

Integrating the Healthcare Enterprise



5

**IHE IT Infrastructure
Technical Framework**

Volume 3

(ITI TF-3)

10

**Cross-Transaction Specifications and
Content Specifications**

15

Revision 13.0 – Final Text

September 9, 2016

20

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

CONTENTS

4	Metadata used in Document Sharing profiles	3
25	4.1 Abstract Metadata Model.....	4
	4.1.1 Metadata Object Types	5
	4.1.2 Association Types.....	7
	4.1.3 Metadata Attributes	8
	4.1.4 Submission Request.....	14
30	4.2 ebRIM Representation	15
	4.2.1 Metadata Object Types	16
	4.2.2 Association Types.....	25
	4.2.3 Metadata Attributes	47
	4.2.4 Success and Error Reporting	100
35	4.2.5 Metadata Vocabulary.....	107
	4.3 Additional Document Sharing Requirements	108
	4.3.1 Submission Metadata Attribute Optionality	108
	4.3.2 Requirements on Query Type Transactions	110
40	5 IHE Content Specifications.....	113
	5.1 Basic Patient Privacy Consents Module	113
	5.1.1 References	113
	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.....	113
45	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	117
	5.2 Scanned Documents Content Module.....	118
	5.2.1 Referenced Standards	118
	5.2.2 XDS Metadata	119
	5.2.3 Specification	121
50	5.2.4 Complete Example (Wrapped PDF).....	132
	5.3 Intentionally Left Blank	135
	5.4 XDW Workflow Content Module	136
	5.4.1 Referenced Standards	136
	5.4.2 Discussion of Content Standards	136
55	5.4.3 Content Specification	143
	5.4.4 Complete Example	156
	5.4.5 Workflow Document Management	159
	5.4.6 XDS Metadata	164

60 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

65 (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 include:

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

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.

75 Document Sharing profiles manage two aspects of the documents being shared, the metadata about the document and the contents of the document.

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. 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.

80 A Document is a byte sequence managed by actors in one or more IHE Document Sharing profiles. There is no concept of equivalent Documents. If the byte sequence is different, then the Documents are different.

85 Note that this definition of a Document is stricter than document definitions in DICOM^{®1}, HL7, HL7 CDA^{®2}, XML, and elsewhere. Those standards have concepts like "insignificant whitespace". Documents with different byte sequences, that those standards consider equivalent, will be treated as different Documents by IHE Document Sharing.

90 This stricter definition of a Document enables the Document Sharing actors to manage any document format, without the need to include a parser that understands the details of all the

¹ DICOM is the registered trademark of the National Electrical Manufacturers Association for its standards publications relating to digital communications of medical information.

² HL7 and CDA are registered trademarks of Health Level Seven International.

different document formats. It also enables most document indexing, registration, etc. to be done using metadata alone, without examining the document contents.

95 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).

Content Profiles may impose restrictions on metadata attributes. See Section 5 for the restrictions imposed by ITI Content Profiles.

4.1 Abstract Metadata Model

100 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:

- SubmissionSet – metadata describing a collection of Folders, DocumentEntries, and Associations submitted together.
- Folder – metadata describing a collection of related DocumentEntries.
- 105 • DocumentEntry – metadata describing a Document.

The two Association types are:

- HasMember – represents membership of an object in a collection. The four variations of HasMember are described in Section 4.1.2.
- 110 • Relationship – represents a relationship (such as a transform) between two Documents (represented by DocumentEntries). The Relationship associations are described in Section 4.1.2.

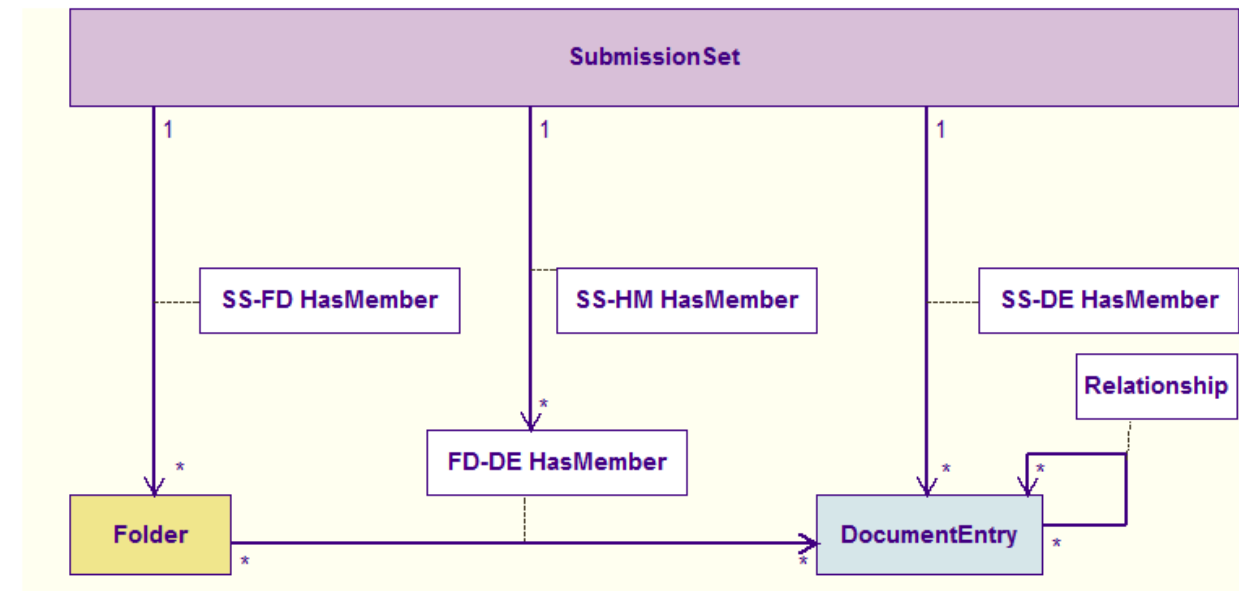


Figure 4.1-1: Document Sharing Objects and Associations

115 **4.1.1 Metadata Object Types**

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

- SubmissionSet
- Folder
- DocumentEntry

120

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.

125

130 SubmissionsSets, once submitted, are a permanent record of the collection of submitted content. 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.

135

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

140 **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 multiple 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.

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

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.

150 **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.

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

155 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 Registry in lieu of submitting the document creates a separation of concerns, allowing the
160 Registry to specialize in indexing, while the Repository manages document storage.

There are two DocumentEntry Types: Stable DocumentEntry and On-Demand Document Entry. The following sections describe these types in detail.

- **Stable DocumentEntry**

165 A Stable Document Entry contains metadata about an already created document available for retrieval. Each Stable DocumentEntry represents a single document. This document is 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 Section 4.2.5.2 for the UUID) and availabilityStatus equal to Approved or Deprecated. All metadata
170 fields contain valid values.

The uniqueID metadata attribute of a Stable DocumentEntry identifies the specific document associated with the entry. It is used in a retrieve request to identify which specific document should be retrieved.

175 If the document returned on a retrieve request is CDA, it will have in the ClinicalDocument/id field in the HL7 CDA R2 document header the same value as the value of the DocumentEntry uniqueId.

- **On-Demand Document Entry**

180 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 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 DocumentEntry has objectType equal to the UUID for On-Demand (see Section 4.2.5.2 for the UUID) and availabilityStatus equal to Approved or Deprecated. An On-Demand Document Entry may be replaced and 185 deprecated. If an On-Demand Document Entry is deprecated, the retrieval of that uniqueID may not have the most recent information and should return an error.

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.

190 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 Document Entry uniqueID is valid for as long as the entry has availabilityStatus equal to 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.

195 When a retrieve request is received specifying an On-Demand Document Entry 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 200 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 metadata about the content.

Unless otherwise specified in a transaction, On-Demand DocumentEntry objects shall not be included in the transaction.

205 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.

There are two types of Associations: HasMember and Relationship.

210 • **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.

- 215 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.
- 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).
- 220 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.
- 225 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.
- 230
- **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.
 - Transform - indicates the transformation of a previous document into a new document.
 - 235 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.
 - Signs – indicates a new document is a signature for a previous document, as in new document signs previous document.

240 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 Section 4.2.3.

4.1.3.1 The Purpose of Metadata Attributes (Informative)

245 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.

- **Patient Identity** – Attributes that describe the subject of the document. This includes patient Id, patient name, and other demographics.

- 250 • **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.
- 255 • **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.
- 260 • **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. 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.
- 265
- 270 • **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.
- 275 • **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.

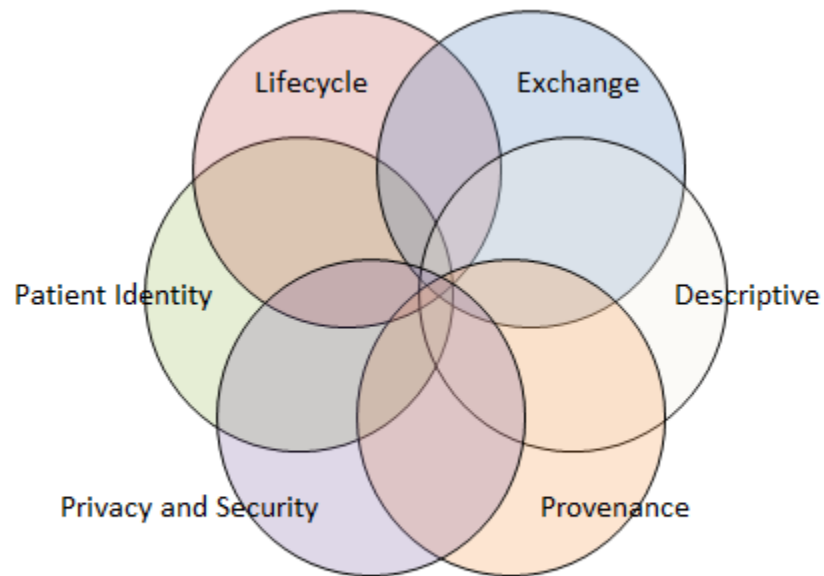


Figure 4.1.3.1-1: Pictorial of Overlapping Document Sharing Metadata Purpose

280 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
285 be consistently assigned and processed.

4.1.3.2 DocumentEntry Metadata Attributes

290 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 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	Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
author	The humans and/or machines that authored the document. This attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty and authorTelecommunication.		X	X	X		X
availabilityStatus	The lifecycle status of the DocumentEntry			X		X	
classCode	The code specifying the high-level use classification of the document type (e.g., Report, Summary, Images, Treatment Plan, Patient Preferences, Workflow).			X	X		
comments	Comments associated with the document.				X		
confidentialityCode	The code specifying the level of confidentiality of the document.			X			
creationTime	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	X		
formatCode	The code specifying the detailed technical format of the document.				X		X
hash	The hash of the contents of the document.			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	Represents a participant within an authorInstitution who has legally authenticated or attested the document.		X	X		X	

DocumentEntry Metadata Attribute	Description	Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
limitedMetadata	Indicates whether the DocumentEntry was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source.		X		X	X	X
contentType	MIME type of the document.				X		X
objectType	The type of DocumentEntry (e.g., On-Demand 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		
referenceIdList	A list of identifiers related to the document		X	X		X	
repositoryUniqueId	The globally unique identifier of the repository where the document can be accessed.			X			X
serviceStartTime	The start time of the service being documented.			X	X		
serviceStopTime	The stop time of the service being documented.			X	X		
size	Size in bytes of the document.			X			X
sourcePatientId	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.	X	X	X			
sourcePatientInfo	This attribute contains demographic information of the source patient to whose medical record this document belongs.	X	X				
title	The title of the document.				X		
typeCode	The code specifying the precise type of document from the user perspective (e.g., LOINC code).				X		
uniqueId	Globally unique identifier assigned to the document by its creator.			X			X
URI	The URI for the document.						X

295

4.1.3.3 SubmissionSet Metadata Attributes

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
author	The humans and/or machines that authored the SubmissionSet. 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 the associated content in the 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 organizations or persons for whom the SubmissionSet is intended.			X			X
limitedMetadata	A flag that the associated SubmissionSet was created using the less rigorous metadata requirements as defined for the Metadata-Limited Document Source.		X		X	X	X
patientId	The patientId represents the primary subject of care of the SubmissionSet.	X		X	X		
sourceId	Identifier of the entity that contributed the SubmissionSet.		X	X	X		
submissionTime	Point in time at the creating entity when the SubmissionSet was created.			X	X	X	
title	The title of the SubmissionSet.				X		
uniqueId	Globally unique identifier for the SubmissionSet assigned by the creating entity.			X			X

4.1.3.4 Folder Metadata Attributes

305 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.

310 **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	X	
codeList	The set of codes specifying the type of clinical activities that resulted in placing DocumentEntry objects in the 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		X	
limitedMetadata	A flag that the associated Folder was created using the less rigorous metadata requirements as defined for the Metadata-Limited Document Source.		X		X	X	X
patientId	The patientId represents the primary subject of care of the Folder.	X		X	X		
title	The title of the Folder.				X		
uniqueId	Globally unique identifier for the Folder.			X			X

4.1.4 Submission Request

A Submission Request is a collection of metadata transferred between one Document Sharing Actor and another. A Submission Request shall contain:

- One SubmissionSet
- 315 • At least one of the following:
 - One or more DocumentEntry objects
 - One or more Folders
 - One or more FD-DE HasMember Associations
- Zero or more Relationship Associations

- 320
- For each object above (except the SubmissionSet), one HasMember Association linking the SubmissionSet to that object

Individual transactions or profiles may impose restrictions on the content.

4.2 ebRIM Representation

325 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>.

330 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.
- 335

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.

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

345

350 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 excludes the requirements found in ebRIM 3.0 Section 2.5.9 which state that "each RegistryObject instance MUST have a life cycle status indicator." For some RegistryObjects the life cycle status indicator is required by IHE, and this requirement is stated within IHE's description of use of the object. For other objects, where a requirement is not explicitly stated by IHE, the life cycle status indicator is optional.

355

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.

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.

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.

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.

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.

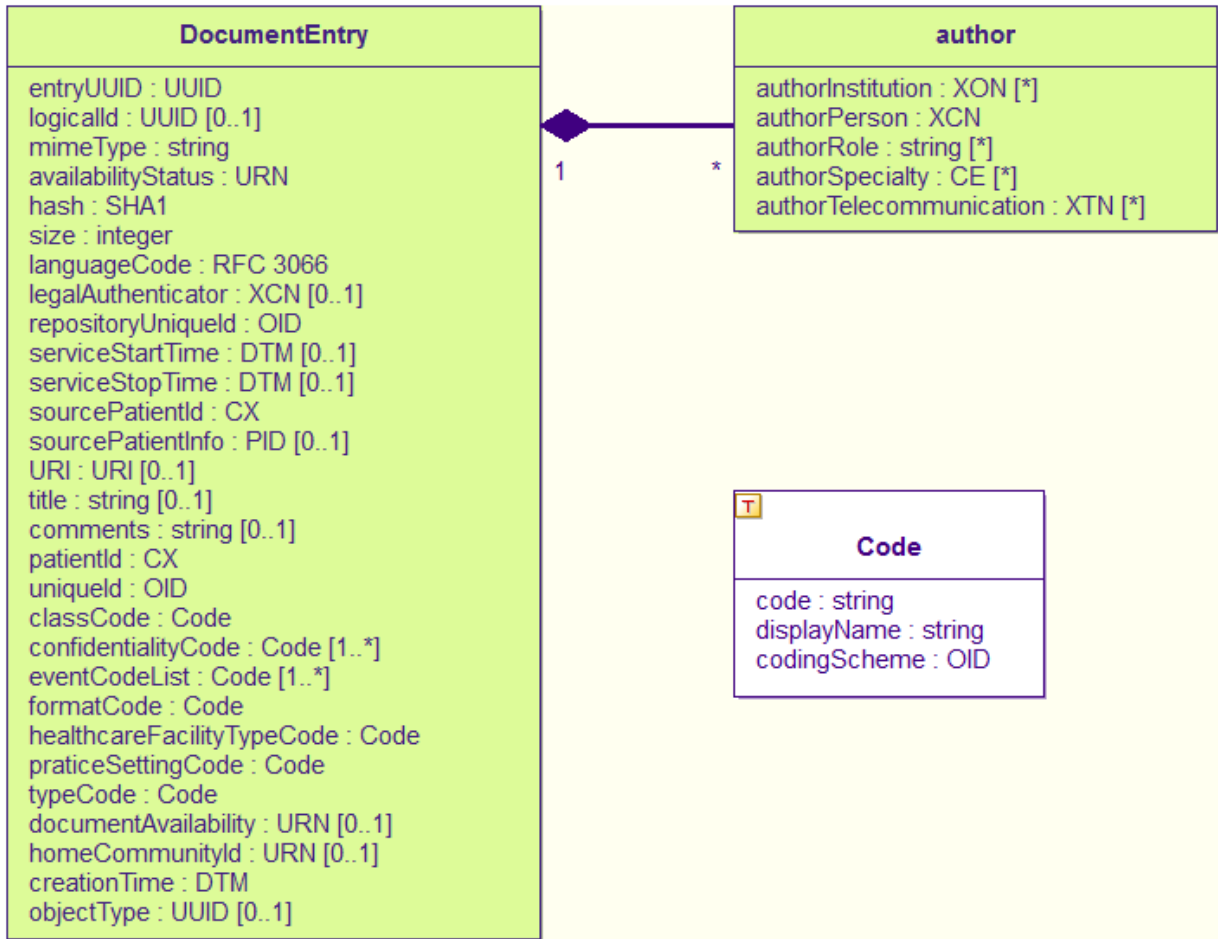


Figure 4.2.1.1-1: DocumentEntry Metadata Attributes (Informative)

380

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

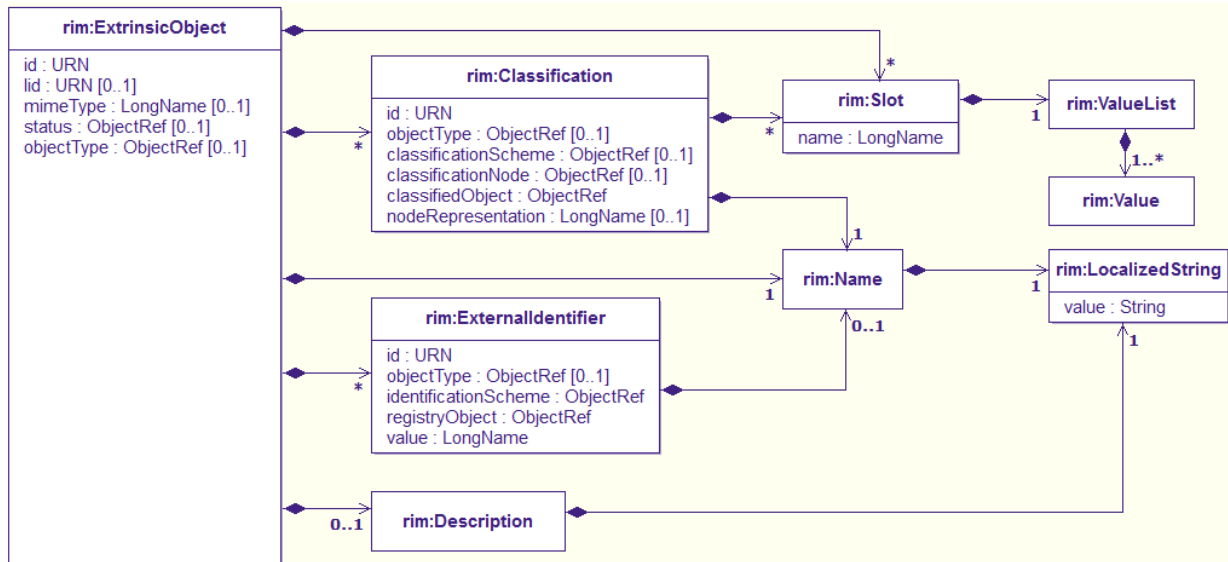


Figure 4.2.1.1-2: UML diagram for rim:ExtrinsicObject (Informative)

385

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.

390

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, Description and ExternalIdentifier), and holds the healthcare specific attributes in general purpose Slots or Classifications.

395

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

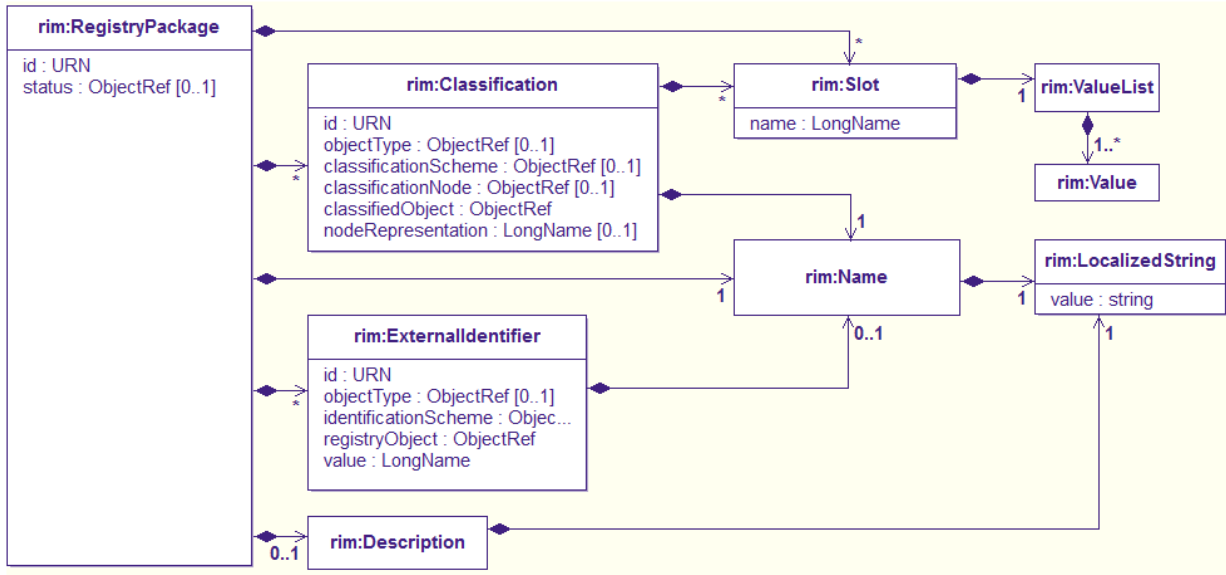
As described in Section 4.1.1, there are two DocumentEntry types: Stable Document Entry and On-Demand Document Entry. A Stable Document Entry is an XDSDocument Entry with objectType equal to the UUID for Stable (see Section 4.2.5.2 for the UUID). An On-Demand DocumentEntry has an objectType equal to 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 uniqueId attribute.

4.2.1.2 SubmissionSet

405

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.



410

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

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.

415

A SubmissionSet has a set of attributes that are described in Section 4.1.3.3 SubmissionSet Metadata.

SubmissionSets exist for two reasons:

420

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

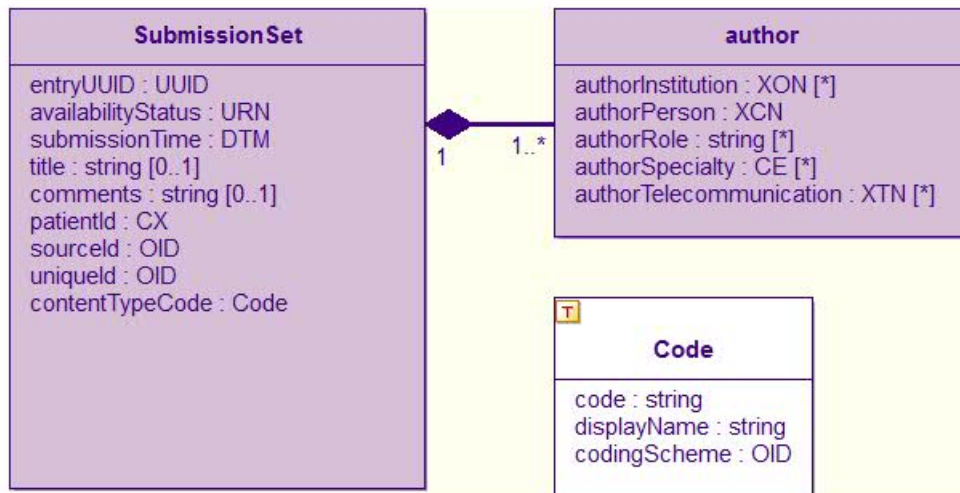


Figure 4.2.1.2-2: SubmissionSet Metadata Attributes (Informative)

425

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.

