



**Open Applications Group, Inc.  
OAGIS Semantic Refinement Method and Tool**

***Specification for Profile BOD XML Schema Representation***

Editor: Serm Kulvatunyou, NIST

Creation Date: 09/27/2015

Last Updated: 10/20/2017 7:15 AM

Version: 1.0

Status: Draft

Contributors:

David Connelly, OAGi

Jim Wilson, KCX & OAGi

Michael Rowell, OAGi/Oracle

Jeff Rice, Boeing

Scott Nieman, Land O’Lakes

*Document #20141014SMRAV1.0*

Change Tracking

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Editor | Version | Change Detail |
| 09/27/2015 | Serm Kulvatunyou & Jaehun Lee | 1.0 | Initialized the first draft. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Change Tracking 2](#_Toc468787021)

[Table of Contents 3](#_Toc468787022)

[1. Purpose 4](#_Toc468787023)

[2. Scope 5](#_Toc468787024)

[3. Relevant Specifications 6](#_Toc468787025)

[4. Glossary 7](#_Toc468787026)

[5. Overall Design 8](#_Toc468787027)

[5.1 One global element 8](#_Toc468787028)

[5.2 Forgoing type inheritances 8](#_Toc468787029)

[5.3 Forgoing the xsd:group 9](#_Toc468787030)

[6. aComplex Type Representation 10](#_Toc468787031)

[7. Code List Representation 11](#_Toc468787032)

[8. Schema Package 12](#_Toc468787033)

[9. Documentation Representation 13](#_Toc468787034)

# Purpose

The purpose of this document is to describe the XML schema convention to represent the profile BOD XML schema. The purpose of a profile BOD XML schema is to provide precise semantic restrictions of a BOD usage information in an XML schema while also simplify the schema structure as much as possible.

Profile BOD is a kind of a standalone BOD where all the content is self-contained in a single XML Schema file. The profile BOD is different from the standalone BOD in two respects.

First, the profile BOD employs, for the most part, the Russian Doll[[1]](#footnote-1) XML schema design while the standalone BOD retains the Garden of Eden design. In Russian Doll, there is one root element (the BOD) and types are *anonymous-type*. In the profile BOD, there is one root element and most types are anonymous except code list and OAGIS defined built-in types.

The objective of the profile BOD XML schema design is to make the content model as precise as possible. Using the Russian Doll design allows the same types defined in the OAGIS Model (e.g., supplier party type, business data types) used in multiple places within a single BOD to have specific contents (or restrictions). With the profile BOD XML schema design, integration interfaces will be able to declare precise integration requirements.

The other difference is that the profile BOD implements the Core Component Specification (CCS) context mechanism. Therefore, the profile BOD contains data elements which are only subset of the model or standalone BOD it corresponds to. In addition, it can contain meta-data (e.g., component identity and version information), *context* information, and context specific documentations. The context information and context specific information describes suitable usage information of the profile BOD.

# Scope

A profile BOD (which is a Business Information Entity (BIE) in Core Component Specification terminology) can have many expressions such as XML Schema, JSON, or others. The scope of this document is limited to the XML Schema expression.

The size of a plain Russian-Doll-based profile BOD schema can be very large if it contains thousands of data fields because types are repeatedly defined even for the common attributes used throughout a BOD. The schema can be made more compact by *globally* defining types that do not have different content or restriction. In other words, anonymous types that have a common content model can be globally defined and reused. For example, if there is no difference in the SupplierParty used in the header and the line of a profile BOD, only a single global SupplierPartyType needs to be defined. Similarly, if there is no different restriction between a TaxAmount and a TotalAmount field, a single global XyzAmountType can be defined and reused. The actual profile BOD can be more complicated to optimize than these two examples such as when there are a few Amount typed fields with one content model and there are other few Amount typed fields with another content model and so on.

At present, the working group member see no need for such optimization as profile BODs are typically small. Therefore, the optimization to make the profile BOD most compact, yet precise, is not in scope of the current specification. The current specification only reduces the size of the profile BOD by using global types for code list definitions and OAGIS-defined built-in types (the naming convention, enforced in the Semantic Refinement Tool, makes the code list and built-in type names unique).

There are also requirements from the user community to represent semantic restrictions beyond the capability of the XML schema by itself. These are documented here and may be considered for future versions of the specification.

1. To represent IF-THEN type of rules for conditional restrictions across data elements.
2. To represent specific values needed in a particular occurrences of a repeated data element, e.g., when two IDs of the Party element are needed; and the requirement is that one is the DUNS number and the other is the Tax ID.

# Relevant Specifications

Core Component Specification

OAGIS Release 10.X Naming and Design Rules - <http://www.oagi.org/oagi/downloads/ResourceDownloads/UNCEFACT_XML_NDR_V3p0.pdf>

OAGIS Repository Data Model

# Glossary

OAGIS Model XML schemas = The OAGIS canonical XML schemas as defined in the OAGIS enterprise edition under the Model folder.

Standalone OAGIS XML schemas = The all-inclusive OAGIS BOD XML schemas as defined in the Standalone folder of OAGIS enterprise and standard edition.

SRT = Semantic Refinement Tool

CCS = Core Component Specification

CC = Core Component per CCS.

BCC = Basic Core Component per CCS. It can also refer to the BCC table in the OAGIS Repository.

ACC = Aggregate Core Component. It can also refer to the ACC table in the OAGIS Repository.

ASCC = Association Core Component. It can also refer to the ASCC table in the OAGIS Repository.

ASCCP = Association Core Component Property. It can also refer to the ASCCP table in the OAGIS Repository.

BDT = Business Data Type

SC = Supplementary Component (of a BDT)

BIE = Business Information Entity per CCS.

BBIE = Basic Business Information Entity per CCS. It can also refer to the BBIE table in the OAGIS Repository.

ABIE = Aggregate Business Information Entity per CCS. It can also refer to the ABIE table in the OAGIS Repository.

ASBIE = Association Business Information Entity per CCS. It can also refer to the ASBIE table in the OAGIS Repository.

ASBIEP = Association Property Business Information Entity per CCS. It can also refer to the ASBIEP table in the OAGIS Repository.

BBIE\_SC = A table containing SC restrictions for a BBIE.

OAGIS Model = The canonical representation of OAGIS where BODs are represented with reusable components and have full content.

Model BOD = A BOD in the OAGIS Model

Standalone BOD = A Model BOD with all of its specification contained in one physical file

Profile BOD = A trimmed down or subset version of the Model BOD

Expression = A syntax-specific representation of a specification.

OAGIS Repository = OAGIS specifications in a database management system.

