

Integrating the Healthcare Enterprise



5

**IHE IT Infrastructure
Technical Framework Supplement**

10

**Restricted Metadata Update
(RMU)**

15

Revision 1.2 – Trial Implementation

20

Date: August 28, 2020
Author: IHE ITI Technical Committee
Email: iti@ihe.net

25

Please verify you have the most recent version of this document. See [here](#) for Trial Implementation and Final Text versions and [here](#) for Public Comment versions.

Foreword

This is a supplement to the IHE IT Infrastructure Technical Framework V17.0. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is published on August 28, 2020 for trial implementation and may be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the IT Infrastructure Technical Framework. Comments are invited and can be submitted at

http://www.ihe.net/ITI_Public_Comments.

This supplement describes changes to the existing technical framework documents.

“Boxed” instructions like the sample below indicate to the Volume Editor how to integrate the relevant section(s) into the relevant Technical Framework volume.

<i>Amend Section X.X by the following:</i>
--

Where the amendment adds text, make the added text **bold underline**. Where the amendment removes text, make the removed text **~~bold strikethrough~~**. When entire new sections are added, introduce with editor’s instructions to “add new text” or similar, which for readability are not bolded or underlined.

General information about IHE can be found at www.ihe.net.

Information about the IHE IT Infrastructure domain can be found at ihe.net/IHE_Domains.

Information about the organization of IHE Technical Frameworks and Supplements and the process used to create them can be found at http://ihe.net/IHE_Process and <http://ihe.net/Profiles>.

The current version of the IT Infrastructure Technical Framework can be found at http://ihe.net/Technical_Frameworks.

CONTENTS

	Introduction to this Supplement.....	5
	Open Issues and Questions	5
65	Closed Issues	5
	General Introduction and Shared Appendices	9
	Appendix A – Actor Summary Definitions	9
	Appendix B – Transaction Summary Definitions.....	9
	Appendix D – Glossary	9
70	Volume 1 – Profiles	10
	48 Restricted Metadata Update (RMU) Profile	10
	48.1 RMU Actors, Transactions, and Content Modules	10
	48.1.1 Actor Descriptions and Actor Profile Requirements.....	11
	48.1.1.1 Update Initiator.....	11
75	48.1.1.2 Update Responder	11
	48.2 RMU Actor Options	11
	48.2.1 Forward Update Option.....	11
	48.2.2 XCA Persistence Option	12
	48.2.3 XDS Persistence Option	12
80	48.2.4 XDS Version Persistence Option	12
	48.3 Required Actor Groupings	12
	48.4 Overview	13
	48.4.1 Concepts	13
	48.4.2 Use Cases	14
85	48.4.2.1 Use Case #1 Restricting Document Access.....	14
	48.4.2.1.1 Use Case Description.....	14
	48.4.2.1.2 Process Flow	14
	48.4.2.2 Use Case #2: Allowing Document Access	16
	48.4.2.2.1 Use Case Description.....	16
90	48.4.2.2.2 Process Flow	17
	48.5 Security Considerations	19
	48.6 Cross Profile Considerations.....	20
	48.6.1 Grouping an XDS Document Consumer.....	20
	48.6.2 Grouping an XCA Initiating Gateway	21
95	48.6.3 Grouping within an XCA/XDS Affinity Domain Option Environment	21
	Volume 2c – Transactions	23
	3.92 Restricted Update Document Set	23
	3.92.1 Scope	23
	3.92.2 Actor Roles.....	23
100	3.92.3 Referenced Standard	23
	3.92.4 Messages	24
	3.92.4.1 Restricted Update Document Set Request Message.....	24
	3.92.4.1.1 Trigger Events	24

	3.92.4.1.2 Message Semantics	24
105	3.92.4.1.2.1 Content	26
	3.92.4.1.2.2 Metadata Annotations	26
	3.92.4.1.2.2.1 PreviousVersion	26
	3.92.4.1.2.2.2 AssociationPropagation	27
	3.92.4.1.3 Expected Actions	28
110	3.92.4.1.3.1 Forward Update Option	28
	3.92.4.1.3.2 XCA Persistence Option	29
	3.92.4.1.3.3 XDS Persistence Option	29
	3.92.4.1.3.4 Version Persistence Option	29
	3.92.4.1.3.4.1 Storage Requirements	29
115	3.92.4.1.3.5 Request Validation	30
	3.92.4.2 Restricted Update Document Set Response Message	31
	3.92.4.2.1 Trigger Events	31
	3.92.4.2.2 Message Semantics	31
	3.92.4.2.3 Expected Actions	32
120	3.92.5 Security Considerations	32
	3.92.5.1.4.1 Audit Record Considerations	32
	3.92.5.1.4.1.1 Update Initiator Audit Message	33
	3.92.5.1.4.1.2 Update Responder Audit Message	34
	Volume 3 – Cross-Transaction and Content Specifications.....	37
125	4.1 Abstract Metadata Model	37
	4.1.5 Metadata Object Versioning Semantics	37
	4.1.5.1 Association Propagation	39
	4.2.3 Metadata Attributes	39
	4.2.3.2 Document Metadata Attribute Definition	39
130	4.2.3.2.30 DocumentEntry.documentAvailability	40
	4.2.3.2.31 DocumentEntry.logicalID	40
	4.2.3.2.32 DocumentEntry.version	41
	4.2.4 Success and Error Reporting	42
	4.2.4.1 RegistryError Element	42
135	4.3 Additional Document Sharing Requirements	44
	4.3.1 Submission Metadata Attribute Optionality	44
	4.3.2 Requirements on Query Type Transactions	46
	4.3.2.1 Query Type Metadata Attribute Optionality	46
140		

Introduction to this Supplement

Attention Readers:

Material originally released in the XDS Metadata Update Supplement is used for the development of a new transaction in this profile that includes the attributes and semantics used for versioning metadata.

The ITI Technical Committee determined that it was appropriate to reuse the work already developed as it met the criteria for the proposed use case(s).

By only introducing the Update Document Set [ITI-57] transaction's Update DocumentEntry Metadata operation for this RMU Profile, complexity is reduced for implementers that do not need full administrative capabilities included in the XDS Metadata Update Supplement. This profile does not include support for updating Submission Set, Folder, and Association objects.

While the RMU Profile can make use of versioning semantics for the new transaction, this does not preclude using it in environments that cannot store and maintain full metadata history.

As this RMU Profile supports updating only DocumentEntry metadata objects, the receiving actor will be expected to perform Association Propagation to maintain a consistent view of the metadata. How the implementer interprets this requirement depends on the actor option selected.

The ITI Technical Committee intends on maintaining compatibility with the feature set between both this profile and the Metadata Update Supplement.

The Restricted Metadata Update (RMU) Profile provides a mechanism for changing Document Sharing metadata both within and across community boundaries in a controlled manner. Certain metadata attributes have been restricted from modification to allow for predictable and safe use in a wide range of operating environments. Communities can use this Profile as part of meeting legislative requirements for patient control of their distributed medical information.

Open Issues and Questions

None.

Closed Issues

RMU_001: Do we need to define a new or extra metadata attribute that indicates the DocumentEntry object was modified by an actor from another community? Should this be left for implementers to decide?

Resolution: Determined that the Submission Set object already provides the metadata attribute, sourceId, which can be used for determining the originating source of the request. No additional metadata attributes needed.

RMU_002: Are DocumentEntry objects the only objects which shall be modifiable by this profile?

Resolution (F2F, 14-Feb-2018): yes, because use cases relate only to DocumentEntry objects.

180 **RMU_003: Do we need a white list/a black list of DocumentEntry attributes which can be modified by this profile?**

Resolution (F2F, 14-Feb-2018): yes, white list is created.

RMU_004: Shall DocumentEntry attributes creationTime, serviceStartTime, serviceStopTime be modifiable?

185 Resolution (F2F, 14-Feb-2018): yes. These attributes may have a special meaning e.g., in the BPPC context, but in this case the receiving actor can reject the change.

RMU_005: Shall DocumentEntry attributes patientId, sourcePatientId, sourcePatientInfo be modifiable?

Resolution (F2F, 14-Feb-2018):

- patientId is not modifiable because this is a structural attribute.
- 190 • sourcePatientId is not modifiable because it contains the patient ID which was valid at the submission time.
- sourcePatientInfo is modifiable because its value does not influence anything.

RMU_006: Do we need a new actor “RMU Document Administrator” in the Initiating Community?

195 Resolution (TCON, 04-April-2018): No. The Update Initiator provides similar capabilities.

RMU_007: Do we need the Asynchronous WS Option?

Resolution: Not at this time but could be added in the future. At this time, most use cases indicate that synchronous web services would be used.

RMU_008: Do we need two transactions or is one enough?

200 Resolution (F2F, 14-Feb-2018): Because of their similarity, the intra and cross-community transactions have been coalesced into a single transaction being differentiated by the use of the homeCommunityId.

RMU_009: unintentionally skipped

205 **RMU_010: Should the Update Initiator be grouped with an XDS Document Consumer with Metadata Update Option instead of embedding the transaction within the RMU actor? This could allow for the use of X-Community Retrieve Document set, as well.**

Resolution: While likely needed for the successful execution of the RMU, it was determined that the formal grouping of an XDS Document Consumer distracted from the central functionality being added by the profile. Instead, it was added as Cross Profile Consideration.

210 **RMU_011: Is UpdateAvailabilityStatus operation in or out?**

Resolution: Out – It was determined that to keep implementation simple, this operation would not be introduced into the transaction.

RMU_012: Should the homeCommunityId be set in @home attribute, RequestList slot, and/or SOAP header?

215 *Resolution: The decision was made to use the @home attribute following the pattern established by XCA. It was felt that this would allow the Update Initiator to more easily compose the request message using the previously obtained response. The XCDR approach was discussed, but participants agreed that verifying the homeCommunityId in both the SOAP header and body was not desirable as currently written.*

220 **RMU_013: unintentionally skipped**

RMU_014: Should a workflow that uses Remove Metadata and Document (RMD) Profile be included?

Resolution: Out of scope. A new work item maybe submitted for consideration next year.

RMU_015: Should the XCA with XDS Affinity Domain Option be included in the profile?

225 *Resolution: No-See RMU_010.*

RMU_016: Should Restricted Metadata Update be considered a replacement for the Metadata Update Supplement?

230 *Resolution: The Restricted Metadata Update (RMU) Profile currently supports only the updating of metadata for DocumentEntry objects. The primary goal of this profile was to enable cross-community support for metadata updates, but still relies on nearly all of its mechanics previously established in the Metadata Update Supplement. The mechanics apply whether used within a community or across multiple communities.*

235 *As a result, this profile was written in way so that either this profile or the supplement could reach final text independent of one another. Thus, it should be considered as a refinement of the existing Update Document Set transaction, but not as a direct replacement.*

RMU_017: Can Restricted Metadata Update be used for updates within a "federated" community (e.g.: Update Responder forward the request to all known holders of the data)?

Resolution: No – The remote community must be identified prior to issuing the Restricted Update Metadata transaction.

240 **RMU_018: Should this profile include any discussion or support for the DocumentAvailability attribute?**

Resolution: Determined that this profile could include this attribute being carried forward from the Metadata Update Supplement. It was agreed this attribute, though, cannot be change using [ITI-92].

245

RMU_019: How should we incorporate the requirements for the XDS Option, Document Metadata Update Option? If brought into this profile, this could introduce a required grouping with XDS for the RMU actors.

