

Integrating the Healthcare Enterprise



5

**IHE IT Infrastructure  
Technical Framework**

**Volume 3  
(ITI TF-3)**

10

**Cross-Transaction Specifications and Content  
Specifications**

15

20

**Revision 11.0 – Final Text  
September 23, 2014**

25

**Please verify you have the most recent version of this document, which is published [here](#).**

## CONTENTS

4	Metadata used in Document Sharing profiles .....	4
30	4.1 Abstract Metadata Model.....	4
	4.1.1 Metadata Object Types .....	5
	4.1.2 Association Types.....	8
	4.1.3 Metadata Attributes .....	9
	4.1.3.1 The Purpose of Metadata Attributes (Informative) .....	9
35	4.1.3.2 DocumentEntry Metadata Attributes.....	10
	4.1.3.3 SubmissionSet Metadata Attributes .....	12
	4.1.3.4 Folder Metadata Attributes.....	13
	4.2 ebRIM Representation .....	14
	4.2.1 Metadata Object Types .....	15
40	4.2.1.1 DocumentEntry.....	15
	4.2.1.2 SubmissionSet .....	18
	4.2.1.3 Folder.....	20
	4.2.1.4 Registry Object List.....	22
	4.2.2 Association Types.....	23
45	4.2.2.1 HasMember .....	26
	4.2.2.2 Document Relationship .....	33
	4.2.3 Metadata Attributes .....	44
	4.2.3.1 General Information about Metadata Attributes.....	44
	4.2.3.2 DocumentEntry Attributes.....	58
50	4.2.3.3 SubmissionSet Attributes .....	80
	4.2.3.4 Folder Attributes.....	88
	4.2.4 Error Reporting.....	94
	4.2.4.1 RegistryErrors Element .....	94
	4.2.4.2 Error responses .....	98
55	4.2.5 Metadata Vocabulary.....	100
	4.2.5.1 Submission Set Object UUIDs .....	100
	4.2.5.2 Document Entry Object.....	100
	4.2.5.3 Folder Object .....	101
	4.3 Additional Document Sharing Requirements .....	101
60	4.3.1 Requirements on Submission Type Transactions.....	101
	4.3.1.1 Submission Metadata Attribute Optionality .....	102
	4.3.1.2 XDS Specific Requirements.....	104
	4.3.2 Requirements on Query Type Transactions .....	111
	4.3.2.1 Query Type Metadata Attribute Optionality .....	111
65	5 IHE Content Specifications.....	115
	5.1 Basic Patient Privacy Consents Module .....	115
	5.1.1 References .....	115
	5.1.2 Patient Privacy Consent Acknowledgment Document Specification	
	1.3.6.1.4.1.19376.1.5.3.1.1.7 – With no Scanned Document Part.....	115

70	5.1.2.1 XDS Metadata .....	115
	5.1.2.2 Specification .....	116
	5.1.3 Patient Privacy Consent Acknowledgment Document Specification	
	1.3.6.1.4.1.19376.1.5.3.1.1.7.1 – With Scanned Document .....	119
	5.1.3.1 XDS Metadata .....	119
75	5.1.3.3 Specification .....	119
	5.1.3.4 Conformance .....	119
	5.2 Scanned Documents Content Module.....	119
	5.2.1 Referenced Standards .....	120
	5.2.1.1 Discussion of Content Standards.....	120
80	5.2.2 XDS Metadata .....	121
	5.2.2.1 XDS DocumentEntry Metadata.....	121
	5.2.2.2 XDS SubmissionSet Metadata .....	122
	5.2.2.3 XDS Folder Metadata.....	122
	5.2.3 Specification.....	123
85	5.2.3.1 ClinicalDocument child-less elements .....	124
	5.2.3.2 ClinicalDocument/recordTarget .....	125
	5.2.3.3 ClinicalDocument/author (original) .....	126
	5.2.3.4 ClinicalDocument/author (scanner).....	127
	5.2.3.5 ClinicalDocument/dataEnterer .....	128
90	5.2.3.6 ClinicalDocument/custodian .....	129
	5.2.3.7 ClinicalDocument/legalAuthenticator .....	130
	5.2.3.8 ClinicalDocument/documentationOf.....	131
	5.2.3.9 ClinicalDocument/component/nonXMLBody .....	131
	5.2.4 Complete Example (Wrapped PDF) .....	132
95		

## 4 Metadata used in Document Sharing profiles

This section describes the metadata that is used in IHE profiles designed for sharing documents (Document Sharing profiles). The Document Sharing profiles are implementing the Document Sharing concept outlined in the ITI whitepaper entitled *Health Information Exchange: Enabling Document Sharing Using IHE Profiles* available on the IHE website ([http://ihe.net/Technical\\_Frameworks/#IT](http://ihe.net/Technical_Frameworks/#IT)). This section assumes understanding of the concepts presented in the white paper.

The ITI Document Sharing profiles which use this metadata are:

- Cross-Enterprise Document Sharing (XDS.b)
- 105 • Cross-Enterprise Document Reliable Interchange (XDR)
- Cross-Enterprise Document Media Interchange (XDM)
- Multi-Patient Queries (MPQ)
- Cross-Community Access (XCA)

110 It is likely that future ITI profiles will also use Document Sharing metadata. Profiles from IHE domains other than ITI that use or constrain this metadata are not listed here. Those profiles will document their use of this metadata.

Document Sharing profiles manage two aspects of the documents being shared, the metadata about the document and the contents of the document. If you think of a document as a book in a library, the index card in the library's card catalog is the document metadata.

115 Metadata encodes the properties of documents, the environments they come from, circumstances of their submission, terms to reference in queries, and grouping with other documents.

Section 4 first explains the metadata at a conceptual level (Section 4.1), then at an implementation level (Section 4.2) and then provides some profile- and transaction-specific metadata constraints and considerations (Section 4.3).

### 120 4.1 Abstract Metadata Model

The metadata used in Document Sharing profiles is characterized by three types of objects and two types of Associations. In figure 4.1-1, the three objects types and two Association types are depicted using UML to show their relationships. The three object types are:

- 125 • SubmissionSet – represents a collection of Folders, Documents and Associations submitted together.
- Folder – represents a collection of related Documents.
- DocumentEntry – represents a Document.
- The two Association types are:

130

- HasMember – represents membership of objects. There are four variations of HasMember that are described in ITI TF-3:4.1.2.
- Relationship – represents a relationship between DocumentEntries.

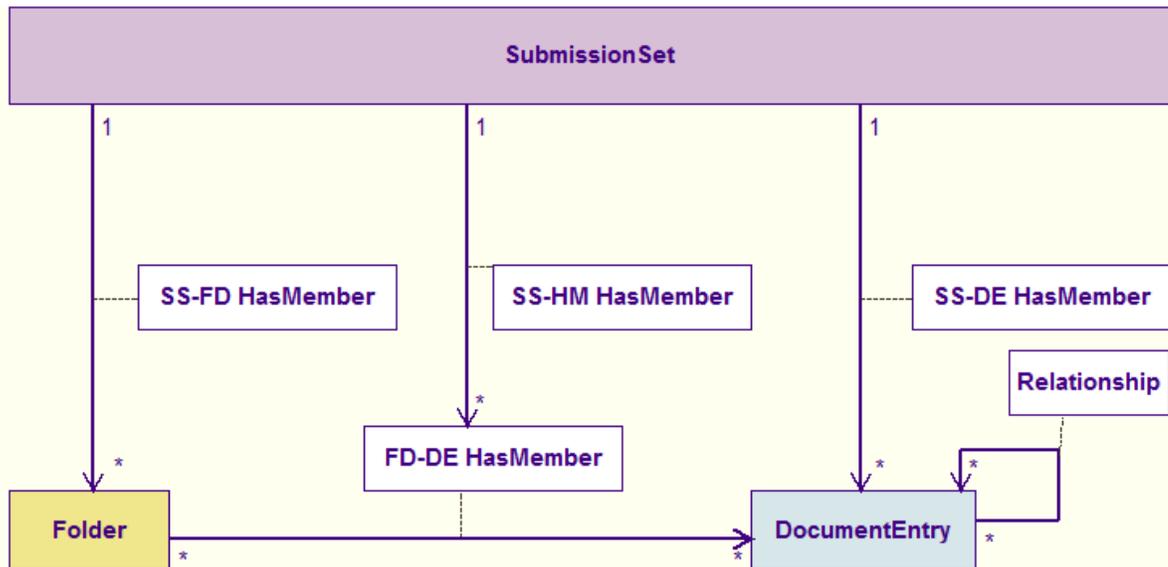


Figure 4.1-1: Document Sharing Objects and Associations

#### 4.1.1 Metadata Object Types

135

There are three metadata object types supported by the Document Sharing metadata, as seen in Figure 4.1-1:

- SubmissionSet
- Folder
- DocumentEntry

140

**SubmissionSet** – The SubmissionSet can be thought of as the packing slip of a postal package. The details of the submission of DocumentEntries, Folders, and Associations are captured in the SubmissionSet object. The creating entity of each submission must group the DocumentEntries, Folders and Associations into a unique SubmissionSet. The Document Sharing profiles ensure that the documents are treated as a unit for submission purposes – either all of the documents arrive at their destination, or none of them do. An example of the use of a Submission Set is packaging all documents related to a care episode at the end of the hospital stay. The EHR system can submit the package. If the submission fails, none of the documents made it to their destination, and a retry is possible.

145

150

DocumentEntries may be bundled into a SubmissionSet by a human, machine, or process. For example, a laboratory machine might automatically submit results associated with a given lab order when they are ready, rather than waiting for a human to bundle them. SubmissionSets may

contain DocumentEntries for multiple patients, but there are specific limitations on how this is done.

155 A SubmissionSet shall be the source of at least one Association of type SS-FD HasMember, SS-HM HasMember, and/or SS-DE HasMember.

**Folder** – A Folder is a logical collection of DocumentEntries that are related in some way, such as to a clinical use case. A Folder is an arbitrary grouping relationship. Folders may be updated by multiple SubmissionSets sent from multiples departments that are submitting their DocumentEntry objects at different times. For example, a Folder may be used to collect the DocumentEntry objects for the patient’s documents that relate to an exam event, such as the exam request and prior results as well as the eventual exam results. As the exam results become available, the DocumentEntry objects can be added to the Folder for the exam records.

All DocumentEntries in a Folder shall be for the same patient.

165 The metadata structure discussed in this volume only specifies how to describe a Folder, and imposes no requirements for when or how a Folder should be used. Additional detail on when and how to use a Folder may be described in IHE profiles.

**DocumentEntry** – DocumentEntry is a metadata object representing a document. This metadata object does not contain the contents of the document; instead it contains attributes describing the document.

170 There are two XDSDocumentEntry Types: Stable Document Entry and On-Demand Document Entry. The following sections describe these types in detail and refer to identifiers of various types. How these identifiers are used and relate to each other is central to understanding the mechanisms used to describe and retrieve content.

- 175 • Stable DocumentEntry uniqueID – is the identifier found in the uniqueID metadata field of a Stable DocumentEntry. It is a unique identifier for the entry and also uniquely defines the document associated with the entry. It is used in a retrieve request to identify which specific document should be retrieved.
- 180 • CDA Header id – Every HL7 CDA R2 document header includes a value for the ClinicalDocument/id field which uniquely identifies the document. This value is always the same as the DocumentEntry uniqueID for the Stable Document Entry which contains metadata about this document. Although we address CDA specifically in the explanation, the concepts also apply to any type of document content where a unique identifier exists in a code field within the document. Some document content does not include a unique identifier so these explanations do not apply to that situation.
- 185 • On-Demand DocumentEntry uniqueID – is the identifier found in the uniqueID metadata field of an On-Demand Document Entry. It is a unique identifier for the entry but does not identify any document.

Details on how documents and metadata are managed depend on the requirements in a particular Document Sharing Profile.

190 For example, in XDS, a Stable DocumentEntry is the logical representation in the Registry of the Document that the Source submitted to a Repository. An entire document's contents can constitute several megabytes, but can be described in a few kilobytes of metadata. The DocumentEntry metadata that describes the document are sufficient for the purposes of storing, organizing and locating documents for retrieval. Submitting a Stable DocumentEntry to a  
195 Registry in lieu of submitting the document creates a separation of concerns, allowing the Registry to specialize in indexing, while the Repository manages document storage.

- **Stable DocumentEntry**

A Stable Document Entry contains metadata about an already created document available for retrieval. Each Stable DocumentEntry represents a single document. This document is  
200 stable because the contents have been effectively combined in the exact representation that will be returned in a Retrieve Document Set. A Stable Document Entry is an XDSDocument Entry with objectType equal to the UUID for Stable (see 4.2.53.1.2 for the UUID) and availabilityStatus = Approved or Deprecated. All metadata fields contain valid values. If the document returned on a retrieve request is CDA, it will have in the  
205 ClinicalDocument/id in the HL7 CDA R2 header the same value of the DocumentEntry uniqueId.

- **On-Demand Document Entry**

An On-Demand Document Entry provides a unique identifier which can be used to create an on-demand document which collects the latest, most recent available information at  
210 the time of retrieval. On-Demand Document Entries never reflect actual document content, but rather the potential for a document with the characteristics described in the metadata of the entry. An On-Demand Document Entry may be replaced and deprecated. If an On-Demand Document Entry is deprecated, of that uniqueID may not have most recent information and should return an error.

215 The uniqueID associated with an On-Demand Document Entry will never represent an actual document. A retrieve request specifying an On-Demand Document Entry uniqueID will return content identified by a uniqueID different than the specified uniqueID.

Every On-Demand Document Entry with the same uniqueID will refer to the same potential content. Actual content depends on the time of retrieval. The On-Demand  
220 Document Entry uniqueID is valid for as long as the entry has availabilityStatus=Approved. The holder of the uniqueID may re-use it in a retrieve request to get the latest information, without the need for an additional query.

When a retrieve request is received specifying an On-Demand Document Entry  
225 uniqueID, the responder may choose to persist the document generated as a result and allow the requestor future access to the metadata and document. This capability is declared through the Persistence of Retrieved Documents Option on the On-Demand Document Source and Responding Gateway Actors. The persistence refers not only to the saving of the content for re-use, but more specifically, to the ability of the requester to use retrieve to access that exact, possibly now historic, content and use a query to get  
230 metadata about the content.

#### 4.1.2 Association Types

Associations represent a link from the source object to a target object. Association objects describe all aspects of this link including references to source and target objects, the specific variant or name of the Association, and status and version information.

235 There are two types of Associations: HasMember and Relationship.

- **HasMember**- defines a membership relationship between two objects. There are four variants of the HasMember Association depending on the types of the source and target object, see Figure 4.1-1.

240 SS-FD HasMember: An association from a SubmissionSet to a Folder identifies the Folder as a member of that SubmissionSet. It identifies the Submission Set that contained the initial creation of the Folder.

245 FD-DE HasMember: An association from a Folder to a DocumentEntry identifies that DocumentEntry as a member of that Folder. Folders have a many-to-many relationship to DocumentEntries (i.e., one folder may be linked to many DocumentEntries, and one DocumentEntry may be linked to many folders).

SS-HM HasMember: An association from a SubmissionSet to a FD-DE HasMember Association identifies that FD-DE HasMember as a member of that SubmissionSet. This makes it possible to identify the Submission Set in which the link between the Folder and the DocumentEntry was created.

250 SS-DE HasMember: An association from a SubmissionSet to a DocumentEntry identifies the DocumentEntry as a member of that SubmissionSet. The association between the SubmissionSet and the DocumentEntry provides information about the submission of the Documents. With this association of a DocumentEntry, you can find the Submission Set and know when the document was submitted, who the author of the submission was, and other information contained in the attributes of that SubmissionSet.

255

- **Relationship** – defines an association between two DocumentEntry objects. There are five variants based on the type of relationship between the DocumentEntry objects.

Replace – indicates the replacement of a previous document with a new document.

260

Transform - indicates the transformation of a previous document into a new document.

Append – indicates a new document that appends to the contents of a previous document.

Transform and Replace – indicates a transformed replacement of a previous document with a new document.

265 Signs – indicates a new document is a signature for a previous document, as in new document signs previous document.

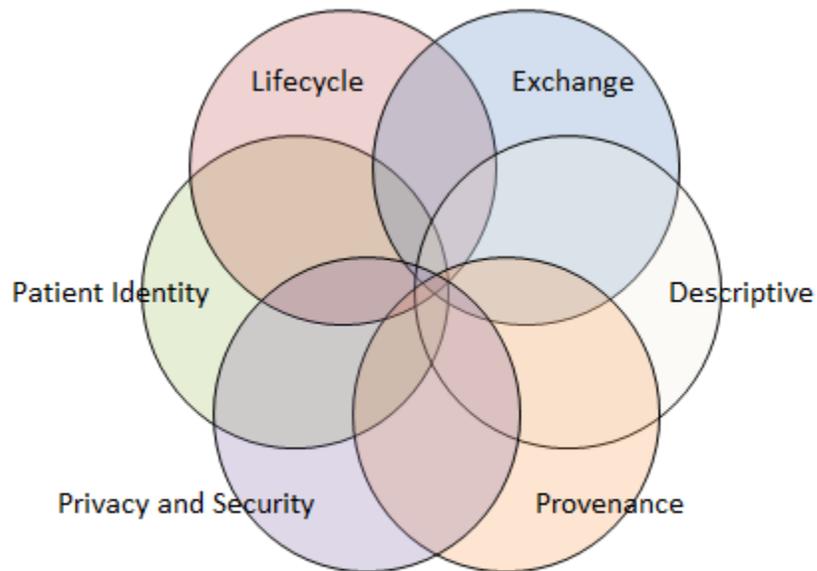
### 4.1.3 Metadata Attributes

Each metadata object holds attributes used for a variety of purposes. This section outlines the variety of purposes metadata attributes serve as well as a general description of each attribute. Detail about the coding of attributes is described in ITI TF-3:4.2.3.

#### 270 4.1.3.1 The Purpose of Metadata Attributes (Informative)

Metadata attributes can be categorized according to specific document-handling purposes. Each metadata attribute typically has more than one purpose, although some have only one. Metadata in the Document Sharing profiles has one or more of these purposes.

- 275 • **Patient Identity** – Attributes that describe the subject of the document. This includes patient Id, patient name, and other demographics.
- **Provenance** – Attributes that describe where the document comes from. These items are highly influenced by medical records regulations. This includes human author, identification of system that authored, the organization that authored, predecessor documents, successor documents, and the pathway that the document took.
- 280 • **Security & Privacy** – Attributes that are used by Privacy and Security rules to appropriately control the document. These values enable conformance to Privacy and Security regulations. These characteristics would be those referenced in Privacy or Security rules. These characteristics would also be used to protect against security risks to confidentiality, integrity, and availability.
- 285 • **Descriptive** – Attributes that are used to describe the clinical value, so they are expressly healthcare-specific. These values are critical for query models and enable workflows in all exchange models. The number of attributes in this category is kept to minimum so the metadata doesn't simply duplicate the document, and to keep disclosure risk to a minimum. Thus the metadata attribute values tend to be from a small set of codes.
- 290 Because this category is close to the clinical values it tends to have few mandatory attributes, allowing policy to choose to not populate. For healthcare documents, this is typically very closely associated with the clinical workflows but also must recognize other uses of healthcare documents such as quality reporting, public health reporting, authorized clinical research, patient access, etc.
- 295 • **Object Lifecycle** – Attributes that describe the current lifecycle state of the document including relationships to other documents. This would include classic lifecycle states of created, published, replaced, transformed, and deprecated.
- **Exchange** -- Attributes that enable the transfer of the document for both push type transfers, and pull type transfers. These attributes are used for low-level automated processing of the document. These attributes are not the workflow routing, but rather the administrative overhead necessary to make the transfer. This includes the document unique Id, location, size, MIME types, and document format.
- 300



**Figure 4.1.3.1-1: Pictorial of Overlapping Document Sharing Metadata Purpose**

305 All metadata attributes describe the document and are not a replacement for the document. Not  
 all metadata attributes are always required; indeed some metadata attributes would be used only  
 for specific uses. Care has been taken to limit the metadata to the minimum metadata attributes  
 necessary to achieve the goal. Each metadata element was assessed for risks posed by exposing it  
 as metadata. All metadata attributes are defined to assure that when the element is needed that it  
 310 be consistently assigned and processed.

**4.1.3.2 DocumentEntry Metadata Attributes**

Table 4.1.3.2-1 provides a conceptual view of the metadata attributes associated with a  
 DocumentEntry object. The table describes each attribute and provides a mapping between the  
 attribute and the purposes that attribute is designed to support. The full DocumentEntry metadata  
 315 attribute definition, including data type and coding is in Section 4.2.3.2.

**Table 4.1.3.2-1: DocumentEntry Metadata Attribute Definition**

DocumentEntry Metadata Attribute	Description	Purpose					
		Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
author	Characterizes the humans and/or machines that		X	X	X		X

DocumentEntry Metadata Attribute	Description	Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
	authored the document. This attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty and authorTelecommunication.						
availabilityStatus	Characterizes the lifecycle status of the DocumentEntry					X	
classCode	A high-level classification of documents that indicates the kind of document, e.g., report, summary, note, consent.			X	X		
comments	Comments associated with the document.				X		
confidentialityCode	The code specifying the level of confidentiality of the document.			X			
creationTime	Characterizes the time the author created the document.		X	X	X	X	
entryUUID	A globally unique identifier used to manage the entry.			X	X	X	X
eventCodeList	This list of codes represents the main clinical acts, such as a colonoscopy or an appendectomy, being documented.				X		
formatCode	Code globally uniquely specifying the format of the document.				X		X
hash	Hash of the document itself.			X			
healthcareFacility TypeCode	This code represents the type of organizational setting of the clinical encounter during which the documented act occurred.		X	X	X		
homeCommunityId	A globally unique identifier for a community.						X
languageCode	Specifies the human language of character data in the document.				X		
legalAuthenticator	Characterizes a participant who has legally authenticated or attested the document within the authorInstitution.		X	X		X	
limitedMetadata	Indicates whether the Document Entry was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source.		X		X	X	X
mimeType	MIME type of the document.				X		X
objectType	The type of DocumentEntry						X
patientId	The patientId represents the subject of care of the document.	X		X	X		
practiceSettingCode	The code specifying the clinical specialty where the act that resulted in the document was performed (e.g., Family Practice, Laboratory, Radiology).		X	X	X		

DocumentEntry Metadata Attribute	Description	Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
repositoryUniqueId	The globally unique identifier of the repository where the document is stored.						X
serviceStartTime	Represents the start time the service being documented took place.			X	X		
serviceStopTime	Represents the stop time the service being documented took place.			X	X		
size	Size in bytes of the document.			X			X
sourcePatientId	The sourcePatientId represents the subject of care medical record Identifier (e.g., Patient Id) in the local patient Identifier Domain of the creating entity.	X	X				
sourcePatientInfo	This attribute contains demographic information of the source patient to whose medical record this document belongs.	X	X				
title	Represents the title of the document.				X		
typeCode	A low-level classification of documents within a classCode that describes class, event, specialty, and setting.				X		
uniqueId	The globally unique identifier assigned by the document creator to this document.			X			X
URI	The URI for the document.						X

#### 4.1.3.3 SubmissionSet Metadata Attributes

320 Table 4.1.3.3-1 provides a conceptual view of the metadata attributes associated with a SubmissionSet object. The table describes each attribute and provides a mapping between the attribute and the purposes that attribute is designed to support. The full SubmissionSet metadata attribute definition, including data type and coding is in Section 4.2.3.3.

**Table 4.1.3.3-1: SubmissionSet Metadata Attribute Definition**

Submission Set Metadata Attribute	Description	Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
-----------------------------------	-------------	------------------	------------	--------------------	-------------	------------------	----------

Submission Set Metadata Attribute	Description	Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
author	The humans and/or machines that created the submission set. This attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty, authorTelecommunication.		X	X	X		
availabilityStatus	The lifecycle status of the SubmissionSet				X	X	
comments	Comments associated with the SubmissionSet.				X		
contentTypeCode	The code specifying the type of clinical activity that resulted in placing these documents in this SubmissionSet.		X	X	X		
entryUUID	A globally unique identifier used to manage the entry.			X	X		X
homeCommunityId	A globally unique identifier for a community.						X
intendedRecipient	The organization(s) or person(s) for whom the Submission Set is intended.			X			X
patientId	The patientId represents the primary subject of care whose longitudinal record is being reflected in this Submission Set.	X		X	X		
sourceId	Identifier of the Document Source that created the SubmissionSet.		X	X	X		
submissionTime	Point in Time at the Document Source when the Submission Set was created.			X	X	X	
title	The title of the SubmissionSet.				X		
uniqueId	Globally unique identifier for the SubmissionSet assigned by the Document Source.			X			X

325 **4.1.3.4 Folder Metadata Attributes**

Table 4.1.3.4-1 provides a conceptual view of the metadata attributes associated with a Folder object. The table describes each attribute and provides a mapping between the attribute and the purposes that attribute is designed to support. The full Folder metadata attribute definition, including data type and coding is in Section 4.2.3.4.

330

**Table 4.1.3.4-1: Folder Metadata Attribute Definition**

Folder Metadata Attribute	Description	Patient Identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
availabilityStatus	The lifecycle status of the Folder				X	X	
codeList	The set of codes specifying the type of clinical activities that resulted in placing documents in this Folder.		X	X	X		
comments	Comments associated with the Folder.				X		
entryUUID	A globally unique identifier used to manage the entry.			X	X		X
homeCommunityId	A globally unique identifier for a community.						X
lastUpdateTime	Most recent point in time that the Folder has been modified.					X	
patientId	The patientId represents the subject of care of documents within the Folder.	X		X	X		
title	The name of the Folder.				X		
uniqueId	Globally unique identifier for the Folder.			X			X

## 4.2 ebRIM Representation

This section details the representation of the metadata objects and their attributes using classes provided by OASIS ebXML RegRep 3.0 specification at <http://docs.oasis-open.org/regrep/v3.0/regrep-3.0-os.zip>.

The Electronic Business using eXtensible Markup Language (ebXML) Registry and Repository (RegRep) specification describes a way to implement registry and repository servers and clients using standard interfaces, protocols and an information model for publishing, management, discovery and retrieval of arbitrary content and metadata that describes it.

The ebXML RegRep specification is made of two parts:

- **ebRIM:** The "ebXML Registry Information Model version 3.0" (ebRIM) defines the types of metadata and content that can be stored in an ebXML Registry.
- **ebRS:** The "ebXML Registry Services Specification version 3.0" (ebRS) defines the services and protocols for an ebXML Registry.

IHE highly constrains the use of ebRIM and ebRS in Document Sharing profiles to fit the requirements for expression of metadata objects and to communicate the objects between actors. This section focuses on expression of the objects, and IHE transactions and profiles detail the communication.

When document sharing was first introduced in IHE, XDS was the only document sharing model. In the initial XDS Profile, the Document Registry Actor could be implemented as an

355 adaptor to an ebXML Registry. As such, all XDS content is valid in terms of the ebRIM, but XDS introduces additional restrictions on the data that may be transmitted. Only a limited number of the classes in ebRIM are supported by XDS and the contents and semantics of those classes are further restricted. While an XDS Registry may be implemented as an adaptor to an ebXML Registry, or without an underlying ebXML Registry, it should not be assumed that features available from a pure ebXML Registry are available in an IHE environment. Features of an ebXML Registry should be considered as not available unless they are explicitly defined by individual IHE profiles.

360 Now that document sharing in IHE has grown beyond the XDS model, Document Sharing metadata applies to profiles beyond XDS. In those other environments, it is highly unlikely to be implemented using an ebXML Registry.

IHE Technical Framework documentation conventionally refers to the ebRIM namespace using the “rim:” prefix, for example rim:ExtrinsicObject, rim:RegistryPackage, rim:Slot, rim:Classification, etc.

365

**Table 4.2-1: ebRIM/Document Sharing Correspondence**

Document Sharing Object/Association	ebRIM class
DocumentEntry	rim:ExtrinsicObject
SubmissionSet	rim:RegistryPackage
Folder	
MemberOf	rim:Association
Relationship	

The DocumentEntry object type is modeled through the rim:ExtrinsicObject class.

370 The SubmissionSet and Folder object types are conveyed through the rim:RegistryPackage class. Since the ebRIM standard does not allow for subclassing the RegistryPackage class, these two objects are implemented as rim:RegistryPackages. A rim:Classification is used to distinguish between the SubmissionSet and Folder object types.

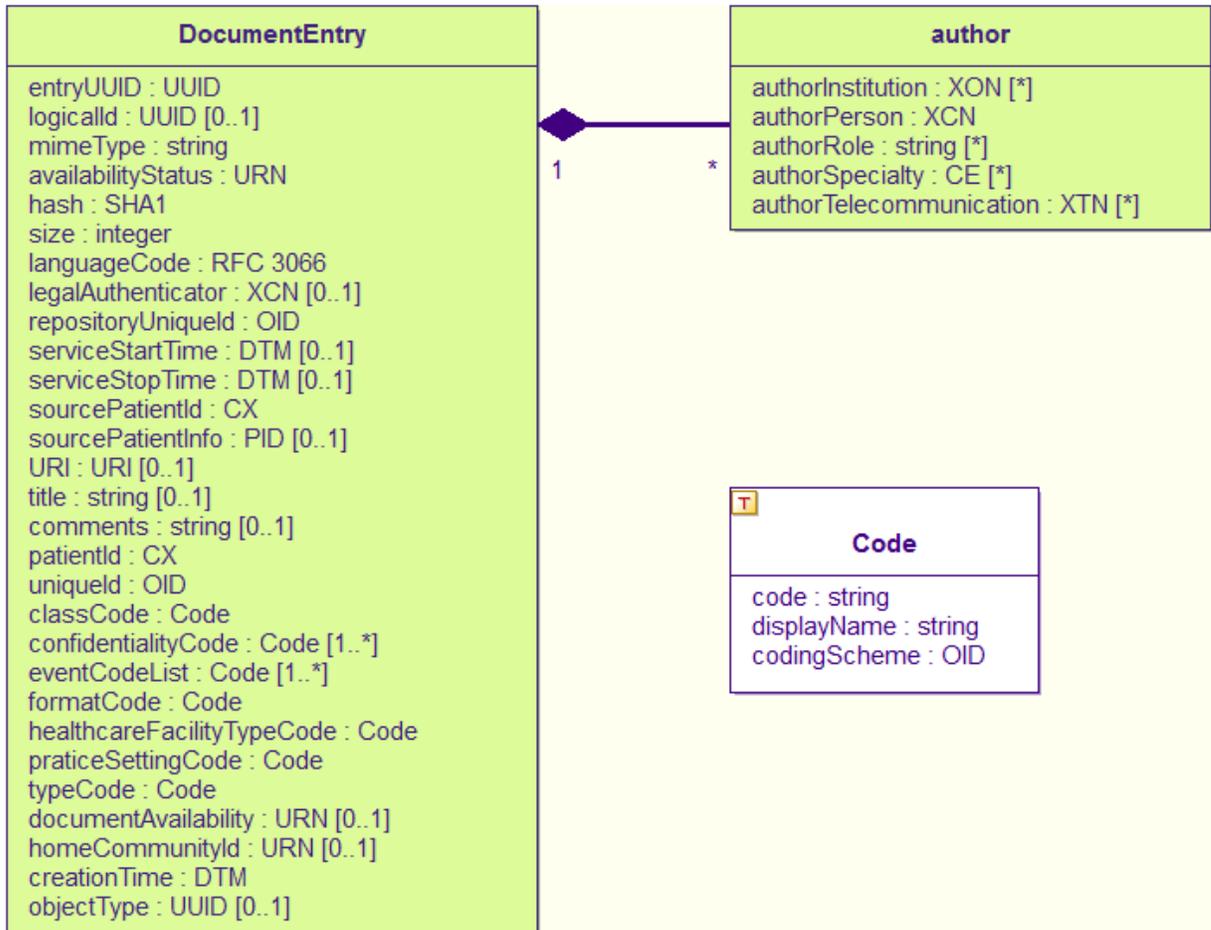
The HasMember and Relationship Association concepts are conveyed through the rim:Association class.

## 375 4.2.1 Metadata Object Types

### 4.2.1.1 DocumentEntry

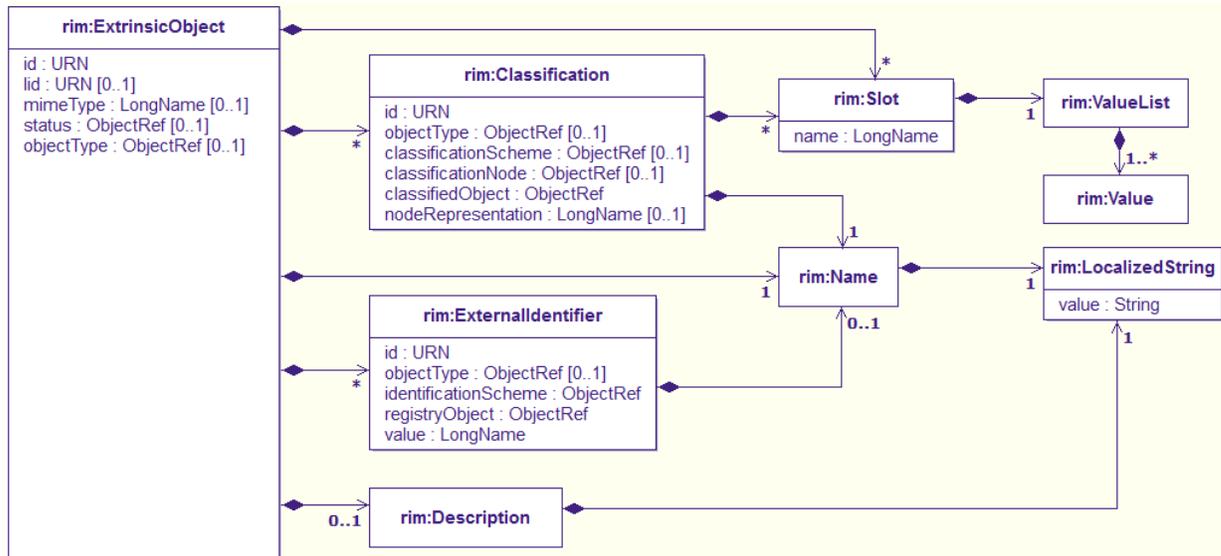
The DocumentEntry does not contain the contents of the document; instead it contains attributes describing the document. Further details regarding the DocumentEntry object type can be found in Section 4.1.1.

380 Figure 4.2.1.1-1 represents the DocumentEntry and its attributes. Detailed descriptions of all the attributes of a DocumentEntry are described in section 4.2.3.2.



**Figure 4.2.1.1-1: DocumentEntry Metadata Attributes (Informative)**

385 The abstract concept of a DocumentEntry is expressed through an ebRIM RegistryPackage classified as a DocumentEntry.



**Figure 4.2.1.1-2: UML diagram for rim:ExtrinsicObject (Informative)**

390 Figure 4.2.1.1-2 represents rim:ExtrinsicObject as a structure made of classes and attributes of the eBRIM subset used for Document Sharing. This diagram is read from left to right and rim:ExtrinsicObject is considered as the root class.

