**Integrating the Healthcare Enterprise**

# IT Infrastructure
# Technical Framework

# Volume 3
# (ITI TF-3)
# Cross-Transaction Specifications
# and
# Content Specifications

**Revision 10.0 – Final Text**

**September 27, 2013**

_____

## CONTENTS

_____

_____

_____

_____

## 4 Metadata used in Document Sharing profiles

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

100 The ITI Document Sharing profiles which use this metadata are:

- Cross-Enterprise Document Sharing (XDS.b)

- Cross-Enterprise Document Reliable Interchange (XDR)

- Cross-Enterprise Document Media Interchange (XDM)

- Multi-Patient Queries (MPQ)

105 - 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.

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

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

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

## 4.1 Abstract Metadata Model

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

- SubmissionSet – represents a collection of Folders, Documents and Associations
  submitted together.

- Folder – represents a collection of related Documents.

- DocumentEntry – represents a Document.

125 - The two Association types are:

_____

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

- Relationship – represents a relationship between DocumentEntries.
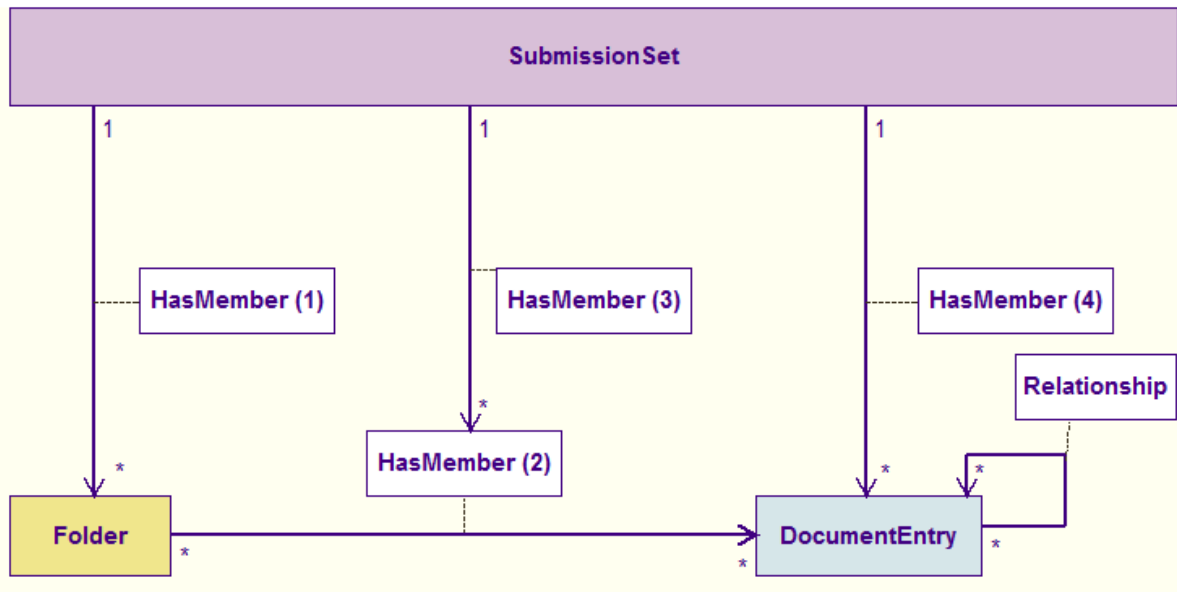


130    **Figure 4.1-1: Document Sharing Objects and Associations**

### 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

135    - Folder

- DocumentEntry

**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,

140    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

145    destination, and a retry is possible.

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
150    done.

A SubmissionSet shall be the source of at least one Association of type HasMember (1),
HasMember (3), and/or HasMember (4).

**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
155    by multiple SubmissionSets sent from multiples departments that are submitting their
DocumentEntry objects at different times. For example, a Folder may be used to collect the
DocumentEntry objects for the patient's documents that relate to an exam event, such as the
exam request and prior results as well as the eventual exam results. As the exam results become
available, the DocumentEntry objects can be added to the Folder for the exam records.

160    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.

**DocumentEntry** – DocumentEntry is a metadata object representing a document. This metadata
165    object does not contain the contents of the document; instead it contains attributes describing the
document. Each DocumentEntry represents a single document. Details on how documents and
metadata are managed depend on the requirements in a particular Document Sharing profile.

For example, in XDS, a DocumentEntry is the logical representation in the Registry of the
Document that the Source submitted to a Repository. An entire document's contents can
170    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 DocumentEntry to a Registry in
lieu of submitting the document creates a separation of concerns, allowing the Registry to
specialize in indexing, while the Repository manages document storage.

175    ## 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.

180    - **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.

   HasMember (1): An association from a SubmissionSet to a Folder identifies the Folder as
   a member of that SubmissionSet. It identifies the Submission Set that contained
185    the initial creation of the Folder.

_____

_____

HasMember (2): 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).

190      HasMember (3): An association from a SubmissionSet to a HasMember (2) Association identifies that HasMember (2) 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.

HasMember (4): An association from a SubmissionSet to a DocumentEntry identifies the 195      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 200      that SubmissionSet.

- **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.

205      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.

210 **4.1.3  Metadata Attributes**

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

**4.1.3.1  The Purpose of Metadata Attributes (Informative)**

215 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.

220 - **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.

- **Security & Privacy** – Attributes that are used by Privacy and Security rules to
225    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.

- **Descriptive** – Attributes that are used to describe the clinical value, so they are expressly
230    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
235    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.

- **Object Lifecycle** – Attributes that describe the current lifecycle state of the document
240    including relationships to other documents. This would include classic lifecycle states of
created, published, replaced, transformed, and deprecated.

- **Exchange --** Attributes that enable the transfer of the document for both push type
transfers, and pull type transfers. These attributes are used for low-level automated
processing of the document. These attributes are not the workflow routing, but rather the
245    administrative overhead necessary to make the transfer. This includes the document
unique Id, location, size, MIME types, and document format.

_____

_____
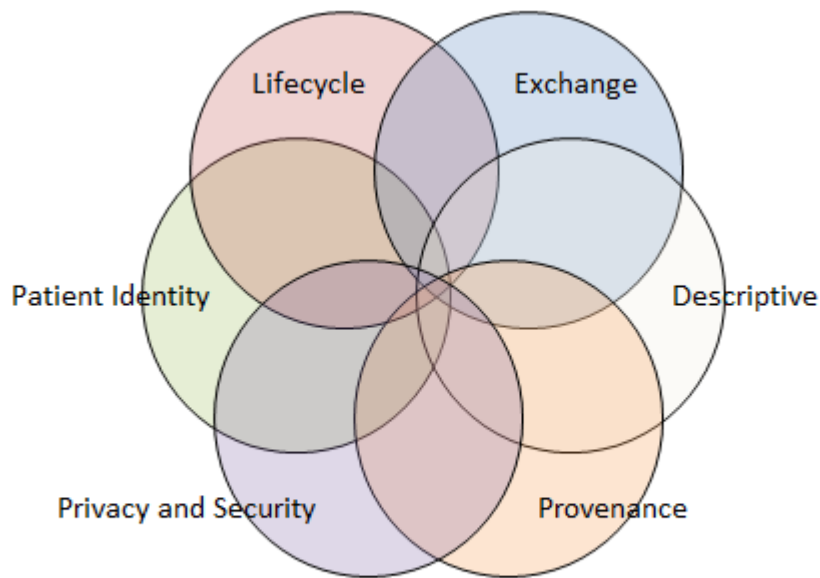


**Figure 4.1.3.1-1: Pictorial of Overlapping Document Sharing Metadata Purpose**

250 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 be consistently assigned and processed.

255 **4.1.3.2 DocumentEntry Metadata Attributes**

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

260

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

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

_____

_____

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

_____

_____

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

### 4.1.3.3  SubmissionSet Metadata Attributes

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.

265

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

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

_____

_____

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

### 4.1.3.4  Folder Metadata Attributes

270    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.

_____

_____

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

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

## 4.2  ebRIM Representation

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

280   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:

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

- **ebRS:** The "ebXML Registry Services Specification version 3.0" (ebRS) defines the services and protocols for an ebXML Registry.

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

_____

_____

When document sharing was first introduced in IHE, XDS was the only document sharing
model. In the initial XDS profile, the Registry actor could be implemented as an adaptor to an
295 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
300 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.

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
305 implemented using an ebXML Registry.

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

310                      **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
315 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

320 ### 4.2.1.1  DocumentEntry

The DocumentEntry does not contain the contents of the document; instead it contains attributes
describing the document. Each DocumentEntry represents a single document which is identified

_____

_____

by the uniqueId attribute. Further details regarding the DocumentEntry object type can be found in 4.1.1.

325    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.

**DocumentEntry**

entryUUID : UUID
logicalId : UUID [0..1]
mimeType : string
availabilityStatus : URN
hash : SHA1
size : integer
languageCode : RFC 3066
legalAuthenticator : XCN [0..1]
repositoryUniqueId : OID
serviceStartTime : DTM [0..1]
serviceStopTime : DTM [0..1]
sourcePatientId : CX
sourcePatientInfo : PID [0..1]
URI : URI [0..1]
title : string [0..1]
comments : string [0..1]
patientId : CX
uniqueId : OID
classCode : Code
confidentialityCode : Code [1..*]
eventCodeList : Code [1..*]
formatCode : Code
healthcareFacilityTypeCode : Code
praticeSettingCode : Code
typeCode : Code
documentAvailability : URN [0..1]
homeCommunityId : URN [0..1]
creationTime : DTM

**author**

authorInstitution : XON [*]
authorPerson : XCN
authorRole : string [*]
authorSpecialty : CE [*]
authorTelecommunication : XTN [*]

1    *

**Code**

code : string
displayName : string
codingScheme : OID

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

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

_____

_____
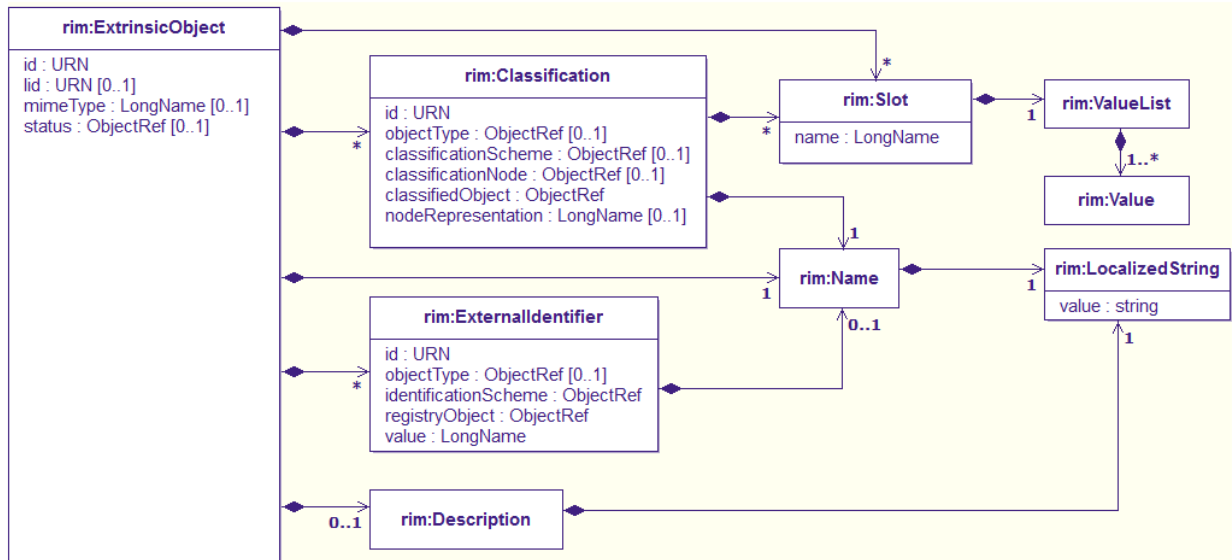


**Figure 4.2.1.1-2: UML diagram for rim:ExtrinsicObject (Informative)**

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

The expression of the DocumentEntry is done by mapping the abstract DocumentEntry metadata attributes into rim:ExtrinsicObject class attributes, elements and other associated classes. This mapping uses, wherever possible, the parts of rim:ExtrinsicObject as intended (such as Name,

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

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

_____

### 4.2.1.2  SubmissionSet

345  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.
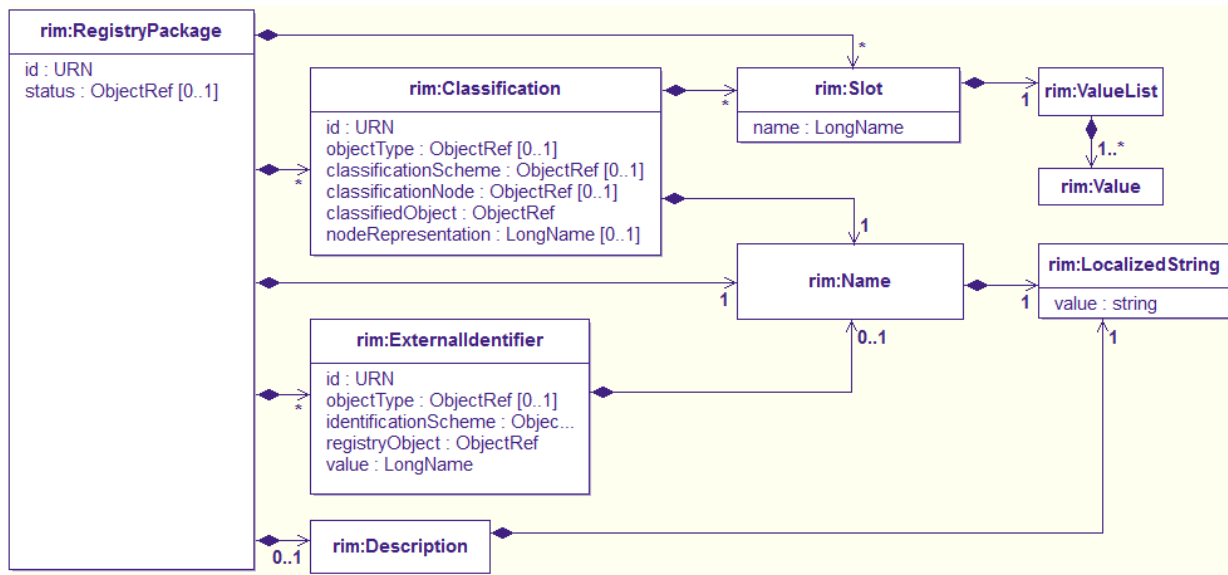


350

**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
355  identify a RegistryPackage as a SubmissionSet versus a Folder.

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

SubmissionSets exist for two reasons:

1.  To support atomic submissions

360  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

_____
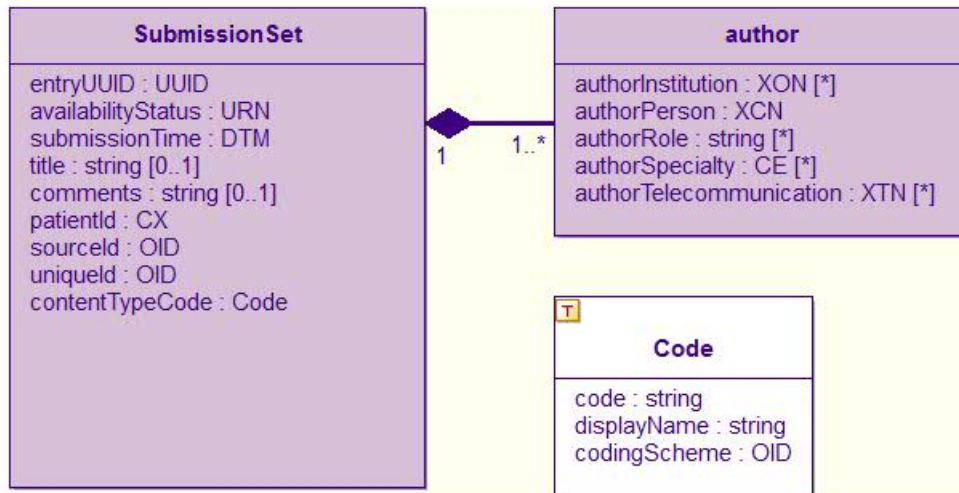


**Figure 4.2.1.2-2: SubmissionSet Metadata Attributes (Informative)**

365 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.

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

370 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.

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

385 **Classification encoded outside the RegistryPackage object**
```
<RegistryPackage id="SubmissionSetId">
</RegistryPackage>
<rim:Classification
    classifiedObject="SubmissionSetId"
390    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"
/>
```
395

_____

| 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.<br><br>Accepted values:<br><br>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

400   The abstract concept of a Folder is expressed through an ebRIM RegistryPackage classified as Folder (see the UML representation of the ebRIM RegistryPackage, Figure 4.2.1.3-1). A Folder is used to bundle DocumentEntry objects. Further details regarding the Folder object type can be found in section 4.1.1.
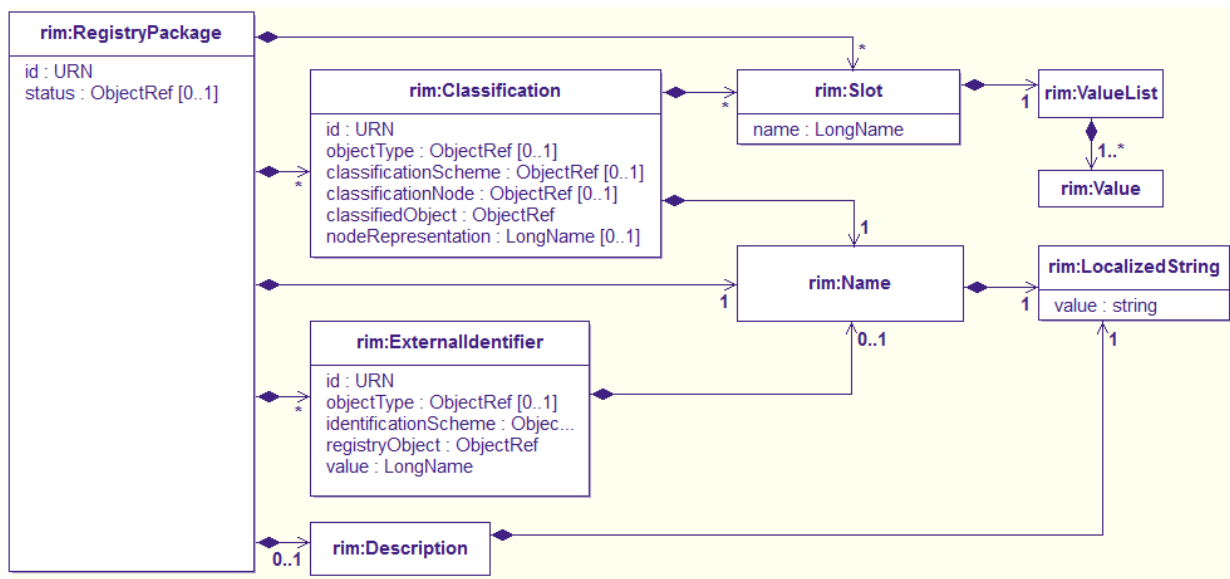


405              **Figure 4.2.1.3-1: UML diagram for Folder (Informative)**

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.

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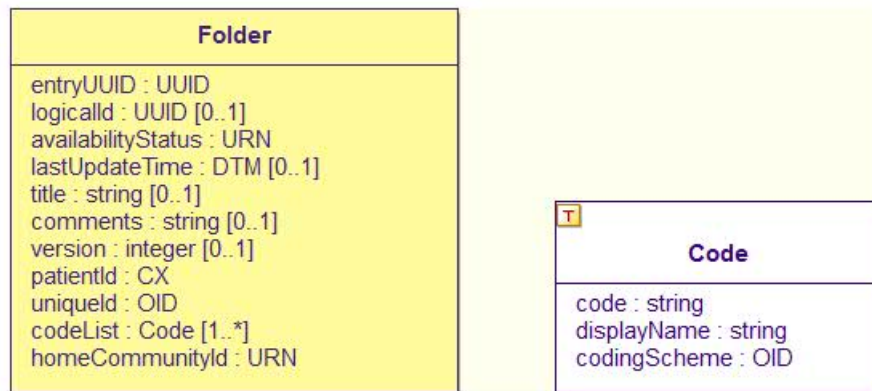


**Figure 4.2.1.3-2: Folder Metadata Attributes (Informative)**

## 415   4.2.1.3.1 Creating a Folder object from a RegistryPackage element

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

420     the Classification.

