

UBL Naming and Design Rules Checklist

This document is a subset of the UBL Naming and Design Rules Master Document. It reflects the rules used to create the UBL 1.0 Final Release schemas. The only appendix included in this checklist is referenced in Rules GNR4 and 6 and is Appendix B. You will find this at the end of this document.

Rules as of: 19 April 2004

Attribute Declaration Rules	
[ATD1]	User defined attributes SHOULD NOT be used. When used, user defined attributes MUST only convey CCT:SupplementaryComponent information.
[ATD2]	The CCT:SupplementaryComponents for the ID CCT:CoreComponent MUST be declared in the following order: Identifier. Content Identification Scheme. Identifier Identification Scheme. Name. Text Identification Scheme. Agency. Identifier Identification Scheme. Agency Name. Text Identification Scheme. Version. Identifier Identification Scheme. Uniform Resource. Identifier Identification Scheme Data. Uniform Resource. Identifier
[ATD3]	If a UBL xsd:SchemaExpression contains one or more common attributes that apply to all UBL elements contained or included or imported therein, the common attributes MUST be declared as part of a global attribute group.
[ATD4]	Within the ccts:CCT xsd:extension element an xsd:attribute MUST be declared for each ccts:SupplementaryComponent pertaining to that ccts:CCT.
[ATD5]	For each ccts:CCT simpleType xsd:Restriction element, an xsd:base attribute MUST be declared and set to the appropriate xsd:datatype.
[ATD6]	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.
[ATD7]	The xsd built in nillable attribute MUST NOT be used for any UBL declared element.
[ATD8]	The xsd:any attribute MUST NOT be used.

Attribute Naming Rules	
[ATN1]	Each CCT:SupplementaryComponent xsd:attribute "name" MUST be the dictionary entry name object class, property term and representation term of the ccts:SupplementaryComponent with the separators removed.

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.
[CDL4]	All UBL maintained or used Code Lists MUST be enumerated using the UBL Code List Schema Module.
[CDL5]	The name of each UBL Code List Schema Module MUST be of the form: {Owning Organization} {Code List Name} {Code List Schema Module}
[CDL6]	An xsd:Import element MUST be declared for every code list required in a UBL schema.
[CDL7]	Users of the UBL Library MAY identify any subset they wish from an identified code list for their own trading community conformance requirements.
[CDL8]	The xsd:schemaLocation MUST include the complete URI used to identify the relevant code list schema.

ComplexType Definition Rules	
[CTD1]	For every class identified in the UBL model, a named xsd:complexType MUST be defined.
[CTD2]	Every ccts:ABIE xsd:complexType definition content model MUST use the xsd:sequence element with appropriate global element references, or local element declarations in the case of ID and Code, to reflect each property of its class as defined in the corresponding UBL model.
[CTD3]	Every ccts:BBIEProperty xsd:complexType definition content model MUST use the xsd:simpleContent element.
[CTD4]	Every ccts:BBIEProperty ComplexType content model xsd:simpleContent element MUST consist of an xsd:extension element.
[CTD5]	Every ccts:BBIEProperty xsd:complexType content model xsd:base attribute value MUST be the ccts:CCT of the unspecialised or specialised UBL datatype as appropriate.
[CTD6]	For every datatype used in the UBL model, a named xsd:complexType or xsd:simpleType MUST be defined.
[CTD7]	Every unspecialised Datatype must be based on a ccts:CCT represented in the CCT schema module and must represent an approved primary or secondary representation term identified in the CCTS.
[CTD8]	Each unspecialised Datatype xsd:complexType must be based on its corresponding CCT xsd:complexType.
	Every unspecialised Datatype that represents a primary representation term whose

