

CCSDS Registry Information Model

CCSDS Registry Information Model Team

September 17, 2008

Draft White Book

Version 0.080303

Contents

1	Introduction	5
1.1	Background	5
1.2	Scope	5
1.3	Approach	7
1.4	Terminology	8
1.5	Document Contents	9
2	ebXML Object Classes	10
2.1	Action	13
2.2	AdhocQuery	13
2.3	Association	14
2.4	AssociationType	15
2.5	AuditableEvent	16
2.6	Classification	17
2.7	ClassificationNode	19
2.8	ClassificationScheme	21
2.9	ContentInformation	22
2.10	ContentManagementService	23
2.11	DataType	24
2.12	DeletionScopeType	25
2.13	EBXML_Classes	26
2.14	EmailAddress	27
2.15	EmailType	27
2.16	ErrorHandlingModel	28
2.17	ErrorSeverityType	29
2.18	EventType	30
2.19	ExternalIdentifier	31
2.20	ExternalLink	33
2.21	ExtrinsicObject	33
2.22	Federation	34
2.23	Identifiable	35
2.24	InvocationModel	35
2.25	Model	36
2.26	NameSpace	37
2.27	NodeType	38
2.28	Notification	39
2.29	NotificationOptionType	40
2.30	NotifyAction	41
2.31	ObjectRef	42
2.32	ObjectType	42
2.33	Organization	43
2.34	Person	44

2.35	PersonName	45
2.36	PhoneType	46
2.37	PostalAddress	46
2.38	QueryExpression	47
2.39	QueryLanguage	47
2.40	Registry	48
2.41	RegistryObject	49
2.42	RegistryPackage	50
2.43	ResponseStatusType	51
2.44	Service	52
2.45	ServiceBinding	53
2.46	Slot	54
2.47	SpecificationLink	56
2.48	StatusType	56
2.49	SubjectGroup	57
2.50	SubjectRole	58
2.51	Subscription	59
2.52	TelecommandService	60
2.53	TelemetryServices	61
2.54	TelephoneNumber	62
2.55	User	63
2.56	VersionInfo	63
2.57	XMLSchema	65
3	Content Unit Object Classes	66
4	Data Dictionary	67

List of Figures

1	Features of an ebXML Registry	6
2	ebXML Concept Map	6
3	ebXML Information Model	7
4	OAIS Content Information Data Model	8
5	CASPAR Conceptual Architecture Model	9
6	ebXML UML Class Diagram	12
7	Content Unit UML Class Diagram	66

1 Introduction

This document presents a draft data model for the Consultative Committee for Space Data Systems (CCSDS) registries. This draft model is based on the "Electronic Business using the eXtensible Markup Language" (ebXML) Information Model, the "Cultural, Artistic and Scientific knowledge for Preservation, Access and Retrieval" (CASPAR) data model, and the Open Archival Information System (OAIS) Reference Model (RM). In particular, the ebXML information model provides the basic classes for the registry schema and administration and the OAIS RM provide the classes for content information. CASPAR, having implemented an archive registry using components of the OAIS RM provided the example of how to integrate the models.

1.1 Background

The CCSDS has undertaken the writing of a series of CCSDS Recommendations and Reports meant to provide CCSDS registry and repository recommendations to accommodate the current computing environment and meet evolving requirements. The Registry White book proposes a staged set of CCSDS Recommendations for registries and repositories that meet current CCSDS agency requirements and can be implemented to demonstrate practical, near-term results.

At the CCSDS Fall meeting there was significant progress in putting together a plan for defining registry standards within CCSDS. Several agencies have agreed to input information including NASA, CNES and ESA. This document is the second deliverable in that plan, a draft Information Model. The consensus Information Model will be incorporated into the Registry White Book.

1.2 Scope

The scope of application of the information model is the entire space informatics domain from operational messaging to science archives. In recognition of this varied user community, this document proposes aggressive use of current and emerging W3C and Web Services standards to provide advanced data access techniques and adherence to the OAIS Reference Model information model.

In the following figure, a Concept Map of the EBXML Information Model is provided. This concept map illustrates the essential elements of the information model. These elements are developed fuller in the sections below.

