Integrating the Healthcare Enterprise

IT Infrastructure Technical Framework

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

Revision 9.0 – Final Text

August 31, 2012

CONTENTS

4 Cross-Transaction Specifications

4.1 XDS Metadata

The following sections specify the mapping of XDS concepts to ebRS and ebRIM semantics:

- XDS Document
- XDS Submission Request
- XDS Submission Set
- XDS Folder
- Document Relationships

The next sections specify the metadata definitions to support the above concepts. The following are discussed:

- XDS Document
- XDS Submission Request
- XDS Submission Set
- XDS Folder

The remaining two sections discuss the following topics:

- XDS Registry Adaptor function
- General Metadata issues

Transactions that Reference this Section 4.1	
Registry Stored Query	ITI-18
Register Document Set – b	ITI-42
Provide and Register Document Set – b	ITI-41
Retrieve Document Set	ITI-43

Transactions that reference specific subsections of this Section 4.1	
Distribute Document Set on Media (4.1.1, 4.1.7, 4.1.8, 4.1.12)	ITI-32

4.1.1 Class Diagram

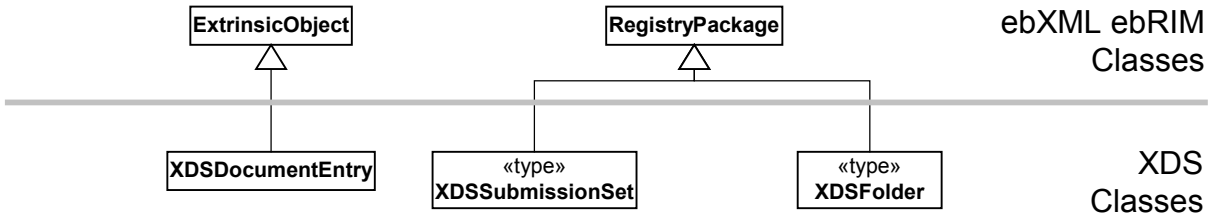

Figure 4.1-1: ebXML Class Diagram of the Register Document Metadata

The XDSDocumentEntry class is derived from the ebXML ExtrinsicObject class. The XDSSubmissionSet and XDSFolder classes are derived from the ebXML[1] RegistryPackage class. Since the ebXML Registry standard does not allow for subclassing the RegistryPackage class, these two classes are implemented as ebXML RegistryPackages. Type information (submission set vs. folder) is coded as an ebXML Classification against two object types created by the XDS profile, XDSSubmissionSet and XDSFolder.

4.1.2 Document Specification

A new registry object type is declared as a subclass of ebXML ExtrinsicObject. Its name is XDSDocumentEntry. An object of this type in the XDS registry is used to represent a document in an XDS repository.

An XDSDocumentEntry object in the registry contains a reference to a single document in a single 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.

4.1.3 XDS Submission Request Specification

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

There are two types of submission requests: XDS Registry Submission Request and XDS Repository Submission Request. Both are described below.

Appropriate protocol bindings are used to transfer this content between systems when the actors are not implemented together on the same system. The bindings are described in "Protocol Selection" section of the appropriate transaction.

[1] ebXML Registry terms such as RegistryPackage are shown with an ebXML prefix to help distinguish ebXML Registry terms from XDS terms. Unless otherwise indicated, references to 'ebXML' in XDS refer to the ebXML Registry specifications as opposed to other ebXML specifications. The short term is used for readability.

The two types of XDS Submission Requests are described next.

4.1.3.1 XDS Registry Submission Request

An XDS Registry Submission Request is the collection of metadata transferred between a Document Repository and a Document Registry in a single ebXML SubmitObjectsRequest. This request contains:

A collection of metadata to be stored in the registry including:

- Metadata for new documents
- Folders to be created
- Documents to be added to folders

A single XDS Submission Set object, contained within the metadata, organizing the metadata

This request is part of the Register Document Set-b [ITI-42] transaction.

4.1.3.2 XDS Repository Submission Request

An XDS Repository Submission Request is the collection of metadata and documents transferred between a Document Source and a Document Repository using a single ebXML SubmitObjectsRequest. This request contains:

- Metadata
- Zero or more documents; each document is represented by an XDSDocumentEntry object in the metadata. Submissions that add metadata to the registry without adding documents to the repository are possible.

This request is the information payload of the Provide and Register Document Set message of the Provide and Register Document Set-b [ITI-41] transaction.

4.1.3.3 Atomicity Requirements for XDS Submission Requests

XDS Submission requests shall be atomic operations. The result of a Submission Request is to update either:

- a Registry or
- a Registry and a Repository.

All changes requested are successfully applied or no net changes are made. More specifically:

1. Atomicity shall be managed by an XDS registry adaptor (see ITI TF-3: 4.1.11 for details on registry adaptor addressing the fact that the ebXML Registry specification does not guarantee that a SubmitObjectsRequest is atomic). XDS specifies the mechanism through which atomicity is to be implemented and where it is needed.

2. All objects shall have their Status attribute set to Submitted when the objects are first created in the ebXML registry. An ebXML ApproveObjectsRequest, shall be issued within the XDS Registry Adaptor to change the Status attribute to Approved. This completes the transaction.

3. The following types of objects shall have their status set to Approved to be considered publicly available:

- XDSSubmissionSet (ebXML RegistryPackage)
- XDSFolder (ebXML RegistryPackage)
- XDSDocumentEntry (subclass of ebXML ExtrinsicObject)

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 registry, then the following actions are performed:

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

Registry queries from the Registry Query transaction shall not find XDS Submission Sets, XDS Folders or XDSDocumentEntry objects until after the above atomic operation that creates them has completed successfully and the status attributes have been set to Approved.

4.1.3.4 Other Properties of Submission Requests

A Submission Request may contain metadata beyond the XDS Submission Set, XDS Folder, and XDSDocumentEntry objects. These are:

- ebXML Associations linking XDSDocumentEntry objects to XDSFolder objects. There are no restrictions on whether the XDSDocumentEntry objects or XDSFolder objects are in this Submission Request. Such an Association is the ebXML mechanism for including objects in an ebXML RegistryPackage (the basis of XDSFolder). Each of these Associations shall be accompanied by another association that links it with the XDSSubmissionSet object. This additional association allows for the identification of the Document Source actor which linked a particular document with a particular folder. See ITI TF-3: 4.1.5 for more information.
- Associations linking existing (already contained in the registry) XDSDocumentEntry objects to the XDSSubmissionSet RegistryPackage contained in this Submission Request. This option is discussed in the next section.

4.1.3.5 Attribute Value Length

All attribute values shall conform to the size specification of ebRIM version 3.0 that is detailed in section 2.2 of that specification - Data Types. More specifically, all Slots shall conform to the specification of ebRIM version 3.0, which is detailed in section 2.8.1 of that specification.

4.1.4 Submission Set Specifications

Submission Sets exist for two reasons:

1. To support atomic submission to the registry
2. To make a permanent record in the registry of
   a) The existence and status of the submission
   b) The XDS Folders and XDSDocumentEntry objects included in the submission.

Submission Sets, once shared, are immutable.

An XDS SubmissionSet is an ebXML RegistryPackage, classified as XDSSubmissionSet that is used to bundle XDSDocumentEntry, XDSFolder and Association objects for submission.

A Submission Set has a set of attributes that are described in ITI TF-3: 4.1.8 Submission Set Metadata.

4.1.4.1 Inclusion of Documents in a SubmissionSet

Documents may be included in a Submission Set in two ways: inclusion by value and inclusion by reference.

Inclusion by value: A new document is being submitted to the registry. The Submission Set contains the XDSDocumentEntry object with associated attributes.

Inclusion by reference: Existing documents in the registry can be referenced by a Submission Set. These documents are included because of their clinical relevance to the rest of the Submission Set.

Linking document metadata to submission set: An XDSSubmissionSet shall be represented by an ebXML RegistryPackage. Document metadata (XDSDocumentEntry objects) shall be linked to the RegistryPackage via ebXML Associations according to the ebXML Registry standard.

For documents included by reference, the Submission Request shall include the Association object used to link the document. For documents included by value, the Submission Request shall include the XDSDocumentEntry object and the Association object used to link the document.

Submission Set Association labeling: Two types of association labels are defined: original (submission by value), or reference (Submission by reference). This allows finding the submission set that first submitted any document. It also supports proper rollback in case of a submission error. 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.) The following labeling of the Associations is required.

Table 4.1-1: Submission Set Association Labeling

Inclusion type	Rollback	Association Labeling
By Value	Yes	Slot: Name=SubmissionSetStatus Value=Original
By Reference	No	Slot: Name=SubmissionSetStatus

Inclusion type	Rollback	Association Labeling
		Value=Reference

Submission Sets and patients: A Submission Set is restricted in terms of mixing documents from different patients. All documents included by value in a Submission Set shall have their patientId attribute set to the same value. This restriction does not apply to documents included by reference.

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

Duplicate registration of a document - A document and its metadata are submitted to the repository as part of a Repository Submission Request. This document already exists in one or more repositories and is already represented in the registry. It is submitted with a new (not previously used) UUID for the XDSDocumentEntry and associated ancillary objects. The registry shall accept such duplicate registration of the documents.

Duplicate document id submitted to repository - A document with its associated metadata is part of a Repository Submission Request. A document with the same XDSDocumentEntry.uniqueID is present in the repository but the XDSDocumentEntry.hash is different. This is an error and the Submission Request shall be rejected by the 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 returned by the registry.

Duplicate document ID submitted to registry - Metadata representing a document (XDSDocumentEntry) is part of a Registry Submission Request. An XDSDocumentEntry 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 registry adaptor.

A document, once generated outside of the XDS environment, can be registered by multiple Document Sources with the same uniqueId, same hash, different UUID, 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 XDSDocumentEntry objects with that same uniqueId.

4.1.4.2 Inclusion of Folders in a SubmissionSet

Linking folder metadata to submission set: An XDSSubmissionSet shall be represented by an ebXML RegistryPackage. Folder metadata (XDSFolder objects) shall be linked to the RegistryPackage via ebXML Associations according to the ebXML Registry standard.

Linking associations to a submission set: A document can be linked to a folder to indicate that this document is a member of a particular folder. This link shall be represented via an ebXML Association according to the ebXML Registry standard. Each of these Associations shall be accompanied by another ‘HasMember’ Association that links it with the XDSSubmissionSet

object. This additional association allows for the identification of the Document Source actor which linked a particular document with a particular folder and shall be as follows:

- The targetObject shall contain the id of the Association that links the document and the folder.
- The sourceObject shall contain the id of the XDSSubmissionSet object.

It is not necessary that the XDSSubmissionSet object which links to this Association also contain the XDSDocumentEntry metadata or the XDSFolder metadata that correspond to the referenced document and folder. This allows for documents to be placed in folders at a later date and time. If the XDSSubmissionSet object does contain the corresponding XDSDocumentEntry or XDSFolder, then these should be linked to the XDSSubmissionSet object as previously described.

4.1.5 Folder Specification

An XDS Folder is an ebXML RegistryPackage classified as XDSFolder. This folder is used to bundle XDSDocumentEntry objects. Folders shall not be nested inside other folders. The patientId attribute of the XDSDocumentEntry objects it contains shall match the patientId attribute on the folder itself. This shall be enforced by the Registry Actor.

Note: The nesting of folders may be considered as a future extension to this transaction.

Linking documents to a folder: A document can be linked to a folder to indicate that this document is a member of a particular folder. This link shall be represented via an ebXML Association according to the ebXML Registry standard. This association shall have an id attribute which shall be a UUID. Each of these Associations shall be accompanied by another association that links it with the XDSSubmissionSet object. This additional association allows for the identification of the Document Source actor which linked a particular document with a particular folder. See ITI TF-3: 4.1.4.2 for more information regarding this accompanying Association object.

4.1.6 Document Relationships and Associations

4.1.6.1 Document Relationships from HL7

Relationships between documents can be established with XDS. XDS adopts the document relationship semantics defined in HL7 CDA. The supported relationships are listed below in Table 4.1-2. The semantics behind each of these relationships are documented in HL7 CDA Release 2, Committee Ballot 2.

To create a document relationship in the registry, submit:

- A new document (XDSDocumentEntry)
- An Association linking the new document to an existing document.

The association type defines the document relationship. The new document and the association must be submitted in the same Submission Set. The existing document must be an Approved

object already in the registry. The identity (registry UUID) of the existing document must be known because the Document Source pre-assigned the UUID prior to submission or discovered it via registry query.

The association types used for document relationships are defined by XDS and an XDS Registry must be initialized with their definitions. See ITI TF-2x: Appendix H for details.