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.

430

Immutability of Submission Sets: Submission Sets are immutable; that is, after they are submitted, they can never be changed. No associations to or from the submission set can be created later on.

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.

440

Classification encoded inside the RegistryPackage object

445

```

<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-
  regrep:ObjectType:RegistryObject:Classification"
  />
</RegistryPackage>
    
```

450 **Classification encoded outside the RegistryPackage object**

```

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

```

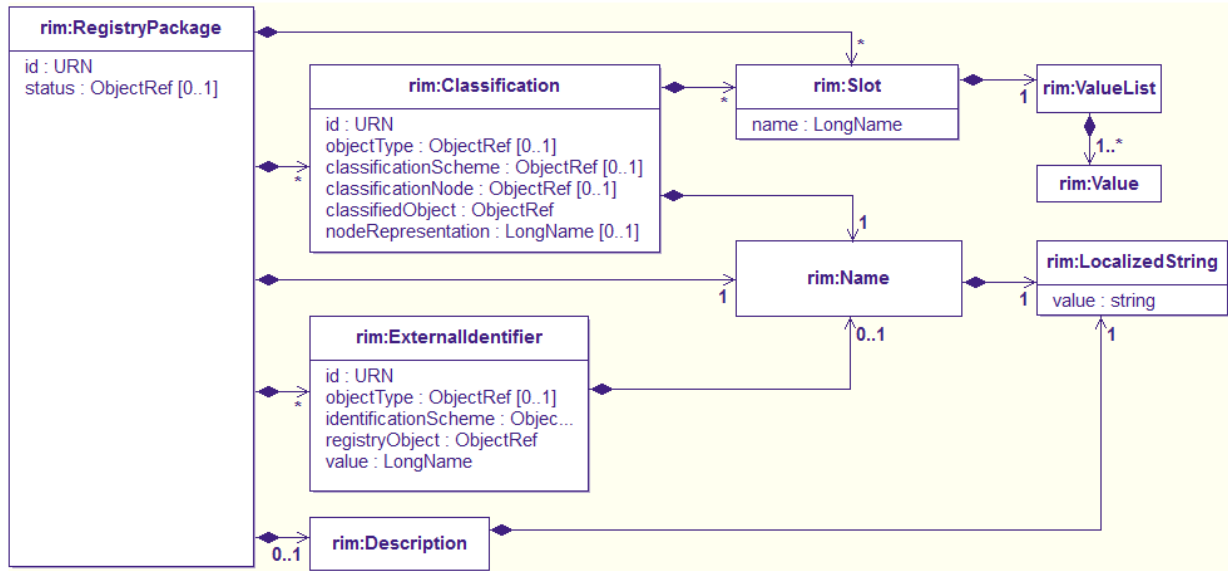
460

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".

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.



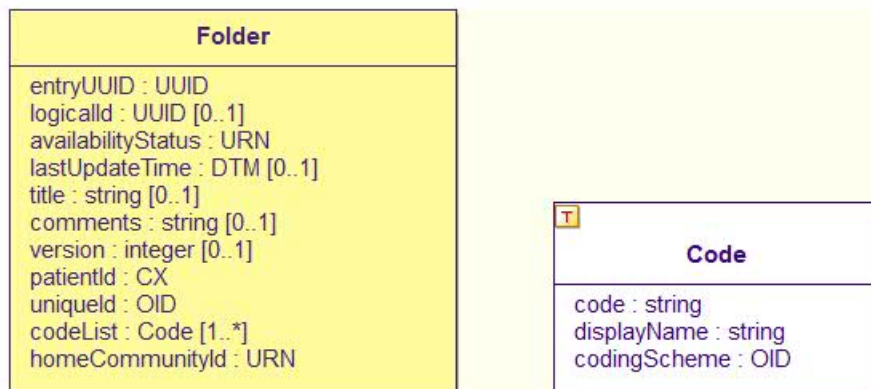
470

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

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.



480

Figure 4.2.1.3-2: Folder Metadata Attributes (Informative)

4.2.1.3.1 Creating a Folder object from a RegistryPackage element

485 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

```

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

Classification encoded outside the RegistryPackage object

```

505 <RegistryPackage id="Fol">
...
</RegistryPackage>
<!-- Classify registry package as Folder -->
<Classification
510   classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2"
   classifiedObject="Fol"
   id="IdExample_066"
   objectType="urn:oasis:names:tc:ebXML-
   regrep:ObjectType:RegistryObject:Classification"
   />
515 <...>
    
```

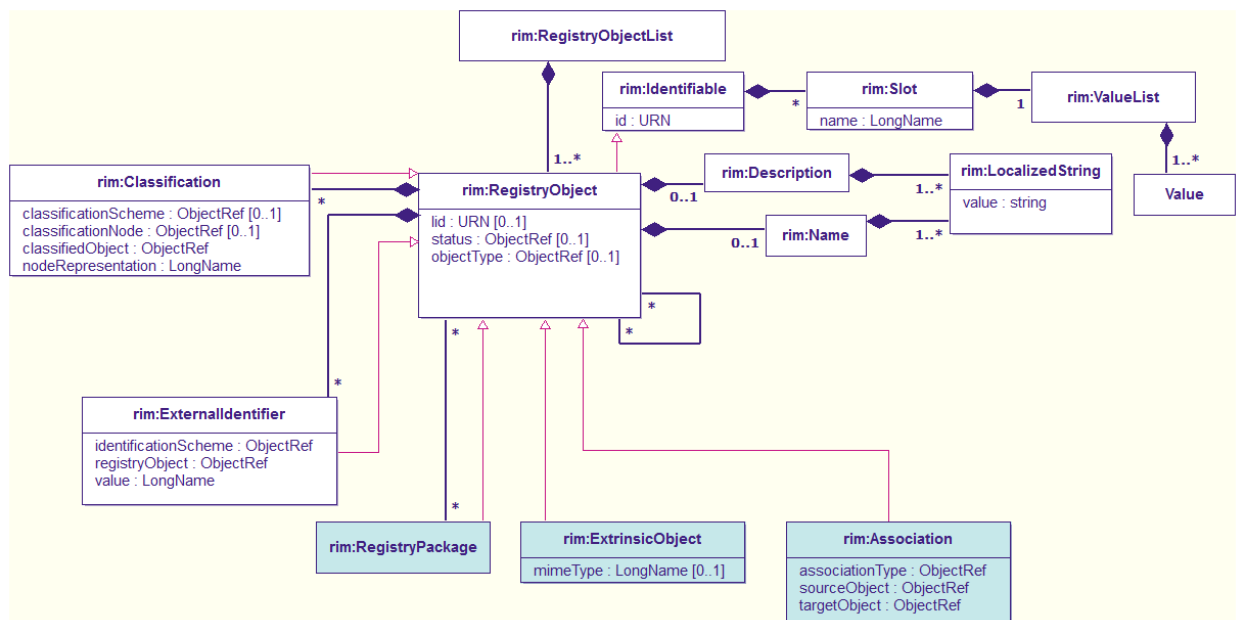
Item	Description
Classification/@classifiedObject	The @id attribute of the RegistryPackage being classified.
Classification/@classificationNode	A fixed value identifying the type of object the RegistryPackage represents. Accepted values: Folder: urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2
Classification/@id	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

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

Figure 4.2.1.4-1 shows in detail the content of the rim:RegistryObjectList used to exchange Document Sharing metadata; a subset of the ebXML Registry Information Model (ebRIM).



525

Figure 4.2.1.4-1: Registry Object List (Informative)

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

530

```

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

535

4.2.1.5 Submission Request

See Section 4.1.4 for a definition of Submission Request. The Submission Request is implemented as an < lcm:SubmitObjectsRequest >.

The <lcm:SubmitObjectsRequest> element shall contain one <rim:RegistryObjectList> element.

540 The <rim:RegistryObjectList> element shall contain:

- One rim:RegistryPackage for the Submission Set
- One rim:ExtrinsicObject for each DocumentEntry in the Submission Request
- One rim:Association for each Association in the Submission Request
- One rim:RegistryPackage for each Folder in the Submission Request

545 The rim:RegistryObjectList may also contain rim:Classification elements that identify the RegistryPackage elements as a SubmissionSet or Folder. See Sections 4.2.1.2.1 and 4.2.1.3.1.

A full XML Schema Document for the XDS types is available online on the IHE FTP site, see ITI TF-2x: Appendix W.

550 Any id attributes or references to other objects in the submission may contain valid UUIDs or symbolic ids (see Section 4.2.3.1.5). References to objects that are not contained in the submission request must be in UUID format.

4.2.2 Association Types

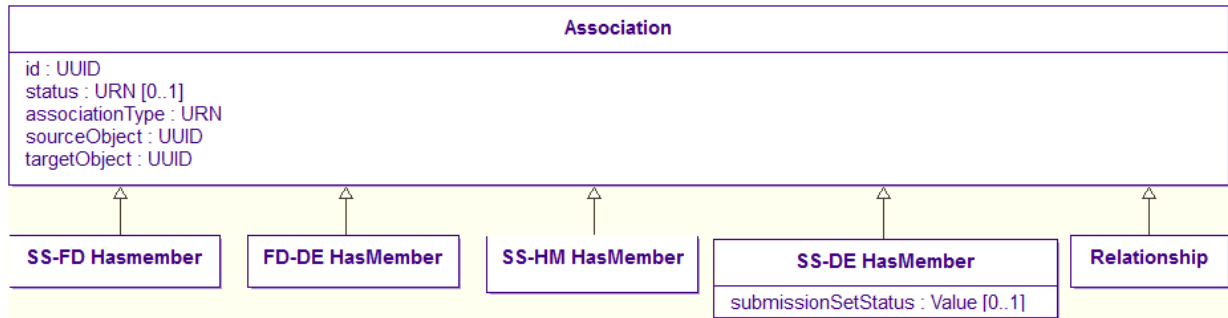
555 All relationships between metadata objects are handled through Associations. An Association is an object that describes a named relationship between two metadata objects. These relationships are used to form submissions and express some query responses. The relationship between the DocumentEntry 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:

- A SubmissionSet and a DocumentEntry – SS-DE HasMember
- 560 • 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

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

565 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.



570

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.

575

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

580

- sourceObject
- targetObject
- associationType

These attributes can be thought to make a small sentence:

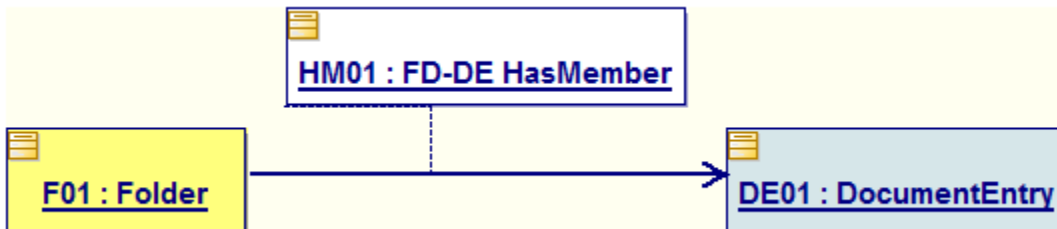
- sourceObject AssociationType targetObject

585

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

- Folder HasMember DocumentEntry

Graphically this example Association looks like:



590

Figure 4.2.2-2: Folder HasMember DocumentEntry (Informative)

Association Type formatting

An Association type shall be specified as a URN.

595 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.

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

600

Example of basic Association

605

```
<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" />
```

4.2.2.1 HasMember

610 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 - SubmissionSetHasMemberDocumentEntry. In the underlying eBRIM model, all of these relationships are created using the eBRIM HasMember Association type.

615 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

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.

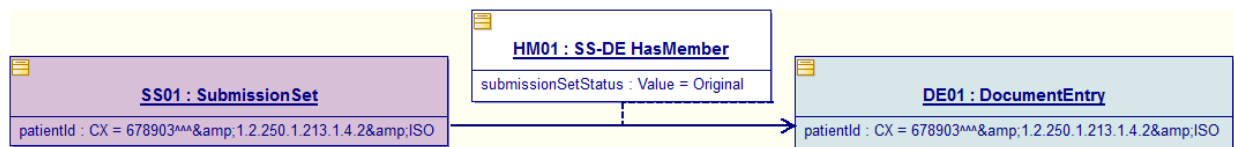
SubmissionSet Association labeling

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

Submission of an original Document (inclusion by value)

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**.

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.



635

Figure 4.2.2.1.1-1: SubmissionSet HasMember DocumentEntry (Informative)

When submitting an existing document by value:

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

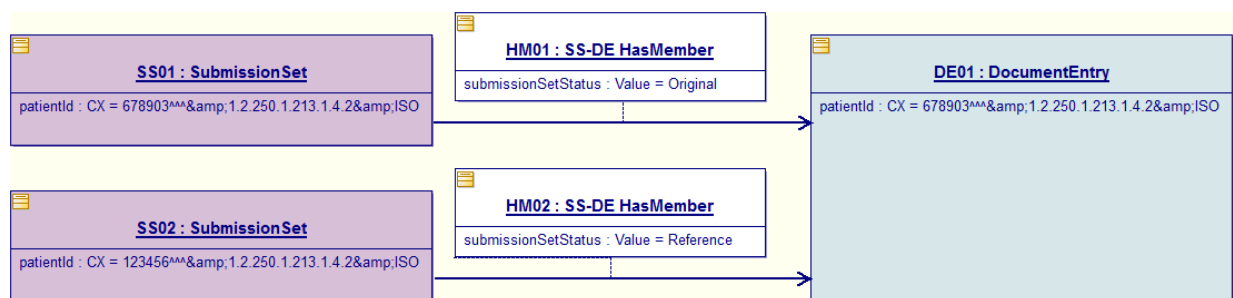
The following XML example demonstrates how to encode a submission by value.

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

655 **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.



670 **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.

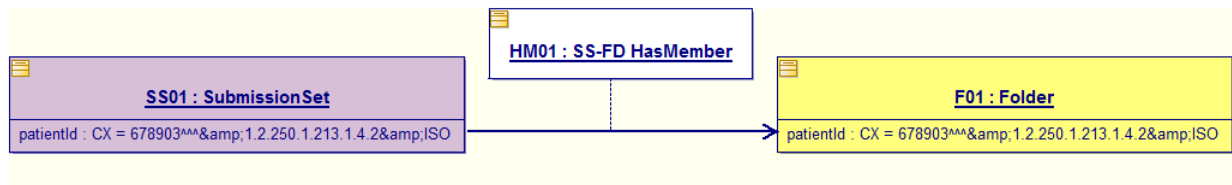
The following XML example demonstrates how to encode a submission by reference.

675

```
680 <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>
685 </rim:Slot>
    </rim:Association>
```

4.2.2.1.2 SS-FD HasMember

690 **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.



695 **Figure 4.2.2.1.2-1: SubmissionSet HasMember Folder (Informative)**

When submitting a Folder:

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

700

Example SubmissionSet – Folder HasMember Association

```
705 <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.

710 **4.2.2.1.3 FD-DE HasMember**

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.

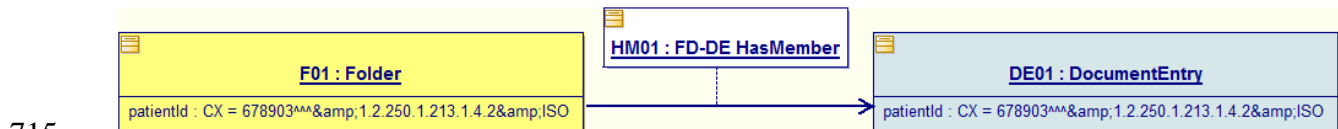


Figure 4.2.2.1.3-1: Folder HasMember DocumentEntry (Informative)

Linking documents to a folder

720 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:

- 725
- The targetObject shall contain the Id of the DocumentEntry object.
 - The sourceObject shall contain the Id of the Folder object.

Example Folder HasMember Association