ComplexType Definition Rules	
[CTD9]	corresponding ccts:CCT is defined as an xsd:simpleType MUST also be defined as an xsd:simpleType and MUST be based on the same xsd:simpleType.
[CTD10]	Every unspecialised Datatype that represents a secondary representation term whose corresponding ccts:CCT is defined as an xsd:simpleType MUST also be defined as an xsd:simpleType and MUST be based on the same xsd:simpleType.
[CTD11]	Each unspecialised Datatype xsd:complexType definition must contain one xsd:simpleContent element.
[CTD12]	The unspecialised Primary Representation Term Datatype xsd:complexType definition xsd:simpleContent element must contain one xsd:restriction element with an xsd:base attribute whose value is equal to the corresponding cct:complexType.
[CTD13]	For every ccts:CCT whose supplementary components are not equivalent to the properties of a built-in xsd:datatype, the ccts:CCT MUST be defined as a named xsd:complexType in the ccts:CCT schema module.
[CTD14]	Each ccts:CCT xsd:complexType definition MUST contain one xsd:simpleContent element
[CTD15]	The ccts:CCT xsd:complexType definition xsd:simpleContent element MUST contain one xsd:extension element. This xsd:extension element MUST include an xsd:base attribute that defines the specific xsd:built-inDatatype required for the ccts:ContentComponent of the ccts:CCT.
[CTD16]	Each CCT:SupplementaryComponent xsd:attribute "type" MUST define the specific xsd:built-in Datatype or the user defined xsd:simpleType for the ccts:SupplementaryComponent of the ccts:CCT.
[CTD17]	Each ccts:SupplementaryComponent xsd:attribute user-defined xsd:simpleType MUST only be used when the ccts:SupplementaryComponent is based on a standardized code list for which a UBL conformant code list schema module has been created.
[CTD18]	Each ccts:SupplementaryComponent xsd:attribute user defined xsd:simpleType MUST be the same xsd:simpleType from the appropriate UBL conformant code list schema module for that type.
[CTD19]	Each ccts:Supplementary Component xsd:attribute "use" MUST define the occurrence of that ccts:SupplementaryComponent as either "required", or "optional.

ComplexType 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 the representation term of the shared ccts:BasicBusinessInformationEntity, with the

ComplexType Naming Rules	
	separators removed and with the "Type" suffix appended after the representation term.
[CTN3]	A UBL xsd:complexType for a cct:UnspecialisedDatatype used in the UBL model MUST have the name of the corresponding ccts:CoreComponentType, with the separators removed and with the "Type" suffix appended.
[CTN4]	A UBL xsd:complexType for a cct:UnspecialisedDatatype based on a ccts:SecondaryRepresentationTerm used in the UBL model MUST have the name of the corresponding ccts:SecondaryRepresentationTerm, with the separators removed and with the "Type" suffix appended.
[CTN5]	A UBL xsd:complexType name based on a ccts:CoreComponentType MUST be the Dictionary entry name of the ccts:CoreComponentType, with the separators removed.

Documentation Rules	
[DOC1]	<p>The xsd:documentation element for every Datatype MUST contain a structured set of annotations in the following sequence and pattern:</p> <ul style="list-style-type: none"> • ComponentType (mandatory): The type of component to which the object belongs. For Datatypes this must be "DT". • DictionaryEntryName (mandatory): The official name of a Datatype. • Version (optional): An indication of the evolution over time of the Datatype. • Definition(mandatory): The semantic meaning of a Datatype. • ObjectClassQualifier (optional): The qualifier for the object class. • ObjectClass(optional): The Object Class represented by the Datatype. • RepresentationTerm (mandatory): A Representation Term is an element of the name which describes the form in which the property is represented. • DataTypeQualifier (optional): semantically meaningful name that differentiates the Datatype from its underlying Core Component Type. • DataType (optional): Defines the underlying Core Component Type.
[DOC2]	<p>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:</p> <ul style="list-style-type: none"> • RestrictionType (mandatory): Defines the type of format restriction that applies to the Content Component. • RestrictionValue (mandatory): The actual value of the format restriction that applies to the Content Component. • ExpressionType (optional): Defines the type of the regular expression of the restriction value.

