, is tracked in a JIRA ticket that is assigned to a designated subcommittee for disposition. The JIRA ticket has the designated component "Documents and business objects". See [Section 4, "Release cycle"](#) for the link to view the JIRA tickets.

The designated subcommittee uses their part of the collaboration environment to establish the disposition of each of the tickets they are assigned.

## 2.4 Creating working drafts

The subcommittees forward to the technical committee their finalized dispositions of tickets and their recommendations of additions for consideration in including in the next release of UBL. The technical committee agrees on the recommendations and then:

- directs the designated model editors to incorporate the recommendations in the master spreadsheet; and
- directs the designated specification editors to incorporate the recommendations in the documentation.

The subcommittees review the changes made by the designated model editors and deem that the collaboration spreadsheet is ready to go to SGTG to create the schemas.

The designated specification editors incorporate PSC and TSC descriptions and SGTG schemas into a complete working draft. Working drafts are stored in the committee's Kagi repository of documents as ZIP files at:

[https://www.oasis-open.org/committees/documents.php?wg\\_abbrev=ubl](https://www.oasis-open.org/committees/documents.php?wg_abbrev=ubl)

The subcommittees review the changes made by the designated specification editors and deem from their perspective that the working draft package is ready to go towards standardization. The technical committee then proceeds with the Process that governs the steps that follow. See [Section 4, "Release cycle"](#) for the handling of working drafts towards the completed standard.



## 3 Proposing additions to the committee

### 3.1 Important terminology and concepts

#### 3.1.1 Business Information Entity or component types

XML documents are tree hierarchies of elements expressing content. Each element in the document markup represents a component found in the document model. Each element/component in UBL is comprised of either only other elements/components or only text. In UBL there is no use of the XML concept of "mixed content", that is, the mixing of text and elements (such as found in a paragraph with embedded element markup) as children of a single parent element.

A submitter asking to add new XML elements in a UBL document has the responsibility to describe in detail all of the corresponding new components to be added to the UBL model. This is described in [Section 3.2, "Contributions of user proposals"](#). Such prose details include the context of use and objectives of each component. Such spreadsheet details include the component's type, name, description, examples, and any alternative business terms that are considered equivalent.

The types of the components, or business information entities (BIE), used in UBL have well-defined roles and have abbreviated names for reference. The basic terminology and concepts behind the components is from the ISO/TS 15000-5:2005 Core Component Technical Specification found in the OASIS archive at:

<https://www.oasis-open.org/committees/download.php/6232/CEFACT-CCTS-Version-2pt01.zip>

The basic types of component are as follows:

- ABIE - Aggregate Business Information Entity
  - an aggregate entity represents a class of detailed information that is comprised only of basic entities and association entities, that is, it describes an element that is comprised only of other elements
  - in UBL all of the basic entities in an aggregate entity are positioned within the aggregate entity before all of the aggregate's association entities
  - inside the UBL model there are two kinds of aggregate entity, though these are distinguished only by how they are used and they are not distinguished as different component types in the document models:
    - Library ABIE - Library Aggregate Business Information Entity
      - a library aggregate exists only to define the contents of an association entity referencing it from elsewhere in the model
      - a library aggregate not fulfilling the role of describing an association entity referencing it from elsewhere has no purpose and is removed from the library in order to prevent the library from growing unnecessarily large for no useful purpose
      - in an XML tree perspective, a Library ABIE is the general shape of an association branch that may be found in multiple places in the tree, but the Library ABIE is not, itself, a branch
      - the XML element name used for the association branch may be, but is not required to be, the same name used for the Library ABIE branch shape
    - Document ABIE - Document Aggregate Business Information Entity