Note to implementers: A Document Source pre-assigning UUIDs and using the saved UUIDs in future transactions can run into consistency problems if a second Document Source submits to the Registry causing the document to be deprecated. Once a document is deprecated, new Associations to that document cannot be accepted by the Registry.

Table 4.1-2: 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

A Document Relationship refers to any of the relationships listed in Table 4.1-2 Document Relationships above.

A Document Source actor creates a document relationship by submitting a Submission Set containing:

XDSDocumentEntry – this defines the new document being submitted

The uniqueId attribute must be unique.

The UUID must be unique or symbolic (the registry assigns)

Association – this links the original XDSDocumentEntry (already in the registry) with the new XDSDocumentEntry being submitted.

The targetObject attribute of the Association object references the existing document in the registry.

The sourceObject attribute of the Association object references the XDSDocumentEntry contained in the Submission Set.

The Association Type is one of the relationships in Table 4.1-2.

The targetObject attribute of the Association is the registry UUID representing the existing document in the registry. This UUID must be discovered via registry query.

The existing document shall be deprecated by the following rules (based on CDA R2):

The APND and XFRM relationships leave the original document with its status unchanged (Approved).

The RPLC and XFRM_RPLC relationships change the status of the original document to Deprecated. All transformations (XFRM) and addenda (APND) of the original document shall also be deprecated.

Note to implementers: if you are doing a replace where original has addenda, you should be real careful - may have been important comment from another author.

The Registry Adaptor manages document deprecation. See ITI TF-3: 4.1.11 XDS Registry Adaptor for details.

Only the most recent version of a document shall be replaceable. The most recent version of a document carries a status of Approved while older versions carry a status of Deprecated.

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

Associations of type XFRM, APND, RPLC, and XFRM_RPLC may include documentation describing the association (type of transformation, reason for replacement, etc.). If included, it shall be specified as a Classification on the Association as shown in the example below. See also XDS Document Entry attribute parentDocumentRelationshipCode.

Example:

```
<rim:Association id="ThisAssociation"
  associationType="urn:ihe:iti:2007:AssociationType:XFRM"
  sourceObject="source"
  targetObject="target"
   objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Association"
  id="ID_042">
  <rim:Classification
     classificationScheme="urn:uuid:abd807a3-4432-4053-87b4-fd82c643d1f3"
     classifiedObject="ThisAssociation"
      id="ID_043"
      objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"
     nodeRepresentation="French">
    <rim:Name>
      <rim:LocalizedString value="Translation into French"/>
    </rim:Name>
    <rim:Slot name="codingScheme">
      <rim:ValueList>
        <rim:Value>Connect-a-thon translation types</rim:Value>
      </rim:ValueList>
    </rim:Slot>
  </rim:Classification>
</rim:Association>
```


When a document is replaced and that document is a member of one or more folders, a new HasMember Association shall be created by the Registry Adaptor connecting the replacement document to each folder that held the original document as a member. The result is that a folder contains both the original and replacement document differentiated by their status. The Document Registry actor shall detect this condition and generate the necessary Associations.

Table 4.1-3 lists all metadata associated with XDSDocumentEntry objects. The attribute XDSDocumentEntry.parentDocumentId is a reference to the targetObject attribute of the new Association. The attribute XDSDocumentEntry.parentDocumentRelationship is a reference to the Association Type. This represents two distinct naming conventions, HL7 CDA and ebXML Registry.

Document relationship metadata may coexist with other metadata in a Submission Set.

The new documents (related to original document by RPLC, APND, XFRM, or XFRM_RPLC Associations) are assigned their own uniqueId attribute unrelated to the original document's.

See ITI TF-1: 10.4.11.1 for further detail on the use and meaning of document relationships.

4.1.6.2 Association type signs

An ebRIM Association with associationType of *signs* shall be used to link an XDSDocumentEntry representing a Digital Signature with the XDSDocumentEntry representing the document being signed. Details of how Digital Signatures are represented in XDS are found in the Document Content Profile on Digital Signatures. In constructing this association, the attributes are:

sourceObject: references the XDSDocumentEntry representing the Digital Signature

targetObject: references the XDSDocumentEntry representing the document being signed

associationType: signs

Other requirements on the use of this Association may exist in the Document Content Profile on Digital Signatures.

4.1.6.3 Association Type formatting

ebRIM 3.0 requires a namespace qualified name (urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember) for an Association Type. The urn:oasis:names:tc:ebxml-regrep:AssociationType: namespace prefix only applies to Association Types defined by ebRIM 3.0 (HasMember only). Association Types defined by XDS and related profiles shall use the IHE specific namespace urn:ihe:iti:2007:AssociationType:.

Table 4.1-2.1: Association Types used in XDS and related profiles

ebRIM 3.0 Format
urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember
urn:ihe:iti:2007:AssociationType:RPLC
urn:ihe:iti:2007:AssociationType:XFRM

ebRIM 3.0 Format
urn:ihe:iti:2007:AssociationType:APND
urn:ihe:iti:2007:AssociationType:XFRM_RPLC
urn:ihe:iti:2007:AssociationType:signs

4.1.7 Document Definition Metadata

Several data types are used in the tables below describing the document metadata. These data types are derived from other standards, and encoded in the registry as described in the following table. For entries where no Data Type is specified the entry is any string of bytes that fits within the length defined by the schema.

For the data types derived from HL7 standards, XDS requires that the default HL7 separators be used to represent the structure of HL7 V2 data types:

Field Separator	|
Component Separator	^
Subcomponent Separator	&
Repetition Separator	~

Table 4.1-3: Data Types

XDS Data Type	Source Standard	Encoding Specification
CX	HL7 V2 Identifier	This is an identifier. HL7 Identifier type CX consist 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 represented using a Universal ID and Universal ID Type. In XDS specification, 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: 543797436^^^&1.2.840.113619.6.197&ISO Note that the '&' character must be properly encoded in the XML content. See the examples in the tables below for the appropriate representation.
DTM	HL7 V2 Date Time	This is a date/time value, represented as precisely as possible. All date time values in the registry are stored using universal coordinated time [UTC]. "UTC" implies that the source and the consumer shall convert the time from/to the local time.

XDS Data Type	Source Standard	Encoding Specification
		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
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 represented without leading 0 digits unless the value is zero. 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
Field	HL7 V2 Message Segment	Specified as the Field identifier, followed by a pipe (|) and then the data value represented with corresponding HL7 V2 data type as defined in HL7 standard. Note that if a HL7 data type is used to derive XDS data type (as shown in this table), the derived XDS data type shall be 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
SHA1	Document hash calculated with SHA1 algorithm	See RFC 3174 US Secure Hash Algorithm 1 (SHA1), September 2001 The encoding is the Lexical Representation of hexBinary ([0-9a-fA-F]).
URI	Uniform Resource Identifier	See RFC 2616
UUID	Universally Unique Identifier	A DCE Universally Unique Identifier, represented in registry attributes using the URN syntax for UUIDs: urn:uuid:9e0110f8-4748-4f1e-b0a8-cecae32209c7
XCN	HL7 V2 Extended Person Name	This data type describes a person along with the identifier by which he is known in some domain (either the source domain or the XDS affinity domain), using the HL7 v2.5 XCN data type. This data type contains,

XDS Data Type	Source Standard	Encoding Specification
		amongst others, Identifier Last Name First Name Second and Further Given Names Suffix Prefix Assigning Authority All of the HL7 v2.5 fields may be specified as optional components with the following restrictions: 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 bytes, is not exceeded. If component 1 (ID Number) is specified, component 9 (Assigning Authority) shall be present if available. The XDS XCN Component 9 is subject to the same the restrictions as defined for the XDS 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'. Any empty component shall be treated by the Document Registry as not specified. This is in compliance with HL7 v2.5. Trailing delimiters are recommended to be trimmed off. Document Registries shall ignore trailing delimiters. This is in compliance with HL7 v2.5. An example of person name with ID number using this data type is as follows: 11375^Welby^Marcus^J^Jr. MD^Dr^^^&1.2.840.113619.6.197&ISO
XON	HL7 V2 Organization Name	This type provides the name and identification of an organization. This specification restricts the coding to the following fields: XON.1 – Organization Name – this field is required XON.6.2 – Assigning Authority Universal Id – this field is required if XON.10 is valued and not an OID XON.6.3 – Assigning Authority Universal Id Type – this field is required if XON.10 is valued and not an OID and shall have the value "ISO" XON.10 – Organization Identifier – this field is optional No other fields shall be specified. The XON data type in XDS 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

XDS Data Type	Source Standard	Encoding Specification
		organization. The XDS affinity domain must ensure that this assumption is correct, through appropriate policies for assigning authorities. 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 Extended Telecommunica-tions Number	This type provides the telecommunications address of an entity (for example author, intended recipient). 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. The XTN data type in XDS Metadata results in a valid encoding of an HL7 V2.5 XTN data type. Example: ^^Internet^radiology@healthcare.example.org

The source/query column indicates which attributes are required during submission, and whether the registry must support the ability to execute queries against them.

395 **Table 4.1-4: Codes for Source Column**

Code	Meaning
R	Required
R2	Required if Known
O	Optional
P	Registry is not required to support query of this attribute.
Cp	Computed/Assigned by Repository, required in register transaction.
Cg	Computed/Assigned by Registry
Cx	Optionally Computed/Assigned by a Document Registry or Initiating Gateway

The XDSDocumentEntry object type is created in ebXML Registry by extending the ebXML Registry ObjectType Classification Scheme[2].

The following metadata elements shall be used to describe an XDS Document. They shall be
400 provided by the Document Repository Actor in the Register Document Set Transaction either directly if grouped with a Document Source Actor or forwarded from a Provide and Register Document Set Transaction.

The XDSDocumentEntry.repositoryUniqueId shall be supplied by the Document Repository Actor. It identifies the repository that is storing the document.

[2] The specific requirement in ebRIM that object types be user extendable was introduced after version 2.0.

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

Table 4.1-5: Document Metadata Attribute Definition

XDSDocumentEntry Attribute	Definition	Source	Constraints

author	Represents the humans and/or machines that authored the document. This attribute contains the following sub-attributes:
	• authorInstitution
	• authorPerson
	• authorRole
	• authorSpecialty
	• authorTelecommunication
	which are individually defined below.
	
	The author attribute is defined as a Classification which contains the above sub-attributes. The author attribute itself does not have a simple value. It defines a structure to hold its sub-attributes. An instance of this Classification shall be considered a single value of the author attribute. If present, the author attribute shall have one or more sub-attributes. Each instance of this Classification shall contain:
	• Zero or one instance of the authorPerson sub-attribute
	• Zero or more instances of the authorInstitution sub-attribute
	• Zero or more instances of the authorRole sub-attribute
	• Zero or more instances of the authorSpecialty sub-attribute
	• Zero or more instances of the authorTelecommunication sub-attribute
	
	At least an authorPerson, authorTelecommunication, or authorInstitution sub-attribute shall be present when the author attribute is included in the document metadata.
	The following example shows the definition of a single author. The classification shows the authorPerson Slot holding the required single value. Single values are shown for authorInstitution, authorRole, authorSpecialty, and authorTelecommunication. Multiple values for the three multi-valued sub-attributes, if present, shall be coded as additional Value elements within the Slot/ValueList having the correct name.
	