**Classification encoded inside the RegistryPackage object**

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

**Classification encoded outside the RegistryPackage object**

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

| Item | Description |
|---|---|
| Classification/@classifiedObject | The @id attribute of the RegistryPackage being classified. |
| Classification/@classificationNode | A fixed value identifying the type of object the RegistryPackage represents. Accepted values: 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

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

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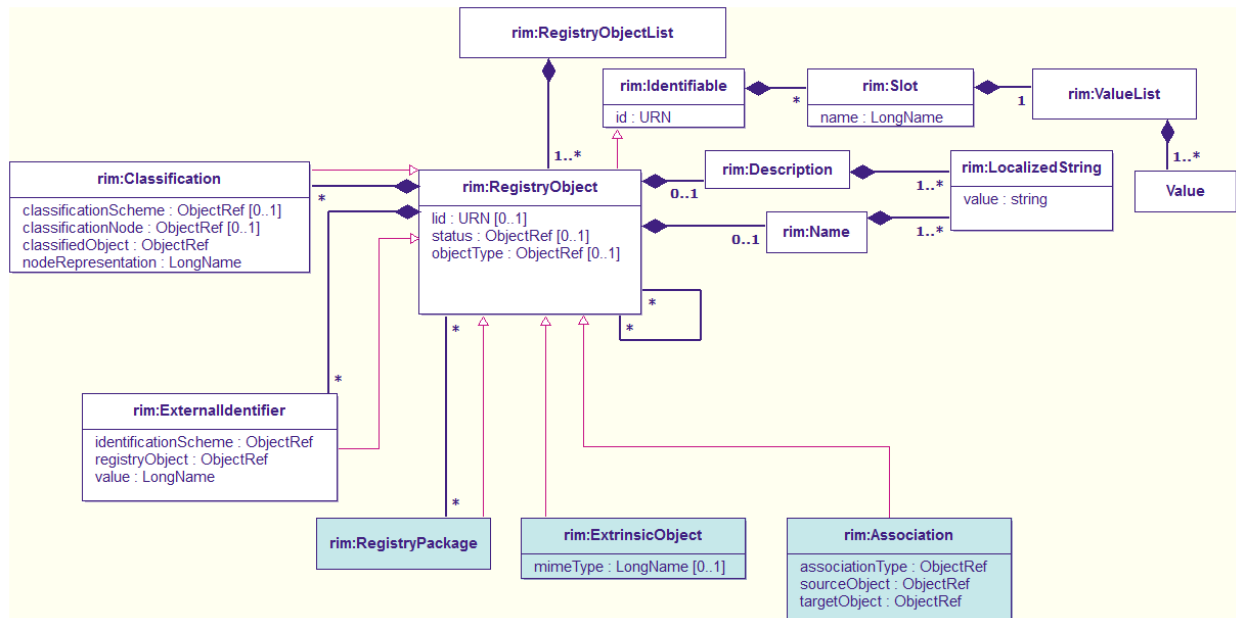


455 **Figure 4.2.1.1-1: Registry Object List (Informative)**

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

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

460

_____

## 4.2.2  Association Types

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

Associations can be used to build relationships between:

470    • A SubmissionSet and a DocumentEntry – HasMember (4)

• A SubmissionSet and a Folder – HasMember (1)

• A Folder and a DocumentEntry – HasMember (2)

• A SubmissionSet and an Association – HasMember (3)

• A DocumentEntry and another DocumentEntry – Relationship

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

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

480    Note:    The Association can be in a different SubmissionSet than the DocumentEntry. For example, a Signs relationship
between a document signature and a document can be provided when the document and the signature are
submitted via different Submission Sets.
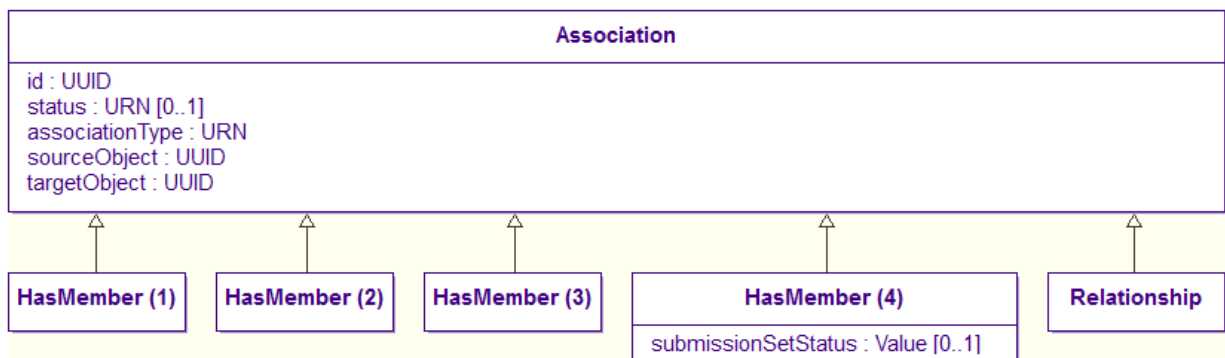


**Figure 4.2.2-1: Association (Informative)**

485    Figure 4.2.1.1-1 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 HasMember (4) (SubmissionSet to DocumentEntry) also has the
submissionSetStatus metadata attribute in addition to the Association class attributes.

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

490    • sourceObject

_____

- targetObject
- associationType

These attributes can be thought to make a small sentence:

- sourceObject AssociationType targetObject

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

- Folder HasMember DocumentEntry

Graphically this example Association looks like:
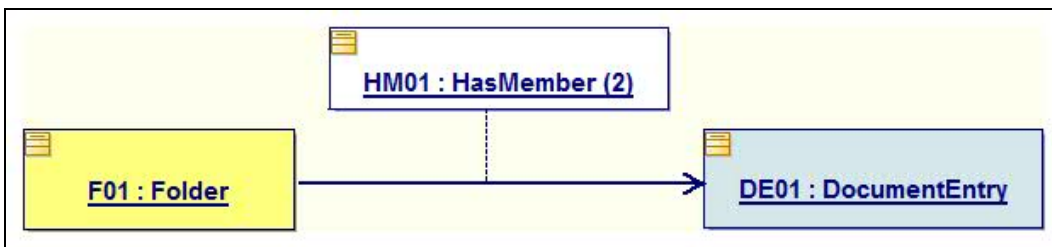


**Figure 4.2.2-2: Folder HasMember DocumentEntry (Informative)**

500   **Association Type formatting**

An Association type shall be specified as a URN.

The valid Association types are specified in the following table.

**Table 4.2.2-1: Association Types**

| Association function | Association Type |
|---|---|
| ebRIM HasMember | `urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember` |
| HL7 Document Relationship Replace | `urn:ihe:iti:2007:AssociationType:RPLC` |
| HL7 Document Relationship Transform | `urn:ihe:iti:2007:AssociationType:XFRM` |
| HL7 Document Relationship Append | `urn:ihe:iti:2007:AssociationType:APND` |
| HL7 Document Relationship Transform and Replace | `urn:ihe:iti:2007:AssociationType:XFRM_RPLC` |
| IHE Signature | `urn:ihe:iti:2007:AssociationType:signs` |

505   ## 4.2.2.1  HasMember

In the Document Sharing abstract metadata model, many different relationships are defined between SubmissionSet, DocumentEntry and Folder objects. In this section, each of these relationships is given its own name, like HasMember (4) - SubmissionSetHasMemberDocumentEntry. In the underlying ebRIM model, all of these
510   relationships are created using the ebRIM HasMember Association type.

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

### 4.2.2.1.1 HasMember (4)

**HasMember -** a DocumentEntry shall be submitted as part of a SubmissionSet by connecting the
objects with a HasMember Association. This is shown as HasMember (4) in figure 4.1-1.
515    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 any
520    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
525    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 HasMember (4) Association connecting
530    them.
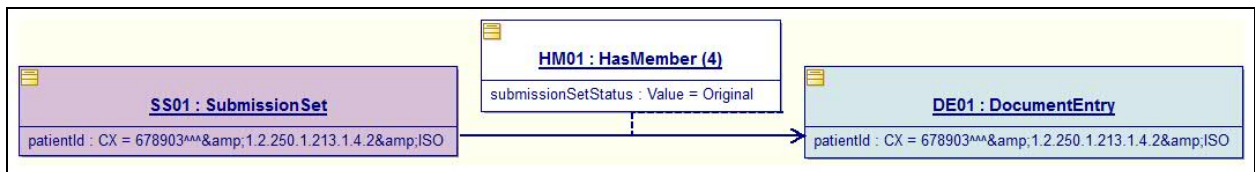


**Figure 4.2.2.1.1-1: SubmissionSet HasMember DocumentEntry (Informative)**

When submitting an existing document by value:

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

535    - The sourceObject shall contain the Id of the SubmissionSet object

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

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

_____

```
</rim:Association>
```

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

550    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
555    **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 HasMember (4) Association with the 'Value=Reference' connects the SubmissionSet to a DocumentEntry already present in the
560    receiving actor.
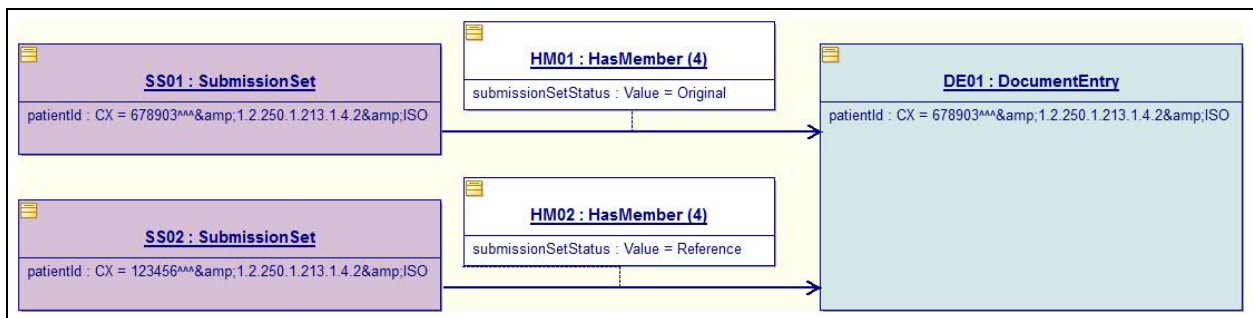


**Figure 4.2.2.1.1-2: Submission of a reference to an existing Document (Informative)**

When submitting a reference to an existing document:

565    • 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.

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

### 4.2.2.1.2 HasMember (1)

**HasMember -** Submit a Folder. The submission includes the Folder object and a HasMember
580    Association linking the Folder to the SubmissionSet. This is shown as HasMember (1) in figure

_____

4.2.2.1.1-2. The value of the SubmissionSet.patientId attribute shall match the value of the Folder.patientId attribute.
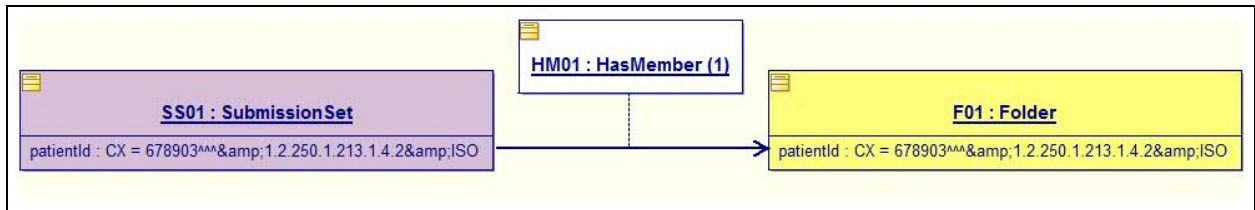


585        **Figure 4.2.2.1.2-1: SubmissionSet HasMember Folder (Informative)**

When submitting a Folder:

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

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

### 4.2.2.1.3 HasMember (2)

590    **HasMember (2) -** 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.
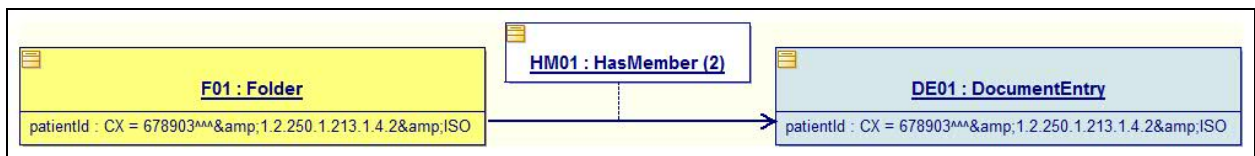


**Figure 4.2.2.1.3-1: Folder HasMember DocumentEntry (Informative)**

**Linking documents to a folder**

595    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 HasMember (2) Association shall be accompanied by a HasMember (3) Association that links the HasMember (2) 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.

600    When adding a DocumentEntry to a Folder:

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

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

### 4.2.2.1.4 HasMember (3) –

**HasMember -** a HasMember Association linking a SubmissionSet to a HasMember (2)
605    Association, which is in turn an Association between a Folder and a DocumentEntry. This is shown as HasMember (3) 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
610   identify the entity that created the link between a particular document and a particular Folder and
shall be as follows:

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

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

615   ## 4.2.2.1.5 Adding DocumentEntries to Folders

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

1. The DocumentEntry can be submitted as part of the Folder in a single submission.

2. The DocumentEntry and Folder are already present. The new submission makes the
   DocumentEntry a member of the Folder by adding the Association.

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

4. The Folder is already present. The new submission includes the DocumentEntry and the
   Association to make the DocumentEntry part of the Folder.

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

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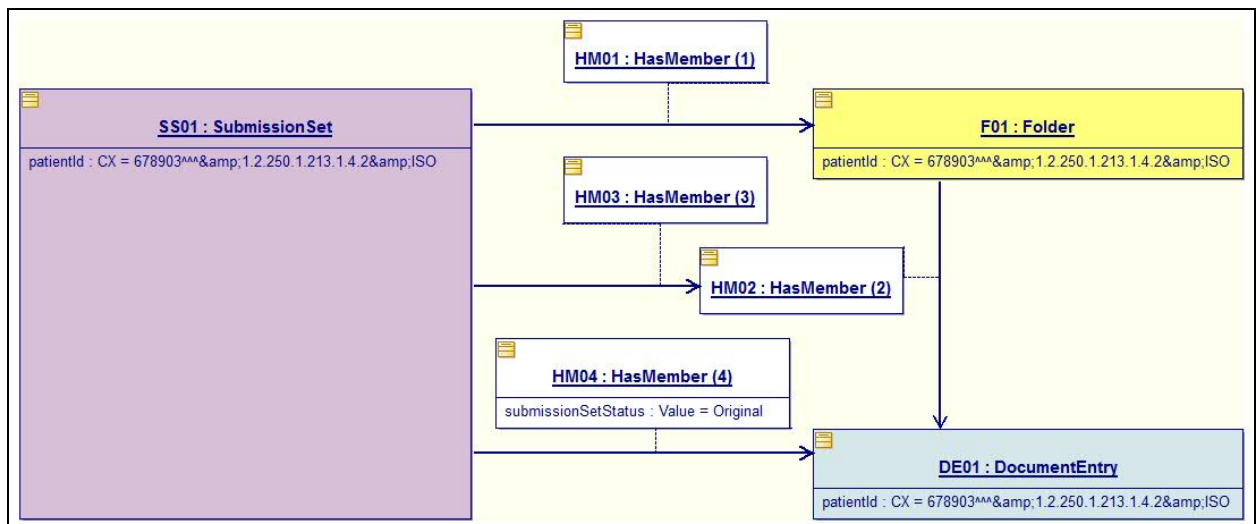


**Figure 4.2.2.1.5-1: Scenario 1 - DocumentEntry submitted as part of the Folder
(Informative)**

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

630    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 HasMember (2) will not have as member either the DocumentEntry or the Folder that correspond to the referenced document and Folder.
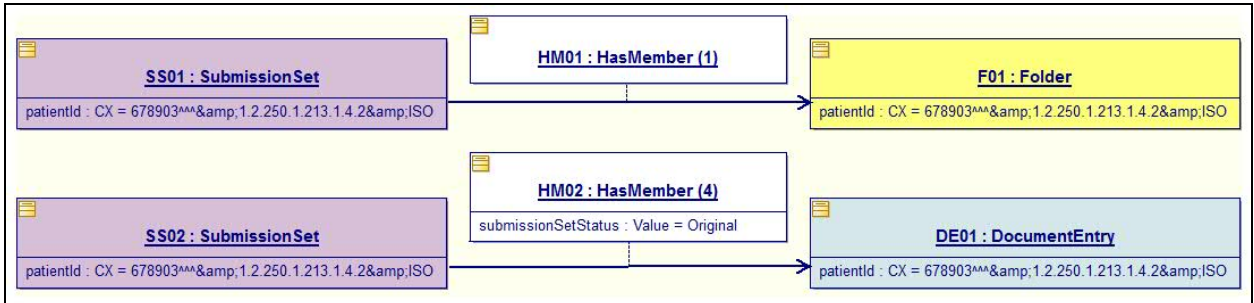


635    **Figure 4.2.2.1.5-2: Scenario 2 - Existing DocumentEntry and existing Folder (Informative)**
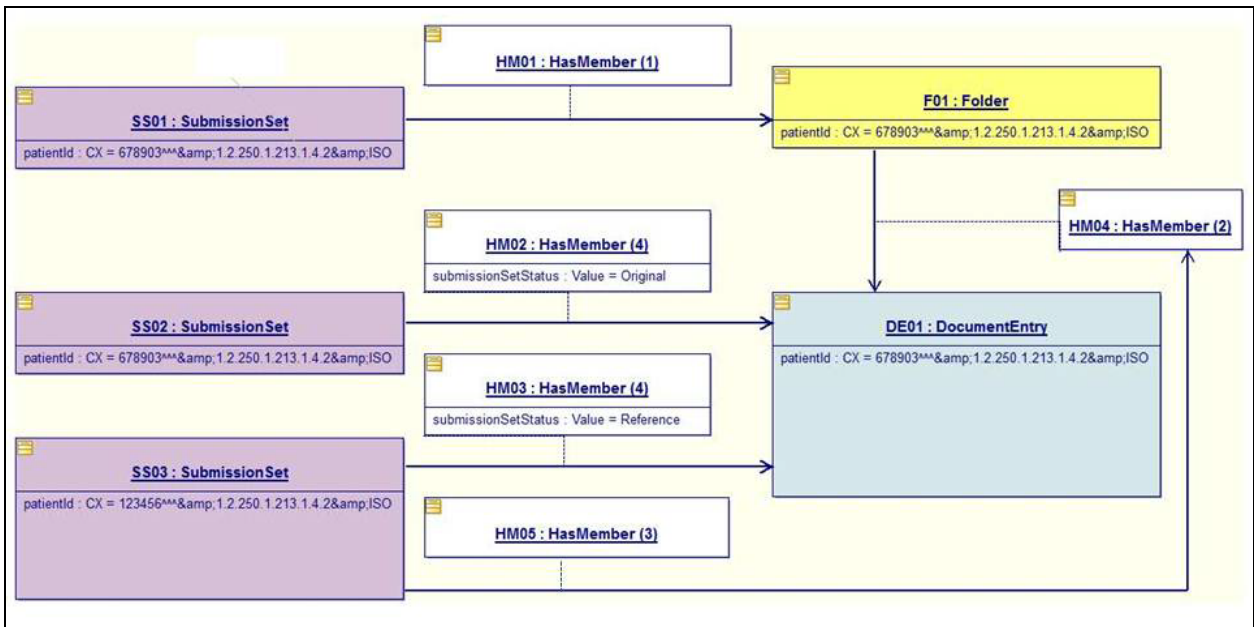