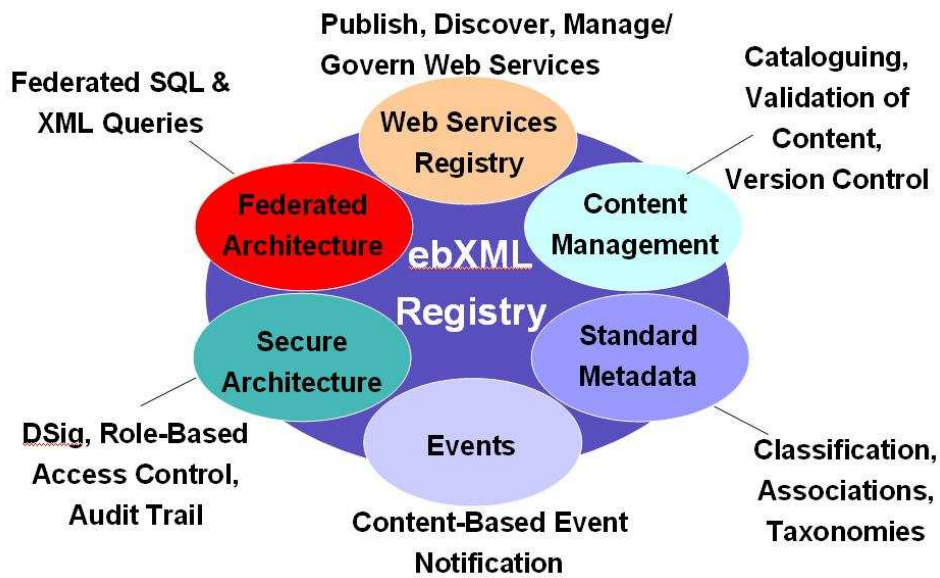


Figure 1: Features of an ebXML Registry

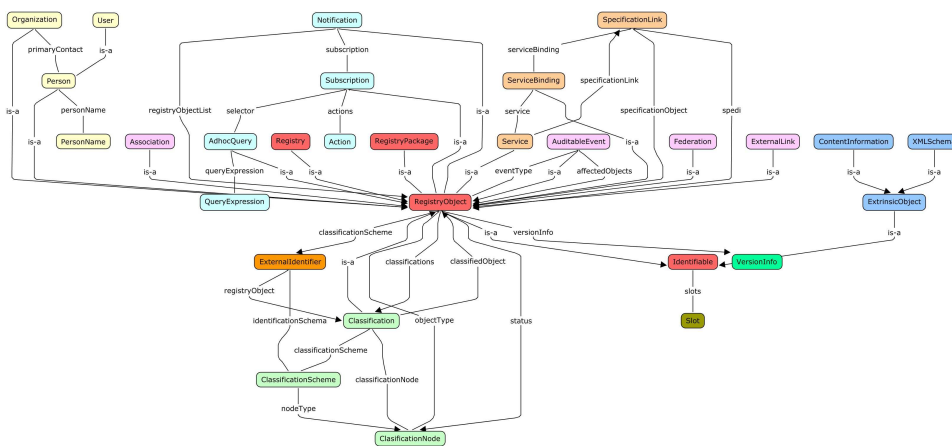


Figure 2: ebXML Concept Map

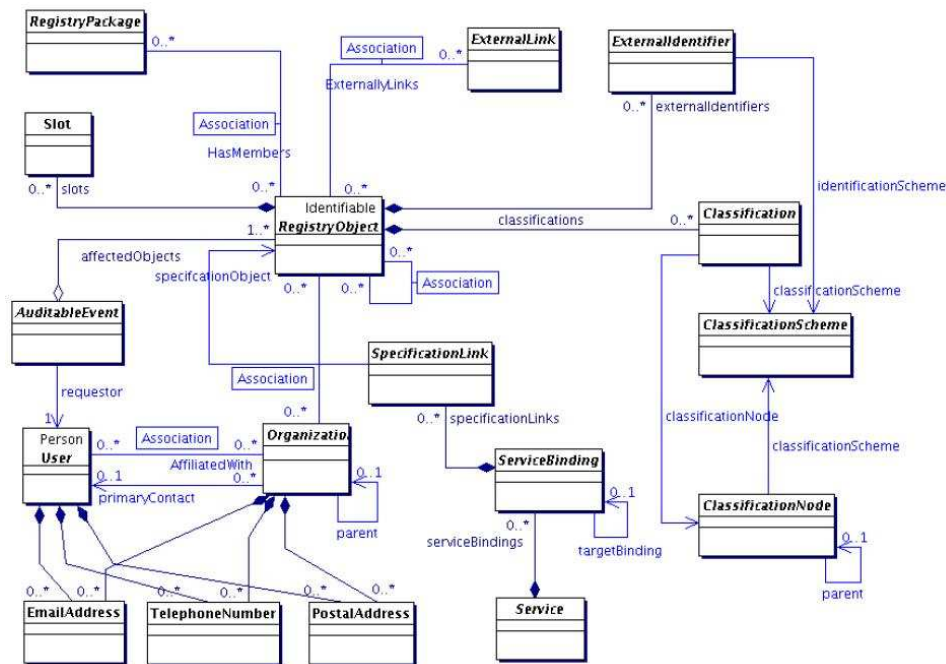


Figure 3: ebXML Information Model

1.3 Approach

The draft data model for the CCSDS registry is primarily based on the ebXML Information Model , Version 3.0.1. A UML class diagram is depicted in Figure 3. An ebXML Registry stores any type of content as RepositoryItems in a repository and stores standardized metadata describing the content as RegistryObjects in a registry. The core classes include the Identifiable and RegistryObject classes.

The Identifiable class is the common super class for most classes in the information model. Information model Classes whose instances have a unique identity are descendants of the Identifiable Class.

The RegistryObject class extends the Identifiable class and serves as a common super class for most classes in the information model. For defining domain specific RegistryObjects, the Extrinsic Object is the primary metadata class for a RepositoryItem. The OAIS RM provides the classes for content information. As shown in Figure 4. The Content Information class is a component of an Archival Information Package. For this draft, only the Content Information class and its components are considered. The Content Information class is equivalent to the OAIS Information Object class. CASPAR provides an example of how to integrate the ebXML and

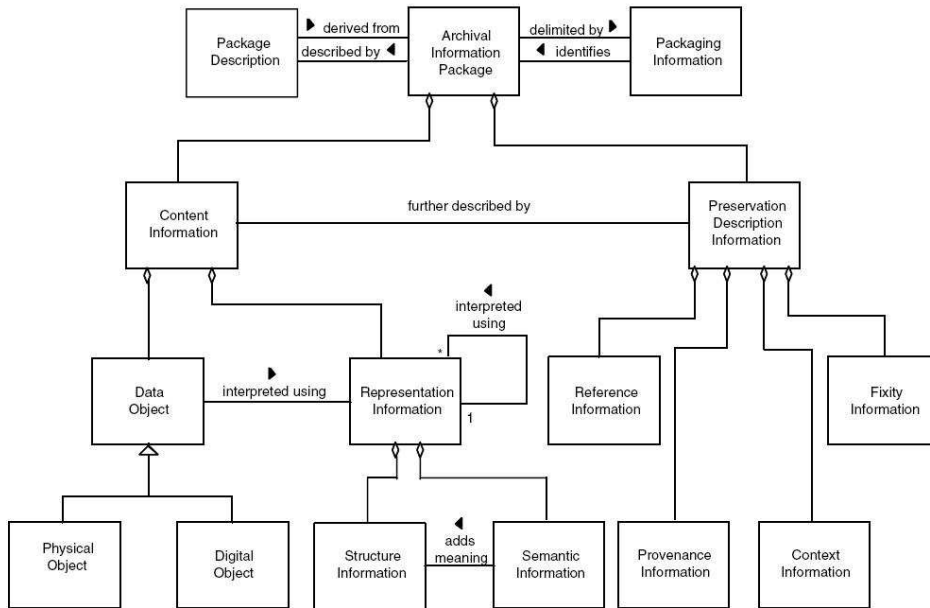


Figure 4: OAIS Content Information Data Model

OAIS models. The CASPAR information model as shown in Figure 5 has the OAIS Content Information as a component of an archive package. In the CCSDS registry model, the OAIS Content Information class is modeled as a subtype of the ebXML ExtrinsicObject. This suggests a model for a data registry such as that used by space science missions. Other registry types can be modeled using additional ExtrinsicObject subclasses such as XMLSchema and Service.

1.4 Terminology

This document uses very specific engineering terminology to describe the various structures involved. Please consult the Glossary for definitions whenever there is any possibility of confusion.

Following are some definitions of essential terms used throughout this document.

An "attribute" is a property or characteristic that allows both identification and distinction.

A "class" is the set of attributes which identifies a family. A class is generic – a template from which individual members of each family may be constructed.

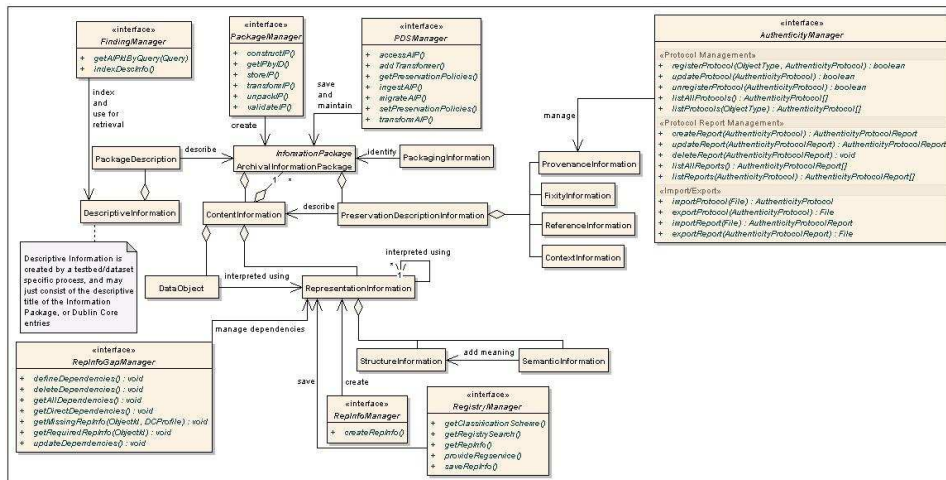


Figure 5: CASPAR Conceptual Architecture Model

An "object" is a specific instance of a class.

An "association" is a defined relationship between classes. It has one direction. The association in the opposite direction is called an inversion relation and is sometimes named by adding a postfix "_I" as in "has_Instrument_I".

"Cardinality" is the number of values allowed to an attribute or association in a single class. Cardinality in general is stated as a range with a minimum and maximum. For example, an attribute that may be multi-valued will have a cardinality of "1..*". A cardinality where the minimum and maximum are the same is often shown as the single value. For example, an attribute required to have exactly one value will have a cardinality of "1". When a value is required the minimum cardinality is at least 1.

"Entity" is a generic term used to refer to specific attributes or associations listed in a class definition.

1.5 Document Contents

Each of the following sections begins with a brief outline, including a hierarchy of the definitions which follow. To facilitate cross-referencing, the classes are listed alphabetically within each section. Subsections begin with a note on the position within the hierarchy and a brief description of the class. The heart of each subsection is the class definition table. Sections are often accompanied by a UML diagram which shows the relationships

among classes graphically.

Class definition tables comprise five columns. The left column is used to separate the table into functional blocks of contiguous rows. The "hierarchy" block restates the position of the class within the definitional hierarchy, and the "subclass" block identifies any subclasses which may exist (be derived from the current class). Attribute and Association blocks list the properties, characteristics, and relationships of the class, some of which may be inherited from parent classes. The "referenced from" block lists classes which may "call" the class being defined.

Within Attribute blocks, the "entity" column lists the properties and characteristics which identify the class and distinguish it from others. The "Indicator" column (far right) tells whether the attribute is optional (O), restricted (R), or both; a restricted attribute has been inherited from a parent class but its use is more narrow than the parent would allow. The "Cardinality" column (middle) shows the number of values allowed. A required attribute for which only one value is allowed will have cardinality "1". A required attribute for which one or more values is allowed will have cardinality "1..*". If a parent's attribute has cardinality "1..*" but the child's cardinality is "1", the Indicator column should show "R". The "Value" column (fourth) includes the indicator Data Dictionary (DD) when a set of valid values for the attribute are provided in the dictionary. A few attributes that represent types have their valid values included in this column.

The Association blocks are handled similarly. The "Entity" column lists relationships among classes using fabricated, but intuitive, names which are unique and consistent across the Specification. The "Value" column (fourth), which is rarely used in the Attribute blocks, lists the class to which the relationship is made.