```
730 <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" />
735 <Association
    associationType="urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember"
    sourceObject="SubmissionSet"
    targetObject="FolderToDocAssoc" />
```

740 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.

745 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

750 **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 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.

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:

- 755
- 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.

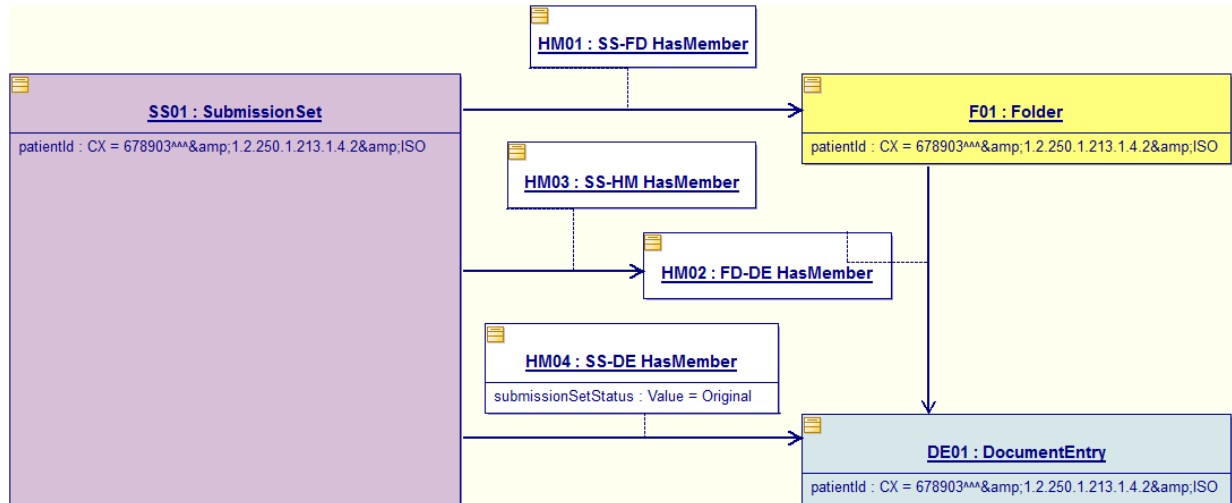
4.2.2.1.5 Adding DocumentEntries to Folders

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

- 760
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.
 3. The DocumentEntry is already present. The new submission includes the Folder and the Association to make the DocumentEntry part of the Folder.
 - 765 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.



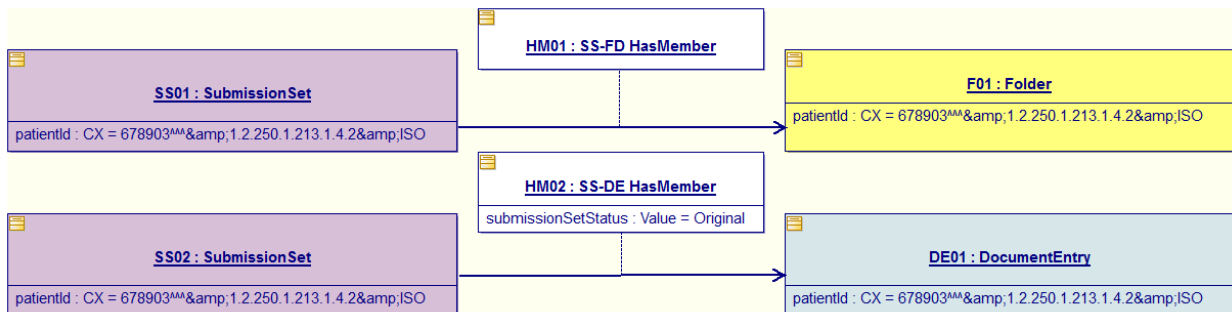
770

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

775

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.



780

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

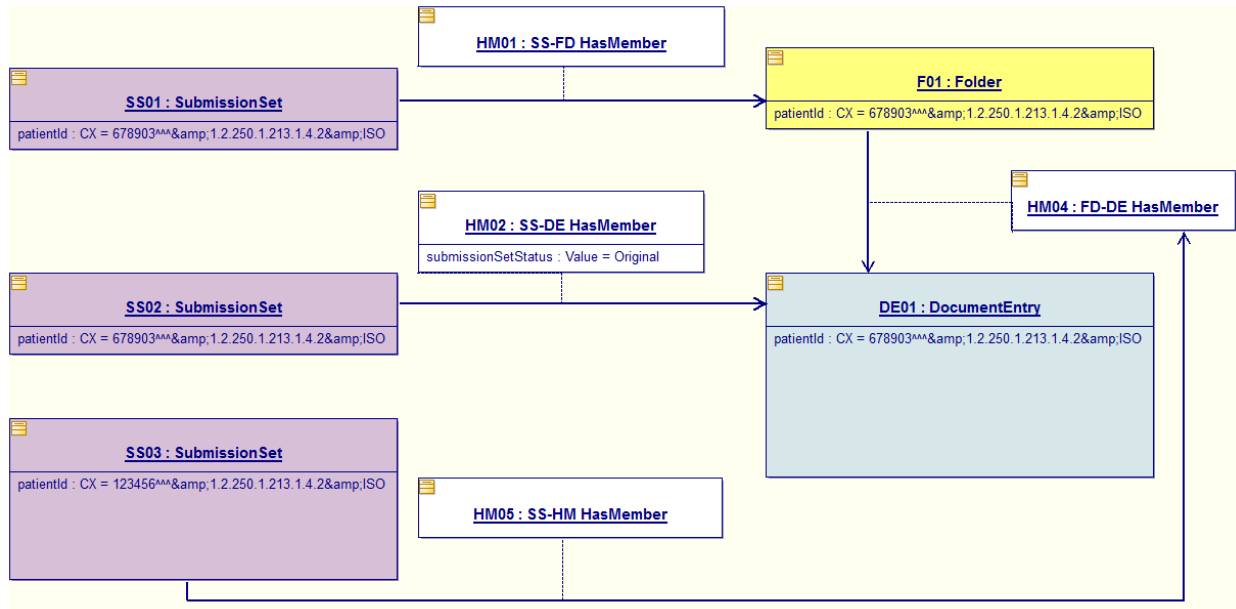
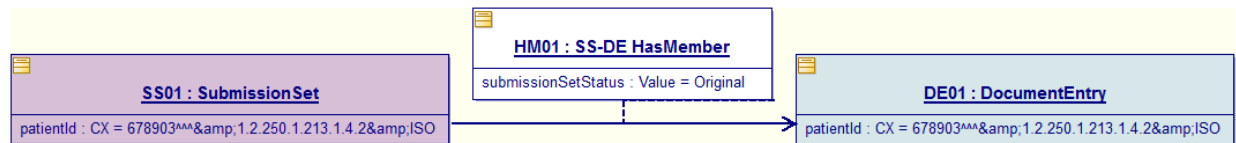


Figure 4.2.2.1.5-3: Scenario 2 - Add existing DocumentEntry to existing Folder (Informative)

785 **Scenario 3 – Folder submitted and existing DocumentEntry added to it**

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.



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

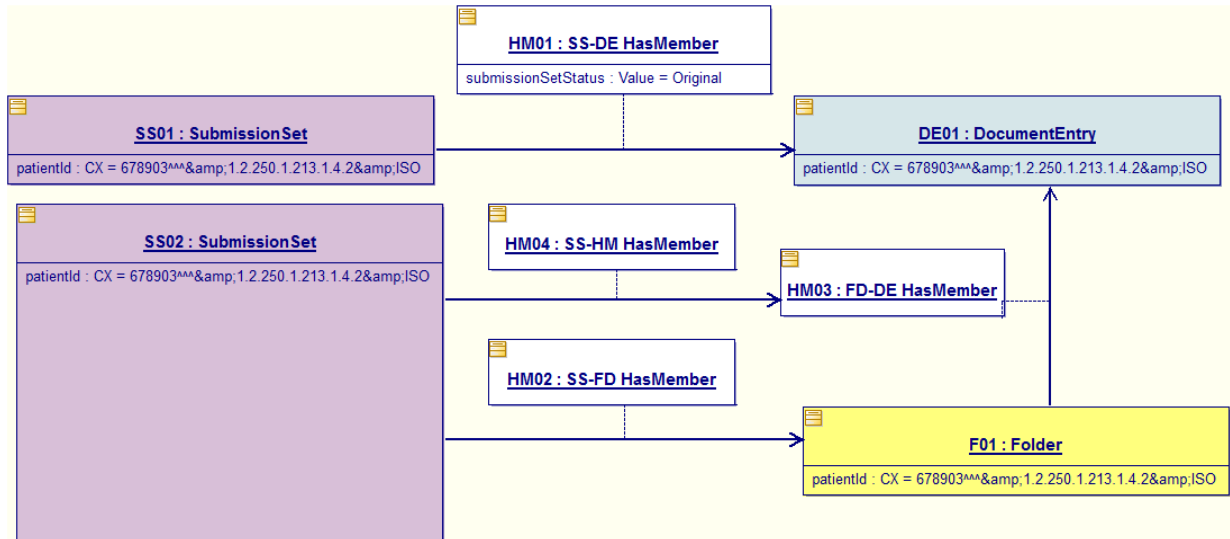
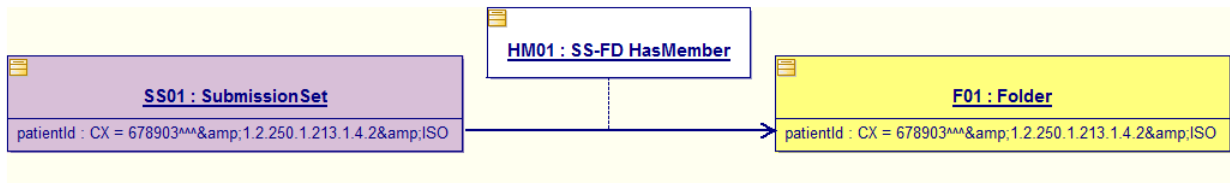


Figure 4.2.2.1.5-5: Scenario 3 - Folder submitted and existing DocumentEntry added to it (Informative)

795 **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.



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

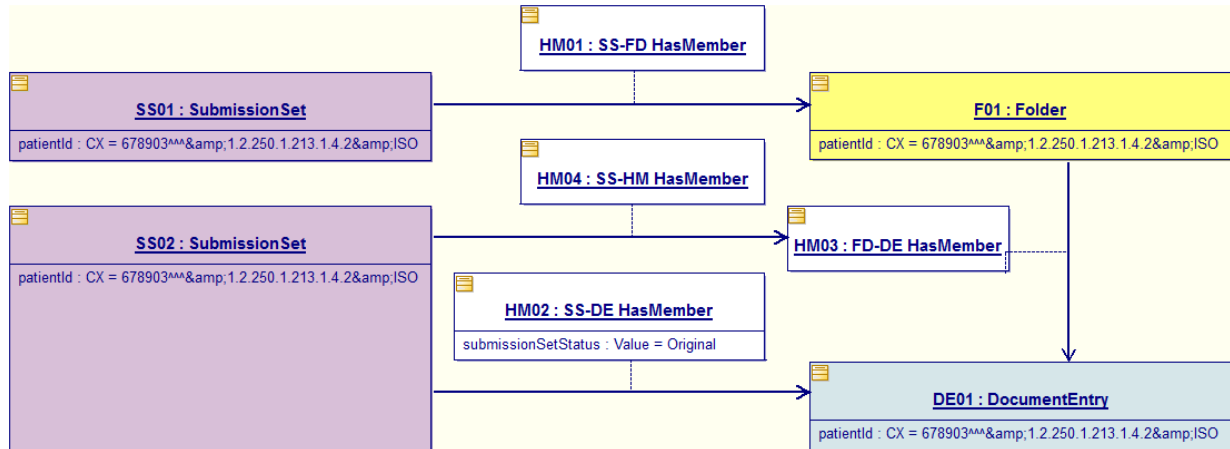


Figure 4.2.2.1.5-7: Scenario 4 - DocumentEntry submitted and added to existing Folder (Informative)

805 **4.2.2.2 Document Relationship**

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.

810 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.

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

815 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.

820 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 known because the creating entity assigned the entryUUID prior to submission or discovered it (e.g., under XDS via Registry Stored Query [ITI-18]).

825 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.

830 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

The uniqueId attribute shall be unique

835 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.

The targetObject attribute of the Association object references the existing DocumentEntry known to the receiving entity.

840 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.

845 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:

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

880 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).

885 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 ITI TF-2x: Appendix B "Definition of uniqueIds".

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

4.2.2.2.1 APND

890 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).

895

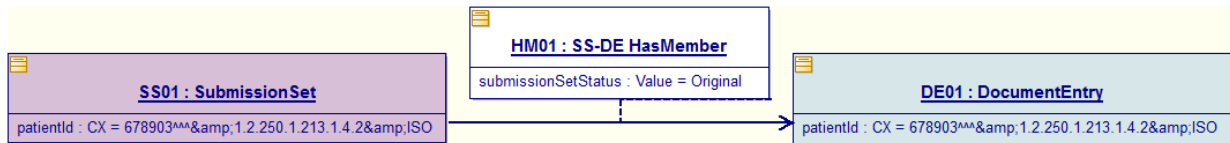


Figure 4.2.2.2.1-1: Starting Point (Informative)

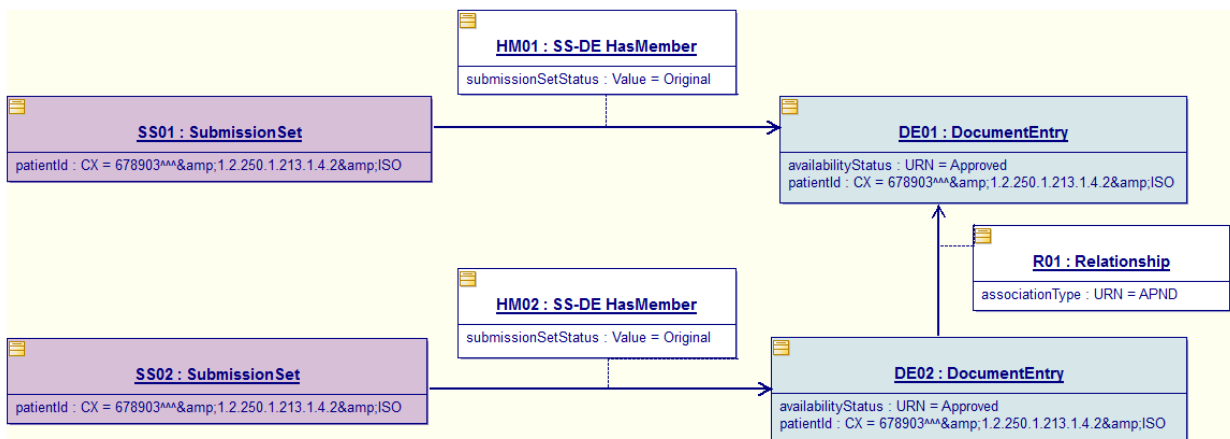


Figure 4.2.2.2.1-2: New DocumentEntry and Document forming an addendum to an existing DocumentEntry/Document (Informative)

900

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

905

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).

910

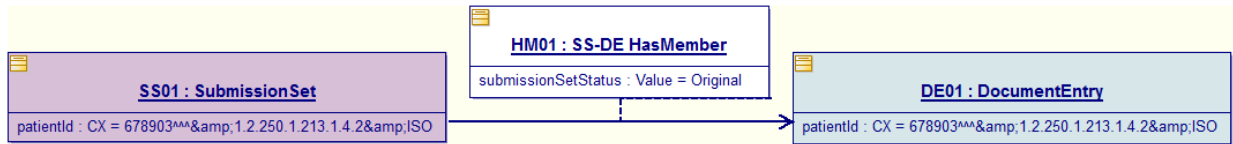
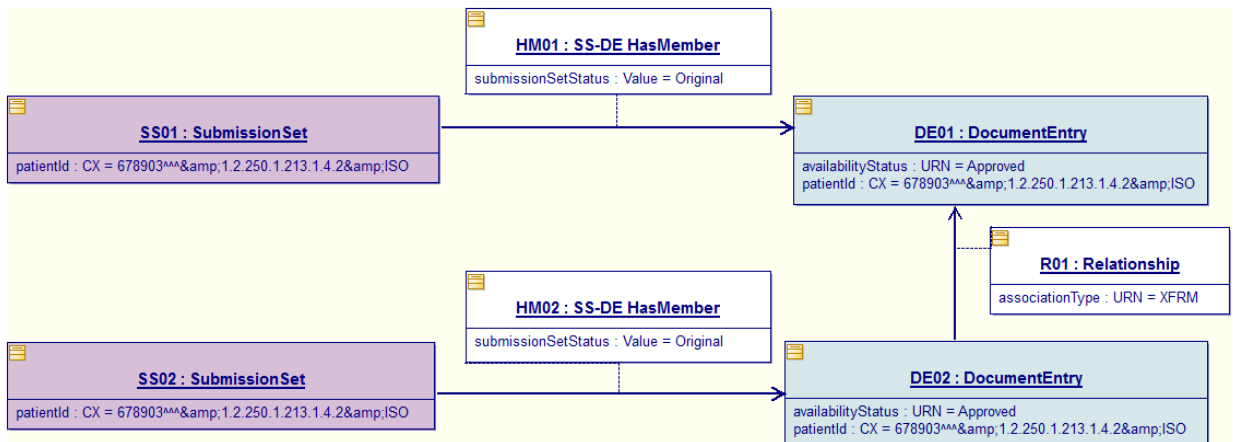


Figure 4.2.2.2-1: Starting Point (Informative)



915

Figure 4.2.2.2-2: New DocumentEntry/Document defining a transformation of an existing DocumentEntry/Document (Informative)

4.2.2.2.3 RPLC

920

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.).

925

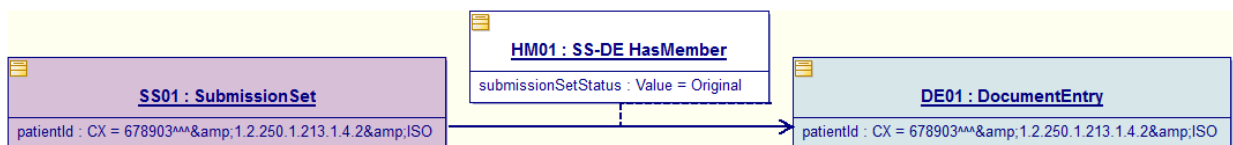
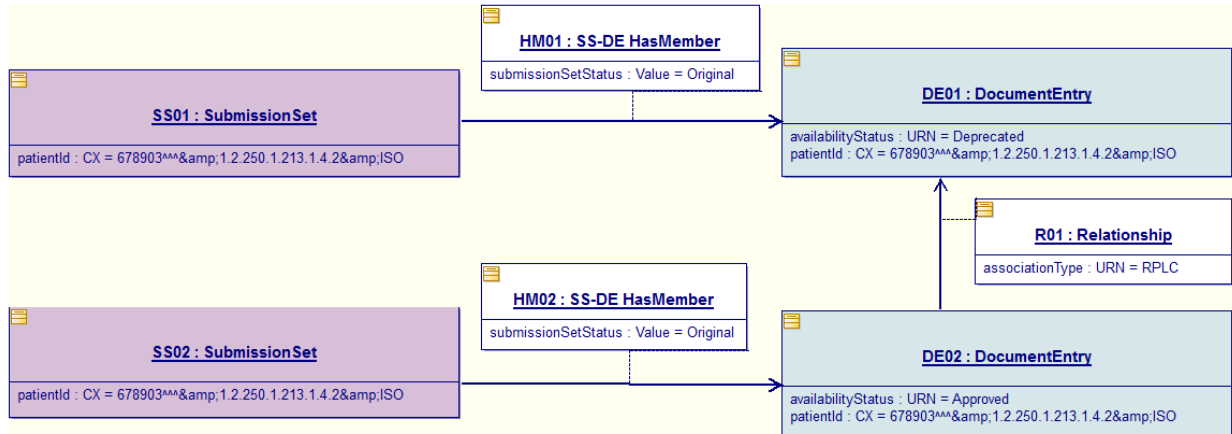


Figure 4.2.2.2.3-1: Starting Point (Informative)



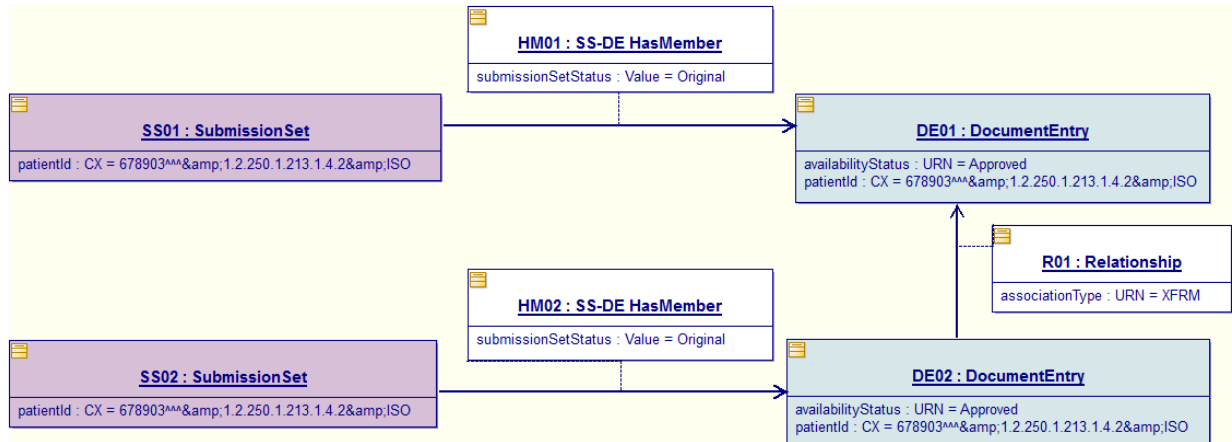
930

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

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.

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:



940

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

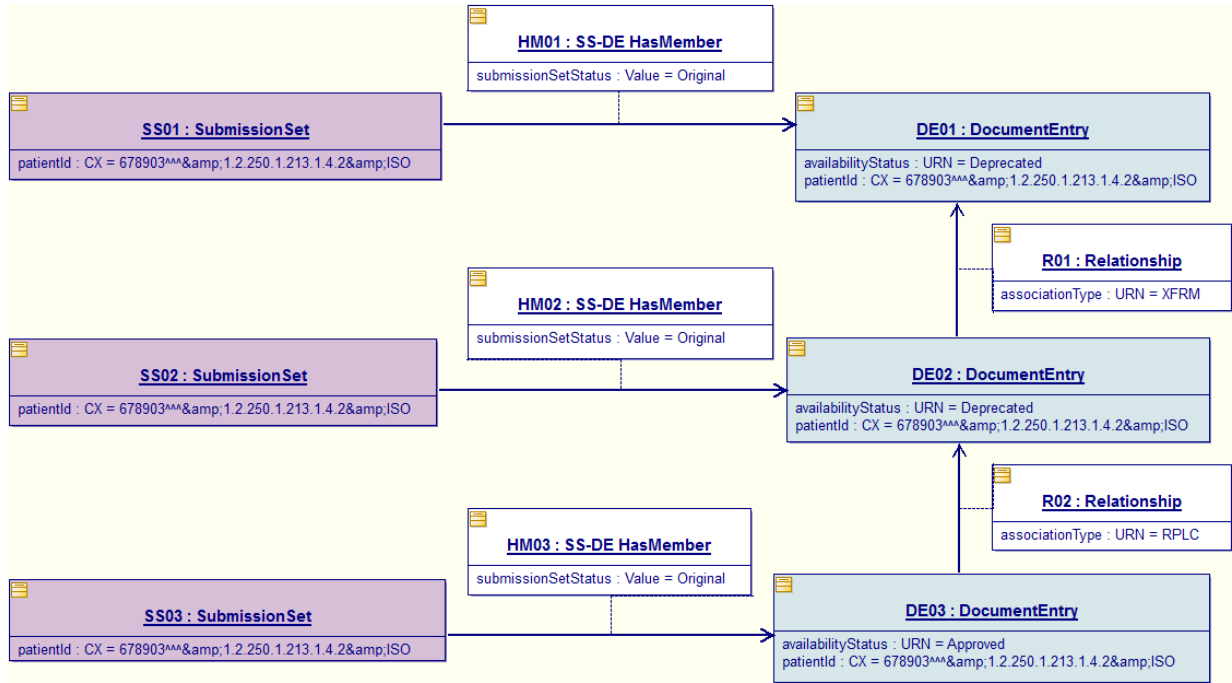
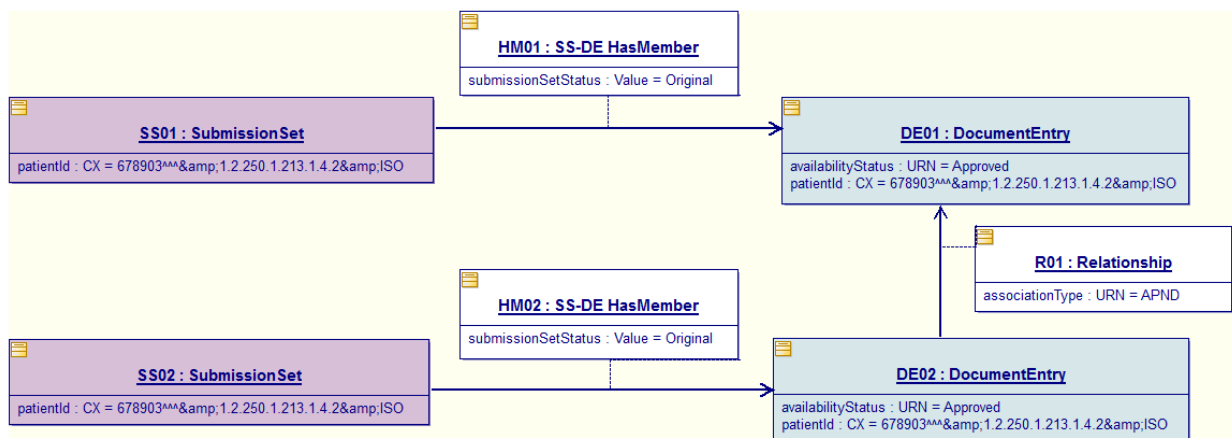


Figure 4.2.2.2.3-4: New DocumentEntry/Document replacing a transformation (XFRM) of an existing DocumentEntry/Document (Informative)

945

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.



950

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

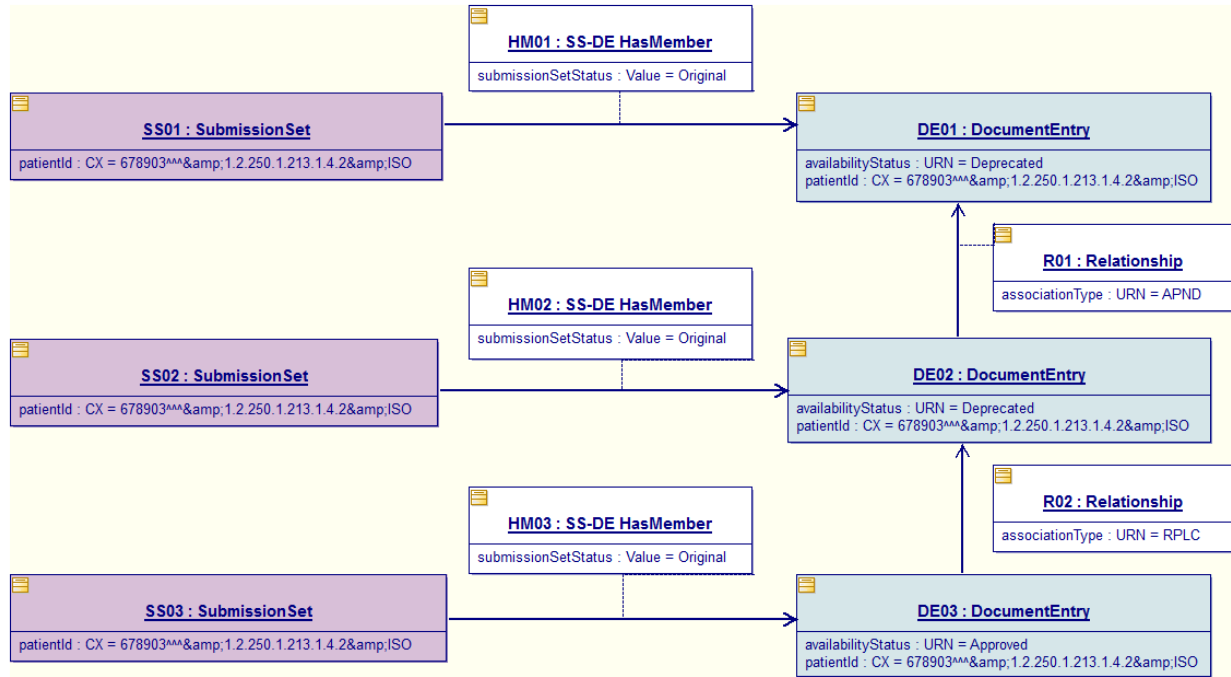


Figure 4.2.2.3-6: New DocumentEntry/Document replacing an addendum (APND) of an existing DocumentEntry/Document (Informative)

955

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.

960

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).

965

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.

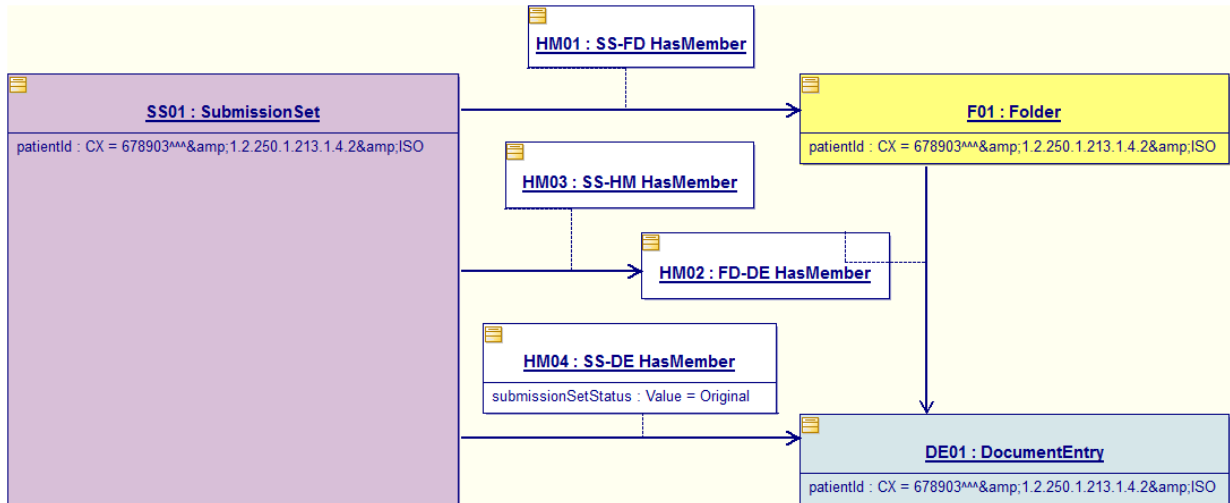
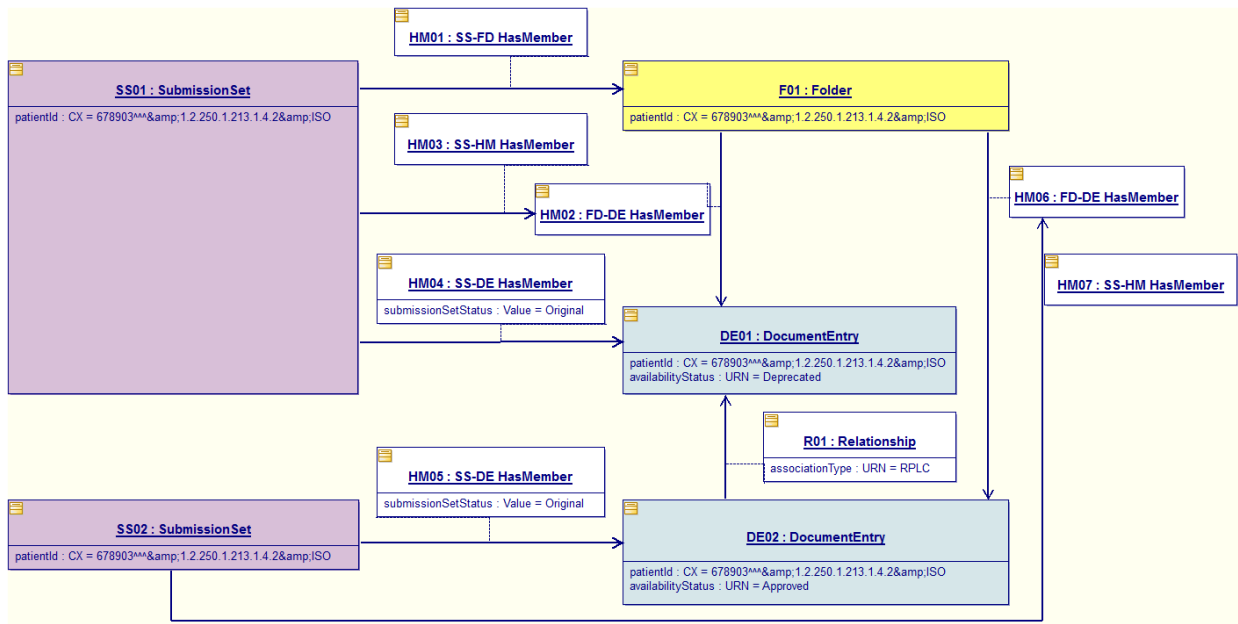


Figure 4.2.2.3-7: DocumentEntry submitted as part of the Folder in a single submission – Starting point (Informative)



970

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

4.2.2.2.4 XFRM_RPLC

975

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 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.

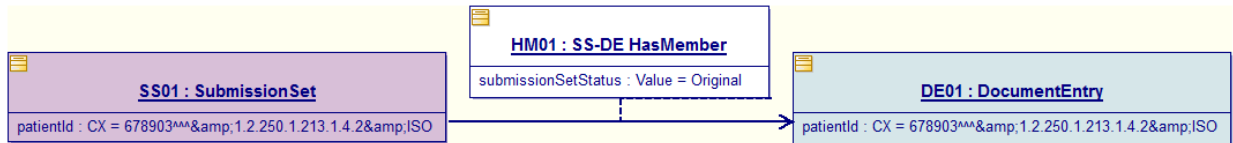


Figure 4.2.2.2.4-1: Starting Point (Informative)

985

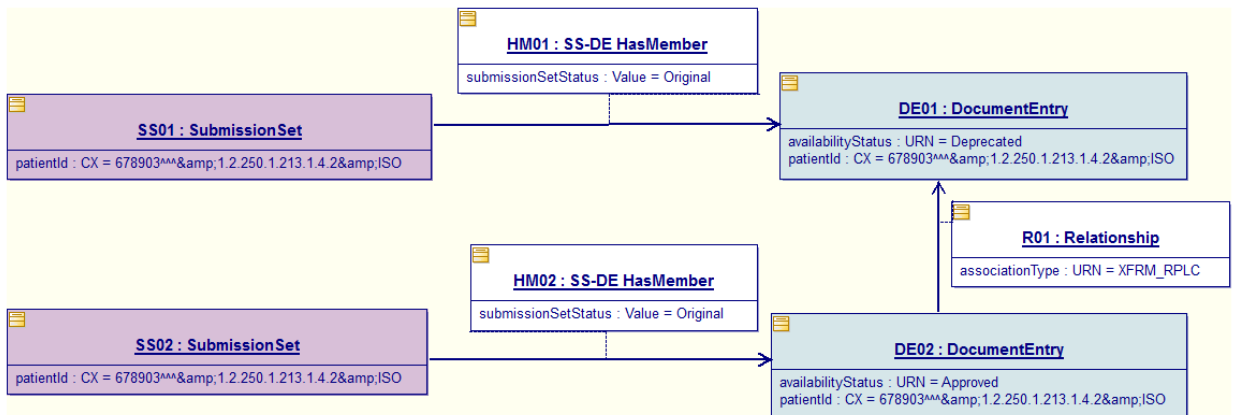


Figure 4.2.2.2.4-2: New DocumentEntry/Document defining a transformation of an existing DocumentEntry/Document that replaces this existing DocumentEntry/Document (Informative)

990

4.2.2.2.5 Signs

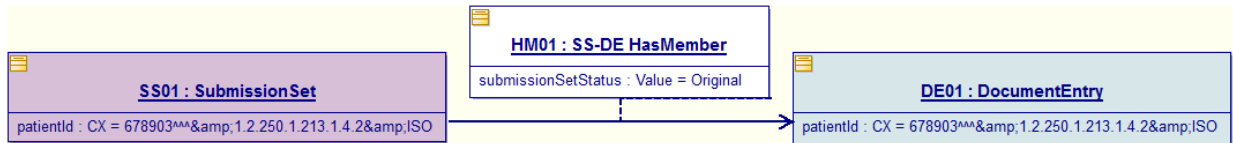
The submission contains a new DocumentEntry and associated Document and a Signs Association linking this new DocumentEntry with an existing DocumentEntry. This new Document is a Digital Signature that signs the existing Document.

995 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 Profile.

In constructing this Association, the attributes are:

1000 **sourceObject:** references the DocumentEntry representing the Digital Signature

targetObject: references the DocumentEntry representing the document being signed
associationType: signs



1005

Figure 4.2.2.5-1: Starting Point (Informative)

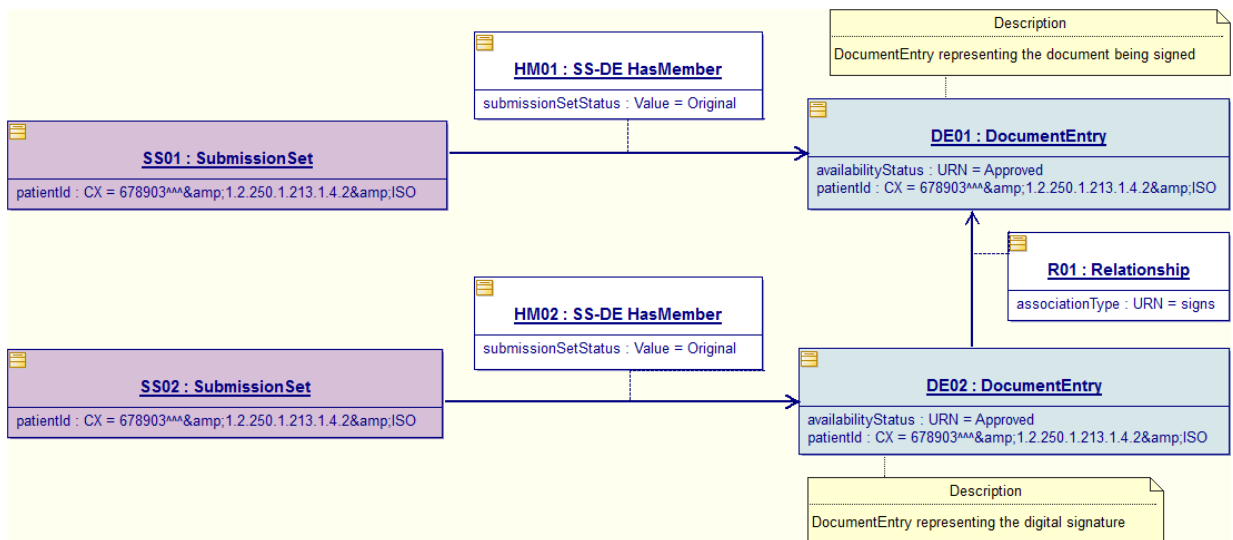


Figure 4.2.2.5-2: New DocumentEntry/Digital signature document signing the existing DocumentEntry/Document (Informative)

1010

4.2.2.2.6 IsSnapshotOf

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.

1015

Example IsSnapshotOf Association