**Figure 4.2.2.1.5-3: Scenario 2 - Add existing DocumentEntry to existing Folder (Informative)**

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

640    When a new Folder is submitted, an existing DocumentEntry can be added to that Folder. In this case, the SubmissionSet object will not contain the DocumentEntry metadata that correspond to the referenced document.
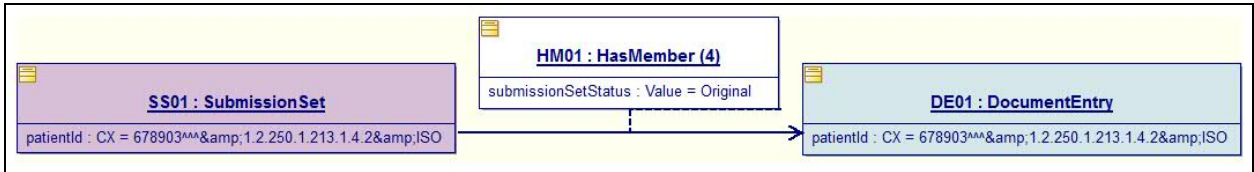
**Figure 4.2.2.1.5-4: Scenario 3 - Starting point - Existing DocumentEntry (Informative)**
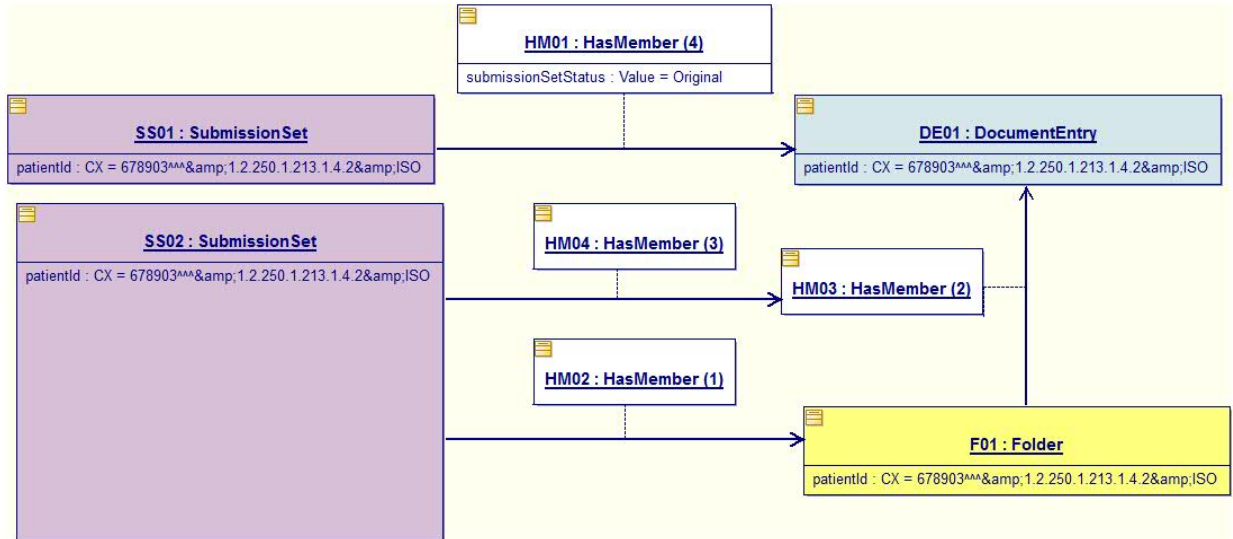


645

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

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

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

650
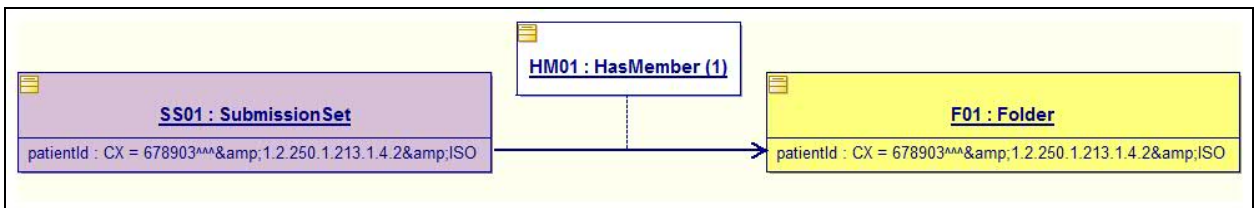


**Figure 4.2.2.1.5-6: Scenario 4 - Starting point - Existing Folder (Informative)**

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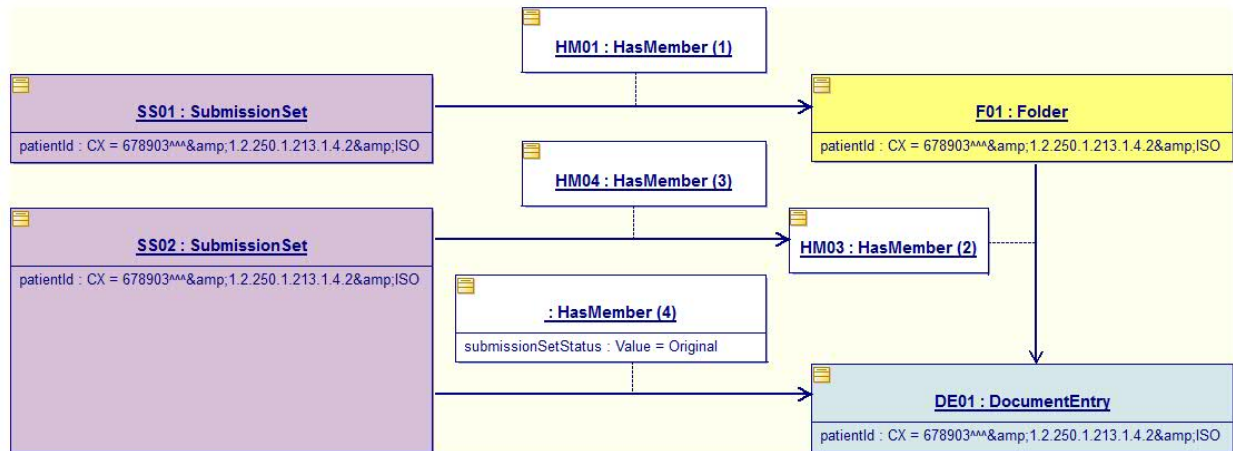
655     **Figure 4.2.2.1.5-7: Scenario 4 - DocumentEntry submitted and added to existing Folder (Informative)**

### 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
660    triggers specific processing in the receiving actor in some profiles. This is documented in the Expected Actions section of the relevant transaction in an IHE profile.

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

665

**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. |

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).

670    An Association linking the new document to an existing document.

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,

_____

675   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 ITI-18).

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

680     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:

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

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

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

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

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

695   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:

700
```
<rim:Association
    associationType="urn:ihe:iti:2007:AssociationType:XFRM"
    sourceObject="source"
    targetObject="urn:uuid:XXX"
    objectType="urn:oasis:names:tc:ebxml-
705     regrep:ObjectType:RegistryObject:Association"
    id="IdExample_042"
    >
  <rim:Classification
    classificationScheme="urn:uuid:abd807a3-4432-4053-87b4-fd82c643d1f3"
710     classifiedObject="IdExample_042"
    id="IdExample_043"
    objectType="urn:oasis:names:tc:ebxml-
        regrep:ObjectType:RegistryObject:Classification"
    nodeRepresentation="LanguageExample"
715     >
    <rim:Name>
      <rim:LocalizedString value="Translation into LanguageExample" />
    </rim:Name>
    <rim:Slot name="codingScheme">
720     <rim:ValueList>
        <rim:Value>Example translation types</rim:Value>
```

_____

```
      </rim:ValueList>
    </rim:Slot>
  </rim:Classification>
725 </rim:Association>
```

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

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

730 **4.2.2.2.1 APND**

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
735 the addendum, etc.) The APND relationship leaves the original DocumentEntry with its
availabilityStatus unchanged (Approved).
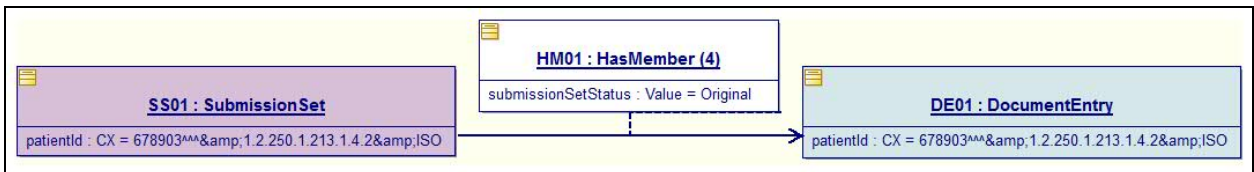


**Figure 4.2.2.2.1-1: Starting Point (Informative)**
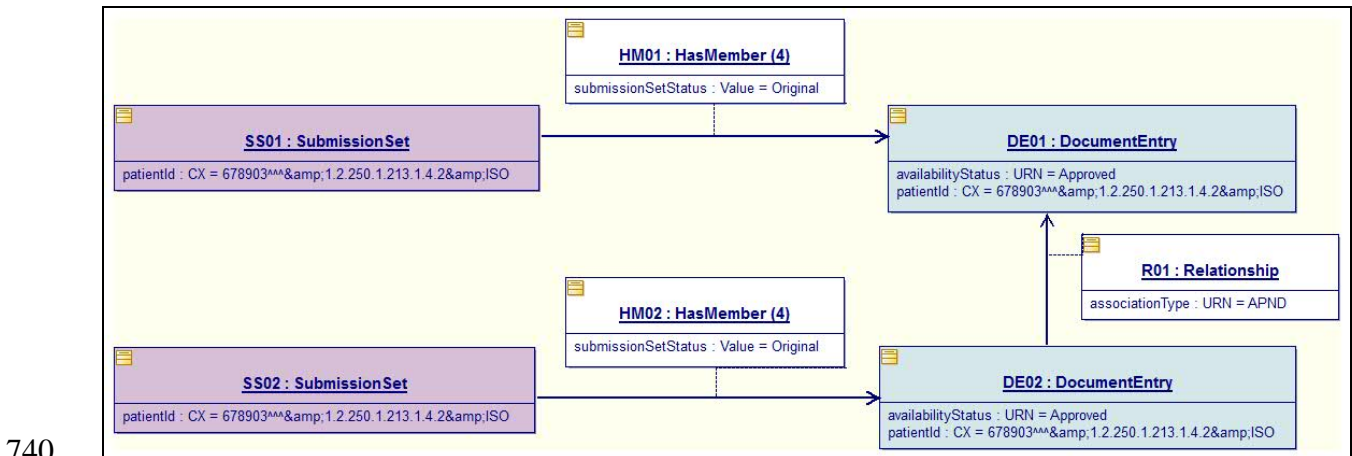


740

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

**Interactions between XFRM and APND**

_____

745    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

The submission shall contain a new DocumentEntry and associated Document and a XFRM Association linking this new DocumentEntry with an existing DocumentEntry. This new
750    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).

755
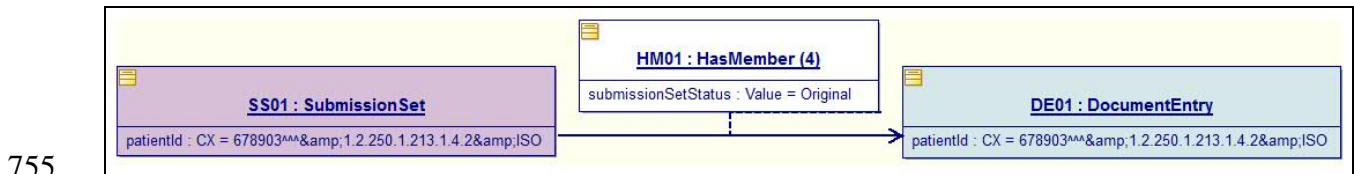
**Figure 4.2.2.2.2-1: Starting Point (Informative)**
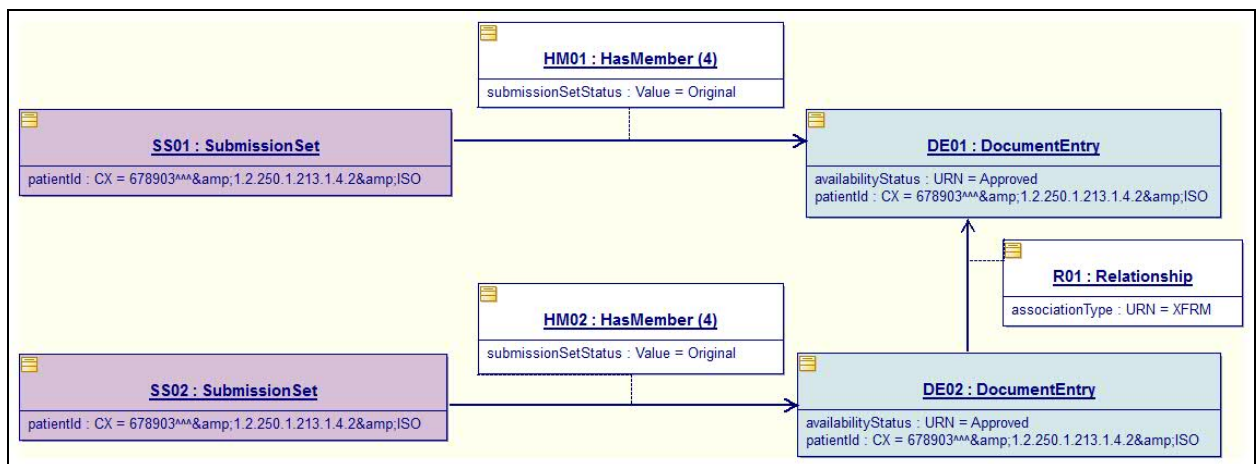
**Figure 4.2.2.2.2-2: New DocumentEntry/Document defining a transformation of an**
760    **existing DocumentEntry/Document (Informative)**

### 4.2.2.2.3 RPLC

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
765    existing DocumentEntry and Document become a deprecated version. The RPLC Association may include documentation describing the replacement (e.g., reason for replacement, etc.).

_____

_____
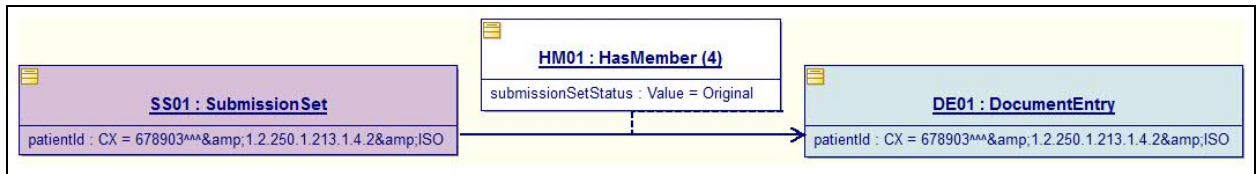


**Figure 4.2.2.2.3-1: Starting Point (Informative)**

770
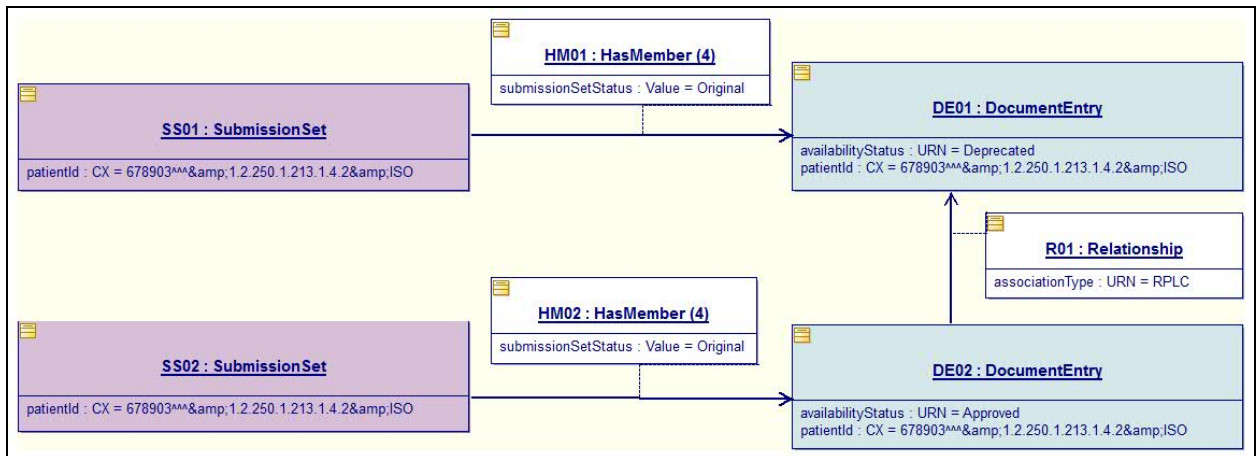


**Figure 4.2.2.2.3-2: New DocumentEntry and Document replacing existing DocumentEntry/Document (Informative)**


775 **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
780 transformation (XFRM) of an existing DocumentEntry/Document:

_____

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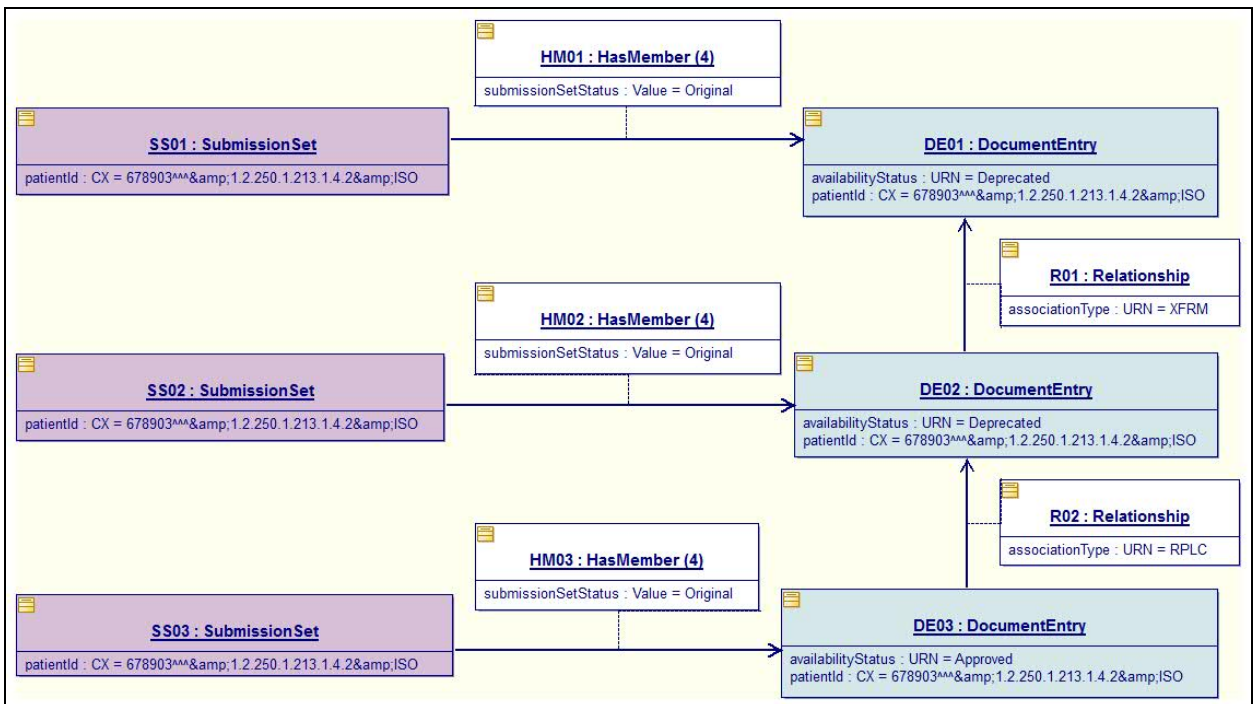


785    **Figure 4.2.2.2.3-4: New DocumentEntry/Document replacing a transformation (XFRM) of an existing DocumentEntry/Document (Informative)**

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.

