Integrating the Healthcare Enterprise

IT Infrastructure Technical Framework

Volume 2a (ITI TF-2a) Transactions Part A – Sections 3.1 – 3.28

Revision 6.0 – Final Text

August 10, 2009

Contents

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 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 6.0 for Final Text, specifies the IHE transactions defined and implemented as of August 2009 . 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. Volume 1 (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.

Volumes 2a, 2b, and 2x of the IT Infrastructure Technical Framework provide detailed technical descriptions of each IHE transaction used in the IT Infrastructure Integration Profiles. Volume 3 contains content specification and specifications used by multiple transactions. These 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. For example, the following IHE Technical Framework(s) are some of those which 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 ITI TF-2a: 1.6.3.

1.2 Overview of IT Infrastructure Technical Framework Volumes 2a, 2b, and 2x, and 3

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. Section 3 is divided into two parts:

- Volume 2a: Sections 3.1 - 3.28 corresponding to transactions [ITI-1] through [ITI-28].
- Volume 2b: Sections 3.29 - 3.57 corresponding to transactions [ITI-29] through [ITI-57].

Volume 2x contains all appendices providing technical details associated with the transactions.

Volume 3, Section 4 contains specifications that are used by multiple transactions.

Volume 3, Section 5 contains Content Specifications.

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 ITI TF-2x: 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:

Lisa Spellman
Senior Director Informatics
230 East Ohio St., Suite 500
Chicago, IL USA 60611
Email: ihe@himss.org

1.7 Copyright Permission

2 Conventions

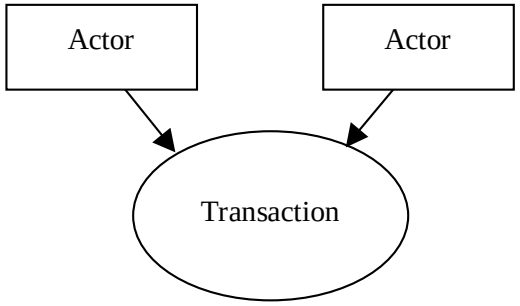
This document has adopted the following conventions for representing the framework concepts and specifying how the standards upon which the IHE IT Infrastructure Technical Framework is based should be applied.

2.1 The Generic IHE Transaction Model

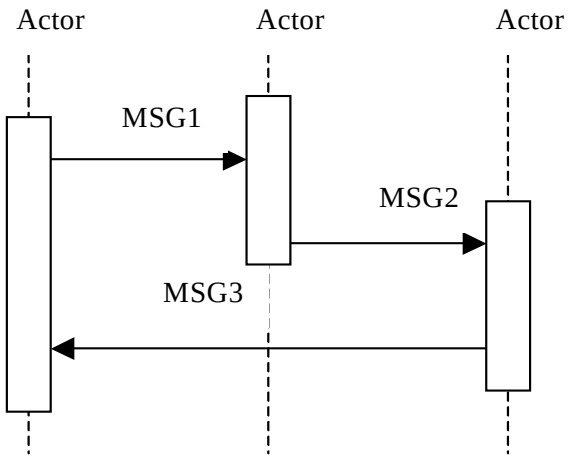
Transaction descriptions are provided in Section 3. In each transaction description, the actors, the roles they play, and the transactions between them are presented as use cases.

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 ITI TF-2x: Appendix Cfor 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.

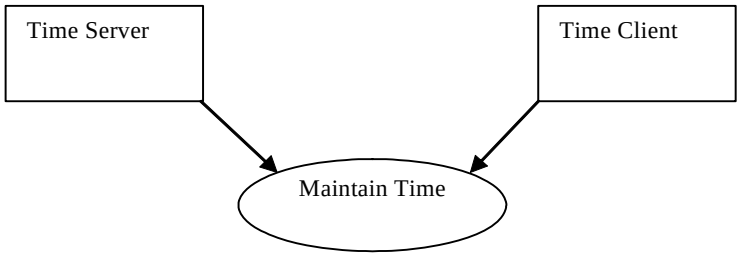
3.1 Maintain Time

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) RFC4330

3.1.4 Interaction Diagram

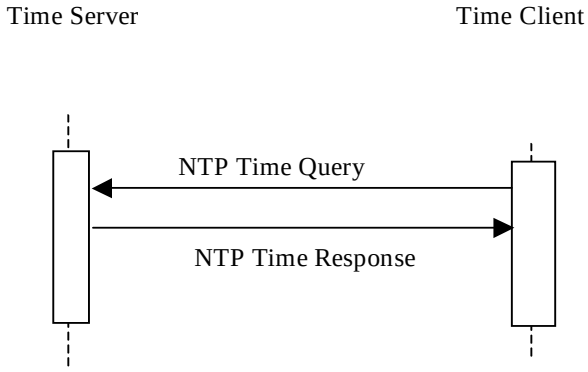


Figure 3.1.4-1. Maintain Time Messages

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) RFC4330 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

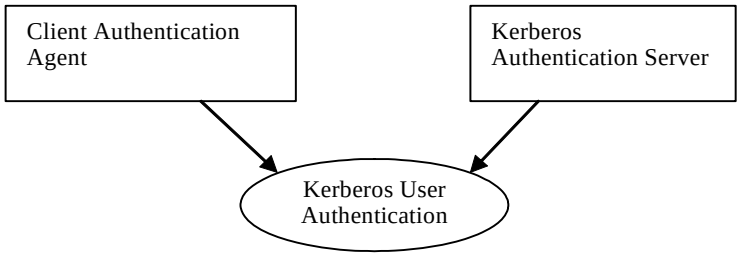
This section corresponds to Transaction ITI-2 of the IHE IT Infrastructure Technical Framework. Transaction ITI-2 is used by the Client Authentication Agent and Kerberos Authentication Server actors.