```
1020 <rim:Association
      associationType="urn:ihe:iti:2010:AssociationType:IsSnapshotOf"
      sourceObject="Snapshot"
      targetObject="urn:uuid:e0985823-dc50-45a5-a6c8-a11a829893bd"/>
      <rim:ExtrinsicObject id="Snapshot"
        objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
        . . .
      </rim:ExtrinsicObject>
```

1025

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 Document Registry.

The Document Registry Actor shall verify upon submission:

1030

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

1035

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

4.2.3 Metadata Attributes

1040

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

1045

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).

4.2.3.1.2 Creating Coded Attributes

1050

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.
2. Code Display Name - The display name used to represent the code (code values are not necessarily human-friendly). Must be non-zero length.
- 1055 3. Coding Scheme - The name/identification of the coding scheme that the code comes from.

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

```

1060 <rim:Classification
      classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-
      e362475b143a"
      classifiedObject="ExampleDocument"
      id="IdExample_046"
      objectType="urn:oasis:names:tc:ebxml-
1065 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="10160-0">
      <rim:Name>
        <rim:LocalizedString value="History of Medication Use"/>
      </rim:Name>
1070 <rim:Slot name="codingScheme">
      <rim:ValueList>
        <rim:Value>2.16.840.1.113883.6.1</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1075 </rim:Classification>
  
```

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.

1080 **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

Item	Description
	"10160-0".
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.

1085 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.

1090 **displayName for the code** – shall be used in the Name element. It is expected that the name be semantically consistent with the code value. The same code value may have multiple valid human-readable representations in different languages or in different, but semantically equivalent, wording. The displayName is a convenience provided by the actor that creates the metadata (“metadata creator”), indicating the meaning of the code in a local language used at the “metadata creator”. It is the responsibility of the actor that consumes the metadata (“metadata consumer”) to resolve code values in metadata into human-readable display values. A “metadata consumer” may use the displayName found in the metadata provided by the “metadata creator”, or it may choose another local designation for the code, for example to translate it from English into French.

1100 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.

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

4.2.3.1.3 Creating External Identifiers

1105 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.

1110 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:

```

1115 <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-
1120 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="DocumentEntry01">
      <rim:Name>
        <rim:LocalizedString value="XSDDocumentEntry.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.

1125 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

1130 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.

- authorPerson – zero or one
- authorInstitution – zero or more
- authorRole – zero or more
- authorSpecialty – zero or more
- 1135 • authorTelecommunication – zero or more

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

Coding:

- 1140 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.
- 1145 The @classifiedObject attribute references the object being classified, either ExtrinsicObject or RegistryPackage. The Classification must be embedded within the referenced object and the @classifiedObject attribute must match the @id attribute of the referenced object, thus providing dual context. In the example below, the Classification references the DocumentEntry object that the author value is associated with.
- 1150 The following example shows the definition of a single author for a DocumentEntry.

```
1155 <rim:Classification
      classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d"
      classifiedObject="ExampleDocument"
      id="IdExample_045"
      objectType="urn:oasis:names:tc:ebxml-
1160 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>
1165 <rim:Slot name="authorInstitution">
      <rim:ValueList>
        <rim:Value>Some Hospital^^^^^^^1.2.3.5.8.9.1789.45</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1170 <rim:Slot name="authorRole">
      <rim:ValueList>
        <rim:Value>name of role</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1175 <rim:Slot name="authorSpecialty">
      <rim:ValueList>
        <rim:Value>specialty of author</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1180 <rim:Slot name="authorTelecommunication">
      <rim:ValueList>
        <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1185 </rim:Classification>
```

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

```
1190 <rim:Classification
      classificationScheme="urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d"
      classifiedObject="theSubmission"
      id="Id_045"
      objectType="urn:oasis:names:tc:ebxml-
1195 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>
1200 <rim:Slot name="authorInstitution">
      <rim:ValueList>
        <rim:Value>Some Hospital^^^^^^^1.2.3.5.8.9.1789.45</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1205 <rim:Slot name="authorRole">
      <rim:ValueList>
        <rim:Value>name of role</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1210 <rim:Slot name="authorSpecialty">
      <rim:ValueList>
        <rim:Value>specialty of author</rim:Value>
      </rim:ValueList>
      </rim:Slot>
1215 <rim:Slot name="authorTelecommunication"> <!-- may be multivalued -->
      <rim:ValueList>
        <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value>
      </rim:ValueList>
      </rim:Slot>
</rim:Classification>
```

1220

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:

1225 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.

1230 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.

1235

```
<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>
</rim:Slot>
```

1240

4.2.3.1.4.2 authorPerson

Description:

1245 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:

1250 The format of the authorPerson value is XCN; see Table 4.2.3.1.7-2 for description of XCN format.

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.

1255

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

1260

4.2.3.1.4.3 authorRole

Description:

1265 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:

The format of the authorRole value is String.

1270 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.

```
1275 <rim:Slot name="authorSpecialty">
    <rim:ValueList>
        <rim:Value>Cardiologist</rim:Value>
    </rim:ValueList>
</rim:Slot>
```

4.2.3.1.4.4 authorSpecialty

1280 **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.

Coding:

1285 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.

```
1290 <rim:Slot name="authorSpecialty">
    <rim:ValueList>
        <rim:Value>Cardiology</rim:Value>
    </rim:ValueList>
</rim:Slot>
```

1295

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.

1300 **Coding:**

The format of the authorTelecommunication value is XTN; see Table 4.2.3.1.7-2 for description of XTN format.

1305 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.

1310

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

4.2.3.1.5 UUIDs

1315

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:

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

1320

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.

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

1325

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.

The following conditions shall apply.

1330

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.
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).
 - 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).

1335

1340 **4.2.3.1.7 Metadata Attribute Data types**

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.

1345 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:

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	ITITF	See Section 4.2.3.1.2
CX	HL7 V2.5 Identifier	<p>This is an identifier. 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 style="text-align: center;">IdNumber^^^&OIDofAA&ISO</p> <p>No other values/modifications in other components or subcomponents are allowed. Specifically, components 2 and 3 shall be empty as listed above. An explicit example is:</p> <p style="text-align: center;">543797436^^^&1.2.840.113619.6.197&ISO</p> <p>Note that the '&' 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"> • CXi.1 shall be present and hold the identifier value. This component may exceed the HL7 conformance length of 15 characters. • CXi.2 and CXi.3 shall be empty. • CXi.4 (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. Either Universal ID with Universal ID Type or Namespace ID is permitted. • CXi.5 (Identifier Type Code) shall be present and chosen from either a URN defined by IHE, or a locally defined value. • No other components shall be present. <p>The following IHE values for CXi.5 (Identifier Type Codes) are defined:</p>

Data Type	Source Standard	Encoding Specification
		<p>urn:ihe:iti:xds:2013:uniqueId 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”, then the CXi value is 11379^^^&1.3.6367.3.7&ISO^urn:ihe:iti:xds:2013:uniqueId</p> <p>urn:ihe:iti:xds:2013:accession This code shall be used when the identifier is an accession number. It shall contain</p> <ul style="list-style-type: none"> • The accession number and • for accession values that are not globally unique, the Assigning Authority shall be included. <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^^^&1.2.3.4.5.6&ISO^urn:ihe:iti:xds:2013:accession</p> <p>urn:ihe:iti:xds:2013:referral 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>urn:ihe:iti:xds:2013:order Order number and assigning authority shall be present. For example: 134467653^^^&1.2.3.4.5.42.1&ISO^urn:ihe:iti:xds:2013:order</p> <p>urn:ihe:iti:xdw:2013:workflowInstanceId 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 components shall be present: For example, if the workflow identifier is “2.16.840.1” the CXi value is: 2.16.840.1^^^urn:ihe:iti:xdw:2013:workflowInstanceId</p> <p>urn:ihe:iti:xds:2016:studyInstanceUID This code shall be used when the identifier is a DICOM Study Instance UID. CXi.1 shall contain the value of Study Instance UID. Only the CXi.1 and CXi.5 components shall be present. For example when the Study Instance UID has a value of “1.2.3.45.678” then the CXi value is 1.2.3.45.678^^^urn:ihe:iti:xds:2016:studyInstanceUID</p> <p>urn:ihe:iti:xds:2015:encounterId Encounter identifier (also known as visit number) and assigning authority shall be present. For example: 245348841^^^&1.2.840.113619.6.197&ISO^urn:ihe:iti:xds:2015:encounterId</p>

Data Type	Source Standard	Encoding Specification
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.</p> <p>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 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 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^^^&1.3.6.1.4.1.21367.2005.3.7& ISO</p>
Integer	W3C XML Schema Part 2	XML Schema Part 2: Data Types Section 3.3.13
MIME Type	RFC 2046	https://www.iana.org/assignments/media-types
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	RFC 3061 . OID in URN syntax	See RFC 3061 . An example is urn:oid: 1.3.6.1.4.1.21367.2005.3.7.
Predefined URN	RFC 2141	Specific URNs are defined and assigned in this framework. An example is urn:ihe:iti:2007:ResponseStatusType:PartialSuccess
SHA1	RFC 3174	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	RFC 2616	Uniform Resource Identifier

Data Type	Source Standard	Encoding Specification
UTF-8	RFC 3629	Unicode standard
UUID	RFC 4122	<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"> Identifier Last Name First Name Second and Further Given Names Suffix Prefix Assigning Authority <p>All of the HL7 v2.5 fields may be specified as optional components with the following restrictions:</p> <p>Either name or an identifier shall be present. Inclusion of other components is optional provided the slot value length restrictions imposed by ebXML3.0, 256 characters, is not exceeded.</p> <p>If component 1 (Id Number) is specified, component 9 (Assigning Authority) shall be present if available.</p> <p>The XCN Component 9 is subject to the same the restrictions as defined for the CX data type component 4. Thus: the first subcomponent shall be empty, the second subcomponent must be an ISO OID (e.g., 1.2.840.113619.6.197), and the third subcomponent shall read 'ISO'.</p> <p>Any empty component shall be treated by the Document Registry as not specified. This is in compliance with HL7 v2.5.</p> <p>Trailing delimiters are recommended to be trimmed off. Receiving actors shall ignore trailing delimiters. This is in compliance with HL7 v2.5.</p> <p>An example of person name with Id number using this data type is as follows:</p> <p>11375^Welby^Marcus^J^Jr. MD^Dr^^&1.2.840.113619.6.197&ISO</p>
XON	HL7 V2.5 Organization Name	<p>This type provides the name and identification of an organization. This specification restricts the coding to the following fields:</p> <ul style="list-style-type: none"> XON.1 – Organization Name – this field is required XON.6.2 – Assigning Authority Universal Id – this field is required if XON.10 is valued and not an OID XON.6.3 – Assigning Authority Universal Id Type – this field is required if XON.10 is valued and not an OID and shall have the value "ISO" XON.10 – Organization Identifier – this field is optional <p>No other fields shall be specified. The XON data type in Document Sharing metadata results in a valid encoding of an HL7 v2.5 XON encoding, with the exception of length limitations. Component length restrictions are unobserved; however, the total length including delimiters shall not exceed the limit of the ebXML Slot Value.</p> <p>It is common for organizations to be uniquely identified by an OID. In such cases, the Organization (Identifier component 10) may contain the organization's OID. If the Organization Identifier is not an OID, the</p>

Data Type	Source Standard	Encoding Specification
		<p>metadata use assumes that it has been assigned so that the composite Id created by combining components 6 and 10 is a unique identifier for the organization.</p> <p>Examples:</p> <p>Some Hospital</p> <p>Some Hospital^^^^^^1.2.3.4.5.6.7.8.9.1789.45</p> <p>Some Hospital^^^^&1.2.3.4.5.6.7.8.9.1789&ISO^^^^45</p>
XTN	<p>HL7 V2.5 Extended Telecommunications Number</p> <p>Chapter 2A Section 89</p>	<p>This type provides the telecommunications address of an entity (for example author, intended recipient). This specification restricts the use to the following fields:</p> <p>For phone numbers:</p> <p>XTN.2 – Indicates the type of phone number (Optional)</p> <p>XTN.3 – The type of telecommunication address (Required)</p> <p>XTN.5 – Country Code (Optional)</p> <p>XTN.6 – Area/City Code (Optional)</p> <p>XTN.7 – Subscriber Number (Required)</p> <p>XTN.8 – Extension (Optional)</p> <p>For email addresses:</p> <p>XTN.2 – Optional. If present, SHALL have the value "NET".</p> <p>XTN.3 – SHALL have the value "Internet".</p> <p>XTN.4 – the telecommunications address, e.g., name@example.com (Required)</p> <p>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. Use of ITU E.123 notation for telephone numbers is recommended.</p> <p>Example:</p> <p>^^Internet^radiology@healthcare.example.org_org (e-mail address)</p> <p>^PRS^PH^^31^^(042) 1123 4567 (national phone number)</p> <p>^PRS^PH^^^^+31 42 1123 4567 (international phone number)</p>

1350

4.2.3.1.8 General format of DocumentEntry, Folder and SubmissionSet attribute tables

The metadata attribute definition tables each have five columns:

- 1355 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.
2. The second column contains a brief description of the attribute.
- 1360 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.
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:

- 1365 • **ebRIM Classification** – indicates that the attribute is coded as a classification object, see Section 4.2.3.1.2.

- **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"
... >
```

- 1370 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>
```

- 1380 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).

- 1385 • **ebRIM Name**– indicates this attribute is held in an ebRIM Name object. For example the title attribute is:

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

- 1390 • **ebRIM Description**– indicates this attribute is held in an ebRIM Description object. For example the comment attribute is:

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

1395

5. 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.

- 1400 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.

- 1405 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.

4.2.3.1.10 classificationScheme vs. classificationNode

Classification elements in ebRIM use one of two attributes: classificationScheme and classificationNode.

- 1415 The attributes serve two different purposes:

- **classificationScheme**

- 1420 As the name suggests, the classificationScheme attribute specifies the category ("scheme") by which the referenced object is being classified. For example, classificationScheme="urn:uuid:ccc5598-8b07-4b77-a05e-ae952c785ead" indicates that a DocumentEntry is being classified by practiceSetting. The classificationScheme does not indicate WHICH practiceSetting applies; that information is recorded elsewhere in the Classification element.

The use of classificationScheme corresponds with "external classifications", as described in ebRIM. This is the predominant type of classification in Document Sharing.

- 1425
- **classificationNode**

The classificationNode attribute provides an actual classification for the referenced object. For example, classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd" indicates that a RegistryPackage is a SubmissionSet.

- 1430 The use of classificationNode corresponds with "internal classifications", as described in ebRIM. Document Sharing metadata uses this type of classification for three purposes:

- To label a RegistryPackage as a SubmissionSet
- To label a RegistryPackage as a Folder
- To label a SubmissionSet, Folder, or DocumentEntry as containing Limited Metadata

4.2.3.2 DocumentEntry Attributes

- 1435 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.

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.

1440 Please see Section 4.2.3.1.8 for the general format of DocumentEntry, Folder and SubmissionSet attribute tables.

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.	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	The code specifying the detailed technical format of the document.	Code	ebRIM Classification	4.2.3.2.9
hash	The hash of the contents of the document.	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.	String	ebRIM Slot	4.2.3.2.13
legalAuthenticator	Represents a participant within an authorInstitution who has legally authenticated or attested the document.	XCN	ebRIM Slot	4.2.3.2.14
limitedMetadata	Indicates whether the DocumentEntry was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document	4.2.3.2.29	ebRIM Classification	4.2.3.2.29

DocumentEntry Metadata Attribute	Description	Data Type (Table 4.2.3.1.7-2)	Coding (Section 4.2.3.1.8)	Detail (See Section)
	Source.			
contentType	MIME type of the document.	MIME type	XML attribute	4.2.3.2.15
objectType	The type of DocumentEntry (e.g., On-Demand 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
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 related to the document.	CXi	ebRIM Slot	4.2.3.2.28
repositoryUniqueId	The globally unique identifier of the repository where the document can be accessed.	OID	ebRIM Slot	4.2.3.2.18
serviceStartTime	The start time of the service being documented.	DTM	ebRIM Slot	4.2.3.2.19
serviceStopTime	The stop time of the service being documented.	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's medical record identifier (e.g., Patient Id) in the local patient identifier domain of the creating entity.	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	Globally unique identifier assigned to the document by its creator.	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

1445 Description:

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

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

4.2.3.2.2 DocumentEntry.availabilityStatus

1450 Description:

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

Approved The document is available for patient care.

