ACC, HIMSS and RSNA

Integrating the Healthcare Enterprise

IT Infrastructure Technical Framework

Volume 2 (ITI TF-2) Transactions

Revision 2.0 – Final Text

August 15, 2005

Contents

1 Introduction 3
1.1 Overview of the Technical Framework 3
1.2 Overview of IT Infrastructure Technical Framework Volume II 4
1.3 Audience 5
1.4 Relationship to Standards 5
1.5 Relationship to Real-world Architectures 5
1.6 Comments 6
1.7 Copyright Permission 6
2 Conventions 7
2.1 The Generic IHE Transaction Model 7
2.2 HL7 Profiling Conventions 8
2.3 Use of Coded Entities and Coding Schemes 8
3 IHE Transactions 9
3.1 Maintain Time 10
3.2 Get User Authentication 13
3.3 Get Service Ticket 16
3.4 Kerberized Communication 19
3.5 Join Context 24
3.6 Change Context 30
3.7 Leave Context 37
3.8 Patient Identity Feed 40
3.9 PIX Query 50
3.10 PIX Update Notification 60
3.11 Retrieve Specific Information for Display 64
3.12 Retrieve Document for Display 75
3.13 Follow Context 80
3.14 Register Document Set 86
3.15 Provide and Register Document Set 128
3.16 Query Registry 139
3.17 Retrieve Document 160
3.18 Intentionally Left Blank (ITI-18) 165
3.19 Authenticate Node 166
3.20 Record Audit Event 171
3.21 Patient Demographics Query 187
3.22 Patient Demographics and Visit Query 199
3.23 Find Personnel White Pages 212
3.24 Query Personnel White Pages 215
Appendix A: Web Service Definition for Retrieve Specific Information for Display and Retrieve Document for Display Transaction 229
Appendix B: Definition of Document Unique IDs 235
B.1: Requirements for Document UIDs 235
B.2: Structure of a Document UID 235
B.3: Document UID encoding rules 236
B.4: How to obtain a UID registration root? 236
B.5: Example of a Document UID 236

Appendix C: HL7 Profiling Conventions 238
C.1: HL7 Implementation Notes 239
Appendix D: Cross-Profile Interactions of PIX and PSA 242
D.1: Namespace Translation from PIX Query to CCOW 244
D.2: Processing Multiple Identifiers 245
Appendix E: Usage of the CX Data Type in PID-3-Patient Identifier List 246
E.1: Patient Identifier Cross-reference Manager actor requirements 247
E.2: Other actor requirements 248
E.3: E.3 Examples of use 249
E.3.1:Data sent by source systems 249
E.3.2:Data sent by the Patient Identifier Cross-reference Manager 250
Appendix F: Intentionally Left Blank 252
Appendix G: Transition from Radiology Basic Security to ATNA 253
G.1: Message Transformation 253
Appendix H: Required Registry Initialization and Schema 264
H.1: Initialization 264
H.2: Schema 264
H.3: Location 264
Appendix I: Required Initialization of the Affinity Domain 265
Appendix J: Example Submissions and Query Results 266
Appendix K: XDS Security Environment 267
K.1: Security Environment 267
K.1.1: Threats 267
K.1.2: Security and Privacy Policy 269
K.1.3: Security Usage Assumptions 270
K.2: Security Objectives 270
K.2.1: XDS Component Security Objectives 271
K.2.2: Environment Security Objectives 273
K.3: Functional Environment 273
Appendix L: Relationship of Document Entry Attributes and Document Headers 276
Appendix M: Using Patient Demographics Query in a Multi-Domain Environment 284
M.1: HL7 QBP^Q22 Conformance Model 284
M.2: IHE PDQ Architecture 284
M.3: Implementing PDQ in a multi-domain architecture 285
GLOSSARY 287

1 Introduction

Integrating the Healthcare Enterprise (IHE) is an initiative designed to stimulate the integration of the information systems that support modern healthcare institutions. Its fundamental objective is to ensure that in the care of patients all required information for medical decisions is both correct and available to healthcare professionals. The IHE initiative is both a process and a forum for encouraging integration efforts. It defines a technical framework for the implementation of established messaging standards to achieve specific clinical goals. It includes a rigorous testing process for the implementation of this framework. And it organizes educational sessions and exhibits at major meetings of medical professionals to demonstrate the benefits of this framework and encourage its adoption by industry and users.

The approach employed in the IHE initiative is to support the use of existing standards, e.g HL7, ASTM, DICOM, ISO, IETF, OASIS and others as appropriate, rather than to define new standards. IHE profiles further constrain configuration choices where necessary in these standards to ensure that they can be used in their respective domains in an integrated manner between different actors. When clarifications or extensions to existing standards are necessary, IHE refers recommendations to the relevant standards bodies.

This initiative has numerous sponsors and supporting organizations in different medical specialty domains and geographical regions. In North America the primary sponsors are the American College of Cardiology (ACC), the Healthcare Information and Management Systems Society (HIMSS) and the Radiological Society of North America (RSNA). IHE Canada has also been formed. IHE Europe (IHE-EUR) is supported by a large coalition of organizations including the European Association of Radiology (EAR) and European Congress of Radiologists (ECR), the Coordination Committee of the Radiological and Electromedical Industries (COCIR), Deutsche Röntgengesellschaft (DRG), the EuroPACS Association, Groupement pour la Modernisation du Système d'Information Hospitalier (GMSIH), Société Francaise de Radiologie (SFR), Società Italiana di Radiologia Medica (SIRM), and the European Institute for health Records (EuroRec). In Japan IHE-J is sponsored by the Ministry of Economy, Trade, and Industry (METI); the Ministry of Health, Labor, and Welfare; and MEDIS-DC; cooperating organizations include the Japan Industries Association of Radiological Systems (JIRA), the Japan Association of Healthcare Information Systems Industry (JAHIS), Japan Radiological Society (JRS), Japan Society of Radiological Technology (JSRT), and the Japan Association of Medical Informatics (JAMI). Other organizations representing healthcare professionals are invited to join in the expansion of the IHE process across disciplinary and geographic boundaries.

1.1 Overview of the Technical Framework

This document, the IHE IT Infrastructure Technical Framework (ITI TF), defines specific implementations of established standards to achieve integration goals that promote appropriate sharing of medical information to support optimal patient care. It is expanded annually, after a

period of public review, and maintained regularly through the identification and correction of errata. The current version, rev. 2.0 for Final Text, specifies the IHE transactions defined and implemented as of August 2005. The latest version of the document is always available via the Internet at http://www.ihe.net/Technical_Framework .

The IHE IT Infrastructure Technical Framework identifies a subset of the functional components of the healthcare enterprise, called IHE actors, and specifies their interactions in terms of a set of coordinated, standards-based transactions. It describes this body of transactions in progressively greater depth. The present volume (ITI TF-1) provides a high-level view of IHE functionality, showing the transactions organized into functional units called integration profiles that highlight their capacity to address specific IT Infrastructure requirements.

Volume 2 of the IT Infrastructure Technical Framework (ITI TF-2) provides detailed technical descriptions of each IHE transaction used in the IT Infrastructure Integration Profiles. These two volumes are consistent and can be used in conjunction with the Integration Profiles of other IHE domains.

The other domains within the IHE initiative also produce Technical Frameworks within their respective areas that together form the IHE Technical Framework. Currently, the following IHE Technical Framework(s) are available:

- IHE IT Infrastructure Technical Framework
- IHE Cardiology Technical Framework
- IHE Laboratory Technical Framework
- IHE Patient Care Coordination Technical Framework
- IHE Radiology Technical Framework

Where applicable, references are made to other technical frameworks. For the conventions on referencing other frameworks, see Section 1.6.3 within this volume.

1.2 Overview of IT Infrastructure Technical Framework Volume II

The remainder of Section 1 further describes the general nature, purpose and function of the Technical Framework. Section 2 presents the conventions used in this volume to define IHE transactions.

Section 3 defines transactions in detail, specifying the roles for each Actor, the standards employed, the information exchanged, and in some cases, implementation options for the transaction.

The appendices following the main body of this volume provide technical details associated with the transactions.

1.3 Audience

The intended audience of this document is:

- IT departments of healthcare institutions
- Technical staff of vendors planning to participate in the IHE initiative
- Experts involved in standards development
- Those interested in integrating healthcare information systems and workflows

1.4 Relationship to Standards

The IHE Technical Framework identifies functional components of a distributed healthcare environment (referred to as IHE actors), solely from the point of view of their interactions in the healthcare enterprise. At its current level of development, it defines a coordinated set of transactions based on ASTM, DICOM, HL7, IETF, ISO, OASIS and W3C standards. As the scope of the IHE initiative expands, transactions based on other standards may be included as required.

In some cases, IHE recommends selection of specific options supported by these standards; however, IHE does not introduce technical choices that contradict conformance to these standards. If errors in or extensions to existing standards are identified, IHE's policy is to report them to the appropriate standards bodies for resolution within their conformance and standards evolution strategy.

IHE is therefore an implementation framework, not a standard. Conformance claims for products must still be made in direct reference to specific standards. In addition, vendors who have implemented IHE integration capabilities in their products may publish IHE Integration Statements to communicate their products' capabilities. Vendors publishing IHE Integration Statements accept full responsibility for their content. By comparing the IHE Integration Statements from different products, a user familiar with the IHE concepts of actors and integration profiles can determine the level of integration between them. See Appendix C for the format of IHE Integration Statements.

1.5 Relationship to Real-world Architectures

The IHE actors and transactions described in the IHE Technical Framework are abstractions of the real-world healthcare information system environment. While some of the transactions are traditionally performed by specific product categories (e.g. HIS, Clinical Data Repository, Radiology Information Systems, Clinical Information Systems or Cardiology Information Systems), the IHE Technical Framework intentionally avoids associating functions or actors with such product categories. For each Actor, the IHE Technical Framework defines only those functions associated with integrating information systems. The IHE definition of an Actor should therefore not be taken as the complete definition of any product that might implement it, nor

should the framework itself be taken to comprehensively describe the architecture of a healthcare information system.

The reason for defining actors and transactions is to provide a basis for defining the interactions among functional components of the healthcare information system environment. In situations where a single physical product implements multiple functions, only the interfaces between the product and external functions in the environment are considered to be significant by the IHE initiative. Therefore, the IHE initiative takes no position as to the relative merits of an integrated environment based on a single, all-encompassing information system versus one based on multiple systems that together achieve the same end. IHE demonstrations emphasize the integration of multiple vendors' systems based on the IHE Technical Framework.

1.6 Comments

HIMSS and RSNA welcome comments on this document and the IHE initiative. They should be directed to the discussion server at http://ihe.rsna.org/ihetf/ or to:

Chris Carr
Director of Informatics
820 Jorie Boulevard
Oak Brook, IL 60523
Email: ihe@rsna.org

Joyce Sensmeier
Director of Professional Services
230 East Ohio St., Suite 500
Chicago, IL 60611
Email: ihe@himss.org

1.7 Copyright Permission

2 Conventions

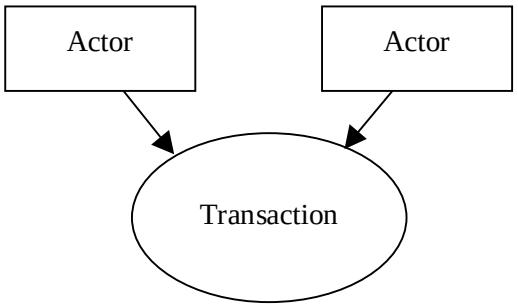
This document has adopted the following conventions for representing the framework concepts and specifying how the standards upon which the IHE IT Infrastructure Technical Framework is based should be applied.

2.1 The Generic IHE Transaction Model

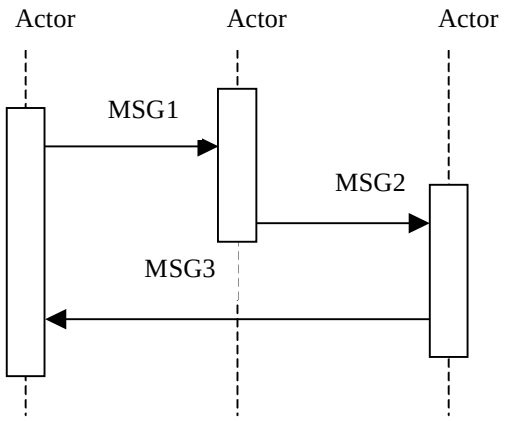
Transaction descriptions are provided in Section 3. In each transaction description, the actors, the roles they play, and the transactions between them are presented as use cases.

The generic IHE transaction description includes the following components:

- Scope: a brief description of the transaction.
- Use case roles: textual definitions of the actors and their roles, with a simple diagram relating them, e.g.:



- *Referenced Standards*: the standards (stating the specific parts, chapters or sections thereof) to be used for the transaction.
- *Interaction Diagram*: a graphical depiction of the actors and messages that support the transaction, with related processing within an Actor shown as a rectangle and time progressing downward, similar to:



The interaction diagrams used in the IHE IT Infrastructure Technical Framework are modeled after those described in Grady Booch, James Rumbaugh, and Ivar Jacobson, *The Unified Modeling Language User Guide*, ISBN 0-201-57168-4. Simple acknowledgment messages are often omitted from the diagrams for brevity. One or more messages may be required to satisfy a transaction. Each message is represented as an arrow starting from the Actor initiating the message.

- *Message definitions*: descriptions of each message involved in the transaction, the events that trigger the message, its semantics, and the actions that the message triggers in the receiver.

2.2 HL7 Profiling Conventions

See Appendix C in this volume for the HL7 profiling conventions as well as the networking implementation guidelines.

2.3 Use of Coded Entities and Coding Schemes

IHE does not produce, maintain or otherwise specify a coding scheme or other resource for controlled terminology (coded entities). Where applicable, coding schemes required by the HL7 and DICOM standards take precedence. In the cases where such resources are not explicitly identified by standards, implementations may utilize any resource (including proprietary or local) provided any licensing/copyright requirements are satisfied.

3 IHE Transactions

This section defines each IHE transaction in detail, specifying the standards used, the information transferred, and the conditions under which the transaction is required or optional.

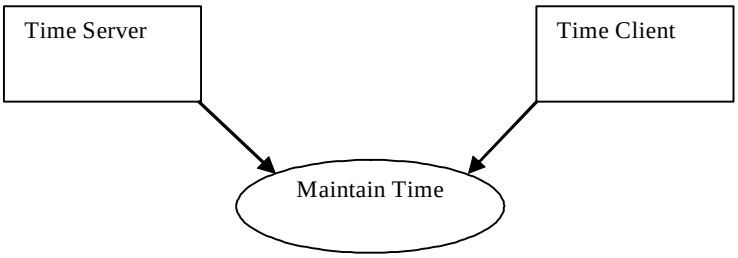
3.1 Maintain Time

This section corresponds to Transaction ITI-1 of the IHE IT Infrastructure Technical Framework. Transaction ITI-1 is used by the Time Server and Time Client actors.

3.1.1 Scope

This transaction is used to synchronize time among multiple systems.

3.1.2 Use Case Roles



Actor: Time Server

Role: Responds to NTP time service queries.

Actor: Time Client

Role: Uses NTP or SNTP time service responses to maintain synchronization with Time Servers and maintain the local system clock.

3.1.3 Referenced Standard

NTP	Network Time Protocol Version 3. RFC1305
SNTP	Simple Network Time Protocol (SNTP) RFC2030

3.1.4 Interaction Diagram

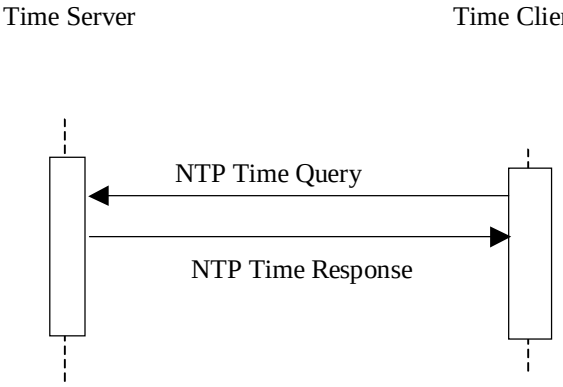


Figure 3.1.4-1. Maintain Time Messages

3.1.4.1 Maintain Time

The NTP transactions are described in detail in RFC1305. There is also extensive documentation on the transactions and recommendations on configurations and setup provided at http://www.ntp.org. Rather than reproduce all of that material as part of this Framework, readers are strongly encouraged to explore that site. The most common mode is the query-response mode that is described below. For other forms, see RFC1305 and the material on http://www.ntp.org.

The Time Server shall support NTP (which implicitly means that SNTP clients are also supported). Secure NTP may also be supported. The Time Client shall utilize NTP when it is grouped with a Time Server, or when high accuracy is required. For ungrouped Time Clients with 1 second accuracy requirements, SNTP may be useable. Time Clients may also support Secure NTP.

Table 3.1.4-1 Permissible Protocol Selections

Protocol	Time Server	Time Client grouped with a Time Server	Time Client (1s accuracy)	Time Client (High accuracy)
SNTP	Must Support	prohibited	permitted	prohibited
NTP	Must Support	Must Support	permitted	permitted
Secure NTP	Optional	Optional	Optional	Optional

3.1.4.1.1 Trigger Events

In a query-response mode the Time Client queries the Time Server and receives a response. This transaction includes timing estimation of network delays.

3.1.4.1.2 Message Semantics

The Time Client uses the Network Time Protocol (NTP) to synchronize its time with the Time Server. NTP clients can be configured to use a specific NTP server at a specific IP address, to obtain the NTP server address automatically from DHCP, and/or to discover the NTP server address automatically. Time clients shall support at least manual configuration and may support all three modes. Time Clients usually maintain time synchronization by adjusting the system clock, so that applications continue to use the system clock facilities. The specific precision of synchronization depends upon the requirements of specific actors.

Implementations must support a time synchronization accuracy of at least one second.

There is a Simple Network Time Protocol (SNTP) RFC2030 defined that can provide one second accuracy for Time Clients. It uses the exact same protocol as NTP, but does not include the measurement data used by the NTP high-accuracy statistical estimation algorithm. It has a lower implementation cost because it omits the measurements and statistical estimation needed to achieve higher accuracy. This omission of the statistical estimation makes it unsuitable for use when grouped with a Time Server. Its use is permitted for Time Clients that are not grouped with a Time Server and that do not need better synchronization for another reason.

Note: The Time Client Actor can often be implemented by using components provided by operating systems. Some offer only SNTP while others offer the choice of SNTP or NTP clients.

The use of Secure NTP is not required. The risk of subversion of the time base to conceal penetration is considered very low, and the operational costs of maintaining Secure NTP too high in most environments.

3.1.4.1.3 Expected Actions

The Time Server and Time Client will maintain synchronization to UTC. The Time Client maintains a statistical estimation process utilizing time estimates and network delay estimates from one or more Time Servers. This statistical estimation process yields a time estimate that is used to continually adjust the system clock.

Note: The relationship between the local reported time, UTC, and battery-backed clock is often a source of confusion. Different hardware and operating systems have different configuration requirements. These should be clearly documented and made clear in the user interface so that field service and operational staff do not introduce errors.

3.2 Get User Authentication

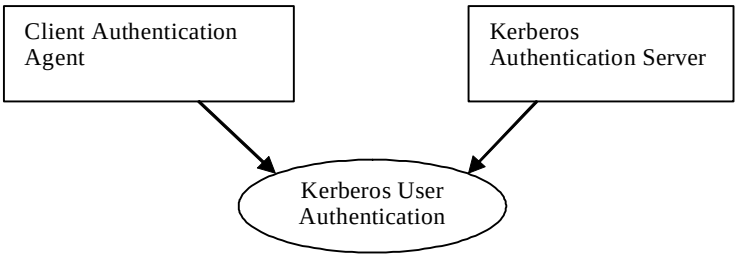
This section corresponds to Transaction ITI-2 of the IHE IT Infrastructure Technical Framework. Transaction ITI-2 is used by the Client Authentication Agent and Kerberos Authentication Server actors.

3.2.1 Scope

This transaction is used to authenticate an enterprise-wide user identity. A challenge-response method verifies that the user knows the correct password. Once the user is authenticated, the Kerberos Authentication Server sends a Ticket Granting Ticket (TGT) to the Client Authentication Agent to permit optimization of subsequent interactions. The TGT acts as a substitute for repeated login/password type activity.

This transaction is equivalent to what is called the "Authentication Service" in RFC1510.

3.2.2 Use Case Roles



Actor: Client Authentication Agent.

Role: Communicates authentication information to the Kerberos Authentication Server, receives a TGT, and performs internal TGT management.

Actor: Kerberos Authentication Server. In RFC1510 this is called a Key Distribution Center (KDC).

Role: Verifies the authentication information, creates a TGT, and sends it to the Client Authentication Agent.

3.2.3 Referenced Standard

RFC1510 The Kerberos Network Authentication Service (V5)

3.2.4 Interaction Diagram

The Client Authentication Agent communicates to the Kerberos Authentication Server a Kerberos Authentication Service Request (KRB_AS_REQ). This message identifies the user, the

name of the ticket-granting service and authentication data. The authentication data is usually a timestamp encrypted with the user's long-term key. (See RFC1510 for the exception cases.)

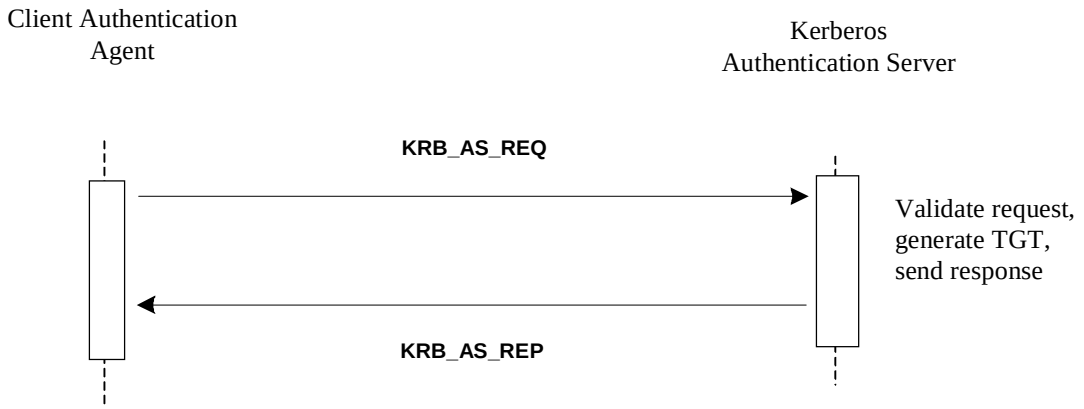


Figure 3.2.4-1. Get User Authentication Messages

3.2.4.1 Get User Authentication (Request/Response)

3.2.4.1.1 Trigger Events

The Kerberos User Authentication transactions normally take place:

1. Upon login or session start for a new user, and
2. Shortly before expiration of a TGT. TGT timeouts are selected to minimize the need for this transaction, but they may expire prior to user logout/ session complete.

When the Client Authentication Agent supports the Authentication for User Context Option, the Client Authentication Agent shall resolve any Context Manager interface issues before starting the user authentication. For instance the Client Authentication Agent needs to be sure that it will be accepted by the Context Manager as the one and only user authenticator in the context for this user session. Similar issues may apply with non-IHE uses of CCOW.

3.2.4.1.2 Message Semantics

The Client Authentication Agent shall support use of this transaction with the Kerberos user name/password system defined in RFC 1510. The username and password shall consist of the 94 printable characters specified in the International Reference Version of ISO-646/ECMA-6 (aka U.S. ASCII).

3.2.4.1.3 Expected Actions

The Client Authentication Agent shall perform TGT management, so that subsequent activities can re-use TGTs from a credentials cache. The Client Authentication Agent shall ensure that a user has access to only to his or her own tickets (both TGT and Service Tickets). This is most often done by clearing the credentials cache upon user logout or session completion.

When the Client Authentication Agent supports the Authenticator for User Context Option, the agent shall perform the Change Context Transaction to set the user identity in the context managed by the Context Manager Actor.

When the user session ends, the Client Authentication Agent shall remove the user credentials from its cache. If it supports the Authenticator for User Context Option, the agent shall perform the Change Context Transaction to set the user to NULL prior to removing the user credentials.

3.2.5 Extended Authentication Methods

The Kerberos challenge-response system used by this Integration Profile can be used to verify users by means of many authentication mechanisms. The mechanism specified in this profile is the Kerberos username and password system. Other methods such as smart cards and biometrics have also been documented but not standardized. (See ITI TF-1: Appendix D for a discussion of alternate authentication mechanisms.)

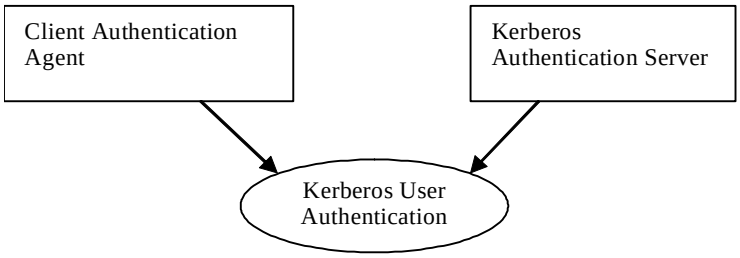
3.3 Get Service Ticket

This section corresponds to Transaction ITI-3 of the IHE IT Infrastructure Technical Framework. Transaction ITI-3 is used by the Client Authentication Agent and Kerberos Authentication Server Actors.

3.3.1 Scope

The Client Authentication Agent uses this transaction to obtain the service ticket that will be sent to a Kerberized Server to authenticate this user to a Kerberized Server.

3.3.2 Use Case Roles



Actor: Client Authentication Agent.

Role: Client communicates authentication information to the Kerberos Authentication Server, receives a Service Ticket, and performs internal ticket management.

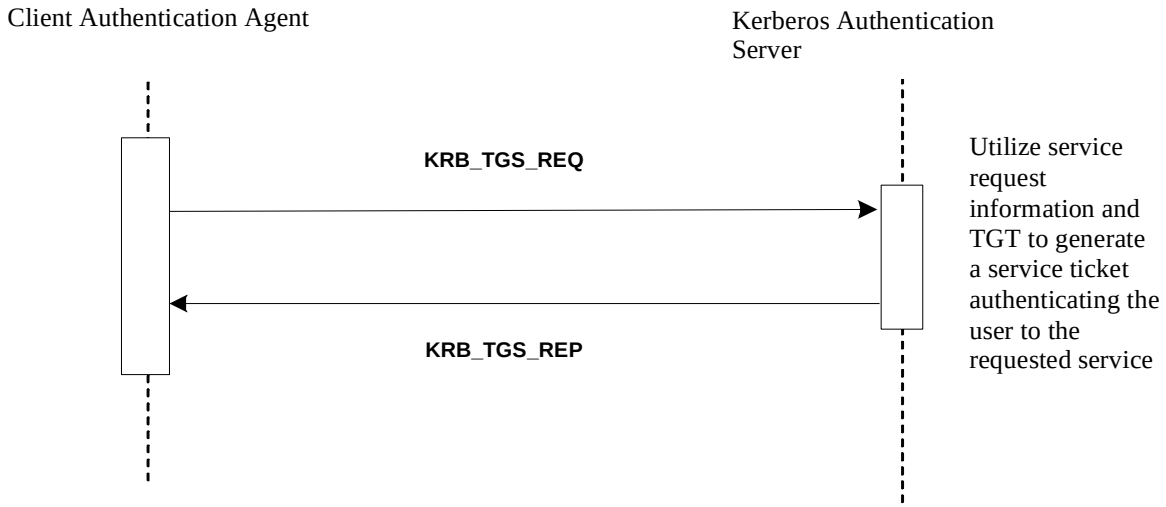
Actor: Kerberos Authentication Server. In RFC1510 this is called a Key Distribution Center (KDC).

Role: Verifies the authentication information, creates a ticket, and sends it to the Client Authentication Agent Actor.

3.3.3 Referenced Standard

RFC1510 The Kerberos Network Authentication Service (V5)

3.3.4 Interaction Diagram



3.3.4.1 Kerberos Service Ticket

3.3.4.1.1 Trigger Events

A service ticket is requested prior to communicating with a Kerberized Server. This ticket will be provided to that service as part of the Kerberized communication process.

3.3.4.1.2 Message Semantics

The Client Authentication Agent Actor requests credentials for a service by sending the Kerberos Authentication Server a Kerberos Ticket-Granting Service Request (KRB_TGS_REQ). This message includes the user's name, an authenticator encrypted with the user's logon session key, the TGT obtained in the Get User Authentication Transaction, and the name of the service for which the user wants a ticket.

When the Kerberos Authentication Server receives KRB_TGS_REQ, it decrypts the TGT with its own secret key, extracting the logon session key. It uses the logon session key to decrypt the authenticator and evaluates that. If the authenticator passes the test, the Kerberos Authentication Server extracts the authorization data from the TGT and invents a session key for the client to share with the Kerberized Server Actor that supports the service. The Kerberos Authentication Server encrypts one copy of this session key with the user's logon session key. It embeds another copy of the session key in a ticket, along with the authorization data, and encrypts this ticket with the service's long-term key. The Kerberos Authentication Server then sends these credentials back to the client in a Kerberos Ticket-Granting Service Reply (KRB_TGS_REP).

There are no IHE specific extensions or modifications to the Kerberos messaging.

3.3.4.1.3 Expected Actions

When the Client Authentication Agent receives the reply, it uses the logon session key to decrypt the session key to use with the service, and stores the key in its credentials cache. Then it extracts the ticket for the service and stores that in its cache.

The client shall maintain the ticket in the credentials cache for later use.

3.3.4.1.4 Service Registration

The Kerberized Communication services supported in an enterprise shall be registered on the Kerberos Authentication Server according to the RFC1510 protocol specification used. The registration of the service on the KDC is outside the scope of this profile.

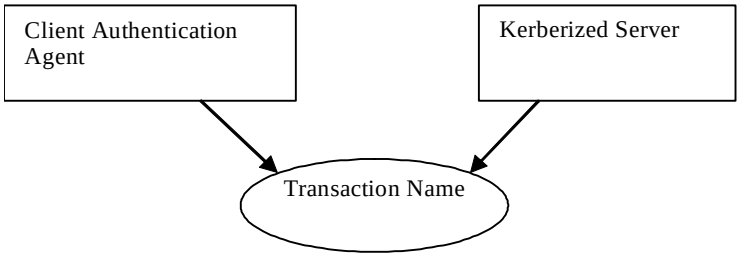
3.4 Kerberized Communication

This section corresponds to Transaction ITI-4 of the IHE IT Infrastructure Technical Framework. Transaction ITI-4 is used by the Client Authentication Agent and Kerberized Server Actors.

3.4.1 Scope

This section specifies the details of the association of a Kerberos user identity with a session for a session oriented protocol, or a transaction for a transaction oriented protocol.

3.4.2 Use Case Roles



Actor: Client Authentication Agent

Role: Provides appropriate ticket as part of the connection or session management for another protocol.

Actor: Kerberized Server

Role: Accepts and verifies the ticket to perform user-identity-related services as part of the connection or session management for another protocol.

3.4.3 Referenced Standard

RFC1510 The Kerberos Network Authentication Service (V5)

3.4.4 Interaction Diagram

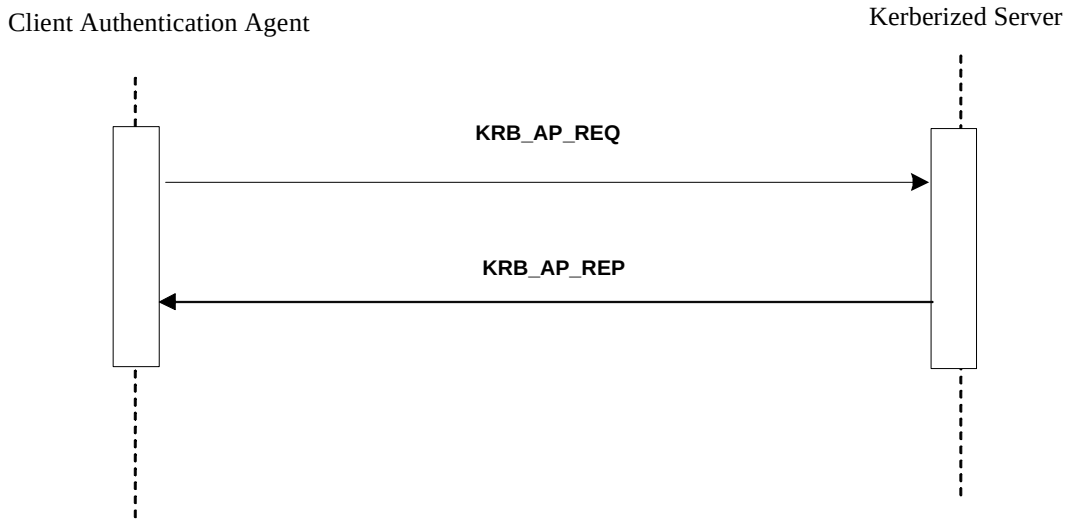


Figure 3.4-1 Kerberized Communications

3.4.4.1 Kerberized Communications

The sequence diagram above describes information flow that can be encapsulated in a variety of different protocol startup sequences. The specific details for this encapsulation are defined as part of the definition of Kerberizing a specific kind of communication protocol.

3.4.4.1.1 Trigger Events

This occurs at the beginning of a session or as part of each session-less transaction.

3.4.4.1.2 Message Semantics

The Client Authentication Agent Actor requests service from a Kerberized Server by sending the server a Kerberos Application Request (KRB_AP_REQ). This message contains an authenticator encrypted with the session key, the ticket obtained in the Get Service Ticket Transaction, and a flag indicating whether the client wants mutual authentication. (The setting of this flag is either specified by the rules of the Kerberized communications, or is an option of the specific Kerberized protocol.)

The Kerberized Server receives KRB_AP_REQ, decrypts the ticket, and extracts the authorization data and the session key. The server uses the session key to decrypt the authenticator and then evaluates the timestamp inside. If the authenticator passes the test, the server looks for a mutual authentication flag in the client's request for protocols that support mutual authentication. If the flag is set, the server uses the session key to encrypt the time

supplied by the Client Authentication Actor and returns the result in a Kerberos Application Reply (KRB_AP_REP).

The actual encoding and exchange of the KRB_AP_REQ and KRB_AP_REP are defined as part of the definition of the specific Kerberized protocol.

3.4.4.1.3 Expected Actions

When the Client Authentication Actor receives KRB_AP_REP, it decrypts the server's authenticator with the session key it shares with the server and compares the time returned by the service with the time in the client's original authenticator. If the times match, the client knows that the service is genuine, and the connection proceeds.

If no mutual authentication is requested, the other IHE actors proceed with their IHE transactions. These transactions are identified as being requested by the authenticated user. The other actors will utilize this information for other purposes, such as confirming user authorization or logging user actions into audit trails.

3.4.4.2 Kerberized HTTP

Kerberized HTTP shall use SPNEGO-HTTP
(see http://www.ietf.org/internet-drafts/draft-brezak-spnego-http-04.txt)

Note: At the time of publication there were no Kerberized HTTP normative standards. There are three relatively well-documented non-normative specifications. In addition, there are commercial and open source implementations of this specification for web and application servers. It was decided to use the Kerberized HTTP specification that is implemented by Microsoft Internet Explorer (MSIE) because many healthcare desktops use MSIE.

The following Figure shows a typical message sequence for Kerberized HTTP.

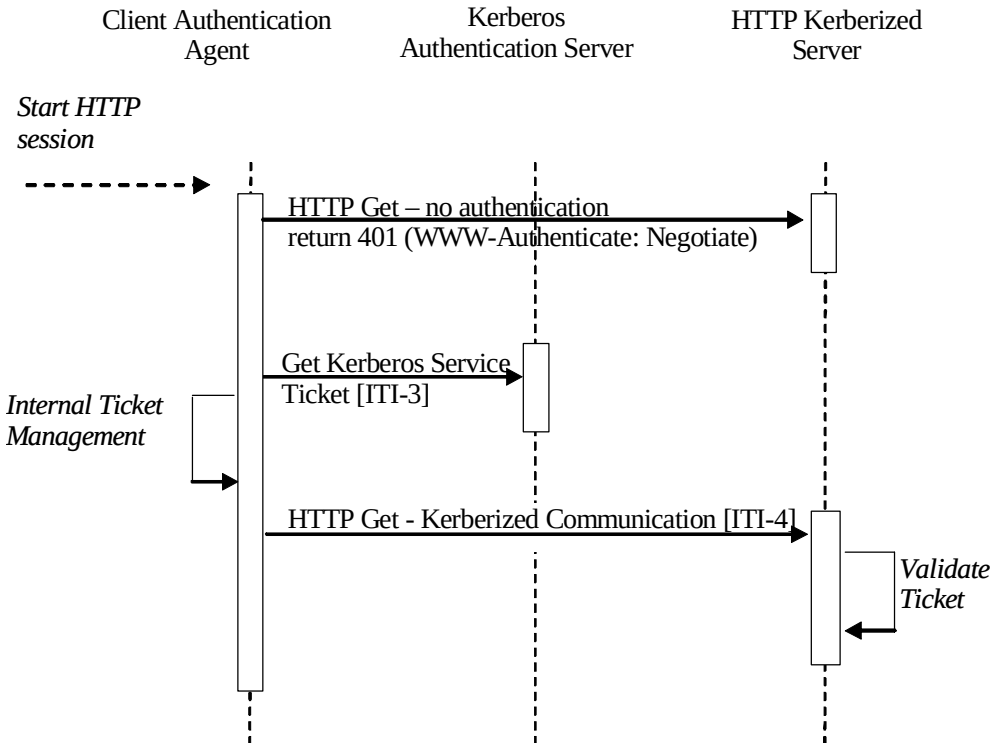


Figure 3.4-2 Kerberized HTTP

There is also documentation on the transactions, configuration, and troubleshooting these configurations. Rather than reproduce all of that material as part of this Framework, readers are strongly encouraged to explore these references.
(See http://support.microsoft.com/default.aspx?scid=kb;ben-us;326985)

3.4.4.2.1 Trigger Events

This transaction occurs at the beginning of each HTTP transaction.

Note: When the workstation is properly configured utilizing Microsoft Internet Explorer these transactions are transparent. A prompt for username, password, and domain is an indication of an improperly configured component.

3.4.4.2.2 Message Semantics

This IHE profile recognizes that the SPNEGO-HTTP method allows the client side to return Kerberos credentials or NTLM credentials. This IHE profile thus restricts the transactions to the Kerberized credentials.

3.4.4.3 Kerberized DICOM

The Kerberization of DICOM has been proposed and is under development. There is not a finished standard at this time.

3.4.4.4 Kerberized HL7

The Kerberization of HL7 has been proposed and is under development. There is not a finished standard at this time.

3.5 Join Context

This section corresponds to Transaction ITI-5 of the IHE IT Infrastructure Technical Framework. Transaction ITI-5 is used by the Patient Context Participant, User Context Participant, Client Authentication Agent and Context Manager Actors.

3.5.1 Scope

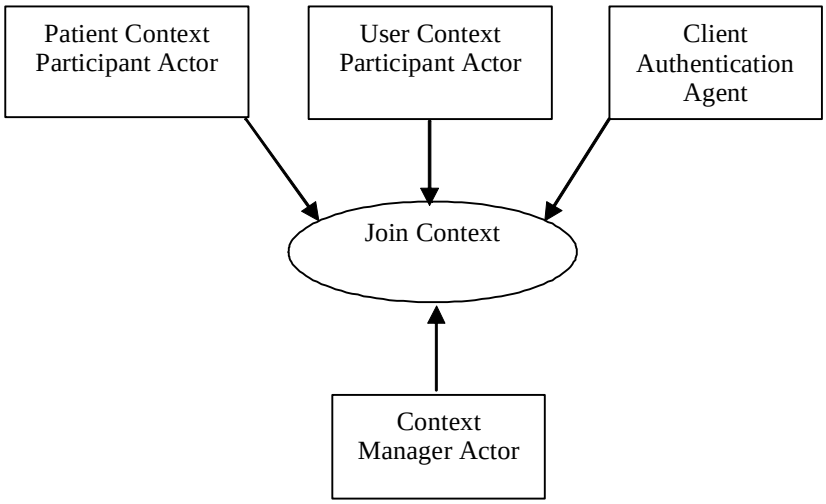
Any of the context participant actors using this Transaction (Patient Context Participant, User Context Participant, and Client Authentication Agent) may locate and join a context management session specific to the workstation on which the instigating user is interacting.

A Context Participant Actor shall first locate the instance of the Context Manager Actor via technology specific methods as defined in the *HL7 Context Management "CCOW"* technology mapping documents. Once the context manager reference is returned, the Context Participant Actor issues a join method to the context manager, which returns a unique participant identifier. User Context Participant and Client Authentication Agent shall use this identifier along with a shared secret as inputs to a two stage secure binding process, which results in the exchange of public keys between the two actors.

If an implementation groups two or more context participant actors, this Transaction shall be performed only once on a launch of an application in which those actors are grouped. All grouped actors share the same common context. If at least one of the grouped actors is a User Context Participant or a Client Authentication Agent, this transaction shall include the two-stage secure binding process.

The semantics of the methods used in this Transaction are defined in the documents HL7 Context Management "CCOW" Standard: Component Technology Mapping: ActiveX or HL7 Context Management "CCOW" Standard: Component Technology Mapping: Web. A Context Participant Actor can implement either technology. The Context Manager Actor shall support both technologies in order to interoperate with joining participants implementing the technology of their choice.

3.5.2 Use Case Roles



Actor: Patient Context Participant

Role: Initiates establishment of context session connection with the Context Manager so as to be able to change and follow Patient Subject changes in the common context.

Actor: User Context Participant

Role: Initiates establishment of a secure context session connection with the Context Manager so as to be able to follow User Subject changes in the common context.

Actor: Client Authentication Agent

Role: Initiates establishment of a secure context session connection with the Context Manager so as to be able to perform User Subject changes in the common context.

Actor: Context Manager

Role: Responds to the request to join the context session from the context participant.

3.5.3 Referenced Standard

HL7 Context Management "CCOW" Standard, Version 1.4:

Technology and Subject Independent Architecture

Component Technology Mapping: ActiveX

Component Technology Mapping: Web

3.5.4 Interaction Diagrams

The Join Context Transaction involves a different set of messages depending on the type of subjects the context participant is interested in, either Patient subject, User subject or both Patient and User subjects.

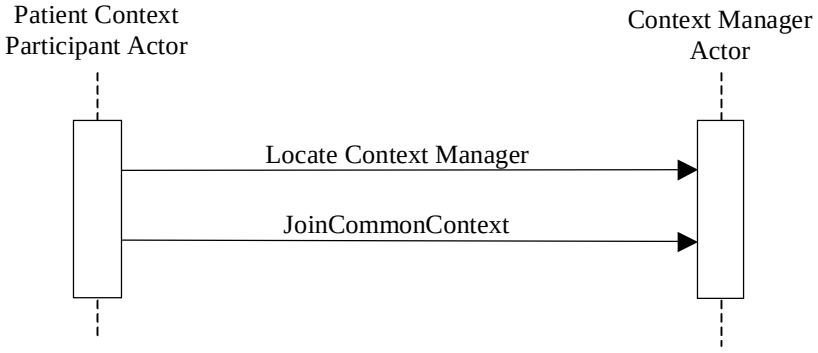


Figure 3.5-1 Patient Subject Join Context Interaction Diagram

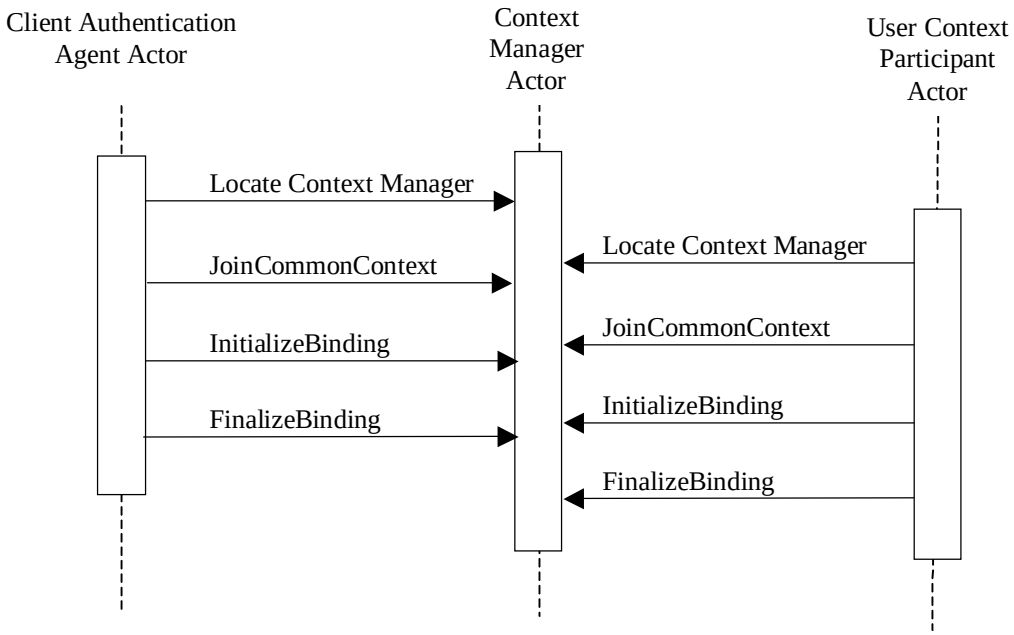


Figure 3.5-2 User Subject Join Context Interaction Diagram

3.5.4.1 Join Context – Locate Method

To join the common context upon launch of an application, it is necessary for the context participant to locate the Context Manager that supports context management for the user's

workstation. This is achieved by the invocation of the Locate method in accordance with specifications of the *HL7 Context Management "CCOW" Standard*.

3.5.4.1.1 Trigger Events

The Locate method is triggered by the user launch of an application that contains one of the following actors: Patient Context Participant, User Context Participant or Client Authentication Agent.

3.5.4.1.2 Message Semantics

In a Web/HTTP implementation, Locate is defined as a method of the ContextManagementRegistry interface. The IHE Context Manager Actor provides this interface for the context participants to call upon, and thus implements the CCOW defined Context Management Registry, which is used to locate the appropriate instance of the Context Manager.

In an ActiveX implementation, the context participants determine the location of the instance of Context Manager from the operating system registry.

3.5.4.1.3 Expected Actions

The Locate method invocation is specific to the Web technology mapping. In this case, the Content Manager shall return the valid URL of the Context Manager instance or a CCOW defined UnableToLocate exception. Refer to the *HL7 Context Management "CCOW" Standard: Component Technology Mapping: Web/HTTP*, Chapter 3 for the details of the response specifications.

3.5.4.2 Join Context – JoinCommonContext Method

The JoinCommonContext method is invoked by the one of the following actors: Patient Context Participant, User Context Participant or Client Authentication Agent.

3.5.4.2.1 Trigger Events

The JoinCommonContext method is triggered by the valid response of the Locate method with a reference to the context manager.

3.5.4.2.2 Message Semantics

JoinCommonContext is defined as a method on the ContextManager interface. It shall be invoked by a Context Participant Actor to complete the establishment of the secure context session. A Context Participant Actor shall provide parameters for this method as specified in the CCOW Standard.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.3, for a detailed description of the parameters associated with this method.

3.5.4.2.3 Expected Actions

If the JoinCommonContext method is successful, the Context Manager shall issue the invoking Actor a unique context participant identifier which is to be used until the context session is terminated by either a Context Participant Actor or the Context Manager Actor.

If the method fails a descriptive CCOW exception will be returned.

After the context session is established, the Context Manager Actor shall periodically verify availability of a Context Participant Actor by invoking the Ping method on the ContextParticipant interface as specified in the CCOW Standard. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.7.6, for a detailed description of the parameters associated with this method.

Should the Context Manager Actor need to terminate an established context session (for example, in a case of restart), it shall inform the context participants of such action by invocation of the CommonContextTerminated method on the ContextParticipant interface as specified in the CCOW Standard. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.7.5, for a detailed description of the parameters associated with this method.

The success of this method signifies completion of the Join Context Transaction for the actors intending to participate only in the patient context.

3.5.4.3 Join Context – InitializeBinding Method

The InitializeBinding method is invoked by the one of the following actors intending to participate in a user context: User Context Participant or Client Authentication Agent.

3.5.4.3.1 Trigger Events

The InitializeBinding method is triggered by the valid response of the JoinContext method.

3.5.4.3.2 Message Semantics

InitializeBinding is defined as a method on the SecureBinding interface and allows a Context Participant Actor and Context Manager to verify each other's identity and supply the Context Manager's public key to the requesting context participant.

In the invocation of this method, context participant supplies the application identification and a digest produced from that identification concatenated with a shared secret. The shared secret is known in CCOW terms as an applications passcode. The passcode shall be site configurable.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.12.2, for a description of the parameters associated with this method, to be issued by the Context Participant Actor.

3.5.4.3.3 Expected Actions

Performing the InitializeBinding method, the Context Manager verifies the identity of a requesting context participant and responds with the message containing its public key. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.12.2, for the specifics of the response formation.

3.5.4.4 Join Context – FinalizeBinding Method

The FinalizeBinding method is invoked by the one of the following actors: User Context Participant or Client Authentication Agent.

3.5.4.4.1 Trigger Events

The FinalizeBinding method is triggered by the valid response of the InitializeBinding method.

3.5.4.4.2 Message Semantics

FinalizeBinding is defined as a method on the SecureBinding interface and allows a Context Participant Actor to supply the Context Manager with its public key.

In the invocation of this method, the context participant supplies its public key and a digest digitally signed with its private key.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.12.3, for a description of the parameters associated with this method, to be issued by the Context Participant Actor.

3.5.4.4.3 Expected Actions

Performing the FinalizeBinding method, the Context Manager verifies the identity of a requesting context participant and accepts or rejects its public key. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.12.3, for the specifics of the response formation.

The success of this method signifies completion of the Join Context Transaction for the actors intending to participate in the user context.

3.6 Change Context

This section corresponds to Transaction ITI-6 of the IHE IT Infrastructure Technical Framework. Transaction ITI-6 is used by the Context Participant and Context Manager actors.

3.6.1 Scope

This transaction allows for an application supporting the Context Participant Actor to change the values for one or more context subjects, forcing other Context Participant actors to synchronize based on the new context values.

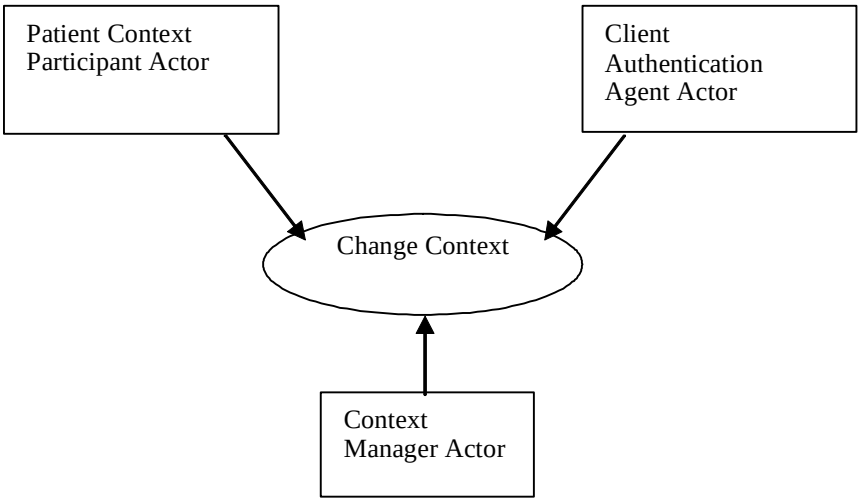
The Change Context Transaction is composed of multiple methods as defined by the *HL7 Context Management "CCOW" Standard*. There are two key characteristics to this transaction. The first is that the transaction has multiple phases consisting of instigating the change, surveying the other participants, and finally publishing the decision as to whether the context changed or not. The second characteristic is that the context change involves a specific subject. For the Patient Context Participant Actor the subject being changed is the patient subject. For the Client Authentication Agent Actor the subject being changed is the user subject. Applications that implement only the Patient Context Participant Actor shall not expect the user subject to be set in context.

The semantics of the methods used are defined in the documents HL7 Context Management "CCOW" Standard: Component Technology Mapping: ActiveX or HL7 Context Management "CCOW" Standard: Component Technology Mapping: Web, in conjunction with the HL7 Context Management "CCOW" Standard: Subject Data Definitions document. The Context Participant Actor can choose the technology implementation it wishes to implement. The Context Manager Actor must support both technology implementations in order to accommodate whichever implementation a participant ends up choosing.

In the case where Patient Context Participant Actors use identifiers from different patient identifier domains the Context Manager Actor shall be grouped with the Patient Identifier Cross-reference Consumer Actor and the corresponding PIX Query Transaction as defined in ITI TF-2: 3.9 to retrieve all identifiers the patient is known by. The IHE Context Manager Actor encompasses more than a CCOW context manager function. See ITI TF-2: Appendix D for a complete discussion of the grouping of these two actors.

The CCOW architecture is defined as a set of components that implement defined interfaces and their detailed methods as specified in the *HL7 Context Management "CCOW" Standard: Technology Independent Architecture* document. This structure is different than the traditional IHE network transaction. As is depicted in the interaction diagram in Section 3.6.4, the IHE Change Context Transaction is composed of multiple CCOW-defined methods.

3.6.2 Use Case Roles



Actor: Client Authentication Agent

Role: Initiates context change for user subject by supplying new context values.

Actor: Patient Context Participant

Role: Initiates context change for patient subject by supplying new context values. After receiving the context survey results it finalizes context change decision. Applications containing this Actor without a patient lookup function would not use this transaction.

Actor: Context Manager

Role: Manages Change Context Transaction lifecycle.