2 ebXML Object Classes

The upper level object model defines the object classes a CCSDS registry will either manage or use for administration. The concept of content unit describes what a registry will manage. Other classes are used by the registry to support the management of content units and support classification, authorization, and related administration functions. For example, "

The upper level object class hierarchy is illustrated in the following diagram. This diagram presents the subclassOf relation for each object class in a hierarchical (tree) format and provides a visual representation of the object classes in relation to their parent classes. As currently modeled

the upper level class hierarchy is flat however it will become more complex as we continue to develop the model.

```
+ EBXML_Classes
+ + Action
+ + + NotifyAction
+ + EmailAddress
+ + Identifiable
+ + + RegistryObject
+ + + + AdhocQuery
+ + + + Association
+ + + + AuditableEvent
+ + + + Classification
+ + + + ClassificationNode
+ + + + ClassificationScheme
+ + + + + AssociationType
+ + + + + ContentManagementService
+ + + + + DataType
+ + + + + DeletionScopeType
+ + + + + EmailType
+ + + + + ErrorHandlingModel
+ + + + + ErrorSeverityType
+ + + + + EventType
+ + + + + InvocationModel
+ + + + + NodeType
+ + + + + NotificationOptionType
+ + + + + ObjectType
+ + + + + PhoneType
+ + + + + QueryLanguage
+ + + + + ResponseStatusType
+ + + + + StatusType
+ + + + + SubjectGroup
+ + + + + SubjectRole
+ + + + ExternalIdentifier
+ + + + ExternalLink
+ + + + ExtrinsicObject
+ + + + + ContentInformation
+ + + + + Model
+ + + + + NameSpace
+ + + + + TelecommandService
+ + + + + TelemetryServices
+ + + + + XMLSchema
+ + + + Federation
+ + + + Notification
```

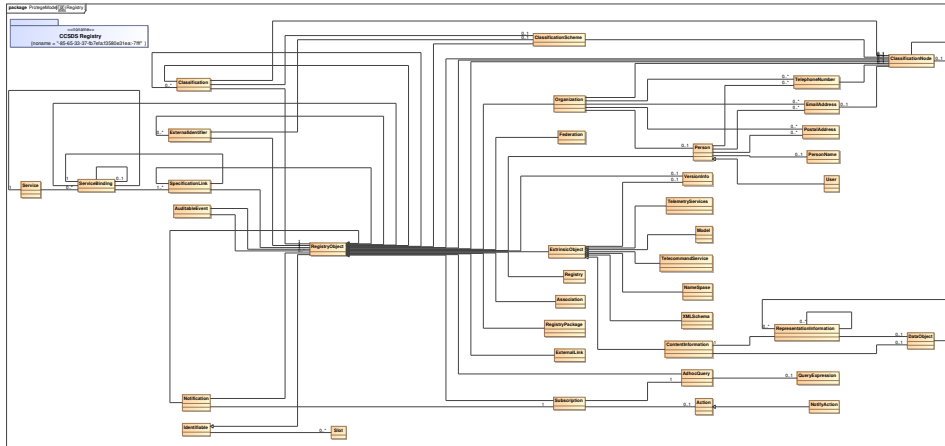


Figure 6: ebXML UML Class Diagram

```

+ + + + Organization
+ + + + Person
+ + + + User
+ + + + Registry
+ + + + RegistryPackage
+ + + + Service
+ + + + ServiceBinding
+ + + + SpecificationLink
+ + + + Subscription
+ + ObjectRef
+ + PersonName
+ + PostalAddress
+ + QueryExpression
+ + Slot
+ + TelephoneNumber
+ + VersionInfo

```

The class hierarchy above includes 57 unique classes.

The upper level object model is illustrated using the UML class hierarchy diagram in Figure 6. The relations between object classes are one directional. Inverse relations are defined when necessary. The following sections present the upper level object classes in a table format. The table includes the class hierarchy, class attributes, and class associations. The class attributes and associations listed include both those used to define the object class and those inherited from parent classes. Cardinalities are provided where appropriate.

2.1 Action

Root Class: EBXML_Classes

Class Description: The Action class is an abstract super class that specifies what the registry must do when an event matching the action's Subscription transpires. A registry uses Actions within a Subscription to asynchronously deliver event Notifications to the subscriber. If no Actions are defined within the Subscription it implies that the user does not wish to be notified asynchronously by the registry and instead intends to periodically poll the registry and pull the pending Notifications.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Action			
Subclass	NotifyAction			
Attribute	none			
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	Subscription			

2.2 AdhocQuery

Root Class: EBXML_Classes

Class Description: The AdhocQuery class is a container for an ad hoc query expressed in a query syntax that is supported

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . AdhocQuery			
Subclass	none			
Attribute	none			
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	queryExpression	0..1	QueryExpression	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	Subscription			

2.3 Association

Root Class: EBXML_Classes

Class Description: Association instances are used to define many-to-many associations among RegistryObjects in the information model. An instance of the Association class represents an association between two RegistryObjects.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Association			
Subclass	none			
Attribute	associationType sourceObject targetObject	1 1 1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.4 AssociationType

Root Class: EBXML_Classes

Class Description: Defines the types of associations between Registry-Objects.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme AssociationType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.5 AuditableEvent

Root Class: EBXML_Classes

Class Description: AuditableEvent instances provide a long-term record of events that effected a change in a RegistryObject. A RegistryObject is associated with an ordered Set of AuditableEvent instances that provide a complete audit trail for that RegistryObject. AuditableEvents are usually a result of a client-initiated request. AuditableEvent instances are generated by the Registry Service to log such Events. Often such events effect a change in the life cycle of a RegistryObject. For example a client request could Create, Update, Deprecate or Delete a RegistryObject. An AuditableEvent is typically created when a request creates or alters the content or ownership of a RegistryObject. Read-only requests typically do not generate an AuditableEvent.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . AuditableEvent			
Subclass	none			
Attribute	requestId timestamp	1 1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	affectedObjects eventType user	1 1 1..*	RegistryObject RegistryObject	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.6 Classification

Root Class: EBXML_Classes

Class Description: A Classification instance classifies a RegistryObject instance by referencing a node defined within a particular Classification-Scheme. An internal Classification will always reference the node directly, by its id, while an external Classification will reference the node indirectly by specifying a representation of its value that is unique within the external classification scheme. The attributes for the Classification class are intended to allow for representation of both internal and external classifications in order to minimize the need for a submission or a query to distinguish between internal and external classifications.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Classification			
Subclass	none			
Attribute	nodeRepresentation	0..1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	classificationNode classificationScheme classifiedObject	0..1 0..1 1	ClassificationNode ClassificationScheme RegistryObject	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	AdhocQuery Association AssociationType AuditableEvent Classification ClassificationNode ClassificationScheme ContentInformation ContentManagementService DataType DeletionScopeType EmailType ErrorHandlingModel ErrorSeverityType EventType ExternalIdentifier ExternalLink ExtrinsicObject Federation InvocationModel Model NameSpase NodeType Notification NotificationOptionType ObjectType Organization Person PhoneType QueryLanguage Registry RegistryObject			

2.7 ClassificationNode

Root Class: EBXML_Classes

Class Description: ClassificationNode instances are used to define tree structures where each node in the tree is a ClassificationNode. Such ClassificationScheme trees are constructed with ClassificationNode instances under a ClassificationScheme instance, and are used to define Classification schemes or ontologies.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationNode			
Subclass	none			
Attribute	code path	0..1 0..1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	parent	0..1	ClassificationNode	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	AdhocQuery Association AssociationType AuditableEvent Classification ClassificationNode ClassificationScheme ContentInformation ContentManagementService DataType DeletionScopeType EmailAddress EmailType ErrorHandlingModel ErrorSeverityType EventType ExternalIdentifier ExternalLink ExtrinsicObject Federation InvocationModel Model NameSpase NodeType Notification NotificationOptionType ObjectType Organization Person PhoneType QueryLanguage Registry RegistryObject			

2.8 ClassificationScheme

Root Class: EBXML_Classes

Class Description: A ClassificationScheme instance describes a taxonomy. The taxonomy hierarchy may be defined internally to the registry by instances of ClassificationNode, or it may be defined externally to the Registry, in which case the structure and values of the taxonomy elements are not known to the Registry. In the first case the classification scheme is said to be internal and in the second case the classification scheme is said to be external.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme			
Subclass	NotificationOptionType EmailType SubjectRole ErrorSeverityType ErrorHandlingModel DataType EventType PhoneType ResponseStatusType ObjectType QueryLanguage DeletionScopeType StatusType InvocationModel AssociationType NodeType SubjectGroup ContentManagementService			
Attribute	isInternal	1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	nodeType	0..1	ClassificationNode	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	Classification ExternalIdentifier			

2.9 ContentInformation

Root Class: EBXML_Classes

Class Description: OAIS Content Information: The set of information that is the original target of preservation. It is an Information Object

comprised of its Content Data Object and its Representation Information. An example of Content Information could be a single table of numbers representing, and understandable as, temperatures, but excluding the documentation that would explain its history and origin, how it relates to other observations, etc.