The expression of the DocumentEntry is done by mapping the abstract DocumentEntry metadata attributes into rim:ExtrinsicObject class attributes, elements and other associated classes. This mapping uses, wherever possible, the parts of rim:ExtrinsicObject as intended (such as Name, 395 Description and ExternalIdentifier), and holds the healthcare specific attributes in general purpose Slots or Classifications.

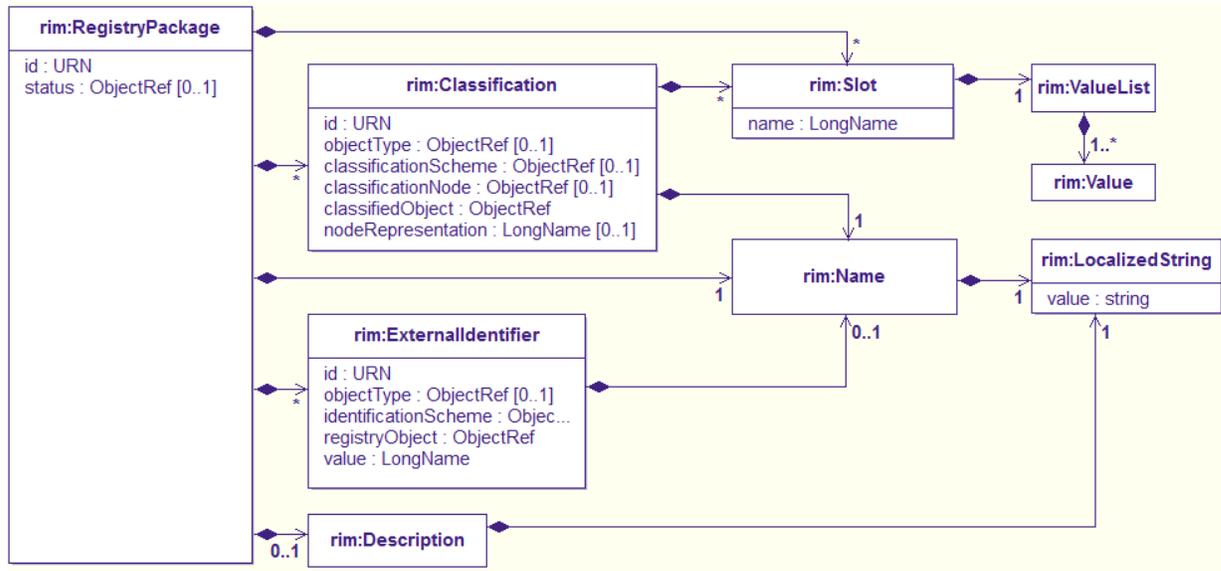
Requirements for matching SubmissionSet.patientId to included or referenced DocumentEntries' patientId are detailed in Section 4.2.2.1.1.

#### 4.2.1.1.1 DocumentEntry types

400 As described in Section 4.1.1, there are two DocumentEntry types: Stable Document Entry and On-Demand Document Entry. A Stable Document Entry is an XDSDocument Entry with objectType equal to the UUID for Stable (see Section 4.2.5.2 for the UUID). An On-Demand DocumentEntry has an objectType equal to the UUID for on-demand (see Section 4.2.5.2 for the UUID). Each Stable DocumentEntry represents a single document which is identified by the 405 uniqueId attribute.

#### 4.2.1.2 SubmissionSet

410 The abstract concept of a SubmissionSet is expressed through an ebRIM RegistryPackage classified as a SubmissionSet. The SubmissionSet bundles DocumentEntry, Folder and Association objects for submission. Further details regarding the SubmissionSet object type can be found in Section 4.1.1.



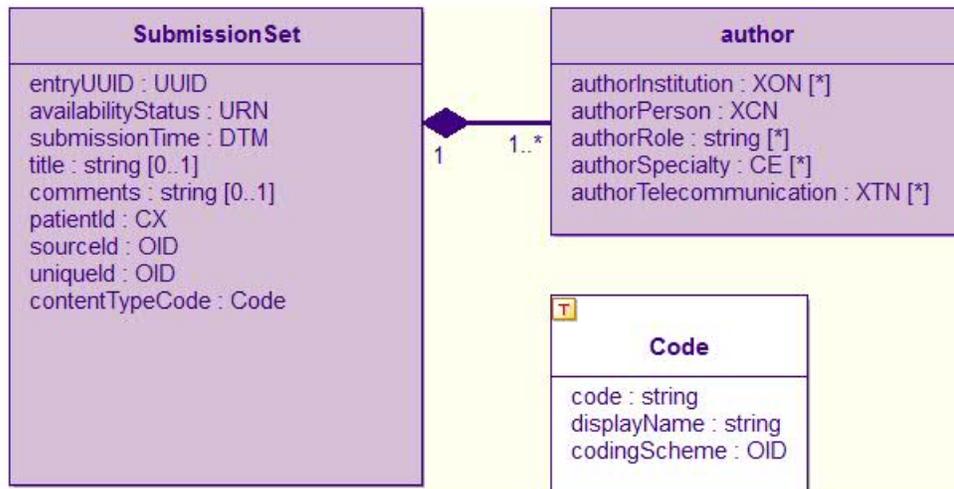
**Figure 4.2.1.2-1: UML diagram for SubmissionSet (Informative)**

415 This expression is done by mapping the abstract SubmissionSet metadata attributes into, wherever possible, the parts of RegistryPackage as intended and holding the healthcare-specific attributes in general-purpose Slots and Classification. An ebRIM Classification class is used to identify a RegistryPackage as a SubmissionSet versus a Folder.

A SubmissionSet has a set of attributes that are described in ITI TF-3: 4.1.3.3 SubmissionSet Metadata.

420 SubmissionSets exist for two reasons:

1. To support atomic submissions
2. To provide a permanent record of:
  - the existence and status of the submission
  - the Folders and DocumentEntry objects and Associations included in the submission



425

**Figure 4.2.1.2-2: SubmissionSet Metadata Attributes (Informative)**

The value of the patientId attribute of the DocumentEntry objects that a SubmissionSet contains shall match the value of the patientId attribute on the SubmissionSet itself.

430 Requirements for matching the value of SubmissionSet.patientId to the value of Patient Id in referenced DocumentEntry objects are detailed in Section 4.2.2.1.1.

#### 4.2.1.2.1 Creating a SubmissionSet object from a RegistryPackage element

435 A SubmissionSet object shall be created from a RegistryPackage element by labeling it with a Classification of type urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd. A receiver of metadata shall accept the Classification element encoded within the RegistryPackage element or on the same level as the RegistryPackage. The following XML example demonstrates these two valid approaches to encoding the Classification.

##### Classification encoded inside the RegistryPackage object

```

440 <RegistryPackage id="SubmissionSetId">
  <rim:Classification
    classifiedObject="SubmissionSetId"
    classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd"
    id="urn:uuid:1d4d08bc-85cc-4596-8fdc-4b5410a6feae"
    objectType="urn:oasis:names:tc:ebxml-
445   regrep:ObjectType:RegistryObject:Classification"
  />
</RegistryPackage>

```

**Classification encoded outside the RegistryPackage object**

```

450 <RegistryPackage id="SubmissionSetId">
</RegistryPackage>
<rim:Classification
  classifiedObject="SubmissionSetId"
  classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd"
455 id="urn:uuid:1d4d08bc-85cc-4596-8fdc-4b5410a6feae"
  objectType="urn:oasis:names:tc:ebxml-
  regrep:ObjectType:RegistryObject:Classification"
/>

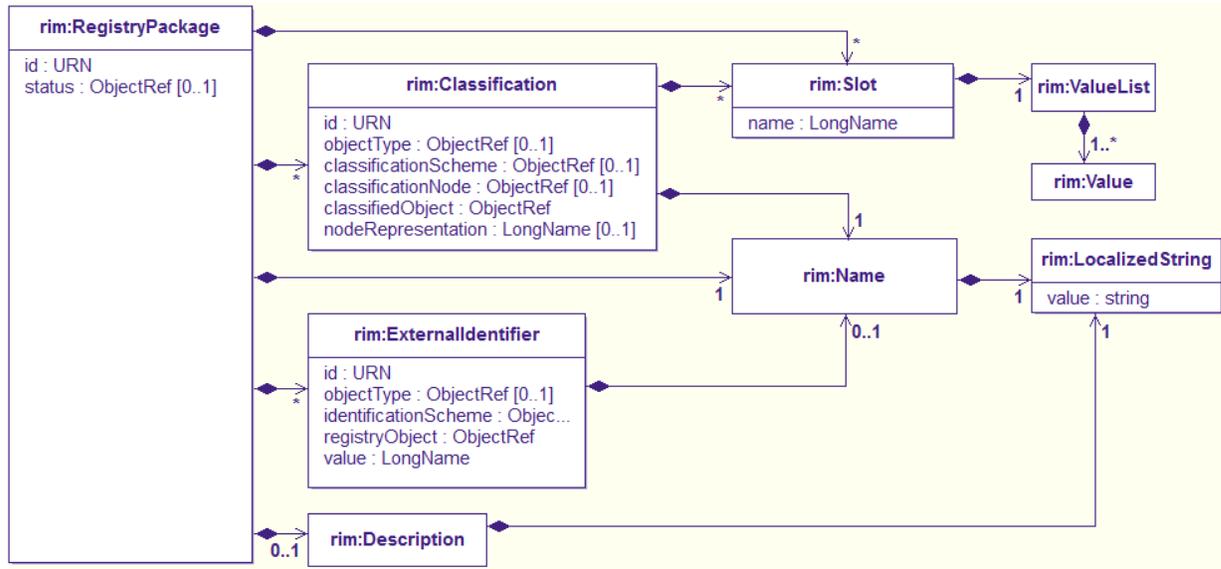
```

Item	Description
Classification/@classifiedObject	The @id attribute of the RegistryPackage being classified.
Classification/@classificationNode	A fixed value identifying the type of object the RegistryPackage represents. Accepted values: urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd
Classification/@id	Symbolic id or UUID identifying this Classification. See section 4.2.3.1.5 for details.
Classification/@objectType	Fixed value as specified by ebRIM. Optional upon submission of objects, required upon retrieval. If set, must be "urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification".

460 This Classification shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

**4.2.1.3 Folder**

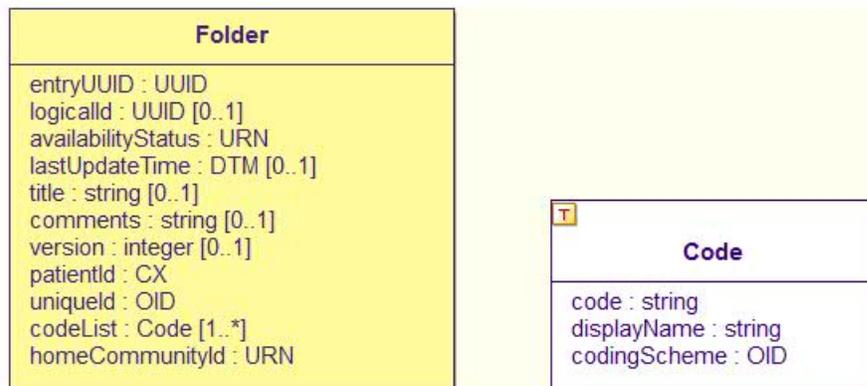
465 The abstract concept of a Folder is expressed through an ebRIM RegistryPackage classified as Folder (see the UML representation of the ebRIM RegistryPackage, Figure 4.2.1.2-1). A Folder is used to bundle DocumentEntry objects. Further details regarding the Folder object type can be found in Section 4.1.1.



**Figure 4.2.1.3-1: UML diagram for Folder (Informative)**

470 This expression is done by mapping the abstract Folder metadata attributes into, wherever possible, the parts of RegistryPackage as intended, and holds the healthcare-specific attributes in general purpose Slots and Classifications. An eBRIM Classification class is used to identify a RegistryPackage as a Folder versus a SubmissionSet.

475 Folders shall not be nested inside other Folders. The value of the patientId attribute of the DocumentEntry objects it contains shall match the value of the patientId attribute on the folder itself.



**Figure 4.2.1.3-2: Folder Metadata Attributes (Informative)**

#### 4.2.1.3.1 Creating a Folder object from a RegistryPackage element

480 A Folder object shall be created from a RegistryPackage element by labeling it with a Classification of type urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2. A receiver of metadata shall accept the Classification element encoded within the RegistryPackage element or on the

same level. The following XML example demonstrates these two valid approaches to encoding the Classification.

**Classification encoded inside the RegistryPackage object**

```

485 <...>
    <RegistryPackage id="Fol">
    <!-- Classify registry package Fol as being a Folder -->
    <Classification
490     classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2"
     classifiedObject="Fol"
     id="IdExample_066"
     objectType="urn:oasis:names:tc:ebxml-
495     regrep:ObjectType:RegistryObject:Classification"
     />
    </RegistryPackage>
    <...>
  
```

**Classification encoded outside the RegistryPackage object**

```

500 <RegistryPackage id="Fol">
    ...
    </RegistryPackage>
    <!-- Classify registry package as Folder -->
    <Classification
505     classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2"
     classifiedObject="Fol"
     id="IdExample_066"
     objectType="urn:oasis:names:tc:ebXML-
510     regrep:ObjectType:RegistryObject:Classification"
     />
    <...>
  
```

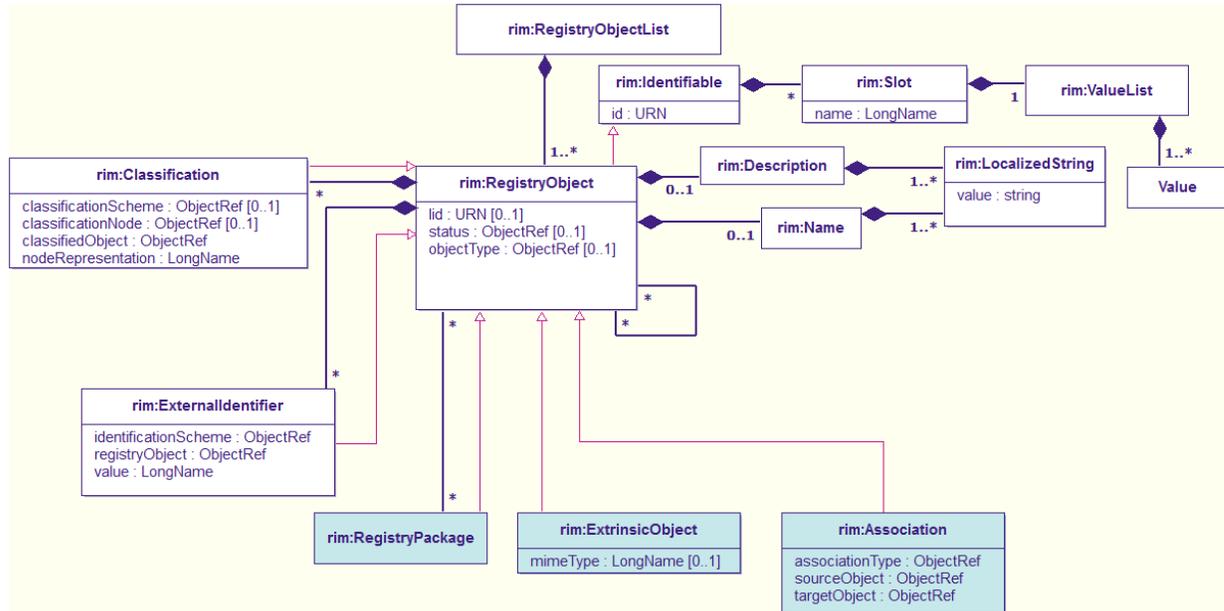
Item	Description
Classification/@classifiedObject	The @id attribute of the RegistryPackage being classified.
Classification/@classificationNode	A fixed value identifying the type of object the RegistryPackage represents. Accepted values: Folder: urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2
Classification/@id	Symbolic id or UUID identifying this Classification. See section 4.2.3.1.5 for details.
Classification/@objectType	Fixed value as specified by ebRIM. Optional upon submission of objects, required upon retrieval. If set, must be "urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification".

This Classification shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

**4.2.1.4 Registry Object List**

In submission requests and query responses, a Registry Object List contains a list of Folders, SubmissionSets, DocumentEntries and Associations.

Figure 4.2.1.1-1 shows in detail the content of the rim:RegistryObjectList used to exchange Document Sharing metadata; a subset of the ebXML Registry Information Model (ebRIM).



**Figure 4.2.1.1-1: Registry Object List (Informative)**

520 The following XML example demonstrates the encoding of several metadata objects grouped within a rim:RegistryObjectList:

```

525 <rim:RegistryObjectList>
    <rim:RegistryPackage id="SubmissionSet01"> ... </rim:RegistryPackage>
    <rim:Association id="Document01InSubmissionSet01" ... />
    <rim:ExtrinsicObject id="Document01"> ... </rim:ExtrinsicObject>
</rim:RegistryObjectList>

```

## 4.2.2 Association Types

530 All relationships between metadata objects are handled through Associations. An Association is an object that describes a named relationship between two metadata objects. The relationship between the DocumentEntry and the Document it represents is made with the DocumentEntry.uniqueId attribute, and not an Association since the Document is not a metadata object.

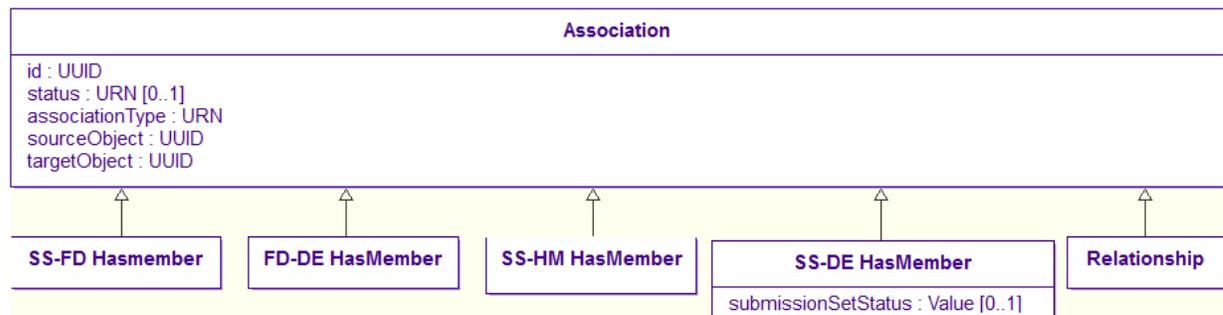
Associations can be used to build relationships between:

- 535
- A SubmissionSet and a DocumentEntry – SS-DE HasMember
  - A SubmissionSet and a Folder – SS-FD HasMember
  - A Folder and a DocumentEntry – FD-DE HasMember
  - A SubmissionSet and an Association – SS-HM HasMember

- A DocumentEntry and another DocumentEntry – Relationship

540 Once deprecated, a DocumentEntry shall not be referenced by future associations.

The abstract concept of a HasMember or Relationship Association is expressed through an ebRIM Association illustrated in the diagram below. This expression is done by mapping the abstract Association metadata attributes into Association class attributes and other associated classes. Further details regarding the Association object type can be found in Section 4.1.2.



545

**Figure 4.2.2-1: Association (Informative)**

Figure 4.2.2-1 represents the attributes of an Association. This diagram demonstrates that the various HasMember and Relationship Associations inherit the attributes from the Association class, and that the SS-DE HasMember (SubmissionSet to DocumentEntry) also has the submissionSetStatus metadata attribute in addition to the Association class attributes. All Associations shall have an id (entryUUID) attribute. It may have UUID or symbolic format depending on where they are used. Symbolic format is allowable only in submissions.

550

Associations have 3 other required attributes (see Figure 4.2.2-1):

- sourceObject
- targetObject
- associationType

555

These attributes can be thought to make a small sentence:

- sourceObject AssociationType targetObject

The sentence is composed of noun-verb-object for example:

560

- Folder HasMember DocumentEntry

Graphically this example Association looks like:

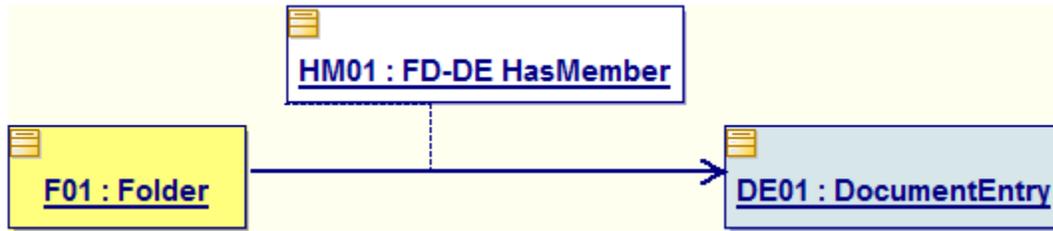


Figure 4.2.2-2: Folder HasMember DocumentEntry (Informative)

**Association Type formatting**

565 An Association type shall be specified as a URN.

The sourceObject and targetObject are UUID or symbolic format depending on where they are used. Symbolic format is allowable only in submissions. The status attribute shall not be submitted but shall be returned from queries.

The valid Association types are specified in the following table.

570

**Table 4.2.2-1: Association Types**

Meaning	Association Type
Membership in a Registry Package (SubmissionSet or Folder)	urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember
Replace	urn:ihe:iti:2007:AssociationType:RPLC
Transformation	urn:ihe:iti:2007:AssociationType:XFRM
Addendum	urn:ihe:iti:2007:AssociationType:APND
Replace with Transformation	urn:ihe:iti:2007:AssociationType:XFRM_RPLC
Digital Signature	urn:ihe:iti:2007:AssociationType:signs
Snapshot of On-Demand document entry	urn:ihe:iti:2010:AssociationType:IsSnapshotOf

**Example of basic Association**

575

```
<rim:Association
  id="urn:uuid:95e9115b-3d90-46ae-9610-ed34fd683d96"
  status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
  associationType="urn:ihe:iti:2007:AssociationType:RPLC"
  sourceObject="urn:uuid:3cce0135-cedb-4a26-ba00-8698ee8dde04"
  targetObject="urn:uuid:e0985823-dc50-45a5-a6c8-a11a829893bd" />
```

580

#### 4.2.2.1 HasMember

In the Document Sharing abstract metadata model, many different relationships are defined between SubmissionSet, DocumentEntry and Folder objects. In this section, each of these relationships is given its own name, like SS-DE HasMember -  
585 SubmissionSetHasMemberDocumentEntry. In the underlying ebRIM model, all of these relationships are created using the ebRIM HasMember Association type.

Note: There are four variants of the HasMember Association. See section 4.1.2 for an overview.

##### 4.2.2.1.1 SS-DE HasMember

590 **HasMember** - a DocumentEntry shall be submitted as part of a SubmissionSet by connecting the objects with a HasMember Association. This is shown as SS-DE HasMember in Figure 4.1-1. DocumentEntries may be included in a SubmissionSet in two ways: inclusion by value and inclusion by reference.

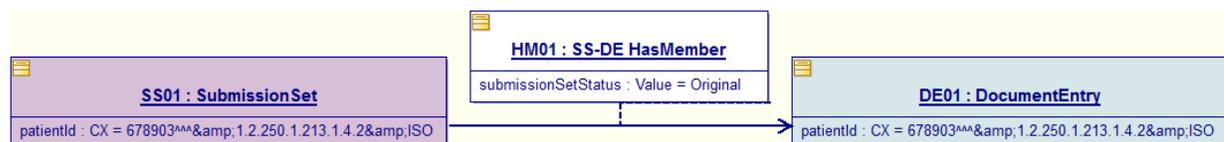
##### SubmissionSet Association labeling

595 Two types of Association labels are defined: original (submission by value), or reference (Submission by reference). This enables finding the SubmissionSet that first submitted any document.

##### Submission of an original Document (inclusion by value)

600 When the creating entity has a new document to be submitted, it shall submit a DocumentEntry by value in the SubmissionSet. This means that the DocumentEntry (and corresponding Document) are part of the submission. The HasMember Association shall contain a slot with the name SubmissionSetStatus with the value set to **original**.

605 All DocumentEntries submitted in a SubmissionSet, included by value, shall have their patientId attribute set to the same value. The value of the SubmissionSet.patientId attribute shall match the value of the DocumentEntry.patientId attribute. The metadata of this submission contains the SubmissionSet, the DocumentEntry, and the original SS-DE HasMember Association connecting them.



**Figure 4.2.2.1.1-1: SubmissionSet HasMember DocumentEntry (Informative)**

When submitting an existing document by value:

- 610
- The targetObject shall contain the Id of the DocumentEntry object.
  - The sourceObject shall contain the Id of the SubmissionSet object

The following XML example demonstrates how to encode a submission by value.

```
615 <rim:Association
      associationType="urn:oasis:names:tc:ebxml-
      regrep:AssociationType:HasMember"
      sourceObject="SubmissionSet01"
      targetObject="Document01">
620   <rim:Slot name="SubmissionSetStatus">
        <rim:ValueList>
          <rim:Value>Original</rim:Value>
        </rim:ValueList>
      </rim:Slot>
    </rim:Association>
```

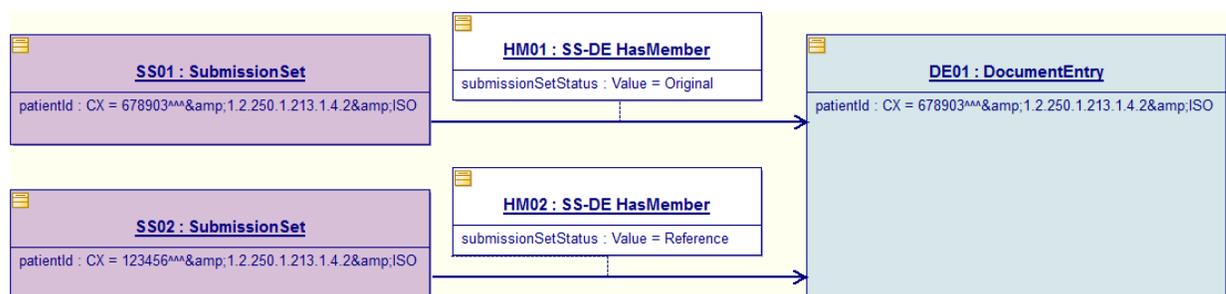
625

### Submission of a reference to an existing Document (inclusion by reference)

Existing documents can be referenced by a SubmissionSet. This means that the DocumentEntry (and corresponding Document) are not part of the submission; they have been previously submitted and already exist in the receiving actor. Documents that were submitted in a previous SubmissionSet may be referenced by subsequent SubmissionSets. In this case, the HasMember Association shall contain a slot with the name SubmissionSetStatus with the value set to **Reference**.

The value of the SubmissionSet.patientId attribute is not required to match the value of the DocumentEntry.patientId attribute of a DocumentEntry included by reference. The metadata of this submission contains the SubmissionSet. The SS-DE HasMember Association with the 'Value=Reference' connects the SubmissionSet to a DocumentEntry already present in the receiving actor.

635



640 **Figure 4.2.2.1.1-2: Submission of a reference to an existing Document (Informative)**

When submitting a reference to an existing document:

- The targetObject shall contain the Id of the DocumentEntry object.
- The sourceObject shall contain the Id of the SubmissionSet object.
- The following XML example demonstrates how to encode a submission by reference.

645

```
650 <rim:Association
      associationType="urn:oasis:names:tc:ebxml-
      regrep:AssociationType:HasMember"
      sourceObject="SubmissionSet01"
      targetObject="Document01">
      <rim:Slot name="SubmissionSetStatus">
        <rim:ValueList>
          <rim:Value>Reference</rim:Value>
        </rim:ValueList>
655 </rim:Slot>
    </rim:Association>
```

#### 4.2.2.1.2 SS-FD HasMember

660 **HasMember** - Submit a Folder. The submission includes the Folder object and a HasMember Association linking the Folder to the SubmissionSet. This is shown as SS-FD HasMember in Figure 4.2.2.1.2-1. The value of the SubmissionSet.patientId attribute shall match the value of the Folder.patientId attribute.

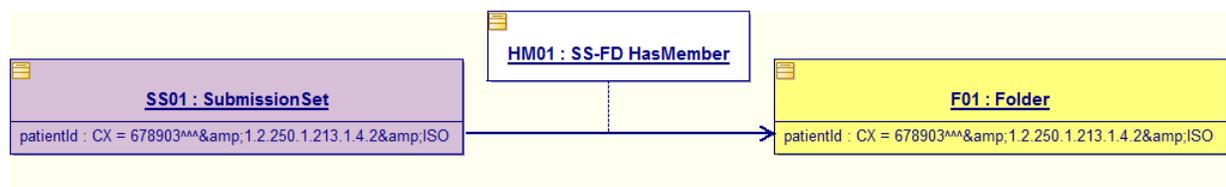


Figure 4.2.2.1.2-1: SubmissionSet HasMember Folder (Informative)

665

When submitting a Folder:

- The targetObject shall contain the Id of the Folder object.
- The sourceObject shall contain the Id of the SubmissionSet object.

670

#### Example SubmissionSet – Folder HasMember Association

```
675 <rim:Association
      associationType="urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember"
      sourceObject="SubmissionSet01"
      targetObject="Folder01">
    </rim:Association>
```

The sourceObject and targetObject attributes are shown using symbolic names to reference the other objects in the submission. UUID format values could have been used if those objects were coded that way.

#### 4.2.2.1.3 FD-DE HasMember

680 **FD-DE HasMember** - a HasMember Association linking a Folder to a DocumentEntry. The value of the Folder.patientId attribute shall match the value of the DocumentEntry.patientId attribute.

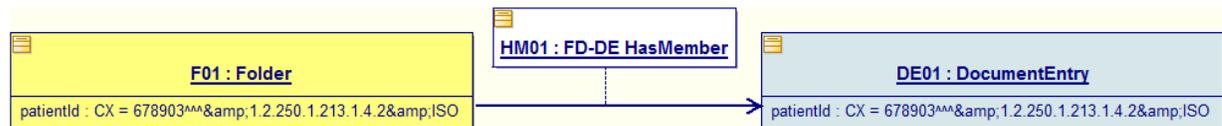


Figure 4.2.2.1.3-1: Folder HasMember DocumentEntry (Informative)

#### 685 Linking documents to a folder

A document can be linked to a Folder to indicate that this document is a member of a particular Folder. This is colloquially called “putting the document into the folder.” Each FD-DE HasMember Association shall be accompanied by a SS-HM HasMember Association that links the FD-DE HasMember Association with the SubmissionSet object (see Section 4.2.2.1.4). See Section 4.2.2.15 for the four ways a DocumentEntry can be added to a Folder.

When adding a DocumentEntry to a Folder:

- The targetObject shall contain the Id of the DocumentEntry object.
- The sourceObject shall contain the Id of the Folder object.

#### 695 Example Folder HasMember Association

```
<Association
  id="FolderToDocAssoc"
  associationType="urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember"
  sourceObject="urn:uuid:e0985823-dc50-45a5-a6c8-a11a8298aabb"
  targetObject="urn:uuid:e0985823-dc50-45a5-a6c8-a11a829893bd" />

<Association
  associationType="urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember"
  sourceObject="SubmissionSet"
  targetObject="FolderToDocAssoc" />
```

In the first association, since both the sourceObject and targetObject attributes in the example are in UUID format (and not symbolic id format), the Folder and the DocumentEntry referenced could be part of this submission or already present in the registry.

The second Association shown is a SS-HM HasMember which is between the SubmissionSet and the first Association documenting which submission added the DocumentEntry to the Folder.

#### 4.2.2.1.4 SS-HM HasMember

**HasMember** - a HasMember Association linking a SubmissionSet to a FD-DE HasMember Association, which is in turn an Association between a Folder and a DocumentEntry. This is

715 shown as SS-HM HasMember in Figure 4.2.2.1.5-1. This shall be used to record the SubmissionSet responsible for adding the DocumentEntry to the Folder. The values of SubmissionSet.patientId, Folder.patientId, and DocumentEntry.patientId shall all be the same.

720 This kind of Association is used when adding a document to an existing Folder. It is used to identify the entity that created the link between a particular document and a particular Folder and shall be as follows:

- The targetObject shall contain the Id of the Association that links the DocumentEntry and the Folder.
- The sourceObject shall contain the Id of the SubmissionSet object.

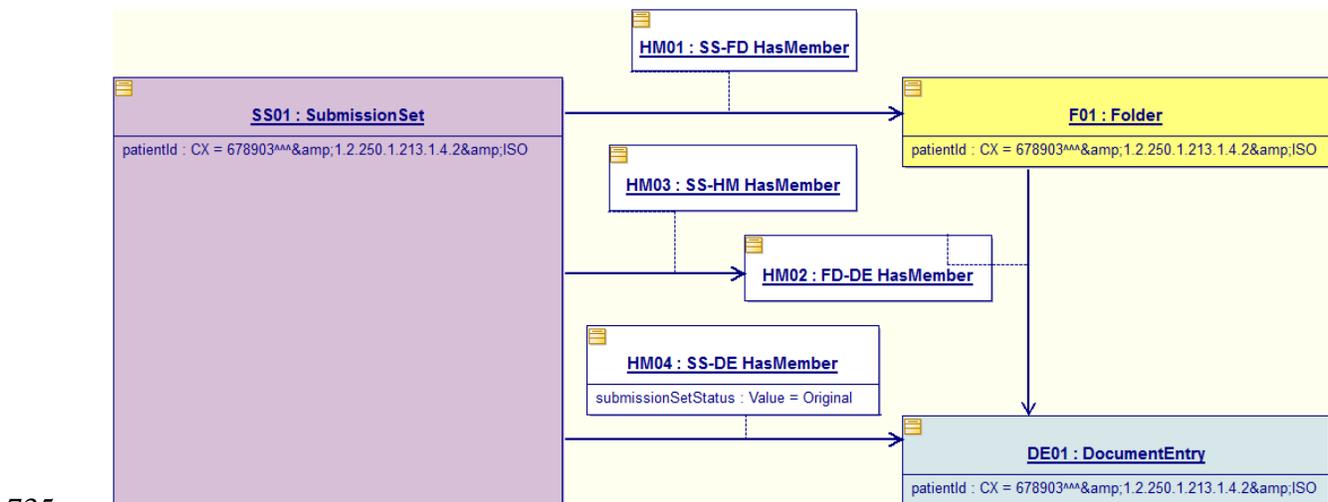
#### 4.2.2.1.5 Adding DocumentEntries to Folders

725 A DocumentEntry can be added to a Folder in one of four ways:

1. The DocumentEntry can be submitted as part of the Folder in a single submission.
2. The DocumentEntry and Folder are already present. The new submission makes the DocumentEntry a member of the Folder by adding the Association.
- 730 3. The DocumentEntry is already present. The new submission includes the Folder and the Association to make the DocumentEntry part of the Folder.
4. The Folder is already present. The new submission includes the DocumentEntry and the Association to make the DocumentEntry part of the Folder.

