
UBL NDR 2.0 Checklist

Editors

Michael Grimley
Mavis Cournane

The following checklist contains all UBL XML naming and design rules as defined in *UBL Naming and Design Rules version 2.0*, 30 August 2006. The checklist is in alphabetical sequence as follows:

Attribute Declaration Rules (ATD)
Code List Rules (CDL)
ComplexType Definition Rules (CTD)
ComplexType Naming Rules (CTN)
Documentation Rules (DOC)
Element Declaration Rules (ELD)
Element Naming Rules (ELN)
General Naming Rules (GNR)
General Type Definition Rules (GTD)
General XML Schema Rules (GXS)
Modeling Constraints Rules (MDC)
Naming Constraints Rules (NMC)
Namespace Rules (NMS)
Root Element Declaration Rules (RED)
Schema Structure Modularity Rules (SSM)
Versioning Rules (VER)

A.1 Attribute Declaration rules	
[ATD6]	(See GXS15)
[ATD7]	(See GXS16)
[ATD8]	(See GXS17)

A.2 Code List rules	
[CDL1]	All UBL Codes MUST be part of a UBL or externally maintained Code List.
[CDL2]	The UBL Library SHOULD identify and use external standardized code lists rather than develop its own UBL-native code lists.
[CDL3]	The UBL Library MAY design and use an internal code list where an existing external code list needs to be extended, or where no suitable external code list exists.

A.3 ComplexType Definition rules	
[CTD1]	For every class identified in the UBL model, a named <code>xsd:complexType</code> MUST be defined.
[CTD2]	Every <code>ccts:ABIE xsd:complexType</code> definition content model MUST use the <code>xsd:sequence</code> element containing references to the appropriate global element declarations.
[CTD3]	Every <code>ccts:BBIEProperty xsd:complexType</code> definition content model MUST use the <code>xsd:simpleContent</code> element.
[CTD4]	Every <code>ccts:BBIEProperty xsd:complexType</code> content model <code>xsd:simpleContent</code> element MUST consist of an <code>xsd:extension</code> element.
[CTD5]	Every <code>ccts:BBIEProperty xsd:complexType</code> content model <code>xsd:base</code> attribute value MUST be the UN/CEFACT Unqualified Datatype or UBL qualified Datatype as appropriate.
[CTD6]	For every Qualified Datatype used in the UBL model, a named <code>xsd:complexType</code> or <code>xsd:simpleType</code> MUST be defined.
[CTD20]	A <code>ccts:QualifiedDataType</code> MUST be based on an unqualified data type and add some semantic and/or technical restriction to the unqualified data type.
[CTD21]	The name of a <code>ccts:QualifiedDataType</code> MUST be the name of its base <code>ccts:UnqualifiedDataType</code> with separators and spaces removed and with its qualifier term added.
[CTD22]	Every qualified datatype based on an unqualified datatype <code>xsd:complexType</code> whose supplementary components map directly to the properties of an XSD built-in data type MUST be defined as an <code>xsd:simpleType</code> MUST contain one <code>xsd:restriction</code> element MUST include an <code>xsd:base</code> attribute that defines the specific XSD built-in data type required for the content component
[CTD23]	Every qualified datatype based on an unqualified datatype <code>xsd:complexType</code> whose supplementary components do not map directly to the properties of an XSD built-in data type MUST be defined as an <code>xsd:complexType</code> MUST contain one <code>xsd:simpleContent</code> element MUST contain one <code>xsd:restriction</code> element MUST include the unqualified datatype as its <code>xsd:base</code> attribute
[CTD24]	Every qualified datatype based on an unqualified datatype <code>xsd:simpleType</code> MUST contain one <code>xsd:restriction</code> element MUST include the unqualified datatype as its <code>xsd:base</code> attribute
[CTD25]	For every <code>ccts:BBIEProperty</code> identified in the UBL model a named <code>xsd:complexType</code> must be defined.