	Entity	Card	Value/Class	In
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExtrinsicObject ContentInformation			
Subclass	none			
Attribute	URI	0..1		
Inherited Attribute	isOpaque mimeType home id description lid name	0..1 0..1 0..1 1 0..1 0..1 0..1		
Association	comprised_of_data_object comprised_of_representation...	0..1 1	DataObject RepresentationInformation	
Inherited Association	contentVersionInfo slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	VersionInfo Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.10 ContentManagementService

Root Class: EBXML_Classes

Class Description: Defines the types of content management services.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme ContentManagementService			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.11 DataType

Root Class: EBXML_Classes

Class Description: Defines the data types for attributes in classes defined by this document.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme DataType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.12 DeletionScopeType

Root Class: EBXML_Classes

Class Description: Defines the values for the deletionScope attribute in RemoveObjectsRequest protocol message.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme DeletionScopeType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.13 EBXML_Classes

Root Class: EBXML_Classes

Class Description: OASIS/ebXML Registry Information Model

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes			
Subclass	Action VersionInfo PersonName Slot QueryExpression ObjectRef PostalAddress TelephoneNumber EmailAddress Identifiable			
Attribute	none			
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	none			

2.14 EmailAddress

Root Class: EBXML_Classes

Class Description: This class defines attributes of an email address.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . EmailAddress			
Subclass	none			
Attribute	address	1		
Inherited Attribute	none			
Association	type	0..1	ClassificationNode	
Inherited Association	none			
Referenced from	Organization Person User			

2.15 EmailType

Root Class: EBXML_Classes

Class Description: Defines the types of email addresses.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme EmailType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.16 ErrorHandlingModel

Root Class: EBXML_Classes

Class Description: Defines the types of error handling models for content management services.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme ErrorHandlingModel			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.17 ErrorSeverityType

Root Class: EBXML_Classes

Class Description: Defines the different error severity types encountered by registry during processing of protocol messages.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme ErrorSeverityType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.18 EventType

Root Class: EBXML_Classes

Class Description: Defines the types of events that can occur in a registry.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme EventType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.19 ExternalIdentifier

Root Class: EBXML_Classes

Class Description: ExternalIdentifier instances provide the additional identifier information to RegistryObject such as DUNS number, Social Security Number, or an alias name of the organization. The attribute identificationScheme is used to reference the identification scheme (e.g., "DUNS", "Social Security #"), and the attribute value contains the actual information (e.g., the DUNS number, the social security number). Each RegistryObject MAY contain 0 or more ExternalIdentifier instances.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExternalIdentifier			
Subclass	none			
Attribute	value	1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	IdentificationScheme registryObject	0..1 1	ClassificationScheme RegistryObject	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	AdhocQuery Association AssociationType AuditableEvent Classification ClassificationNode ClassificationScheme ContentInformation ContentManagementService DataType DeletionScopeType EmailType ErrorHandlingModel ErrorSeverityType EventType ExternalIdentifier ExternalLink ExtrinsicObject Federation InvocationModel Model NameSpace NodeType Notification NotificationOptionType ObjectType Organization Person PhoneType QueryLanguage Registry RegistryObject RegistryPackage			

2.20 ExternalLink

Root Class: EBXML_Classes

Class Description: ExternalLinks use URIs to associate content in the registry with content that MAY reside outside the registry. For example, an organization submitting an XML Schema could use an ExternalLink to associate the XML Schema with the organization's home page.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExternalLink			
Subclass	none			
Attribute	externalURI	1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.21 ExtrinsicObject

Root Class: EBXML_Classes

Class Description: The ExtrinsicObject class is the primary metadata class for a RepositoryItem.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExtrinsicObject			
Subclass	TelecommandService Model TelemetryServices XMLSchema ContentInformation NameSpase			
Attribute	isOpaque mimeType	0..1 0..1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	contentVersionInfo	0..1	VersionInfo	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.22 Federation

Root Class: EBXML_Classes

Class Description: Federation instances are used to represent a registry federation.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Federation			
Subclass	none			
Attribute	replicationSyncLatency	0..1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.23 Identifiable

Root Class: EBXML_Classes

Class Description: The Identifiable class is the common super class for most classes in the information model. Information model Classes whose instances have a unique identity are descendants of the Identifiable Class.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable			
Subclass	RegistryObject			
Attribute	home id	0..1 1		
Inherited Attribute	none			
Association	slots	0..*	Slot	
Inherited Association	none			
Referenced from	none			

2.24 InvocationModel

Root Class: EBXML_Classes

Class Description: Defines the different ways that a content management service may be invoked by the registry.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme InvocationModel			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.25 Model

Root Class: EBXML_Classes

Class Description: The Deep Space Network Information Service Architecture (DISA) is a set of information services and information models to enable the Deep Space Network and Advance Multi-mission Operation System (AMMOS) to become a service-oriented architecture. As such, DISA has identified several services needed to support movement towards a SOA. A Registry Service is one such service that has identified needs for managing models, schemas, services, elements, and namespaces.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExtrinsicObject Model			
Subclass	none			
Attribute	none			
Inherited Attribute	isOpaque mimeType home id description lid name	0..1 0..1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	contentVersionInfo slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	VersionInfo Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.26 Namespace

Root Class: EBXML_Classes

Class Description: The Deep Space Network Information Service Architecture (DISA) is a set of information services and information models to enable the Deep Space Network and Advance Multi-mission Operation System (AMMOS) to become a service-oriented architecture. As such, DISA has identified several services needed to support movement towards a SOA. A Registry Service is one such service that has identified needs for managing models, schemas, services, elements, and namespaces.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExtrinsicObject NameSpace			
Subclass	none			
Attribute	none			
Inherited Attribute	isOpaque mimeType home id description lid name	0..1 0..1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	contentVersionInfo slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	VersionInfo Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.27 NodeType

Root Class: EBXML_Classes

Class Description: Defines the different ways in which a Classification-Scheme may assign the value of the code attribute for its ClassificationNodes.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme NodeType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.28 Notification

Root Class: EBXML_Classes

Class Description: The Notification class represents a Notification from the registry regarding an event that matches a Subscription. A registry may use a Notification instance to notify a client of an event that matches a Subscription they have registered. This is a push model of notification. A client may also pull events from the registry using the AdhocQuery protocol defined by [ebRS].

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Notification			
Subclass	none			
Attribute	none			
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	registryObjectList subscription	0..* 1	RegistryObject Subscription	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.29 NotificationOptionType

Root Class: EBXML_Classes

Class Description: Defines the different ways in which a client may wish to be notified by the registry of an event within a Subscription.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme NotificationOptionType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.30 NotifyAction

Root Class: EBXML_Classes

Class Description: The NotifyAction class is a sub-class of Action class. An instance of NotifyAction represents an Action that the registry MUST perform in order to notify the subscriber of a Subscription of the events of interest to that subscriber.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Action . . NotifyAction			
Subclass	none			
Attribute	endPoint notificationOption	1 0..1		
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	none			

2.31 ObjectRef

Root Class: EBXML_Classes

Class Description: The information model supports the ability for an attribute in an instance of an information model class to reference a RegistryObject instance using an object reference. An object reference is modeled in this specification with the ObjectRef class. An instance of the ObjectRef class is used to reference a RegistryObject. A RegistryObject MAY be referenced via an ObjectRef instance regardless of its location within a registry or that of the object referring to it.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . ObjectRef			
Subclass	none			
Attribute	createReplica home id	0..1 0..1 1		
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	none			

2.32 ObjectType

Root Class: EBXML_Classes

Class Description: Defines the different types of RegistryObjects a registry may support.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme ObjectType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.33 Organization