3.6.3 Referenced Standard

HL7 Context Management "CCOW" Standard, Version 1.4:

Technology and Subject Independent Architecture

Component Technology Mapping: ActiveX

Component Technology Mapping: Web

Subject Data Definitions

3.6.4 Interaction Diagram

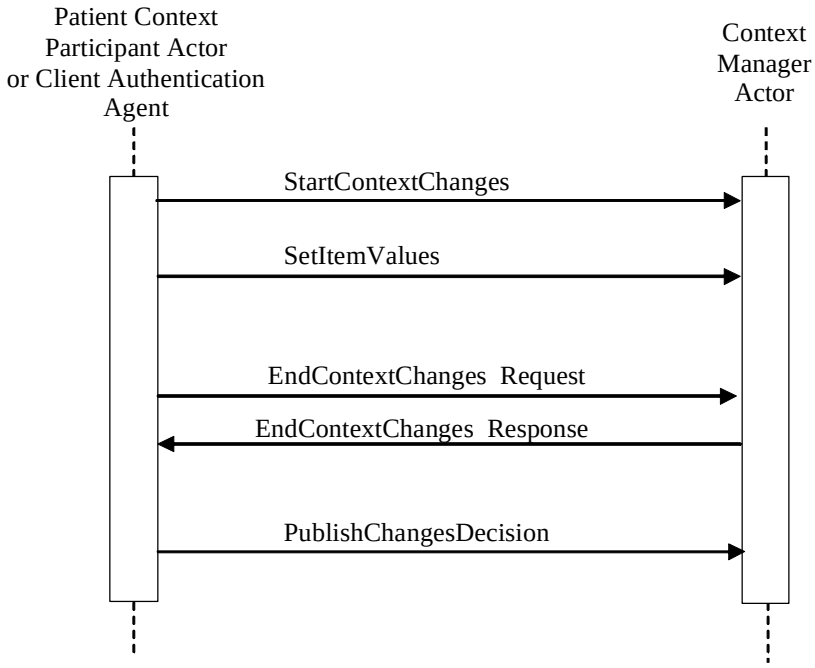


Figure 3.6-1 Change Context sequence

3.6.4.1 Context Change – StartContextChanges Method

3.6.4.1.1 Trigger Events

This method is triggered by a specific user gesture. The user gesture that triggers this transaction in for the Patient Context Participant Actor is one of selecting a patient. The user gesture that triggers this transaction for the Client Authentication Agent Actor is authentication of a user.

3.6.4.1.2 Message Semantics

The Patient Context Participant and/or the Client Authentication Agent Actor will issue a StartContextChanges method of the ContextManager interface. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.5, for a more detailed description of the parameters associated with this method. IHE specifies no restrictions or extensions to the CCOW definition of the StartContextChanges method.

3.6.4.1.3 Expected Actions

The Context Manager Actor returns the pending context coupon. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document,

Section 17.3.6.5, for a more detailed description of the response issued by the Context Manager Actor. IHE specifies no restrictions or extensions to the CCOW definition of the StartContextChanges method.

3.6.4.2 Change Context – SetItemValues Method

3.6.4.2.1 Trigger Events

The SetItemValues method is triggered by the return of a context coupon in response to the StartContextChanges method.

3.6.4.2.2 Message Semantics

3.6.4.2.2.1 Patient Context Participant Actor support for CCOW Patient Subject

The Patient Context Participant Actor issues an invocation of the SetItemValues method of the ContextData interface to the Context Manager Actor. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.4.4, for a more detailed description of the parameters associated with this method, to be issued by the Patient Context Participant Actor. The Patient Context Participant Actor supports synchronization around the CCOW patient subject. A Patient Context Participant Actor performing a Change Context Transaction shall set the Patient.Id.IdList.1 patient identifier item. All other patient identifier items as defined by the CCOW standard and shown in Table 3.6.4.2-1 Patient Subject Identifier Items, are subject to deprecation in future releases of the standard.

Table 3.6.4.2-1 Patient Subject Identifier Items

Patient Subject Identifier Item Name	HL7 Meaning	HL7 Data Type	HL7 Semantic Constraints on Values	Case Sensitive
Patient.Id.MRN.*Suffix*	Patient's medical record number, per PID-2	ST	HL7 Table 0203Identifier Type = MR	No
Patient.Id.MPI	Patient's identifier in the "Master Patient Index", per PID-2	ST	HL7 Table 0203Identifier Type = PT or PI (as agreed upon by context sharing systems) and Assigning Authority represents the MPI system	No
Patient.Id.NationalIdNumber	Patient's national identifier number, per PID-2	ST	HL7 Table 0203Identifier Type = PT and Assigning Authority represents agreed-upon National Authority	No
Patient.Id.IdList	A list of patient identifiers for a patient, per PID-3	CX	May be a repeating set of CX item values each of which contains an identifier that denotes the same patient	No

Adapted from the HL7 Context Management "CCOW" Standard, version 1.4

The Patient.Id.IdList.1 item shall populate component 1, (the patient identifier), and either sub-component 1, (namespace ID), of component 4, (the assigning authority), of the CX data item. This is to be consistent with the requirements for the patient identifier as defined in the PIX Query transaction documented in ITI TF-2: 3.9.4.1.2.2.

The Patient Context Participant Actor should use the SetItemValues associated with the ContextData interface, as defined in Sections 17.3.4.4 and 17.3.4.5 respectively of the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document.

3.6.4.2.2.2 Client Authentication Agent Actor support for CCOW User Subject

- The Client Authentication Agent Actor supports synchronization around the CCOW user subject. A Client Authentication Agent Actor performing a Change Context Transaction shall set the User.Id.Logon.*Suffix* identifier item, where the *Suffix* is assigned as Kerberos. This would make the item name to be used by the Client Authentication Agent Actor User.Id.Logon.Kerberos. The value of User.Id.Kerberos shall be the username@realm.

The Client Authentication Agent Actor shall use the SetItemValues associated with SecureContextData interface as defined in Section 17.3.13.3 of the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document.

3.6.4.2.3 Expected Actions

The Context Manager Actor returns an acknowledgement of the changed data. IHE specifies no restrictions or extensions to the CCOW definition of the SetItemValues method. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.4.4, for a more detailed description of the response issued by the Context Manager Actor to the Patient Context Participant Actor. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.13.3, for a more detailed description of the response issued by the Context Manager Actor to the Client Authentication Agent Actor.

3.6.4.3 Context Change – EndContextChanges

3.6.4.3.1 Trigger Events

The EndContextChanges method is triggered by the completion of the SetItemValues method.

3.6.4.3.2 Message Semantics

The Patient Context Participant and Client Authentication Agent Actors issue an EndContextChanges method of the ContextManager interface to the Context Manager Actor. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.6, for a description of the parameters

associated with this method. IHE specifies no restrictions or extensions to the CCOW definition of the EndContextChanges method.

3.6.4.3.3 Expected Actions

The EndContextChanges method triggers the ContextChangesPending method as defined in ITI TF-2: 3.13.4.1. The Context Manager Actor returns the results of the context survey to the instigating Patient Context Participant or Client Authentication Agent Actor.

If the instigating Patient Context Participant or Client Authentication Agent Actor receives a unanimous acceptance in the survey results, then it triggers an accept in the PublishChangesDecision method.

If the instigating Patient Context Participant or Client Authentication Agent Actor receives one or more Conditional Accept responses in the survey results, then the application containing the Actor must ask the user to continue, suspend context participation, or cancel the pending context change transaction. The user's decision to continue will result in the context change being accepted. The user's decision to suspend context participation will cancel the change transaction and allow the user to temporarily use the application without affecting the current context session. The user's decision to cancel will cancel the pending context change transaction. At this point the Patient Context Participant or Client Authentication Agent Actor triggers the PublishChangesDecision with the user's response.

In the event a participant application does not respond to the survey, after a configurable period of time the Context Manager Actor will deem the application as "busy". If the instigating participant application receives one or more busy responses, it shall only present the suspend or cancel choices. This prevents an application from inadvertently becoming out of synch with the context, unbeknownst to the user.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.6, for a more detailed description of the response issued by the Context Manager Actor and actions required by the Patient Context Participant and or Client Authentication Agent Actors. IHE specifies no restrictions or extensions to the CCOW definition of the EndContextChanges method.

3.6.4.4 Context Change – PublishChangesDecision

3.6.4.4.1 Trigger Events

The PublishChangesDecision method is triggered by the return of EndContextChanges method.

3.6.4.4.2 Message Semantics

The Patient Context Participant and Client Authentication Agent Actors shall issue either an accept or cancel via the PublishChangesDecision method of the ContextManager interface to the

Context Manager Actor. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.8, for a more detailed description of the parameters associated with this method. IHE specifies no restrictions or extensions to the CCOW definition of the PublishChangesDecision method.

3.6.4.4.3 Expected Actions

When the PublishChangesDecision method is received by the Context Manager Actor it triggers the ContextChangesAccepted or ContextChangesCancelled method as defined in ITI TF-2: 3.13.4.2 or ITI TF-2: 3.13.4.3 respectively. IHE specifies no restrictions or extensions to the CCOW definition of the PublishChangesDecision method. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.8, for a description of the response issued by the Context Manager Actor.

3.7 Leave Context

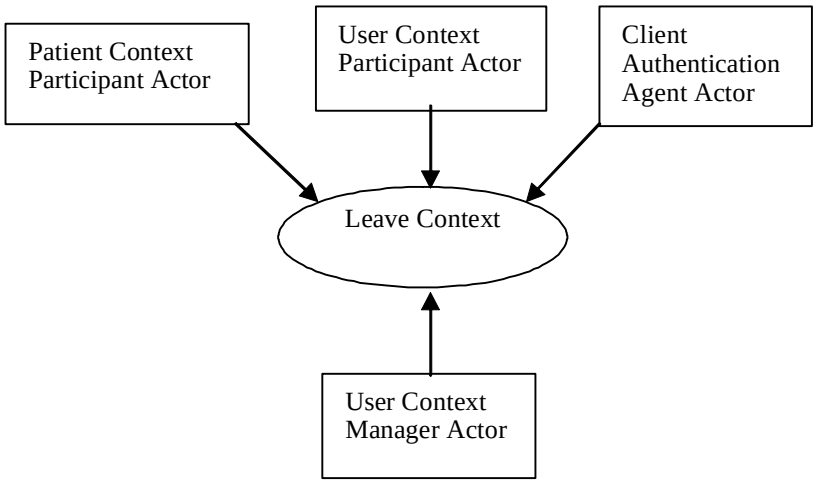
This section corresponds to Transaction ITI-7 of the IHE IT Infrastructure Technical Framework. Transaction ITI-7 is used by the Patient Context Participant, User Context Participant, Client Authentication Agent, and Context Manager Actors.

3.7.1 Scope

This transaction allows for an application supporting the Patient Context Participant, User Context Participant, or Client Authentication Agent Actor to terminate participation in a context management session in which it is participating.

A Context Participant Actor notifies the Context Manager Actor that is leaving the common context. The semantics of the methods used are defined in the documents *HL7 Context Management "CCOW" Standard: Component Technology Mapping: ActiveX* or *HL7 Context Management "CCOW" Standard: Component Technology Mapping: Web*. The Context Participant Actor can choose the technology implementation it wishes to implement. The Context Manager Actor must support both technology implementations in order to accommodate whichever implementation a joining participant ends up choosing.

3.7.2 Use Case Roles



Actor: Patient Context Participant

Role: Initiates notification to the Context Manager that it will no longer be participating in the context management session.

Actor: User Context Participant

Role: Initiates notification to the Context Manager that it will no longer be participating in the context management session.

Actor: Client Authentication Agent

Role: Initiates notification to the Context Manager that it will no longer be participating in the context management session.

Actor: Context Manager

Role: Responds to the request to leave the context session from the context participant.

3.7.3 Referenced Standard

HL7 Context Management "CCOW" Standard, Version 1.4:

Technology and Subject Independent Architecture

Component Technology Mapping: ActiveX

Component Technology Mapping: Web

3.7.4 Interaction Diagram

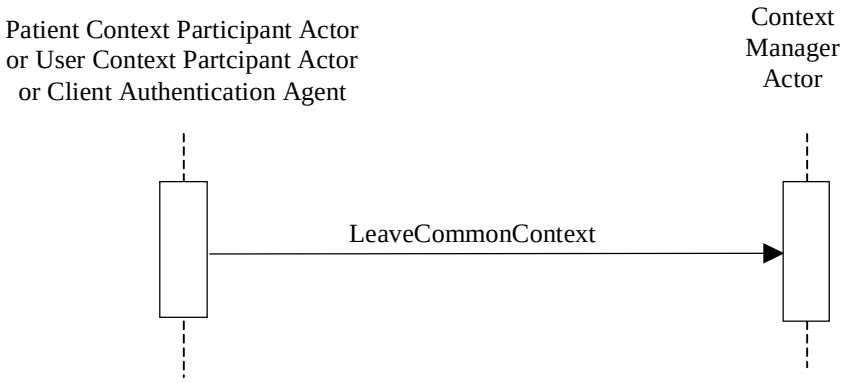


Figure 3.7-1 Leave Context Sequence

3.7.4.1 Leave Context – LeaveCommonContext Method

3.7.4.1.1 Trigger Events

This transaction is triggered by the user closing an application that contains a Patient Context Participant Actor, a User Context Participant Actor, or Client Authentication Agent Actor.

3.7.4.1.2 Message Semantics

LeaveContext is defined as a method on the ContextManager interface. It shall be invoked by a Context Participant Actor to announce its departure from the secure context session. A Context Participant Actor shall provide parameters for this method as specified in the CCOW Standard.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.4, for a description of the parameters associated with this method.

3.7.4.1.3 Expected Actions

The Context Manager Actor acknowledges the receipt of the notification. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.6.4, for a description of the response issued by the Context Manager Actor.

The context participant is expected to dispose of all context manager interface references upon receipt of the message reply. No further context change transactions will be processed by the Context Manager for this context participant.

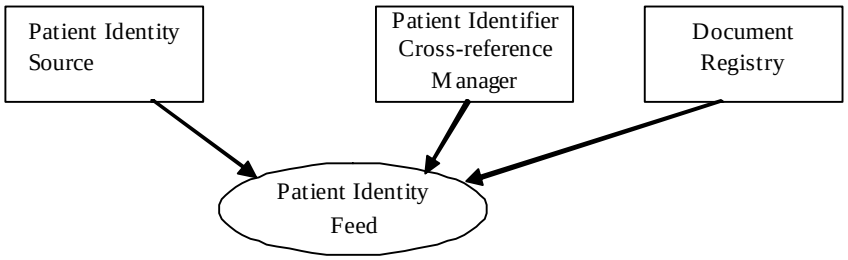
3.8 Patient Identity Feed

This section corresponds to Transaction ITI-8 of the IHE IT Infrastructure Technical Framework. Transaction ITI-8 is used by the Patient Identity Source, Patient Identifier Cross-reference Manager and Document Registry actors.

3.8.1 Scope

This transaction communicates patient information, including corroborating demographic data, after a patient's identity is established, modified or merged or after the key corroborating demographic data has been modified.

3.8.2 Use Case Roles



Actor: Patient Identity Source

Role: Provides notification to the Patient Identifier Cross-reference Manager and Document Registry for any patient identification related events including: creation, updates, merges, etc.

Actor: Patient Identifier Cross-reference Manager

Role: Serves a well-defined set of Patient Identification Domains. Based on information provided in each Patient Identification Domain by a Patient Identification Source Actor, it manages the cross-referencing of patient identifiers across Patient Identification Domains.

Actor: Document Registry

Role: Uses patient identifiers provided by Patient Identity Source to ensure that XDS Documents metadata registered is associated with a known patient and updates patient identity in document metadata by tracking identity change operations (e.g. merge).

3.8.3 Referenced Standards

HL7 Version 2.3.1 Chapter 2 – Control, Chapter 3 – Patient Administration

HL7 version 2.3.1 was selected for this transaction for the following reasons:

- It provides a broader potential base of Patient Identity Source Actors capable of participating in the profiles associated with this transaction.

- It allows existing ADT Actors from within IHE Radiology to participate as Patient Identity Source Actors.

3.8.4 Interaction Diagram

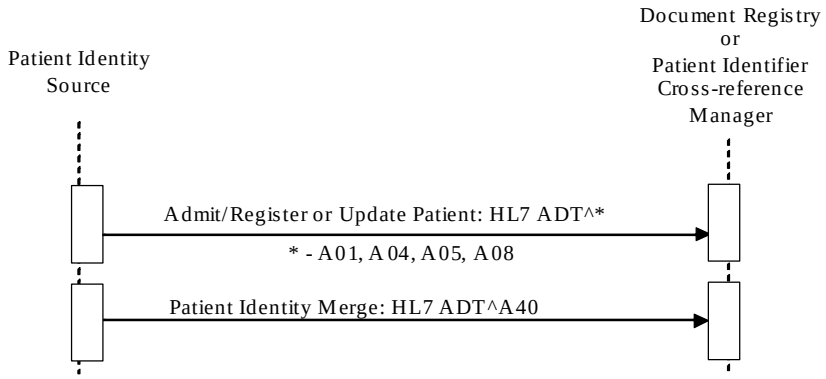


Figure 3.8-1 Patient Identity Sequence

3.8.4.1 Patient Identity Management – Admit/Register or Update Patient

3.8.4.1.1 Trigger Events

The following events from a Patient Identity Source Actor will trigger one of the Admit/Register or Update messages:

- A01 – Admission of an in-patient into a facility
- A04 – Registration of an outpatient for a visit of the facility
- A05 – Pre-admission of an in-patient (i.e., registration of patient information ahead of actual admission).

Changes to patient demographics (e.g., change in patient name, patient address, etc.) shall trigger the following Admit/Register or Update message:

- A08 – Update Patient Information

The Patient Identifier Cross-reference Manager shall only perform cross-referencing logic on messages received from Patient Identity Source Actors. For a given Patient Identifier Domain there shall be one and only one Patient Identity Source Actor, but a given Patient Identity Source Actor may serve more than one Patient Identifier Domain.

3.8.4.1.2 Message Semantics

The Patient Identity Feed transaction is conducted by the HL7 ADT message, as defined in the subsequent sections. The Patient Identity Source Actor shall generate the message whenever a patient is admitted, pre-admitted, or registered, or when some piece of patient demographic data

changes. Pre-admission of inpatients shall use the A05 trigger event. The segments of the message listed below are required, and their detailed descriptions are provided in the following subsections.

Note: Conventions used in this section as well as additional qualifications to the level of specification and HL7 profiling are stated in Appendix C and C.1 in this Volume.

Required segments are defined below. Other segments are optional

Table 3.8-1 ADT Patient Administration Messages

ADT	Patient Administration Message	Chapter in HL7 2.3.1
MSH	Message Header	2
EVN	Event Type	3
PID	Patient Identification	3
PV1	Patient Visit	3

Each message shall be acknowledged by the HL7 ACK message sent by the receiver of ADT message to its sender. See Appendix C.1.3, "Acknowledgement Modes", for definition and discussion of the ACK message.

This transaction does not require Patient Identity Source Actors to include any attributes not already required by the corresponding HL7 message (as is described in the following sections). This minimal set of requirements enables inclusion of the largest range of Patient Identity Source Actor systems.

This transaction **does** place additional requirements on the Patient Identifier Cross-reference Manager and Document Registry Actors, requiring them to accept aset of HL7 attributes beyond what is required by HL7. (See Section 3.8.4.1.3 for a description of these additional requirements)..

3.8.4.1.2.1 MSH Segment

The MSH segment shall be constructed as defined in Appendix C.1.2 "Message Control".

Field *MSH-9 Message Type* shall have at least two components. The first component shall have a value of **ADT**; the second component shall have values of **A01**, **A04**, **A05** or **A08** as appropriate. The third component is optional; however, if present, it shall have a value of **ADT_A01**.

3.8.4.1.2.2 EVN Segment

The Patient Identity Source Actor is not required to send any attributes within the EVN segment beyond what is specified in the HL7 standard. See Table C.1-4 in Appendix C.1.4 "Common Segment Definitions" for the specification of this segment.

3.8.4.1.2.3PID Segment

The Patient Identity Source Actor is not required to send any attributes within the PID segment beyond what is specified in the HL7 standard.

This message shall use the field PID-3 Patient Identifier List to convey the Patient ID uniquely identifying the patient within a given Patient Identification Domain.

The Patient Identity Source Actor shall provide the patient identifier in the ID component (first component) of the PID-3 field (PID-3.1). If the Patient Identity Source Actor provides component PID-3.4, Assigning Authority, then either the first subcomponent (namespace ID) or the second and third subcomponents (universal ID and universal ID type) shall be populated. If all three subcomponents are populated, the first subcomponent shall reference the same entity as is referenced by the second and third components.

3.8.4.1.2.4PV1 Segment

The Admit/ Register or Update Patient message is not required to include any attributes within the PV1 segment beyond what is specified in the HL7 standard.

3.8.4.1.3 Expected Actions – Patient Identifier Cross-reference Manager

The Patient Identifier Cross-reference Manager shall be capable of accepting attributes in the PID segment as specified in Table 3.8-2. This is to ensure that the Patient Identifier Cross-reference Manager can handle a sufficient set of corroborating information in order to perform its cross-referencing function.

Table 3.8-2 IHE Profile - PID segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	4	SI	O		00104	Set ID - Patient ID
2	20	CX	O		00105	Patient ID
3	250	CX	R		00106	Patient Identifier List
4	20	CX	O		00107	Alternate Patient ID
5	250	XPN	R		00108	Patient Name
6	250	XPN	R+		00109	Mother's Maiden Name
7	26	TS	R+		00110	Date/Time of Birth
8	1	IS	R+	0001	00111	Administrative Sex
9	250	XPN	O		00112	Patient Alias
10	250	CE	O	0005	00113	Race
11	250	XAD	R2		00114	Patient Address
12	4	IS	O	0289	00115	County Code
13	250	XTN	R2		00116	Phone Number - Home
14	250	XTN	R2		00117	Phone Number - Business

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
15	250	CE	O	0296	00118	Primary Language
16	250	CE	O	0002	00119	Marital Status
17	250	CE	O	0006	00120	Religion
18	250	CX	O		00121	Patient Account Number
19	16	ST	R2		00122	SSN Number – Patient
20	25	DLN	R2		00123	Driver's License Number - Patient
21	250	CX	O		00124	Mother's Identifier
22	250	CE	O	0189	00125	Ethnic Group
23	250	ST	O		00126	Birth Place
24	1	ID	O	0136	00127	Multiple Birth Indicator
25	2	NM	O		00128	Birth Order
26	250	CE	O	0171	00129	Citizenship
27	250	CE	O	0172	00130	Veterans Military Status
28	250	CE	O	0212	00739	Nationality
29	26	TS	O		00740	Patient Death Date and Time
30	1	ID	O	0136	00741	Patient Death Indicator

Adapted from the HL7 standard, Version 2.3.1

Note: This table reflects attributes required to be handled by the Patient Identifier Cross-reference Manager (receiver). It is likely that not all attributes marked as R2 or R+ above will be sent in some environments.

If the PID-3.4 (assigning authority) component is not included in the message (as described in Section 3.8.4.1.2.3) the Patient Identifier Cross-reference Manager shall fill PID-3.4 prior to storing the ID information and performing its cross-referencing activities. The information filled by the Patient Identifier Cross-reference Manager is based on the configuration associating each of the Patient Identity Source actors with the subcomponents of the correct assigning authority (namespace ID, UID and UID type). (See 3.8.4.1.3.1 below for a list of required Patient Identifier Cross-reference Manager configuration parameters).

A single Patient Identity Source Actor can serve multiple Patient Identification domains as long as it explicitly provides a fully qualified assigning authority. The Patient Identifier Cross-reference Manager Actor shall only recognize (by configuration) a single Patient Identity Source Actor per domain. (See 3.8.4.1.3.1 below for a list of required Patient Identifier Cross-reference Manager configuration parameters).

The cross-referencing process (algorithm, human decisions, etc.) is performed within the Patient Identifier Cross-reference Manager Actor, but its specification is beyond the scope of IHE.

Once the Patient Identifier Cross-reference Manager has completed its cross-referencing function, it shall make the newly cross-referenced identifiers available to PIX queries and send out notification to any Patient Identifier Cross-reference Consumers that have been configured (as

being interested in receiving such notifications) using the PIX Update Notification transaction (see Section 3.10 for the details of that transaction).

3.8.4.1.3.1 Required Patient Identifier Cross-reference Manager Configuration

The following items are expected to be parameters that are configurable on the Patient Identifier Cross-reference Manager Actor. For each Identification Domain included in the Identification Cross-reference Domain managed by a Patient Identifier Cross-reference Manager Actor, the following configuration information is needed:

- Identifier of the Identification Domain itself. This identifier shall specify the 3 components of the HL7 assigning authority (including the namespace ID and/or both the universal ID and universal ID type subcomponents) of the PID-3 field for the identification domain/source.
- Patient Identity Source Actor for that domain. This is expected to be the MSH-3 Sending Application field in the HL7 ADT message. (Alternative identification schemes might include IP address of the Patient Identity Source Actor or Node Authentication if the Basic Security Profile defined by IHE Radiology is used.)
- Details about where in the HL7 ADT message (Identity Feed transaction) that the Source Actor will provide the patient identifier for the domain (e.g., first component of the PID-3 field). If a Source Actor is managing multiple Patient Identifier Domains then the Source Actor is required to include the assigning authority in its Identity Feed Transactions.

3.8.4.1.4 Expected Actions – Document Registry

The Document Registry shall be capable of accepting attributes in the PID segment as specified in Table 3.8-2. The Patient Identity Feed transaction contains more than what the XDS Document Registry needs for its operation.

Table 3.8-2 IHE Profile - PID segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
3	250	CX	R		00106	Patient Identifier List

Adapted from the HL7 standard, Version 2.3.1

Note: This table reflects only the attributes required to be handled by the Document Registry (receiver). Other attributes of the PID Segment may be ignored.

If the PID-3.4 (assigning authority) component is not included, or subcomponents 2 and 3 (the universal ID and the universal ID Type of Assigning Authority) of PID-3.4 are not filled in the message (as described in Section 3.8.4.1.2.3) the Document Registry shall fill subcomponents 2 and 3 of PID-3.4 prior to storing the patient identity in the registry. The assigning authority information filled by the Document Registry is based on its configuration associating to each of the Patient Identity Source actors, which shall include the universal ID and the universal ID Type of the correct assigning authority. (See 3.8.4.1.4.1 below for a list of required Document Registry configuration parameters).

A single Patient Identity Source Actor can serve multiple Patient Identification domains, and therefore may include multiple patient identifiers in a message, as long as it explicitly specifies a fully qualified assigning authority for each patient identifier. The Document Registry Actor shall only recognize (by configuration) one single Patient Identity Source Actor per domain. (See 3.8.4.1.4.1 below for a list of required Document Registry configuration parameters).

The Document Registry shall store only the patient identifiers of the patient identification domain designated by the Affinity Domain for document sharing in the registry. Patient identifiers of other patient identification domains (assigning authorities), if present in a received message, shall be ignored.

3.8.4.1.4.1 Required Document Registry Configuration

The following items are expected to be parameters that are configurable on the Document Registry Actor:

- Identifier of the Patient Identification Domain itself. This identifier shall be specified with 2 components of the HL7 assigning authority (date type HD): universal ID and universal ID type. The universal ID shall be an OID (ISO Object Identifier), and therefore the universal ID Type must be "ISO".

3.8.4.2 Patient Identity Management –Patient Identity Merge (Merge Patient ID)

3.8.4.2.1 Trigger Events

When two patients' records are found to identify the same patient by a Patient Identity Source Actor in a Patient Identifier Domain and are merged, the Patient Identity Source shall trigger the following message:

- A40 – Merge Patient – Internal ID

An A40 message indicates that the Patient Identity Source Actor has done a merge within a specific Patient Identification Domain. That is, MRG-1 (patient ID) has been merged into PID-3 (Patient ID).

3.8.4.2.2 Message Semantics

The Patient Identity Feed transaction is an HL7 ADT message. The message shall be generated by the system (Patient Identity Source Actor) that performs the update whenever two patient records are found to reference the same person.

Note: Conventions used in this section as well as additional qualifications to the level of specification and HL7 profiling are stated in Appendix C and C.1 in this Volume.

The segments of the HL7 Merge Patient message listed below are required, and the detailed description of the message is provided in Section 3.8.4.2.2.1–3.8.4.2.2.6. The PV1 segment is optional.

Table 3.8-3 ADT A40 Patient Administration Message

ADT A40	Patient Administration Message	Chapter in HL7 v2.3.1
MSH	Message Header	2
EVN	Event Type	3
PID	Patient Identification	3
MRG	Merge Information	3
[PV1]	Patient Visit	3

Each message shall be acknowledged by the HL7 ACK message sent by the receiver of ADT message to its sender. See Appendix C.1.3 “Acknowledgement Modes” for definition and discussion of the ACK message.

A separate merge message shall be sent for each pair of patient records to be merged. For example, if Patients A, B, and C are all to be merged into Patient B, two ADT^A40 messages would be sent. In the first ADT^A40 message, patient B would be identified in the PID segment and Patient A would be identified in the MRG segment. In the second ADT^A40 message, patient B would be identified in the PID segment, and Patient C would be identified in the MRG segment.

Modification of any patient demographic information shall be done by sending a separate Update Patient Information (A08) message for the current Patient ID. An A40 message is the only method that may be used to update a Patient ID.

3.8.4.2.2.1 MSH Segment

MSH segment shall be constructed as defined in the Appendix C.1.2 “Message Control”.

Field *MSH-9 Message Type* shall have at least two components. The first component shall have a value of **ADT**; the second component shall have value of **A40**. The third component is optional; however, if present, it shall have a value of **ADT_A39**.

3.8.4.2.2.2 EVN Segment

See Appendix C.1.4 for the list of all required and optional fields within the EVN segment.

3.8.4.2.2.3 PID Segment

The PID segment shall be constructed as defined in Section 3.8.4.1.2.3.

3.8.4.2.2.4 MRG Segment

The PID and PV1 segments contain the dominant patient information, including Patient ID (and Issuer of Patient ID). The MRG segment identifies the “old” or secondary patient records to be

de-referenced. HL7 does not require that the ”old” record be deleted; it does require that the "old" identifier shall not be referenced in future transactions following the merge.

IHE does not require that the Patient Identity Source Actor send any attributes within the MRG segment beyond what is specified in the HL7 standard. If the assigning authority component of MRG-1 is present (MRG-1.4), it shall be equal to the assigning authority component of PID-3 (PID-3.4)

3.8.4.2.2.5PV1 Segment

PV1 segment shall be constructed as defined in Section 3.8.4.1.2.4.

3.8.4.2.3 Expected Actions

The Patient Identifier Cross-reference Manager shall be capable of accepting attributes in the MRG segment as specified in Table 3.8-4.

Table 3.8-4 IHE Profile - MRG segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	250	CX	R		00211	Prior Patient Identifier List
2	250	CX	O		00212	Prior Alternate Patient ID
3	250	CX	O		00213	Prior Patient Account Number
4	250	CX	R2		00214	Prior Patient ID
5	250	CX	O		01279	Prior Visit Number
6	250	CX	O		01280	Prior Alternate Visit ID
7	250	XPN	R2		01281	Prior Patient Name

Adapted from the HL7 Standard, Version 2.3.1

In addition, the Patient Identifier Cross-reference Manager shall perform the Expected Actions as specified in Section 3.8.4.1.3.

When the Patient Identifier Cross-reference Manager receives the ADT^A40 message type of the Patient Identity Feed transaction, it shall cross-reference the patient identifiers provided in the PID-3 and MRG-1 fields of the message by replacing any references it is maintaining internally to the patient ID provided in the MRG-1 field by the patient ID included in the PID-3 field. After the identifier references are replaced, the Patient Identifier Cross-reference Manager shall reapply its internal cross-referencing logic/ policies before providing the updated information via either the PIX Query or PIX Notification Transactions.

3.8.4.2.4 Expected Actions – Document Registry

The Document Registry shall be capable of accepting attributes in the MRG segment as specified in Table 3.8-4. Other attributes may exist, but the Document Registry shall ignore them.

Table 3.8-4 IHE Profile - MRG segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	250	CX	R		00211	Prior Patient Identifier List
2	250	CX	O		00212	Prior Alternate Patient ID
3	250	CX	O		00213	Prior Patient Account Number
4	250	CX	R2		00214	Prior Patient ID
5	250	CX	O		01279	Prior Visit Number
6	250	CX	O		01280	Prior Alternate Visit ID
7	250	XPN	R2		01281	Prior Patient Name

Adapted from the HL7 Standard, Version 2.3.1

In addition, the Document Registry shall perform the Expected Actions as specified in Section 3.8.4.1.4.

When the Document Registry receives the ADT^A40 message type of the Patient Identity Feed transaction, it shall merge the patient identity specified in MRG-1 (secondary patient identity) into the patient identity specified in PID-3 (primary patient identity) in its registry. After the merge, all Document Submission Sets (including all Documents beneath them) under the secondary patient identity before the merge shall point to the primary patient identity. The secondary patient identity shall no longer be referenced in the future services provided by the Document Registry.

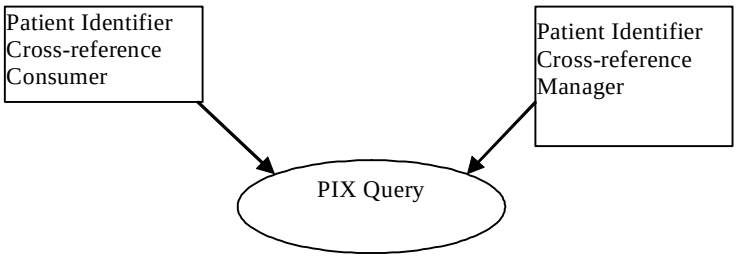
3.9 PIX Query

This section corresponds to Transaction ITI-9 of the IHE IT Infrastructure Technical Framework. Transaction ITI-9 is used by the Patient Identifier Cross-reference Consumer and Patient Identifier Cross-reference Manager actors.

3.9.1 Scope

This transaction involves a request by the Patient Identifier Cross-reference Consumer Actor for a list of patient identifiers that correspond to a patient identifier known by the consumer. The request is received by the Patient Identifier Cross-reference Manager. The Patient Identifier Cross-reference Manager immediately processes the request and returns a response in the form of a list of corresponding patient identifiers, if any.

3.9.2 Use Case Roles



Actor: Patient Identifier Cross-reference Consumer

Role: Queries the Patient Identifier Cross-reference Manager for a list of corresponding patient identifiers, if any

Actor: Patient Identifier Cross-reference Manager

Role: Manages the cross-referencing of patient identifiers across Patient Identification Domains. Upon request it returns a list of corresponding patient identifiers, if any.

3.9.3 Referenced Standard

HL7 2.5, Chapter 2 – Control, Chapter 3 – Patient Administration, Chapter 5 – Query

HL7 version 2.5 was selected for this transaction for the following reasons:

It was considered the most stable version that contained the functionality required by transactions ITI-9 and ITI-10.

3.9.4 Interaction Diagram

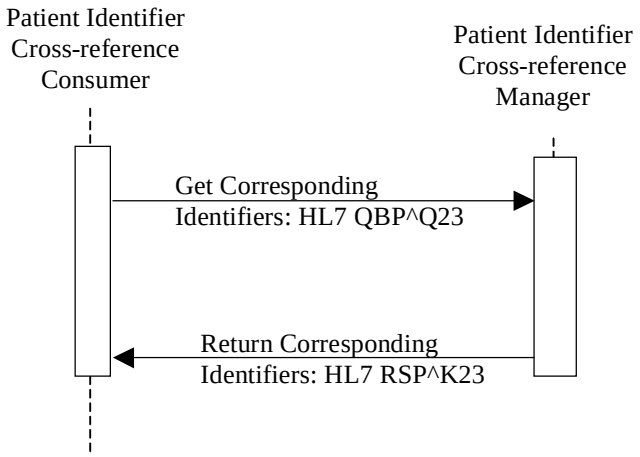


Figure 3.9-1 Get Corresponding Identifiers Sequence

3.9.4.1 Get Corresponding Identifiers

3.9.4.1.1 Trigger Events

A Patient Identifier Cross-reference Consumer's need to get the patient identifier associated with a domain for which it needs patient related information will trigger the request for corresponding patient identifiers message based on the following HL7 trigger event:

- Q23 – Get Corresponding Identifiers

3.9.4.1.2 Message Semantics

The Request for Corresponding Patient Identifiers transaction is conducted by the HL7 QBP^Q23 message. The Patient Identifier Cross-reference Consumer Actor shall generate the query message whenever it needs to obtain a corresponding patient identifier(s) from other Patient Identification Domain(s). The segments of the message listed below are required, and their detailed descriptions are provided in the following subsections.

Note: Conventions used in this section as well as additional qualifications to the level of specification and HL7 profiling are stated in Appendix C and C.1 in this Volume.

Table 3.9-1 QBP Query By Parameter

QBP	Query By Parameter	Chapter in HL7 2.5
MSH	Message Header	2
QPD	Query Parameter Definition	5

QBP	Query By Parameter	Chapter in HL7 2.5
RCP	Response Control Parameter	5

The receiver shall respond to the query by sending the RSP^K23 response message. This satisfies the requirements of original mode acknowledgment; no intermediate ACK message is to be sent.

3.9.4.1.2.1 MSH Segment

The MSH segment shall be constructed as defined in Appendix C.1.2 “Message Control”.

Field *MSH-9 Message Type* shall have at least two components. The first component shall have a value of QBP; the second component shall have the value of Q23. The third component is optional; however, if present, it shall have a value of QBP_Q21.

3.9.4.1.2.2 QPD Segment

The Patient Identifier Cross-reference Consumer Actor is required to send attributes within the QPD segment as described in Table 3.9-2.

Table 3.9-2 IHE Profile - QPD segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	250	CE	R	0471	01375	Message Query Name
2	32	ST	R+		00696	Query Tag
3	250**	CX	R			Person Identifier
4	250	CX	O			What Domains Returned

Adapted from the HL7 Standard, version 2.5

** Note: This value assumes completion of an HL7 erratum to correct an error identified in the standard.

This message shall use the field QPD-3 *Person Identifier* to convey a single Patient ID uniquely identifying the patient within a given Patient Identification Domain.

The Patient Identifier Cross-reference Consumer Actor shall provide the patient identifier in the ID component (first component) of the QPD-3 field (QPD-3.1).

The Patient Identifier Cross-reference Consumer Actor shall provide component QPD-3.4, Assigning Authority, by including either the first subcomponent (namespace ID) or the second and third subcomponents (universal ID and universal ID type) If all three subcomponents are populated, the first subcomponent shall reference the same entity as is referenced by the second and third components.

If the requesting system wishes to select the domains from which they wish to receive Patient IDs, it does so by populating *QPD-4-What Domains Returned* with as many repetitions as

domains for which it wants to receive Patient IDs. Each repetition of QPD-4 shall contain an instance of data type CX in which only the fourth component (Assigning Authority) is populated; the remaining components shall be empty. The responding system shall return the Patient ID value for each requested domain if a value is known.

If QPD-4 is empty, the Patient Identifier Cross-reference Manager Actor shall return Patient IDs for all domains for which it possesses a corresponding Patient ID (subject to local publication restrictions).

3.9.4.1.2.3 RCP Segment

Although HL7 requires that the RCP Segment be sent in all QBP messages, IHE does not require that the Patient Identifier Cross-reference Consumer Actor send any attributes within the RCP segment, as is specified in the HL7 standard.

3.9.4.1.2.3.1 3.9.4.1.2.3.1 Populating RCP-1-Query Priority

Field *RCP-1-Query Priority* shall always contain **I**, signifying that the response to the query is to be returned in Immediate mode.

3.9.4.1.3 Expected Actions

The Patient Identifier Cross-reference Manager shall be capable of accepting attributes in the QPD segment as specified in Table 3.9-2.

The Patient Identifier Cross-reference Manager Actor must be capable of receiving all valid combinations of subcomponents that make up the Assigning Authority component (i.e., all valid combinations of QPD-3.4).

The Patient Identifier Cross-reference Manager Actor shall be capable of accepting multiple concurrent PIX Query requests (Get Corresponding Identifiers messages) and responding correctly using the Return Corresponding Identifiers message.

3.9.4.2 Return Corresponding Identifiers

3.9.4.2.1 Trigger Events

The Patient Identifier Cross-reference Manager's response to the Get Patient Identifiers message will trigger the following message:

- K23 – Corresponding patient identifiers

3.9.4.2.2 Message Semantics

The Return Corresponding Identifiers transaction is conducted by the HL7 RSP^K23 message. The Patient Identifier Cross-reference Manager Actor shall generate this message in direct

response to the QBP^Q23 query message previously received. This message satisfies the Application Level, Original Mode Acknowledgement for the HL7 QBP^Q23 message. The segments of the message listed without enclosing square brackets in the Table below are required. Detailed descriptions of all segments listed in the table below are provided in the following subsections. Other segments of the message are optional.

Note: Conventions used in this section as well as additional qualifications to the level of specification and HL7 profiling are stated in Appendix C and C.1 in this Volume.

Table 3.9-3 RSP Segment Pattern Response

RSP	Segment Pattern Response	Chapter in HL7 2.5
MSH	Message Header	2
MSA	Message Acknowledgement	2
[ERR]	Error segment	2
QAK	Query Acknowledgement	5
QPD	Query Parameter Definition	5
[PID]	Patient Identification	3

3.9.4.2.2.1 MSH Segment

The MSH segment shall be constructed as defined in Appendix C.1.2, “Message Control”.

Field *MSH-9-Message Type* shall have at least two components. The first component shall have a value of RSP; the second component shall have the value of K23. The third component is optional; however, if present, it shall have a value of RSP_K23.

3.9.4.2.2.2 MSA Segment

The Patient Identifier Cross-reference Manager Actor is not required to send any attributes within the MSA segment beyond what is specified in the HL7 standard. See Appendix C.1.3 for the list of all required and optional fields within the MSA segment.

3.9.4.2.2.3 QAK Segment

The Patient Identifier Cross-reference Manager Actor shall send attributes within the QAK segment as defined in Table 3.9-4. For the details on filling in QAK-2 (Query Response Status) refer to Section 3.9.4.2.2.6.

Table 3.9-4 IHE Profile - QAK segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	32	ST	R		00696	Query Tag
2	2	ID	R+	0208	00708	Query Response Status

Adapted from the HL7 standard, version 2.5

3.9.4.2.2.4 QPD Segment

The Patient Identifier Cross-reference Manager Actor shall echo the QPD Segment value that was sent in the QBP^Q23 message.

3.9.4.2.2.5 PID Segment

The Patient Identifier Cross-reference Manager Actor shall return only those attributes within the PID segment that are required by the HL7 standard: *PID-3-Patient IdentifierList* and *PID-5-Patient Name*.

The PID segment is returned only when the Patient Identifier Cross-reference Manager Actor recognizes the specified Patient Identification Domain and Patient ID and an identifier exists for the specified patient in at least one other domain. See Section 3.9.4.2.2.6, "Patient Identifier Cross-reference Manager Actor Query Response Behavior," for a detailed description of how the Patient Identifier Cross-reference Manager Actor responds to the query request under various circumstances.

The Patient Identifier Cross-reference Manager Actor shall use the field PID-3 Patient Identifier List to convey the Patient ID uniquely identifying the patient within each Patient Identification Domain for which a Patient ID exists for the specified patient. Each resulting ID returned in PID-3 shall include a fully qualified Assigning Authority component. In other words, the Assigning Authority component returned shall include ALL subcomponents (namespace ID, Universal ID, and Universal ID type).

To eliminate the issue of conflicting name values between Patient Identifier Domains, the Patient Identifier Cross-reference Manager Actor shall return in an empty (not present) value in the first repetition of field PID-5-Patient Name, and shall return a second repetition of field *PID-5-Patient Name* in which the only populated component is Component 7 (Name Type Code). Component 7 of repetition 2 shall contain a value of S (Coded Pseudo-name to assure anonymity). All other components of repetition 2 shall be empty (not present).

3.9.4.2.2.6 Patient Identifier Cross-reference Manager Actor Query Response Behavior

It is wholly the responsibility of the Patient Identifier Cross-reference Manager Actor to perform the matching of patient identifiers based on the patient traits it receives. The information provided by the Patient Identifier Cross-reference Manager Actor to Patient Identifier Cross-reference Consumer Actors is a list of cross-referenced identifiers in two or more of the domains managed by the cross-referencing Actor. The list of cross-references is not made available until the set of policies and processes for managing the cross-reference function have been completed. The policies of administering identities adopted by the cooperating domains are completely internal to the Patient Identifier Cross-reference Manager Actor and are outside of the scope of

this framework. Possible matches should not be communicated until the healthcare institution policies and processes embodied in the Patient Identifier Cross-reference Manager Actor reach a positive matching decision.

The Patient Identifier Cross-reference Manager Actor shall respond to the query request as described by the following 6 cases:

Case 1: The Patient Identifier Cross-reference Manager Actor recognizes the specified Patient Identification Domain and Patient ID sent by the Patient Identifier Cross-reference Consumer in QPD-3, and corresponding identifiers exist for the specified patient in at least one of the domains requested in QPD-4 (one identifier per domain). (See Case 6 below for the required behavior if there are multiple identifiers recognized within a given Identifier Domain by the Patient Identifier Cross-reference Manager Actor.)

AA (application accept) is returned in MSA-1.

OK (data found, no errors) is returned in QAK-2.

A single PID segment is returned in which one repetition of *PID-3 Patient Identifier List* is populated for each of the domains, if any, that the Patient Identifier Cross-reference Manager Actor did recognize in which a single identifier exists for the requested patient, not including the queried-for patient identifier that is returned in QPD-3.

Case 2: The Patient Identifier Cross-reference Manager Actor recognizes the Patient Identification Domain and Patient ID sent in QPD-3, but no identifier exists for that patient in any of the domains sent in QPD-4.

AA (application accept) is returned in MSA-1.

NF (no data found, no errors) is returned in QAK-2.

No PID segment is returned.

Case 3: The Patient Identifier Cross-reference Manager Actor recognizes the specified Patient Identification Domain sent in the fourth component of QPD-3, but does not recognize the Patient ID sent in the first component of QPD-3.

AE (application error) is returned in MSA-1 and in QAK-2.

An ERR segment is returned in which the components of *ERR-1-Error Code* and Location are valued as follows.

COMP #	COMPONENT NAME	VALUE
1	Segment ID	QPD
2	Sequence	1

COMP #	COMPONENT NAME	VALUE
3	Field Position	3
4	Field Repetition	1
5	Component Number	1
6	Sub-Component Number	*(empty)*

As specified by HL7, *ERR-2.6-Sub-Component Number* is not valued because we are referring to the entire fourth component of field QPD-4.

ERR-3-HL7 Error Code is populated with the error condition code **204** (unknown key identifier). Together with the values in ERR-2, this signifies that the Patient Identifier Cross-reference Manager Actor did not recognize the value in the first component of QPD-3.

Case 4: The Patient Identifier Cross-reference Manager Actor does not recognize the Patient Identification Domain of the identifier sent in QPD-3.

AE (application error) is returned in MSA-1 and in QAK-2.

An ERR segment is returned in which the components of *ERR-2-Error Location* are valued as follows.

COMP #	COMPONENT NAME	VALUE
1	Segment ID	QPD
2	Sequence	1
3	Field Position	3
4	Field Repetition	1
5	Component Number	4
6	Sub-Component Number	*(empty)*

As specified by HL7, *ERR-2.6-Sub-Component Number* is not valued because we are referring to the entire fourth component of field QPD-3.

ERR-3-HL7 Error Code is populated with the error condition code **204** (unknown key identifier). Together with the values in ERR-2, this signifies that the Patient Identifier Cross-reference Manager Actor did not recognize the value in the fourth component of QPD-3.

Case 5: The Patient Identifier Cross-reference Manager Actor does not recognize one or more of the Patient Identification Domains for which an identifier has been requested.

AE (application error) is returned in MSA-1 and in QAK-2.

For each domain that was not recognized, an ERR segment is returned in which the components of *ERR-2-Error Location* are valued as follows.

COMP #	COMPONENT NAME	VALUE
1	Segment ID	QPD
2	Sequence	1
3	Field Position	4
4	Field Repetition	*(see below)*
5	Component Number	*(empty)*
6	Sub-Component Number	*(empty)*

As specified by HL7, *ERR-2.5-Component Number* and *ERR-2.6-Sub-Component Number* are not valued because we are referring to the entire field QPD-4.

ERR-3-HL7 Error Code is populated with the error condition code **204** (unknown key identifier). Together with the values in ERR-2, this signifies that the Patient Identifier Cross-reference Manager Actor did not recognize the domain for the occurrence of *QPD-4-What Domains Returned* whose ordinal number is returned as an integer in ERR-2.4.

Case 6: The Patient Identifier Cross-reference Manager Actor recognizes the specified Patient Identification Domain and Patient ID sent by the Patient Identifier Cross-reference Consumer in QPD-3, and corresponding identifiers exist for the specified patient in at least one of the domains requested in QPD-4, and there are multiple identifiers within at least one of the requested domains.

AA (application accept) is returned in MSA-1.

OK (data found, no errors) is returned in QAK-2.

A single PID segment is returned in which one repetition of *PID-3-Patient Identifier List* is populated for each of the identifiers, not including the queried-for patient identifier that is returned in QPD-3. If the Patient Identifier Cross-reference Manager Actor chooses to return multiple identifiers associated with the same domain, it shall return these identifiers grouped in successive repetitions within the *PID-3-Patient Identifier List*.

3.9.4.2.3 Expected Actions

The Patient Identifier Cross-reference Consumer will use the list of patient identifier aliases provided by the Patient Identifier Cross-reference Manger to perform the functions for which it requested the list.

In the case where the returned list of identifiers contains multiple identifiers for a single domain, the Patient Identifier Cross-reference Consumer shall either use ALL of the multiple identifiers from the given domain or it shall ignore ALL of the multiple identifiers from the given domain.

This allows Patient Identifier Cross-reference Consumer Actors capable of handling multiple identities for a single patient within a single domain (i.e., those that can correctly aggregate the information associated with the different identifiers) to do so. For those Patient Identifier Cross-

reference Consumer Actors not capable of handling this situation, ignoring the entire list of different identifiers prevents the consumer from presenting incomplete data.

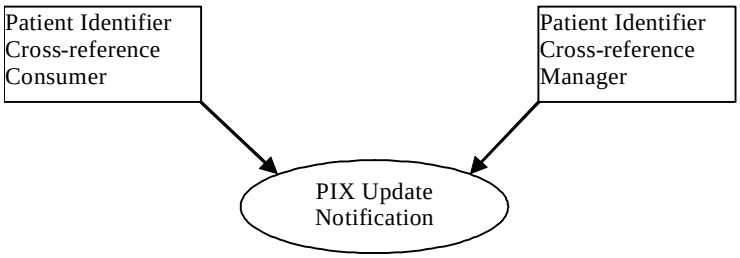
3.10 PIX Update Notification

This section corresponds to Transaction ITI-10 of the IHE IT Infrastructure Technical Framework. Transaction ITI-10 is used by the Patient Identifier Cross-reference Consumer and Patient Identifier Cross-reference Manager actors.

3.10.1 Scope

This transaction involves the Patient Identifier Cross-reference Manager Actor providing notification of updates to patient identifier cross-reference associations to Patient Identifier Cross-reference Consumers that have registered (by configuration on the Cross-reference Manager) their interest in receiving such notifications. This transaction uses HL7's generic 'Update Person Information' message to communicate this patient-centric information.

3.10.2 Use Case Roles



Actor: Patient Identifier Cross-reference Manager

Role: It serves a well-defined set of Patient Identification Domains. The Patient Identifier Cross-reference Manager manages the cross-referencing of patient identifiers across Patient Identification Domains by providing a list of patient ID "aliases" via notification to a configured list of interested Patient Identifier Cross-reference Consumers.

Actor: Patient Identifier Cross-reference Consumer

Role: Receives notifications from the Patient Identifier Cross-reference Manager of changes to patient ID aliases. Typically the Patient Identifier Cross-reference Consumer Actor uses this information to maintain information links about patients in a different patient ID domain.

3.10.3 Referenced Standard

HL7 Version 2.5, Chapter 2 – Control, Chapter 3 – Patient Administration

HL7 version 2.5 was selected for this transaction for the following reason:

It was considered the most stable version that contained the functionality required by Transaction ITI-9 and ITI-10.

3.10.4 Interaction Diagram

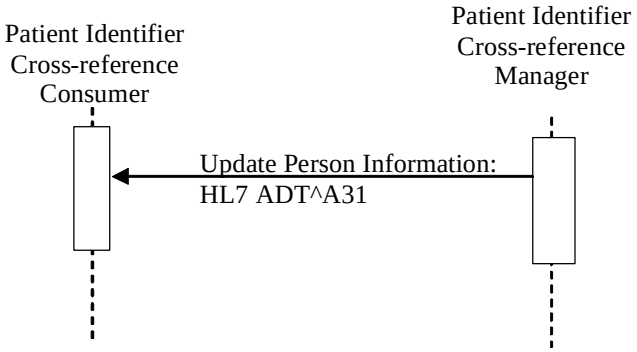


Figure 3.10-1 Update Person Information Sequence

3.10.4.1 Update Person Information

3.10.4.1.1 Trigger Events

The Patient Identifier Cross-reference Manager shall notify a Patient Identifier Cross-reference Consumer when there is a change in a set of cross-referenced patient identifiers for any of the patient identifiers belonging to Patient Identifier Domains of interest to the consumer. The configuration of the domains of interest to a Patient Cross-reference Consumer is maintained by the Patient Cross-reference Manager Actor.

Several notifications may have to be issued to communicate a single update to a set of cross-reference patient identifiers as required to reflect all the changes on the resulting sets of cross-reference patient Identifiers belonging to Patient Identifier Domains of interest to the Patient Identifier Cross-referencing Consumer.

The following HL7 trigger event will be used to update to the list of patient identifiers:

- A31 – Update Person Information

3.10.4.1.2 Message Semantics

The PIX Update Notification transaction is conducted by the ADT^A31 message. The Patient Identifier Cross-reference Manager Actor initiates this transaction whenever identifier list information is updated for a patient.

It is wholly the responsibility of the Patient Identifier Cross-reference Manager Actor to perform the matching of patient identifiers based on the patient traits it receives. The information provided by the Patient Identifier Cross-reference Manager Actor to Patient Identifier Cross-

reference Consumer Actors is a list of cross-referenced identifiers in two or more of the domains managed by the cross-referencing Actor. The list of cross-references is not made available until the set of policies and processes for managing the cross-reference function have been completed. The policies of administering identities adopted by the cooperating domains are completely internal to the Patient Identifier Cross-reference Manager Actor and are outside of the scope of this standard. Possible matches should not be communicated until the healthcare institution policies and processes embodied in the Patient Identifier Cross-reference Manager Actor reach a positive matching decision.

The Patient Identifier Cross-reference Manager Actor Configuration is expected to have configuration indicating which Identity Consumers are interested in receiving the PIX Update Notification Transactions. This configuration information shall include identification of the identity consumer systems interested in receiving notifications and, for each of those systems, a list of the patient identifier domains of interest.