	`<rim:Classification`
	`classificationScheme="urn:uuid:93606bcf-9494-43ec-9b4e-a7748d1a838d"`
	`classifiedObject="theDocument"`
	`id=”ID_045”`
	`objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"`
	`nodeRepresentation="">`
	`<!-- nodeRepresentation intentionally left blank -->`
		R2	ebRIM

XDSDocumentEntry Attribute	Definition	Source	Constraints
	<rim:Slot name="authorPerson"> <!-- shall be single valued, includes optional ID --> <rim:ValueList> <rim:Value>D12398^Doe^John^^^^^^&1.2.3.4.5.6.7.8.9.1789.45.1&ISO</rim:Value> </rim:ValueList> </rim:Slot> <rim:Slot name="authorInstitution"> <!-- may be multivalued --> <rim:ValueList> <rim:Value> Some Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45</rim:Value> </rim:ValueList> </rim:Slot> <rim:Slot name="authorRole"> <!-- may be multivalued --> <rim:ValueList> <rim:Value>name of role</rim:Value> </rim:ValueList> </rim:Slot> <rim:Slot name="authorSpecialty"> <!-- may be multivalued --> <rim:ValueList> <rim:Value>specialty of author</rim:Value> </rim:ValueList> </rim:Slot> <rim:Slot name="authorTelecommunication"> <!-- shall be single valued --> <rim:ValueList> <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
authorInstitution (sub-attribute of author)	Represents a specific healthcare facility under which the human and/or machines authored the document. A specific case is that of homecare. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	R2	XON
authorPerson (sub-attribute of author)	Represents the humans and/or machines that authored the document within the authorInstitution. The document author may be the patient itself. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	R2	XCN

XDSDocumentEntry Attribute	Definition	Source	Constraints
authorRole (sub-attribute of author)	A code that represents the role of the author with respect to the patient when the document was created. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	R2	
authorSpecialty (sub-attribute of author)	Represents a specific specialty within a healthcare facility under which the human and/or machines authored the document. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	R2	
authorTelecommunication	Represents the telecommunications address (e.g. email) of the document author, intended to assist with automated routing of other messages intended for the document author. See the author attribute for definition of the requirements of usage. See author for example	O	XTN
availabilityStatus	An XDS Document shall have one of two availability statuses: Approved available for patient care Deprecated obsolete This attribute is always set to Approved as part of the submission of new XDS Documents. It may be changed to Deprecated under the primary responsibility of the Document Source with possible patient supervision. Although XDS supports the ability to delete documents, there is no such state as "the Document Entry is removed" (only an audit trail is kept if such a deletion is allowed). This list may be extended in the future. See ITI TF-3: 4.1.3.3 Atomicity Requirements for XDS Submission Requests for additional details. If present, shall have a single value. The example below shows the status attribute, however, this attribute is only returned on query, not set during any registry or repository transaction. `<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" mimeType="application/octet-stream" > …`	Cg	
classCode	The code specifying the particular kind of document (e.g., Prescription,	R	XDS

XDSDocumentEntry Attribute	Definition	Source	Constraints

	Discharge Summary, Report). It is suggested that the XDS Affinity Domain draws these values from a coding scheme providing a coarse level of granularity (about 10 to 100 entries). Shall have a single value.
	
	`<rim:Classification`
	`classificationScheme=`
	`"urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"`
	`classifiedObject="theDocument"`
	`id="ID_046"`
	`objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"`
	`nodeRepresentation="classCode"`
	`>`
	`<rim:Name>`
	`<rim:LocalizedString`
	`value="classCodeDisplayName"/>`
	`</rim:Name>`
	`<rim:Slot name="codingScheme">`
	`<rim:ValueList>`
	`<rim:Value>XDS Affinity Domain Specific Value</rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	`</rim:Classification>`
			Affinity Domain specific

classCode DisplayName	The name to be displayed for communicating to a human the meaning of the classCode. Shall have a single value for each value of classCode. See classCode for example.	R	XDS Affinity Domain specific

comments	Comments associated with the Document. Free form text with an XDS Affinity Domain specified usage.
	
	`<rim:Description>`
	`<rim:LocalizedString value = "comments"/>`
	`</rim:Description>`
		O	XDS Affinity Domain specific

confidentialityCode	The code specifying the level of confidentiality of the XDS Document. These codes are specific to an XDS Affinity Domain. Enforcement and issues related to highly sensitive documents are beyond the scope of XDS (see security section). These issues are expected to be addressed in later years. confidentialityCode is part of a codification scheme and value set enforced by the Document Registry. Shall have one or more values. Code multiple values by creating multiple classification objects.
	`<rim:Classification`
	`classificationScheme=`
	`"urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"`
	`classifiedObject="theDocument"`
	`id="ID_047"`
	`objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"`
	`nodeRepresentation="confidentialityCode"`
	`>`
	`<rim:Name>`
	`<rim:LocalizedString value="displayName"/>`
	`</rim:Name>`
		R	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source	Constraints
	<rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>XDS Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
creationTime	Represents the time the author created the document in the Document Source. Shall have a single value. <rim:Slot name="creationTime"> <rim:ValueList> <rim:Value>20041225212010</rim:Value> </rim:ValueList> </rim:Slot>	R	DTM
entryUUID	This globally unique identifier is primarily intended for use as a document registry management identifier. It is not meant to be an external reference (outside of the Document Registry) for XDS Documents (e.g., in links within other documents). The uniqueId is meant for that purpose so that such links remain valid beyond the XDS Affinity Domain. Values of this attribute are in one of two formats: properly formatted UUID (including urn:uuid: prefix) or symbolic name (anything that does not have the urn:uuid: prefix). Provide and Register Document Set-b [ITI-41] and Register Document Set-b [ITI-42] transactions may carry UUID or symbolic formats. All other transactions shall carry only the UUID format. In processing the Register Document Set-b [ITI-42] transaction, the Document Registry actor shall accept and store the offered UUID values and assign UUID values to all symbolic values. In the XML form, entryUUID is represented by the id attribute. In the example below, the entryUUID is a6e06ca8-0c75-4064-9e5c-88b9045a96f6 <rim:ExtrinsicObject mimeType="application/pdf" id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6" objectType= "urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1" > …	R	UUID
eventCodeList	This list of codes represents the main clinical acts, such as a colonoscopy or an appendectomy, being documented. In some cases, the event is inherent in the typeCode, such as a "History and Physical Report" in which the procedure being documented is necessarily a "History and Physical" act. An event can further specialize the act inherent in the typeCode, such as where it is simply "Procedure Report" and the procedure was a "colonoscopy". If one or more eventCodes are included, they shall not conflict with the values inherent in the classCode, practiceSettingCode or typeCode, as such a conflict would create an ambiguous situation. This short list of codes is provided to be used as “key words” for certain types of queries. If present, shall have one or more values. Code multiple values by creating multiple classification objects.	O	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source	Constraints
	<rim:Classification classificationScheme= "urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4" classifiedObject="theDocument" id="ID_048" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification" nodeRepresentation="eventCode" > <rim:Name> <rim:LocalizedString value="eventCodeDisplayName"/> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>XDS Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
eventCodeListDisplayName	The list of names to be displayed for communicating to human reader the meaning of the eventCode. If present, shall have a single value corresponding to each value in eventCodeList. See eventCodeList for an example.	O[3]	XDS Affinity Domain specific
formatCode	Code globally uniquely specifying the format of the document. Along with the typeCode, it should provide sufficient information to allow any potential XDS Document Consumer to know if it will be able to process 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). Shall have a single value. Format codes may be specified by multiple organizations. Format codes defined by ITI 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 Affinity Domain shall have names with the prefix urn:ad:'name of affinity domain': Affinity Domains shall be unique. The prefixes described here are not assumed to be exhaustive. <rim:Classification classificationScheme=	R	XDS Affinity Domain specific

[3] Required if eventCode has a value.

XDSDocumentEntry Attribute	Definition	Source	Constraints
	"urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d" classifiedObject="theDocument" id="ID_049" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification" nodeRepresentation="formatCode" > <rim:Name> <rim:LocalizedString value="name"/> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>XDS Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
formatCodeDisplayName	The name to be displayed for communicating to human reader the meaning of the formatCode. Shall have a single value corresponding to the value in formatCode. See formatCode for an example.	R	XDS Affinity Domain specific
hash	Hash key of the XDS Document itself. This value is computed by the Document Repository and used by the Document Registry for detecting the improper resubmission of XDS Documents. If present, shall have a single value. If this attribute is received in a Provide & Register Document Set-b [ITI-41] transaction, it shall be verified by the repository with the actual hash value of the submitted document; an XDSRepositoryMetadataError shall be returned on mismatch. <rim:Slot name="hash"> <rim:ValueList> <rim:Value> da39a3ee5e6b4b0d3255bfef95601890afd80709 </rim:Value> </rim:ValueList> </rim:Slot>	Cp	SHA1 hash See Table 4.1-3
healthcareFacilityTypeCode	This code represents the type of organizational setting of the clinical encounter during which the documented act occurred. In some cases, the setting of the encounter is inherent in the typeCode, such as "Diabetes Clinic Progress Note". healthcareFacilityTypeCode shall be equivalent to or further specialize the value inherent in the typeCode; for example, where the typeCode is simply "Clinic Progress Note" and the value of healthcareFacilityTypeCode is "private clinic". The value shall not conflict with the value inherent in the typeCode, as such a conflict would create an ambiguous situation. Shall have a single value. <rim:Classification classificationScheme= "urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1" classifiedObject="theDocument" id="ID_050"	R	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source	Constraints
	objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification" nodeRepresentation="healthcareFacilityTypeCode"> <rim:Name> <rim:LocalizedString value="healthcareFacilityTypeCodeDisplayName"/> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>XDS Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
healthcareFacility TypeCodeDisplay Name	The name to be displayed for communicating to a human the meaning of the healthcareFacilityTypeCode. Shall have a single value for each value of healthcareFacilityTypeCode. See healthcareFacilityTypeCode for an example.	R	XDS Affinity Domain specific
homeCommunityId	A globally unique identifier for a community.	Cx	64 character OID in URI syntax See TF-2a: 3.18.4.1.2.3.8 ITI TF-2b: 3.38.4.1.2.1
languageCode	Specifies the human language of character data in the document. The values of the attribute are language identifiers as described by the IETF (Internet Engineering Task Force) RFC 3066. This value may further be restricted by the registry according to XDS Affinity Domain specific policy. Shall have a single value. <rim:Slot name="languageCode"> <rim:ValueList> <rim:Value>en-us</rim:Value> </rim:ValueList> </rim:Slot>	R	
legalAuthenticator	Represents a participant who has legally authenticated or attested the document within the authorInstitution. Legal authentication implies that a document has been signed manually or electronically by the legalAuthenticator. This attribute may be absent if not applicable. If present, shall have a single value <rim:Slot name="legalAuthenticator"> <rim:ValueList> <rim:Value>^Welby^Marcus^^^Dr^MD</rim:Value>	O	XCN

XDSDocumentEntry Attribute	Definition	Source	Constraints
	</rim:ValueList> </rim:Slot>		
mimeType	MIME type of the document in the Repository. Shall have a single value. <rim:ExtrinsicObject mimeType="application/pdf" id="theDocument" objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1" > …	R	
patientId	The patientId represents the subject of care of the document. This identifier shall be from the Assigning Authority Domain supporting the XDS Affinity Domain in which the Document Registry operates. It shall contain two parts: Authority Domain Id (enforced by the Registry) An Id in the above domain. No other values are allowed, as specified for the CX type in Table 4.1-3 above. Using HL7 terminology, no other values are allowed in the components of the coded value, nor are further subcomponents allowed. The value of the patientId shall be the same for all new documents of a Submission Set. Shall have a single value. <rim:ExternalIdentifier identificationScheme= "urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446" value="6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO" id=”ID_051" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject=”DocumentEntry01”> <rim:Name> <rim:LocalizedString value = "XDSDocumentEntry.patientId"/> </rim:Name> </rim:ExternalIdentifier>	R	CX
practiceSettingCode	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 XDS Affinity Domain draws these values from a coding scheme providing a coarse level of granularity (about 10 to 100 entries). Shall have a single value. <rim:Classification classificationScheme= "urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead" classifiedObject="theDocument" id=”ID_052" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification" nodeRepresentation="practiceSettingCode"	R	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source	Constraints
	> <rim:Name> <rim:LocalizedString value="practiceSettingCodeDisplayName" /> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>XDS Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
practiceSettingCode DisplayName	The name to be displayed for communicating to a human the meaning of the practiceSettingCode. Shall have a single value for each value of practiceSettingCode. See practiceSettingCode for an example.	R	XDS Affinity Domain specific
repositoryUniqueId	The globally unique identifier of the repository where the document is stored, assigned by the Document Repository. This unique identifier for the Document Repository may be used to identify and connect to the specific Document Repository where the document is stored once its metadata has been retrieved from a Document Registry. This repositoryUniqueId is intended to respond to the following types of usage: The means to reference the Document Repository where this XDS document is stored. The repositoryUniqueId represents an immutable id for the Document Repository. The means to ensure that a XDS Document is retrieved from the appropriate Document Repository. Shall have a single value. <rim:Slot name="repositoryUniqueId"> <rim:ValueList> <rim:Value>1.3.6.1.4…</rim:Value> </rim:ValueList> </rim:Slot>	Cp	ITI TF-2b: 3.41.4.1 3.42.4.1.2
serviceStartTime	Represents the start time 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 XDS metadata but may be coded in documents managed by XDS. This time is expressed as (date/time/UTC). If present, shall have a single value. Note: Other times, such as document creation or approval are to be recorded, if needed, within the document. <rim:Slot name="serviceStartTime"> <rim:ValueList> <rim:Value>20041225212010</rim:Value> </rim:ValueList> </rim:Slot>	R2	HL7 V2 DTM
serviceStopTime	Represents the stop time the service being documented took place	R2	HL7 V2

XDSDocumentEntry Attribute	Definition	Source	Constraints

	(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 XDS metadata but may be coded in documents managed by XDS. This time is expressed as (date/time/UTC). If the Service happens at a point in time, this attribute shall contain the same value as the serviceStartTime. If present, shall have a single value.
	