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.)

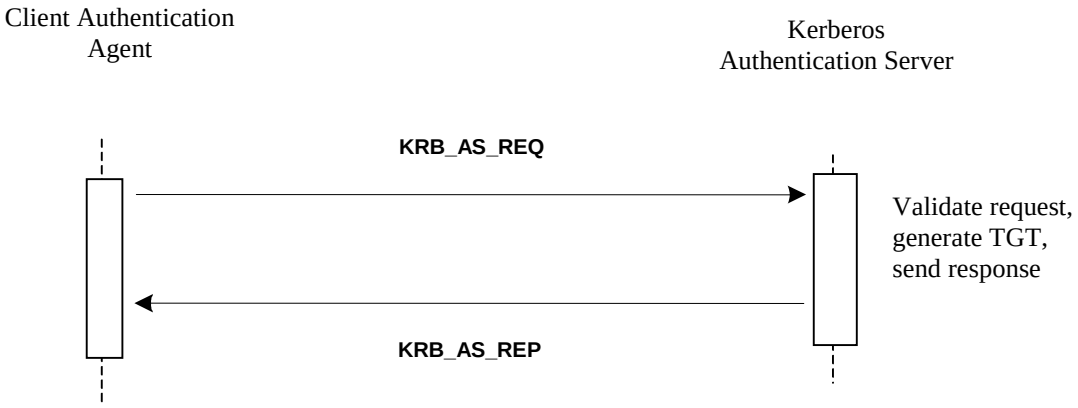


Figure 3.2.4-1. Get User Authentication Messages

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.)

3.2.6 Audit Record Considerations

The Client Authentication Agent shall produce the ATNA UserAuthenticated event for each Get Authentication [ITI-2] transaction with the EventTypeCode equal to Login or Failure as appropriate. If the application knows about logout, this shall produce a UserAuthentication event with the eventTypeCode of Logout. The UserName element shall be the Kerberos identity in the form of username@realm.

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110114, DCM, "UserAuthenticated")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV(110122, DCM, "Login") EV(110123, DCM, "Logout")
Source (1)			
Human Requestor (1)			
Destination (0)			
Audit Source (Client Authentication Agent) (1)			
Participant Object (0)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	the process ID as used within the local operating system in the local system logs.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110150, DCM, "Application")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	*U*	*not specialized*
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

480

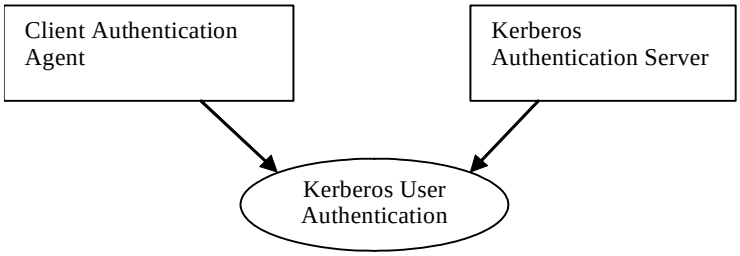
3.3 Get Service Ticket

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.

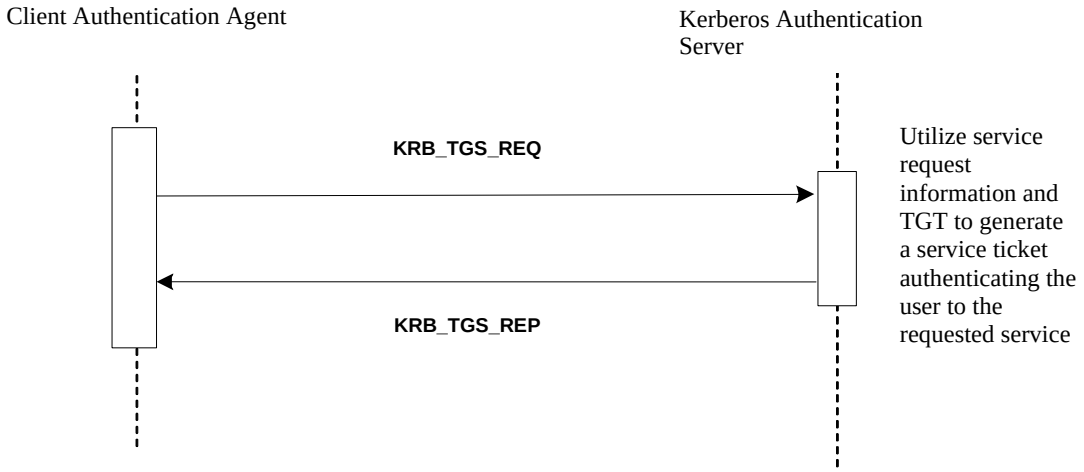
Actor: Kerberos Authentication Server. In RFC1510 this is called a Key Distribution Center (KDC).

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.

3.3.5 Security Considerations

The Get Service Ticket [ITI-3] Transaction is not required to log an ATNA UserAuthentication event in the case of successful communications. An ATNA UserAuthentication event shall be logged when the communications fails for the purpose of authentication failure.