The segments of the message listed in the Table below are required. Other segments are optional.

Table 3.10-1 ADT Patient Administration Message

ADT	Patient Administration Message	Chapter in HL7 2.5
MSH	Message Header	2
EVN	Event Type	3
PID	Patient Identification	3
PV1	Patient Visit	3

Each message shall be acknowledged by the HL7 ACK message sent by the receiver of ADT message to its sender. See Appendix C.1.3, “Acknowledgement Modes” for the definition and discussion of the ACK message.

3.10.4.1.2.1 MSH Segment

The MSH segment shall be constructed as defined in Appendix C.1.2, “Message Control”.

Field *MSH-9 Message Type* shall have at least two components. The first component shall have a value of ADT; the second component shall have the value of A31. The third component is optional; however, if present, it shall have a value of ADT_A31.

3.10.4.1.2.2 EVN Segment

See Appendix C.1.4 for the list of all required and optional fields within the EVN segment.

3.10.4.1.2.3 PID Segment

The Patient Identifier Cross-reference Manager Actor shall provide only those attributes within the PID segment that are required by the HL7 standard: *PID-3-Patient Identifier List* and *PID-5-Patient Name*.

The Patient Identifier Cross-reference Manager Actor shall use the field *PID-3 Patient Identifier List* to convey the Patient IDs uniquely identifying the patient within each Patient Identification Domain for which a Patient ID exists for the specified patient. Each resulting ID returned in PID-3 shall include a fully qualified Assigning Authority component. In other words, the Assigning Authority component returned shall include ALL subcomponents (namespace ID, Universal ID, and Universal ID type).

To eliminate the issue of multiple name values between Patient Identifier Domains, the Patient Identifier Cross-reference Manager Actor shall return a single space character in field *PID-5-Patient Name*.

A single PID segment is sent in which one repetition of *PID-3-Patient Identifier List* is populated for each of the identifiers in the notification. If the Patient Identifier Cross-reference Manager Actor chooses to send multiple identifiers associated with the same domain, it shall return these identifiers grouped in successive repetitions within the *PID-3-Patient Identifier List*.

3.10.4.1.2.4 PV1 Segment

As is specified by the HL7 Standard, Version 2.5, the PV1 Segment is required. The required field *PV1-2-patient class* shall contain **N** (not applicable) to indicate the transmission of patient information outside the context of a visit or encounter. Other fields shall be left blank.

Table 3.10-2 IHE Profile – PV1 segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
2	1	IS	R	0004	00132	Patient Class

Adapted from the HL7 Standard, version 2.5

3.10.4.1.3 Expected Actions

The Patient Identifier Cross-reference Consumer, when it receives the ADT^A31 message, shall update its internal identifier information for the affected patient(s) in all domains in which it is interested whenever it receives updated identifier information that results in a change to the cross-referencing of a patient.

In the case where the returned list of identifiers contains multiple identifiers for a single domain, the Patient Identifier Cross-reference Consumer shall either use ALL of the multiple identifiers from the given domain or it shall ignore ALL of the multiple identifiers from the given domain.

This allows Patient Identifier Cross-reference Consumer Actors capable of handling multiple identities for a single patient within a single domain (i.e., those that can correctly aggregate the information associated with the different identifiers) to do so. For those Patient Identifier Cross-reference Consumer Actors not capable of handling this situation, ignoring the entire list of different identifiers prevents the consumer from presenting incomplete data.

3.11 Retrieve Specific Information for Display

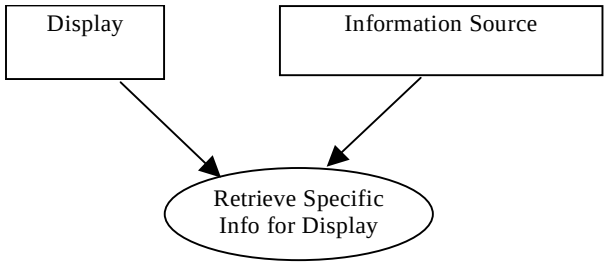
This section corresponds to Transaction ITI-11 of the IHE IT Infrastructure Technical Framework. Transaction ITI-11 is used by the Information Source and Display actors.

3.11.1 Scope

This transaction involves the query of information for presentation purposes. This may occur when a user attempts to lookup information associated with certain patient that is stored on a different system. Note that the retrieved information is always related to a well-identified patient (Patient ID), but its content, although of a specific type (lab summary, or radiology summary, list of allergies), is generally dynamic (i.e., retrieving the same type of specific information at a different point in time is likely to result in different content); for example, a list of allergies may have been updated between two requests.

To support a wide range of display capabilities, the information provided is formatted into well-formed XHTML. Such formatting shall be done using XHTML Basic and W3C HTML Compatibility Guidelines provided in the Appendix C of the W3C XHTML 1.0 Recommendation.

3.11.2 Use Case Roles



Actor: Display

Role: A system that requests specific information for display, and displays it.

Actor: Information Source

Role: A system that provides specific information in response to the request from the Display Actor, in a presentation-ready format.

3.11.3 Referenced Standards

- IETF RFC1738, Uniform Resource Locators (URL), December 1994, http://www.faqs.org/rfcs/rfc1738.html

IETF RFC2616 HyperText Transfer Protocol HTTP/1.1

Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. http://www.w3.org/TR/REC-xml.

Web Services Description Language (WSDL) 1.1. W3C Note 15 March 2001. http://www.w3.org/TR/wsdl.

XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).A Reformulation of HTML 4 in XML 1.0. W3C Recommendation 26 January 2000, revised 1 August 2002. http://www.w3.org/TR/xhtml1.

XHTML™ Basic. W3C Recommendation 19 December 2000. http://www.w3.org/TR/xhtm-basic.

Interaction Diagram

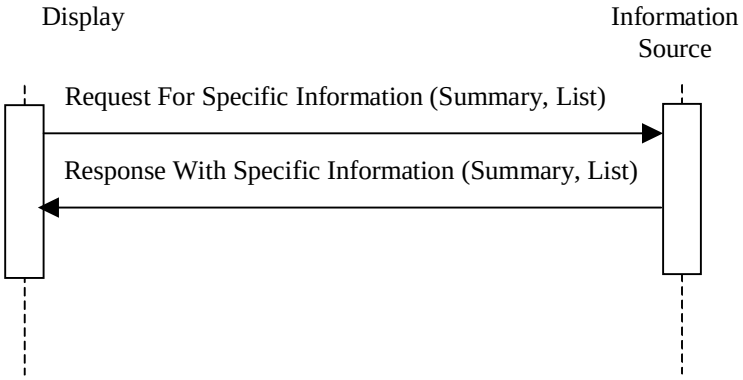


Figure 3.11-1 Request For Specific Information – Summary sequence

3.11.3.1 Request For Specific Information - Summary

3.11.3.1.1 Trigger Events

The following event will trigger a Request for Specific Information:

- User of the Display Actor needs to review a summary list of information/ reports that are part of a patient's clinical history (i.e., summary of lab reports, summary of radiology exam reports, etc.) with the intent of selecting a specific item off the list for subsequent retrieval as a persistent object via the Retrieve Document for Display Transaction

3.11.3.1.2 Message Semantics

The Retrieve Specific Information for Display transaction is performed by the invocation of a web service. The Display Actor shall generate a web service request whenever a user needs to review the information stored as part of a patient's clinical history on the Information Source Actor.

To specify the type of information that needs to be returned, a web service request shall include the following parameters (keys) to filter the subset of information (See Table 3.11.4-1). All parameter names and values (see Table 3.11.4-2) are case-sensitive.

Table 3.11.4-1 Web Service Request Keys

Parameter Name	REQ	Description	Notes
requestType	R	requestType specifies what type of information shall be retrieved. This parameter shall always be valued.	See Table 3.11.4-2 for the list of possible values.
patientID	R	This attribute identifies the subject of the results being queried for. Its value shall include identification of assigning authority.	PatientID value shall be formatted as HL7 CX data type (including assigning authority) according to the requirements specified for the Patient Identity Feed transaction (see Section 3.8.4.1.2.3)
lowerDateTime	O	Used to constrain the earliest date/time of creation of information.	This value shall be encoded in the XML primitive dateTime format.
upperDateTime	O	Used to constrain the latest date/time of creation of information.	This value shall be encoded in the XML primitive dateTime format.
mostRecentResults	R	The numeric value that indicates the number of most recent results to be included into the response, *i.e.*, 1 indicates to provide the latest result.	Value of 0 indicates that all available results shall be returned.

Table 3.11.4-2 Web Service Request Types

requestType value	Description
SUMMARY	Summary of all reports known to the Information Source
SUMMARY-RADIOLOGY	Summary of radiology reports
SUMMARY-CARDIOLOGY	Summary of cardiology reports
SUMMARY-LABORATORY	Summary of laboratory reports
SUMMARY-SURGERY	Summary of surgery reports
SUMMARY-EMERGENCY	Summary of emergency reports
SUMMARY-DISCHARGE	Summary of discharge reports
SUMMARY-ICU	Summary of intensive care reports
SUMMARY-RX	Summary of Prescriptions

Note: parameter values that contain reserved characters need to be encoded using %<hex><hex> notation. Reserved characters include slash (/, encode as %2f) and ampersand (&, encode as %26).

Formal definition of the web service in WSDL is provided in the Appendix A.

The only binding required for both the Display Actor and Information Source Actor is the binding to the HTTP-GET. In this binding the sample message will be formatted as follows:

http://<location>/IHERetrieveSummaryInfo?requestType=SUMMARY&patientID=9999841
0^^^%26www.mlhlife.com%26DNS &lowerDateTime=2003-01-
01T00:00:00&upperDateTime=2003-01-01T23:59:59&mostRecentResults=1

The <location> part of the URL is configurable by the implementation, and must contain the host name, an optional port address, and may be followed by an optional path. The path if present may not contain a '?' character. The remainder of the URL, including IHERetrieveSummaryInfo and the following request parameters are specified by the WSDL and may not be changed.

More specifically, using the definitions from RFC 1738, the <location> part of the URL must match the production for location from the figure below:

```
location       = hostport [ "/" hpath ]

hostport       = host [ ":" port ]
host           = hostname | hostnumber
hostname       = *[ domainlabel "." ] toplabel
domainlabel    = alphadigit | alphadigit *[ alphadigit | "-" ] alphadigit
toplabel       = alpha | alpha *[ alphadigit | "-" ] alphadigit
alphadigit     = alpha | digit
hostnumber     = digits "." digits "." digits "." digits
port           = digits

hpath          = hsegment *[ "/" hsegment ]
hsegment       = *[ uchar | ";" | ":" | "@" | "&" | "=" ]
lowalpha       = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" |
                 "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" |
                 "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" |
                 "y" | "z"
hialpha        = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" |
                 "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |
                 "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"

alpha          = lowalpha | hialpha
digit          = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                 "8" | "9"
safe           = "$" | "-" | "_" | "." | "+"
extra          = "!" | "*" | "'" | "(" | ")" | ","

hex            = digit | "A" | "B" | "C" | "D" | "E" | "F" |
                 "a" | "b" | "c" | "d" | "e" | "f"
escape         = "%" hex hex

unreserved     = alpha | digit | safe | extra
uchar          = unreserved | escape
```

The following location values are legal according to this specification:

<location> value	Resulting URL
Myhost	http://myhost/IHERetrieveSummaryInfo?…
myhost:8080	http://myhost:8080/IHERetrieveSummaryInfo?…
myhost/MyAspPageThatLooksLikeItCouldBeAFolder.aspx	http://myhost/MyAspPageThatLooksLikeItCouldBeAFolder.aspx/IHERetrieveSummaryInfo?…
myhost:8080/MyAspPageThatLooksLikeItCouldBeAFolder.aspx	http://myhost:8080/MyAspPageThatLooksLikeItCouldBeAFolder.aspx/IHERetrieveSummaryInfo?…
myhost/MyJspPage.jsp	http://myhost/MyJspPage.jsp/IHERetrieveSummaryInfo?…
myhost:8080/MyJspPageThatLooksLikeItCouldBeAFolder.jsp	http://myhost/MyJspPage.jsp/ IHERetrieveSummaryInfo?…

The following location values are not legal:

<location> value	Resulting URL
My+Computer	'+' is not a legal character in a host name.
myhost:99999	99999 is not a valid port.
myhost/myPath.jsp?request=	'?' is not valid in a path.

1615

In addition, the Display Actor shall support the following field of the HTTP request:

Table 3.11.4-3 HTTP Request and Response Fields

HTTP Field	REQ	Description	Values
Accept-Language	O	This field restricts the set of natural languages that are preferred as a response to the request.	Any valid value according to RFC2616

The Information Source actor shall support the following field of the HTTP response.

Table 3.11.4-4 HTTP Response Fields

HTTP Field	REQ	Description	Values
Expires	R	This field gives the date/time after which the response is considered stale	Shall be 0. This is now deprecated usage, but it is the widely supported means of specifying no cacheing.
Cache-Control	R	This field indicates that this response should not be cached.	Shall be no-cache

If necessary, the Display Actor may perform the request to the web service utilizing HTTPS protocol.

- Information Source Actors may return HTTP redirect responses (responses with values of 301, 302, 303 or 307) in response to a request. Display Actors can expect to receive an error response, or the data requested, or a request to look elsewhere for the data. A Display Actor must follow redirects, but if a loop is detected, it may report an error.

3.11.3.1.3 Expected Actions

Upon reception of the Request for Specific Information, the Information Source Actor shall parse the request and if there are no errors, return the Response with Specific Information as specified in Section 3.11.4.2, and HTTP response code 200 - OK.

To specify the type of information that needs to be processed, an Information Source Actor shall support the following parameters (keys) to filter the subset of information (See Table 3.11.4-5).

Table 3.11.4-5 Web Service Request Keys

Parameter Name	REQ	Description	Notes
requestType	R	requestType specifies what type of information shall be retrieved. This parameter shall always be valued.	See Table 3.11.4-2 for the list of possible values.
patientID	R	This attribute identifies the subject of the results being queried for. Its value shall include identification of assigning authority.	PatientID value shall be formatted as HL7 CX data type (including assigning authority) according to the requirements specified for the Patient Identity Feed transaction (see Section 3.8.4.1.2.3)
lowerDateTime	R	Used to constrain the earliest date/time of creation of information.	This value shall be encoded in the XML primitive dateTime format.
upperDateTime	R	Used to constrain the latest date/time of creation of information.	This value shall be encoded in the XML primitive dateTime format.
mostRecentResults	R	The numeric value that indicates the number of most recent results to be included into the response, *i.e.*, 1 indicates to provide the latest result.	Value of 0 indicates that all available results shall be returned.

If the requestType specified is not supported, the Information Source Actor shall return HTTP response-code 404 (not found) with the suggested reason-phrase “requestType not supported”.. If the Information Source Actor is not able to format the document in any content types listed in the 'Accept' field, it shall return HTTP response code 406 – Not Acceptable.

If the Patient ID specified by the Display Actor is not known to the Information Source Actor, it shall return HTTP response-code 404 (not found) with the suggested reason-phrase “Patient ID not found”. If the Display Actor provides the Patient ID from a different domain than the one the Information Source Actor belongs to, and the Information Source Actor is grouped with the Patient ID Consumer Actor, it may attempt to obtain a mapping of the provided Patient ID into its domain before responding.

Note: Other HTTP response codes may be returned by the Information Source Actor, indicating conditions outside of the scope of this profile, for example, 401 – Authentication Failed might be returned if Information Source Actor is grouped with the Kerberized Server Actor.

Note: It is recommended that the Information Source Actor complement the returned error code with a human readable description of the error condition.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Display Actor.

If lowerDateTime and/or upperDateTime parameters are specified, they shall define the lower and/or upper inclusive boundary of the temporal range in which returned information should have been created. The value of the mostRecentResults parameter shall be interpreted within such specified date/time range.

3.11.3.2 Response with Specific Information - Summary

3.11.3.2.1 Trigger Events

This message is sent by the Information Source Actor in response to the Request For Specific Information web service request.

3.11.3.2.2 Message Semantics

Information Source Actor shall support at least one of the values of the requestType parameter specified in Table 3.11.4-2.

The Information Source shall set an expiration of zero to ensure no caching. The message shall be formatted using XHTML Basic and W3C HTML Compatibility Guidelines provided in the Appendix C of the W3C XHTML 1.0 Recommendation.

The Display Actor may request the Information Source Actor to provide any specific information including a summary of reports of different types pertaining to a particular patient. The exact content of the summary is determined by the Information Source Actor and may be regulated by the institution policy. For example, it may contain the hyperlink to a persistent object so that it

can be retrieved by using the Retrieve Document for Display Transaction. In the case of retrieving a summary of documents (requestType of SUMMARY[-xx]), it is strongly recommended to include a link to the relevant documents, for each item of the summary. If present, the link will have to be formatted as a web service request in accordance to the requirements in Section 3.12. It may also contain a hyperlink representing the invocation of the Request for Specific Information for display, as specified in this Section.

3.11.3.2.3 Expected Actions

The Display Actor shall render the received response for the user. It shall not assume that the content of the document may be meaningfully parsed beyond determination of XHTML tags necessary for accurate presentation of provided information.

When the summary responses include links to documents or other specific information, Information Source Actors are strongly encouraged to format them according to the requirements stated in Sections 3.11 and 3.12, to facilitate retrieval of information from other information sources.

3.11.3.3 Request For Specific Information - List

3.11.3.3.1 Trigger Events

The following event will trigger a Request for Specific Information:

- User of the Display Actor needs to review a particular subset of information that is part of a patient's clinical history (i.e., lab report, radiology exam report, list of medications, etc.) that is stored on the Information Source system.

3.11.3.3.2 Message Semantics

The Retrieve Specific Information for Display transaction is performed by the invocation of a web service. The Display Actor shall generate a web service request whenever a user needs to review the information stored as part of a patient's clinical history on the Information Source Actor.

To specify the type of information to be returned, a web service request shall include the following parameters (keys) to filter the subset of information (See Table 3.11.4-7). All parameter names and values (see Table 3.11.4-7) are case-sensitive.

Table 3.11.4-6 Web Service Request Keys

Parameter Name	REQ	Description	Notes
requestType	R	requestType specifies what type of information shall be retrieved. This parameter shall always be valued.	See Table 3.11.4-7 for the list of possible values.

Parameter Name	REQ	Description	Notes
patientID	R	This attribute identifies the subject of the results being queried for. Its value shall include identification of assigning authority.	PatientID value shall be formatted as HL7 CX data type (including assigning authority) according to the requirements specified for the Patient Identity Feed transaction (see Section 3.8.4.1.2.3)

Table 3.11.4-7 Web Service Request Types

requestType value	Description
LIST-ALLERGIES	List of allergies and adverse reactions for a patient known to the Information Source
LIST-MEDS	List of medications currently taken by or administered to a patient

Formal definition of the web service in WSDL is provided in the Appendix A.

The only binding required for both Display Actor and Information Source Actor is the binding to the HTTP-GET. In this binding the sample message will be formatted as follows:

http://<location>/IHERetrieveListInfo?requestType=LIST-MEDS&patientID=99998410^^^%26www.mlhlife.com%26DNS

The <location> part of the URL is configurable by the implementation, and must contain the host name, an optional port address, and may be followed by an optional path. The path if present may not contain a '?' character. The remainder of the URL, including IHERetrieveListInfo and the following request parameters are specified by the WSDL and may not be changed. See the discussion about location in section 3.11.4.1.2 Message Semantics above.

In addition, the Display Actor shall support the following field of the HTTP request:

Table 3.11.4-8 HTTP Request and Response Fields

HTTP Field	REQ	Description	Values
Accept-Language	O	This field restricts the set of natural languages that are preferred as a response to the request.	Any valid value according to RFC2616

The Information Source actor shall support the following field of the HTTP response.

Table 3.11.4-9 HTTP Request Fields

HTTP Field	REQ	Description	Values
Expires	R	This field gives the date/time after which the response is considered stale	Shall be 0. This is now deprecated usage, but it is the widely supported means of specifying no cacheing.
Cache-Control	R	This field indicates that this response should not be cached.	Shall be no-cache

If necessary, the Display Actor may perform the request to the web service utilizing HTTPS protocol.

-

- Information Source Actors may return HTTP redirect responses (responses with values of 301, 302, 303 or 307) in response to a request. Display Actors can expect to receive an error response, or the data requested, or a request to look elsewhere for the data. A Display Actor must follow redirects, but if a loop is detected, it may report an error.

3.11.3.3.3 Expected Actions

Upon reception of the Request for Specific Information, the Information Source Actor shall parse the request and if there are no errors, shall return the Response with Specific Information as specified in Section 3.11.4.2, and HTTP response code 200 - OK.

If the requestType specified is not supported, the Information Source Actor shall return HTTP response-code 404 (not found) with the suggested reason-phrase “requestType not supported”. If the Information Source Actor is not able to format the document in any content types listed in the 'Accept' field, it shall return HTTP response code 406 – Not Acceptable.

If the Patient ID specified by the Display Actor is not known to the Information Source Actor, it shall return HTTP response-code 404 (not found) with the suggested reason-phrase “Patient ID not found”. If the Display Actor provides the Patient ID from a different domain than the one the Information Source Actor belongs to, and the Information Source Actor is grouped with the Patient ID Consumer Actor, it may attempt to obtain a mapping of the provided Patient ID into its domain before responding.

Note: Other HTTP response codes may be returned by the Information Source Actor, indicating conditions outside of the scope of this profile, for example, 401 – Authentication Failed might be returned if Information Source Actor is grouped with the Kerberized Server Actor.

Note: It is recommended that the Information Source Actor complement returned error code with a human readable description of the error condition.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Display Actor.

3.11.3.4 Response with Specific Information - List

3.11.3.4.1 Trigger Events

This message is sent by the Information Source Actor in response to the Request For Specific Information web service request.

3.11.3.4.2 Message Semantics

Information Source Actor shall support at least one of the values of the requestType parameter specified in Table 3.11.4-7.

The Information Source shall set an expiration of zero to ensure no caching. The message shall be formatted using XHTML Basic and W3C HTML Compatibility Guidelines provided in the Appendix C of the W3C XHTML 1.0 Recommendation.

The Display Actor may request the Information Source Actor to provide a list of information items (pertaining to a particular patient) that the Information Source has presently recorded. The exact content of the list is determined by the Information Source Actor.

The Display Actor shall not use the lowerDateTime, upperDateTime or mostRecentResults parameters in a query. The Information Source shall ignore them if they are specified.

3.11.3.4.3 Expected Actions

The Display Actor shall render the received response for the user. It shall not assume that the content of the document may be meaningfully parsed beyond determination of XHTML tags necessary for accurate presentation of provided information.

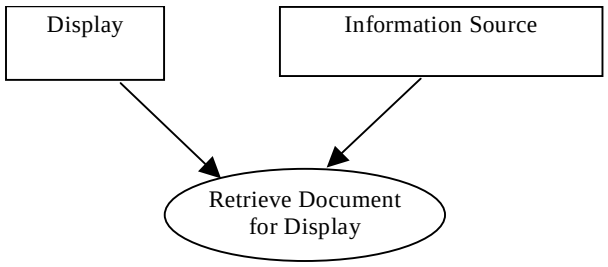
3.12 Retrieve Document for Display

This section corresponds to Transaction ITI-12 of the IHE IT Infrastructure Technical Framework. Transaction ITI-12 is used by the Information Source and Display actors.

3.12.1 Scope

This transaction involves the retrieval of a document (persistent object) for presentation purposes. The uniquely identifiable persistent object means that retrieving the same document instance at a different point in time will provide the same semantics for its presented content. The information content of the document is immutable even if the presentation of such content is provided with the use of different formats, stylesheets, etc.

3.12.2 Use Case Roles



Actor: Display

Role: A system that requests a document/object for display, and displays it.

Actor: Information Source

Role: A system that provides specific information in response to the request from the Display Actor, in a presentation-ready format.

3.12.3 Referenced Standards

IETF RFC2616 HyperText Transfer Protocol HTTP/1.1

Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation 6 October 2000. http://www.w3.org/TR/REC-xml.

Web Services Description Language (WSDL) 1.1. W3C Note 15 March 2001. http://www.w3.org/TR/wsdl.

3.12.4 Interaction Diagram

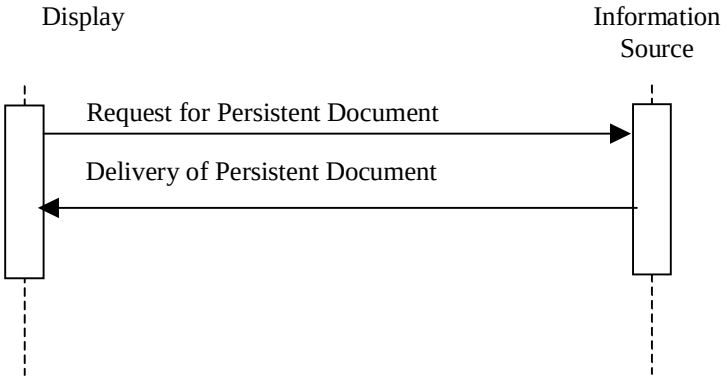


Figure 3.12-1 Request for Persistent Document Sequence

3.12.4.1 Request for Persistent Document

3.12.4.1.1 Trigger Events

The request for a document is triggered when a user of the Display Actor needs to review a particular document that is stored by the Information Source Actor.

3.12.4.1.2 Message Semantics

The Retrieve Document for Display transaction is performed by the invocation of a web service. The Display Actor shall generate the web service request whenever a user needs to review the document stored as part of a patient's clinical history on the Information Source Actor.

The web service request shall include the following parameters (keys) to identify the document to be returned and its format See Table (3.12.4-1). All parameter names and values are case-sensitive.

Table 3.12.4-1 Query Keys

Parameter Name	REQ	Description	Values
requestType	R	This parameter is required to have a value of DOCUMENT.	DOCUMENT
documentUID	R	Identifies document's UID as known to both actors.	This value shall be a properly defined Object identifier (OID) as specified in Volume 2, Appendix B.
preferredContentType	R	This parameter is required to identify the preferred format the document is to be provided in (as MIME content type).	Display may specify one of the following formats: image/jpeg application/x-hl7-cda-level-one+xml (see note) application/pdf (see note)

Note: see IANA registry for details about hl7-cda-level-one and PDF, such as version. Applications creating PDF may use this MIME type for other versions of PDF up to 1.3. Receivers shall support document encoded in this version and previous versions.

Note: see HL7 CDA framework release 1.0 for details about application/x-hl7-cda-level-one+xml.

Formal definition of the web service in WSDL is provided in Appendix A.

The only binding required for both the Display Actor and Information Source Actor is the binding to the HTTP-GET. In this binding the sample message will be formatted as follows:

http://<location>/IHERetrieveDocument?requestType=DOCUMENT&documentUID=1.2.3
&preferredContentType=application%2fpdf

The <location> part of the URL is configurable by the implementation, and must contain the host name, an optional port address, and may be followed by an optional path. The path if present may not contain a '?' character. The remainder of the URL, including IHERetrieveDocument and the following request parameters are specified by the WSDL and may not be changed. See the discussion about location in section 3.11.4.1.2 Message Semantics above.

In addition, the Display Actor shall support the following fields of the HTTP request:

Table 3.12.4-3 HTTP Request and Response Fields

HTTP Field	REQ	Description	Values
Accept	O	This field may be used to specify certain media types which are acceptable for the response	At least one of the following values: image/jpeg application/x-hl7-cda-level-one+xml application/pdf */* Other values may be included as well
Accept-Language	O	This field is similar to Accept, but restricts the set of natural languages that are preferred as a response to the request.	Any valid value according to RFC2616
Expires	R	This field gives the date/time after which the response is considered stale	Any valid value according to RFC2616, or 0

The Information Source actor shall support the following field of the HTTP response.

Table 3.12.4-4 HTTP Response Fields

HTTP Field	REQ	Description	Values
Expires	R	This field gives the date/time after which the response is considered stale	Any valid value according to RFC2616, or 0

The Display Actor may provide list of content types it supports in the HTTP Accept field. If the HTTP Accept Field is absent, it means that any content type is acceptable by the Display Actor.

The preferredContentType parameter shall specify the content type desired by the Display Actor. The value of the preferredContentType parameter of the request shall be one of the values from the Table 3.12.4-1 and shall not contradict values specified in the HTTP Accept field.

The Information Source shall provide info in preferredContentType if capable, otherwise it shall only use a type specified in the Accept Field as appropriate given the information to be returned.

If necessary, the Display Actor may perform the request to the web service utilizing HTTPS protocol.

-

- Information Source Actors may return HTTP redirect responses (responses with values of 301, 302, 303 or 307) in response to a request. Display Actors can expect to receive an error response, or the data requested, or a request to look elsewhere for the data. A Display Actor must follow redirects, but if a loop is detected, it may report an error.

3.12.4.1.3 Expected Actions

Upon reception of the Request for Specific Information, the Information Source Actor shall parse the request and shall return the retrieved document as specified in Section 3.12.4.2, and HTTP response code 200 - OK.

If the requestType specified is a not a legal value according to this profile, the Information Source Actor shall return HTTP response-code 403 (forbidden) with the suggested reason-phrase "requestType not supported".

If the Information Source Actor is not able to format the document in any content types listed in the 'Accept' field, it shall return HTTP response code 406 – Not Acceptable.

If the specified documentUID is not known to the Information Source Actor, it shall return HTTP response-code 404 (not found) with the suggested reason-phrase "Document UID not found".

If the documentUID, preferredContentType or requestType parameters are missing, the Information Source Actor shall return HTTP response code 400 - Bad Request.

If the documentUID or preferredContentType parameters are malformed, the Information Source Actor shall return HTTP response code 400 - Bad Request.

If the specified preferredContentType is not consistent with the setting of the HTTP Accept field, the Information Source Actor shall return HTTP response code 400 – Bad Request.

Note: Other HTTP response codes may be returned by the Information Source Actor, indicating conditions outside of the scope of this profile, for example, 401 – Authentication Failed might be returned if Information Source Actor is grouped with the Kerberized Server Actor.

Note: It is recommended that the Information Source Actor complement returned error code with a human readable description of the error condition.

If an error condition cannot be automatically recovered, at a minimum, the error should be displayed to the user by the Display Actor.

3.12.4.2 Delivery of Persistent Document

3.12.4.2.1 Trigger Events

The Delivery of Persistent Document message is the transmission of the requested document in specified format from the Information Source Actor to the Display Actor. This transmission will happen if such document, identified by the documentUID parameter in the request, has been successfully located by the Information Source Actor.

3.12.4.2.2 Message Semantics

In response to the request from the Display Actor, the Information Source Actor shall format the document according to the preferredContentType specified, and return it in the HTTP response. See Section 3.12.4.1.2 for a discussion of the rules related to preferredContentType.

The Information Source Actor shall maintain global uniqueness of object identifiers.

The Information Source Actor shall set an expiration date compatible with the policies associated with the possible removal of instances of persistent documents (no more than a week).

3.12.4.2.3 Expected Actions

The Display Actor shall render the received document for the user.

3.13 Follow Context

This section corresponds to Transaction ITI-13 of the IHE IT Infrastructure Technical Framework. Transaction ITI-13 is used by the Patient Context Participant, User Context Participant and Context Manager Actors.

3.13.1 Scope

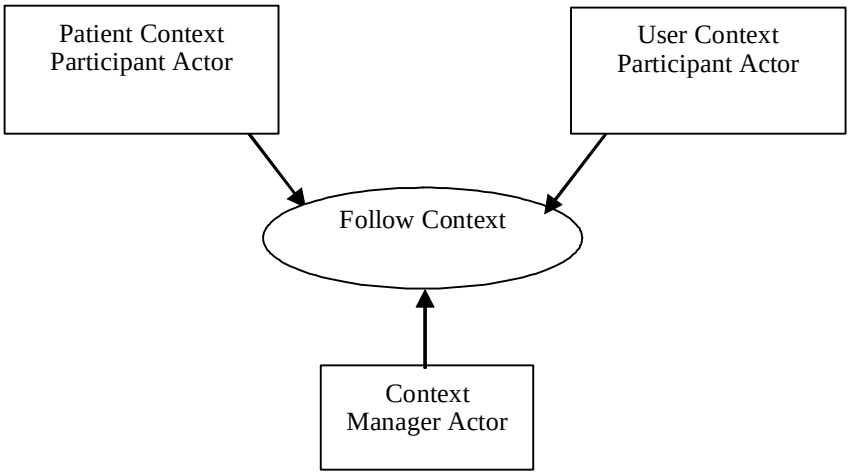
This transaction allows the Context Manager Actor to force other context participant actors to synchronize based on the new context values.

This transaction is composed of multiple methods as defined by the *HL7 Context Management "CCOW" Standard*. It has multiple phases consisting of surveying the participants, indication to them of final decision as to whether the context changed or not, and retrieval of the new context values by the context participants.

Each of the context participant actors follows a specific subject. The Patient Context Participant Actor follows the patient subject and does not expect the user subject to be set in context. The User Context Participant follows the user subject.

The semantics of the methods used are defined in the documents HL7 Context Management "CCOW" Standard: Component Technology Mapping: ActiveX or HL7 Context Management "CCOW" Standard: Component Technology Mapping: Web, in conjunction with the HL7 Context Management "CCOW" Standard: Subject Data Definitions document. A Context Participant Actor can implement either technology. The Context Manager Actor shall support both technologies in order to interoperate with joining participants implementing the technology of their choice.

3.13.2 Use Case Roles



Actor: Patient Context Participant

Role: Responds to context survey. Synchronizes display to new value(s) in the patient subject of a context it follows.

Actor: User Context Participant

Role: Responds to context survey. Synchronizes display to new value(s) in the patient subject of a context it follows.

Actor: Context Manager

Role: Conducts context survey, notifies the context participants of acceptance or cancellation of a change, and provides context values.

3.13.3 Referenced Standard

HL7 Context Management "CCOW" Standard, Version 1.4

Technology and Subject Independent Architecture

Component Technology Mapping: ActiveX

Component Technology Mapping: Web

Subject Data Definitions

3.13.4 Interaction Diagram

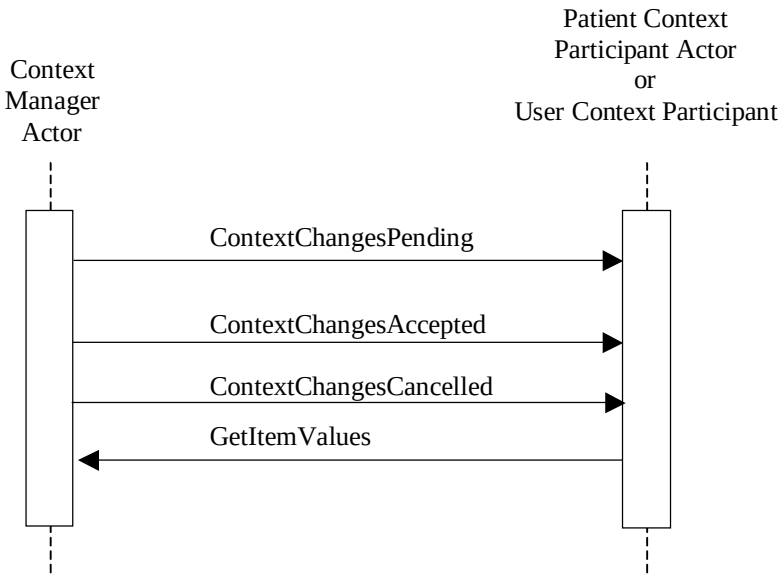


Figure 3.13-1 Follow Context – ContextChangesPending Method Sequeice

3.13.4.1 Follow Context – ContextChangesPending Method

The ContextChangesPending method is invoked by the Context Manager Actor to survey context participant actors with regard to acceptability of changes proposed by a Patient Context Participant or Client Authentication Agent Actors.

3.13.4.1.1 Trigger Events

The ContextChangesPending method is triggered when the Context Manager receives invocation of the EndContextChanges method.

3.13.4.1.2 Message Semantics

ContextChangesPending is defined as a method on the ContextParticipant interface and allows the Context Manager to survey a context participant as to whether or not it is ready to follow the changes in the context.

In the invocation of this method, the Context Manager shall provide the pending context's coupon.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.7.2, for a description of the parameters associated with this method.

3.13.4.1.3 Expected Actions

Performing the ContextChangesPending method, the Patient Context Participant or User Context Participant Actor makes a decision whether or not it can accept change of context (for example due to operation being in progress). To reach this decision, it may invoke the GetItemValues method to inspect proposed new values in the context.

As a response, a Context Participant Actor will respond with an indication to Accept or Conditionally Accept the proposed change. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.7.2, for the specifics of the response formation.

3.13.4.2 Follow Context – ContextChangesAccepted Method

The ContextChangesAccepted method is invoked by the Context Manager Actor to confirm to the context participants that instigator of change accepted proposed changes.

3.13.4.2.1 Trigger Events

The ContextChangesAccepted method is triggered when the Context Manager receives invocation of the PublishChangesDecision method indicating that the changes have been accepted.

3.13.4.2.2 Message Semantics

ContextChangesAccepted is defined as a method on the ContextParticipant interface and allows the Context Manager to inform a context participant that the context value(s) have been changed.

In the invocation of this method, the Context Manager provides the new context coupon.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture,* Section 17.3.7.3 for a description of the parameters associated with this method.

3.13.4.2.3 Expected Actions

Performing the ContextChangesAccepted method, the Patient Context Participant or User Context Participant Actor accepts new context and can subsequently retrieve new values using the GetItemValues method.

It responds with confirmation of success or an exception. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.7.3, for the specifics of the response formation.

3.13.4.3 Follow Context – ContextChangesCancelled Method

The ContextChangesCancelled method is invoked by the Context Manager Actor to inform the context participants that instigator of change cancelled proposed changes.

3.13.4.3.1 Trigger Events

The ContextChangesCancelled method is triggered when the Context Manager receives invocation of the PublishChangesDecision method indicating that the changes have been cancelled.

3.13.4.3.2 Message Semantics

ContextChangesCancelled is defined as a method on the ContextParticipant interface and allows the Context Manager inform a context participant that the pending context change has been cancelled.

In the invocation of this method, the Context Manager provides the pending context's coupon.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture,* Section 17.3.7.4 for a description of the parameters associated with this method.

3.13.4.3.3 Expected Actions

Performing the ContextChangesCancelled method, the Patient Context Participant or User Context Participant Actor keeps its current context and destroys information about a pending context change that has been cancelled.

It responds with confirmation of success or an exception. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture* document, Section 17.3.7.4, for the specifics of the response formation.

3.13.4.4 Follow Context – GetItemValues Method

The GetItemValues method is invoked by a Context Participant Actor to retrieve value(s) from the context it follows.

3.13.4.4.1 Trigger Events

The GetItemValues method is triggered by a Context Participant Actor after it receives the context coupon as a result of the ContextChangesPending, ContextChangesAccepted or GetContextCoupon methods.

3.13.4.4.2 Message Semantics

GetItemValues is defined as a method on the ContextData or SecureContextData interface. If the context is not secured when a participant actor has joined the context (i.e., Patient Context Participant that only follows patient context), then this method should be invoked on the ContextData interface. Otherwise, it shall be invoked on the SecureContextData interface.

By invocation of this method without specification of the list of item names, a context participant retrieves values of all items presently set in context. It can also first invoke the GetItemNames method on the same interface (as specified in CCOW Standard) and use the list of items for selective retrieval of item values from the context via GetItemValues method. The Patient Context Participant needs to search through the resulting list of Patient.Id.IdList.<n> values until a recognized Patient Domain is found. The Patient Context Participant may choose to be grouped with a PIX Patient Identifier Cross-reference Consumer to handle the cases where no known Patient Domain is found in the resulting IdList.

Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture document*, Section 17.3.4.5, for the Patient Context Participant Actor, and Section 17.3.13.2, for the User Context Participant, for a description of parameters associated with this method.

3.13.4.4.3 Expected Actions

Context Manager shall return the values of requested items or an exception. Refer to the *HL7 Context Management "CCOW" Standard: Technology and Subject-Independent Architecture document*, Section 17.3.4.5, for the Patient Context Participant Actor, and Section 17.3.13.2, for the User Context Participant, for a description of the response issued by the Context Manager Actor.

3.14 Register Document Set

This section corresponds to Transaction ITI-14 of the IHE IT Infrastructure Technical Framework. Transaction ITI-14 is used by the Document Repository Actor to register a set of documents with the Document Registry.

3.14.1 Scope

The Register Document Set transaction passes a Submission Request from a Document Repository Actor to a Document Registry Actor.

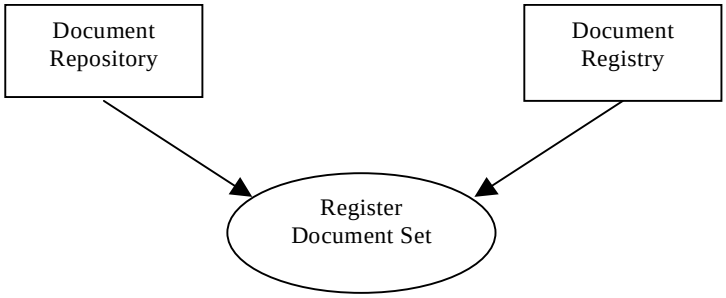
A Register Document Set transaction carries:

Metadata describing zero or more documents

XDS Submission Set definition along with the linkage to new documents and references to existing documents

XDS Folder definitions along with linkage to new or existing documents

3.14.2 Use Case Roles



Actor: Document Repository

Role: A document storage system that submits document metadata to a Document Registry.

Actor: Document Registry

Role: A document indexing system that receives and stores document metadata.

3.14.3 Referenced Standards

ebRIM OASIS/ebXML Registry Information Model v2.0

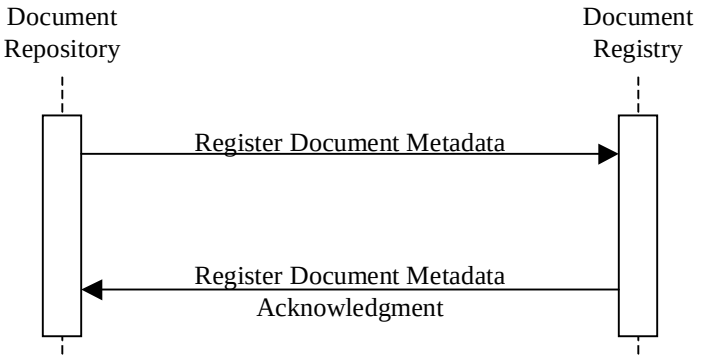
ebRS OASIS/ebXML Registry Services Specifications v2.0

HTTP HyperText Transfer Protocol HTTP/1.1 (IETF RFC2616)

CDA HL7 Clinical Document Architecture (ANSI/HL7 CDA R1-2000)

HL7V2 HL7 Version 2.5

3.14.4 Interaction Diagram



3.14.4.1 Register Document Metadata

The Document Repository sends metadata for a set of documents to the Document Registry.

3.14.4.1.1 Trigger Events

The Register Document Metadata message is triggered when:

1. A Document Repository wants to register metadata for a set of documents it holds.
2. A Document Repository receives a Provide and Register Document Set transaction (ITI-15)

3.14.4.1.2 Message Semantics

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

Metadata definitions to support the above are discussed as follows:

XDS Document

XDS Submission Request

XDS Submission Set

XDS Folder

In addition the following topics are discussed:

Protocol requirements

XDS registry adaptor function

General metadata issues

Security requirements

Sequencing Requirements

3.14.4.1.2.1 Class Diagram

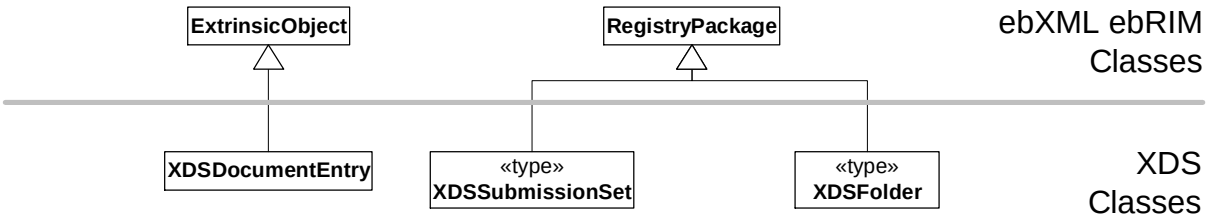


Figure 3.14.4.1-1 ebXML Class Diagram of the Register Document Metadata

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.

3.14.4.1.2.2 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.

[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.

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.

Appendix H defines the metadata to initialize an ebXML registry to serve as an XDS Document Registry.

3.14.4.1.2.3 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 section 3.14.4.1.2.11.

The two types of XDS Submission Requests are described next.

3.14.4.1.2.3.1 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, contained within the metadata, organizing the metadata

This request is part of the Register Document Set transaction.

3.14.4.1.2.3.2 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 transaction ITI-15.

Unless otherwise stated, the XDS Submission Set requirements specified hereafter apply to both types of XDS Submission Requests

3.14.4.1.2.3.3 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 section 3.14.4.1.2.12 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 be 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.

3.14.4.1.2.3.4 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).
- 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.

3.14.4.1.2.3.5 Attribute Size

All attribute values shall conform to the size specification of ebRIM version 2.1 that is detailed in section 7.2 Data Types of that specification. More specifically, all Slots shall conform to the specification of ebRIM version 2.1, which is detailed in section 7.6.1 of that specification. The version 2.0-ebRIM specification is overly limiting in this respect. Without adopting the newer size limits, many typical patient record values could not be encoded.

3.14.4.1.2.4 Submission Set Specifications

Submission Sets exist for two reasons:

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

An XDS SubmissionSet is an ebXML RegistryPackage, classified as XDSSubmissionSet that is used to bundle XDSDocumentEntry objects.

A Submission Set has a set of attributes that are described in section 3.14.4.1.2.8 Submission Set Metadata.

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 3.14.4.1-1 Submission Set Association Labeling

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

3.14.4.1.2.5 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.

3.14.4.1.2.6 Document Relationships and Associations

3.14.4.1.2.6.1 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 3.14.4.1-2. 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 discovered 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 Appendix H for details.

Table 3.14.4.1-2 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.

Adapted from HL7 CDA Release 2, Committee Ballot 2

A Document Relationship refers to any of the relationships listed in Table 3.14.4.1-2 above. Section 3.15.5.1 documents for the Document Source which of these operations are required and optional.

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 3.14.4.1-2.

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.

The Registry Adaptor manages document deprecation. See section 3.14.4.1.2.12 XDS Registry Adaptor for details.

Table 3.14.4.1-3 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 Assocition 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 document (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 Vol-1: 10.4.11.1 for further detail on the use and meaning of document relationships.

3.14.4.1.2.6.2 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.

3.14.4.1.2.7 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 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 3.14.4.1-3 Data Types

XDS Data Type	Source Standard	Encoding Specification
CX	HL7 V2	This is an identifier. HL7 Identifier type CX consist of

	Identifier	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 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.
		The format of these values is defined as the following regular expression:
		YYYY[MM[DD[hh[mm[ss]]]]]
		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-2 : 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`
URI	Uniform Resource Identifer	See RFC 2616 `http://www.ihe.net`
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 includes the ID Number and Name of a person, specified with first 6 components of the HL7 data type XCN: Identifier Last Name First Name Second and Further Given Names Suffix Prefix A example of person name with ID number using this data type is as follows: `11375^Welby^Marcus^J^Jr. MD^Dr.`
XON	HL7 V2 Organization Name	This is the organization name, specified with the first component (XON.1) of the HL7 data type XON: Organization Name An example of organization name using this data type is as follows: `Fairview Hospital`

The source/query column indicates which attributes are required, and whether they may be used in queries according to the table below.

Table 3.1.14.1-4 Codes for Source/Query Column

Code	Meaning
R	Required
R2	Required if Known
O	Optional
P	Not supported in query.
Cp	Computed/Assigned by Repository, required in register transaction.
Cg	Computed/Assigned by Registry

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 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.URI shall be supplied by the Document Repository Actor. Its value is dependent on how the repository stores the document.

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 3.14.4.1-5 Document Metadata Attribute Definition

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
authorInstitution	Represents a specific healthcare facility under which the human and/or machines authored the document. A specific case is that of homecare.	R2/R	XON

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

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	<rim:Slot name="authorInstitution"> <rim:ValueList> <rim:Value>Fairview Hospital</rim:Value> </rim:ValueList> </rim:Slot>		
authorPerson	Represents the humans and/or machines that authored the document within the authorInstitution. The document author may be the patient itself. This attribute may be multi-valued. <rim:Slot name="authorPerson"> <rim:ValueList> <rim:Value>^Welby^Marcus^^^Dr^MD</rim:Value> </rim:ValueList> </rim:Slot>	R2/R	XCN
authorRole	A code that represents the role of the author with respect to the patient when the document was created. <rim:Slot name="authorRole"> <rim:ValueList> <rim:Value>***theAuthorRole***</rim:Value> </rim:ValueList> </rim:Slot>	R2/O	
authorSpecialty	Represents a specific specialty within a healthcare facility under which the human and/or machines authored the document. <rim:Slot name="authorSpecialty"> <rim:ValueList> <rim:Value>***theAuthorSpecialty***</rim:Value> </rim:ValueList> </rim:Slot>	R2/O	
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. The example below shows the status attribute, however, this	Cg/R	

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type

	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="Approved"**`
	`  mimeType="application/octet-stream"`
	`> …`		

classCode	The code specifying the particular kind of document (e.g. Prescription, 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).
	`<rim:Classification`
	`  classificationScheme=`
	`  "urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"`
	`  classifiedObject="theDocument"`
	`  nodeRepresentation="`***`classCode`***`"`
	`>`
	`  <rim:Name>`
	`  <rim:LocalizedString`
	`  value="`***`classCodeDisplayName`***`"/>`
	`  </rim:Name>`
	`  <rim:Slot name="codingScheme">`
	`  <rim:ValueList>`
	`  <rim:Value>Affinity Domain Specific Value</rim:Value>`
	`  </rim:ValueList>`
	`  </rim:Slot>`
	`</rim:Classification>`
		R/R	XDS Affinity Domain specific

classCode DisplayName	The name to be displayed for communicating to a human the meaning of the classCode. See classCode for example.	R/P	XDS Affinity Domain specific

confidentialityCode	The code specifying the level of confidentiality of the XDS Document. These codes are specific to an 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.
	`<rim:Classification`
	`  classificationScheme=`
		R/P	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	"urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f" classifiedObject="theDocument" nodeRepresentation="***confidentialityCode***" > <rim:Name> <rim:LocalizedString value="displayName"/> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>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. <rim:Slot name="creationTime"> <rim:ValueList> <rim:Value>20041225212010</rim:Value> </rim:ValueList> </rim:Slot>	R/R	DTM
entryUUID	The globally unique identifier (may be assigned by either by Source, Repository, or Registry) is primarily intended for use as a document registry management identifier. It is not meant to be an external reference 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. 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" > …	Cg/P	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	O/R	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	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. <rim:Classification classificationScheme= "urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4" classifiedObject="theDocument" nodeRepresentation="***eventCode***" > <rim:Name> <rim:LocalizedString value="***eventCodeDisplayName***"/> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
eventCodeDisplay NameList	The list of names to be displayed for communicating to human reader the meaning of the eventCode. See eventCodeList for an example.	O[3]/P	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). <rim:Classification classificationScheme= "urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d" classifiedObject="theDocument" nodeRepresentation="***formatCode***" > <rim:Name> <rim:LocalizedString value="name"/> </rim:Name>	R/O	XDS Affinity Domain specific

[3] Required if eventCode has a value.

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	<rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
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. <rim:Slot name="hash"> <rim:ValueList> <rim:Value> da39a3ee5e6b4b0d3255bfef95601890afd80709 </rim:Value> </rim:ValueList> </rim:Slot>	Cp/P	SHA1 hash
healthcareFacility TypeCode	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. <rim:Classification classificationScheme= "urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1" classifiedObject="theDocument" nodeRepresentation="***healthcareFacilityTypeCode***" > <rim:Name> <rim:LocalizedString value="***healthcareFacilityTypeCodeDisplayName***"/> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>	R/R	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
healthcareFacility TypeCodeDisplay Name	The name to be displayed for communicating to a human the meaning of the healthcareFacilityTypeCode See healthcareFacilityTypeCode for an example.	R/P	XDS Affinity Domain specific

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.
	`<rim:Slot name="languageCode">`
	`  <rim:ValueList>`
	`  <rim:Value>en-us</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		R/P	

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

mimeType	MIME type of the document in the Repository.
	`<rim:ExtrinsicObject mimeType="`***`application/pdf`***`"`
	`  id="theDocument"`
	`  objectType= "urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"`
	`>` …
		R/P	

parentDocumentId	The identifier of the parentDocument entry that represents the source of a document replacement, addendum, transformation, or signs relationship. May identify a document which is unknown by the Document Registry. `<rim:ObjectRef`	R/P	ebRIM Association

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	id="***urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6***" /> <rim:Association associationType="***parentDocumentRelationShip***" sourceObject="theDocument" targetObject="***urn:uuid:a6e06ca8-0c75-4064-9e5c-88b9045a96f6***" /> If the parent document is in the registry then code as association to it, otherwise create a stub document object, and use its entryUUID as the value for the targetObject attribute. A document stub represents a document that is not in registry but is needed by another object to point at. This association is coded with a type from parentDocumentRelationship. A document may have a single relationship via the RPLC, APND, or XFRM association types. There is no restriction on the number of signs relationships that a document may be part of.		
parentDocumentRelationship	The type of relationship that the document has with the parentDocument (e.g. Replace, addendum, transformation, or signs). See parentDocumentID for an example.	R/P	Use one of the following values: APND RPLC XFRM signs
patientId	The patientId represents the subject of care medical record identifier as selected by the Document Source. This identifier shall be from the Assigning Authority Domain supporting the 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. The value of the patientId shall be the same for all new	R/R	CX

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type

	documents of a Submission Set.
	