250 Resolution: No – The Technical Committee did not want to introduce a mandatory requirement for implementers to use full metadata versioning as described in the XDS Metadata Update Supplement. Instead, it was added as an option to allow implementers to declare this capability.

RMU_020: Should this profile allow the Update Initiator to set the value for Association Propagation to "no"? If not, should this profile address this concept directly? Is it sufficient to leave the requirements as expected actions on the Update Responder?

255 Resolution: In order to maintain consistency with the XDS Metadata Update supplement, the Association Propagation attribute will remain in this Profile, but be forced to "yes".

RMU_021: Should we bring forward ITI TF-1 Section 10.4.14 and ITI TF-2a Section 3.18.4.1.2.3.5.1 from XDS Metadata Update?

260 Resolution: It is the Technical Committee's intent to eventually bring the underlying content together for both profiles. It will not be done for trial implementation.

RMU_022: Should the metadata attributes, hash and size, be restricted? Do we have the list of allowed and restricted attributes correct?

Resolution: At this time, we have allowed modification of the hash and size attributes. Affinity Domains have the ability to restrict these parameters to meet their requirements.

265 **RMU_023: Should this profile allow metadata modifications for On-Demand Documents?**

Resolution: It was agreed that metadata for an On-Demand Document could be updated. This is consistent with Metadata Update supplement. A question was posed as whether or not the metadata of an ODD snapshot persisted in a Document Repository requires separate treatment as the storage of the physical document is for archival purposes only.

270 The ITI Technical Committee felt that the associated metadata may still be updated like any other Stable Document Entry object. If an individual community's use case requires that the metadata cannot be updated after an original submission, the community could create a local policy restricting any request made using the Restricted Update Document Set [ITI-92] transaction.

275

General Introduction and Shared Appendices

The [IHE Technical Framework General Introduction and Shared Appendices](#) are components shared by all of the IHE domain technical frameworks. Each technical framework volume contains links to these documents where appropriate.

*Update the following appendices to the General Introduction as indicated below. Note that these are **not** appendices to Volume 1.*

Appendix A – Actor Summary Definitions

Add the following actors to the IHE Technical Frameworks General Introduction Appendix A:

Actor Name	Definition
Update Initiator	The Update Initiator creates requests to update previously stored metadata.
Update Responder	The Update Responder processes requests to update previously stored metadata.

Appendix B – Transaction Summary Definitions

Add the following transactions to the IHE Technical Frameworks General Introduction Appendix B:

Transaction Name and Number	Definition
Restricted Update Document Set [ITI-92]	The Restricted Update Document Set is used to request updates to a restricted set of metadata attributes for a document in a community.

Appendix D – Glossary

*Add the following **new** glossary terms to the IHE Technical Frameworks General Introduction Appendix D.*

Glossary Term	Definition
Metadata Object Instance	An object representing a single version of a logical metadata object.
Logical Metadata Object	A collection of metadata object instances describing a single object. Each instance represents a different version of the object's metadata.
Metadata Annotation	An XML element used in a submission to further describe or trigger additional behavior by a receiving actor for a submitted metadata object.

Volume 1 – Profiles

Add new Section 48

48 Restricted Metadata Update (RMU) Profile

300 The Restricted Metadata Update (RMU) Profile provides a mechanism for changing Document
Sharing metadata both within and across community boundaries in a controlled manner. RMU
enables modification of a restricted set of Document Sharing metadata attributes (see Section
48.4.1) to allow for predictable and safe use in a wide range of operating environments.
305 Communities can use this Profile as part of meeting legislative requirements for patient control
of their distributed medical information

48.1 RMU Actors, Transactions, and Content Modules

Figures 48.1-1 show the actors directly involved in the RMU Profile and the relevant
transactions between them. If needed for context, other actors that may be indirectly involved
due to their participation in other related profiles are shown in dotted lines. Actors which have a
310 required grouping are shown in conjoined boxes (see Section 48.3).

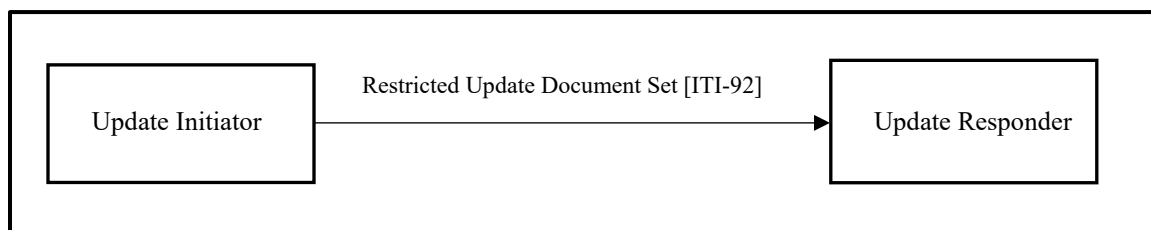


Figure 48.1-1: RMU Actor Diagram

315 Table 48.1-1 lists the transactions for each actor directly involved in the RMU Profile. To claim
compliance with this profile, an actor shall support all required transactions (labeled “R”) and
may support the optional transactions (labeled “O”).

Table 48.1-1: RMU Profile - Actors and Transactions

Actors	Transactions	Initiator or Responder	Optionality	Reference
Update Initiator	Restricted Update Document Set [ITI-92]	Initiator	R	ITI TF-2c: 3.92
Update Responder	Restricted Update Document Set [ITI-92]	Responder	R	ITI TF-2c: 3.92

48.1.1 Actor Descriptions and Actor Profile Requirements

Most requirements are documented in ITI TF-2 Transactions. This section documents any additional requirements on profile's actors.

48.1.1.1 Update Initiator

The Update Initiator creates requests for updating metadata previously stored either within or outside the Document Sharing community. This update may have been initiated by an automated mechanism or manual administrative procedure, in accordance with a predetermined agreement or policy.

The Update Initiator will request the update to the most recent version of the Document Sharing metadata using the patient identifier and terminology known to the storing community. This profile does not specify how the Update Initiator obtains the metadata.

The Restricted Update Document Set [ITI-92] transaction is used by the Update Initiator to transmit the update request to the Update Responder for processing. The success or failure of the transaction is returned in the response.

48.1.1.2 Update Responder

The Update Responder receives and responds to a Restricted Update Document Set [ITI-92] transaction from an Update Initiator. If the Update Responder services the target community for the request, the Update Responder will persist the request. Otherwise, the Update Responder may optionally forward the request to destination community for processing.

48.2 RMU Actor Options

Options that may be selected for each actor in this profile, if any, are listed in the Table 48.2-1. Dependencies between options, when applicable, are specified in notes.

Table 48.2-1: RMU - Actors and Options

Actor	Option Name	Reference
Update Initiator	No options defined	--
Update Responder	Forward Update (Note 1)	ITI TF-1: 48.2.1
	XCA Persistence (Note 1)	ITI TF-1: 48.2.2
	XDS Persistence (Note 1)	ITI TF-1: 48.2.3
	XDS Version Persistence (Note 1)	ITI TF-1: 48.2.4

Note 1: The Update Responder shall support at least one of these options. The Update Responder may support for more than one of these options.

48.2.1 Forward Update Option

The Update Responder supporting the Forward Update Option shall be grouped with an Update Initiator. When it receives a Restricted Update Document Set [ITI-92] transaction request for a

remote Document Sharing community, the grouped Update Initiator shall forward the request to the remote community's Update Responder.

48.2.2 XCA Persistence Option

350 An Update Responder that supports the Persistence Option shall be grouped with an XCA Responding Gateway. The Update Responder shall be capable of processing the Restricted Update Document Set [ITI-92] transaction. Accepted metadata updates shall be reflected in subsequent Cross Gateway Query [ITI-38] transactions from the grouped XCA Responding Gateway.

355 48.2.3 XDS Persistence Option

An Update Responder that supports the Persistence Option shall be grouped with an XDS Document Registry. The Update Responder shall be capable of processing the Restricted Update Document Set [ITI-92] transaction. Accepted metadata updates shall be reflected in subsequent Registry Stored Query [ITI-18] transactions from the grouped XDS Document Registry.

360 48.2.4 XDS Version Persistence Option

An Update Responder that supports the XDS Version Persistence Option shall be grouped with an XDS Document Registry. The Update Responder shall be capable of processing the Restricted Update Document Set [ITI-92] transaction request and store updated metadata with full metadata versioning semantics as described in ITI TF-3: 4.1.5 – Metadata Object Versioning Semantics.

365 Accepted metadata updates shall be reflected in subsequent Registry Stored Query [ITI-18] transactions from the grouped XDS Document Registry using versioning semantics described in ITI TF-2a: 3.18.4.1.2.3.5.1 - Compatibility Issues (currently in the XDS Metadata Update Trial Implementation Supplement).

Note: An Update Responder that is grouped with an XDS Document Administrator may satisfy this requirement.

370

48.3 Required Actor Groupings

An actor from this profile (Column 1) shall implement all of the required transactions and/or content modules in this profile *in addition to all* of the requirements for the grouped actor (Column 3).

375 Section 48.5 describes some optional groupings that may be of interest for security considerations and Section 48.6 describes some optional groupings in other related profiles.

Table 48.3-1: RMU - Required Actor Groupings

RMU Actor	Grouping Condition	Actor to be grouped with	Reference
Update Initiator	Required	ATNA / Secure Node or Secure Application	ITI TF-1: 9.1
	Required	CT / Time Client	ITI TF-1: 7.1

RMU Actor	Grouping Condition	Actor to be grouped with	Reference
Update Responder	Required	ATNA / Secure Node or Secure Application	ITI TF-1: 9.1
	Required	CT / Time Client	ITI TF-1: 7.1
	Forward Update Option	RMU / Update Initiator	ITI TF-1: 48.1.1.1
	XCA Persistence Option	XCA / Responding Gateway	ITI TF-1: 18.1
	XDS Persistence Option	XDS / Document Registry	ITI TF-1: 10.1.1.3
	XDS Version Persistence Option	XDS / Document Registry	ITI TF-1: 10.1.1.3

48.4 Overview

380 48.4.1 Concepts

The Restricted Metadata Update (RMU) Profile provides a mechanism for updating Document Sharing metadata both within and across community boundaries in a controlled manner.

385 The RMU Profile can be combined with other IHE profiles, such as XCA and XDS, to build patient longitudinal records that can be maintained both within and across medical communities and national borders.

The profile maintains the stewardship of the patient record in the community where patient record was stored. When patient records are shared across communities, it is expected that the record will contain the patient identifier and coding system values from the community where the record originated.

390 The profile assumes that the latest version of the stored DocumentEntry metadata object(s) will be used as the basis for the update submission.

In order to maintain interoperability among participating communities, certain metadata attributes are restricted from modification as they describe the current state of DocumentEntry object, or the stored physical document.

395 The RMU Profile permits updating the following metadata attributes:

- author
- classCode
- comments
- confidentialityCode
- 400 • creationTime
- eventCodeList
- formatCode

- hash
- healthcareFacilityTypeCode
- 405 • languageCode
- legalAuthenticator
- mimeType
- practiceSettingCode
- referenceIdList
- 410 • serviceStartTime
- serviceStopTime
- size
- sourcePatientInfo
- title
- 415 • typeCode
- URI

Other IHE Profiles, XDS Affinity Domain policies, or community policies may impose additional restrictions on the metadata attributes they allow to be changed. It is expected that a document sharing agreement will be established among domain participants prior to deploying the capabilities in this Profile.