A.4 Complex Type Naming rules	
[CTN1]	A UBL xsd:complexType name based on an ccts:AggregateBusinessInformationEntity MUST be the ccts:DictionaryEntryName with the separators removed and with the "Details" suffix replaced with "Type".
[CTN2]	A UBL xsd:complexType name based on a ccts:BasicBusinessInformationEntityProperty MUST be the ccts:DictionaryEntryName shared property term and its qualifiers and representation term of the ccts:BasicBusinessInformationEntity, with the separators removed and with the "Type" suffix appended after the representation term.
[CTN6]	A UBL xsd:complexType name based on a ccts:BasicBusinessInformationEntityProperty and with a . ccts:BasicBusinessInformationEntityRepresentationTerm of 'Text' MUST have the word "Text" removed from the end of its name.
[CTN7]	A UBL xsd:complexType name based on a ccts:BasicBusinessInformationEntityProperty and with a . ccts:BasicBusinessInformationEntityRepresentationTerm of 'Identifier' MUST have the word "Identifier" replaced by the word "ID" at the end of its name.
[CTN8]	A UBL xsd:complexType name based on a ccts:BasicBusinessInformationEntityProperty MUST remove all duplication of words that occur as a result of duplicate property terms and representation terms.

A.5 Documentation rules	
[DOC1]	The xsd:documentation element for every Datatype MUST contain a structured set of annotations in the following sequence and pattern (as defined in CCTS Section 7): DictionaryEntryName (mandatory) Version (mandatory): Definition(mandatory) RepresentationTerm (mandatory) QualifierTerm(s) (mandatory, where used) UniqueIdentifier (mandatory) Usage Rule(s) (optional) Content Component Restriction (optional)
[DOC2]	A Datatype definition MAY contain one or more Content Component Restrictions to provide additional information on the relationship between the Datatype and its corresponding Core Component Type. If used the Content Component Restrictions must contain a structured set of annotations in the following patterns: RestrictionType (mandatory): Defines the type of format restriction that

	<p>applies to the Content Component.</p> <p>RestrictionValue (mandatory): The actual value of the format restriction that applies to the Content Component.</p> <p>ExpressionType (optional): Defines the type of the regular expression of the restriction value.</p>
[DOC3]	<p>A Datatype definition MAY contain one or more Supplementary Component Restrictions to provide additional information on the relationship between the Datatype and its corresponding Core Component Type. If used the Supplementary Component Restrictions must contain a structured set of annotations in the following patterns:</p> <p>SupplementaryComponentName (mandatory): Identifies the Supplementary Component on which the restriction applies.</p> <p>RestrictionValue (mandatory, repetitive): The actual value(s) that is (are) valid for the Supplementary Component</p>
[DOC4]	<p>The xsd:documentation element for every Basic Business Information Entity MUST contain a structured set of annotations in the following patterns:</p> <p>ComponentType (mandatory): The type of component to which the object belongs. For Basic Business Information Entities this must be “BBIE”.</p> <p>DictionaryEntryName (mandatory): The official name of a Basic Business Information Entity.</p> <p>Version (optional): An indication of the evolution over time of the Basic Business Information Entity.</p> <p>Definition(mandatory): The semantic meaning of a Basic Business Information Entity.</p> <p>Cardinality(mandatory): Indication whether the Basic Business Information Entity represents a not-applicable, optional, mandatory and/or repetitive characteristic of the Aggregate Business Information Entity.</p> <p>ObjectClassQualifier (optional): The qualifier for the object class.</p> <p>ObjectClass(mandatory): The Object Class containing the Basic Business Information Entity.</p> <p>PropertyTermQualifier (optional): A qualifier is a word or words which help define and differentiate a Basic Business Information Entity.</p> <p>PropertyTerm(mandatory): Property Term represents the distinguishing characteristic or Property of the Object Class and shall occur naturally in the definition of the Basic Business Information Entity.</p> <p>RepresentationTerm (mandatory): A Representation Term describes the form in which the Basic Business Information Entity is represented.</p> <p>DataTypeQualifier (optional): semantically meaningful name that differentiates the Datatype of the Basic Business Information Entity from its underlying Core Component Type.</p> <p>DataType (mandatory): Defines the Datatype used for the Basic Business Information Entity.</p> <p>AlternativeBusinessTerms (optional): Any synonym terms under which the Basic Business Information Entity is commonly known and used in the business.</p>