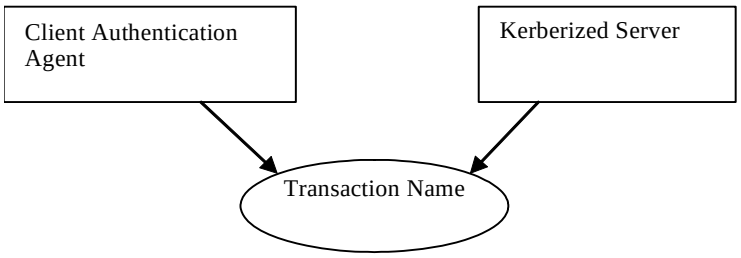
3.4 Kerberized Communication

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

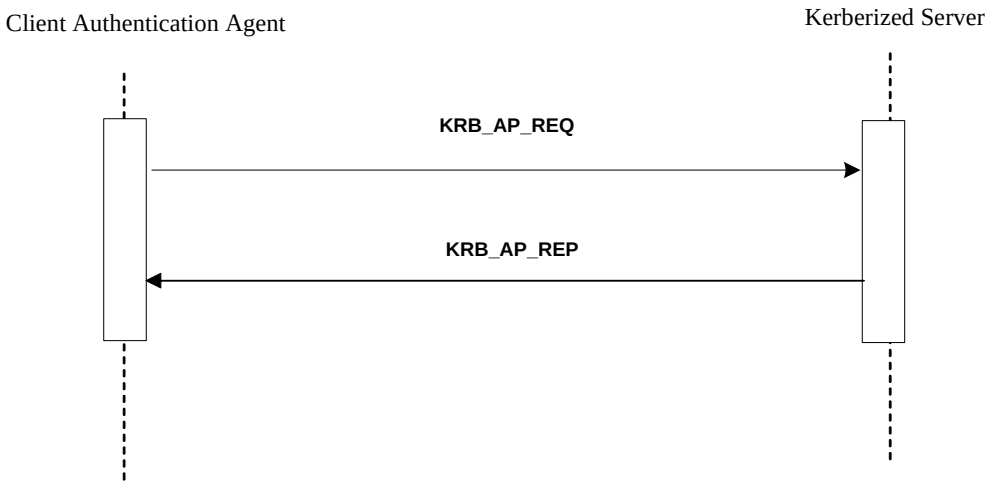


Figure 3.4-1 Kerberized Communications

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.

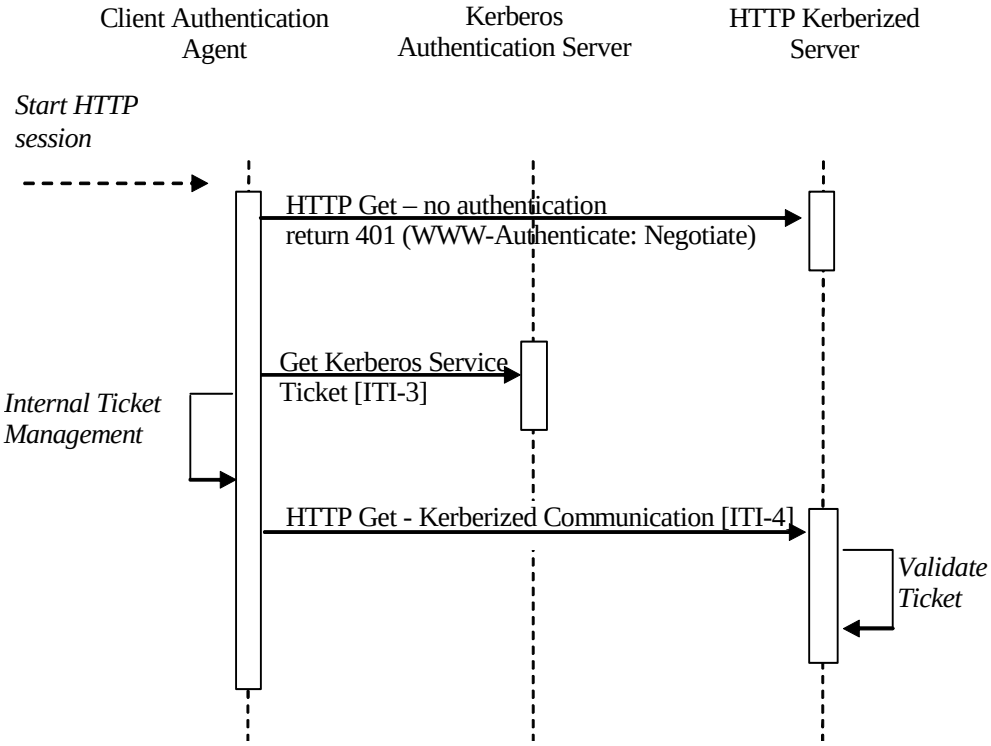


Figure 3.4-2 Kerberized HTTP

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.

3.4.5 Security Considerations

The Kerberized Communications [ITI-4] Transaction is not required to log an ATNA UserAuthentication event in the case of successful communications. An ATNA UserAuthentication event shall be logged when the communications fails for the purpose of authentication failure.