48.4.2 Use Cases

48.4.2.1 Use Case #1 Restricting Document Access

48.4.2.1.1 Use Case Description

A patient believes that certain documents may have been either incorrectly classified or contain confidential information that is accessible by non-physicians within a community.

The patient makes a request through the local Hospital Information System (HIS) patient portal for a list of documents and related metadata stored in the community. Upon review, the patient identifies several records that should be restricted from general view.

On behalf of the patient, the practitioner triggers an action in the local HIS to update the classification of a metadata attribute which restricts these documents from non-physician use.

48.4.2.1.2 Process Flow

The practitioner uses the local HIS to obtain the latest version of the metadata for each document, and then prepares the updated document metadata, reflecting the requested changes.

The HIS uses the value of the homeCommunityId metadata attribute to determine where the documents are stored. When the community is the same community where the documents are stored, this value may not be populated.

Having prepared the metadata, the HIS, as an Update Initiator, issues a metadata update request to Update Responder to process the request against the locally stored metadata.

The Update Responder returns a response to the Update Initiator indicating the success or failure of the request.

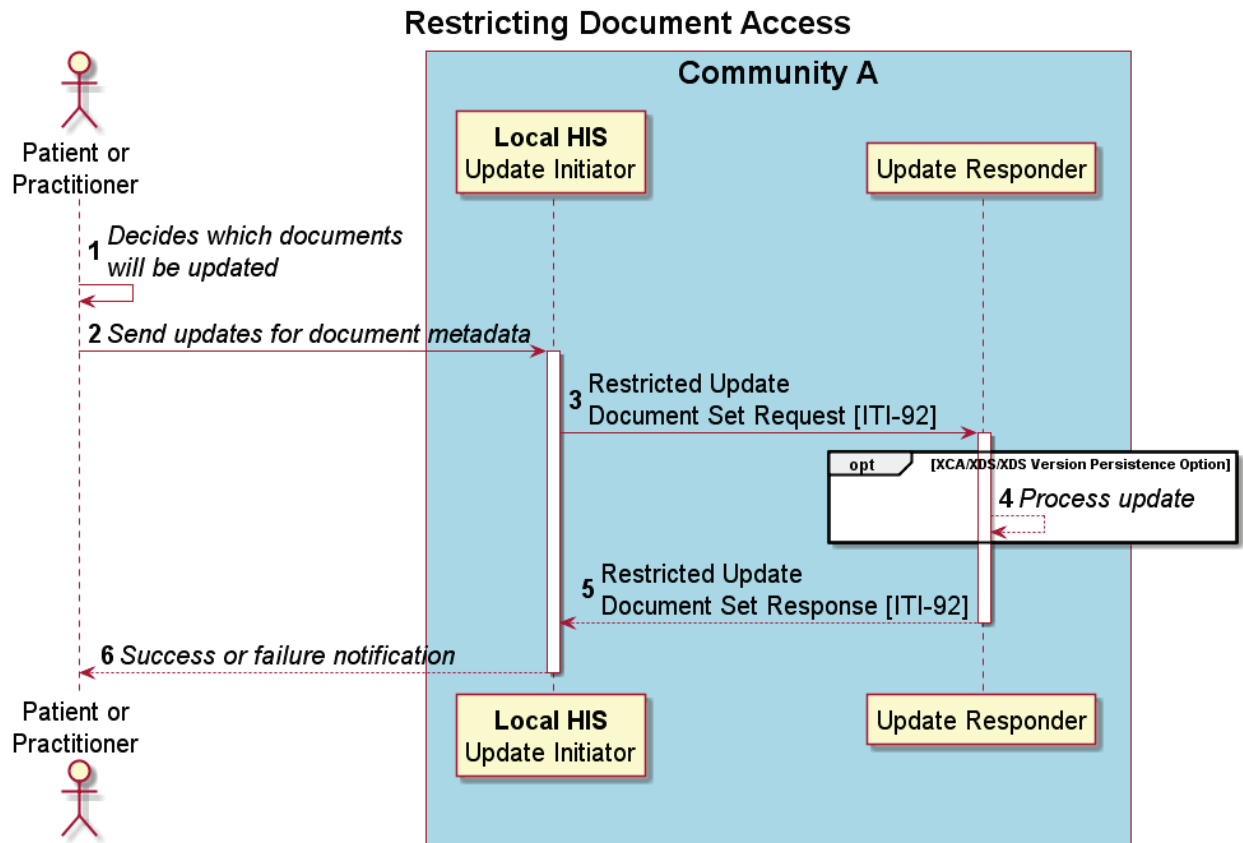


Figure 48.4.2.1-1: Restricting Document Access Process Flow

The text in Figure 48.4.2.1-2 below was used to generate the diagram in Figure 48.4.2.1-1. Readers will generally find the diagram more informative. The text is included here to facilitate editing.

```

@startuml
'Version 1.0

title Restricting Document Access

autonumber
skinparam sequence {
TitleFontSize 25
ArrowFontSize 20
ParticipantFontSize 20
ActorFontSize 20
BoxFontSize 25
ParticipantPadding 20
}
skinparam BoxPadding 10

actor "Patient or\nPractitioner" as USER

box "Community A" #LightBlue
participant "***Local HIS**\nUpdate Initiator" as UIA
participant "Update Responder" as URA
end box

USER->USER: //Decides which documents//\n//will be updated//

USER->UIA: //Send updates for document metadata//
activate UIA
UIA->URA:Restricted Update\nDocument Set Request [ITI-92]
activate URA
opt XCA/XDS/XDS Version Persistence Option
URA-->URA: //Process update//
end opt
URA-->UIA:Restricted Update\nDocument Set Response [ITI-92]
deactivate URA
UIA-->USER: //Success or failure notification//

deactivate UIA

@enduml

```

Figure 48.4.2.1-2: Restricting Document Access Pseudocode**48.4.2.2 Use Case #2: Allowing Document Access****48.4.2.2.1 Use Case Description**

Community A and B has agreed to share medical documents between their two communities. Community A has agreed to make available to practitioners, documents that are classified with both "normal" and "restricted" confidentiality. Community B has agreed only to share records classified as "normal" in order to meet local policy restrictions. As part of their document sharing agreement, a provision was added that allows patients to modify the confidentiality classification of their records.

A patient received treatment in Community B for a disease requiring specialized treatment. As part of this treatment, the patient's medical records are stored in this community with the confidentiality code classified as "restricted".

460 After successful treatment in Community B, the patient returns to Community A to meet with their primary care physician. In order to review the patient's treatment plan, the physician needs access to their medical records in Community B.

Utilizing the provision in the community's document sharing agreement, the patient changes the classification of their medical records in Community B from "restricted" to "normal". This
465 allows their primary care physician in Community A to access these records and review their treatment plan.

48.4.2.2.2 Process Flow

Starting with the latest version of the metadata for each document, the local HIS prepares the updated document metadata, reflecting the requested changes. The HIS uses the value of
470 homeCommunityId in the metadata to determine where the documents are stored.

With the metadata prepared, the HIS issues a metadata update request to Community A's Update Responder. The Update Responder uses the homeCommunityId to lookup the destination address for Community B and interacts with the grouped Update Initiator to forward the metadata to Community B for processing.

475 The Update Responder in the Community B processes the request. The Update Responder will return the response to the Community A's grouped Update Responder/Initiator which returns the response to the Update Initiator. The Update Initiator returns the response to the Local HIS which provides the user the success or failure of their request.

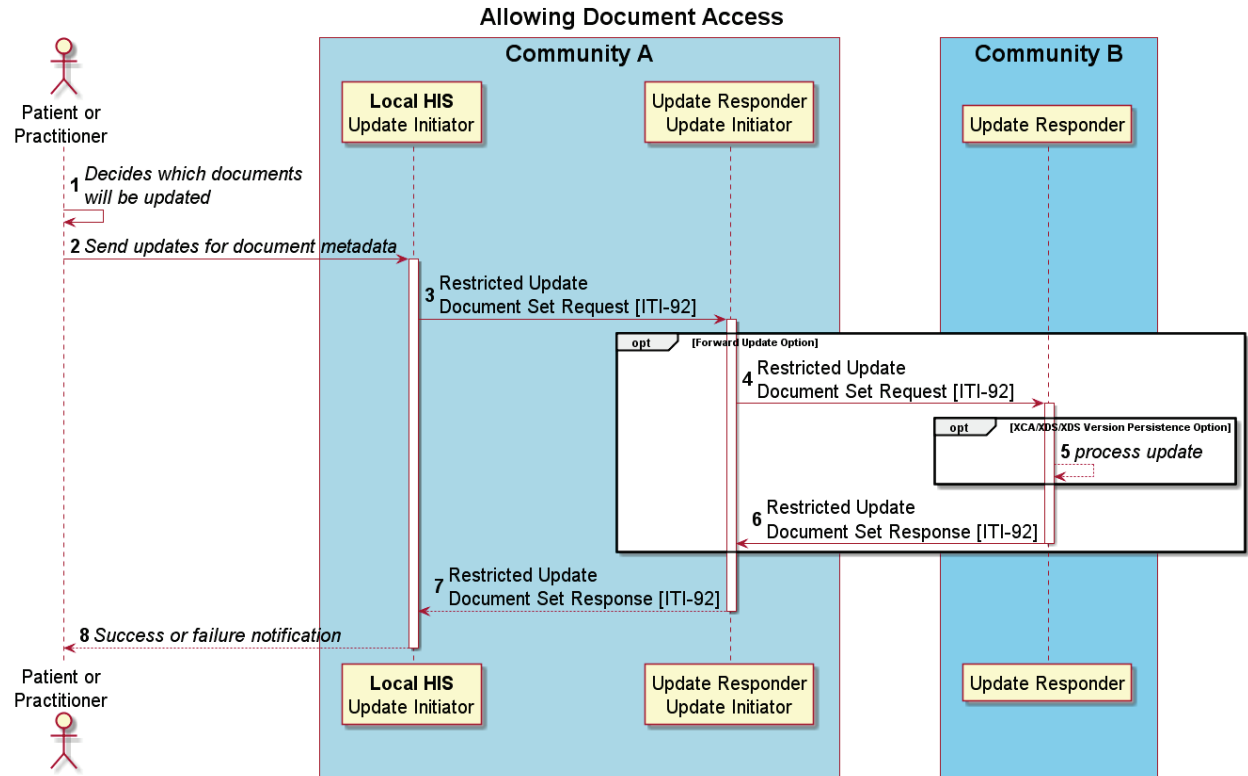


Figure 48.4.2.2-1: Allowing Document Access Process Flow

The text in Figure 48.4.2.2-2 below was used to generate the diagram in Figure 48.4.2.2-1. Readers will generally find the diagram more informative. The text is included here to facilitate editing.

485

```

@startuml
'Version 1.0

title Allowing Document Access

autonumber
skinparam sequence {
TitleFontSize 25
ArrowFontSize 20
ParticipantFontSize 20
ActorFontSize 20
BoxFontSize 25
ParticipantPadding 20
}
skinparam BoxPadding 10

actor "Patient or\nPractitioner" as USER

box "Community A" #LightBlue
participant "***Local HIS**\nUpdate Initiator" as UI
participant "Update Responder\nUpdate Initiator" as URA
end box

box "Community B" #SkyBlue
participant "Update Responder" as URB
end box

USER->USER: //Decides which documents//\n//will be updated//

USER->UI: //Send updates for document metadata//
activate UI
UI->URA:Restricted Update\nDocument Set Request [ITI-92]
activate URA
opt Forward Update Option
URA->URB:Restricted Update\nDocument Set Request [ITI-92]
activate URB
opt XCA/XDS/XDS Version Persistence Option
URB-->URB: //process update//
end opt
URB->URA:Restricted Update\nDocument Set Response [ITI-92]
deactivate URB
end opt
URA-->UI:Restricted Update\nDocument Set Response [ITI-92]
deactivate URA
UI-->USER: //Success or failure notification//

deactivate UI

@enduml

```

Figure 48.4.2.2-2: Allowing Document Access Pseudocode

48.5 Security Considerations

For general security considerations, see ITI TF-1: Appendix G - Security Considerations and ITI TF-2x: Appendix K - XDS Security Environment. Transaction specific security considerations are presented in the Security Considerations section of each transaction in Volume 2.