Documentation Rules	
[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> <ul style="list-style-type: none"> • SupplementaryComponentName (mandatory): Identifies the Supplementary Component on which the restriction applies. • RestrictionValue (mandatory, repetitive): The actual value(s) that is (are) valid for the Supplementary Component
[DOC4]	<p>The xsd:documentation element for every Basic Business Information Entity MUST contain a structured set of annotations in the following sequence and pattern:</p> <ul style="list-style-type: none"> • ComponentType (mandatory): The type of component to which the object belongs. For Basic Business Information Entities this must be “BBIE”. • DictionaryEntryName (mandatory): The official name of a Basic Business Information Entity. • Version (optional): An indication of the evolution over time of the Basic Business Information Entity. • Definition(mandatory): The semantic meaning of a Basic Business Information Entity. • 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. • ObjectClassQualifier (optional): The qualifier for the object class. • ObjectClass(mandatory): The Object Class containing the Basic Business Information Entity. • PropertyTermQualifier (optional): A qualifier is a word or words which help define and differentiate a Basic Business Information Entity. • 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. • RepresentationTerm (mandatory): A Representation Term describes the form in which the Basic Business Information Entity is represented. • DataTypeQualifier (optional): semantically meaningful name that differentiates the Datatype of the Basic Business Information Entity from its underlying Core Component Type. • DataType (mandatory): Defines the Datatype used for the Basic Business Information Entity. • AlternativeBusinessTerms (optional): Any synonym terms under which the Basic Business Information Entity is commonly known and used in the business. • Examples (optional): Examples of possible values for the Basic Business Information Entity.

Documentation Rules

[DOC5]

The xsd:documentation element for every Aggregate Business Information Entity MUST contain a structured set of annotations in the following sequence and pattern:

- **ComponentType (mandatory):** The type of component to which the object belongs. For Aggregate Business Information Entities this must be “ABIE”.
- **DictionaryEntryName (mandatory):** The official name of the Aggregate Business Information Entity .
- **Version (optional):** An indication of the evolution over time of the Aggregate Business Information Entity.
- **Definition(mandatory):** The semantic meaning of the Aggregate Business Information Entity.
- **ObjectClassQualifier (optional):** The qualifier for the object class.
- **ObjectClass(mandatory):** The Object Class represented by the Aggregate Business Information Entity.
- **AlternativeBusinessTerms (optional):** Any synonym terms under which the Aggregate Business Information Entity is commonly known and used in the business.

Documentation Rules

[DOC6]

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:

- **ComponentType (mandatory):** The type of component to which the object belongs. For Association Business Information Entities this must be “ASBIE”.
- **DictionaryEntryName (mandatory):** The official name of the Association Business Information Entity.
- **Version (optional):** An indication of the evolution over time of the Association Business Information Entity.
- **Definition(mandatory):** The semantic meaning of the Association Business Information Entity.
- **Cardinality(mandatory):** Indication whether the Association Business Information Entity represents an optional, mandatory and/or repetitive association.
- **ObjectClass(mandatory):** The Object Class containing the Association Business Information Entity.
- **PropertyTermQualifier (optional):** A qualifier is a word or words which help define and differentiate the Association Business Information Entity.
- **PropertyTerm(mandatory):** Property Term represents the Aggregate Business Information Entity contained by the Association Business Information Entity.
- **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 Information Entity.
- **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.

Documentation Rules	
[DOC7]	<p>The xsd:documentation element for every Core Component Type MUST contain a structured set of annotations in the following sequence and pattern:</p> <ul style="list-style-type: none"> • ComponentType (mandatory): The type of component to which the object belongs. For Core Component Types this must be "CCT". • DictionaryEntryName (mandatory): The official name of the Core Component Type, as defined by [CCTS]. • Version (optional): An indication of the evolution over time of the Core Component Type. • Definition(mandatory): The semantic meaning of the Core Component Type, as defined by [CCTS]. • ObjectClass(mandatory): The Object Class represented by the Core Component Type, as defined by [CCTS]. • PropertyTerm(mandatory): The Property Term represented by the Core Component Type, as defined by [CCTS].