	`<rim:Slot name="serviceStopTime">`
	`  <rim:ValueList>`
	`  <rim:Value>20041225232010</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
			DTM

size	Size in bytes of the byte stream that was provided in the Register and Provide Transaction and stored by the XDS Document Repository. This value is computed by the Document Repository and included in the Register Documents Set Transaction. If present, shall have a single value.
	If this attribute is received in a Provide & Register Document Set-b [ITI-41] transaction, it shall be verified by the repository with the actual size of the submitted document; an XDSRepositoryMetadataError shall be returned on mismatch.
	
	`<rim:Slot name="size">`
	`  <rim:ValueList>`
	`  <rim:Value>3654</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		Cp	Integer
			
			ITI TF-2b:
			3.41.4.1

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. It shall contain two parts:
	Authority Domain Id
	An Id in the above domain (e.g., Patient Id).
	This sourcePatientId is not intended to be updated once the Document is registered (just as the Document content and metadata itself will not be updated without replacing the previous document). As this sourcePatientId may have been merged by the source actor, it may no longer be in use within the Document Source (EHR-CR). It is only intended as an audit/checking mechanism and has occasional use for Document Consumer Actors. There can be only one Slot named sourcePatientId.
	
	`<rim:Slot name="sourcePatientId">`
	`  <rim:ValueList>`
	`  <rim:Value>`
	`j98789^^^&1.2.3.4.343.1&ISO</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		R	CX

sourcePatientInfo	This attribute should contain demographics information of the patient to whose medical record this document belongs, as the Document Source	O	

XDSDocumentEntry Attribute	Definition	Source	Constraints

	knew it at the time of Submission.
	This information typically includes: the patient first and last name, sex, and birth date. The Clinical XDS Affinity Domain policies may require more or less specific information and format.
	This patient information is not intended to be updated once the Document is registered (just as the Document content and metadata itself will not be updated without replacing the previous document). As sourcePatientInfo may have been updated by the source actor, it may no longer be in use within the Document Source (EHR-CR). It is only intended as an audit/checking mechanism and has occasional use for Document Consumer actors. Shall have a single value (only a single sourcePatientInfo slot may be present).
	
	`<rim:Slot name="sourcePatientInfo">`
	`  <rim:ValueList>`
	`<rim:Value>PID-3|DTP-1^^^&1.3.6.1.4.1.21367.2005.3.7 &ISO</rim:Value>`
	`  <rim:Value>PID-5|DICTAPHONE^ONE^^^</rim:Value>`
	`  <rim:Value>PID-7|19650120</rim:Value>`
	`  <rim:Value>PID-8|M</rim:Value>`
	`  <rim:Value>PID-11|100 Main St^^BURLINGTON^MA^01803^USA</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
	PID-3 should include the source patient identifier.
	PID-5 should include the patient name.
	PID-8 should code the patient gender as
	M – MaleF – Female
	O – OtherU – Unknown
	PID-7 should include the patient date of birth.
	PID-11 should include the patient address.
	PID-2, PID-4, PID-12 and PID-19 should not be used.		

title	Represents the title of the document. Clinical documents often do not have a title, and are collectively referred to by the display name of the classCode (e.g., a "consultation" or "progress note"). Where these display names are rendered to the clinician, or where the document has a unique title, the title component shall be used. Max length, 128 bytes, UTF-8. If present, shall have a single value.
	
	`<rim:ExtrinsicObject`
	`  id="theDocument"`
	`  objectType=`
	`  "urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"`
	`  mimeType="application/pdf"`
	`>`
	`  <rim:Name>`
	`  <rim:LocalizedString value="title"/>`
	`  </rim:Name>`
	`  …`
	`</rim:ExtrinsicObject>`
		O	

XDSDocumentEntry Attribute	Definition	Source	Constraints

typeCode	The code specifying the precise kind of document (e.g., Pulmonary History and Physical, Discharge Summary, Ultrasound Report). It is suggested that the XDS Affinity Domain draw these values from a coding scheme providing a fine level of granularity. Shall have a single value.
	
	`<rim:Classification`
	`classificationScheme=`
	`" urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"`
	`classifiedObject="theDocument"`
	`nodeRepresentation="typeCode"`
	`id=”ID_053"`
	`objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"`
	`>`
	`<rim:Name>`
	`<rim:LocalizedString`
	`value="typeCodeDisplayName"`
	`/>`
	`</rim:Name>`
	`<rim:Slot name="codingScheme">`
	`<rim:ValueList>`
	`<rim:Value>XDS Affinity Domain Specific Value</rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	`</rim:Classification>`
		R	XDS Affinity Domain specific

typeCodeDisplay Name	The name to be displayed for communicating to a human the meaning of the typeCode. Shall have a single value for each value of typeCode. See typeCode for an example.	R	XDS Affinity Domain specific

uniqueId	The globally unique identifier assigned by the document creator to this document. This unique identifier may be used in the body of other XDS Documents to reference this document. The length of Unique Identifier shall not exceed 128 bytes. The structure and format of this Id shall be consistent with the specification corresponding to the format attribute. (e.g., for a DICOM standard document a 64 character numeric UID, for an HL7 CDA format a serialization of the CDA Document id extension and root in the form oid^extension, where OID is a 64 digits max, and the ID is a 16 UTF-8 char max). If the oid is coded without the extension then the '^' character shall not be included.
	This uniqueId is intended to respond to the following types of usage:
	The means to reference this XDS document from within the content of another document. Neither the XDS Registry nor the Repository is aware of such references, but the Document Sources and Consumers are.
	Shall have a single value.
	
	`<rim:ExternalIdentifier`
	`identificationScheme=`
	`"urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"`
	`value="1.3.6.1.4.1.21367.2005.3.7^11379"`
	`id=”ID_054"`
	`objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier"`
		R	See ITI TF-3: 4.1.7.2

XDSDocumentEntry Attribute	Definition	Source	Constraints
	registryObject=”DocumentEntry01”> <rim:Name> <rim:LocalizedString value="XDSDocumentEntry.uniqueId"/> </rim:Name> </rim:ExternalIdentifier>		
URI	For XDM the URI element shall point to the file containing the document. If present, shall have a single value. There are two formats for coding this attribute. If the string representing the URI is 128 characters or shorter, the short format may be used: <rim:Slot name="URI"> <rim:ValueList> <rim:Value>http://www.ihe.net</rim:Value> </rim:ValueList> </rim:Slot> If the string is more than 128 characters long, the long format shall be used: <rim:Slot name="URI"> <rim:ValueList> <rim:Value>1|http://www.ihe.net/IHERetrieveDocument? </rim:Value> <rim:Value>2|requestType=DOCUMENT&documentUID=1.2.3 </rim:Value> <rim:Value>3|&preferredContentType=application%2fpdf </rim:Value> </rim:ValueList> </rim:Slot> Each Value is composed of an ordering prefix followed by a portion of the actual URI string. The ordering prefix shall be sequential starting at the value 1. When the long format is used, all Values shall have an ordering prefix. Each value is ordered by its ordering prefix: ordering-prefix :== digit vertical-bar digit :== ‘1’ | ‘2’ | ‘3’ | ‘4’ | ‘5’ | ‘6’ | ‘7’ | ‘8’ | ‘9’ vertical-bar :== ‘|’ The long version may be used for URIs of less than 129 characters. This profile does not specify how a URI is to be broken up into pieces. The following example is valid. <rim:Slot name="URI"> <rim:ValueList> <rim:Value>1|http://www.ihe.net</rim:Value> </rim:ValueList> </rim:Slot>	O	URI ITI TF-2b: 3.32.4.1.2.2

XDSDocumentEntry Attribute	Definition	Source	Constraints
	The long version is assembled into a URI by concatenating the Values without the ordering prefixes in the order specified by the ordering-prefixes.		

4.1.7.1 XDSDocumentEntry.formatCode

In general, the repository holds an octet stream representing the document. The registry metadata describes, among other things, the format of the document. This is coded in XDSDocumentEntry.formatCode. This code will identify document format parameters necessary for interoperability. Rules about handling the formatCode are necessary but are not imposed by XDS. In the future IHE content specific Integration Profiles may be created that specify these rules.

Note: Although only a small number of document standards may be used, a large number of code values may be defined to point to specific templates and archetypes structuring specific document content.

4.1.7.2 XDSDocumentEntry.uniqueId

The specification of the format and encoding for this attribute depends on the document standard defining the content of the XDS Document (e.g., OID with optional extension ID for HL7 CDA, UUID in some cases, SOP Instance UID for DICOM composite objects. Format is: OID^Extension). This attribute shall not exceed 128 bytes in size. It shall be used as an opaque and globally unique identifier for the XDS Document. Document Consumers, Registries, Repositories shall not attempt to interpret its content. When the Extension is not present, the '^' character shall not be included.

4.1.7.3 XDSDocumentEntry.repositoryUniqueId

To better match the Web Services messaging architecture and provide a MTOM/XOP binding for the Retrieve Document Set and the Provide and Register Document Set-b transactions, it is necessary to further specify the location of the document to identify the actual Document Repository that contains it before the Document Repository can be queried to retrieve the actual document.

The Document Repository shall populate the following attribute in the XDSDocumentEntry class:

- repositoryUniqueId: this single-valued attribute of type OID represents the unique id of the Document Repository that stores the document. The attribute is populated by the Document Repository as part of the Provide and Register Document Set-b transaction. The Document Repository id is considered immutable throughout the lifetime of the Document Repository to which it is associated. In other words, once an id has been associated to a Document Repository it can never change. The repositoryUniqueId attributes are defined in a community and assigned to Document Repository actors.

The Document Repository shall populate this attribute before registering documents in the Document Registry. This allows for positive identification of the web service endpoint of the Document Repository for the purposes of retrieving a document or set of documents. The mechanism by which the service endpoints are discovered and associated to the appropriate actors and how that configuration is maintained is out of scope for this transaction.

4.1.8 Submission Set Metadata

The following metadata elements shall be used to describe an XDS Submission Set. They shall be provided by the Document Source Actor in the Provide and Register Document Set transaction. They shall be provided by the Document Repository Actor in the Register Document Set Transaction either directly if grouped with a Document Source Actor or forwarded from a Provide and Register Document Set Transaction.

Each of the attributes listed below is an attribute on the RegistryPackage object defining the Submission Set. The attribute name is defined with a prefix of the object type of XDSSubmissionSet when referenced by other objects, for example XDSSubmissionSet.sourceId.

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.

Table 4.1-6: Submission Set Metadata Attribute Definitions

XDSSubmission Set Attribute	Definition	Source	Constraints
author	Represents the humans and/or machines that authored the submission set. This attribute contains the following sub-attributes: • authorInstitution • authorPerson • authorRole • authorSpecialty • authorTelecommunication which are individually defined below. The author attribute is defined as a Classification which contains the above sub-attributes. The author attribute itself does not have a simple value. It defines a structure to hold its sub-attributes. An instance of this Classification shall be considered a single value of the author attribute. If present, the author attribute shall have one or more sub-attributes. Each instance of this Classification shall contain: • Zero or one instance of the authorPerson sub-attribute • Zero or more instances of the authorInstitution sub-attribute • Zero or more instances of the authorRole sub-attribute • Zero or more instances of the authorSpecialty sub-attribute	R2	ebRIM

XDSSubmission Set Attribute	Definition	Source	Constraints

	• Zero or more instances of the authorTelecommunication sub-attribute
	
	At least an authorPerson, authorTelecommunication, or authorInstitution sub-attribute shall be present when the author attribute is included in the submission set.
	
	The following example shows the definition of a single author. The classification shows the authorPerson Slot holding the required single value. Single values are shown for authorInstitution, authorRole, authorSpecialty, and authorTelecommunication. Multiple values for the three multivalued sub-attributes, if present, shall be coded as additional Value elements within the Slot/ValueList having the correct name.
	