490

Updating patient care records is subject to local policies and government restrictions. As the sensitivity of the data may vary, the deployment will need to perform its own risk analysis and establish a mitigation strategy when enabling this profile's transaction.

495 In most cases, appropriate precautions should be taken to restrict use of this profile's transactions to users with sufficient privileges.

Please see ITI TF-1: Appendix L - XDS Affinity Domain Definition Checklist for more information.

48.6 Cross Profile Considerations

48.6.1 Grouping an XDS Document Consumer

500 The Update Initiator may be grouped with an XDS Document Consumer to obtain the latest version of the DocumentEntry object metadata.

When the Update Initiator is grouped with an XDS Document Consumer, the XDS Document Consumer will query the XDS Document Registry grouped with the Update Responder for the latest version of the metadata using the Registry Stored Query [ITI-18] transaction.

505 The Update Initiator will use those results to create the submission and submit the request using the Restricted Update Document Set [ITI-92] transaction. Update Responder will process and store the results of the Restricted Update Document Set [ITI-92] transaction in grouped XDS Document Registry so that the future query results reflect the requested changes in the metadata.

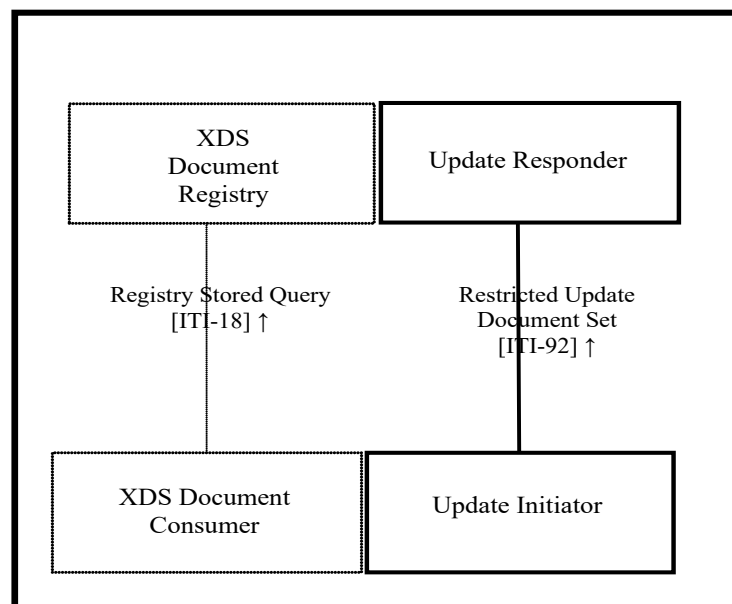


Figure 48.6.1-1: RMU and XDS Actor Groupings

48.6.2 Grouping an XCA Initiating Gateway

The Update Initiator may be grouped with an XCA Initiating Gateway to obtain the latest versions of the DocumentEntry object metadata.

- 515 The Initiating Gateway will use the Cross Gateway Query [ITI-38] transaction to contact Responding Gateway for the remote community servicing the request.

- The metadata in the Cross-Gateway Query [ITI-38] transaction response contains the patient identifier and coded values from the remote community. These are used by the local community's Update Initiator to form the request for the Restricted Update Document Set [ITI-92] transaction.
- 520

The Update Initiator will send the update request to the remote community's Update Responder for processing.

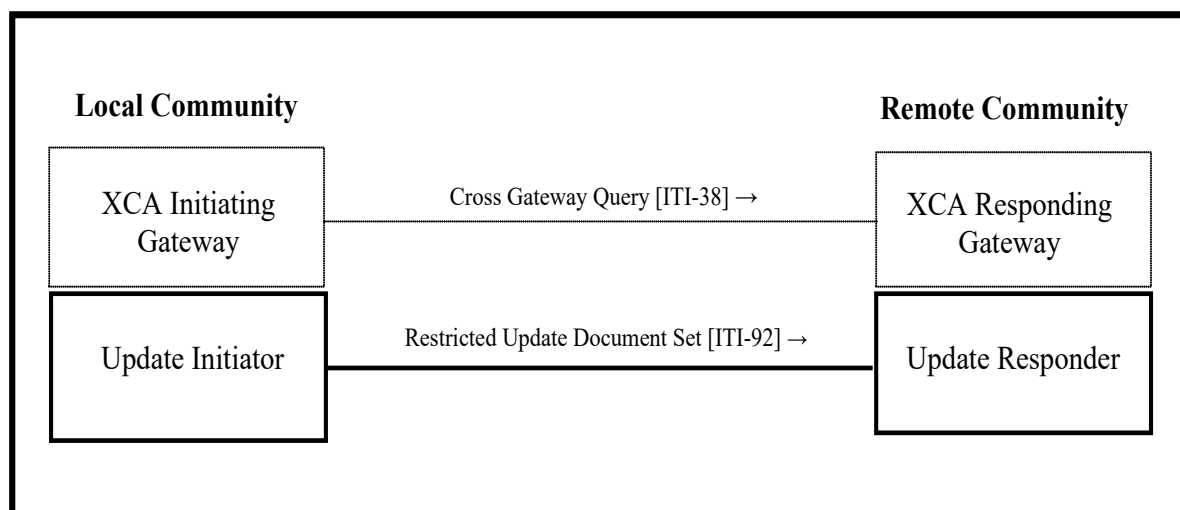


Figure 48.6.2-1: RMU and XCA Actor Groupings

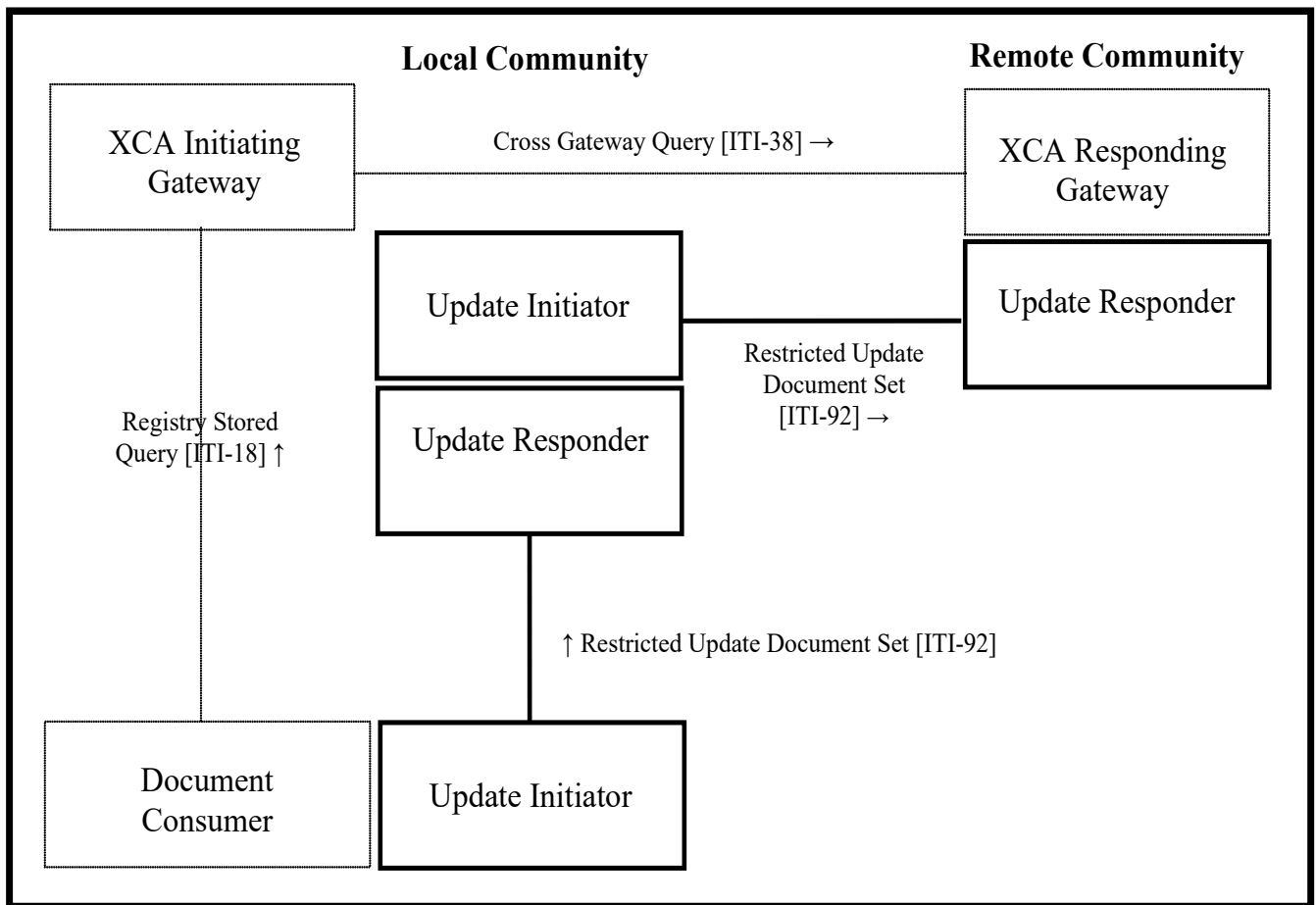
525 48.6.3 Grouping within an XCA/XDS Affinity Domain Option Environment

The Update Initiator in a local community may be grouped with an XDS Document Consumer to obtain the latest versions of the DocumentEntry object metadata.

- The XDS Document Consumer will use the Registry Stored Query [ITI-18] transaction to send a query to an XCA Initiating Gateway supporting the XDS Affinity Domain Option. The XCA Initiating Gateway will use the Cross Gateway Query [ITI-38] transaction to contact Responding Gateways for the remote community servicing the request.
- 530

- The metadata in the Cross-Gateway Query [ITI-38] transaction response contains the patient identifier and coded values from the remote community. These are used by the local community's Update Initiator to form the request for the Restricted Update Document Set [ITI-92] transaction.
- 535

The Update Initiator will send the update request to the local community's Update Responder. The Update Responder will be grouped with another Update Initiator responsible for forwarding the request to the remote community's Update Responder for processing.



540

Figure 48.6.3-1: RMU and XCA/XDS Affinity Domain Option Actor Groupings

Volume 2c – Transactions

Add Section 3.92

545 3.92 Restricted Update Document Set

This section corresponds to the Restricted Update Document Set [ITI-92] transaction of the IHE Technical Framework. Restricted Update Document Set [ITI-92] is used by the Update Initiator and Update Responder.

3.92.1 Scope

550 The Restricted Update Document Set [ITI-92] transaction is used to update DocumentEntry metadata objects from the Update Initiator to the Update Responder.