Root Class: EBXML_Classes

Class Description: Organization instances provide information on organizations such as a Submitting Organization. Each Organization instance MAY have a reference to a parent Organization.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Organization			
Subclass	none			
Attribute	none			
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	addresses emailAddresses parent primaryContact telephoneNumbers	0..* 0..* 0..1 0..1 0..*	PostalAddress EmailAddress ClassificationNode Person TelephoneNumber	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.34 Person

Root Class: EBXML_Classes

Class Description: Person instances represent persons or humans.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Person			
Subclass	User			
Attribute	none			
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	addresses emailAddresses personName telephoneNumbers	0..* 0..* 0..1 0..*	PostalAddress EmailAddress PersonName TelephoneNumber	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	Organization			

2.35 PersonName

Root Class: EBXML_Classes

Class Description: This class defines attributes for a person's name.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . PersonName			
Subclass	none			
Attribute	firstName lastName middleName	0..1 0..1 0..1		
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	Person User			

2.36 PhoneType

Root Class: EBXML_Classes

Class Description: Defines the types of telephone numbers.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme PhoneType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.37 PostalAddress

Root Class: EBXML_Classes

Class Description: PostalAddress defines attributes of a postal address.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . PostalAddress			
Subclass	none			
Attribute	city country postalCode stateOrProvince street streetNumber	0..1 0..1 0..1 0..1 0..1 0..1		
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	Organization Person User			

2.38 QueryExpression

Root Class: EBXML_Classes

Class Description: The QueryExpression class is an extensible wrapper that can contain a query expression in any supported query syntax such as SQL or Filter Query syntax.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . QueryExpression			
Subclass	none			
Attribute	anyType queryLanguage	1 1		
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	AdhocQuery			

2.39 QueryLanguage

Root Class: EBXML_Classes

Class Description: Defines the query languages supported by a registry.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme QueryLanguage			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.40 Registry

Root Class: EBXML_Classes

Class Description: Registry instances are used to represent a single physical OASIS ebXML Registry.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Registry			
Subclass	none			
Attribute	catalogingLatency conformanceProfile operator replicationSyncLatency specificationVersion	0..1 0..1 1 0..1 1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.41 RegistryObject

Root Class: EBXML_Classes

Class Description: The RegistryObject class extends the Identifiable class and serves as a common super class for most classes in the information model.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject			
Subclass	ExternalIdentifier ClassificationScheme Person AdhocQuery RegistryPackage ClassificationNode ExternalLink Classification Notification Federation Subscription ServiceBinding AuditableEvent Association ExtrinsicObject Service Organization SpecificationLink Registry			
Attribute	description lid name	0..1 0..1 0..1		
Inherited Attribute	home id	0..1 1		
Association	classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..1 0..1 0..1	Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Inherited Association	slots	0..*	Slot	
Referenced from	AuditableEvent Classification ExternalIdentifier Notification SpecificationLink			

2.42 RegistryPackage

Root Class: EBXML_Classes

Class Description: RegistryPackage instances allow for grouping of

logically related RegistryObject instances even if individual member objects belong to different Submitting Organizations.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . RegistryPackage			
Subclass	none			
Attribute	none			
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.43 ResponseStatusType

Root Class: EBXML_Classes

Class Description: Defines the different types of status for a RegistryResponse.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme ResponseStatusType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.44 Service

Root Class: EBXML_Classes

Class Description: Service instances describe services, such as web services.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Service			
Subclass	none			
Attribute	none			
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	serviceBindings	0..*	ServiceBinding	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	ServiceBinding			

2.45 ServiceBinding

Root Class: EBXML_Classes

Class Description: ServiceBinding instances are RegistryObjects that represent technical information on a specific way to access a Service instance. An example is where a ServiceBinding is defined for each protocol that may be used to access the service.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ServiceBinding			
Subclass	none			
Attribute	accessURI	0..1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	service specificationLink targetBinding	1 1..* 0..1	Service SpecificationLink ServiceBinding	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	Service ServiceBinding SpecificationLink			

2.46 Slot

Root Class: EBXML_Classes

Class Description: Slot instances provide a dynamic way to add arbitrary attributes to RegistryObject instances. This ability to add attributes dynamically to RegistryObject instances enables extensibility within the information model. A slot is composed of a name, a slotType and a Bag of values.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Slot			
Subclass	none			
Attribute	name_s slotType values	1 0..1 1..*		
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	AdhocQuery Association AssociationType AuditableEvent Classification ClassificationNode ClassificationScheme ContentInformation ContentManagementService DataType DeletionScopeType EmailType ErrorHandlingModel ErrorSeverityType EventType ExternalIdentifier ExternalLink ExtrinsicObject Federation Identifiable InvocationModel Model NameSpace NodeType Notification NotificationOptionType ObjectType Organization Person PhoneType QueryLanguage Registry RegistryObject RegistryPackage ResponseStatusType Service 55 ServiceBinding SpecificationLink StatusType SubjectGroup SubjectRole Subscription TelecommandService			

2.47 SpecificationLink

Root Class: EBXML_Classes

Class Description: A SpecificationLink provides the linkage between a ServiceBinding and one of its technical specifications that describes how to use the service using the ServiceBinding. For example, a ServiceBinding MAY have SpecificationLink instances that describe how to access the service using a technical specification such as a WSDL document or a CORBA IDL document.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . SpecificationLink			
Subclass	none			
Attribute	usageDescription usageParameters	0..1 0..1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	serviceBinding specificationObject	1 1	ServiceBinding RegistryObject	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	ServiceBinding			

2.48 StatusType

Root Class: EBXML_Classes

Class Description: Defines the different types of status for a Registry-Object.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme StatusType			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.49 SubjectGroup

Root Class: EBXML_Classes

Class Description: Defines the groups that a User may belong to for access control purposes.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme SubjectGroup			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.50 SubjectRole

Root Class: EBXML_Classes

Class Description: Defines the roles that may be assigned to a User for access control purposes.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ClassificationScheme SubjectRole			
Subclass	none			
Attribute	none			
Inherited Attribute	isInternal home id description lid name	1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	nodeType slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	ClassificationNode Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.51 Subscription

Root Class: EBXML_Classes

Class Description: Subscription instances are RegistryObjects that define a User's interest in certain types of AuditableEvents. A User MAY create a subscription with a registry if he or she wishes to receive notification for a specific type of event.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Subscription			
Subclass	none			
Attribute	endTime notificationInterval startTime	0..1 0..1 0..1		
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	actions selector	0..1 1	Action AdhocQuery	
Inherited Association	slots classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..1 0..1	Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	Notification			

2.52 TelecommandService

Root Class: EBXML_Classes

Class Description: The SLE Services provide a standard way of passing CCSDS telecommand and telemetry services across the ground segment. By implementing SLE services, TTC Services Providers will be able to provide a standard interface for supplying TTC services to Missions. This will reduce the cost of providing cross support services for spacecraft missions once the standard is in widespread use. In the near future, CCSDS tracking services and security will be added to the SLE capability, to facilitate the implementation of a fully operational SLE service.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExtrinsicObject TelecommandService			
Subclass	none			
Attribute	none			
Inherited Attribute	isOpaque mimeType home id description lid name	0..1 0..1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	contentVersionInfo slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	VersionInfo Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.53 TelemetryServices

Root Class: EBXML_Classes

Class Description: The SLE Services provide a standard way of passing CCSDS telecommand and telemetry services across the ground segment. By implementing SLE services, TTC Services Providers will be able to provide a standard interface for supplying TTC services to Missions. This will reduce the cost of providing cross support services for spacecraft missions once the standard is in widespread use. In the near future, CCSDS tracking services and security will be added to the SLE capability, to facilitate the implementation of a fully operational SLE service.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExtrinsicObject TelemetryServices			
Subclass	none			
Attribute	none			
Inherited Attribute	isOpaque mimeType home id description lid name	0..1 0..1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	contentVersionInfo slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	VersionInfo Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.54 TelephoneNumber

Root Class: EBXML_Classes

Class Description: This class defines attributes of a telephone number.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . TelephoneNumber			
Subclass	none			
Attribute	areaCode countryCode extension number	0..1 0..1 0..1 0..1		
Inherited Attribute	none			
Association	phoneType	0..1	ClassificationNode	
Inherited Association	none			
Referenced from	Organization Person User			