	Examples (optional): Examples of possible values for the Basic Business Information Entity
[DOC5]	<p>The xsd:documentation element for every Aggregate Business Information Entity MUST contain a structured set of annotations in the following sequence and pattern:</p> <p>ComponentType (mandatory): The type of component to which the object belongs. For Aggregate Business Information Entities this must be “ABIE”.</p> <p>DictionaryEntryName (mandatory): The official name of the Aggregate Business Information Entity .</p> <p>Version (optional): An indication of the evolution over time of the Aggregate Business Information Entity.</p> <p>Definition(mandatory): The semantic meaning of the Aggregate Business Information Entity.</p> <p>ObjectClassQualifier (optional): The qualifier for the object class.</p> <p>ObjectClass(mandatory): The Object Class represented by the Aggregate Business Information Entity.</p> <p>AlternativeBusinessTerms (optional): Any synonym terms under which the Aggregate Business Information Entity is commonly known and used in the business.</p>
[DOC6]	<p>The xsd:documentation element for every Association Business Information Entity element declaration MUST contain a structured set of annotations in the following sequence and pattern:</p> <p>ComponentType (mandatory): The type of component to which the object belongs. For Association Business Information Entities this must be “ASBIE”.</p> <p>DictionaryEntryName (mandatory): The official name of the Association Business Information Entity.</p> <p>Version (optional): An indication of the evolution over time of the Association Business Information Entity.</p> <p>Definition(mandatory): The semantic meaning of the Association Business Information Entity.</p> <p>Cardinality(mandatory): Indication whether the Association Business Information Entity represents an optional, mandatory and/or repetitive association.</p> <p>ObjectClass(mandatory): The Object Class containing the Association Business Information Entity.</p> <p>PropertyTermQualifier (optional): A qualifier is a word or words which help define and differentiate the Association Business Information Entity.</p> <p>PropertyTerm(mandatory): Property Term represents the Aggregate Business Information Entity contained by the Association Business Information Entity.</p> <p>AssociatedObjectClassQualifier (optional): Associated Object Class Qualifiers describe the 'context' of the relationship with another ABIE. That is, it is the role the contained Aggregate Business Information Entity plays within its association with the containing Aggregate Business</p>

	<p>Information Entity.</p> <p>AssociatedObjectClass (mandatory); Associated Object Class is the Object Class at the other end of this association. It represents the Aggregate Business Information Entity contained by the Association Business Information Entity.</p>
[DOC8]	<p>The xsd:documentation element for every Supplementary Component attribute declaration MUST contain a structured set of annotations in the following sequence and pattern:</p> <p>Name (mandatory): Name in the Registry of a Supplementary Component of a Core Component Type.</p> <p>Definition (mandatory): A clear, unambiguous and complete explanation of the meaning of a Supplementary Component and its relevance for the related Core Component Type.</p> <p>Primitive type (mandatory): PrimitiveType to be used for the representation of the value of a Supplementary Component.</p> <p>Possible Value(s) (optional): one possible value of a Supplementary Component.</p>
[DOC9]	<p>The xsd:documentation element for every Supplementary Component attribute declaration containing restrictions MUST include the following additional information appended to the information required by DOC8:</p> <p>Restriction Value(s) (mandatory): The actual value(s) that is (are) valid for the Supplementary Component.</p>

A.6 Element Declaration rules	
[ELD2]	All element declarations MUST be global
[ELD3]	For every class and property identified in the UBL model, a global element bound to the corresponding xsd:complexType MUST be declared.
[ELD4]	When a ccts:ASBIE is unqualified, it is bound via reference to the global ccts:ABIE element to which it is associated.
[ELD6]	The code list xsd:import element MUST contain the namespace and schema location attributes.
[ELD7]	Empty elements MUST not be declared, except in the case of extension, where the 'UBLExtensions' element is used.
[ELD9]	(See GXS14)
[ELD11]	When a ccts:ASBIE is qualified, a new element MUST be declared and bound to the xsd:complexType of its associated ccts:ABIE.
[ELD12]	The 'UBLExtensions' element MUST be declared as the first child of the document element with xsd:minOccurs= "0".
[ELD13]	The 'UBLProfileID' element MUST be declared immediately following the 'UBLExtensions' element with xsd:minOccurs= "0".
[ELD14]	The 'UBLSubsetID' element MUST be declared immediately following the 'UBLProfileID' element with xsd:minOccurs= "0".