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

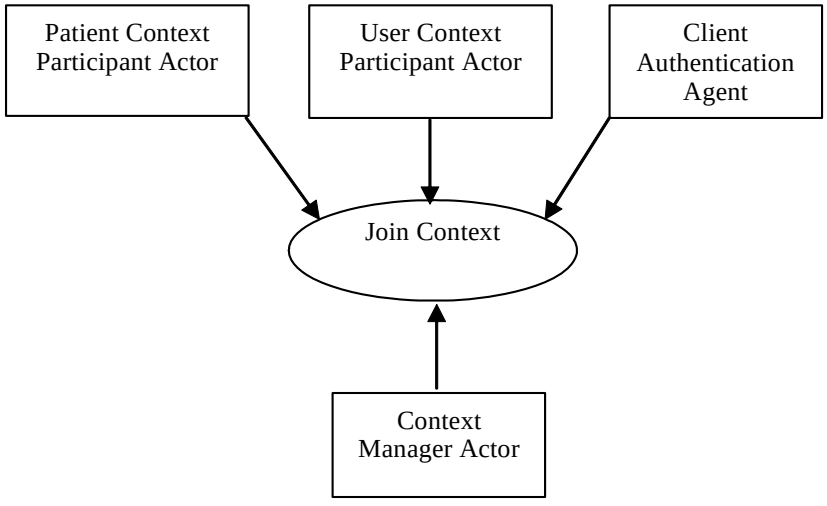
Any of the context participant actors using this Transaction (Patient Context Participant, User Context Participant, and Client Authentication Agent) may locate and join a context management session specific to the workstation on which the instigating user is interacting.

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.

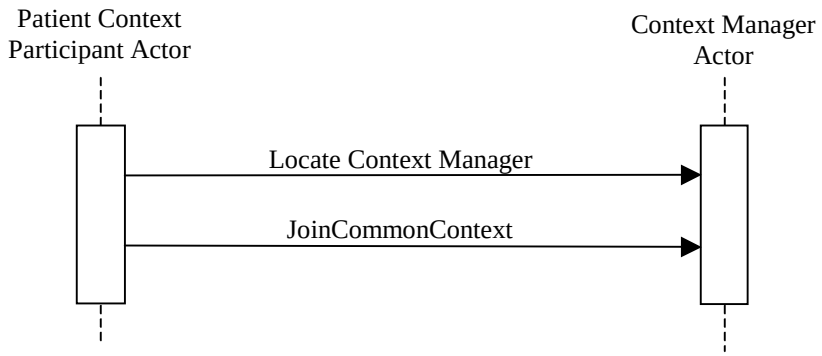


Figure 3.5-1 Patient Subject Join Context Interaction Diagram

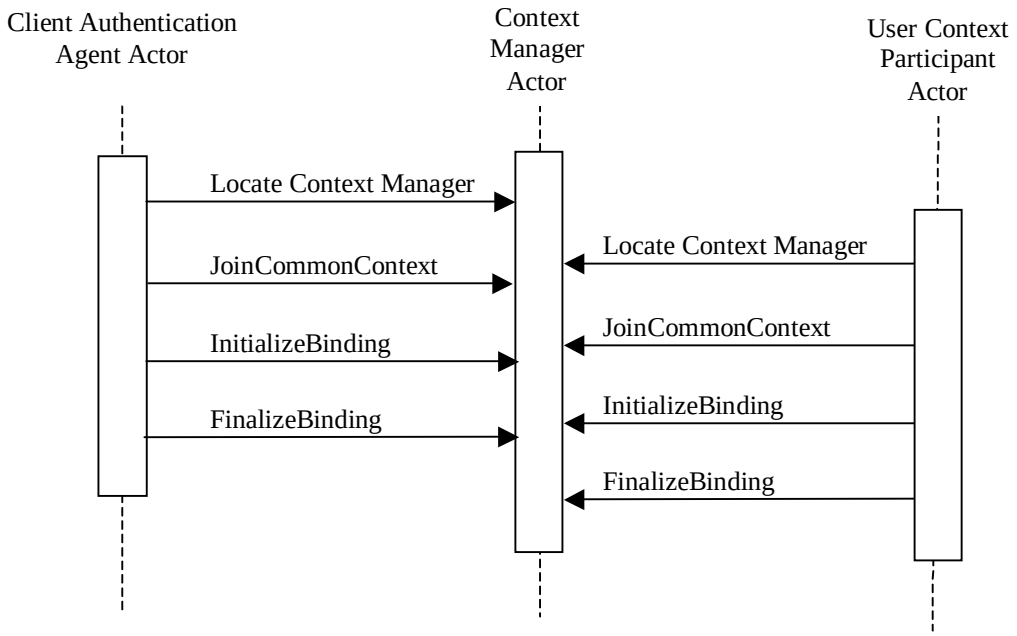


Figure 3.5-2 User Subject Join Context Interaction Diagram

665

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* 670 *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

675 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 680 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.