#### Scenario 1 - DocumentEntry submitted as part of the Folder in a single submission.

The simplest scenario submits all related objects in one submission set, as shown below.



735

Figure 4.2.2.1.5-1: Scenario 1 - DocumentEntry submitted as part of the Folder (Informative)

#### Scenario 2 – Add existing DocumentEntry to existing Folder

740

Documents can be placed in a Folder at a later date and time, as shown in Figures 4.2.2.1.5-2 and 4.2.2.1.5-3. In this case, the SubmissionSet SS03 which links to FD-DE HasMember will not have as member either the DocumentEntry or the Folder that correspond to the referenced document and Folder.

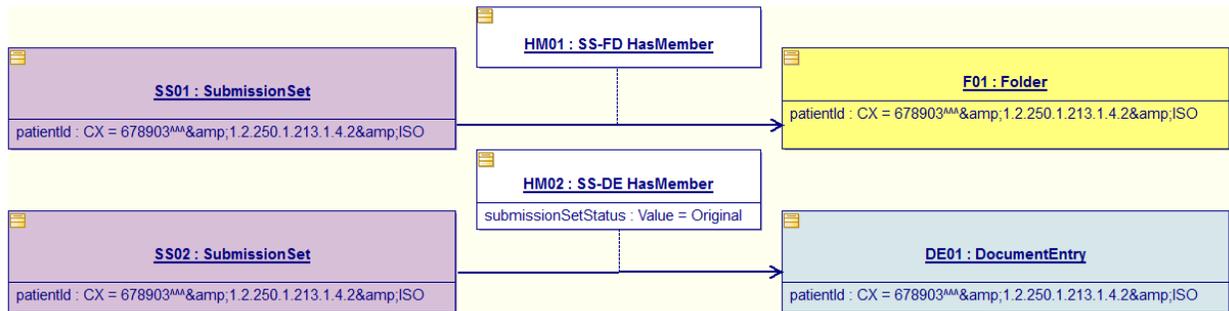
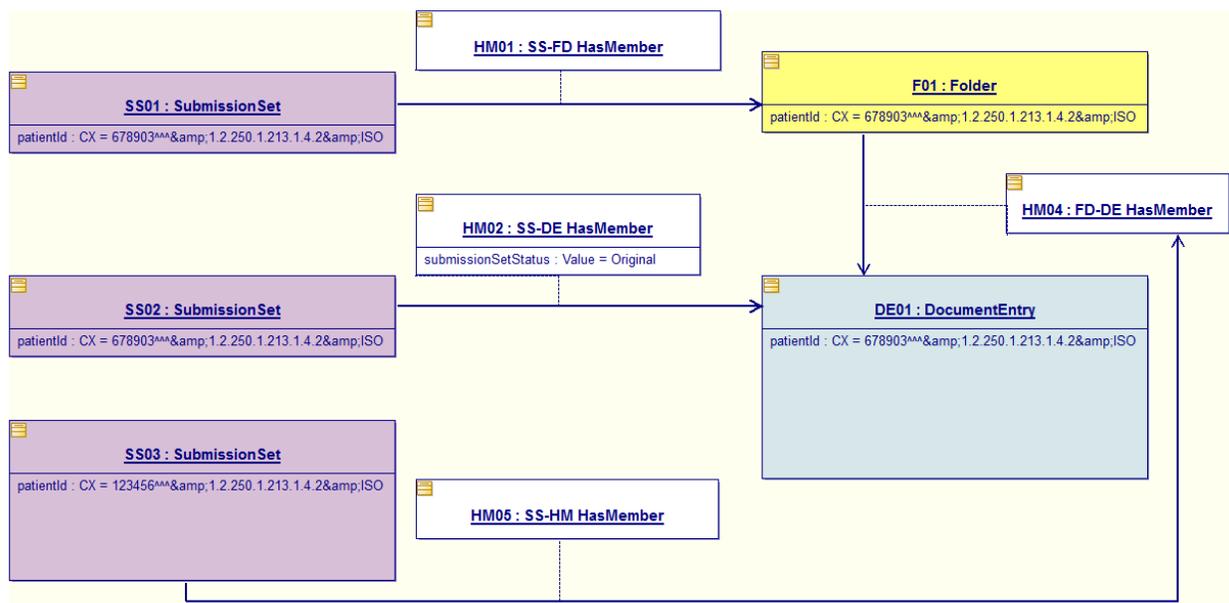


Figure 4.2.2.1.5-2: Scenario 2 - Existing DocumentEntry and existing Folder (Informative)



745

Figure 4.2.2.1.5-3: Scenario 2 - Add existing DocumentEntry to existing Folder (Informative)

### Scenario 3 – Folder submitted and existing DocumentEntry added to it

750

When a new Folder is submitted, an existing DocumentEntry can be added to that Folder. In this case, the SubmissionSet object will not contain the DocumentEntry metadata that correspond to the referenced document.

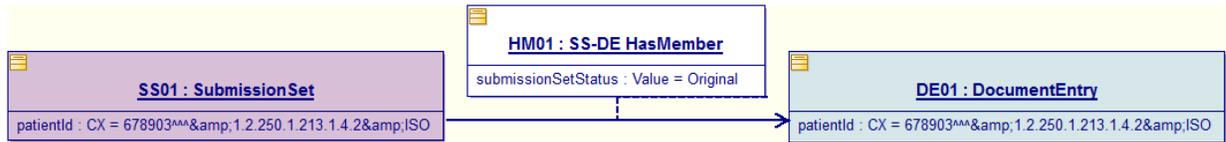
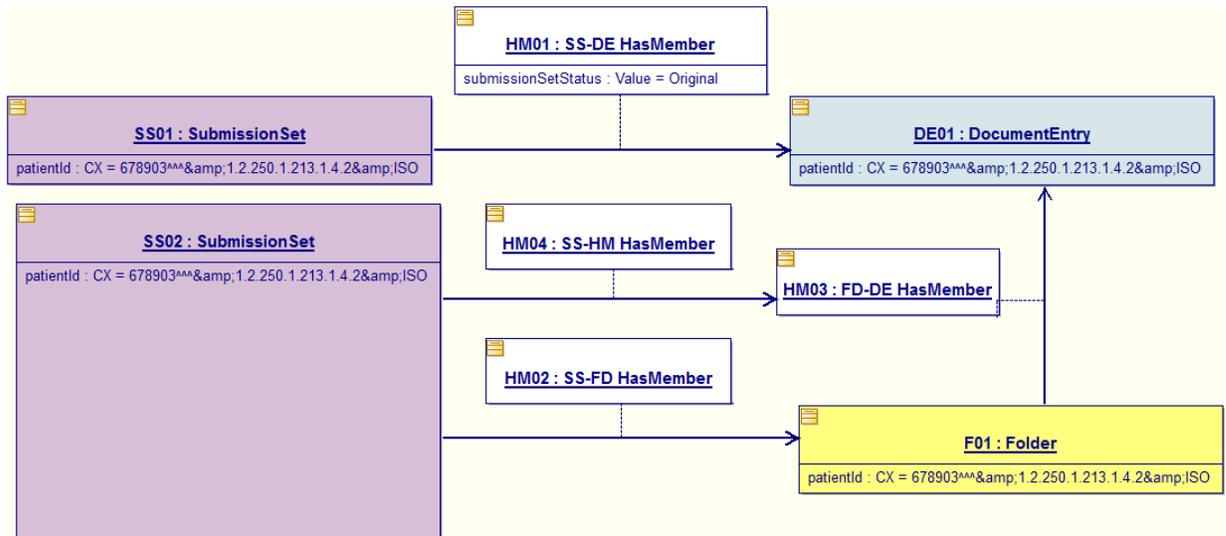


Figure 4.2.2.1.5-4: Scenario 3 - Starting point - Existing DocumentEntry (Informative)



755 Figure 4.2.2.1.5-5: Scenario 3 - Folder submitted and existing DocumentEntry added to it (Informative)

#### Scenario 4 – DocumentEntry submitted and added to existing Folder

When a new DocumentEntry is submitted, it can be added to an existing folder. In this case, the SubmissionSet object will not contain the Folder metadata that correspond to the referenced Folder.

760

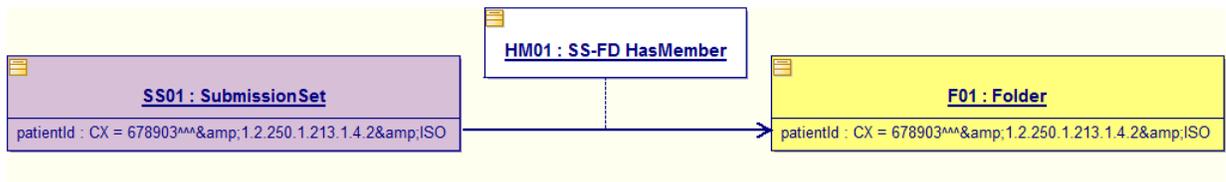
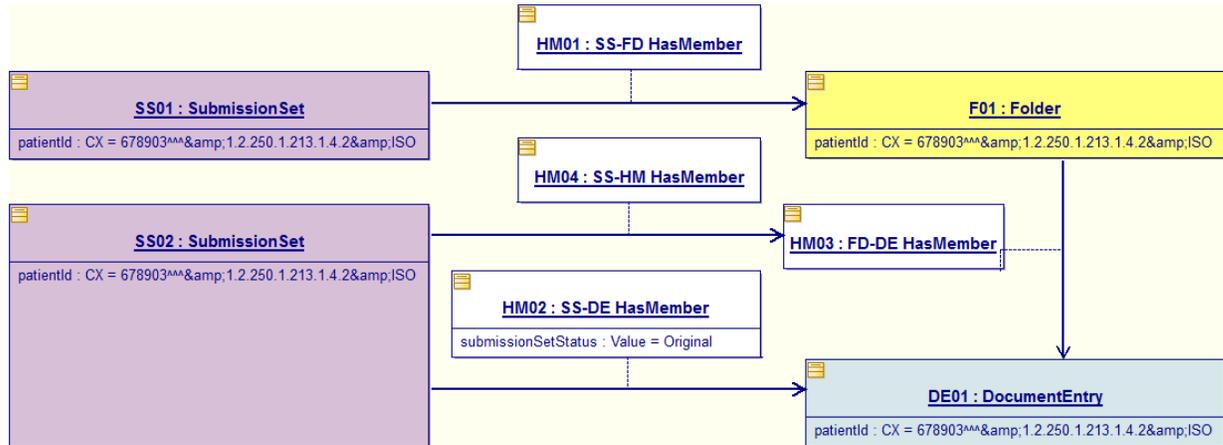


Figure 4.2.2.1.5-6: Scenario 4 - Starting point - Existing Folder (Informative)



765 **Figure 4.2.2.1.5-7: Scenario 4 - DocumentEntry submitted and added to existing Folder (Informative)**

#### 4.2.2.2 Document Relationship

770 Document relationships are association types used to link two DocumentEntry objects and declare a semantic relationship between them. Receiving a document relationship Association triggers specific processing in the receiving actor in some profiles. This is documented in the Expected Actions section of the relevant transaction in an IHE profile.

A document relationship refers to any of the relationships listed in Table 4.2.2.2-1 Document Relationships. The document relationship semantics (except Signs) are defined in HL7 Clinical Document Architecture (CDA) Release 2.

775 **Table 4.2.2.2-1: Document Relationships**

Relationship	Definition
APND (append)	The current document is an addendum to the parent document.
RPLC (replace)	The current document is a replacement of the parent document.
XFRM (transform)	The current document is a transformation of the parent document.
XFRM_RPLC (transform with replace)	The current document is both a transformation and a replacement of the parent document.
Signs	The current document is a Digital Signature which signs the parent document.
Is Snapshot of	The current document is a snapshot in time of the parent which shall be an on-demand document

Adapted from HL7 CDA Release 2, Committee Ballot 2

To create a document relationship in the registry, a creating entity shall submit:

A new Document (DocumentEntry).

An Association linking the new document to an existing document.

780 The Association type defines the document relationship. The new DocumentEntry and the  
Association must be submitted in the same SubmissionSet. The existing DocumentEntry must be  
an Approved object already in the registry (see Section 4.2.3.2.2,  
DocumentEntry.availabilityStatus). The identity (entryUUID) of the existing document will be  
785 known because the creating entity assigned the entryUUID prior to submission or discovered it  
(e.g., under XDS via ITI-18).

The targetObject attribute of the Association is the entryUUID representing the existing DocumentEntry known to the receiving entity. This entryUUID may have been discovered via Registry Stored Query [ITI-18], or other means.

790 Note to implementers: An XDS Document Source using saved UUIDs in future transactions can run into consistency problems. A second XDS Document Source can deprecate any document. Once a document is deprecated, new Associations to that document cannot be accepted by the XDS Document Registry. The saved UUID will point to the deprecated document and not the replacement document.

A submitting actor creates a document relationship by submitting a SubmissionSet containing:

**DocumentEntry** – this describes the new document being submitted

795 The uniqueId attribute shall be unique  
The entryUUID shall be unique or a symbolic Id as described in Section 4.2.3.1.5.

**Association** – this links the original DocumentEntry (already known to the receiving entity) with the new DocumentEntry being submitted.

800 The targetObject attribute of the Association object references the existing DocumentEntry known to the receiving entity.  
The sourceObject attribute of the Association object references the DocumentEntry contained in the SubmissionSet.

The Association Type is one of the relationships in Table 4.2.2.2-1: Document Relationships.

805 Associations of type XFRM, APND, RPLC, and XFRM\_RPLC may include documentation describing the Association (type of transformation, reason for replacement, etc.). If one of these is included, it shall be specified as a Classification on the Association as shown in the example below.

Example of a partial submission request:

```
810 <rim:Association
      associationType="urn:ihe:iti:2007:AssociationType:XFRM"
      sourceObject="source"
      targetObject="urn:uuid:XXX"
      objectType="urn:oasis:names:tc:ebxml-
815     regrep:ObjectType:RegistryObject:Association"
      id="IdExample_042"
    >
    <rim:Classification
      classificationScheme="urn:uuid:abd807a3-4432-4053-87b4-fd82c643d1f3"
820     classifiedObject="IdExample_042"
      id="IdExample_043"
      objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="LanguageExample"
    >
825   <rim:Name>
     <rim:LocalizedString value="Translation into LanguageExample" />
   </rim:Name>
   <rim:Slot name="codingScheme">
     <rim:ValueList>
830     <rim:Value>Example translation types</rim:Value>
     </rim:ValueList>
   </rim:Slot>
   </rim:Classification>
 </rim:Association>
835 <rim:ExtrinsicObject id="source"
      objectType=" urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1" />
  . . .
</rim:ExtrinsicObject>
```

840 Since the targetObject always references an object already in the Registry; it shall be coded in UUID format, as shown.

A stub DocumentEntry is shown to illustrate the id reference coding (Association sourceObject attribute referencing the id attribute of the DocumentEntry/ExtrinsicObject in the submission).

845 The new documents (related to original document by RPLC, APND, XFRM, or XFRM\_RPLC Associations) shall be assigned their own uniqueId attribute unrelated to the original document's uniqueId. See IHE ITI TF-2x: Appendix B "Definition of uniqueIds".

See IHE ITI TF-1: 10.4.10 for further detail on the use and meaning of document relationships.

#### 4.2.2.2.1 APND

850 The submission shall contain a new DocumentEntry and associated Document and an APND Association linking this new DocumentEntry with an existing DocumentEntry. This new Document/DocumentEntry forms an addendum to the existing Document/DocumentEntry. The APND Association may include documentation describing the addendum (e.g., reason for adding the addendum, etc.) The APND relationship leaves the original DocumentEntry with its availabilityStatus unchanged (Approved).

855

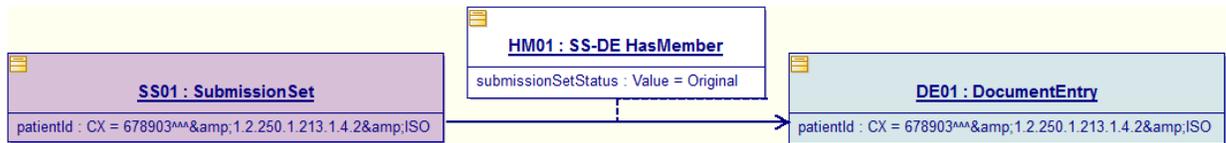


Figure 4.2.2.2.1-1: Starting Point (Informative)

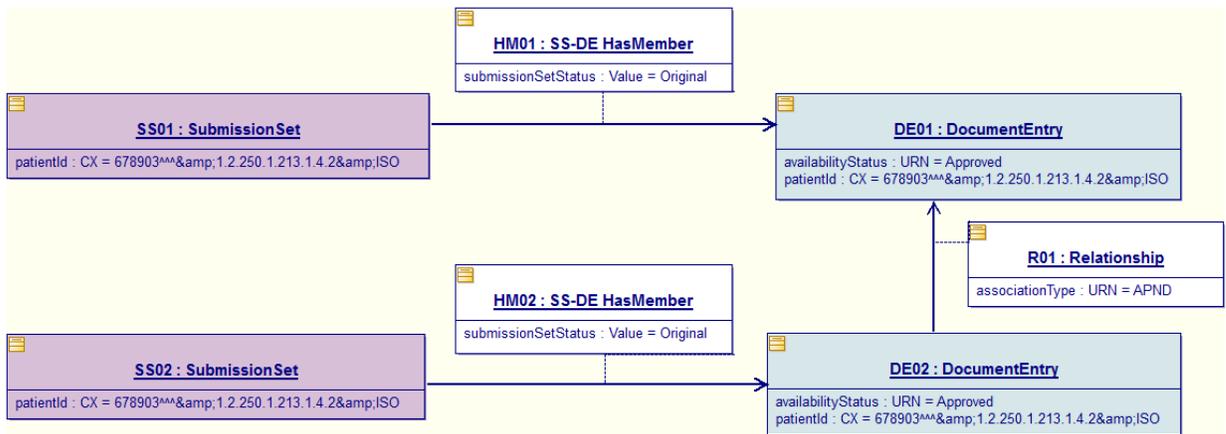


Figure 4.2.2.2.1-2: New DocumentEntry and Document forming an addendum to an existing DocumentEntry/Document (Informative)

860

### Interactions between XFRM and APND

A transformation (connected to original document with XFRM Association) is an alternate form of an original document. Therefore, a transformation shall not be appended (APND).

#### 4.2.2.2.2 XFRM

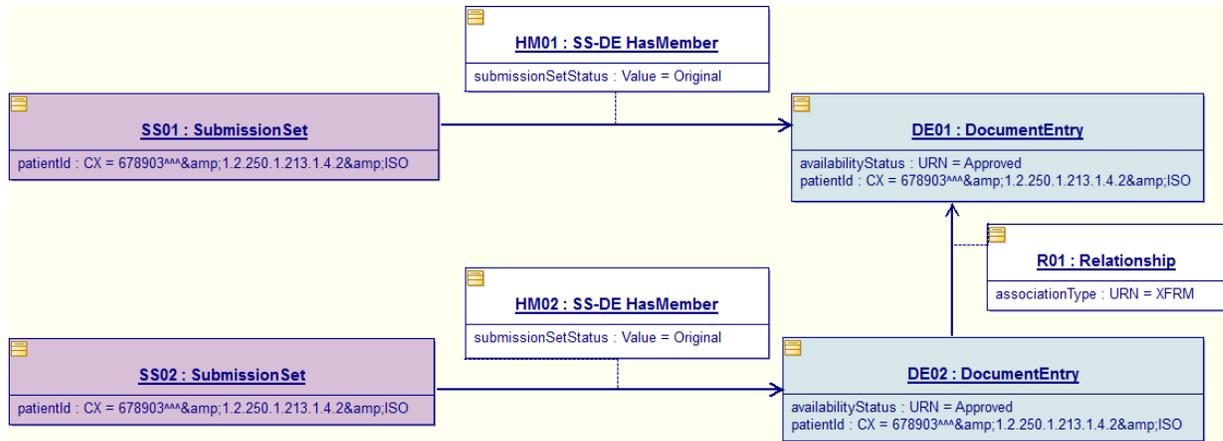
865

The submission shall contain a new DocumentEntry and associated Document and a XFRM Association linking this new DocumentEntry with an existing DocumentEntry. This new Document/DocumentEntry defines a transformation of the existing Document/DocumentEntry. The XFRM Association may include documentation describing the transformation (e.g., reason for transformation, etc.). The XFRM relationship leaves the original DocumentEntry with its availabilityStatus unchanged (Approved).

870



Figure 4.2.2.2.2-1: Starting Point (Informative)

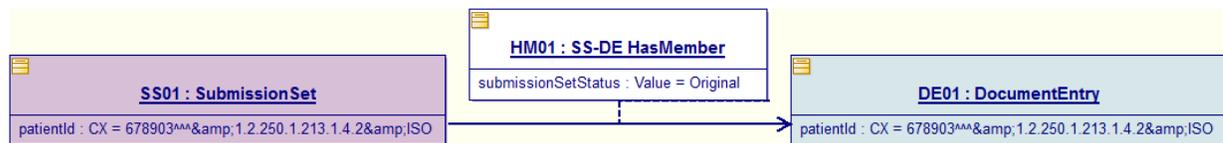


875

**Figure 4.2.2.2.2-2: New DocumentEntry/Document defining a transformation of an existing DocumentEntry/Document (Informative)**

#### 4.2.2.2.3 RPLC

880 The submission shall contain a new DocumentEntry and associated Document and a RPLC Association linking this new DocumentEntry with an existing DocumentEntry. The new DocumentEntry and Document are considered the approved version of the document; the existing DocumentEntry and Document become a deprecated version. The RPLC Association may include documentation describing the replacement (e.g., reason for replacement, etc.).



885

**Figure 4.2.2.2.3-1: Starting Point (Informative)**

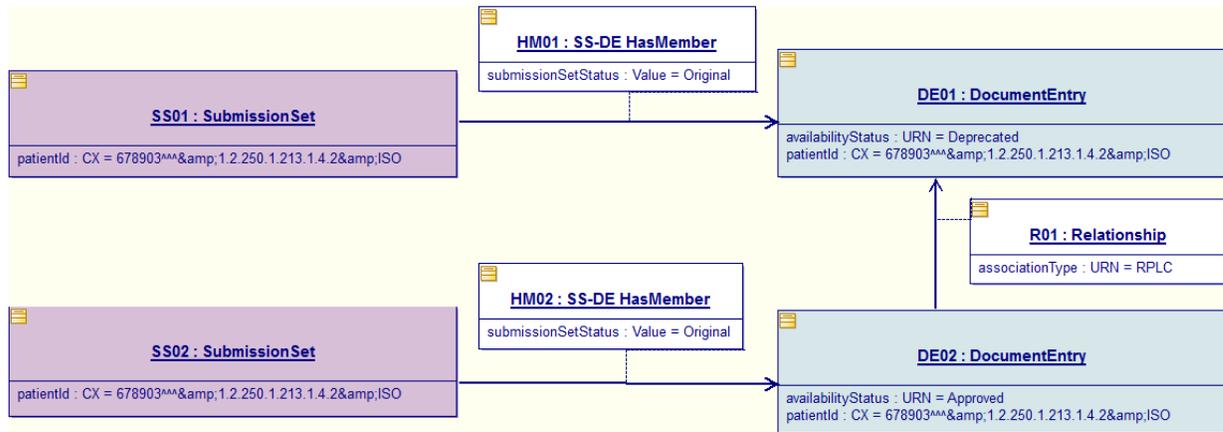


Figure 4.2.2.2.3-2: New DocumentEntry and Document replacing existing DocumentEntry/Document (Informative)

890

### RPLC interactions with other Associations

The submission of a RPLC relationship shall change the availabilityStatus of the original DocumentEntry to Deprecated. All transformations (XFRM) and addenda (APND) of the original document shall also be deprecated.

895

Figures 4.2.2.2.3-2 and 4.2.2.2.3-3 show a new DocumentEntry/Document replacing a transformation (XFRM) of an existing DocumentEntry/Document:

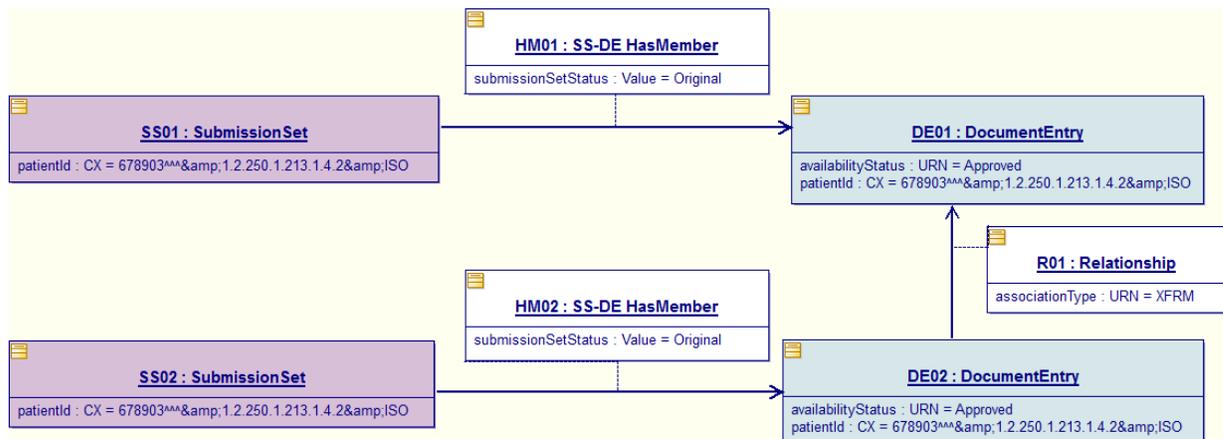
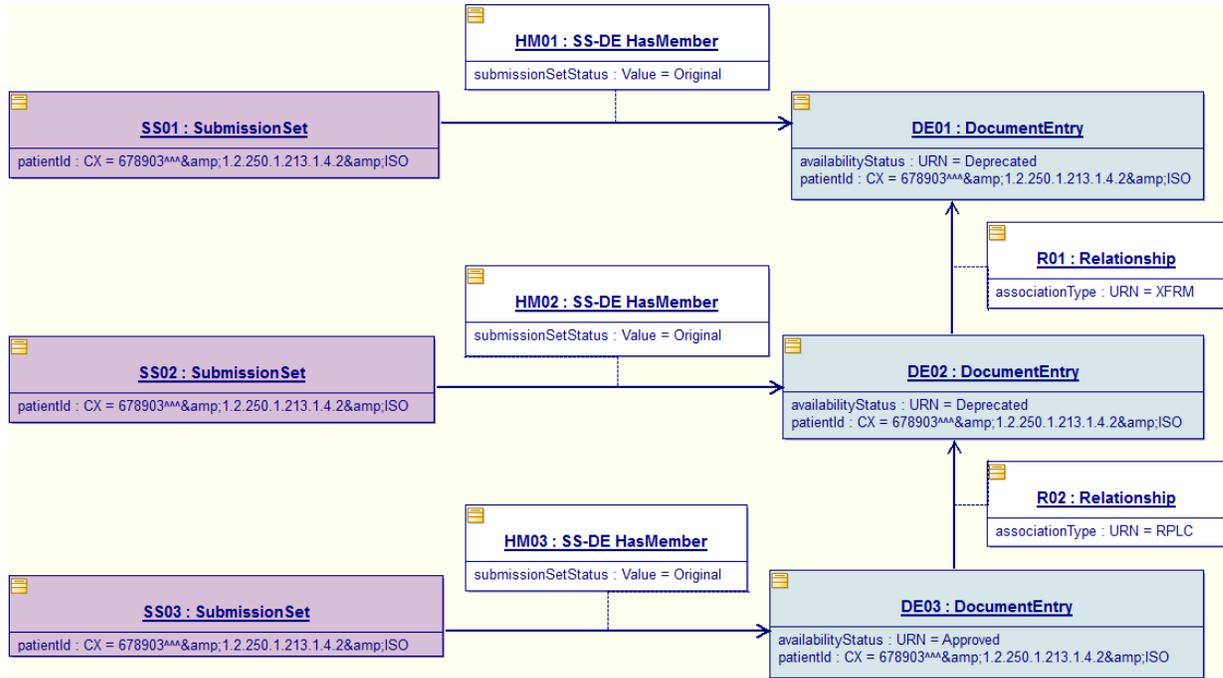


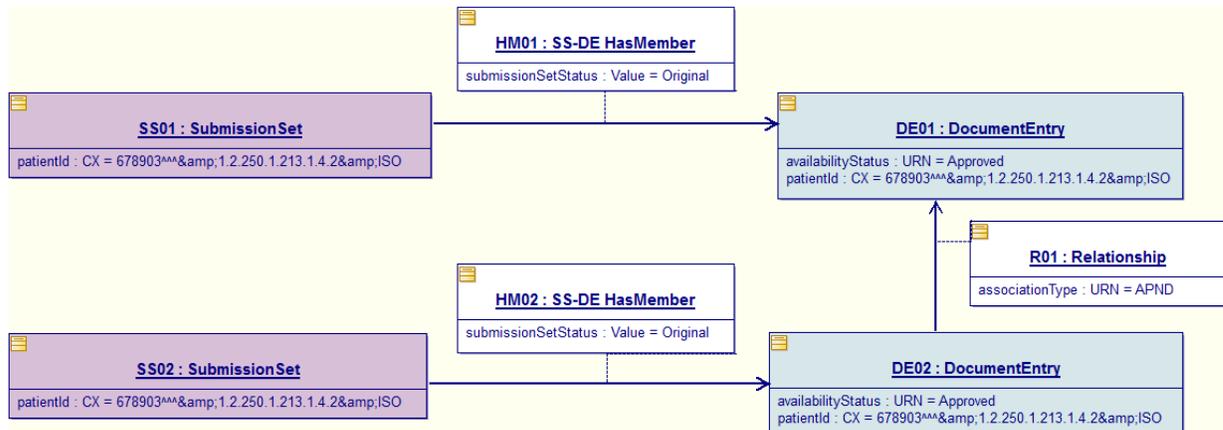
Figure 4.2.2.2.3-3: New DocumentEntry/Document defining a transformation of an existing DocumentEntry/Document – Starting point (Informative)

900

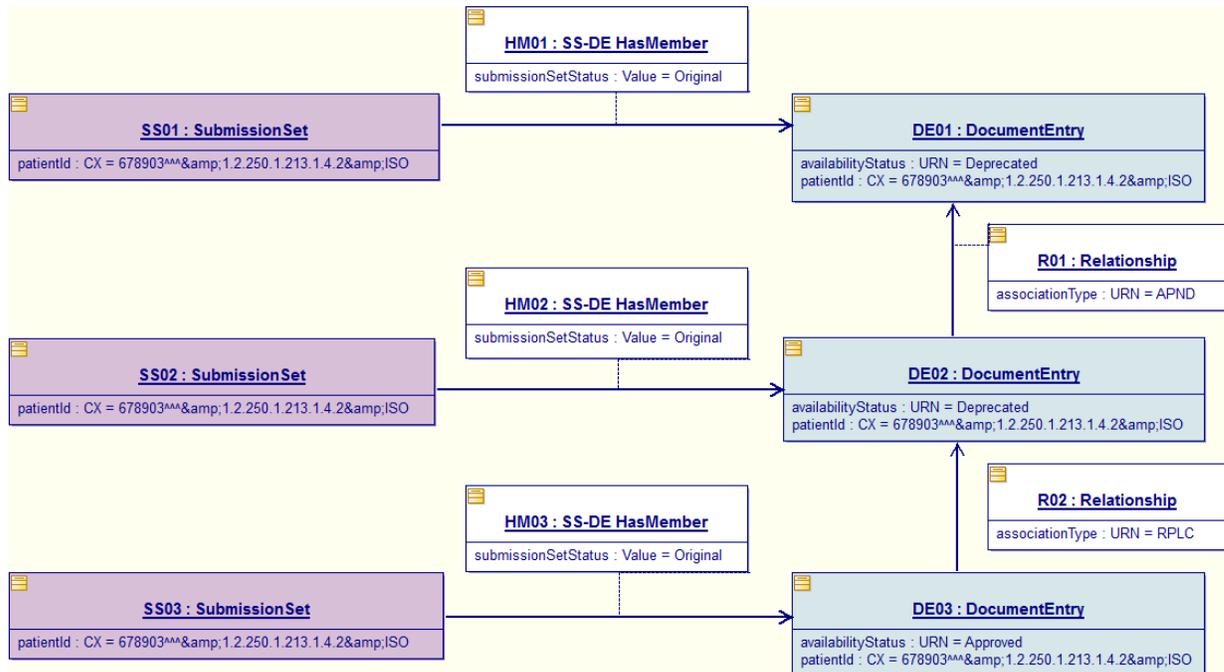


**Figure 4.2.2.2.3-4: New DocumentEntry/Document replacing a transformation (XFRM) of an existing DocumentEntry/Document (Informative)**

905 Figures 4.2.2.2.3-4 and 4.2.2.2.3-5 show a new DocumentEntry/Document replacing an addendum (APND) of an existing DocumentEntry/Document.



**Figure 4.2.2.2.3-5: New DocumentEntry and Document forming an addendum to an existing DocumentEntry/Document – Starting point (Informative)**



910

**Figure 4.2.2.3-6: New DocumentEntry/Document replacing an addendum (APND) of an existing DocumentEntry/Document (Informative)**

915

Only an Approved DocumentEntry is replaceable. The most recent version of a DocumentEntry carries an availabilityStatus of Approved while older versions carry an availabilityStatus of Deprecated.

A transformation (connected to original DocumentEntry with XFRM Association) is an alternate form of an original document. Therefore, a transformation is permitted to be replaced (RPLC).

920

When a DocumentEntry is replaced and that DocumentEntry is a member of one or more Folders, new FD-DE HasMember and SS-HM HasMember Associations shall be created by the receiving actor, connecting the replacement DocumentEntry to each Folder that held the original DocumentEntry as a member. The result is that a Folder contains both the original and replacement DocumentEntry differentiated by their availabilityStatus.