2.55 User

Root Class: EBXML_Classes

Class Description: User instances represent users that have registered with a registry. User instances are also used in an AuditableEvent to keep track of the identity of the requestor that sent the request that generated the AuditableEvent. User class is a sub-class of Person class that inherits all attributes of the Person class and does not add any new attributes.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . Person User			
Subclass	none			
Attribute	none			
Inherited Attribute	home id description lid name	0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	slots addresses emailAddresses personName telephoneNumbers classifications externalIdentifiers objectType status versionInfo	0..* 0..* 0..* 0..1 0..* 0..* 0..* 0..1 0..1 0..1	Slot PostalAddress EmailAddress PersonName TelephoneNumber Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

2.56 VersionInfo

Root Class: EBXML_Classes

Class Description: VersionInfo class encapsulates information about the specific version of a RegistryObject.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . VersionInfo			
Subclass	none			
Attribute	Comment versionName	0..1 1		
Inherited Attribute	none			
Association	none			
Inherited Association	none			
Referenced from	AdhocQuery Association AssociationType AuditableEvent Classification ClassificationNode ClassificationScheme ContentInformation ContentManagementService DataType DeletionScopeType EmailType ErrorHandlingModel ErrorSeverityType EventType ExternalIdentifier ExternalLink ExtrinsicObject Federation InvocationModel Model NameSpase NodeType Notification NotificationOptionType ObjectType Organization Person PhoneType QueryLanguage Registry RegistryObject RegistryPackage ResponseStatusType Service ServiceBinding SpecificationLink StatusType SubjectGroup SubjectRole Subscription TelecommandService TelemetryServices User			

2.57 XMLSchema

Root Class: EBXML_Classes

Class Description: The Deep Space Network Information Service Architecture (DISA) is a set of information services and information models to enable the Deep Space Network and Advance Multi-mission Operation System (AMMOS) to become a service-oriented architecture. As such, DISA has identified several services needed to support movement towards a SOA. A Registry Service is one such service that has identified needs for managing models, schemas, services, elements, and namespaces.

	Entity	Card	Value/Class	Ind
Hierarchy	EBXML_Classes . Identifiable . . RegistryObject . . . ExtrinsicObject XMLSchema			
Subclass	none			
Attribute	none			
Inherited Attribute	isOpaque mimeType home id description lid name	0..1 0..1 0..1 1 0..1 0..1 0..1		
Association	none			
Inherited Association	contentVersionInfo slots classifications externalIdentifiers objectType status versionInfo	0..1 0..* 0..* 0..* 0..1 0..1 0..1	VersionInfo Slot Classification ExternalIdentifier ClassificationNode ClassificationNode VersionInfo	
Referenced from	none			

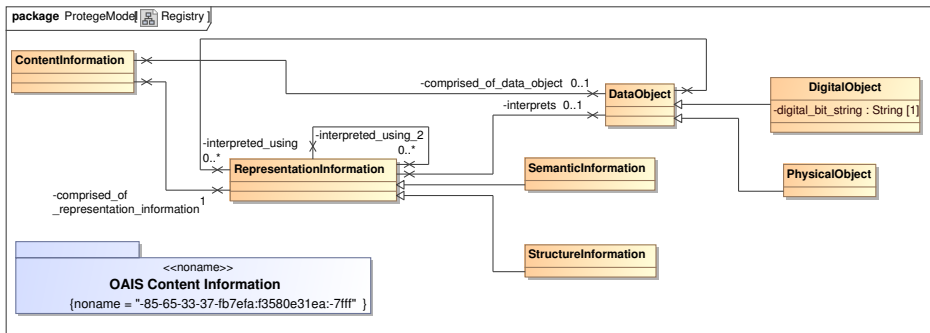


Figure 7: Content Unit UML Class Diagram

3 Content Unit Object Classes

The upper level object model defines the object classes a CCSDS registry will either manage or use for administration. The concept of content unit describes what a registry will manage. Other classes are used by the registry to support the management of content units and support classification, authorization, and related administration functions. For example, ”

The upper level object class hierarchy is illustrated in the following diagram. This diagram presents the subclassOf relation for each object class in a hierarchical (tree) format and provides a visual representation of the object classes in relation to their parent classes. As currently modeled the upper level class hierarchy is flat however it will become more complex as we continue to develop the model.

The class hierarchy above includes 0 unique classes.

The upper level object model is illustrated using the UML class hierarchy diagram in Figure 7. The relations between object classes are one directional. Inverse relations are defined when necessary. The following sections present the upper level object classes in a table format. The table includes the class hierarchy, class attributes, and class associations. The class attributes and associations listed include both those used to define the object class and those inherited from parent classes. Cardinalities are provided where appropriate.

4 Data Dictionary

The following data dictionary lists the attributes and relationships used in the information model along with their definitions.

Comment Each VersionInfo instance MAY have comment. This attribute defines the comment associated with the VersionInfo for a specific RegistryObject version. The value of the comment attribute is indirectly provided by the client as the value of the comment attribute of the `prim:Request` object. The value for this attribute MUST be set by the Registry implementation based upon the `prim:Request` comment attribute value provided by the client if any.

IdentificationScheme Each ExternalIdentifier instance MUST have an identificationScheme attribute that references a ClassificationScheme. This ClassificationScheme defines the namespace within which an identifier is defined using the value attribute for the RegistryObject referenced by the RegistryObject attribute.

URI TBD description

accessURI A ServiceBinding MAY have an accessURI attribute that defines the URI to access that ServiceBinding. This attribute is ignored if a targetBinding attribute is specified for the ServiceBinding. If the URI is a URL then a registry MUST validate the URL to be resolvable at the time of submission before accepting a ServiceBinding submission to the registry.

actions TBD description

address Each EmailAddress instance MUST have an address attribute that provides the actual email address.

addresses Each Organization instance MAY have an addresses attribute that is a Set of PostalAddress instances. Each PostalAddress provides a postal address for that organization. An Organization SHOULD have at least one PostalAddress.

affectedObjects Each AuditableEvent MUST have an affectedObjects attribute that identifies the Set of RegistryObjects instances that were affected by this event.

anyType This attribute is extensible and therefor MAY be of any type depending upon the queryLanguage specified. For SQL queryLanguage it MUST be an SQL query string. For Filter query it MUST be a FilterQueryType defined by [RR-QUERY-XSD].

areaCode Each TelephoneNumber instance MAY have an areaCode attribute that provides the area code for that telephone number.

associationType Each Association MUST have an associationType attribute that identifies the type of that association. The value of the associationType attribute MUST be a reference to a ClassificationNode within the canonical AssociationType ClassificationScheme. While the AssociationType scheme MAY easily be extended, a Registry MUST support the canonical association types as defined by the canonical AssociationType ClassificationScheme.

catalogingLatency Each Registry instance MAY have an attribute named catalogingLatency that specifies the maximum latency between the time a submission is made to the registry and the time it gets cataloged by any cataloging services defined for the objects within the submission.

city Each PostalAddress MAY have a city attribute identifying the city for that address

classificationNode If the Classification instance represents an internal classification, then the classificationNode attribute is required. The classificationNode value MUST reference a ClassificationNode instance.

classificationScheme If the Classification instance represents an external classification, then the classificationScheme attribute is required. The classificationScheme value MUST reference a ClassificationScheme instance.

classifications Each RegistryObject instance MAY have a Set of zero or more Classification instances that are composed within the RegistryObject. These Classification instances classify the RegistryObject.

classifiedObject For both internal and external classifications, the classifiedObject attribute is required and it references the RegistryObject instance that is classified by this Classification.

code Each ClassificationNode MAY have a code attribute. The code attribute contains a code within a standard coding scheme. The code attribute of a ClassificationNode MUST be unique with respect to all sibling ClassificationNodes that are immediate children of the same parent ClassificationNode or ClassificationScheme.

comprised_of_data_object associated data object

comprised_of_representation_information TBD description

conformanceProfile Each Registry instance MAY have an attribute named conformanceProfile that declares the conformance profile that the registry supports. The conformance profiles choices are "registryLite" and "registryFull" as defined by [ebRS].

contentVersionInfo Each ExtrinsicObject instance MAY have a contentVersionInfo attribute. The value of the contentVersionInfo attribute MUST be of type VersionInfo. The contentVersionInfo attribute provides information about the specific version of the RepositoryItem associated with an ExtrinsicObject. The contentVersionInfo attribute is set by the registry.