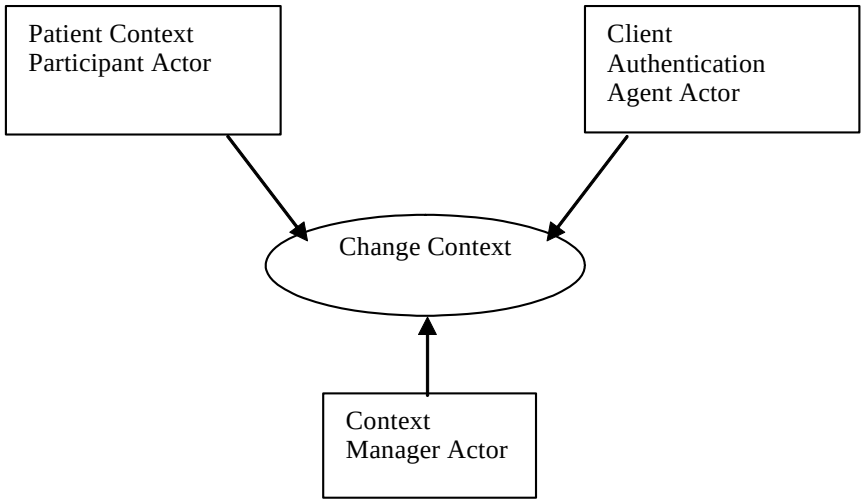
The Change Context Transaction is composed of multiple methods as defined by the *HL7 Context Management "CCOW" Standard*. There are two key characteristics to this transaction. The first is that the transaction has multiple phases consisting of instigating the change, surveying the other participants, 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-2a: 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-2x: 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



790

Actor: Client Authentication Agent

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

Actor: Patient Context Participant

795 **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.

800 ### 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

805 Subject Data Definitions

3.6.4 Interaction Diagram

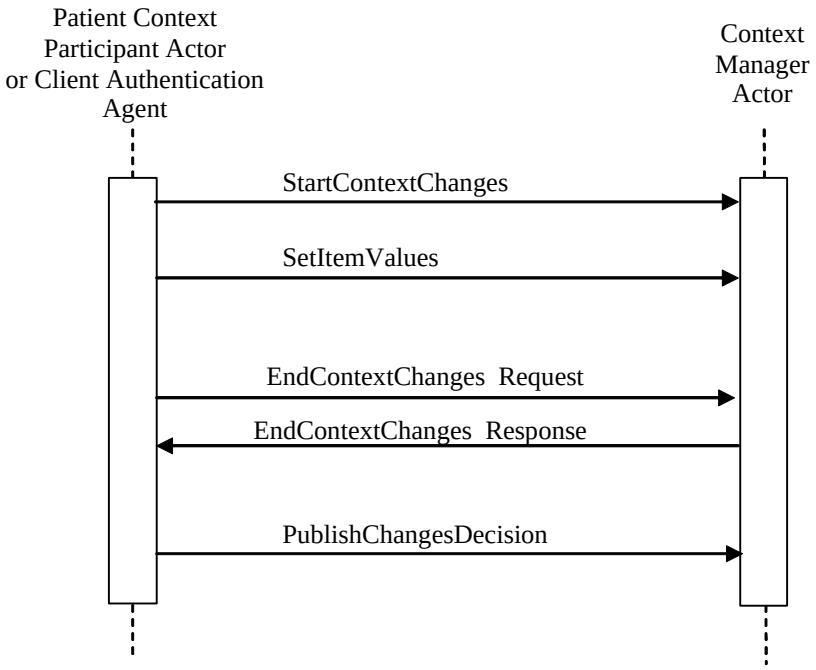


Figure 3.6-1 Change Context sequence

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-2a: 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-2a: 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-2a: 3.13.4.2 or ITI TF-2a: 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

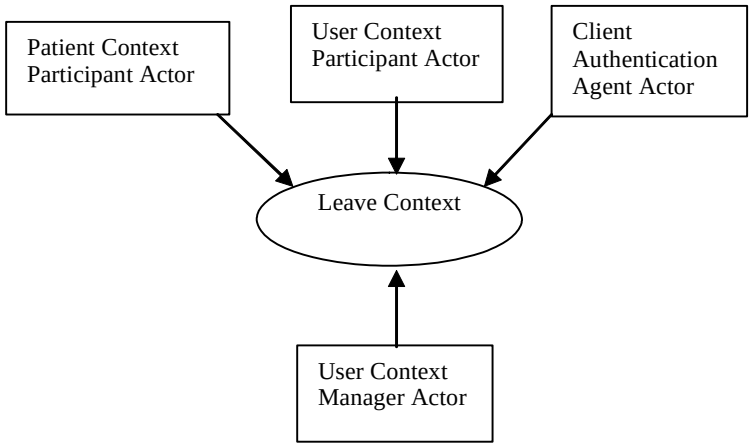
This section corresponds to Transaction ITI-7 of the IHE IT Infrastructure Technical Framework. Transaction ITI-7 is used by the Patient Context Participant, User Context Participant, Client Authentication Agent, and Context Manager Actors.

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

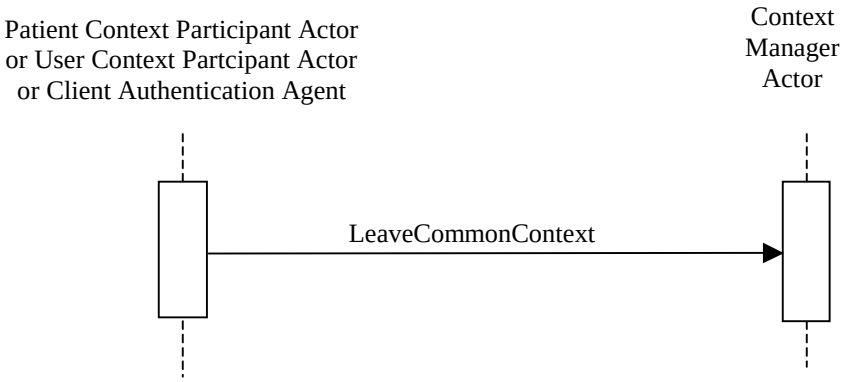
945 HL7 Context Management "CCOW" Standard, Version 1.4:

Technology and Subject Independent Architecture

Component Technology Mapping: ActiveX

Component Technology Mapping: Web

3.7.4 Interaction Diagram



950

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
955 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.