Syntax independent [X] where [X] is OAGIS Model, Model BOD, Standalone BOD, or Profile BOD = OAGIS Model, Model BOD, Standalone BOD, and Profile BOD represented via Core Component Specification in the OAGIS Repository. It is important to note that the terms OAGIS Model, Model BOD, Standalone BOD, and Profile BOD are always used in this abstract, syntax-independent, sense in this specification. To refer to a syntax-specific representation of these concepts, these terms are suffixed with the name of the particular syntax (expression), e.g., OAGIS Model XML schema, Profile BOD XML Schema, Profile BOD JSON schema.

BOD element = The data element representing an OAGIS BOD containing an OAGIS Verb and OAGIS Noun, e.g., ProcessPurchaseOrder, SyncBOM.

GUID = Globally Unique Identifier

XML Schema = The W3C XML Schema standard.

XML schema = An instantiation of the W3C XML Schema standard.

XML instance = An instantiation of an XML schema.

# Overall Design

The profile BOD XML schema generally follows the OAGIS Release 10.X Naming and Design Rules (see Relevant Specification) except those described in this section. The design rules described in this section shall override relevant rules in the OAGIS Release 10.X Naming and Design Rules.

The overall schema design objective is for it to be simple yet provides precise semantic constraints. Therefore, the Russian Doll XML schema design pattern is used and layers of type inheritances (extension and restriction) are reduced to a single type definition and XML Schema group wrappers are skipped. These overall design details are explained in this section.

## Compatibility

A profile BOD XML schema shall be compatible with the corresponding Model BOD XML schema. In other words, XML instances valid according to a profile BOD XML schema shall be valid according to the corresponding Model BOD XML schema.

## XML Schema design pattern

The profile BOD XML schema employs the Russian Doll design pattern (i.e., local elements and anonymous types). **Only** code lists and OAGIS defined built-in type primitives are defined globally and are reusable. Other specifics of this pattern are specified in section 7.

## GUID

GUID of the elements and types in the profile BODs are newly assigned. In order words, they are different from their correspondences in the OAGIS Model. The reason is the content in the profile BOD consists of Business Information Entities (BIEs) which are derived from Core Components (CCs) in the Model and they may have unique restrictions. Each BIE derivation results in a new GUID assigned. The only exception is the GUIDs of the code lists and the OAGIS defined built-in types, which are constant, because code lists are treated as primitives.

GUID of the root BOD element is GUID of the corresponding ASBIEP. GUIDs of local elements are GUIDs of the corresponding ASBIEs or BBIEs.

GUID of an attribute that is derived from a BCC is the GUID of the corresponding BBIE. GUID of an attribute that is derived from an SC is the GUID of the corresponding BBIE\_SC.

GUID of a local complex type with a complex content (one with an xsd:sequence) is the GUID of the ABIE corresponding to the owner ASBIEP element.

GUID of a local complex type with a simple content is the GUID of the BDT corresponding to the owner BBIE (and BBIEP) element.

GUID of a local simple type is the GUID of the BDT corresponding to the owner BBIE (and BBIEP) element or attribute.

GUID of a global simple type is the GUID of a code list or an OAGIS built-in type.

## Global element - ASBIEP

A profile BOD XML schema generally contains a global xsd:element representing the root BOD element (i.e., conveying Verb + Noun semantics). It is an ASBIEP in the CCTS term; and it is derived from an ASCCP. All types are local and anonymous except the type definitions for code lists and OAGIS defined built-in type primitives used in the profile BOD. For example:

<xsd:element name="ProcessPurchaseOrder">

<xsd:complexType>

<!-- Specification of the ProcessPurchaseOrder profile BOD goes here. -->

</xsd:complexType>

</xsd:element>

<xsd:simpleType name="xbt\_DateHourMinuteUTCType" id="oagis-id-3b54f63b41304d2681ec4483a32e93a5">

<!-- Specification of the xbt\_DateHourMinuteUTCType primitive used in the profile BOD goes here. -->

</xsd:simpleType>

<xsd:simpleType name="companyA\_2011\_LanguageCodeContentType\_oagis-id-555596b8c9fb46cda2724a1770fa5555" id="oagis-id-555596b8c9fb46cda2724a1770fa5555">

<!-- Specification of the code list used in the profile BOD goes here. -->

</xsd:simpleType>

However, the profile BOD specification also supports the notion of “schema package”. In the case of schema package, a profile BOD XML schema can contain multiple global xsd:element. Each global xsd:element must be a BOD element. For example, a profile BOD schema package may contain:

<xsd:element name="ProcessPurchaseOrder">

<xsd:complexType>

<!-- Specification of the profiled ProcessPurchaseOrder BOD goes here. -->

</xsd:complexType>

</xsd:element>

<xsd:element name="AcknowledgePurchaseOrder">

<xsd:complexType>

<!-- Specification of the profiled AcknowledgePurchaseOrder BOD goes here. -->

</xsd:complexType>

</xsd:element>

<!-- Code list and primitive types used by the profile BODs in the same schema package are defined globally and shared across the profile BODs. -->

<xsd:simpleType name="xbt\_DateHourMinuteUTCType" id="oagis-id-3b54f63b41304d2681ec4483a32e93a5">

<!-- Specification of the xbt\_DateHourMinuteUTCType primitive goes here. -->

</xsd:simpleType>

<xsd:simpleType name="companyA\_2011\_LanguageCodeContentType\_oagis-id-555596b8c9fb46cda2724a1770fa5555" id="oagis-id-555596b8c9fb46cda2724a1770fa5555">

<!-- Specification of the code list goes here. -->

</xsd:simpleType>

## Forgoing the type inheritance

Type inheritances are forgone in the profile BOD XML schema (this is also the way the BIE entities are represented in the database – no inheritance). Examples below illustrate these for the cases of complex content extension, simple content restriction, and simple type restriction.

**Complex content type extension example**

Below is a snippet of the model specification of the Party element (ASCCP). The associated PartyType complex type (ACC) has two inheritances, i.e., it extends the PartyBaseType ACC, which in turn extends the PartyIdentificationType ACC in the model specification.