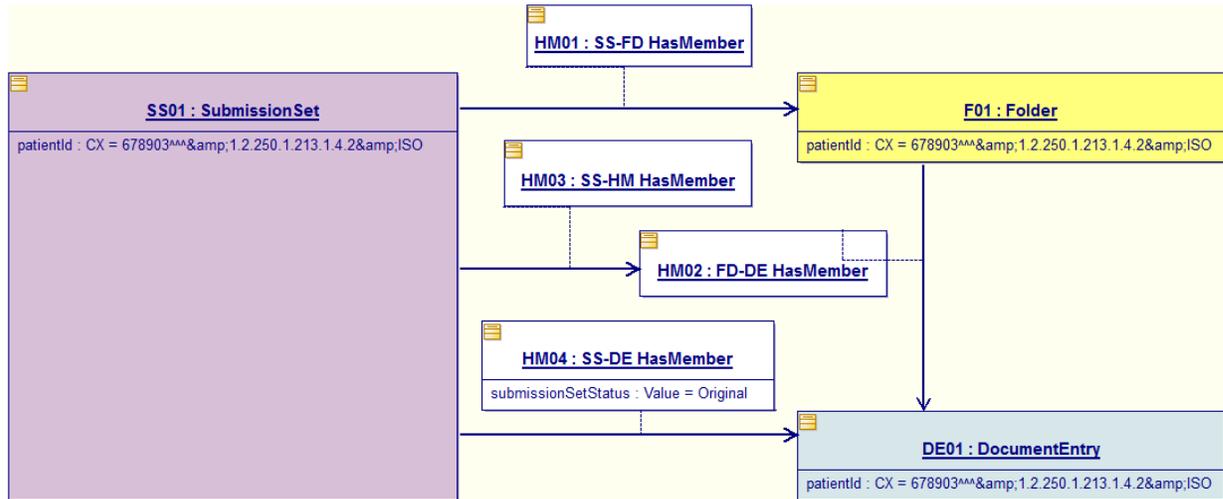


Figure 4.2.2.3-7: DocumentEntry submitted as part of the Folder in a single submission – Starting point (Informative)

925

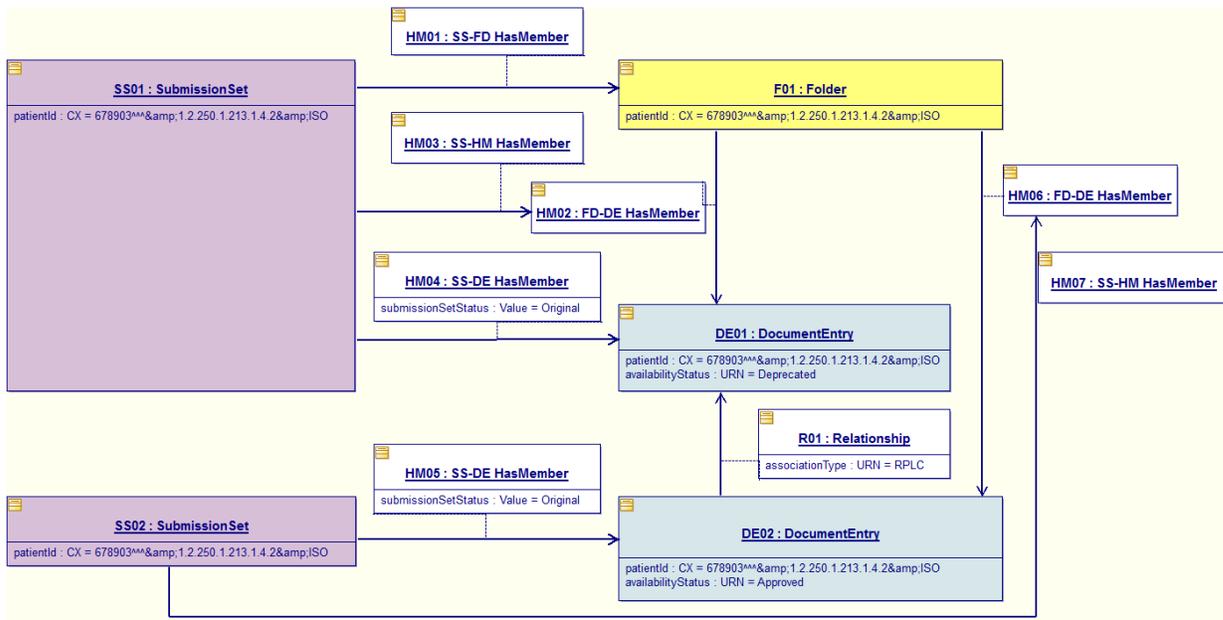


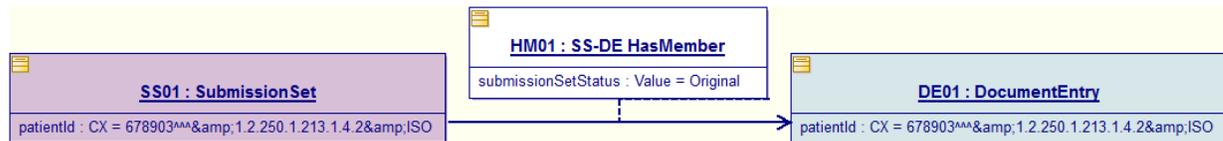
Figure 4.2.2.3-8: Submission of a new DocumentEntry replacing the DocumentEntry part of an existing Folder (Informative)

930

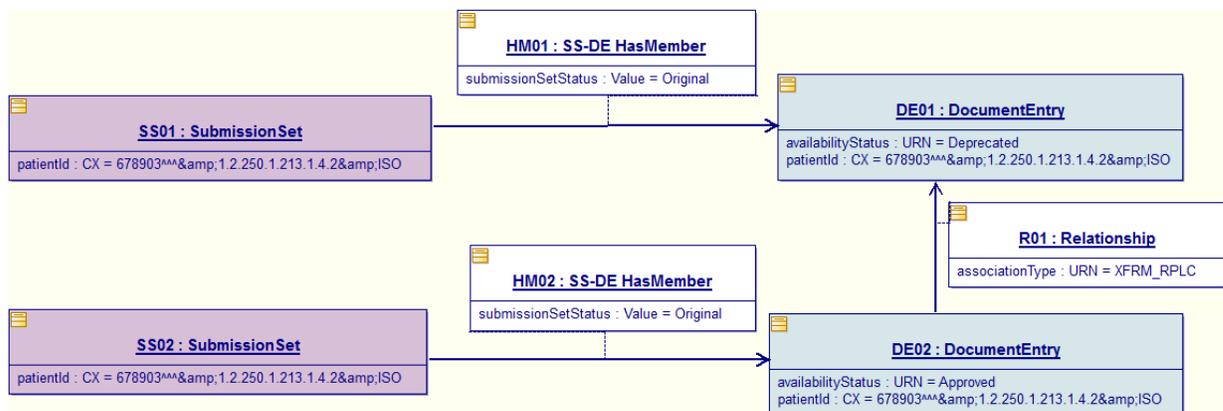
#### 4.2.2.2.4 XFRM\_RPLC

The submission contains a new DocumentEntry and associated Document and a XFRM\_RPLC Association linking this new DocumentEntry with an existing DocumentEntry. This new Document/DocumentEntry defines a transformation of the existing Document/DocumentEntry that replaces the existing Document/DocumentEntry. The XFRM\_RPLC Association may

935 include documentation describing the transformation/replacement (e.g., reason, etc.). The XFRM\_RPLC can be thought of as a RPLC followed immediately by a XFRM. All behavior associated with a RPLC association shall also apply to the XFRM\_RPLC association.



940 **Figure 4.2.2.4-1: Starting Point (Informative)**



**Figure 4.2.2.4-2: New DocumentEntry/Document defining a transformation of an existing DocumentEntry/Document that replaces this existing DocumentEntry/Document (Informative)**

945 **4.2.2.2.5 Signs**

The submission contains a new DocumentEntry and associated Document and a Signs Association linking this new DocumentEntry with an existing DocumentEntry. This new Document is a Digital Signature that signs the existing Document.

950 An eBRIM Association with associationType of *signs* shall be used to link a DocumentEntry representing a Digital Signature with the DocumentEntry representing the document being signed. Details of how Digital Signatures are represented are found in the ITI Digital Signatures content profile.

In constructing this Association, the attributes are:

**sourceObject:** references the DocumentEntry representing the Digital Signature

955 **targetObject:** references the DocumentEntry representing the document being signed

**associationType:** signs

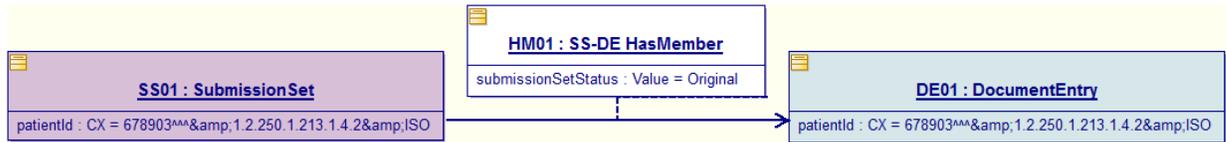


Figure 4.2.2.5-1: Starting Point (Informative)

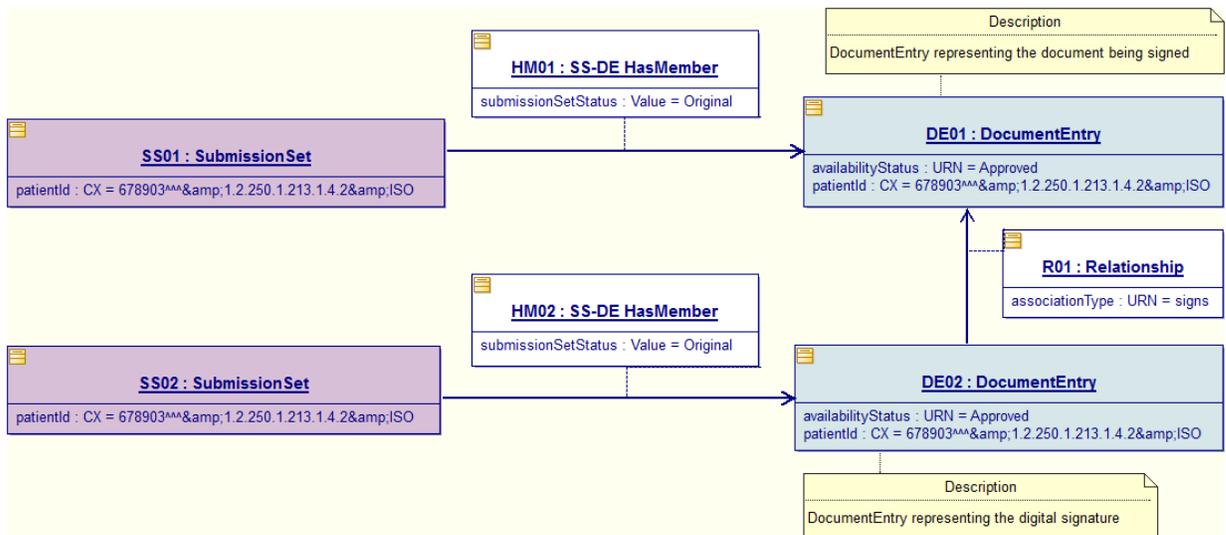


Figure 4.2.2.5-2: New DocumentEntry/Digital signature document signing the existing DocumentEntry/Document (Informative)

960

#### 4.2.2.2.6 IsSnapshotOf

965

When the content of an On-Demand DocumentEntry is retrieved, the retrieved version can be saved in a Document Repository as a Document and recorded in the Document Registry as a Stable DocumentEntry. When this happens, the saved version is linked to the On-Demand DocumentEntry through the IsSnapshotOf Association.

### Example IsSnapshotOf Association

```
970 <rim:Association
      associationType="urn:ihe:iti:2010:AssociationType:IsSnapshotOf"
      sourceObject="Snapshot"
      targetObject="urn:uuid:e0985823-dc50-45a5-a6c8-a11a829893bd" />
975 <rim:ExtrinsicObject id="Snapshot"
      objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
      . . .
</rim:ExtrinsicObject>
```

980 This Association shall be submitted together with the Stable DocumentEntry representing the snapshot of the On-Demand Document. The sourceObject attribute references the snapshot and the targetObject references the On-Demand DocumentEntry already in the Registry.

The Document Registry Actor shall verify upon submission:

- The sourceObject references a DocumentEntry in the submission
- The targetObject references a DocumentEntry in the Registry
- The targetObject DocumentEntry has availabilityStatus of Approved
- 985 • Verify that the objectType attribute of the sourceObject DocumentEntry is urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1 (Stable)
- Verify that the objectType attribute of the targetObject DocumentEntry is urn:uuid:34268e47-fdf5-41a6-ba33-82133c465248 (On-Demand)

990 Note: this example does not show the SubmissionSet object or the SubmissionSet to DocumentEntry HasMember association.

## 4.2.3 Metadata Attributes

This section details the coding of metadata attributes using in Document Sharing profiles.

### 4.2.3.1 General Information about Metadata Attributes

#### 4.2.3.1.1 Attribute Value Length

995 All attribute values are limited by the size specifications of ebRIM version 3.0. The ebRIM Slots are limited to 256 characters in length. Individual components of ebRIM Classification, ebRIM ExternalIdentifier, and ebRIM Name are strings, attributes, and Slots. Those that are ebRIM Slots are limited to 256 characters in length. The other components are naturally short (e.g., language).

#### 1000 4.2.3.1.2 Creating Coded Attributes

Many attributes of DocumentEntry, SubmissionSet, and Folder (see Table 4.2.3.2-1, Table 4.2.3.3-1 and Table 4.2.3.4-1) are coded attributes defined as ebRIM Classifications. Three details are required to describe a coded value:

1. Code Value – contains the assigned value of the code.
- 1005 2. Code Display Name - The display name used to represent the code (code values are not necessarily human-friendly). Must be non-zero length.
3. Coding Scheme - The name/identification of the coding scheme that the code comes from.

1010 These three values combine to define a single coded element. An example of a coded attribute in a submission request is:

```

1015 <rim:Classification
      classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-
e362475b143a"
      classifiedObject="ExampleDocument"
1020      id="IdExample_046"
      objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="10160-0">
      <rim:Name>
1025      <rim:LocalizedString value="History of Medication Use"/>
      </rim:Name>
      <rim:Slot name="codingScheme">
      <rim:ValueList>
      <rim:Value>2.16.840.1.113883.6.1</rim:Value>
      </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
  
```

1030 A code is constructed as a Classification object. The relevant parts of this classification are described in table 4.2.3.1.2-1 below.

**Table 4.2.3.1.2-1: Classifications and Descriptions**

Item	Description
Classification/@classificationScheme	This UUID defines which metadata attribute this Classification is supplying a value for. See Section 4.2.5 for a list of all values for this attribute. In the example above, this attribute indicates that the Classification contains a value for the DocumentEntry.classCode metadata attribute.
Classification/@classifiedObject	This references the object in metadata being classified. In the example this references the DocumentEntry object that the classCode value is for. Note that the Classification must also be embedded within the referenced object, thus providing dual context. This value must match the @id attribute of the containing ExtrinsicObject or RegistryPackage.
Classification/@id	UUID identifying this Classification.
Classification/@objectType	ebXML requires this for Query/Response type transactions but not for submissions. When required, contains the value shown "urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"
Classification/@nodeRepresentation	The code value. Required. The example above shows a code value of "10160-0".

Item	Description
Name/LocalizedString/@value	The code displayName. The example above shows a display name of "History of Medication Use".
Slot/@name	Fixed value that must always be the string "codingScheme".
Slot/ValueList/Value/text()	The code codingScheme. Shall have exactly one value. In the example above it is "2.16.840.1.113883.6.1".

This Classification shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

1035 The creating entity defines the local configuration for each coded metadata attribute. Specifically, it defines for each code:

**Name of the coding scheme** – shall be used in the codingScheme Slot.

**Value for the code** – shall be used in the nodeRepresentation attribute.

1040 **One or more display Name values for each code value** – A name shall be used in the Name element. It is expected, but not enforced, that the name should be semantically consistent with the code value. (This permits submitting using different local languages to a single Registry.)

Some coded attributes allow multiple values. EventCodeList is an example. These attributes may contain the letters 'List' in their name. These codes are XML encoded identically to the above example except the entire Classification element may be repeated to specify additional values.

1045 Note: the attribute DocumentEntry.languageCode attributed is not encoded as a coded attribute as shown above. See Table 4.2.3.1.7-2 for details.

1050 Note: The same code value may have multiple valid human readable representations in different languages or in different, but semantically equivalent, wording. An XDS Affinity Domain MAY restrict the displayName values it accepts from XDS Document Sources to a list of allowed values. It is the XDS Document Consumer's responsibility to resolve code values in metadata into human readable display values. A XDS Document Consumer may use the displayName found in the metadata provided by the XDS Document Source, or it may choose another local designation for the code, for example to translate it from English into French.

#### 4.2.3.1.3 Creating External Identifiers

1055 Several attributes of DocumentEntry, SubmissionSet and Folder (Table 4.2.3.2-1, Table, 4.2.3.3-1 and Table 4.2.3.4-1) are identifiers defined as ebRIM ExternalIdentifiers. These identifiers, such as patient IDs or document unique IDs are considered to be real-world identifiers that have global meaning external to an ebXML registry or other transaction.

1060 The identifiers consist of a single, opaque string stored in the value attribute of an ExternalIdentifier element. The meaning and format depend on the particular identifier. An example of an identifier in a submission request is:

```

1065 <rim:ExternalIdentifier
      identificationScheme="urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446"
      value="6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO"
      id="IdExample_051"
      objectType="urn:oasis:names:tc:ebxml-
1070 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="DocumentEntry01">
      <rim:Name>
        <rim:LocalizedString value="XSDSDocumentEntry.patientId"/>
      </rim:Name>
    </rim:ExternalIdentifier>
  
```

**Table 4.2.3.1.3-1: Identifiers and Descriptions**

Item	Description
ExternalIdentifier/@identificationScheme	Fixed value denoting the specific identifier. See Section 4.2.5 for a list of valid values.
ExternalIdentifier/@value	The identifier value.
ExternalIdentifier/@id	UUID identifying this ExternalIdentifier.
ExternalIdentifier/@objectType	ebXML requires this for Query/Response type transactions but not for submissions. When required, contains the value: "urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier"
ExternalIdentifier/@registryObject	This references the object in metadata being identified. In the example this references the DocumentEntry object that the patient identifier value is for. Note that the ExternalIdentifier element must also be embedded within the referenced object, thus providing dual context. This value must match the @id attribute of the containing ExtrinsicObject or RegistryPackage.
Name/LocalizedString/@value	The readable name for the identifier. Fixed value according to the particular identifier.

1075 This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

#### 4.2.3.1.4 Creating Author Attribute

The author attributes of DocumentEntry and SubmissionSet represent the humans and/or machines that authored the document or SubmissionSet. The author attribute defines a structure to hold its sub-attributes that are individually defined in Sections 4.2.3.1.4.1 through 4.2.3.1.4.5.

- 1080
- authorPerson – zero or one
  - authorInstitution – zero or more
  - authorRole – zero or more
  - authorSpecialty – zero or more
  - authorTelecommunication – zero or more

1085 At least an authorPerson, authorTelecommunication or authorInstitution sub-attribute shall be present when the author attribute is included in the metadata.

**Coding:**

1090 The author attribute is as an ebRIM Classification which contains sub-attributes in ebRIM Slots. An instance of the author Classification shall be considered a single author. Multiple authors are valid and are represented by multiple author ebRIM Classification objects. For the author metadata attribute, the value of classificationScheme is different for DocumentEntries and SubmissionSets; see Sections 4.2.3.2.1 and 4.2.3.3.1, respectively. The nodeRepresentation attribute shall be empty.

The following example shows the definition of a single author for a DocumentEntry.

1095

```
<rim:Classification
  classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d"
  classifiedObject="ExampleDocument"
  id="IdExample_045"
  objectType="urn:oasis:names:tc:ebxml-
  regrep:ObjectType:RegistryObject:Classification"
  nodeRepresentation=""><!-- nodeRepresentation intentionally left blank-->
  <rim:Slot name="authorPerson">
    <rim:ValueList>
      <rim:Value>^Welby^Marcus^^MD^Dr</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="authorInstitution">
    <rim:ValueList>
      <rim:Value>Some Hospital^^^^^^^1.2.3.5.8.9.1789.45</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="authorRole">
    <rim:ValueList>
      <rim:Value>name of role</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="authorSpecialty">
    <rim:ValueList>
      <rim:Value>specialty of author</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="authorTelecommunication">
    <rim:ValueList>
      <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value>
    </rim:ValueList>
  </rim:Slot>
</rim:Classification>
```

1130 The following example shows the definition of a single author for a SubmissionSet.

```
1135 <rim:Classification
      classificationScheme="urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d"
      classifiedObject="theSubmission"
      id="Id_045"
      objectType="urn:oasis:names:tc:ebxml-
1140 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation=""><!-- nodeRepresentation intentionally left blank-->
      <rim:Slot name="authorPerson">
        <rim:ValueList>
1145   <rim:Value>^Welby^Marcus^^MD^Dr</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="authorInstitution">
        <rim:ValueList>
1155   <rim:Value>Some Hospital^^^^^^^^^1.2.3.5.8.9.1789.45</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="authorRole">
        <rim:ValueList>
1160   <rim:Value>name of role</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="authorSpecialty">
        <rim:ValueList>
1165   <rim:Value>specialty of author</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="authorTelecommunication"> <!-- may be multivalued -->
        <rim:ValueList>
1170   <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value>
        </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
```

1165 The author Classification shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

#### 4.2.3.1.4.1 authorInstitution

##### Description:

1170 Represents a specific healthcare facility under which the human and/or machines authored the document or SubmissionSet. This is a sub-attribute of the author attribute.

##### Coding:

The format of the authorInstitution value is XON. See Table 4.2.3.1.7-2 for description of XON format.

1175 The value is coded as an optional ebRIM Slot within an author Classification. See Section 4.2.3.1.4 for an example of author Classification. There may be multiple values within the ebRIM Slot.

This example shows the authorInstitution Slot with two elements reflecting two names for the same organization.

```
1180 <rim:Slot name="authorInstitution">
      <rim:ValueList>
        <rim:Value>MercyHospital</rim:Value>
        <rim:Value>NY Mercy Hospital^^^^^^^^^1.2.3.9.1789.45</rim:Value>
      </rim:ValueList>
1185 </rim:Slot>
```

#### 4.2.3.1.4.2 authorPerson

##### Description:

1190 Represents the humans and/or machines that authored the document or SubmissionSet within the authorInstitution. The author may be the patient itself. This is a sub-attribute of the author attribute.

##### Coding:

The format of the authorPerson value is XCN; see Table 4.2.3.1.7-2 for description of XCN format.

1195 The value is coded as a required ebRIM Slot within an author Classification; see Section 4.2.3.1.4 for an example of author Classification. There may be only a single value within the ebRIM Slot.

This example shows the authorPerson Slot holding the required single value.

```
1200 <rim:Slot name="authorPerson">
      <rim:ValueList>
        <rim:Value>^Welby^Marcus^^MD^Dr</rim:Value>
      </rim:ValueList>
      </rim:Slot>
```

#### 4.2.3.1.4.3 authorRole

1205 **Description:**

A string that represents the role of the author with respect to the patient at the time the document or SubmissionSet was created. This is a sub-attribute of the author attribute. If this is a coded value, it should follow the CX encoding in Table 4.2.3.1.7-2

##### Coding:

1210 The format of the authorRole value is String.

The value is coded as an optional ebRIM Slot within an author Classification. See Section 4.2.3.1.4 for an example of author Classification. There may be multiple values within the ebRIM Slot.

```
1215 <rim:Slot name="authorSpecialty">
      <rim:ValueList>
        <rim:Value>Cardiologist</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

#### 1220 **4.2.3.1.4.4 authorSpecialty**

##### **Description:**

Represents a specific specialty within a healthcare facility under which the human and/or machines authored the document or SubmissionSet. This is a sub-attribute of the author attribute. If this is a coded value, it should follow the CX encoding in Table 4.2.3.1.7-2.

##### 1225 **Coding:**

The format of the authorSpecialty value is String.

The value is coded as an optional ebRIM Slot within an author Classification. See Section 4.2.3.1.4 for an example of author Classification. There may be multiple values within the ebRIM Slot.

```
1230 <rim:Slot name="authorSpecialty">
      <rim:ValueList>
        <rim:Value>Cardiology</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

1235

#### **4.2.3.1.4.5 authorTelecommunication**

##### **Description:**

Represents the telecommunications address (e.g., email) of the document or SubmissionSet author. This is a sub-attribute of author.

##### 1240 **Coding:**

The format of the authorTelecommunication value is XTN; see Table 4.2.3.1.7-2 for description of XTN format. This specification restricts the use to the following fields:

XTN.3 – The type of telecommunication address. For example, e-mail addresses will have this valued with "Internet".

1245 XTN.4 – the telecommunications address, e.g., name@example.com

No other fields shall be specified and both of these fields shall be present.

The value is coded as an optional ebRIM Slot within an author Classification. See Section 4.2.3.1.4 for an example of author Classification. There may be multiple values within the ebRIM Slot.

1250

```
<rim:Slot name="authorTelecommunication">
  <rim:ValueList>
    <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

1255

#### 4.2.3.1.5 UUIDs

UUIDs shall be formatted according to [RFC 4122](#). Furthermore, values 10 through 15 shall be formatted in hexadecimal using only lower case 'a'-'f'. An example of a properly formatted UUID is:

1260

```
urn:uuid:10b545ea-725c-446d-9b95-8aeb444eddf3
```

Alternatively, symbolic Ids may be used when referencing objects in the same submission request. A symbolic Id is any string which does not start with urn:uuid:. For example, in XDS, a Document Source may create a submission request using symbolic Ids; the Document Registry replaces these with assigned properly formatted UUIDs upon acceptance of the submission.

1265

Once a UUID-format Id value is assigned to a Registry Object, that value is permanent and shall not be changed.

#### 4.2.3.1.6 Extra Metadata Attributes

Metadata beyond what is specified in this volume may be specified by a creating entity. This extra metadata may be ignored by the recipient but its presence shall not cause an error.

1270

The following conditions shall apply.

1. All extra metadata content shall be in the form of Slots.
2. These Slots may be attached to SubmissionSet, DocumentEntry, Folder, or Association objects.
3. Receiving actors shall ignore extra metadata elements they do not understand.
- 1275 4. If an actor accepts extra metadata Slots (no warning on submission) then it shall return these Slots in query results.
5. The Name attribute of extra Slots shall conform to the following rules:
  - a) Name shall be a valid URN.
  - b) Name shall begin with 'urn:' prefix (formatted as a valid URN).
  - 1280 c) The prefix 'urn:ihe' shall not be used.
6. Note that ebRIM requires that the name of a Slot be unique within the containing object (DocumentEntry, SubmissionSet, Folder, or Association).

#### 4.2.3.1.7 Metadata Attribute Data types

1285 Several data types are used in the tables below describing the Document Entry, Folder and Submission Set metadata. These data types are derived from other standards, and encoded as described in Table 4.2.3.1.7-2.

For the data types derived from HL7 standards, IHE Document Sharing Profiles require that the default HL7 separators in Table 4.2.3.1.7-1 be used to represent the structure of HL7 v2.5 data types:

1290

**Table 4.2.3.1.7-1: HL7 Separators**

HL7 Separators	
Field Separator	
Component Separator	^
Subcomponent Separator	&
Repetition Separator	~

**Table 4.2.3.1.7-2: Data Types (previously Table 4.1-3)**

Data Type	Source Standard	Encoding Specification
Code	ITI TF	See Section 4.2.3.1.2
CX	HL7 V2.5 Identifier	<p>entifier. HL7 Identifier type CX consists of several components, but this specification restricts them to the use of two components, the Id Number, and the Assigning Authority (AA). The Assigning Authority identifies the "domain" over which the Id Number represents a unique entity. Furthermore, the AA is characterized by a Universal Id and Universal Id Type. In Document Sharing profiles, ISO Object Identifiers (see OID below) must be used as Universal Id. Therefore, Universal Id Type is always ISO. The required format is:</p> <p>^&amp;OIDofAA&amp;ISO</p> <p>ues/modifications in other components or subcomponents are allowed. Specifically, components 2 and 3 shall be empty as listed above.</p> <p>Example is:</p> <p>^^&amp;1.2.840.113619.6.197&amp;ISO</p> <p>Note that the '&amp;' character must be properly encoded in the XML content.</p>
CXi	HL7 V2 Identifier	<p>This is an identifier of a reference object, distinct from the use of CX for Patient Identifiers. HL7 Identifier type CX consists of several components.</p> <ul style="list-style-type: none"> <li>• CXi.1 shall be present and hold the identifier value.</li> <li>• CXi4 (Assigning Authority) shall be present when the identifier in CXi.1 is not globally unique and holds the identifier of the "domain" over which the ID Number represents a unique entity. It is formatted just like CX.4 in the CX datatype above.</li> <li>• CXi.5 (Identifier Type Code) shall be present and chosen from either a URN defined by IHE, or a locally defined value.</li> <li>• When the homeCommunityId is known, CX.6 shall be present and holds the homeCommunityId encoded as ISO. See CX.4 in the CX datatype above.</li> <li>• No other components shall be present.</li> </ul>

Data Type	Source Standard	Encoding Specification
		<p>The following IHE values for CXi.5 (Identifier Type Codes) are defined:</p> <p><b>urn:ihe:iti:xds:2013:uniqueId</b>  This code shall be used when the identifier is a uniqueID from Document Sharing metadata attribute.  For example if uniqueID equals “1.3.6367.3.7^11379”, and homeCommunityID is “1.2.3.4” then the CXi value is  11379^^^&amp;1.3.6367.3.7&amp;ISO^urn:ihe:iti:xds:2013:uniqueId^1.2.3.4&amp;ISO</p> <p><b>urn:ihe:iti:xds:2013:accession</b>  This code shall be used when the identifier is an accession number. It shall contain</p> <ul style="list-style-type: none"> <li>• The accession number and</li> <li>• for accession values that are not globally unique, the Assigning Authority shall be included.</li> </ul> <p>For example when the accession number has a value of “2013001” and the assigning authority is “1.2.3.4.5.6” then the CXi value is  2013001^^^&amp;1.2.3.4.5.6&amp;ISO^urn:ihe:iti:xds:2013:accession</p> <p><b>urn:ihe:iti:xds:2013:referral</b>  Referral number and assigning authority shall be present.  For example:  201300001^^^1.2.3.4.5.6^urn:ihe:iti:xds:2013:referral</p> <p><b>urn:ihe:iti:xds:2013:order</b>  Order number and assigning authority shall be present.  For example:  134467653^^^1.2.3.4.5.42.1^urn:ihe:iti:xds:2013:order</p> <p><b>urn:ihe:iti:xdw:2013:workflowId</b>  This code shall be used when the identifier is an XDW workflow identifier. The workflow identifier shall be an OID. Only the CXi.1 and CXi.5 component shall be present:  For example, if the workflow identifier is “2.16.840.1” the value of referenceIdList attribute is:  2.16.840.1^^^urn:ihe:iti:xdw:2013:workflowId</p>
DTM	HL7 V2.5 Date Time	<p>This is a date/time value, represented as precisely as possible. All date time values in the registry are stored using Coordinated Universal Time [UTC]. "UTC" implies that time shall be converted from/to the local time.  The format of these values is defined as the following regular expression:  YYYY[MM[DD[hh[mm[ss]]]]]</p> <p>Where:  YYYY is the four digit year i.e., 2006  MM is the two digit month 01-12, where Jan is 01, Feb is 02, etc.  DD is the two digit day of the month 01-31  HH is the two digit hour, 00-23, where 00 is midnight, 01 is 1 am, 12 is noon, 13 is 1 pm, etc.  mm is the two digit minute, 00-59</p>

Data Type	Source Standard	Encoding Specification
		<p>ss is the two digit seconds, 00-59</p> <p>The following are legal date time values with increasing precision representing the date and time January 2, 2005, 3:04:05am</p> <p>2005 200501 20050102 2005010203 200501020304 20050102030405</p>
Field	HL7 V2.5 Message Segment	<p>Specified as the Field identifier, followed by a pipe ( ) and then the data value characterized as a corresponding HL7 V2.5 data type as defined in HL7 standard. Note that if a Document Sharing data type is based on an HL7 data type, then the Document Sharing data type is used to represent the value.</p> <p>An example of field Patient Identifier List (the third field of PID segment) is as follows: PID-3 DTP-1^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp; ISO</p>
Integer	<a href="#">W3C XML Schema Part 2</a>	XML Schema Part 2: Data Types section 3.3.13
MIME Type	<a href="#">RFC 2046</a>	<a href="https://www.iana.org/assignments/media-types">https://www.iana.org/assignments/media-types</a>
OID	ISO Object Identifier	<p>An ISO Object identifier. Limited in length to 64 characters, and made up of characters from the set [0-9.]. It must start with an integer, and is followed by one or more additional integer values, separated by periods. Integers are characterized without leading 0 digits unless the value is zero. (e.g.,1.3.6.1.4.1.21367.2005.3.7)</p> <p>In the attribute tables below, when an OID format is specified, it shall follow the assignment and format rules defined for document UID in ITI TF-2x: Appendix B</p>
OID URN	<a href="#">RFC 3061</a> . OID in URN syntax	<a href="#">See RFC 3061</a> . An example is urn:oid: 1.3.6.1.4.1.21367.2005.3.7.
Predefined URN	<a href="#">RFC 2141</a>	Specific URNs are defined and assigned in this framework. An example is urn:ihe:iti:2007:ResponseStatusType:PartialSuccess
SHA1	<a href="#">RFC 3174</a>	Document hash calculated with SHA1 algorithm. The encoding is the Lexical Representation of hexBinary ([0-9a-fA-F]).
String	XML	XML Schema Part 2: Data Types section 3.2.1
URI	<a href="#">RFC 2616</a>	Uniform Resource Identifier
UTF-8	<a href="#">RFC 3629</a>	Unicode standard
UUID	<a href="#">RFC 4122</a>	<p>A DCE Universally Unique Identifier, represented in registry attributes using the URN syntax for UUIDs e.g.,urn:uuid:9e0110f8-4748-4f1e-b0a8-cecae32209c7</p> <p>Some Document Sharing profiles may allow use of symbolic Ids in certain conditions or locations.</p>
XCN	HL7 V2.5 Extended Person Name	<p>This data type describes a person along with the identifier by which he is known in some domain (e.g., the XDS affinity domain), using the HL7 v2.5 XCN data type. This data type contains, amongst others:</p> <ul style="list-style-type: none"> <li>Identifier</li> <li>Last Name</li> <li>First Name</li> <li>Second and Further Given Names</li> </ul>



Data Type	Source Standard	Encoding Specification
		No other fields shall be specified. The XTN data type in Document Sharing metadata results in a valid encoding of an HL7 V2.5 XTN data type. Example: <a href="mailto:^^Internet^radiology@healthcare.example.org">^^Internet^radiology@healthcare.example.org</a>

1295 **4.2.3.1.8 General format of DocumentEntry, Folder and SubmissionSet attribute tables**

The metadata attribute definition tables each have five columns:

1. The first column contains the name used to refer to the attribute in IHE documentation. It is not always related to the way the attribute is coded in eBRIM.
- 1300 2. The second column contains a brief description of the attribute.
3. The third column identifies how the value of the attribute is coded. Table 4.2.3.1.7-2 lists all the possible types of data. In isolated instances a value may be complex enough to require a separate section to describe, in which case a reference to that section is included. For example, author.
- 1305 4. The fourth column describes where to find the encoding of the attribute within the DocumentEntry/Folder/SubmissionSet. In other words, how the value of the attribute is held within the enclosing object. The possible values in this column are:
  - **ebRIM Classification** – indicates that the attribute is coded as a classification object, see Section 4.2.3.1.2.
  - 1310 • **XML attribute** – indicates that the attribute is the value of an XML attribute of the enclosing object. For example:

```
<rim:ExtrinsicObject ... id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6" ... >
```

1315 Showing the entryUUID attribute coded as a value of the *idXML* attribute of the ExtrinsicObject representing the DocumentEntry. Note that the name of the attribute – entryUUID – is not referenced in its actual representation.

- **ebRIM Slot**– indicates that the attribute is encoded within a slot. For example:

```
<rim:ExtrinsicObject ...>
<rim:Slot name="creationTime">
  <rim:ValueList>
    <rim:Value>20041225212010</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

1325 Showing the creationTime attribute within a Slot of the ExtrinsicObject representing the Document Entry. In this case the name of the slot is the attribute name.