country Each PostalAddress MAY have a country attribute identifying the country for that address.

countryCode Each TelephoneNumber instance MAY have a countryCode attribute that provides the country code for that telephone number.

createReplica Every ObjectRef instance MAY have a createReplica attribute. The createReplica attribute is a client supplied hint to the registry. When createReplica is true a registry SHOULD create a local replica for the RegistryObject being referenced if it happens to be a remote ObjectRef.

description Each RegistryObject instance MAY have textual description in a human readable and user-friendly form. This attribute is I18N capable and therefore of type InternationalString.

emailAddresses Each Organization instance MAY have an attribute emailAddresses that is a Set of EmailAddress instances. Each EmailAddress provides an email address for that Organization. An Organization SHOULD have at least one EmailAddress.

endPoint This attribute specifies a URI that identifies a service endpoint that MAY be used by the registry to deliver notifications. Currently this attribute can either be a "mailto" URI (e.g. mailto:someone@acme.com) or a "urn:uuid" URI. If endpoint is a "mailto" URI then the registry MUST use the specified email address to deliver the notification via email. Email configuration parameters such as the "from" email address and SMTP server configuration MAY be specified in a registry specific manner. If endpoint is a "urn:uuid" URI then it MUST be a reference to a ServiceBinding object to a Service that implements the RegistryClient interface as defined by [ebRS]. In this case the registry MUST deliver the notification by web service invocation as defined by the ServiceBinding object.

- endTime** This attribute denotes the time after which the subscription expires and is no longer active. If this attribute is missing the subscription never expires.
- eventType** Each AuditableEvent MUST have an eventType attribute which identifies the type of event recorded by the AuditableEvent. The value of the eventType attribute MUST be a reference to a ClassificationNode in the canonical EventType ClassificationScheme. A Registry MUST support the event types as defined by the canonical EventType ClassificationScheme. The canonical EventType ClassificationScheme MAY easily be extended by adding additional ClassificationNodes to the canonical EventType ClassificationScheme.
- extension** Each TelephoneNumber instance MAY have an extension attribute that provides the extension number, if any, for that telephone number.
- externalIdentifiers** Each RegistryObject instance MAY have a Set of zero or more ExternalIdentifier instances that are composed within the RegistryObject. These ExternalIdentifier instances serve as alternate identifiers for the RegistryObject.
- externalURI** Each ExternalLink instance MUST have an externalURI attribute defined. The externalURI attribute provides a URI to the external resource pointed to by this ExternalLink instance. If the URI is a URL then a registry MUST validate the URL to be resolvable at the time of submission before accepting an ExternalLink submission to the registry.
- firstName** Each PersonName SHOULD have a firstName attribute that is the first name of the person.
- home** An Identifiable instance MAY have a home attribute. The home attribute, if present, MUST contain the base URL to the home registry for the RegistryObject instance. The home URL MUST be specified for instances of the Registry class that is defined later in this specification.
- id** Each Identifiable instance MUST have a unique identifier which is used to refer to that object. Note that classes in the information model that do not inherit from Identifiable class do not require a unique id. Examples include classes such as TelephoneNumber, PostalAddress, EmailAddress and PersonName. An Identifiable instance MUST have an id that MUST conform to the rules defined in section title "Unique ID Generation" in [ebRS].

isInternal When submitting a ClassificationScheme instance the submitter MUST declare whether the ClassificationScheme instance represents an internal or an external taxonomy. This allows the registry to validate the subsequent submissions of ClassificationNode and Classification instances in order to maintain the type of ClassificationScheme consistent throughout its lifecycle.

isOpaque Each ExtrinsicObject instance MAY have an isOpaque attribute defined. This attribute determines whether the content catalogued by this ExtrinsicObject is opaque to (not readable by) the registry. In some situations, a Submitting Organization may submit content that is encrypted and not even readable by the registry.

lastName Each PersonName SHOULD have a lastName attribute that is the last name of the person.

lid Each RegistryObject instance MUST have a lid (Logical Id) attribute . The lid is used to refer to a logical RegistryObject in a version independent manner. All versions of a RegistryObject MUST have the same value for the lid attribute. Note that this is in contrast with the id attribute that MUST be unique for each version of the same logical RegistryObject. The lid attribute MAY be specified by the submitter when creating the original version of a RegistryObject. If the submitter assigns the lid attribute when submitting the original version of a RegistryObject, she must guarantee that it is a globally unique URN. A registry MUST honor a valid submitter supplied LID. If the submitter does not specify a LID then the registry MUST assign a LID and the value of the LID attribute MUST be identical to the value of the id attribute of the first (originally created) version of the logical RegistryObject. Note that classes in the information model that do not inherit from RegistryObject class do not require a lid. Examples include Entity classes such as TelephoneNumber, PostalAddress, EmailAddress and PersonName.

middleName Each PersonName SHOULD have a middleName attribute that is the middle name of the person.

contentType Each ExtrinsicObject instance MAY have a contentType attribute defined. The contentType provides information on the type of repository item catalogued by the ExtrinsicObject instance. The value of this attribute SHOULD be a registered MIME media type at <http://www.iana.org/assignments/media-types>.

name Each RegistryObject instance MAY have a human readable name. The name does not need to be unique with respect to other Registry-

Object instances. This attribute is I18N capable and therefore of type `InternationalString`.

name_s Each Slot instance MUST have a name. The name is the primary means for identifying a Slot instance within a `RegistryObject`. Consequently, the name of a Slot instance MUST be locally unique within the `RegistryObject` instance.

nodeRepresentation If the `Classification` instance represents an external classification, then the `nodeRepresentation` attribute is required. It is a representation of a taxonomy element from a classification scheme. It is the responsibility of the registry to distinguish between different types of `nodeRepresentation`, like between the classification scheme node code and the classification scheme node canonical path. This allows the client to transparently use different syntaxes for `nodeRepresentation`.

nodeType When submitting a `ClassificationScheme` instance the Submitting Organization MUST declare the structure of taxonomy nodes within the `ClassificationScheme` via the `nodeType` attribute. The value of the `nodeType` attribute MUST be a reference to a `ClassificationNode` within the canonical `NodeType` `ClassificationScheme`. A Registry MUST support the node types as defined by the canonical `NodeType` `ClassificationScheme`. The canonical `NodeType` `ClassificationScheme` MAY easily be extended by adding additional `ClassificationNodes` to it. The following canonical values are defined for the `NodeType` `ClassificationScheme`: `UniqueCode`: This value indicates that each node of the taxonomy has a unique code assigned to it. `EmbeddedPath`: This value indicates that the unique code assigned to each node of the taxonomy also encodes its path. This is the case in the NAICS taxonomy. `NonUniqueCode`: In some cases nodes are not unique, and it is necessary to use the full path (from `ClassificationScheme` to the node of interest) in order to identify the node. For example, in a geography taxonomy Moscow could be under both Russia and the USA, where there are five cities of that name in different states.

notificationInterval This attribute denotes the duration that a registry MUST wait between delivering successive notifications to the client. The client specifies this attribute in order to control the frequency of notification communication between registry and client.

notificationOption This attribute controls the specific type of event notification content desired by the subscriber. It is used by the subscriber to control the granularity of event notification content communicated by the registry to the subscriber. The value of the `notificationOption` attribute MUST be a reference to a `ClassificationNode` within

the canonical NotificationOptionType ClassificationScheme. A Registry MUST support the notificationOption types as defined by the NotificationOptionType ClassificationScheme. The canonical NotificationOptionType ClassificationScheme MAY easily be extended by adding additional ClassificationNodes to it.

number Each TelephoneNumber instance MAY have a number attribute that provides the local number (without area code, country code and extension) for that telephone number.

objectType Each RegistryObject instance has an objectType attribute. The value of the objectType attribute MUST be a reference to a ClassificationNode in the canonical ObjectType ClassificationScheme. A Registry MUST support the object types as defined by the ObjectType ClassificationScheme. The canonical ObjectType ClassificationScheme may easily be extended by adding additional ClassificationNodes to the canonical ObjectType ClassificationScheme. The objectType for almost all objects in the information model matches the ClassificationNode that corresponds to the name of their class. For example the objectType for a Classification is a reference to the ClassificationNode with code "Classification" in the canonical ObjectType ClassificationScheme. The only exception to this rule is that the objectType for an ExtrinsicObject or an ExternalLink instance MAY be defined by the submitter and indicates the type of content associated with that object. A registry MUST set the correct objectType on a RegistryObject when returning it as a response to a client request. A client MAY set the objectType on a RegistryObject when submitting the object. A client SHOULD set the objectType when the object is an ExternalLink or an ExtrinsicObject since content pointed to or described by these types may be of arbitrary objectType.