A.7 Element Naming rules	
[ELN1]	A UBL global element name based on a <code>ccts:ABIE</code> MUST be the same as the name of the corresponding <code>xsd:complexType</code> to which it is bound, with the word “Type” removed.
[ELN2]	A UBL global element name based on a <code>ccts:BBIEProperty</code> MUST be the same as the name of the corresponding <code>xsd:complexType</code> to which it is bound, with the word “Type” removed.
[ELN3]	A UBL global element name based on a <code>ccts:ASBIE</code> MUST be the <code>ccts:ASBIE</code> dictionary entry name property term and its qualifiers; and the object class term and qualifiers of its associated <code>ccts:ABIE</code> . All <code>ccts:DictionaryEntryName</code> separators MUST be removed..

A.8 General Naming rules	
[GNR1]	UBL XML element and type names MUST be in the English language, using the primary English spellings provided in the Oxford English Dictionary.
[GNR2]	UBL XML element and type names MUST be consistently derived from CCTS conformant dictionary entry names.
[GNR3]	UBL XML element and type names constructed from <code>ccts:DictionaryEntryNames</code> MUST NOT include periods, spaces, other separators, or characters not allowed by W3C XML 1.0 for XML names
[GNR4]	UBL XML element, and simple and complex type names MUST NOT use acronyms, abbreviations, or other word truncations, except those in the list of exceptions maintained and published by the UBL TC.
[GNR6]	The acronyms and abbreviations listed in the UBL-approved list MUST always be used in place of the word or phrase they represent.
[GNR7]	UBL XML element, and type names MUST be in singular form unless the concept itself is plural.
[GNR8]	The UpperCamelCase (UCC) convention MUST be used for naming elements and types.
[GNR10]	Acronyms and abbreviations at the beginning of an attribute name MUST appear in all lower case. All other acronym and abbreviation usage in an attribute declaration MUST appear in upper case.
[GNR11]	Acronyms and abbreviations MUST appear in all upper case for all element declarations and type definitions.

A.9 General Type Definition Rules	
[GTD1]	All types MUST be named.
[GTD2]	The predefined XML Schema type <code>xsd:anyType</code> MUST NOT be used.

A.10 General XML Schema Rules

<p>[GXS1] UBL Schema MUST conform to the following physical layout as applicable:</p> <pre> <!-- ===== XML Declaration===== --> <?xml version="1.0" encoding="UTF-8"?> <!-- ===== Schema Header ===== --> Document Name: < Document name as indicated in Section 3.6 > Generated On: < Date schema was generated > <!-- ===== Copyright Notice ===== --> "Copyright , 2001-2004 The Organization for the Advancement of Structured Information Standards (OASIS). All rights reserved. <!-- ===== xsd:schema Element With Namespaces Declarations ===== -- > xsd:schema element to include version attribute and namespace declarations in the following order: xmlns:xsd Target namespace Default namespace CommonAggregateComponents CommonBasicComponents CoreComponentTypes Unspecialized Unqualified Datatypes Specialized Qualified Datatypes Identifier Schemes Code Lists Attribute Declarations – elementFormDefault=""qualified""" attributeFormDefault=""unqualified""" Version Attribute <!-- ===== Imports ===== --> CommonAggregateComponents schema module CommonBasicComponents schema module Unspecialized Unqualified Types schema module Specialized Qualified Types schema module <!-- ===== Global Attributes ===== --> Global Attributes and Attribute Groups <!-- ===== Root Element ===== --> Root Element Declaration Root Element Type Definition <!-- ===== Element Declarations ===== --> alphabetized order <!-- ===== Type Definitions ===== --> All type definitions segregated by basic and aggregates as follows <!-- ===== Aggregate Business Information Entity Type Definitions ===== --> </pre>
--