	`<rim:Classification`
	`classificationScheme="urn:uuid:a7058bb9-b4e4-4307-ba5b-e3f0ab85e12d"`
	`classifiedObject="theSubmissionSet"`
	`id=”ID_055"`
	`objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"`
	`nodeRepresentation="">`
	`<!-- nodeRepresentation intentionally left blank -->`
	
	`<rim:Slot name="authorPerson">`
	`<!-- shall be single valued, includes optional ID -->`
	`<rim:ValueList>`
	`<rim:Value> D12398^Doe^John^^^^^^&1.2.3.4.5.6.7.8.9.1789.45.1&ISO </rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	
	`<rim:Slot name="authorInstitution">`
	`<!-- may be multivalued -->`
	`<rim:ValueList>`
	`<rim:Value> Some Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45</rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	
	`<rim:Slot name="authorRole">`
	`<!-- may be multivalued -->`
	`<rim:ValueList>`
	`<rim:Value>name of role</rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	
	`<rim:Slot name="authorSpecialty">`
	`<!-- may be multivalued -->`
	`<rim:ValueList>`
	`<rim:Value>specialty of author</rim:Value>`
	`</rim:ValueList>`		

XDSSubmission Set Attribute	Definition	Source	Constraints
	</rim:Slot> <rim:Slot name="authorTelecommunication"> <!-- shall be single valued --> <rim:ValueList> <rim:Value>^^Internet^john.doe@healthcare.example.org</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
authorInstitution (sub-attribute of author)	Represents a specific healthcare facility under which the human and/or machines authored the Submission Set. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	R2	XON
authorPerson (sub-attribute of author)	Represents the human and/or machines that authored the Submission Set. The document author may be the patient itself. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	O	XCN
authorRole (sub-attribute of author)	A code that represents the role of the author with respect to the patient when the submission set was created. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	R2	
authorSpecialty (sub-attribute of author)	Represents a specific specialty within a healthcare facility under which the human and/or machines authored the submission set. This is a sub-attribute of the author attribute. See the author attribute for definition of the requirements of usage. See author for example.	R2	
authorTelecommunication	Represents the telecommunications address (e.g. email) of the submission set author, intended to assist with automated routing of other messages intended for the submission set author. See the author attribute for definition of the requirements of usage. See author for example	O	XTN
availabilityStatus	An XDSSubmissionSet shall have one of two availability statuses: Submitted submission transaction not completed, not available for patient care Approved available for patient care This attribute is always set to Approved as part of a successful submission of new XDS Documents. XDS does not allow for the	Cg	

XDSSubmission Set Attribute	Definition	Source	Constraints
	deprecation of Submission Sets. See ITI TF-3: 4.1.3.3 Atomicity Requirements for XDS Submission Requests for additional details. If present, shall have a single value. The example below shows the status attribute, however, this attribute is only returned on query, not set during any registry or repository transaction. `<rim:RegistryPackage id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855" status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved" > ...`		
comments	Comments associated with the Submission Set. Free form text with an XDS Affinity Domain specified usage. `<rim:Description> <rim:LocalizedString value = "comments"/> </rim:Description>`	O	Use specific to XDS Affinity Domain.
contentTypeCode	The code specifying the type of clinical activity that resulted in placing these XDS Documents in this XDS-Submission Set. These values are to be drawn for a vocabulary defined by the XDS Affinity Domain. Shall have a single value. `<rim:Classification classificationScheme= "urn:uuid:aa543740-bdda-424e-8c96-df4873be8500" classifiedObject="submissionSet" id="ID_056" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification" nodeRepresentation="contentTypeCode" > <rim:Name> <rim:LocalizedString value="contentTypeCodeDisplayName" /> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>XDS Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>`	R	XDS Affinity Domain specific
contentTypeCode DisplayName	The name to be displayed for communicating to a human the meaning of the contentTypeCode. Shall have a single value for each value of contentTypeCode. See contentTypeCode for an example.	R	XDS Affinity Domain specific

XDSSubmission Set Attribute	Definition	Source	Constraints
entryUUID	This globally unique identifier is primarily intended for use as a document registry management identifier. It is not meant to be an external reference (outside of the Document Registry) for XDS Documents (e.g., in links within other documents). The uniqueId is meant for that purpose so that such links remain valid beyond the XDS Affinity Domain. Values of this attribute are in one of two formats: properly formatted UUID (including urn:uuid: prefix) or symbolic name (anything that does not have the urn:uuid: prefix). Provide and Register.b and Register.b transactions may carry UUID or symbolic formats. All other transactions shall carry only the UUID format. In processing the Register.b transaction, the Document Registry actor shall accept and store the offered UUID values and assign UUID values to all symbolic values. In the XML form, entryUUID is represented by the id attribute. In the example below, the entryUUID is `urn:uuid:a6e06ca8-0c75-4064-9e5d-88b9045a9ab6` `<rim:RegistryPackage mimeType="application/pdf" id="urn:uuid:a6e06ca8-0c75-4064-9e5d-88b9045a9ab6" >` ...	R	UUID
homeCommunityId	A globally unique identifier for a community.	Cx	64 character OID in URI syntax See TF-2a: 3.18.4.1.2.3.8 ITI TF-2b: 3.38.4.1.2.1
intendedRecipient	Represents the organization(s) or person(s) for whom the Submission set is intended. If present, shall have one or more values. Each slot value shall include at least one of the organization, person, or telecommunications address fields described below. Example below shows two doctors from the same organization, another doctor without precision of the organization, another organization without the precision of the person, and just a telecommunications address. The receipt of this attribute in any transaction will not cause the request to be rejected but the attribute may be ignored. Note: It is highly recommended to define the organization for all the persons, avoiding errors in the transmission of the documents internally at the Document Recipient side. There is a "|" character separator between the organization and the person, and between the person and the telecommunications address, which is required when the person or telecommunications address information is present. `<rim:Slot name="intendedRecipient">`	O	XON/XCN

XDSSubmission Set Attribute	Definition	Source	Constraints
	<rim:ValueList> <rim:Value> Some Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45|^Wel^Marcus ^^^Dr^MD|^^Internet^mwel@healthcare.example.org </rim:Value> <rim:Value> Some Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45|^Al^Peter^^ ^Dr^MD</rim:Value> <rim:Value>|12345^John^Smith^^^Dr^MD</rim:Value >| <rim:Value>Main Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.2364</rim:Valu e> <rim:Value>||^^Internet^dr.oz@healthcare.exampl e.org</rim:Value> </rim:ValueList></rim:Slot>		
patientId	The patientId represents the medical record identifier of subject of care whose longitudinal record is being maintained, as selected by the Document Source. Attaching an existing document for patient A to a folder for patient B is presumed in this case to be an update to the longitudinal record for patient B. In this case, the Submission Set patientId would be that of patient B. This identifier shall be from the Assigning Authority Domain supporting the XDS Affinity Domain in which the Document Registry operates. It shall contain two parts: Authority Domain Id (enforced by the Registry) An Id in the above domain. No other values are allowed, as specified for the CX type in Table 4.1-3 above. Using HL7 terminology, no other values are allowed in the components of the coded value, nor are further subcomponents allowed. The value of the patientId shall be the same for all new documents of a Submission Set. Shall have a single value. <rim:ExternalIdentifier identificationScheme= " urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446" value="***6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO***" id=”ID_057" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject=”SubmissionSet01”> <rim:Name> <rim:LocalizedString value = "XDSSubmissionSet.patientId"/> </rim:Name> </rim:ExternalIdentifier>	R	CX
sourceId	OID identifying the instance of the Document Source that contributed the Submission Set. When a "broker" is involved in sending submission sets from a collection of client systems, it should use a different source ID for submissions from each separate system to allow for tracking. Shall have a single value.	R	OID

XDSSubmission Set Attribute	Definition	Source	Constraints
	<rim:ExternalIdentifier identificationScheme= "urn:uuid:554ac39e-e3fe-47fe-b233-965d2a147832" value="1.3.6.1.4.1.21367.2005.3.7" id="ID_058" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject="SubmissionSet01"> <rim:Name> <rim:LocalizedString value = "XDSSubmissionSet.sourceId"/> </rim:Name> </rim:ExternalIdentifier>		
submissionTime	Point in Time at the Document Source when the Submission Set was created and issued for registration to the Document Registry. Shall have a single value. This shall be provided by the Document Source (in case of e-mail with significant delay). <rim:Slot name="submissionTime"> <rim:ValueList> <rim:Value>20041225212010</rim:Value> </rim:ValueList> </rim:Slot>	R	DTM
title	Represents the title of the Submission Set. If present, shall have a single value. <rim:Name> <rim:LocalizedString value="title"/> </rim:Name>	O	XDS Affinity Domain specific
uniqueId	Globally unique identifier for the submission-set instance assigned by the Document Source in OID format. Shall have a single value. <rim:ExternalIdentifier identificationScheme= " urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8" id="ID_059" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier" value="***1.3.6.1.4.1.21367.2005.3.7.3670984664***" registryObject="SubmissionSet01"> <rim:Name> <rim:LocalizedString value = "XDSSubmissionSet.uniqueId"/> </rim:Name> </rim:ExternalIdentifier>	R	OID See ITI TF-2x: Appendix B

460 #### 4.1.8.1 Creating an XDSSubmissionSet object from a RegistryPackage element

An XDSSubmissionSet 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 coded within the RegistryPackage element or on

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

Classification coded inside the RegistryPackage object

```
<…>
    <RegistryPackage id=”SS”>
        <!-- Classify registry package SS as being an XDSSubmissionSet -->
        <Classification classificationNode=”urn:uuid:a54d6aa5-d40d-43f9-88c5-
b4633d873bdd” classifiedObject=”SS” id=”ID_060"
objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"/>
    </RegistryPackage>
    <RegistryPackage id=”Fol”>
        <!-- Classify registry package Fol as being an XDSFolder -->
        <Classification classificationNode=”urn:uuid:d9d542f3-6cc4-48b6-8870-
ea235fbc94c2” classifiedObject=”Fol” id=”ID_061"
objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"/>
    </RegistryPackage>
<…>
```

Classification coded outside the RegistryPackage object

```
<…>
    <RegistryPackage id=”SS”>
    </RegistryPackage>
    <!-- Classify registry package as XDSSubmissionSet -->
    <Classification classificationNode=”urn:uuid:a54d6aa5-d40d-43f9-88c5-
b4633d873bdd” classifiedObject=”SS” id=”ID_060"
objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"/>
    <RegistryPackage id=”Fol”>
    </RegistryPackage>
    <!-- Classify registry package as XDSFolder  -->
    <Classification classificationNode=”urn:uuid:d9d542f3-6cc4-48b6-8870-
ea235fbc94c2” classifiedObject=”Fol” id=”ID_061"
objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"/>
<…>
```


The following UUIDs shall be used to label RegistryPackage elements as Submission Set or Folder

Object being coded	UUID used on the Classification
Submission Set	urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd
Folder	urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2

4.1.9 Folder Metadata

The following metadata elements shall be used to describe an XDS Folder. They shall be provided by the Document Source Actor in the Provide and Register Document Set transaction. They shall be provided by the Document Repository Actor in the Register Document Set transaction if this transaction is used outside the context of a Provide and Register Document Set transaction.

Each of the attributes listed below is an attribute on the RegistryPackage object defining the Folder. The attribute name is defined with a prefix of the object type of XDSFolder when referenced by other objects, for example XDSFolder.patientId.

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.

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 be expected to hold the folder name.

Table 4.1-7: Folder Metadata Attribute Definitions

XDSFolder Attribute	Definition	Source	Constraints
availabilityStatus	An XDSFolder shall have one of two availability statuses: Submitted submission transaction not completed, not available for patient care Approved available for patient care This attribute is always set to Approved as part of a successful submission of new XDS Folders. XDS does not allow for the deprecation of Folders. See ITI TF-3: 4.1.3.3 Atomicity Requirements for XDS Submission Requests for additional details. If present, shall have a single value. The example below shows the status attribute, however, this attribute is only returned on query, not set during any registry or repository transaction. <rim:RegistryPackage	Cg	

XDSFolder Attribute	Definition	Source	Constraints
	id="urn:uuid:fbeacdb7-5421-4474-9267-985007cd8855" status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved" > ...		

codeList	Shall contain the set of codes specifying the type of clinical activity that resulted in placing XDS Documents in this XDSFolder. These values shall be drawn from a vocabulary or coding scheme defined by the XDS Affinity Domain.
	Shall have one or more values. Code multiple values by creating multiple classification objects.
	