	`<rim:ExternalIdentifier`
	`identificationScheme=`
	`"urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"`
	`value="`***`6578946^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO`***`"`
	`>`
	`<rim:Name>`
	`<rim:LocalizedString value =`
	`"XDSDocumentEntry.patientId"/>`
	`</rim:Name>`
	`</rim:ExternalIdentifier>`		

practiceSettingCode	The code specifying the clinical specialty where the act that resulted in the document was performed (e.g. Familly 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)
	
	`<rim:Classification`
	`classificationScheme=`
	`"urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead"`
	`classifiedObject="theDocument"`
	`nodeRepresentation="`***`practiceSettingCode`***`"`
	`>`
	`<rim:Name>`
	`<rim:LocalizedString`
	`value="`***`practiceSettingCodeDisplayName`***`"`
	`/>`
	`</rim:Name>`
	`<rim:Slot name="codingScheme">`
	`<rim:ValueList>`
	`<rim:Value>Affinity Domain Specific Value</rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	`</rim:Classification>`
		R/R	XDS Affinity Domain specific

practiceSettingCode DisplayName	The name to be displayed for communicating to a human the meaning of the practiceSettingCode. See practiceSettingCode for an example.	R/P	XDS Affinity Domain specific
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. This time is expressed as (date/time/UTC).	R2/R	HL7 V2 DTM

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type

	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>`		

serviceStopTime	Represents the stop 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. 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.
	
	`<rim:Slot name="serviceStopTime">`
	`  <rim:ValueList>`
	`  <rim:Value>20041225232010</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		R2/R	HL7 V2 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.
	
	`<rim:Slot name="size">`
	`  <rim:ValueList>`
	`  <rim:Value>3654</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		Cp/P	Integer

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	R/P	CX

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	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. <rim:Slot name="sourcePatientId"> <rim:ValueList> <rim:Value>j98789^^^id.domain</rim:Value> </rim:ValueList> </rim:Slot>		
sourcePatientInfo	This attribute contains demographics information of the patient to whose medical record this document belongs, as the Document Source knew it at the time of Submission. This information typically includes: the patient first and last name, sex, and birth date. The Clinical Affinity Domain policies may require more 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. <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>	R[4]/P	

[4] Certain segments are required, see definition.

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	PID-3 is required and must include the source patient identifier. PID-5 is required and must include the patient name. PID-8 is required and must code the patient gender as M – Male F – Female O – Other U – Unknown PID-7 is required if known, and must include the patient date of birth. PID-11 is required if known, and must include the patient address. PID-2, PID-4, PID-12 and PID-19 should not be used. Other PID segments are optional.		

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.
	`<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/P	

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.
	`<rim:Classification`
	`classificationScheme=`
	`"urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"`
	`classifiedObject="theDocument"`
	`nodeRepresentation="`***`typeCode`***`"`
	`>`
		R/R	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	<rim:Name> <rim:LocalizedString value="***typeCodeDisplayName***" /> </rim:Name> <rim:Slot name="codingScheme"> <rim:ValueList> <rim:Value>Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
typeCodeDisplayName	The name to be displayed for communicating to a human the meaning of the typeCode. See typeCode for an example.	R/P	XDS Affinity Domain specific

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
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). 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. The means to ensure that when a XDS Document is retrieved from the XDS Document Repository using the URI component, the selected XDS Document is the correct one. `<rim:ExternalIdentifier identificationScheme= "urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab" value="`***`1.3.6.1.4.1.21367.2005.3.7^11379`***`" > <rim:Name> <rim:LocalizedString value="XDSDocumentEntry.uniqueId"/> </rim:Name> </rim:ExternalIdentifier>`	R/R	See section 3.14.4.1.2.7.2
URI	The URI of the XDS Document to be used for retrieval. XDS does not constraint the format of this URI beyond RFC 2616. However, the IHE Retrieve Information for Display Integration Profile defined format may be used in cases where the Document repository is grouped with a RID Information Source Actor (See ITI TF-1:Appendix E.5) RID links can be used only if they yield the document in full	Cp/P	URI

XDSDocumentEntry Attribute	Definition	Source/ Query	Data Type
	fidelity. <rim:Slot name="URI"> <rim:ValueList> <rim:Value>http://www.ihe.net</rim:Value> </rim:ValueList> </rim:Slot>		

3.14.4.1.2.7.1 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.

3.14.4.1.2.7.2 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.

3.14.4.1.2.8 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.

2345 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-2 : Appendix B.

Table 3.14.4.1-6 Submission Set Metadata Attribute Definitions

XDSSubmission Set Attribute	Definition	Source/ Query	Data Type

authorInstitution	Represents a specific healthcare facility under which the human and/or machines authored the Submission Set.
	
	`<rim:Slot name="authorInstitution">`
	`  <rim:ValueList>`
	`  <rim:Value>Fairview Hospital</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		R2/R	XON

authorPerson	Represents the human and/or machines that authored the Submission Set. The document author may be the patient itself.
	
	`<rim:Slot name="authorPerson">`
	`  <rim:ValueList>`
	`  <rim:Value>^Welby^Marcus^^^Dr^MD</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		O/R	XCN

authorRole	A code that represents the role of the author with respect to the patient when the document was created.
	
	`<rim:Slot name="authorRole">`
	`  <rim:ValueList>`
	`  <rim:Value>`***`theAuthorRole`***`</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		R2/O	

authorSpecialty	Represents a specific specialty within a healthcare facility under which the human and/or machines authored the document.
	
	`<rim:Slot name="authorSpeciality">`
	`  <rim:ValueList>`
	`  <rim:Value>`***`theAuthorSpeciality`***`</rim:Value>`
	`  </rim:ValueList>`
	`</rim:Slot>`
		R2/O	

comments	Comments associated with the Submission Set. Free form text with an Affinity Domain specified usage.
	
	`<rim:Description>`
	`  <rim:LocalizedString value = "`***`comments`***`"/>`
	`</rim:Description>`
		R2/R	Use specific to XDS Affinity Domain.

XDSSubmission Set Attribute	Definition	Source/ Query	Data Type

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 Affinity Domain.
	
	`<rim:Classification`
	`classificationScheme=`
	`"urn:uuid:aa543740-bdda-424e-8c96-df4873be8500"`
	`classifiedObject="submissionSet"`
	`nodeRepresentation="`***`contentTypeCode`***`"`
	`>`
	`<rim:Name>`
	`<rim:LocalizedString`
	`value="`***`contentTypeCodeDisplayName`***`" />`
	`</rim:Name>`
	`<rim:Slot name="codingScheme">`
	`<rim:ValueList>`
	`<rim:Value>Affinity Domain Specific Value</rim:Value>`
	`</rim:ValueList>`
	`</rim:Slot>`
	`</rim:Classification>`
		R/R	XDS Affinity Domain specific

contentTypeCode DisplayName	The name to be displayed for communicating to a human the meaning of the contentTypeCode. See contentTypeCode for an example.	R/P	XDS Affinity Domain specific
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 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. The value of the patientId shall be the same for all new documents of a Submission Set.	R/R	CX

XDSSubmission Set Attribute	Definition	Source/ Query	Data Type
	<rim:ExternalIdentifier identificationScheme= "urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427" value="***6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO***" > <rim:Name> <rim:LocalizedString value = "patientId"/> </rim:Name> </rim:ExternalIdentifier>		
sourceId	Globally unique identifier for the instance of the Document Source that contributed the Submission Set. The assigning authority for these identifiers is specified by the XDS Affinity Domain. 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. <rim:ExternalIdentifier identificationScheme= "urn:uuid:554ac39e-e3fe-47fe-b233-965d2a147832" value="8449607624^^^***&1.3.6.1.4.1.21367.2005.3.7&ISO***" > <rim:Name> <rim:LocalizedString value = "XDSSubmissionSet.sourceId"/> </rim:Name> </rim:ExternalIdentifier>	R/R	CX
submissionTime	Point in Time at the Document Source when the Submission Set was created and issued for registration to the Document Registry. 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/R	DTM
uniqueId	Globally unique identifier for the submission-set instance assigned by the Document Source in OID format. <rim:ExternalIdentifier identificationScheme= "urn:uuid:4b052cba-b03b-4233-8b27-e8d5e3f8d3e4" value="***1.3.6.1.4.1.21367.2005.3.7.3670984664***"> <rim:Name> <rim:LocalizedString value =	R/R	OID See Appendix B

XDSSubmission Set Attribute	Definition	Source/ Query	Data Type
	"XDSSubmissionSet.uniqueId"/> </rim:Name> </rim:ExternalIdentifier>		

3.14.4.1.2.9 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-2 : Appendix B.

Table 3.14.4.1-7 Folder Metadata Attribute Definitions

XDSFolder Attribute	Definition	Source/ Query	Data Type
codeList	The list of codes specifying the type of clinical activity that resulted in placing these XDS Documents in this XDSFolder. These values are to be drawn for a vocabulary or coding scheme defined by the Clinical Affinity Domain. When a new submission request associates XDS Documents (new submission or previously submitted) to an XDS Folder, the Code included in the codeList is appended to the existing list of codes for this Folder (if any) unless this code is already present in the list managed by the Registry for the same XDS-Folder. Only one code may be assigned to the Folder when a XDS Document is placed in a Folder. <rim:Classification classificationScheme= 'urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5' classifiedObject='Foloder' nodeRepresentation='***codeList***'	R/R	Multi-Valued. XDS Affinity Domain specific

XDSFolder Attribute	Definition	Source/ Query	Data Type
	> <rim:Name> <rim:LocalizedString value='***codeListCodeDisplayName***' /> </rim:Name> <rim:Slot name='codingScheme'> <rim:ValueList> <rim:Value>Affinity Domain Specific Value</rim:Value> </rim:ValueList> </rim:Slot> </rim:Classification>		
codeDisplayName List	The list of human readable descriptions of the meaning of each on of the codes present in the codeList. Only one code may be assigned to the Folder when a XDS Document is placed in such a Folder. See codeList for an example.	R/P	Multi-valued.
comments	Comments associated with the Folder. Free form text with an Affinity Domain specified usage. <rim:Description> <rim:LocalizedString value = "***comments***"/> </rim:Description>	R2/R	XDS Affinity Domain specific.
lastUpdateTime	Point in time at the Document Registry when an XDS Document was registered and placed in the XDS Folder. <rim:Slot name="submissionTime"> <rim:ValueList> <rim:Value>20041225212010</rim:Value> </rim:ValueList> </rim:Slot>	Cg/R	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 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.
	The value of the patientId shall be the same for all new documents of a Folder.
	<rim:ExternalIdentifier
	  identificationScheme=
	  "urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"
		R/R	CX

XDSFolder Attribute	Definition	Source/ Query	Data Type
	value="***6578946^^^&1.3.6.1.4.1.21367.2005.3.7&ISO***" > <rim:Name> <rim:LocalizedString value = "patientId"/> </rim:Name> </rim:ExternalIdentifier>		
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. <rim:ExternalIdentifier identificationScheme= "urn:uuid:4b052cba-b03b-4233-8b27-e8d5e3f8d3e4" value="***1.3.6.1.4.1.21367.2005.3.7.3670984664***"/>	R/R	OID See Appendix B

3.14.4.1.2.10 Registry Adaptor Enforcement of Attributes

2365 **Table 3.14.4.1-8 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 match classCode
confidentialityCode	Coding Scheme and Code Value
creationTime	No enforcement
entryUUID	No enforcement
eventCodeList	Coding Scheme and Code Value

eventCodeDisplayNameList	Must match eventCodeList
formatCode	Coding Scheme and Code Value
hash	No enforcement
healthcareFacilityTypeCode	Coding Scheme and Code Value
healthcareFacilityTypeCodeDisplayName	Must match healthcareFacilityTypeCode
legalAuthenticator	No enforcement
languageCode	Optionally enforced by Affinity Domain
mimeType	Code Value
parentDocumentRelationship	One of three values
parentDocumentId	Existing UUID
patientId	Authority Domain Id Patient Id (known from patient identity feed)
practiceSettingCode	Coding Scheme and Code Value
practiceSettingCode DisplayName	Must match practiceSettingCode
serviceStartTime	No enforcement
serviceStopTime	Verifies serviceStartTime <= serviceStopTime
size	No enforcement
sourcePatientId	No enforcement
sourcePatientInfo	No enforcement
Title	No enforcement
typeCode	No enforcement
typeCodeDisplayName	Must match typeCode

uniqueId	No identical existing uniqueId in registry (assigned to XDSDocumentEntry, XDSSubmissionSet, or XDSFolder)
URI	No enforcement

Table 3.14.4.1-9 SubmissionSet Metadata Attribute Enforcement

XDSSubmissionSet Attribute	Registry Enforcement
authorInstitution	No enforcement
authorPerson	No enforcement
authorSpecialty	No enforcement
comments	No enforcement
contentTypeCode	Coding Scheme and Code value
contentTypeCodeDisplayName	Must match contentTypeCode
patientId	Authority Domain Id Patient Id (known from patient identity feed)
sourceId	Coding Scheme and Code value
submissionTime	No enforcement
uniqueId	No identical existing uniqueId in registry (assigned to XDSDocumentEntry, XDSSubmissionSet, or XDSFolder)

Table 3.14.4.1-10 Folder Metadata Attribute Enforcement

XDSFolder Attribute	Registry Enforcement
codeList	Coding Scheme and Code value
codeListDisplayName	Must match codeList
comments	No enforcement

lastUpdateTime	No enforcement
patientId	The value of the patientId shall be the same for all new documents of a Folder.
uniqueId	No identical existing uniqueId in registry (assigned to XDSDocumentEntry, XDSSubmissionSet, or XDSFolder)

3.14.4.1.2.11 Protocol Requirements

SOAP with Attachments shall be used as the protocol between the Document Repository and the Document Registry when these two actors are implemented separately. The protocol is specified in ITI TF-2 : 3.15.4.1.2.3.1 (On-line protocol binding).

3.14.4.1.2.12 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 IDs (XDSDocumentEntry.patientId attribute) shall be a known patient ID and registered against the Patient ID Domain of the XDS Affinity Domain managed by the patient Identity Source Actor.

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 Affinity Domain constrains code values, to verify them (See Section 3.14.4.1.2.10).

Ensure submissions are atomic - The adaptor shall make submission to registry an atomic operation – see section 3.14.4.1.2.3.3 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:

- 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. An error shall be thrown if this object is not contained in the registry.
- 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.

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.

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

3.14.4.1.2.13 General Metadata Issues

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

3.14.4.1.2.13.1 Association Type naming

XDS requires that Association names be specified as text names and not UUIDs. This is consistent with version 2.0 and 2.1 of ebRIM. XDS requires the use of the following standard Associations:

HasMember – for linking RegistryPackage objects to their contents

ExternallyLinks – for binding an ExternalLink object to an ExtrinsicObject.

In addition, XDS defines a collection of Association types defined in section 3.14.4.1.2.6 Document Relationships and Associations.

3.14.4.1.2.13.2 Assigning Codes to Documents

Many attributes of XDSDocumentEntry, XDSSubmissionSet, and XDSFolder (Tables 3.14.4.1-3, 3.14.4.1-4, and 3.14.4.1-5) are coded attributes defined as ebRIM Classifications. Three details are required to describe a coded value:

2. The value of the code
4. The display name of the code (raw codes are not human-friendly)
5. 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"
   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 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

Name for each code – which must be used in the Name element and must match the value for the code.

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 Affinity Domain.

Note: the attribute XDSDocumentEntry.languageCode is not encoded as shown above. See Tables 3.14.4.1-3 for details.

3.14.4.1.2.14 Sequencing Requirements

The Repository actor shall:

3. Make a new document available for retrieval via the Retrieve Document transaction before it initiates the Register Document Metadata transaction with the Registry actor.

This is necessary because:

4. The Document Registry actor may choose to validate URIs contained in metadata before acknowledging the Register Document Metadata transaction.

6. The Document Consumer actor may retrieve the document before the Register Document Metadata Acknowledgement is received by the Repository actor.

3.14.4.1.2.15 Security Requirements

This profile requires all actors be grouped with a Secure Node Actor as defined in the IHE Audit Trail and Node Authentication Integration profile. This use of the ATNA profile in an XDS Affinity Domain does not require a centralized affinity domain Audit Repository Actor.

The use of ATNA along with XDS does require that each member of the Affinity Domain does have audit and security mechanisms in place. See appendix ITI TF-1: Appendix G and ITI-TF-2: Appendix K.

The individual actors involved are often members of different secure domains, as illustrated in Figure 3.14.4.1-2. The data transfers between different secure domains need different protection than transfers within a secure domain. They shall be either:

- Encrypted, with TLS authentication of both hosts, for online transfers, or
- Encrypted, using S/MIME secure encoding and digital signature by the sender, for offline transfers.

Transfers within a single secure domain may choose to omit encryption if it is unnecessary, so it is recommended that the online transfer security mechanisms be configurable. Certificate management and exchange is defined as part of the affinity domain business relationships and no IHE Integration Profile is specified at this time, see ITI TF-1: Appendix L.

Each transaction will result in audit records describing the transaction. Each secure domain has its own audit server to capture the records for the actors that are within that domain. Access to audit records by other enterprises within the affinity domain is managed and controlled by the business relationship terms of the affinity domain. There is no automatic IHE transaction for such access.

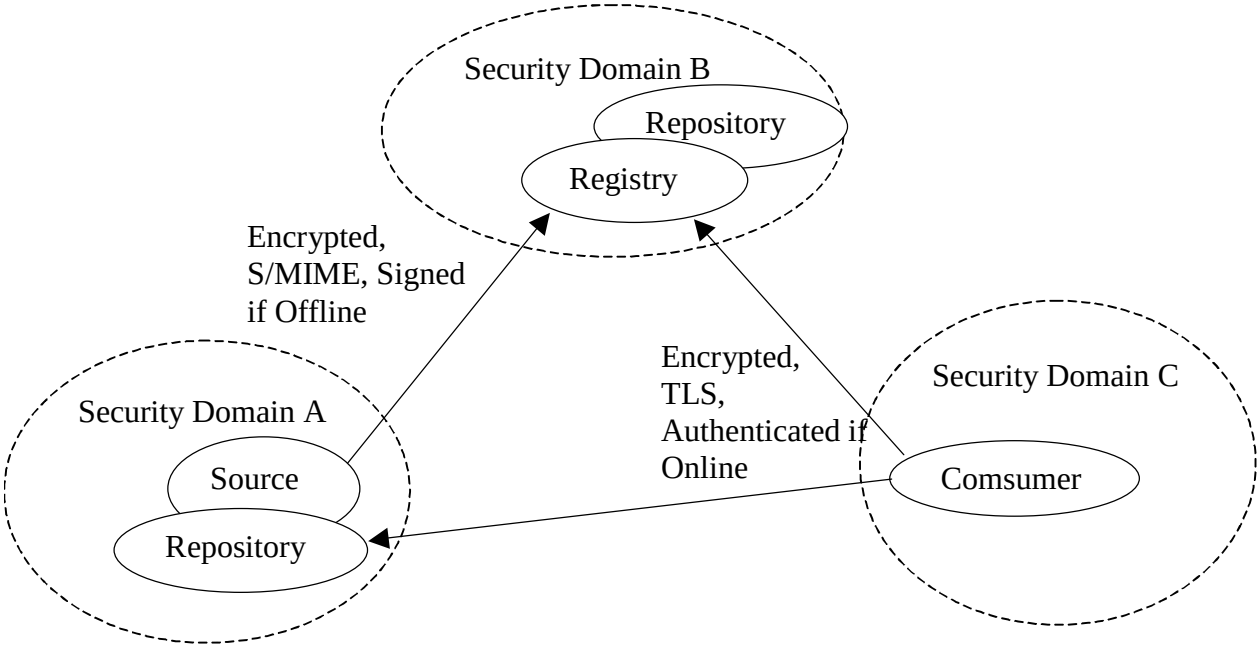
The audit records that should be generated (references IHE ATNA Integration Profile) by normal XDS activities are:

- For the Register Document Set, and the Provide and Register Document Set Transactions:
  - The Source Actor shall generate “Export” events describing the export of PHI from the Source to the Registry Actor. There should be one report for each transaction.
  - The Registry Actor shall generate “Import” events describing the import of PHI from the Source to the Registry Actor. There should be one report for each transaction.
- For the Query Documents Transaction:

- The Registry Actor shall generate a "Query" event describing the query, and shall generate an "Export" event if the query results in a reply that contains PHI.

- For the Retrieve Document Transaction:
  - The Repository Actor shall generate an "Export" event. This may be an event for each Retrieve Document Transaction, or multiple transactions for the same patient may be heuristically combined. The heuristics for this combination are not specified by IHE. It is intended to reduce the volume of audit records. Combination is permitted when the active participants and patient are the same, and the time difference is considered insignificant.
  - The Document Consumer Actor shall generate an "Import" event. This may be one event per transaction, or multiple transactions may be reported as a single event using a heuristic for combining transactions. Combination is permitted when the active participants and patient are the same, and the time difference is considered insignificant.

Figure 3.14.4.1-2 - Example Security Domain Relationships



All Actors are part of the same Clinical Affinity Domain

3.14.4.1.3 Expected Actions

Upon receipt of a Register Document Metadata message, the Document Registry with the aid of the Registry Adaptor shall do the following:

Accept all valid SubmitObjectsRequests.

Perform validations

Update the registry with the contained metadata

Return a RegistryResponse message given the status of the operation.

If the registry rejects the metadata, then, the following occurs:

An error is returned

The error status includes an error message

The request is rolled back

3.14.4.2 Register Document Metadata Acknowledgment

3.14.4.2.1 Trigger Events

The Document Registry finishes processing a Register Document Metadata request and shall respond with:

Register Document Metadata Acknowledgment

This message corresponds to the ebXML RequestResponse message.

3.14.4.2.2 Message Semantics

The ebXML RequestResponse message carries the status of the requested operation and an error message if the requested operation failed. The conditions of failure and possible error messages are given in the ebRS standard.

3.14.4.2.3 Expected Actions

The Document Repository now knows that the transaction succeeded/failed and can continue. The metadata added to the registry as a result of this transaction is now available for discovery via query transactions.

3.15 Provide and Register Document Set

This section corresponds to Transaction ITI-15 of the IHE Technical Framework. Provide and Register Document Set is used by the Document Source to provide a set of documents to the Document Repository, and to request that the repository store these documents and then register them with the Document Registry.

The Provide and Register Document Set transaction describes only the interaction between the Document Source and Document Repository actors. The interaction between the Document Repository and the Document Registry is described separately in the Register Document Set Transaction (ITI-14).

This transaction aligns with the Registry Services standard (ebRS). The ebRS standard covers the interaction with a service that includes a registry with integrated repository. From the point of view of the Document Source, the separate nature of the XDS Document Registry and Repository actors is hidden. This transaction exactly matches the registry service for submitting registry/repository content found in ebRS.

By specifying separate registry and repository actors, XDS offers additional flexibility of having a single registry index content for multiple repositories. The ebRIM portion of the registry standard supports this possibility though the ExternalLink object type.

The documents and metadata go to the repository actor and then the metadata is forwarded on to the registry actor. They move in this direction for several reasons:

- Allows best reuse of ebXML Registry specified protocols
- Document Source only needs to know the identity of the Document Repository. Repository knows the identity of the registry. If Provide and Register Document Set transaction were sent to the registry then routing decisions for documents would be more complex.
- Resulting protocols are simpler
- Simplifies the common case where the Document Source and the Document Repository are grouped.

3.15.1 Scope

The Provide Register Document Set transaction passes a Repository Submission Request (see ITI TF-2: 3.14.4.1.2.3.2) from a Document Source to a Document Registry.

A Provider and Register Document Set transaction carries:

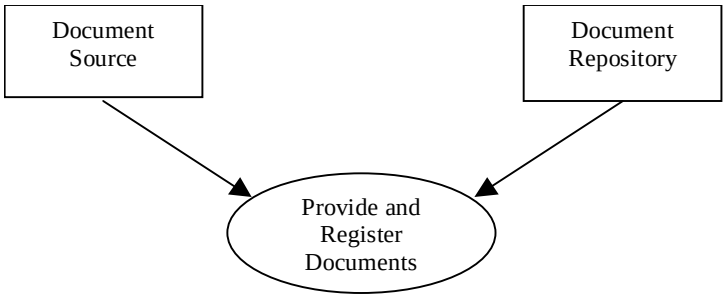
Metadata describing zero or more new documents

Submission Set definition along with the linkage to new documents and references to existing documents

Zero or more XDS Folder definitions along with linkage to new or existing documents

Zero or more documents

3.15.2 Use Case Roles



Actor: Document Source

Role: A system that submits documents and associated metadata to a Document Repository. Detail requirements for this actor are discussed in section 3.15.5.1.

Actor: Document Repository

Role: A document storage system that receives documents and associated metadata and:

Stores the documents

Enhances submitted metadata with repository information to enable later retrieval of documents

Forwards the enhanced metadata to the Document Registry.

3.15.3 Referenced Standards

ebMS OASIS/ebXML Messaging Services Specifications v2.0

ebRIM OASIS/ebXML Registry Information Model v2.0

ebRS OASIS/ebXML Registry Services Specifications v2.0

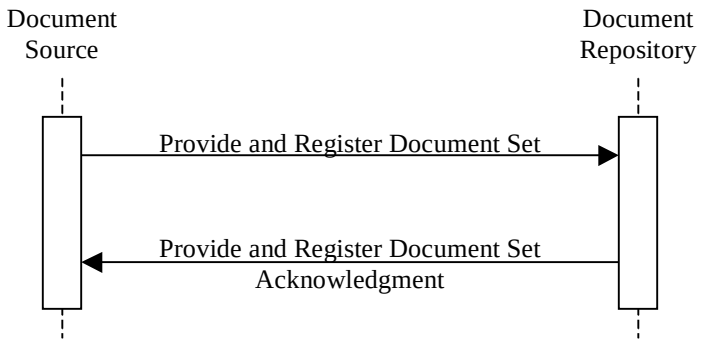
HTTP HyperText Transfer Protocol HTTP/1.1 (IETF RFC2616)

MIME Multipurpose Internet Message Extensions (RFC 2045 to RFC 2049)

SMTP Simple Mail Transfer Protocol (RFC2821)

multipart/related The MIME Multipart/Related Content-type (RFC2387)

3.15.4 Interaction Diagram



3.15.4.1 Provide and Register Document Set Message

A Document Source sends documents and associated metadata to a Document Repository that has an associated Document Registry. This message corresponds to an ebRS SubmitObjectsRequest with associated documents.

3.15.4.1.1 Trigger Events

The Document Source, based on a human decision or the application of a certain rule of automatic operation, wants to submit

- A set of one or more documents to the Document Repository and
- The associated metadata to the Document Registry.

3.15.4.1.2 Message Semantics

Message semantics are discussed as follows:

1. Metadata
2. Security Requirements
3. Protocol Selection (On-Line Protocol binding and Off-Line Protocol binding)

3.15.4.1.2.1 Metadata

The Register Document Set message shall include the metadata attributes (as defined in section 3.14.4.1.2.7) that will be forwarded by the Document Repository to the Document Registry using the Register Document Set Transaction (ITI-14).

The Document Source supplies all necessary registry object attributes with the exception of the URI attribute of an XDSDocumentEntry that must be assigned by the Document Repository.

Therefore, the Document Repository must add this attribute to the metadata before initiating the Register Document Set transaction to the registry.

3.15.4.1.2.2 Security Requirements

Relevant security requirements are discussed in the Register Document transaction (see ITI TF-1: 3.14.4.1.2.14).

3.15.4.1.2.3 Protocol Selection

There are two types of network relationships between the Document Source and Document Repository:

On-line – the Document Source constructs a direct connection (i.e, socket) to the Document Repository.

Off-line – the Document Source connects to the Document Repository via SMTP.

3.15.4.1.2.3.1 On-Line Protocol Binding

3.15.4.1.2.3.1.1 General structure and header

This is a MIME multipart/related message. The first attachment inside the payload of the SOAP request bears the registry metadata in an XML file containing the SubmitObjectsRequest.

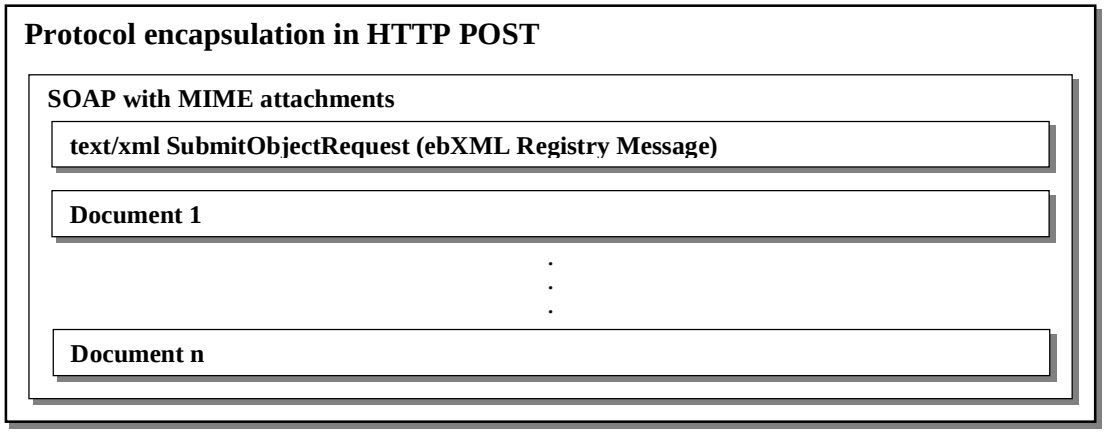


Figure 3.15.4.1-1 General Diagram of the Main message composing the On-Line Provide and Register Document Set Transaction

3.15.4.1.2.3.1.2 Associated Documents

The next attachments will contain the document(s) to be provided and registered, as MIME parts. There are one or more parts that contain byte streams representing documents[5].

The multipart packaging transmits the MIME-type of each part. The metadata part shall be of type text/xml. Parts containing documents destined for the Document Repository can have any MIME type, either single part or multipart. Each part containing a document has associated with it a document ID that is unique within the scope of this message. The Registry Metadata contained within one part of this message uses these document IDs to bind pieces of metadata to documents.

The registry metadata will be valid according to ebRIM and will contain the definition of one or more ebXML ExtrinsicObjects. An ExtrinsicObject is a registry object that represents a repository document within the registry. Each ExtrinsicObject will contain an **id** attribute. The format of this **id** follows the ebXML Registry definition. It is either a valid UUID or a symbolic name.

The value of this **id** attribute is used to link an ExtrinsicObject (XDSDocumentEntry) to a single part of the multipart that contains the attachments to the message. The header of the relevant part of the multipart will have a Content-Id header whose value is this **id** attribute surrounded by angle brackets as in the following example.

The metadata includes:

```
<ExtrinsicObject id="myDocument" ...
```

which links to the following MIME multipart part:

```
--------Boundary
Content-Type: text/xml
Content-Id: <myDocument>

This sentence is the value of the document.
--------Boundary
```

3.15.4.1.2.3.2 Off-Line Protocol Binding

3.15.4.1.2.3.2.1 General structure and header

As shown on Figure 3.15.4.1-2, the Off-Line transaction will be based on the ebXML Message Service Binding, as defined in the ebXML Registry Service (ebRS), with an Asynchronous Message and responses as defined in ebXML Messaging Services (ebMS). The re-use of ebXML enables implementers to integrate the Provide and Register Document Set transaction into a server which supports more comprehensive services, including some using Collaboration-Protocol Profiles (CPP) and Collaboration-Protocol Agreement (CPA) as supported by ebXML.

[5] This section is written independent of which protocol binding is used to package this multipart message. The protocol choice is documented elsewhere in this profile.

Because IHE is aiming to specify such as plug-and-play mechanisms, the Off-Line Protocol Binding is entirely defined into the present document. This specification does not mandate the use of a CPA between the Document Repository acting as "ebRS Registry" and the Document Source acting as "ebRS Registry Client". Such protocol agreement aspects are beyond the scope of the XDS Profile. The Document Source has only to know the Document Repository e-mail address to be able to provide and register a document set.

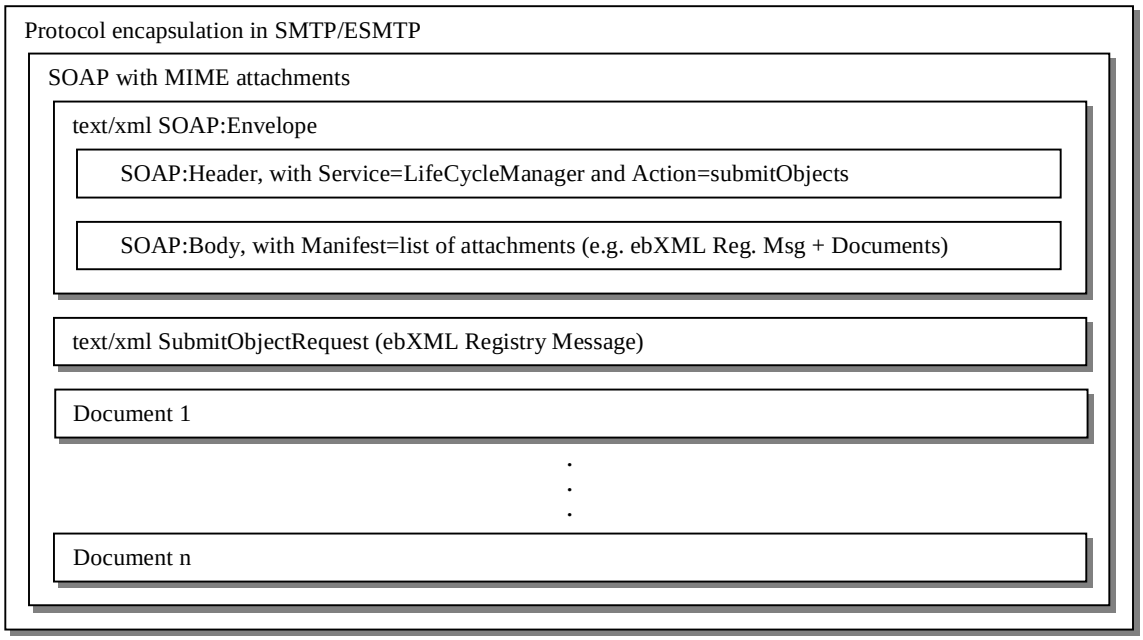


Figure 3.15.4.1-2 General Diagram of the Main message composing the Off-Line Provide and Register Document Set Transaction

The message is an e-mail message (which the ebXML Messaging Services can split into several messages if a single message would be too big) containing the following fields:

- The **From:** e-mail address of the sender (Document Source).
- Optionally, a **Reply-to:** address if the Document Source wants the response messages to be sent to another e-mail address.
- The **To:** e-mail address of the recipient (Document Repository). In case the Document Repository is able to register a document set to more than one Document Registry, it will have a different e-mail address for each one of the Repository-Registry peer.
- **Date:** is the date and time of the Provide and Register Document Set Transaction.
- **Subject: XDS/1.0/PnR/** (followed optionally by indication of XDS "subprofile" name. It SHALL NOT contain any Patient related information)
- **MIME-Version: 1.0**.
- **SOAPAction: "ebXML"**.

This is a MIME multipart/related message. The first attachment is the text/xml SOAP:Envelope part containing the ebMS header. The character set of the ebMS header is UTF-8.

The Header is described in the ebMS standard. It contains the following ebRS tag values:

- The header of the message, in /SOAP:Envelope/ SOAP:Header/eb:MessageHeader/ as shown in the table below.

Table 3.15.4.1-1 ebXML Message Header

Location ("@" for attributes)	Description
eb:From/eb:PartyId	Identification of the message sender (its email address, preceded by **mailto:**)
eb:From/eb:Role	String indicating the authorized role of the sender formatted as a URI per ebXML messaging specification: http://www.ihe.net/roles/iti/xds/DocumentSource
eb:To/eb:PartyId	Identification of intended recipient of the message (its email address, preceded by **mailto:**)
eb:To/eb:Role	String indicating the authorized role of the sender formatted as a URI per ebXML messaging specification: http://www.ihe.net/roles/iti/xds/DocumentRepository
eb:CPAId	Identification of a Collaboration Protocol Agreement between the sender and receiver. This shall contain the trading partner agreed CPA text reference, if it exists (e.g., the URI of the XML file describing the partnership agreement). If there is no CPA, this element shall be the concatenation of eb:From/eb:PartyId and the eb:To/eb:PartyId, separated by the hyphen character (-).
eb:ConversationId	In the absence of a local trading partner agreement, shall be CCYYMMDD-HHMMSS-mmmmm based upon the sending ebXML message generation. When generating responses the eb:ConversationID is taken from the original message.
eb:Service	Shall be LifeCycleManager
eb:Action	Shall be submitObjects
eb:MessageData/eb:MessageId	A unique message identifier generated by the sender: either a concatenation of message elements to create a globally unique identifier, or a single message element if that element is globally unique.

eb:MessageData/eb:Timestamp	UTC Time that the message header was created in XMLSchema dateTime format. Example: 2004-12-25T23:50:50
eb:DuplicateElimination	If present, duplicate messages should be eliminated.
eb:Description	Description of the Submission Set (equivalent to the XDSSubmissionSet.comments attribute).
eb:AckRequested	Optional in ebMS, required here to indicate that the repository shall acknowledge the message. This element has the following attributes: SOAP:mustUnderstand="1" eb:version="2.0" eb:signed="false"

2740

- List of references to document, in /SOAP:Envelope/ SOAP:Body/eb:Manifest/eb:Reference as shown in the table below.

Table 3.15.4.1-2 ebXML Message References

Location ("@" for attributes)	Description
@eb:id	Identification of the document, which is the OID of the XDSDocument. However, the first reference shall be to the SubmitObjectsRequest XML file, with id set to SubmitObjectsRequest.
@xlink:href	The relative URI of the document in the payload of the ebMS message, cid: followed by the OID. Used only for a newly submitted XDS Document.
@xlink:role	Shall be present only for the first reference, and be set to **http://www.ihe.net/roles/iti/xds/SubmitObjectsRequest**
eb:Schema	Shall be present only for the first reference, and has following attributes: eb:location=**http://www.ihe.net/schemas/iti/xds/SubmitObjectsRequest** eb:version=1.0
eb:Description	To be set to the XDSDocumentEntry.title. However, for the first reference, shall be set to the meaning of SubmitObjectsRequest in the local language (i.e. lang="en-US", "Provide and Register Document Set Metadata").

2745 The following attachment inside the payload of the SOAP request bears the registry metadata in an XML file containing the SubmitObjectsRequest.

3.15.4.1.2.3.2.2 Associated Documents

See the subsection "Associated Documents" in the On-Line Binding section (ITI TF-2: 3.15.4.1.2.3.1.1). Any document that has a reference xlink:href and contains a URI that is a content id (URI scheme "cid") shall be included in the payload.

3.15.4.1.3 Expected Actions

The Document Repository will receive this message. Each document within the message will be stored into the repository as an octet stream with an associated MIME type. A detected failure will result in an error result message being returned to the Document Source thus terminating this transaction.

The Document Repository will modify the received registry metadata adding:

- A URI identifier (xdsDocumentEntry.URI) must be created that can be used by a Document Consumer to reference the document.
- A hash value (xdsDocumentEntry.hash)
- A size (xdsDocumentEntry.size).

A Register Document Set transaction with this modified metadata will be issued to the XDS Document Registry.

The repository will ensure that any Document Retrieve Transaction received including the URI identifying the XDS Document, this document shall be provided to the Document Consumer unchanged from the octet stream that was submitted (full fidelity repository).

3.15.4.2 Provide and Register Document Set Acknowledgment

The Document Repository sends a Provide and Register Document Set Acknowledgment when the processing of a Provide and Register Document Set is complete. This message is identical to the RegistryResponse message specified in ebRS. It shall be conveyed in the same protocol as the request.

3.15.4.2.1 Trigger Events

The following events can trigger this message:

Documents stored to repository successfully and metadata stored to registry successfully (The registry part is carried out as part of a Register Document Set transaction)

Documents stored to repository successfully but an error occurred in storing the metadata to the registry

Documents were not successfully stored to the repository

3.15.4.2.2 Message Semantics

An ebRS RegistryResponse message is returned containing status and an error message if necessary.

Additional relevant semantics for both the repository and registry are described in the Register Document Set transaction.

3.15.4.2.3 Expected Actions

The Document Source now knows that the transaction succeeded/failed and continue. The metadata added to the registry as a result of this transaction is now available for discovery via query transactions. The document(s) added to the repository are now available for retrieval.

3.15.5 Actor Requirements

This section summarizes the capabilities of one or more actors relevant to this transaction. The details regarding how to perform these operations are documented elsewhere in this transaction or possibly in other transactions.

3.15.5.1 Document Source

An implementation of the Document Source Actor shall be capable of the following operations:

1. Submit a single document
2. Submit a document as a replacement for another document already in the registry/repository

An implementation of the Document Source Actor may support one or more of the following XDS Options:

1. **Multiple Documents Submission Option**. In this option the Document Source offers the ability to include multiple documents in a single Submission Request.
2. **Document Life Cycle Management** In this option the Document Source offers the ability to perform the following operation:
   - Submit a document as an addendum to another document already in the registry/repository
   - Submit a document as a transformation of another document already in the registry/repository

   Note: In order to support document replacement/addendum/transformation grouping with the Document Consumer may be necessary in order to Query the registry (e.g. for UUIDs of existing document entries)

3. **Folder Management Option.** In this option the Document Source offers the ability to perform the following operation:

   - Create a folder
   - Add one or more documents to a folder

   Note: In order to support document addition to an existing folder, grouping with the Document Consumer may be necessary in order to Query the registry (e.g. for UUIDs of existing folder).

These operations are discussed in section 3.14.4.1.2.3.4 Other Properties of Submission Requests.

3.15.5.2 Document Repository

A Document Repository may validate the following metadata elements received as part of a Provide and Register transaction:

XDSDocumentEntry.uniqueId – a submission may be rejected if not unique within the repository.

XDSSubmissionSet.sourceId – a repository may choose to accept submissions only from certain sources and use this field to perform the filtering.

3.16 Query Registry

This section corresponds to Transaction ITI-16 of the IHE Technical Framework. Transaction ITI-16 is used by the Document Consumer to query the Document Registry for information about documents indexed in the registry.

Note: This is a very general query mechanism that allows very broad use. Future extensions to XDS may introduce restrictions or specified the use of canned queries. Proposals for restricting the search mechanism are requested.

3.16.1 Scope

The Query Registry Transaction supports a variety of types of queries. Examples include the following:

Query by patient (Id) for a time interval, by document type(s), by practice setting(s), by author person

Query by Document Source

Query for XDS Folders updated during a time interval

Query for all documents in a Folder or Submission Set

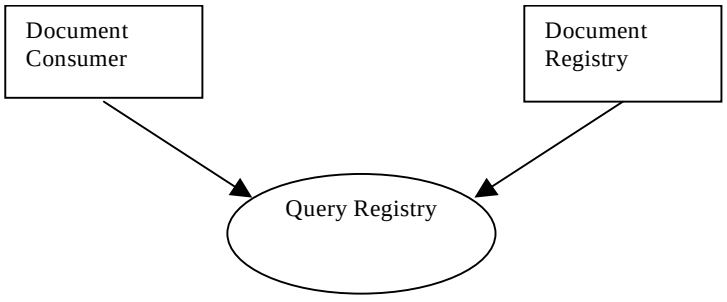
Query by time of submission

The list of XDS registry entries attributes that can be the target of a query are defined in Section 3.14.4. This transaction will document the basic syntax and semantics of XDS Document Registry queries.

All queries return:

- Metadata for one or more registry objects, or
- Object references for one or more registry objects (registry UUIDs).

3.16.2 Use Case Roles



Actor: Document Consumer

Role: Generates Query Registry messages and sends them to the Document Registry.

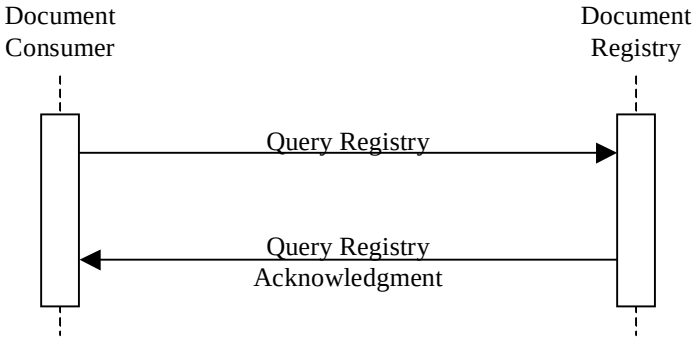
Actor: Document Registry

Role: Receives Query Registry messages and executes a query against registry metadata to select and return matching data to the Document Consumer.

3.16.3 Referenced Standard

ebRS	OASIS/ebXML Registry Services Specifications v2.0
SQL	ISO/IEC 9075 Database Language SQL

3.16.4 Interaction Diagram



3.16.4.1 Query Registry

This is the query request to the registry from a Document Consumer.

3.16.4.1.1 Trigger Events

This message is initiated when the Document consumer wants to retrieve document metadata.

3.16.4.1.2 Message Semantics

XDS specifies the use of SQL as a query language to the registry. There are 2 significant parameters to an AdHocQueryRequest (HTTP-SOAP):

- returnType
- SQL query text

3.16.4.1.2.1 Parameter returnType

XDS supports the following values for the parameter returnType:

- ObjectRef – a list of object UUIDs (references)
- LeafClass – list of XML elements representing the leaf class of the object returned

3.16.4.1.2.2 SQL query text

SQL queries submitted to an XDS Document Registry shall conform to the ebRS Registry Services specification, which maps elements of the information model (ebRIM) into a collection of SQL views.

The next sections show the details of several useful queries. This is not an exhaustive list. Any valid SQL query written against the registry information model (ebRIM+XDS specialization) may be used. The specific SQL subset used by registry is specified in Appendix D of ebRS.

3.16.4.1.2.3 Security Requirements

Relevant security requirements are discussed in the Register Document transaction (see ITI TF-1: 3.14.4.1.2.14).

3.16.4.1.3 Expected Actions

The registry returns a Query Registry Acknowledgment message.

3.16.4.1.4 Minimum Query Catalog

The queries documented in this section form a minimal set of queries needed by Document Consumers to discover documents in XDS.

It is the responsibility of the Document Consumer to package the SQL from any of these Minimum Queries listed below into a Query transaction.

All implementations of the Registry actor shall support all queries, including parts labeled optional, that are documented in this section. Document Consumer actors shall be able to depend on these queries to be supported by XDS Registry actors. XDS Registry actors may reject queries not in this query catalog. For example, XDS Registry actors may reject queries using the SQL keyword 'LIKE' except where noted in the following queries.

Each query is documented as a pseudo-function showing a simple name and a list of arguments. Many parameters end in '|*' indicating that the parameter is optional. Each parameter is numbered. The tables showing the SQL are labeled with these numbers so individual lines of SQL can be associated with a parameter of the pseudo-function defining the query.

Queries whose names start with 'Find' are broad, keyword-based searches focused on a single patient ID. Queries whose names start with 'Get' are simpler retrieval-style searches.

Query Parameters

Each query is represented as a function with parameters. The parameters are numbered and the *Parm* column in each query definition table indicates which parameter a particular row of the table supports. Additionally, each query parameter is supported by one or more detail parameters. For example, the query parameter *fromDateTime* is supported by detail parameters *$timeSlot, $lowerTime,* and *$uppertime* where *$timeSlot* indicates the name of the slot (there are 3) that is being tested and *$lowerTime* and *$upperTime* give the time range of interest. If a query parameter like *fromDateTime* is used then all of its detail parameters must be filled in. If this query parameter is not used, then all rows with a *Parm* showing that query's number are to be removed from the query.

All DateTime values are formatted as YYMMDDHHMMSS. All time comparisons are:

LowerDateTime <= DateTime < UpperDateTime

Some parameters are labeled as being in 'value list' format. A value list has the format:

('value1', 'value2')

The single quotes around the list items are required. The list format, parentheses and comma separation are required.

All values (constants) are set into single quotes, for example a dateTime value of '200412252359'.

When using the LIKE clause, the wildcard character is '%'.

3.16.4.1.4.1 FindDocuments

`FindDocuments((1)patientId, (2)classCode|*, (3)dateTimeRange|*, (4)practiceSettingCode|*, (5)healthcareFacilityTypeCode|*, (6)eventCodeList|*, (7)status)`

Find documents (XDSDocumentEntry) objects in the registry for a given patientID given a host of parameters to match.

Query Parameter	Detail Parameters
(1) patientID	$patientId – patientID including domain
(2) classCode	$classCodes – value list of classCodes
(3) dateTimeRange	$dateTimeAtt – slot name of dateTime attribute
	$dateTimeFrom – lower dateTime bound
	$dateTimeTo – upper dateTime bound

(4) practiceSettingCode	$psCodes – value list of practiceSettingCodes
(5) healthcareFacilityTypeCode	$hcftCodes – value list of healthcareFacilityTypeCodes
(6) eventCodeList	$evCodes – value list of eventCodes
(7) status	$status – value list – choose either ('Approved') OR ('Approved', 'Deprecated') OR ('Deprecated')

2930

- Returns a sequence of XDSDocumentEntry metadata for documents.
- Date/Time based on serviceStartTime or serviceStopTime or creationTime.
- Query can target documents that are Approved or Deprecated or both

Parm	SQL	Opt	Comments
	SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier patId		Returns ExtrinsicObjects
2	, Classification clCode	Yes	Required only if using classCode
3	, Slot dateTime	Yes	Required if selecting on a DateTime attribute
5	, Classification psc	Yes	Required if using XDSDocumentEntry.practiceSettingCode
6	, Classification hftc	Yes	Required if using XDSDocumentEntry.healthcareFacilityTypeCodes
7	, Classification ecl	Yes	Required if using XDSDocumentEntry.eventCodeList
	WHERE		
	doc.objectType = 'urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1'		Select XDSDocumentEntry objects only

1	AND (doc.id = patId.registryobject AND patId.identificationScheme='urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427' AND patId.value = $patientId)		Select on patientID
2	AND (clCode.classifiedobject = doc.id AND clCode.classificationScheme = 'urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a' AND clCode.nodeRepresentation IN $classCodes)	Yes	Select on classCode.
3	AND (dateTime.parent = doc.id AND dateTime.name = $dateTimeAtt AND dateTime.value >= $dateTimeFrom AND dateTime.value < $dateTimeTo)	Yes	Select on dateTimeRange
4	AND (psc.classifiedObject = doc.id AND psc.classificationScheme= 'urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead' AND psc.nodeRepresentation IN $psCodes)	Yes	Select on practiceSettingCodes
5	AND (hftc.classifiedObject = doc.id AND hftc.classificationScheme = 'urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1' AND hftc.nodeRepresentation IN $hcftCodes)	Yes	Select on healthcareFacilityTypeCodes
6	AND (ecl.classifiedObject = doc.id AND ecl.classificationScheme = 'urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4' AND ecl.nodeRepresentation IN	Yes	Select on eventCodeList

	$evCodes)		
7	AND doc.status IN $status		Select on document status

3.16.4.1.4.2 FindSubmissionSets

FindSubmissionSets((1)patientId, (2)sourceId|*, (3)dateTimeRange|*, (4)authorPerson|*, (5)contentTypeCode|*)

Find submission sets (XDSSubmissionSet objects) in the registry for a given patientID given a host of parameters to match.

Query Parameter	Detail Parameters
(1) patientID	$patientId – patientID including domain
(2) sourceId	$sourceIds – value list of sourceId
(3) dateTimeRange	$dateTimeAtt – slot name of dateTime attribute
	$dateTimeFrom – lower dateTime bound
	$dateTimeTo – upper dateTime bound
(4) authorPerson	$authorPattern – LIKE pattern for authorPerson
(5) contentTypeCode	$contentTypeCodes – value list of contentTypeCode

- This query returns XDSSubmissionSet metadata only. Use GetSubmissionSetContents () to drill down.

Parm	SQL	Opt	Comments
	SELECT ss.id FROM RegistryPackage ss, ExternalIdentifier patId, Classification c		Returns SubmissionSets
2	, ExternalIdentifier sid	Yes	sourceId
3	, Slot dateTime	Yes	dateTime range
4	, Slot ap	Yes	authorPerson
5	, Classification ctc	Yes	contentTypeCode
	WHERE		

	c.classifiedObject = ss.id AND c.classificationNode = 'urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd'		Select XDSSubmissionSet object only
	AND ss.status = 'Approved'		Select only Approved submission sets
1	AND (ss.id = patId.registryobject AND patId.identificationScheme= 'urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446' AND patId.value = $patientId)		Select on patientID
2	AND (sid.registryobject = ss.id AND sid.identificationScheme = 'urn:uuid:554ac39e-e3fe-47fe-b233-965d2a147832' AND sid.value IN $sourceIds)	Yes	Select on sourceId
3	AND (dateTime.parent = ss.id AND dateTime.name = $dateTimeAtt AND dateTime.value >= $dateTimeFrom AND dateTime.value < $dateTimeTo)	Yes	Select on dateTimeRange
4	AND (ap.parent = ss.id AND ap.name = 'authorPerson' AND ap.value LIKE $authorPattern)	Yes	Select on authorPerson
5	AND (ctc.classifiedObject = ss.id AND ctc.classificationScheme= 'urn:uuid:aa543740-bdda-424e-8c96-df4873be8500' AND ctc.nodeRepresentation IN $contentTypeCodes)	Yes	Select on contentTypeCode

3.16.4.1.4.3 FindFolders

FindFolders((1)patientId, (2)updatedSince|*, (3)codeList|*)

Find folder (XDSFolder object) in the registry for a given patientID given a host of parameters to match.

Query Parameter	Detail Parameters
(1) patientID	$patientId – patientID including domain
(2) updatedSince	$lastUpdateTime – dateTime value
(3) codeList	$codes – value list from codeList

- This query returns XDSFolder object only.

Parm	SQL	Opt	Comments
	SELECT fol.id FROM RegistryPackage fol, ExternalIdentifier patId, Classification c		Returns Folders
2	, Slot lut	Yes	lastUpdateTime
3	, Classification cl	Yes	codeList
	WHERE		
	(c.classifiedObject = fol.id AND c.classificationNode = 'urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2')		Select XDSFolder objects only
	AND fol.status = 'Approved'		Select only Approved folders
1	AND (patId.registryobject = fol.id AND patId.identificationScheme = 'urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a' AND patId.value = $patientId)		Select on patientID
2	AND (lut.parent = fol.id AND lut.name = 'lastUpdateTime'	Yes	Select on lastUpdateTime

	AND lut.value >= $lastUpdateTime)		
3	AND (cl.classifiedObject = fol.id AND cl.classificationScheme = 'urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5' AND cl.nodeRepresentation IN $codes)	Yes	Select on codeList

3.16.4.1.4.4 GetAll

GetAll((1)patientId, (2)status)

Get all XDSSubmissionSet, XDSDocumentEntry, and XDSFolder instances associated with the specified patient ID.