<xsd:complexType name="PartyIdentificationType" id="oagis-id-00012d4229984113976240713ed38906">

<xsd:sequence>

<xsd:element ref="ID" id="oagis-id-00022d4229984113976240713ed38906"minOccurs="0" maxOccurs="unbounded" />

<xsd:element ref="PartyIDSet" id="oagis-id-00032d4229984113976240713ed38906" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

<xsd:attribute name="typeCode" id="oagis-id-00042d4229984113976240713ed38906" type="CodeType\_1E7368" use="optional"/>

<xsd:attribute name="role" id="oagis-id-00052d4229984113976240713ed38906" type="PartyRoleCodeContentType" use="optional"/>

</xsd:complexType>

<xsd:complexType name="PartyBaseType" id="oagis-id-00062d4229984113976240713ed38906" abstract="true">

<xsd:complexContent>

<xsd:extension base="PartyIdentificationType">

<xsd:sequence minOccurs="1" maxOccurs="1">

<xsd:element ref="AccountID" id="oagis-id-00072d4229984113976240713ed38906" minOccurs="0" maxOccurs="unbounded"/>

<xsd:element ref="Name" id="oagis-id-00082d4229984113976240713ed38906" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:complexType name="PartyType" id="oagis-id-00092d4229984113976240713ed38906" abstract="false">

<xsd:complexContent>

<xsd:extension base="PartyBaseType">

<xsd:sequence minOccurs="1" maxOccurs="1">

<xsd:element name="Extension" id="oagis-id-00102d4229984113976240713ed38906" type="PartyExtensionType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:extension>

</xsd:complexContent>

</xsd:complexType>

<xsd:element name="Party" id="oagis-id-00112d4229984113976240713ed38906" type="PartyType">

Below shows how the Party element (ASBIEP) shall appear inside a profile BOD (assuming the Party element is a child under the PurchaseOrderHeader element (ASBIEP). Notice that there is no type extension under the Party ASBIEP, the content of the two based types in the model are merged into the Party’s complex type definition, i.e., all children of the two based types in the model are direct children of the Party’s type (ABIE) definition.

<xsd:element name="PurchaseOrderHeader" id="oagis-id-00122d4229984113976240713ed38906">

<xsd:complexType id="oagis-id-00132d4229984113976240713ed38906">

<xsd:sequence>

<xsd:element name="Party" id="oagis-id-00142d4229984113976240713ed38906">

<xsd:complexType id="oagis-id-00152d4229984113976240713ed38906">

<xsd:sequence>

<xsd:element name="ID" id="oagis-id-00162d4229984113976240713ed38906">

<xsd:complexType id="oagis-id-00172d4229984113976240713ed38906">

<!-- Specification of the ID element -->

</xsd:complexType>

</xsd:element>

<xsd:element name="PartyIDSet" id="oagis-id-00182d4229984113976240713ed38906">

<xsd:complexType id="oagis-id-00192d4229984113976240713ed38906">

<!-- Specification of the PartyIDSet element -->

</xsd:complexType>

</xsd:element>

<xsd:element name="AccountID" id="oagis-id-00202d4229984113976240713ed38906">

<xsd:complexType id="oagis-id-00212d4229984113976240713ed38906">

<!-- Specification of the AccountID element -->

</xsd:complexType>

</xsd:element>

<xsd:element name="Name" id="oagis-id-00222d4229984113976240713ed38906">

<xsd:complexType id="oagis-id-00232d4229984113976240713ed38906">

<!-- Specification of the Name element -->

</xsd:complexType>

</xsd:element>

<xsd:element name="Extension" id="oagis-id-00242d4229984113976240713ed38906">

<xsd:complexType id="oagis-id-00252d4229984113976240713ed38906">

<!-- Specification of the Extension element -->

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

**Simple content type extension example**

In the example below the Name element (BCCP) is bound to OpenNameType (BDT). OpenNameType is defined in the OAGIS Model with two inheritances through xsd:extension based on NameType BDT and NameType\_02FC2Z (BDT) as follows.

<xsd:complexType name="OpenNameType" id="oagis-id-3cd82d4229984113976240713ed38906">

<xsd:simpleContent>

<xsd:extension base="NameType">

<xsd:attribute name="typeCode" type="xsd:token" use="optional" id="oagis-id-82d03758dfd844bea1676759edf0d653"/>

</xsd:extension>

</xsd:simpleContent>

</xsd:complexType>

<xsd:complexType name="NameType" id="oagis-id-bf66e0afea2c4c2da7bc69af14ca23c9">

<xsd:simpleContent>

<xsd:extension base="NameType\_02FC2Z">

<xsd:attribute name="sequenceNumber" type="xsd:integer" id="oagis-id-84fa20db74b942449e1885cff79b24df">

</xsd:attribute>

</xsd:extension>

</xsd:simpleContent>

</xsd:complexType>

<xsd:complexType name="NameType\_02FC2Z" id="oagis-id-8ef2aeaecfa645088c4bf4b424905596">

<xsd:simpleContent>

<xsd:extension base="xsd:string">

<xsd:attribute name="languageCode" type="clm56392A20081107\_LanguageCodeContentType" use="optional" id="oagis-id-42e59d799de147b8ab49c8a27ec85ff1">

</xsd:attribute>

</xsd:extension>

</xsd:simpleContent>

</xsd:complexType>

<xsd:element name="Name" type="OpenNameType">

The Name element (BIE) shall be defined in the profile BOD as shown below. Notice that all attributes from the two based types become direct attributes of the Name’s type definition:

<xsd:element name="Name" id="oagis-id-975496b8c9fb46cda2724a1770facf54">

<xsd:complexType id="oagis-id-000096b8c9fb46cda2724a1770fa1111">

<xsd:simpleContent>

<xsd:extension base="xsd:string">

<xsd:attribute name="languageCode" id="oagis-id-222296b8c9fb46cda2724a1770fa2222" type="CompanyA\_2011\_LanguageCodeContentType\_ oagis-id-555596b8c9fb46cda2724a1770fa5555" use="optional"/>

<xsd:attribute name="sequenceNumber" id="oagis-id-333396b8c9fb46cda2724a1770fa3333" type="xsd:positiveInteger" use="optional"/>

<xsd:attribute name="typeCode" id="oagis-id-444496b8c9fb46cda2724a1770fa4444" type=" CompanyA\_2012\_NameTypeCodeContentType\_ oagis-id-666696b8c9fb46cda2724a1770fa6666" use="optional"/>

</xsd:extension>

</xsd:simpleContent>

</xsd:complexType>

</xsd:element>

<xsd:simpleType name="clCompanyA\_2011\_LanguageCodeContentType\_oagis-id-555596b8c9fb46cda2724a1770fa5555" id="oagis-id-555596b8c9fb46cda2724a1770fa5555">

<xsd:restriction base="xsd:token">

<xsd:enumeration value="US-EN"/>

<xsd:enumeration value="FR"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="clCompanyA\_2012\_NameTypeCodeContentType\_oagis-id-666696b8c9fb46cda2724a1770fa6666" id="oagis-id-666696b8c9fb46cda2724a1770fa6666">

<xsd:restriction base="xsd:token">

<xsd:enumeration value="Firstname"/>

<xsd:enumeration value="Lastname"/>

<xsd:enumeration value="Middlename"/>

</xsd:restriction>

</xsd:simpleType>

Notice the followings:

1. The type hierarchy from OpenNameType to NameType\_02FC2Z was reduced to one type definition based on xsd:string per the definition of the root type, NameType\_02FC2Z, in the model. Therefore, the Name element (BBIEP) in the profile BOD is simply based on xsd:string. If, within the SRT, the user restricted the primitive of the Name element to xsd:token, then the base type will be xsd:token.
2. Although the languageCode attribute (SC) in the model uses clm56392A20081107\_LanguageCodeContentType code list, the user may use the SRT to assign another code list in the BOD profile. In this example, the user assigned the languageCode attribute with a Company A’s code list (which must be based on the clm56392A20081107\_LanguageCodeContentType code list). In the profile BOD expression, code lists are always defined using a global type (there is no name clashing because the SRT ensures that combination of the code list’s agency ID, list ID, and version ID are unique. See section 7.8 how the name of the code list type is generated).
3. Similar to the languageCode attribute, the user assigned a specific code list to the typeCode attribute (SC).