Deprecated The document is obsolete.

1455 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:

1460 The format of the availabilityStatus value is a URN.

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.

1465

```
<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"
>
...

```

1470

4.2.3.2.3 DocumentEntry.classCode

Description:

1475 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.

1480

Coding:

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

1485

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".

```
1490 <rim:Classification
      classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"
      classifiedObject="ExampleDocument"
      id="IdExample_046"
      objectType="urn:oasis:names:tc:ebxml-
1495 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="10160-0"
      >
      <rim:Name>
      <rim:LocalizedString
1500       value="History of Medication Use"/>
      </rim:Name>
      <rim:Slot name="codingScheme">
      <rim:ValueList>
1505       <rim:Value>2.16.840.1.113883.6.1</rim:Value>
      </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
```

4.2.3.2.4 DocumentEntry.comments

1510 **Description:**

Contains comments associated with the document.

Coding:

Max length is unbounded.

1515 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.

```
1520 <rim:Description>
      <rim:LocalizedString value = "comment associated with the Document"/>
    </rim:Description>
```

4.2.3.2.5 DocumentEntry.confidentialityCode

Description:

1525 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.

1530 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.

- 1535
- [1...1] Confidentiality Security Classification Label Field
 - [0...*] Sensitivity Security Category Label Field
 - [0...*] Compartment Security Category Label Field
 - [0...*] Integrity Security Category Label Field
 - [0...*] Handling Caveat Security Category Field

1540 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.

1545 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):

- 1550
- 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).
 - 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.
 - 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.
- 1555
- 1560

- 1565
- 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.

1570

 - 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.

1575 **Coding:**

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
1580 urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f.

The following example shows two confidentialityCode 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
1585 the DocumentEntry labeled "ExampleDocument".

```
1590 <rim:Classification
      classificationScheme=
        "urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
      classifiedObject="ExampleDocument"
      id="IdExample_046"
      objectType="urn:oasis:names:tc:ebxml-
1595 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="N"
      <rim:Name>
        <rim:LocalizedString
          value="Normal Clinical Data"/>
        </rim:Name>
1600 <rim:Slot name="codingScheme">
      <rim:ValueList>
        <rim:Value>2.16.840.1.113883.5.25</rim:Value>
        </rim:ValueList>
      </rim:Slot>
1605 </rim:Classification>
      <rim:Classification
        classificationScheme=
          "urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
        classifiedObject="ExampleDocument"
1610 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"/>
        </rim:Name>
1615 <rim:Slot name="codingScheme">
      <rim:ValueList>
        <rim:Value>2.16.840.1.113883.1.11.20471</rim:Value>
1620 </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
```

4.2.3.2.6 DocumentEntry.creationTime

1625 **Description:**

Represents the time the author created the document.

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.

1630 The value is coded as a single value within an ebRIM Slot in the DocumentEntry.

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

```
1635 <rim:Slot name="creationTime">
      <rim:ValueList>
        <rim:Value>20041225212010</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

1640 4.2.3.2.7 DocumentEntry.entryUUID

Description:

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:

1645 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.

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-

1650 9e5c-88b9045a96f6

```
1655 <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:

1660 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, 1665 practiceSettingCode or typeCode as such a conflict would create an ambiguous situation.

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

1670 multiple classification objects. For the eventCodeList metadata attribute, the classificationScheme shall be urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4.

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

1675

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

1680

1685

1690

1695 **4.2.3.2.9 DocumentEntry.formatCode**

Description:

The code specifying the detailed technical 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.

1700 The mimeType indicates the base format; the formatCode indicates the detailed-level technical format. 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). The formatCode is often an indicator of the IHE Document Content Profile to which the document conforms.

1705

Format codes may be specified by multiple organizations. Format codes for Document Content Profiles defined by the ITI domain shall be in URN format and have names with the prefix urn:ihe:iti:

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

1710

urn:ihe:'domain initials':

Format codes defined by non-IHE domains should be a valid unique URN.

Coding:

1715 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 classificationScheme shall be urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d. Any valid URN may be used as a formatCode.

1720 The following example using the XDS-SD document content profile specifies formatCode="urn:ihe:iti:xds-sd:pdf:2008" with display name "XDS-Scanned Documents" and coding scheme "urn" for the DocumentEntry labeled "ExampleDocument".

```
1725 <rim:Classification
      classificationScheme=
        "urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"
      classifiedObject="ExampleDocument"
      id="IdExample_049"
      objectType="urn:oasis:names:tc:ebxml-
1730 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="urn:ihe:iti:xds-sd:pdf:2008"
    >
    <rim:Name>
      <rim:LocalizedString value="XDS-Scanned Documents"/>
    </rim:Name>
    <rim:Slot name="codingScheme">
1735   <rim:ValueList>
     <rim:Value>1.3.6.1.4.1.19376.1.2.3</rim:Value>
   </rim:ValueList>
  </rim:Slot>
</rim:Classification>
```

1740

4.2.3.2.10 DocumentEntry.hash

Description:

The hash of the contents of the document.

Coding:

1745 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.

1750

1755

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

4.2.3.2.11 DocumentEntry.healthcareFacilityTypeCode

Description:

1760

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

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".

1765

Coding:

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.

1770

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

1775

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

1780

1785

1790

4.2.3.2.12 DocumentEntry.homeCommunityId

Description:

1795 A globally unique identifier for a community where the DocumentEntry and document can be accessed.

Coding:

1800 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 ITI TF-2a: 3.18.4.1.2.3.8 and ITI TF-2b: 3.38.4.1.2.1.

```
1805 <rim:ExtrinsicObject home="urn:oid:1.2.3" ...>
    ...
</rim:ExtrinsicObject>
```

4.2.3.2.13 DocumentEntry.languageCode

Description:

Specifies the human language of character data in the document.

1810 **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.

The following example shows a languageCode of en-CA.

```
1815 <rim:Slot name="languageCode">
    <rim:ValueList>
        <rim:Value>en-CA</rim:Value>
    </rim:ValueList>
1820 </rim:Slot>
```

4.2.3.2.14 DocumentEntry.legalAuthenticator

Description:

1825 Represents a participant within an 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:

1830 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.

```
1835 <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

1840 **Description:**

MIME type of the document in the Repository

Coding:

1845 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.

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

```
1850 <rim:ExtrinsicObject mimeType="application/pdf"
    id="ExampleDocument"
    objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1">
```

4.2.3.2.16 DocumentEntry.patientId

Description:

1855 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).

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

Coding:

1860 The format of the patientId value is CX, see Table 4.2.3.1.7-2.

It shall contain two parts:

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

An Id from the above Assigning Authority.

1865 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.

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.

1870 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.

```
1875 <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-
1880       regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="DocumentEntry01">
  <rim:Name>
    <rim:LocalizedString value="XDSDocumentEntry.patientId" />
  </rim:Name>
</rim:ExternalIdentifier>
```

1885

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

4.2.3.2.17 DocumentEntry.practiceSettingCode

Description:

1890 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).

Coding:

1895 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:ccccf5598-8b07-4b77-a05e-ae952c785ead.

1900 The following example specifies practiceSettingCode="ExamplepracticeSettingCode" with display name "ExamplepracticeSettingCodeDisplayName" and coding scheme "Example Practice Setting Code Scheme" for the DocumentEntry labeled "ExampleDocument".

```
1905 <rim:Classification
      ClassificationScheme="urn:uuid:ccccf5598-8b07-4b77-a05e-ae952c785ead"
      classifiedObject="ExampleDocument"
      id="IdExample_052"
      objectType="urn:oasis:names:tc:ebxml-
1910 regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="ExamplepracticeSettingCode">
      <rim:Name>
        <rim:LocalizedString
          value="ExamplepracticeSettingCodeDisplayName"/>
        </rim:Name>
      <rim:Slot name="codingScheme">
1915 <rim:ValueList>
          <rim:Value>Example Practice Setting Code Scheme</rim:Value>
        </rim:ValueList>
      </rim:Slot>
    </rim:Classification>
```

1920

4.2.3.2.18 DocumentEntry.repositoryUniqueId

Description:

1925 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 ebRIM Slot in the DocumentEntry.

The following example shows a repositoryUniqueId of 1.3.6.1.4.5.

1930

```
1935 <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:

1940 Represents the start time of the service being documented (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.

1945 **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.

1950 The following example shows a serviceStartTime of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

1955

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

4.2.3.2.20 DocumentEntry.serviceStopTime

Description:

1960 Represents the stop time of the service being documented (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.

1965 If the service happens at a point in time, this attribute shall contain the same value as the serviceStartTime.

Coding:

Max length is 256 characters. The format of the serviceStopTime 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.

1970 The following example shows a stop time of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

1975

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

4.2.3.2.21 DocumentEntry.size

Description:

1980 Size in bytes of the byte stream that comprises the document.

Coding:

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.

1985

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

1990

4.2.3.2.22 DocumentEntry.sourcePatientId

Description:

1995 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.

Coding:

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:

2000 An Id (e.g., Patient Id CX.1).

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

All other CX components shall be omitted.

2005

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

2010 4.2.3.2.23 DocumentEntry.sourcePatientInfo

Description:

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

2015 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.

2020 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:

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).

2025

```
<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>
  </rim:ValueList>
</rim:Slot>
```

2030

2035

sourcePatientInfo should include values for:

- 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)

2040

M – Male F – Female

O – Other U – Unknown

2045 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:

Represents the title of the document.

2050 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.

2055 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.

```
2060 <rim:ExtrinsicObject
      id="ExampleDocument"
      objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
      mimeType="application/pdf">
      <rim:Name>
      <rim:LocalizedString value="Example Document Title"/>
      </rim:Name>
2065      ...
    </rim:ExtrinsicObject>
```

4.2.3.2.25 DocumentEntry.typeCode

Description:

2070 The code specifying the precise type of document from the user's 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:

2075 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".

```
2080 <rim:Classification
      classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"
      classifiedObject="ExampleDocument"
      nodeRepresentation="ExemplotypeCode"
      id="IdExample_053"
2085      objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification">
      <rim:Name>
        <rim:LocalizedString value="typeCodeDisplayName" />
      </rim:Name>
2090 <rim:Slot name="codingScheme">
      <rim:ValueList>
        <rim:Value>Example Type Code Scheme</rim:Value>
      </rim:ValueList>
      </rim:Slot>
2095 </rim:Classification>
```

4.2.3.2.26 DocumentEntry.uniqueId

Description:

Globally unique identifier assigned to the document by its creator.

2100 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.

2105 Two documents MAY be assigned the same uniqueId if the documents have the same byte sequence, but they SHALL NOT be assigned the same uniqueId if the documents have different byte sequences when communicated via a Document Sharing protocol. The size and hash metadata attributes allow for a quick comparison, so it is adequate to treat two documents as having the same byte sequence if their size and hash attributes are the same.

2110 For formats where different byte sequences can be functionally equivalent (XML, for example, where whitespace between elements is ignored), it is important to note that documents with different byte sequences SHALL have different uniqueIds even if they are functionally equivalent when communicated via a Document Sharing protocol.

Coding:

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

2115 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.

2120 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.)

```
2125 <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-
2130 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="DocumentEntry01">
      <rim:Name>
        <rim:LocalizedString value="XDSDocumentEntry.uniqueId"/>
      </rim:Name>
    </rim:ExternalIdentifier>
```

2135

4.2.3.2.27 DocumentEntry.URI

Description:

The URI attribute contains the URI for the document.

Coding:

2140 Coded as an eBRIM Slot. Shall have only a single value. Max length is 256 characters.

```
2145 <rim:Slot name="URI">
      <rim:ValueList>
        <rim:Value>DOC001.XML</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

4.2.3.2.28 DocumentEntry.referenceIdList

Description:

2150 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, XDW Workflow Instance Identifiers, etc. The referenceIdList contains Identifiers CXi encoded, as specified in Table 4.2.3.1.7-2.

Coding:

2155 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 two values in the referenceIdList:

```
2160 <rim:Slot name="urn:ihe:iti:xds:2013:referenceIdList ">
      <rim:ValueList>
        <rim:Value>
2165 2013001^^^&1.2.3.4.5.6&ISO^urn:ihe:iti:xds:2013:accession
        </rim:Value>
        <rim:Value>
        1.2.3.12.78.23^^^&1.2.3.4&ISO^urn:ihe:iti:xds:2013:uniqueId^&1.2.3.4&ISO
        </rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

2170 **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:

2175 The Document Entry is flagged using an ebRIM Classification with a classificationNode of urn:uuid:ab9b591b-83ab-4d03-8f5d-f93b1fb92e85.

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

```
2180 <ExtrinsicObject id="DocEntry">
  (...)
  <Classification classifiedObject="DocEntry"
    classificationNode="urn:uuid:ab9b591b-83ab-4d03-8f5d-f93b1fb92e85" />
  (...)
2185 </ExtrinsicObject>
```

4.2.3.2.30 DocumentEntry.objectType

Description:

2190 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 a 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
2195 available information at the time of retrieval. It is designed by setting an objectType equal to the 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.

2200 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.

```
2205 <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.

2210 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.

See Section 4.2.3.1.8 for the General format of DocumentEntry, Folder and SubmissionSet attribute tables.

2215

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 SubmissionSet. This attribute contains the sub-attributes: authorInstitution, authorPerson, authorRole, authorSpecialty, authorTelecommunication.	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 associated content in the SubmissionSet.	Code	ebRIM Classification	4.2.3.3.4
entryUUID	A globally unique identifier used to identify the entry.	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 organizations or persons 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	See 4.2.3.3.13	ebRIM Classification	4.2.3.3.13

SubmissionSet Metadata Attribute	Description	Form	Coding	Detail
	Document Source.			
patientId	The patientId represents the primary subject of care of the SubmissionSet.	CX	ebRIM ExternalIdentifier	4.2.3.3.8
sourceId	Identifier of the entity that contributed the SubmissionSet.	OID	ebRIM ExternalIdentifier	4.2.3.3.9
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	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:

2220 Represents the humans and/or machines that authored the SubmissionSet. See Section 4.2.3.1.4 for details on creating the structure.

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

4.2.3.3.2 SubmissionSet.availabilityStatus

Description:

2225 Represents the status of the SubmissionSet. Since the deprecation of SubmissionSets is not allowed, this value shall always be Approved.

Coding:

The availabilityStatus value shall be

urn:oasis:names:tc:ebxml-regrep:StatusType:Approved.

2230

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

2235

4.2.3.3.3 SubmissionSet.comments

Description:

Contains comments associated with the SubmissionSet.

Coding:

2240 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.

```
2245 <rim:Description>
      <rim:LocalizedString value = "comment associated with SubmissionSet"/>
    </rim:Description>
```

4.2.3.3.4 SubmissionSet.contentTypeCode

2250 **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.

Coding:

2255 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

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

```
2260 <rim:Classification
      classificationScheme="urn:uuid:aa543740-bdda-424e-8c96-df4873be8500"
      classifiedObject="ExampleSubmissionSet"
      id="IdExample_056"
2265     objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="ExamplecontentTypeCode"
      >
2270   <rim:Name>
      <rim:LocalizedString value="Example contentTypeCodeDisplayName" />
    </rim:Name>
      <rim:Slot name="codingScheme">
2275     <rim:ValueList>
      <rim:Value>Example Specific Value</rim:Value>
    </rim:ValueList>
    </rim:Slot>
  </rim:Classification>
```


4.2.3.3.5 SubmissionSet.entryUUID

2280 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.

2285 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.

```
2290 <rim:RegistryPackage
      id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6"
      objectType=
        "urn:oasis:names:tc:ebxml-
      regrep:Object:RegistryObject:RegistryPackage"
    > ...
```

2295

4.2.3.3.6 SubmissionSet.homeCommunityId

Description:

A globally unique identifier for a community.

Coding:

2300 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 ITI TF-2a: 3.18.4.1.2.3.8 and ITI TF-2b: 3.38.4.1.2.1.

```
2305 <rim:RegistryPackage home="urn:oid:1.2.3" ...>
      ...
    </rim:RegistryPackage>
```

4.2.3.3.7 SubmissionSet.intendedRecipient

Description:

2310 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:

2315 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.

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.

2325

```
<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:Value>|^Internet^dr.oz@healthcare.example.org</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

2330

2335

2340

4.2.3.3.8 SubmissionSet.patientId

Description:

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

2345

Coding:

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

It shall contain two parts:

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

An Id from the above Assigning Authority.

2350

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.

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

2355 RegistryPackage representing the SubmissionSet. There shall be zero or one single 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.

```
2360 <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-
2365 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="SubmissionSet01">
      >
      <rim:Name>
      <rim:LocalizedString value = "XDSSubmissionSet.patientId"/>
2370 </rim:Name>
</rim:ExternalIdentifier>
```

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

2375 **4.2.3.3.9 SubmissionSet.sourceId**

Description:

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.

2380 **Coding:**

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

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.

2385 In the example below the sourceId is 1.3.6.1.4.1.21367.2005.3.7.

```
2390 <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-
2395 regrep:ObjectType:RegistryObject:ExternalIdentifier"
      registryObject="SubmissionSet01">
      <rim:Name>
        <rim:LocalizedString value = "XDSSubmissionSet.sourceId"/>
      </rim:Name>
    </rim:ExternalIdentifier>
```

4.2.3.3.10 SubmissionSet.submissionTime

2400 **Description:**

Represents the point in time at the creating entity when the SubmissionSet was created.

This shall be provided by the submitting system.

Coding:

2405 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).

```
2410 <rim:Slot name="submissionTime">
      <rim:ValueList>
        <rim:Value>20041225212010</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

2415

4.2.3.3.11 SubmissionSet.title

Description:

Shall contain the title of the SubmissionSet.

Coding:

2420 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.

2425

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

2430 **4.2.3.3.12 SubmissionSet.uniqueId**

Description:

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

Coding:

2435 The format of the SubmissionSet.uniqueId value is OID (see Table 4.2.3.1.7-2).

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.

2440

```
<rim:ExternalIdentifier
  identificationScheme="urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8"
  id="IdExample_059"
  objectType="urn:oasis:names:tc:ebxml-
2445 regrep:ObjectType:RegistryObject:ExternalIdentifier"
  value="1.2.3.4.5"
  registryObject="SubmissionSet01">
  <rim:Name>
    <LocalizedString value = "XDSSubmissionSet.uniqueId"/>
2450 </rim:Name>
</rim:ExternalIdentifier>
```

4.2.3.3.13 SubmissionSet.limitedMetadata

Description:

2455 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 classificationNode of urn:uuid:5003a9db-8d8d-49e6-bf0c-990e34ac7707. Zero or one may be present.

2460 The following example marks the "SubmissionSet" SubmissionSet as created via the less rigorous metadata requirements.

```

2465 <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"/>
2470      (...)
    </RegistryPackage>

```

4.2.3.4 Folder Attributes

2475 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	The set of codes specifying the type of clinical activities that resulted in placing DocumentEntry objects in the Folder.	Code	ebRIM Classification	4.2.3.4.2
comments	Comments associated with the Folder.	String	ebRIM Description	4.2.3.4.3
entryUUID	A globally unique identifier used to identify the entry.	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	Represents the primary subject of care of the Folder.	CX	ebRIM ExternalIdentifier	4.2.3.4.7
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

2480 **4.2.3.4.1 Folder.availabilityStatus**

Description:

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

Coding:

2485 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.

```
2490 <rim:RegistryPackage
      id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855"
      status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
    > ...
```

2495 **4.2.3.4.2 Folder.codeList**

Description:

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

2500 **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 more values. Code multiple values by creating multiple classification objects.

2505 The following example specifies contentTypeCode="ExamplecontentTypeCode" with display name "ExamplecodeList CodeDisplayName" and coding scheme "Example coding scheme" for the Folder labeled "ExampleFolder".

```
2510 <rim:Classification
      classificationScheme= "urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5"
      classifiedObject="ExampleFolder"
      id="IdExample_062"
      objectType="urn:oasis:names:tc:ebxml-
2515       regrep:ObjectType:RegistryObject:Classification"
      nodeRepresentation="Examplecode"
      >
      <rim:Name>
        <rim:LocalizedString
          value="ExamplecodeList CodeDisplayName" />
2520 </rim:Name>
      <rim:Slot name="codingScheme">
        <rim:ValueList>
          <rim:Value>Example coding scheme</rim:Value>
          </rim:ValueList>
        </rim:Slot>
2525 </rim:Classification>
```

4.2.3.4.3 Folder.comments

Description:

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

2530 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.

Coding:

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

2535 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 Folder.

```
2540 <rim:Description>
      <rim:LocalizedString value = "comments"/>
      </rim:Description>
```

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).

Coding:

2550 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.

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

2555 urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a9ab6

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

2560

4.2.3.4.5 Folder.homeCommunityId

Description:

A globally unique identifier for a community.

Coding:

2565 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 ITI TF-2a: 3.18.4.1.2.3.8 and ITI TF-2b: 3.38.4.1.2.1.

2570

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

4.2.3.4.6 Folder.lastUpdateTime

Description:

2575 Most recent point in time when a DocumentEntry was placed in the Folder.

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.

2580 The following example shows a last update of December 25, 2004 21:20:10 Coordinated Universal Time (UTC).

```
2585 <rim:Slot name="lastUpdateTime">
      <rim:ValueList>
        <rim:Value>20041225212010</rim:Value>
      </rim:ValueList>
    </rim:Slot>
```

4.2.3.4.7 Folder.patientId

2590 **Description:**

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

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:

2595 Max length unbounded. The format of the patientId value is CX (see Table 4.2.3.1.7-2).

It shall contain two parts:

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

An Id from the Assigning Authority.

2600 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.

```
2605 <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-
      regrep:ObjectType:RegistryObject:ExternalIdentifier"
2610 >
      <rim:Name>
        <rim:LocalizedString value = "XDSFolder.patientId"/>
      </rim:Name>
    </rim:ExternalIdentifier>
```

2615 This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or External Identifier elements except as described above.

4.2.3.4.8 Folder.title

Description:

Shall contain the title of the Folder.

2620 Note: Prior to the availability of this attribute, the comments attribute might have been used to hold the title of the folder. With the addition of this attribute, the comments attribute shall not hold the folder title.

Coding:

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

2625 Max length is 256 characters.

In this example the title is "title".

```
2630 <Name>
    <LocalizedString value="title"/>
</Name>
```

4.2.3.4.9 Folder.uniqueId

Description:

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

2635 **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.

```
2640 <rim:ExternalIdentifier
    identificationScheme="urn:uuid:75df8f67-9973-4fbe-a900-df66cefec5a"
    id="IdExample_059"
    objectType="urn:oasis:names:tc:ebxml-
2645 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>
2650 </rim:ExternalIdentifier>
```

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

2655 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.

Coding:

2660 The Folder is flagged using an ebRIM Classification with a classificationNode 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.