3.92.2 Actor Roles

The roles in this transaction are defined in the following table and may be performed by the actors shown here:

555 **Table 3.92.2-1: Actor Roles**

Actor:	Update Initiator
Role:	Prepares and issues an update to DocumentEntry metadata objects.
Actor:	Update Responder
Role:	Accepts requests for updates to DocumentEntry metadata objects.

3.92.3 Referenced Standard

ebRIM	OASIS/ebXML Registry Information Model v3.0 This model defines the types of metadata and content that can be stored in an ebXML Registry and forms the basis for the Document Sharing metadata model.
ebRS	OASIS/ebXML Registry Services Specifications v3.0 This defines the services and protocols for an ebXML Registry, used as the basis for the XDS Document Registry
ITI TF-2x: Appendix V	Web Services for IHE Transactions
ITI TF-3:4	Metadata Used in Document Sharing Profiles

3.92.4 Messages

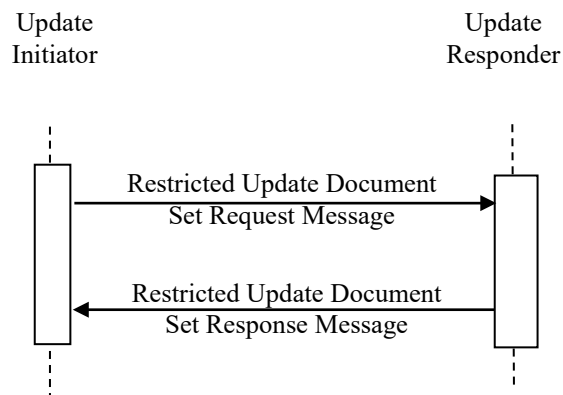


Figure 3.92.4-1: Interaction Diagram

3.92.4.1 Restricted Update Document Set Request Message

The Restricted Update Document Set Request message provides the ability to submit updated attributes for a DocumentEntry object in a community.

3.92.4.1.1 Trigger Events

The Restricted Update Document Set Request message is triggered when an Update Initiator needs to transmit updated DocumentEntry object metadata.

3.92.4.1.2 Message Semantics

The Restricted Update Document Set Request message shall use SOAP 1.2 and Simple SOAP. Implementers of this transaction shall comply with all requirements described in ITI TF-2x: Appendix V.3: Synchronous and Asynchronous Web Services.

XML namespace prefixes used in text and in examples below are for informative purposes only and are documented in ITI TF-2x: V.2.4.

The requirements for the request message are:

- <wsa:Action/> shall contain the value urn:ihe:iti:2018:RestrictedUpdateDocumentSet
- <soap12:Body/> shall contain one <lcm:SubmitObjectsRequest/> element representing the Submission Request (see ITI TF-3: 4.2.1.4 for details of expressing a Submission Request).
- <lcm:SubmitObjectsRequest/> shall contain one <rim:RegistryObjectList/>
- <rim:RegistryObjectList/> element shall include the following objects:

- One <rim:RegistryPackage/> classified as a Submission Set object (see ITI TF-3: 4.2.1.2).
- For each DocumentEntry metadata object:
 - One <rim:ExtrinsicObject/> that includes all required metadata attributes (see Section 3.92.4.1.2.1 Content) and ebRIM attribute, @lid representing the logicalID attribute which is the entryUUID of the initial version of the object.
 - One matching <rim:Association/> that represents a SS-HM HasMember relationship between the Submission Set and DocumentEntry Object. This Association shall include two instances of <rim:Slot/> for the SubmissionSetStatus and PreviousVersion metadata annotations. The Association may optionally include a <rim:Slot/> for the AssociationPropagation metadata annotation (see Section 3.92.4.1.2.2) with a value set to "yes".

The WSDL for this transaction is available online: see ITI TF-2x: Appendix W.

Below is an example of the SOAP Body for a Restricted Update Document Set Request message.

```

<soap12:Body>
  <lcm:SubmitObjectsRequest>
    <!-- Submission Request contents - See ITI TF-3: 4.2.1.4 -->
    <rim:RegistryObjectList>
      <rim:RegistryPackage id="SubmissionSet01" home="urn:oid:1.2.3.4.5.6.2333.23">
        <!-- Submission Set goes here -->
      </rim:RegistryPackage>
      <rim:ExtrinsicObject id="Document01" lid="urn:uuid:0000-0-00000000"
        home="urn:oid:1.2.3.4.5.6.2333.23">
        <!-- DocumentEntry metadata goes here -->
      </rim:ExtrinsicObject>
      <rim:Association
        associationType="urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember"
        sourceObject="SubmissionSet01"
        targetObject="Document01">
        <rim:Slot name="SubmissionSetStatus">
          <rim:ValueList>
            <rim:Value>Original</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="PreviousVersion">
          <rim:ValueList>
            <rim:Value>1</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="AssociationPropagation">
          <rim:ValueList>
            <rim:Value>yes</rim:Value>
          </rim:ValueList>
        </rim:Slot>
      </rim:Association>
    </rim:RegistryObjectList>
  </lcm:SubmitObjectsRequest>
</soap12:Body>

```

A full example is available online: see ITI TF-2x: Appendix W.

3.92.4.1.2.1 Content

635 The Update Initiator shall ensure that all metadata attributes are consistent with the requirements specified in ITI TF-3: Table 4.3.1-3 - Sending Actor Metadata Attribute Optionality column "RMU-SD" (Stable Document Entry) or "RMU-OD" (On-Demand Document Entry) as appropriate for the DocumentEntry objectType.

The Update Initiator shall use the latest version of the DocumentEntry object metadata as the basis for the update submission.

These metadata attributes shall not be modified by the Update Initiator:

- 640 • availabilityStatus
- entryUUID
- homeCommunityId
- logicalID
- version
- 645 • patientId
- sourcePatientId
- documentAvailability
- uniqueId
- repositoryUniqueId
- 650 • objectType

These metadata attributes describe the current state of the DocumentEntry object, or physical document stored in a repository.

Other IHE profiles, XDS Affinity Domain policies, or community policies may impose restrictions on updating metadata attributes not included in the preceding list.

655 3.92.4.1.2.2 Metadata Annotations

Metadata Annotations are added to SS-DE HasMember Association to specify the expected state or indicate the new state of a target object instance. The Update Responder shall use these annotations when processing the request for storage as specified below.

660 Metadata Annotations are coded as <rim:Slot/> child elements within a <rim:Association/> object.

3.92.4.1.2.2.1 PreviousVersion

The PreviousVersion annotation is constructed as a <rim:Slot/> with a name attribute equal to *PreviousVersion*. This slot contains a <rim:ValueList/> element containing a single

<rim:Value/>. The <rim:Value/> is populated with the version number of the existing object being updated in the submission.

If processed successfully, the submitted version of the metadata object will receive the value of the PreviousVersion annotation plus one.

```
<rim:Association
  associationType="urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember"
  sourceObject="SubmissionSet01"
  targetObject="Document01">
  <rim:Slot name="SubmissionSetStatus">
    <rim:ValueList>
      <rim:Value>Original</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="PreviousVersion">
    <rim:ValueList>
      <rim:Value>1</rim:Value>
    </rim:ValueList>
  </rim:Slot>
</rim:Association>
```

Figure 3.92.4.1.2.2.1-1: PreviousVersion Example

3.92.4.1.2.2.2 AssociationPropagation

The AssociationPropagation annotation triggers the Update Responder to create Association objects for the updated metadata object based on the Association objects linked to the existing metadata object version. The default value for Association Propagation is "yes".

This example shows a SS-DE HasMember Association for a new version of a DocumentEntry object where the AssociationPropagation annotation triggers association propagation by the Update Responder for the updated metadata object, Document01.

```

695 <rim:Association
      associationType="urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember"
      sourceObject="SubmissionSet01"
      targetObject="Document01">
        <rim:Slot name="SubmissionSetStatus">
          <rim:ValueList>
700           <rim:Value>Original</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="PreviousVersion">
          <rim:ValueList>
705           <rim:Value>1</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="associationPropagation">
          <rim:ValueList>
710           <rim:Value>yes</rim:Value>
          </rim:ValueList>
        </rim:Slot>
      </rim:Association>

```

Figure 3.92.4.1.2.2-1: Example SubmissionSet – DocumentEntry HasMember Association as part of an update including AssociationPropagation annotation

3.92.4.1.3 Expected Actions

The Update Responder shall process the Restricted Update Document Set Request message according to the capabilities described in the following sections.

Any error that occurs during the processing of the Restricted Update Document Set Request message shall cause the entire transaction to fail and no change made to the existing objects. The Update Responder shall return the status and any error codes incurred during the processing of the request in its response message.

3.92.4.1.3.1 Forward Update Option

The Update Responder shall validate the value of the homeCommunityId.

If the value is not known, the Update Responder shall return the error code, *XDSUnknownCommunity*. If the value is omitted, the Update Responder shall return the error code, *XDSMissingHomeCommunityId*.

If valid, the request shall be forwarded to the Update Responder servicing the community configured for the homeCommunityId contained in the request. The Update Initiator may return the error code, *XDSUnavailableCommunity*, if the Update Responder could not be contacted or is unable to process the request.

3.92.4.1.3.2 XCA Persistence Option

735 The Update Responder shall validate the request according to the requirements in Section 3.92.4.1.3.5 - Request Validation. If the request is valid, the Update Responder shall persist the data in the grouped XCA Responding Gateway. If successful, the updated metadata shall be returned in a subsequent Cross Gateway Query [ITI-38] transaction.

3.92.4.1.3.3 XDS Persistence Option

740 The Update Responder shall validate the request according to the requirements in Section 3.92.4.1.3.5 - Request Validation. If the request is valid, the Update Responder shall persist the data in the grouped with the Document Registry. If successful, the updated metadata shall be returned in a subsequent Registry Stored Query [ITI-18] transaction.

3.92.4.1.3.4 Version Persistence Option

745 The Update Responder shall validate the request according to the requirements in Section 3.92.4.1.3.5 - Request Validation. If the request is valid, the Update Responder shall persist the data in the grouped with the Document Registry according to the requirements in Section 3.92.4.1.3.4.1.

750 If successful, the updated metadata shall be returned in a subsequent Registry Stored Query [ITI-18] transaction response consistent with metadata versioning semantics described in ITI TF-2a: 3.18.4.1.2.3.5.1 - Compatibility Issues (currently in the XDS Metadata Update Trial Implementation Supplement).

3.92.4.1.3.4.1 Storage Requirements

The Update Responder updates the metadata of a DocumentEntry object by persisting a new version of the DocumentEntry object.

755 The Update Responder shall be capable of storing multiple versions of a logical DocumentEntry metadata object.

For each DocumentEntry object, the following actions are performed:

1. Store the new DocumentEntry
 - a. Set the version attribute value to PreviousVersion plus one.
- 760 2. Change the availabilityStatus attribute for the existing DocumentEntry object to Deprecated if the availabilityStatus is Approved..
3. If Association Propagation slot is missing or has a value equalled to "yes", the Update Responder shall perform the following actions:
 - a. Scan for existing approved HasMember Associations for the existing DocumentEntry object. For each HasMember Association found:
 - 765 i. If the Association links the existing DocumentEntry object to a Folder, create a new FD-DE HasMember Association linking the new DocumentEntry object with

the Folder. The existing FD-DE Has Member Association availabilityStatus will remain approved.

770 In addition, create an additional SS-HM HasMember Association linking the created FD-DE HasMember Association to the submitted SubmissionSet object.