	`<rim:Classification`
	`classificationScheme=`
	`"urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5"`
	`classifiedObject="Folder"`
	`id="ID_062"`
	`objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification"`
	`nodeRepresentation="codeList"`
	`>`
	`<rim:Name>`
	`<rim:LocalizedString`
	`value="codeListCodeDisplayName" />`
	`</rim:Name>`
	`<rim:Slot name="codingScheme">`
	`<rim:ValueList>`
	`<rim:Value>XDS Affinity Domain Specific Value</rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	`</rim:Classification>`
		R	Multi-Valued. XDS Affinity Domain specific

codeListDisplayName	The list of human readable descriptions of the meaning of each of the codes present in the codeList. Shall have a single value corresponding to each value in codeList. Only one code may be assigned to the Folder when an XDS Document is placed in such a Folder. See codeList for an example.	R	Multi-valued.

comments	Comments associated with the Folder. Free form text with an XDS Affinity Domain specified usage.
	
	`<rim:Description>`
	`<rim:LocalizedString value = "comments"/>`
	`</rim:Description>`
		O	XDS Affinity Domain specific.

entryUUID	This globally unique identifier is primarily intended for use as a document registry management identifier. It is not meant to be an external reference (outside of the Document Registry) for XDS Documents (e.g., in links within other documents). The uniqueId is meant for that purpose so that such links remain valid beyond the XDS Affinity Domain. Values of this attribute are in one of two formats: properly formatted UUID (including urn:uuid: prefix) or symbolic name (anything that does not have the urn:uuid: prefix). Provide and Register.b and Register.b transactions may carry UUID or symbolic formats. All	R	UUID

XDSFolder Attribute	Definition	Source	Constraints
	other transactions shall carry only the UUID format. In processing the Register.b transaction, the Document Registry actor shall accept and store the offered UUID values and assign UUID values to all symbolic values. In the XML form, entryUUID is represented by the id attribute. In the example below, the entryUUID is urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a9ab6 `<rim:RegistryPackage mimeType="application/pdf" id="urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a9ab6" > ...`		
homeCommunityId	A globally unique identifier for a community.	Cx	64 character OID in URI syntax See TF-2a: 3.18.4.1.2.3.8 ITI TF-2b: 3.38.4.1.2.1
lastUpdateTime	Point in time at the Document Registry when an XDS Document was registered and placed in the XDS Folder. If present, shall have a single value. `<rim:Slot name="lastUpdateTime"> <rim:ValueList> <rim:Value>20041225212010</rim:Value> </rim:ValueList> </rim:Slot>` The Document Registry shall set lastUpdateTime on submission of folder. The value in the submission request (if present), shall be ignored.	Cg	DTM
patientId	The patientId represents the subject of care medical record Identifier as defined by the Document Source. This identifier shall be from the Assigning Authority Domain supporting the XDS Affinity Domain in which the Document Registry operates. It shall contain two parts: Authority Domain Id (enforced by the Registry) An Id in the above domain. No other values are allowed, as specified for the CX type in Table 4.1-3 above. Using HL7 terminology, no other values are allowed in the components of the coded value, nor are further subcomponents allowed. The value of the patientId shall be the same for all new documents of a Folder. Shall have a single value. `<rim:ExternalIdentifier identificationScheme=" urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a "`	R	CX

XDSFolder Attribute	Definition	Source	Constraints
	value="6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO" id="ID_063" objectType="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:ExternalIdentifier" registryObject="Folder01"> <rim:Name> <rim:LocalizedString value = "XDSFolder.patientId"/> </rim:Name> </rim:ExternalIdentifier>		
title	Represents the title of the Folder. If present, shall have a single value. <rim:Name> <rim:LocalizedString value="title"/> </rim:Name>	O	XDS Affinity Domain specific
uniqueId	Globally unique identifier for the XDS-Folder in which one or more XDS Documents are placed. It is assigned by the Document Source at the time the XDS Folder is created in OID format. Shall have a single value. <rim:ExternalIdentifier identificationScheme= "urn:uuid:75df8f67-9973-4fbe-a900-df66cefecc5a" id="ID_064" 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>	R	OID See ITI TF-2x: Appendix B

4.1.9.1 Creating an XDSFolder object from a RegistryPackage element

490 An XDSFolder 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 coded within the RegistryPackage element or on the same level. The following XML example demonstrates these two valid approaches to coding the Classification.

495 **Classification coded inside the RegistryPackage object**

```
<…>
    <RegistryPackage id="SS">
        <!-- Classify registry package SS as being an XDSSubmissionSet -->
        <Classification classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-
```



```
b4633d873bdd" classifiedObject="SS" id="ID_065"
objectType="urn:oasis:names:tc:ebxmlregrep:ObjectType:RegistryObject:Classific
ation"/>
    </RegistryPackage>
    <RegistryPackage id="Fol">
        <!-- Classify registry package Fol as being an XDSFolder -->
        <Classification classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-
ea235fbc94c2" classifiedObject="Fol" id="ID_066"
objectType="urn:oasis:names:tc:ebxmlregrep:ObjectType:RegistryObject:Classific
ation"/>
    </RegistryPackage>
<…>
```

Classification coded outside the RegistryPackage object

```
<…>
    <RegistryPackage id="SS">
    </RegistryPackage>
    <!-- Classify registry package as XDSSubmissionSet -->
    <Classification classificationNode="urn:uuid:a54d6aa5-d40d-43f9-88c5-
b4633d873bdd" classifiedObject="SS" id="ID_065"
objectType="urn:oasis:names:tc:ebxmlregrep:ObjectType:RegistryObject:Classific
ation"/>
    <RegistryPackage id="Fol">
    </RegistryPackage>
    <!-- Classify registry package as XDSFolder  -->
    <Classification classificationNode="urn:uuid:d9d542f3-6cc4-48b6-8870-
ea235fbc94c2" classifiedObject="Fol" id="ID_066"
objectType="urn:oasis:names:tc:ebxmlregrep:ObjectType:RegistryObject:Classific
ation"/>
<…>
```

The following UUIDs shall be used to label RegistryPackage elements as Submission Set or Folder

Object being coded	UUID used on the Classification
Submission Set	urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd
Folder	urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2

4.1.10 Registry Adaptor Enforcement of Attributes

ebRIM version 3.0 datatype Slot/ValueList/Value is limited to 256 characters by that standard. Sometimes HL7 datatypes, which the attribute tables show as being encoded as a Slot, can be

larger. The Document Source shall encoded these Slots so they fit into the 256 character space allocated to them. This may require some information to be excluded. This profile gives no
505 guidance as to how information is to be excluded to make this coding limit.

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

Table 4.1-8: Document Metadata Attribute Enforcement

XDSDocumentEntry Attribute	Registry Enforcement
availabilityStatus	No enforcement
authorInstitution	No enforcement
authorPerson	No enforcement
authorRole	No enforcement
authorSpecialty	No enforcement
classCode	Coding Scheme and Code Value.
classCodeDisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
confidentialityCode	Coding Scheme and Code Value
confidentialityCodeDisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
creationTime	No enforcement
entryUUID	No enforcement
eventCodeList	Coding Scheme and Code Value
eventCodeDisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
formatCode	Coding Scheme and Code Value
formatCodeDisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
hash	Hash value enforcement
healthcareFacilityTypeCode	Coding Scheme and Code Value
healthcareFacilityTypeCodeDisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
legalAuthenticator	No enforcement
languageCode	Optionally enforced by XDS Affinity Domain
mimeType	Code Value
parentDocumentRelationship	One of four values
parentDocumentId	Existing UUID
patientId	Authority Domain Id Patient Id (known from patient identity feed)

XDSDocumentEntry Attribute	Registry Enforcement
practiceSettingCode	Coding Scheme and Code Value
practiceSettingCode DisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
serviceStartTime	No enforcement
serviceStopTime	Verifies serviceStartTime <= serviceStopTime
size	No enforcement
sourcePatientId	No enforcement
sourcePatientInfo	Some parts required
title	No enforcement
typeCode	Coding Scheme and Code Value
typeCodeDisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
uniqueId	See ITI TF-3: 4.1.7.2
URI	No enforcement

510

Table 4.1-9: SubmissionSet Metadata Attribute Enforcement

XDSSubmissionSet Attribute	Registry Enforcement
authorInstitution	No enforcement
authorPerson	No enforcement
authorRole	No enforcement
authorSpecialty	No enforcement
comments	No enforcement
contentTypeCode	Coding Scheme and Code value
contentTypeCodeDisplayName	Must be non-zero length and may be restricted by XDS Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
patientId	Authority Domain Id Patient Id (known from patient identity feed)
sourceId	No enforcement
submissionTime	No enforcement
uniqueId	No identical existing uniqueId in registry according to rules in ITI TF-3: 4.1.7.2

Table 4.1-10: Folder Metadata Attribute Enforcement

XDSFolder Attribute	Registry Enforcement
codeList	Coding Scheme and Code value
codeListDisplayName	Must be non-zero length and may be restricted by XDS

XDSFolder Attribute	Registry Enforcement
	Affinity Domain (see 4.1.12.2 Assigning Codes to Documents)
comments	No enforcement
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 XDSDocumentEntry, XDSSubmissionSet, or XDSFolder)

4.1.11 XDS Registry Adaptor

The XDS Registry Adaptor is a set of functionality that is not provided for in the ebXML registry standard, but is instead specified by XDS to support integration into the healthcare environment. This adaptor has the following responsibility:

Validate patient ID – patient identifiers (XDSDocumentEntry.patientId, XDSSubmissionSet.patientId, XDSFolder.patientId) shall be validated by:

- 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 merged patient identifiers received from the Patient Identity Feed [ITI-8]. A submission containing a merged 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

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 (See ITI TF-3: 4.1.10).

Ensure submissions are atomic - The adaptor shall make submission to registry an atomic operation – see ITI TF-3: 4.1.3.3 Atomicity Requirements for Submission Requests for atomicity requirements.

If the registry submission is successful then the adaptor shall label all Document Entry, Folder, and Submission Set objects as Approved. The ebRIM specification provides the ApproveObjectsRequest for this purpose.

If the registry submission fails then the adaptor shall remove from the registry all objects stored as part of this submission set. The ebRIM specification provides the RemoveObjectsRequest for this purpose.

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

Document to be replaced must have status = Approved.

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 section 3.8.4.2.4 of the Technical Framework.

The association's sourceObject attribute shall contain the **id** (UUID or symbolic id) of an ExtrinsicObject representing an XDSDocumentEntry included in the Submission Set.

The association's targetObject attribute shall contain the UUID of an ExtrinsicObject (XDSDocumentEntry) already in the registry.

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

Verify the ExtrinsicObject pointed to by the Association's targetObject attribute is present in the registry and has status of `Approved`. The error `XDSReplaceFailed` shall be thrown 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 ebRIM requests ApproveObjectsRequest and DeprecateObjectsRequest are available to do this.

If the Document being replaced is a member of one or more Folders, generate HasMember Associations connecting the replacement Document with each of the Folders holding the original Document. This makes the replacement Document a member of all Folders where the original Document is a member.

Validate patientIDs in Folders - The adaptor shall verify that all documents in a folder are for the same patient. Specifically, verify that the patientId attribute of the folder matches the patientId attribute of each document in the folder. This comparison shall take into consideration patient identity merges as described in section 3.8.4.2.4 of the Technical Framework.

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

Maintain Folder attribute 'lastUpdateTime' - The XDS Folder attribute lastUpdateTime shall be updated by the adaptor every time a new document is added to an XDS Folder.

Validate patientID on documents being added to a Folder - The patientId attribute of an XDSDocumentEntry object shall match the patientId attribute on any folder that holds it.

Validate coding - The adaptor shall enforce the number of classifications offered against a document. Code lists are allowed to be multiples. Codes are required to be singular.

Accept submissions containing multiple documents – The adapter shall be capable of accepting submissions containing multiple documents.

4.1.11.1 Required Initialization of the XDS Affinity Domain

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 ITI TF-3: 4.1.11. How this information is loaded into the Registry Adaptor or how the Registry Adaptor is implemented is not defined by this profile.

1. List of acceptable mimeTypes for documents indexed by the registry.
2. PIX domain name (Assigning Authority) for XDS Affinity Domain. PatientIds attached to metadata submitted to this registry must come from this PIX Assigning Authority.
3. Acceptable values for all coded attributes represented in the registry by ebXML External Classifications. These include classCode, eventCode, confidentialityCode, healthCareFacilityTypeCode, formatCode for XDS Document and XDSSubmissionSet.code and XDSFolder.codeList.

4.1.11.2 Required Registry Initialization and Schema

A standard ebXML Registry must be initialized with key Classification Schemes and object types to support XDS. An ebXML Registry SubmitObjectsRequest is available to perform this initialization. It includes:

- Classification Schemes that anchor the definition of ExternalIdentifiers

- Additions to the ObjectType ClassificationScheme that introduces a general XDS ClassificationNode that anchors these additions. The usable new ClassificationNodes are: XDSDocumentEntry, XDSDocumentEntryStub, XDSFolder, and XDSSubmissionSet. XDSDocumentEntry and XDSDocumentEntryStub are used as new objectTypes for use in an ExtrinsicObject to create XDS specific object types. XDSFolder and XDSSubmissionSet are used to classify RegistryPackage objects to label them as XDS Folders or XDS SubmissionSets.
- External Classification Schemes to support attribute coding.