Query Parameter	Detail Parameters
(1) patientID	$patientId – patientID including domain
(2) status	$status – value list – choose either ('Approved') OR ('Approved', 'Deprecated') OR ('Deprecated')

- Query can target documents that are Approved or Deprecated or both
- Association objects are included

Parm	SQL	Opt	Comments
	SELECT roAndAss.id FROM RegistryObject ro, ExtrinsicObject doc, RegistryPackage		Returns RegistryObjects (submisson sets, folders, docs, associations)

	ss, RegistryPackage fol, Association ass		
	WHERE		
	doc.id IN (SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier patId WHERE doc.objectType = 'urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1' AND patId.registryObject = doc.id AND patId.identificationScheme = 'urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427' AND patId.value = $patientId)		Select documents on patientId
	AND ss.id IN (SELECT ss.id FROM RegistryPackage ss, Classification ssCl, ExternalIdentifier patId WHERE ssCL.classifiedObject = ss.id AND ssCL.classificationScheme = 'urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd' AND patId.registryObject = ss.id AND patId.identificationScheme = 'urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446' AND patId.value = $patientId)		Select submission sets on patientId
	AND fol.id IN (SELECT fol.id FROM RegistryPackage fol, Classification folCl, ExternalIdentifier patId WHERE folCL.classifiedObject = fol.id AND folCL.classificationScheme = 'urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2' AND		Select folders on patientId

	patId.registryObject = fol.id AND patId.identificationScheme = 'urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a' AND patId.value = $patientId)		
	AND ro.id IN (ss.id, fol.id, doc.id)		Collect submission sets, folders, and documents
	AND ro.status IN $status		Status is appropriate
	AND ass.id IN (SELECT ass.id FROM Association ass WHERE ass.sourceObject IN (ro.id) AND ass.targetObject IN (ro.id))		Collect associations linking the above documents, submission sets, and folders
	AND roAndAss in (ro.id, ass.id)		Combine submission sets, folders, documents, and assocations

3.16.4.1.4.5 GetDocument

GetDocument((1)XDSDocument.UUID|*, (2)XDSDocument.uniqueId|*)

This query returns the specified document given either the entry UUID or uniqueId of the document.

Query Parameter	Detail Parameters
(1) UUID	$uuid – UUID of document
(2) uniqueId	$uniqueId – uniqueId of document

- Parameter 1 OR 2 must be specified.

Parm	SQL	Opt	Comments
	SELECT doc.id FROM ExtrinsicObject doc		

	WHERE		
	doc.id IN (		
1	$uuid	Yes	Select by UUID
2	SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier uniqId WHERE uniId.registryobject = doc.id AND uniId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND uniId.value=$uniqueId	Yes	Select by uniqueId
	)		

3.16.4.1.4.6 GetSubmissionSetContents

GetSubmissionSetContents((1)XDSSubmissionSet.UUID, (2)XDSSubmissionSet.uniqueId)

This query returns the specified submission set, all contained documents and folders, and the associations that relate these elements.

Query Parameter	Detail Parameters
(1) UUID	$uuid – UUID of submission set
(2) uniqueId	$uniqueId – uniqueId of submission set

- Parameter 1 OR 2 must be specified.

Parm	SQL	Opt	Comments
	SELECT roAndAss.id FROM RegistryObject ro, ExtrinsicObject doc, RegistryPackage ss, RegistryPackage fol, Association ass		Returns RegistryObjects (submisson sets, folders, docs, associations)
	WHERE		
	ss.id IN (		Select submission set

1	$uuid	Yes	
2	SELECT ss.id FROM RegistryPackage ss, ExternalIdentifier uniqId WHERE uniqId.registryObject = ss.id AND uniqId.identificationScheme = 'urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8' AND uniqId.value = $uniqueId	Yes	
	) AND doc.id IN (SELECT doc.id FROM ExtrinsicObject doc, Association a WHERE a.associationType = 'HasMember' AND a.sourceObject = ss.id AND a.targetObject = doc.id AND doc.objectType = 'urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1')		Select documents
	AND fol.id IN (SELECT fol.id FROM RegistryPackage fol, Classification folCl, Association a WHERE a.associationType = 'HasMember' AND a.sourceObject = ss.id AND a.targetObject = fol.id AND folCL.classifiedObject = fol.id AND folCL.classificationScheme = 'urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2') AND		Select folders
	ro.id IN (ss.id, fol.id, doc.id) AND		Collect submission sets, folders, and documents
	ass.id IN (SELECT ass.id FROM Association ass		Collect associations linking the above documents, submission sets, and folders

	WHERE ass.sourceObject IN (ro.id) AND ass.targetObject IN (ro.id)) AND		
	roAndAss in (ro.id, ass.id)		Combine submission sets, folders, documents, and assocations

3.16.4.1.4.7 GetFolderContents

GetFolderContents((1)XDSFolder.UUID, (2)XDSFolder.uniqueId)

This query returns an XDSFolder, the XDSSubmissionSet it is associated with and the collection of documents currently associated with the folder.

Query Parameter	Detail Parameters
(1) UUID	$uuid – UUID of folder
(2) uniqueId	$uniqueId – uniqueId of folder

- This query only retrieves metadata for Approved documents.

Parm	SQL	Opt	Comments
	SELECT ro.id FROM RegistryObject ro, RegistryPackage fol, RegistryPackage ss, ExtrinsicObject docEnt WHERE		
1	fol.id = $uuid	Yes	Use only if selecting by UUID
2	fol.id IN (SELECT fol.id FROM RegistryPackage fol, ExternalIdentifier uniId WHERE	Yes	Use only if selecting by uniqueId

	AND fol.id = uniId.registryObject AND uniId.identificationScheme = 'urn:uuid:75df8f67-9973-4fbe-a900-df66cefecc5a' AND uniId.value = $uniqueId)		
	AND ss.id IN (SELECT ss.id FROM RegistryPackage ss, Association ass, ExternalIdentifier uniqId WHERE ss.id = ass.sourceObject AND ass.associationtype='HasMember' AND ass.targetObject = fol.id AND uniqId.registryObject = ss.id AND uniqId.identificationScheme= 'urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8')		Select XDSSubmission Set
	AND docEnt.id IN (SELECT docEnt.id FROM ExtrinsicObject docEnt, Association ass, ExternalIdentifier uniqId WHERE fol.id = ass.sourceObject AND ass.associationtype='HasMember' AND ass.targetObject = docEnt.id AND uniqId.registryObject = docEnt.id AND uniqId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab')		Select XDSDocuments in XDSFolder
	AND ro.id IN (fol.id, ss.id, docEnt.id)		Gather Folder, Submission Set, and Documents
	AND ro.status = 'Approved'		Approved object only

3.16.4.1.4.8 GetFoldersForDocument

GetFoldersForDocument((1)XDSDocument.UUID|*, (2)XDSDocument.uniqueId|*)

3000 Return a list of XDSFolder metadata for each folder associated with a specified document.

Query Parameter	Detail Parameters
(1) UUID	$uuid – UUID of document
(2) uniqueId	$uniqueId – uniqueId of document

- This query can identify the document by UUID or uniqueId.

Parm	SQL	Opt	Comments
	SELECT fol.id FROM RegistryPackage fol, Association ass, Classification class, ExtrinsicObject doc		
	WHERE		
1	doc.id = $uuid	Yes	Use only if selecting by UUID.
2	doc.id IN (SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier uniId WHERE uniId.registryobject = doc.id AND uniId.identificationscheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND uniId.value = $uniqueId)	Yes	Use only if selecting by uniqueId.
	AND (ass.associationtype = 'HasMember' AND ass.sourceobject = fol.id AND ass.targetObject = doc.id)		Linkage from document to RegistryPackage.
	AND (class.classifiedObject = fol.id AND class.classificationnode = 'urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2')		Verify RegistryPackage is XDSFolder.
	AND fol.status = 'Approved'		

3005

3.16.4.1.4.9 GetAddendums

GetAddendums((1)documentUUID, (2)documentUniqueId)

Query Parameter	Detail Parameters
(1) UUID	$uuid – UUID of document
(2) uniqueId	$uniqueId – uniqueId of document

Given a document UUID or uniqueId, return all addendums for this document.

Parm	SQL	Opt	Comments
	SELECT add.id FROM ExtrinsicObject add, ExtrinsicObject doc, Association a		
	WHERE		
1	doc.id = $uuid	Yes	Select document by UUID
2	doc.id IN (SELECT ExtrinsicObject doc, ExternalIdentifier uniqId WHERE uniId.registryobject = doc.id AND uniId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND uniId.value=$uniqueId)	Yes	Select document by uniqueId
	AND (a.associationType = 'APND' AND a.sourceObject = add.id AND a.targetObject = doc.id)		

3.16.4.1.4.10 GetHistory

GetHistory((1)documentUUID, (2)documentUniqueId)

Given a document UUID or uniqueId, return the document linked to this document by a RPLC association if it exists.

Query Parameter	Detail Parameters
(1) UUID	$uuid – UUID of document
(2) uniqueId	$uniqueId – uniqueId of document

The document identified in this query need not be an Approved document.

Parm	SQL	Opt	Comments
	SELECT rplc.id FROM ExtrinsicObject rplc, ExtrinsicObject doc, Association a		
	WHERE		
1	doc.id = $uuid	Yes	Select document by UUID
2	doc.id IN (SELECT ExtrinsicObject doc, ExternalIdentifier uniqId WHERE uniId.registryobject = doc.id AND uniId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND uniId.value=$uniqueId)	Yes	Select document by uniqueId
	AND (a.associationType = 'RPLC' AND a.targetObject = rplc.id AND a.sourceObject = doc.id)		

3.16.4.1.4.11 GetTransformations

GetTransformations((1)documentUUID, (2)documentUniqueId)

Given a document UUID or uniqueId, return all transformations for the document.

Query Parameter	Detail Parameters
(1) UUID	$uuid – UUID of document
(2) uniqueId	$uniqueId – uniqueId of document

Parm	SQL	Opt	Comments
	SELECT xfrm.id FROM ExtrinsicObject xfrm, ExtrinsicObject doc, Association a		
	WHERE		
1	doc.id = $uuid	Yes	Select by UUID
2	doc.id IN (SELECT ExtrinsicObject doc, ExternalIdentifier uniqId WHERE uniId.registryobject = doc.id AND uniId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND uniId.value=$uniqueId)	Yes	Select by uniqueId
	AND (a.associationType = 'XFRM' AND a.sourceObject = xfrm.id AND a.targetObject = doc.id)		

3.16.4.2 Query Registry Acknowledgement

This is the response to the Query Registry message.

3.16.4.2.1 Trigger Events

Completion of query initiated by a Query Registry message.

3.16.4.2.2 Message Semantics

The Query Registry Acknowledgement (AdhocQueryResponse) is returned in one of three forms:

1. List of ObjectRefs
2. Registry metadata describing objects found by query
3. Error message

3.16.4.2.3 Expected Actions

The Document Consumer may process the returned registry data, retrieve documents based on the metadata if the necessary metadata was returned, or handle returned errors

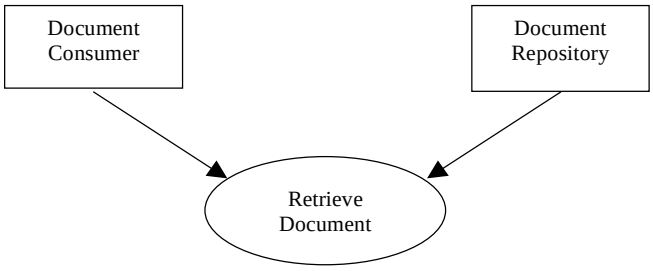
3.17 Retrieve Document

This section corresponds to Transaction ITI-17 of the IHE Technical Framework. The Document Consumer and Document Repository actors use transaction ITI-17.

3.17.1 Scope

This transaction is used by the Document Consumer to retrieve a document from the Document Repository. The Document Consumer has already obtained the URI information from the Document Registry by means of the Query Registry transaction.

3.17.2 Use Case Roles



Actor: Document Consumer

Role: Obtains document.

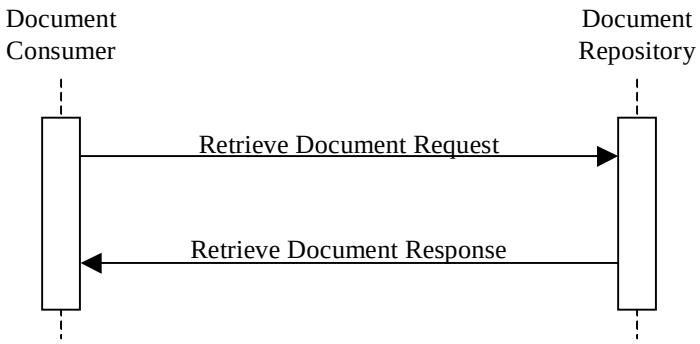
Actor: Document Repository

Role: Provides documents.

3.17.3 Referenced Standard

HTTP	Hyper Text Transfer Protocol HTTP 1.1 (RFC 2616)
MIME	Multipurpose Internet Message Extensions (RFC 2045 to RFC 2049)
SMTP	Simple Mail Transfer Protocol (RFC 2821)
Multipart/Related	The MIME Multipart/Related Content-type (RFC 2387)

3.17.4 Interaction Diagram



3.17.4.1 Retrieve Document Request

3.17.4.1.1 Trigger Events

The Document Consumer obtains document URIs via the

Query Registry transaction.

3.17.4.1.2 Message Semantics

The URI specifies the protocol and protocol parameters that are to be used to retrieve the document. The Document Repository shall support the following parameters for protocol in the URI:

- HTTP

The details of URI handling are specified in the HTTP standard (RFC 2616).

The Document Repository shall fully implement support for any protocol parameters that are required by the HTTP standard.

3.17.4.1.2.1 Request Headers

The HTTP Protocol specifies a variety of request headers that can affect the result returned by the server. Document Consumers may use any request header allowed by the HTTP Protocol[6]. However, XDS Repositories are not required to acknowledge or support of these headers not required by the protocol, and may be required in certain cases to ignore certain headers. See the table below for details.

Request Header	Repository	Comments

[6] Ed Note: To allow common web browsers to be used without restriction.

	Support	
Accept Accept-Charset Accept-Language	Always Ignored	These headers, if used by the Repository could in fact alter the content returned from the repository, and so must be ignored by the repository. [inconsistent with RID].
Accept-Encoding	O	This header requests that an encoded form the data be returned [e.g., gzip or compress]. Repositories may support this header, but are not required to. Document Consumers must support responses that ignore this content header.
Authorization	O	This header may be sent in environments where EUA is used with XDS. See the EUA profile for more details.
If-Modified-Since	O	Since Repositories are not expected to change documents once stored, they are free to ignore this header or respond as appropriate.

3.17.4.1.2.2 Security Requirements

Relevant security requirements are discussed in the Register Document transaction (see ITI TF-1: 3.14.4.1.2.14).

3.17.4.1.3 Expected Actions

A Retrieve Document Response will be generated in return. Details are specified in the HTTP standard.

3.17.4.2 Retrieve Document Response

3.17.4.2.1 Trigger Events

This message is triggered by the:

Retrieve Document Request.

3.17.4.2.2 Message Semantics

XDS Repositories are required to return the following values:

Response Code	When to Return	Support
200 – OK	If the request is valid and data is available.	R
304 – Not Modified	If the request is a valid conditional GET [see HTTP	O

	specification], and the document has not been modified since the requested modification date.	
400 – Bad Request	If the request is not valid.	R
401 – Authorization Required	If the request requires authentication, and an Authorization header is not present, or is not valid. Used in conjunction with EUA.	O
403 – Forbidden	If access needs to be denied for reasons other than authentication failure [e.g., because the request comes from a Node that is not allowed access to the document].	R
404 – Not Found	If the request is syntactically valid, but the document cannot be located, or does not otherwise exist [see RID].	R
410 – Gone	If the request is valid, and the document once existed, but is no longer available [e.g., the document may have been removed at the patients request].	O
5XX – Server Error	The server may return any error code beginning with the digit 5 to indicate a server error.	O

3.17.4.2.2.1 Response Headers

The HTTP Protocol specifies a variety of response headers that provide more information about the response. The use of these headers is described in the table below:

3095

Response Header	Repository Support	Comments
Expires	R	Any valid value according to RFC2616, or 0 [c.f. RID volume]
Content-Encoding	O	If the Document consumer requested encoding of the response, and the repository is able to fulfill that request, it must return the appropriate value in this header.
Content-Type	R	These headers correspond to the mimeType, languageCode, and size attributes of the

Content-Language Content-Length	O	XDSDocumentEntry. Content-Type is required in the response[7]. The other two are optional, but if present, must be the same as the values provided to the registry.
Last-Modified	R	This header should correspond to the date the document was first stored in the repository [if known], or the date of document creation [XDSDocumentEntry.creationTime].
WWW-Authenticate	O	If the XDS Repository requires authentication and the request did not contain valid credentials, this header must be returned in the 401 response.

3.17.4.2.2.2 Security Requirements

Relevant security requirements are discussed in the Register Document transaction (see ITI TF-1: 3.14.4.1.2.14).

3.17.4.2.3 Expected Actions

The Document Consumer now has the content of the document to process.

[7] This is to allow browser-based document consumers to activate the appropriate viewer based on the type of data present, without requiring that information to be known in advance before the request is made.

3.18 Intentionally Left Blank (ITI-18)

3.19 Authenticate Node

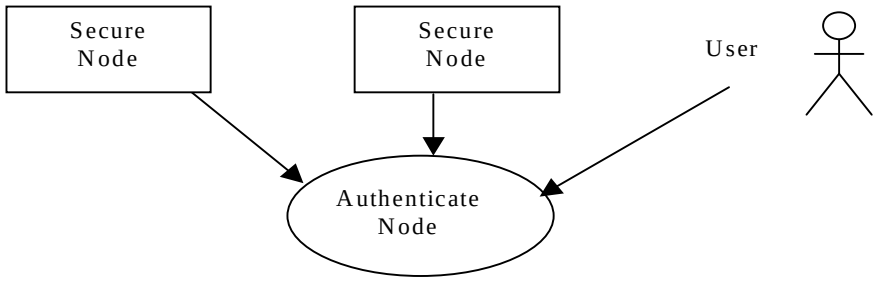
This section corresponds to Transaction 19 of the IHE ITI Technical Framework. Transaction 19 is used by the Secure Node actors

3.19.1 Scope

In the Authenticate Node transaction, the local Secure Node presents its identity to a remote Secure Node, and authenticates the identity of the remote node. After this mutual authentication other secure transactions may take place through this secure pipe between the two nodes.

In addition, the Secure Node authenticates the identity of the user who requests access to the node. This user authentication is a local operation that does not involve communication with a remote node.

3.19.2 Use Case Roles



Actor: Secure Node

Role: Establish a protocol specific trust relationship between two nodes in a network. Establishes the identity of a user, and authorizes access to the patient data and applications at the node.

Actor: User

Role: Someone who wants to have access to the data and applications available at the node.

3.19.3 Referenced Standards

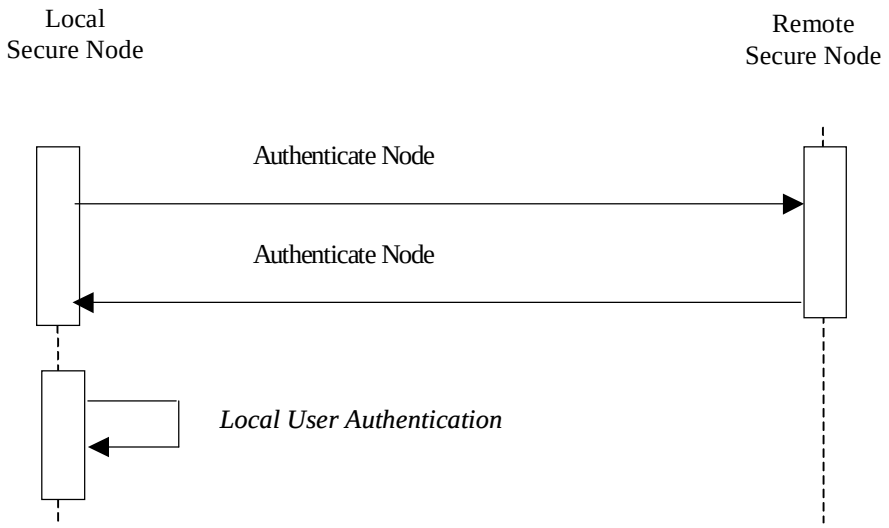
DICOM 2003 PS 3.15:
Security Profiles. Annex B1: The Basic TLS Secure Transport Connection profile.

IETF: Transport Layer Security (TLS) 1.0 (RFC 2246)

ITU-T: Recommendation X.509 (03/00). “Information technology - Open Systems Interconnection - The directory: Public-key and attribute certificate frameworks"

3.19.4 Interaction Diagram

Note: This diagram does not imply sequencing of Authentication Node and Local User Authentication.



3.19.5 Trigger Events

The Local Secure Node starts the authentication process with the Remote Secure Node when information exchange between the two nodes is requested. The first transaction shall be the Authenticate Node transaction, and all other PHI transactions performed by IHE actors shall be secure transactions. This authentication process is needed when a secure connection is established.

The Basic Secure Node shall always apply the Authenticate Node process to every DICOM, HTTP, or HL7 connection.

3.19.6 Message Semantics

3.19.6.1 DICOM and HL7 Connections

The Local Secure Node uses the TLS protocol to establish a trust relationship with the Remote Secure Node. The configuration settings for the TLS protocol depends upon the messaging standard that are used for the connection. HL7 and DICOM transactions are required to adhere

to the specifications in this section. All Secure Nodes shall be configurable for use on a physically secured network or not on a physically secured network.

When configured for use on a physically secured network, the normal DICOM and HL7 connection mechanisms shall be used.

When configured for use not on a physically secured network implementations shall use the TLS protocol, and the following cyphersuite shall be supported:

TLS_RSA_WITH_NULL_SHA

If the ATNA Encryption Option is implemented, the following cyphersuite shall also be supported:

TLS_RSA_WITH_AES_128_CBC_SHA.

The recommended "well-known port 2762" as specified by DICOM shall be used when the Secure node is configured for use not on a physically secured network. When the secure node is configured for use on a physically secured network, a different port number shall be used, preferably the standard port 104. HL7 does not specify port numbers, but the port number used when configured for use on a physically secured network shall be different than the port number used when configured for use not on a physically secured network.

The Authenticate node transaction involves the exchange of certificates representing the identities of the trusted nodes. Certificates shall use the X.509 standard.

The healthcare enterprise should define the maximum expiration time for certificates in its security policy. The IHE Technical Framework recommends a maximum expiration time of 2 years.

The Secure node shall provide means for installing of the required certificates, for example, via removable media or network interchange. Implementation must support the configuration of a list of authorized nodes (see ITI TF-2: Appendix A). Connections shall only be accepted from the authorized nodes on the list and connections only attempted to nodes on the list.

If Secure Node is configured for physical security, then it shall use the non-TLS DICOM port and protocol.

3.19.6.2 HTTP Connections

3.19.6.2.1 Expected Actions

A trusted association will be established between the two nodes. This association will be used for all further secure transactions between the IHE actors in two nodes.

The HTTP connection shall be made using a TLS connection in the same manner as HL7 and DICOM TLS connections described above, although the port number shall be configurable.

Note: Most web browsers do not support the mutual authentication of both nodes by means of TLS. They usually only support authentication of the server node by means of TLS, leaving the client node un-authenticated.

SSL connections are similar in only authenticating the server node. This is not suitable for use in the transfer of PHI. Secured HTTP communications will require the use of either browser extensions like applets, modified browsers, or application software that uses mutually authenticating TLS connections.

If Secure Node is configured for physical security, then it shall use the normal HTTP protocol.

3.19.7 Local User Authentication

The Secure Node starts the authentication process with a User when the User wants to log on to the node. The secure node shall not allow access to PHI to an operator who has not successfully completed the local user authentication. Local user authentication is not an IHE specified network transaction, although it may utilize a network system for user authentication.

This is a local invocation of functions at the Secure Node. The identity of the User will be established by the Secure Node actor based on methods such as:

- Username with Password
- Biometrics
- Smart card
- Magnetic Card

The User shall log in using his or her own unique individually assigned identity. Identities must be unique across the secure domain. A user may have more than one identity. The Secure Node shall be configurable to maintain a list of authorized users for the Secure Node.

The rules for assignment of unique individual identities to users is part of the Security Policy of the healthcare enterprise. Development of these rules is outside the scope of the IHE Technical Framework. The following examples list a few special cases related to user identification that may occur in practice.

3.19.7.1 Example: Team approach

When the operator is part of a team performing a procedure, the other members of the team involved in creating and accessing the data should be manually identified and recorded in the procedure log (which may be paper or electronic), and it is assumed that all have accessed the data even though they were not (and cannot be in most cases) actually logged on to the piece of equipment.

During some procedures, it may be necessary for one operator to relieve the operator who has already been authenticated by the system. It is recommended that the first operator log off and that the system authenticate the new operator.

3.19.7.2 Example: Access to locked exam room, no user logon on modality.

There may be situations where the acquisition modality has no user logon features, and access to the equipment is controlled by controlling access to the examination room. In these situations an

equipment-specific user ID will be used, and access to the room should be recorded in the procedure log (which may be paper or electronic).

3.19.7.3 Example: Enterprise User Authentication

The healthcare enterprise may implement local user authentication using the Enterprise User Authentication Profile (EUA). This implementation may be mixed with other non-EUA access to the secure domain, based upon each node's internal use an EUA availability.

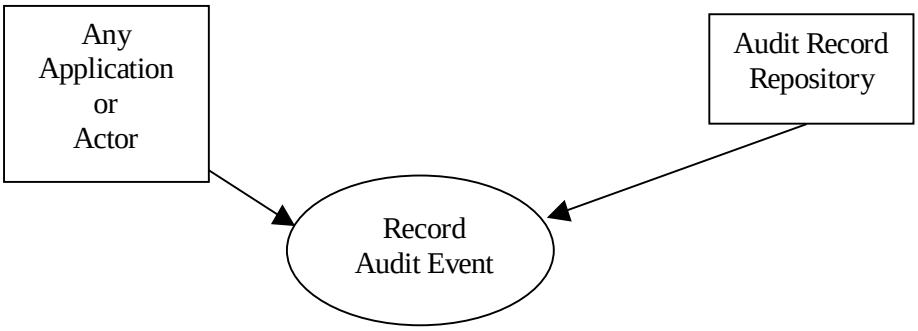
3.20 Record Audit Event

This section corresponds to Transaction 20 of the IHE ITI Technical Framework. Transaction 20 is used by the all IHE actors that support the Audit Trail and Node Authentication Integration Profile to communicate with the Audit Record Repository actors.

3.20.1 Scope

In the Record Audit Event transaction, the IHE actor creates an entry in the Audit Log at the Audit Record Repository.

3.20.2 Use Case Roles



Application or Actor: Any actor or any other application that is grouped with the Secure Node Actor.

Role: Create an audit record and transmit this record to the Audit Record Repository.

Actor: Audit Record Repository

Role: Receive an audit record from the Audit Record Creator and store this for audit purposes.

3.20.3 Referenced Standards

IETF: The BSD Syslog Protocol. (RFC 3164);

Reliable Delivery for Syslog (RFC 3195);

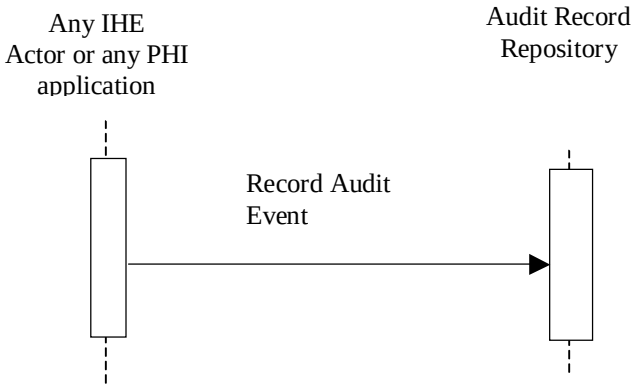
Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications (RFC 3881).

DICOM: Supplement 95

ASTM: E2147-01 Standard Specification for Audit and Disclosure Logs for Use in Health Information Systems.

W3C: Recommendation: Extensible Markup Language (XML) 1.0

3.20.4 Interaction Diagram



3.20.5 Record Audit Event

The Audit Record Repository shall accept the Audit Record message. The usage of the result by the Audit Record Repository is beyond the scope of the IHE Technical Framework.

3.20.6 Trigger Events and Message semantics

An Audit Log is a record of actions performed on data by users. Actions are queries, views, additions, deletions and changes. The IHE actor creates an Audit Record when an IHE transaction-related event occurs or when a non-transaction event occurs.

IHE specifies that events defined in Table 3.20.6-1 shall be reportable by means of the IHE Audit Trail. Radiology devices may also find that their subset of events is reportable by means of the IHE Provisional Audit Message Format. This is not recommended other than as a strategy for managing the upgrade of products and systems to the DICOM Audit Message Standard with IHE Extensions.

Table 3.20.6-1. Audit Record trigger events

Trigger Event	Description	Source Vocabulary
Actor-start-stop	Startup and shutdown of any actor. Applies to all actors. Is distinct from hardware powerup and shutdown.	DICOM (Sup 95) "Application Activity"
Audit-Log-Used	The audit trail repository has been accessed or modified by something other than the arrival of audit trail messages.	DICOM (Sup 95) "Audit Log Used"
Begin-storing-instances	Begin storing SOP Instances for a study. This may be a mix of instances.	DICOM (Sup 95) "Begin Transferring DICOM

Trigger Event	Description	Source Vocabulary
		Instances"
Health-service-event	Health services scheduled and performed within an instance or episode of care. This includes scheduling, initiation, updates or amendments, performing or completing the act, and cancellation. See note below.	IHE Extension (section 3.20.7.3) "Health Services Provision Event"
Instances-deleted	SOP Instances are deleted from a specific study. One event covers all instances deleted for the particular study.	DICOM (Sup 95) "DICOM Instances Accessed" or "DICOM Study Deleted"
Instances-Stored	Instances for a particular study have been stored on this system. One event covers all instances stored for the particular study.	DICOM (Sup 95) "DICOM Instances Transferred"
Medication	Medication orders and administration within an instance or episode of care. This includes initial order,dispensing, delivery, and cancellation. See note below.	IHE Extension (section 3.20.7.3) "Medication Event"
Mobile-machine-event	Mobile machine joins or leaves secure domain.	DICOM (Sup 95) "Network Entry"
Node-Authentication-failure	A secure node authentication failure has occurred during TLS negotiation, e.g. invalid certificate.	DICOM (Sup 95) "Security Alert"
Order-record-event	Order record created, accessed, modified or deleted. Involved actors: Order Placer. This includes initial order, updates or amendments, delivery, completion, and cancellation. See note below.	DICOM (Sup 95) "Order Record"
Patient-care-assignment	Staffing or participant assignment actions relevant to the assignment of healthcare professionals, caregivers attending physician, residents, medical students, consultants, etc. to a patient It also includes change in assigned role or authorization, e.g., relative to healthcare status change, and de-assignment	IHE Extension (section 3.20.7.3) "Patient Care Resource Assignment"
Patient-care-episode	Specific patient care episodes or problems that occur within an instance of care. This includes initial assignment, updates or amendments, resolution, completion, and cancellation. See note below.	IHE Extension (section 3.20.7.3) "Patient Care Episode"
Patient-care-protocol	Patient association with a care protocol. This includes initial assignment, scheduling, updates or amendments, completion, and cancellation. See note below.	IHE Extension (section 3.209.7.3) "Patient Care Protocol"
Patient-record-event	Patient record created, modified, or accessed.	DICOM (Sup 95) "Patient Record"
PHI-export	Any export of PHI on media, either removable physical media such as CD-ROM or electronic transfer of files such as email. Any printing activity, paper or film, local or remote, that prints PHI.	DICOM (Sup 95) "Export"
PHI-import	Any import of PHI on media, either removable physical media such as CD-ROM or electronic transfers of files such as email.	DICOM (Sup 95) "Import"
Procedure-record-event	Procedure record created, modified, accessed or deleted.	DICOM (Sup 95)

Trigger Event	Description	Source Vocabulary
		"Procedure Record"
Query Information	A query has been received, either as part of an IHE transaction, or as part other products functions. For example: 1. Modality Worklist Query 2. Instance or Image Availability Query	DICOM (Sup 95) "Query"
Security Administration	Administrative actions create, modify, delete, query, and display the following: 1. Security attributes for data sets, data groups, or classes plus their atomic data elements or attributes. 2. Security attributes and auditable events for the application functions used for patient management, clinical processes, registry of business objects and methods, program creation and maintenance, etc. 3. Security domains according to various organizational categories such as entity-wide, institutional, departmental, etc. 4. Security categories or groupings for functions and data such as patient management, nursing, clinical, etc. 5. The allowable access permissions associated with functions and data, such as create, read, update, delete, and execution of specific functional units or object access or manipulation methods. 6. Security roles according to various task-grouping categories such as security administration, admissions desk, nurses, physicians, clinical specialists, etc. It also includes the association of permissions with roles for role-based access control. 7. User accounts. This includes assigning or changing password or other authentication data. It also includes the association of roles with users for role-based access control, or permissions with users for user-based access control. 8. Unauthorized user attempt to use security administration functions. 9. Audit enabling and disabling. 10. User authenticaton, authentication failure, authentication revocation, or signoff. 11. Configuration and other changes, e.g., software updates, that affect any software that processes protected information. Hardware changes may also be reported in this event. Security administration events should always be audited.	DICOM (Sup 95) "Security Alert"
Study-Object-Event	Study is created, modified, accessed, or deleted. This reports on addition of new instances to existing studies as well as creation of new studies.	DICOM (Sup 95) "DICOM Instances Accessed"

Trigger Event	Description	Source Vocabulary
Study-used	SOP Instances from a specific study are created, modified or accessed. One event covers all instances used for the particular study.	DICOM (Sup 95) "DICOM Instances Accessed"

Note: The IHE extension has reduced the scope of many of the IETF events to remove phrases like "checking for clinical contra-indications". This is done to highlight that the events should be reported are those that are related to the access, use, creation, and distribution of PHI. This audit log is not intended to be a general purpose monitoring system to track all kinds of medical activity. As a result, many clinically significant events will not be separately reported.

3.20.6.1 Audit Record Transportation

This profile defines two transport mechanisms for the audit messages:

1. Transport utilizing the Reliable Syslog protocol in "cooked" mode as defined in RFC-3195.
2. Transport utilizing the BSD Syslog protocol defined in RFC-3164.

The Audit repository shall support both transport mechanisms for the receipt of messages. Individual IHE Actors may choose to utilize either of the two transport mechanisms, unless they also comply with another Profile that further restricts the use. IHE recommends the use of reliable syslog because it deals with issues such as delivery confirmation, message loss prevention, and message truncation prevention.

The Reliable Syslog protocol specifies the use of local cache and storage. Messages are preserved locally until they are confirmed to have been successfully stored at the recipient. After delivery they may be removed at the convenience of the local machine and local functions.

3.20.6.2 Audit Record format

The IHE defines several audit record formats, and future profiles may define more message formats. An IHE actor shall utilize one or more of these audit record formats. All audit record formats utilize XML encoding and are defined by XML schema.

The present list of audit record schema are:

5. The IHE Audit Trail format. This is a schema based on the standards developed and issued by the IETF, HL7, and DICOM organizations to meet the medical auditing needs as specified by ASTM.
4. IHE Provisional Audit Record format, defined below. This was previously defined as part of the IHE Radiology technical framework. Its use is deprecated, this implies that no extensions will be made and new applications should use the new IHE Audit Trail format.

3.20.6.3 Audit Message Transports

The IHE actor will create the Audit Record and transmit this to the Audit Record Repository as soon as possible. When for some reason the Audit Record repository is not available, the IHE actor shall store the Audit Record in a local buffer until the Audit Record Repository is available again. The local Audit Record at the IHE actor may be deleted when this record has been transmitted to the Audit Record Repository.

Note: The Reliable Syslog protocol has explicit support for management of occasionally connected and mobile devices.

3.20.6.3.1 Reliable Syslog

The Reliable Syslog "cooked" mode defined in RFC-3195 shall be used to transport the audit messages. The schema used for the messages shall be identified as part of the "cooked" connection establishment.

3.20.6.3.2 BSD Syslog

The BSD syslog is appropriate in some situations, it was defined in the IHE Rad Technical Framework, and it is widely used legacy protocol. The XML messages are permitted to violate the BSD limitations in the following ways:

- The syslog port number shall be configurable, with the BSD port number (514) as the default.
- Messages are limited in length to 32768 bytes. Note that the underlying transport might not accept messages longer than 1024. They may be truncated. The Audit Repository must be prepared for arbitrary truncation of messages. The IHE Provisional schema uses shortened names to reduce the size of messages, but some may exceed 1024 bytes. When these are truncated the resulting XML will be incorrect and will need to be corrected by the Audit Repository to close the truncated portions of the message.
- The XML may contain Unicode characters that are encoded using the UTF-8 encoding rules. UTF-8 avoids utilizing the control characters that are mandated by the syslog protocol, but it may appear to be gibberish to a system that is not prepared for UTF-8. Audit repositories must accept UTF-8 encodings and store them without damage, e.g. preserve all 8 bits.

3.20.7 Audit Message Formats

3.20.7.1 RFC-3881 format

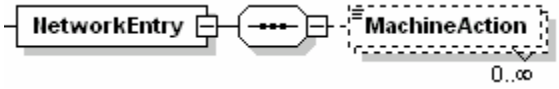
A common XML schema was defined based upon joint work by IHE, HL7, DICOM, ASTM E31, and the Joint NEMA/COCIR/JIRA Security and Privacy Committee. The IHE IT Infrastructure technical framework prefers use of this schema for audit records generated by all IHE actors. The schema can be found at:
http://www.xml.org/xml/schema/7f0d86bd/healthcare-security-audit.xsd

The DICOM Standard, Supplement 95 Audit Trail Messages provides vocabulary and further specification of the use of these schema elements for events that may occur in the context of DICOM equipment. IHE has evaluated this and determined that it is more broadly applicable, and extended it for more general healthcare use.

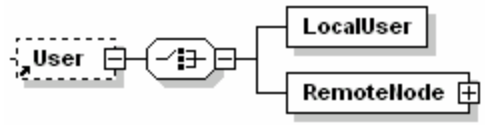
For reference, the schema elements are diagrammed below. The diagrams are read from left to right: elements to the right are part of the lefthandside element.

Required single element. A NetworkEntry element consists of exactly one MachineAction element.

Optional single element. A NetworkEntry element consists of zero or one MachineAction element.



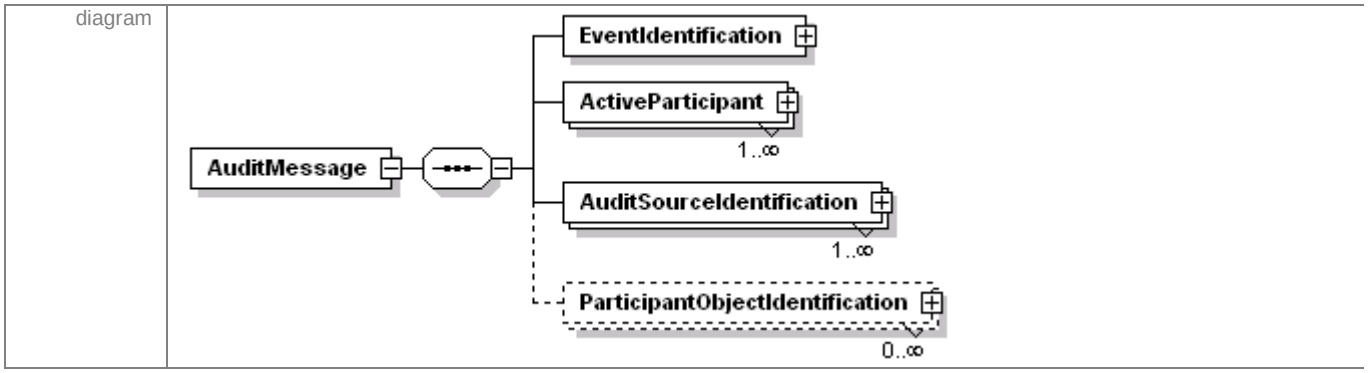
Optional multiple elements. A NetworkEntry element consists of zero or any number of MachineAction elements.



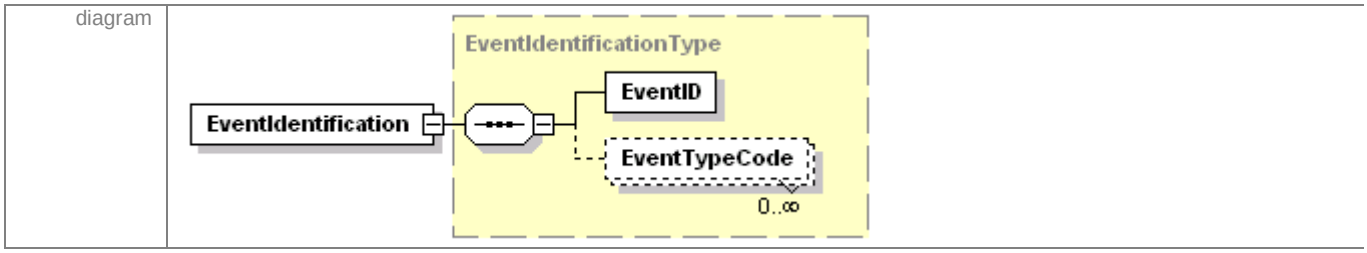
Selections of one out of several elements. A user consists either of a LocalUser element or of a RemoteNode element.

Compound element: The "+" in an element box means that the element consists of further elements. If these expansion elements have not occurred up to this point in the document, can be expected to follow below in the document.

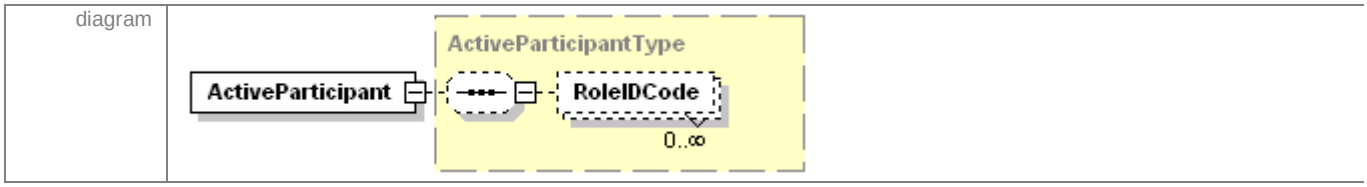
- element AuditMessage



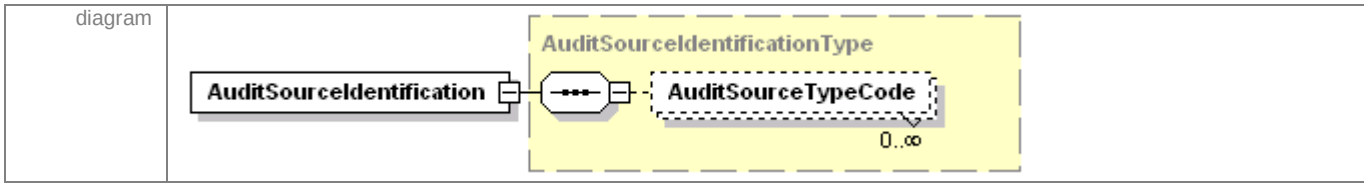
- element AuditMessage/EventIdentification



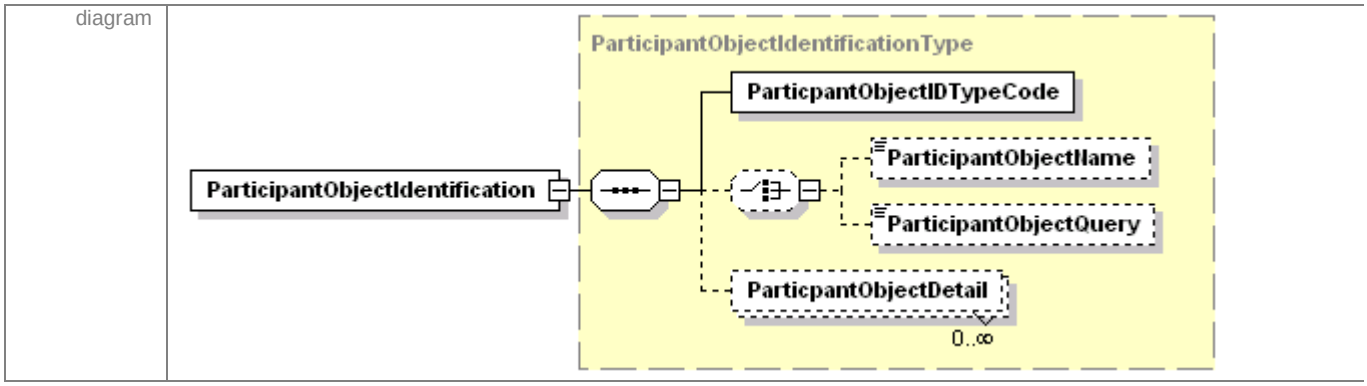
- element AuditMessage/ActiveParticipant



- element AuditMessage/AuditSourceIdentification



- element AuditMessage/ParticipantObjectIdentification



3.20.7.2 DICOM Audit Trail

A Secure Node actor shall be able to detect events that are defined by the DICOM standard in Supplement 95, and generate Record Audit Event transactions that conform to the DICOM standard when these events take place.

The DICOM Standard provides a schema for the basic messages and states that extensions are valid. This profile does not restrict private extensions that comply with the W3C XML encoding rules for the use of schemas, namespaces, etc.

3.20.7.3 IHE Audit Trail

The DICOM standard does not address all the kinds of security and privacy events that can take place in the healthcare environment. The following additional events are defined by IHE for use in healthcare. Secure nodes shall use these events when the DICOM standard events do not apply.

The notation used in these tables is that used in the DICOM standard. The messages shall be encoded as instances based on the RFC-3881 schema. This profile does not restrict private extensions to the RFC-3881 schema that comply with the W3C XML encoding rules for the use of schemas, namespaces, etc.

3.20.7.3.1 Health Services Provision Event

This message may be generated whenever health services are scheduled and performed within an instance or episode of care. These include scheduling, initiation, updates or amendments, performing or completing an act, and cancellation.

	Field Name	Opt.	Value Constraints
Event	Event ID	M	EV (IHE0001, IHE, "Health Services Provision Event")
	Event Action Code	M	EV: "C" (create) "R" (read) "U" (update) "D" (delete)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	*U*	*not specialized*
User (1..n)	User ID	M	The identity of the persons or processes manipulating the data. All that are known shall be included.
	UserName	*U*	*not specialized*
	UserIsRequestor	*U*	*not specialized*
	RoleIDCode	*U*	*not specialized*
Patients (1)	Participant Object Type Code	M	EV 1 (person)
	Participant Object Type Code Role	M	EV 1 (patient)
	Participant Object ID Type Code	M	EV 2 (patient ID)
	Participant Object ID	M	The patient ID
	Participant Object Name	U	The patient name
	Participant Object Detail	U	Description of the event

In cases where there is an event that applies to more than one patient, there shall be a separate audit message for each patient.

3.20.7.3.2 Medication Event

This message may be generated whenever there are medication orders or administration within an instance or episode of care. These include initial order, dispensing, delivery, and cancellation.

3380

	Field Name	Opt.	Value Constraints
Event	Event ID	M	EV (IHE0002, IHE, "Medication Event")
	Event Action Code	M	EV: "C" (create) "R" (read) "U" (update) "D" (delete)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	*U*	*not specialized*
User (1..n)	User ID	M	The identity of the persons or processes manipulating the data. All that are known shall be included.
	UserName	*U*	*not specialized*
	UserIsRequestor	*U*	*not specialized*
	RoleIDCode	*U*	*not specialized*
Patients (1)	Participant Object Type Code	M	EV 1 (person)
	Participant Object Type Code Role	M	EV 1 (patient)
	Participant Object ID Type Code	M	EV 2 (patient ID)
	Participant Object ID	M	The patient ID
	Participant Object Name	U	The patient name
	Participant Object Detail	U	Description of the event

In cases where there is an event that applies to more than one patient, there shall be a separate audit message for each patient.

3.20.7.3.3 Patient Care Resource Assignment Event

3385 This message may be generated whenever there are staffing or participant assignment actions relevant to the assignment of healthcare professionals, caregivers, attending physician, residents, medical students, consultants, etc. to a patient. It is also generated as a result of a change in assigned role or authorization, e.g., relative to healthcare status change, and de-assignment.

	Field Name	Opt.	Value Constraints
Event	Event ID	M	EV (IHE0003, IHE, "Patient Care Resource Assignment")
	Event Action Code	M	EV: "C" (create) "R" (read) "U" (update) "D" (delete)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	*U*	*not specialized*
User (1..n)	User ID	M	The identity of the persons or processes manipulating the data. All that are known shall be included.
	UserName	*U*	*not specialized*
	UserIsRequestor	*U*	*not specialized*
	RoleIDCode	*U*	*not specialized*
Patients (1)	Participant Object Type Code	M	EV 1 (person)
	Participant Object Type Code Role	M	EV 1 (patient)
	Participant Object ID Type Code	M	EV 2 (patient ID)
	Participant Object ID	M	The patient ID
	Participant Object Name	U	The patient name
	Participant Object Detail	U	Description of the event

3390

In cases where there is an event that applies to more than one patient, there shall be a separate audit message for each patient.

3.20.7.3.4 Patient Care Episode Event

3395 This message may be generated whenever there are specific patient care episodes or problems that occur within an instance of care. These include initial assignment, updates or amendments, resolution, completion, and cancellation.

	Field Name	Opt.	Value Constraints
Event	Event ID	M	EV (IHE0004, IHE, "Patient Care Episode")
	Event Action Code	M	EV: "C" (create) "R" (read) "U" (update) "D" (delete)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	*U*	*not specialized*
User (1..n)	User ID	M	The identity of the persons or processes manipulating the data. All that are known shall be included.
	UserName	*U*	*not specialized*
	UserIsRequestor	*U*	*not specialized*
	RoleIDCode	*U*	*not specialized*
Patients (1)	Participant Object Type Code	M	EV 1 (person)
	Participant Object Type Code Role	M	EV 1 (patient)
	Participant Object ID Type Code	M	EV 2 (patient ID)
	Participant Object ID	M	The patient ID
	Participant Object Name	U	The patient name
	Participant Object Detail	U	Description of the event

In cases where there is an event that applies to more than one patient, there shall be a separate audit message for each patient.

3.20.7.3.5 Patient Care Protocol Event

This message may be generated whenever there is a patient association with a care protocol. These include initial assignment, scheduling, updates or amendments, completion, and cancellation.

	Field Name	Opt.	Value Constraints
Event	Event ID	M	EV (IHE0005, IHE, "Patient Care Protocol")
	Event Action Code	M	EV: "C" (create) "R" (read) "U" (update) "D" (delete)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	*U*	*not specialized*
User (1..n)	User ID	M	The identity of the persons or processes manipulating the data. All that are known shall be included.
	UserName	*U*	*not specialized*
	UserIsRequestor	*U*	*not specialized*
	RoleIDCode	*U*	*not specialized*
Patients (1)	Participant Object Type Code	M	EV 1 (person)
	Participant Object Type Code Role	M	EV 1 (patient)
	Participant Object ID Type Code	M	EV 2 (patient ID)
	Participant Object ID	M	The patient ID
	Participant Object Name	U	The patient name
	Participant Object Detail	U	Description of the event

In cases where there is an event that applies to more than one patient, there shall be a separate audit message for each patient.

3410 #### 3.20.7.4 Other event reports

Events that do not correspond to DICOM events or IHE Extension events can be reported. They shall comply with RFC-3881.

3.20.7.5 Controlled Terminology for IHE Extensions

This profile defines the following controlled terminology for use in the IHE extensions.

Context ID ccc1
Audit Event ID
Type: Extensible Version: 2004xxxx

Coding Scheme Designator	Coding Scheme Version	Code Value	Code Meaning
IHE		IHE0001	Health Services Provision Event
IHE		IHE0002	Medication Event
IHE		IHE0003	Patient Care ResourceAssignment
IHE		IHE0004	Patient Care Episode
IHE		IHE0005	Patient Care Protocol

IHE Code Definitions (Coding Scheme Designator "IHE" Coding Scheme Version "2004")

Code Value	Code Meaning	Definition	Notes
IHE0001	Health Services Provision Event	Health services scheduled and performed within an instance or episode of care. This includes scheduling, initiation, updates or amendments, performing or completing the act, and cancellation.	
IHE0002	Medication Event	Medication orders and administration within an instance or episode of care. This includes initial order,dispensing, delivery, and cancellation.	
IHE0003	Patient Care Resource Assignment	Staffing or participant assignment actions relevant to the assignment of healthcare professionals, caregivers attending physician, residents, medical students, consultants, etc. to a patient It also includes change in assigned role or authorization, e.g., relative to healthcare status change, and de-assignment	
IHE0004	Patient Care Episode	Specific patient care episodes or problems that occur within an instance of care. This includes initial assignment, updates or amendments, resolution, completion, and cancellation.	
IHE0005	Patient Care Protocol	Patient association with a care protocol. This includes initial assignment, scheduling, updates or amendments, completion, and cancellation.	

3.20.7.6 IHE Provisional Audit Message Form

A provisional XML Schema was defined for the contents of the audit records generated by the IHE actors in the deprecated Basic Security Integration Profile as part of the IHE Radiology domain. The ATNA profile includes this schema as an alternative format for audit messages. It is less flexible than the IHE Audit Trail format, and is no longer the recommended format for IHE use. The preferred format is the IHE Audit Trail format with extensions that is described above.

However, the IHE Provisional Audit Message format is suitable for many diagnostic equipment settings and can be transformed into an equivalent IHE Audit Trail format. It is also installed and in use at many locations. So the IHE Provisional Audit Message format is part of the IHE IT profile. The transition from its format to the IHE Audit Trail format is encouraged to reduce the burden on Audit Repositories which may result from processing this alternative format.

A provisional XML Schema has been defined for the contents of the audit records generated by the IHE actors in the Basic Security Integration Profile from the radiology technical framework. The audit records are used to generate an audit record log for activities related to protected health information.