**Simple type restriction example**

In the example below, the CreationDateTime element (BCCP) uses DateTimeType (BDT). DateTimeType has two inheritances through a restriction on DateTimeType\_AD9DD9 (BDT), which is in turn a union of a number of types, all of which are based on xsd:token.

<xsd:element name="CreationDateTime" type="DateTimeType" id="oagis-id-4ba8e6b8c9fb46cda2724a1770fa9baf">

</xsd:element>

<xsd:simpleType name="DateTimeType" id="oagis-id-dd0c8f86b160428da3a82d2866a5b48d">

<xsd:restriction base="DateTimeType\_AD9DD9"/>

</xsd:simpleType>

<xsd:simpleType name="DateTimeType\_AD9DD9" id="oagis-id-a5cfd20385314a63afc1ffcf6357a08b" final="union">

<xsd:union memberTypes=" xbt\_CenturyType xbt\_DateType xbt\_DayOfWeekType …… xbt\_YearWeekDayTimeType xbt\_YearWeekDayTimeUTCType xbt\_YearWeekDayTimeUTCOffsetType">

</xsd:union>

</xsd:simpleType>

<xsd:simpleType name="xbt\_CenturyType" id="oagis-id-433b017552a14828a92821fd2540d790">

<xsd:restriction base="xsd:token">

<xsd:pattern value="[0-9]{2}"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="xbt\_YearWeekDayTimeUTCOffsetType" id="oagis-id-bd2e6cf3f01a4a90b9ec141fc908f531">

<xsd:restriction base="xsd:token">

<xsd:pattern value="[0-9]{4}-W(0[1-9]|[1-4][0-9]|5[0123])-[1-7]T((([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](|(\.[0-9]+)))|(24:00:00))([\+|\-]([0-1][0-9]|2[0-3]):[0-5][0-9])"/>

</xsd:restriction>

</xsd:simpleType>

The element CreationDateTime (BBIEP) would appear in the Profile BOD schema as follows, assuming that the user constrains CreationDateTime to xbt\_YearWeekDayTimeUTCOffsetType in the SRT.

<xsd:element name="CreationDateTime" type="DateTimeType" id="oagis-id-257896b8c9fb46cda2724a1770fa9baf">

<xsd:restriction base=" xbt\_YearWeekDayTimeUTCOffsetType ">

</xsd:restriction>

</xsd:element>

<xsd:simpleType name="xbt\_YearWeekDayTimeUTCOffsetType" id="oagis-id-57a296b8c9fb46cda2724a1770f3b5a5">

<xsd:restriction base="xsd:token">

<xsd:pattern value="[0-9]{4}-W(0[1-9]|[1-4][0-9]|5[0123])-[1-7]T((([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](|(\.[0-9]+)))|(24:00:00))([\+|\-]([0-1][0-9]|2[0-3]):[0-5][0-9])"/>

</xsd:restriction>

</xsd:simpleType>

## Forgoing the xsd:group

An xsd:group in the model specification is imported as ACC and ASCCP. However, when a BIE which uses the xsd:group is created, there is no ABIE nor ASBIEP created which is corresponding to the xsd:group. Instead, associations are established directly between the parent of the group and children of the group. The snippet below shows an example.

Model specification:

<xsd:element name="ProductionOrder" type="ProductionOrderType" id="oagis-id-111296b8c9fb46cda2724a1770fa1112"/>

<xsd:complexType name="ProductionOrderType" id="oagis-id-111396b8c9fb46cda2724a1770fa1113">

<xsd:sequence>

<xsd:element ref="ProductionOrderHeader" id="oagis-id-111496b8c9fb46cda2724a1770fa1114"/>

<!-- some other elements here, not shown -->

</xsd:sequence>

</xsd:complexType>

<xsd:element name="ProductionOrderHeader" type="ProductionOrderHeaderType" id="oagis-id-111596b8c9fb46cda2724a1770fa1115"/>

<xsd:complexType name="ProductionOrderHeaderType" id="oagis-id-111696b8c9fb46cda2724a1770fa1116">

<xsd:sequence>

<!-- some other elements here, not shown -->

<xsd:group ref="FreeFormTextGroup" id="oagis-id-111696b8c9fb46cda2724a1770fa1116"/>

<!-- some other elements here, not shown -->

</xsd:sequence>

</xsd:complexType>

<xsd:group name="FreeFormTextGroup" id="oagis-id-111796b8c9fb46cda2724a1770fa1117">

<xsd:sequence>

<xsd:element ref="Description" id="oagis-id-111896b8c9fb46cda2724a1770fa1118" minOccurs="0" maxOccurs="unbounded"/>

<xsd:element ref="Note" id="oagis-id-111

1996b8c9fb46cda2724a1770fa1119" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:group>

Profile BOD specification:

<xsd:element name="ProductionOrder" id="oagis-id-1120296b8c9fb46cda2724a1770fa1120">

<xsd:complexType id="oagis-id-112196b8c9fb46cda2724a1770fa1121">

<xsd:sequence>

<xsd:element name="ProductionOrderHeader" id="oagis-id-112296b8c9fb46cda2724a1770fa1122">

<xsd:complexType id="oagis-id-112396b8c9fb46cda2724a1770fa1123">

<xsd:sequence>

<!-- some other elements here, not shown -->

<xsd:element name="Description" id="oagis-id-112496b8c9fb46cda2724a1770fa1124" minOccurs="0" maxOccurs="unbounded">

<xsd:complexType id="oagis-id-112596b8c9fb46cda2724a1770fa1125">

<!-- Specification of the Description element -->

</xsd:complexType>

</xsd:element>

<xsd:element name="Note" minOccurs="0" maxOccurs="unbounded">

<xsd:complexType id="oagis-id-112596b8c9fb46cda2724a1770fa1125">

<!-- Specification of the Note element -->

</xsd:complexType>

</xsd:element>

<!-- some other elements here, not shown -->

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<!-- some other elements here, not shown -->

</xsd:sequence>

</xsd:complexType>

</xsd:element>

# Schema Package

Schema package is the case where multiple profile BODs are included in a single schema file. It is commonly used for supporting Web Services deployment. Section 5.4 outlines an example Schema Package.