Element Declaration Rules	
[ELD1]	Each UBL:ControlSchema MUST identify one and only one global element declaration that defines the document ccts:AggregateBusinessInformationEntity being conveyed in the Schema expression. That global element MUST include an xsd:annotation child element which MUST further contain an xsd:documentation child element that declares "This element MUST be conveyed as the root element in any instance document based on this Schema expression."
[ELD2]	All element declarations MUST be global with the exception of ID and Code which MUST be local.
[ELD3]	For every class 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. When an ccts:ABIE is qualified, a new element MUST be declared and bound to the xsd:complexType of its associated ccts:AggregateBusinessInformationEntity. □
[ELD5]	For each ccts:CCT simpleType, an xsd:restriction element MUST be declared.
[ELD6]	The code list xsd:import element MUST contain the namespace and schema location attributes.
[ELD7]	Empty elements MUST not be declared.
[ELD8]	Global elements declared for Qualified BBIE Properties must be of the same type as its corresponding Unqualified BBIE Property. (i.e. Property Term + Representation Term.)
[ELD9]	The xsd:any element MUST NOT be used.

Element Naming Rules	
[ELN1]	A UBL global element name based on a ccts:ABIE MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the word "Type" removed.
[ELN2]	A UBL global element name based on an unqualified ccts:BBIEProperty MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the word "Type" removed.
[ELN3]	A UBL global element name based on a qualified ccts:ASBIE MUST be the ccts:ASBIE dictionary entry name property term and its qualifiers; and the object class term and qualifiers of its associated ccts:ABIE. All ccts:DictionaryEntryName separators MUST be removed. Redundant words in the ccts:ASBIE property term or its qualifiers and the associated ccts:ABIE object class term or its qualifiers MUST be dropped.
[ELN4]	A UBL global element name based on a Qualified ccts:BBIEProperty MUST be the same as the name of the corresponding xsd:complexType to which it is bound, with the Qualifier prepended(?) and with the word "Type" removed.

General Naming Rules	
[GNR1]	UBL XML element, attribute and type names MUST be in the English language, using the primary English spellings provided in the Oxford English Dictionary.
[GNR2]	UBL XML element, attribute and type names MUST be consistently derived from CCTS conformant dictionary entry names.
[GNR3]	UBL XML element, attribute and type names constructed from ccts:DictionaryEntryNames MUST NOT include periods, spaces, other separators, or characters not allowed by W3C XML 1.0 for XML names.
[GNR4]	UBL XML element, attribute, and simple and complex type names MUST NOT use acronyms, abbreviations, or other word truncations, except those in the list of exceptions published in Appendix B.
[GNR5]	Acronyms and abbreviations MUST only be added to the UBL approved acronym and abbreviation list after careful consideration for maximum understanding and reuse.
[GNR6]	The acronyms and abbreviations listed in Appendix B MUST always be used.
[GNR7]	UBL XML element, attribute 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.
[GNR9]	The lowerCamelCase (LCC) convention MUST be used for naming attributes.

General Type Definition Rules	
[GTD1]	All types MUST be named.

[GTD2]	The xsd:any Type MUST NOT be used.
--------	------------------------------------

General XML Schema Rules

[GXS1]	<p>UBL Schema MUST conform to the following physical layout as applicable: XML Declaration <!-- ===== 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 Datatypes Identifier Schemes Code Lists Attribute Declarations – elementFormDefault=”qualified” attributeFormDefault=”unqualified” <!-- ===== Imports ===== -->CommonAggregateComponents schema module CommonBasicComponents schema module Representation Term schema module (to include CCT module) Unspecialised Types schema module Specialised 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 ===== --> alphabetized order of ccts:AggregateBusinessInformationEntity xsd:TypeDefinitions <!-- =====Basic Business Information Entity Type Definitions ===== --> alphabetized order of ccts:BasicBusinessInformationEntities <!-- ===== Copyright Notice ===== --> Required OASIS full copyright notice.</p>
[GXS2]	UBL MUST provide two normative schemas for each transaction. One schema