- **ebRIM ExternalIdentifier**– indicates the value is held within an ExternalIdentifier object, using the XML element rim:ExternalIdentifier, providing an additional identifier to a registry object (see Section 4.2.3.1.3).
- **ebRIM Name**– indicates this attribute is held in an ebRIM Name object. For example the title attribute is:

1330

```
<rim:Name>  
  <rim:LocalizedString value="ExampleTitle"/>  
</rim:Name>.
```

- **ebRIM Description**– indicates this attribute is held in an ebRIM Description object. For example the comment attribute is:

1335

```
<rim:Description>  
  <rim:LocalizedString value="ExampleComment"/>  
</rim:Description>.
```

1340

1. The fifth column links to a section which describes the attribute in more detail, including further detail and examples regarding its use and its coding.

#### 4.2.3.1.9 Metadata Attribute Cardinality

Metadata attributes have several dimensions of cardinality.

1345

One dimension is whether or not any value is required, or the attribute is optional. This dimension is dependent on the profile, actor and transaction in which the attribute is being specified and is detailed in Sections 4.2.3.2, 4.2.3.3 and 4.2.3.4.

Another dimension is the required handling of the receiving actor. This is specified within each transaction, or in the case of XDS Document Registry, Section 4.3.1.2.

1350

The last dimension of cardinality is whether the attribute can contain multiple values and, when multiple values are allowed, how multiple values are expressed. This is specified within the text explaining each of the attributes in Sections 4.2.3.2.1 through 4.2.3.2.27 for DocumentEntry, Sections 4.2.3.3.1 through 4.2.3.3.12 for SubmissionSet, and Sections 4.2.3.4.1 through 4.2.3.4.9 for Folder. The text indicates whether multiple values are allowed and, if they are allowed, how to express them. For example, for codes expressed using a Classification element, if multiple values are allowed they are coded by specifying multiple Classification elements.

1355

#### 4.2.3.2 DocumentEntry Attributes

The following metadata attributes shall be used to describe a Document Sharing DocumentEntry. Optionality is determined by specific transaction requirements; see Section 4.2.3.1.9.

1360

Each attribute shown below is an attribute on the DocumentEntry object. The attribute name is defined with a prefix of the object type of DocumentEntry when referenced by other objects, for example DocumentEntry.patientId.

Please see Section 4.2.3.1.8 for the general format of DocumentEntry, Folder and SubmissionSet attribute tables.

1365

**Table 4.2.3.2-1: DocumentEntry Metadata Attribute Definition** (previously Table 4.1-5)

DocumentEntry Metadata Attribute	Description	Data Type (Table 4.2.3.1.7-2)	Coding (Section 4.2.3.1.8)	Detail (See Section)
author	The humans and/or machines that authored the document. This attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty and authorTelecommunication.	See Section 4.2.3.2.1	ebRIM Classification	4.2.3.2.1
availabilityStatus	The lifecycle status of the DocumentEntry	Predefined URN	XML attribute	4.2.3.2.2
classCode	The code specifying the high-level use classification of the document type (e.g., Report, Summary, Images, Treatment Plan, Patient Preferences, Workflow).	Code	ebRIM Classification	4.2.3.2.3
comments	Comments associated with the Document.	String	ebRIM Description	4.2.3.2.4
confidentialityCode	The code specifying the level of confidentiality of the Document.	Code	ebRIM Classification	4.2.3.2.5
creationTime	The time the author created the document. Shall have a single value.	DTM	ebRIM Slot	4.2.3.2.6
entryUUID	A globally unique identifier used to identify the entry.	UUID	XML attribute	4.2.3.2.7
eventCodeList	This list of codes represents the main clinical acts, such as a colonoscopy or an appendectomy, being documented.	Code	ebRIM Classification	4.2.3.2.8
formatCode	Code globally uniquely specifying the format of the document.	Code	ebRIM Classification	4.2.3.2.9
hash	Hash of the Document itself.	SHA1 hash	ebRIM Slot	4.2.3.2.10
healthcareFacility TypeCode	This code represents the type of organizational setting of the clinical encounter during which the documented act occurred.	Code	ebRIM Classification	4.2.3.2.11
homeCommunityId	A globally unique identifier for a community.	OID URN	home XML attribute	4.2.3.2.12
languageCode	Specifies the human language of character data in the document. Using codes from IETF <a href="#">RFC 5646</a>	String	ebRIM Slot	4.2.3.2.13
legalAuthenticator	Represents a participant who has legally authenticated or attested the document within the authorInstitution.	XCN	ebRIM Slot	4.2.3.2.14
limitedMetadata	Indicates whether the Document Entry was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source.	4.2.3.2.29	ebRIM Classification	4.2.3.2.29
mime Type	MIME type of the document.	MIME type	XML attribute	4.2.3.2.15
objectType	The type of DocumentEntry	UUID	XML attribute	4.2.3.2.30
patientId	The patientId represents the subject of care of the document.	CX	ebRIM ExternalIdentifier	4.2.3.1.3

DocumentEntry Metadata Attribute	Description	Data Type (Table 4.2.3.1.7-2)	Coding (Section 4.2.3.1.8)	Detail (See Section)
practiceSettingCode	The code specifying the clinical specialty where the act that resulted in the document was performed (e.g., Family Practice, Laboratory, Radiology).	Code	ebRIM Classification	4.2.3.2.17
referenceIdList	A list of identifiers that apply to the document.	CXi	ebRIM Slot	4.2.3.2.28
repositoryUniqueId	The globally unique identifier of the repository where the document is stored.	OID	ebRIM Slot	4.2.3.2.18
serviceStartTime	The start time the service being documented took place.	DTM	ebRIM Slot	4.2.3.2.19
serviceStopTime	The stop time the service being documented took place.	DTM	ebRIM Slot	4.2.3.2.20
size	Size in bytes of the document.	Integer	ebRIM Slot	4.2.3.2.21
sourcePatientId	The sourcePatientId represents the subject of care medical record Identifier (e.g., Patient Id) in the local patient Identifier Domain of the Document Source.	CX	ebRIM Slot	4.2.3.2.22
sourcePatientInfo	This attribute contains demographic information of the patient to whose medical record this document belongs.	CX	ebRIM Slot	4.2.3.2.23
title	The title of the document.	UTF-8	ebRIM Name	4.2.3.2.24
typeCode	The code specifying the precise type of document from the user perspective (e.g., LOINC code).	Code	ebRIM Classification	4.2.3.2.25
uniqueId	The globally unique identifier assigned by the document creator to this document.	OID	ebRIM ExternalIdentifier	4.2.3.1.3
URI	The URI for the document.	URI	ebRIM Slot	4.2.3.2.27

#### 4.2.3.2.1 DocumentEntry.author

##### Description:

Represents the humans and/or machines that authored the document. See Section 4.2.3.1.4 for details on creating the structure.

1370 The classificationScheme shall be urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d

#### 4.2.3.2.2 DocumentEntry.availabilityStatus

##### Description:

Represents the status of the DocumentEntry. A DocumentEntry shall have one of two availability statuses:

1375 Approved: The document is available for patient care.

Deprecated: The document is obsolete.

1380 This attribute is typically omitted in a submission of new documents. If present in a submission, the submitted value is ignored. It is always set to Approved as a result of the successful submission of new documents. It may be changed to Deprecated under the primary responsibility of the creating entity.

**Coding:**

The format of the availabilityStatus value is a URN.

1385 In a query response the value is coded in the status attribute of the ExtrinsicObject representing the DocumentEntry and shall be "urn:oasis:names:tc:ebxml-regrep:StatusType:Approved" or "urn:oasis:names:tc:ebxml-regrep:StatusType:Deprecated". The example below shows the status attribute.

```
1390 <ExtrinsicObject
      id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855"
      objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
      status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
    >
    ...
```

#### 4.2.3.2.3 DocumentEntry.classCode

1395 **Description:**

The code specifying the high-level use classification of the document type (e.g., Report, Summary, Images, Treatment Plan, Patient Preferences, Workflow). The typeCode specifies the precise type of document from the user perspective. Valid values for classCode attribute are specified by the policies of the creating entity. It is recommended that the creating entity draws these values from a coding scheme providing a coarse level of granularity (about 10 to 100 entries). For example, XDS specifies that the XDS Affinity Domain will establish this list.

**Coding:**

1405 There shall be exactly zero or one eBRIM Classification containing a classCode for any DocumentEntry. See Section 4.2.3.1.2 for a description of coding an eBRIM Classification. For the classCode metadata attribute, the classificationScheme shall be urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a.

The following example specifies classCode=10160-0 with display name "History of Medication Use" and coding scheme "2.16.840.1.113883.6.1" for the DocumentEntry labeled "ExampleDocument".

1410

```
1415 <rim:Classification
      classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"
      classifiedObject="ExampleDocument"
      id="IdExample_046"
      objectType="urn:oasis:names:tc:ebxml-
1420 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="10160-0"
      >
      <rim:Name>
      <rim:LocalizedString
        value="History of Medication Use"/>
      </rim:Name>
      <rim:Slot name="codingScheme">
1425 <rim:ValueList>
        <rim:Value>2.16.840.1.113883.6.1</rim:Value>
      </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
```

#### 4.2.3.2.4 DocumentEntry.comments

##### 1430 **Description:**

Contains comments associated with the document.

##### **Coding:**

Max length is unbounded.

1435 The value of the comments attribute is coded in XML as the "value" attribute of the LocalizedString element within the ebRIM Description structure. There can be at most one ebRIM Description structure per DocumentEntry.

The following example shows a comment for the document.

```
1440 <rim:Description>
      <rim:LocalizedString value = "comment associated with the Document"/>
    </rim:Description>
```

#### 4.2.3.2.5 DocumentEntry.confidentialityCode

##### **Description:**

1445 The code specifying the security and privacy tags of the document. These codes are set by policy of the participants in the exchange, e.g., XDS affinity domain. confidentialityCode is part of a codification scheme.

1450 The confidentialityCode can carry multiple vocabulary items. HL7 has developed an understanding of security and privacy tags that might be desirable in a Document Sharing environment, called HL7 Healthcare Privacy and Security Classification System (HCS). The following specification is recommended but not mandated by IHE, as the vocabulary bindings are an administrative domain responsibility. The use of this method is up to the policy domain

such as the XDS Affinity Domain or other Trust Domain where all parties including sender and recipients are trusted to appropriately tag and enforce.

- [1...1] Confidentiality Security Classification Label Field
- 1455 • [0...\*] Sensitivity Security Category Label Field
- [0...\*] Compartment Security Category Label Field
- [0...\*] Integrity Security Category Label Field
- [0...\*] Handling Caveat Security Category Field

1460 In the HL7 Healthcare Privacy and Security Classification System In the HL7 Healthcare Privacy and Security Classification (HCS) there are code systems specific to Confidentiality, Sensitivity, Integrity, and Handling Caveats. Some values would come from a local vocabulary as they are related to workflow roles and special projects.

1465 The decision to include a code is the responsibility of the publisher/sender (e.g., Access Control decision) and is dependent on the Policy rules and Trust Framework in place for the exchange. Use of Sensitivity tags expose the nature of the sensitivity and should be used only when the end-to-end confidentiality of the tags can be assured.

When using the HL7 Healthcare Privacy and Security Classification System (HCS):

- 1470 • The confidentialityCode SHALL contain exactly one value from the HL7 code system V:Confidentiality (@codeSystem="2.16.840.1.113883.5.25" i.e., U, L, M, N, R, or V), to indicate the Confidentiality coding of the content.
  - The value represents the most restrictive content in the identified document (aka. High water mark).
- 1475 • The confidentialityCode MAY contain values from the HL7 code system V:InformationSensitivityPolicy (@codeSystem="2.16.840.1.113883.1.11.20428"), to indicate the Sensitivity coding of the content.
  - Multiple values are all applicable to the content. This means that a consuming system/user must have rights to all Sensitivity classes indicated.
- 1480 • The confidentialityCode MAY contain values from the HL7 code system V:Compartment (@codeSystem="2.16.840.1.113883.1.11.20478"), to indicate the Compartment of the content.
  - Multiple values are all applicable to the content. This means that a consuming system/user must have rights to all Compartments indicated.
- 1485 • The confidentialityCode MAY contain values from the HL7 code system V:SecurityIntegrityObservationValue (@codeSystem="2.16.840.1.113883.1.11.20469"), to indicate the Integrity of the content.
  - Multiple values are all applicable to the content.

- The confidentialityCode MAY contain values from the HL7 code system V:SecurityControlObservationValue (@codeSystem="2.16.840.1.113883.1.11.20471"), to address the Handling Caveats that must be applied to the use of the content.
- 1490
- Multiple values all applicable to the content. A consuming system must enforce all Handling Caveats indicated.
  - Other value-sets and codesystems MAY be used as agreed between the communicating partners.

**Coding:**

- 1495 Each confidentialityCode is coded within an ebRIM Classification object. See Section 4.2.3.1.2 for a description of coding an ebRIM Classification. There shall be zero or more ebRIM Classification containing a confidentiality code (some profiles require at least one). Multiple values of confidentialityCode are coded by specifying multiple classification objects. For the confidentialityCode metadata attribute, the classificationScheme shall be urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f.
- 1500

The following example shows two confidentialityCodes values. It specifies normal confidentiality with code "N", display name "Normal Clinical Data", and coding scheme "2.16.840.1.113883.5.25", and an obligation not to reuse with code "NOREUSE", display name "prohibit use beyond purpose of use", and coding scheme "2.16.840.1.113883.1.11.20471", for the DocumentEntry labeled "ExampleDocument".

1505

```
1510 <rim:Classification
      classificationScheme=
        "urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
      classifiedObject="ExampleDocument"
      id="IdExample_046"
      objectType="urn:oasis:names:tc:ebxml-
1515 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="N"
      <rim:Name>
        <rim:LocalizedString
          value="Normal Clinical Data"/>
        </rim:Name>
      <rim:Slot name="codingScheme">
1520 <rim:ValueList>
          <rim:Value>2.16.840.1.113883.5.25</rim:Value>
        </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
1525 <rim:Classification
      classificationScheme=
        "urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
      classifiedObject="ExampleDocument"
1530 id="IdExample_046"
      objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="NOREUSE">
      <rim:Name>
        <rim:LocalizedString value="prohibit reuse beyond purpose of use"/>
1535 </rim:Name>
      <rim:Slot name="codingScheme">
      <rim:ValueList>
        <rim:Value>2.16.840.1.113883.1.11.20471</rim:Value>
        </rim:ValueList>
1540 </rim:Slot>
    </rim:Classification>
```

#### 4.2.3.2.6 DocumentEntry.creationTime

##### Description:

Represents the time the author created the document.

##### 1545 Coding:

Max length is 256 characters. The format of the creationTime value is DTM; see Table 4.2.3.1.7-2 for a description of DTM format.

The value is coded as a single value within an ebRIM Slot in the DocumentEntry.

1550 The following example shows a creationTime of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

1555

```
<rim:Slot name="creationTime">
  <rim:ValueList>
    <rim:Value>20041225212010</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

#### 4.2.3.2.7 DocumentEntry.entryUUID

##### Description:

1560

entryUUID is a globally unique identifier primarily intended for internal document management purposes. In contrast, the uniqueId attribute is used for external references (e.g., links, etc.).

##### Coding:

Max length is unbounded. The format of the entryUUID value is UUID. Certain Document Sharing transactions may allow symbolic Ids (anything that does not have the urn:uuid: prefix) to be used in certain conditions.

1565

The value of the entryUUID is coded in the id XML attribute on the ExtrinsicObject representing the DocumentEntry. In the example below, the entryUUID is urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6

1570

```
<rim:ExtrinsicObject mimeType="application/pdf"
  id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6"
  objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
> ...
```

#### 4.2.3.2.8 DocumentEntry.eventCodeList

##### Description:

1575

This list of codes represents the main clinical acts, such as a colonoscopy or an appendectomy, being documented. In some cases, the event is inherent in the typeCode, such as a "History and Physical Report" in which the procedure being documented is necessarily a "History and Physical" act. An event can further specialize the act inherent in the typeCode, such as where it is simply "Procedure Report" and the procedure was a "colonoscopy". When defining the value sets for eventCodes, they should not conflict with the values inherent in the classCode, practiceSettingCode or typeCode as such a conflict would create an ambiguous situation.

1580

##### Coding:

The eventCodeList is coded with eBRIM Classification objects. See Section 4.2.3.1.2 for a description of coding an eBRIM Classification. There may be zero or more eBRIM Classifications containing a code and additional eventCode entries are coded by specifying multiple classification objects. For the eventCodeList metadata attribute, the classificationScheme shall be urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4.

1585

1590 The following example specifies an eventCode="ExampleeventCode" with display name "eventCodeDisplayName" and coding scheme "Example Event Code Scheme" for the DocumentEntry labeled "ExampleDocument".

```
1595 <rim:Classification
      classificationScheme=
        "urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4"
      classifiedObject="ExampleDocument"
      id="IdExample_048"
      objectType="urn:oasis:names:tc:ebxml-
1600 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="ExampleeventCode"
    >
    <rim:Name>
      <rim:LocalizedString value="eventCodeDisplayName" />
    </rim:Name>
    <rim:Slot name="codingScheme">
1605 <rim:ValueList>
      <rim:Value>Example Event Code Scheme</rim:Value>
    </rim:ValueList>
    </rim:Slot>
  </rim:Classification>
```

#### 1610 4.2.3.2.9 DocumentEntry.formatCode

**Description:**

Globally unique code specifying the format of the document. Along with the typeCode, it should provide sufficient information to allow potential consumer to know if it will be able to process the document.

1615 The formatCode shall be sufficiently specific to ensure processing/display by identifying a document encoding, structure and template (e.g., for a CDA Document, the fact that it complies with a CDA schema, possibly a template and the choice of a content-specific style sheet).

Format codes may be specified by multiple organizations. Format codes defined by the ITI domain shall have names with the prefix

1620 urn:ihe:iti:

Format codes defined by other IHE domains shall have names with the prefix

urn:ihe:'domain initials':

Format codes defined by the creating entity shall have names with the prefix

urn:ad:'creating entity':

1625 The value of 'creating entity' is intended to be unique.

**Coding:**

There shall be zero or one ebRIM Classification containing a formatCode. See Section 4.2.3.1.2 for a description of coding an ebRIM Classification. For the formatCode metadata attribute, the

1630 classificationScheme shall be urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d. Any valid URN may be used as a formatCode.

The following example specifies formatCode="ExampleformatCode" with display name "ExampleformatCodeDisplayName" and coding scheme "Example Format Code Scheme" for the DocumentEntry labeled "ExampleDocument".

```
1635 <rim:Classification
      classificationScheme=
        "urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"
      classifiedObject="ExampleDocument"
      id="IdExample_049"
1640      objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="ExampleformatCode"
      >
      <rim:Name>
1645       <rim:LocalizedString value="ExampleformatCodeDisplayName" />
      </rim:Name>
      <rim:Slot name="codingScheme">
        <rim:ValueList>
1650         <rim:Value>Example Format Code Scheme</rim:Value>
        </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
```

#### 4.2.3.2.10 DocumentEntry.hash

##### Description:

1655 The hash of the contents of the document. If this attribute is present in a transaction, it may be validated; the rules for validation are specified in the relevant profile. If this attribute is missing, it may be created using the rules in the relevant profile.

##### Coding:

1660 Max length is 256 characters. The format of the hash value is SHA1 hash; see Table 4.2.3.1.7-2 for a description of SHA1.

The value is coded as a case-insensitive single value within an ebRIM Slot in the DocumentEntry.

The following example shows a hash of da39a3ee5e6b4b0d3255bfef95601890afd80709.

1665

```
<rim:Slot name="hash">
  <rim:ValueList>
    <rim:Value>da39a3ee5e6b4b0d3255bfef95601890afd80709</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

#### 1670 4.2.3.2.11 DocumentEntry.healthcareFacilityTypeCode

##### Description:

This code represents the type of organizational setting of the clinical encounter during which the documented act occurred.

1675 In some cases, the setting of the encounter is inherent in the typeCode, such as "Diabetes Clinic Progress Note". healthcareFacilityTypeCode shall be equivalent to or further specialize the value inherent in the typeCode; for example, where the typeCode is simply "Clinic Progress Note" and the value of healthcareFacilityTypeCode is "private clinic".

##### Coding:

1680 There shall be zero or one ebRIM Classification containing a healthcareFacilityTypeCode. See Section 4.2.3.1.2 for a description of coding an ebRIM Classification. For the healthcareFacilityTypeCode metadata attribute the classificationScheme shall be urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1.

1685 The following example specifies healthcareFacilityTypeCode = "ExamplehealthcareFacilityTypeCode " with display name "healthcareFacilityTypeCodeDisplayName " and coding scheme "Example Healthcare Facility Scheme" for the DocumentEntry labeled "ExampleDocument".

```
1690 <rim:Classification
  classificationScheme=
    "urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"
  classifiedObject="ExampleDocument"
  id="IdExample_050"
  objectType="urn:oasis:names:tc:ebxml-
1695 regrep:ObjectType:RegistryObject:Classification"
  nodeRepresentation="ExamplehealthcareFacilityTypeCode">
  <rim:Name>
    <rim:value="healthcareFacilityTypeCodeDisplayName" />
  </rim:Name>
  <rim:Slot name="codingScheme">
1700   <rim:ValueList>
     <rim:Value>Example Healthcare Facility Scheme</rim:Value>
   </rim:ValueList>
  </rim:Slot>
</rim:Classification>
```

#### 1705 4.2.3.2.12 DocumentEntry.homeCommunityId

##### Description:

A globally unique identifier for a community where the DocumentEntry and document can be accessed.

**Coding:**

1710 Max length is unbounded. Contained in the ebRS ExtrinsicObject home attribute for the ExtrinsicObject that corresponds to the DocumentEntry. It is an OID URN (see Table 4.2.3.1.7-2).

See IHE ITI TF-2a:3.18.4.1.2.3.8 and IHE ITI TF-2b: 3.38.4.1.2.1.

```
1715 <rim:ExtrinsicObject home="urn:oid:1.2.3" ...>
      ...
    </rim:ExtrinsicObject>
```

#### 4.2.3.2.13 DocumentEntry.languageCode

**Description:**

1720 Specifies the human language of character data in the document.

**Coding:**

Max length is 256 characters. The values of the attribute are language identifiers as described by the IETF (Internet Engineering Task Force) [RFC 5646](#).

The value is coded as a single value within an ebRIM Slot in the DocumentEntry.

1725 The following example shows a languageCode of en-CA.

```
1730 <rim:Slot name="languageCode">
      <rim:ValueList>
        <rim:Value>en-CA</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

#### 4.2.3.2.14 DocumentEntry.legalAuthenticator

**Description:**

1735 Represents a participant within the authorInstitution who has legally authenticated or attested the document. Legal authentication implies that a document has been signed manually or electronically by the legalAuthenticator.

**Coding:**

1740 Max length is 256 characters. This attribute shall be absent if not applicable. The value is coded as a single value within an ebRIM Slot in the DocumentEntry. The format of the legalAuthenticator value is XCN; see Table 4.2.3.1.7-2 for description of XCN format.

The following example shows a legalAuthenticator of ^Welby^Marcus^^ Dr^MD.

1745

```
<rim:Slot name="legalAuthenticator">
  <rim:ValueList>
    <rim:Value>^Welby^Marcus^^^Dr^MD</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

#### 4.2.3.2.15 DocumentEntry.mimeType

##### Description:

1750 MIME type of the document in the Repository

##### Coding:

Max length is unbounded. Shall have only a single value. Encoded in the ebRS ExtrinsicObject mimeType attribute for the ExtrinsicObject that corresponds to the DocumentEntry. See ebRS Schema RIM.XSD.

1755 In this example the MIME type is “application/pdf”.

```
<rim:ExtrinsicObject mimeType="application/pdf"
  id="ExampleDocument"
  objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
```

#### 1760 4.2.3.2.16 DocumentEntry.patientId

##### Description:

The patientId represents the subject of care of the document. For XDS the patient identifier domain is the XDS Affinity Domain Patient Identifier Domain (XAD-PID).

1765 Within a submission request, the value of patientId of the DocumentEntries shall match that of the SubmissionSet.

##### Coding:

The format of the patientId value is CX, see Table 4.2.3.1.7-2.

It shall contain two parts:

1770 Assigning Authority Domain Id (organization that issued the Id),

An Id from the above Assigning Authority.

No other values are allowed, as specified for the CX type. Using HL7 terminology, no other values are allowed in the components of the coded value, nor are further subcomponents allowed.

1775 Coded as an ebRIM ExternalIdentifier. See Section 4.2.3.1.3 for a description of coding an ebRIM ExternalIdentifier. This element references, and is contained in, the ExtrinsicObject representing the DocumentEntry. There shall be zero or one patientId value.

In the example below the patientId is 6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO where 6578946 is the ID and 1.3.6.1.4.1.21367.2005.3.7 is the assigning authority.

1780

```
<rim:ExternalIdentifier
  identificationScheme=
    "urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"
  value="6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO"
  id="IdExample_051"
  objectType="urn:oasis:names:tc:ebxml-
    regrep:ObjectType:RegistryObject:ExternalIdentifier"
  registryObject="DocumentEntry01">
  <rim:Name>
    <rim:LocalizedString value="XDSDocumentEntry.patientId" />
  </rim:Name>
</rim:ExternalIdentifier>
```

1785

1790

This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

1795

#### **4.2.3.2.17 DocumentEntry.practiceSettingCode**

##### **Description:**

The code specifying the clinical specialty where the act that resulted in the document was performed (e.g., Family Practice, Laboratory, Radiology). It is suggested that the creating entity draws these values from a coding scheme providing a coarse level of granularity (about 10 to 100 entries).

1800

##### **Coding:**

There shall be zero or one ebRIM Classification containing a practiceSettingCode. See Section 4.2.3.1.2 for a description of coding an ebRIM Classification. For the practiceSettingCode metadata attribute, the classificationScheme shall be urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead.

1805

The following example specifies practiceSettingCode="ExamplepracticeSettingCode" with display name "ExamplepracticeSettingCodeDisplayName" and coding scheme "Example Practice Setting Code Scheme" for the DocumentEntry labeled "ExampleDocument".

```
1810 <rim:Classification
      ClassificationScheme="urn:uuid:ccccf5598-8b07-4b77-a05e-ae952c785ead"
      classifiedObject="ExampleDocument"
      id="IdExample_052"
      objectType="urn:oasis:names:tc:ebxml-
1815 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="ExamplepracticeSettingCode">
      <rim:Name>
        <rim:LocalizedString
          value="ExamplepracticeSettingCodeDisplayName"/>
1820 </rim:Name>
      <rim:Slot name="codingScheme">
        <rim:ValueList>
          <rim:Value>Example Practice Setting Code Scheme</rim:Value>
          </rim:ValueList>
1825 </rim:Slot>
      </rim:Classification>
```

#### 4.2.3.2.18 DocumentEntry.repositoryUniqueId

##### Description:

1830 The globally unique, immutable, identifier of the repository where the document referenced by the Document Entry can be accessed. This unique identifier for the repository may be used to identify and connect to the specific repository to access the document.

##### Coding:

Max length is 64 characters. The format of the repositoryUniqueId value is OID.

The value is coded as a single value within an eRIM Slot in the DocumentEntry.

1835 The following example shows a repositoryUniqueId of 1.3.6.1.4.5.

```
1840 <rim:Slot name="repositoryUniqueId">
      <rim:ValueList>
        <rim:Value>1.3.6.1.4.5</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

#### 4.2.3.2.19 DocumentEntry.serviceStartTime

##### Description:

1845 Represents the start time of the service being documented took place (clinically significant, but not necessarily when the document was produced or approved). This may be the same as the encounter time in case the service was delivered during an encounter. Encounter time is not coded in metadata but may be coded within the document.

Note: If needed, other times associated with the document, such as time of approval, are to be recorded within the document.

1850 **Coding:**

Max length is 256 characters. The format of the serviceStartTime value is DTM; see table 4.2.3.1.7-2 for a description of DTM format.

The value is coded as a single value within an ebRIM Slot Attribute in the DocumentEntry.

1855 The following example shows a creationTime of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

```
1860 <rim:Slot name="serviceStartTime">
  <rim:ValueList>
    <rim:Value>20041225212010</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

#### 4.2.3.2.20 DocumentEntry.serviceStopTime

##### Description:

1865 Represents the stop time of the service being documented took place (clinically significant, but not necessarily when the document was produced or approved). This may be the same as the encounter time in case the service was delivered during an encounter. Encounter time is not coded in metadata but may be coded within the document.

If the service happens at a point in time, this attribute shall contain the same value as the serviceStartTime.

##### 1870 Coding:

Max length is 256 characters. The format of the serviceStartTime value is DTM; see table 4.2.3.1.7-2 for a description of DTM format.

The value is coded as a single value within an ebRIM Slot Attribute in the DocumentEntry.

1875 The following example shows a stop time of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

```
1880 <rim:Slot name="serviceStopTime">
  <rim:ValueList>
    <rim:Value>20041225232010</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

#### 4.2.3.2.21 DocumentEntry.size

##### Description:

Size in bytes of the byte stream that comprises the document.

##### Coding:

1885 Max length of the encoded size is 256 characters. Coded as an ebRIM Slot. Shall have only a single value.

The following example shows a size value of 3654 bytes.

```
1890 <rim:Slot name="size">
      <rim:ValueList>
        <rim:Value>3654</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

#### 4.2.3.2.22 DocumentEntry.sourcePatientId

##### 1895 Description:

The sourcePatientId represents the subject of care's medical record Identifier (e.g., Patient Id) in the local patient identifier domain of the creating entity at the time the document entry was submitted.

##### Coding:

1900 Max length is 256 characters. Coded as an ebRIM Slot with the value encoded according the HL7 v2.5 CX datatype (See table 4.2.3.1.7-2). Shall contain zero or one value.

It shall contain only two parts:

An Id (e.g., Patient Id CX.1).

The Authority Domain Id that assigned the Id (CX.4).

1905 All other CX components shall be omitted.

```
1910 <rim:Slot name="sourcePatientId">
      <rim:ValueList>
        <rim:Value>j98789^^^&1.2.3.4.343.1&ISO</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

#### 4.2.3.2.23 DocumentEntry.sourcePatientInfo

##### Description:

1915 This attribute contains demographics information at the time of submission of the patient to whose medical record this document belongs.

This information typically includes: the patient first and last name, sex, and birth date. Policies at the creating entity may require more or less specific information and format.

1920 This patient information is not intended to be updated once the document is registered (just as the document content and metadata itself will not be updated without replacing the previous document). As sourcePatientInfo may have been updated by the source actor, it may no longer be in use within the Document Source (EHR-CR). It is only intended as an audit/checking mechanism and has occasional use for Document Consumer Actors.

##### Coding:

1925 Max length is 256 characters. Coded as an ebRIM Slot. Shall contain zero or one value list of demographic elements, where each element in the list is identified by fields from the HL7 PID segment (See HL7 v2.5, chapter 3, section 3.4.2).

```
1930 <rim:Slot name="sourcePatientInfo">
  <rim:ValueList>
    <rim:Value>PID-3|DTP-1^^^&1.3.6.1.4.1.21367.2005.3.7&ISO</rim:Value>
    <rim:Value>PID-5|DICTAPHONE^ONE^^^</rim:Value>
    <rim:Value>PID-7|19650120</rim:Value>
    <rim:Value>PID-8|M</rim:Value>
    <rim:Value>PID-11|100 Main St^^BURLINGTON^MA^01803^USA</rim:Value>
1935 </rim:ValueList>
  </rim:Slot>
```

sourcePatientInfo should include values for:

- 1940 • PID-3 (source patient identifier list. A list uses the repetition separator ~, for example: DTP 1^^^&1.3.6&ISO~XTP1^^^&1.3.11&ISO.)
- PID-5 (source patient name)
- PID-7 (source patient date of birth)
- PID-8 (source patient gender)

M – Male F – Female

1945 O – Other U – Unknown

sourcePatientInfo should not include values for PID-2 (patient id), PID-4 (alternate patient id), PID-12 (country code), or PID-19 (social security number).

#### 4.2.3.2.24 DocumentEntry.title

##### Description:

1950 Represents the title of the document.

Clinical documents often do not have a title, in such case the classCode (e.g., a "consultation" or "progress note") is often used as the title. In that case, the title is usually omitted.

##### Coding:

The format of DocumentEntry.title shall be any string of length less than 128 characters.

1955 DocumentEntry.title is represented in ebXML as the "value" attribute of the LocalizedString element within the ebRIM Name structure. There can be only one ebRIM Name structure per DocumentEntry.

The following example shows a title for the DocumentEntry.

```
1960 <rim:ExtrinsicObject
      id="ExampleDocument"
      objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
      mimeType="application/pdf">
1965   <rim:Name>
      <rim:LocalizedString value="Example Document Title"/>
      </rim:Name>
      ...
    </rim:ExtrinsicObject>
```

#### 4.2.3.2.25 DocumentEntry.typeCode

##### 1970 **Description:**

The code specifying the precise type of document from the user perspective. It is recommended that the creating entity draw these values from a coding scheme providing a fine level of granularity such as LOINC.

##### **Coding:**

1975 Coded as an ebRIM classification. See Section 4.2.3.1.2 for a description on creating classifications. Shall have zero or one value.

The following example specifies typeCode="ExampletypeCode" with display name "typeCodeDisplayName " and coding scheme " Example Type Code Scheme " for the DocumentEntry labeled "ExampleDocument".