- a document aggregate exists to define a document type, that is, the top-level elements or the beginning of a UBL document
- at the time of writing, no UBL document aggregate is referenced by any association entity (though unexpected, this may change in the future if needs require it)
- in an XML tree perspective, a Document ABIE is the trunk of the document tree
- the XML document element always uses the Document ABIE name as the XML element name
- so far in UBL a Document ABIE shape is not found elsewhere in the tree as an association branch
- outside the UBL model there is a third kind of aggregate entity, used in the definition of extensions for UBL documents
  - UBL Common Extension ABIE
    - in all document models there are two reserved ABIEs within which user extensions are placed
- BBIE - Basic Business Information Entity
  - a basic entity represents a single and indivisible piece of textual information in a single component of the class, that is, it is an element that is comprised only of text
  - all of the text content of a UBL XML document is found only in the elements of BBIE components
  - a basic entity may have imposed on it a strict lexical structure in the text string, such as a date format where the string "2015-01-01" represents New Year's Day in 2015 independent of any location, whereas the string "2015-01-01+05:00" represents New Year's Day in 2015 in the Eastern Time Zone; a text value such as "January 1, 2015" would violate the ISO 8601 lexical constraints of a component declared to be of type "Date" and so would not be schema-valid when validated
  - a basic entity may have imposed on it a value constraint such as one expressed in a code list that applies in a particular business context; the UBL schemas do not impose value constraints; see the discussion on two-phase validation in the UBL 2.1 specification at <http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html#A-UBL-2.1-CODE-LISTS-AND-TWO-PHASE-VALIDATION> for more information regarding value constraints
  - in an XML tree perspective, a BBIE is a leaf in the tree, attached to either a branch or the trunk
  - outside the UBL model there is a second kind of basic entity, used in the definition of extensions for UBL documents
    - UBL Common Extension BBIE
      - there are a number of Extension BBIEs within each of which a single piece of extension metadata is placed
- ASBIE - Association Business Information Entity
  - an association entity is a placeholder representing the entire contents of another class (aggregate) as a single associated component of the class, that is, it is a child element that is comprised only of other elements as defined by the associated class
  - in an XML tree perspective, an ASBIE is a branch in the tree and the shape of that branch is defined by the associated ABIE

- the XML element name used for the ASBIE branch may be, but is not required to be, the same name used for the associated ABIE branch shape
- outside the UBL model there is a second kind of association entity, used in the definition of extensions for UBL documents
  - UBL Common Extension ASBIE
    - there is a single extension ASBIE within which the collection of Extension BBIEs and Library BBIEs comprise the set of extension metadata and the extension content

There are also BIEs used to define UBL extensions that are modeled using CCTS. Extensions are defined by users to be available to add their own information to the UBL model while not violating the model's constraints. The UBL committee also defines extensions that are available to users to add across all UBL documents. The first of the extensions defined by the UBL committee is the digital signature extension comprised of the following types of BIE:

- Signature Apex ABIE
  - analogous to a Document ABIE in that it is the tree trunk of the signature extension structure
- Signature Library ABIE
  - analogous to a Library ABIE in that it the shape of a tree branch identified by a Signature ASBIE as having signature information
  - an extension ABIE may reference standard UBL model ABIEs (through ASBIEs) and BBIEs without the need to redeclare the components as extension components
- Signature BBIE
  - analogous to a BBIE in that it is a leaf of signature information in the signature tree

There is no obligation for UBL users to use CCTS to model extensions. However, if a user submits to the UBL committee the content of their extensions for the committee's consideration as additions to the UBL model, it should likewise be modeled using CCTS. Accordingly, a user's extension would also have the following types of BBIE:

- User Extension Apex ABIE
  - analogous to a Document ABIE in that it is the tree trunk of the user's extension structure
- User Extension Library ABIE
  - analogous to a Library ABIE in that it the shape of a tree branch identified by a user extension ASBIE as having extension information
  - an extension ABIE may reference standard UBL model ABIEs (through ASBIEs) and BBIEs without the need to redeclare the components as extension components
- User Extension BBIE
  - analogous to a BBIE in that it is a leaf of extension information in the user extension tree

### 3.1.2 Namespaces

Namespaces are critically important in XML, and thus are critically important in UBL. Unfortunately, the simple concept embodied by XML namespaces is often misunderstood, leading to confusion with users. Regrettably, the verbose nature of UBL namespaces contributes to the mistaken impression that the namespaces somehow are adding to the complexity of the XML in UBL documents.