- ii. If the Association links the existing DocumentEntry object to an existing SubmissionSet with the SubmissionSetStatus attribute set to "Reference", then deprecate this Association and create a new Association between the existing SubmissionSet and the submitted version of the DocumentEntry.

775 Note: In this case, the associated objects are not required to have the same patientId attribute value.

- b. Scan for approved relationship associations linked to the existing DocumentEntry object (see ITI TF-3: 4.2.2.2). For each association found, create a copy of the association and replace the entryUUID referencing the existing DocumentEntry object with the entryUUID of the submitted DocumentEntry object.
- 780 c. Verify that a generated Association object has not been created more than once during processing (a.k.a.: overlapping updates). If so, the duplicate Association object shall be discarded. This condition is detected by checking the sourceObject and targetObject attributes of the Association object.
- d. Store the generated Association objects.

785 3.92.4.1.3.5 Request Validation

The following rules shall be used by the Update Responder to validate the submission.

Unless a more specific code is provided within the validation, the Update Responder shall return the error code, *XDSMetadataUpdateError*, for any error returned during processing:

- 790 1. Verify all metadata objects contained in the request has the same value for the homeCommunityId and matches the value configured for the Update Responder's community. If this value does not match, the Update Responder shall return the error code, *XDSUnknownCommunity*. If the value is omitted, the Update Responder may return the error code, *XDSMissingHomeCommunityId*.
- 795 2. Verify the request does not contain DocumentEntry object updates that have a value for Association Propagation equaled to "no". If found, the error code, *XDSMetadataAnnotationError*, should be returned.
- 800 3. Verify the request does not include original versions of DocumentEntry objects (i.e.: entryUUID equals the logicalID or the logicalID is missing). If an original version of a DocumentEntry is received, the error code, *XDSInvalidRequestException*, should be returned.
- 4. Verify the metadata object submitted for updating is a DocumentEntry object. Otherwise, the error code, *XDSObjectTypeError*, should be returned.

- 805 5. Verify the system contains an existing DocumentEntry (Stable or On-Demand) metadata object instance. If the DocumentEntry cannot be located, the error code, *UnresolvedReferenceException*, shall be returned.
6. Verify that the SS-DE HasMember Association is present and has a Slot with name PreviousVersion. The value of the PreviousVersion Slot shall match the version number of the existing DocumentEntry. Otherwise, the error code, *XDSMetadataVersionError*, shall be returned.
- 810 7. Verify the submitted and existing DocumentEntry objects have the same values for both the logicalID and uniqueId attribute. If these values are not identical, the error code, *XDSMetadataIdentifierError*, should be returned.
8. Verify the submitted and existing DocumentEntry objects have the same values for the patientId attributes. If these values are not identical, the error code, *XDSPatientIDReconciliationError*, should be returned.
- 815 9. Check the submitted DocumentEntry metadata and determine if it contains any change to an unmodifiable attribute, as described in Section 3.92.4.1.2.1. If so, the error code, *UnmodifiableMetadataError*, should be returned.
- 820 10. If any portion of the submitted DocumentEntry metadata contains modifications to an attribute that violates local policy, the error code, *LocalPolicyRestrictionError*, should be returned.
- 825 11. All metadata objects must conform to the rules for content and format defined in ITI TF-3: 4.2 and 4.3. Stable Document Entry object metadata must confirm to the requirements defined in ITI TF-2b: 3.42 for the Register Document Set-b [ITI-42] transaction. On-Demand Document Entry objects must conform to the requirements defined in ITI TF-2b: 3.61 for the Register On-Demand Document Entry [ITI-61] transaction. If an error is encountered, the appropriate error code from ITI TF-3: Table 4.2.4.1-2 shall be returned.

3.92.4.2 Restricted Update Document Set Response Message

830 The Restricted Update Document Set Response is sent when the Update Responder has completed processing of the Restricted Update Document Set Request.

3.92.4.2.1 Trigger Events

The Restricted Update Document Set Request message processing has been completed.

3.92.4.2.2 Message Semantics

835 The Restricted Update Document Set Response message shall use SOAP 1.2 and Simple SOAP. Implementers of this transaction shall comply with all requirements described in ITI TF-2x: Appendix V.3: Synchronous and Asynchronous Web Services.

XML namespace prefixes used in text and in examples below are for informational purposes only and are documented in ITI TF-2x: Appendix V, Table V.2.4-1.

The requirements for the response message are as follows:

- 840
- `<wsa:Action/>` shall contain the value
`urn:ihe:iti:2018:RestrictedUpdateDocumentSetResponse`
 - `<soap12:Body/>` shall contain one `<rs:RegistryResponse/>` element

The `rs:RegistryResponse/@status` attribute provides the overall status of the request. It shall contain one of two values:

- 845
- If all metadata in the request was updated successfully, the Update Responder shall set the status equal to `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success`.
 - If the metadata could not be updated successfully, then the Update Responder shall set the status equal to `urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Failure`.
- 850

If an error occurs when updating a metadata object, then a `rs:RegistryResponse/rs:RegistryErrorList/rs:RegistryError` element shall be returned in the response with:

- `@severity` is `urn:oasis:names:tc:ebxml-regrep:ErrorSeverityType:Error`.
- 855 • `@errorCode` contains an error code from ITI TF-3: Table 4.2.4.1-2.
- `@codeContext` contains the error message and the entryUUID for the object that caused the error.

See ITI TF-3: 4.2.4.1 for examples of response messages.

3.92.4.2.3 Expected Actions

- 860
- When the Update Initiator receives a success response, the metadata objects were successfully updated and the transaction is complete. The Update Initiator can continue processing normally.

If an error response was received, the Update Initiator may need to perform additional steps to determine the cause and correct the error. These steps are not specified by this transaction.

3.92.5 Security Considerations

- 865
- Additional security considerations that may apply are discussed in ITI TF-1: 48.5 - RMU Security Considerations.

3.92.5.1.4.1 Audit Record Considerations

The Restricted Update Document Set [ITI-92] transaction is PHI-Patient Record event, as defined in ITI TF-2a: Table 3.20.4.1.1.1-1 with the following exceptions:

870

3.92.5.1.4.1.1 Update Initiator Audit Message

	Field Name	Opt	Value Constraints
Event <i>AuditMessage/ EventIdentification</i>	EventID	M	EV(110106, DCM, “Export”)
	EventActionCode	M	“U” (Update)
	<i>EventDateTime</i>	M	<i>not specialized</i>
	<i>EventOutcomeIndicator</i>	M	<i>not specialized</i>
	EventTypeCode	M	EV(“ITI-92”, “IHE Transactions”, “Restricted Update Document Set”)
Source (Update Initiator) (1)			
Human Requestor (0..n)			
Destination (Update Responder) (1)			
Audit Source (Update Initiator) (1)			
Patient (1)			
SubmissionSet (1)			

Where:

Source <i>AuditMessage/ ActiveParticipant</i>	UserID	U	<i>not specialized</i>
	AlternativeUserID	M	Process ID as used within the local operating system in the local system logs.
	<i>UserName</i>	U	<i>not specialized</i>
	UserIsRequestor	U	<i>not specialized</i>
	RoleIDCode	M	EV(110153, DCM, “Source”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address

Human Requestor (if known) <i>AuditMessage/ ActiveParticipant</i>	UserID	M	Identity of the human that initiated the transaction.
	<i>AlternativeUserID</i>	U	<i>not specialized</i>
	<i>UserName</i>	U	<i>not specialized</i>
	<i>UserIsRequestor</i>	U	<i>not specialized</i>
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	<i>NetworkAccessPointTypeCode</i>	U	<i>not specialized</i>
	<i>NetworkAccessPointID</i>	U	<i>not specialized</i>

Destination <i>AuditMessage/ ActiveParticipant</i>	UserID	M	SOAP endpoint URI.
	<i>AlternativeUserID</i>	U	<i>not specialized</i>
	<i>UserName</i>	U	<i>not specialized</i>
	UserIsRequestor	M	<i>not specialized</i>
	RoleIDCode	M	EV(110152, DCM, “Destination”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address

875

Audit Source (AuditMessage/ AuditSourceIdentification)	<i>AuditSourceID</i>	<i>U</i>	<i>not specialized</i>
	<i>AuditEnterpriseSiteID</i>	<i>U</i>	<i>not specialized</i>
	<i>AuditSourceTypeCode</i>	<i>U</i>	<i>not specialized</i>

Patient (AuditMessage/ ParticipantObjectIdentification)	<i>ParticipantObjectTypeCode</i>	<i>M</i>	“1” (Person)
	<i>ParticipantObjectTypeCodeRole</i>	<i>M</i>	“1” (Patient)
	<i>ParticipantObjectDataLifeCycle</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectIDTypeCode</i>	<i>M</i>	<i>not specialized</i>
	<i>ParticipantObjectSensitivity</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectID</i>	<i>M</i>	The patient ID in HL7 ¹ CX format.
	<i>ParticipantObjectName</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectQuery</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectDetail</i>	<i>U</i>	<i>not specialized</i>

Submission Set (AuditMessage/ ParticipantObjectIdentification)	<i>ParticipantObjectTypeCode</i>	<i>M</i>	“2” (System)
	<i>ParticipantObjectTypeCodeRole</i>	<i>M</i>	“20” (job)
	<i>ParticipantObjectDataLifeCycle</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectIDTypeCode</i>	<i>M</i>	EV(“urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd”, “IHE XDS Metadata”, “submission set classificationNode”)
	<i>ParticipantObjectSensitivity</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectID</i>	<i>M</i>	SubmissionSet uniqueID
	<i>ParticipantObjectName</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectQuery</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectDetail</i>	<i>C</i>	If known, set the “urn:ihe:iti:xca:2010:homeCommunityId” as the value of the attribute type and the value of the homeCommunityId as the value of the attribute value.

3.92.5.1.4.1.2 Update Responder Audit Message

	Field Name	Opt	Value Constraints
Event (AuditMessage/ EventIdentification)	<i>EventID</i>	<i>M</i>	EV(110107, DCM, “Import”)
	<i>EventActionCode</i>	<i>M</i>	“U” (Update)
	<i>EventDateTime</i>	<i>M</i>	<i>not specialized</i>
	<i>EventOutcomeIndicator</i>	<i>M</i>	<i>not specialized</i>
	<i>EventTypeCode</i>	<i>M</i>	EV(“ITI-92”, “IHE Transactions”, “Restricted Update Document Set”)
Source (Update Initiator) (1)			
Destination (Update Responder) (1)			

¹ HL7 is the registered trademark of Health Level Seven International.