960 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*
965 *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.

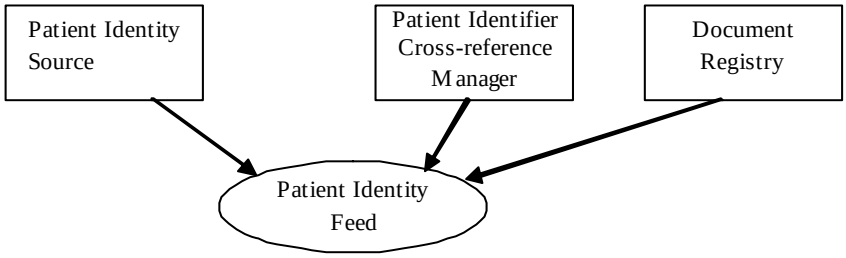
3.8 Patient Identity Feed

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

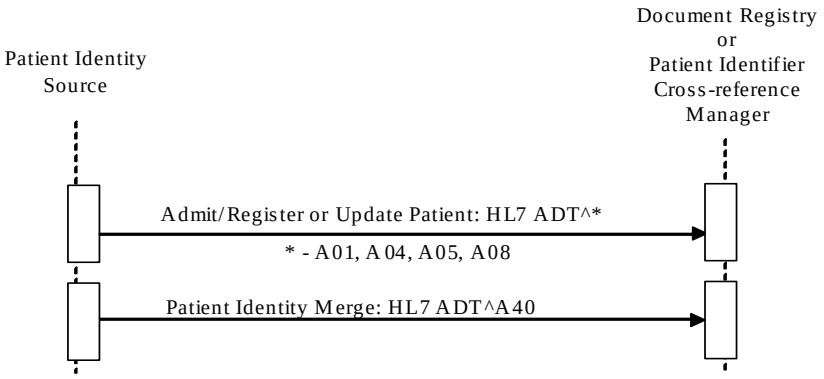


Figure 3.8-1 Patient Identity Sequence

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 ITI TF-2x: Appendix C and C.1.

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 ITI TF-2x: 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 ITI TF-2a: 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 ITI TF-2x: 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 one of the values of **A01**, **A04**, **A05** or **A08** as appropriate. The third component is optional; however, if present, it shall have the following value for each corresponding message type:

- ADT_A01 for A01 message type
- ADT_A01 for A04 message type
- ADT_A05 for A05 message type
- ADT_A01 for A08 message type

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 ITI TF-2x: C.1.4 "Common Segment Definitions" for the specification of this segment.

3.8.4.1.2.3 PID 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). The Patient Identity Source Actor shall use component PID-3.4 to convey the assigning authority (Patient Identification Domain) of the patient identifier.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.

3.8.4.1.2.4 PV1 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
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

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
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 ITI TF-2a: 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. The Patient Identifier Cross-reference Manager Actor shall only recognize (by configuration) a single Patient Identity Source Actor per domain. (See ITI TF-2a: 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 ITI TF-2a: 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 Patient 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 Domain. This identifier shall specify all 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 of the domain.
- Patient Identity Source Actor for the domain. This is expected to be the MSH-3 Sending Application and the corresponding MSH-4 Sending Facility fields in the HL7 ADT message. (Alternative identification schemes might include IP address of the Patient Identity Source Actor or Node Authentication if the Audit Trail and Node Authentication Integration Profile is used.)

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 triggers and data than what the XDS Document Registry needs for its operation. In particular, A08 – Update Patient Information, if received shall be ignored.

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 subcomponents 2 and 3 (the universal ID and the universal ID Type of Assigning Authority) of the Patient Identification Domain of the XDS Affinity Domain in PID-3.4 are not filled in the message (as described in ITI TF-2a: 3.8.4.1.2.3) the Document Registry shall fill subcomponents 2 and 3 of the Patient Identification Domain of the XDS Affinity Domain prior to storing the patient identity in the registry. The assigning authority information filled by the Document Registry is based on its configuration of the Patient Identification Domain of the XDS Affinity Domain (See ITI TF-2a: 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 XDS 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 of the XDS Affinity Domain. This identifier shall be specified with 3 components of the HL7 assigning authority (data type HD): namespaceID, universal ID and universal ID type. The universal ID shall be an ISO OID (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 ITI TF-2x: Appendix C and C.1.

The segments of the HL7 Merge Patient message listed below are required, and the detailed description of the message is provided in ITI TF-2a: 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 ITI TF-2x: 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 ITI TF-2x: 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 ITI TF-2x: 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 ITI TF-2a: 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 identifier and the issuing assigning authority. 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.

The Patient Identity Source Actor shall send the “old” patient identifier (to be merged) in MRG-1, with the identifier value in the component MRG-1.1 and the assigning authority in the component MRG-1.4. The Patient Identity Source Actor shall populate the same value of the assigning authority in PID-3.4, in the component MRG-1.4.