## Note

This governance document does not attempt to relate the technical declaration syntax for XML namespaces. The focus of this section of governance is to guide the reader regarding the selection of XML namespaces used in UBL and in user contributions to the committee.

The sole role of namespaces in any XML document, including UBL documents, is to distinguish element names found in separate collections of element names where the element names otherwise would be ambiguous because they have the same spelling. A simple example in UBL is the comparison of the ABIE named "Location" and the BBIE named "Location". These are different Business Information Entities and so the components have different XML element structures. The ABIE named "Location" is comprised of elements as children. The BBIE named "Location" is comprised only of text. They cannot both be expressed in a single XML document using the markup `<Location>` as neither the user nor the XML processor interpreting the markup knows which ASBIE (pointing to the ABIE) or BBIE component is being referenced. Therefore, to avoid ambiguity, the identically-named components must come from separate collections.

Users of UBL have the responsibility not to create in their XML any new non-UBL components in any of the UBL collections. Unfortunately it is easy to break this rule and so users of UBL need to be diligent not to break this rule. This obligates users to use unique user-defined collections for user-defined components, and to re-use standard UBL-defined collections for standard UBL-defined components.

The name of a collection of element names is the XML namespace of those names. A namespace is identified using a Uniform Resource Identifier (URI). This may be an absolute Uniform Resource Locator (URL) typically used as an Internet address, but it need not be. An XML namespace never dereferences any information at its URI, even if that URI is a URL, so there is no obligation for a URL to actually deliver any information. There is no access to the Internet when one uses a namespace URI, even if that URI is a URL. In Standard UBL each URI string is a Uniform Resource Name (URN).

Because the syntax of a URI is not allowed inside of an element's angle brackets, an abbreviated proxy name (referred to in XML as "the namespace prefix") is used to reference the namespace URI, thus identifying in which collection a given element's name is found. Adding to the earlier example, the element named `<cac:Location>` uses the string "cac:" as the abbreviated prefix to the Library ABIE collection name. Similarly, the element named `<cbc:Location>` uses the string "cbc:" as the abbreviated prefix to the BBIE collection name. Thus, the two elements `cac:Location` and `cbc:Location` are easily distinguished to human readers and to an XML processor.

The only restriction on an abbreviated prefix name is that it be structured in the same way as an XML name. Prefix names are separated from element names using the colon. It is possible, with the appropriate namespace declarations, to abbreviate the prefix for one collection at a time all the way down to the empty string without a colon, such as is used in the element `<Invoice>`. Prefix names are never standardized and should never be imposed or assumed. With the appropriate namespace URI declarations one can successfully use in a UBL document the elements named `<harry:Location>` and `<sally:Location>` to distinguish the `Location` ASBIE and BBIE. However, while the choice of prefix name is irrelevant to the XML processor, such a nonsensical example does not promote human legibility when reading the raw angle brackets. Accordingly, the UBL committee chooses consistently to use in its documentation and artefacts the same set of abbreviations for all of the namespace prefix names to UBL-standard URI collection identifiers. The committee also chooses consistently to use the empty prefix for XML document elements (the top-most element or very first element that surrounds all information; the one corresponding to the Document ABIE).

In the UBL model all Library ABIEs belong in a single collection. Since ASBIEs are placeholders for Library ABIEs, the ASBIEs are also in the ABIE collection. The name of this collection is the namespace URI:

`urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2`

Abbreviated in UBL documentation using `cac:` as in `<cac:BuyerCustomerParty>`.

In the UBL model each Document ABIE belongs to a separate collection of only one element, that being itself. The name for each collection incorporates in the namespace URI the Document ABIE name as follows:

`urn:oasis:names:specification:ubl:schema:xsd:{Document-ABIE-Name}-2`

`e . g .`  
`urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2`

Abbreviated in UBL documentation using the empty prefix as in  
<ApplicationResponse>.