Audit Source (Update Responder) (1)
Patient (1)
SubmissionSet (1)

880

Where:

Source (AuditMessage/ ActiveParticipant)	<i>UserID</i>	<i>U</i>	<i>not specialized</i>
	<i>AlternativeUserID</i>	<i>U</i>	<i>not specialized</i>
	<i>UserName</i>	<i>U</i>	<i>not specialized</i>
	<i>UserIsRequestor</i>	<i>U</i>	<i>not specialized</i>
	<i>RoleIDCode</i>	<i>M</i>	EV(110153, DCM, “Source”)
	<i>NetworkAccessPointTypeCode</i>	<i>M</i>	“1” for machine (DNS) name, “2” for IP address
	<i>NetworkAccessPointID</i>	<i>M</i>	The machine name or IP address

Destination (AuditMessage/ ActiveParticipant)	<i>UserID</i>	<i>M</i>	SOAP endpoint URI
	<i>AlternativeUserID</i>	<i>M</i>	Process ID as used within the local operating system in the local system logs.
	<i>UserName</i>	<i>U</i>	<i>not specialized</i>
	<i>UserIsRequestor</i>	<i>M</i>	“false”
	<i>RoleIDCode</i>	<i>M</i>	EV(110152, DCM, “Destination”)
	<i>NetworkAccessPointTypeCode</i>	<i>M</i>	“1” for machine (DNS) name, “2” for IP address
	<i>NetworkAccessPointID</i>	<i>M</i>	The machine name or IP address

Audit Source (AuditMessage/ AuditSourceIdentification)	<i>AuditSourceID</i>	<i>U</i>	<i>not specialized</i>
	<i>AuditEnterpriseSiteID</i>	<i>U</i>	<i>not specialized</i>
	<i>AuditSourceTypeCode</i>	<i>U</i>	<i>not specialized</i>

Patient (AuditMessage/ ParticipantObjectIdentification)	<i>ParticipantObjectTypeCode</i>	<i>M</i>	“1” (Person)
	<i>ParticipantObjectTypeCodeRole</i>	<i>M</i>	“1” (Patient)
	<i>ParticipantObjectDataLifeCycle</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectIDTypeCode</i>	<i>M</i>	<i>not specialized</i>
	<i>ParticipantObjectSensitivity</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectID</i>	<i>M</i>	The patient ID in HL7 CX format.
	<i>ParticipantObjectName</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectQuery</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectDetail</i>	<i>U</i>	<i>not specialized</i>

Submission Set (AuditMessage/ ParticipantObjectIdentification)	<i>ParticipantObjectTypeCode</i>	<i>M</i>	“2” (System)
	<i>ParticipantObjectTypeCodeRole</i>	<i>M</i>	“20” (job)
	<i>ParticipantObjectDataLifeCycle</i>	<i>U</i>	<i>not specialized</i>

	ParticipantObjectIDTypeCode	<i>M</i>	EV(“urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd”, “IHE XDS Metadata”, “submission set classificationNode”)
	<i>ParticipantObjectSensitivity</i>	<i>U</i>	<i>not specialized</i>
	ParticipantObjectID	<i>M</i>	SubmissionSet uniqueID
	<i>ParticipantObjectName</i>	<i>U</i>	<i>not specialized</i>
	<i>ParticipantObjectQuery</i>	<i>U</i>	<i>not specialized</i>
	ParticipantObjectDetail	<i>C</i>	If known, set the “urn:ihe:iti:xca:2010:homeCommunityId” as the value of the attribute type and the value of the homeCommunityId as the value of the attribute value.

885

Volume 3 – Cross-Transaction and Content Specifications

4.1 Abstract Metadata Model

Editor: Update Table 4.1.3.2-1: DocumentEntry Metadata Attribute Definition to add the following rows in alphabetical order in the table

Table 4.1.3.2-1: DocumentEntry Metadata Attribute Definition

DocumentEntry Metadata Attribute	Description	Patient identity	Provenance	Security & Privacy	Descriptive	Object Lifecycle	Exchange
<u>documentAvailability</u>	<u>The status of the Document in the Document Repository</u>					<u>X</u>	<u>X</u>
<u>logicalID</u>	<u>A globally unique identifier used to identify the logical entry.</u>		<u>X</u>			<u>X</u>	<u>X</u>
<u>version</u>	<u>Version number of a DocumentEntry.</u>		<u>X</u>			<u>X</u>	<u>X</u>

Editor: Add new Section 4.1.5.

4.1.5 Metadata Object Versioning Semantics

One part of updating metadata is the management of metadata object versioning as specified in ebRIM 3.0. ebRIM 3.0 version control introduces the following concepts to support versioning:

Metadata Object Instance – a single metadata object representing a single version of an object

Logical Metadata Object – the collection of metadata object instances that are the versions of a single object. Each metadata object instance is a different version of the logical metadata object.

Before the ability to update metadata was added to some Document Sharing profiles, a logical metadata object was always represented by a single instance so differentiating logical and instance was not important.

A logical DocumentEntry metadata object represents a single document in a Document Repository. The logical DocumentEntry object encompasses all versions (instances) of the DocumentEntry object that have historically represented the document in the Document Repository.

An association, through its sourceObject and targetObject attributes, references metadata object instances (particular versions of the objects).

Metadata versions are identified/managed using two metadata attributes: logicalID and version:

915 **logicalID**

- Each object instance is assigned a logicalID (@lid) along with its entryUUID (@id)
- The first version of an object has entryUUID equal to logicalID
- Each metadata object instance has a unique value for the entryUUID attribute
- Each logical object has a unique value for the logicalID attribute
- 920 • Each logical object is represented by one or more object instances.
- All objects with the same logicalID shall be of the same object type. The logicalID shall identify a group of DocumentEntry objects with the same objectType attribute.
- The rules for interpreting logicalID are:
 1. All object instances with the same logicalID are versions of the same logical object
 - 925 2. Each object instance has a unique entryUUID
 3. The first version of a logical object has logicalID equals the entryUUID.
 4. The second and later versions of a logical object have logicalID not equal the entryUUID
 - 930 5. If an object instance is submitted with no logicalID attribute, the value for logicalID defaults to the value of the entryUUID for that object instance becoming the first version of the logical metadata object.

version

- Instances of a logical metadata object are assigned a version through the version attribute as described in ITI TF-3: Tables 4.2.3.2-1.
- 935 • The highest numbered version of an object instance shall have availabilityStatus of urn:oasis:names:tc:ebxml-regrep:StatusType:Approved or urn:oasis:names:tc:ebxml-regrep:StatusType:Deprecated. All older versions shall have availabilityStatus of urn:oasis:names:tc:ebxml-regrep:StatusType:Deprecated.
- 940 • When updates are submitted, they reference the version being updated.
- Changes shall only be accepted for the most recent version.

945 The uniqueId, logicalID, and objectType attributes of a logical DocumentEntry object shall not be altered through versioning. They are required to be consistent across all object instances within a logical object. Document Replacement shall be used when any of these attributes are required to be changed.

4.1.5.1 Association Propagation

When an updated version of a DocumentEntry object is submitted, certain HasMember and Relationship Associations will be replicated from the existing to the submitted new version of a DocumentEntry object. The receiving actor is responsible for performing these actions. This responsibility is called Association Propagation.

4.2.3 Metadata Attributes

...

4.2.3.2 Document Metadata Attribute Definition

Editor: Update ITI TF-3: Table 4.2.3.2-1 Document Metadata Attribute Definition as shown below

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

DocumentEntry Metadata Attribute	Description	Data Type	Coding (Section 4.2.3.1.8)	Detail (See Section)
...				
creationTime	The time the author created the document. Shall have a single value.	DTM	ebRIM Slot	4.2.3.2.6
<u>documentAvailability</u>	<u>The status of the Document in the Document Repository</u>	<u>Predefined URN</u>	<u>ebRIM Slot</u>	<u>4.2.3.2.30</u>
entryUUID	A globally unique identifier used to identify the entry.	UUID	XML attribute	4.2.3.2.7
limitedMetadata	Indicates whether the Document Entry was created using the less rigorous requirements of metadata as defined for the Metadata-Limited Document Source.	4.2.3.2.29	ebRIM Classification	4.2.3.2.29
<u>logicalID</u>	<u>A globally unique identifier used to identify the logical entry.</u>	<u>UUID</u>	<u>XML attribute</u>	<u>4.2.3.2.31</u>
contentType	MIME type of the document.	MIME type	XML attribute	4.2.3.2.15
...				
URI	The URI for the document.	URI	ebRIM Slot	4.2.3.2.27
<u>version</u>	<u>Version number of a DocumentEntry.</u>	<u>Integer</u>	<u>ebRIM VersionInfo</u>	<u>4.2.3.2.32</u>

<i>Editor: Add the following new sub-sections to Section 4.2.3.2 as shown</i>

965 **4.2.3.2.30 DocumentEntry.documentAvailability**

Description:

The attribute represents the status of the physical document in the Document Repository. Online indicates the document is available to be retrieved. Offline indicates the document is not available to be retrieved.

970 **Coding:**

If present, this attribute is coded as an ebRIM Slot and shall have a single value, either:

urn:ihe:iti:2010:DocumentAvailability:Online, or

urn:ihe:iti:2010:DocumentAvailability:Offline. If this attribute is not present in metadata, the default value is urn:ihe:iti:2010:DocumentAvailability:Online.

975 The following example indicates the document associated with the DocumentEntry is not available to be retrieved.

980

```
<rim:Slot name="documentAvailability">
  <rim:ValueList>
    <rim:Value>urn:ihe:iti:2010:DocumentAvailability:Offline</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

4.2.3.2.31 DocumentEntry.logicalID

985 **Description:**

A globally unique identifier used to identify the logical DocumentEntry object. All versions of a DocumentEntry object carry the same logicalID.

990 The first version of a DocumentEntry object has logicalID equal to entryUUID. Other versions have logicalID not equal to entryUUID. If not present in a submission, logicalID defaults to the value of the entryUUID attribute. See Section 4.1.5 - Metadata Object Versioning Semantics for more information.

Coding:

Max length is unbounded. The format of the logicalID value is UUID. The logicalID shall never be submitted in symbolic form.

995 The value of the logicalID is coded in the lid XML attribute on the ebRIM ExtrinsicObject representing the DocumentEntry object.

The following two forms, with logicalID (@lid) missing or entryUUID (@id) equal to the logicalID (@lid), shall only be submitted in a transaction supporting original submissions (e.g., Register Document Set [ITI-42]):

1000

```
<rim:ExtrinsicObject
  id="urn:uuid:3cce0135-cedb-4a26-ba00-8698ee8dde04">
...
</rim:ExtrinsicObject>
```

1005

```
<rim:ExtrinsicObject
  id="urn:uuid:3cce0135-cedb-4a26-ba00-8698ee8dde04"
  lid="urn:uuid:3cce0135-cedb-4a26-ba00-8698ee8dde04">
...
</rim:ExtrinsicObject>
```

1010

The following form, with entryUUID (@id) different from logicalID (@lid), shall only be submitted in a transaction supporting metadata updates (e.g., Update Document Set [ITI-57]):

1015

```
<rim:ExtrinsicObject
  id="urn:uuid:3cce0135-cedb-4a26-ba00-8698ee8dde04"
  lid="urn:uuid:e0985823-dc50-45a5-a6c8-a11a829893bd">
...
</rim:ExtrinsicObject>
```

1020

4.2.3.2.32 DocumentEntry.version

Description:

This is the version number of a DocumentEntry object assigned by the storing actor (e.g.: Document Registry). The first version of a DocumentEntry shall have a value of 1. Subsequent versions get values of 2, 3, etc.

1025

This attribute shall be returned in query responses. If present in a submission, it shall be ignored.

Coding:

Max length is unbounded.

The value of the version attribute is coded in XML as the versionName attribute on the ebRIM VersionInfo element.

1030