	<p>alphabetized order of ccts:AggregateBusinessInformationEntity xsd:TypeDefinitions</p> <p><!-- =====Basic Business Information Entity Type Definitions ===== --></p> <p>alphabetized order of ccts:BasicBusinessInformationEntities</p> <p><!-- ===== Copyright Notice ===== --></p> <p>Required OASIS full copyright notice.</p>
[GXS2]	UBL MUST provide two schemas for each transaction. One normative schema shall be fully annotated. One non-normative schema shall be a run-time schema devoid of documentation..
[GXS3]	Built-in XSD Simple Types SHOULD be used wherever possible.
[GXS4]	All W3C XML Schema constructs in UBL Schema and schema modules MUST contain the following namespace declaration on the xsd schema element: <code>xmlns:xsd="http://www.w3.org/2001/XMLSchema"</code>
[GXS5]	The xsd:substitutionGroup feature MUST NOT be used.
[GXS6]	The xsd:final attribute MUST be used to control extensions where there is a desire to prohibit further extensions.
[GXS7]	xsd:notation MUST NOT be used.
[GXS8]	The xsd:all element MUST NOT be used.
[GXS9]	The xsd:choice element SHOULD NOT be used where customisation and extensibility are a concern.
[GXS11]	The xsd:union technique MUST NOT be used except for Code Lists. The xsd:union technique MAY be used for Code Lists.
[GXS12]	UBL designed schema SHOULD NOT use xsd:appinfo. If used, xsd:appinfo MUST only be used to convey non-normative information.
[GXS13]	Complex Type extension or restriction MAY be used where appropriate.
[GXS14]	The xsd:any element MUST NOT be used except within the 'ExtensionContentType' type definition, and with xsd:processContents= "skip" for non-UBL namespaces.
[GXS15]	Each xsd:schemaLocation attribute declaration MUST contain a system-resolvable URL, which at the time of release from OASIS shall be a relative URL referencing the location of the schema or schema module in the release package.
[GXS16]	The built in xsd:nillable attribute MUST NOT be used for any UBL declared element.
[GXS17]	The xsd:anyAttribute MUST NOT be used.

A.11 Modelling constraint rules	
[MDC1]	UBL Libraries and Schemas MUST only use ebXML Core Component approved ccts:CoreComponentTypes, except in the case of extension, where the 'UBL Extensions' element is used

[MDC2]	Mixed content MUST NOT be used except where contained in an <code>xsd:documentation</code> element
--------	--

A.12 Naming constraint rules

[NMC1]	Each dictionary entry name MUST define one and only one fully qualified path (FQP) for an element or attribute.
--------	---

A.13 Namespace Rules

[NMS1]	Every UBL-defined -or -used schema module, except internal schema modules, MUST have a namespace declared using the <code>xsd:targetNamespace</code> attribute.
[NMS2]	Every UBL-defined-or -used major version schema set MUST have its own unique namespace.
[NMS3]	UBL namespaces MUST only contain UBL developed schema modules.
[NMS4]	The namespace names for UBL Schemas holding committee draft status MUST be of the form: <code>urn:oasis:names:tc:ubl:schema:<subtype>:<document-id></code>
[NMS5]	The namespace names for UBL Schemas holding OASIS Standard status MUST be of the form: <code>urn:oasis:names:specification:ubl:schema:<subtype>:<document-id></code>
[NMS6]	UBL published namespaces MUST never be changed.
[NMS7]	The <code>ubl:CommonAggregateComponents</code> schema module MUST reside in its own namespace.
[NMS8]	The <code>ubl:CommonAggregateComponents</code> schema module namespace MUST be represented by the namespace prefix “cac” when referenced in other schemas.
[NMS9]	The <code>ubl:CommonBasicComponents</code> schema module MUST reside in its own namespace.
[NMS10]	The <code>UBL:CommonBasicComponents</code> schema module namespace MUST be represented by the namespace prefix “cbc” when referenced in other schemas.
[NMS15]	The <code>ubl:QualifiedDatatypes</code> schema module MUST reside in its own namespace.
[NMS16]	The <code>ubl:QualifiedDatatypes</code> schema module namespace MUST be represented by the namespace prefix “qdt” when referenced in other schemas.
[NMS17]	The <code>ccts:UnqualifiedDatatypes</code> schema module namespace MUST be represented by the token “udt” when referenced in other schemas.
[NMS18]	The <code>CommonExtensionComponents</code> schema module namespace MUST be represented by the namespace prefix ‘ext’ when referenced in other schemas.