In the UBL model all BBIEs belong in a single collection. The name of this collection is the namespace URI:

`urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2`

Abbreviated in UBL documentation using `cbc:` as in <cbc:IssueDate>.

Outside the UBL model all common extension ABIEs, ASBIEs and BBIEs belong in a single collection. The name of this collection is the namespace URI:

`urn:oasis:names:specification:ubl:schema:xsd:CommonExtensionComponents-2`

Abbreviated in UBL documentation using `ext:` as in <ext:UBLExtensions>.

The Signature Apex ABIE, Library ABIE and BBIE collections have, respectively, the following namespace URI (only one URI for each collection):

`urn:oasis:names:specification:ubl:schema:xsd:CommonSignatureComponents-2`

`urn:oasis:names:specification:ubl:schema:xsd:SignatureAggregateComponents-2`

`urn:oasis:names:specification:ubl:schema:xsd:SignatureBasicComponents-2`

Abbreviated in UBL documentation, respectively using `sig:`, `sac:` and `sbc:`

Again, it cannot be underscored enough that users creating their own semantics and elements must not use the UBL namespace URIs as the identifiers of those collections. And if they are using UBL-standard semantics and elements they must use the UBL namespace URIs to identify the UBL-standard collections. If a UBL namespace URI is used to identify a user component that is not identical to the component defined by the UBL Technical Committee, then interoperability is broken (and, often, silently broken until systems fall apart).

XML namespace URIs should be considered as organizational assets, and as such, they should be managed accordingly. Care should be taken when defining a URI string for a collection that that string has not already been used for a different collection elsewhere in the organization's XML corpus.

URL strings have a strict (but unenforced) syntax of an absolute URL that should use a domain owned by the user. URN strings have a strict (but unenforced) syntax of allowed values in the second field, called the "URN Namespace" (for example, OASIS has formally registered "oasis" as a URN namespace and so is allowed to use it). There is a back door in the URN scheme that allows anyone to use what is termed an "experimental" URN namespace. As such there is no attempt to assert the namespace is anything other than something used temporarily or for experimental purposes.

Thus, if a user is not managing namespace URIs as organizational assets, it would be entirely appropriate to use the experimental URN namespace for the User Extension Apex ABIE, User Document ABIE, User Library ABIE and User BBIE collections along the lines as follows (only one URI for each collection):

`urn:X-MyCompany:Extension`

urn:X-MyCompany:Document

urn:X-MyCompany:Aggregate

urn:X-MyCompany:Basic

Abbreviated, respectively, using prefix names along the lines of `myx:`, `myd:`, `mya:` and `myb:`

An example of a managed set of namespaces would be to use URLs to the organizational web site. This ensures no other organization following proper etiquette would create an ambiguously-identical namespace URI string, whereas two organizations could very well create identical experimental URN strings. Using a URL for the URI string provides an added benefit that the URL could deliver information to a browser should anyone attempt to dereference the URL. A common practice is to create an XHTML page with RDDL attributes that guides the reader to information about the XML namespace. As mentioned before, an XML processor uses the namespace URI as just a string, so the fact that the string happens also to be a URL does not require the URL to actually exist. Examples of URLs to use as namespace URIs are:

`http://myCompany.com/ns/ubl/Extension`

`http://myCompany.com/ns/ubl/Document`

`http://myCompany.com/ns/ubl/Aggregate`

`http://myCompany.com/ns/ubl/Basic`

Abbreviated, respectively, using prefix names along the lines of `mx:`, `md:`, `ma:` and `mb:`

### 3.1.3 Cardinality

The cardinality of a component is the number of times the component is allowed to exist at the particular point in the document tree, between the immediately-preceding component of another name and the immediately-following component of another name. The cardinality applies only to BBIEs and ASBIEs as these are the components realized in XML syntax as leaves and branches of the tree. An ABIE describes only the shape of a tree branch, and it is not, itself, a branch in any tree.