General XML Schema Rules	
	shall be fully annotated. One 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: xmlns:xsd="http://www.w3.org/2001/XMLSchema"
[GXS5]	The xsd:SubstitutionGroups feature MUST NOT be used.
[GXS6]	The xsd:final attribute MUST be used to control extensions.
[GXS7]	xsd:notations 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.
[GXS10]	The xsd:include feature MUST only be used within a document schema.
[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.

Instance Document Rules	
[IND1]	All UBL instance documents MUST validate to a corresponding schema.
[IND2]	All UBL instance documents MUST always identify their character encoding with the XML declaration.
[IND3]	In conformance with ISO/IETF/ITU/UNCEFACT Memorandum of Understanding Management Group (MOUMG) Resolution 01/08 (MOU/MG01n83) as agreed to by OASIS, all UBL XML SHOULD be expressed using UTF-8.
[IND4]	All UBL instance documents MUST contain the following namespace declaration in the root element: xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
[IND5]	UBL conformant instance documents MUST NOT contain an element devoid of content or null values.
[IND6]	The absence of a construct or data in a UBL instance document MUST NOT carry meaning.

Modeling Constraints Rules	
[MDC1]	UBL Libraries and Schemas MUST only use ebXML Core Component approved ccts:CoreComponentTypes.
[MDC2]	Mixed content MUST NOT be used except where contained in an

	xsd:documentation element.
--	----------------------------

Naming Constraints Rules

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

Namespace Rules

[NMS1]	Every UBL-defined or -used schema module MUST have a namespace declared using the xsd:targetNamespace attribute.
[NMS2]	Every UBL defined or used schema set version 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: urn:oasis:names:tc:ubl:schema:<subtype>:<document-id>
[NMS5]	The namespace names for UBL Schemas holding OASIS Standard status MUST be of the form: urn:oasis:names:specification:ubl:schema:<subtype>:<document-id>
[NMS6]	UBL Schema modules MUST be hosted under the UBL committee directory: <a href="http://www.oasis-open.org/committees/ubl/schema/<subtype>/UBL-<document-id>.<filetype>">http://www.oasis-open.org/committees/ubl/schema/<subtype>/UBL-<document-id>.<filetype>
[NMS7]	UBL published namespaces MUST never be changed.
[NMS8]	The ubl:CommonAggregateComponents schema module MUST reside in its own namespace.
[NMS9]	The ubl:CommonAggregateComponents schema module MUST be represented by the token "cac".
[NMS10]	The ubl:CommonBasicComponents schema module MUST reside in its own namespace.
[NMS11]	The UBL:CommonBasicComponents schema module MUST be represented by the token "cbc".
[NMS12]	The ccts:CoreComponentType schema module MUST reside in its own namespace.
[NMS13]	The ccts:CoreComponentType schema module namespace MUST be represented by the token "cct".
[NMS14]	The ccts:UnspecialisedDatatype schema module MUST reside in its own namespace.
[NMS15]	The ccts:UnspecialisedDatatype schema module namespace MUST be represented by the token "udt".
[NMS16]	The ubl:SpecialisedDatatypes schema module MUST reside in its own namespace.
[NMS17]	The ubl:SpecialisedDatatypes schema module namespace MUST be represented

Namespace Rules	
	by the token "sdt".
[NMS18]	Each UBL:CodeList schema module MUST be maintained in a separate namespace.

Root Element Declaration Rules	
[RED1]	Every UBL instance document must use the global element defined as the root element in the schema as its root element.