The IHE Provisional Audit Message Schema is described in ITI TF-2: Appendix F.

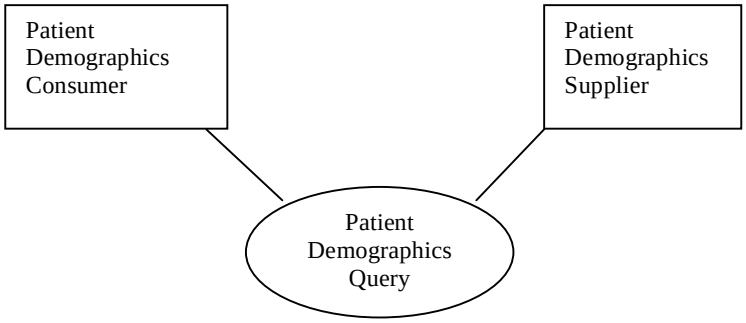
3.21 Patient Demographics Query

This section corresponds to Transaction ITI-21 of the IHE Technical Framework. Transaction ITI-21 is used by the Patient Demographics Consumer and Patient Demographics Supplier actors.

3.21.1 Scope

This transaction involves a request by the Patient Demographics Consumer Actor for information about patients whose demographic data match data provided in the query message. The request is received by the Patient Demographics Supplier Actor. The Patient Demographics Supplier Actor immediately processes the request and returns a response in the form of demographic information for matching patients.

3.21.2 Use Case Roles



Actor: Patient Demographics Consumer

Role: Requests a list of patients matching a minimal set of demographic criteria (*e.g.*, ID or partial name) from the Patient Demographics Supplier. Populates its attributes with demographic information received from the Patient Demographics Supplier.

Actor: Patient Demographics Supplier

Role: Returns demographic information for all patients matching the demographic criteria provided by the Patient Demographics Consumer.

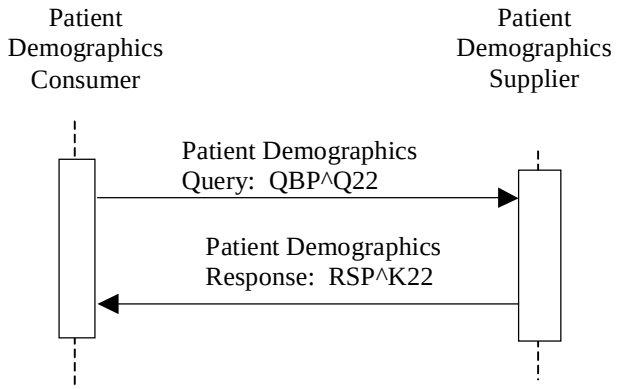
3.21.3 Referenced Standards

HL7: Version 2.5, Chapter 2 – Control

Version 2.5, Chapter 3 – Patient Administration

Version 2.5, Chapter 5 – Query

3.21.4 Interaction Diagram



3.21.4.1 Patient Demographics Query

3.21.4.1.1 Trigger Events

A Patient Demographics Consumer's need to select a patient based on demographic information about patients whose information matches a minimal set of known data will trigger the Patient Demographics Query based on the following HL7 trigger event:

Q22 – Find Candidates

3.21.4.1.2 Message Semantics

The Patient Demographics Query is conducted by the HL7 QBP^Q22 message. The Patient Demographics Consumer actor shall generate the query message whenever it needs to select from a list of patients whose information matches a minimal set of demographic data. The segments of the message listed below are required, and their detailed descriptions are provided in the following subsections.

Table 3.21-1 QBP Query by Parameter

QBP	Query by Parameter	Chapter in HL7 2.5
MSH	Message Header	2
QPD	Query Parameter Definition	5
RCP	Response Control Parameter	5

The receiver shall respond to the query by sending the RSP^K22 message. This satisfies the requirements of original mode acknowledgment; no intermediate ACK message is to be sent.

3.21.4.1.2.1 3.21.4.1.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2: Appendix C.1.2).

The Patient Demographics Query is always targeted at a single source of patient demographic information (referred to in this Transaction as the ***patient information source***). When more than one patient information source is available, Field *MSH-5-Receiving Application* specifies the patient information source that this query is targeting. The Patient Demographics Supplier shall return this value in *MSH-3-Sending Application* of the ACK message and of the RSP^K22 response. Note that, in a multi-domain environment, the patient information source identified by MSH-5 may or may not be associated with any of the Patient ID Domains listed in *QPD-8-What Domains Returned*.

A list shall be published of all Receiving Applications that the Patient Demographics Supplier supports, for the Patient Demographics Consumer to choose from. Each query is processed against patient demographic information associated with one and only one Patient ID Domain.

Field *MSH-9-Message Type* shall have at least two components. The first component shall have a value of **QBP**; the second component shall have a value of **Q22**. The third component is optional; however, if present, it shall have a value of **QBP_Q21**.

3.21.4.1.2.2 3.21.4.1.2.2 QPD Segment

The Patient Demographics Consumer Actor shall send attributes within the QPD segment as described in Table 3.21-2.

Table 3.21-2. IHE Profile - QPD segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	250	CE	R	0471	01375	Message Query Name
2	32	ST	R+		00696	Query Tag
3		QIP	R			Demographics Fields
8		CX	O			What Domains Returned

Adapted from the HL7 standard, version 2.5

3.21.4.1.2.2.1 3.21.4.1.2.2.1 Populating QPD-3-Demographics Fields

Field *QPD-3-Demographics Fields* consists of one or more repetitions, each of which contains two components that together contain the name and value of a distinct parameter to the query. Acceptable segments are PID and PD1.

The first component of each parameter contains the name of an HL7 element in the form

@<seg>.<field no>.<component no>.<subcomponent no>

The above format is populated according to common HL7 usage for specifying elements used in query parameters, as follows:

<seg> represents a 3-character segment ID from the HL7 Standard.

<field no> is the number of a field within the segment as shown in the SEQ column of the segment attribute table for the segment selected.

<component no>, for fields whose data types contain multiple components, shall contain the cardinal number of the component being valued. For fields whose data types do not contain multiple components, <component no> shall not be valued and its preceding period shall not appear.

<subcomponent no>, for components whose data types contain multiple subcomponents, shall contain the cardinal number of the subcomponent being valued. For components whose data types do not contain multiple subcomponents, <subcomponent no> shall not be valued and its preceding period shall not appear.

The second subcomponent of each parameter contains the value that is to be matched. If it is desired to constrain the quality of a match within the bounds of an algorithm known to the Supplier, the algorithm and constraint values may be specified in Fields QPD-4 through QPD-7.

At a minimum, the Patient Demographics Consumer may specify, and the Patient Demographics Supplier shall support, the fields in the following table.

Table 3.21-3. PDQ Profile – QPD-3 fields required to be supported

FLD	ELEMENT NAME
PID.3	Patient Identifier List
PID.5	Patient Name
PID.7	Date/Time of Birth
PID.8	Administrative Sex
PID.11	Patient Address
PID.18	Patient Account Number

An example of parameter expressions in QPD-3:

```
@PID.5.1^SMITH~@PID.8^F
```

requests all patients whose family name (first component of *PID-5-Patient Name*) matches the value SMITH and whose sex (*PID-8-Sex*) matches the value ‘female’.

3.21.4.1.2.2.2 3.21.4.1.2.2.1 Populating QPD-8-What Domains Returned

As is specified in the discussion of the Find Candidates (Q22) Query in Chapter 3 of the HL7 Standard, field QPD-8 restricts the set of domains for which identifiers are returned in PID-3:

In a multiple-domain environment, QPD-8 may be used to identify one or more domains of interest to the Patient Demographics Consumer and from which the Consumer wishes to obtain a value for *PID-3-Patient Identifier*. Note that the patient information source designated by MSH-5 may or may not be associated with any of the Patient ID Domains listed in *QPD-8-What Domains Returned*.

In a single-domain environment, QPD-8 may be ignored by the Patient Demographics Supplier. The Supplier shall always return both identifiers from the Patient ID Domain associated with the patient information source designated by *MSH-5-Receiving Application*.

Within field QPD-8, only component 4 (Assigning Authority) shall be valued.

The Patient Demographics Supplier may or may not be able to supply additional identifiers from the domains specified in QPD-8. A discussion of how QPD-8 is processed is included in the architectural discussion in the "Using Patient Data Query (PDQ) in a Multi-Domain Environment" section (ITI TF-2: Appendix M).

3.21.4.1.2.3 3.21.4.1.2.3 RCP Segment

The Patient Demographics Consumer Actor shall send attributes within the RCP segment as described in Table 3.21-5. Fields not listed are optional and may be ignored.

Table 3.21-5. IHE Profile - RCP segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	1	ID	R	0091	00027	Query Priority
2	10	CQ	O	0126	00031	Quantity Limited Request

Adapted from the HL7 standard, version 2.5

3.21.4.1.2.3.1 3.21.4.1.2.3.1 Populating RCP-1-Query Priority

Field *RCP-1-Query Priority* shall always contain **I**, signifying that the response to the query is to be returned in Immediate mode.

3.21.4.1.2.3.2 3.21.4.1.2.3.2 Populating RCP-2-Quantity Limited Request

The Patient Demographics Consumer Actor may request that responses to the query be sent, using the HL7 Continuation Protocol, in increments of a specified number of patient records. (In the context of the HL7 query, a patient record is defined as the PID segment and any segments accompanying it for each patient.) It is desirable to request an incremental response if the query could result in hundreds or thousands of matches or "hits."

The Patient Demographics Supplier Actor shall support the HL7 Continuation Protocol.

Field RCP-2 is of data type CQ, which contains two components. The first component contains the number of increments, while the second component contains the kind of increment, always `RD` to signify that incremental replies are specified in terms of records.

For example, `50^RD` requests 50 records at a time.

See the "Incremental Response Processing" section (ITI TF-2: 3.21.4.1.3.3) and the "Expected Actions" section of the Patient Demographics Query Response message (ITI TF-2: 3.21.4.2.3) for more information on the implementation of the continuation protocol.

3.21.4.1.3 Expected Actions

3.21.4.1.3.1 3.21.4.1.3.1 Immediate Acknowledgement

The Patient Demographics Supplier shall immediately return an RSP^K22 response message as specified below in Section 3.21.4.2, "Patient Demographics Response." The RSP^K22 response message incorporates original mode application acknowledgment as specified in the "Acknowledgment Modes" section (ITI TF-2: C.1.3). The Supplier shall use *MSH-3-Sending Application* of the RSP^K22 to return the value it received from the Patient Demographics Consumer in Field *MSH-5-Receiving Application* of the QBP^Q22 message.

3.21.4.1.3.2 3.21.4.1.3.2 Query Parameter Processing

The Patient Demographics Supplier Actor shall be capable of accepting, searching on, and responding with attributes in the QPD segment as specified in Table 3.21-2.

The Patient Demographics Supplier Actor must be capable of receiving all valid combinations of subcomponents that make up the Assigning Authority component (*i.e.*, all valid combinations of QPD-3.8).

Handling of phonetic issues, alternate spellings, upper and lower case, wildcards, accented characters, etc., if deemed appropriate, is to be supported by the Patient Demographics Supplier rather than by the Patient Demographics Consumer. The Supplier shall return at least all exact matches to the query parameters sent by the Consumer; IHE does not further specify matching requirements.

3.21.4.1.3.3 3.21.4.1.3.3 Incremental Response Processing

The Patient Demographics Supplier Actor shall be capable of accepting and processing attributes in the RCP segment as listed in Table 3.21-5. In particular, the Patient Demographics Supplier Actor shall respond in immediate mode (as specified by a *RCP-1-Query Priority* value of **I**).

Also, the Patient Demographics Supplier Actor shall be able to interpret *RCP-2-Quantity Limited Request* to return successive responses of partial lists of records according to the HL7 Continuation Protocol, as described in Section 3.21.4.2 below and in the HL7 Standard.

3.21.4.2 Patient Demographics Response

3.21.4.2.1 Trigger Events

The Patient Demographics Supplier's response to the Find Candidates message shall be the following message:

K22 – Find Candidates response

3.21.4.2.2 Message Semantics

The Patient Demographics Response is conducted by the RSP^K22 message. The Patient Demographics Supplier Actor shall generate this message in direct response to the QBP^Q22 message previously received. This message satisfies the Application Level, Original Mode Acknowledgement for the HL7 QBP^Q22 message.

The segments of the message listed without enclosing square brackets in the Table below are required. Detailed descriptions of all segments listed in the table below are provided in the following subsections. Other segments of the message are optional.

Table 3.21-6 RSP Segment Pattern Response

RSP	Segment Pattern Response	Chapter in HL7 2.5
MSH	Message Header	2
MSA	Message Acknowledgement	2
[{ERR}]	Error	2
QAK	Query Acknowledgement	5
QPD	Query Parameter Definition	5
[{ PID	Patient Identification	3
[PD1]		
[QRI] }]	Query Response Instance	5
[DSC]	Continuation Pointer	2

3.21.4.2.2.1 3.21.4.2.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2: C.1.2).

Field *MSH-3-Sending Application* specifies the patient information source that processed the query. The Patient Demographics Supplier shall use Field *MSH-3-Sending Application* of the RSP^K22 message to return the value it received from the Patient Demographics Consumer in Field *MSH-5-Receiving Application* of the QBP^Q22 message.

Field *MSH-9-Message Type* shall have at least two components. The first component shall have a value of **RSP**; the second component shall have a value of **K22**. The third component is optional; however, if present, it shall have a value of **RSP_K22**.

3.21.4.2.2.2 3.21.4.2.2.2 MSA Segment

The Patient Demographics Supplier Actor is not required to send any attributes within the MSA segment beyond what is specified in the HL7 standard. See the "Acknowledgment Modes" section (ITI TF-2: C.1.3) for the list of all required and optional fields within the MSA segment.

3.21.4.2.2.3 3.21.4.2.2.3 QAK Segment

The Patient Demographics Supplier Actor shall send attributes within the QAK segment as defined in table 3.21-7. For the details on filling in QAK-2 (Query Response Status) refer to the "Patient Demographics Supplier Actor Query Response Behavior" section (ITI TF-2: 3.21.4.2.2.8).

Table 3.21-7. PDQ Profile - QAK segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	32	ST	R		00696	Query Tag
2	2	ID	R+	0208	00708	Query Response Status

Adapted from the HL7 standard, version 2.5

3.21.4.2.2.4 3.21.4.2.2.4 QPD Segment

The Patient Demographics Supplier Actor shall echo the QPD Segment value that was sent in the QBP^Q22 message.

3.21.4.2.2.5 3.21.4.2.2.5 PID Segment

The Patient Demographics Supplier Actor shall return one PID segment group (*i.e.*, one PID segment plus any segments associated with it in the message syntax shown in Table 3.21-5) for

each matching patient record found. The Supplier shall return the attributes within the PID segment as specified in Table 3.21-8. In addition, the Patient Demographics Supplier Actor shall return all other attributes within the PID segment for which it is able to supply values.

Table 3.21-8. PDQ Profile - PID segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
3	250	CX	R		00106	Patient Identifier List
5	250	XPN	R		00108	Patient Name
7	26	TS	R2		00110	Date/Time of Birth
8	1	IS	R2	0001	00111	Administrative Sex
11	250	XAD	R2		00114	Patient Address
18	250	CX	R2		00121	Patient Account Number

Adapted from the HL7 standard, version 2.5

The Patient Demographics Supplier may or may not be able to supply additional identifiers from the domains specified in QPD-8. Inability to supply an identifier in a particular domain is not an error, provided that the domain is recognized.

The PID segment and its associated PD1 and QRI segments are returned only when the Patient Demographics Supplier Actor is able to associate the search information in QPD-3 with one or more patient records in the patient information source associated with *MSH-5-Receiving Application*. See the "Patient Demographics Supplier Actor Query Response Behavior" section (ITI TF-2: 3.21.4.2.2.8) for a detailed description of how the Patient Demographics Supplier Actor responds to the query request under various circumstances.

3.21.4.2.2.6 3.21.4.2.2.6 QRI Segment

For each patient for which the Patient Demographics Supplier Actor returns a PID Segment, it may optionally return the QRI (Query Response Instance) segment, but is not required to do so. Refer to the HL7 Standard, Version 2.5, Chapter 5, Section 5.5.5, for more information.

3.21.4.2.2.7 3.21.4.2.2.7 DSC Segment

If a number of records is specified in *RCP-2-Quantity Limited Request*, the Patient Demographics Supplier Actor shall return an incremental response of that number of records when the number of matching records it finds exceeds the number of records specified in RCP-2.

As long as the Patient Demographics Supplier Actor has records to return in addition to those returned in the incremental response, the Supplier shall return a DSC Segment. The single field of the DSC Segment shall contain a unique alphanumeric value (the Continuation Pointer) that the Patient Demographics Consumer may return in the MSH segment of its acknowledgement of the RSP^K22 message to request the next increment of responses. The Supplier shall return

increments as many times as the Consumer requests them (and there are increments to return), and shall stop when the Consumer returns an acknowledgement that does not request additional increments (or when there are no more increments to return).

3.21.4.2.2.8 3.21.4.2.2.8 Patient Demographics Supplier Actor Query Response Behavior

The Patient Demographics Supplier shall perform the matching of patient data based on the query parameter values it receives. The information provided by the Patient Demographics Supplier Actor to Patient Demographics Consumer Actors is a list of possible matching patients from the patient information source associated with the value that the Consumer sent in *MSH-5-Receiving Application* of the query message.

If domains are specified in *QPD-8-What Domains Returned* and are recognized by the Patient Demographics Supplier, the response will also, for each patient, contain any Patient ID values found in the specified domains.

The mechanics of the matching algorithms used are internal to the Patient Demographics Supplier Actor and are outside of the scope of this framework.

The Patient Demographics Supplier Actor shall respond to the query request as described by the following 3 cases:

Case 1: The Patient Demographics Supplier Actor finds (in the patient information source associated with *MSH-5-Receiving Application*) at least one patient record matching the criteria sent in *QPD-3-Demographics Fields*. No patient identifier domains are requested in *QPD-8-What Domains Returned*.

AA (application accept) is returned in MSA-1.

OK (data found, no errors) is returned in QAK-2.

One PID segment group (*i.e.*, one PID segment plus any segments associated with it in the message syntax shown in Table 3.21-5) is returned from the patient information source for each patient record found. If the Patient Demographics Supplier Actor returns data for multiple patients, it shall return these data in successive occurrences of the PID segment group.

Within each PID segment, field *PID-3-Patient Identifier List* contains one or more identifiers from the Patient ID Domain associated with the patient data source identified by *MSH-5-Receiving Facility*.

If an incremental number of records is specified in *RCP-2-Quantity Limited Request*, and the number of records to be sent exceeds that incremental number, the Supplier returns only the incremental number of records, followed by a DSC segment containing a uniquely valued Continuation Pointer.

Case 2: The Patient Demographics Supplier Actor finds (in the patient information source associated with *MSH-5-Receiving Application*) at least one patient record matching the criteria sent in *QPD-3-Demographics Fields*. One or more patient identifier domains are requested in *QPD-8-What Domains Returned*; the Supplier recognizes all the requested domains.

AA (application accept) is returned in MSA-1.

OK (data found, no errors) is returned in QAK-2.

One PID segment group (*i.e.*, one PID segment plus any segments associated with it in the message syntax shown in Table 3.21-5) is returned for each matching patient record found. If the Patient Demographics Supplier Actor returns data for multiple patients, it shall return these data in successive occurrences of the PID segment group.

Within each PID segment, field *PID-3-Patient Identifier List* contains, in successive occurrences delimited by the repetition separator, the identifiers from all the Patient ID Domains requested in QPD-8. In each occurrence of PID-3, component 4 contains the assigning authority value for one Patient ID Domain, and component 1 contains the Patient ID value in that domain (or is left blank if an identifier does not exist in that domain).

If an incremental number of records is specified in *RCP-2-Quantity Limited Request*, and the number of records to be sent exceeds that incremental number, the Supplier returns only the incremental number of records, followed by a DSC segment containing a uniquely valued Continuation Pointer.

Case 3: The Patient Demographics Supplier Actor does not recognize one or more of the domains in *QPD-8-What Domains Returned*.

AE (application error) is returned in MSA-1 and in QAK-2.

For each domain that was not recognized, an ERR segment is returned in which the components of *ERR-2-Error Location* are valued as follows.

COMP #	COMPONENT NAME	VALUE
1	Segment ID	**QPD**
2	Sequence	**1**
3	Field Position	**8**
4	Field Repetition	*(see below)*
5	Component Number	*(empty)*
6	Subcomponent Number	*(empty)*

ERR-2.4-Field Repetition identifies the ordinal occurrence of QPD-8 that contained the unrecognized domain. As specified by HL7, *ERR-2.5-Component Number* and *ERR-2.6-Subcomponent Number* are not valued because we are referring to the entire field QPD-8.

ERR-3-HL7 Error Code is populated with the error condition code **204** (unknown key identifier). Together with the values in ERR-2, this signifies that the Patient Demographics Supplier Actor did not recognize the domain for *QPD-8-What Domains Returned*.

3.21.4.2.3 Expected Actions

3740 The Patient Demographics Consumer will use the demographic information provided by the Patient Demographics Supplier to perform the functions for which it requested the information, *e.g.*, providing a pick list to the user.

The Consumer will return an original mode acknowledgement as described in the HL7 Standard, populating field *MSH-14-Continuation Pointer* as follows.

3745 If the Supplier returned a value in *DSC-1-Continuation Pointer*, indicating that there are more matching patients remaining to be sent than could be sent in a single increment, ***and*** if the Consumer wishes to receive the next increment, the Consumer shall populate MSH-14 with the value it received in DSC-1. The Supplier will then send the next increment of matching records, and will again populate DSC-1 (with a new unique value) if there are yet more records to be sent.

3750 If the Supplier did not return a value in *DSC-1-Continuation Pointer*, or if the Supplier did return a value in DSC-1 but the Consumer does not wish to receive the next increment, the Consumer shall return an empty value in MSH-14.

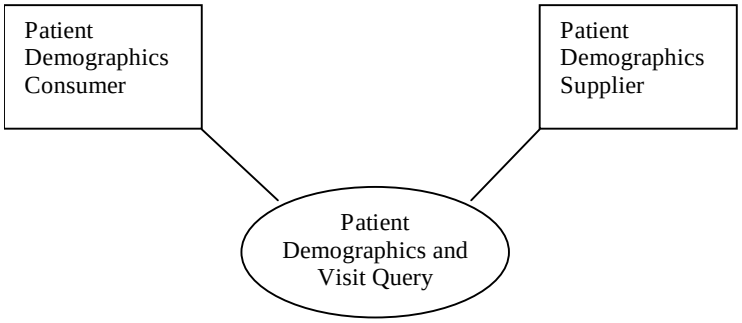
3.22 Patient Demographics and Visit Query

This section corresponds to Transaction ITI-22 of the IHE Technical Framework. Transaction ITI-22 is used by the Patient Demographics Consumer and Patient Demographics Supplier actors.

3.22.1 Scope

This transaction involves a request by the Patient Demographics Consumer Actor for information about patients whose demographic and visit data match data provided in the query message. The request is received by the Patient Demographics Supplier actor. The Patient Demographics Supplier actor immediately processes the request and returns a response in the form of demographic and visit information for matching patients.

3.22.2 Use Case Roles



Actor: Patient Demographics Consumer

Role: Requests a list of patients matching a minimal set of demographic (*e.g.*, ID or partial name) and visit criteria from the Patient Demographics Supplier. Populates its attributes with demographic and visit information received from the Patient Demographics Supplier.

Actor: Patient Demographics Supplier

Role: Returns demographic and visit information for all patients matching the demographic and visit criteria provided by the Patient Demographics Consumer.

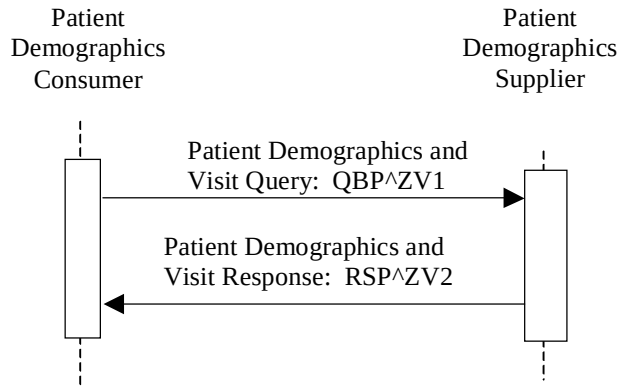
3.22.3 Referenced Standards

HL7: Version 2.5, Chapter 2 – Control

Version 2.5, Chapter 3 – Patient Administration

Version 2.5, Chapter 5 – Query

3.22.4 Interaction Diagram



3.22.4.1 Patient Demographics and Visit Query

3.22.4.1.1 Trigger Events

A Patient Demographics Consumer's need to select a patient based on demographic and visit information about patients whose information matches a minimal set of known data will trigger the Patient Demographics and Visit Query based on the following HL7 trigger event:

ZV1 – Find Candidates from Visit Information

3.22.4.1.2 Message Semantics

The Patient Demographics and Visit Query transaction is conducted by the HL7 QBP^ZV1 message. The Patient Demographics Consumer actor shall generate the query message whenever it needs to select from a list of patients whose information matches a minimal set of demographic and visit data. The segments of the message listed below are required, and their detailed descriptions are provided in the following subsections.

Table 3.22-1. QBP Query by Parameter

QBP	Query by Parameter	Chapter in HL7 2.5
MSH	Message Header	2
QPD	Query Parameter Definition	5
RCP	Response Control Parameter	5

The receiver shall respond to the query by sending the RSP^ZV2 message. This satisfies the requirements of original mode acknowledgment; no intermediate ACK message is to be sent.

3.22.4.1.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2: C.1.2).

The Patient Demographics and Visit Query is always targeted at a single source of patient demographic information (referred to in this Transaction as the ***patient information source***). When more than one patient information source is available, Field *MSH-5-Receiving Application* specifies the patient information source that this query is targeting. The Patient Demographics Supplier shall return this value in *MSH-3-Sending Application* of the ACK message and of the RSP^ZV2 response. Note that, in a multi-domain environment, the patient information source identified by MSH-5 may or may not be associated with any of the Patient ID Domains listed in *QPD-8-What Domains Returned*.

A list shall be published of all Receiving Applications that the Patient Demographics Supplier supports, for the Patient Demographics Consumer to choose from. Each query is processed against patient demographic information associated with one and only one Patient ID Domain.

Field *MSH-9-Message Type* shall have at least two components. The first component shall have a value of **QBP**; the second component shall have a value of **ZV1**. The third component is optional; however, if present, it shall have a value of **QBP_Q21**.

3.22.4.1.2.2 QPD Segment

The Patient Demographics Consumer Actor shall send attributes within the QPD segment as described in Table 3.22-2.

Table 3.22-2. PDQ Profile - QPD segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	250	CE	R	0471	01375	Message Query Name
2	32	ST	R+		00696	Query Tag
3		QIP	R			Demographics and Visit Fields
8		CX	O			What Domains Returned

Adapted from the HL7 standard, version 2.5

3.22.4.1.2.2.1 Parameters in QPD-3-Demographics and Visit-Related Fields

Field *QPD-3-Demographics and Visit-Related Fields* consists of one or more repetitions, each of which contains two components that together contain the name and value of a distinct parameter to the query. Acceptable segments are PID, PD1, PV1, and PV2.

The first component of each parameter contains the name of an HL7 element in the form

@<seg>.<field no>.<component no>.<subcomponent no>

The above format is populated according to common HL7 usage for specifying elements used in query parameters, as follows:

<seg> represents a 3-character segment ID from the HL7 Standard.

<field no> is the number of a field within the segment as shown in the SEQ column of the segment attribute table for the segment selected.

<component no>, for fields whose data types contain multiple components, shall contain the cardinal number of the component being valued. For fields whose data types do not contain multiple components, <component no> should not be valued and its preceding period should not appear.

<subcomponent no>, for components whose data types contain multiple subcomponents, shall contain the cardinal number of the subcomponent being valued. For components whose data types do not contain multiple subcomponents, <subcomponent no> should not be valued and its preceding period should not appear.

The second subcomponent of each parameter contains the value that is to be matched. If it is desired to constrain the quality of a match within the bounds of an algorithm known to the Supplier, the algorithm and constraint values may be specified in Fields QPD-4 through QPD-7.

At a minimum, the Patient Demographics Consumer may specify, and the Patient Demographics Supplier must support, the fields in the following table. Support for other fields is optional.

Table 3.22-3. PDQ Profile – QPD-3 fields required to be supported

FLD	ELEMENT NAME
PID.3	Patient Identifier List
PID.5	Patient Name
PID.7	Date/Time of Birth
PID.8	Administrative Sex
PID.11	Patient Address
PID.18	Patient Account Number

In addition, it is recommended that the Patient Demographics Supplier support the fields in the following table. Some fields may not be relevant to particular care settings (*e.g.*, inpatient, day patient) and will thus not be supportable by domains in those care settings.

Table 3-22.4. PDQ Profile – QPD-3 fields recommended to be supported

FLD	ELEMENT NAME
PV1.2	Patient Class
PV1.3	Assigned Patient Location
PV1.7	Attending Doctor
PV1.8	Referring Doctor

PV1.9	Consulting Doctor
PV1.10	Hospital Service
PV1.17	Admitting Doctor
PV1.19	Visit Number

Examples of parameter expressions in QPD-3:

```
@PID.5.1^SMITH~@PID.8^F
```

requests all patients whose family name (first component of *PID-5-Patient Name*) matches the value SMITH and whose sex (*PID-8-Sex*) matches the value 'female'.

```
@PV1.3.2^389~@PV1.3.3^2
```

requests all patients whose room number (second component of *PV1-3-Assigned Patient Location*) matches the value 389 and whose bed number (third component of *PV1-3-Assigned Patient Location*) matches the value 2.

3.22.4.1.2.2.2 Populating QPD-8-What Domains Returned

As in the Patient Demographics Query (Transaction ITI-21), field QPD-8 restricts the set of domains for which identifiers are returned in PID-3:

In a multiple-domain environment, QPD-8 may be used to identify one or more domains of interest to the Patient Demographics Consumer and from which the Consumer wishes to obtain a value for *PID-3-Patient Identifier*. Note that the patient information source designated by MSH-5 may or may not be associated with any of the Patient ID Domains listed in *QPD-8-What Domains Returned*.

In a single-domain environment, QPD-8 may be ignored by the Patient Demographics Supplier. The Supplier shall always return both identifiers from the Patient ID Domain associated with the patient information source designated by *MSH-5-Receiving Application*.

Within field QPD-8, only component 4 (Assigning Authority) shall be valued.

The Patient Demographics Supplier may or may not be able to supply additional identifiers from the domains specified in QPD-8. A discussion of how QPD-8 is processed is included in the architectural discussion in the "Using Patient Data Query (PDQ) in a Multi-Domain Environment" section (ITI TF-2: Appendix M).

3.22.4.1.2.3 RCP Segment

The Patient Demographics Consumer Actor shall send attributes within the RCP segment as described in Table 3.22-5. Fields not listed are optional.

Table 3.22-5. IHE Profile - RCP segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	1	ID	R	0091	00027	Query Priority
2	10	CQ	O	0126	00031	Quantity Limited Request

Adapted from the HL7 standard, version 2.5

3.22.4.1.2.3.1 Populating RCP-1-Query Priority

Field *RCP-1-Query Priority* shall always contain **I**, signifying that the response to the query is to be returned in Immediate mode.

3.22.4.1.2.3.2 Populating RCP-2-Quantity Limited Request

The Patient Demographics Consumer Actor may request that responses to the query be sent, using the HL7 Continuation Protocol, in increments of a specified number of patient records. (In the context of the HL7 query, a patient record is defined as the PID segment and any segments accompanying it for each patient.) It is desirable to request an incremental response if the query could result in hundreds or thousands of matches or "hits."

The Patient Demographics Supplier Actor shall support the HL7 Continuation Protocol.

Field RCP-2 is of data type CQ, which contains two components. The first component contains the number of increments, while the second component contains the kind of increment, always `RD` to signify that incremental replies are specified in terms of records.

For example, `50^RD` requests 50 records at a time.

See the "Incremental Response Processing" section (ITI TF-2: 3.22.4.1.3.3) and the "Expected Actions" section of the Patient Demographics Query Response message (ITI TF-2: 3.22.4.2.3) for more information on the implementation of the continuation protocol.

3.22.4.1.3 Expected Actions

3.22.4.1.3.1 Immediate Acknowledgement

The Patient Demographics Supplier shall immediately return an RSP^ZV2 response message as specified below in Section 3.22.4.2, "Patient Demographics Response." The RSP^ZV2 response message incorporates original mode application acknowledgment as specified in the "Acknowledgment Modes" section (ITI TF-2: C.1.3). The Supplier shall use Field *MSH-3-Sending Application* of the RSP^ZV2 message to return the value it received from the Patient Demographics Consumer in Field *MSH-5-Receiving Application* of the QBP^ZV1 message.

3.22.4.1.3.2 Query Parameter Processing

The Patient Demographics Supplier Actor shall be capable of accepting, searching on, and responding with attributes in the QPD segment as specified in Table 3.22-2.

The Patient Demographics Supplier Actor must be capable of receiving all valid combinations of subcomponents that make up the Assigning Authority component (*i.e.*, all valid combinations of QPD-3.8).

Handling of phonetic issues, alternate spellings, upper and lower case, wildcards, accented characters, etc., if deemed appropriate, is to be supported by the Patient Demographics Supplier rather than by the Patient Demographics Consumer. The Supplier shall return at least all exact matches to the query parameters sent by the Consumer; IHE does not further specify matching requirements.

3.22.4.1.3.3 Incremental Response Processing

The Patient Demographics Supplier Actor shall be capable of accepting and processing attributes in the RCP segment as listed in Table 3.22-5. In particular, the Patient Demographics Supplier Actor shall respond in immediate mode (as specified by a *RCP-1-Query Priority* value of **I**).

Also, the Patient Demographics Supplier Actor shall be able to interpret *RCP-2-Quantity Limited Request* to return successive responses of partial lists of records according to the HL7 Continuation Protocol, as described in Section 3.22.4.2 below and in the HL7 Standard.

3.22.4.2 Patient Demographics and Visit Response

3.22.4.2.1 Trigger Events

The Patient Demographics Supplier's response to the Find Candidates with Visit Information message shall be the following message:

ZV2 – Find Candidates with Visit Information response

3.22.4.2.2 Message Semantics

The Patient Demographics and Visit Response transaction is conducted by the RSP^ZV2 message. The Patient Demographics Supplier Actor shall generate this message in direct response to the QBP^ZV1 message previously received. This message satisfies the Application Level, Original Mode Acknowledgement for the HL7 QBP^ZV1 message.

The segments of the message listed without enclosing square brackets in Table 3.22-6 are required. Detailed descriptions of all segments listed in the table below are provided in the following subsections. Other segments of the message are optional.

Table 3.22-6 RSP Segment Pattern Response

RSP	Segment Pattern Response	Chapter in HL7 2.5
MSH	Message Header	2
MSA	Message Acknowledgement	2
[{ERR}]	Error	2
QAK	Query Acknowledgement	5
QPD	Query Parameter Definition	5
[{ PID	Patient Identification	3
[PD1]	Additional Patient Demographics	3
PV1	Patient Visit	3
[PV2]	Patient Visit – Additional Information	3
[QRI] }]	Query Response Instance	5
[DSC]	Continuation Pointer	2

3.22.4.2.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2: C.1.2).

Field *MSH-3-Sending Application* specifies the patient information source that processed the query. The Patient Demographics Supplier shall use Field *MSH-3-Sending Application* of the RSP^ZV2 message to return the value it received from the Patient Demographics Consumer in Field *MSH-5-Receiving Application* of the QBP^Q22 message.

Field *MSH-9-Message Type* shall have at least two components. The first component shall have a value of **RSP**; the second component shall have a value of **ZV2**. The third component is optional; however, if present, it shall have a value of **RSP_ZV2**.

3.22.4.2.2.2 MSA Segment

The Patient Demographics Supplier Actor is not required to send any attributes within the MSA segment beyond what is specified in the HL7 standard. See the "Acknowledgment Modes" section (ITI TF-2: C.1.3) for the list of all required and optional fields within the MSA segment.

3.22.4.2.2.3 QAK Segment

The Patient Demographics Supplier Actor shall send attributes within the QAK segment as defined in table 3.22-7. For the details on filling in QAK-2 (Query Response Status) refer to the "Patient Demographics Supplier Actor Query Response Behavior" section (ITI TF-2: 3.22.4.2.2.11).

Table 3.22-7. IHE Profile - QAK segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	32	ST	R		00696	Query Tag
2	2	ID	R+	0208	00708	Query Response Status

Adapted from the HL7 standard, version 2.5

3.22.4.2.2.4 QPD Segment

The Patient Demographics Supplier Actor shall echo the QPD Segment value that was sent in the QBP^ZV1 message.

3.22.4.2.2.5 PID Segment

The Patient Demographics Supplier Actor shall return one PID segment group (*i.e.*, one PID segment plus any segments associated with it in the message syntax shown in Table 3.22-6) for each matching patient record found. The Supplier shall return the attributes within the PID segment as specified in Table 3.22-8. In addition, the Patient Demographics Supplier Actor shall return all other attributes within the PID segment for which it is able to supply values.

Table 3.22-8. PDQ Profile - PID segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
3	250	CX	R		00106	Patient Identifier List
5	250	XPN	R		00108	Patient Name
7	26	TS	R2		00110	Date/Time of Birth
8	1	IS	R2	0001	00111	Administrative Sex
11	250	XAD	R2		00114	Patient Address
18	250	CX	R2		00121	Patient Account Number

Adapted from the HL7 standard, version 2.5

The Patient Demographics Supplier may or may not be able to supply additional identifiers from the domains specified in QPD-8. Inability to supply an identifier in a particular domain is not an error, provided that the domain is recognized.

The PID segment and the PD1, PV1, PV2, and QRI segments that are associated with it are returned only when the Patient Demographics Supplier Actor is able to associate the search information in QPD-3 with one or more patient records in the patient information source associated with *MSH-5-Receiving Application*. See the "Patient Demographics Supplier Actor Query Response Behavior" section (ITI TF-2: 3.22.4.2.2.11) for a detailed description of how the Patient Demographics Supplier Actor responds to the query request under various circumstances.

3.22.4.2.2.6 PD1 Segment

For each patient for which the Patient Demographics Supplier Actor returns a PID segment, it may optionally return the PD1 (Patient Additional Demographics) segment, but is not required to do so.

3.22.4.2.2.7 PV1 Segment

For each patient for which the Patient Demographics Supplier Actor returns a PID segment, it shall also return a PV1 Segment in which attributes are populated as specified in Table 3.22-9. In addition, the Patient Demographics Supplier Actor shall return all other attributes within the PV1 segment for which it is able to supply values.

Table 3.22-9. PDQ Profile – PV1 segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
2	1	IS	R	0004	00132	Patient Class
3	80	PL	R2		00133	Assigned Patient Location
7	250	XCN	R2	0010	00137	Attending Doctor
8	250	XCN	R2	0010	00138	Referring Doctor
9	250	XCN	R2	0010	00139	Consulting Doctor
10	3	IS	R2	0069	00140	Hospital Service
17	250	XCN	R2	0010	00147	Admitting Doctor
19	250	CX	R2		00149	Visit Number

Adapted from the HL7 standard, version 2.5

3.22.4.2.2.8 PV2 Segment

For each patient for which the Patient Demographics Supplier Actor returns a PID segment, it may optionally return the PV2 (Patient Visit – Additional Information) segment, but is not required to do so.

3.22.4.2.2.9 QRI Segment

For each patient for which the Patient Demographics Supplier Actor returns a PID segment, it may optionally return the QRI (Query Response Instance) segment, but is not required to do so. Refer to the HL7 Standard, Version 2.5, Chapter 5, Section 5.5.5, for more information.

3.22.4.2.2.10 DSC Segment

If a number of records is specified in *RCP-2-Quantity Limited Request*, the Patient Demographics Supplier Actor shall return an incremental response of that number of records when the number of matching records it finds exceeds the number of records specified in RCP-2.

As long as the Patient Demographics Supplier Actor has records to return in additional to those returned in the incremental response, the Supplier shall return a DSC Segment. The single field of the DSC Segment shall contain a unique alphanumeric value (the Continuation Pointer) that the Patient Demographics Consumer may return in the MSH segment of its acknowledgement of the RSP^ZV2 message to request the next increment of responses. The Supplier shall return increments as many times as the Consumer requests them (and there are increments to return), and shall stop when the Consumer returns an acknowledgement that does not request additional increments (or when there are no more increments to return).

3.22.4.2.2.11 Patient Demographics Supplier Actor Query Response Behavior

The Patient Demographics Supplier shall perform the matching of patient data based on the query parameter values it receives. The information provided by the Patient Demographics Supplier Actor to Patient Demographics Consumer Actors is a list of possible matching patients from the patient information source associated with the value that the Consumer sent in *MSH-5-Receiving Application* of the query message.

If domains are specified in *QPD-8-What Domains Returned* and are recognized by the Patient Demographics Supplier, the response will also, for each patient, contain any Patient ID values found in the specified domains.

The mechanics of the matching algorithms used are internal to the Patient Demographics Supplier Actor and are outside of the scope of this framework.

The Patient Demographics Supplier Actor shall respond to the query request as described by the following 3 cases:

Case 1: The Patient Demographics Supplier Actor finds (in the patient information source associated with *MSH-5-Receiving Application*) at least one patient record matching the criteria sent in *QPD-3-Demographics Fields*. No patient identifier domains are requested in *QPD-8-What Domains Returned*.

AA (application accept) is returned in MSA-1.

OK (data found, no errors) is returned in QAK-2.

One PID-PV1 segment group (*i.e.*, one PID segment and one PV1 segment, plus any segments associated with them in the message syntax shown in Table 3.22-6) is returned from the patient information source for each patient record found. If the Patient Demographics Supplier Actor returns data for multiple patients, it shall return these data in successive occurrences of the PID-PV1 segment group.

Within each PID segment, field *PID-3-Patient Identifier List* contains one or more identifiers from the Patient ID Domain associated with the patient data source identified by *MSH-5-Receiving Facility*.

If an incremental number of records is specified in *RCP-2-Quantity Limited Request*, and the number of records found exceeds that incremental number, the Supplier returns only the incremental number of records, followed by a DSC segment containing a uniquely valued Continuation Pointer.

Case 2: The Patient Demographics Supplier Actor finds (in the patient information source associated with *MSH-5-Receiving Application*) at least one patient record matching the criteria sent in *QPD-3-Demographics Fields*. One or more patient identifier domains are requested in *QPD-8-What Domains Returned*; the Supplier recognizes all the requested domains.

AA (application accept) is returned in MSA-1.

OK (data found, no errors) is returned in QAK-2.

One PID-PV1 segment group (*i.e.*, one PID and one PV1 segment plus any segments associated with them in the message syntax shown in Table 3.22-6) is returned for each matching patient record found. If the Patient Demographics Supplier Actor returns data for multiple patients, it shall return these data in successive occurrences of the PID segment group.

Within each PID segment, field *PID-3-Patient Identifier List* contains, in successive occurrences delimited by the repetition separator, the identifiers from all the Patient ID Domains requested in QPD-8. In each occurrence of PID-3, component 4 contains the assigning authority value for one Patient ID Domain, and component 1 contains the Patient ID value in that domain (or is left blank if an identifier does not exist in that domain).

If an incremental number of records is specified in *RCP-2-Quantity Limited Request*, and the number of records to be sent exceeds that incremental number, the Supplier returns only the incremental number of records, followed by a DSC segment containing a uniquely valued Continuation Pointer.

Case 3: The Patient Demographics Supplier Actor does not recognize one or more of the domains in *QPD-8-What Domains Returned*.

AE (application error) is returned in MSA-1 and in QAK-2.

For each domain that was not recognized, an ERR segment is returned in which the components of *ERR-2-Error Location* are valued as follows.

COMP #	COMPONENT NAME	VALUE
1	Segment ID	**QPD**
2	Sequence	**1**
3	Field Position	**8**
4	Field Repetition	*(see below)*
5	Component Number	*(empty)*

COMP #	COMPONENT NAME	VALUE
6	Subcomponent Number	*(empty)*

ERR-2.4-Field Repetition identifies the ordinal occurrence of QPD-8 that contained the unrecognized domain. As specified by HL7, *ERR-2.5-Component Number* and *ERR-2.6-Subcomponent Number* are not valued because we are referring to the entire field QPD-8.

ERR-3-HL7 Error Code is populated with the error condition code **204** (unknown key identifier). Together with the values in ERR-2, this signifies that the Patient Demographics Supplier Actor did not recognize the domain for *QPD-8-What Domains Returned*.

3.22.4.2.3 Expected Actions

The Patient Demographics Consumer will use the demographic information provided by the Patient Demographics Supplier to perform the functions for which it requested the information, *e.g.*, providing a pick list to the user.

The Consumer will return an original mode acknowledgement as described in the HL7 Standard, populating field *MSH-14-Continuation Pointer* as follows.

If the Supplier returned a value in *DSC-1-Continuation Pointer*, indicating that there are more matching patients remaining to be sent than could be sent in a single increment, ***and*** if the Consumer wishes to receive the next increment, the Consumer shall populate MSH-14 with the value it received in DSC-1. The Supplier will then send the next increment of matching records, and will again populate DSC-1 (with a new unique value) if there are yet more records to be sent.

If the Supplier did not return a value in *DSC-1-Continuation Pointer*, or if the Supplier did return a value in DSC-1 but the Consumer does not wish to receive the next increment, the Consumer shall return an empty value in MSH-14.

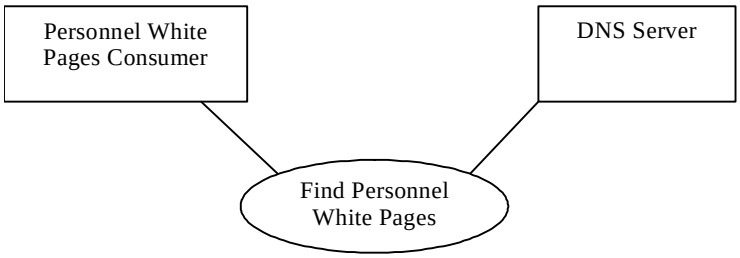
3.23 Find Personnel White Pages

This section corresponds to Transaction ITI-23 of the IHE Technical Framework. Transaction ITI-23 is used by the Personnel White Pages Consumer and the DNS Server Actors.

3.23.1 Scope

This Transaction is used to locate the Personnel White Pages directory.

3.23.2 Use Case Roles



Actor: Personnel White Pages Consumer

Role: Requests Locating information for the Personnel White Pages Directory

Actor: DNS Server

Role: Provides locating information about the Personnel White Pages Directory

3.23.3 Referenced Standard

IETF: RFC-2181 Clarifications to the DNS Specification

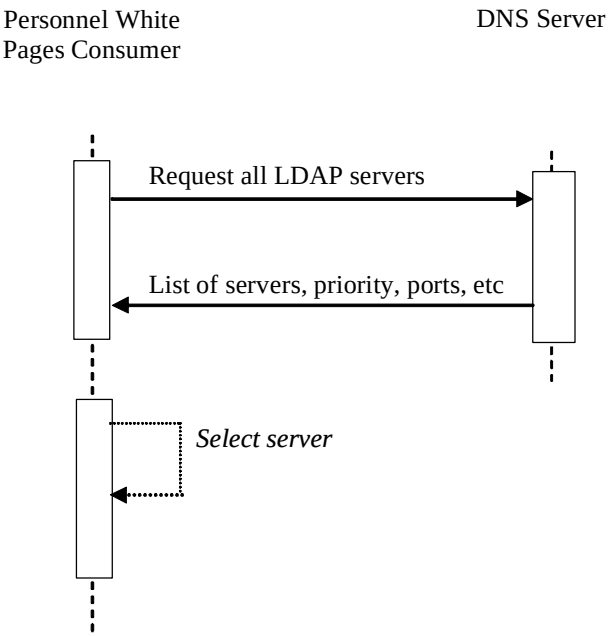
RFC-2219 Use of DNS Aliases for Network Services

RFC-2782 A DNS RR for specifying the location of services (DNS SRV)

DICOM: DICOM Supplement 67 – Configuration Management, January 14, 2004.

Note: Normative RFC's are frequently updated by issuance of subsequent RFC's. The original older RFC is not modified to include references to the newer RFC. This profile lists the applicable RFC's in effect at the time of publication. Subsequent updates and clarifications to these RFC's should also be applied.

3.23.4 Interaction Diagram



3.23.4.1 Request all LDAP servers

The RFC-2782 DNS RR is used for specifying the location of services (DNS SRV). It specifies a mechanism for requesting the names and rudimentary descriptions for machines that provide network services. The DNS client requests the descriptions for all machines that are registered as offering a particular service name. In this case the service name requested will be "_ldap._tcp". The DNS server may respond with multiple names for a single request.

3.23.4.1.1 Trigger Events

This transaction is used by the Personnel White Pages Consumer prior to any access to the Personnel White Pages Directory.

3.23.4.1.2 Message Semantics

The Personnel White Pages Consumer shall request a list of all the LDAP servers available. The Personnel White Pages Consumer shall use the priority, capacity, and location information provided by DNS as part of the server selection process. (RFC-2782 recommends the proper use of these parameters).

Note:

Multiple LDAP servers providing access to a common replicated LDAP database is a commonly supported configuration. This permits LDAP servers to be located where appropriate for best performance and fault tolerance. The DNS server response information provides guidance for selecting the most appropriate server.

There may also be multiple LDAP servers providing different databases. In this situation the client may have to examine several servers to find the one that supports the Personnel White Pages Directory (See ITI TF-2:3.24.4.1.2.2).

The client may have a mechanism for manual default selection of the LDAP server to be used if the DNS server does not provide an LDAP server location.

3.23.4.1.3 Expected Actions

The DNS Server shall return all known LDAP servers in accordance with RFC-2782.

3.24 Query Personnel White Pages

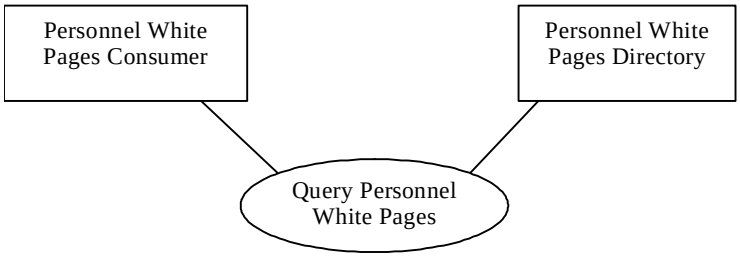
This section corresponds to Transaction ITI-24 of the IHE Technical Framework. Transaction ITI-24 is used by the Personnel White Pages Consumer and the Personnel White Pages Directory Actors.

3.24.1 Scope

This Transaction is used to retrieve information from the Personnel White Pages directory.

The RFC-2251 “Lightweight Directory Access Protocol (v3)” specifies a mechanism for making queries of a database corresponding to an LDAP schema. The LDAP client can compose requests in the LDAP query language, and the LDAP server will respond with the results for a single request.

3.24.2 Use Case Roles



Actor: Personnel White Pages Consumer

Role: Requests information about a human workforce member(s)

Actor: Personnel White Pages Directory

Role: Provides information about one or more human workforce member

3.24.3 Referenced Standard

IETF: RFC-2181 Clarifications to the DNS Specification

RFC 1766 Tags for the Identification of Languages

RFC 2251 - Lightweight Directory Access Protocol (v3)

RFC 2252 - Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions

RFC 2253 - Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names

RFC 2256 - A Summary of the X.500(96) User Schema for use with LDAPv3

RFC 2798 - Definition of the inetOrgPerson LDAP Object Class

RFC 2829 Authentication Methods for LDAP

RFC 2830 LDAPv3: Extension for Transport Layer Security

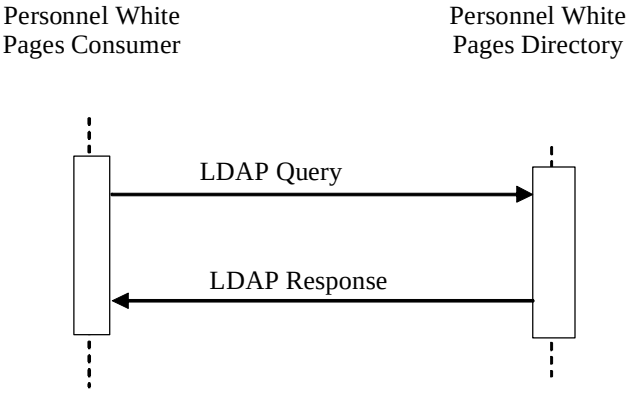
ISO: ISO/TS 17090 directory standard for healthcare identity management

CRU: Projet de schémas d'annuaires et de schémas de registres de resources numériques interopérables pour les administrations Document technique – v1, novembre 2002

ASTM: E.123: Notation for national and international telephone numbers

HL7: HL7 Version 2.5, Chapter 2 – Control

3.24.4 Interaction Diagram



3.24.5 LDAP Query/Response

The Personnel White Pages Consumer may make a wide variety of queries and cascaded queries using LDAP. The Personnel White Pages Consumer and Personnel White Pages Directory shall support the data model described here.

A commonly supported configuration type has multiple LDAP servers providing access to a common replicated LDAP database. This permits LDAP servers to be located where appropriate for best performance and fault tolerance. The replication rules chosen for the LDAP servers affect the visible data consistency. LDAP permits inconsistent views of the database during updates and replications. This inconsistency may result in a consumer receiving the person's previous demographics or contact information. This should not be a problem for our use-cases as none of them are life critical.

3.24.5.1 Trigger Events

Personnel White Pages Consumer requires some Personnel White Pages information on one or more human workforce members.

3.24.5.2 Message Semantics

The transaction uses standard LDAP v3 query/response mechanisms.

3.24.5.2.1 User Authentication

Some of the attributes to be retrieved using this transaction may be considered sensitive to the healthcare personnel. It is the responsibility of the Personnel White Pages Directory to enforce these protections. To protect records and/or attributes, the Personnel White Pages Consumer may be called upon to provide user credentials.

Anonymous authentication shall be implemented on Personnel White Pages Directory and is optional for Personnel White Pages Consumer. Anonymous authentication shall be implemented as described in LDAP v3 section 4.2 Bind Operation.