```
1980 <rim:Classification
      classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"
      classifiedObject="ExampleDocument"
      nodeRepresentation="ExampletypeCode"
1985   id="IdExample_053"
      objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification">
      <rim:Name>
1990   <rim:LocalizedString value="typeCodeDisplayName" />
      </rim:Name>
      <rim:Slot name="codingScheme">
1995   <rim:ValueList>
      <rim:Value>Example Type Code Scheme</rim:Value>
      </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
```

#### 4.2.3.2.26 DocumentEntry.uniqueId

##### **Description:**

Globally unique identifier assigned to the document by the creating entity.

2000 A DocumentEntry representing a single document is identified by the uniqueId attribute; the linkage between DocumentEntry and the document it represents is made with the uniqueId attribute.

This unique identifier may be used in other documents to reference this document.

**Coding:**

2005 The format of the DocumentEntry.uniqueId value is OID see Table 4.2.3.1.7-2.

Coded as an ebRIM ExternalIdentifier (see Section 4.2.3.1.3 for a description of coding an ebRIM ExternalIdentifier) which references, and is contained in, the ExtrinsicObject representing the DocumentEntry. There shall be only a single uniqueId value.

2010 The structure and format of this Id shall be consistent with the specification corresponding to the formatCode attribute. (e.g., for a DICOM standard document a 64 character numeric UID, for an HL7 CDA format a serialization of the CDA Document Id extension and root in the form oid^extension, where OID is a 64 digits max, and the Id is a 16 UTF-8 char max. If the OID is coded without the extension then the '^' character shall not be included.)

```
2015 <rim:ExternalIdentifier
      identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"
      value="1.2.3.4.5.6.78901.2345.6.7^123456"
      id="IdExample_054"
      objectType="urn:oasis:names:tc:ebxml-
2020 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="DocumentEntry01">
  <rim:Name>
    <rim:LocalizedString value="XDSDocumentEntry.uniqueId"/>
  </rim:Name>
2025 </rim:ExternalIdentifier>
```

#### 4.2.3.2.27 DocumentEntry.URI

**Description:**

The URI attribute contains the URI for the document.

**Coding:**

2030 Coded as an ebRIM Slot. Shall have only a single value. Max length is 256 characters.

```
2035 <rim:Slot name="URI">
  <rim:ValueList>
    <rim:Value>DOC001.XML</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

#### 4.2.3.2.28 DocumentEntry.referenceIdList

**Description:**

2040 This list contains zero or more Identifiers. These Identifiers may be internal or external identifiers, e.g., Identifiers may be Accession Numbers, Order Numbers, Referral Request Identifiers, XDS Workflow Instance Identifiers, etc. The referenceIdList contains Identifiers CXi encoded, as specified in Table 4.2.3.1.7-2.

2045 XDS Document Registry Actors supporting the ReferenceId Option shall preserve this value. When the ReferenceId Option is not declared, the Document Registry shall handle this extra metadata according to the historic rules outlined in Section 4.2.3.1.6 “Extra Metadata Elements”.

**Coding:**

Coded as an ebRIM Slot. May have multiple values. Max length for each value is 256 characters. The name of the slot in the metadata shall be “urn:ihe:iti:xds:2013:referenceIdList”.

For example encoding 2 values in the referenceIdList

```
2050 <rim:Slot name="urn:ihe:iti:xds:2013:referenceIdList ">
    <rim:ValueList>
      <rim:Value>
2055 2013001^^^&1.2.3.4.5.6&ISO^urn:ihe:iti:xds:2013:accession
      </rim:Value>
      <rim:Value>
2055 1.2.3.12.78.23^^^urn:ihe:iti:xds:2013:uniqueId^&1.2.3.4&ISO
      </rim:Value>
    </rim:ValueList>
  </rim:Slot>
```

2060 **4.2.3.2.29 DocumentEntry.limitedMetadata**

**Description:**

Indicates whether the Document Entry was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source.

**Coding:**

2065 The Document Entry is flagged using an ebRIM Classification with a classificationScheme of urn:uuid:ab9b591b-83ab-4d03-8f5d-f93b1fb92e85.

The following example marks the “DocEntry” Document Entry as created via the less rigorous metadata requirements.

```
2070 <ExtrinsicObject id="DocEntry">
  (...)
  <Classification classifiedObject="DocEntry"
    classificationNode="urn:uuid:ab9b591b-83ab-4d03-8f5d-f93b1fb92e85" />
  (...)
2075 </ExtrinsicObject>
```

**4.2.3.2.30 DocumentEntry.objectType**

**Description:**

2080 The objectType attribute reflects the type of DocumentEntry. As described in Section 4.1.1, there are two DocumentEntry types: Stable Document Entry and On-Demand Document Entry. A Stable Document Entry contains metadata about an already created document available for retrieval is an Document Entry and is designated by setting objectType equal to the UUID for Stable (see Section 4.2.5.2 for the UUID). An On-Demand DocumentEntry contains metadata which can be used to create an on-demand document which collects the latest, most recent available information at the time of retrieval. It is designed by setting an objectType equal to the  
 2085 UUID for on-demand (see Section 4.2.5.2 for the UUID).

**Coding:**

Max length is unbounded. The format of the objectType value is UUID.

2090 The value of the objectType is coded in the objectType XML attribute on the ExtrinsicObject representing the DocumentEntry. In the example below, the objectType is urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1 and reflects a stable DocumentEntry.

```

    <rim:ExtrinsicObject mimeType="application/pdf"
      id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6"
      objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
    > ...
  
```

**4.2.3.3 SubmissionSet Attributes**

The following metadata elements shall be used to describe a SubmissionSet.

2100 Each attribute shown is an attribute on the RegistryPackage object defining the SubmissionSet. The attribute name is defined with a prefix of the object type of SubmissionSet when referenced by other objects, for example SubmissionSet.sourceId.

Please see section 8 for the General format of DocumentEntry, Folder and SubmissionSet attribute tables.

2105 **Table 4.2.3.3-1: Submission Set Metadata Attribute Definitions** (previously Table 4.1-6)

SubmissionSet Metadata Attribute	Description	Form	Coding	Detail
author	The humans and/or machines that authored the document in the SubmissionSet. This attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty.	4.2.3.3.1	ebRIM Classification	4.2.3.3.1
availabilityStatus	The lifecycle status of the SubmissionSet	Predefined URN	XML attribute	4.2.3.3.2
comments	Comments associated with the SubmissionSet.	String	ebRIM Description	4.2.3.3.3
contentTypeCode	The code specifying the type of clinical activity that resulted in placing the documents in this	Code	ebRIM	4.2.3.3.4

SubmissionSet Metadata Attribute	Description	Form	Coding	Detail
	SubmissionSet.		Classification	
entryUUID	A globally unique identifier used to identify the SubmissionSet.	UUID	XML attribute	4.2.3.3.5
homeCommunityId	A globally unique identifier for a community.	OID URN	home XML attribute	4.2.3.3.6
intendedRecipient	The organization(s) or person(s) for whom the SubmissionSet is intended.	See 4.2.3.3.7	ebRIM Slot	4.2.3.3.7
limitedMetadata	A flag that the associated SubmissionSet was created using the less rigorous metadata requirements as defined for the Metadata-limited Document Source.	See 4.2.3.3.13	ebRIM Classification	4.2.3.3.13
patientId	The patientId represents the subject of care whose longitudinal record is being reflected in this SubmissionSet.	CX	ebRIM ExternalIdentifier	4.2.3.3.8
sourceId	OID identifying the instance of the creating entity that contributed the SubmissionSet.	OID	ebRIM ExternalIdentifier	4.2.3.3.8
submissionTime	Point in time at the creating entity when the SubmissionSet was created.	DTM	ebRIM Slot	4.2.3.3.10
title	The title of the SubmissionSet.	UTF-8	ebRIM Name	4.2.3.3.11
uniqueId	A globally unique identifier for the SubmissionSet assigned by the creating entity.	OID	ebRIM ExternalIdentifier	4.2.3.3.12

#### 4.2.3.3.1 SubmissionSet.author

##### Description:

Represents the humans and/or machines that authored the submission set. See Section 4.2.3.1.4 for details on creating the structure.

2110 The classificationScheme shall be `urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d`

#### 4.2.3.3.2 SubmissionSet.availabilityStatus

##### Description:

Represents the status of the SubmissionSet. Since the deprecation of SubmissionSets is not allowed, this value shall always be Approved.

2115 **Coding:**

The availabilityStatus value shall be "urn:oasis:names:tc:ebxml-regrep:StatusType:Approved".

2120

```
<rim:RegistryPackage
  id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855"
  status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
> ...
```

#### 4.2.3.3.3 SubmissionSet.comments

##### Description:

Contains comments associated with the SubmissionSet.

2125

##### Coding:

Max length is unbounded. The value of the comments attribute is coded in XML as the "value" attribute of the LocalizedString element within the ebRIM Description structure. There can be at most one ebRIM Description structure per SubmissionSet.

The following example shows a comment for the SubmissionSet.

2130

```
<rim:Description>
  <rim:LocalizedString value = "comment associated with SubmissionSet"/>
</rim:Description>
```

2135

#### 4.2.3.3.4 SubmissionSet.contentTypeCode

##### Description:

The code specifying the type of clinical activity that resulted in placing these DocumentEntries, Folders, and/or Associations in this SubmissionSet. These values are to be drawn from a vocabulary defined by the creating entity that contributed the SubmissionSet.

2140

##### Coding:

Coded as an ebRIM Classification. See Section 4.2.3.1.2 for a description of coding an ebRIM Classification. Shall have zero or one value

2145

The following example specifies contentTypeCode="ExamplecontentTypeCode" with display name "Example contentTypeCodeDisplayName" and coding scheme "Example contentTypeCode Scheme" for the SubmissionSet labeled "ExampleSubmissionSet".

```
2150 <rim:Classification
      classificationScheme="urn:uuid:aa543740-bdda-424e-8c96-df4873be8500"
      classifiedObject="ExampleSubmissionSet"
      id="IdExample_056"
      objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="ExamplecontentTypeCode"
      >
2155 <rim:Name>
      <rim:LocalizedString value="Example contentTypeCodeDisplayName" />
      </rim:Name>
      <rim:Slot name="codingScheme">
2160 <rim:ValueList>
      <rim:Value>Example Specific Value</rim:Value>
      </rim:ValueList>
      </rim:Slot>
      </rim:Classification>
```

#### 4.2.3.3.5 SubmissionSet.entryUUID

2165 entryUUID is a globally unique identifier primarily intended for internal document management purposes. In contrast, the uniqueId attribute is used for external references (e.g., links, etc.).

##### Coding:

The format of the entryUUID value is UUID. Certain Document Sharing transactions may allow symbolic Ids (any string that does not have the urn:uuid: prefix) to be used.

2170 The value of the entryUUID is coded in the id XML attribute on the RegistryPackage representing the Submission Set. In the example below, the entryUUID is urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6

```
2175 <rim:RegistryPackage
      id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6"
      objectType=
      "urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:RegistryPackage"
      > ...
```

#### 2180 4.2.3.3.6 SubmissionSet.homeCommunityId

##### Description:

A globally unique identifier for a community.

##### Coding:

2185 The ebRIM RegistryPackage home attribute for the RegistryPackage that corresponds to the SubmissionSet. It is an OID URN. See table 4.2.3.1.7-2.

See TF-2a:3.18.4.1.2.3.8 and ITI TF-2b: 3.38.4.1.2.1.

2190 

```
<rim:RegistryPackage home="urn:oid:1.2.3" ...>
    ...
</rim:RegistryPackage>
```

#### 4.2.3.3.7 SubmissionSet.intendedRecipient

##### Description:

2195 Represents the organization(s) or person(s) for whom the SubmissionSet is intended at time of submission. Each slot value shall include at least one of the organization, person, or telecommunications address fields described below. It is highly recommended to define the organization for all the persons, avoiding errors in the transmission of the documents.

##### Coding:

2200 Max length is 256. The format of the SubmissionSet.intendedRecipient value is XON|XCN|XTN where XON identifies the organization, XCN identifies a person and XTN identifies the telecommunications, see table 4.2.3.1.7-2 for description of XON, XCN and XTN format. There is a "|" character separator between the organization and the person, and between the person and the telecommunications address, which is required when the person or the telecommunications address information is present.

The value is coded as zero or more values within a single ebRIM Slot in the SubmissionSet.

2205 The following example shows two doctors from the same organization, another doctor without the organization details and another organization without the person details, and just a telecommunications address.

2210 

```
<rim:Slot name="intendedRecipient">
  <rim:ValueList>
    <rim:Value>
      Some
      Hospital^^^^^^^^1.2.3.9.1789.45|^Wel^Marcus^^^Dr^MD|^Internet^mwel@healthca
      re.example.org
    </rim:Value>
    <rim:Value>
      Some Hospital^^^^^^^^1.2.3.9.1789.45|^Peirre^LaPointe^^^Dr^MD
    </rim:Value>
    <rim:Value>|12345^LaShawn^James^^Dr^MD</rim:Value>
    <rim:Value>MainHospital^^^^^^^^1.2.3.4.5.6.7.8.9.1789.2364</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

#### 4.2.3.3.8 SubmissionSet.patientId

##### Description:

2225 The patientId represents the primary subject of care of the SubmissionSet.

##### Coding:

The format of the patientId value is CX (see table 4.2.3.1.7-2).

It shall contain two parts:

2230 Assigning Authority Domain Id (organization that issued the Id).

An Id from the above Assigning Authority.

No other values are allowed, as specified for the CX type. Using HL7 terminology, no other values are allowed in the components of the coded value, nor are further subcomponents allowed.

2235 The value is coded within a single ebRIM ExternalIdentifier element (see Section 4.2.3.1.3 for a description of coding an ebRIM ExternalIdentifier) which references, and is contained in, the RegistryPackage representing the SubmissionSet. There shall be zero or one single patientId value.

2240 In the example below the patientId is 6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO where 6578946 is the ID and 1.3.6.1.4.1.21367.2005.3.7 is the assigning authority.

```
2245 <rim:ExternalIdentifier
      identificationScheme="urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446"
      value="6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO"
      id="IdExample_057"
      objectType="urn:oasis:names:tc:ebxml-
2250 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="SubmissionSet01">
      >
      <rim:Name>
        <rim:LocalizedString value = "XDSSubmissionSet.patientId"/>
      </rim:Name>
    </rim:ExternalIdentifier>
```

2255 This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

#### 4.2.3.3.9 SubmissionSet.sourceId

##### Description:

2260 The globally unique, immutable, identifier of the entity that contributed the SubmissionSet. When a "broker" is involved in sending SubmissionSets from a collection of client systems, it shall use a different sourceId for submissions from each separate system to allow for tracking.

##### Coding:

The format of the sourceId value is OID (see table 4.2.3.1.7-2).

2265 The value is coded within a single ebRIM ExternalIdentifier element (see Section 4.2.3.1.3 for a description of coding an ebRIM ExternalIdentifier) which references, and is contained in, the RegistryPackage representing the SubmissionSet. There shall be only a single sourceId value.

In the example below the sourceId is "1.3.6.1.4.1.21367.2005.3.7".

```
2270 <rim:ExternalIdentifier
      identificationScheme="urn:uuid:554ac39e-e3fe-47fe-b233-965d2a147832"
      value="1.3.6.1.4.1.21367.2005.3.7"
      id="IdExample_058"
      objectType="urn:oasis:names:tc:ebxml-
2275 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="SubmissionSet01">
      <rim:Name>
        <rim:LocalizedString value = "XDSSubmissionSet.sourceId"/>
      </rim:Name>
    </rim:ExternalIdentifier>
```

#### 4.2.3.3.10 SubmissionSet.submissionTime

##### 2280 **Description:**

Represents the point in time at the creating entity when the Submission Set was submitted. This shall be provided by the submitting system.

##### **Coding:**

2285 Max length is 256. The format of the submissionTime value is DTM. See table 4.2.3.1.7-2 for a description of DTM format.

The value is coded as a single value within an eBRIM Slot Attribute in the SubmissionSet.

The following example shows a submissionTime of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

```
2290 <rim:Slot name="submissionTime">
      <rim:ValueList>
        <rim:Value>20041225212010</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

2295

#### 4.2.3.3.11 SubmissionSet.title

##### **Description:**

Shall contain the title of the Submission Set.

##### **Coding:**

2300 The format of SubmissionSet.title shall be any string of length less than 256 characters.

SubmissionSet.title is represented in ebXML as the "value" attribute of the LocalizedString element within the ebRIM Name structure. There can be only one ebRIM Name structure per SubmissionSet.

The following example shows a title for the SubmissionSet.

2305

```
<rim:Name>
  <LocalizedString value="Example Submission Set Title"/>
</rim:Name>
```

#### 4.2.3.3.12 SubmissionSet.uniqueId

2310 **Description:**

The globally unique identifier for the SubmissionSet assigned by the entity that contributed the SubmissionSet.

**Coding:**

The format of the SubmissionSet.uniqueId value is OID (see Table 4.2.3.1.7-2).

2315 The value is coded within a single ebRIM ExternalIdentifier element which references, and is contained in, the RegistryPackage representing the SubmissionSet. See Section 4.2.3.1.3 for a description of coding an ebRIM ExternalIdentifier. There shall be only a single uniqueId value.

The following example shows a uniqueId of 1.2.3.4.5.

2320

```
<rim:ExternalIdentifier
  identificationScheme="urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8"
  id="IdExample_059"
  objectType="urn:oasis:names:tc:ebxml-
  regrep:ObjectType:RegistryObject:ExternalIdentifier"
  value="1.2.3.4.5"
  registryObject="SubmissionSet01">
  <rim:Name>
    <rim:LocalizedString value = "XDSSubmissionSet.uniqueId"/>
  </rim:Name>
</rim:ExternalIdentifier>
```

2325

2330

#### 4.2.3.3.13 SubmissionSet.limitedMetadata

**Description:**

2335 Indicates whether the SubmissionSet was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source Actor.

**Coding:**

The SubmissionSet is flagged using an ebRIM Classification with a classificationScheme of urn:uuid:5003a9db-8d8d-49e6-bf0c-990e34ac7707. Zero or one may be present.

2340 The following example marks the "SubmissionSet" SubmissionSet as created via the less rigorous metadata requirements.

```

2345 <RegistryPackage id="SubmissionSet">
      (...)
      <Classification classifiedObject="SubmissionSet"
        classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd"/>
      <Classification classifiedObject="SubmissionSet"
        classificationNode="urn:uuid:5003a9db-8d8d-49e6-bf0c-990e34ac7707"/>
      (...)
2350 </RegistryPackage>
  
```

#### 4.2.3.4 Folder Attributes

The following metadata elements shall be used to describe a Folder. Each attribute shown below is an attribute on the RegistryPackage object defining the Folder. The attribute name is defined with a prefix of the object type of Folder when referenced by other objects, for example Folder.patientId. See Section 4.2.3.1.8 for the general format of DocumentEntry, Folder and SubmissionSet attribute tables.

**Table 4.2.3.4-1: Folder Metadata Attribute Definitions** (previously Table 4.1-7)

Folder Metadata Attribute	Description	Form	Coding	Detail (See Section)
availabilityStatus	The lifecycle status of the Folder	Predefined URN	XML attribute	4.2.3.4.1
codeList	Shall contain the set of codes specifying the type of clinical activities that resulted in placing DocumentEntries in this Folder.	Code	ebRIM Classification	4.2.3.4.2
comments	Comments associated with the Folder.	String	ebRIM Description	4.2.3.4.3
entryUUID	This globally unique identifier used to identify the Folder.	UUID	XML attribute	4.2.3.4.4
homeCommunityId	A globally unique identifier for a community.	OID URN	XML attribute	4.2.3.4.5
lastUpdateTime	Most recent point in time that the Folder has been modified.	DTM	ebRIM Slot	4.2.3.4.6
limitedMetadata	A flag that the associated Folder was created using the less rigorous metadata requirements as defined for the Metadata-Limited	4.2.3.4.10	ebRIM Classification	4.2.3.4.10
patientId	The patientId represents the subject of care of the DocumentEntries within the Folder.	CX	ebRIM ExternalIdentifier	4.2.3.1.3
title	The title of the Folder.	String	ebRIM Name	4.2.3.4.8
uniqueId	Globally unique identifier for the Folder.	OID	ebRIM ExternalIdentifier	4.2.3.4.9

#### 4.2.3.4.1 Folder.availabilityStatus

2360 **Description:**

Represents the status of the Folder. Since the deprecation of Folders is not allowed, this value shall always be Approved.

**Coding:**

2365 Max length is unbounded. The availabilityStatus value shall be "urn:oasis:names:tc:ebxml-regrep:StatusType:Approved".

The example below shows the status attribute; however, this attribute is only returned on query, not set during any transaction.

```
2370 <rim:RegistryPackage
      id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855"
      status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
    > ...
```

#### 4.2.3.4.2 Folder.codeList

**Description:**

2375 Shall contain the set of codes specifying the type of clinical activity that resulted in placing Documents in this Folder. These values shall be drawn from a vocabulary or coding scheme defined by the creating entity.

**Coding:**

2380 Coded as an ebRIM classification. See Section 4.2.3.1.2 for a description of coding an ebRIM Classification. Shall have zero or more values. Code multiple values by creating multiple classification objects.

The following example specifies contentTypeCode="ExamplecontentTypeCode" with display name "ExamplecodeList CodeDisplayName" and coding scheme "Example coding scheme" for the Folder labeled "ExampleFolder".

2385

```
2390 <rim:Classification
      classificationScheme= "urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5"
      classifiedObject="ExampleFolder"
      id="IdExample_062"
      objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="Examplecode"
      >
2395 <rim:Name>
      <rim:LocalizedString
        value="ExamplecodeList CodeDisplayName" />
      </rim:Name>
      <rim:Slot name="codingScheme">
2400 <rim:ValueList>
      <rim:Value>Example coding scheme</rim:Value>
      </rim:ValueList>
      </rim:Slot>
</rim:Classification>
```

#### 4.2.3.4.3 Folder.comments

##### 2405 **Description:**

Comments associated with the Folder. Free form text or IHE profile-specified usage.

Note: Prior to the availability of the Folder.title attribute, the comments attribute might have been used to hold the title of the folder (folder name). With the addition of the title attribute, the comments attribute shall not hold the folder name.

##### 2410 **Coding:**

The format of Folder.comments is any string. Max length is unbounded.

The value of the comments attribute is coded in XML as the "value" attribute of the LocalizedString element within the ebRIM Description structure. There can be at most one ebRIM Description structure per DocumentEntry.

2415 The following example shows a comment for the Folder.

```
<rim:Description>
  <rim:LocalizedString value = "comments"/>
</rim:Description>
```

#### 2420 4.2.3.4.4 Folder.entryUUID

##### **Description:**

This globally unique identifier is primarily intended for use as a document registry management identifier. It is not meant to be an external reference (e.g., outside of the Document Registry) for documents (e.g., in links within other documents).

##### 2425 **Coding:**

Max length is unbounded. The format of the entryUUID value is UUID. Certain Document Sharing transactions may allow symbolic Ids (anything that does not have the urn:uuid: prefix) to be used.

2430 The value of the entryUUID is coded in the id XML attribute on the ExtrinsicObject representing the DocumentEntry. In the example below, the entryUUID is

urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a9ab6

```
2435 <rim:RegistryPackage
      id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a9ab6"
    > ...
```

#### 4.2.3.4.5 Folder.homeCommunityId

##### Description:

A globally unique identifier for a community.

##### Coding:

2440 Max length is unbounded. The ebRS RegistryPackage home attribute for the RegistryPackage that corresponds to the Folder. It is an OID URN (see Table 4.2.3.1.7-2).

See IHE ITI TF-2a:3.18.4.1.2.3.8 and ITI TF-2b: 3.38.4.1.2.1.

```
2445 <rim:RegistryPackage home="urn:oid:1.2.3" ...>
      ...
    </rim:RegistryPackage>
```

#### 4.2.3.4.6 Folder.lastUpdateTime

##### Description:

Most recent point in time when a DocumentEntry was placed in the Folder.

2450 **Coding:**

Max length is 256 characters. The format of the lastUpdateTime value is DTM. See table 4.2.3.1.7-2 for a description of DTM format.

The value is coded as a single value within an ebRIM Slot Attribute in the DocumentEntry.

2455 The following example shows a last update of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

```
2460 <rim:Slot name="lastUpdateTime">
      <rim:ValueList>
        <rim:Value>20041225212010</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

#### 4.2.3.4.7 Folder.patientId

##### Description:

The patientId represents the primary subject of care of the Folder.

2465 The value of the patientId shall be the same for all new documents of a SubmissionSet. All DocumentEntries placed in a Folder shall have the same patientId as the Folder.

##### Coding:

Max length unbounded. The format of the patientId value is CX (See Table 4.2.3.1.7-2).

It shall contain two parts:

2470 Assigning Authority Domain Id (organization that issued the Id).

An Id from the Assigning Authority.

No other values are allowed, as specified for the CX type in Table 4.2.3.1.7-2 above. Using HL7 terminology, no other values are allowed in the components of the coded value, nor are further subcomponents allowed.

```
2475 <rim:ExternalIdentifier
      identificationScheme="urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a"
      value="6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO"
      id="IdExample_057"
      objectType="urn:oasis:names:tc:ebxml-
2480 regrep:ObjectType:RegistryObject:ExternalIdentifier"
    >
      <rim:Name>
        <rim:LocalizedString value = "XDSFolder.patientId"/>
2485 </rim:Name>
    </rim:ExternalIdentifier>
```

This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

#### 2490 4.2.3.4.8 Folder.title

##### Description:

Shall contain the Folder name.

Note: Prior to the availability of this attribute the comments attribute might have been used to hold the title of the folder (folder name). With the addition of this attribute the comments attribute shall not hold the folder name.

2495 **Coding:**

Encoded in the name element within the ebRS ExtrinsicObject for the document (see ebRS Schema RIM.XSD). Shall have only a single value.

Max length is 256 characters.

In this example the title is “title”.

2500

```
<Name>
  <LocalizedString value="this is the title of the Folder"/>
</Name>
```

#### 4.2.3.4.9 Folder.uniqueId

2505 **Description:**

Globally unique identifier for the folder instance assigned by the creating entity.

**Coding:**

Max length is 256 characters. Encoded as the ExternalIdentifier. See Section 4.2.3.1.3 for a description of coding an ebRIM ExternalIdentifier. Shall have only a single value. Shall be of OID format.

2510

```
<rim:ExternalIdentifier
  identificationScheme="urn:uuid:75df8f67-9973-4fbe-a900-df66cefec5a"
  id="IdExample_059"
  objectType="urn:oasis:names:tc:ebxml-
  regrep:ObjectType:RegistryObject:ExternalIdentifier"
  value="1.3.6.1.4.1.21367.2005.3.7.3670984664"
  registryObject="Folder01">
  <rim:Name>
    <rim:LocalizedString value = "XDSFolder.uniqueId"/>
  </rim:Name>
</rim:ExternalIdentifier>
```

2515

2520

2525 This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

#### 4.2.3.4.10 Folder.limitedMetadata

**Description:**

Indicates whether the Folder was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source Actor.

2530 **Coding:**

The Folder is flagged using an ebRIM Classification with a classificationScheme of urn:uuid:2c144a76-29a9-4b7c-af54-b25409fe7d03. Zero or one may be present.

The following example marks the “Folder” Folder as created via the less rigorous metadata requirements.

2535

```

<RegistryPackage id="Folder">
  (...)

  <Classification classifiedObject="Folder"
  classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2"
  id="ID_061" objectType="urn:oasis:names:tc:ebxml-
  regrep:ObjectType:RegistryObject:Classification" />

  <Classification classifiedObject="Folder"
  classificationNode="urn:uuid:2c144a76-29a9-4b7c-af54-b25409fe7d03" />
  (...)
</RegistryPackage>

```

2540

2545

#### 4.2.4 Error Reporting

2550

Registry Services schema (ebRS 3.0) defines the RegistryError element for reporting details of errors or warnings.

##### 4.2.4.1 RegistryErrors Element

The RegistryError element contains the attributes in Table 4.2.4.1-1.

2555

All IHE transactions that carry the RegistryError element shall return these attributes with each error reported.

The body of all RegistryError elements shall be empty.

**Table 4.2.4.1-1: RegistryError Element Attributes**

Attribute	Optionality	Value
errorCode	R	Shall be taken from Table 4.2.4.1-2 when one of those codes is appropriate. All extensions to the list of error codes shall be unique.
codeContext	R	Supplies additional detail for the errorCode
severity	R	Indicates the severity of the error. Shall be one of: oasis:names:tc:ebxml- regrep:ErrorSeverityType:Error urn:oasis:names:tc:ebxml- regrep:ErrorSeverityType:Warning
location	O	Supplies the location of the error: module name and line number or stack trace if appropriate.

2560 An example of an error response reporting two errors using ebRS and ebRIM 3.0 is:

```

2565 <RegistryResponse
      xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0"
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure">
2575   <RegistryErrorList
        highestSeverity="urn:oasis:names:tc:ebxml-
          regrep:ErrorSeverityType:Error">
2570     <RegistryError
      errorCode="XDSPatientIdDoesNotMatch"
      codeContext="Patient Id in Document (Document1) does not match
        SubmissionSet"
      location=""
      severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
2575     <RegistryError
      errorCode="XDSRegistryMetadataError"
      codeContext="RegistryPackage (SubmissionSet) is not labeled as
        SubmissionSet or Folder"
      location=""
      severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
2580   </RegistryErrorList>
</RegistryResponse>

```

Table 4.2.4.2-1 through Table 4.2.4.2-4 control the reporting of errors for transactions that use the Document Sharing metadata attributes.

2585 **Table 4.2.4.1-2: Error Codes** (previously Table 4.1-11)

Error Code <sup>1</sup>	Discussion	Transaction (See Note 1)
DocumentQueued	An XDR Recipient queued the document for future manual matching to a patient	P
InvalidDocumentContent	<p>The recipient has rejected this submission because it detected that one of the documents does not match the metadata (e.g., formatCode) or has failed other requirements for the document content.</p> <p>When the RegistryError element contains this error code, the @location shall contain the DocumentUniqueID of the document in error.</p> <p>If multiple documents are in error, there shall be a separate RegistryError element for each document in error.</p>	P
PartialAppendContentNotProcessed	An XDR Document Recipient did not process some part of the content.	P

<sup>1</sup> The XDS Metadata Update Supplement adds error codes to this table.

Error Code <sup>1</sup>	Discussion	Transaction (See Note 1)
	Specifically the parts not processed are Append semantics	
PartialFolderContentNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Folder semantics	P
PartialReplaceContentNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Replacement semantics	P
PartialTransformNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Transform semantics	P
PartialTransformReplaceNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Transform and Replace semantics	P
XDSDocumentUniqueIdError	The document associated with the uniqueId is not available. This could be because the document is not available, the requestor is not authorized to access that document or the document is no longer available.	RS, XGR
XDSDuplicateUniqueIdInRegistry	UniqueId received was not unique. UniqueId could have been attached to SubmissionSet or Folder. codeContext shall indicate which and the value of the non-unique uniqueId. This error cannot be thrown for DocumentEntry. See XDSNonIdenticalHash.	P,R P
XDSExtraMetadataNotSaved	This warning is returned if extra metadata was present but not saved.	P,R
XDSMissingDocument	DocumentEntry exists in metadata with no corresponding attached document	P
XDSMissingDocumentMetadata	MIME package contains MIME part with Content-Id header not found in metadata	P
XDSMissingHomeCommunityId	A value for the homeCommunityId is required and has not been specified	SQ, XGQ, RS, XGR
XDSNonIdenticalHash	Document being registered was a duplicate (uniqueId already in registry) but hash does not match. codeContext indicates UniqueId.	R
XDSNonIdenticalSize	Document being registered was a duplicate (uniqueId already in registry) but size does not match. CodeContext indicates UniqueId.	R
XDSPatientIdDoesNotMatch	This error is thrown when the patient Id is required to match and does not. The codeContext shall indicate the value of the	P,R

IHE IT Infrastructure Technical Framework, Volume 3 (ITI TF-3): Cross-Transaction and Content Specifications

Error Code <sup>1</sup>	Discussion	Transaction (See Note 1)
	Patient Id and the nature of the conflict.	
XDSRegistryBusy XDSRepositoryBusy	Too much activity	P,R,SQ, XGQ P, RS, XGR
XDSRegistryDeprecatedDocumentError	The transaction was rejected because it submitted an Association referencing a deprecated document.	P,R
XDSRegistryDuplicateUniqueIdInMessage XDSRepositoryDuplicateUniqueIdInMessage	A UniqueId value was found to be used more than once within the submission. Error code indicates where error was detected. codeContext shall indicate the duplicate UniqueId.	P,R
XDSRegistryError XDSRepositoryError	Internal Error The error codes XDSRegistryError or XDSRepositoryError shall be returned if and only if a more detailed code is not available from this table for the condition being reported. If one of these error codes is returned, the attribute codeContext shall contain details of the error condition that may be implementation-specific.	P,R, SQ, XGQ P, RS, XGR
XDSRegistryMetadataError XDSRepositoryMetadataError	Error detected in metadata. Actor name indicates where error was detected. (Document Recipient uses Repository error). codeContext indicates nature of problem.	P,R
XDSRegistryNotAvailable	Repository was unable to access the Registry	P
XDSRegistryOutOfResources XDSRepositoryOutOfResources	Resources are low.	P,R,SQ, XGQ P, RS, XGR
XDSResultNotSinglePatient	This error signals that the single Stored Query would have returned metadata for multiple Patient Ids	SQ, RS
XDSSStoredQueryMissingParam	A required parameter to a stored query is missing.	SQ, XGQ
XDSSStoredQueryParamNumber	A parameter which only accepts a single value is coded with multiple values	SQ, XGQ
XDSTooManyResults	Query resulted in too many results. No results are returned.	Q,SQ, XGQ
XDSUnavailableCommunity	A community which would have been contacted was not available. See Note 2.	SQ, RS
XDSUnknownCommunity	A value for the homeCommunityId is not recognized	SQ, XGQ, RS, XGR
XDSUnknownPatientId	Patient Id referenced in metadata is not known by the receiving actor. The codeContext shall include the value of patient Id in question.	P,R, XGQ Note: this error code is not used in the response to Registry Stored

Error Code <sup>1</sup>	Discussion	Transaction (See Note 1)
		Query
XDSUnknownRepositoryId	The repositoryUniqueId value could not be resolved to a valid document repository or the value does not match the repositoryUniqueId.	RS, XGR
XDSUnknownStoredQuery	The Query Id provided in the request is not recognized.	SQ, XGQ

Note 1:

P = Provide and Register-b

R = Register-b

SQ = Stored Query

2590

RS = Retrieve Document Set

XGQ = Cross Gateway Query

XGR = Cross Gateway Retrieve

Note 2:

Two examples of the use of error code XDSUnavailableCommunity are:

2595

A Cross Gateway Query or Cross Gateway Retrieve fails because the community identified by a homeCommunityId could not be contacted.

A Cross Gateway Query based on Patient ID could not contact some known communities to relay the query.

The error would be generated by the Initiating Gateway and returned in the Registry Stored Query or Retrieve Document Set. This would only apply when XDS Affinity Domain Option was used.

2600

#### 4.2.4.2 Error responses

The RegistryError elements shall be contained within an ebXML RegistryErrorList element in the ebXML RegistryResponse or ebXML AdhocQueryResponse.

The status attribute of the RegistryResponse or AdhocQueryResponse shall be one of the following values:

2605

urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success

urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure

urn:ihe:iti:2007:ResponseStatusType:PartialSuccess

The highestSeverity attribute may be present in a RegistryErrorList. If present, it shall contain the severity of the most severe of the RegistryErrors in the RegistryErrorList.

2610

The following tables explain the meaning of the status attribute in RegistryResponse and AdhocQueryResponse. Tables 4.2.4.2-1, 4.2.4.2-2, 4.2.4.2-3 and 4.2.4.2-4 below indicate whether the RegistryErrorList element shall be present and what other content shall be part of the response.

2615

**Table 4.2.4.2-1: [ITI-41] Provide and Register Document Set-b Responses**

RegistryResponse status	RegistryErrorList element	Result
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success	May be present. If present will contain one or more RegistryError elements with warning severity, none with error severity.	All metadata defined in this volume, and documents were successfully registered. Extra metadata may or may not be saved, based on the presence of the XDSExtraMetadataNotSaved warning.
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure	Present, contains one or more RegistryError elements. At least one has error severity others may have warning severity.	Metadata and documents not stored

**Table 4.2.4.2-2: [ITI-42] Register Document Set-b Responses**

RegistryResponse status	RegistryErrorList element	Result
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success	May be present. If present will contain one or more RegistryError elements with warning severity, none with error severity	All metadata defined in this volume was successfully registered Extra metadata may or may not be saved, based on the presence of the XDSExtraMetadataNotSaved warning.
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure	Present, contains one or more RegistryError elements. At least one has error severity, others may have warning severity.	Metadata not stored

**Table 4.2.4.2-3: [ITI-18] Stored Query and [ITI-38] Cross Gateway Query Responses**

AdhocQueryResponse status	RegistryErrorList element	Result
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success	May be present. If present will contain one or more RegistryError elements with warning severity; none with error severity	Results shall be returned. Results may contain zero or more entries.
urn:ihe:iti:2007:ResponseStatusType:PartialSuccess	Present, contains one or more RegistryError elements. At least one has error severity; others may have warning severity.	Results shall be returned. Results may contain zero or more entries.
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure	Present, contains one or more RegistryError elements. At least one has error severity; others may have warning severity.	Results not returned

2620

**Table 4.2.4.2-4: [ITI-43] Retrieve Document Set and [ITI-39] Cross Gateway Retrieve Responses**

Registry Response status	RegistryErrorList element	Result
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success	May be present. If present will contain one or more RegistryError elements with warning severity; none with error severity	All documents were successfully retrieved
urn:ihe:iti:2007:ResponseStatusType:PartialSuccess	Present, contains one or more RegistryError elements. At least one has error severity; others may have warning severity.	Some documents were successfully retrieved
urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure	Present, contains one or more RegistryError elements. At least one has error severity; others may have warning severity.	No documents were successfully retrieved

Note: Complete details on how these elements shall be populated are available at ITI TF-2b: 3.43.5 Protocol Requirements.

## 4.2.5 Metadata Vocabulary

2625 The UUIDs in the following tables shall be used in constructing and interpreting XDS metadata.

### 4.2.5.1 Submission Set Object UUIDs

UUID	Use/meaning
urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd	SubmissionSet ClassificationNode
urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d	author External Classification Scheme
urn:uuid:aa543740-bdda-424e-8c96-df4873be8500	contentTypeCode External Classification Scheme
urn:uuid:6b5ae1a-874d-4603-a4bc-96a0a7b38446	patientId External Identifier
urn:uuid:554ac39e-e3fe-47fe-b233-965d2a147832	sourceId External Identifier
urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8	uniqueId External Identifier

### 4.2.5.2 Document Entry Object

UUID	Use/meaning
urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1	DocumentEntry objectType for Stable Document Entries
urn:uuid:34268e47-fdf5-41a6-ba33-82133c465248	DocumentEntry objectType for On-Demand Document Entries
urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d	author External Classification Scheme
urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a	classCode External Classification Scheme
urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f	confidentialityCode External Classification Scheme

UUID	Use/meaning
urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4	eventCodeList External Classification Scheme
urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d	formatCode External Classification Scheme
urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1	healthCareFacilityTypeCode External Classification Scheme
urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427	patientId ExternalIdentifier
urn:uuid:ccc5598-8b07-4b77-a05e-ae952c785ead	practiceSettingCode External Classification Scheme
urn:uuid:f0306f51-975f-434e-a61c-c59651d33983	typeCode External Classification Scheme
urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab	uniqueId ExternalIdentifier

### 4.2.5.3 Folder Object

UUID	Use/meaning
urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2	Folder ClassificationNode
urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5	codeList External Classification Scheme
urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a	patientId External Identifier
urn:uuid:75df8f67-9973-4fbe-a900-df66cefec5a	uniqueId External Identifier

## 4.3 Additional Document Sharing Requirements

2630 This section includes profile specific requirements.

### 4.3.1 Requirements on Submission Type Transactions

A Submission Type Transaction transfers a collection of Document Sharing metadata to a receiving actor for processing. There are three Submission Type transactions:

- ITI-42 Register Document Set-b – transfers only Document Sharing metadata
- 2635 • ITI-41 Provide and Register Document Set-b – transfers Document Sharing metadata along with document content
- ITI-32 Distribute Document Set on Media - transfers Document Sharing metadata along with document content

2640 The Document Sharing metadata Objects found in the Submission Request of a Submission Type Transaction are:

- SubmissionSet metadata
- Metadata for new or existing documents
- Folders to be created
- Associations (various relationships and HasMember as needed)

2645 An XDS Registry Submission Request is the collection of metadata transferred between a Document Repository and a Document Registry. This request is part of the Register Document Set-b [ITI-42] and Register On-Demand Document Entry [ITI-61] transactions.

2650 An XDS Repository Submission Request is the collection of metadata and documents transferred between a Document Source and a Document Repository. This request is the information payload of the Provide and Register Document Set message of the Provide and Register Document Set-b [ITI-41] transaction.

#### 4.3.1.1 Submission Metadata Attribute Optionality

This section lists which metadata attributes an actor shall provide when initiating a Submission Type Transaction.

2655 The Actor/Transaction pairs addressed by this section are as follows:

**Table 4.3.1.1-1: Sending Actor/Transaction Pairs**

Actor	Transaction	Shortname
XDS Document Source	ITI-41 Provide and Register Document Set-b	XDS DS
XDS Document Repository	ITI-42 Register Document Set-b	XDS DR
XDM Portable Media Creator	ITI-32 Distribute Document Set on Media	XDM MC
XDR Document Source	ITI-41 Provide and Register Document Set-b	XDR DS
XDR Metadata-Limited Document Source	ITI-41 Provide and Register Document Set-b	XDR MS
XDS On-Demand Document Source	ITI-61 Register On-Demand Document Entry	XDS OD

2660 For each actor/transaction pair across the top of table 4.3.1.1-3, and each metadata attribute row, the cell indicates the requirement for that actor when creating a submission request using the paired transaction. The requirements are expressed through these codes:

**Table 4.3.1.1-2: Sending Actor Metadata Attribute Optionality Code Definitions**

Code	Meaning
R	Required – A value for the attribute shall be supplied by the sending Actor when sending the submission
R2	Required if Known – A value for the attribute shall be supplied by the sending Actor when sending the submission unless the Actor does not have any value for the attribute
O	Optional – The sending Actor may or may not supply a value for this attribute
X	Prohibited – when sending a submission, a value for the attribute shall not be supplied by the sending Actor.

2665 **Table 4.3.1.1-3: Sending Actor Metadata Attribute Optionality**

Metadata Element	Metadata Attribute	XDS DS	XDS DR	XDM MC	XDR DS	XDR MS	XDS OD
DocumentEntry	author	R2	R2	R2	R2	R2	R2
DocumentEntry	availabilityStatus	O	O	O	O	O	O
DocumentEntry	classCode	R	R	R2	R	R2	R

IHE IT Infrastructure Technical Framework, Volume 3 (ITI TF-3): Cross-Transaction and Content Specifications

Metadata Element	Metadata Attribute	XDS DS	XDS DR	XDM MC	XDR DS	XDR MS	XDS OD
DocumentEntry	comments	O	O	O	O	O	O
DocumentEntry	confidentialityCode	R	R	R2	R	R2	R
DocumentEntry	creationTime	R	R	R2	R	R2	X
DocumentEntry	entryUUID	R	R	R	R	R	R
DocumentEntry	eventCodeList	O	O	O	O	O	O
DocumentEntry	formatCode	R	R	R2	R	R2	R
DocumentEntry	hash	O	R	R	O	O	X
DocumentEntry	healthcareFacilityTypeCode	R	R	R2	R	R2	R
DocumentEntry	homeCommunityId	O	O	O	O	O	O
DocumentEntry	languageCode	R	R	R2	R	R2	R
DocumentEntry	legalAuthenticator	O	O	O	O	O	O
DocumentEntry	limitedMetadata	X	X	O	X	R	X
DocumentEntry	mimeType	R	R	R	R	R	R
DocumentEntry	objectType	R	R	R	R	R	R
DocumentEntry	patientId	R	R	R2	R	R2	R
DocumentEntry	practiceSettingCode	R	R	R2	R	R2	R
DocumentEntry	referenceIdList	O	O	O	O	O	O
DocumentEntry	repositoryUniqueId	O	R	O	O	O	<u>O</u>
DocumentEntry	serviceStartTime	R2	R2	R2	R2	R2	O
DocumentEntry	serviceStopTime	R2	R2	R2	R2	R2	O
DocumentEntry	size	O	R	R	O	O	X
DocumentEntry	sourcePatientId	R	R	R2	R	R2	R
DocumentEntry	sourcePatientInfo	O	O	R2	O	R2	O
DocumentEntry	title	O	O	O	O	O	O
DocumentEntry	typeCode	R	R	R2	R	R2	R
DocumentEntry	uniqueId	R	R	R	R	R	R
DocumentEntry	URI	O	O	R	O	O	O
SubmissionSet	author	R2	R2	R2	R2	R2	R2
SubmissionSet	availabilityStatus	O	O	O	O	O	O
SubmissionSet	comments	O	O	O	O	O	O
SubmissionSet	contentTypeCode	R	R	R2	R	R2	R
SubmissionSet	entryUUID	R	R	R	R	R	R
SubmissionSet	homeCommunityId	O	O	O	O	O	O
SubmissionSet	intendedRecipient	O	O	R2	R2	R2	O
SubmissionSet	limitedMetadata	X	X	O	X	R	X
SubmissionSet	patientId	R	R	R2	R	R2	R
SubmissionSet	sourceId	R	R	R	R	R	R
SubmissionSet	submissionTime	R	R	R	R	R	R

Metadata Element	Metadata Attribute	XDS DS	XDS DR	XDM MC	XDR DS	XDR MS	XDS OD
SubmissionSet	title	O	O	O	O	O	O
SubmissionSet	uniqueId	R	R	R	R	R	R
Folder	availabilityStatus	O	O	O	O	O	O
Folder	codeList	R	R	R2	R	R2	R
Folder	comments	O	O	O	O	O	O
Folder	entryUUID	R	R	R	R	R	R
Folder	homeCommunityId	O	O	O	O	O	O
Folder	lastUpdateTime	O	O	O	O	O	O
Folder	limitedMetadata	X	X	O	X	R	X
Folder	patientId	R	R	R2	R	R2	R
Folder	title	R	R	O	R	O	R
Folder	uniqueId	R	R	R	R	R	R

#### 4.3.1.2 XDS Specific Requirements

In an XDS setting, a DocumentEntry is stored in the XDS Document Registry and its related document is stored in a XDS Document Repository.

2670 A DocumentEntry in the XDS Document Registry is used to represent a document in a XDS Document Repository.

A DocumentEntry object in the XDS Document Registry contains a reference to a single document in a single XDS Document Repository.

Note: A repository may hold documents that are not indexed in the registry.

2675 ITI TF-2x: Appendix H defines the metadata to initialize an ebXML registry to serve as an XDS Document Registry.

SubmissionSets are immutable, except in cases where patientID must be updated.

A Submission Request is the collection of information that is transferred to an XDS Document Registry or Repository.

##### 4.3.1.2.1 Provide and Register Document Set-b Transaction [ITI-41]

2680 XDS uses ITI-41 to transfer the collection of metadata and documents between an XDS Document Source and an XDS Document Repository using a single ebXMLSubmitObjectsRequest. This request contains:

- Metadata
  - Submission Set metadata
  - Document Metadata
  - Folders to be created
  - Associations (various relationships and HasMember as needed)

2685

- 2690
- Zero or more documents (Each document is represented by a DocumentEntry object in the metadata. Submissions that add metadata to the Document Registry without adding documents to the Document Repository are possible. If the metadata contains a DocumentEntry, then the Document must be present.)

The metadata is relayed by the XDS Document Repository to the XDS Document Registry through a Register Document Set-b [ITI-42] transaction. Relayed metadata may include fields added by the XDS Document Repository.

- 2695
- Id attributes and references: The submission may contain valid UUIDs or symbolic ids (or both) as specified in section 4.3.1.2.2.

#### 4.3.1.2.2 Register Document Set-b Transaction [ITI-42]

XDS uses ITI-42 to transfer the collection of metadata between a XDS Document Repository and a XDS Document Registry in a single ebXMLSubmitObjectsRequest. This request contains:

- 2700
- SubmissionSet metadata
  - DocumentEntry metadata
  - Folders to be created
  - Associations (various relationships and HasMember as needed)

2705 **Document duplication:** There are several conditions regarding the duplication of document that can occur.

- Duplicate registration of a document - A document and its metadata are submitted to the XDS Document Repository as part of a Provide and Register Document Set [ITI-41] transaction. This document already exists in one or more Document Repositories and is already represented in the Document Registry. It is submitted with a new (not previously used) entryUUID for the DocumentEntry and associated ancillary objects. The Document Registry shall accept such duplicate registration of the documents.
- Duplicate document id submitted to repository - A document with its associated metadata is submitted to an XDS Document Repository. A document with the same DocumentEntry.uniqueId is present in the repository but the DocumentEntry.hash is different. This is an error and the Submission Request shall be rejected by the Document Repository.

2715

Note: There are two approaches to detecting this fault. First, this can be detected at the repository if repository logic can validate the hashes and has record of the document id to compare. Otherwise the request can be forwarded on to the registry and let the fault be detected by the registry (see next bullet). The repository then deals with the error returned by the registry.

2720

- Duplicate document Id submitted to registry - Metadata representing a document (DocumentEntry) is part of a Registry Submission Request. A DocumentEntry object with the same uniqueId is present in the registry but the hash is different. This is an error and the Submission Request shall be rejected by the XDS Document Registry.

2725 Note: A document can be registered by multiple Document Sources with the same uniqueId, same hash, different entryUUID, and with other metadata attributes not the same as described above. As a result, a Document Consumer may issue a GetDocuments Stored Query with a uniqueId parameter and have returned two or more DocumentEntry objects with that same uniqueId.

2730 **Id attributes and references:** Registries shall only accept and produce lowercase UUIDs as specified in section 4.2.3.1.4.

Id references, or pointers, between elements of metadata are necessary for forming submissions and understanding query responses. The sourceObject and targetObject attributes of an Association are id references. Classifications and ExternalIdentifier type attributes are nested inside the objects they describe but they also contain id references to their parent objects.

2735 Id references to objects in the Registry shall be in UUID format. Id references to other objects within a submission may be UUID or symbolic format. All Id references returned in a Stored Query shall be in UUID format. A responsibility of a Document Registry when accepting a Register Document Set transaction is to translate any symbolic Ids into UUIDs before storing them in the registry.

2740 Symbolic Id format is any string that does not have the prefix 'urn:uuid:'.

Once a UUID format Id value is assigned to a Registry Object, that value shall be permanent and shall not be changed.

#### 4.3.1.2.3 XDS Atomicity of Submissions

2745 XDS Submission requests shall be atomic operations. If any part of the Submission Request Processing fails, the entire transaction shall fail and no changes result. The result of a successful Submission Request is to update either:

a Document Registry or

a Document Registry and a Document Repository.

2750 The XDS Document Registry addresses the fact that the ebXML Registry specification does not guarantee that SubmitObjectsRequest are atomic. XDS specifies the mechanism through which atomicity is to be implemented and where it is needed.

If an error occurs, storing documents in the repository then all documents stored as part of the Repository Submission Request shall be removed.

2755 If an error occurs, storing metadata in the Document Registry, then the following actions shall be performed:

All metadata stored as part of the Registry Submission Request shall be removed from the Document Registry

All documents stored as part of the Repository Submission Request shall be removed. This only applies if the Registry Submission Request is a result of a Repository Submission Request.

2760 Registry queries from the Registry Stored Query [ITI-18] transaction shall not find XDS SubmissionSets, XDS Folders or DocumentEntry objects until after the above atomic operation

that creates them has completed successfully and the availabilityStatus attributes of the Folders and DocumentEntries have been set to Approved.

2765 For document metadata included by value, a rollback of the submission shall delete the document metadata and the Association. For document metadata included by reference, a rollback of the submission shall not delete the document metadata but shall still delete the Association. The document whose Association is being deleted existed before this submission and shall be maintained.

#### 4.3.1.2.4 XDS Registry Enforcement of Attributes

2770 The XDS Document Registry enforces rules on metadata content. Other document sharing profiles have their own different enforcement specifications. A summary of attribute enforcement is provided in Tables 4.3.1.2-1, 4.3.1.2-2 and 4.3.1.2-3; details on certain attributes are given in Section 4.3.1.2.5 XDS Registry Responsibilities.

2775 The Registry shall reject any submission which includes attribute values whose size exceeds the specification in the standard.

2780 ebRIM version 3.0 data type Slot/ValueList/Value is limited to 256 characters before XML escaping by that standard. Document Sharing metadata may have restricted this to 128 characters or less. Some HL7 data types may be longer according to HL7; however, when encoded as slots, the Document Source shall restrict these to fit into the 256 character space allocated. This may require some information to be excluded. The XDS Profile gives no guidance as to how information is to be excluded to make this coding limit.

The XDS Registry shall enforce the cardinality rules as per Section 4.2.3.1.9.

**Table 4.3.1.2-1: DocumentEntry Metadata Attribute Enforcement**

DocumentEntry Attribute	Registry Enforcement
availabilityStatus	Enforcement of format only
authorInstitution	Enforcement of format only
authorPerson	Enforcement of format only
authorRole	Enforcement of format only
authorSpecialty	Enforcement of format only
authorTelecommunication	Enforcement of format only
classCode	Coding Scheme and Code Value and CodeDisplayName
confidentialityCode	Coding Scheme and Code Value and CodeDisplayName
creationTime	Enforcement of format only
entryUUID	Enforcement of format only
eventCodeList	Coding Scheme and Code Value and CodeDisplayName
formatCode	Coding Scheme and Code Value and

DocumentEntry Attribute	Registry Enforcement
	CodeDisplayName
hash	Hash value enforcement
healthcareFacilityTypeCode	Coding Scheme and Code Value and CodeDisplayName
legalAuthenticator	Enforcement of format only
languageCode	Optionally enforced by XDS Affinity Domain
mimeType	Code Value
patientId	Authority Domain Id Patient Id (known from patient identity feed)
practiceSettingCode	Coding Scheme and Code Value and CodeDisplayName
referenceIdList	If present and not empty then shall be CXi encoded
serviceStartTime	Enforcement of format only
serviceStopTime	Verifies serviceStartTime <= serviceStopTime
size	Enforcement of format only
sourcePatientId	Enforcement of format only
sourcePatientInfo	Some parts required
title	Enforcement of format only
typeCode	Coding Scheme and Code Value and CodeDisplayName
uniqueId	No identical existing uniqueId in registry (assigned to DocumentEntry, SubmissionSet, or Folder) , unless there is duplicate Document registration as described in the Document Duplication text in section 4.3.1.2.2
URI	Enforcement of format only

2785

**Table 4.3.1.2-2: SubmissionSet Metadata Attribute Enforcement**

SubmissionSet Attribute	Registry Enforcement
authorInstitution	Enforcement of format only
authorPerson	Enforcement of format only
authorRole	Enforcement of format only
authorSpecialty	Enforcement of format only
authorTelecommunication	Enforcement of format only
comments	Enforcement of format only
contentTypeCode	Coding Scheme and Code value and CodeDisplayName
patientId	Authority Domain Id Patient Id (known from patient identity feed)
sourceId	Enforcement of format only
submissionTime	Enforcement of format only

SubmissionSet Attribute	Registry Enforcement
uniqueId	No identical existing uniqueId in registry (assigned to DocumentEntry, SubmissionSet, or Folder)

**Table 4.3.1.2-3: Folder Metadata Attribute Enforcement**

Folder Attribute	Registry Enforcement
codeList	Coding Scheme and Code value and CodeDisplayName
comments	Enforcement of format only
lastUpdateTime	Shall be set to the current time on submission of folder or folder content changes.
patientId	The value of the patientId shall be the same for all documents of a Folder.
uniqueId	No identical existing uniqueId in registry (assigned to DocumentEntry, SubmissionSet, or Folder)

#### 4.3.1.2.5 XDS Registry Responsibilities

2790 The XDS Document Registry extends functionality beyond the ebXML registry standard. The Document Registry performs validation as specified below:

2795 **Validate patient Id** –patientIds (such as the DocumentEntry.patientId attribute) shall be a known patient Id and registered against the Patient Id Domain of the XDS Affinity Domain managed by the Patient Identity Source Actor for the XDS Affinity Domain. This validation requirement does not apply to DocumentEntry.sourcePatientId.

- Patient identifiers shall be verified against the patient identifiers received from the Patient Identity Feed [ITI-8] indicating that it is registered against the Patient ID Domain of the Affinity Domain.
- Patient identifiers shall be verified against known subsumed patient identifiers received from the Patient Identity Feed [ITI-8]. A submission containing a subsumed patient identifier shall return an XDSUnknownPatientId error. See Section 3.8.4.2.4 for background on patient identifier merge.

2800 **Validate submitted metadata** – the adaptor shall verify that submitted metadata meets XDS Registry metadata specification

2805 **Verify coded values** – the adaptor shall verify that coded fields (ebXML external classifications) contain valid XDS specified values or where the XDS Affinity Domain constrains code values, to verify them. Optionally, verify that the code display names match the code value and schema.

2810 **Ensure submissions are atomic** - The adaptor shall make submission to registry an atomic operation – see Section 4.3.1.2.3 XDS Atomicity of Submissions for Submission Requests for atomicity requirements.

If the registry submission is successful then the adaptor shall label all submitted DocumentEntry, Folder, and SubmissionSet objects as Approved. If the registry submission fails then the adaptor shall remove from the registry all objects stored as part of this SubmissionSet.

2815 **Support Folder membership** - When the Document Registry Actor detects a HasMember association in the submission with a sourceObject attribute which references a Folder in the submission or in the registry then it shall verify that:

- The targetObject references a DocumentEntry in the submission or registry.
- If the targetObject is in the registry then it shall have availabilityStatus of Approved.
- There is also a HasMember association linking the SubmissionSet to the Folder-DocumentEntry association.

2820

**Support document relationship associations** - When a Submission Request includes a 'APND', 'XFRM', 'RPLC', 'XFRM\_RPLC' or 'Signs' association the Document Registry shall verify the association attributes as follows.

- The sourceObject references a DocumentEntry in the submission
- The targetObject references a DocumentEntry in the Registry
- The targetObject DocumentEntry has availabilityStatus of Approved

2825

**Support document replacement** - When a Submission Request includes a 'RPLC' or 'XFRM\_RPLC' Association indicating that a DocumentEntry is being replaced, the following shall be true:

- DocumentEntry to be replaced shall have status = Approved.
- The Association's sourceObject attribute shall contain the Id (UUID or symbolic Id) of an ExtrinsicObject representing a DocumentEntry included in the SubmissionSet.
- The Association's targetObject attribute shall contain the UUID of an ExtrinsicObject (DocumentEntry) already in the registry.

2830

When the 'RPLC' or 'XFRM\_RPLC' Association is detected by the Registry Adaptor it shall:

- Return the error XDSReplaceFailed if this object is not contained in the registry, or has status other than Approved. This ensures that only the most recent version of a document can be replaced.
- Submit the Submission Request to the registry.

2840

- If the submission is successful, label the replacement document as Approved and the replaced document as Deprecated.
- The existing document and the document replacing it shall have the same patient identifier (XSDDocumentEntry.patientId attribute). This comparison shall take into consideration patient identity merges as described in ITI TF-2a: 3.8.4.2.4.

2845 **Validate patientIds in Folders** - The adaptor shall verify that all DocumentEntries in a Folder are for the same patient. Specifically, verify that the value of the patientId attribute of the Folder matches the value of the patientId attribute of each DocumentEntry in the Folder. This comparison shall take into consideration patient identity merges as described in ITI TF-2a: 3.8.4.2.4.

2850 **Validate patientId in SubmissionSets:**

- Must match included Folders
- Must match include DocumentEntries (by value)

**Validate MIME types** - The adaptor shall validate that the mimeType DocumentEntry attribute for all DocumentEntries received is on the approved list for this XDS Affinity Domain.

2855 **Maintain Folder attribute ‘lastUpdateTime’** – The Folder attribute lastUpdateTime shall be updated by the adaptor every time a new DocumentEntry is added to a Folder.

**Validate patientId on documents being added to a Folder** - The value of the patientId attribute of a DocumentEntry object shall match the value of the patientId attribute on any folder that holds it.

2860 **Accept submissions containing multiple DocumentEntries**– The adaptor shall be capable of accepting submissions containing multiple DocumentEntries.

#### **4.3.1.2.6 Required Initialization of the XDS Affinity Domain**

2865 This initialization supports the operation of the Registry Adaptor. The following information must be provided by the XDS Affinity Domain administrator and loaded into the Registry Adaptor. This supports the functionality specified for the Registry Adaptor in Section 4.3.1.2. How this information is loaded into the Registry Adaptor or how the Registry Adaptor is implemented is not defined by the XDS Profile.

1. List of acceptable mimeTypes for documents indexed by the registry.
- 2870 2. Domain name (Patient Id Assigning Authority) for XDS Affinity Domain. PatientIds attached to metadata submitted to this registry shall come from this Assigning Authority.
3. Acceptable values for all coded attributes characterized in the registry by ebXML External Classifications.

#### **4.3.2 Requirements on Query Type Transactions**

##### **4.3.2.1 Query Type Metadata Attribute Optionality**

2875 This section lists which attributes shall contain content in a response to a query transaction. The query transactions covered in Table 4.3.2.1-1. The content of the query request can be found within the referenced transaction.

**Table 4.3.2.1-1: Responding Actor/Transaction Pairs**

Actor	Transaction	Shortname
XDS Document Registry	ITI-18 Registry Stored Query	XDS DR
XCA Initiating Gateway	ITI-18 Registry Stored Query	XCA IG
XCA Responding Gateway	ITI-38 Cross-Gateway Query	XCA RG
MPQ Document Registry	ITI-51 Multi-Patient Stored Query	MPQ DR

2880 For each actor/transaction pair across the top of the table, and each metadata attribute row, the cells indicates the requirement for that actor when creating a query response to the paired transaction. The requirements are expressed through these codes:

**Table 4.3.2.1-2: Responding Actor Metadata Attribute Optionality Code Definitions**

Code	Meaning
R	Required – A value for the attribute shall be supplied by the responding Actor when responding to a query
R2	Required if Known – A value for the attribute shall be supplied by the responding Actor when responding to the query if a value is available to the Actor. For the Document Registry it must supply the value specified in the submission request.
O	Optional – The responding Actor may or may not supply a value for this attribute. For the Document Registry it must supply the value specified in the submission request.
X	Prohibited – When responding to a query, a value for the attribute shall not be supplied by the responding Actor.
R3	Required for Stable DocumentEntries and not allowed for On-Demand DocumentEntries.

2885

**Table 4.3.2.1-3: Responding Actor Metadata Attribute Optionality**

Metadata Element	Metadata Attribute	XDS DR	XCA IG	XCA RG	MPQ DR
DocumentEntry	author	R2	R2	R2	R2
DocumentEntry	availabilityStatus	R	R	R	R
DocumentEntry	classCode	R	R	R	R
DocumentEntry	comments	O	O	O	O
DocumentEntry	confidentialityCode	R	R	R	R
DocumentEntry	creationTime	R3	R3	R3	R3
DocumentEntry	entryUUID	R	R	R	R
DocumentEntry	eventCodeList	O	O	O	O
DocumentEntry	formatCode	R	R	R	R
DocumentEntry	hash	R3	R3	R3	R3
DocumentEntry	healthcareFacility TypeCode	R	R	R	R
DocumentEntry	homeCommunityId	O	R	R	O
DocumentEntry	languageCode	R	R	R	R
DocumentEntry	legalAuthenticator	O	O	O	O

Metadata Element	Metadata Attribute	XDS DR	XCA IG	XCA RG	MPQ DR
DocumentEntry	limitedMetadata	X	X	X	X
DocumentEntry	mimeType	R	R	R	R
DocumentEntry	objectType	R	R	R	R
DocumentEntry	patientId	R	R	R	R
DocumentEntry	practiceSettingCode	R	R	R	R
DocumentEntry	referenceIdList	O	O	O	O
DocumentEntry	repositoryUniqueId	R	R	R	R
DocumentEntry	serviceStartTime	R2	R2	R2	R2
DocumentEntry	serviceStopTime	R2	R2	R2	R2
DocumentEntry	size	R3	R3	R3	R3
DocumentEntry	sourcePatientId	R	R	R	R
DocumentEntry	sourcePatientInfo	O	O	O	O
DocumentEntry	title	O	O	O	O
DocumentEntry	typeCode	R	R	R	R
DocumentEntry	uniqueId	R	R	R	R
DocumentEntry	URI	O	O	O	O
SubmissionSet	author	R2	R2	R2	R2
SubmissionSet	availabilityStatus	R	R	R	R
SubmissionSet	comments	O	O	O	O
SubmissionSet	contentTypeCode	R	R	R	R
SubmissionSet	entryUUID	R	R	R	R
SubmissionSet	homeCommunityId	O	R	R	O
SubmissionSet	intendedRecipient	O	O	O	O
SubmissionSet	limitedMetadata	X	X	X	X
SubmissionSet	patientId	R	R	R	R
SubmissionSet	sourceId	R	R	R	R
SubmissionSet	submissionTime	R	R	R	R
SubmissionSet	title	O	O	O	O
SubmissionSet	uniqueId	R	R	R	R
Folder	availabilityStatus	R	R	R	R
Folder	codeList	R	R	R	R
Folder	comments	O	O	O	O
Folder	entryUUID	R	R	R	R
Folder	homeCommunityId	O	R	R	O
Folder	lastUpdateTime	O	O	O	O
Folder	limitedMetadata	X	X	X	X
Folder	patientId	R	R	R	R
Folder	title	R	R	R	R

<b>Metadata Element</b>	<b>Metadata Attribute</b>	<b>XDS DR</b>	<b>XCA IG</b>	<b>XCA RG</b>	<b>MPQ DR</b>
Folder	uniqueId	R	R	R	R

## 2890 **5 IHE Content Specifications**

This section follows the documentation pattern found in the IHE PCC Technical Framework. The reader should be familiar with the IHE PCC Technical Framework.

### **5.1 Basic Patient Privacy Consents Module**

2895 This section describes the encoding requirements for the Basic Patient Privacy Consents Document Content.

The BPPC document has two possible document templates, one without a scanned document part, and one in ITI TF-3: 5.1.2 defines the requirements of the BPPC document without a scanned document part; ITI TF-3: 5.1.3 explains with a scanned document part.

#### **5.1.1 References**

- 2900
- HL7 CDA Release 2.0 (denoted HL7 CDA R2, or just CDA, in subsequent text)

#### **5.1.2 Patient Privacy Consent Acknowledgment Document Specification 1.3.6.1.4.1.19376.1.5.3.1.1.7 – With no Scanned Document Part**

2905 A patient acknowledgement of a Patient Privacy Consent Policy is a document that contains machine readable indication. This specification describes the BPPC document without a scanned part. When the Patient Privacy Consent Acknowledgment Document contains a Scanned Document (XDS-SD), it will conform to ITI TF-3: 5.1.3.

##### **5.1.2.1 XDS Metadata**

###### **5.1.2.1.1 XDS DocumentEntry Metadata**

2910 BPPC leverages the XDS DocumentEntry Metadata requirements in the IHE PCC TF-2: 4.1.1 unless otherwise specified below.

###### **5.1.2.1.1.1 XDSDocumentEntry.classCode**

The LOINC code for these documents is “57016-8” “Privacy Policy Acknowledgement Document” and the codeSystem is 2.16.840.1.113883.6.1.

For backward compatibility historic documents may be registered with:

- 2915
- classCode -- This attributes may have been set to the value "Consent".
  - classCodeDisplayName -- This attributes may have been set to the value "Consent".

###### **5.1.2.1.1.2 XDSDocumentEntry.eventCodeList**

- eventCodeList -- the eventCodeList shall be populated using the Patient Privacy Policy Identifiers that have been acknowledged to within the document.

- 2920
- /ClinicalDocument/documentationOf/serviceEvent[templateId/@root='1.3.6.1.4.1.19376.1.5.3.1.2.6']/code/@code
  - eventCodeDisplayNameList -- The eventCodeDisplayNameList shall be populated using the display names for those policies.
- 2925
- /ClinicalDocument/documentationOf/serviceEvent[templateId/@root='1.3.6.1.4.1.19376.1.5.3.1.2.6']/code/@displayName

#### 5.1.2.1.1.3 XSDDocumentEntry.formatCode

The XSDDocumentEntry format code for this content shall be **urn:ihe:iti:bppc:2007**. The formatCode codeSystem shall be **1.3.6.1.4.1.19376.1.2.3**.

#### 5.1.2.1.1.4 XSDDocumentEntry.uniqueld

- 2930
- This value shall be the `ClinicalDocument/id` in the HL7 CDA R2 header. The root attribute is required, and the extension attribute is optional. The total length is limited to 256 characters. Additionally see IHE PCC TF-2: 4.1.1, for further content specification.

#### 5.1.2.1.2 XDS SubmissionSet Metadata

No additional constraints. For more information, see IHE PCC TF-2: 4.1.2

- 2935
- #### 5.1.2.1.3 XDS Folder Metadata

No additional requirements. For more information, see IHE PCC TF-2: 4.1.3

#### 5.1.2.2 Specification

- 2940
- CDA Release 2.0 documents that conform to the requirements of this content module shall indicate their conformance by the inclusion of the appropriate `<templateId>` elements in the header of the document. This is shown in the sample document below. A CDA Document may conform to more than one template. This content module inherits from the Medical Document content module, and so must conform to the requirements of that template as well, thus all `<templateId>` elements shown in the example below shall be included.