There are four values for cardinality:

- "0..1" - the component is optional and cannot be repeated if it exists
- "1" - the component is mandatory and cannot be repeated
- "0..n" - the component is optional and if it exists it can be repeated with immediate like-named component siblings
- "1..n" - the component is mandatory and can be repeated with immediate like-named component siblings

In order to preserve backward compatibility with all previous versions of UBL 2.x, the cardinality of all proposed new or modified components is strictly managed. Backward compatibility is the nature that, for example, every conceivable schema-valid instance of UBL 2.0 is also a schema-valid instance of UBL 2.1 and all subsequent minor revisions of UBL. Similarly every schema-valid instance of UBL 2.1 shall also be a schema-valid instance of UBL 2.2 and all subsequent minor revisions of UBL.

The following scenarios are anticipated to preserve backward compatibility:

- an existing ABIE can have only optional BBIE and/or ASBIE components added to its children

- a new ABIE can have any new or existing BBIE and/or ASBIE children, each with any cardinality, including being mandatory
- an existing BBIE or ASBIE can have its minimum cardinality decreased (which rarely happens because the absence of a component is not meant to convey information) or its maximum cardinality increased (more likely); all of the possible changes are as follows:
  - from "0..1" to "0..n"
  - from "1" to "1..n" or "0..1" or "0..n"
  - from "1..n" to "0..n"

Guaranteeing forward compatibility is not possible with UBL. Because of the added components, schema validity is broken in prior versions of UBL. For example, a UBL 2.1 instance is not guaranteed to be a schema-valid instance of UBL 2.0 because the user may have used in the document a UBL 2.1 component not defined in UBL 2.0. However it is true that a user of UBL 2.1 choosing only to use UBL 2.0 components in the UBL 2.1 document will have an instance that is schema-valid with the UBL 2.0 schemas.

The UBL 2 Customization specification at <http://docs.oasis-open.org/ubl/guidelines/UBL2-Customization1.0cs01.pdf> makes reference to potential forward-compatible processing by pruning from a document those components that a system does not recognize. Thus while the original UBL 2.1 document may have components foreign to UBL 2.0, a pruned version of that UBL 2.1 document could then be schema-valid to both UBL 2.0 and UBL 2.1. One caveat being that pruning the document might lose critical information intended to be sent by the sender. The benefit, however, being that the receiving system is allowed to process at least what it knows from the source document. Another caveat is that a change in cardinality that decreases the minimum value will create an instance where dummy information must be added to the massaged document to make it schema-valid to the earlier schema that required the component. This is another reason to deprecate lowering the minimum cardinality of a component in a revision of the model.


### 3.1.4 Text-based components

In UBL there is no concept of "paragraphs". A single BBIE of data type Text or Name is a continuous string of text from the start tag of the element to the end tag of the element.

The entry of a text or name semantic that is not language-based, such a serial number or street name, would typically not have a repeatable cardinality.

The entry of a text or name semantic that is language-based, such as a description, would typically have the repeatable cardinality. This is not provided for the purposes of expressing paragraphs. Each instance of such a repeatable text field should have a different language identifier, thus representing different translations of the semantic's value.

## 3.2 Contributions of user proposals

A submission to the committee that is incomplete or is only a contribution of a statement along the lines of "UBL should contain a..." will receive less attention and consideration than a submission filed complete with the information the committee needs to consider and implement the suggested components. 

Those contributors working with CCTS will find the UBL 2.1 model, the UBL 2 digital signature extension model and a prototypical empty model all available as spreadsheets to be copied as proforma models with which to create one's own models. These resources are linked from the help document describing the colour conventions and instructions:

<https://docs.google.com/document/d/1RSvewwdS-lp4sl-gYHI5gjNxrgXXqCc5wuA6UI5hh-WA/view>

For guidance on the many CCTS fields in the model columns of the spreadsheet, please review the latest version of the UBL Naming and Design Rules Version 3.0 found at:

[https://www.oasis-open.org/committees/document.php?document\\_id=54964](https://www.oasis-open.org/committees/document.php?document_id=54964)

### 3.2.1 Proposing new document types

A proposal that the committee include in UBL a new suite of document types should include the following details:

- the name of the collection of document types expressing the suite of semantics as a whole
  - an example in UBL 2.1 is the addition of the collection of Collaborative, Planning, Forecasting and Replenishment documents described in <http://docs.oasis-open.org/ubl/os-UBL-2.1/UBL-2.1.html#S-COLLABORATIVE-PLANNING-FORECASTING-AND-REPLENISHMENT>
  - this package of document types was incorporated by a voting committee member at the time, providing all of the information needed to seamlessly include the documents in the UBL 2.1 document set
- a thorough description of the context of the use of the document types in enough detail to describe the objectives of use of the document suite and the roles of the players involved in their exchange
- a set of detailed swim-lane diagrams illustrating the choreography of the exchange of instances of the document types
- a spreadsheet of the proposed Document ABIEs and any required new Library ABIEs and BBIEs
- schemas generating using the contributor's namespace URIs for all components (no use of Standard-UBL namespace URIs)
- example instances conforming to the proposed document model schemas but using the contributor's namespace URIs

### 3.2.2 Proposing new business objects

A proposal that the committee include in UBL new business objects in existing ABIEs should include the following details:

- a thorough description of the context of the use of the new information augmenting existing components
- a spreadsheet of the proposed new business objects showing in each row the object class to which the proposed object is to be added
  - such a spreadsheet does not have an ABIE row for the miscellany of business objects that are being added to existing ABIEs
  - such a spreadsheet does have an ABIE row for new ABIEs that are needed to satisfy new ASBIEs, plus all of the business objects for that new ABIE

## 3.3 Demonstrating contributions using schemas

### Note



It is out of the scope of this governance document to tutor the reader on the configuration and execution of publicly-available tools that may be found listed in the web sites cited in [Section 1.1](#), "Exclusions".



### 3.3.1 Using extensions to remain compatible

If the proposer wishes instances of their demonstration schemas to remain compatible with UBL documents, then the proposed components are obliged to be placed in an extension. This is only for the proposal as the committee will consider incorporating the proposed components in the master UBL model.


Copying the UBL Signature Extension spreadsheet as an example, and hand-crafting schemas or using publicly-available tools to generate the schemas, the extension model is presented in XML using three custom non-Standard-UBL namespace URI strings. One string is for the extension apex, one is for the non-Standard-UBL aggregates and association components, and one is for the non-Standard-UBL basic components. The Standard-UBL namespace URI strings are used for those UBL components that already exist in the Standard-UBL model and schemas.

### 3.3.2 Using incompatible demonstration schemas

If the proposer wishes instances of their demonstration schemas and instances to mimic the desired end result of the committee incorporating their additions, then the proposer is obliged to use non-Standard-UBL namespace URI strings for all components. While being incompatible at the time of the submission, it would be true that if all of the proposers changes are adopted then changing only the namespace URIs in the demonstration instances to be the Standard-UBL namespace URIs would create UBL-schema-valid instances.


To accomplish this one copies the UBL spreadsheet and edits the new spreadsheet with the changes. Using publicly-available tools to generate the schemas, the tool configuration must include non-Standard-UBL namespace URI strings so as to create schemas that cannot be mistaken ambiguously to be Standard-UBL schemas.

## 3.4 Submitting proposals to the committee

A non-committee-member proposer zips up all of the contribution into a single submission, subscribes to the UBL public comment list and then posts the submission to that list.  The mail list is not meant for general discussion purposes, the committee is obliged to solicit answers from the submitter to questions about the submission through the same list so as to treat the answers as additional information included with the submission.

All contributions must be submitted to the UBL public comment list under the terms of the OASIS Feedback License, which assures potential implementers and OASIS that the contribution may be used safely by the Committee in a way that is consistent with its charter and the licensing terms set by its IPR Mode and the OASIS IPR Policy. Please see the feedback page for detailed links:

[https://www.oasis-open.org/committees/comments/index.php?wg\\_abbrev=ubl](https://www.oasis-open.org/committees/comments/index.php?wg_abbrev=ubl)

All contributions to the UBL public comment list, be they questions or submissions of any kind, are publicly-archived and individual posts may not be removed from the archive: 

<https://lists.oasis-open.org/archives/ubl-comment/>

## 4 Release cycle

The UBL 2.1 specification was released in late 2013. The committee targets releasing subsequent revisions every three years, thus, at the time of writing, the next release is targeted for late 2016.