# Mapping of OAGIS Repository Entities to Profile BOD XML Schema Constructs

This section formally specifies the maps from table entities (which also correspond to CCS entities) in the OAGIS Repository to XML schema constructs (see the OAGIS Repository Data Model for reference). It indicates how the profile XML schema shall be generated. This section includes only the map related to data structure and semantic documentation. Meta-data and context information are described in the next section; and a full example profile BOD is provided in Section 9.

## ASBIEP corresponding to the OAGIS BOD

ASBIEP corresponding to the OAGIS BOD is generated as a XML Schema global element in a profile BOD XML schema. ASBIEP’s GUID is generated as the ‘id’ attribute of the global element.

Lemma : An ASBIEP is derived from ASCCP. An ASCCP is imported from a global or local element declaration that has a complex content in the OAGIS Model XML schema. An ASBIEP uses an ABIE (only DataArea is a local element in the OAGIS Model XML schema).

## ABIE

An ABIE is generated as an XML Schema anonymous complex type in a profile BOD XML schema. The anonymous complex type is a child of an ASBIEP schema element that uses the ABIE.

Lemma : An ABIE is derived from an ACC. An ACC is imported from a global complex type declaration with complex content in the OAGIS Model XML schema.

## ASBIE and ASBIEP not corresponding to the OAGIS BOD

An ASBIE and the ASBIEP it associates to are together generated as an XML Schema local element in a profile BOD XML schema. The local element is a child of the XML Schema sequence under an ABIE complex type.

Lemma : An ASBIE is derived from an ASCC. An ASCC is an association from an ACC to an ASCCP. An ASCC is imported from an XML Schema element reference or local element within an ACC in the OAGIS Model XML schema.

Corollary 3.1: An ASBIE is an association from an ABIE to an ASBIEP.

## BBIE with the ‘element’ entity type and its BBIEP

A BBIE with the ‘element’ entity type and the BBIEP it associates to are together generated as an XML Schema local element in a profile BOD XML schema. The local element is a child of the XML Schema sequence under an ABIE complex type.

Lemma : A BBIE is derived from a BCC which may have an entity type ‘attribute’ or ‘element’. A BBIEP is derived from a BCCP. A BCC is an association from an ACC to a BCCP. A BCCP is imported from an XML Schema global element declaration that has a simple content in the OAGIS Model XML schema or an XML Schema attribute, which is not characterized as an SC (i.e., an attribute of an ACC XML Schema type definition). A BCC is imported from an XML Schema element reference to a BCCP within an ACC.

Corollary 4.1: A BBIE is an association from an ABIE to a BBIEP.

Lemma : A BCCP uses a BDT.

Corollary 5.1: A BBIEP also uses a BDT.

## BBIE with the ‘attribute’ entity type and its BBIEP

A BBIE with the ‘attribute’ entity type and the BBIEP it associates to are together generated as an XML Schema attribute declaration in a profile BOD XML schema. The attribute is a child of the XML Schema complex type (an ABIE). See related lemma and corollary in 7.4.

## BDT (DT table) and BBIE\_SC

Because of Corollary 5.1, a BDT is normally generated as a XML Schema complex type with simple content or not generated at all when there is no supplementary component (however, core component documentation can be generated as part of the BBIE annotation). In the former case, the simple content extends the XML Schema built-in type, OAGIS built-in type, code list type, or agency ID list type per the BBIE’s BDT\_PRI\_RESTR\_ID, CODE\_LIST\_ID, or AGENCY\_ID\_LIST\_ID column. In the latter case, the BBIEP XML Schema element directly uses the XML Schema built-in type, OAGIS built-in type, code list type, or agency ID list type directly. The OAGIS built-in types, code list type, and agency ID list type are declared globally as indicated in section 7.7 and 7.8, respectively.

A BBIE\_SC, if any, is generated as an XML Schema attribute of the BDT’s complex type. Since an SC must always have a simple content, the XML Schema attribute directly uses an XML Schema built-in type, OAGIS built-in type, code list type, or agency ID list type.

Lemma : Per CCS, an SC is part of a BDT. However, in the SRT implementation the BDTs are CC artifacts and are not reflected explicitly as a BIE artifact. That is, a primitive restriction (which can be a built-in type, code list, or agency ID list) is applied directly in the BBIE and the BBIE\_SC tables (as opposed to creating another DT record). For this reason, the based type of or the type used by the BBIE and BBIE\_SC should be fetched from the primitive restriction column captured in the BBIE and BBIE\_SC table.

## OAGIS Built-in Type (XBT table)

XBT table stores both the XML Schema built-in type and OAGIS built-in type definitions. Only OAGIS built-in types used in the profile BOD are generated in the profile BOD XML schema. They are generated as the XML Schema simple type definitions at the global level. The definition is taken from the XBT’s SCHEMA\_DEFINITION column.

## Code List and Agency ID List

Code list as well as agency ID list is simply represented as one, global simple type restriction based on the xsd:token with all the enumerated values. See an example of code list representation in section 5.5. An agency ID list is represented in the same way.

The name of the code list shall be the concatenation of the ‘cl’, AGENCY\_ID, ‘\_’, VERSION\_ID, ‘\_’, NAME, ‘ContentType’, ‘\_’, LIST\_ID. If NAME is empty, remove the ‘\_’ in its front. Also, remove whitespaces in the NAME to make it camel case (LIST\_ID and VERSION\_ID columns shall not have space).

The name of the agency ID list shall be the concatenation of the ‘il’, AGENCY\_ID\_LIST\_VALUE, VERSION\_ID, ‘\_’, NAME, ‘ContentType’, ‘\_’, LIST\_ID. Note that the first token, AGENCY\_ID\_LIST\_VALUE, is the de-reference of the AGENCY\_ID\_LIST\_VALUE\_ID column to VALUE column in the AGENCY\_ID\_LIST\_VALUE table. This token indicates the owner agency of the agency ID list. If NAME is empty, remove the ‘\_’ in its front. Also, remove whitespaces in the NAME column value to make it camel case (LIST\_ID and VERSION\_ID columns shall not have space).

## Others

Other XML Schema features such as ‘default’, ‘fix’, ‘minOccurs’, ‘maxOccurs’, etc. are instantiated according to the data in the corresponding tables identified above. The mappings between the column names and these XML Schema features are straightforward so their details are omitted in this document.

## Documentation

A profile BOD schema can contain two types of documentations. First is the schema element, type, and attribute semantic definitions, which map to the information in the Definition table in the OAGIS Repository; and the other is meta-data documentation which includes CCS and OAGi-specific meta-data. Generation of these documentations are optional (the SRT will provide various options associated with the generation of documentations). This section describes how the semantic definitions shall be generated. Section 8 provides details about the meta-data documentation.

Because anonymous types and local elements are used, there can be semantic documentations from more than one OAGIS Repository entities associated with a schema construct. The order of semantic documentations shall be generated as follows – ASBIE, ASBIEP, ABIE, BBIE, BBIEP, BDT, BBIE\_SC, each in a separate xsd:documentation element.