Simple Authentication shall be implemented on the Personnel White Pages Directory and is optional for the Personnel White Pages Consumer. Simple authentication shall be implemented as described in LDAP v3 section 4.2 Bind Operation. This authentication type is not recommended for use over networks that are not otherwise secured as the username and password are transferred in the clear. The use of SSL-Simple Authentication is a better choice.

SSL-Simple Authentication shall be implemented on the Personnel White Pages Directory and is optional for the Personnel White Pages Consumer. SSL-Simple Authentication is not defined in any normative text, but is consistently implemented and often referred to as “ldaps”. The PWP Consumer shall connect to port 636 using SSL against the PWP Directory Certificate. The LDAP v3 conversation then continues with Simple Authentication as defined in LDAP v3 section 4.2 Bind Operation.

PWP specifies read operations on personnel demographics. The use of bi-directional TLS authentication, such as that defined in ATNA Profile, is not necessary as this profile does not provide access to Protected Health Information (PHI). The use of SSL to cover the authentication and query process is sufficient in this Profile.

3.24.5.2.2 Base DN Discovery

The Personnel White Pages represents a branch within the “LDAP” directory. Branches in LDAP are defined by a “Base DN”. The list of Base DNs that are provided by a LDAP directory can be found by doing a LDAP Query with a NULL (i.e. “”) Base DN, and ObjectClass=”DN”. The Personnel White Pages Directory shall contain a person object with the cn=”IHE-ITI-PWP”. The Personnel White Pages Consumer may thus search through the list of Base DNs that the LDAP Directory contains for this cn object. The Personnel White Pages Directory identified in this way

shall contain person/inetOrgPerson objects that conform to the Query Personnel White Pages Directory Transaction.

Note: The first LDAP server that yields a result on the search for IHE-ITI-PWP can be used. There is no need to search further.

3.24.5.2.3 Query Encoding

Note that the LDAP transactions utilize UTF-8 encoding unless otherwise noted. The schema shown here is the commonly used schema found in X.500 Schema for LDAP and inetOrgPerson. Extensions beyond this schema are not recommended. The base schema must be preserved to ensure interoperability. Schema extensions shall not introduce attributes that duplicate the meaning of any attribute specified in this Profile.

These attributes are multi-valued unless explicitly defined as single-valued. At this time there is no universally implemented method to distinguish the purpose for any of the instances in a multi-valued attribute. The IHE recommends that the first entry contain the preferred value, and that applications use the first entry whenever a single value must be selected.

The following table shows the attributes found in Person (OrganizationalPerson and ResidentialPerson) as defined in RFC 2256 and inetOrgPerson as defined in RFC 2798. The first three columns contain the definitions from the standards for reference. Within the table the fourth column is the IHE recommendation for use with further discussion found in the fifth column.

KEY for IHE REQ Column:

R – The Personnel White Pages Directory shall contain valid values for these attributes. These values are critical to Healthcare workflow.

R2– The Personnel White Pages Directory shall contain valid values for these attributes if the value is available. These attributes are sufficiently useful that the provider should utilize it in the defined way. Personnel White Pages Consumers should expect that the information in these attributes are valid, but shall be robust to empty values.

O – The Personnel White Paged Directory may contain values for these optional attributes. The IHE has identified sufficiently useful purpose or defined an interoperable way to use the value. The IHE may profile these values in future profiles.

D – Although these attributes are defined in inetOrgPerson/Person, their use is discouraged. This is typically due to the attribute being obsolete, poorly implemented, or not available for query.

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
aliasedObjectName	RFC 2256	• Alias Object Name • Optional • The aliasedObjectName attribute is used by the directory service if the entry containing this attribute is an alias.	O	
Audio	RFC 2798	• Audio • Optional • Not well defined	D	The audio format defined is obsolete.
businessCategory	RFC 2798	• Business Category • Optional • describes the kind of business performed by an organization	D	Not well defined
CarLicense	RFC 2798	• Vehicle license or registration plate • Optional • Used to record the values of the license or registration plate associated with an individual • (e.g. 6ABC246)	O	
Cn	RFC 2256	• Common Name • Required • This is the X.500 commonName attribute, which contains a name of an object. If the user is a person, it is typically the person's full name. • (e.g. Barbara Jensen)	R	See 3.24.4.1.2.3.1 Use of language tag and HL7 Name Data Type (XPN)
departmentNumber	RFC 2798	• Department Number • Optional • Identifies a department within an organization. This can be numeric or alphanumeric • (e.g. Radiology)	O	
Description	RFC 2798	• Description • Optional • This attribute contains a human-readable description of the object.	D	
destinationIndicator	RFC 2256	• Destination Indicator • Optional • This attribute is used for the telegram service	D	Originally defined as part of telegram addressing.

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
displayName	RFC 2798	• Display Name • Optional • Singular • When displaying a person's name, especially within a one-line summary list, it is useful to be able to identify a name to be used. Since other attribute types such as 'cn' are multivalued, an additional attribute type is needed. Display name is defined for this purpose. • (e.g. Babs Jensen)	R	
employeeNumber	RFC 2798	• Employee Number • Optional • Singular • Numeric or alphanumeric identifier assigned to a person, typically based on order of hire or association with an organization. • (e.g. 42)	O	
employeeType	RFC 2798	• Employee Type • Optional • Used to identify the employer to employee relationship. Typical values used will be "Contractor", "Employee", "Intern", "Temp", "External", and "Unknown" but any value may be used. • (e.g. External)	O	
facsimileTelephoneNumber	RFC 2256	• FAX Number • Optional • A value of this attribute is a telephone number for a facsimile terminal (and, optionally, its parameters). • (e.g. +1 408 555 1992)	R2	See 3.24.4.1.2.3.3 Phone Numbers
GivenName	RFC 2798	• Name • Optional • The givenName attribute is used to hold the part of a person's name which is not their surname nor middle name. • (e.g. Barbara)	R2	
homePhone	RFC 2798	• Home Phone • Optional • (e.g. +1 408 555 1862)	O	

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
homePostalAddress	RFC 2798	• Home Postal Address • Optional • This attribute contains a home address used by a Postal Service to perform services for the object.	O	
Initials	RFC 2798	• Initials • Optional • The initials attribute contains the initials of some or all of an individuals names, but not the surname(s). • (e.g. BJJ)	R2	
internationaliSDNNumber	RFC 2798	• International ISDN Number • Optional	D	
jpegPhoto	RFC 2798	• JPEG Photograph • Optional • Used to store one or more images of a person using the JPEG File Interchange Format	O	
L	RFC 2256	• Locality Name • Optional • This is the X.500 localityName attribute, which contains the name of a locality, such as a city, county or other geographic region.	O	
labeledURI	RFC 2798	• URI • Optional • (e.g. http://www.ihe.net IHE Home)	O	
Mail	RFC 2798	• E-Mail Address • Optional • User's e-mail address in RFC 822 compliant form • (e.g. bjensen@siroe.com)	R2	
manager	RFC 2798	• Manager • Optional • Distinguished Name of the Manager	O	In Healthcare the manager of an individual is not clear. The manager attribute does not include enough information to determine the type of manager indicated.

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
Mobile	RFC 2798	• Mobile/cellular phone number • Optional • A value of this attribute is a telephone number complying with ITU Recommendation E.123. • (e.g. +1 408 555 1941)	R2	This attribute should contain only business use mobile phone numbers. See 3.24.4.1.2.3.3 Phone Numbers
O	RFC 2798	• Organization • Optional • Highest-level organization name, e.g., a company name, to which ou attribute entries belong. • (e.g. Saint-ihe-hospital.local)	R2	
objectClass	RFC 2256	• Object Class • Required • The values of the objectClass attribute describe the kind of object which an entry represents. The objectClass attribute is present in every entry, with at least two values. One of the values is either "top" or "alias". • (e.g. top, person, organizationalPerson, inetOrgPerson)	R	
ou	RFC 2256	• Organizational Unit Name • Optional • This is the X.500 organizationalUnitName attribute, which contains the name of an organizational unit. • (e.g. Radiologists)	R2	
pager	RFC 2798	• Pager phone number • Optional • A value of this attribute is a telephone number complying with ITU Recommendation E.123.	R2	This attribute should contain only business use mobile phone numbers. See 3.24.4.1.2.3.3 Phone Numbers
photo	RFC 2798	• Photo • Optional • Photo attribute values are encoded in G3 fax format with an ASN.1 wrapper.	D	The format is too cumbersome. See jpegPhoto.

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
physicalDeliveryOfficeName	RFC 2256	• Post Office Name • Optional • This attribute contains the name that a Postal Service uses to identify a post office.	R2	
postalAddress	RFC 2256	• Postal Address • Optional • This attribute contains an address used by a Postal Service to perform services for the object.	R2	
postalCode	RFC 2256	• Postal Code • Optional • This attribute contains a code used by a Postal Service to identify a postal service zone, such as a US ZIP code	R2	
postOfficeBox	RFC 2256	• Post Office Box • Optional • This attribute contains the number that a Postal Service uses when a customer arranges to receive mail at a box on premises of the Postal Service.	R2	
preferredDeliveryMethod	RFC 2798	• Delivery Method • Optional • Singular • Coded value (delivery-value) • (e.g. any, physical, telephone)	O	
preferredLanguage	RFC 2798	• Preferred Language • Optional • Singular • Preferred written or spoken language for a person. Values for this attribute type MUST conform to the definition of the Accept-Language header field defined in [RFC2068] with one exception: the sequence "Accept-Language" ":" should be omitted. • The following example indicates that this person prefers French, prefers British English 80%, and general English 70%. (e.g. fr, en-gb;q=0.8, en;q=0.7)	R2	

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
registeredAddress	RFC 2256	• Registered Address • Optional • A postal address suitable for reception of expedited documents, where it is necessary to have the recipient accept delivery.	O	
roomNumber	RFC 2798	• Room Number • Optional	O	
secretary	RFC 2798	• Secretary • Optional • Distinguished name of the secretary	O	
seeAlso	RFC 2798	• See Also references • Optional • Distinguished name of other interesting Objects	D	
sn	RFC 2256	• Surname • Required • This is the X.500 surname attribute, which contains the family name of a person • (e.g. Jensen)	R	
st	RFC 2256	• State or Province • Optional • This is the X.500 stateOrProvinceName attribute, which contains the full name of a state or province	R2	
street	RFC 2256	• Street Address • Optional • This is the X.500 streetAddress attribute, which contains the physical address of the object to which the entry corresponds, such as an address for package delivery.	R2	
telephoneNumber	RFC 2256	• Telephone number • Optional • A value of this attribute is a telephone number complying with ITU Recommendation E.123.	R2	See 3.24.4.1.2.3.3 Phone Numbers
teletexTerminalIdentifier	RFC 2798	• Teletex Terminal Identifier • Optional	D	
telexNumber	RFC 2798	• Telex Number • Optional	D	

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
title	RFC 2256	• Title • Optional • This attribute contains the title, such as "Vice President", of a person in their organizational context. The "personalTitle" attribute would be used for a person's title independent of their job function. • (e.g. manager, product development)	R2	
uid	RFC 2798	• User ID • Required • The user ID use for system login. • (e.g. bjensen)	R	See 3.24.4.1.2.3.2 Use of uid
userCertificate	RFC 2798	• User Identity Certificate • Optional • This attribute is to be stored and requested in the binary form, as 'userCertificate;binary'.	D	The PKCS12 format includes the private key and shall not be publicly available.
userPassword	RFC 2256	• User password • Optional • Passwords are stored using an Octet String syntax and are not encrypted. Transfer of cleartext passwords are strongly discouraged where the underlying transport service cannot guarantee confidentiality and may result in disclosure of the password to unauthorized parties.	D	Generally Not Accessible
userPKCS12	RFC 2798	• User PKCS #12 • Optional • PKCS #12 [PKCS12] provides a format for exchange of personal identity information. When such information is stored in a directory service, the userPKCS12 attribute should be used. This attribute is to be stored and requested in binary form, as 'userPKCS12;binary'. The attribute values are PFX PDUs stored as binary data.	D	The PKCS12 format includes the private key and shall not be publicly available.

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
userSMIMECertificate	RFC 2798	• User S/MIME Certificate • Optional • A PKCS#7 [RFC2315] SignedData, where the content that is signed is ignored by consumers of userSMIMECertificate values. It is recommended that values have a `contentType' of data with an absent `content' field. Values of this attribute contain a person's entire certificate chain and an smimeCapabilities field [RFC2633] that at a minimum describes their SMIME algorithm capabilities. Values for this attribute are to be stored and requested in binary form, as 'userSMIMECertificate;binary'. If available, this attribute is preferred over the userCertificate attribute for S/MIME applications.	O	
x121Address	RFC 2256	• Address for X.121 • Optional	D	
X500uniqueIdentifier	RFC 2798	• Unique identifier • Required • The x500UniqueIdentifier attribute is used to distinguish between objects when a distinguished name has been reused. This is a different attribute type from both the "uid" and "uniqueIdentifier" types.	R	

3.24.5.2.3.1 Use of language tag and HL7 Name Data Type (XCN)

Many people have different variations of their name to be used depending on the context and language. This is easily supported in LDAP through the use of the language tag as documented in RFC 1766. This language tag can be applied to any attribute but is most useful on names.

HL7 has a well-defined format for encoding names (HL7 XCN). LDAP ‘name’ attributes marked with a language tag of “lang-x-ihe” shall be encoded using the HL7 XCN Data Type. UTF-8 shall be used for any characters outside ASCII.

Example use of the language tag:

```
objectclass: Top
objectclass: person
objectclass: organizationalPerson
```

```
objectclass: inetOrgPerson
dn: cn=Wang XiaoDong, ou=Radiologists, o=Saint-ihe-hospital.local
cn: Wang XiaoDong
cn: XiaoDong, Wang, Florida Department of Health:123456789
cn;lang-cn: 王 小東
cn;lang-x-ihe: Wang^XiaoDong^^^^^^A~王^小東^^^^^^
sn: Wang
givenname: XiaoDong
givenname;lang-cn: 小東
sn;lang-cn: 王
ou: People
uid: XiaoDong
title: Sample HL7 person
mail: Wang.XiaoDong@foo.bar.com
telephonenumber: 555-555-5678
```

3.24.5.2.3.2 Use of uid.

The uid attribute is a multi-valued attribute that is intended to be used for User ID. It is likely that one of the values for uid will be the enterprise User ID. Enterprises that implement the PWP Profile shall implement the following values for the uid attribute:

1. If an enterprise has implemented both IHE ITI EUA and PWP profiles, one of the uid attributes shall contain the IHE ITI EUA user identity in <user>@<realm> format.
2. If an enterprise has implemented a UPIN, one of the uid attributes shall contain the UPIN value in the format <UPIN>@UPIN. Where a UPIN is the Universal Physician Identification Number as assigned by the assigning authority in which the facility operates (e.g. CMS in the USA).

3.24.5.2.3.3 Phone Numbers

Phone numbers shall be represented in the PWP Directory using E.123 notation. E.123 is a notation for national and international telephone numbers. Recommendation E.123 defines a standard way to write telephone numbers, e-mail addresses, and web addresses. It recommends the following formats (when dialing the area code is optional for local calling):

Telephone number:

National notation (042) 123 4567

International notation +31 42 123 4567

E.123 also recommends that a hypen (-), space ( ), or period (.) be used to visually separate groups of numbers. The parentheses are used to indicate digits that are sometimes not dialed. A slash (/) is used to indicate alternate numbers. This information is important if you want to make sure people know how to dial a phone number in a specific country.

The use of National notation and International notation will be a local PWP Directory policy. PWP Consumers shall expect to receive both notations.

3.24.5.2.4 Expected Actions

The Personnel White Pages Directory shall provide the appropriate response to the indicated query given LDAP query rules, local access control policy, and the current information it the directory.

Note: Any attribute is valid to query on, the results of the query may be quick or may take a long time to complete. Each Personnel White Pages Directory will be optimized differently based on architecture and configuration. We expect that the following attributes will be query keys more often than others (cn, displayname, objectclass, sn, uid, givenName, initials, mail, o, ou, and employeeNumber).

Directory shall support Anonymous, Simple, and SSL-Simple Authentications.

Appendix A: Web Service Definition for Retrieve Specific Information for Display and Retrieve Document for Display Transaction

The following is an example WSDL definition of web services used in Transactions ITI-11 and ITI-12. This code is provided as an example and is not intended to replace the formal specification of Transactions ITI-11 and ITI-12 in Volume 2. Also, the definitions of summaryRequestType, listRequestType and contentType shall correspond to the capabilities of the Information Source Actor.

```
<?xml version="1.0" encoding="utf8"?>

<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:s="http://www.w3.org/2001/XMLSchema"
    xmlns:s0="http://rsna.org/ihe/IHERetrieveForDisplay"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    targetNamespace="http://rsna.org/ihe/IHERetrieveForDisplay"
    xmlns="http://schemas.xmlsoap.org/wsdl/">

  <!-- Defines the types available for the parameters -->
  <!-- May also include the return type defintiions -->
  <types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://rsna.org/ihe/IHERetrieveForDisplay">
      <!-- Add any items that control the returned values list or type here -->
      <!-- Add or remove items in the actual supplied WSDL to show the available types. -->
      <s:simpleType name="summaryRequestType">
        <s:restriction base="s:string">
           <s:enumeration value="SUMMARY" />
           <s:enumeration value="SUMMARY-RADIOLOGY" />
           <s:enumeration value="SUMMARY-CARDIOLOGY" />
           <s:enumeration value="SUMMARY-LABORATORY" />
           <s:enumeration value="SUMMARY-SURGERY" />
           <s:enumeration value="SUMMARY-EMERGENCY" />
           <s:enumeration value="SUMMARY-DISCHARGE" />
           <s:enumeration value="SUMMARY-ICU" />
        </s:restriction>
      </s:simpleType>

      <s:simpleType name="listRequestType">
        <s:restriction base="s:string">
           <s:enumeration value="LIST-ALLERGIES" />
           <s:enumeration value="LIST-MEDS" />
        </s:restriction>
      </s:simpleType>

      <!-- Please list all content types available, and remove those not available. -->
      <s:simpleType name="contentType">
        <s:restriction base="s:string">
           <s:enumeration value="text/html" />
        </s:restriction>
      </s:simpleType>

      <!-- Indicates that this item is a returned rows restriction -->
      <s:simpleType name="ReturnedResultCount" type="s:positiveInteger" />
```

```
      <!-- Please use the string "Search" as a prefix for all search criteria, and list below -->
      <!-- Indicates that this item is a search string -->
      <s:simpleType name="SearchString" type="s:string" />


  </s:schema>
</types>

<message name="RetrieveSummaryInfoHttpGetIn">
  <!-- Add other parameters here if they are available, using types defined above. -->
  <part name="requestType" type="summaryRequestType" />
  <part name="patientID" type="SearchString" />
  <part name="lowerDateTime" type="s:dateTime" />
  <part name="upperDateTime" type="s:dateTime" />
  <part name="mostRecentResults" type="ReturnedResultCount" />
</message>

<message name="RetrieveSummaryInfoHttpGetOut">
  <!-- If a complex type is defined for the return value, then it is suggested that -->
  <!-- it be used here instead of s0:string.  If a complex type is allowed as one -->
  <!-- of the options, but an arbitrarily formatted string is also allowed, then create -->
  <!-- a union type here that allows either option. -->
  <part name="Body" element="s0:string" />
</message>

<message name="RetrieveListInfoHttpGetIn">
  <!-- Add other parameters here if they are available, using types defined above. -->
  <part name="requestType" type="listRequestType" />
  <part name="patientID" type="SearchString" />
</message>

<message name="RetrieveListInfoHttpGetOut">
  <!-- If a complex type is defined for the return value, then it is suggested that -->
  <!-- it be used here instead of s0:string.  If a complex type is allowed as one -->
  <!-- of the options, but an arbitrarily formatted string is also allowed, then create -->
  <!-- a union type here that allows either option. -->
  <part name="Body" element="s0:string" />
</message>
<message name="RetrieveDocumentHttpGetIn">
  <!-- Add other parameters here if they are available, using types defined above. -->

  <!-- It is recommended that one of the sub-types of SearchUID is chosen here -->
  <!-- Especially if SearchStudyUID is allowed, then the display client can know that -->
  <!-- it is permissable to use a dicom uid here -->
  <part name="documentUID" type="SearchString" />
  <part name="contentType" type="contentType" />
</message>

<message name="RetrieveDocumentHttpGetOut">
  <!-- If a complex type is defined for the return value, then it is suggested that -->
  <!-- it be used here instead of s:string.  If a complex type is allowed as one -->
  <!-- of the options, but an arbitrarily formatted string is also allowed, then create -->
  <!-- a union type here that allows either option. -->
  <part name="Body" element="s:string" />
</message>

<portType name="IHERetrieveForDisplayHttpGet">
  <operation name="RetrieveSummaryInfo">
    <input message="s0:RetrieveSummaryInfoHttpGetIn" />
    <output message="s0:RetrieveSummaryInfoHttpGetOut" />
  </operation>
     <operation name="RetrieveListInfo">
    <input message="s0:RetrieveListInfoHttpGetIn" />
```

```
      <output message="s0:RetrieveListInfoHttpGetOut" />
    </operation>
    <operation name="RetrieveDocument">
      <input message="s0:RetrieveDocumentHttpGetIn" />
      <output message="s0:RetrieveDocumentHttpGetOut" />
    </operation>
  </portType>

  <binding name="IHERetrieveForDisplayHttpGet" type="s0:IHERetrieveForDisplayHttpGet">
    <http:binding verb="GET" />
    <operation name="RetrieveSummaryInfo">
      <http:operation location="/IHERetrieveSummaryInfo" />
      <input>
        <http:urlEncoded />
      </input>

      <output>
        <mime:content type="text/html" />
      </output>
    </operation>

    <operation name="RetrieveListInfo">
      <http:operation location="/IHERetrieveListInfo" />
      <input>
        <http:urlEncoded />
      </input>

      <output>
        <mime:content type="text/html" />
      </output>
    </operation>

    <operation name="RetrieveDocument">
      <http:operation location="/IHERetrieveDocument" />
      <input>
        <http:urlEncoded />
      </input>

      <!-- The type of the output should be restricted on a per-server basis to the types -->
      <!-- actually provided. -->
      <output>
        <mime:content type="text/html" />
        <mime:content type="application/x-hl7-cda-level-one+xml" />
        <mime:content type="application/pdf" />
        <mime:content type="image/jpeg" />
      </output>
    </operation>
  </binding>

  <!-- Bind the actual service here -->
  <service name="IHERetrieveForDisplay">
    <port name="IHERetrieveForDisplayHttpGet" binding="s0:IHERetrieveForDisplayHttpGet">
      <http:address location="http://localhost/" />
    </port>
</service>
<?xml version="1.0" encoding="utf8"?>

<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:s="http://www.w3.org/2001/XMLSchema"
    xmlns:s0="http://rsna.org/ihe/IHERetrieveForDisplay"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    targetNamespace="http://rsna.org/ihe/IHERetrieveForDisplay"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
```

```
  <!-- Defines the types available for the parameters -->
  <!-- May also include the return type defintiions -->
  <types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://rsna.org/ihe/IHERetrieveForDisplay">
      <!-- Add any items that control the returned values list or type here -->
      <!-- Add or remove items in the actual supplied WSDL to show the available types. -->
      <s:simpleType name="summaryRequestType">
        <s:restriction base="s:string">
           <s:enumeration value="SUMMARY" />
           <s:enumeration value="SUMMARY-RADIOLOGY" />
           <s:enumeration value="SUMMARY-CARDIOLOGY" />
           <s:enumeration value="SUMMARY-LABORATORY" />
           <s:enumeration value="SUMMARY-SURGERY" />
           <s:enumeration value="SUMMARY-EMERGENCY" />
           <s:enumeration value="SUMMARY-DISCHARGE" />
           <s:enumeration value="SUMMARY-ICU" />
           <s:enumeration value="SUMMARY-RX" />
        </s:restriction>
      </s:simpleType>

      <s:simpleType name="listRequestType">
        <s:restriction base="s:string">
           <s:enumeration value="LIST-ALLERGIES" />
           <s:enumeration value="LIST-MEDS" />
        </s:restriction>
      </s:simpleType>

      <!-- Please list all content types available, and remove those not available. -->
      <s:simpleType name="contentType">
        <s:restriction base="s:string">
           <s:enumeration value="text/html" />
        </s:restriction>
      </s:simpleType>

      <!-- Indicates that this item is a returned rows restriction -->
      <s:simpleType name="ReturnedResultCount" type="s:positiveInteger" />

      <!-- Please use the string "Search" as a prefix for all search criteria, and list below -->
      <!-- Indicates that this item is a search string -->
      <s:simpleType name="SearchString" type="s:string" />


    </s:schema>
  </types>

  <message name="RetrieveSummaryInfoHttpGetIn">
    <!-- Add other parameters here if they are available, using types defined above. -->
    <part name="requestType" type="summaryRequestType" />
    <part name="patientID" type="SearchString" />
    <part name="lowerDateTime" type="s:dateTime" />
    <part name="upperDateTime" type="s:dateTime" />
    <part name="mostRecentResults" type="ReturnedResultCount" />
  </message>

  <message name="RetrieveSummaryInfoHttpGetOut">
    <!-- If a complex type is defined for the return value, then it is suggested that -->
    <!-- it be used here instead of s0:string. If a complex type is allowed as one -->
    <!-- of the options, but an arbitrarily formatted string is also allowed, then create -->
    <!-- a union type here that allows either option. -->
    <part name="Body" element="s0:string" />
  </message>
```

```
<message name="RetrieveListInfoHttpGetIn">
  <!-- Add other parameters here if they are available, using types defined above. -->
  <part name="requestType" type="listRequestType" />
  <part name="patientID" type="SearchString" />
</message>

<message name="RetrieveListInfoHttpGetOut">
  <!-- If a complex type is defined for the return value, then it is suggested that -->
  <!-- it be used here instead of s0:string. If a complex type is allowed as one -->
  <!-- of the options, but an arbitrarily formatted string is also allowed, then create -->
  <!-- a union type here that allows either option. -->
  <part name="Body" element="s0:string" />
</message>
<message name="RetrieveDocumentHttpGetIn">
  <!-- Add other parameters here if they are available, using types defined above. -->

  <!-- It is recommended that one of the sub-types of SearchUID is chosen here -->
  <!-- Especially if SearchStudyUID is allowed, then the display client can know that -->
  <!-- it is permissable to use a dicom uid here -->
  <part name="documentUID" type="SearchString" />
  <part name="contentType" type="contentType" />
</message>

<message name="RetrieveDocumentHttpGetOut">
  <!-- If a complex type is defined for the return value, then it is suggested that -->
  <!-- it be used here instead of s:string. If a complex type is allowed as one -->
  <!-- of the options, but an arbitrarily formatted string is also allowed, then create -->
  <!-- a union type here that allows either option. -->
  <part name="Body" element="s:string" />
</message>

<portType name="IHERetrieveForDisplayHttpGet">
  <operation name="RetrieveSummaryInfo">
    <input message="s0:RetrieveSummaryInfoHttpGetIn" />
    <output message="s0:RetrieveSummaryInfoHttpGetOut" />
  </operation>
     <operation name="RetrieveListInfo">
    <input message="s0:RetrieveListInfoHttpGetIn" />
    <output message="s0:RetrieveListInfoHttpGetOut" />
  </operation>
  <operation name="RetrieveDocument">
    <input message="s0:RetrieveDocumentHttpGetIn" />
    <output message="s0:RetrieveDocumentHttpGetOut" />
  </operation>
</portType>

<binding name="IHERetrieveForDisplayHttpGet" type="s0:IHERetrieveForDisplayHttpGet">
  <http:binding verb="GET" />
  <operation name="RetrieveSummaryInfo">
    <http:operation location="/IHERetrieveSummaryInfo" />
    <input>
      <http:urlEncoded />
    </input>

    <output>
      <mime:content type="text/html" />
    </output>
  </operation>

  <operation name="RetrieveListInfo">
    <http:operation location="/IHERetrieveListInfo" />
    <input>
      <http:urlEncoded />
```

```
      </input>

      <output>
        <mime:content type="text/html" />
      </output>
    </operation>

    <operation name="RetrieveDocument">
      <http:operation location="/IHERetrieveDocument" />
      <input>
        <http:urlEncoded />
      </input>

      <!-- The type of the output should be restricted on a per-server basis to the types -->
      <!-- actually provided. -->
      <output>
        <mime:content type="text/html" />
        <mime:content type="application/x-hl7-cda-level-one+xml" />
        <mime:content type="application/pdf" />
        <mime:content type="image/jpeg" />
      </output>
    </operation>
  </binding>

  <!-- Bind the actual service here -->
  <service name="IHERetrieveForDisplay">
    <port name="IHERetrieveForDisplayHttpGet" binding="s0:IHERetrieveForDisplayHttpGet">
      <http:address location="http://localhost/" />
    </port>
</service>
```

Appendix B: Definition of Document Unique IDs

The Retrieve Information for Display Integration Profile in its Retrieve Persistent Document transaction relies on a globally unique identification of persistent objects. It is the Information Source Actor's responsibility, when a specific document instance is available for retrieval, to assign to this document instance a globally unique identifier, thus allowing Display Actors to retrieve the same document instance at different points in time and to obtain the same semantics for its presented content.

This appendix describes how unique identifiers for documents shall be created. A unique identifier may be created by the Information Source Actor or by any other system to which the information source is connected. The requirements specified in this appendix are derived from the common practices and definitions of OIDs in ISO 8824, HL7 V3 and CDA and UIDs in DICOM. They guarantee uniqueness across multiple countries, sites, vendors and equipment.

B.1: Requirements for Document UIDs

The UID identification scheme is based on the OSI Object Identification (numeric form) as defined by the ISO 8824 standard.

All Unique Identifiers, used within the context of this transaction shall be registered values as defined by ISO 9834-3 to ensure global uniqueness. These requirements result in the following structure for unique Ids.

B.2: Structure of a Document UID

Each Document UID is composed of two parts, an <org root> and a <suffix> separated by a "period". Therefore: UID = <org root>.<suffix>

The <org root> portion of the UID uniquely identifies an organization, (e.g., manufacturer, research organization, hospital, etc.), and is composed of a number of numeric components as defined by ISO 8824. The <suffix> portion of the UID is also composed of a number of numeric components, and shall be unique within the scope of the <org root>. This implies that the organization identified in the <org root> is responsible for guaranteeing <suffix> uniqueness by providing registration policies. These policies shall guarantee <suffix> uniqueness for all UID's created by that organization. Unlike the <org root>, which may be common for UID's in an organization, the <suffix> shall take different unique values between different UID's that identify different objects. The <org root> is used only for uniqueness and not for any other purpose.

Although a specific implementation may choose some particular structure for its generated UIDs, it should never assume that a UID carries any semantics. A UID shall not be "parsed" to find a particular value or component. Component definition (for the suffix) is implementation-specific and may change as long as uniqueness is maintained. Parsing UID's (including extracting the root) may jeopardize the ability to inter-operate as implementations evolve.

B.3: Document UID encoding rules

The UID encoding rules are defined as follows:

- Each component of a UID is a number and shall consist of one or more digits. The first digit of each component shall not be zero unless the component is a single digit.

Note: Registration authorities may distribute components with non-significant leading zeroes. The leading zeroes should be ignored when being encoded (ie. “00029” would be encoded “29”).

- Each component numeric value shall be encoded using the characters 0-9 of the Basic G0 Set of the International Reference Version of ISO 646:1990. This particular encoding is the same as the UTF-8 encoding for these characters in UNICODE.
- Components shall be separated by the character "." (2EH).
- UIDs shall not exceed 64 total characters, including the digits of each component, and separators between components.

B.4: How to obtain a UID registration root?

Organizations that define UIDs are responsible for properly registering their UIDs (at least obtain a registered <Org Root>) as defined for OSI Object Identifiers (ISO 9834-3). The organization defining the UID shall accept the responsibility of ensuring its uniqueness. IHE will not register UIDs or issue registered organization roots. There are a large number of means to obtain free or for a reasonable fee an organization root.

A useful resource that is often used by the DICOM community lists the many ways to obtain a registered UID Root for a small fee or even for free, anywhere in the world.

http://www.dclunie.com/medical-image-faq/html/part8.html#UIDRegistration

The manner in which the suffix of a Document UID is defined is not constrained by any IHE Integration Profile. Only the guarantee of its uniqueness by the defining organization is required by IHE.

B.5: Example of a Document UID

This example presents a particular choice made by a specific organization in defining its suffix to guarantee uniqueness. A variant is discussed.

"1.2.840.xxxxx.4076078054086.11059664469.235212"

(root) (suffix)

In this example, the root is:

1	Identifies ISO
2	Identifies ANSI Member Body
840	Country code of a specific Member Body (U.S. for ANSI)

xxxxx Identifies a specific Organization.(provided by ANSI)

In this example the remaining components of the suffix relate to the identification of a specific document instance:

4076078054086	802.3 MAC Address (004 076 078 054 086)
11059664469	Time system was booted (July 31, 2033 10:14:29)
235212	Monotonically increasing sequence number

In this example, the organization has chosen these components to guarantee uniqueness. Other organizations may choose an entirely different series of components to uniquely identify its documents.

Because of the flexibility allowed in creating Document UIDs, implementations should not depend on any assumed structure of UIDs and should not attempt to parse UIDs to extract the semantics of some of its components.

Appendix C: HL7 Profiling Conventions

The HL7 tables included in this document have been modified from the HL7 2.5 standard document. Such a modification is called a profile. Refer to the HL7 2.5 standard for the meanings of specific columns in the table.

The profiling tables in this document leverage the ongoing HL7 profile definition. To maintain this specification at a generic level, the following differences have been introduced:

- Message specifications do not indicate the cardinality of segments within a message.
- For fields composed of multiple components, there is no indication of the size of each component.
- Where a table containing enumerated values is referenced from within a segment profile table, the enumerated values table is not always present.
- The number of times a repeating field can repeat is not indicated.
- The conditions that would require inclusion of conditional fields are not defined when they depend on functional characteristics of the system implementing the transaction and they do not affect data consistency.

The following terms refer to the OPT column, which has been profiled:

R	Required
R2	This is an IHE extension. If the sending application has data for the field, it is required to populate the field. If the value is not known, the field may not be sent.
R+	This is an IHE extension. This is a field that IHE requires that was listed as optional within the HL7 standard.
O	Optional
C	Conditional

IHE requires that Z-segments be present in HL7 transactions only when defined by the IHE IT Infrastructure Technical Framework.

According to the HL7 standard, if the value of a field is not present, the receiver shall not change corresponding data in its database. However, if sender includes explicit NULL value (i.e., two double-quotes “”), it shall cause removal of any values for that field in the receiver’s database.

Table C-1 provides a sample profile for an imaginary HL7 segment. Tables for real segments are copied from the HL7 2.5 standard with modifications made only to the OPT column.

Table C-1 Sample HL7 Profile

SEQ	LEN	DT	OPT	TBL#	ITEM #	ELEMENT NAME
1	1	ST	R		xx001	Element 1
2	4	ST	O		xx002	Element 2

3	180	HD	R2		xx003	Element 3
4	180	HD	C		xx004	Element 4
5	180	HD	O		xx005	Element 5
6	180	HD	R+		xx006	Element 6

C.1: HL7 Implementation Notes

Network Guidelines

The HL7 2.5 standard does not define a network communications protocol. Beginning with HL7 2.2 , the definitions of lower layer protocols were moved to the Implementation Guide, but are not HL7 requirements. The IHE Framework makes these recommendations:

1. Applications shall use the Minimal Lower Layer Protocol defined in Appendix C of the HL7 Implementation Guide.
5. An application that wants to send a message (initiate a transaction) will initiate a network connection to start the transaction. The receiver application will respond with an acknowledgement or response to query but will not initiate new transactions on this network connection.

Message Control

According to the HL7 standard, each message shall begin with the MSH (message header) segment. Table C.1-1 identifies all required fields in this message. This table shall be interpreted according to the HL7 Standard unless otherwise noted in Appendix C.

Table C.1-1 IHE Profile - MSH segment

SEQ	LEN	DT	OPT	TBL#	ITEM #	ELEMENT NAME
1	1	ST	R		00001	Field Separator
2	4	ST	R		00002	Encoding Characters
3	180	HD	R+		00003	Sending Application
4	180	HD	R+		00004	Sending Facility
5	180	HD	R+		00005	Receiving Application
6	180	HD	R+		00006	Receiving Facility
7	26	TS	R		00007	Date/Time Of Message
8	40	ST	O		00008	Security
9	13	CM	R	0076/ 0003	00009	Message Type
10	20	ST	R		00010	Message Control ID
11	3	PT	R		00011	Processing ID
12	60	VID	R	0104	00012	Version ID
13	15	NM	O		00013	Sequence Number
14	180	ST	O		00014	Continuation Pointer

15	2	ID	O	0155	00015	Accept Acknowledgment Type
16	2	ID	O	0155	00016	Application Acknowledgment Type
17	3	ID	O	0399	00017	Country Code
18	16	ID	C	0211	00692	Character Set
19	250	CE	O		00693	Principal Language Of Message
20	20	ID	O	0356	01317	Alternate Character Set Handling Scheme
21	10	ID	O	0449	01598	Conformance Statement ID #

Adapted from the HL7 Standard, version 2.5 and version 2.3.1

Note: This element is only applicable in HL7 version 2.5 and thus is only applicable for those transactions based on HL7 v2.5

The IHE IT Infrastructure Technical Framework requires that applications support HL7-recommended values for the fields *MSH-1-Field Separator* and *MSH-2-Encoding Characters*.

Field *MSH-18-Character Set* shall only be valued if the message utilizes character sets other than ISO IR-6, also known as ASCII.

Implementations supporting sequence number protocol (and using the field *MSH-13-Sequence Number)* shall be configurable to allow them to perform transactions without such protocol.

Acknowledgment Modes

Applications that receive HL7 messages shall conform to the acknowledgment and response requirements using the HL7 Original Mode (versus Enhanced Acknowledgment Mode).

The IHE IT Infrastructure Technical Framework provides for each HL7 message to be acknowledged by the HL7 ACK or by the appropriate HL7 Query/Response message sent by the receiver of an HL7 message to its sender. The segment requirements for responses to queries can be found within the respective sections for the transactions. The segments of the ACK message listed below are required, and their detailed descriptions are provided in tables C.1-2, C.1-3 and corresponding notes. The ERR segment is optional and may be included if the *MSA-1-Acknowledgment Code* field identifies an error condition.

ACK	Acknowledgement Message	Chapter in HL7 2.3.1 / 2.5
MSH	Message Header	2
MSA	Message Acknowledgement	2
[ERR]	Error Comments	2

Table C.1-2 IHE Profile - MSA segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	2	ID	R	0008	00018	Acknowledgment Code

2	20	ST	R		00010	Message Control ID
3	80	ST	O		00020	Text Message
4	15	NM	O		00021	Expected Sequence Number
5	1	ID	O	0102	00022	Delayed Acknowledgment Type
6	250	CE	O	0357	00023	Error Condition

Adapted from the HL7 standard, version 2.5 and version 2.3.1

Field *MSA-2-Message Control ID* shall contain the Message ID from the *MSH-10-Message Control ID* of the incoming message for which this acknowledgement is sent.

Table C.1-3 IHE Profile - ERR segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	80	ID	R		00024	Error code and location

Adapted from the HL7 standard, version 2.5 and version 2.3.1

Common Segment Definitions

The following table specifies the contents of the EVN segment that is common to several HL7-based transaction messages defined in this volume.

Table C.1-4 IHE Profile - EVN segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	3	ID	O	0003	00099	Event Type Code
2	26	TS	R		00100	Recorded Date/Time
3	26	TS	O		00101	Date/Time Planned Event
4	3	IS	O	0062	00102	Event Reason Code
5	60	XCN	O	0188	00103	Operator ID
6	26	TS	R2		01278	Event Occurred
7	180	HD	O		01534	Event Facility #

Adapted from the HL7 Standard, version 2.5 and version 2.3.1

Note: This element is only applicable in HL7 version 2.5 and thus is only applicable for those transactions based on HL7 v2.5

Field *EVN-1-Event Type Code* is optional; however, if present, its value shall be equal to the second component of the field *MSH-9-Message Type*.

Appendix D: Cross-Profile Interactions of PIX and PSA

When the Context Manager Actor in a Patient Synchronized Application Integration Profile is grouped with a Patient Identifier Cross-reference Consumer in a Patient Identifier Cross-referencing Integration Profile, patient identifiers must be accessible to both actors in a consistent manner. This Appendix provides the necessary mapping rules.

The Patient Identifier Cross-Referencing (PIX) Integration Profile defines a general-purpose mapping of a Patient ID within a Patient Identification Domain to aliases in other Patient Identification Domains. This mapping is intended to be used across all IHE systems that require patient identification in transactions crossing Patient Identification Domains. The PIX Integration Profile relies on HL7 V2 Transactions.

The Patient Synchronized Application Integration Profile relies on HL7 CCOW which, confronted with a similar need, has defined a Patient Mapping API within its architecture. The HTTP Technology mapping for the CCOW Patient Mapping Agent API supports its operation over a network interface, thus creating an alternative to HL7 V2 messages.

As IHE strives to avoid the inclusion in its integration profiles of incompatible but functionally equivalent variants, it has decided to use HL7 V2 ADT messages for the Patient Identifier Cross-referencing Integration Profiles. In consequence, the combined use of the Patient Synchronized (CCOW based) Integration Profile and of the Patient Identifier Cross-referencing Integration profiles requires that the IHE Context Manager Actor uses the services of the PIX Integration Profile. To do so, the Patient Identifier Cross-reference Consumer Actor in communication with the Patient Identifier Cross-reference Manager Actor operates as a substitute for the CCOW Patient Mapping Agent. This is shown in diagram D-1 below as a dashed oval surrounding the Patient Cross-reference Manager and the Patient Identifier Cross-reference Consumer actors. As a result it is likely that a context management solution would bundle a PMA proxy application that would implement the PIX Query in support of the Patient Identifier Cross-reference Consumer Actor.

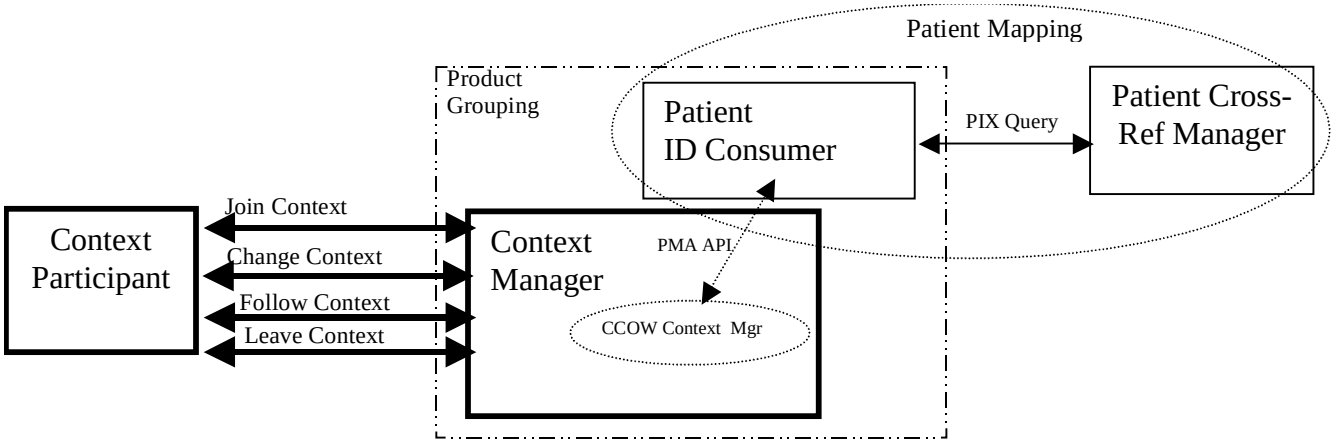


Figure D-1: Actor Grouping Diagram

This Appendix provides the definition of the mapping of the CCOW Patient Mapping Agent API methods onto the PIX Query Transaction (HL7 V2 QBP^Q23/RSP^K23) as defined by the PIX Integration Profile.

Table D-1 shows the definition of the Patient Mapping Methods parameters as implemented in Web technology. Most of these Arguments relate to the normal operations of the Patient Mapping Agent methods that pose no mapping challenge except for the ItemNames and ItemValues which pose some constraints. The first constraint comes from the translation of Patient Identity Domains for both query and response from and to a CCOW defined name / value pair. The second one comes from the fact that CCOW participant applications can set more than one identifier in context the ability to detect when these identifiers represent the identities of more than one patient. IHE has taken steps to mitigate these issues by further restricting how the IHE Context Participant implements the methods. Each of these constraints is addressed in sections below.

Table D-1 ContextChangesPending

HTTP Request Message		
Argument Name	**Data Type**	**Comment**
Interface	string	"ContextAgent"
Method	string	"ContextChangesPending"
agentCoupon	long	"-1"
contextManager	string	URL for the Context Manager that is requesting the patient id cross-reference
itemNames	string[]	One or more item names (e.g. Patient.Id.IdList)
itemValues	string[]	The patient identifiers corresponding to the domains identified in item names
contextCoupon	long	Context Coupon value for pending context change transaction

HTTP Request Message		
Argument Name	**Data Type**	**Comment**
managerSignature	string	Not required
HTTP Reply Message		
agentCoupon	long	"-1"
itemNames	string[]	See below for valid item names for patient subject
itemValues	string[]	See below for any constraints on item values
contextCoupon	long	Return the value provided in request
agentSignature	string	Not required
Decision	string	"valid" or "invalid"
Reason	string	Reason text if mapping is invalid

Adapted from the HL7 Context Management "CCOW" Standard, version 1.4

D.1: Namespace Translation from PIX Query to CCOW

The CCOW standard defines multiple identifier items that may be set into the context by an instigating participant application. The current list of valid identifier names are listed in Table D-2.

Table D-2 Patient Subject Identifiers

Patient Subject Identifier Item Name	HL7 Meaning	HL7 Data Type	HL7 Semantic Constraints on Values	Case Sensitive
Patient.Id.MRN.*Suffix*	Patient medical record number, per PID-2	ST	HL7 Table 0203 Identifier Type = MR	No
Patient.Id.MPI	Patient identifier in the "Master Patient Index", per PID-2	ST	HL7 Table 0203 Identifier Type = PT or PI (as agreed upon by context sharing systems) and Assigning Authority represents the MPI system	No
Patient.Id.NationalIdNumber	Patient national identifier number, per PID-2	ST	HL7 Table 0203 Identifier Type = PT and Assigning Authority represents agreed upon National Authority	No
Patient.Id.IdList	A list of patient identifiers for a patient, per PID-3	CX	May be a repeating set of CX item values (per Section 1.7 of the HL7 Context Management "CCOW" Standard: Subject Data Definitions document), each of which contains an identifier that denotes the same patient	No

Adapted from the HL7 Context Management "CCOW" Standard, version 1.4

IHE has specified in the Context Change Transaction as documented in ITI TF-2 that the Context Participant Actor shall use the Patient.Id.IdList item. The intent is to eliminate translation as the Patient.Id.IdList value maps directly to PIX Query Transaction QPD-3.

Applications using in their identifier items Patient.Id.MRN.Suffix will need to migrate to the Patient.Id.IdList item as expected by the HL7 CCOW standard.

D.2: Processing Multiple Identifiers

CCOW participant applications are permitted to populate as many patient identifiers as they have available to them. This means that when a user selects a patient in one of these applications the context is populated with multiple identifiers for the selected patient. When the CCOW Patient Mapping Agent (PMA) accepts multiple patient identifiers as input, the PMA has the responsibility of invalidating patient mapping and causing the context change transaction to be cancelled if it determines that the multiple identifiers supplied as part of the transaction identify more than one patient.

The QPD segment as defined in the IHE PIX Query Transaction specifies a single identifier uniquely identifying one patient within a given Patient Identification Domain. In the case where multiple identifiers are populated, the context manager may have to process the response to the initial PIX Query Transaction to evaluate if the other identifiers in context are included. If so, no further processing is required. Otherwise, an additional PIX Query will need to be issued and the results processed. Should a non-null result be returned, indicating the identifier uniquely identifies a different patient for the given domain, the context manager shall assume “invalid” in the decision field and “multiple patients identified” in the reason field.

In order to mitigate this condition, IHE specifies that all context participants supporting the Patient Synchronized Applications profile shall only set one identifier for the patient when a Patient Identifier Cross-referencing Integration Profile is used by the context manager. This means that the context participant for those applications that manage multiple patient identifiers will need to be configurable as to which identifier item is passed in the Change Context Transaction.

Appendix E: Usage of the CX Data Type in PID-3-Patient Identifier List

The Health Level Seven Standard (HL7) uses data type CX to express various identifiers, including the Patient ID in the third field of the PID segment. We discuss here how IHE IT Infrastructure expects the CX data type to be populated in the *PID-3-Patient Identifier List* fields of messages that it defines.

Requirements for populating the elements of *PID-3-Patient Identifier List* vary slightly, depending on what actor is originating the transaction in which the PID segment is sent. If the Patient Identifier Cross-reference Manager is the source of the PID segment, the requirements (specifically, with respect to populating the Assigning Authority subcomponents) are more rigorous than otherwise.

PID-3-Patient Identifier List permits multiple occurrences of the CX data type. Data type CX contains 8 components as shown below. This structure allows expression of the value and context for each identifier that the system knows.

Table E-1: Components of HL7 Data Type CX

Cmp	Len	DT	Opt	Tbl	Name
1	15	ST	R		ID
2		ST	O		Check digit
3		ID	O	0061	Code identifying the check digit scheme employed
4	227	HD	R		Assigning authority
5		ID	O	0203	Identifier type code
6		HD	O		Assigning facility
7		DT	O		Effective date
8		DT	O		Expiration date

Adapted from the HL7 Standard, Version 2.5

Each occurrence of *PID-3-Patient Identifier List* contains, at a minimum, an identifier value in Component 1 and an assigning authority in Component 4. The assigning authority unambiguously provides the context for the identifier. It is also common practice to provide an identifier type code in Component 5, but this is not required by IHE. Other components are optional and will not be discussed here; implementers may refer to HL7 Version 2.5 for more information.

Component 1 of Data Type CX, **ID**, is of data type ST. This data type allows a free text value of up to 15 characters.[8]

Component 4 of Data Type CX, **Assigning Authority**, is of data type HD. This data type contains 3 components that, when implemented at the component level, become subcomponents of Component 4. The requirements for the subcomponents of Component 4 vary by actor.

E.1: Patient Identifier Cross-reference Manager actor requirements

The Patient Identifier Cross-reference Manager Actor is expected to have access to complete internal and external identifier information for the Assigning Authority of the patient identifier. To facilitate interoperability, it is required that the Patient Identifier Cross-reference Manager Actor populate all subcomponents of the Assigning Authority component. The usage of these subcomponents will be explained in the examples below.

This requirement applies to the response portion of Transaction ITI-9 (PIX Query) and to Transaction ITI-10 (PIX Update Notification).

Table E-2: Usage of HL7 Data Type CX by the PIX Manager Actor

Cmp	Sbc	Len	DT	Opt	Tbl	Name	Conditionality predicate
1		15	ST	R		ID	
2			ST	O		Check digit	
3			ID	O	0061	Code identifying the check digit scheme employed	
4		227	HD	R		Assigning authority	Subcomponent 1 must refer to the same entity as Subcomponents 2 and 3.
4	1	20	IS	R	0363	Namespace ID	
4	2	199	ST	R		Universal ID	
4	3	6	ID	R	0301	Universal ID type	
5			ID	O	0203	Identifier type code	
6			HD	O		Assigning facility	If all three subcomponents are populated, they must refer to the same entity.
6	1		IS	O	0300	Namespace ID	
6	2		ST	C		Universal ID	Populated if, and only if, Subcomponent 3 is populated.
6	3		ID	C	0301	Universal ID type	Populated if, and only if, Subcomponent 2 is populated

[8] As implemented in HL7 Version 2.5. Prior to Version 2.5, HL7 did not specify the length of individual components. Although the profiles in IHE-ITI are based Versions 2.3.1 and 2.4 of HL7, they use the component length constraints provided by Version 2.5 to support forward compatibility.

Cmp	Sbc	Len	DT	Opt	Tbl	Name	Conditionality predicate
7			DT	O		Effective date	
8			DT	O		Expiration date	

IHE specifies that the Patient Identifier Cross-reference Manager actor must populate all 3 subcomponents of Component 4. The following rules apply:

Subcomponent 1 of Component 4, **Namespace ID**, is of data type IS. HL7 specifies that when valued in the Patient ID field, the value in this subcomponent be a code taken from user-defined Table 0363, *Assigning Authority*. Version 2.5 of HL7 provides suggested values for assigning authorities in various local jurisdictions, such as **USSSA** for U.S. Social Security Administration. Sites may add values to this table, but for interoperability must ensure that added values (and meanings) are agreed upon by all communicating systems.

Subcomponent 2 of Component 4, **Universal ID**, is of data type ST. This subcomponent contains a value from either a known external domain or a specified internal domain. The domain is given in Subcomponent 3.

Subcomponent 3, **Universal ID Type**, is of data type ID. This subcomponent contains a code taken from HL7 Table 0301, *Universal ID Type*. Table 0301 contains values for various known external identifier domains such as **DNS** (Internet dotted name) and **ISO** (International Standards Organization Object Identifier, or OID), as well as the values **L, M,** and **N** to permit the use of internal identifier domains.

Subcomponent 1 must refer to the same entity as Subcomponents 2 and 3.

E.2: Other actor requirements

The PID segment may also appear in messages generated by other IHE Actors, including the Patient ID Cross-reference Consumer and the Information Source. These actors must also populate the Assigning Authority.

However, IHE specifies that they need not populate all three subcomponents of Assigning Authority. They must populate either Namespace ID (an entry from a user-defined table), or Universal ID and Universal ID Type (allowing the use of an externally defined identifier scheme).

This requirement applies to Transaction 8 (Patient Identity Feed), to the query portion of Transaction ITI-9 (PIX Query), and to any other transaction (except for the response portion of ITI-9 and for ITI-10) that populates *PID-3-Patient Identifier List*.