Per the Process, the steps are summarized as follows (please see the Process for the latest detail on these steps) in order to document likely dates during the release year of completion of each of the stages:

1. working drafts are developed on an arbitrary as-needed basis using documentation from the editors and schemas from the Schema Generation Task Group

### Note

The working drafts are prepared and discussed until about February of the release year.

2. after a meeting where it is decided that a working draft is stable and ready to proceed, no more additions will be accepted for the given release cycle
3. the editors prepare both a Committee Specification Draft (CSD) 1 package and a Public Review Draft (PRD) 1 package, both dated for the following meeting, based on the working draft with changes only to the cover page and any other non-material edits appropriate to the package (such as in the release notes and revision history annexes)
4. at the following meeting the committee approves the following two motions each with a full majority vote:
  - to approve the specific working draft as the given Committee Specification Draft (CSD) package
  - to approve the given Public Review Draft package for a public review period of 30 days
5. input from outside of the committee is accepted only through the public comment facility linked from the committee home page
6. every comment received either from within the committee or from the public comment facility is tracked in JIRA as a separate ticket

### Note

The UBL JIRA tickets related to new documents and business objects are added as related to the component named "Documents and business objects" and can be viewed at <https://issues.oasis-open.org/issues/?filter=11670&jql=project%20%3D%20UBL%20AND%20component%20%3D%20%22Documents%20and%20business%20objects%22>

All of the UBL JIRA tickets can be viewed at <https://issues.oasis-open.org/issues/?filter=11670>.

In both cases the view can be changed using the box icon at top right; the list view provides the most summary information, without the detail view obscuring the screen space.

7. at the end of the public review the editors create a proposed Disposition of Comments in which every comment received is handled as required

### Note

The end of the first public review is about April of the release year

8. any substantive changes that impact on validation and conformance will trigger creating new working drafts, a CSD2 package and PRD2 package for a public review period of 15 days (if only

fixing issues that were added) or 30 days (if substantial additional content had to be added on an exceptional basis)

9. the previous step repeats until all comments received are accommodated with only non-material changes that impact on validation and conformance, after which no more changes will be accepted for the given release cycle unless testing or review reveal a backward compatibility or other serious issue requiring a restart of the process

## Note

Assuming a second public review but not a third, the reviews end about May of the release year

10. committee members begin soliciting OASIS member companies to express interest in submitting a "Statement of Use" of the revised specification
11. the editors prepare Committee Specification 1, dated for the meeting following the point 7 days after the end of the last public review, based on the final Committee Specification Draft with changes only to the cover page and any other non-material edits appropriate to the package (such as in the release notes annex and the act of removing the revision history annex)
12. at the meeting following the point 7 days after the last public review, the committee approves the following two motions each with a full majority vote:
  - to approve the proposed Disposition of Comments as finalized
  - to approve requesting TC Administration hold a Special Majority Vote for the advancement of the Committee Specification Draft as the prepared Committee Specification 1

## Note

With no inordinate delays the committee specification is ready about the end of June of the release year

13. the editors prepare Candidate OASIS Standard 1, dated to correspond with the end of a Special Majority Vote scheduled in negotiation with OASIS TC Administration
14. at a meeting following the approval of the Committee Specification, the committee approves the following motion with a full majority vote:
  - to approve requesting OASIS TC Administration hold a Special Majority Vote to submit the Committee Specification as the given Candidate OASIS Standard
15. after the ballot is completed the editors prepare a complete submission package as described in the Process, including the received Statements of Use
16. OASIS TC Administration reviews the package for at least 15 days and then conducts a 60-day public review

## Note

With no inordinate delays the public review ends about mid-October of the release year