```
2665 <RegistryPackage id="Folder">
      (...)
      <Classification classifiedObject="Folder"
2670 classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2"
      id="ID_061" objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"/>
      <Classification classifiedObject="Folder"
2675 classificationNode="urn:uuid:2c144a76-29a9-4b7c-af54-b25409fe7d03"/>
      (...)
    </RegistryPackage>
```

4.2.4 Success and Error Reporting

2680 The ebXML RegistryResponse (for submission transactions) or ebXML AdhocQueryResponse (for query transactions) element will contain the status of the requested operation. The status attribute reflects the status of the operation and shall be one of the following values:

```
2685 urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success
      urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure
      urn:ihe:iti:2007:ResponseStatusType:PartialSuccess
```

A response to a Submission Request that indicates success with no warnings is:

```
2690 <rs:RegistryResponse
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success"/>
```

A response to a query request that indicates success with no warnings and no documents found is:

2695

```
<query:AdhocQueryResponse
  status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
<rim:RegistryObjectList/>
</query:AdhocQueryResponse >
```

2700

If the transaction resulted in warnings or errors, the ebXML RegistryResponse or ebXML AdhocQueryResponse shall contain an ebXML RegistryErrorList, which shall contain corresponding RegistryError elements.

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. See Section 4.2.4.2 for more details.

2705

4.2.4.1 RegistryErrors Element

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

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

2710

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.

2715

An example response to a Submission Request that reports two errors and one warning is:

```
2720 <rs:RegistryResponse
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure">
  <rs:RegistryErrorList
    highestSeverity="urn:oasis:names:tc:ebxml-
2725     regrep:ErrorSeverityType:Error">
    <rs:RegistryError
      errorCode="XDSPatientIdDoesNotMatch"
      codeContext="Patient Id in Document (Document1) does not match
        SubmissionSet"
      location=""
      severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
    <rs:RegistryError
2730     errorCode="XDSRegistryMetadataError"
      codeContext="RegistryPackage (SubmissionSet) is not labeled as
        SubmissionSet or Folder"
      location=""
      severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
    <rs:RegistryError
2735     errorCode="XDSExtraMetadataNotSaved"
      codeContext="Extra Metadata - localinfo - not saved"
      location=""
      severity="urn:oasis:names:tc:ebxml-
2740     regrep:ErrorSeverityType:Warning"/>
  </rs:RegistryErrorList>
</rs:RegistryResponse>
```

An example response to a query request that reports two errors is:

```
2745 <query:AdhocQueryResponse
      status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure">
  <rs:RegistryErrorList
    highestSeverity="urn:oasis:names:tc:ebxml-
2750     regrep:ErrorSeverityType:Error">
    <rs:RegistryError
      errorCode="XDSStoredQueryMissingParam"
      codeContext="Required parameter $XDSDocumentEntryStatus is missing"
      location=""
      severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
    <rs:RegistryError
2755     errorCode="XDSStoredQueryParamNumber"
      codeContext="$XDSDocumentEntryServiceStartTimeFrom contains multiple
        entries and is restricted to a single entry"
      location=""
      severity="urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error"/>
2760   </rs:RegistryErrorList>
</query:AdhocQueryResponse >
```

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.

2765

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

Error Code ³	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. Specifically the parts not processed are Append semantics	P
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
UnresolvedReferenceException	The recipient cannot resolve an entryUUID reference in the transaction.	P, R
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 and XDSNonIdenticalSize.	P, R
XDSExtraMetadataNotSaved	This warning is returned if extra metadata was present but not saved.	P, R

³ The XDS Metadata Update Supplement adds error codes to this table.

Error Code ³	Discussion	Transaction (See Note 1)
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 Document Registry) but hash does not match. codeContext indicates UniqueId.	R
XDSNonIdenticalSize	Document being registered was a duplicate (uniqueId already in Document 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 Patient Id and the nature of the conflict.	P, R
XDSRegistry Busy XDSRepository Busy	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
XDSReplaceFailed <i>Note: This error code is deprecated. See ITI TF-2b: 3.42.4.1.3.5.</i>	Error detected by the Document Registry during a document replacement.	P, R
XDSResultNotSinglePatient	This error signals that the single Stored Query would have returned metadata for multiple Patient Ids	SQ, RS
XDSStoredQueryMissingParam	A required parameter to a stored query is missing.	SQ, XGQ

Error Code ³	Discussion	Transaction (See Note 1)
XDSStoredQueryParamNumber	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.	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 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
- RS = Retrieve Document Set
- XGQ = Cross Gateway Query
- XGR = Cross Gateway Retrieve

2770

Note 2:

Two examples of the use of error code XDSUnavailableCommunity are:

2775

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.

2780

4.2.4.2 Error responses

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.

2785

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

2790

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

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

2795 4.2.5 Metadata Vocabulary

The UUIDs in the following tables shall be used in constructing and interpreting Document Sharing 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:6b5aea1a-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

2800 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:75df8f67-9973-4fbc-a900-df66cefec5a	uniqueId External Identifier

4.3 Additional Document Sharing Requirements

2805 This section includes profile specific requirements.

4.3.1 Submission Metadata Attribute Optionality

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

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

2810

Table 4.3.1-1: 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

2815

For each actor/transaction pair across the top of Table 4.3.1-3, 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 4.3.1-2: 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.

Table 4.3.1-3: 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
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	healthcareFacility TypeCode	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	R
DocumentEntry	serviceStartTime	R2	R2	R2	R2	R2	O

Metadata Element	Metadata Attribute	XDS DS	XDS DR	XDM MC	XDR DS	XDR MS	XDS OD
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
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

2820

4.3.2 Requirements on Query Type Transactions

4.3.2.1 Query Type Metadata Attribute Optionality

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.

2825

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

2830 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.

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

Metadata Element	Metadata Attribute	XDS DR	XCA IG	XCA RG	MPQ DR
DocumentEntry	languageCode	R	R	R	R
DocumentEntry	legalAuthenticator	O	O	O	O
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

Metadata Element	Metadata Attribute	XDS DR	XCA IG	XCA RG	MPQ DR
Folder	patientId	R	R	R	R
Folder	title	R	R	R	R
Folder	uniqueId	R	R	R	R

2835 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

2840 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 Section 5.1.2 defines the requirements of the BPPC document without a scanned document part; Section 5.1.3 explains with a scanned document part.

5.1.1 References

- 2845
- 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

2850 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 Section 5.1.3.

5.1.2.1 XDS Metadata

5.1.2.1.1 XDS DocumentEntry Metadata

2855 BPPC leverages the XDS DocumentEntry Metadata requirements in 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 Acknowledgment Document” and the codeSystem is 2.16.840.1.113883.6.1.

For backward compatibility historic documents may be registered with:

- 2860
- 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 XSDDocumentEntry.eventCodeList

- eventCodeList -- the eventCodeList shall be populated using the Patient Privacy Policy Identifiers that have been acknowledged to within the document.
- 2865 • /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.
- 2870 • /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:bpsc:2007**. The formatCode codeSystem shall be **1.3.6.1.4.1.19376.1.2.3**.

5.1.2.1.1.4 XSDDocumentEntry.uniqueId

- 2875 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 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 PCC TF-2: 4.1.2

2880 5.1.2.1.3 XDS Folder Metadata

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

5.1.2.2 Specification

- 2885 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.

```

2890 <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=' ' />
2895 <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'
2900   codeSystem='2.16.840.1.113883.5.25' codeSystemName='Confidentiality' />
      <languageCode code='en-US' />
      :
      <component><structuredBody>
2905   </structuredBody></component>
</ClinicalDocument>
    
```

Figure 5.1.2.2-1: Sample Consent to Share Information Document

2910 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.

2915 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

2920 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.

```
2925 <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' />
2930        <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' />
        <effectiveTime>
          <low value='20101130' />
          </effectiveTime>
2935      </serviceEvent>
    </documentationOf>
```

Figure 5.1.2.2-2: Patient Privacy Acknowledgement Service Events Example

5.1.2.2.2 <documentationOf typeCode='DOC'>

2940 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'>

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' />

2945 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=' ' />

2950 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=' ' />

2955 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.

5.1.2.2.7 <effectiveTime><low value=' ' /><high value=' ' /></effectiveTime>

2960 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

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

2965 1.3.6.1.4.1.19376.1.5.3.1.1.7.1 – With Scanned Document

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

2970 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

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

2975 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

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

2980 5.1.3.1.3 XDS Folder Metadata

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

5.1.3.2 Specification

This BPPC document shall conform to the XDS-SD (Section 5.2) specification and shall have the additional requirements stated in Section 5.1.2.3.

2985 5.1.3.3 Conformance

See Section 5.1.2.2.

5.2 Scanned Documents Content Module

2990 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
2995 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
define the following nomenclature for entity roles concerned in forming the wrapper content.

- 3000 • *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.
- *(Scanner) Operator* – Person assembling the scanned content.

3005 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)
- HL7 CDA Release 2.0 (denoted HL7 CDA R2, or just CDA, in subsequent text)
- 3010 • [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
subset of Adobe PDF version 1.4 intended to be suitable for long-term preservation of page-
3015 oriented documents. PDF/A attempts to maximize:

- Device independence
- Self-containment
- Self-documentation

The constraints imposed by PDF/A include:

- 3020 • 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
 - Colorspaces specified in a device-independent manner
- 3025
- 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

3030 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.

3035 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)

3040 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.

3045 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.

HL7 CDA R2 header schema is constrained so that pertinent metadata values and scanning facility, technology and operator information shall be present (see Section 5.2.3).

3050 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

3055 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.

5.2.2.1.1 XDSDocumentEntry.formatCode

3060 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.uniquelid

3065 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

3070 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. Other specific examples may exist and this profile does not preclude the notion of a parent document for these cases.

3075

5.2.2.2 XDS SubmissionSet Metadata

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

3080 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/documentaionOf/serviceEvent/effectiveTime	R	5.2.3.8	SA / O	Denotes the time/date range of the original content.
ClinicalDocument/component/no nXMLBody	R	5.2.3.9	SA	The scanned/encoded content.

5.2.3.1 ClinicalDocument child-less elements

3085 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.
- 3090 • 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.
- 3095 • 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.
- 3100 • 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.
- 3105 • Attributes `confidentialityCode@code` and `confidentialityCode@codeSystem` shall be present.
- 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 Section 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.
- 3110

Example:

3115

```
<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"/>
</ClinicalDocument>
```

5.2.3.2 ClinicalDocument/recordTarget

3120

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.

3125

- 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.
- 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.
- At least one `ClinicalDocument/recordTarget/patientRole/patient/name` element shall be at least one given subelement and one family subelement.

3130

- 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.

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>
```

3135

5.2.3.3 ClinicalDocument/author (original)

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.

- 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.
- The ClinicalDocument/author/assignedAuthor/id element if known shall include both the root and 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 attribute. Refer to PCC TF-2: 4.1.1 for more details.

3150

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="aaaaabbbbb" root="1.3.5.35.1.4436.7"/>
      <name>Dr. Wiseman's Clinic</name>
    </representedOrganization>
  </assignedAuthor>
</author>
```

3155

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.

3160

- 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.
- 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.
- 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.

3165

3170

3175

- The ClinicalDocument/author/assignedAuthor/assignedAuthoringDevice/manufacturerModelName 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 manufacturerModelName referenced here refers to the makers of the technology that was used to produce the embedded content.
3180
- 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.
3185
- The ClinicalDocument/author/assignedAuthor/representedOrganization/id element shall be present. The root attribute shall be set to the oid of the scanning facility.

3190 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>
```

5.2.3.5 ClinicalDocument/dataEnterer

3195 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.

- 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.
- 3200 • 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.
- 3205 • 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>
```

5.2.3.6 ClinicalDocument/custodian

3210 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.

- The ClinicalDocument/assignedCustodian/representedOrganization/name shall be present.
- 3215 • At least one ClinicalDocument/assignedCustodian/representedOrganization/addr element shall include at least the country sub element.

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>
```

3220 5.2.3.7 ClinicalDocument/legalAuthenticator

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 attribute. 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>
```


3230 **5.2.3.8 ClinicalDocument/documentationOf**

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>
```

3240 **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.

- 3245
- 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.
- 3250
- 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.
- 3255

- 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.

3260 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//SSUswbFHx/JzBLEu5yYxOIze8bPcRWqdaGDmcZO
      BWc/9bfUNOPfOte4409jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
      K7CXIe2be+/1DzXQP+RlBmRzdHJlYW0KZW5kb2JqCjYgMCAvYmoKMjAxMjYyLjE1LjE1
      SW5mbyAyIDAgUgovSUQgWzZGNENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
      RjRDQzdBRUIlNEIzNkZCMjgzQzVDMzI0Nz1BRDI4M0Y+XQo+PgpzdGFydHhZWyKMzAx
      MjYyLjE1LjE1MgolJUVPRgo=
    </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//SSUswbFHx/JzBLEu5yYxOIze8bPcRWqdaGDmcZO
      BWc/9bfUNOPfOte4409jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
      K7CXIe2be+/1DzXQP+RlBmRzdHJlYW0KZW5kb2JqCjYgMCAvYmoKMjAxMjYyLjE1LjE1
      SW5mbyAyIDAgUgovSUQgWzZGNENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
      RjRDQzdBRUIlNEIzNkZCMjgzQzVDMzI0Nz1BRDI4M0Y+XQo+PgpzdGFydHhZWyKMzAx
      MjYyLjE1LjE1MgolJUVPRgo=
    </text>
  </nonXMLBody>
</component>
</ClinicalDocument>
```

3265 **5.2.4 Complete Example (Wrapped PDF)**

```
3265 <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">
3270 <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>
3275 <effectiveTime value="20050329224411+0500"/>
<confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
<languageCode code="en-US"/>
<recordTarget>
  <patientRole>
3280   <id extension="12345" root="2.16.840.1.113883.3.933"/>
   <addr>
     <streetAddressLine>17 Daws Rd.</streetAddressLine>
     <city>Blue Bell</city>
     <state>MA</state>
3285     <postalCode>02368</postalCode>
     <country>USA</country>
   </addr>
   <patient>
     <name>
3290       <prefix>Mrs.</prefix>
       <given>Ellen</given>
       <family>Ross</family>
     </name>
     <administrativeGenderCode code="F"
3295 codeSystem="2.16.840.1.113883.5.1"/>
     <birthTime value="19600127"/>
   </patient>
  </patientRole>
</recordTarget>
3300 <author>
  <templateId root="1.3.6.1.4.1.19376.1.2.20.1"/>
  <time value="19990522"/>
  <assignedAuthor>
3305   <id extension="11111111" root="1.3.5.35.1.4436.7"/>
   <assignedPerson>
     <name>
       <prefix>Dr.</prefix>
       <given>Bernard</given>
       <family>Wiseman</family>
3310       <suffix>Sr.</suffix>
     </name>
   </assignedPerson>
   <representedOrganization>
3315   <id extension="aaaaabbbbb" root="1.3.5.35.1.4436.7"/>
   <name>Dr. Wiseman's Clinic</name>
   </representedOrganization>
```

```

    </assignedAuthor>
  </author>
  <author>
3320 <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>
3325 <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>
3330 <representedOrganization>
        <id root="1.3.6.4.1.4.1.2835.2"/>
        <name>SOME Scanning Facility</name>
          <addr>
3335 <streetAddressLine>21 North Ave</streetAddressLine>
            <city>Burlington</city>
            <state>MA</state>
            <postalCode>01803</postalCode>
            <country>USA</country>
          </addr>
3340 </representedOrganization>
        </assignedAuthor>
      </author>
    <dataEnterer>
3345 <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>
3350 <name>
          <prefix>Mrs.</prefix>
          <given>Bernice</given>
          <family>Smith</family>
        </name>
        </assignedPerson>
3355 </assignedEntity>
      </dataEnterer>
    <custodian>
      <assignedCustodian>
        <representedCustodianOrganization>
3360 <id root="1.3.6.4.1.4.1.2835.2"/>
        <name>SOME Scanning Facility</name>
          <addr>
3365 <streetAddressLine>21 North Ave</streetAddressLine>
            <city>Burlington</city>
            <state>MA</state>
            <postalCode>01803</postalCode>
            <country>USA</country>
          </addr>
        </representedCustodianOrganization>

```