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.

3.8.4.2.2.5 PV1 Segment

PV1 segment shall be constructed as defined in ITI TF-2a: 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	O		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 ITI TF-2a: 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 ITI TF-2a: 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.

3.8.5 Security Considerations

3.8.5.1 Audit Record Considerations – Admit/Register or Update Patient

The Patient Admit/Register transactions (A01, A04, A05) and Update Patient Information (A08) transaction are to be audited as “Patient Record” events, as defined in table 3.20.6-1. The actors involved in the transaction shall create audit data in conformance with DICOM (Supp 95) “Patient Record”. The following tables show items that are required to be part of the audit record for these specific PIX transactions.

3.8.5.1.1 Patient Identity Source Actor audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110110, DCM, "Patient Record")
	EventActionCode	M	"C" (create) for A01, A04, A05 "U" (update) for A08
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-8", "IHE Transactions", "Patient Identity Feed")
Source (Patient Identity Source Actor) (1)			
Human Requestor (0..n)			
Destination (Patient Identifier Cross-reference Manager or Document Registry) (1)			
Audit Source (Patient Identity Source Actor) (1)			
Patient (1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identity Source Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifier Cross-reference Manager or Document Registry facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*M*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (person)
	ParticipantObjectTypeCodeRole	M	"1" (patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format..
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

1220

3.8.5.1.2 Patient Identifier Cross-reference Manager or Document Registry Actor audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110110, DCM, "Patient Record")
	EventActionCode	M	"C" (create) for A01, A04, A05 "U" (update) for A08
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-8", "IHE Transactions", "Patient Identity Feed")
Source (Patient Identity Source Actor) (1)			
Destination (Patient Identifier Cross-reference Manager or Document Registry) (1)			
Audit Source (Patient Identifier Cross-reference Manager or Document Registry) (1)			
Patient(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identity Source Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

	UserID	M	The identity of the Patient Identifier Cross-reference Manager or Document Registry facility and receiving application from the HL7 message; concatenated together, separated by the | character.

	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

1225

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (person)
	ParticipantObjectTypeCodeRole	M	"1" (patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

3.8.5.2 Audit Record Considerations – Patient Identity Merge (Merge Patient ID)

The Patient Identity Merge transaction (A40) is to be audited as a "Patient Record" event, as defined in Table 3.20.6-1. The source of the transaction shall create audit data in conformance with DICOM (Supp 95) "Patient Record". The following tables show items that are required to be part of the audit record for the Patient Identity Merge transaction. Logically, a merge operation consists of a delete on one patient record, and an update of another patient record. Separate audit records shall be written for the delete operation and the update operation.

1235 **3.8.5.2.1 Patient Identity Source Actor audit message:**

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110110, DCM, "Patient Record")
	EventActionCode	M	"D" (delete) for the Delete operation "U" (update) for the Update operation
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-8", "IHE Transactions", "Patient Identity Feed")
Source (Patient Identity Source Actor) (1)			
Human Requestor (0..n)			
Destination (Patient Identifier Cross-reference Manager or Document Registry) (1)			
Audit Source (Patient Identity Source Actor) (1)			
Patient(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identity Source Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifier Cross-reference Manager or Document Registry facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (person)
	ParticipantObjectTypeCodeRole	M	"1" (patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

3.8.5.2.2 Patient Identifier Cross-reference Manager or Document Registry Actor audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110110, DCM, "Patient Record")
	EventActionCode	M	"D" (delete) for the Delete audit record "U" (update) for the Update audit record
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-8", "IHE Transactions", "Patient Identity Feed")
Source (Patient Identity Source Actor) (1)			
Destination (Patient Identifier Cross-reference Manager or Document Registry) (1)			
Audit Source (Patient Identifier Cross-reference Manager or Document Registry) (1)			
Patient(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identity Source Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*M*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

	UserID	M	The identity of the Patient Identifier Cross-reference Manager or Document Registry facility and receiving application from the HL7 message; concatenated together, separated by the | character.

	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

1245

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (person)
	ParticipantObjectTypeCodeRole	M	"1" (patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

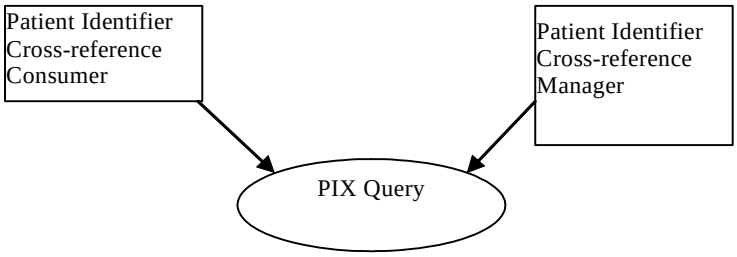
3.9 PIX Query

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

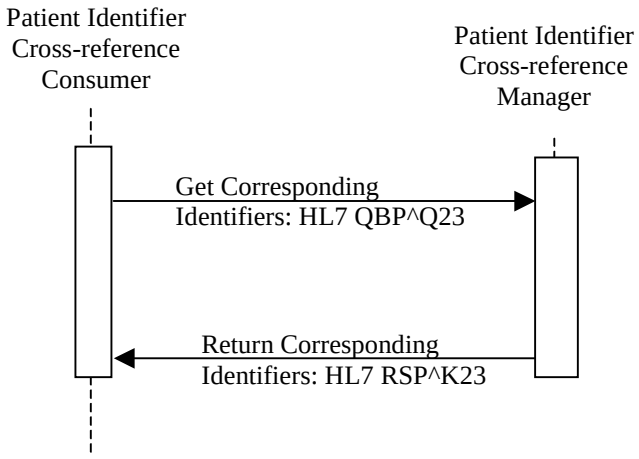


Figure 3.9-1 Get Corresponding Identifiers Sequence

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 ITI TF-2x: Appendix C and C.1.