This initialization includes the assignment of UUIDs to these definitions. These pre-assigned UUIDs shall be used when implementing XDS.

This initialization is available online on the IHE FTP site, see ITI TF-2x: Appendix W, as ITI/packages/XDSb.Registry.Initialization.zip

4.1.12 General Metadata Issues

This section documents ebXML Registry issues that are confusing, under documented, or are in conflict between various versions of the registry specification.

4.1.12.1 Association Type naming

XDS requires that Association names be specified as text names with proper namespaces and not UUIDs. This is consistent with version 3.0 of ebRIM. XDS requires the use of the following standard Associations:

HasMember – for linking RegistryPackage objects to their contents

In addition, XDS defines a collection of Association types defined in ITI TF-3: 4.1.6 Document Relationships and Associations.

4.1.12.2 Assigning Codes to Documents

Many attributes of XDSDocumentEntry, XDSSubmissionSet, and XDSFolder (Tables 4.1-5, 4.1-6, and 4.1-7) are coded attributes defined as ebRIM Classifications. Three details are required to describe a coded value:

1. The value of the code
2. A display name used by Document Source to represent the code (raw codes are not human-friendly)
3. The name of the coding scheme that the code comes from.

These three values combine to define a single coded element.

As described in ebXML Registry metadata, a coded attribute looks like:


```
<!--++++++++++++++++++++++++++++++++++++++++++++++++++
  --
  --       XdsDocumentEntry.classCode
  --
++++++++++++++++++++++++++++++++++++++++++++++++++ -->
<rim:Classification
   classificationScheme=
     "urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"
   classifiedObject="theDocument"
   id=”ID_067"
   objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Classification"
   nodeRepresentation="My Class Code">

   <!-- ++++++++++++++++++++++++++++++++++++++++++++++++
     XdsDocumentEntry.classCodeDisplayName
   ++++++++++++++++++++++++++++++++++++++++++++++++ -->
   <rim:Name>
      <rim:LocalizedString value="Display Name for My Class Code"/>
   </rim:Name>

   <!-- ++++++++++++++++++++++++++++++++++++++++++++++++
     Coding scheme for classCode
   ++++++++++++++++++++++++++++++++++++++++++++++++ -->
   <rim:Slot name="codingScheme">
      <rim:ValueList>
         <rim:Value>Name of the Coding Scheme (LOINC for
example)</rim:Value>
      </rim:ValueList>
   </rim:Slot>
</rim:Classification>
```

A code is constructed as a Classification object. The relevant parts of this classification are:

Classification – this element wraps the definition

classificationScheme attribute – this UUID references a Classification Scheme object already present in the registry. This Classification Scheme object and its UUID are predefined by XDS and serve as the defining ‘type’ for the code.

classifiedObject attribute – this references the object in metadata being classified. This can be specified as a UUID or as a symbolic name as shown in the example above.

nodeRepresentation attribute – this is the value of the code.

Name element - this is the display name for the code.

codingScheme Slot (Value sub-element) - this is the name of the coding scheme.

The XDS Affinity Domain defines the local configuration for each coding scheme. Specifically, it defines:

Name of the coding scheme – which must be used in the codingScheme Slot

Values for the code – one of which must be used in the nodeRepresentation attribute

One or more display Name values for each code value – which should be used in the Name element and should be semantically consistent with the value for the code.

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

Some code types allow multiple values. EventCodeList is an example. These codes contain the letters 'List' in their name. These codes are XML coded identically to the above example with one exception. The entire Classification element may be repeated to specify additional values.

The Registry Adaptor Function is responsible for validating codes against the configuration of the XDS Affinity Domain.

Note: the attribute XDSDocumentEntry.languageCode is not encoded as shown above. See Tables 4.1-5 for details.

4.1.12.3 Formatting of UUIDs

UUIDs shall be formatted according to RFC4122. Furthermore, values 10 through 15 shall be formatted in hexadecimal using lower case 'a'-'f'. An example of a properly formatted UUID is:

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

Registries shall only accept and produce lowercase UUIDs.

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.

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.

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.

4.1.12.4 XML Namespaces

The Register Document Set, Provide and Register Document Set, and Query Registry transactions are SOAP requests/responses containing valid XML. All elements shall be namespace qualified. Namespaces must be present in all elements. All referenced namespaces must be defined within the transmission.

4.1.13 Error Reporting

Registry Services schema (ebRS 3.0) defines the RegistryError element for reporting details of errors or warnings. RegistryError contains two required attributes, errorCode and codeContext. All transactions that carry the RegistryError element shall return these two attributes with each error reported. Codes reported in errorCode shall be taken from Table 4.1-11 when one of those codes is appropriate. All extensions to the list of error codes shall be unique. 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. The attribute codeContext shall contain details of the error condition that may be implementation specific.

The following attributes are required on the RegistryError element when reporting errors or warnings:

- **errorCode** shall be a value taken from Table 4.1-11
- **codeContext** supplies additional detail for the errorCode
- **severity** supplies a coded indication of the severity of the error:
    - urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error
    - urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Warning

The body of all RegistryError elements shall be empty.

The following attributes on the RegistryError element are optional:

- **location** supplies the location of the error: module name and line number or stack trace if appropriate.
- **highestSeverity** – supplies the severity of the most severe error

The value of the status attribute of either the RegistryResponse or AdHocQueryResponse elements shall be taken from the following lists. For Version 3.0 ebRIM/ebRS:

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

Tables 4.1-12 through 4.1-15 control the reporting of errors for transactions that use the ebRS/ebRIM schemas.

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
Submission Set"
      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>
```

Table 4.1-11: Error Codes

Error Code[1]	Discussion	Transaction (See Note 1)
XDSMissingDocument	XDSDocumentEntry exists in metadata with no corresponding attached document	P
XDSMissingDocumentMetadata	MIME package contains MIME part with Content-Id header not found in metadata	P
XDSRegistryNotAvailable	Repository was unable to access the Registry	P
XDSRegistryError XDSRepositoryError	Internal Registry/Repository Error.	P,R, SQ, XGQ P, RS. XGR
XDSRegistryDuplicateUniqueIdInMessage XDSRepositoryDuplicateUniqueIdInMessage	A UniqueId value was found to be used more than once within the submission. Error code indicates where error was detected. CodeContext shall indicate the duplicate UniqueId.	P,R
XDSDuplicateUniqueIdInRegistry	UniqueId received was not unique within the Registry. UniqueId could have been attached to XDSSubmissionSet or XDSFolder. CodeContext shall indicate which and the value of the non-unique uniqueId. This error cannot be thrown for XDSDocumentEntry. See XdsNonIdenticalHash.	P,R P
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
XDSRegistryBusy XDSRepositoryBusy	Too much activity	P,R,SQ, XGQ P, RS, XGR
XDSRegistryOutOfResources XDSRepositoryOutOfResources	Resources are low.	P,R,SQ, XGQ P, RS, XGR
XDSRegistryMetadataError	Error detected in metadata. Actor name indicates where error was detected.	P,R

Error Code[1]	Discussion	Transaction (See Note 1)
XDSRepositoryMetadataError	CodeContext indicates nature of problem.	
XDSTooManyResults		Q,SQ, XGQ
XDSExtraMetadataNotSaved	This warning is returned if extra metadata was present but not saved in the registry.	P,R
XDSUnknownPatientId	Patient ID referenced in metadata is not known to the Registry actor via the Patient Identity Feed or is unknown because of patient identifier merge or other reasons. 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
XDSPatientIdDoesNotMatch	XDS specifies where patient IDs must match between documents, submission sets, and folders. 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
XDSUnknownStoredQuery	The Query ID provided in the request is not recognized.	SQ, XGQ
XDSStoredQueryMissingParam	A required parameter to a stored query is missing.	SQ, XGQ
XDSStoredQueryParamNumber	A parameter which only accepts a single value is coded with multiple values	SQ, XGQ
XDSRegistryDeprecatedDocumentError	The Register transaction was rejected because it submitted an Association referencing a deprecated document.	P,R
XDSUnknownRepositoryId	The repositoryUniqueId value could not be resolved to a valid document repository or the value does not match the repositoryUniqueId of the Document Repository	RS, XGR
XDSDocumentUniqueIdError	The document associated with the DocumentUniqueId is not available. This could be because the document is not available to the Document Repository, the requestor is not authorized to access that document or the document is no longer available.	RS, XGR
XDSResultNotSinglePatient	This error signals that the single Stored Query would have returned metadata assigned to multiple Patient IDs	SQ, RS
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

Error Code[1]	Discussion	Transaction (See Note 1)
ParitalTransformNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Transform semantics.	P
ParitalAppendContentNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Append semantics.	P
ParitalTransformReplaceNotProcessed	An XDR Document Recipient did not process some part of the content. Specifically the parts not processed are Transform and Replace semantics.	P
DocumentQueued	An XDR Document Recipient queued the document for future manual matching to a patient.	P
XDSUnknownCommunity	A value for the homeCommunityId is not recognized	SQ, XGQ, RS, XGR
XDSMissingHomeCommunityId	A value for the homeCommunityId is required and has not been specified	SQ, XGQ, RS, XGR
XDSUnavailableCommunity	A community which would have been contacted was not available . See Note 2.	SQ, RS

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

Note 1:

P = Provide and Register, Provide and Register-b

R = Register, Register-b

Q = Query

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:

1. A Cross Gateway Query or Cross Gateway Retrieve fails because the community identified by a homeCommunityId could not be contacted.
2. 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.

The following tables explain the meaning of the status attribute in responses from the Registry or Repository.

In the following tables, the values shown in the RegistryResponse Status and AdhocQueryResponse Status columns shall be prefixed by the namespace

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

or

- urn:ihe:iti:2007:ResponseStatusType: (for PartialSuccess)

when used with ebRS 3.0.

Table 4.1-12: Provide and Register Document Set-b Responses

RegistryResponse status	RegistryErrorList element	Result
Success	May be present. If present will contain one or more RegistryError elements with warning severity, none with error severity	All metadata and documents were successfully registered
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 4.1-13: Register Document Set-b Responses

RegistryResponse status	RegistryErrorList element	Result
Success	May be present. If present will contain one or more RegistryError elements with warning severity, none with error severity	All metadata was successfully registered
Failure	Present, contains one or more RegistryError elements. At least one has error severity, others may have warning severity.	Metadata not stored

Table 4.1-15: Stored Query and Cross Gateway Query Responses

AdhocQueryResponse status	RegistryErrorList element	Result
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.
PartialSuccess	Present, contains one or more RegistryError elements. At least one has error severity, others may have warning severity.	Results shall be returned. Results may contain zero or more entries.

AdhocQueryResponse status	RegistryErrorList element	Result
Failure	Present, contains one or more RegistryError elements. At least one has error severity, others may have warning severity.	Results not returned

Table 4.1-16: Retrieve Document Set and Cross Gateway Retrieve Responses

Registry Response status	RegistryErrorList element	Result
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
PartialSuccess	Present, contains one or more RegistryError elements. At least one has error severity, others may have warning severity.	Some documents were successfully retrieved
Failure	Present, contains one or more RegistryError elements. At least one has error severity, others may have warning severity.	No documents were successfully retrieved

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

4.1.14 Extra Metadata Elements

XDS transactions may contain metadata not defined in the XDS Profile. 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.
2. These Slots may be attached to XDSSubmissionSet, XDSDocumentEntry, XDSFolder, or Association objects.
3. If the Document Registry actor is not capable of storing extra metadata and extra metadata is provided in a Register Document Set transaction, it shall return a warning with an error code of XDSExtraMetadataNotSaved. The XDS defined metadata shall be saved.
4. Document Consumer actors shall ignore extra metadata elements they do not understand.
5. If a Document Registry actor accepts extra metadata Slots (no warning on submission) then it shall return these Slots in query results.
6. 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

7. Note that ebRIM requires that the name of a Slot be unique within the containing object (Document Entry, Submission Set, Folder, Association).

4.2 Character String Comparisons

All character string comparisons shall be done in conformance with the rules of the Unicode standard (http://www.unicode.org/versions/latest/) using the normalized form C defined in Unicode Technical Report 15 (http://unicode.org/reports/tr15).

Note: Latin alphabet case-insensitive NFC matching corresponds to byte string matching. The primary impact of this is for non-Latin alphabets. They need to be converted into normalized form before comparison. The TR 15 approach is consistent with the working documents of W3C, although W3C has not yet issued a balloted recommendation that Unicode normalized form C be used. See http://www.w3.org/TR/WD-charreq, http://www.w3.org/International/charlint/, and the current W3C draft (http://www.w3.org/TR/charmod-norm).

See the following references for more details:

Unicode Technical Report #15, Unicode Consortium UAX #15: Unicode Normalization Forms (http://www.unicode.org/reports/tr15/)

Unicode Standard Unicode Consortium. The Unicode Standard, (http://www.unicode.org/versions/latest/)

4.3 XDS Metadata Vocabulary

4.3.1 Metadata UUIDs

The UUIDs in the following sections shall be used in constructing and interpreting XDS metadata. The assigning authority “IHE XDS Metadata” shall be used for these codes.

4.3.1.1 Submission Set Object

UUID	Use/meaning
urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd	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

4.3.1.2 Document Entry Object

UUID	Use/meaning
urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1	XDSDocumentEntry 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

4.3.1.3 Folder Object

UUID	Use/meaning
urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2	XDSFolder 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

5 IHE Content Specifications

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

5.1 Basic Patient Privacy Consents Module

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

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

5.1.1 References

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

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

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

5.1.2.1 XDS Metadata

5.1.2.1.1 XDS DocumentEntry Metadata

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

5.1.2.1.1.1 XDSDocumentEntry.classCode

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

For backward compatibility historic documents may be registered with:

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

5.1.2.1.1.2 XDSDocumentEntry.eventCodeList

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

- /ClinicalDocument/documentationOf/serviceEvent[templateId/@root='1.3.6.1.4.1.19376.1.5.3.1.2.6']/code/@code

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

5.1.2.1.1.3 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 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

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

5.1.2.3 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 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.