```

2945 <ClinicalDocument xmlns='urn:hl7-org:v3'>
      <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
      <templateId root='1.3.6.1.4.1.19376.1.5.3.1.1.1' />
      <templateId root='1.3.6.1.4.1.19376.1.5.3.1.1.7' />
      <id root=' ' extension=' ' />
2950 <code code='57016-8' displayName='PATIENT PRIVACY ACKNOWLEDGEMENT'
      codeSystem='2.16.840.1.113883.6.1' codeSystemName='LOINC' />
      <title>Consent to Share Information</title>
      <effectiveTime value='20070619012005' />
      <confidentialityCode code='N' displayName='Normal'
2955   codeSystem='2.16.840.1.113883.5.25' codeSystemName='Confidentiality' />
      <languageCode code='en-US' />
      :
      <component><structuredBody>
2960   </structuredBody></component>
</ClinicalDocument>
    
```

**Figure 5.1.2.2-1: Sample Consent to Share Information Document**

A Patient Privacy Acknowledgement Document shall contain a text description of what the patient consented to, a list of codes indicating the policy(s) agreed to, and a time range indicating the effective time of the consent. It may be attested to using an electronic digital signature, conforming to the ITI Digital Signature Profile.

A Patient Privacy Acknowledgement Document shall have one or more <serviceEvent> elements in the header identifying the policies acknowledged by the document (see section 4.2.3.4 of CDA R2). Each <serviceEvent> element indicates an acknowledgement of a Patient Privacy Policy.

More than one policy may be acknowledged to within a given document.

Data Element Name	Opt	Template ID
Patient Privacy Acknowledgement Service Event At least one, and possibly more than one acknowledgement can be provided within the document.	R	1.3.6.1.4.1.19376.1.5.3.1.2.6

**5.1.2.2.1 Patient Privacy Acknowledgement Service Events  
1.3.6.1.4.1.19376.1.5.3.1.2.6**

Within a Patient Privacy Acknowledgement Document, there shall be a Patient Privacy Acknowledgement Service Event with the effective time of the consent shall be specified within the documentationOf/serviceEvent element.

```
2980 <documentationOf typeCode='DOC'>
    <serviceEvent classCode='ACT' moodCode='EVN'>
        <templateId root='1.3.6.1.4.1.19376.1.5.3.1.2.6' />
        <id root='1.2.3.4.5.6' />
        <code code='9.8.7.6.5.4.3.2.1' displayName='Chicken Costume Consent'
codeSystem='9.8.7.6.5.4.3.2.2' codeSystemName='Silly Policy Land' />
2985     <effectiveTime>
        <low value='20101130' />
        </effectiveTime>
    </serviceEvent>
</documentationOf>
```

**Figure 5.1.2.2-2: Patient Privacy Acknowledgement Service Events Example**

#### 2990 **5.1.2.2.2 <documentationOf typeCode='DOC'>**

At least one <documentationOf> element shall exist, describing the service event. This element shall have a typeCode attribute with the value DOC.

#### **5.1.2.2.3 <serviceEvent classCode='ACT' moodCode='EVN'>**

2995 One <serviceEvent> shall exist for each Patient Privacy Policy acknowledged. This element shall have a classCode attribute set to ACT, and a moodCode attribute of EVN.

#### **5.1.2.2.4 <templateId root='1.3.6.1.4.1.19376.1.5.3.1.2.6' />**

The <templateId> element shall be recorded exactly as shown above, and identifies this <serviceEvent> as recording the acknowledgement of a Patient Privacy Policy.

#### **5.1.2.2.5 <id root=' ' />**

3000 The service event shall have one <id> element, providing an identifier for the service event. The root attribute of this element shall be present, and shall be a GUID or OID. The extension attribute shall not be present.

#### **5.1.2.2.6 <code code=' ' displayName=' ' codeSystem=' ' codeSystemName=' ' />**

3005 The <code> element shall be present, and shall indicate the Patient Privacy Policy acknowledged. The code attribute indicates the Patient Privacy Policy Identifier, and the codeSystem attribute indicates the code system for the Patient Privacy Policy Domain from which this Patient Privacy Policy Identifier is assigned. The displayName attribute may be present, and describes the Patient Privacy Policy acknowledgement. The codeSystemName attribute may be present, and describes the Patient Privacy Policy Domain code system.

#### 3010 **5.1.2.2.7 <effectiveTime><low value=' ' /><high value=' ' /></effectiveTime>**

The <effectiveTime> element shall be present, and shall indicate the effective time range over which acknowledgement is given. The low value must be provided. The high value may be present. If present, it shall indicate the maximum effective time of the consent. The

3015 effectiveTime must be the same for all ServiceEvents in one Patient Privacy Acknowledgement Document. If different effectiveTimes are needed then different documents should be generated.

### **5.1.3 Patient Privacy Consent Acknowledgment Document Specification**

#### **1.3.6.1.4.1.19376.1.5.3.1.1.7.1 – With Scanned Document**

3020 A patient acknowledgement of a Patient Privacy Consent Policy is a document that contains machine readable indication. This section specifies the BPPC document with a scanned document part.

#### **5.1.3.1 XDS Metadata**

The BPPC document shall conform to the requirements in section 5.1.2.1 with the formatCode exception listed below

#### **5.1.3.1.1 XDS DocumentEntry Metadata**

3025 The BPPC document shall conform to the XDS DocumentEntry Metadata requirements in the IHE PCC TF-2:5.1.1.1.1 unless otherwise specified below.

#### **5.1.3.1.1.1 XDSDocumentEntry.formatCode**

The XDSDocumentEntry format code for this content is **urn:ihe:iti:bppc-sd:2007**. The formatCode codeSystem shall be 1.3.6.1.4.1.19376.1.2.3.

#### **5.1.3.1.2 XDS SubmissionSet Metadata**

3030 No additional constraints. For more information, see IHE PCC TF-2: 5.1.1.1.2

#### **5.1.3.1.3 XDS Folder Metadata**

No additional requirements. For more information, see IHE PCC TF-2: 5.1.1.1.3

#### **5.1.3.3 Specification**

3035 This BPPC document shall conform to the XDS-SD (ITI TF-3: 5.2) specification and shall have the additional requirements stated in ITI TF-3: 5.1.2.3.

#### **5.1.3.4 Conformance**

See IHE ITI TF-3: 5.1.2.4.

## **5.2 Scanned Documents Content Module**

3040 This section outlines the content of the HL7 CDA R2 constraints for the document. We note here that requirements specified below are to ensure the presence of a minimum amount of wrapper data in order to enhance description and facilitate sharing of the document. Implementers of this profile can and should make use of additional annotation within the CDA header to provide richer context. The examples in the following sections contain the minimal amount of wrapper

3045 data, as specified, and in many cases do make use of additional CDA header elements for enriched context.

**Assumptions and Definitions:** We assume that the scanning facility and equipment within it are assigned an OID and that the scanning facility assembles the wrapped scanned content. More information regarding the construction of OIDs can be found in ITI TF-2x: Appendix B. We  
3050 define the following nomenclature for entity roles concerned in forming the wrapper content.

- *Original content* – Legacy paper or electronic document intended for wrapping.
- *Scanned content* – Scanned or appropriately converted/encoded electronic version of the original content.
- *Original author* – Author of the original content.
- 3055 • *(Scanner) Operator* – Person assembling the scanned content.

### 5.2.1 Referenced Standards

- PDF RFC 3778, The application/pdf Media Type (informative)
- PDF/A ISO 19005-1b. Document management - Electronic document file format for long-term preservation - Part 1: Use of PDF (PDF/A)
- 3060 • HL7 CDA Release 2.0 (denoted HL7 CDA R2, or just CDA, in subsequent text)
- [RFC 5646](#), Tags for the identification of languages

#### 5.2.1.1 Discussion of Content Standards

PDF and plaintext documents intended for wrapping can consist of multiple pages. Encoding of multiple page PDF documents are subject to the PDF/A standard. This ISO standard, PDF/A, is a  
3065 subset of Adobe PDF version 1.4 intended to be suitable for long-term preservation of page-oriented documents. PDF/A attempts to maximize:

- Device independence
- Self-containment
- Self-documentation

3070 The constraints imposed by PDF/A include:

- Audio and video content are forbidden
- JavaScript and executable file launches are prohibited
- All fonts must be embedded and also must be legally embeddable for unlimited, universal rendering
- 3075 • Colorspaces specified in a device-independent manner
- Encryption is disallowed (although the enclosing document and transport may provide encryption external to the PDF content)

- Compression methods are restricted to a standard list

3080 The PDF/A approach has several advantages over TIFF or JPEG. First, there are more image compressions and format flexibility in PDF, so that the image files sizes can be kept smaller. There are many simple programs available for converting TIFF and JPEG into PDF with various other features for improving compression or adding other information. The PDF/A enables devices that produce vectorized output. Unlike TIFF, JPEG, or BMP, a PDF/A image has the ability to provide several "layers" of information. This allows the creation of PDF searchable images.

3085 A PDF searchable image is a PDF document with an exact bitmapped replica of the scanned paper pages and with text information stored behind the bitmap image of the page. This approach retains the look of the original pages while enabling text searchability and computer analysis. This approach is especially suitable for documents that have to be searchable while retaining the original scan details. The text layer is created by an Optical Character Recognition (OCR) application that scans the text on each page. It then creates a PDF file with the recognized text stored in a layer beneath the image of the text. Unrecognized graphics areas and annotations are preserved with full fidelity in the image. The text form may be incomplete or the OCR confused by some words, but the original image is preserved and available.

3095 Plaintext as well as PDF/A documents shall be base-64 encoded before wrapped in a HL7 CDA R2 header. The PDF/A documents shall conform to PDF/A-1b. Creators are encouraged to conform to PDF/A-1a to the maximum extent possible, but a simple document scanner may be unable to fully conform to PDF/A-1a. Other profiles may require PDF/A-1a conformance.

3100 HL7 CDA R2 header schema is constrained so that pertinent metadata values and scanning facility, technology and operator information shall be present (see IHE ITI TF-3: 5.2.3).

Medical imagery and photographs are outside the scope of this profile. Diagnostic or intervention medical imagery will be supported through DICOM (which includes the use of JPEG and MPEG). Additionally audio and video recorded content is not covered by this profile.

## 5.2.2 XDS Metadata

3105 XDS-SD is a CDA R2 document and thus conforms to the XDS Metadata requirements in the PCC TF-2:4 unless otherwise specified below.

### 5.2.2.1 XDS DocumentEntry Metadata

XDS-SD leverages the XDS DocumentEntry Metadata requirements in the PCC TF-2: 4.1.1 unless otherwise specified below.

#### 3110 5.2.2.1.1 XDSDocumentEntry.formatCode

The XDSDocumentEntry.formatCode shall be **urn:ihe:iti:xds-sd:pdf:2008** when the document is scanned pdf and **urn:ihe:iti:xds-sd:text:2008** when the document is scanned text. The formatCode codeSystem shall be **1.3.6.1.4.1.19376.1.2.3**.

#### **5.2.2.1.2 XDSDocumentEntry.uniqueId**

- 3115 This value shall be the `ClinicalDocument/id` in the HL7 CDA R2 header. The root attribute is required, and the extension attribute is optional. In accordance with the XDS Profile, total length is limited to 256 characters. Additionally see PCC TF-2: 4.1.1, for further content specification.

#### **5.2.2.1.3 Relating instances of XDS-SD documents**

- 3120 In general, most instances of XDS-SD will not have parent documents. It is possible, however, in some specific use cases that instances of XDS-SD documents are related. For example, for a particular document it may be the case that both the PDF scanned content and somewhat equivalent plaintext need to be wrapped and submitted. Each document would correspond to separate `XDSDocumentEntries` linked via an `XFRM Association` that indicates one document is a transform of the other. These can be submitted in a single submission set, or in separate ones.
- 3125 Other specific examples may exist and this profile does not preclude the notion of a parent document for these cases.

#### **5.2.2.2 XDS SubmissionSet Metadata**

- 3130 No additional constraints. Particular to this profile, a legitimate use of submission sets would be to maintain a logical grouping of multiple XDS-SD documents. We encourage such usage. For more information, see PCC TF-2: 4.1.2

#### **5.2.2.3 XDS Folder Metadata**

No additional requirements. For more information, see PCC TF-2: 4.1.3

### 5.2.3 Specification

HL7 CDA R2 header element	CDA as constrained by XDS-SD	Section Number of Extended Discussion	Source Type	Source / Value
ClinicalDocument/typeId	R	5.2.3.1	FM	Fixed, per CDA R2 version in use.
ClinicalDocument/templateId	R	5.2.3.1	FM	Fixed, per this specification
ClinicalDocument/id	R	5.2.3.1	DS	Computable.
ClinicalDocument/code	R	5.2.3.1	O / FM	Entered by operator or appropriately fixed for scanned content
ClinicalDocument/title	R2	5.2.3.1	SA / O	Entered by operator, or possibly can be taken from the scanned content.
ClinicalDocument/confidentiality Code	R	5.2.3.1	O	Assigned by the operator
ClinicalDocument/effectiveTime	R	5.2.3.1	DS	Computed. This is the scan time.
ClinicalDocument/languageCode	R	5.2.3.1	O	Entered by operator
ClinicalDocument/recordTarget	R	5.2.3.2	SA / O	Taken from scanned content, supplemented by operator.
ClinicalDocument/author/assignedAuthor/assignedPerson	R2	5.2.3.3	SA / O	Taken from scanned content, supplemented by operator. This is the original author.
ClinicalDocument/author/assignedAuthor/authoringDevice	R	5.2.3.4	DS / FM / O	Can be computed or fixed based on the scanning device and software. This is the information about the scanning device.
ClinicalDocument/dataEnterer	R	5.2.3.5	DS / O	Can be computed by the scanner or supplemented by operator. This is the information about the scanner operator.
ClinicalDocument/custodian	R	5.2.3.6	DS / FM	Retains original HL7 CDA Context. To be computed or fixed appropriately to denote guardianship of the scanned and wrapped content.
ClinicalDocument/legalAuthenticator	O	5.2.3.7	O	Most likely supplemented by the operator, when applicable or mandated.
ClinicalDocument/documentationOf/serviceEvent/effectiveTime	R	5.2.3.8	SA / O	Denotes the time/date range of the original content.
ClinicalDocument/component/normalizedXMLBody	R	5.2.3.9	SA	The scanned/encoded content.

### 5.2.3.1 ClinicalDocument child-less elements

- 3135 In this section we further discuss `id`, `code`, `effectiveTime`, `confidentialityCode` and `languageCode` elements of the `ClinicalDocument`.
- The `ClinicalDocument/templateId` element shall be present. The root attribute shall contain the oid, '1.3.6.1.4.1.19376.1.2.20', to indicate this document is an XDS-SD document.
- 3140
- The `ClinicalDocument/id` element shall be present. The root attribute shall contain the oid for the document, in which case the extension attribute shall be empty, or an oid that scopes the set of possible unique values for the extension attribute, in which case the extension shall be populated with a globally unique identifier within the scope of the root oid.
- 3145
- The `ClinicalDocument/code` will in most cases be provided by the operator. Values for this code are dictated by the CDA R2 documentation, but are permissible to extend to fit the particular use case. Attributes `code@code` and `code@codeSystem` shall be present.
  - The `ClinicalDocument/title` shall be present if known.
  - The `ClinicalDocument/effectiveTime` shall denote the time at which the original content was scanned. At a minimum, the time shall be precise to the day and shall include the time zone offset from GMT.
- 3150
- The `ClinicalDocument/confidentialityCode` shall be assigned by the operator in accordance with the scanning facility policy. The notion or level of confidentiality in the header may not be the same as that in the Affinity Domain, but in certain cases could be used to derive a confidentiality value among those specified by the Affinity Domain. Attributes `confidentialityCode@code` and `confidentialityCode@codeSystem` shall be present.
- 3155
- The `ClinicalDocument/languageCode`, in accordance with the HL7 CDA R2 documentation, shall denote the language used in the character data of the wrapper CDA header. If the scanned content, when rendered, is in a language different than that of the header, the language context of the CDA will be overwritten at the body level (see IHE ITI TF-3: 5.2.3.9 `ClinicalDocument/component/nonXMLBody` for an example). Attribute `code@code` shall be present. Attribute `code@codeSystem` shall be IETF (Internet Engineering Task Force) RFC 5646 in accordance with the HL7 CDA R2 documentation.
- 3160

3165 Example:

```
<ClinicalDocument xmlns="urn:hl7-org:v3">
  <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
  <templateId root="1.3.6.1.4.1.19376.1.2.20"/>
  <id root="1.3.6.4.1.4.1.2835.2.7777"/>
  <code code="34133-9" codeSystem="2.16.840.1.113883.6.1"
    codeSystemName="LOINC" displayName="SUMMARIZATION OF EPISODE NOTE"/>
  <title>Good Health Clinic Care Record Summary</title>
  <effectiveTime value="20050329224411+0500"/>
  <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
  <languageCode code="en-US"/>
```

### 5.2.3.2 ClinicalDocument/recordTarget

3170 The `ClinicalDocument/recordTarget` contains identifying information about the patient concerned in the original content. In many cases this will have to be supplied by the operator. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- The `ClinicalDocument/recordTarget/patientRole/id` element shall include both the root and the extension attributes. Refer to PCC TF-2: 4.1.1 for more details.
- 3175 • At least one `ClinicalDocument/recordTarget/patientRole/addr` element shall include at least the country subelement. The `addr` element has an unbounded upper limit on occurrences. It can, and should, be replicated to include additional addresses for a patient, each minimally specified by the country sub element.
- 3180 • At least one `ClinicalDocument/recordTarget/patientRole/patient/name` element shall be at least one given subelement and one family subelement.
- The `ClinicalDocument/recordTarget/patientRole/patient/administrativeGenderCode` element shall be present.
- The `ClinicalDocument/recordTarget/patientRole/patient/birthTime` element shall be present with precision to the year.

3185 Example:

```
<recordTarget>
  <patientRole>
    <id extension="12345" root="2.16.840.1.113883.3.933"/>
    <addr>
      <streetAddressLine>17 Daws Rd.</streetAddressLine>
      <city>Blue Bell</city>
      <state>MA</state>
      <postalCode>02368</postalCode>
      <country>USA</country>
    </addr>
    <patient>
      <name>
        <prefix>Mrs.</prefix>
        <given>Ellen</given>
        <family>Ross</family>
      </name>
      <administrativeGenderCode code="F"
        codeSystem="2.16.840.1.113883.5.1"/>
      <birthTime value="19600127"/>
    </patient>
  </patientRole>
</recordTarget>
```

### 5.2.3.3 ClinicalDocument/author (original)

3190 This ClinicalDocument/author element represents the author of the original content. It additionally can encode the original author's institution in the subelement representedOrganization. Information regarding the original author and his/her institution shall be included, if it is known. In many cases this will have to be supplied by the operator. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- 3195 • The ClinicalDocument/author/templateId element shall be present. The root attribute shall contain the oid, '1.3.6.1.4.1.19376.1.2.20.1', to indicate this is the original author.
- The ClinicalDocument/author/time represents the day and time of the authoring of the original content. This value is not restricted beyond statements made in the HL7 CDA R2 documentation.
- 3200 • The ClinicalDocument/author/assignedAuthor/id element if known shall include both the root and the extension attributes. Refer to PCC TF-2: 4.1.1 for more details.
- The ClinicalDocument/author/assignedAuthor/representedOrganization/id element if known shall include both the root and the extension attributes. Refer to PCC TF-2: 4.1.1 for more details.

Example:

```
<author>
  <templateId root="1.3.6.1.4.1.19376.1.2.20.1"/>
  <time value="19990522"/>
  <assignedAuthor>
    <id extension="11111111" root="1.3.5.35.1.4436.7"/>
    <assignedPerson>
      <name>
        <prefix>Dr.</prefix>
        <given>Bernard</given>
        <family>Wiseman</family>
        <suffix>Sr.</suffix>
      </name>
    </assignedPerson>
    <representedOrganization>
      <id extension="aaaaabbbb" root="1.3.5.35.1.4436.7"/>
      <name>Dr. Wiseman's Clinic</name>
    </representedOrganization>
  </assignedAuthor>
</author>
```

3205

#### 5.2.3.4 ClinicalDocument/author (scanner)

This ClinicalDocument/author element shall be present and represent the scanning device and software used to produce the scanned content. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- 3210
  - The ClinicalDocument/author/templateId element shall be present. The root attribute shall contain the oid, '1.3.6.1.4.1.19376.1.2.20.2', to indicate this author is the scanning device and software.
- 3215
  - The ClinicalDocument/author/time shall denote the time at which the original content was scanned. This value shall be equal to that of ClinicalDocument/effectiveTime. At a minimum, the time shall be precise to the day and shall include the time zone offset from GMT.
- 3220
  - The ClinicalDocument/author/assignedAuthor/id element shall be at least the root oid of the scanning device.
  - The ClinicalDocument/author/assignedAuthor/assignedAuthoringDevice/code element shall be present. The values set here are taken from appropriate DICOM vocabulary. The value of code@codeSystem shall be set to "1.2.840.10008.2.16.4". The value of code@code shall be set to "CAPTURE" for PDF scanned content and "WSD" for plaintext. The value of code@displayName shall be set to "Image Capture" for PDF scanned content and "Workstation" for plaintext.
- 3225
  - The ClinicalDocument/author/assignedAuthor/assignedAuthoringDevice/manufacturerModel

- 3230 Name element shall be present. The mixed content shall contain string information that specifies the scanner product name and model number. From this information, features like bit depth and resolution can be inferred. In the case of virtually scanned documents (for example, print to PDF), the manufactureModelName referenced here refers to the makers of the technology that was used to produce the embedded content.
- 3235
- The ClinicalDocument/author/assignedAuthor/assignedAuthoringDevice/softwareName element shall be present. The mixed content shall contain string information that specifies the scanning software name and version. In the case of virtually scanned documents, the softwareName referenced here refers to the technology that was used to produce the embedded content.
  - The ClinicalDocument/author/assignedAuthor/representedOrganization/id element shall be present. The root attribute shall be set to the oid of the scanning facility.

Example:

```
<author>
  <templateId root="1.3.6.1.4.1.19376.1.2.20.2" />
  <time value="20050329224411+0500" />
  <assignedAuthor>
    <id root="1.3.6.4.1.4.1.2835.2.1234" />
    <assignedAuthoringDevice>
      <code code="CAPTURE" displayName="Image Capture" codeSystem="
1.2.840.10008.2.16.4" />
      <manufacturerModelName>SOME SCANNER NAME AND MODEL
      </manufacturerModelName>
      <softwareName>SCAN SOFTWARE NAME v0.0</softwareName>
    </assignedAuthoringDevice>
    <representedOrganization>
      <id root="1.3.6.4.1.4.1.2835.2" />
      <name>SOME Scanning Facility</name>
      <addr>
        <streetAddressLine>21 North Ave</streetAddressLine>
        <city>Burlington</city>
        <state>MA</state>
        <postalCode>01803</postalCode>
        <country>USA</country>
      </addr>
    </representedOrganization>
  </assignedAuthor>
</author>
```

3240

### 5.2.3.5 ClinicalDocument/dataEnterer

This ClinicalDocument/dataEnterer element shall represent the scanner operator who produced the scanned content. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- 3245
- The ClinicalDocument/dataEnterer/templateId element shall be present. The root attribute shall contain the oid, '1.3.6.1.4.1.19376.1.2.20.3', to indicate this is the scanner operator.
  - The ClinicalDocument/dataEnterer/time shall denote the time at which the original content was scanned. This value shall be equal to that of ClinicalDocument/effectiveTime. At a minimum, the time shall be precise to the day and shall include the time zone offset from GMT.
- 3250
- The ClinicalDocument/dataEnterer/assignedEntity/id element shall be both the root and the extension attributes the root shall be the oid of the scanning facility and the extension shall be an appropriately assigned, facility unique id of the operator.

Example:

```
<dataEnterer>
  <templateId root="1.3.6.1.4.1.19376.1.2.20.3" />
  <time value="20050329224411+0500" />
  <assignedEntity>
    <id extension="22222222" root="1.3.6.4.1.4.1.2835.2" />
    <assignedPerson>
      <name>
        <prefix>Mrs.</prefix>
        <given>Bernice</given>
        <family>Smith</family>
      </name>
    </assignedPerson>
  </assignedEntity>
</dataEnterer>
```

3255

### 5.2.3.6 ClinicalDocument/custodian

The ClinicalDocument/custodian shall be present. Its context is left up to the scanning facility to refine in accordance with local policies and to reflect the entity responsible for the scanned content. In most cases this will be the scanning facility. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

3260

- The ClinicalDocument/assignedCustodian/representedOrganization/name shall be present.
- At least one ClinicalDocument/assignedCustodian/representedOrganization/addr element shall include at least the country sub element.

3265 Example:

```
<custodian>
  <assignedCustodian>
    <representedCustodianOrganization>
      <id root="1.3.6.4.1.4.1.2835.2"/>
      <name>SOME Scanning Facility</name>
      <addr>
        <streetAddressLine>21 North Ave</streetAddressLine>
        <city>Burlington</city>
        <state>MA</state>
        <postalCode>01803</postalCode>
        <country>USA</country>
      </addr>
    </representedCustodianOrganization>
  </assignedCustodian>
</custodian>
```

### 5.2.3.7 ClinicalDocument/legalAuthenticator

3270 The ClinicalDocument/legalAuthenticator may be present and its context is left up to the scanning facility to refine in accordance with local policies. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- The ClinicalDocument/legalAuthenticator/assignedEntity/id element if known shall include both the root and the extension attributes. Refer back to PCC TF-2: 4.1.1 for more details.

Example:

```
<legalAuthenticator>
  <time value="19990522"/>
  <signatureCode code="S"/>
  <assignedEntity>
    <id extension="11111111" root="1.3.5.35.1.4436.7"/>
    <assignedPerson>
      <name>
        <prefix>Dr.</prefix>
        <given>Bernard</given>
        <family>Wiseman</family>
        <suffix>Sr.</suffix>
      </name>
    </assignedPerson>
  </assignedEntity>
</legalAuthenticator>
```

3275

### 5.2.3.8 ClinicalDocument/documentationOf

3280 This `ClinicalDocument/documentationOf` element is used to encode the date/time range of the original content. If the original content is representative of a single point in time then the endpoints of the date/time range shall be the same. Information regarding this date/time range shall be included, if it is known. In many cases this will have to be supplied by the operator. This profile does not restrict the `documentationOf` element beyond statements made in the HL7 CDA R2 documentation.

Example:

```
<documentationOf>
  <serviceEvent >
    <effectiveTime>
      <low value="19800127" />
      <high value="19990522" />
    </effectiveTime>
  </serviceEvent>
</documentationOf>
```

### 3285 5.2.3.9 ClinicalDocument/component/nonXMLBody

This `ClinicalDocument/component/nonXMLBody` element shall be present and used to wrap the scanned content. The `nonXMLBody` element is guaranteed to be unique; thus the x-path to recover the scanned content is essentially fixed. All subelements of the `nonXMLBody` retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- 3290 • If the human-readable language of the scanned content is different than that of the wrapper (specified in `ClinicalDocument/languageCode`), then `ClinicalDocument/component/nonXMLBody/languageCode` shall be present. Attribute `code@code` shall be present. Attribute `code@codeSystem` shall be IETF (Internet Engineering Task Force) RFC 5646 in accordance with the HL7 CDA R2 documentation.
- 3295 • The `ClinicalDocument/component/nonXMLBody/text` element shall be present and encoded using `xs:base64Binary` encoding and shall contain the scanned content.
  - `ClinicalDocument/component/nonXMLBody/text@mediaType` shall be “application/pdf” for PDF, or “text/plain” for plaintext.
  - When the character encoding of text/plain content is other than UTF-8, the `@mediaType` shall be `text/plain;charset=XXX`, where XXX is the name of the encoding as specified by RFC-2046.
  - 3300 • `ClinicalDocument/component/nonXMLBody/text@representation` shall be present. The `@representation` for both PDF and plaintext scanned content will be “B64”, because this profile requires the base-64 encoding of both formats.

3305 Example (PDF scanned content is in the *same* language as the wrapper):

```
<component>
  <nonXMLBody>
    <text mediaType="application/pdf" representation="B64">
      JVBERi0xLjMKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0
      ZURlY29kZT4+CnN0cmVhbQp4nGWPmWsdMqYFd/8KjfJwqmVbkr0GQqFbg7fQoSRNWuhB
      Q/4/1L67TEEYme+9J1s3CMQQRm39NLUxg8H17gK89nN1N8eLAbZ2mmHXuql2QDVUhnZx
      a5iBcyQtoMIUM7TZHbH5KZEVDgm//SSUsbwFHx/JzBLEu5yYxOIE8bPcRWqdaGDmcZO
      BWc/9bfUNOPfOte4409jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
      K7CXIe2be+/1DzXQP+RlbnRzdHJlYW0KZW5kb2JqCjYgMCAvYmoKMjAxMmVudG9iag0
      ...
      SW5mbyAyIDAgUgovSUQgWzxnENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
      RjRDQzdBRUIlNEIzNkZCMjgzQzVDMzI0Nz1BRDI4M0Y+XQo+PgpzdGFydHhZWyKMzAx
      Mgo1JUVPRgo=
    </text>
  </nonXMLBody>
</component>
</ClinicalDocument>
```

Example (PDF scanned content is in a *different* language than the wrapper):

```
<component>
  <nonXMLBody>
    <languageCode code="zh-CN" />
    <text mediaType="application/pdf" representation="B64">
      JVBERi0xLjMKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0
      ZURlY29kZT4+CnN0cmVhbQp4nGWPmWsdMqYFd/8KjfJwqmVbkr0GQqFbg7fQoSRNWuhB
      Q/4/1L67TEEYme+9J1s3CMQQRm39NLUxg8H17gK89nN1N8eLAbZ2mmHXuql2QDVUhnZx
      a5iBcyQtoMIUM7TZHbH5KZEVDgm//SSUsbwFHx/JzBLEu5yYxOIE8bPcRWqdaGDmcZO
      BWc/9bfUNOPfOte4409jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
      K7CXIe2be+/1DzXQP+RlbnRzdHJlYW0KZW5kb2JqCjYgMCAvYmoKMjAxMmVudG9iag0
      ...
      SW5mbyAyIDAgUgovSUQgWzxnENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
      RjRDQzdBRUIlNEIzNkZCMjgzQzVDMzI0Nz1BRDI4M0Y+XQo+PgpzdGFydHhZWyKMzAx
      Mgo1JUVPRgo=
    </text>
  </nonXMLBody>
</component>
</ClinicalDocument>
```

## 5.2.4 Complete Example (Wrapped PDF)

3310

```
<ClinicalDocument xmlns="urn:hl7-org:v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" classCode="DOCCLIN"
moodCode="EVN" xsi:schemaLocation="urn:hl7-org:v3 CDA.xsd">
```

3315

```
<typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
<templateId root="1.3.6.1.4.1.19376.1.2.20"/>
```

3320

```
<id root="1.3.6.4.1.4.1.2835.2.7777"/>
<code code="34133-9" codeSystem="2.16.840.1.113883.6.1"
codeSystemName="LOINC" displayName="SUMMARIZATION OF EPISODE NOTE"/>
<title>Good Health Clinic Care Record Summary</title>
<effectiveTime value="20050329224411+0500"/>
<confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
<languageCode code="en-US"/>
```

3325

```
<recordTarget>
<patientRole>
<id extension="12345" root="2.16.840.1.113883.3.933"/>
```

3330

```
<addr>
<streetAddressLine>17 Daws Rd.</streetAddressLine>
<city>Blue Bell</city>
<state>MA</state>
<postalCode>02368</postalCode>
<country>USA</country>
```

3335

```
</addr>
<patient>
<name>
<prefix>Mrs.</prefix>
<given>Ellen</given>
<family>Ross</family>
</name>
<administrativeGenderCode code="F"
codeSystem="2.16.840.1.113883.5.1"/>
```

3340

```
<birthTime value="19600127"/>
</patient>
```

3345

```
</patientRole>
</recordTarget>
<author>
<templateId root="1.3.6.1.4.1.19376.1.2.20.1"/>
```

3350

```
<time value="19990522"/>
<assignedAuthor>
<id extension="11111111" root="1.3.5.35.1.4436.7"/>
```

3355

```
<assignedPerson>
<name>
<prefix>Dr.</prefix>
<given>Bernard</given>
<family>Wiseman</family>
<suffix>Sr.</suffix>
</name>
</assignedPerson>
<representedOrganization>
```

3360

```
<id extension="aaaaabbbbb" root="1.3.5.35.1.4436.7"/>
<name>Dr. Wiseman's Clinic</name>
</representedOrganization>
```

```

    </assignedAuthor>
  </author>
<author>
3365   <templateId root="1.3.6.1.4.1.19376.1.2.20.2" />
      <time value="20050329224411+0500" />
      <assignedAuthor>
        <id root="1.3.6.4.1.4.1.2835.2.1234" />
        <assignedAuthoringDevice>
3370   <code code="CAPTURE" displayName="Image Capture" codeSystem="
1.2.840.10008.2.16.4" />
      <manufacturerModelName>SOME SCANNER NAME AND MODEL </manufacturerModelName>
        <softwareName>SCAN SOFTWARE NAME v0.0</softwareName>
        </assignedAuthoringDevice>
3375   <representedOrganization>
        <id root="1.3.6.4.1.4.1.2835.2" />
        <name>SOME Scanning Facility</name>
          <addr>
            <streetAddressLine>21 North Ave</streetAddressLine>
3380           <city>Burlington</city>
            <state>MA</state>
            <postalCode>01803</postalCode>
            <country>USA</country>
          </addr>
3385   </representedOrganization>
        </assignedAuthor>
      </author>
<dataEnterer>
3390   <templateId root="1.3.6.1.4.1.19376.1.2.20.3" />
      <time value="20050329224411+0500" />
      <assignedEntity>
        <id extension="22222222" root="1.3.6.4.1.4.1.2835.2" />
        <assignedPerson>
3395   <name>
          <prefix>Mrs.</prefix>
          <given>Bernice</given>
          <family>Smith</family>
        </name>
        </assignedPerson>
3400   </assignedEntity>
      </dataEnterer>
      <custodian>
        <assignedCustodian>
          <representedCustodianOrganization>
3405   <id root="1.3.6.4.1.4.1.2835.2" />
          <name>SOME Scanning Facility</name>
            <addr>
              <streetAddressLine>21 North Ave</streetAddressLine>
3410             <city>Burlington</city>
              <state>MA</state>
              <postalCode>01803</postalCode>
              <country>USA</country>
            </addr>
          </representedCustodianOrganization>

```