operator Each Registry instance MUST have an attribute named operator that is a reference to the Organization instance representing the organization for the registry's operator. Since the same Organization MAY operate multiple registries, it is possible that the home registry for the Organization referenced by operator may not be the local registry.

parent Each ClassificationNode MAY have a parent attribute. The parent attribute either references a parent ClassificationNode or a ClassificationScheme instance in case of first level classificationNode instances.

path Each ClassificationNode MAY have a path attribute. A registry MUST set the path attribute for any ClassificationNode that has a non-null code attribute value, when the ClassificationNode is retrieved from the registry. The path attribute MUST be ignored by the registry when it is specified by the client at the time the object is submitted to

the registry. The path attribute contains the canonical path from the root ClassificationScheme or ClassificationNode within the hierarchy of this ClassificationNode as defined by the parent attribute. The path attribute of a ClassificationNode MUST be unique within a registry.

personName Each Person instance MAY have a personName attribute that provides the name for that user.

phoneType Each TelephoneNumber instance MAY have a phoneType attribute that provides the type for the TelephoneNumber. The value of the phoneType attribute MUST be a reference to a ClassificationNode in the canonical PhoneType ClassificationScheme.

postalCode Each PostalAddress MAY have a postalCode attribute identifying the postal code (e.g., zip code) for that address.

primaryContact Each Organization instance SHOULD have a primaryContact attribute that references the Person instance for the person that is the primary contact for that organization.

queryExpression Each AdhocQuery instance MAY have a queryExpression attribute that contains the query expression for the AdhocQuery depending upon the use case as follows. When an AdhocQuery is submitted to the registry it MUST contain a queryExpression. When a stored AdhocQuery is included in an AdhocQueryRequest to invoke a stored query as defined by the stored query feature defined in [ebRS] it SHOULD NOT contain a queryExpression.

queryLanguage The queryLanguage attribute specifies the query language that the query expression conforms to. The value of this attribute MUST be a reference to a ClassificationNode within the canonical QueryLanguage ClassificationScheme. A Registry MUST support the query languages as defined by the canonical QueryLanguage ClassificationScheme. The canonical QueryLanguage ClassificationScheme MAY easily be extended by adding additional ClassificationNodes to it to allow a registry to support additional query language syntaxes.

registryObject Each ExternalIdentifier instance MUST have a registryObject attribute that references the parent RegistryObject for which this is an ExternalIdentifier.

registryObjectList This attribute specifies a Set of ObjectRefs or a Set of RegistryObject instances that represent the objects that were impacted by the event that matched the Subscription. The registry MUST include ObjectRef or RegistryObject instances as Set elements depending upon the notificationOption specified for the Subscription.

replicationSyncLatency Each Federation instance MAY specify a replicationSyncLatency attribute that describes the time duration that is the amount of time within which a member of this Federation MUST synchronize itself with the current state of the Federation. Members of the Federation MAY use this parameter to periodically synchronize the federation metadata they MUST cache locally about the state of the Federation and its members. Such synchronization MAY be based upon the registry event notification capability.

requestId Each AuditableEvent MUST have a requestId attribute that identifies the client request instance that affected this event.

selector This attribute defines the selection criteria that determine which events match this Subscription and are of interest to the User. The selector attribute references a pre-defined query that is stored in the registry as an instance of the AdhocQuery class. This AdhocQuery instance specifies or "selects" events that are of interest to the subscriber.

service A ServiceBinding MUST have a service attribute whose value MUST be the id of its parent Service.

serviceBinding A SpecificationLink instance MUST have a serviceBinding attribute that provides a reference to its parent

serviceBindings A Service MAY have a serviceBindings attribute that defines the service bindings that provide access to that Service.

slotType Each Slot instance MAY have a slotType that allows different slots to be grouped together. The slotType attribute MAY also be used to indicate the data type or value domain for the slot value(s).

slots An Identifiable instance MAY have a Set of zero or more Slot instances that are composed within the Identifiable instance. These Slot instances serve as extensible attributes that MAY be defined for the Identifiable instance.

sourceObject Each Association MUST have a sourceObject attribute that references the RegistryObject instance that is the source of that Association.

specificationLink A ServiceBinding MAY have a specificationLinks attribute defined that is a Set of references to SpecificationLink instances. Each SpecificationLink instance links the ServiceBinding to a particular technical specification that MAY be used to access the Service for the ServiceBinding.

specificationObject A SpecificationLink instance MUST have a specificationObject attribute that provides a reference to a RegistryObject instance that provides a technical specification for the parent ServiceBinding. Typically, this is an ExtrinsicObject instance representing the technical specification (e.g., a WSDL document). It may also be an ExternalLink object in case the technical specification is a resource that is external to the registry.

specificationVersion Each Registry instance MUST have an attribute named specificationVersion that is the version of the ebXML Registry Services Specification [ebRS].

startTime This attribute denotes the time at which the subscription becomes active. If this attribute is missing subscription starts immediately.

stateOrProvince Each PostalAddress MAY have a stateOrProvince attribute identifying the state, province or region for that address.

status Each RegistryObject instance MUST have a life cycle status indicator. The status is assigned by the registry. A registry MUST set the correct status on a RegistryObject when returning it as a response to a client request. A client SHOULD NOT set the status on a RegistryObject when submitting the object as this is the responsibility of the registry. A registry MUST ignore the status on a RegistryObject when it is set by the client during submission or update of the object. The value of the status attribute MUST be a reference to a ClassificationNode in the canonical StatusType ClassificationScheme. A Registry MUST support the status types as defined by the StatusType ClassificationScheme. The canonical StatusType ClassificationScheme MAY easily be extended by adding additional ClassificationNodes to the canonical StatusType ClassificationScheme.

street Each PostalAddress MAY have a street attribute identifying the street name for that address.

streetNumber Each PostalAddress MAY have a streetNumber attribute identifying the street number (e.g., 65) for the street address.

subscription This attribute specifies a reference to a Subscription instance within the registry. This is the Subscription that matches the event for which this Notification is about.

targetBinding A ServiceBinding MAY have a targetBinding attribute defined that references another ServiceBinding. A targetBinding MAY be specified when a service is being redirected to another service. This allows the rehosting of a service by another service provider.

- targetObject** Each Association MUST have a targetObject attribute that references the RegistryObject instance that is the target of that Association.
- telephoneNumbers** Each Person instance MAY have a telephoneNumbers attribute that contains the Set of TelephoneNumber instances defined for that user. A Person SHOULD have at least one TelephoneNumber.
- timestamp** Each AuditableEvent MUST have a timestamp attribute that records the date and time that this event occurred.
- type** Each EmailAddress instance MAY have a type attribute that provides the type for that email address. The value of the type attribute MUST be a reference to a ClassificationNode in the canonical EmailType ClassificationScheme as referenced in appendix .
- usageDescription** A SpecificationLink instance MAY have a usageDescription attribute that provides a textual description of how to use the optional usageParameters attribute described next. The usageDescription is of type InternationalString, thus allowing the description to be in multiple languages.
- usageParameters** A SpecificationLink instance MAY have a usageParameters attribute that provides a Bag of Strings representing the instance specific parameters needed to use the technical specification (e.g., a WSDL document) specified by this SpecificationLink object.
- user** Each AuditableEvent MUST have a user attribute that identifies the User that sent the request that generated this event affecting the RegistryObject instance.
- value** Each ExternalIdentifier instance MUST have a value attribute that provides the identifier value for this ExternalIdentifier (e.g., the actual social security number).
- values** A Slot instance MUST have a Sequence of values. The Sequence of values MAY be empty. Since a Slot represent an extensible attribute whose value MAY be a Sequence, therefore a Slot is allowed to have a Sequence of values rather than a single value.
- versionInfo** Each RegistryObject instance MAY have a versionInfo attribute. The value of the versionInfo attribute MUST be of type VersionInfo. The versionInfo attribute provides information about the specific version of a RegistryObject. The versionInfo attribute is set by the registry.

versionName Each VersionInfo instance MUST have versionName. This attribute defines the version name identifying the VersionInfo for a specific RegistryObject version. The value for this attribute MUST be automatically generated by the Registry implementation.