```
3370     </assignedCustodian>
        </custodian>
        <legalAuthenticator>
          <time value="19990522"/>
          <signatureCode code="S"/>
3375     <assignedEntity>
          <id extension="11111111" root="1.3.5.35.1.4436.7"/>
          <assignedPerson>
            <name>
              <prefix>Dr.</prefix>
3380              <given>Bernard</given>
              <family>Wiseman</family>
              <suffix>Sr.</suffix>
            </name>
            </assignedPerson>
3385          </assignedEntity>
        </legalAuthenticator>
        <documentationOf>
          <serviceEvent >
            <effectiveTime>
3390              <low value="19800127"/>
              <high value="19990522"/>
            </effectiveTime>
          </serviceEvent>
        </documentationOf>
3395     <component>
          <nonXMLBody>
            <text mediaType="application/pdf" representation="B64">
3400 JVBeri0xLjMKJcfsj6IKNSAwIG9iago8PC9MZw5ndGggNiAwIFivRmlsdGVyIC9GbGF0
ZURlY29kZT4+CnN0cmVhbQp4nGWPMWsDMQyFd/8KjFJwqmVbkr0GQqFbg7fQoSRNWuhB
Q/4/1L67TEEYme+9J1s3CMQQRm39NLUxg8H17gK89nN1N8eLAbZ2mmHXuql2QDVUhnZx
a5iBcyQtOmiUM7TZhbH5KZEVDgm//SSUswbFHx/JzBLEu5yYxOizE8bPcRWqdaGDmcZO
BwC/9bfUNOPfOte4409jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
K7CXIe2be+/1DzXQP+RlBmRzdHJlYW0KZW5kb2JqcjYgMCAvYmoKMjAxcmVuZG9iago0
...
3405 SW5mbyAyIDAgUgovSUQgWzxCNENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
RjRDQzdBRUI1NEIzNkZCMjgzQzVDMzI0NzI0BRDI4M0Y+XQo+PgpzdGFydHhyZWYKMzAx
MgolJUVPRgo=
          </text>
          </nonXMLBody>
3410     </component>
  </ClinicalDocument>
```

5.3 Intentionally Left Blank

3415

5.4 XDW Workflow Content Module

3420 This section defines the XDW Workflow Document by providing a schema and explaining its use. This document does not include clinical information about the patient directly. It shall only contain information necessary for organizing and defining work tasks. All clinical information regarding any task shall be provided through separate documents that are referenced from the associated input or output documents.

5.4.1 Referenced Standards

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

Web Services – Human Task (WS-HumanTask) Specification Version 1.1, OASIS

3425 5.4.2 Discussion of Content Standards

3430 The XDW Workflow Document is a document that incorporates elements from the HL7 CDA document structure and from the WS-HumanTask structure. The Workflow Document exists to coordinate the activities of multiple people in different organizations. They agree to share these documents as a method of exchanging work information. These documents are used by these organizations to feed what is often considered their own internal task management systems and have their own administrative rules for managing activities.

3435 Sharing clinical documents is often accomplished as a normal part of providing healthcare. The XDW workflow allows the work information to be shared in the same way as other patient related clinical information. Integrating the internal workflow management systems of independent organizations with independent administrative rules, and perhaps in different legal and regulatory systems, is avoided.

3440 The XDW Workflow Document does not contain clinical information about the patient. The input, output, and other elements of the task data shall contain references to documents (DocumentEntry.uniqueId) that contain the clinical information.

3440 XDW Workflow uses the XDS lifecycle management tools to coordinate updates to the Workflow Document instead of requiring an integration of all the different task management systems in the different organizations.

The XDW Workflow Document builds upon two other standards, HL7 CDA and OASIS WS-Human Task.

3445 The XDW Workflow Document shall comply with the XDW XML Schema that includes elements from the CDA and OASIS Human Task standards. Access to the schema files from those standards will be needed.

The figure below represents the main level structure of the Workflow Document with the first level of the elements that composed the structure.

3450 It is possible to divide the structured into four parts:

- Part 1: elements derived from HL7 CDA standard (Type of the element: CDA),

- Part 2: two elements, patient and author, defined in the XDWSchema with the structure derived from HL7 R-MIM standard (Type of the element: tXDWpatient and tXDWauthor),
- 3455
- Part 3: elements defined by IHE XDW Profile
 - Part 4: the element <TaskList> in which is defined by elements derived from the OASIS WS-HumanTask standard. In this last section the <TaskList> is a list of elements <XDWTask> composed of the HumanTask <taskData> (all data that define the XDWTask) and the HumanTask <taskEventHistory> that contains a list of
- 3460
- elements <taskEvent>.

All the elements of the Figure 5.4.2-1 are described in Section 5.4.3.

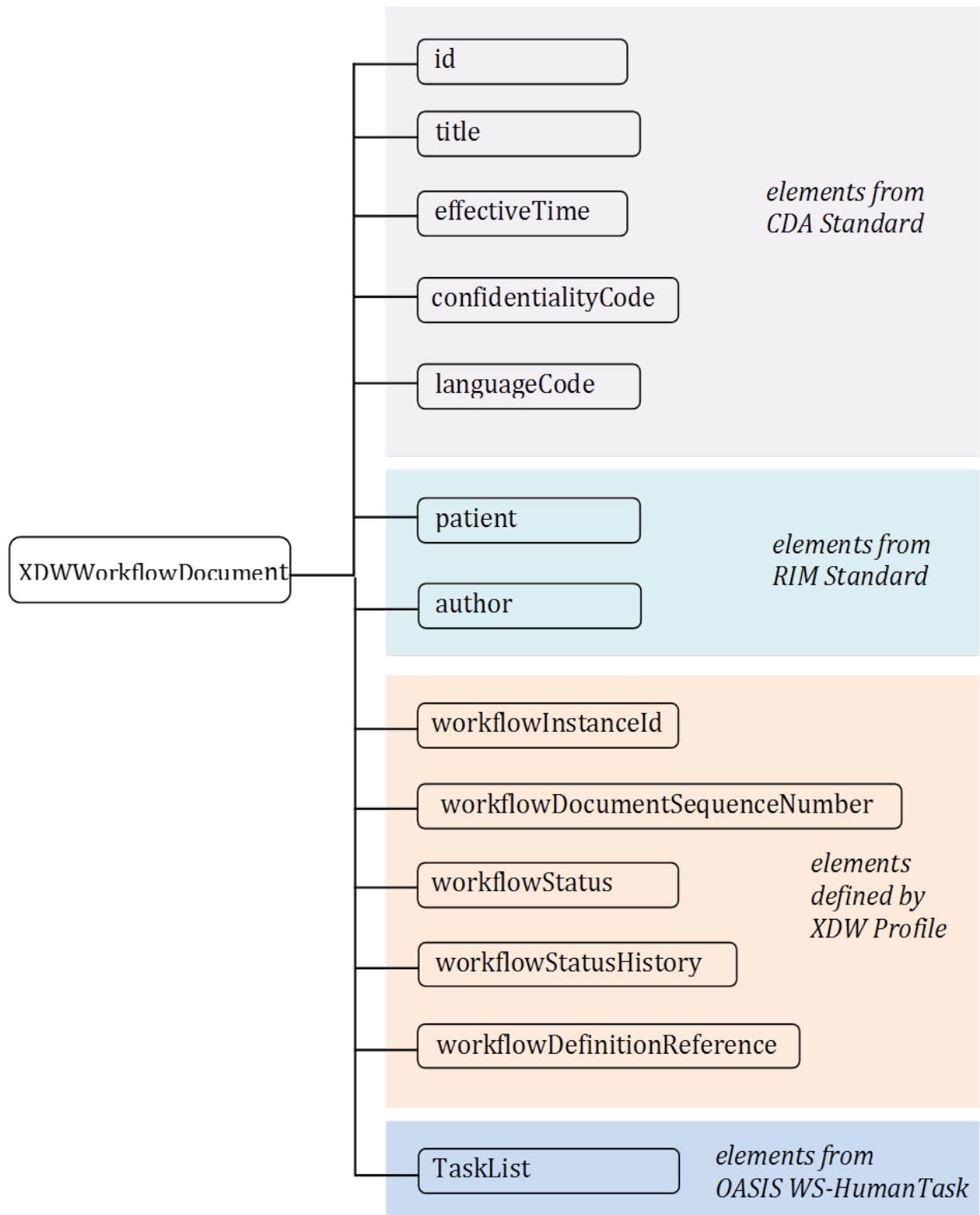


Figure 5.4.2-1: XDW.WorkflowDocument Structure

3465 **5.4.2.1 XDW Workflow Document Elements from HL7 CDA Standard**

Some elements are incorporated directly from the HL7 CDA standard. This means that the elements, their definitions, and the rules for interpreting them are in the HL7 standard. These are summarized here for convenience.

3470 <patient> and <author> elements have been defined based upon the HL7 CDA R-MIM. The XDW schema defines these elements using elements from CDA, and was derived by eliminating all elements that are not needed for workflow identification purposes. The R-MIM includes elements that are of clinical value. These have been removed for workflow use.

5.4.2.2 XDW Workflow Document Elements defined by IHE XDW Profile

The XDW Workflow Document also has elements that are defined by IHE (see Table 5.4.3-1):

- 3475 • <workflowInstanceId> Every version of the Workflow Document shall have the same workflowInstanceId value. This value shall be an OID. It is conveyed in the DocumentEntry.referenceIdList attribute of the workflow document's metadata. It shall be globally unique, because it is shared by many organizations.
- 3480 • <workflowDocumentSequenceNumber> This is used to simplify management of the changes to the Workflow Document as the workflow is executed. It shall be created as "1", and be incremented for each update to the Workflow Document.
- <workflowStatus> This shall be either
 - 3485 OPEN– which means that further updates are expected for this Workflow Document. These updates could be modifications to existing tasks or addition of new tasks or update to an existing task. Tasks shall not be deleted.
 - CLOSED– which means that further updates to this Workflow Document are not expected. A workflow with a CLOSED workflowStatus may continue to be updated, after which the value of workflowStatus may be transitioned back to OPEN or remain CLOSED. These constraints will be defined by the Workflow Definition referenced.
- 3490 • <workflowStatusHistory> This element represents the history of changes of status of the workflow document. It consists of sub-elements named documentEvent. Each documentEvent describes a change of status of the workflow document. In case that the workflowDefinitionReference describes a type of workflow that can't change its status from CLOSED to OPEN, the workflowStatusHistory contain at most two
- 3495 documentEvent elements, one for the opening of the workflow corresponding to the creation of the workflow document, and one to track the closing of the process related. Instead, if the workflowDefinitionReference permits the change of status from CLOSED to OPEN (e.g., OPEN-->CLOSED-->OPEN...) the element workflowStatusHistory will contain from 1 to N documentEvent elements to track these changes. A documentEvent is
- 3500 described by sub-elements defined in Table 5.4.3-5.

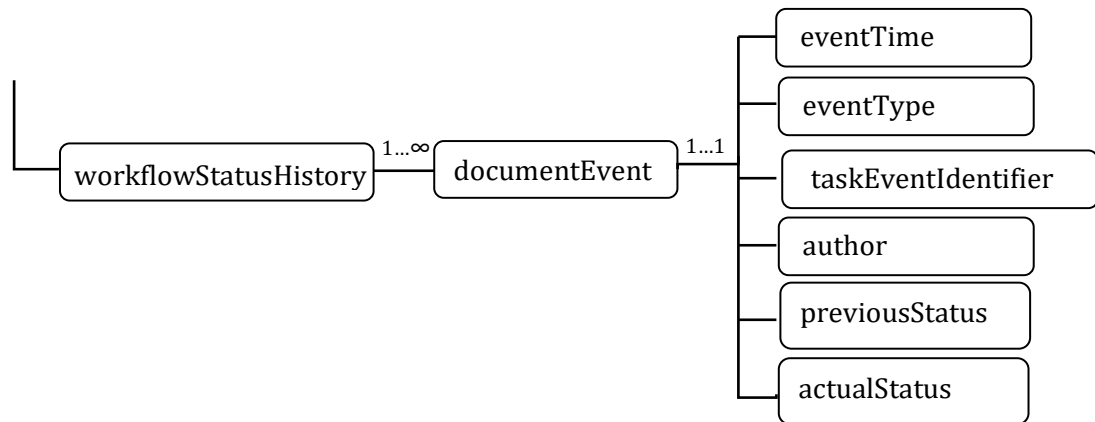


Figure 5.4.2.2-1: workflowStatusHistory Element

- 3505
- `<workflowDefinitionReference>`. This is the reference to the workflow definition. This is usually contained in policy or procedure document or may be defined by IHE as a specific workflow definition profile. This profile places no restriction on the style used to document such Workflow definition. It is recommended to assign an OID to those. It shall be recorded by the creator of the initial Workflow Document and shall be preserved unchanged in all subsequent versions of the document.
- 3510

5.4.2.3 XDW Workflow Document Elements from the OASIS Human Task

The descriptions of a task and of `<taskEvent>` are taken from the OASIS Human Task standard. This standard defines a way to describe a human task. It was defined as an extension to the BPEL and BPMN workflow standards. These standards are in use to manage the workflow of automated tasks under the control of an integrated task management system. It was recognized that while these standards do not have the ability to control human task, they needed a way to describe tasks to be performed by humans and other organizations.

3515

The element `<XDWTask>` groups all information about one task in the workflow, the `<XDWTask>` is structured in two sub elements: `<taskData>` and `<taskEventHistory>`.

- 3520
- `<taskData>` describes a single task. This is a list of details about the task, a description, the inputs to the task (e.g., documents), the outputs from the task (e.g., documents), fault descriptions and comments. The `<taskDetails>` include elements like the task ID, description, state, etc. (see Table 5.4.3-8)
 - `<taskEventHistory>` contains a list of the `<taskEvent>` elements that describe the changes of the task. For each task, there is one or more `<taskEvent>` that describes the history of the task. There is a list of the `<taskEvent>`: an `<eventType>`, a description, the inputs to the `<taskEvent>` (e.g., documents), the outputs from the `<taskEvent>` (e.g., documents), fault descriptions, comments, and
- 3525

3530 attachments (other documents that do not represent outputs). The details include elements like the task ID, status, etc. (see Table 5.4.3-10)

The definitions and rules such as the state machine that defines status are in the Human Task standard. There are other datatypes and web services also defined in Human Task standard that are not used by XDW.

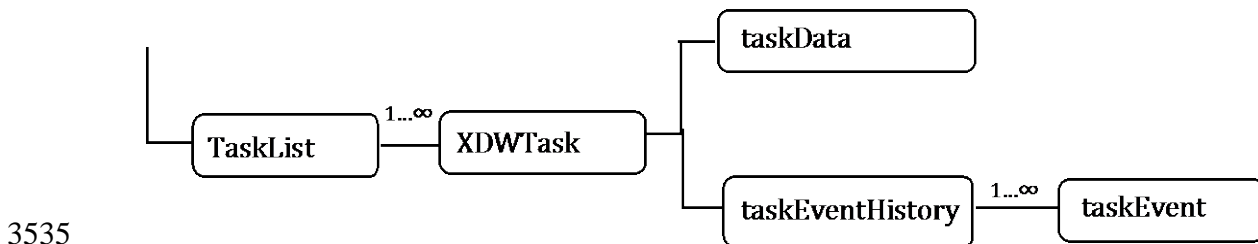


Figure 5.4.2.3-1: XDW Workflow elements derived from OASIS WS-HumanTask

5.4.2.4 Relationship between Task and <taskEvent>

3540 When a Task is generated it has a first <taskEvent>. A Task can either have only one <taskEvent> if the status of the task is not modifiable and it is born just completed or it can have more status and so more taskEvents. In this case at any time the task changed a new <taskEvent> is created.

3545 When a new Task is generated, zero or more references to external documents, associated with the Task, either as input or output, are put in the respective element of the Task. As a Task changes new input or output documents may be added (cumulative list of references). However, for each Task Event, only the input and output document related to the specific task Event shall be included. The inputs documents of a <taskEvent> are the documents that have been used as input for performing the Task change. The Output documents are those that have been created as a result of the Task Change. As a consequence, all input and output document references, 3550 present one or more times in the task Events list shall be listed (without duplicates) in the Task. Likewise for output document references.

3555 The clinical documents referenced in the input or output data elements of Tasks and task Events shall be accessible in the affinity domain (if XDW is used along with XDS) or Media Interchange (if used along with XDM) or Point-to-point submission set (if used along with XDR). In anticipation of the use of XDW in a cross-community environment, both the document uniqueId and homeCommunityId are permitted to be included.

The XDW Workflow Document defines a task list which is a series of task descriptions. The relationship between the task, the order of the elements in this list and the possible status of a task, all this rules are defined in the Workflow Definition Document.

3560 The XDW Profiles define the recommended statuses processable in a Task with the `<taskEvent>`. These statuses are a subset of the HumanTask Standard. There are other task status values possible, but these are not normally used.

Table 5.4.2.4-1: Description of Task Status

Task status	Description
CREATED	The workflow is open, the task is created but not assigned to an owner
READY	The task created is assigned to an owner and is ready to be performed
IN_PROGRESS	The task is started and the owner is performing the task actions
FAILED	The task is completed with fault response (it is not possible conclude the action of the task)
COMPLETED	The task is completed with response

3565

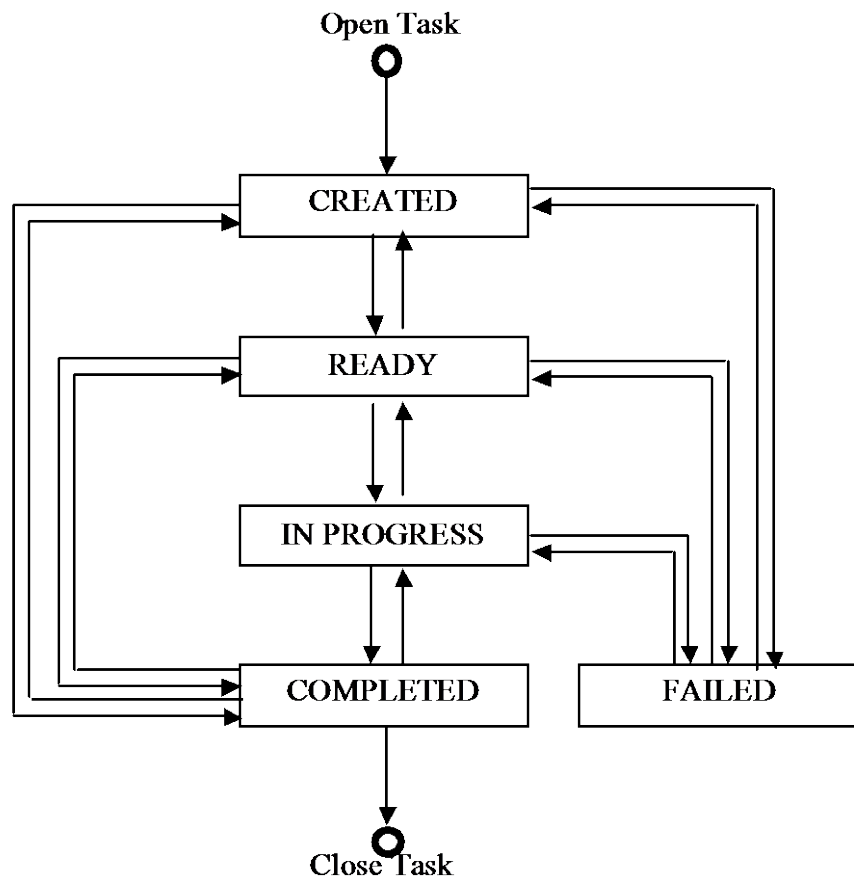


Figure 5.4.2.4-1: Task Status Transition

3570 The element `<XDWTask>` and XDW `<taskEvent>` is constrained by XDW with a minimal set of elements required. These elements are fully extensible with any kind of attributes defined by Human Task standard. This allows specific Workflow Definition profiles to add elements defined in Human Task to manage for example intertask relationships, additional status, etc. to address more advanced specific workflow requirements.

3575 **5.4.3 Content Specification**

The tables below represent all Workflow Document elements. The tables show for each element the Optionality and the standard from which the definition and the structure of the element derive.

Optionality:

3580 R = element Required for XDW Profile

R2 = element Required if known for XDW Profile

O = element Optional for XDW Profile

X = element shall not be used

3585 Inside the tables the column description is used to constrain the use of the attribute when referring to element defined in the underlining standard. When the description in blank no constrains is required. When the element is defined by XDW this is the complete description.

There are three functional roles for interacting with these elements.

- The "create" role specifies what elements shall be created. The Content Creator is permitted to include any optional element, and may include other elements.
- 3590 • The "view" role specifies what elements shall be presented by Content Consumer or Content Updater that support viewing of the document. It may present for viewing any other element that it understands or has a means of presenting. There are elements that are required for viewing, while being optional for both creation and viewing.
- 3595 • The "update" role specifies what elements shall be maintained with correct values when updating a document. An "update" operation shall preserve the value of all elements that are present, even if their meaning is unknown. This means that an updater might not update the contents of optional elements when updating a workflow document.

If one of the following tables does not specify separate values for the three roles, then the specified value applies to all three roles.

3600 Tables uses the following namespace conventions:

- cda="urn:hl7-org:v3";
- ht="http://docs.oasis-open.org/ns/bpe4people/ws-humantask/types/200803"
- xs="http://www.w3.org/2001/XMLSchema"
- xdw="urn:ihe:iti:xdw:2011"

3605

- <XDW.WorkflowDocument>

Table 5.4.3-1: Elements of the Workflow Document

XDW.WorkflowDocument element	Standard	Data Type	Optionality	Description
id	HL7 CDA	cda:II	R	Document ID
title	HL7 CDA	cda:ST	O	Displayable title
effectiveTime	HL7 CDA	cda:TS	R	Time of most recent update
confidentialityCode	HL7 CDA	cda:CE	R	
languageCode	HL7 CDA	cda:CS	O	

XDW.WorkflowDocument element	Standard	Data Type	Optionality	Description
patient	HL7 CDA	xdw:tX DWpatient	R	Patient information derived from R-MIM. Restricted to non-clinical necessary content. See Table 5.4.3-2
author	HL7 CDA	xdw:tX DWAutor	R	Author information derived from R-MIM. Restricted to non-clinical necessary content. See Table 5.4.3-3
workflowInstanceId	IHE	OID	R	Conveys the workflow identifier. It shall contain the same value as the CXi.1 component of the DocumentEntry.referenceList metadata attribute.
workflowDocumentSequenceNumber	IHE	xs:int	R	
workflowStatus	IHE	xs:token	R	OPEN if modifications are permitted to the document contents. CLOSED if modifications are not expected.
workflowStatusHistory	IHE	xdw:work flow Status History _type	R	List of changes of the workflowStatus See Table 5.4.3-4
workflowDefinitionReference	IHE	xs:any URI	R	References (urn:OID:) to the documents that define this kind of workflow.
TaskList	IHE	xdw:Ta skList _type	R	List of all tasks and their history See Table 5.4.3-6

3610 • <patient>

Table 5.4.3-2: Patient Element

Patient element	Standard	Data Type	Optionalit y	Description
id	HL7 CDA	cda:II	R	
name	HL7 CDA	cda:PN	O	
administrativeGenderCode	HL7 CDA	cda:CE	O	
birthTime	HL7 CDA	cda:TS	O	
maritalStatusCode	HL7 CDA	cda:CE	O	

- <author>

3615

Table 5.4.3-3: Author Element

Author element	Standard	Data Type	Optionalit y	Definition
assignedAuthor	HL7 CDA	cda:PO CD_MT0 00040. AssignedAuth or	R	Either assignedAuthoringDe vice or assignedPerson should be specified

- <workflowStatusHistory>

3620

Table 5.4.3-4: workflowStatusHistory Element

TaskList element	Standard	Data Type	Optionalit y	Description
documentEvent	IHE	xdw:tX DWdocu mentEv ent_ty pe	R	A detailed event that represents a change of the workflowStatus The first documentEvent element is added when the workflow document is created. A documentEvent element is then added whenever the workflowStatus of the workflow document changes.

TaskList element	Standard	Data Type	Optionalit y	Description
				See Table 5.4.3-5

- <documentEvent>

Table 5.4.3-5: documentEvent Element

documentEvent element	Standard	Data Type	Optionalit y	Description
eventTime	OASIS_WS-HumanTask	xs:dateTime	R	Time when the specific documentEvent element is added to the workflow document.
eventType	OASIS_WS-HumanTask	ht:tTaskEventtype	R	The type of event that happens that solicits the modification of the workflowStatus. It should be valorized with one of these types: create, stop, suspend, resume, fail, complete. These types are defined in the HumanTask specification (C. WS-HumanTask Data Types Schema, <!-- Defines the human task event types -->).
taskEventIdentifier	IHE	xs:anyURI	R	Element that permits to track the reference to the taskEvent that solicits the modification of the workflowStatus. It stores the same value of the element taskEvent/identifier of the taskEvent of reference.
author	IHE	xs:string	R	Actual owner of the workflow after the event
previousStatus	IHE	xs:token	R	The previous value of workflowStatus.

documentEvent element	Standard	Data Type	Optionality	Description
				Either "OPEN" or "CLOSED". In case of a Workflow Document just created this element shall be valorized with ""
actualStatus	IHE	xs:token	R	Equal to the current value of the workflowStatus element. Either "OPEN" or "CLOSED".

3625

- <TaskList>

Table 5.4.3-6: TaskList Element

TaskList element	Standard	Data Type	Optionality	Description
XDWTask	IHE	xdw:tXDWTask	R	List of tasks See Table 5.4.3-7

3630

- <XDWTask>

Table 5.4.3-7: XDWTask Element

XDWTask element	Standard	Data Type	Optionality	Description
taskData	OASIS_WS-HumanTask	ht:tTaskInstanceData	R	Description of the current task (status, inputs, outputs, etc.) See Table 5.4.3-8
taskEventHistory	IHE	xdw:tXDWEventHistory	R	History of the changes to the current task (dates, changes, etc.) See Table 5.4.3-11

- <taskData>

3635 The XDW Profile adds the following restrictions to the OASIS definition for taskDetails:

- 3640 • The taskData/input shall contain a taskData/input/part for every clinical document or workflow that is to be used as input to the task. This element is of type tMessagePartsData. An element <part> shall have a child element <AttachmentInfo> of type tAttachmentInfo. Table 5.4.3-9 describes how to assign values to each AttachmentInfo child elements.
- 3645 • The taskData/output shall contain a taskData/output/part for every clinical document or workflow that is created as a result of the task that is to be shared. This element is of type tMessagePartsData. An element <part> shall have a child element <AttachmentInfo> of type tAttachmentInfo. Table 5.4.3-9 describes how to assign values to each AttachmentInfo child element.
- Any clinical documents that are registered in an XDS Document Registry shall be identified in the taskData/input/part, taskData/output/part, or taskData/attachmentInfos/info as described in Table 5.4.3-9.

3650 The element <part> shall have an attribute @name. The value of this attribute identifies the role played by the referenced object within the task. A Workflow Definition profile shall define a list of acceptable values for this attribute. If no Workflow Definition profile is supported and if no values are defined by local policies, this value shall be set to “XDSRegisteredDocument”.

Table 5.4.3-8: taskData Element

taskData element	Standard	Data Type	Optionality			Description
			Create	View	Update	
taskDetails	OASIS_WS - HumanTask	ht:tTaskDetails	R	R	R	See Table 5.4.3-10
description	OASIS_WS - HumanTask	xs:string	R	R	R	Textual description
input	OASIS_WS - HumanTask	ht:tMessagePartsData	R	R	R	This element lists documents/workflows referenced by the task as inputs, using a child <part> elements for each document/workflow.
output	OASIS_WS - HumanTask	ht:tMessagePartsData	R	R	R	This element lists documents/workflows referenced by the task as outputs, using a child <part> elements for each document/workflow.
fault	OASIS_WS - HumanTask	ht:tFaultData	O	R	O	Description of fault
comments	OASIS_WS	xs:string	O	R	O	Simple text comments

taskData element	Standard	Data Type	Optionality			Description
			Create	View	Update	
	- HumanTask	g				about the task

3655

- <AttachmentInfo>

Each document referenced in input or output elements is structured using a tAttachmentInfo data type. The XDW Profile extends this data type, adding a new optional child element (homeCommunityId) that can be used to convey the home community Id of the referenced document. The structure of the <AttachmentInfo> element is described in Table 5.4.3-9.

3660

An <AttachmentInfo> element can be used to refer to another workflow. An <AttachmentInfo> element that stores a reference to a child or parent workflow shall contain an accessType with the value urn:ihe:iti:xdw:2013:workflowInstanceId.

3665

Table 5.4.3-9: AttachmentInfo Element

AttachedInfoelement	Standard	Data Type	Optionality			Description
			Create	View	Update	
identifier	OASIS_WS-HumanTask	xs:anyURI	R	R	R	<p>If the accessType is urn:ihe:iti:xdw:2011:XDSr egistered, the identifier shall contain the value of DocumentEntry.uniqueId</p> <p>If the accessType is urn:ihe:iti:xdw:2013:workf lowInstanceId, the identifier shall contain the value of the DocumentEntry.referenceI dList in the referenced workflow. See Note 1.</p>
name	OASIS_WS-HumanTask	xs:string	R	R	R	Stores the same value of the part/@name attribute

AttachedInfoelement	Standard	Data Type	Optionality			Description
			Create	View	Update	
accessType	OASIS_WS-HumanTask	xs:string	R	R	R	<p>If the attachment is a document, the value of accessType shall be urn:ihe:iti:xdw:2011:XDSregistered.</p> <p>If the part element references another workflow, the value of accessType shall be urn:ihe:iti:xdw:2013:workflowInstanceId. See Note 1.</p>
contentType	OASIS_WS-HumanTask	xs:string	R	R	R	Conveys the MIME type of the referenced document. If the attachment refers to a child/parent workflow then this element shall be empty.
contentCategory	OASIS_WS-HumanTask	xs:anyURI	R	R	R	Fixed value http://www.iana.org/assignments/media-types
attachedTime	OASIS_WS-HumanTask	xs:dateTime	R	R	R	The date/time when the document is attached as reference
attachedBy	OASIS_WS-HumanTask	ht:tUser	R	R	R	The owner that attached the reference to the task
homeCommunityId	IHE	OID	O	O	O	The home community Id of the referenced document

Note 1: The XDW Profile allows for reference to objects other than XDS documents or XDW Workflows. In this case the <identifier> element identifies the uid of the referenced object. The <accessType> of this referenced objects shall be “URL”. No further constraints are defined for other elements.

3670

- <taskDetails>

Table 5.4.3-10: taskDetails Element

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	
id	OASIS_WS - HumanTask	xs:anyURI	R	R	R	Internal ID for the task
taskType	OASIS_WS - HumanTask	xs:string	R	R	R	
name	OASIS_WS - HumanTask	xs:QName	R	R	R	The name of the task
status	OASIS_WS - HumanTask	ht:tStatus	R	R	R	Recommend limiting values to the statuses described above.
priority	OASIS_WS - HumanTask	ht:tPriority	O	R	O	
taskInitiator	OASIS_WS - HumanTask	ht:tUser	O	O	O	
taskStakeholders	OASIS_WS - HumanTask	ht:tOrganizationalEntity	O	O	O	
potentialOwners	OASIS_WS - HumanTask	ht:tOrganizationalEntity	O	O	O	Owners in Human Task terminology are people/organizations/ etc. that perform the task.
businessAdministrators	OASIS_WS - HumanTask	ht:tOrganizationalEntity	O	O	O	
actualOwner	OASIS_WS - HumanTask	ht:tUser	R	R	R	The actual performer of the task.

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	
notificationRecipients	OASIS_WS - HumanTask	ht:tOrganizationalEntity	O	R	O	Notification Recipient may be used to contain information about persons to be notified. Use of this element does not imply that Human Task "notification" will be used. This element may be used to trigger notification mechanisms outside of XDW (e.g., IHE DSUB Profile). This combined use is not part of this profile specification
createdTime	OASIS_WS - HumanTask	xs:dateTime	R	R	O	
createdBy	OASIS_WS - HumanTask	ht:tUser	R	R	O	
lastModifiedTime	OASIS_WS - HumanTask	xs:dateTime	R (Note 1)	R	R	
lastModifyBy	OASIS_WS - HumanTask	ht:tUser	O	R	R	
activationTime	OASIS_WS - HumanTask	xs:dateTime	O	R	O	
expirationTime	OASIS_WS - HumanTask	xs:dateTime	O	R	O	
isSkipable	OASIS_WS - HumanTask	xs:boolean	O	R	O	
hasPotentialOwners	OASIS_WS - HumanTask	xs:boolean	O	O	O	
startedByTimeExists	OASIS_WS - HumanTask	xs:boolean	X	X	X	
completedByTimeExists	OASIS_WS - HumanTask	xs:boolean	X	X	X	

taskDetails element	Standard	Data Type	Optionality			Description
			Create	View	Update	
presentationName	OASIS_WS - HumanTask	ht:tPresentationName	O	O	O	
presentationSubject	OASIS_WS - HumanTask	ht:tPresentationSubject	O	O	O	
renderingMethodExists	OASIS_WS - HumanTask	xs:boolean	R	R	R	Value shall be "false"
hasOutput	OASIS_WS - HumanTask	xs:boolean	X	X	X	
hasFault	OASIS_WS - HumanTask	xs:boolean	X	X	X	
hasAttachments	OASIS_WS - HumanTask	xs:boolean	X	X	X	
hasComments	OASIS_WS - HumanTask	xs:boolean	X	X	X	
escalated	OASIS_WS - HumanTask	xs:boolean	O	R	O	
searchBy	OASIS_WS - HumanTask	xs:string	X	X	X	
outcome	OASIS_WS - HumanTask	xs:string	X	X	X	
parentTaskId	OASIS_WS - HumanTask	xs:anyURI	X	X	X	XDW prohibits use of subTasks
hasSubTasks	OASIS_WS - HumanTask	xs:boolean	X	X	X	XDW prohibits use of subTasks.

3675

Note 1: lastModifiedTime shall be the same as createdTime

- <taskEventHistory>

Table 5.4.3-11: taskEventHistory Element

taskEventHistory element	Standard	Data Type	Optionality	Description
taskEvent	OASIS_WS-HumanTask	ht:taskEvent_type	R	See Table 5.4.3-12

3680 • <taskEvent>

Table 5.4.3-12: taskEvent Element

taskEvent element	Standard	Data Type	Optionality	Description
id	OASIS_WS-HumanTask	xs:integer	R	
eventTime	OASIS_WS-HumanTask	xs:dateTime	R	
identifier	OASIS_WS-HumanTask	xs:anyURI	R	
principal	OASIS_WS-HumanTask	xs:string	O	
eventType	OASIS_WS-HumanTask	ht:tTaskEventType	R	The type of event that happens that solicits the modification of the status of the task (adding a new taskEvent). It should be valorized with one of these types: create, stop, suspend, resume, fail, complete. These types are defined in the HumanTask specification (C. WS-HumanTask Data Types Schema, <!-- Defines the human task event types -->).
startOwner	OASIS_WS-HumanTask	xs:string	O	
endOwner	OASIS_WS-HumanTask	xs:string	O	
status	OASIS_WS-HumanTask	ht:tStatus	R	
hasData	OASIS_WS-HumanTask	xs:Boolean	O	

taskEvent element	Standard	Data Type	Optionality	Description
eventData	OASIS_WS-HumanTask	xs:any Type	R2	This structure includes the data elements that were changed by this event.
faultName	OASIS_WS-HumanTask	xs:string	0	

5.4.4 Complete Example

3685 In the example in Figure 5.4.4-1 represents the XML of the XDW Workflow Document for the use case described in ITI TF-1: 30.4.2.1. This example represents the complete Workflow Document at the end of the process (see column C of ITI TF-1: Figure 30.4.2.1.1-1).

In this case there are two tasks:

- 3690 • the first task has been created in status “COMPLETED” and so it has only one taskEvent in the taskEventHistory;
- the second task ends the process in status “COMPLETED” and it has two taskEvent.