```
<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'
    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' />
  <languageCode code='en-US'/>
     :
  <component><structuredBody>

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

Figure 5.1.2.3-1: Sample Consent to Share Information Document

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

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

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

5.1.2.3.1 Patient Privacy Acknowledgement Service Events 1.3.6.1.4.1.19376.1.5.3.1.2.6

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


```
<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 5.1.2.3-2: Patient Privacy Acknowledgement Service Events Example

5.1.2.3.2 <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.

5.1.2.3.3 <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.

5.1.2.3.4 <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..

5.1.2.3.5 <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.

5.1.2.3.6 <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.

5.1.2.3.7 <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.

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.

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 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

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 the additional requirements stated in ITI TF-3: 5.1.2.3.

5.1.3.4 Conformance

See 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 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 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.

- *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.

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)
- 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-oriented documents. PDF/A attempts to maximize:

- Device independence
- Self-containment
- Self-documentation

The constraints imposed by PDF/A include:

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

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

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

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

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

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

5.2.2 XDS Metadata

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

5.2.2.1 XDS DocumentEntry Metadata

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

5.2.2.1.1 XDSDocumentEntry.formatCode

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 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 particular document it may be the case that both the PDF scanned content and somewhat equivalent plaintext need to be wrapped and submitted. Each document would correspond to separate XDSDocumentEntries linked via an XFRM Association that indicates one document is a transform of the other. These can be submitted in a single submission set, or in separate ones. Other specific examples may exist and this profile does not preclude the notion of a parent document for these cases.

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

5.2.2.3 XDS Folder Metadata

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

5.2.3 Specification

HL7 CDA R2 header element	CDA as constrained by XDS-SD	Section Number of Extended Discussion	Source Type	Source / Value
ClinicalDocument/typeId	R	5.2.3.1	FM	Fixed, per CDA R2 version in use.
ClinicalDocument/templateId	R	5.2.3.1	FM	Fixed, per this specification
ClinicalDocument/id	R	5.2.3.1	DS	Computable.
ClinicalDocument/code	R	5.2.3.1	O / FM	Entered by operator or appropriately fixed for scanned content
ClinicalDocument/title	R2	5.2.3.1	SA / O	Entered by operator, or possibly can be taken from the scanned content.
ClinicalDocument/confidentiality Code	R	5.2.3.1	O	Assigned by the operator
ClinicalDocument/effectiveTime	R	5.2.3.1	DS	Computed. This is the scan time.
ClinicalDocument/languageCode	R	5.2.3.1	O	Entered by operator
ClinicalDocument/recordTarget	R	5.2.3.2	SA / O	Taken from scanned content, supplemented by operator.
ClinicalDocument/author/assigned Author/assignedPerson	R2	5.2.3.3	SA / O	Taken from scanned content, supplemented by operator. This is the original author.
ClinicalDocument/author/assigned Author/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 `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 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 this code are dictated by the CDA R2 documentation, but are permissible to extend to fit the particular use case. Attributes code@code and code@codeSystem shall be present.
- The ClinicalDocument/title shall be present if known.
- The ClinicalDocument/effectiveTime shall denote the time at which the original content was scanned. At a minimum, the time shall be precise to the day and shall include the time zone offset from GMT.
- The ClinicalDocument/confidentialityCode shall be assigned by the operator in accordance with the scanning facility policy. The notion or level of confidentiality in the header may not be the same as that in the Affinity Domain, but in certain cases could be used to derive a confidentiality value among those specified by the Affinity Domain. Attributes confidentialityCode@code and confidentialityCode@codeSystem shall be present.
- The ClinicalDocument/languageCode, in accordance with the HL7 CDA R2 documentation, shall denote the language used in the character data of the wrapper CDA header. If the scanned content, when rendered, is in a language different than that of the header, the language context of the CDA will be overwritten at the body level (see 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"/>
```


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 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 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.
- 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>
```


5.2.3.3 ClinicalDocument/author (original)

This ClinicalDocument/author element represents the author of the original content. It additionally can encode the original author's institution in the subelement representedOrganization. Information regarding the original author and his/her institution shall be included, if it is known. In many cases this will have to be supplied by the operator. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- The ClinicalDocument/author/templateId element shall be present. The root attribute shall contain the oid, '1.3.6.1.4.1.19376.1.2.20.1', to indicate this is the original author.
- The ClinicalDocument/author/time represents the day and time of the authoring of the original content. This value is not restricted beyond statements made in the HL7 CDA R2 documentation.
- The ClinicalDocument/author/assignedAuthor/id element if known shall include both the root and the extension attributes. Refer to PCC TF-2: 4.1.1 for more details.
- The ClinicalDocument/author/assignedAuthor/representedOrganization/id element if known shall include both the root and the extension attributes. Refer to PCC TF-2: 4.1.1for 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>
```

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 shall contain the oid, '1.3.6.1.4.1.19376.1.2.20.2', to indicate this author is the scanning device and software.
- The ClinicalDocument/author/time shall denote the time at which the original content was scanned. This value shall be equal to that of ClinicalDocument/effectiveTime. At a minimum, the time shall be precise to the day and shall include the time zone offset from GMT.
- The ClinicalDocument/author/assignedAuthor/id element shall be at least the root oid of the scanning device.
- The ClinicalDocument/author/assignedAuthor/assignedAuthoringDevice/code element shall be present. The values set here are taken from appropriate DICOM vocabulary. The value of code@codeSystem shall be set to "1.2.840.10008.2.16.4". The value of code@code shall be set to "CAPTURE" for PDF scanned content and "WSD" for plaintext. The value of code@displayName shall be set to "Image Capture" for PDF scanned content and "Workstation" for plaintext.
- The ClinicalDocument/author/assignedAuthor/assignedAuthoringDevice/manufacturerModelName element shall be present. The mixed content shall contain string information that specifies the scanner product name and model number. From this information, features like bit depth and resolution can be inferred. In the case of virtually scanned documents (for example, print to PDF), the manufactureModelName referenced here refers to the 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 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>
```

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.
- 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.
- 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>
```

1215

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
1220 content. In most cases this will be the scanning facility. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

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

5.2.3.7 ClinicalDocument/legalAuthenticator

The `ClinicalDocument/legalAuthenticator` may be present and its context is left up to the scanning facility to refine in accordance with local policies. All subelements retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- The ClinicalDocument/legalAuthenticator/assignedEntity/id element if known shall include both the root and the extension 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>
```


5.2.3.8 ClinicalDocument/documentationOf

This `ClinicalDocument/documentationOf` element is used to encode the date/time range of the original content. If the original content is representative of a single point in time then the endpoints of the date/time range shall be the same. Information regarding this date/time range shall be included, if it is known. In many cases this will have to be supplied by the operator. This profile does not restrict the `documentationOf` element beyond statements made in the HL7 CDA R2 documentation.

Example:

```
<documentationOf>
  <serviceEvent >
    <effectiveTime>
      <low value="19800127"/>
      <high value="19990522"/>
    </effectiveTime>
  </serviceEvent>
</documentationOf>
```

5.2.3.9 ClinicalDocument/component/nonXMLBody

This ClinicalDocument/component/nonXMLBody element shall be present and used to wrap the scanned content. The nonXMLBody element is guaranteed to be unique; thus the x-path to recover the scanned content is essentially fixed. All subelements of the nonXMLBody retain their original definition as defined by the HL7 CDA R2 specification, unless noted below.

- If the human-readable language of the scanned content is different than that of the wrapper (specified in ClinicalDocument/languageCode), then ClinicalDocument/component/nonXMLBody/languageCode shall be present. Attribute code@code shall be present. Attribute code@codeSystem shall be IETF (Internet Engineering Task Force) RFC 3066 in accordance with the HL7 CDA R2 documentation.
- The ClinicalDocument/component/nonXMLBody/text element shall be present and encoded using xs:base64Binary encoding and shall contain the scanned content.
    - ClinicalDocument/component/nonXMLBody/text@mediaType shall be "application/pdf" for PDF, or "text/plain" for plaintext.
    - When the character encoding of text/plain content is other than UTF-8, the @mediaType shall be text/plain;charset=XXX, where XXX is the name of the encoding as specified by RFC-2046.
    - 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
      ...
      SW5mbyAyIDAgUgovSUQgWzxGNENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
      RjRDQzdBRUI1NEIzNkZCMjgzQzVDMzI0NzlBRDI4M0Y+XQo+PgpzdGFydHhyZWYKMzAx
      MgolJUVPRgo=
      </text>
    </nonXMLBody>
  </component>
</ClinicalDocument>
```

Example (PDF scanned content is in a *different* language than the wrapper):

```
  <component>
    <nonXMLBody>
      <languageCode code="zh-CN"/>
      <text mediaType="application/pdf" representation="B64">
      JVBERi0xLjMKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9GbGF0
      ZURlY29kZT4+CnN0cmVhbQp4nGWPMWsDMQyFd/8KjfJwqmVbkr0GQqFbg7fQoSRNWuhB
      Q/4/1L67TEEYme+9J1s3CMQQRm39NLuXg8H17gK89nN1N8eLAbZ2mmHXuql2QDVUhnZx
      a5iBcyQtoMIUM7TZHbH5KZEVDgm//SSUswbFHx/JzBLeu5yYxOIzE8bPcRWqdaGDmcZO
      BWc/9bfUNOPfOte44O9jxtcIKskqp0JZouJ5deYqeBn58ZmKtIU+2ptjqWQRJpGyrHDu
      K7CXIe2be+/1DzXQP+RlbmRzdHJlYW0KZW5kb2JqCjYgMCBvYmoKMjAxCmVuZG9iago0
      ...
      SW5mbyAyIDAgUgovSUQgWzxGNENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
      RjRDQzdBRUI1NEIzNkZCMjgzQzVDMzI0NzlBRDI4M0Y+XQo+PgpzdGFydHhyZWYKMzAx
      MgolJUVPRgo=
      </text>
    </nonXMLBody>
  </component>
</ClinicalDocument>
```


5.2.4 Complete Example (Wrapped PDF)

```
<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"/>
  <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"/>
  <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>
  <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>
    <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>
   <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>
    <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>
  <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>
  <documentationOf>
    <serviceEvent >
      <effectiveTime>
        <low value="19800127"/>
        <high value="19990522"/>
      </effectiveTime>
    </serviceEvent>
  </documentationOf>
  <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
      ...
      SW5mbyAyIDAgUgovSUQgWzxGNENDN0FFQjU0QjM2RkIyODNDNUMzMjQ3OUFEMjgzRj48
      RjRDQzdBRUI1NEIzNkZCMjgzQzVDMzI0NzlBRDI4M0Y+XQo+PgpzdGFydHhyZWYKMzAx
      MgolJUVPRgo=
      </text>
    </nonXMLBody>
  </component>
</ClinicalDocument>
```