_____
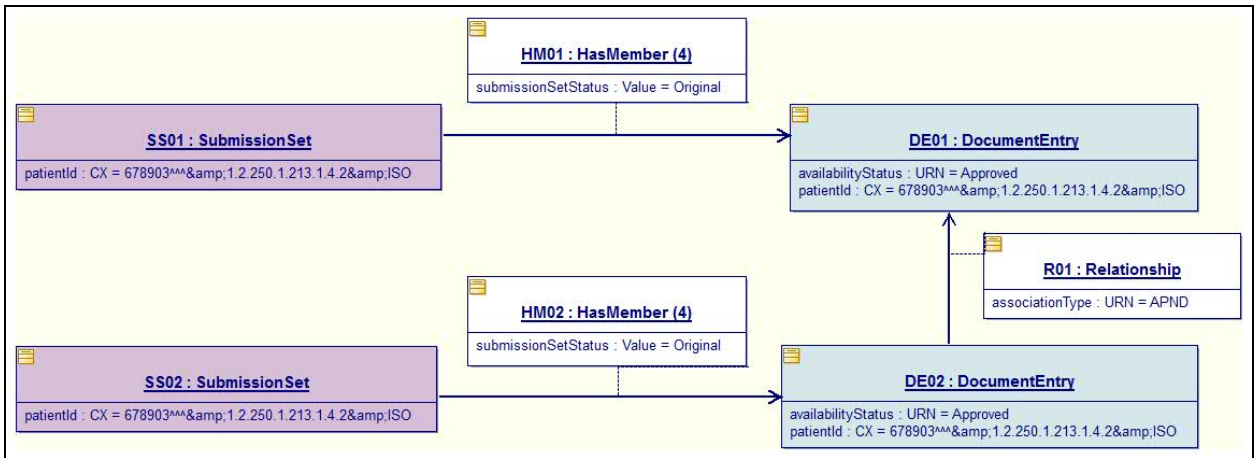


790

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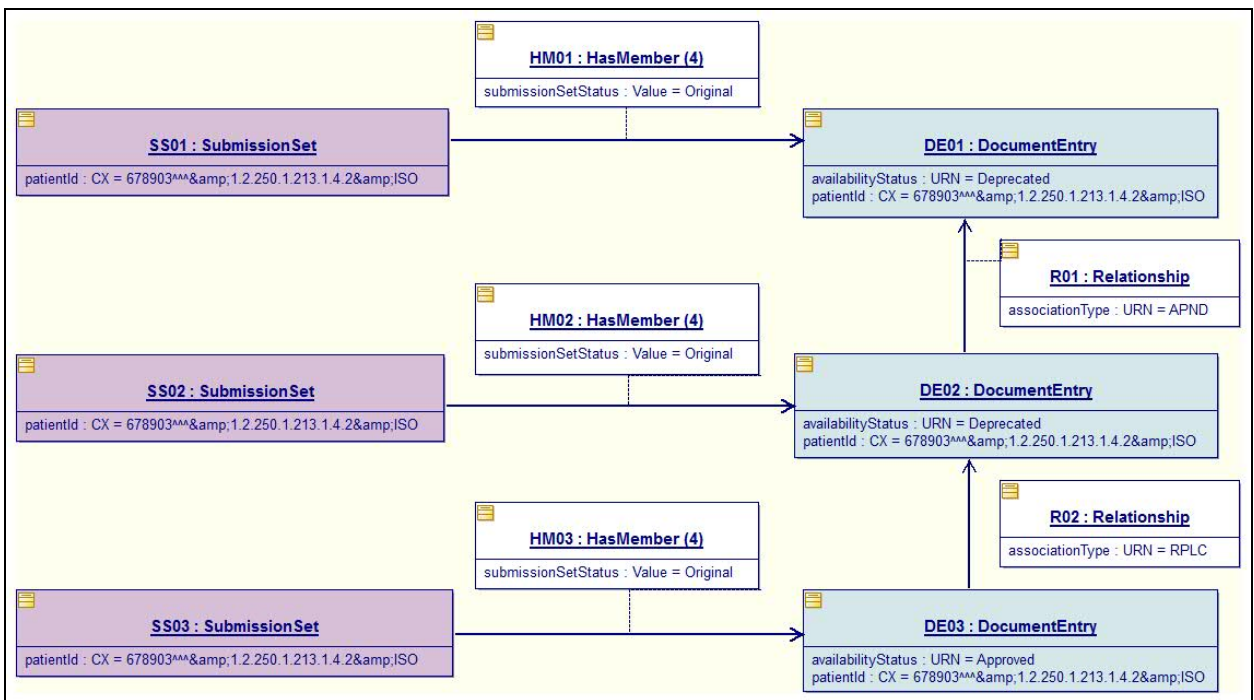


**Figure 4.2.2.2.3-6: New DocumentEntry/Document replacing an addendum (APND) of an existing DocumentEntry/Document (Informative)**

795

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

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

800

_____

_____

When a DocumentEntry is replaced and that DocumentEntry is a member of one or more Folders, new HasMember (2) and HasMember (3) 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.

805
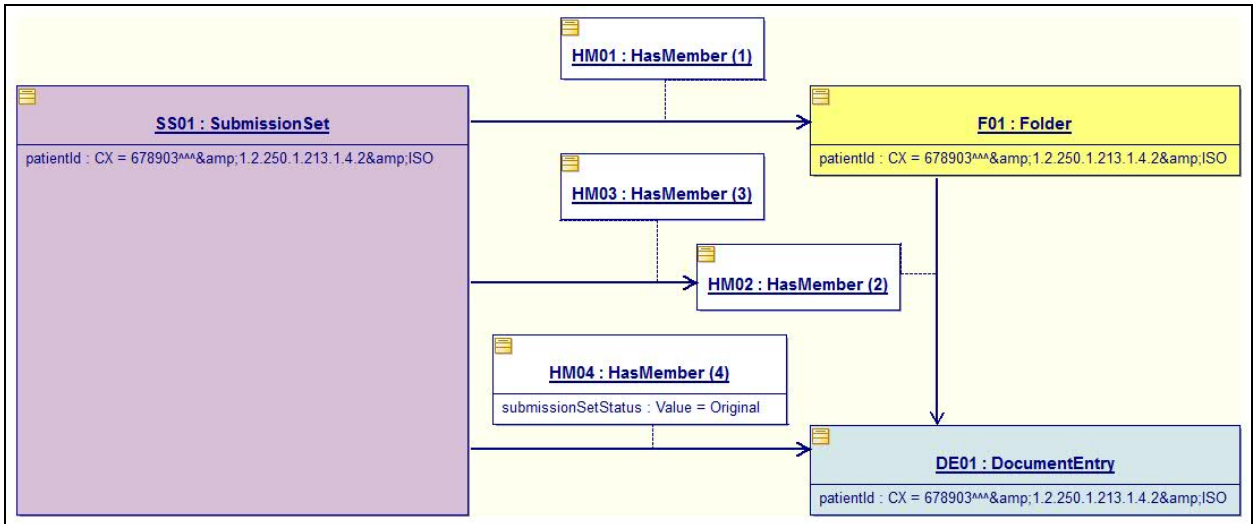


**Figure 4.2.2.2.3-7: DocumentEntry submitted as part of the Folder in a single submission – Starting point (Informative)**



810
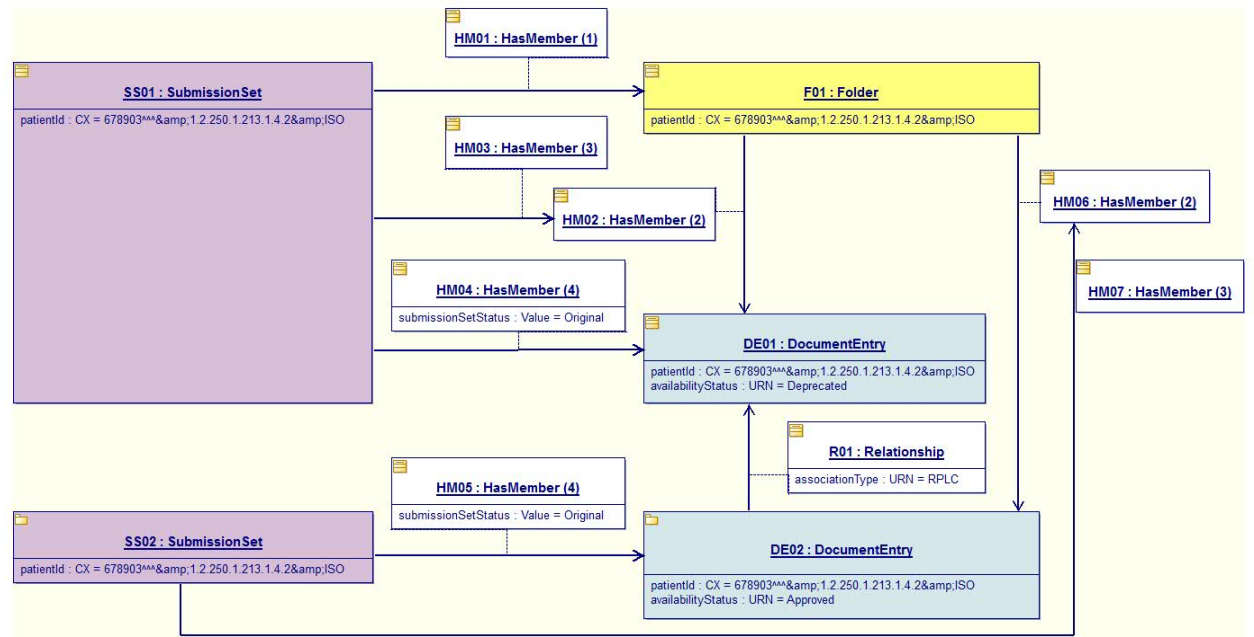
**Figure 4.2.2.2.3-8: Submission of a new DocumentEntry replacing the DocumentEntry part of an existing Folder (Informative)**

_____

_____

### 4.2.2.2.4 XFRM_RPLC

815

820

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.
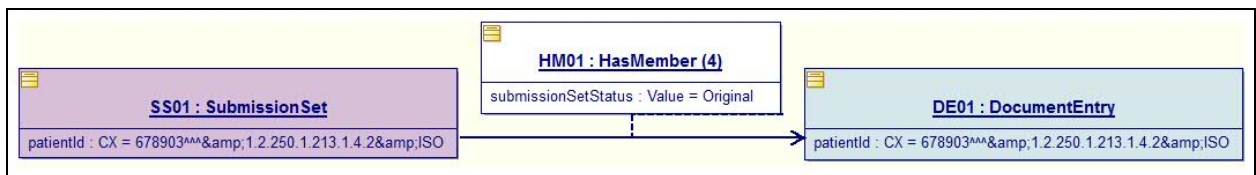


**Figure 4.2.2.2.4-1: Starting Point (Informative)**
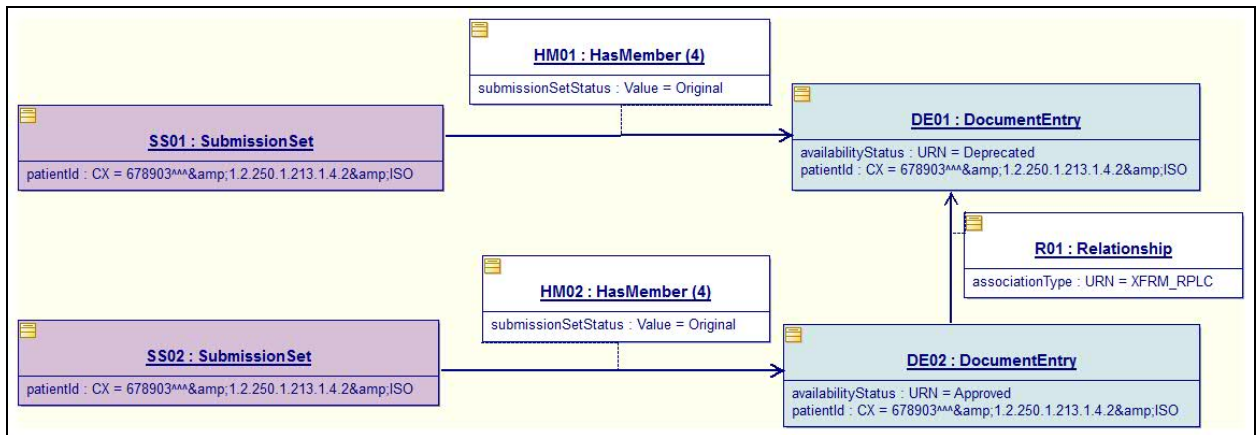


825

**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)**

### 4.2.2.2.5 Signs

830

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.

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

835

content profile.

In constructing this Association, the attributes are:

_____

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

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

**associationType**: signs

840



**Figure 4.2.2.2.5-1: Starting Point (Informative)**
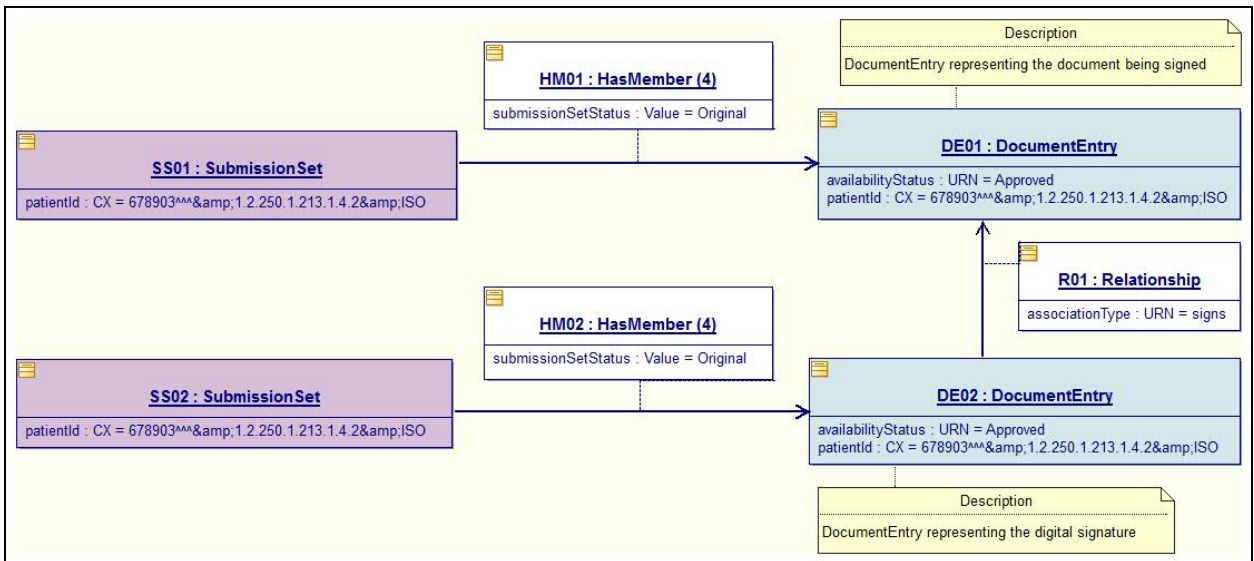


**Figure 4.2.2.2.5-2: New DocumentEntry/Digital signature document signing the existing**
845                   **DocumentEntry/Document (Informative)**

## 4.2.3 Metadata Attributes

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

### 4.2.3.1 General Information about Metadata Attributes

### 4.2.3.1.1 Attribute Value Length

850 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).

_____

855 **4.2.3.1.2 Creating Coded Attributes**

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

1. Code Value – contains the assigned value of the code.

860 2. Code Display Name - The display name used to represent the code (code values are not necessarily human-friendly). Must be non-zero length.

3. Coding Scheme - The name/identification of the coding scheme that the code comes from.

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

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

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

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

| Item | Description |
|---|---|
| Classification/@classification Scheme | 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/@classifiedObj ect | 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. <br><br> 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" |

_____

_____

| Item | Description |
|---|---|
| Classification/@nodeRepresentation | The code value. Required. The example above shows a code value of "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.

890    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.

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

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

900    Note: the attribute DocumentEntry.languageCode attributed is not encoded as a coded attribute as shown above. See table 4.2.3.2-1 for details.

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

### 4.2.3.1.3 Creating External Identifiers

Several attributes of DocumentEntry, SubmissionSet and Folder (table 4.2.3.2-1, table 4.2.3.3-1
910    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.

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
915    example of an identifier in a submission request is:

```
<rim:ExternalIdentifier
    identificationScheme="urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446"
    value="6578946^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO"
    id="IdExample_051"
920    objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier"
    registryObject="DocumentEntry01">
```

_____

_____

```
  <rim:Name>
    <rim:LocalizedString value="XDSDocumentEntry.patientId"/>
  </rim:Name>
</rim:ExternalIdentifier>
```
925

**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. |

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

930 **4.2.3.1.4 Creating Author Attribute**

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

- authorPerson – zero or one

935
- authorInstitution – zero or more

- authorRole – zero or more

- authorSpecialty – zero or more

- authorTelecommunication  – zero or more

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

**Coding:**

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
945 metadata attribute, the value of classificationScheme is different for DocumentEntries and

_____

SubmissionSets; see sections 4.2.3.2.1 and 4.2.3.3.1, respectively. The nodeRepresentation attribute shall be empty.

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

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

The following example shows the definition of a single author for a SubmissionSet.
```
985   <rim:Classification
          classificationScheme="urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d"
          classifiedObject="theSubmission"
          id="Id_045"
          objectType="urn:oasis:names:tc:ebxml-
990          regrep:ObjectType:RegistryObject:Classification"
          nodeRepresentation=""><!-- nodeRepresentation intentionally left blank-->
        <rim:Slot name="authorPerson">
          <rim:ValueList>
            <rim:Value>^Welby^Marcus^^MD^Dr</rim:Value>
995        </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="authorInstitution">
          <rim:ValueList>
            <rim:Value>Some Hospital^^^^^^^^^1.2.3.5.8.9.1789.45</rim:Value>
1000       </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="authorRole">
          <rim:ValueList>
            <rim:Value>name of role</rim:Value>
1005       </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="authorSpecialty">
          <rim:ValueList>
```

```
        <rim:Value>specialty of author</rim:Value>
1010    </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="authorTelecommunication">  <!-- shall be single valued -->
      <rim:ValueList>
        <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value>
1015    </rim:ValueList>
    </rim:Slot>
</rim:Classification>
```

1020 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:**

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.

1025 **Coding:**

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

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
1030 ebRIM Slot.

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

```
<rim:Slot name="authorInstitution">
1035  <rim:ValueList>
        <rim:Value>MercyHospital</rim:Value>
        <rim:Value>NY Mercy Hospital^^^^^^^^^1.2.3.9.1789.45</rim:Value>
      </rim:ValueList>
</rim:Slot>
```

1040 ### 4.2.3.1.4.2 authorPerson

**Description:**

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.

1045 **Coding:**

The format of the authorPerson value is XCN; see table 4.2.3.1.6-2 for description of XCN format.

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

1050 This example shows the authorPerson Slot holding the required single value.

_____

```
      <rim:Slot name="authorPerson">
        <rim:ValueList>
          <rim:Value>^Welby^Marcus^^MD^Dr</rim:Value>
1055      </rim:ValueList>
      </rim:Slot>
```

### 4.2.3.1.4.3 authorRole

**Description:**

1060 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.

The value is coded as an optional ebRIM Slot within an author Classification. See section
1065 4.2.3.1.4 for an example of author Classification. There may be multiple values within the
ebRIM Slot.

```
      <rim:Slot name="authorSpecialty">
        <rim:ValueList>
1070      <rim:Value>Cardiologist</rim:Value>
        </rim:ValueList>
      </rim:Slot>
```

### 4.2.3.1.4.4 authorSpecialty

**Description:**

1075 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 4.2.3.1.7-2.

**Coding:**

The format of the authorSpecialty value is String.