```
<rim:VersionInfo versionName="2"/>
```

4.2.4 Success and Error Reporting

...

1035 4.2.4.1 RegistryError Element

Editor: Update ITI TF-3: Table 4.2.4.1-2 Error Codes as shown. Update Note 1 and replace Note 2 as shown.

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

Error Code	Discussion	Transaction (See Note 1)
...		
XDSMissingHomeCommunityId	A value for the homeCommunityId is required and has not been specified	SQ, XGQ, RS, XGR, <u>RU</u>
XDSUnknownCommunity	A value for the homeCommunityId is not recognized	SQ, XGQ, RS, XGR, <u>RU</u>
XDSUnavailableCommunity	A community which would have been contacted was not available <u>or capable of processing the request.</u> See Note 2.	SQ, RS, <u>RU</u>
InvalidDocumentContent	The recipient has rejected this submission because it detected that one of the documents does not match the metadata (e.g., formatCode) or has failed other requirements for the document content. When the RegistryError element contains this error code, the @codeContext shall contain the DocumentUniqueID of the document in error. If multiple documents are in error, there shall be a separate RegistryError element for each document in error.	P, <u>RU</u>
UnresolvedReferenceException	The recipient cannot resolve an entryUUID reference in the transaction.	P, R, <u>RU</u>
...		
<u>XDSMetadataUpdateError</u>	<u>General metadata update error. Use only when more specific error code is not available or appropriate.</u>	<u>RU</u>
<u>XDSPatientIDReconciliationError</u>	<u>Update encountered an error where patient identifiers did not match</u>	<u>RU</u>
<u>XDSMetadataVersionError</u>	<u>The version number included in the update request did not match the existing object. One cause of this is multiple simultaneous update attempts.</u>	<u>RU</u>
...		

Error Code	Discussion	Transaction (See Note 1)
<u>XDSObjectTypeError</u>	<u>The receiving actor cannot store the request as the objectType is either not supported or does not match the value of the existing object.</u>	<u>RU</u>
<u>XDSMetadataIdentifierError</u>	<u>The receiving actor cannot store the request because the identifier is in conflict with an existing known value.</u>	<u>RU</u>
<u>XDSInvalidRequestException</u>	<u>The receiving actor detected that an initial version of a metadata object instance was received in an update transaction.</u>	<u>RU</u>
<u>UnmodifiableMetadataError</u>	<u>An update for a metadata object includes changes to values for attributes which are prohibited for the transaction.</u>	<u>RU</u>
<u>LocalPolicyRestrictionError</u>	<u>The receiving actor has detected that the request is not permitted because of local policy restrictions or violation of a previously reached agreement.</u>	<u>RU</u>
<u>XDSMetadataAnnotationError</u>	<u>The receiving actor has detected an error with a SS-HM annotation.</u>	<u>RU</u>

1040

Note 1:

P = Provide and Register-b

R = Register-b

SQ = Stored Query

1045

RS = Retrieve Document Set

XGQ = Cross Gateway Query

XGR = Cross Gateway Retrieve

RU = Restricted Update Document Set

1050

Note 2:**Two examples of the use of error code XDSUnavailableCommunity are:****A Cross Gateway Query or Cross Gateway Retrieve fails because the community identified by a homeCommunityId could not be contacted.****A Cross Gateway Query based on Patient ID could not contact some known communities to relay the query.**

1055

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.**In the following situations, the error code XDSUnavailableCommunity may be generated by an Initiating Gateway supporting the XDS Affinity Domain Option and returned in the Registry Stored Query or Retrieve Document Set:**

1060

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 code, **XDSUnavailableCommunity** may be returned by an Update Initiator when a community's Update Responder, as identified by a homeCommunityId, could not be contacted or was not capable of processing the request.

1065 4.3 Additional Document Sharing Requirements

...

4.3.1 Submission Metadata Attribute Optionality

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

1070 The Actor/Transaction pairs addressed by this section are as follows:

Editor: Update Table 4.3.1-1 as shown adding actors RMU Update Initiator (Stable Document Entry) and RMU Update Initiator (On-Demand Document Entry).

Table 4.3.1-1: Sending Actor/Transaction Pairs

Actor	Transaction	Shortname
XDS Document Source	Provide and Register Document Set-b [ITI-41]	XDS DS
XDS Document Repository	Register Document Set-b [ITI-42]	XDS DR
XDM Portable Media Creator	Distribute Document Set on Media [ITI-32]	XDM MC
XDR Document Source	Provide and Register Document Set-b [ITI-41]	XDR DS
XDR Metadata-Limited Document Source	Provide and Register Document Set-b [ITI-41]	XDR MS
XDS On-Demand Document Source	Register On-Demand Document Entry [ITI-61]	XDS OD
<u>RMU Update Initiator (Stable Document Entry)</u>	<u>Restricted Update Document Set [ITI-92]</u>	<u>RMU SD</u>
<u>RMU Update Initiator (On-Demand Document Entry)</u>	<u>Restricted Update Document Set [ITI-92]</u>	<u>RMU OD</u>

1075 For each actor/transaction pair across the top of Table 4.3.1-3, and each metadata attribute row, the cell indicates the requirement for that actor when creating a submission request using the paired transaction. The requirements are expressed through these codes:

Editor: Update Table 4.3.1-2 as shown adding code "I" – "Immutable".

Table 4.3.1-2: Sending Actor Metadata Attribute Optionality Code Definitions

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

Code	Meaning
X	Prohibited – when sending a submission, a value for the attribute shall not be supplied by the sending actor.
I	<u>Immutable – when sending a submission, a value for the attribute may not be changed from its original value</u>

1080

Editor: Update Table 4.3.1-3 as shown add the new attributes (bold) in alphabetical order and new columns and values for RMU SD and RMU OD.

Table 4.3.1-3: Sending Actor Metadata Attribute Optionality

Metadata Element	Metadata Attribute	XDS DS	XDS DR	XDM MC	XDR DS	XDR MS	XDS OD	<u>RMU SD</u>	<u>RMU OD</u>
DocumentEntry	author	R2	R2	R2	R2	R2	R2	<u>R2</u>	<u>R2</u>
DocumentEntry	availabilityStatus	O	O	O	O	O	O	<u>I</u>	<u>I</u>
DocumentEntry	classCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
DocumentEntry	comments	O	O	O	O	O	O	<u>O</u>	<u>O</u>
DocumentEntry	confidentialityCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
DocumentEntry	creationTime	R	R	R2	R	R2	X	<u>R</u>	<u>X</u>
<u>DocumentEntry</u>	<u>documentAvailability</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>I</u>	<u>I</u>
DocumentEntry	entryUUID	R	R	R	R	R	R	<u>R</u>	<u>R</u>
DocumentEntry	eventCodeList	O	O	O	O	O	O	<u>O</u>	<u>O</u>
DocumentEntry	formatCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
DocumentEntry	hash	O	R	R	O	O	X	<u>I</u>	<u>X</u>
DocumentEntry	healthcareFacility TypeCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
DocumentEntry	homeCommunityId	O	O	O	O	O	O	<u>I</u>	<u>I</u>
DocumentEntry	languageCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
DocumentEntry	legalAuthenticator	O	O	O	O	O	O	<u>O</u>	<u>O</u>
DocumentEntry	limitedMetadata	X	X	O	X	R	X	<u>X</u>	<u>X</u>
<u>DocumentEntry</u>	<u>logicalID</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>I</u>	<u>I</u>
DocumentEntry	contentType	R	R	R	R	R	R	<u>R</u>	<u>R</u>
DocumentEntry	objectType	R	R	R	R	R	R	<u>I</u>	<u>I</u>
DocumentEntry	patientId	R	R	R2	R	R2	R	<u>I</u>	<u>I</u>
DocumentEntry	practiceSettingCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
DocumentEntry	referenceIdList	O	O	O	O	O	O	<u>O</u>	<u>O</u>
DocumentEntry	repositoryUniqueId	O	R	O	O	O	R	<u>I</u>	<u>I</u>
DocumentEntry	serviceStartTime	R2	R2	R2	R2	R2	O	<u>O</u>	<u>O</u>
DocumentEntry	serviceStopTime	R2	R2	R2	R2	R2	O	<u>O</u>	<u>O</u>
DocumentEntry	size	O	R	R	O	O	X	<u>I</u>	<u>X</u>
DocumentEntry	sourcePatientId	R	R	R2	R	R2	R	<u>I</u>	<u>I</u>

Metadata Element	Metadata Attribute	XDS DS	XDS DR	XDM MC	XDR DS	XDR MS	XDS OD	RMU SD	RMU OD
DocumentEntry	sourcePatientInfo	O	O	R2	O	R2	O	<u>O</u>	<u>O</u>
DocumentEntry	title	O	O	O	O	O	O	<u>O</u>	<u>O</u>
DocumentEntry	typeCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
DocumentEntry	uniqueId	R	R	R	R	R	R	<u>I</u>	<u>I</u>
DocumentEntry	URI	O	O	R	O	O	O	<u>O</u>	<u>O</u>
DocumentEntry	version	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>I</u>	<u>I</u>
SubmissionSet	author	R2	R2	R2	R2	R2	R2	<u>R2</u>	<u>R2</u>
SubmissionSet	availabilityStatus	O	O	O	O	O	O	<u>I</u>	<u>I</u>
SubmissionSet	comments	O	O	O	O	O	O	<u>O</u>	<u>O</u>
SubmissionSet	contentTypeCode	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
SubmissionSet	entryUUID	R	R	R	R	R	R	<u>R</u>	<u>R</u>
SubmissionSet	homeCommunityId	O	O	O	O	O	O	<u>O</u>	<u>O</u>
SubmissionSet	intendedRecipient	O	O	R2	R2	R2	O	<u>O</u>	<u>O</u>
SubmissionSet	limitedMetadata	X	X	O	X	R	X	<u>X</u>	<u>X</u>
SubmissionSet	patientId	R	R	R2	R	R2	R	<u>R</u>	<u>R</u>
SubmissionSet	sourceId	R	R	R	R	R	R	<u>R</u>	<u>R</u>
SubmissionSet	submissionTime	R	R	R	R	R	R	<u>R</u>	<u>R</u>
SubmissionSet	title	O	O	O	O	O	O	<u>O</u>	<u>O</u>
SubmissionSet	uniqueId	R	R	R	R	R	R	<u>R</u>	<u>R</u>
Folder	availabilityStatus	O	O	O	O	O	O	<u>X</u>	<u>X</u>
Folder	codeList	R	R	R2	R	R2	R	<u>X</u>	<u>X</u>
Folder	comments	O	O	O	O	O	O	<u>X</u>	<u>X</u>
Folder	entryUUID	R	R	R	R	R	R	<u>X</u>	<u>X</u>
Folder	homeCommunityId	O	O	O	O	O	O	<u>X</u>	<u>X</u>
Folder	lastUpdateTime	O	O	O	O	O	O	<u>X</u>	<u>X</u>
Folder	limitedMetadata	X	X	O	X	R	X	<u>X</u>	<u>X</u>
Folder	patientId	R	R	R2	R	R2	R	<u>X</u>	<u>X</u>
Folder	title	R	R	O	R	O	R	<u>X</u>	<u>X</u>
Folder	uniqueId	R	R	R	R	R	R	<u>X</u>	<u>X</u>

1085 4.3.2 Requirements on Query Type Transactions

4.3.2.1 Query Type Metadata Attribute Optionality

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

1090

Table 4.3.2.1-1: Responding Actor/Transaction Pairs

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

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

1095

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

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

Editor: Update Table 4.3.2.1-3 as shown adding the new attributes in alphabetical order.

Table 4.3.2.1-3: Responding Actor Metadata Attribute Optionality

Metadata Element	Metadata Attribute	XDS DR	XCA IG	XCA RG	MPQ DR
<u>DocumentEntry</u>	<u>documentAvailability</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>
<u>DocumentEntry</u>	<u>logicalID</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>
<u>DocumentEntry</u>	<u>version</u>	<u>O</u>	<u>O</u>	<u>O</u>	<u>O</u>