A.14 Root element declaration rules	
[RED2]	The root element MUST be the only global element declared in document schemas.
A.15 Schema structure modularity rules	
[SSM1]	UBL Schema expressions MAY be split into multiple schema modules.
[SSM2]	A document schema in one UBL namespace that is dependent upon type definitions or element declarations defined in another namespace MUST only import the document schema from that namespace.
[SSM3]	A document schema in one UBL namespace that is dependant upon type definitions or element declarations defined in another namespace MUST NOT import internal schema modules from that namespace.
[SSM5]	UBL schema modules MUST either be treated as external schema modules or as internal schema modules of the document schema.
[SSM6]	All UBL internal schema modules MUST be in the same namespace as their corresponding document schema.
[SSM7]	Each UBL internal schema module MUST be named {ParentSchemaModuleName} {InternalSchemaModuleFunction} {schema module}
[SSM8]	A UBL schema module MAY be created for reusable components.
[SSM9]	A schema module defining all UBL Common Aggregate Components MUST be created.
[SSM10]	The UBL Common Aggregate Components schema module MUST be identified as CommonAggregateComponents in the document name within the schema header.
[SSM11]	A schema module defining all UBLCommon Basic Components MUST be created.
[SSM12]	The UBL Common Basic Components schema module MUST be identified as CommonBasicComponents in the document name within the schema header.
[SSM18]	A schema module defining all UBL Qualified Datatypes MUST be created.
[SSM19]	The UBL Qualified Datatypes schema module MUST be identified as QualifiedDatatypes in the document name in the schema header.
[SSM20]	The UBL Qualified Datatypes schema module MUST import the ccts:UnQualifiedDatatypes schema module.
SSM21	The UBL extensions schema module MUST be identified as CommonExtensionComponents in the document name within the schema header.
A.16 Standards Adherence rules	

A.17 Versioning rules	
[VER1]	Every UBL Schema and schema module major version committee draft MUST have an RFC 3121 document-id of the form <code><name>-<major>[.<revision>]</code>
[VER2]	Every UBL Schema and schema module major version OASIS Standard MUST have an RFC 3121 document-id of the form <code><name>-<major></code>
[VER3]	Every minor version release of a UBL schema or schema module committee draft MUST have an RFC 3121 document-id of the form <code><name>-<major>[.<revision>]</code>
[VER4]	Every minor version release of a UBL schema or schema module OASIS Standard MUST have an RFC 3121 document-id of the form <code><name>-<major></code>
[VER5]	For UBL Minor version changes the namespace name MUST not change
[VER6]	Every UBL Schema and schema module major version number MUST be a sequentially assigned, incremental number greater than zero.
[VER7]	Every UBL Schema and schema module minor version number MUST be a sequentially assigned, incremental non-negative integer.
[VER10]	UBL Schema and schema module minor version changes MUST not break semantic compatibility with prior versions.
[VER11]	Every UBL Schema and schema module major version committee draft MUST capture its version number in the <code>xsd:version</code> attribute of the <code>xsd:schema</code> element in the form <code><major>.0[.<revision>]</code>
[VER12]	Every UBL Schema and schema module major version OASIS Standard MUST capture its version number in the <code>xsd:version</code> attribute of the <code>xsd:schema</code> element in the form <code><major>.0</code>
[VER13]	Every minor version release of a UBL schema or schema module committee draft MUST capture its version information in the <code>xsd:version</code> attribute in the form <code><major>. <non-zero>[.<revision>]</code>
[VER14]	Every minor version release of a UBL schema or schema module OASIS Standard MUST capture its version information in the <code>xsd:version</code> attribute in the form <code><major>. <non-zero></code>
[VER15]	Every UBL document schema MUST declare an optional element named "UBLVersionID" immediately following the optional 'UBL Extensions' element.

Notices

Copyright © OASIS Open 2006. 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.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.