1080 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.

```
      <rim:Slot name="authorSpecialty">
1085    <rim:ValueList>
          <rim:Value>Cardiology</rim:Value>
        <rim:ValueList>
      </rim:Slot>
```

### 4.2.3.1.4.5 authorTelecommunication

1090 **Description:**

_____

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

**Coding:**

The format of the authorTelecommunication value is XTN; see table 4.2.3.1.7-2 for description
1095   of XTN format. This specification restricts the use to the following fields:

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

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

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

1100   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.

```
<rim:Slot name="authorTelecommunication">
  <rim:ValueList>
    <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value>
  </rim:ValueList>
</rim:Slot>
```
1105

### 4.2.3.1.5 UUIDs

1110   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
```

Alternatively, symbolic Ids may be used when referencing objects in the same submission
1115   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.

1120   ### 4.2.3.1.6 Extra Metadata Attributes

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

The following conditions shall apply.

1. All extra metadata content shall be in the form of Slots.

1125   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.

1130    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
1135    (DocumentEntry, SubmissionSet, Folder, or Association).

### 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.

1140    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 | ~ |

1145

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

| Data Type | Source Standard | Encoding Specification |
|---|---|---|
| Code | ITI TF | See section 4.2.3.1.2 |
| CX | HL7 V2.5 Identifier | 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: IdNumber^^^&OIDofAA&ISO 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: |

_____

_____

| Data Type | Source Standard | Encoding Specification |
|---|---|---|
| | | 543797436^^^&1.2.840.113619.6.197&ISO<br><br>Note that the '&' character must be properly encoded in the XML content. |
| CXi | HL7 V2 Identifier | 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.<br><br>• CXi.1 shall be present and hold the identifier value.<br><br>• CXi4 (Assigning Authority) shall be present when the identifier in CXi.1 is not globally unique and holds the identifier of the "domain" over which the ID Number represents a unique entity. It is formatted just like CX.4 in the CX datatype above.<br><br>• CXi.5 (Identifier Type Code) shall be present and chosen from either a URN defined by IHE, or a locally defined value.<br><br>• When the homeCommunityId is known, CX.6 shall be present and holds the homeCommunityId encoded as ISO, see CX.4 in the CX datatype above.<br><br>• No other components shall be present.<br><br>The following IHE values for CXi.5 (Identifier Type Codes) are defined:<br><br>**urn:ihe:iti:xds:2013:uniqueId**<br><br>This code shall be used when the identifier is a uniqueID from Document Sharing metadata attribute.<br><br>For example if uniqueID equals "1.3.6367.3.7^11379", and homeCommunityID is "1.2.3.4" then the CXi value is<br><br>`11379^^^&1.3.6367.3.7&ISO^urn:ihe:iti:xds:2013:uniqueId^&1.2.3.4&ISO`<br><br>**urn:ihe:iti:xds:2013:accession**<br><br>This code shall be used when the identifier is an accession number. It shall contain<br><br>• The accession number and<br><br>• for accession values that are not globally unique, the Assigning Authority shall be included.<br><br>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<br><br>`2013001^^^&1.2.3.4.5.6&ISO^urn:ihe:iti:xds:2013:accession`<br><br>**urn:ihe:iti:xds:2013:referral**<br><br>Referral number and assigning authority shall be present.<br><br>For example:<br><br>`201300001^^^1.2.3.4.5.6^urn:ihe:iti:xds:2013:referral`<br><br>**urn:ihe:iti:xds:2013:order**<br><br>Order number and assigning authority shall be present.<br><br>For example: |

_____

_____

| Data Type | Source Standard | Encoding Specification |
|---|---|---|
| | | `134467653^^^1.2.3.4.5.42.1^urn:ihe:iti:xds:2013:order` |
| DTM | HL7 V2.5 Date Time | This is a date/time value, represented as precisely as possible. All date time values in the registry are stored using Coordinated Universal Time [UTC]. "UTC" implies that time shall be converted from/to the local time. The format of these values is defined as the following regular expression: YYYY[MM[DD[hh[mm[ss]]]]] 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 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 |
| Field | HL7 V2.5 Message Segment | 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. 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 |
| 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 | 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) 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 |
| 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 |

_____

| Data Type | Source Standard | Encoding Specification |
|-----------|-----------------|------------------------|
| 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 |
| UTF-8 | RFC 3629 | Unicode standard |
| UUID | RFC 4122 | A DCE Universally Unique Identifier, represented in registry attributes using the URN syntax for UUIDs<br><br>e.g.,urn:uuid:9e0110f8-4748-4f1e-b0a8-cecae32209c7<br><br>Some Document Sharing profiles may allow use of symbolic Ids in certain conditions or locations. |
| XCN | HL7 V2.5 Extended Person Name | 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:<br><br>Identifier<br>Last Name<br>First Name<br>Second and Further Given Names<br>Suffix<br>Prefix<br>Assigning Authority<br><br>All of the HL7 v2.5 fields may be specified as optional components with the following restrictions:<br><br>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.<br><br>If component 1 (Id Number) is specified, component 9 (Assigning Authority) shall be present if available.<br><br>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'.<br><br>Any empty component shall be treated by the Document Registry as not specified. This is in compliance with HL7 v2.5.<br><br>Trailing delimiters are recommended to be trimmed off. Receiving actors shall ignore trailing delimiters. This is in compliance with HL7 v2.5.<br><br>An example of person name with Id number using this data type is as follows:<br><br>11375^Welby^Marcus^J^Jr. MD^Dr^^^&1.2.840.113619.6.197&ISO |
| XON | HL7 V2.5 Organization Name | This type provides the name and identification of an organization. This specification restricts the coding to the following fields:<br><br>XON.1 – Organization Name – this field is required<br><br>XON.6.2 – Assigning Authority Universal Id – this field is required if XON.10 is valued and not an OID<br><br>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"<br><br>XON.10 – Organization Identifier – this field is optional |

_____

| Data Type | Source Standard | Encoding Specification |
|---|---|---|
|  |  | 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. |
|  |  | 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 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. |
|  |  | Examples: |
|  |  | Some Hospital |
|  |  | Some Hospital^^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45 |
|  |  | Some Hospital^^^^^&1.2.3.4.5.6.7.8.9.1789&ISO^^^^45 |
| XTN | HL7 V2.5 Extended Telecommunications Number<br><br>Chapter 2A Section 89 | This type provides the telecommunications address of an entity (for example author, intended recipient). This specification restricts the use to the following fields:<br><br>XTN.3 – The type of telecommunication address. For example, e-mail addresses will have this valued with "Internet".<br>XTN.4 – the telecommunications address, e.g., name@example.com<br><br>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.<br><br>Example:<br><br>^^Internet^radiology@healthcare.example.org |

## 4.2.3.1.8 General format of DocumentEntry, Folder and SubmissionSet attribute tables

1150 The metadata attribute definition tables each have five columns:

1. The first column contains the name used to refer to the attribute in IHE documentation. It is not always related to the way the attribute is coded in ebRIM.

2. The second column contains a brief description of the attribute.

3. The third column identifies how the value of the attribute is coded. Table 4.2.3.1.7-2lists
1155 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
1160 held within the enclosing object. The possible values in this column are:

_____

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

- **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" ... >
```

Showing the entryUUID attribute coded as a value of the *id* XML 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>
```

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

- **ebRIM ExternalIdentifier** – indicates the value is held within an ExternalIdentifier object, using the XML element rim:ExternalIdentifier, providing an additional identifier to a registry object (see section 4.2.3.1.3).

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

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

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

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

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

### 4.2.3.1.9 Metadata Attribute Cardinality

Metadata attributes have several dimensions of cardinality.

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

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

_____

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,
4.2.3.3.1 through 4.2.3.3.12 for SubmissionSet, and 4.2.3.4.1 through 4.2.3.4.9 for Folder. The
1205   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.

### 4.2.3.2  DocumentEntry Attributes

The following metadata attributes shall be used to describe a Document Sharing DocumentEntry.
1210   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.

Please see section 4.2.3.1.8 for the general format of DocumentEntry, Folder and SubmissionSet
1215   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 |
|---|---|---|---|---|
| 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 particular kind of document (e.g., Prescription, Discharge Summary, Report). | Code | ebRIM Classification | 4.2.3.2.3 |
| comments | Comments associated with the Document. | String | ebRIM Description | 4.2.3.2.4 |
| confidentialityCode | The code specifying the level of confidentiality of the Document. | Code | ebRIM Classification | 4.2.3.2.5 |
| creationTime | The time the author created the document. Shall have a single value. | DTM | ebRIM Slot | 4.2.3.2.6 |
| entryUUID | A globally unique identifier used to identify the entry. | UUID | XML attribute | 4.2.3.2.7 |
| eventCodeList | This list of codes represents the main clinical acts, such as a colonoscopy or an appendectomy, being documented. | Code | ebRIM Classification | 4.2.3.2.8 |
| formatCode | Code globally uniquely specifying the format of the document. | Code | ebRIM Classification | 4.2.3.2.9 |
| Hash | Hash of the Document itself. | SHA1 hash | ebRIM Slot | 4.2.3.2.10 |

_____

_____

| DocumentEntry Metadata Attribute | Description | Data Type (Table 4.2.3.1.7-2) | Coding (Section 4.2.3.1.8) | Detail |
|---|---|---|---|---|
| healthcareFacility TypeCode | This code represents the type of organizational setting of the clinical encounter during which the documented act occurred. | Code | ebRIM Classification | 4.2.3.2.11 |
| homeCommunityId | A globally unique identifier for a community. | OID URN | home XML attribute | 4.2.3.2.12 |
| languageCode | Specifies the human language of character data in the document. Using codes from IETF RFC 3066 | String | ebRIM Slot | 4.2.3.2.13 |
| legalAuthenticator | Represents a participant who has legally authenticated or attested the document within the authorInstitution. | XCN | ebRIM Slot | 4.2.3.2.14 |
| limitedMetadata | Indicates whether the Document Entry was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source. | 4.2.3.2.29 | ebRIM Classification | 4.2.3.2.29 |
| mimeType | MIME type of the document. | MIME type | XML attribute | 4.2.3.2.15 |
| 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 that apply to the document. | CXi | ebRIM ExternalIdentifier | 4.2.3.2.28 |
| repositoryUniqueId | The globally unique identifier of the repository where the document is stored. | OID | ebRIM Slot | 4.2.3.2.18 |
| serviceStartTime | The start time the service being documented took place. | DTM | ebRIM Slot | 4.2.3.2.19 |
| serviceStopTime | The stop time the service being documented took place. | DTM | ebRIM Slot | 4.2.3.2.20 |
| Size | Size in bytes of the document. | Integer | ebRIM Slot | 4.2.3.2.21 |
| sourcePatientId | The sourcePatientId represents the subject of care medical record Identifier (e.g., Patient Id) in the local patient Identifier Domain of the Document Source. | CX | ebRIM Slot | 4.2.3.2.22 |
| sourcePatientInfo | This attribute contains demographic information of the patient to whose medical record this document belongs. | CX | ebRIM Slot | 4.2.3.2.23 |
| Title | The title of the document. | UTF-8 | ebRIM Name | 4.2.3.2.24 |
| typeCode | The code specifying the precise kind of document (e.g., Pulmonary History and Physical, Discharge Summary, Ultrasound Report). | Code | ebRIM Classification | 4.2.3.2.25 |
| uniqueId | The globally unique identifier assigned by the document creator to this document. | OID | ebRIM ExternalIdentifier | 4.2.3.1.3 |

_____

| DocumentEntry Metadata Attribute | Description | Data Type (Table 4.2.3.1.7-2) | Coding (Section 4.2.3.1.8) | Detail |
|---|---|---|---|---|
| URI | The URI for the document. | URI | ebRIM Slot | 4.2.3.2.27 |

### 4.2.3.2.1 DocumentEntry.author

**Description:**

1220 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

**Description:**

1225 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.

This attribute is typically omitted in a submission of new documents. If present in a submission,
1230 the submitted value is ignored. It is always set to Approved as a result of the successful submission of new documents. It may be changed to Deprecated under the primary responsibility of the creating entity.

**Coding:**

The format of the availabilityStatus value is a URN.

1235 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; however, this attribute is only returned on query, not specified in a submission type transaction.

1240

```
<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"
1245    >
...
```

### 4.2.3.2.3 DocumentEntry.classCode

**Description:**

1250 The code specifying the particular kind of document (e.g., Prescription, Discharge Summary, Report). Valid values for this metadata attribute are specified by the policies of the creating entity. It is recommended that the creating entity draws these values from a coding scheme providing a coarse level of granularity (about 10 to 100 entries). For example, XDS specifies that the XDS Affinity Domain will establish this list.

**Coding:**

1255 There shall be exactly zero or one ebRIM Classification containing a classCode for any DocumentEntry. See section 4.2.3.1.2 for a description of coding an ebRIM Classification. For the classCode metadata attribute, the classificationScheme shall be urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a.

The following example specifies classCode=10160-0 with display name "History of Medication
1260 Use" and coding scheme "2.16.840.1.113883.6.1" for the DocumentEntry labeled "ExampleDocument".

```
<rim:Classification
    classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"
1265    classifiedObject="ExampleDocument"
    id="IdExample_046"
    objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
    nodeRepresentation="10160-0"
1270    >
  <rim:Name>
    <rim:LocalizedString
        value="History of Medication Use"/>
  </rim:Name>
1275  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>2.16.840.1.113883.6.1</rim:Value>
    </rim:ValueList>
  </rim:Slot>
1280 </rim:Classification>
```

### 4.2.3.2.4 DocumentEntry.comments

**Description:**

Contains comments associated with the document.

**Coding:**

1285 Max length is unbounded.

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

The following example shows a comment for the document.
1290

_____

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

### 4.2.3.2.5 DocumentEntry.confidentialityCode

1295  **Description:**

The code specifying the level of confidentiality of the document. These codes are set by policy of the creating entity and issues related to highly sensitive documents are beyond the scope of metadata definition. These issues are expected to be addressed separately. confidentialityCode is part of a codification scheme.

1300  **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 and multiple values of confidentialityCode are coded by specifying multiple classification objects. For the confidentialityCode metadata
1305  attribute, the classificationScheme shall be urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f.

The following example specifies confidentialityCode="Example confidentialityCode Value" with display name "ExampleconfidentialityCodeDisplayName" and coding scheme "Example Scheme" for the DocumentEntry labeled "ExampleDocument".

```
1310  <rim:Classification
          classificationScheme=
          "urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
          classifiedObject="ExampleDocument"
          id="IdExample_046"
1315      objectType="urn:oasis:names:tc:ebxml-
              regrep:ObjectType:RegistryObject:Classification"
          nodeRepresentation="Example confidentialityCode Value"
        <rim:Name>
          <rim:LocalizedString
1320        value="ExampleconfidentialityCodeDisplayName"/>
        </rim:Name>
        <rim:Slot name="codingScheme">
          <rim:ValueList>
            <rim:Value>Example Scheme</rim:Value>
1325      </rim:ValueList>
        </rim:Slot>
      </rim:Classification>
```

### 4.2.3.2.6 DocumentEntry.creationTime

**Description:**

1330  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.

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

_____

1335    The following example shows a creationTime of December 25, 2004 21:20:10 Coordinated
        Universal Time (UTC).

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

### 4.2.3.2.7 DocumentEntry.entryUUID

**Description:**

1345    entryUUID is a globally unique identifier primarily intended for internal document management
        purposes. In contrast, the uniqueId attribute is used for external references (e.g., links, etc.).

**Coding:**

        Max length is unbounded. The format of the entryUUID value is UUID. Certain Document
        Sharing transactions may allow symbolic Ids (anything that does not have the urn:uuid: prefix) to
1350    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-
        9e5c-88b9045a96f6

```
1355   <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

1360    **Description:**

        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
1365    simply "Procedure Report" and the procedure was a "colonoscopy". When defining the value sets
        for eventCodes, they should not conflict with the values inherent in the classCode,
        practiceSettingCode or typeCode as such a conflict would create an ambiguous situation.

        **Coding:**

        The eventCodeList is coded with ebRIM Classification objects. See section 4.2.3.1.2 for a
1370    description of coding an ebRIM Classification. There may be zero or more ebRIM
        Classifications containing a code and additional eventCode entries are coded by specifying
        multiple classification objects. For the eventCodeList metadata attribute, the
        classificationScheme shall be urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4.

The following example specifies an eventCode="ExampleeventCode" with display name
1375  "eventCodeDisplayName" and coding scheme "Example Event Code Scheme" for the
DocumentEntry labeled "ExampleDocument".

```
       <rim:Classification
           classificationScheme=
1380       "urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4"
           classifiedObject="ExampleDocument"
           id="IdExample_048"
           objectType="urn:oasis:names:tc:ebxml-
               regrep:ObjectType:RegistryObject:Classification"
1385      nodeRepresentation="ExampleeventCode"
           >
         <rim:Name>
           <rim:LocalizedString value="eventCodeDisplayName"/>
         </rim:Name>
1390     <rim:Slot name="codingScheme">
           <rim:ValueList>
             <rim:Value>Example Event Code Scheme</rim:Value>
           </rim:ValueList>
         </rim:Slot>
1395   </rim:Classification>
```

### 4.2.3.2.9 DocumentEntry.formatCode

**Description:**

Globally unique code specifying the format of the document. Along with the typeCode, it should
provide sufficient information to allow potential consumer to know if it will be able to process
1400  the document.

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

Format codes may be specified by multiple organizations. Format codes defined by the ITI
1405  domain shall have names with the prefix

```
urn:ihe:iti:
```

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

```
urn:ihe:'domain initials':
```

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

1410  ```
urn:ad:'creating entity':
```

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

**Coding:**

There shall be zero or one ebRIM Classification containing a formatCode. See section 4.2.3.1.2
for a description of coding an ebRIM Classification. For the formatCode metadata attribute, the
1415  classificationScheme shall be urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d. Any valid
URN may be used as a formatCode.

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

1420
```
<rim:Classification
    classificationScheme=
    "urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"
    classifiedObject="ExampleDocument"
    id="IdExample_049"
    objectType="urn:oasis:names:tc:ebxml-
        regrep:ObjectType:RegistryObject:Classification"
    nodeRepresentation="ExampleformatCode"
    >
  <rim:Name>
    <rim:LocalizedString value="ExampleformatCodeDisplayName"/>
  </rim:Name>
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>Example Format Code Scheme</rim:Value>
    </rim:ValueList>
  </rim:Slot>
</rim:Classification>
```

### 4.2.3.2.10    DocumentEntry.hash

1440 **Description:**

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

**Coding:**

1445 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.

1450
```
<rim:Slot name="hash">
  <rim:ValueList>
    <rim:Value>da39a3ee5e6b4b0d3255bfef95601890afd80709</rim:Value>
  </rim:ValueList>
1455 </rim:Slot>
```

### 4.2.3.2.11    DocumentEntry.healthcareFacilityTypeCode

**Description:**

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

1460 In some cases, the setting of the encounter is inherent in the typeCode, such as "Diabetes Clinic
Progress Note". healthcareFacilityTypeCode shall be equivalent to or further specialize the value

inherent in the typeCode; for example, where the typeCode is simply "Clinic Progress Note" and
the value of healthcareFacilityTypeCode is "private clinic".

**Coding:**

1465   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`.

The following example specifies healthcareFacilityTypeCode =
1470   "ExamplehealthcareFacilityTypeCode " with display name
"healthcareFacilityTypeCodeDisplayName " and coding scheme "Example Healthcare Facility
Scheme" for the DocumentEntry labeled "ExampleDocument".

```
       <rim:Classification
1475       classificationScheme=
           "urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"
           classifiedObject="ExampleDocument"
           id="IdExample_050"
           objectType="urn:oasis:names:tc:ebxml-
1480          regrep:ObjectType:RegistryObject:Classification"
           nodeRepresentation="ExamplehealthcareFacilityTypeCode">
         <rim:Name>
           <rim:value="healthcareFacilityTypeCodeDisplayName"/>
         </rim:Name>
1485     <rim:Slot name="codingScheme">
           <rim:ValueList>
             <rim:Value>Example Healthcare Facility Scheme</rim:Value>
           </rim:ValueList>
         </rim:Slot>
1490   </rim:Classification>
```

### 4.2.3.2.12   DocumentEntry.homeCommunityId

**Description:**

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

1495   **Coding:**

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

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

```
1500   <rim:ExtrinsicObject home="urn:oid:1.2.3" ...>
          ...
       </rim:ExtrinsicObject>
```

### 4.2.3.2.13   DocumentEntry.languageCode

**Description:**

1505   Specifies the human language of character data in the document.

_____

**Coding:**

Max length is 256 characters. The values of the attribute are language identifiers as described by the IETF (Internet Engineering Task Force) RFC 3066.

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

1510 The following example shows a languageCode of en-CA.

```
<rim:Slot name="languageCode">
  <rim:ValueList>
    <rim:Value>en-CA</rim:Value>
  </rim:ValueList>
</rim:Slot>
```
1515

### 4.2.3.2.14    DocumentEntry.legalAuthenticator

**Description:**

Represents a participant within the authorInstitution who has legally authenticated or attested the
1520 document. Legal authentication implies that a document has been signed manually or electronically by the legalAuthenticator.

**Coding:**

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
1525 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.