The documentation of an ASBIEP shall be generated under the corresponding XML Schema element’s annotation.

The documentation of an ASBIE shall be generated under the corresponding XML Schema element’s annotation.

The documentation of an ABIE shall be generated under the corresponding XML Schema complex type’s annotation.

The documentation of a BBIE shall be generated under the corresponding XML Schema element’s or attribute’s annotation.

The documentation of a BBIEP shall be generated under the corresponding XML Schema element’s or attribute’s annotation.

The documentation of a BBIE\_SC shall be generated under the corresponding XML Schema attribute’s annotation.

The documentation of a BDT shall be generated under the corresponding XML Schema element’s, attribute’s, complex type, or simple type annotation.

## Usage Rule

Usage rules will be generated as the XML Schema app info. It is included in the CCS meta-data, and hence described in section 8.

# Core Component and OAGi-specific Documentations

Core component and OAGi-specific documentations are optional documentations. If they are generated, they shall be included in a separate xsd:documentation or xsd:appinfo from the XML Schema element, type, and attribute semantic documentation (a.k.a. the Definition column in the OAGIS Repository). The element representing core component documentation will be prefixed ccts\_. The element representing OAGi-specific documentation will be prefixed oag\_. OAGi-specific documentation are meta-data not included in (i.e., extending) the core component specification such as the WHO columns.

Subsections below provide more details about how these documentations shall be generated. In particular, section 8.3 provides a type definition, CCTSAndOAGIS\_BIEDocumentationType, for capturing these documentations. The detail in each subsection below describes how the type definition shall be used.

## CCTSAndOAGIS\_BIEDocumentationType Overview

The figure below shows the top-level content in the type definition. It should be noted that the ‘\_’ is also used for separating 2 consecutive acronyms (which are in all-cap) in addition to indicating the ccts\_ and oag\_ prefix.

The data elements in the CCTSAndOAGIS\_BIEDocumentationType shall be instantiated under an XML Schema documentation (xsd:documentation) associated with the corresponding BIE entity.

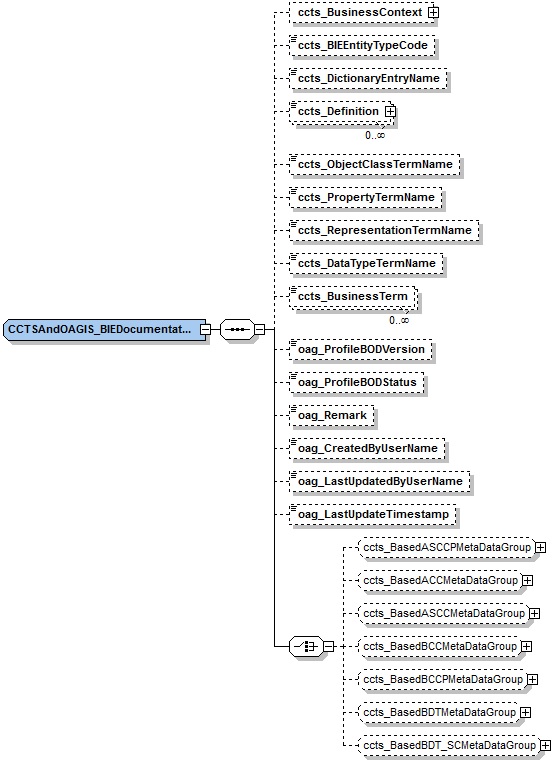


Figure : Top-level content of CCTSAndOAGIS\_BIEDocumentationType

## Business context information

Business Context information is exported from a record in the BIZ\_CTX table and related records in the CTX\_CATEGORY, CTX\_SCHEME, CTX\_SCHEME\_VALUE tables. It describes the usage context, to which the profile BOD is applicable.

The business context information is captured with the ccts\_BusinessContext element, the first element in CCTS\_AndOAGIS\_BIEDocumentationType shown in Figure 1. Figure 2 below gives an overview of the content inside the ccts\_BusinessContext element.

The ccts\_BusinessContext element shall be used only within the root profile BOD XML Schema element (the ASBIEP).

See section 8.4 for detail explanation about each of the elements in ccts\_BusinessContext.

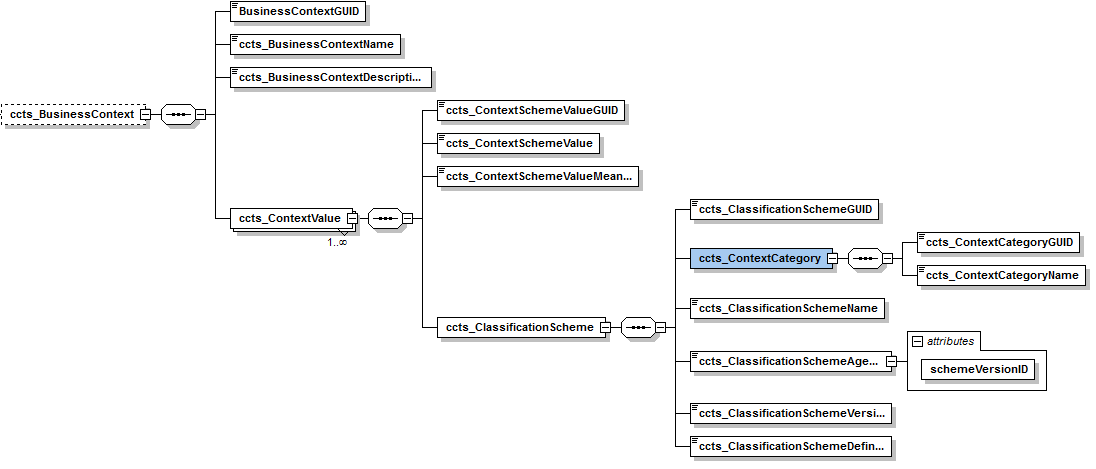


Figure : Content of the ccts\_BusinessContext element

## BIE Documentation

BIE documentation is grouped into two parts. The first part includes the intrinsic information of the BIE. The other part includes information about the CC on which the BIE is based. The first part is represented by a set of elements highlighted in the blue box in Figure 1, while the second part is represented by the XML Schema groups highlighted in the orange box. These XML Schema groups are mutually exclusive; and the specific group to use corresponds to the value in the BIEEntityTypeCode element. This is clearly documented in the normative schema definition in 8.4. Figure 3 below illustrates the content of ccts\_BasedASCCPMetaDataGroup as an example. Definitions of other groups follow this same 3-element-set pattern.

Data for populating the BIE documentation elements described in the previous paragraph are drawn directly from the corresponding BIE tables in the SRT database. The mapping is straightforward for readers who are knowledgeable about the CCS meta-model and hence is not described in detail here.