17. if changes are required the entire process restarts back at the Committee Specification Draft stage towards Committee Specification 2
18. if no changes are required then the committee chair requests (a full majority vote is not required) OASIS TC Administration hold a Special Majority Ballot for the TC to approve continuing with the OASIS Standard ballot

19. OASIS TC Administration hold a 14-day ballot for all OASIS Members to approve the proposed OASIS Standard
20. the editors prepare the putative OASIS Standard deliverable, dated to correspond with the end of the OASIS Member ballot

### **Note**

With no inordinate delays the member ballot ends early- to mid-November of the release year

21. at a meeting following the approval of the OASIS Standard the committee approves the following motion with a full majority vote:
  - to approve requesting OASIS TC Administration hold a Special Majority Vote of committee to request OASIS TC Administration submit the new UBL standard to JTC 1 as an amendment to ISO/IEC 19845
22. the editors prepare a publication of the new UBL standard in the format conforming to ISO/IEC Directives (this may happen periodically through the process so as not to have any surprises after the Standard is completed)

### **Note**

Only the first version of any submission is allowed to be published as an ISO/IEC Standard in a format that does not conform to ISO/IEC Directives. All subsequent submissions must conform.

23. after successful conclusion of OASIS TC Administration conducting a 4-week Member Review for member feedback regarding the submission request, OASIS considers the request and decides to forward the Standard as an amendment

### **Note**

With no inordinate delays the submission should happen about mid-January of the year following the release year

24. the process dictated by ISO/IEC Directives then engages in order to bring the amended version of ISO/IEC 19845 to a new JTC 1 Standard

### **Note**

Readers may have noted that the UBL specification phase document numbers are single digits rather than the two digits used in UBL committee notes and other work products created by the committee and requested by OASIS TC Administration. Historically for a few projects dated before the requirement was established only single digits were used and only for this remaining one project will single digits continue to be used for continuity. Other work products now created by the committee are using two digits as required for the phase document number.

## Appendix A Release Notes

### A.1 Availability

Online and downloadable versions of this release are available from the locations specified at the top of this document.

### A.2 Status of this release

Release of this package marks the beginning of its internal committee review and development.

**THIS RELEASE IS SUBJECT TO CHANGE. IT IS PROVIDED FOR TESTING PURPOSES ONLY AND SHOULD NOT BE USED FOR PRODUCTION SYSTEMS.**

### A.3 Package structure

This OASIS Committee Note is published as a zip archive in the <http://docs.oasis-open.org/ubl/UBL-Governance/v1.0/cnd01wd02/> directory. Unzipping this archive creates a directory tree containing a master DocBook XML file (UBL-Governance-v1.0-cnd01wd02.xml), a generated hypertext version of this file (UBL-Governance-v1.0-cnd01wd02.html), a generated PDF version of this file (UBL-Governance-v1.0-cnd01wd02.pdf), and a number of subdirectories. The files in these subdirectories contain the various components of this release. A description of each subdirectory is given below. Note that while the UBL-Governance-v1.0-cnd01wd02.xml file is the “original” of this specification, it may not be viewable in all currently available web browsers.

**db**

DocBook documentation support files

## Appendix B Acknowledgements (Non-Normative)

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Oriol Bausa Peris, Individual  
Kenneth Bengtsson, Alfa1lab  
Peter Borresen, Document Engineering Services Limited  
Jon Bosak, Individual  
Kees Duvekot, RFS Holland Holding B.V.  
Martin Forsberg, Swedish Association of Local Authorities & Regions  
G. Ken Holman, Crane Softwrights Ltd.  
Ole Madsen, Danish Agency for Digitisation, Ministry of Finance  
Tim McGrath, Document Engineering Services Limited  
Andrew Schoka, Individual

---

## Appendix C Revision History (Non-Normative)

Revision	Date	Edit- or	Changes made
cnd01wd01	16 January 2015	GKH	Initial version in XML
cnd01wd02	1 February 2015 22:40z	GKH	A complete first draft of content for internal review.