```
<rim:Slot name="legalAuthenticator">
  <rim:ValueList>
    <rim:Value>^Welby^Marcus^^^Dr^MD</rim:Value>
  </rim:ValueList>
</rim:Slot>
```
1530

### 4.2.3.2.15    DocumentEntry.mimeType

**Description:**

1535 MIME type of the document in the Repository

**Coding:**

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

1540 In this example the MIME type is "application/pdf".

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

_____

_____

1545 **4.2.3.2.16    DocumentEntry.patientId**

**Description:**

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

Within a submission request, the value of patientId of the DocumentEntries shall match that of
1550    the SubmissionSet.


**Coding:**

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

It shall contain two parts:

1555    Assigning Authority Domain Id (organization that issued the Id),

An Id from the above Assigning Authority.

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

1560    Coded as an ebRIM ExternalIdentifier. See section 4.2.3.1.3 for a description of coding an ebRIM ExternalIdentifier. This element references, and is contained in, the ExtrinsicObject representing the DocumentEntry. There shall be zero or one patientId value.

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

1565
```
<rim:ExternalIdentifier
    identificationScheme=
      "urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"
    value="6578946^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO"
1570    id="IdExample_051"
    objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:ExternalIdentifier"
    registryObject="DocumentEntry01">
  <rim:Name>
1575    <rim:LocalizedString value="XDSDocumentEntry.patientId "/>
  </rim:Name>
</rim:ExternalIdentifier>
```

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

1580 **4.2.3.2.17    DocumentEntry.practiceSettingCode**

**Description:**

The code specifying the clinical specialty where the act that resulted in the document was performed (e.g., Family Practice, Laboratory, Radiology). It is suggested that the creating entity

_____

1585 draws these values from a coding scheme providing a coarse level of granularity (about 10 to 100 entries).

**Coding:**

There shall be zero or one ebRIM Classification containing a practiceSettingCode. See section 4.2.3.1.2 for a description of coding an ebRIM Classification. For the practiceSettingCode metadata attribute, the classificationScheme shall be urn:uuid:cccf5598-8b07-4b77-a05e-
1590 ae952c785ead.

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

```
1595    <rim:Classification
           ClassificationScheme="urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead"
           classifiedObject="ExampleDocument"
           id="IdExample_052"
           objectType="urn:oasis:names:tc:ebxml-
1600         regrep:ObjectType:RegistryObject:Classification"
           nodeRepresentation="ExamplepracticeSettingCode">
         <rim:Name>
           <rim:LocalizedString
             value="ExamplepracticeSettingCodeDisplayName"/>
1605     </rim:Name>
         <rim:Slot name="codingScheme">
           <rim:ValueList>
             <rim:Value>Example Practice Setting Code Scheme</rim:Value>
           </rim:ValueList>
1610     </rim:Slot>
       </rim:Classification>
```

### 4.2.3.2.18    DocumentEntry.repositoryUniqueId

**Description:**

The globally unique, immutable, identifier of the repository where the document is stored.

1615 **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.

```
1620    <rim:Slot name="repositoryUniqueId">
         <rim:ValueList>
           <rim:Value>1.3.6.1.4.5</rim:Value>
         </rim:ValueList>
       </rim:Slot>
```

1625 ### 4.2.3.2.19    DocumentEntry.serviceStartTime

**Description:**

_____

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

1630

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

**Coding:**

Max length is 256 characters. The format of the serviceStartTime value is DTM; see table
1635   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.

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

1640
```
<rim:Slot name="serviceStartTime">
  <rim:ValueList>
    <rim:Value>20041225212010</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

1645   ### 4.2.3.2.20   DocumentEntry.serviceStopTime

**Description:**

Represents the stop time of the service being documented took place (clinically significant, but not necessarily when the document was produced or approved). This may be the same as the encounter time in case the service was delivered during an encounter. Encounter time is not
1650   coded in metadata but may be coded within the document.

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

**Coding:**

Max length is 256 characters. The format of the serviceStartTime value is DTM; see table
1655   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.

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

1660
```
<rim:Slot name="serviceStopTime">
  <rim:ValueList>
    <rim:Value>20041225232010</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

### 4.2.3.2.21   DocumentEntry.size

1665   **Description:**

_____

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.

1670    The following example shows a size value of 3654 bytes.

```
<rim:Slot name="size">
  <rim:ValueList>
    <rim:Value>3654</rim:Value>
  </rim:ValueList>
</rim:Slot>
```
1675

### 4.2.3.2.22    DocumentEntry.sourcePatientId

**Description:**

The sourcePatientId represents the subject of care's medical record Identifier (e.g., Patient Id) in
1680    the local patient identifier domain of the creating entity at the time the document entry was submitted.

**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.

1685    It shall contain only two parts:

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.

1690
```
<rim:Slot name="sourcePatientId">
  <rim:ValueList>
    <rim:Value>j98789^^^&amp;1.2.3.4.343.1&amp;ISO</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

1695    ### 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.

This information typically includes: the patient first and last name, sex, and birth date. Policies at
1700    the creating entity may require more or less specific information and format.

**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).

1705

```
<rim:Slot name="sourcePatientInfo">
  <rim:ValueList>  <rim:Value>PID-3|DTP-1^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;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>
```

1710

1715    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)

1720    - PID-8 (source patient gender)

M – Male    F – Female

O – Other    U – Unknown

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

1725    ### 4.2.3.2.24    DocumentEntry.title

**Description:**

Represents the title of the document.

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

1730    **Coding:**

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

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.

1735    The following example shows a title for the DocumentEntry.

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

1740

_____

```
        <rim:LocalizedString value="Example Document Title"/>
      </rim:Name>
      ...
1745  </rim:ExtrinsicObject>
```

### 4.2.3.2.25    DocumentEntry.typeCode

**Description:**

The code specifying the precise kind of document (e.g., Pulmonary History and Physical,
Discharge Summary, Ultrasound Report). It is recommended that the creating entity draw these
1750  values from a coding scheme providing a fine level of granularity.

**Coding:**

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
1755  "typeCodeDisplayName " and coding scheme " Example Type Code Scheme " for the
DocumentEntry labeled "ExampleDocument".

```
      <rim:Classification
          classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"
1760      classifiedObject="ExampleDocument"
          nodeRepresentation="ExampletypeCode"
          id="IdExample_053"
          objectType="urn:oasis:names:tc:ebxml-
              regrep:ObjectType:RegistryObject:Classification">
1765    <rim:Name>
          <rim:LocalizedString value="typeCodeDisplayName" />
        </rim:Name>
        <rim:Slot name="codingScheme">
          <rim:ValueList>
1770        <rim:Value>Example Type Code Scheme</rim:Value>
          </rim:ValueList>
        </rim:Slot>
      </rim:Classification>
```

### 4.2.3.2.26    DocumentEntry.uniqueId

1775  **Description:**

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

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.

1780  This unique identifier may be used in other documents to reference this document.

**Coding:**

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

_____

Coded as an ebRIM ExternalIdentifier (see section 4.2.3.1.3 for a description of coding an
ebRIM ExternalIdentifier) which references, and is contained in, the ExtrinsicObject
1785   representing the DocumentEntry. There shall be only a single uniqueId value.

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
1790   coded without the extension then the '^' character shall not be included.)

```
<rim:ExternalIdentifier
    identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"
    value="1.2.3.4.5.6.78901.2345.6.7^123456"
1795    id="IdExample_054"
    objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier"
    registryObject="DocumentEntry01">
  <rim:Name>
    <rim:LocalizedString value="XDSDocumentEntry.uniqueId"/>
1800  </rim:Name>
</rim:ExternalIdentifier>
```

### 4.2.3.2.27    DocumentEntry.URI

**Description:**

The URI attribute contains the URI for the document.

1805   **Coding:**

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

```
<rim:Slot name="URI">
  <rim:ValueList>
1810    <rim:Value>DOC001.XML</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

### 4.2.3.2.28    DocumentEntry.referenceIdList

**Description:**

1815   This list contains zero or more Identifiers. These Identifiers may be internal or external
identifiers, e.g., Identifiers may be Accession Numbers, Order Numbers, Referral Request
Identifiers, XDS Workflow Instance Identifiers, etc. The referenceIdList contains Identifiers CXi
encoded, as specified in table 4.2.3.1.7-2.

XDS Document Registry actors supporting the ReferenceId option shall preserve this value.
1820   When the ReferenceId option is not declared, the Document Registry shall handle this extra
metadata according to the historic rules outlined in section 4.2.3.1.6 "Extra Metadata Elements".

**Coding:**

Coded as an ebRIM Slot. May have multiple values. Max length for each value is 256 characters.

The name of the slot in the metadata shall be "urn:ihe:iti:xds:2013:referenceIdList".

1825 For example encoding 2 values in the referenceIdList

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

### 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.

1840 **Coding:**

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

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

1845

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

### 4.2.3.3 SubmissionSet Attributes

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

Each attribute shown is an attribute on the RegistryPackage object defining the SubmissionSet.
1855 The attribute name is defined with a prefix of the object type of SubmissionSet when referenced by other objects, for example SubmissionSet.sourceId.

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

1860     **Table 4.2.3.3-1: Submission Set Metadata Attribute Definitions** (previously Table 4.1-6)

| SubmissionSet Metadata Attribute | Description | Form | Coding | Detail |
|---|---|---|---|---|
| author | The humans and/or machines that authored the document in the SubmissionSet. This attribute contains the sub-attributes: authorInstitution, | 4.2.3.3.1 | ebRIM Classification | 4.2.3.3.1 |

_____

| SubmissionSet Metadata Attribute | Description | Form | Coding | Detail |
|---|---|---|---|---|
| | authorPerson, authorRole, authorSpecialty. | | | |
| availabilityStatus | The lifecycle status of the SubmissionSet | Predefined URN | XML attribute | 4.2.3.3.2 |
| comments | Comments associated with the SubmissionSet. | String | ebRIM Description | 4.2.3.3.3 |
| contentTypeCode | The code specifying the type of clinical activity that resulted in placing the documents in this SubmissionSet. | Code | ebRIM Classification | 4.2.3.3.4 |
| entryUUID | A globally unique identifier used to identify the SubmissionSet. | UUID | XML attribute | 4.2.3.3.5 |
| homeCommunityId | A globally unique identifier for a community. | OID URN | home XML attribute | 4.2.3.3.6 |
| intendedRecipient | The organization(s) or person(s) for whom the SubmissionSet is intended. | See 4.2.3.3.7 | ebRIM Slot | 4.2.3.3.7 |
| limitedMetadata | A flag that the associated SubmissionSet was created using the less rigorous metadata requirements as defined for the Metadata-limited Document Source. | See 4.2.3.3.13 | ebRIM Classification | 4.2.3.3.13 |
| patientId | The patientId represents the subject of care whose longitudinal record is being reflected in this SubmissionSet. | CX | ebRIM ExternalIdentifier | 4.2.3.3.8 |
| sourceId | OID identifying the instance of the creating entity that contributed the SubmissionSet. | OID | ebRIM ExternalIdentifier | 4.2.3.3.8 |
| submissionTime | Point in time at the creating entity when the SubmissionSet was created. | DTM | ebRIM Slot | 4.2.3.3.10 |
| title | The title of the SubmissionSet. | UTF-8 | ebRIM Name | 4.2.3.3.11 |
| uniqueId | A globally unique identifier for the SubmissionSet assigned by the creating entity. | OID | ebRIM ExternalIdentifier | 4.2.3.3.12 |

### 4.2.3.3.1 SubmissionSet.author

**Description:**

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

1865   The classificationScheme shall be urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d

### 4.2.3.3.2 SubmissionSet.availabilityStatus

**Description:**

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

_____

1870 **Coding:**

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

```
<rim:RegistryPackage
     id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855"
1875     status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
> ...
```

### 4.2.3.3.3 SubmissionSet.comments

**Description:**

Contains comments associated with the SubmissionSet.

1880 **Coding:**

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

The following example shows a comment for the SubmissionSet.

1885
```
<rim:Description>
  <rim:LocalizedString value = "comment associated with SubmissionSet"/>
</rim:Description>
```

1890 ### 4.2.3.3.4 SubmissionSet.contentTypeCode

**Description:**

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

1895 **Coding:**

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

The following example specifies contentTypeCode="ExamplecontentTypeCode" with display name "Example contentTypeCodeDisplayName" and coding scheme "Example
1900 contentTypeCode Scheme" for the SubmissionSet labeled "ExampleSubmissionSet".

```
<rim:Classification
     classificationScheme="urn:uuid:aa543740-bdda-424e-8c96-df4873be8500"
     classifiedObject="ExampleSubmissionSet"
1905     id="IdExample_056"
     objectType="urn:oasis:names:tc:ebxml-
         regrep:ObjectType:RegistryObject:Classification"
     nodeRepresentation="ExamplecontentTypeCode"
     >
1910   <rim:Name>
     <rim:LocalizedString value="Example contentTypeCodeDisplayName" />
   </rim:Name>
   <rim:Slot name="codingScheme">
```

_____

```
1915        <rim:ValueList>
              <rim:Value>Example Specific Value</rim:Value>
            </rim:ValueList>
          </rim:Slot>
        </rim:Classification>
```

### 4.2.3.3.5 SubmissionSet.entryUUID

1920 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.

1925 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`

```
1930    <rim:RegistryPackage
          id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6"
          objectType=
            "urn:oasis:names:tc:ebxml-        regrep:ObjectType:RegistryObject:RegistryPackage"
        > …
```

### 4.2.3.3.6 SubmissionSet.homeCommunityId

1935 **Description:**

A globally unique identifier for a community.

**Coding:**

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.

1940 See TF-2a:3.18.4.1.2.3.8 and ITI TF-2b: 3.38.4.1.2.1.

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

### 4.2.3.3.7 SubmissionSet.intendedRecipient

1945 **Description:**

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
1950 organization for all the persons, avoiding errors in the transmission of the documents.

**Coding:**

_____

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
1955    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
1960    the organization details and another organization without the person details, and just a
telecommunications address.

```
<rim:Slot name="intendedRecipient">
  <rim:ValueList>
    <rim:Value>
      Some
Hospital^^^^^^^^^1.2.3.9.1789.45|^Wel^Marcus^^^Dr^MD|^^Internet^mwel@healthcare.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>
```

### 4.2.3.3.8 SubmissionSet.patientId

**Description:**

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

1980    **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.

1985    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
1990    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.

_____

_____

```
1995    <rim:ExternalIdentifier
            identificationScheme="urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446"
            value="6578946^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO"
            id="IdExample_057"
            objectType="urn:oasis:names:tc:ebxml-
2000            regrep:ObjectType:RegistryObject:ExternalIdentifier"
            registryObject="SubmissionSet01">
        >
        <rim:Name>
            <rim:LocalizedString value = "XDSSubmissionSet.patientId"/>
2005    </rim:Name>
        </rim:ExternalIdentifier>
```

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

### 4.2.3.3.9 SubmissionSet.sourceId

2010    **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.

**Coding:**

2015    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.

In the example below the sourceId is "1.3.6.1.4.1.21367.2005.3.7".

2020
```
        <rim:ExternalIdentifier
            identificationScheme="urn:uuid:554ac39e-e3fe-47fe-b233-965d2a147832"
            value="1.3.6.1.4.1.21367.2005.3.7"
            id="IdExample_058"
2025        objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier"
            registryObject="SubmissionSet01">
        <rim:Name>
            <rim:LocalizedString value = "XDSSubmissionSet.sourceId"/>
        </rim:Name>
2030    </rim:ExternalIdentifier>
```

### 4.2.3.3.10    SubmissionSet.submissionTime

**Description:**

Represents the point in time at the creating entity when the Submission Set was submitted.

This shall be provided by the submitting system.

2035    **Coding:**

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
2040 Universal Time (UTC).

```
<rim:Slot name="submissionTime">
  <rim:ValueList>
    <rim:Value>20041225212010</rim:Value>
2045  </rim:ValueList>
</rim:Slot>
```

### 4.2.3.3.11    SubmissionSet.title

**Description:**

2050 Shall contain the title of the Submission Set.

**Coding:**

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
2055 SubmissionSet.

The following example shows a title for the SubmissionSet.

```
<rim:Name>
  <LocalizedString value="Example Submission Set Title"/>
2060 </rim:Name>
```

### 4.2.3.3.12    SubmissionSet.uniqueId

**Description:**

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

2065 **Coding:**

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.

2070 The following example shows a uniqueId of 1.2.3.4.5.

```
<rim:ExternalIdentifier
    identificationScheme="urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8"
    id="IdExample_059"
2075 objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier"
    value="1.2.3.4.5"
    registryObject="SubmissionSet01">
  <rim:Name>
    <rim:LocalizedString value = "XDSSubmissionSet.uniqueId"/>
```

_____

_____

2080
```
    </rim:Name>
</rim:ExternalIdentifier>
```

### 4.2.3.3.13    SubmissionSet.limitedMetadata

**Description:**

2085 Indicates whether the SubmissionSet was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source actor.

**Coding:**

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

2090 The following example marks the "SubmissionSet" SubmissionSet as created via the less rigorous metadata requirements.

```
<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"/>
 (…)
</RegistryPackage>
```
2095



2100

## 4.2.3.4  Folder Attributes

The following metadata elements shall be used to describe a Folder. Each attribute shown below is an attribute on the RegistryPackage object defining the Folder. The attribute name is defined
2105 with a prefix of the object type of Folder when referenced by other objects, for example Folder.patientId. Please 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 |
|---|---|---|---|---|
| availabilityStatus | The lifecycle status of the Folder | Predefined URN | XML attribute | 4.2.3.4.1 |
| codeList | Shall contain the set of codes specifying the type of clinical activities that resulted in placing DocumentEntries in this Folder. | Code | ebRIM Classification | 4.2.3.4.2 |
| Comments | Comments associated with the Folder. | String | ebRIM Description | 4.2.3.4.3 |
| entryUUID | This globally unique identifier used to identify the Folder. | UUID | XML attribute | 4.2.3.4.4 |
| homeCommunityId | A globally unique identifier for a community. | OID URN | XML attribute | 4.2.3.4.5 |

_____

_____

| Folder Metadata Attribute | Description | Form | Coding | Detail |
|---|---|---|---|---|
| lastUpdateTime | Most recent point in time that the Folder has been modified. | DTM | ebRIM Slot | 4.2.3.4.6 |
| limitedMetadata | A flag that the associated Folder was created using the less rigorous metadata requirements as defined for the Metadata-Limited | 4.2.3.4.10 | ebRIM Classification | 4.2.3.4.10 |
| patientId | The patientId represents the subject of care of the DocumentEntries within the Folder. | CX | ebRIM ExternalIdentifier | 4.2.3.1.3 |
| Title | The title of the Folder. | String | ebRIM Name | 4.2.3.4.8 |
| uniqueId | Globally unique identifier for the Folder. | OID | ebRIM ExternalIdentifier | 4.2.3.4.9 |

### 2110 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:**

2115 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.

```
2120  <rim:RegistryPackage
       id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855"
       status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
      > ...
```

### 4.2.3.4.2 Folder.codeList

2125 **Description:**

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

**Coding:**

2130 Coded as an ebRIM classification. See section 4.2.3.1.2 for a description of coding an ebRIM Classification. Shall have zero or more values. Code multiple values by creating multiple classification objects.

_____

The following example specifies contentTypeCode="ExamplecontentTypeCode" with display
name "ExamplecodeList CodeDisplayName" and coding scheme "Example coding scheme" for
2135 the Folder labeled "ExampleFolder".

```
<rim:Classification
    classificationScheme= "urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5"
    classifiedObject="ExampleFolder"
2140    id="IdExample_062"
    objectType="urn:oasis:names:tc:ebxml-
      regrep:ObjectType:RegistryObject:Classification"
    nodeRepresentation="Examplecode"
    >
2145   <rim:Name>
    <rim:LocalizedString
        value="ExamplecodeList CodeDisplayName" />
   </rim:Name>
   <rim:Slot name="codingScheme">
2150    <rim:ValueList>
       <rim:Value>Example coding scheme</rim:Value>
     </rim:ValueList>
   </rim:Slot>
</rim:Classification>
```

### 2155 4.2.3.4.3 Folder.comments

**Description:**

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

Note: Prior to the availability of the Folder.title attribute, the comments attribute might have been used to hold the title of
the folder (folder name). With the addition of the title attribute, the comments attribute shall not hold the folder
2160 name.

**Coding:**

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

The value of the comments attribute is coded in XML as the "value" attribute of the
LocalizedString element within the ebRIM Description structure. There can be at most one
2165 ebRIM Description structure per DocumentEntry.

The following example shows a comment for the Folder.

```
<rim:Description>
   <rim:LocalizedString value = "comments"/>
2170 </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
2175 documents (e.g., in links within other documents).

**Coding:**

_____

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

2180  The value of the entryUUID is coded in the id XML attribute on the ExtrinsicObject representing
the DocumentEntry. In the example below, the entryUUID is

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

```
<rim:RegistryPackage
  id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a9ab6"
> ...
```
2185

### 4.2.3.4.5 Folder.homeCommunityId

**Description:**

A globally unique identifier for a community.

2190  **Coding:**

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

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

```
<rim:RegistryPackage home="urn:oid:1.2.3" ...>
   ...
</rim:RegistryPackage>
```
2195

### 4.2.3.4.6 Folder.lastUpdateTime

**Description:**

2200  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.

2205  The following example shows a last update of December 25, 2004 21:20:10 Coordinated
Universal Time (UTC).

```
<rim:Slot name="lastUpdateTime">
  <rim:ValueList>
    <rim:Value>20041225212010</rim:Value>
  </rim:ValueList>
</rim:Slot>
```
2210

### 4.2.3.4.7 Folder.patientId

**Description:**

_____

_____

2215    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 patiendId as the Folder.