```

3695 <?xml version="1.0" encoding="UTF-8"?>
<xdw:XDW.WorkflowDocument xmlns:hl7="urn:hl7-org:v3"
  xmlns:ws-ht="http://docs.oasis-open.org/ns/bpel4people/ws-
humantask/types/200803"
  xmlns:xdw="urn:ihe:iti:xdw:2011"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3700 xsi:schemaLocation="urn:ihe:iti:xdw:2011 XDW-2014-12-23.xsd">
  <xdw:id root="1.2.3.4.5"/>
  <xdw:effectiveTime value="20110401031520"/>
  <xdw:confidentialityCode code="1.24.3.3.3"/>
  <xdw:patient>
3705   <xdw:id root="1.3.6.1.4.1.21367.13.20.1000" extension="33333"
     assigningAuthorityName="IHERED"/>
  </xdw:patient>
  <xdw:author>
    <xdw:assignedAuthor>
3710     <hl7:id root="1.2.3.4.5" extension="11111"/>
     <hl7:assignedPerson>
       <hl7:name>
         <hl7:family>Blum</hl7:family>
         <hl7:prefix>Dr.</hl7:prefix>
       </hl7:name>
3715     </hl7:assignedPerson>
    </xdw:assignedAuthor>
  </xdw:author>
  <xdw:workflowInstanceId>1.2.3.4</xdw:workflowInstanceId>

```

```

3720 <xdw:workflowDocumentSequenceNumber>3</xdw:workflowDocumentSequenceNumber>
      <xdw:workflowStatus>CLOSED</xdw:workflowStatus>
      <xdw:workflowStatusHistory>
        <xdw:documentEvent>
3725           <xdw:eventTime>2011-03-28T10:00:12.0Z</xdw:eventTime>
           <xdw:eventType>create</xdw:eventType>
           <xdw:taskEventIdentifier>
urn:oid:1.2.3.4.5</xdw:taskEventIdentifier>
           <xdw:author>Mr. Rossi</xdw:author>
           <xdw:previousStatus/>
3730           <xdw:actualStatus>OPEN</xdw:actualStatus>
        </xdw:documentEvent>
        <xdw:documentEvent>
           <xdw:eventTime>2011-04-01T03:15:20.0Z</xdw:eventTime>
3735           <xdw:eventType>complete</xdw:eventType>
           <xdw:taskEventIdentifier>
urn:oid:1.2.3.4.7</xdw:taskEventIdentifier>
           <xdw:author>Dr. Brum</xdw:author>
           <xdw:previousStatus>OPEN</xdw:previousStatus>
           <xdw:actualStatus>CLOSED</xdw:actualStatus>
3740         </xdw:documentEvent>
      </xdw:workflowStatusHistory>

      <xdw:workflowDefinitionReference>urn:oid:1.2.3.4.5.6.7.8.9</xdw:workflowDefi
3745 nitionReference>
      <xdw:TaskList>
        <xdw:XDWTask>
          <xdw:taskData>
            <ws-ht:taskDetails>
3750               <ws-ht:id>1</ws-ht:id>
               <ws-ht:taskType>Requested</ws-ht:taskType>
               <ws-ht:name>ReferralRequested</ws-ht:name>
               <ws-ht:status>COMPLETED</ws-ht:status>
               <ws-ht:actualOwner>Mr. Rossi</ws-ht:actualOwner>
3755               <ws-ht:createdTime>2011-03-28T10:00:12.0Z</ws-
ht:createdTime>
               <ws-ht:createdBy>Mr. Rossi</ws-ht:createdBy>
               <ws-ht:lastModifiedTime>2011-03-28T10:00:12.0Z</ws-
ht:lastModifiedTime>
               <ws-ht:renderingMethodExists>false</ws-
3760 ht:renderingMethodExists>
            </ws-ht:taskDetails>
            <ws-ht:description>Request for a specialist visit</ws-
ht:description>
            <ws-ht:input/>
3765             <ws-ht:output/>
          </xdw:taskData>
          <xdw:taskEventHistory>
            <xdw:taskEvent>
3770               <xdw:id>l01</xdw:id>
               <xdw:eventTime>2011-03-28T10:00:12.0Z</xdw:eventTime>
               <xdw:identifier>urn:oid:1.2.3.4.5</xdw:identifier>
               <xdw:eventType>create</xdw:eventType>

```

```

        <xdw:status>COMPLETED</xdw:status>
    </xdw:taskEvent>
3775 </xdw:taskEventHistory>
    </xdw:XDWTask>
    <xdw:XDWTask>
        <xdw:taskData>
3780 <ws-ht:taskDetails>
            <ws-ht:id>2</ws-ht:id>
            <ws-ht:taskType>Referral Referred</ws-ht:taskType>
            <ws-ht:name>Referred</ws-ht:name>
            <ws-ht:status>COMPLETED</ws-ht:status>
            <ws-ht:actualOwner>Dr. Brum</ws-ht:actualOwner>
3785 <ws-ht:createdTime>2011-03-29T09:20:01.0Z</ws-
ht:createdTime>
            <ws-ht:createdBy>Dr. Brum</ws-ht:createdBy>
            <ws-ht:lastModifiedTime>2011-04-01T03:15:20.0Z</ws-
3790 ht:lastModifiedTime>
            <ws-ht:renderingMethodExists>false</ws-
ht:renderingMethodExists>
            </ws-ht:taskDetails>
            <ws-ht:description>Specialist visit</ws-ht:description>
            <ws-ht:input>
3795 <!-- one part element for each document in input -->
                <ws-ht:part name="eReferralDoc1">
                    <ws-ht:attachmentInfo>
3800 <ws-ht:identifier>1.2.3.4.56.7.78</ws-
                    <ws-ht:name>eReferralDoc1</ws-ht:name>
                    <ws-ht:accessType>urn:ihe:iti:
xdw:2011:XDSregistered</ws-ht:accessType>
                    <ws-ht:contentType>application/pdf</ws-
3805 ht:contentType>
                    <ws-
ht:contentTypeCategory>http://www.iana.org/assignments/media-types</ws-
ht:contentTypeCategory>
                    <ws-ht:attachedTime>2011-04-
3810 01T03:15:20.0Z</ws-ht:attachedTime>
                    <ws-ht:attachedBy>Dr. Brum</ws-
ht:attachedBy>
                </ws-ht:part>
            </ws-ht:input>
            <xdw:homeCommunityId>urn:oid:1.2.3.4.5</xdw:homeCommunityId>
            </ws-ht:attachmentInfo>
3815 <!--eReferralDoc1-->
            </ws-ht:part>
            </ws-ht:input>
            <ws-ht:output>
3820 <!-- one documentReference element for each document in
            input -->
                <ws-ht:part name="ChildWorkflow">
                    <ws-ht:attachmentInfo>
3825 <ws-ht:identifier>1.2.3.4.12312.34</ws-
                    <ws-ht:name>ChildWorkflow</ws-ht:name>

```

```
3830         <ws-
ht:accessType>urn:ihe:iti:xdw:2013:workflowInstanceId</ws-ht:accessType>
         <ws-ht:contentType>application/xml</ws-
ht:contentType>
3835         <ws-
ht:contentCategory>http://www.iana.org/assignments/media-types</ws-
ht:contentCategory>
         <ws-ht:attachedTime>2011-04-01T03:15:20.0Z</ws-
3840         <ws-ht:attachedBy>Dr. Brum</ws-ht:attachedBy>
         </ws-ht:attachmentInfo>
         </ws-ht:part>
         </ws-ht:output>
3845     </xdw:taskData>
     <xdw:taskEventHistory>
         <xdw:taskEvent>
             <xdw:id>201</xdw:id>
             <xdw:eventTime>2011-03-29T09:20:01.0Z</xdw:eventTime>
             <xdw:identifier>urn:oid:1.2.3.4.6</xdw:identifier>
3850             <xdw:eventType>create</xdw:eventType>
             <xdw:status>IN_PROGRESS</xdw:status>
             </xdw:taskEvent>
             <xdw:taskEvent>
                 <xdw:id>202</xdw:id>
                 <xdw:eventTime>2011-04-01T03:15:20.0Z</xdw:eventTime>
                 <xdw:identifier>urn:oid:1.2.3.4.7</xdw:identifier>
                 <xdw:eventType>complete</xdw:eventType>
                 <xdw:status>COMPLETED</xdw:status>
3855             </xdw:taskEvent>
         </xdw:taskEventHistory>
     </xdw:XDWTask>
</xdw:TaskList>
</xdw:XDW.WorkflowDocument>
```

Figure 5.4.4-1: Sample XDW Workflow Document

3860

5.4.5 Workflow Document Management

5.4.5.1 Workflow Document Lifecycle Management

The Cross-Enterprise Document Workflow Profile takes advantage of the lifecycle management of XDS when used in an XDS environment.

3865 A Workflow Document shall be created and be assigned a workflow identifier. The initial document shall include at least one task on the TaskList, and have a workflowStatus of OPEN. The Workflow Document is updated when:

- The information about a task is modified. This may be due to a change in some other task related information like updating the output information.
 - A new task is added to the <TaskList>.
- 3870

- The workflow status is changed to CLOSED.

Each update shall be done using the XDS Document Replace when in an XDS environment. The series of steps to be taken is:

- 3875 • Update the XDW document to reflect the desired changes. This is often replacement of the `<TaskData>`. It may also be a change by adding a new task to the `<TaskList>` or a new `<taskEvent>` to a Task.
- Use the XDS Replace operation to replace the old document with this modified document. This replacement document shall carry the same workflow identifier as the original Workflow Document.
- 3880 • It is possible that a document replace will be rejected by the XDS Document Registry if another actor has also done a replace in the time since the Workflow Document instance was obtained. In this case (attempting to replace a document already replaced), the XDW Document Creator or Updater shall obtain the most recent version of the Workflow Document which was updated by another XDW Actor, consider the evolution of the workflow, and performed a new content update. This kind of race condition should be very rare because updating is much faster than the rate at which people perform tasks. If certain workflows definitions require reducing the likelihood of such race conditions, one should consider placing in the Workflow Description one or more tasks "In Progress" and requiring that other Actor wait while such tasks are in-progress.

3890 When using XDR or XDM, the receiving actor shall perform an equivalent local update process.

When an XDW Actor decides that a workflow status code shall be placed in a CLOSED status, a final update to set the workflow status code to CLOSED shall be performed. The specific rules for determining when and which XDW Actors are allowed or should set the workflow status code to CLOSED are not specified by the XDW Profile. They may be determined within the Workflow Definition. XDW Content Consumer and Content Updater Actors shall support the means to query for Workflow Documents that are in a workflow status OPEN.

3900 This profile does not further constrain the rules for document lifecycle management, but a specific Workflow Definition may add requirements requiring that certain kinds of tasks be created initially, restricting the kinds of tasks that can be added, and requiring that updates be performed for specific task changes.

5.4.5.2 Associations Types

3905 A clinical document can be referenced by many Workflow Documents in different steps and for different reasons. When the content of a Workflow Document is known, the related clinical documents are always reached through the references (DocumentEntry.uniqueId and homeCommunityId) tracked inside the different task in the "input" and "output" elements.

The use of a workflow identifier is necessary to have a fixed id to identify the whole workflow. Since the Workflow Document will be replaced many times (it is replaced at each step), its DocumentEntry.uniqueId metadata attribute is not useful for maintaining a fixed reference. The

3910 document `uniqueId` of each of the successive XDW documents can be used to identify a particular state of the workflow.

XDW uses a workflow identifier stored in the `DocumentEntry.referenceIdList` metadata attribute of each workflow document to group all versions of the workflow document.

- The Content Creator shall create a workflow identifier, as an OID, when a new workflow is created.
- 3915 • The Content Creator shall create a single value in `DocumentEntry.referenceIdList` containing the workflow identifier. Only the CXi.1 and CXi.5 component shall be present.

An example workflow identifier in `DocumentEntry.referenceIdList` is:

```
2.16.840.1^^^^urn:ihe:iti:xdw:2013:workflowInstanceId
```

- 3920 • The Content Updater shall use the same value for the workflow identifier when it creates a new version of the Workflow Document.

Since every version of the Workflow Document replaces the previous, there is always one and only one approved document with a given workflow identifier.

3925 If a workflow generates another workflow there shall be two different workflow identifiers, one for each workflow. The relationship between the different workflows is always guaranteed to be inside the Workflow Documents using the `DocumentEntry.referenceIdList` as output of the task of the parent Workflow Document and as the input of the first task in the child Workflow Document.

5.4.5.3 Create workflow

3930 When the first step of a new workflow is completed, the XDW Content Creator shall:

- create the first version of the Workflow Document.

Then the XDW Content Creator shall use `Provide` and `Register Document Set-b` [ITI-41] (in the case of XDS):

- 3935 • submit the Workflow Document to the XDS Document Repository, using a new workflow identifier in the document's `DocumentEntry.referencedIdList` metadata attribute, using ITI-41 `Provide` and `Register Document Set-b`.

5.4.5.4 Update Workflow Document

For each subsequent step in the Workflow an XDW Content Updater shall:

- 3940 • obtain the most recent version of the Workflow Document, the only version approved_ with the specific workflow identifier in the `DocumentEntry.referenceIdList` (e.g., using a grouped XDS Document Consumer)
- update the content in the Workflow Document (by adding a new task or updating an existing task with a new `<taskEvent>`);

- 3945
- re-register (update) the Workflow Document by performing a document replace (e.g., in a XDS environment using a grouped XDS Document Source).

This new version of the workflow document has the same workflow identifier as the previous version.

- 3950
- In a Document Sharing infrastructure (e.g., an XDS environment) two different Content Updaters could be in the situation of race condition when both update, in the same time, the same Workflow Document.

In this case two actors (Content Updater A and Content Updater B) retrieve the same Workflow Document (e.g., Workflow Document with document uniqueId 1) and change it.

Content Updater A publishes a new version updated with a new document uniqueId (e.g., document uniqueId 2) and the previous version (with document uniqueId 1) is deprecated..

- 3955
- When Updater Creator B tries to replace the same Workflow Document (document uniqueId 1) with his updated version this transaction generates an error because the document uniqueId 1 is deprecated and replaced with document uniqueId 2.

Content Updater B shall retrieve the current version of the Workflow Document (document uniqueId 2) and update it with a new version of the document with document uniqueId 3.

3960

5.4.5.5 Association of a clinical document to a task and <taskEvent>

Any clinical documents included as input or output documents within the taskData element that are registered in an XDS Document Registry shall be referenced using uniqueId and homeCommunityId of the Clinical Document referenced.

3965

5.4.5.6 Get the Workflow Document and a clinical document associated to the workflow

The most recent version of the Workflow Document may be retrieved at any point during the workflow.

- 3970
- The version of the Workflow Document with an approved status contains the most current information on the workflow and its tasks. So an XDW Content Consumer needs to analyze only the approved version to get all current information.

- 3975
- Any Workflow Document contains details of each task that has been performed. A task or <taskEvent> includes the references (DocumentEntry.uniqueId and homeCommunityId) to zero or more input and/or output clinical documents. These documents may be obtained by means of XDS, or should be included along with the Workflow Document if XDR or XDM is used.

5.4.5.7 Use of the eventCodeList to manage the status of a Workflow Document

An overall workflow status is required to be set by each author of a new step. This value is either OPEN or CLOSED.

3980 This workflow status is required to be present in every workflow step, and shall take either the value OPEN or CLOSED.

By setting this workflow status to OPEN, a step author indicates that, for the workflow definition and the step author further steps are expected to be performed.

By setting this workflow status to CLOSED, a step author indicates that, for the workflow definition and the step author no further steps are expected to be performed.

3985 This workflow status shall be present for all XDW documents in its eventCodeList metadata.

This use of workflow status enables the use of query to locate OPEN or CLOSED workflows with certain other properties.

The EventCodeList contains the workflow status with two possible code values: either OPEN or CLOSED.

3990 **5.4.5.8 Parameters for Required Queries**

The section below documents some examples of the possible queries in an XDS environment (defined in the Registry Stored Query [ITI-18] transaction) to obtain the different documents related to the workflow from some parameters available:

- Find all open Workflow Documents for a patient

3995 A Registry Stored Query “FindDocuments” maybe used with patientId, XDW document formatCode and eventCodeList with the value “urn:ihe:iti:xdw:2011:eventCode:open” for the Workflow Document.

- Find all particular type of open Workflow Documents for a patient

4000 A Registry Stored Query “FindDocuments” may be used with patientId, XDW document formatCode, eventCodeList with the value “urn:ihe:iti:xdw:2011:eventCode:open” for the Workflow Document and a specific XDW document typeCode.

- Get one or more documents referenced in a Workflow Document

4005 A Registry Stored Query “FindDocuments” which retrieves the Workflow Document (like in the first example) and a Registry Stored Query “GetDocuments” with document uniqueId and homeCommunityId to retrieve one or more documents referenced inside the Workflow Document.

- Find the latest version of a Workflow Document for a given workflow identifier

A Registry Stored Query “FindDocumentsByReferenceId” may be used with patientId and the workflow identifier.

4010 **5.4.6 XDS Metadata**

5.4.6.1 Document Metadata

The following metadata elements shall be used to describe the Workflow Document in an XDS Affinity Domain. The XDW Profile does not introduce new metadata and all the metadata elements used are the common Document Sharing specified in Section 4.2.3.2.

4015

Table 5.4.6.1-1: XDW Constraints for Document Metadata Attributes

DocumentEntry Attribute	XDW Constraints
author	Represents the humans and/or machines that authored the document. In the Workflow Document the Author is the human and/or machine which most recently updated the Workflow Document. This means that when a Workflow Document is updated by a different person or machine, the author changes.
authorInstitution (sub-attribute of author)	No special requirements for Workflow Document
authorPerson (sub-attribute of author)	No special requirements for Workflow Document
authorRole (sub-attribute of author)	No special requirements for Workflow Document
authorSpecialty (sub-attribute of author)	No special requirements for Workflow Document
availabilityStatus	No special requirements for Workflow Document
classCode	This code is specified by the XDS Affinity Domain. The XDS Affinity Domain should specify a code for non-clinical Workflow Management documents.
comments	No special requirements for Workflow Document
confidentialityCode	No special requirements for Workflow Document
creationTime	No special requirements for Workflow Document
entryUUID	No special requirements for Workflow Document
eventCodeList	For a Workflow Document, one code of this list shall be used to define the overall status of the workflow. This code shall have one of the following two values: <ul style="list-style-type: none"> code: urn:ihe:iti:xdw:2011:eventCode:open codingScheme: 1.3.6.1.4.1.19376.1.2.3 code: urn:ihe:iti:xdw:2011:eventCode:closed codingScheme: 1.3.6.1.4.1.19376.1.2.3 (See Section 5.4.5.7.)
formatCode	Each XDW Workflow Document shall have the following value for the formatCode attribute: code: urn:ihe:iti:xdw:2011:workflowDoc codingScheme: 1.3.6.1.4.1.19376.1.2.3
hash	No special requirements for Workflow Document
healthcareFacility TypeCode	No special requirements for Workflow Document

DocumentEntry Attribute	XDW Constraints
homeCommunityId	No special requirements for Workflow Document
languageCode	No special requirements for Workflow Document
legalAuthenticator	No special requirements for Workflow Document
mimeType	No special requirements for Workflow Document
patientId	No special requirements for Workflow Document
practiceSettingCode	No special requirements for Workflow Document
referenceIdList	Contains the workflow identifier. Only a single value shall be sent in this list. Only the CXi.1 and CXi.5 components shall be used: CXi.1 shall contain same value as XDW.WorkflowDocument.workflowInstanceId CXi.5 shall contain urn:ihe:iti:xdw:2013:workflowInstanceId.
repositoryUniqueId	No special requirements for Workflow Document
serviceStartTime	Shall be the starting time the service being documented took place For the Workflow Document the serviceStartTime is the time at which work began on the earliest task for this workflow. If present, shall have a single value.
serviceStopTime	No special requirements for Workflow Document
size	No special requirements for Workflow Document
sourcePatientId	No special requirements for Workflow Document
sourcePatientInfo	No special requirements for Workflow Document
title	No special requirements for Workflow Document
typeCode	For Workflow Documents defined by an IHE profile, the profile specifies the value of typeCode used. For other Workflow Documents defined in the XDS Affinity Domain, the XDS Affinity Domain specifies the value for typeCode.
uniqueId	No special requirements for Workflow Document
URI	No special requirements for Workflow Document

5.4.6.2 XDS SubmissionSet Metadata

No additional constraints. See Section 4.2.3.3.

5.4.6.3 XDS Folder Metadata

4020 No additional constraints