It should be noted that the ccts\_Definition element captures the same semantic documentation described in 7.10. The purposes of this duplication are 1) the semantic documentation is available even when the user chooses not to generate the CCS meta-data, and 2) the ccts\_Definition provides an unambiguous association between the semantic documentation and a particular BIE entity, allowing for profile import into the SRT.

## Usage Rule

## XML Schema type definition

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning" elementFormDefault="qualified" attributeFormDefault="unqualified" vc:minVersion="1.1">

<xsd:complexType name="CCTSAndOAGIS\_BIEDocumentationType">

<xsd:sequence>

<xsd:element name="ccts\_BusinessContext" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Convey the business context information associated with the BIE. The business context information represents the situation in which the BIE should be used.</xsd:documentation>

</xsd:annotation>

<xsd:complexType>

<xsd:sequence>

<xsd:element name="BusinessContextGUID" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of the business context. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BusinessContextName" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>A concise (oftentime mnemonic) label of a business context</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BusinessContextDescription" type="xsd:string">

<xsd:annotation>

<xsd:documentation>Explanation describing the business context primarily for human consumption</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ContextValue" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Each context value element formally indicates a context in a particular context dimension (category) that makes up a business context. Context values in different context categories are interpreted as having logical AND relationshipasamong them, while context values in the same context category are interpreted as having logical OR relationships among them.</xsd:documentation>

</xsd:annotation>

<xsd:complexType>

<xsd:sequence>

<xsd:element name="ccts\_ContextSchemeValueGUID" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of the context scheme value. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity. </xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ContextSchemeValue" type="xsd:token">

<xsd:annotation>

<xsd:documentation>The context value from the specified context scheme below.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ContextSchemeValueMeaning" type="xsd:string">

<xsd:annotation>

<xsd:documentation>The meaning of the context value primarily for human consumption</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ClassificationScheme">

<xsd:annotation>

<xsd:documentation>This entiity describes the classification scheme the context value comes from.</xsd:documentation>

</xsd:annotation>

<xsd:complexType>

<xsd:sequence>

<xsd:element name="ccts\_ClassificationSchemeGUID" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of the classification scheme. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ContextCategory">

<xsd:annotation>

<xsd:documentation>Context contegory or dimension the classification scheme is used for describing. For example, industry context category, role context category.</xsd:documentation>

</xsd:annotation>

<xsd:complexType>

<xsd:sequence>

<xsd:element name="ccts\_ContextCategoryGUID" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of the context category. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ContextCategoryName" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>A concise (oftentime mnemonic) label of the context category</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="ccts\_ClassificationSchemeName" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>A concise (oftentime mnemonic) label of the classification scheme</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ClassificationSchemeAgencyID">

<xsd:annotation>

<xsd:documentation>Identifier of the organization that maintains the classification scheme. The ID shall be assumed to be coming from UN/CEFACT 3055 ID list. If the ID is not in the 3055, the ID is assumed to be a company's specific ID. In other words, if the ID is 3055 list, it shall be interpreted per 3055; and if not, it is a company's specific ID understood by the company and its partners.</xsd:documentation>

</xsd:annotation>

<xsd:complexType>

<xsd:simpleContent>

<xsd:extension base="xsd:normalizedString">

<xsd:attribute name="schemeVersionID" type="xsd:normalizedString" use="required">

<xsd:annotation>

<xsd:documentation>Version ID of the Agency ID code list. This shall be the version of the Agency ID used in the SRT.</xsd:documentation>

</xsd:annotation>

</xsd:attribute>

</xsd:extension>

</xsd:simpleContent>

</xsd:complexType>

</xsd:element>

<xsd:element name="ccts\_ClassificationSchemeVersionID" type="xsd:normalizedString">

<xsd:annotation>

<xsd:documentation>Version ID of the classification scheme</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_ClassificationSchemeDefinition" type="xsd:string">

<xsd:annotation>

<xsd:documentation>Description of the classification scheme for human consumption.</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="ccts\_BIEEntityTypeCode" minOccurs="0">

<xsd:annotation>

<xsd:documentation>This indicates the type of BIE this documentation is capturing. For example, if the code is ABIE, it means that all the information, e.g., Dictionary Entry Name, is about an ABIE. If the code is ASBIEP, then only the elements in the ccts\_BasedASCCPMetaDataGroup shall be specified. If the code is ABIE, then only the elements in the ccts\_BasedACCMetaDataGroup shall be specified. If the code is ASBIE, then only the elements in the ccts\_BasedASCCMetaDataGroup shall be specified. If the code is BBIEP, then only the elements in the ccts\_BasedBCCPMetaDataGroup shall be specified. If the code is BBIE, then only the elements in the ccts\_BasedBCCMetaDataGroup shall be specified. If the code is BDT, then only the elements in the ccts\_BasedBDTMetaDataGroup shall be specified. If the code is SC, then only the elements in the ccts\_BasedBDT\_SCMetaDataGroup shall be specified. </xsd:documentation>

</xsd:annotation>

<xsd:simpleType>

<xsd:restriction base="xsd:token">

<xsd:enumeration value="ABIE"/>

<xsd:enumeration value="ASBIE"/>

<xsd:enumeration value="ASIEP"/>

<xsd:enumeration value="BBIE"/>

<xsd:enumeration value="BBIEP"/>

<xsd:enumeration value="BDT"/>

<xsd:enumeration value="SC">

<xsd:annotation>

<xsd:documentation>Supplementary Component</xsd:documentation>

</xsd:annotation>

</xsd:enumeration>

</xsd:restriction>

</xsd:simpleType>

</xsd:element>

<xsd:element name="ccts\_DictionaryEntryName" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Dictionary Entry Name (DEN) of the BIE entity being documented. At this point only one occurrence is supported and the DEN is assumed to be in us-en. </xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_Definition" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>This capture the semantic documentations associated with the BIE entity. The xml:lang, if not populated, is assumed to be us-en.</xsd:documentation>

</xsd:annotation>

<xsd:complexType>

<xsd:simpleContent>

<xsd:extension base="xsd:string">

<xsd:attribute name="source" type="xsd:anyURI"/>

</xsd:extension>

</xsd:simpleContent>

</xsd:complexType>

</xsd:element>

<xsd:element name="ccts\_ObjectClassTermName" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Object class term of the BIE entity being documented. This element is used only when the BIEEntityTypeCode is ABIE, ASBIE, or BBIE.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_PropertyTermName" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Property term of the BIE entity being documented. This element is used only when the BIEEntityTypeCode is ASBIEP, BBIEP, ASBIE, BBIE, or SC.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_RepresentationTermName" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Representation term of the BIE entity being documented. This element is used only when the BIEEntityTypeCode is ASBIEP, BBIEP, ASBIE, BBIE, BDT, or SC.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_DataTypeTermName" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Data type term of the BIE entity being documented. This element is used only when the BIEEntityTypeCode is BDT.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BusinessTerm" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Alternate names of the BIE. The multiple cardinality is for supporting multiple languages. Currently the SRT supports only one BusinessTerm per one BIE entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="oag\_ProfileBODVersion" type="xsd:token" minOccurs="0">