**Coding:**

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

2220    It shall contain two parts:

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

An Id from the Assigning Authority.

No other values are allowed, as specified for the CX type in table 4.2.3.1.7-2 above. Using HL7 terminology, no other values are allowed in the components of the coded value, nor are further
2225    subcomponents allowed.

```
<rim:ExternalIdentifier
    identificationScheme="urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a"
    value="6578946^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO"
2230    id="IdExample_057"
    objectType="urn:oasis:names:tc:ebxml-
        regrep:ObjectType:RegistryObject:ExternalIdentifier"
    >
  <rim:Name>
2235    <rim:LocalizedString value = "XDSFolder.patientId"/>
  </rim:Name>
</rim:ExternalIdentifier>
```

This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or
2240    External Identifier elements except as described above.

### 4.2.3.4.8 Folder.title

**Description:**

Shall contain the Folder name.

2245    Note: Prior to the availability of this attribute the comments attribute might have been used to hold the title of the folder (folder name). With the addition of this attribute the comments attribute shall not hold the folder name.

**Coding:**

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

Max length is 256 characters.

2250    In this example the title is "title".

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

_____

_____

2255 **4.2.3.4.9 Folder.uniqueId**

**Description:**

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

**Coding:**

Max length is 256 characters. Encoded as the ExternalIdentifier. See section 4.2.3.1.3 for a
2260 description of coding an ebRIM ExternalIdentifier. Shall have only a single value. Shall be of
OID format.

```
<rim:ExternalIdentifier
    identificationScheme="urn:uuid:75df8f67-9973-4fbe-a900-df66cefecc5a"
    id="IdExample_059"
    objectType="urn:oasis:names:tc:ebxml-
        regrep:ObjectType:RegistryObject:ExternalIdentifier"
    value="1.3.6.1.4.1.21367.2005.3.7.3670984664"
    registryObject="Folder01">
  <rim:Name>
    <rim:LocalizedString value = "XDSFolder.uniqueId"/>
  </rim:Name>
</rim:ExternalIdentifier>
```

2275 This ExternalIdentifier shall not contain other Slot, Name, Description, Classification, or
External Identifier elements except as described above.

## 4.2.3.4.10    Folder.limitedMetadata

**Description:**

Indicates whether the Folder was created using the less rigorous requirements of metadata as
2280 defined for the Metadata-Limited Document Source actor.

**Coding:**

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

The following example marks the "Folder" Folder as created via the less rigorous metadata
2285 requirements.

```
<RegistryPackage id="Folder">
 (…)
```

2290
```
<Classification classifiedObject="Folder" classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-
ea235fbc94c2" id="ID_061" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"/>
```

2295
```
    <Classification classifiedObject="Folder"
        classificationNode="urn:uuid:2c144a76-29a9-4b7c-af54-b25409fe7d03"/>
 (…)
```

```
</RegistryPackage>
```

_____

_____

### 4.2.4  Error Reporting

2300    Registry Services schema (ebRS 3.0) defines the RegistryError element for reporting details of errors or warnings.

### 4.2.4.1  RegistryErrors Element

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

All IHE transactions that carry the RegistryError element shall return these attributes with each
2305    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 4.2.4.1-1when 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. |

2310    An example of an error response reporting two errors using ebRS and ebRIM 3.0 is:

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

_____

Tables 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.

2335 **Table 4.2.4.1-1: Error Codes** (previously Table 4.1-11)

| Error Code[1] | Discussion | Transaction (See Note 1) |
|---|---|---|
| DocumentQueued | An XDR Recipient queued the document for future manual matching to a patient | 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 |
| XDSDocumentUniqueIdError | The document associated with the uniqueId is not available. This could be because the document is not available, the requestor is not authorized to access that document or the document is no longer available. | RS, XGR |
| XDSDuplicateUniqueIdInRegistry | UniqueId received was not unique. UniqueId could have been attached to SubmissionSet or Folder. codeContext shall indicate which and the value of the non-unique uniqueId. This error cannot be thrown for DocumentEntry. See XDSNonIdenticalHash. | P,R<br>P |
| XDSExtraMetadataNotSaved | This warning is returned if extra metadata was present but not saved. | P,R |
| XDSMissingDocument | DocumentEntry exists in metadata with no corresponding attached document | P |
| XDSMissingDocumentMetadata | MIME package contains MIME part with Content-Id header not found in metadata | P |

_____

[1] The XDS Metadata Update Supplement adds error codes to this table.

_____

| Error Code[1] | Discussion | Transaction (See Note 1) |
|---|---|---|
| XDSMissingHomeCommunityId | A value for the homeCommunityId is required and has not been specified | SQ, XGQ, RS, XGR |
| XDSNonIdenticalHash | Document being registered was a duplicate (uniqueId already in registry) but hash does not match. codeContext indicates UniqueId. | R |
| XDSNonIdenticalSize | Document being registered was a duplicate (uniqueId already in registry) but size does not match. CodeContext indicates UniqueId. | R |
| XDSPatientIdDoesNotMatch | This error is thrown when the patient Id is required to match and does not. The codeContext shall indicate the value of the Patient Id and the nature of the conflict. | P,R |
| XDSRegistryBusy<br><br>XDSRepositoryBusy | Too much activity | P,R,SQ, XGQ<br><br>P, RS, XGR |
| XDSRegistryDeprecatedDocumentError | The transaction was rejected because it submitted an Association referencing a deprecated document. | P,R |
| XDSRegistryDuplicateUniqueIdInMessage<br><br>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<br><br>XDSRepositoryError | Internal Error<br><br>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.<br><br>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<br><br>P, RS. XGR |
| XDSRegistryMetadataError<br><br>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<br><br>XDSRepositoryOutOfResources | Resources are low. | P,R,SQ, XGQ<br><br>P, RS, XGR |
| 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[1] | 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. | Q,SQ, XGQ |
| XDSUnavailableCommunity | A community which would have been contacted was not available. See Note 2. | SQ, RS |
| XDSUnknownCommunity | A value for the homeCommunityId is not recognized | SQ, XGQ, RS, XGR |
| XDSUnknownPatientId | Patient Id referenced in metadata is not known by the receiving actor. The codeContext shall include the value of patient Id in question. | P,R, XGQ  Note: this error code is not used in the response to Registry Stored 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

Note 2:

Two examples of the use of error code XDSUnavailableCommunity are:

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.

## 4.2.4.2  Error responses

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

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

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

_____

_____

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

```
urn:ihe:iti:2007:ResponseStatusType:PartialSuccess
```

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

2360    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.

2365            **Table 4.2.4.2-1: [ITI-41] Provide and Register Document Set-b Responses**

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

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

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

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

| AdhocQueryResponse status | RegistryErrorList element | Result |
|---|---|---|
| urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success | May be present. If present will contain one or more RegistryError elements with warning severity; none with error severity | Results shall be returned. Results may contain zero or more entries. |
| urn:ihe:iti:2007:ResponseStatusType:PartialSuccess | Present, contains one or more | Results shall be returned. |

_____

_____

| AdhocQueryResponse status | RegistryErrorList element | Result |
|---|---|---|
| | RegistryError elements. At least one has error severity; others may have warning severity. | 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 |

2370

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

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

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

### 4.2.5  Metadata Vocabulary

2375   The UUIDs in the following tables shall be used in constructing and interpreting XDS metadata.

### 4.2.5.1  Submission Set Object UUIDs

| UUID | Use/meaning |
|---|---|
| urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd | SubmissionSet ClassificationNode |
| urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d | author External Classification Scheme |
| urn:uuid:aa543740-bdda-424e-8c96-df4873be8500 | contentTypeCode External Classification Scheme |
| urn:uuid: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 |

_____

_____

### 4.2.5.2  Document Entry Object

| UUID | Use/meaning |
|------|-------------|
| urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1 | DocumentEntry ClassificationNode |
| 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 |
| 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:cccf5598-8b07-4b77-a05e-ae952c785ead | practiceSettingCode External Classification Scheme |
| urn:uuid:f0306f51-975f-434e-a61c-c59651d33983 | typeCode External Classification Scheme |
| urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab | uniqueId ExternalIdentifier |

### 4.2.5.3  Folder Object

| UUID | Use/meaning |
|------|-------------|
| urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2 | Folder ClassificationNode |
| urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5 | codeList External Classification Scheme |
| urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a | patientId External Identifier |
| urn:uuid:75df8f67-9973-4fbe-a900-df66cefecc5a | uniqueId External Identifier |

## 4.3  Additional Document Sharing Requirements

2380   This section includes profile specific requirements.

### 4.3.1  Requirements on Submission Type Transactions

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

- ITI-42 Register Document Set-b – transfers only Document Sharing metadata

2385
- ITI-41 Provide and Register Document Set-b – transfers Document Sharing metadata
  along with document content

- ITI-32 Distribute Document Set on Media - transfers Document Sharing metadata along
  with document content

The Document Sharing metadata Objects found in the Submission Request of a Submission Type
2390   Transaction are:

- SubmissionSet metadata

_____

- Metadata for new or existing documents

- Folders to be created

- Associations (various relationships and HasMember as needed)

2395 An XDS Registry Submission Request is the collection of metadata transferred between a
Document Repository and a Document Registry. This request is part of the Register Document
Set-b [ITI-42] transaction.

An XDS Repository Submission Request is the collection of metadata and documents transferred
between a Document Source and a Document Repository. This request is the information
2400 payload of the Provide and Register Document Set message of the Provide and Register
Document Set-b [ITI-41] transaction.

### 4.3.1.1  Submission Metadata Attribute Optionality

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

2405 The Actor/Transaction pairs addressed by this section are as follows:

**Table 4.3.1.1-1: 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 |

For each actor/transaction pair across the top of table 4.3.1.1-3, and each metadata attribute row,
2410 the cell indicates the requirement for that actor when creating a submission request using the
paired transaction. The requirements are expressed through these codes:

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

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

_____

2415

**Table 4.3.1.1-3: Metadata Attribute Optionality**

| Metadata Element | Metadata Attribute | XDS DS | XDS DR | XDM MC | XDR DS | XDR MS |
|---|---|---|---|---|---|---|
| DocumentEntry | author | R2 | R2 | R2 | R2 | R2 |
| DocumentEntry | availabilityStatus | O | O | O | O | O |
| DocumentEntry | classCode | R | R | R2 | R | R2 |
| DocumentEntry | comments | O | O | O | O | O |
| DocumentEntry | confidentialityCode | R | R | R2 | R | R2 |
| DocumentEntry | creationTime | R | R | R2 | R | R2 |
| DocumentEntry | entryUUID | R | R | R | R | R |
| DocumentEntry | eventCodeList | O | O | O | O | O |
| DocumentEntry | formatCode | R | R | R2 | R | R2 |
| DocumentEntry | hash | O | R | R | O | O |
| DocumentEntry | healthcareFacility TypeCode | R | R | R2 | R | R2 |
| DocumentEntry | homeCommunityId | O | O | O | O | O |
| DocumentEntry | languageCode | R | R | R2 | R | R2 |
| DocumentEntry | legalAuthenticator | O | O | O | O | O |
| DocumentEntry | mimeType | R | R | R | R | R |
| DocumentEntry | patientId | R | R | R2 | R | R2 |
| DocumentEntry | practiceSettingCode | R | R | R2 | R | R2 |
| DocumentEntry | repositoryUniqueId | O | R | O | O | O |
| DocumentEntry | serviceStartTime | R2 | R2 | R2 | R2 | R2 |
| DocumentEntry | serviceStopTime | R2 | R2 | R2 | R2 | R2 |
| DocumentEntry | size | O | R | R | O | O |
| DocumentEntry | sourcePatientId | R | R | R2 | R | R2 |
| DocumentEntry | sourcePatientInfo | O | O | R2 | O | R2 |
| DocumentEntry | title | O | O | O | O | O |
| DocumentEntry | typeCode | R | R | R2 | R | R2 |
| DocumentEntry | uniqueId | R | R | R | R | R |
| DocumentEntry | URI | O | O | R | O | O |
| SubmissionSet | author | R2 | R2 | R2 | R2 | R2 |
| SubmissionSet | availabilityStatus | O | O | O | O | O |
| SubmissionSet | comments | O | O | O | O | O |
| SubmissionSet | contentTypeCode | R | R | R2 | R | R2 |
| SubmissionSet | entryUUID | R | R | R | R | R |
| SubmissionSet | homeCommunityId | O | O | O | O | O |
| SubmissionSet | intendedRecipient | O | O | R2 | R2 | R2 |
| SubmissionSet | patientId | R | R | R2 | R | R2 |
| SubmissionSet | sourceId | R | R | R | R | R |
| SubmissionSet | submissionTime | R | R | R | R | R |

_____

_____

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

### 4.3.1.2  XDS Specific Requirements

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

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

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

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

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

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

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

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

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

- Metadata

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

_____

_____

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

2440

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

2445    Id attributes and references: The submission may contain valid UUIDs or symbolic ids (or both) as specified in section 4.3.1.2.2.

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

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

2450
- SubmissionSet metadata

- DocumentEntry metadata

- Folders to be created

- Associations (various relationships and HasMember as needed)

**Document duplication**: There are several conditions regarding the duplication of document that
2455    can occur.

- Duplicate registration of a document - A document and its metadata are submitted to the XDS Document Repository as part of a Provide and Register Document Set [ITI-41] transaction. This document already exists in one or more Document Repositories and is already represented in the Document Registry. It is submitted with a new (not previously
2460    used) entryUUID for the DocumentEntry and associated ancillary objects. The Document Registry shall accept such duplicate registration of the documents.

- Duplicate document id submitted to repository - A document with its associated metadata is submitted to an XDS Document Repository. A document with the same DocumentEntry.uniqueId is present in the repository but the DocumentEntry.hash is
2465    different. This is an error and the Submission Request shall be rejected by the Document Repository.

    Note: There are two approaches to detecting this fault. First, this can be detected at the repository if repository logic can validate the hashes and has record of the document id to compare. Otherwise the request can be forwarded on to the registry and let the fault be detected by the registry (see next bullet). The repository then deals with the error
2470    returned by the registry.

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

_____

2475    Note: A document can be registered by multiple Document Sources with the same uniqueId, same hash, different entryUUID, and with other metadata attributes not the same as described above. As a result, a Document Consumer may issue a GetDocuments Stored Query with a uniqueId parameter and have returned two or more DocumentEntry objects with that same uniqueId.

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

2480

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

2485    Id references to objects in the Registry shall be in UUID format. Id references to other objects within a submission may be UUID or symbolic format. All Id references returned in a Stored Query shall be in UUID format. A responsibility of a Document Registry when accepting a Register Document Set transaction is to translate any symbolic Ids into UUIDs before storing them in the registry.

2490    Symbolic Id format is any string that does not have the prefix 'urn:uuid:'.

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

### 4.3.1.2.3 XDS Atomicity of Submissions

XDS Submission requests shall be atomic operations. If any part of the Submission Request
2495    Processing fails, the entire transaction shall fail and no changes result. The result of a successful Submission Request is to update either:

a Document Registry or

a Document Registry and a Document Repository.

The XDS Document Registry addresses the fact that the ebXML Registry specification does not
2500    guarantee that SubmitObjectsRequest are atomic. XDS specifies the mechanism through which atomicity is to be implemented and where it is needed.

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

If an error occurs, storing metadata in the Document Registry, then the following actions shall be
2505    performed:

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

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

2510    Registry queries from the Registry Stored Query [ITI-18] transaction shall not find XDS SubmissionSets, XDS Folders or DocumentEntry objects until after the above atomic operation

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

2515 For document metadata included by value, a rollback of the submission shall delete the document metadata and the Association. For document metadata included by reference, a rollback of the submission shall not delete the document metadata but shall still delete the Association. The document whose Association is being deleted existed before this submission and shall be maintained.

### 4.3.1.2.4 XDS Registry Enforcement of Attributes

2520 The XDS Document Registry enforces rules on metadata content. Other document sharing profiles have their own different enforcement specifications. A summary of attribute enforcement is provided in tables 4.3.1.2-1, 4.3.1.2-2, 4.3.1.2-3; details on certain attributes are given in section 4.3.1.2.5 XDS Registry Responsibilities.

The Registry shall reject any submission which includes attribute values whose size exceeds the
2525 specification in the standard.

ebRIM version 3.0 data type Slot/ValueList/Value is limited to 256 characters before XML escaping by that standard. Document Sharing metadata may have restricted this to 128 characters or less. Some HL7 data types may be longer according to HL7; however, when encoded as slots, the Document Source shall restrict these to fit into the 256 character space allocated. This may
2530 require some information to be excluded. The XDS profile gives no guidance as to how information is to be excluded to make this coding limit.

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

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

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

_____

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

2535

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

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

_____

_____

| SubmissionSet Attribute | Registry Enforcement |
|---|---|
| sourceId | Enforcement of format only |
| submissionTime | Enforcement of format only |
| uniqueId | No identical existing uniqueId in registry (assigned to DocumentEntry, SubmissionSet, or Folder) |

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

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

### 4.3.1.2.5 XDS Registry Responsibilities

2540 The XDS Document Registry extends functionality beyond the ebXML registry standard. The
Document Registry performs validation as specified below:

**Validate patient Id** –patientIds (such as the DocumentEntry.patientId attribute) shall be a
known patient Id and registered against the Patient Id Domain of the XDS Affinity Domain
managed by the Patient Identity Source Actor for the XDS Affinity Domain. This validation
2545 requirement does not apply to DocumentEntry.sourcePatentId.

- Patient identifiers shall be verified against the patient identifiers received from the Patient
  Identity Feed [ITI-8] indicating that it is registered against the Patient ID Domain of the
  Affinity Domain.

- Patient identifiers shall be verified against known subsumed patient identifiers received
2550  from the Patient Identity Feed [ITI-8]. A submission containing a subsumed patient
  identifier shall return an XDSUnknownPatientId error. See section 3.8.4.2.4 for
  background on patient identifier merge.

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

2555 **Verify coded values** – the adaptor shall verify that coded fields (ebXML external classifications)
contain valid XDS specified values or where the XDS Affinity Domain constrains code values,
to verify them. Optionally, verify that the code display names match the code value and schema.

_____

_____

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

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

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

- DocumentEntry to be replaced shall have status = Approved.

- The Association's sourceObject attribute shall contain the Id (UUID or symbolic Id) of an ExtrinsicObject representing a DocumentEntry included in the SubmissionSet.

- The Association's targetObject attribute shall contain the UUID of an ExtrinsicObject (DocumentEntry) already in the registry.

When the 'RPLC' or 'XFRM_RPLC' Association is detected by the Registry Adaptor it shall:

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

- Submit the Submission Request to the registry.

- If the submission is successful, label the replacement document as Approved and the replaced document as Deprecated.

- The existing document and the document replacing it shall have the same patient identifier (XDSDocumentEntry.patientId attribute). This comparison shall take into consideration patient identity merges as described in ITI TF-2a: 3.8.4.2.4.

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

**Validate patientId in SubmissionSets:**

- Must match included Folders

- Must match include DocumentEntries (by value)

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

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

_____

_____

**Validate patientId on documents being added to a Folder** - The value of the patientId
2595  attribute of a DocumentEntry object shall match the value of the patientId attribute on any folder
that holds it.

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

### 4.3.1.2.6 Required Initialization of the XDS Affinity Domain

2600  This initialization supports the operation of the Registry Adaptor. The following information
must be provided by the XDS Affinity Domain administrator and loaded into the Registry
Adaptor. This supports the functionality specified for the Registry Adaptor in section 4.3.1.2.
How this information is loaded into the Registry Adaptor or how the Registry Adaptor is
implemented is not defined by the XDS profile.

2605      1.  List of acceptable mimeTypes for documents indexed by the registry.

         2.  Domain name (Patient Id Assigning Authority) for XDS Affinity Domain. PatientIds
             attached to metadata submitted to this registry shall come from this Assigning Authority.

         3.  Acceptable values for all coded attributes characterized in the registry by ebXML
             External Classifications.

2610  ### 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.