Table E-3: Usage of HL7 Data Type CX by other IHE Actors

Cmp	Sbc	Len	DT	Opt	Tbl	Name	Conditionality predicate
1		15	ST	R		ID	
2			ST	O		Check digit	
3			ID	O	0061	Code identifying the	

Cmp	Sbc	Len	DT	Opt	Tbl	Name	Conditionality predicate
						check digit scheme employed	
4		227	HD	R		Assigning authority	If all three subcomponents are populated, they must refer to the same entity.
4	1	20	IS	C	0363	Namespace ID	Must be populated if Subcomponents 2 and 3 are not populated.
4	2	199	ST	C		Universal ID	Must be populated if Subcomponent 1 is not populated. Populated if, and only if, Subcomponent 3 is populated.
4	3	6	ID	C	0301	Universal ID type	Must be populated if Subcomponent 1 is not populated. Populated if, and only if, Subcomponent 2 is populated.
5			ID	O	0203	Identifier type code	
6			HD	O		Assigning facility	If all three subcomponents are populated, they must refer to the same entity.
6	1		IS	O	0300	Namespace ID	
6	2		ST	C		Universal ID	Populated if, and only if, Subcomponent 3 is populated.
6	3		ID	C	0301	Universal ID type	Populated if, and only if, Subcomponent 2 is populated.
7			DT	O		Effective date	
8			DT	O		Expiration date	

The definitions of the subcomponents of Component 4 are as given above for the Patient Identifier Cross-reference Manager actor. If all three subcomponents are defined, Subcomponent 1 must refer to the same entity as Subcomponents 2 and 3.

E.3: E.3 Examples of use

Metropolitan Medical Center treats a patient, Jane Smith, for whom 3 identifiers are known. (For this example, assume that the HL7 V2 default delimiters are in use: | for field separator, ^ for component separator, ~ for repetition separator and & for subcomponent separator.)

3.24.6 Data sent by source systems

The source systems provide data to the Patient Identifier Cross-reference Manager. These data are sent either in a Patient Identity Feed transaction [ITI-8] or in response to a PIX Query.

- Patient Smith's Social Security number is **999-99-4452**. This number is assigned by the U.S. Social Security Administration.

The ADT system sends the Social Security number at registration, in an occurrence of *PID-3-Patient Identifier List* that looks like this:

```
999-99-4452^^^USSSA
```

Note that only Subcomponent 1 of Assigning Authority is assigned here, while Subcomponents 2 and 3 are left empty.

- Patient Smith's medical record number is **9990-99497**. This number is assigned by Metropolitan Medical Center, for which no external identifier is known. Metropolitan Medical Center incorporates the Namespace ID **99MMC** for the medical record numbers it assigns.

The ADT system sends the medical record number at registration, in an occurrence of *PID-3-Patient Identifier List* that looks like this:

```
999099497^^^99MMC
```

Note again that only Subcomponent 1 of Assigning Authority is assigned here.

- Patient Smith's medical insurance number is **99998410**. This number is assigned by MLH Life & Casualty Company, whose Internet domain name is **www.mlhlifecasualty.com**.[9]

The billing system sends the medical insurance number in an occurrence of *PID-3-Patient Identifier List* that looks like this:

```
99998410^^^&www.mlhlife.com&DNS
```

Note that only Subcomponents 2 and 3 of Assigning Authority are assigned here. Also note the value **DNS** in the third subcomponent of Component 4 to indicate an Internet domain name.

3.24.7 Data sent by the Patient Identifier Cross-reference Manager

The Patient Identifier Cross-reference Manager implements HL7 Table 0363, *Assigning Authority*, by incorporating the values in HL7 Version 2.5 as well as the values **99MMC** for Metropolitan Medical Center and **99MLHLIFE** for MLH Life & Casualty.[10] It also includes a known ISO Object Identifier for the Social Security Administration, **1.2.mm.nnnnn.555.6666**.[11]

To send the identifiers in *PID-3-Patient Identifier List*, the Patient Identifier Cross-reference Manager builds and concatenates them as follows.

[9] Implementers should take into account the possibility that, as with any domain identifier, Internet domain identifiers – either fully qualified domain names (FQDNs) or IPv4 or IPv6 addresses – are liable to change.

[10] The use of **99** to preface these codes is not mandated by HL7, but reflects the practice directed by Chapter 7 of HL7 Version 2.5 for specifying local coding system values.

[11] This OID is fictitious. The real OID for the SSA should be substituted here.

- In the first occurrence, the Social Security number is sent in the first component, as well as the known internal and external values for SSN assigning authority in the fourth component. Note the value **ISO** in the third subcomponent of Component 4 to indicate an ISO Object Identifier.

```
999-99-4452^^^USSSA&1.2.mm.nnnnn.555.6666&ISO
```

- In the second occurrence, the medical insurance number is sent in the first component, as well as the known internal and external values for insurance number assigning authority in the fourth component.

```
99998410^^^99MLHLIFE&www.mlhlife.com&DNS
```

- In the third occurrence, the medical record number is sent in the first component, as well as the known internal and external values for MRN assigning authority in the fourth component. Note that no external value is known for MRN assigning authority, so the HIS repeats the internal value as an external value and uses the value **L** in the third subcomponent of Component 4 to indicate a locally assigned value.

```
999099497^^^99MMC&99MMC&L
```

In sending all values in a PIX Update Notification transaction [ITI-10], the Patient Identifier Cross-reference Manager concatenates the three *PID-3-Patient Identifier List* values using the repetition separator:

```
|999994452^^^USSSA&1.2.mm.nnnnn.555.6666&ISO~99998410^^^99A
  BCLIFE&www.abclife.com&DNS~999099497^^^99MMC&99MMC&|
```

Appendix F: Intentionally Left Blank

Appendix G: Transition from Radiology Basic Security to ATNA

G.1: Message Transformation

The IHE Provisional messages can be transformed into equivalent Audit Trail messages by various means. One such is the use of an XSLT. The following XSLT is an illustrative example of such a transformation. It has not been tested or verified on the full range of real world messages, nor does it include much error processing, but it does transform properly formed IHE Provisional audit trail messages into the RFC-3881 format. It is provided for educational purposes only.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:msxsl="urn:schemas-microsoft-
com:xslt" xmlns:js="javascript:code">
	<xsl:output method="xml" encoding="UTF-8"/>
	<msxsl:script language="JScript" implements-prefix="js"><![CDATA[
function encode64(i) // base 64 encoding
{
	var k = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";
	var o = "";
	var c1, c2, c3, e1, e2, e3, e4 = "";
	var l = 0;
	do
	{
		c1 = i.charCodeAt(l++);
		c2 = i.charCodeAt(l++);
		c3 = i.charCodeAt(l++);
		e1 = c1 >> 2;
		e2 = ((c1 & 3) << 4) | (c2 >> 4);
		e3 = ((c2 & 15) << 2) | (c3 >> 6);
		e4 = c3 & 63;
		if (isNaN(c2))
			e3 = e4 = 64;
		else if (isNaN(c3))
			e4 = 64;
		o = o + k.charAt(e1) + k.charAt(e2) + k.charAt(e3) + k.charAt(e4);
		c1 = c2 = c3 = e1 = e2 = e3 = e4 = "";
	}
	while (l < i.length);
	return o;
}
]]></msxsl:script>
	<xsl:template match="/IHEYr4">
		<AuditMessage>
			<xsl:attribute
name="xsi:noNamespaceSchemaLocation">http://www.xml.org/xml/schema/7f0d86bd/healthcare-security-
audit.xsd</xsl:attribute>
			<EventIdentification>
				<xsl:attribute name="EventDateTime"><xsl:value-of
select="TimeStamp"/></xsl:attribute>
				<xsl:apply-templates mode="EventIdentification"/>
			</EventIdentification>
			<xsl:apply-templates mode="ActiveParticipant"/>
			<xsl:apply-templates mode="AuditSourceIdentification"/>
			<xsl:apply-templates mode="ParticipantObjectIdentification"/>
```

```
                </AuditMessage>
        </xsl:template>
        <!-- ========================================== -->
        <!-- EventIdentification                        -->
        <!-- ========================================== -->
        <xsl:template mode="EventIdentification" match="ActorConfig">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'E'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'ActorConfig'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="ActorStartStop">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'E'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'ActorStartStop'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="AuditLogUsed">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'R'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'AuditLogUsed'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="BeginStoringInstances">
                <xsl:apply-templates mode="EventIdentification"
select="InstanceActionDescription"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'BeginStoringInstances'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="DICOMInstancesDeleted">
                <xsl:apply-templates mode="EventIdentification" select="ObjectAction"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'DICOMInstancesDeleted'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="DICOMInstancesUsed">
                <xsl:apply-templates mode="EventIdentification" select="ObjectAction"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'DICOMInstancesUsed'"/></xsl:attribute>
```

```
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="DicomQuery">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'E'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'DICOMQuery'"/></xsl:attribute>
                </EventID>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="Export">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'R'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'Export'"/></xsl:attribute>
                </EventID>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="Import">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'C'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'Import'"/></xsl:attribute>
                </EventID>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="InstanceActionDescription">
                <xsl:apply-templates mode="EventIdentification" select="ObjectAction"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="InstancesSent">
                <xsl:apply-templates mode="EventIdentification"
select="InstanceActionDescription"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'InstancesSent'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="InstancesStored">
                <xsl:apply-templates mode="EventIdentification"
select="InstanceActionDescription"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'InstancesStored'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="NetworkEntry">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'E'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'NetworkEntry'"/></xsl:attribute>
                </EventID>
```

```
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="ObjectAction">
                <xsl:attribute name="EventActionCode"><xsl:choose><xsl:when test=". =
'Create'"><xsl:value-of select="'C'"/></xsl:when><xsl:when test=". = 'Access'"><xsl:value-of
select="'R'"/></xsl:when><xsl:when test=". = 'Modify'"><xsl:value-of
select="'U'"/></xsl:when><xsl:when test=". = 'Delete'"><xsl:value-of
select="'D'"/></xsl:when><xsl:otherwise><xsl:value-of
select="'E'"/></xsl:otherwise></xsl:choose></xsl:attribute>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="OrderRecord">
                <xsl:apply-templates mode="EventIdentification" select="ObjectAction"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'OrderRecord'"/></xsl:attribute>
                </EventID>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="PatientRecord">
                <xsl:apply-templates mode="EventIdentification" select="ObjectAction"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'PatientRecord'"/></xsl:attribute>
                </EventID>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="ProcedureRecord">
                <xsl:apply-templates mode="EventIdentification" select="ObjectAction"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'ProcerdureRecord'"/></xsl:attribute>
                </EventID>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="SecurityAlert">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'E'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'SecurityAlert'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="StudyDeleted">
                <xsl:apply-templates mode="EventIdentification" select="ObjectAction"/>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
                        <xsl:attribute name="code"><xsl:value-of
select="'StudyDeleted'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="UserAuthenticated">
                <xsl:attribute name="EventActionCode"><xsl:value-of select="'E'"/></xsl:attribute>
                <xsl:attribute name="EventOutcomeIndicator"><xsl:value-of
select="0"/></xsl:attribute>
                <EventID>
```

```
                        <xsl:attribute name="code"><xsl:value-of
select="'UserAuthenticated'"/></xsl:attribute>
                </EventID>
                <xsl:apply-templates mode="EventTypeCode"/>
        </xsl:template>
        <xsl:template mode="EventIdentification" match="*"/>
        <!-- ========================================= -->
        <!-- EventIdentification / EventTypeCode       -->
        <!-- ========================================= -->
        <xsl:template mode="EventTypeCode" match="Action">
                <EventTypeCode>
                        <xsl:attribute name="code"><xsl:value-of select="."/></xsl:attribute>
                </EventTypeCode>
        </xsl:template>
        <xsl:template mode="EventTypeCode" match="AlertType">
                <EventTypeCode>
                        <xsl:attribute name="code"><xsl:value-of select="."/></xsl:attribute>
                </EventTypeCode>
        </xsl:template>
        <xsl:template mode="EventTypeCode" match="ApplicationAction">
                <EventTypeCode>
                        <xsl:attribute name="code"><xsl:value-of select="."/></xsl:attribute>
                </EventTypeCode>
        </xsl:template>
        <xsl:template mode="EventTypeCode" match="ConfigType">
                <EventTypeCode>
                        <xsl:attribute name="code"><xsl:value-of select="."/></xsl:attribute>
                </EventTypeCode>
        </xsl:template>
        <xsl:template mode="EventTypeCode" match="MachineAction">
                <EventTypeCode>
                        <xsl:attribute name="code"><xsl:value-of select="."/></xsl:attribute>
                </EventTypeCode>
        </xsl:template>
        <xsl:template mode="EventTypeCode" match="*"/>
        <!-- ========================================= -->
        <!-- Active Participant                        -->
        <!-- ========================================= -->
        <xsl:template mode="ActiveParticipant" match="ActorConfig">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="ActorName">
                <ActiveParticipant>
                        <xsl:attribute name="UserID"><xsl:value-of select="."/></xsl:attribute>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="false()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="ActorStartStop">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="AET">
                <xsl:attribute name="AlternativeUserID"><xsl:value-of select="."/></xsl:attribute>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="AuditLogUsed">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="BeginStoringInstances">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="Destination">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
```

```
        <xsl:template mode="ActiveParticipant" match="DICOMInstancesDeleted">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="DICOMInstancesUsed">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="DicomQuery">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="Export">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="Hname">
                <xsl:attribute name="UserName"><xsl:value-of select="."/></xsl:attribute>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="Import">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="InstanceActionDescription">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="InstancesSent">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="InstancesStored">
                <ActiveParticipant>
                        <xsl:apply-templates mode="ActiveParticipant" select="RemoteNode"/>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="true()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="IP">
                <xsl:attribute name="UserID"><xsl:value-of select="'node'"/></xsl:attribute>
                <xsl:attribute name="NetworkAccessPointID"><xsl:value-of
select="."/></xsl:attribute>
                <xsl:attribute name="NetworkAccessPointTypeCode"><xsl:value-of
select="2"/></xsl:attribute>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="LocalPrinter">
                <ActiveParticipant>
                        <xsl:attribute name="UserID"><xsl:value-of select="."/></xsl:attribute>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="false()"/></xsl:attribute>
                        <RoleIDCode>
                                <xsl:attribute name="code"><xsl:value-of
select="'Printer'"/></xsl:attribute>
                        </RoleIDCode>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="LocalUser">
                <xsl:attribute name="UserID"><xsl:value-of select="."/></xsl:attribute>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="LocalUsername">
                <ActiveParticipant>
                        <xsl:attribute name="UserID"><xsl:value-of select="."/></xsl:attribute>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="true()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="MediaID">
                <ActiveParticipant>
                        <xsl:attribute name="UserID"><xsl:value-of select="."/></xsl:attribute>
```

```
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="false()"/></xsl:attribute>
                        <xsl:apply-templates mode="ActiveParticipantRoleIDCode"
select="../MediaType"/>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="NetworkEntry">
                <ActiveParticipant>
                        <xsl:attribute name="UserID"><xsl:value-of
select="../Host"/></xsl:attribute>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="false()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="Node">
                <ActiveParticipant>
                        <xsl:apply-templates mode="ActiveParticipant"/>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="false()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="OrderRecord">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="PatientRecord">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="ProcedureRecord">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="RemoteNode">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="Requestor">
                <ActiveParticipant>
                        <xsl:apply-templates mode="ActiveParticipant"/>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="true()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="RNode">
                <ActiveParticipant>
                        <xsl:apply-templates mode="ActiveParticipant"/>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="false()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="SecurityAlert">
                <xsl:apply-templates mode="ActiveParticipant"/>
                <ActiveParticipant>
                        <xsl:attribute name="UserID"><xsl:value-of
select="../Host"/></xsl:attribute>
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="false()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="StudyDeleted">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="User">
                <ActiveParticipant>
                        <xsl:apply-templates mode="ActiveParticipant"/>
```

```
                        <xsl:attribute name="UserIsRequestor"><xsl:value-of
select="true()"/></xsl:attribute>
                </ActiveParticipant>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="UserAuthenticated">
                <xsl:apply-templates mode="ActiveParticipant"/>
        </xsl:template>
        <xsl:template mode="ActiveParticipant" match="*"/>
        <!-- ========================================== -->
        <!-- Active Participant / RoleIDCode            -->
        <!-- ========================================== -->
        <xsl:template mode="ActiveParticipantRoleIDCode" match="MediaType">
                <RoleIDCode>
                        <xsl:attribute name="code"><xsl:value-of select="."/></xsl:attribute>
                </RoleIDCode>
        </xsl:template>
        <xsl:template mode="ActiveParticipantRoleIDCode" match="*"/>
        <!-- ========================================== -->
        <!-- AuditSourceIdentification                  -->
        <!-- ========================================== -->
        <xsl:template mode="AuditSourceIdentification" match="Host">
                <AuditSourceIdentification>
                        <xsl:attribute name="AuditSourceID"><xsl:value-of
select="."/></xsl:attribute>
                </AuditSourceIdentification>
        </xsl:template>
        <xsl:template mode="AuditSourceIdentification" match="*"/>
        <!-- ========================================== -->
        <!-- ParticipantObjectIdentification            -->
        <!-- ========================================== -->
        <xsl:template mode="ParticipantObjectIdentification" match="BeginStoringInstances">
                <xsl:apply-templates mode="ParticipantObjectIdentification"/>
        </xsl:template>
        <xsl:template mode="ParticipantObjectIdentification" match="DICOMInstancesDeleted">
                <xsl:apply-templates mode="ParticipantObjectIdentification"/>
        </xsl:template>
        <xsl:template mode="ParticipantObjectIdentification" match="DICOMInstancesUsed">
                <xsl:apply-templates mode="ParticipantObjectIdentification"/>
        </xsl:template>
        <xsl:template mode="ParticipantObjectIdentification" match="DicomQuery">
                <ParticipantObjectIdentification>
                        <xsl:attribute name="ParticipantObjectTypeCode"><xsl:value-of
select="2"/></xsl:attribute>
                        <xsl:attribute name="ParticipantObjectTypeCodeRole"><xsl:value-of
select="3"/></xsl:attribute>
                        <xsl:attribute name="ParticipantObjectID"><xsl:value-of
select="CUID"/></xsl:attribute>
                        <ParticipantObjectIDTypeCode>
                                <xsl:attribute name="code"><xsl:value-of
select="'2'"/></xsl:attribute>
                        </ParticipantObjectIDTypeCode>
                        <xsl:apply-templates mode="ParticipantObjectQuery" select="Keys"/>
                        <xsl:apply-templates mode="ParticipantObjectDetail" select="SyntaxUID"/>
                </ParticipantObjectIdentification>
        </xsl:template>
        <xsl:template mode="ParticipantObjectIdentification" match="Export">
                <xsl:apply-templates mode="ParticipantObjectIdentification"/>
        </xsl:template>
        <xsl:template mode="ParticipantObjectIdentification" match="Import">
                <xsl:apply-templates mode="ParticipantObjectIdentification"/>
        </xsl:template>
        <xsl:template mode="ParticipantObjectIdentification" match="InstanceActionDescription">
                <xsl:apply-templates mode="ParticipantObjectIdentification"/>
```

```
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="InstancesSent">
		<xsl:apply-templates mode="ParticipantObjectIdentification"/>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="InstancesStored">
		<xsl:apply-templates mode="ParticipantObjectIdentification"/>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="OrderRecord">
		<xsl:apply-templates mode="ParticipantObjectIdentification"/>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="Patient">
		<ParticipantObjectIdentification>
			<xsl:apply-templates mode="ParticipantObjectIdentification"
select="PatientID"/>
			<xsl:attribute name="ParticipantObjectTypeCode"><xsl:value-of
select="1"/></xsl:attribute>
			<xsl:attribute name="ParticipantObjectTypeCodeRole"><xsl:value-of
select="1"/></xsl:attribute>
			<ParticipantObjectIDTypeCode>
				<xsl:attribute name="code"><xsl:value-of
select="'2'"/></xsl:attribute>
			</ParticipantObjectIDTypeCode>
			<xsl:apply-templates mode="ParticipantObjectIdentification"
select="PatientName"/>
			<xsl:apply-templates mode="ParticipantObjectDetail" select="../SUID"/>
		</ParticipantObjectIdentification>
		<xsl:apply-templates mode="ParticipantObjectIdentification" select="SUID"/>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="PatientID">
		<xsl:attribute name="ParticipantObjectID"><xsl:value-of
select="."/></xsl:attribute>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="PatientName">
		<ParticipantObjectName>
			<xsl:value-of select="."/>
		</ParticipantObjectName>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="PatientRecord">
		<xsl:apply-templates mode="ParticipantObjectIdentification"/>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="ProcedureRecord">
		<xsl:apply-templates mode="ParticipantObjectIdentification"/>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="SecurityAlert">
		<ParticipantObjectIdentification>
			<xsl:attribute name="ParticipantObjectID"><xsl:value-of
select="../Host"/></xsl:attribute>
			<xsl:attribute name="ParticipantObjectTypeCode"><xsl:value-of
select="2"/></xsl:attribute>
			<xsl:attribute name="ParticipantObjectTypeCodeRole"><xsl:value-of
select="13"/></xsl:attribute>
			<ParticipantObjectIDTypeCode>
				<xsl:attribute name="code"><xsl:value-of
select="12"/></xsl:attribute>
			</ParticipantObjectIDTypeCode>
			<xsl:apply-templates mode="ParticipantObjectDetail" select="Description"/>
		</ParticipantObjectIdentification>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="StudyDeleted">
		<xsl:apply-templates mode="ParticipantObjectIdentification"/>
	</xsl:template>
	<xsl:template mode="ParticipantObjectIdentification" match="SUID">
		<ParticipantObjectIdentification>
```

```
                        <xsl:attribute name="ParticipantObjectID"><xsl:value-of
select="."/></xsl:attribute>
                        <xsl:attribute name="ParticipantObjectTypeCode"><xsl:value-of
select="2"/></xsl:attribute>
                        <xsl:attribute name="ParticipantObjectTypeCodeRole"><xsl:value-of
select="3"/></xsl:attribute>
                        <ParticipantObjectIDTypeCode>
                                <xsl:attribute name="code"><xsl:value-of
select="'9'"/></xsl:attribute>
                        </ParticipantObjectIDTypeCode>
                        <xsl:apply-templates mode="ParticipantObjectDetail"
select="../AccessionNumber"/>
                        <xsl:apply-templates mode="ParticipantObjectDetail" select="../CUID"/>
                        <xsl:apply-templates mode="ParticipantObjectDetail"
select="../FillerOrderNumber"/>
                        <xsl:apply-templates mode="ParticipantObjectDetail"
select="../NumberOfInstances"/>
                        <xsl:apply-templates mode="ParticipantObjectDetail" select="../MPPSUID"/>
                        <xsl:apply-templates mode="ParticipantObjectDetail"
select="../PlacerOrderNumber"/>
                </ParticipantObjectIdentification>
        </xsl:template>
        <xsl:template mode="ParticipantObjectIdentification" match="*"/>
        <!-- ========================================================= -->
        <!-- ParticipantObjectIdentification / ParticipantObjectDetail -->
        <!-- ========================================================= -->
        <xsl:template mode="ParticipantObjectDetail" match="AccessionNumber">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'AccessionNumber'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="CUID">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'ContainsSOPClass'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="Description">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'Description'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="FillerOrderNumber">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'FillerOrderNumber'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="MPPSUID">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'ContainsMPPS'"/></xsl:attribute>
```

```
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="NumberOfInstances">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'NumberOfInstances'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="PlacerOrderNumber">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'PlacerOrderNumber'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="SUID">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'ContainsSOPInstances'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="SyntaxUID">
                <ParticipantObjectDetail>
                        <xsl:attribute name="type"><xsl:value-of
select="'TransferSyntax'"/></xsl:attribute>
                        <xsl:attribute name="value"><xsl:value-of
select="js:encode64(string(.))"/></xsl:attribute>
                </ParticipantObjectDetail>
        </xsl:template>
        <xsl:template mode="ParticipantObjectDetail" match="*"/>
        <!-- ========================================================= -->
        <!-- ParticipantObjectIdentification / ParticipantObjectQuery  -->
        <!-- ========================================================= -->
        <xsl:template mode="ParticipantObjectQuery" match="Keys">
                <ParticipantObjectQuery>
                        <xsl:value-of select="."/>
                </ParticipantObjectQuery>
        </xsl:template>
        <xsl:template mode="ParticipantObjectQuery" match="*"/>
        <!-- ========================================== -->
        <!-- Discard everything else                    -->
        <!-- ========================================== -->
        <xsl:template match="*"/>
</xsl:stylesheet>
```

Appendix H: Required Registry Initialization and Schema

H.1: Initialization

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 ExtrinsicObjectto 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.

H.2: Schema

An XML Schema has been defined for XDS.

H.3: Location

These resources be found on the IHE website:

http://www.ihe.net

Select *Resources* tab (one of the tabs listed across the top of the page)

Select *Integration Profiles – Supplemental Information*

Navigate to the XDS section

Appendix I: Required Initialization of the Affinity Domain

This initialization supports the operation of the Registry Adaptor. The following information must be provided by the Affinity Domain administrator and loaded into the Registry Adaptor. This supports the functionality specified for the Registry Adaptor in section 3.14.4.1.2.11. 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.

Appendix J: Example Submissions and Query Results

Sample submissions, queries, and query results may be found on the IHE website at:

http://www.ihe.net

Select *Resources* tab (one of the tabs listed across the top of the page)

Select *Implementors Tools*

Navigate to the XDS section

Appendix K: XDS Security Environment

This Appendix expands on the summary provided in the XDS Volume 1 specification (ITI TF-1: Appendix H).

The XDS operations assume that a suitable security and privacy environment has been established. Almost all of the relevant threats will be managed by agreements, policies, and technologies that are external to the XDS transactions. The few that affect the XDS transactions will be managed by generic security mechanisms that are not unique to XDS. The threats and security objectives that must be addressed are described in sections 1 and 2 below. Only a few of these have issues that are unique to the XDS application.

Section 3 discusses these few threats and objectives in terms of the agreements and policies that need to be established to create a suitable environment for XDS. Establishing these agreements often involves business agreement discussions that are part of establishing the XDS affinity group. These agreements are necessary because the exchange of documents implies agreeing to the delegation of responsibility for maintaining the security of these documents and for providing the necessary audit and record keeping facilities.

K.1: Security Environment

3.24.8 Threats

Specific threats to the overall XDS system are listed below. These threats are identified using the Common Criteria nomenclature defined by ISO 17799. Most of these are mitigated by policies, procedures, and technologies that are not unique to XDS and do not require any special XDS considerations. Many of these mitigations do require that the parties within the XDS affinity group have agreement on details of how they will work together.

T.ADMIN_ERROR Improper administration may result in defeat of specific security features.

T.ADMIN_ROGUE Authorized administrator's intentions may become malicious resulting in TSF data to be compromised.

T.AUDIT_CORRUPT A malicious process or user may cause audit records to be lost or modified, or prevent future records from being recorded by taking actions to exhaust audit storage capacity, thus masking an attacker's actions.

T.CONFIG_CORRUPT A malicious process or user may cause configuration data or other trusted data to be lost or modified.

T.DISASTER System or network may failure due to disaster (e.g. fire, earthquake).

T.DOS A malicious process or user may block others from system resources via a resource exhaustion denial of service attack.

T.EAVESDROP A malicious process or user may intercept transmitted data inside or outside of the enclave. Some of the XDS environments are not concerned with eavesdrop exposure. They may employ external protective mechanisms such as physical network security or VPNs to protect against eavesdropping.

T.HARDWARE Hardware may malfunction.

T.IMPROPER_INSTALLATION XDS components may be delivered, installed, or configured in a manner that undermines security.

T.INSECURE_START Reboot may result in insecure state of the operating system.

T.INTRUSION Malicious software (e.g. virus) may be introduced into the system.

T.MASQUERADE A malicious process or user on one machine on the network may masquerade as an entity on another machine on the same network.

T.OBJECTS_NOT_CLEAN Systems may not adequately remove the data from objects between usage by different users, thereby releasing information to a user unauthorized for the data. This also includes swapping hard disk with PHI during service and repair.

T.POOR_DESIGN Unintentional or intentional errors in requirement specification, design or development of the TOE components may occur.

T.POOR_IMPLEMENTATION Unintentional or intentional errors in implementing the design of the XDS environment may occur.

T.POOR_TEST Incorrect system behavior may result from inability to demonstrate that all functions and interactions within the XDS operation are correct.

T.REPLAY A malicious process or user may gain access by replaying authentication (or other) information.

T.SPOOFING A hostile entity may masquerade itself as part of the XDS affinity group and communicate with authorized users who incorrectly believe they are communicating with authorized members.

T.SYSACC A malicious process or user may gain unauthorized access to the administrator account, or that of other trusted personnel.

T.UNATTENDED_SESSION A malicious process or user may gain unauthorized access to an unattended session.

T.UNAUTH_ACCESS Unauthorized access to data by a user may occur. This includes access via direct user interaction with the device, access via network transactions, and access via removable electronic and printed media.

T.UNAUTH_MODIFICATION Unauthorized modification or use of XDS attributes and resources may occur.

T.UNDETECTED_ACTIONS Failure of the XDS components to detect and record unauthorized actions may occur.

T.UNIDENTIFIED_ACTIONS Failure of the administrator to identify and act upon unauthorized actions may occur.

T.UNKNOWN_STATE Upon failure of XDS components, the security of the XDS environment may be unknown.

T.USER_CORRUPT User data may be lost or tampered with by other users.

3.24.9 Security and Privacy Policy

There are a wide variety of security and privacy regulations established by law and regulation. These are interpreted and extended to create individual enterprise policies. This equipment will be installed into a variety of enterprises that are subject to a variety of laws and regulations. The XDS environment will provide support for the common aspects of these enterprise policies. The policy statements whose enforcement must be provided by the XDS security mechanisms are:

P.ACCOUNT The users of the system shall be held accountable for their actions within the system.

P.AUTHORIZATION The system must limit the extent of each user's abilities in accordance with the TSPP. (See P.PATIENT_CARE)

P.AUTHORIZED_USERS Only those users who have been authorized to access the information within the system may access the system. (See P.PATIENT_CARE)

P.CRYPTOGRAPHY The system shall use standard approved cryptography (methods and implementations) for key management (i.e., generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e., encryption, decryption, signature, hashing, key exchange, and random number generation services).

P.DECLARATIVE_SECURITY The system shall allow the administrator to define security related rules. Examples include defining access control policies and password expiration restriction.

P.I_AND_A All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.

P.OBJECTAUTHORIZATION The XDS components must enforce the policy regarding how authorization is established for protected objects. The policy determines how access control and other policies are enforced. (This is often considered part of P.Authorization, but in the XDS context it may make sense to consider this as a separate policy.)

P.PATIENT_CARE The security and privacy measures should not prevent patient care. In particular, there should be emergency bypass mechanisms to override security when necessary to provide patient care.

P.SYSTEM_INTEGRITY The system must have the ability to periodically validate its correct operation and, with the help of Administrators, Backup and Restore Operators, and Service Personnel, it must be able to recover from any errors that are detected.

P.TRACE The primary method for enforcing the security and privacy policy is the use of auditing. The XDS components must have the ability to review the actions of individuals. The XDS environment must provide sufficient audit information to external audit and monitoring systems to permit the review of actions of individuals by that other system.

P.TRUSTED_RECOVERY Procedures and/or mechanisms shall be provided to assure that, after a system failure or other discontinuity, recovery without a protection compromise is obtained

P.VULNERABILITY_SEARCH The XDS environment must undergo an analysis for vulnerabilities beyond those that are obvious.

3.24.10 Security Usage Assumptions

Assumptions of the use of the XDS environment:

A.PHYSICAL It is assumed that appropriate physical security is provided within the domain for the value of the IT assets and the value of the stored, processed, and transmitted information.

A. AUDIT_REVIEW It is assumed that there will be audit repository and review services provided that can accept audit information from the XDS components in real time.

A.OPERATION It is assumed that networks, firewalls, etc. are deployed and maintained to meet appropriate network security levels.

A.PERSONNEL It is assumed that the organization can assure IT user & other workforce personal integrity/trustworthiness.

A.PKI It is assumed that there will be a facility to provide signed certificates as needed for node and user authentication. The key management maybe done manually or automatically depending on the availability of appropriate technology.

K.2: Security Objectives

This section defines the security objectives for the XDS environment. These objectives are suitable to counter all identified threats and cover all identified organizational security policies and assumptions. Common Criteria nomenclature is used. The XDS component security objectives are identified with “O.” appended to at the beginning of the name and the environment objectives are identified with “OE.” appended to the beginning of the name.

3.24.11 XDS Component Security Objectives

O.ACCESS The XDS components will ensure that users gain only authorized access to it and to the resources that it controls. (See O.EMERGENCY_BYPASS)

O.ACCESS_HISTORY The XDS components will display information (to authorized users) related to previous attempts to establish a session.

O.ADMIN_ROLE The XDS components will provide separate administrator roles to isolate administrative actions. These include a General Administrator role, a Backup and Restore Operator role, a Cryptographic Administrator role, and a Service Personnel role. Additional roles can be defined. These roles are collectively called Administrators.

O.ADMIN_TRAINED The XDS components will provide authorized Administrators with the necessary information for secure management and operation.

O.AUDIT_GENERATION The XDS components will provide the capability to detect and create records of security and privacy relevant events associated with users. The XDS components will reliably transmit this information to the central audit repository, and provide reliable local storage of events until the central audit repository has confirmed receipt. (See OE.AUDIT_REVIEW)

O.AUDIT_PROTECTION Each XDS component will provide the capability to protect audit information within its scope of control.

O.AUDIT_REVIEW If an external central audit repository is not part of the environment, the components will be configured to provide limited capability to analyze and selectively view audit information. (See OE.AUDIT_REVIEW)

O.CONFIG_MGMT All changes to the components and its development evidence will be tracked and controlled.

O.DECLARATIVE_SECURITY The components will allow security functions and access control to be defined by the authorized administrator.

O.DISASTER_RECOVERY The components should allow the authorized Administrators to perform backup and restore of electronic data, and rapid configuration and reconfiguration of device operation. In addition, the TOE should support administrative procedures to restore operation after disasters that may have substantially destroyed portions of the hospital operation and where substitute temporary systems are in place.

O.DISCRETIONARY_ACCESS The components will control accesses to resources based upon the identity of users and the role of users. (See O.EMERGENCY_BYPASS)

O.DISCRETIONARY_USER_CONTROL The components will allow authorized users to specify which resources may be accessed by which users and groups of users. (See O.EMERGENCY_BYPASS)

O.EMERGENCY_BYPASS The XDS components should allow access to any secured data during a declared medical emergency.

O.ENCRYPTED_CHANNEL Based on the environmental policies, encryption may be used to provide confidentiality of protected data in transit over public network.

O.INSTALL The XDS components will be delivered with the appropriate installation guidance in the form of installation manuals and training to establish and maintain component security.

O.INTRUSION_DETECTION The XDS components will ensure intrusion of malicious software (e.g. virus) is detected.

O.MANAGE The XDS components will provide all the functions and facilities necessary to support the authorized Administrators in their management of the security of the TOE.

O.PROTECT The XDS components will provide means to protect user data and resources.

O.RECOVERY Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.

O.REMOTE_SERVICE The XDS components will provide the means for remote service without sacrificing security or privacy policy.

O.RESIDUAL_INFORMATION The XDS components will ensure that any information contained in a protected resource is not released when the resource is reallocated. Information on permanent media such as hard disk shall be secured during service and repair.

O.RESOURCE_SHARING No user will block others from accessing resources.

O.SELF_PROTECTION Each XDS component will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.

O.TRAINED_USERS The XDS environment will provide authorized users with the necessary guidance for secure operation.

O.TRUSTED_PATH **The TOE** will provide a means to ensure users are not communicating with some other entity pretending to be the TOE. This covers entity authentication. (See O.USER_AUTHENTICATION.)

O.TRUSTED_SYSTEM_OPERATION The XDS components will function in a manner that maintains security.

O.USER_AUTHENTICATION The XDS components will verify the claimed identity of the interactive user. (See O.ENTITY_AUTHENTICATION.)

O.USER_IDENTIFICATION The XDS components will uniquely identify the interactive users.

3.24.12 Environment Security Objectives

OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the XDS environment and the value of the stored, processed, and transmitted information.

OE.AUDIT_REVIEW There may be an audit repository and review service provided that can accept audit information from the XDS environment in real time. This facility will provide review and analysis functions. (See O.AUDIT_GENERATION, O.AUDIT_REVIEW)

OE.OPERATION Networks, firewalls, etc. are deployed and maintained to meet appropriate network security levels.

OE.PERSONNEL Assure IT user & other workforce personal integrity/trustworthiness.

OE.PKI There will be a facility to provide signed certificates as needed for node and user authentication.

K.3: Functional Environment

The XDS can be modelled as having four different organizations that have a delegated responsibility relationship where each organization has a different functional responsibility. In some configurations a single organization is responsible for two or more of these functions, which makes delegation much easier. This section discusses the major areas that must be solved.

The four functions are:

Creator – This functional organization has created the PHI and is legally responsible to the patient and others for providing healthcare and for protecting this data.

Repository – This functional organization is responsible for providing access to persistent documents to readers. The creator has delegated responsibility to the repository to provide adequate protection for a subset of the PHI. This subset is called the document.

Registry - This functional organization is responsible for providing query services to readers. The creator has delegated responsibility to the to the registry to provide adequate protection for a subset of the PHI. This subset is called the metadata.

Reader – This functional organization is providing healthcare services that make use of data that is contained in the metadata and the documents.

There are three levels of difficulty in delegation.

"**Trivial**" delegation is that where it is not necessary to delegate the responsibility for implementing the threat mitigation. In those cases it does not matter whether the organizations have the same policy or mitigations. For example, if the registry provides adequate mitigation

against the threat of disaster, it need not be concerned with the disaster related policies of the reader.

"**Easy**" delegation is that where the two organizations have the equivalent policies. In those cases there is an initial difficult phase of discovering that the policies are the same and evaluating that the mitigation strategies are acceptable. This results in a simple binary decision to approve or disapprove a business relationship permitting the exchange of data. With the exception of the three policy classes described as "hard" below, the details of policies are likely to differ, but the goals are sufficiently uniform that a simple business decision can be made.

For the "easy" delegation, the IHE transactions must provide adequate mitigations for the threats so that the business decision to exchange data can be made based simply on review of the partners policies and mitigations. This means that some IHE transactions will have additional security requirements attached. For example, encryption to avoid the threat of eavesdropping may be required. These requirements are not unique to XDS and will be able to use standardized security features like TLS and VPN tools. These requirements may be significantly different from the usual practice within an enterprise, because of the differences in the environment.

"**Hard**" delegation is that where the two organizations have different policies or inconsistent/incompatible mitigation strategies. These are likely to occur for the following policies, where organizations often disagree on the details of the policy goals, and where policies often change:

P.Authorization – The authorized access policies and authorized modification policies often differ, and are often subject to change. The changes that occur are often at a detailed level, e.g. access rights to a particular patient information may change. This means that either there is an agreed mechanism to propogate changes, or an acceptance that policy changes may not be enforced, or there will be restrictions on the data exchange to avoid delegating responsibility for data that is subject to change.

P.Account and P.trace – The policies for accountability and traceability often differ. These are much less subject to change, but it is often difficult to reconcile delegation when these policies differ. This will be an especially difficult issue for repository and registry functions that support multiple different creator organizations.

P.ObjectAuthorization – The policies regarding creation and modification of access rights often differ.

In addition, any of the policy and threat mitigations may be determined to be unacceptable by creator, registry, or repository. In the simple situation where there are only four real world participants this simply means that there is no business relationship. In the more complex world where the registry or repository are in many relationships with many creators and readers it introduces a serious problem. Either the registry and repository must limit its relationship to that small set of creators and readers that mutually accept all the policies and mitigations of all the

other organizations, or there must be a mitigation strategy so that creators can restrict delegations by the registry and repository to only those readers that have policies and mitigations that are acceptable to the creator.

Mitigations for differences include the following:

Limit the data exchange to that data where the differences are not significant. For example, highly sensitive data like psychiatric notes might not be shared, while relatively insignificant data like allergy information is shared.

Provide a revocation mechanism to deal with policy changes, so that future delegations can be prohibited. It is often impractical to revoke past delegations because the PHI has already been disclosed. But the revocation mechanism can stop further delegation from taking place. This revocation mechanism must be part of the P.Authorization and P.ObjectAuthorization policies and must be mutually acceptable for this mitigation to be effective.

Trusted third party inspections and audits can sometimes deal with reconciliation of differences in P.Account and P.Trace.

An "approved delegation" list identifying acceptable and unacceptable creator/reader pairs can mitigate the repository and registry issues when the reader has incompatible policies with the creator. This does require the creator to accept the approved delegation policy and implementation of the repository and registry, but it reduces the combinatorial explosion of policy combinations between creators, repositories, registries, and readers into a linear growth in complexity.

The "approved delegation" may go further into identification of persons, but this is only a viable path when all parties have policies the easily support delegation of personal responsibility. Persons are usually required to comply with organizational policies, and organizations generally use roles rather than persons to establish policies. The often viable exception is the special case of the "deny access to person X". This can be a viable means of dealing with situations involving a conflict of interest. This kind of access denial may be applicable to just a particular subset of the PHI exchanged, (e.g. denying access to an ex-spouse).

These mitigations do not directly change the technical requirements for the XDS transactions. They are policy decisions that may affect how particular actors are configured. The implementation of XDS actors will need to be aware that this kind of site-specific configuration management and policy control will be routinely required.

Appendix L: Relationship of Document Entry Attributes and Document Headers

XDS Document Entry attributes, placed in the XDS Document Registry by Document Sources, may be derived from header data present in the document content. Although the XDS Integration Profile does not mandate a strict relationship, this appendix illustrates sample mappings of XDS Document Entry attributes to header fields of some standard document formats. This relationship does not imply that values are mapped or copied directly as transformations may be needed between conventions in the EHR-CR and EHR-LR (e.g. vocabulary mappings).

Table L-1 Relationship of XDS Document Attributes to Document header fields

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
patientId	levelone >clinical_document_header >>patient >>>id mapped into XDS Affinity Domain patient id domain	ClinicalDocument >recordTarget >>patientRole >>>id mapped into XDS Affinity Domain patient id domain	Class: EHR_EXTRACT attribute: subject_of_care[1]: II mapped into XDS Affinity Domain patient id domain
serviceStartTime	levelone >clinical_document_header>>patient_encounter >>>encounter_tmr	ClinicalDocument >documentationOf >>event >>>effectiveTime low=	Class: CLINICAL_SESSION attribute: session_time[1]: IVL<TS>
serviceStopTime	levelone >clinical_document_header >>patient_encounter >>>encounter_tmr	ClinicalDocument >documentationOf >>event >>>effectiveTime high=	

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
classCode	Inferred from levelone >clinical_document_header >>document_type_cd RT= EX=	Inferred from ClinicalDocument >code codeSystem= code=	Class COMPOSITION Attribute: to be added.
classCodeDisplayName	Inferred from levelone >clinical_document_header >>document_type_cd DN=	Inferred from ClinicalDocument >code codeSystem= code=	
practiceSettingCode	levelone >clinical_document_header >>patient_encounter >>>practice_setting_cd V= S=	Inferred from ClinicalDocument >code codeSystem= code=	(need input from CEN TC 251)
practiceSettingCode DisplayName	levelone >clinical_document_header >>patient_encounter >>>practice_setting_cd DN=	Inferred from ClinicalDocument >code codeSystem= code=	

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
healthcareFacility TypeCode	Inferred from levelone >clinical_document_header >>patient_encounter >>>practice_setting_cd V= S=	Inferred from ClinicalDocument >code codeSystem= code=	Class CLINICAL_SESSION attribute: healthcare_facility[0..1]: II
healthcareFacility TypeCodeDisplay Name	Inferred from levelone >clinical_document_header >>patient_encounter >>>practice_setting_cd DN=	Inferred from ClinicalDocument >code codeSystem= code=	
availabilityStatus	N/A (Generated and maintained by the Registry)	N/A (Generated and maintained by the Registry)	N/A (Generated and maintained by the Registry)
uniqueId	levelone >clinical_document_header >>id	ClinicalDocument >id	Class RECORD_COMPONENT attribute: rc_id[1]: II
typeCode	levelone >clinical_document_header >>document_type_cd RT= EX=	ClinicalDocument >code codeSystem= code=	Class RECORD_COMPONENT attribute: meaning[0..1]: CV

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
typeCodeDisplay Name	levelone >clinical_document_header >>document_type_cd DN=	ClinicalDocument >code displayName=	
formatCode		ClinicalDocument >typeId	Class EHR_EXTRACT attribute: rm_id[1]: String
eventCode	Inferred from levelone >clinical_document_header >>document_type_cd RT= EX=	Inferred from ClinicalDocument >code codeSystem= code=	(need input from CEN TC 251)
eventCodeDisplay Name	Inferred from levelone >clinical_document_header >>document_type_cd RT= EX=	Inferred from ClinicalDocument >code codeSystem= code=	(need input from CEN TC 251)
title	Inferred from levelone >clinical_document_header >>document_type_cd	ClinicalDocument >title	Class: RECORD_COMPONENT attribute: name[1]: TEXT

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
	DN=		
authorInstitution	levelone >clinical_document_header >>originating_organization >>>organization	ClinicalDocument >author >>assignedAuthor >>>representedOrganization >>>>name	Class CLINICAL_SESSION attribute: healthcare_facility[0..1]: II
authorPerson	levelone >clinical_document_header >>originator >>>person	ClinicalDocument >author >>assignedAuthor >>>assignedAuthorChoice >>>>person	Class: COMPOSITION attribute: composer[0..1]: II
legalAuthenticator	levelone >clinical_document_header >>legal_authenticator >>>person	ClinicalDocument >legalAuthenticator >>assignedEntity >>>person	Class FUNCTIONAL_ROLE (association from class ATTESTATION) attribute: performer[1]: II
URI	N/A	N/A	N/A
parentDocumentRelationship	levelone >clinical_document_header >>document_relationship >>>document_relationship.type_cd	ClinicalDocument >relatedDocument typeCode=	IN THE CASE OF REPLACEMENT Class: AUDIT_INFO attribute: revision_status CS_REV_STAT IN THE CASE OF ADDENDUM or

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
			TRANSFORM Class LINK attribute nature: CV
parentDocumentId	levelone >clinical_document_header >>document_relationship >>>related_document >>>>id	ClinicalDocument >relatedDocument >>parentDocument >>>id	IN THE CASE OF REPLACEMENT attribute: previous_version[0..1] : II This attribute uniquely identifies the RECORD_COMPONENT of which the current RECORD_COMPONENT is a revision (null for the first ever version). IN THE CASE OF ADDENDUM or TRANSFORM Class LINK Attribute: target[1]: II
confidentialityCode	levelone >clinical_document_header >>confidentiality_cd RT= EX=	ClinicalDocument >confidentialityCode	Class RECORD_COMPONENT attribute: sensitivity[1]: CS_SENSITIVITY
languageCode	xml:lang attribute	ClinicalDocument >relatedDocument typeCode=	This attribute is a property of all text data types in CEN,

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
			and so we have not defined a separate overall language to govern the whole document. It might be reasonable to assume that the natural language used for the name attribute is considered to be a reasonable guide to the value of this attribute.
patientId AssignBySource	levelone >clinical_document_header >>patient >>>person >>>>id	ClinicalDocument >recordTarget >>patientRole >>>id	Class: EHR_EXTRACT attribute: subject_of_care[1]: II
patientInfo AssignBySource	levelone >clinical_document_header >>patient >>>person >>>>person_name	ClinicalDocument >recordTarget >>patientRole >>>patientPatient >>>>name	
size	N/A Total length of submitted document.	N/A Total length of submitted document.	N/A Total length of submitted document.
hash	N/A Hash of submitted document.	N/A Hash of submitted document.	N/A Hash of submitted document.

Attribute	CDA R1-2000	CDA R2 Draft Aug 2004	EHRCOM
entryUUID	N/A Generated by registry	N/A Generated by registry.	N/A Generated by registry.

Appendix M: Using Patient Demographics Query in a Multi-Domain Environment

M.1: HL7 QBP^Q22 Conformance Model

The HL7 Find Candidates Query (QBP^Q22) defines a patient demographics query between a client application and an MPI system (HL7 V2.5, Page 3-64). This implies that the server maintains a master record of the patient demographics, but may know a number of patient identifiers from other domains.

In the QBP^Q22 Conformance Statement, QPD-8 (What Domains Returned) is defined as "the set of domains for which identifiers are returned in PID-3" (HL7 V2.5, Page 3-66, second table). Note that this field does not cite "demographics information in some domains", but about "identifiers issued in some domains", and explicitly specifies that these identifiers are returned in PID-3 (Patient ID List).

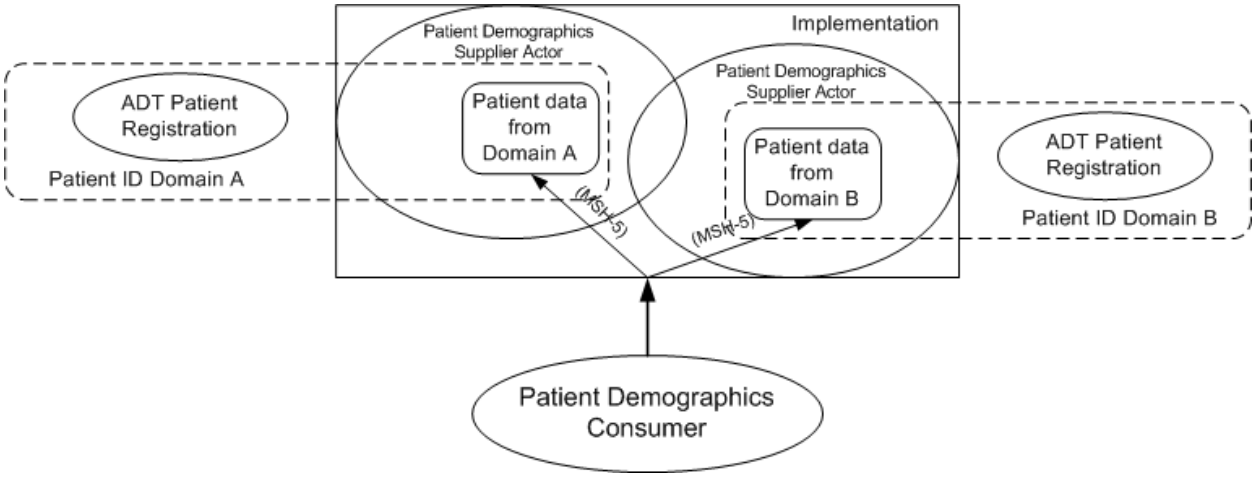
In the example following the Conformance Statement in HL7 2.5, three patient records are included in the query response; each returned patient record includes two identifiers in PID-3 (domains METRO HOSPITAL and SOUTH LAB) as requested in the query. However, one set of demographic information is returned in the remainder of the PID segment. The example does not illustrate or assume a mechanism for returning multiple sets of demographic information.

Thus it appears that QBP^Q22 is not intended to provide a way to issue a single query for patient demographics maintained in multiple different patient registration systems (domains).

M.2: IHE PDQ Architecture

In the PDQ Integration Profile, the supplier is characterized as a Patient Demographics Supplier. The supplier is not assumed nor required to be an MPI system. It may be holding information from only a single patient identification domain, or may instead hold information from multiple identification domains.

The latter case would apply if, for example, the Patient Demographics Supplier is grouped with an actor accepting ADT feeds from multiple patient registration systems in different domains. Equivalently, the Patient Demographics Supplier Actor (or some other Actor with which it is grouped) may manage a set of patient demographics sources, but is not expected to cross-reference them (as a PIX Actor or an MPI system). A conceptual model embracing both multi-domain concepts is shown in the following picture.



Because of the definition of QBP^Q22, it must be determined which patient demographics source a QBP^Q22 query is asking for, before any processing of the query request can proceed. The identification of a need for such determination is the key difference between the IHE PDQ transactions and the original HL7 QBP^Q22 definitions.

Three obvious alternatives exist for determining the patient demographics source.

6. The supplier advertises different application entities for each of the patient demographics sources it manages. By addressing its query to a particular application entity in *MSH-5-Receiving Application*, the consumer explicitly selects a source it is asking for.
7. The consumer is required to populate PID-3.4 in QPD-3 (Query Parameter) with the domain name administered by the corresponding source (patient identifier domain) it is asking for.
8. The consumer includes in QPD-8 (What Domains Returned) the domain name of the corresponding patient information source it is asking for.

In selecting among these alternatives for the PDQ Profile, IHE-ITI took into account the need to constrain the current HL7 QBP^Q22 definition while maintaining the integrity of the HL7 standard query and at the same time to model the IHE PDQ Profile properly to satisfy its real-world purpose. Based on these considerations, alternative 1 is the best selection, although alternative 2 is acceptable. Alternative 3 is not acceptable because it violates the definition of QPD-8 that is stated in the HL7 Standard.

M.3: Implementing PDQ in a multi-domain architecture

There are three possible approaches in using PDQ in a multi-domain environment:

1. Group the PDQ Patient Demographics Supplier Actor with a PIX Patient Identifier Cross-reference Manager Actor. This allows the use of QPD-8 to request ***patient identifiers*** from other domains to be returned in the demographics query response to the PDQ Patient Demographics Consumer.
2. Group the PDQ Patient Demographics Supplier Actor with a PIX Patient Identifier Cross-reference Consumer Actor. This allows the use of QPD-8 to request ***patient identifiers*** from other domains to be returned in the demographics query response to the PDQ Patient Demographics Consumer.
3. Group the PDQ Patient Demographics Consumer Actor with a PIX Patient Identifier Cross-reference Consumer Actor. This obliges the Patient Demographics Consumer to use separate query requests to obtain patient demographics information (PDQ query) and patient identifiers from the domains in which it is interested (PIX query).

Approach 3 is not recommended if Approach 1 or 2 is feasible. To require the Patient Demographics Consumer to issue a separate PIX query increases complexity and might not be permissible in the actual implementation architecture.

When Approach 1 or 2 is implemented, QPD-8 may be used by the Patient Demographics Consumer to ask for patient identifiers from domains other than the domain administered by the requested demographics source, exactly as defined in the HL7 QBP^Q22 Conformance Statement. If this use of QBP-8 is permitted by the implementation, it should be made clear that the patient demographics information returned comes from the patient demographics source that is associated with *MSH-5-Receiving Application*.

In Approach 2, note that the PDQ Patient Demographics Supplier is grouped with the PIX Patient Identifier Cross-reference Consumer. This combined actor will use a PIX Query to satisfy the request of the client from additional patient identifiers and return them in PID-3.

GLOSSARY

See IHE IT Infrastructure Technical Framework. Vol 1: Glossary.