<xsd:annotation>

<xsd:documentation>This represents the user specified version of the profile BOD generated. It is used only in the root element of the profile BOD.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="oag\_ProfileBODStatus" type="xsd:token" minOccurs="0">

<xsd:annotation>

<xsd:documentation>This represents the user speciified status of the profile BOD such as Draft, Test, Production. It is used only in the root element of the profile BOD.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="oag\_Remark" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Remark is a non-normative documentation about the BIE entity. The user may use this, for example, to make note about what to think about, what to do, or other things still uncertained about the BIE entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="oag\_CreatedByUserName" type="xsd:token" minOccurs="0">

<xsd:annotation>

<xsd:documentation>This is a track-n-trace field indicating the original user who created the BIE entity (in the SRT).</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="oag\_LastUpdatedByUserName" type="xsd:token" minOccurs="0">

<xsd:annotation>

<xsd:documentation>This is a track-n-trace field indicating the user who last updated the BIE entity (in the SRT).</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="oag\_LastUpdateTimestamp" type="xsd:dateTime" minOccurs="0">

<xsd:annotation>

<xsd:documentation>This is a track-n-trace field indicating the datetime the BIE entity was last updated.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:choice>

<xsd:group ref="ccts\_BasedASCCPMetaDataGroup" minOccurs="0"/>

<xsd:group ref="ccts\_BasedACCMetaDataGroup" minOccurs="0"/>

<xsd:group ref="ccts\_BasedASCCMetaDataGroup" minOccurs="0"/>

<xsd:group ref="ccts\_BasedBCCMetaDataGroup" minOccurs="0"/>

<xsd:group ref="ccts\_BasedBCCPMetaDataGroup" minOccurs="0"/>

<xsd:group ref="ccts\_BasedBDTMetaDataGroup" minOccurs="0"/>

<xsd:group ref="ccts\_BasedBDT\_SCMetaDataGroup" minOccurs="0"/>

</xsd:choice>

</xsd:sequence>

</xsd:complexType>

<xsd:group name="ccts\_BasedASCCPMetaDataGroup">

<xsd:annotation>

<xsd:documentation>Group of elements used for identifying the ASCCP on which an ASBIEP is based. The GUID and Revision Number together uniquely identify the ASCCP record in the SRT database.</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

<xsd:element name="ccts\_BasedASCCP\_GUID" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of an ASCCP. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedASCCPRevisionNumber" type="xsd:positiveInteger" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Revision number of the ASCCP</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedASCCPDefinition" type="xsd:string" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Description of the ASCCP</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:group>

<xsd:group name="ccts\_BasedACCMetaDataGroup">

<xsd:annotation>

<xsd:documentation>Group of elements used for identifying the ACC on which an ABIE is based. The GUID and Revision Number together uniquely identify the ACC record in the SRT database.</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

<xsd:element name="ccts\_BasedACC\_GUID" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of an ACC. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedACCRevisionNumber" type="xsd:positiveInteger" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Revision number of the ACC</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedACCDefinition" type="xsd:string" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Description of the ACC</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:group>

<xsd:group name="ccts\_BasedASCCMetaDataGroup">

<xsd:annotation>

<xsd:documentation>Group of elements used for identifying the ASCC on which an ASBIE is based. The GUID and Revision Number together uniquely identify the ASCC record in the SRT database.</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

<xsd:element name="ccts\_BasedASCC\_GUID" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of an ASCC. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedASCCRevisionNumber" type="xsd:positiveInteger" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Revision number of the ASCC</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedASCCDefinition" type="xsd:string" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Description of the ASCC</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:group>

<xsd:group name="ccts\_BasedBCCPMetaDataGroup">

<xsd:annotation>

<xsd:documentation>Group of elements used for identifying the BCCP on which a BBIEP is based. The GUID and Revision Number together uniquely identify the BCCP record in the SRT database.</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

<xsd:element name="ccts\_BasedBCCP\_GUID" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of a BCCP. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBCCPRevisionNumber" type="xsd:positiveInteger" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Revision number of the BCCP</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBCCPDefinition" type="xsd:string" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Description of the BCCP</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:group>

<xsd:group name="ccts\_BasedBCCMetaDataGroup">

<xsd:annotation>

<xsd:documentation>Group of elements used for identifying the BCC on which a BBIE is based. The GUID and Revision Number together uniquely identify the BCC record in the SRT database.</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

<xsd:element name="ccts\_BasedBCC\_GUID" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of a BCC. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBCCRevisionNumber" type="xsd:positiveInteger" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Revision number of the BCC</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBCCDefinition" type="xsd:string" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Description of the BCC</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:group>

<xsd:group name="ccts\_BasedBDTMetaDataGroup">

<xsd:annotation>

<xsd:documentation>Group of elements used for identifying the BDT used by the BCCP, on which a BBIEP is based. The BDT maybe further restricted on the BIE side. The GUID and Revision Number together uniquely identify the BDT record in the SRT database.</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

<xsd:element name="ccts\_BasedBDT\_GUID" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of a BDT. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBDT\_RevisionNumber" type="xsd:positiveInteger" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Revision number of the BDT</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBDT\_Definition" type="xsd:string" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Description of the BDT</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:group>

<xsd:group name="ccts\_BasedBDT\_SCMetaDataGroup">

<xsd:annotation>

<xsd:documentation>Group of elements used for identifying the SC of the BDT used by the BCCP, on which a BBIEP is based. The SC maybe further restricted on the BIE side. The GUID and Revision Number together uniquely identify the BDT SC record in the SRT database.</xsd:documentation>

</xsd:annotation>

<xsd:sequence>

<xsd:element name="ccts\_BasedBDT\_SC\_GUID" type="xsd:normalizedString" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Globally unique identifier (GUID) of an SC of a BDT. A GUID of an entity never changes so that while other parts of the entity may evolve over time, it can be recognized as the same entity.</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBDT\_SCRevisionNumber" type="xsd:positiveInteger" minOccurs="0">

<xsd:annotation>

<xsd:documentation>Revision number of the BDT's SC</xsd:documentation>

</xsd:annotation>

</xsd:element>

<xsd:element name="ccts\_BasedBDT\_SCDefinition" type="xsd:string" minOccurs="0" maxOccurs="unbounded">

<xsd:annotation>

<xsd:documentation>Description of the BDT's SC</xsd:documentation>

</xsd:annotation>

</xsd:element>

</xsd:sequence>

</xsd:group>

</xsd:schema>

# Appendix A: Full Profile BOD Schema Example

1. http://www.oracle.com/technetwork/java/design-patterns-142138.html [↑](#footnote-ref-1)