2615

#### Table 4.3.2.1-1: Actor/Transaction Pairs

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

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:

2620

_____

_____

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

| Code | Meaning |
|------|---------|
| R | Required – a value for the attribute shall be supplied by the Actor when responding to a query |
| R2 | Required if Known – a value for the attribute shall be supplied by the 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 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. |

2625

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

| Metadata Element | Metadata Attribute | XDS DR | XDS 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 | R | R | R | R |
| DocumentEntry | entryUUID | R | R | R | R |
| DocumentEntry | eventCodeList | O | O | O | O |
| DocumentEntry | formatCode | R | R | R | R |
| DocumentEntry | hash | R | R | R | R |
| DocumentEntry | healthcareFacility TypeCode | R | R | R | R |
| DocumentEntry | homeCommunityId | O | R | R | O |
| DocumentEntry | languageCode | R | R | R | R |
| DocumentEntry | legalAuthenticator | O | O | O | O |
| DocumentEntry | mimeType | R | R | R | R |
| DocumentEntry | patientId | R | R | R | R |
| DocumentEntry | practiceSettingCode | R | R | R | R |
| DocumentEntry | repositoryUniqueId | R | R | R | R |
| DocumentEntry | serviceStartTime | R2 | R2 | R2 | R2 |
| DocumentEntry | serviceStopTime | R2 | R2 | R2 | R2 |
| DocumentEntry | Size | R | R | R | R |
| 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 |

_____

_____

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

_____

_____

# 5   IHE Content Specifications

2630    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

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

2635    The BPPC document has two possible document templates, one without a scanned document part, and one in ITI TF-3: 5.1.2 defines the requirements of the BPPC document without a scanned document part; ITI TF-3: 5.1.3 explains with a scanned document part.

### 5.1.1 References

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

2640    ### 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

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

2645    Document (XDS-SD), it will conform to ITI TF-3: 5.1.3.

#### 5.1.2.1 XDS Metadata

##### 5.1.2.1.1 XDS DocumentEntry Metadata

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

2650    ##### 5.1.2.1.1.1 XDSDocumentEntry.classCode

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

For backward compatibility historic documents may be registered with:

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

2655    - classCodeDisplayName -- This attributes may have been set to the value "Consent".

##### 5.1.2.1.1.2 XDSDocumentEntry.eventCodeList

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

_____

- /ClinicalDocument/documentationOf/serviceEvent[templateId/@root='1.3.6.1.4.1.193
2660      76.1.5.3.1.2.6']/code/@code

- eventCodeDisplayNameList -- The eventCodeDisplayNameList shall be populated using
  the display names for those policies.

  - /ClinicalDocument/documentationOf/serviceEvent[templateId/@root='1.3.6.1.4.1.193
    76.1.5.3.1.2.6']/code/@displayName

### 2665  5.1.2.1.1.3 XDSDocumentEntry.formatCode

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

### 5.1.2.1.1.4 XDSDocumentEntry.uniqueId

This value shall be the `ClinicalDocument/id` in the HL7 CDA R2 header. The root attribute is
2670    required, and the extension attribute is optional. The total length is limited to 256 characters.
Additionally see IHE PCC TF-2: 4.1.1, for further content specification.

### 5.1.2.1.2 XDS SubmissionSet Metadata

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

### 5.1.2.1.3 XDS Folder Metadata

2675    No additional requirements. For more information, see IHE PCC TF-2: 4.1.3

### 5.1.2.2 Specification

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
2680    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.

_____

_____

```
2685   <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=' '/>
         <code code='57016-8' displayName='PATIENT PRIVACY ACKNOWLEDGEMENT'
2690        codeSystem='2.16.840.1.113883.6.1' codeSystemName='LOINC'/>
         <title>Consent to Share Information</title>
         <effectiveTime value='20070619012005'/>
         <confidentialityCode code='N' displayName='Normal'
           codeSystem='2.16.840.1.113883.5.25' codeSystemName='Confidentiality' />
2695     <languageCode code='en-US'/>
             :
         <component><structuredBody>

         </structuredBody></component>
2700   </ClinicalDocument>
```

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

A Patient Privacy Acknowledgement Document shall contain a text description of what the
patient consented to, a list of codes indicating the policy(s) agreed to, and a time range indicating
the effective time of the consent. It may be attested to using an electronic digital signature,
2705   conforming to the ITI Digital Signature Profile.

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

2710

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

### 5.1.2.2.1 Patient Privacy Acknowledgement Service Events 1.3.6.1.4.1.19376.1.5.3.1.2.6

Within a Patient Privacy Acknowledgement Document, there shall be a Patient Privacy
Acknowledgement Service Event with the effective time of the consent shall be specified within
2715   the documentationOf/serviceEvent element.

_____

_____

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

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

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

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

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

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

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

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

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

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=' '/>

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>

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.

2755 ### 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

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.

2760 #### 5.1.3.1 XDS Metadata

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

##### 5.1.3.1.1 XDS DocumentEntry Metadata

The BPPC document shall conform to the XDS DocumentEntry Metadata requirements in the
2765 IHE PCC TF-2:5.1.1.1.1 unless otherwise specified below.

##### 5.1.3.1.1.1 XDSDocumentEntry.formatCode

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

##### 5.1.3.1.2 XDS SubmissionSet Metadata

2770 No additional constraints. For more information, see IHE PCC TF-2: 5.1.1.1.2

##### 5.1.3.1.3 XDS Folder Metadata

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

#### 5.1.3.3 Specification

This BPPC document shall conform to the XDS-SD (ITI TF-3: 5.2) specification and shall have
2775 the additional requirements stated in ITI TF-3: 5.1.2.3.

#### 5.1.3.4 Conformance

See IHE ITI TF-3: 5.1.2.4.

## 5.2 Scanned Documents Content Module

This section outlines the content of the HL7 CDA R2 constraints for the document. We note here
2780 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 data, as specified, and in many cases do make use of additional CDA header elements for
2785     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.

2790
- *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.

2795 **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)
2800
- RFC 3066, 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-
2805     oriented documents. PDF/A attempts to maximize:

- Device independence
- Self-containment
- Self-documentation

The constraints imposed by PDF/A include:

2810
- 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

_____

_____

2815
- 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

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.
2820 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.

2825 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)
2830 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.

Plaintext as well as PDF/A documents shall be base-64 encoded before wrapped in a HL7 CDA
2835 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 IHE ITI TF-3: 5.2.3).

2840 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

XDS-SD is a CDA R2 document and thus conforms to the XDS Metadata requirements in the
2845 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

2850 The XDSDocumentEntry.formatCode shall be **urn:ihe:iti:xds-sd:pdf:2008** when the document is scanned pdf and **urn:ihe:iti:xds-sd:text:2008** when the document is scanned text. The formatCode codeSystem shall be 1.3.6.1.4.1.19376.1.2.3.

### 5.2.2.1.2 XDSDocumentEntry.uniqueId

This value shall be the `ClinicalDocument/id` in the HL7 CDA R2 header. The root attribute is
2855 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

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
2860 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
2865 document for these cases.

### 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

2870 ### 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/confidentialityCode | R | 5.2.3.1 | O | Assigned by the operator |
| ClinicalDocument/effectiveTime | R | 5.2.3.1 | DS | Computed. This is the scan time. |
| ClinicalDocument/languageCode | R | 5.2.3.1 | O | Entered by operator |
| ClinicalDocument/recordTarget | R | 5.2.3.2 | SA / O | Taken from scanned content, supplemented by operator. |
| ClinicalDocument/author/assignedAuthor/assignedPerson | R2 | 5.2.3.3 | SA / O | Taken from scanned content, supplemented by operator. This is the original author. |
| ClinicalDocument/author/assignedAuthor/authoringDevice | R | 5.2.3.4 | DS / FM / O | Can be computed or fixed based on the scanning device and software. This is the information about the scanning device. |
| ClinicalDocument/dataEnterer | R | 5.2.3.5 | DS / O | Can be computed by the scanner or supplemented by operator. This is the information about the scanner operator. |
| ClinicalDocument/custodian | R | 5.2.3.6 | DS / FM | Retains original HL7 CDA Context. To be computed or fixed appropriately to denote guardianship of the scanned and wrapped content. |
| ClinicalDocument/legalAuthenticator | O | 5.2.3.7 | O | Most likely supplemented by the operator, when applicable or mandated. |
| ClinicalDocument/documentationOf/serviceEvent/effectiveTime | R | 5.2.3.8 | SA / O | Denotes the time/date range of the original content. |
| ClinicalDocument/component/nonXMLBody | R | 5.2.3.9 | SA | The scanned/encoded content. |

_____

### 5.2.3.1 ClinicalDocument child-less elements

In this section we further discuss `id`, `code`, `effectiveTime`, `confidentialityCode` and
2875    `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.

- The ClinicalDocument/id element shall be present. The root attribute shall contain the oid
  2880    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.

- The ClinicalDocument/code will in most cases be provided by the operator. Values for
  2885    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
  2890    time zone offset from GMT.

- 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.
  2895    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
  2900    header, the language context of the CDA will be overwritten at the body level (see IHE
  ITI TF-3: 5.2.3.9 ClinicalDocument/component/nonXMLBody for an example). Attribute
  code@code shall be present. Attribute code@codeSystem shall be IETF (Internet
  Engineering Task Force) RFC 3066 in accordance with the HL7 CDA R2 documentation.

_____

_____

Example:

```
<ClinicalDocument xmlns="urn:hl7-org:v3">
  <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
  <templateId root="1.3.6.1.4.1.19376.1.2.20"/>
  <id root="1.3.6.4.1.4.1.2835.2.7777"/>
  <code code="34133-9"  codeSystem="2.16.840.1.113883.6.1"
      codeSystemName="LOINC" displayName="SUMMARIZATION OF EPISODE NOTE"/>
  <title>Good Health Clinic Care Record Summary</title>
  <effectiveTime value="20050329224411+0500"/>
  <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
  <languageCode code="en-US"/>
```

2905

## 5.2.3.2 ClinicalDocument/recordTarget

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
2910 subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- The ClinicalDocument/recordTarget/patientRole/id element shall include both the root and the extension attributes. Refer back to PCC TF-2: 4.1.1 for more details.

- At least one ClinicalDocument/recordTarget/patientRole/addr element shall include at
2915 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.

2920 - 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>
```

2925

### 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
2930 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.

2935 - 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 the extension attributes. Refer to PCC TF-2: 4.1.1 for more details.

2940 - The ClinicalDocument/author/assignedAuthor/representedOrganization/id element if known shall include both the root and the extension attributes. Refer to PCC TF-2: 4.1.1for more details.

Example:

_____

_____

```
<author>
  <templateId root="1.3.6.1.4.1.19376.1.2.20.1"/>
  <time value="19990522"/>
  <assignedAuthor>
    <id extension="11111111" root="1.3.5.35.1.4436.7"/>
    <assignedPerson>
      <name>
        <prefix>Dr.</prefix>
        <given>Bernard</given>
        <family>Wiseman</family>
        <suffix>Sr.</suffix>
      </name>
    </assignedPerson>
    <representedOrganization>
      <id extension="aaaaabbbbb" root="1.3.5.35.1.4436.7"/>
      <name>Dr. Wiseman's Clinic</name>
    </representedOrganization>
  </assignedAuthor>
</author>
```

2945 **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.

- The ClinicalDocument/author/templateId element shall be present. The root attribute
2950    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
2955    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
2960    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.

- The
2965    ClinicalDocument/author/assignedAuthor/assignedAuthoringDevice/manufacturerModel Name element shall be present. The mixed content shall contain string information that

_____

_____

specifies the scanner product name and model number. From this information, features
like bit depth and resolution can be inferred. In the case of virtually scanned documents
(for example, print to PDF), the manufactureModelName referenced here refers to the
2970      makers of the technology that was used to produce the embedded content.

- 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
2975      embedded content.

- The ClinicalDocument/author/assignedAuthor/representedOrganization/id element shall
  be present. The root attribute shall be set to the oid of the scanning facility.

Example:

```
<author>
  <templateId root="1.3.6.1.4.1.19376.1.2.20.2"/>
  <time value="20050329224411+0500"/>
  <assignedAuthor>
    <id root="1.3.6.4.1.4.1.2835.2.1234"/>
    <assignedAuthoringDevice>
    <code code="CAPTURE" displayName="Image Capture" codeSystem="
    1.2.840.10008.2.16.4" />
      <manufacturerModelName>SOME SCANNER NAME AND MODEL
      </manufacturerModelName>
      <softwareName>SCAN SOFTWARE NAME v0.0</softwareName>
    </assignedAuthoringDevice>
    <representedOrganization>
      <id root="1.3.6.4.1.4.1.2835.2"/>
      <name>SOME Scanning Facility</name>
      <addr>
        <streetAddressLine>21 North Ave</streetAddressLine>
        <city>Burlington</city>
        <state>MA</state>
        <postalCode>01803</postalCode>
        <country>USA</country>
      </addr>
    </representedOrganization>
  </assignedAuthor>
</author>
```

## 2980  5.2.3.5 ClinicalDocument/dataEnterer

This ClinicalDocument/dataEnterer element shall represent the scanner operator who produced
the scanned content. All subelements retain their original definition as defined by the HL7 CDA
R2 specification, unless noted below.

_____

- 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.

2985

- 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.

2990

- 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>
```

## 2995 5.2.3.6 ClinicalDocument/custodian

The ClinicalDocument/custodian shall be present. Its context is left up to the scanning facility to refine in accordance with local policies and to reflect the entity responsible for the scanned content. In most cases this will be the scanning facility. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

3000

- The ClinicalDocument/assignedCustodian/representedOrganization/name shall be present.

- 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>
```

3005

### 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.

3010
- The ClinicalDocument/legalAuthenticator/assignedEntity/id element if known shall
  include both the root and the extension attributes. Refer back to PCC TF-2: 4.1.1 for
  more details.

Example:

```
<legalAuthenticator>
  <time value="19990522"/>
  <signatureCode code="S"/>
  <assignedEntity>
    <id extension="11111111" root="1.3.5.35.1.4436.7"/>
    <assignedPerson>
      <name>
        <prefix>Dr.</prefix>
        <given>Bernard</given>
        <family>Wiseman</family>
        <suffix>Sr.</suffix>
      </name>
    </assignedPerson>
  </assignedEntity>
</legalAuthenticator>
```

_____

_____

3015 ## 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
3020 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>
```

## 5.2.3.9 ClinicalDocument/component/nonXMLBody

3025 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.

- If the human-readable language of the scanned content is different than that of the
3030   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 3066 in accordance with the HL7 CDA R2 documentation.

- The ClinicalDocument/component/nonXMLBody/text element shall be present and
3035   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
3040     encoding as specified by RFC-2046.

  - 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.

_____

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/9bfUNOPfOte44O9jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
       K7CXIe2be+/1DzXQP+RlbmRzdHJlYW0KZW5kb2JqCjYgMCBvYmoKMjAxCmVuZG9iago0
       ...
       SW5mbyAyIDAgUgovSUQgWzxGNENDN0FFQjjU0QjM2RkIyODNDNUMzMjQ3OUFFMjgzRj48
       RjRDQzdBRUI1NEIzNkZCMjgzQzVDMzI0NzlBRDI4M0Y+XQo+PgpzdGFydHhyZWYKMzAx
       MgolJUVPRgo=
       </text>
     </nonXMLBody>
   </component>
 </ClinicalDocument>
```

3045

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/9bfUNOPfOte44O9jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
       K7CXIe2be+/1DzXQP+RlbmRzdHJlYW0KZW5kb2JqCjYgMCBvYmoKMjAxCmVuZG9iago0
       ...
       SW5mbyAyIDAgUgovSUQgWzxGNENDN0FFQjjU0QjM2RkIyODNDNUMzMjQ3OUFFMjgzRj48
       RjRDQzdBRUI1NEIzNkZCMjgzQzVDMzI0NzlBRDI4M0Y+XQo+PgpzdGFydHhyZWYKMzAx
       MgolJUVPRgo=
       </text>
     </nonXMLBody>
   </component>
 </ClinicalDocument>
```

## 5.2.4 Complete Example (Wrapped PDF)

```
3050   <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">
         <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
         <templateId root="1.3.6.1.4.1.19376.1.2.20"/>
3055     <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"/>
3060     <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"/>
         <languageCode code="en-US"/>
         <recordTarget>
           <patientRole>
             <id extension="12345" root="2.16.840.1.113883.3.933"/>
3065         <addr>
               <streetAddressLine>17 Daws Rd.</streetAddressLine>
               <city>Blue Bell</city>
               <state>MA</state>
               <postalCode>02368</postalCode>
3070           <country>USA</country>
             </addr>
             <patient>
               <name>
                 <prefix>Mrs.</prefix>
3075             <given>Ellen</given>
                 <family>Ross</family>
               </name>
               <administrativeGenderCode code="F"
                  codeSystem="2.16.840.1.113883.5.1"/>
3080           <birthTime value="19600127"/>
             </patient>
           </patientRole>
         </recordTarget>
         <author>
3085       <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>
3090           <name>
                 <prefix>Dr.</prefix>
                 <given>Bernard</given>
                 <family>Wiseman</family>
                 <suffix>Sr.</suffix>
3095           </name>
             </assignedPerson>
             <representedOrganization>
                 <id extension="aaaaabbbbb" root="1.3.5.35.1.4436.7"/>
```

_____

```
                <name>Dr. Wiseman's Clinic</name>
3100        </representedOrganization>
          </assignedAuthor>
      </author>
      <author>
        <templateId root="1.3.6.1.4.1.19376.1.2.20.2"/>
3105    <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="
3110    1.2.840.10008.2.16.4" />
              <manufacturerModelName>SOME SCANNER NAME AND MODEL
              </manufacturerModelName>
              <softwareName>SCAN SOFTWARE NAME v0.0</softwareName>
          </assignedAuthoringDevice>
3115      <representedOrganization>
              <id root="1.3.6.4.1.4.1.2835.2"/>
              <name>SOME Scanning Facility</name>
              <addr>
                <streetAddressLine>21 North Ave</streetAddressLine>
3120            <city>Burlington</city>
                <state>MA</state>
                <postalCode>01803</postalCode>
                <country>USA</country>
              </addr>
3125      </representedOrganization>
        </assignedAuthor>
      </author>
      <dataEnterer>
        <templateId root="1.3.6.1.4.1.19376.1.2.20.3"/>
3130    <time value="20050329224411+0500"/>
        <assignedEntity>
          <id extension="22222222" root="1.3.6.4.1.4.1.2835.2"/>
          <assignedPerson>
            <name>
3135          <prefix>Mrs.</prefix>
              <given>Bernice</given>
              <family>Smith</family>
            </name>
          </assignedPerson>
3140    </assignedEntity>
      </dataEnterer>
      <custodian>
        <assignedCustodian>
          <representedCustodianOrganization>
3145        <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>
3150            <state>MA</state>
                <postalCode>01803</postalCode>
```

_____

```
                       <country>USA</country>
                    </addr>
                 </representedCustodianOrganization>
3155       </assignedCustodian>
        </custodian>
        <legalAuthenticator>
           <time value="19990522"/>
           <signatureCode code="S"/>
3160       <assignedEntity>
              <id extension="11111111" root="1.3.5.35.1.4436.7"/>
              <assignedPerson>
                 <name>
                    <prefix>Dr.</prefix>
3165               <given>Bernard</given>
                    <family>Wiseman</family>
                    <suffix>Sr.</suffix>
                 </name>
              </assignedPerson>
3170       </assignedEntity>
        </legalAuthenticator>
        <documentationOf>
           <serviceEvent >
              <effectiveTime>
3175             <low value="19800127"/>
                 <high value="19990522"/>
              </effectiveTime>
           </serviceEvent>
        </documentationOf>
3180    <component>
           <nonXMLBody>
              <text mediaType="application/pdf" representation="B64">
              JVBERi0xLjMKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0
              ZURlY29kZT4+CnN0cmVhbQp4nGWPMWsDMQyFd/8KjfJwqmVbkr0GQqFbg7fQoSRNWuhB
3185          Q/4/1L67TEEYme+9J1s3CMQQRm39NLuXg8H17gK89nN1N8eLAbZ2mmHXuql2QDVUhnZx
              a5iBcyQtoMIUM7TZHbH5KZEVDgm//SSUswbFHx/JzBLeu5yYxOIzE8bPcRWqdaGDmcZO
              BWc/9bfUNOPfOte44O9jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
              K7CXIe2be+/1DzXQP+RlbmRzdHJlYW0KZW5kb2JqCjYgMCBvYmoKMjAxCmVuZG9iago0
              ...
3190          SW5mbyAyIDAgUgovSUQgWzxGNENDN0FFQjjU0QjM2RkIyODNDNUMzMjQ3OUFFMjgzRj48
              RjRDQzdBRUI1NEIzNkZCMjgzQzVDMzI0NzllBRDI4M0Y+XQo+PgpzdGFydHhyZWYKMzAx
              MgolJUVVPRgo=
              </text>
           </nonXMLBody>
3195    </component>
     </ClinicalDocument>
```