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 UBL 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.
[SSM4]	Imported schema modules MUST be fully conformant with UBL naming and design rules.
[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:CommonAggregateComponents MUST be created.
[SSM10]	The ubl:CommonAggregateComponents schema module MUST be named "ubl:CommonAggregateComponents Schema Module"
[SSM11]	A schema module defining all ubl:CommonBasicComponents MUST be created.
[SSM12]	The ubl:CommonBasicComponents schema module MUST be named "ubl:CommonBasicComponents Schema Module"
[SSM13]	A schema module defining all ccts:CoreComponentTypes MUST be created.
[SSM14]	The ccts:CoreComponentType schema module MUST be named "ccts:CoreComponentType Schema Module"
[SSM15]	The xsd:facet feature MUST not be used in the ccts:CoreComponentType schema module.
[SSM16]	A schema module defining all ccts:UnspecialisedDatatypes MUST be created.

Schema Structure Modularity Rules

[SSM17]	The ccts:UnspecialisedDatatype schema module MUST be named "ccts:UnspecialisedDatatype Schema Module"
[SSM18]	A schema module defining all ubl:SpecialisedDatatypes MUST be created.
[SSM19]	The ubl:SpecialisedDatatypes schema module MUST be named "ubl:SpecialisedDatatypes schema module"

Standards Adherence rules

[STA1]	All UBL schema design rules MUST be based on the W3C XML Schema Recommendations: XML Schema Part 1: Structures and XML Schema Part 2: Datatypes.
[STA2]	All UBL schema and messages MUST be based on the W3C suite of technical specifications holding recommendation status.
[STN1]	Each CCTS:CCT simpleType definition name MUST be the ccts:CCT dictionary entry name with the separators removed.

SimpleType Naming Rules

[STN1]	Each CCTS:CCT simpleType definition name MUST be the ccts:CCT dictionary entry name with the separators removed.
--------	--

SimpleType Definition Rules

[STD1]	For every ccts:CCT whose supplementary components map directly onto the properties of a built-in xsd:DataType, the ccts:CCT MUST be defined as a named xsd:simpleType in the ccts:CCT schema module.
--------	--

Versioning Rules

[VER1]	Every UBL Schema and schema module major version committee draft MUST have an RFC 3121 document-id of the form $\langle \text{name} \rangle - \langle \text{major} \rangle . 0 [. \langle \text{revision} \rangle]$
[VER2]	Every UBL Schema and schema module major version OASIS Standard MUST have an RFC 3121 document-id of the form $\langle \text{name} \rangle - \langle \text{major} \rangle . 0$
[VER3]	Every minor version release of a UBL schema or schema module draft MUST have an RFC 3121 document-id of the form $\langle \text{name} \rangle - \langle \text{major} \rangle . \langle \text{non-zero} \rangle [. \langle \text{revision} \rangle]$
[VER4]	Every minor version release of a UBL schema or schema module OASIS Standard MUST have an RFC 3121 document-id of the form

Versioning Rules	
	<name>-<major >.<non-zero>
[VER5]	For UBL Minor version changes, the name of the version construct 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.
[VER8]	A UBL minor version document schema MUST import its immediately preceding version document schema.
[VER9]	UBL Schema and schema module minor version changes MUST be limited to the use of xsd:extension or xsd:restriction to alter existing types or add new constructs.
[VER10]	UBL Schema and schema module minor version changes MUST not break semantic compatibility with prior versions.

Appendix B. Approved Acronyms and Abbreviations

The following Acronyms and Abbreviations have been approved for UBL use:

- ◆ A Dun & Bradstreet number *must* appear as "DUNS". [TBD: need example.]
- ◆ "Identifier" *must* appear as "ID".
- ◆ "Uniform Resource Identifier" *must* appear as "URI"
- ◆ [Example] the "Uniform Resource. Identifier" portion of the **Binary Object. Uniform Resource. Identifier** supplementary component becomes "URI" in the resulting XML name). The use of URI for Uniform Resource Identifier takes precedence over the use of "ID" for "Identifier".