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
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 ITI TF-2x: 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).

The Consumer shall specify “IHE PIX Query” for QPD-1 Message Query Name.

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 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 ITI TF-2x: Appendix C and C.1.

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

RSP	Segment Pattern Response	Chapter in HL7 2.5
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 ITI TF-2x: 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 ITI TF-2x: 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 ITI TF-2a: 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 ITI TF-2a: 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
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.

3.9.5 Security Considerations

3.9.5.1 Audit Record Considerations

The PIX Query Transaction is a Query Information event as defined in Table 3.20.6-1. The Actors involved in the transaction shall create audit data in conformance with DICOM (Supp 95) "Query", with the following exceptions:

3.9.5.1.1 Patient Identifier Cross-reference Consumer audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-9", "IHE Transactions", "PIX Query")
Source (Patient Identifier Cross-reference Consumer) (1)			
Human Requestor (0..n)			
Destination (Patient Identifier Cross-reference Manager) (1)			
Audit Source (Patient Identity Cross-reference Consumer) (1)			
Patient (0..n)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifer Cross-reference Consumer Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

	UserID	M	The identity of the Patient Identifier Cross-reference Manager facility and receiving application from the HL7 message; concatenated together, separated by the | character.

	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (system object)
	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-9", "IHE Transactions", "PIX Query")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	*U*	*not specialized*
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	M	the QPD segment of the query - Base64 encoded
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

3.9.5.1.2 Patient Identifier Cross-reference Manager audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-9", "IHE Transactions", "PIX Query")
Source (Patient Identifier Cross-reference Manager) (1)			
Destination (Patient Identifier Cross-reference Consumer) (1)			
Audit Source (Patient Identifier Cross-reference Manager) (1)			
Patient (0..n)			
Query Parameters(1)			

1485 Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifer Cross-reference Consumer Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifier Cross-reference Manager facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (system object)
	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*Not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-9", "IHE Transactions", "PIX Query")
	ParticipantObjectSensitivity	*U*	*Not specialized*
	ParticipantObjectID	*U*	*not specialized*
	ParticipantObjectName	*U*	*Not specialized*
	ParticipantObjectQuery	M	the QPD segment of the query - Base64 encoded
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

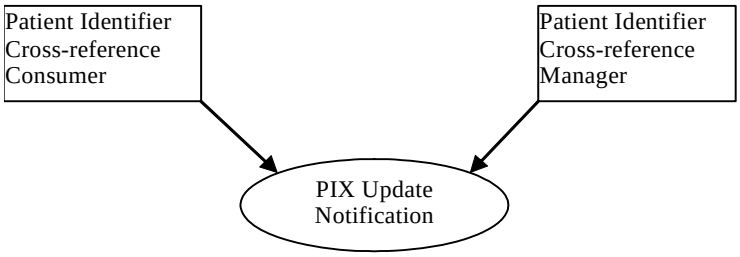
3.10 PIX Update Notification

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

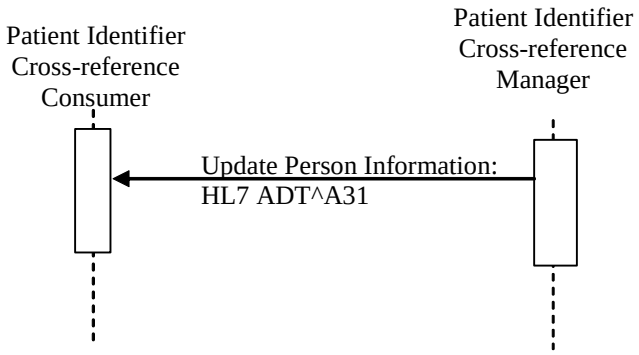


Figure 3.10-1 Update Person Information Sequence

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 shall only contain a list of cross-referenced identifiers for the domains of interest as configured with the Patient Identifier Cross-reference Manager actor in two or more of the domains managed by the cross-referencing Actor. Multiple notifications may need to be sent. For example:

Consumer CON_A is configured to receive update notifications for domains DOM_A and DOM_AD. Notifications are sent as follows:

- A PIX A01 feed is send for a patient for DOM_A. The update notification shall contain the patient identifier and assigning authority for DOM_A.
- A PIX A01 feed is processed for DOM_AD. The Patient Identifier Cross-reference Manager cross references this patient with DOM_A. The update notification shall contain the patient identifier and assigning authority for DOM_A and DOM_AD.
- A PIX A08 feed is processed for DOM_AD changing the patient address. The Patient Identifier Cross-reference Manager cross references determines this patient is no longer the same patient as DOM_A. Two update notifications shall be sent. One containing the patient identifier and assigning authority for DOM_A. The other one containing the patient identifier and assigning authority for DOM_AD.

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 Patient Identifier Cross-reference Manager Actor should account for consumers interested in all domains.

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 ITI TF-2x: 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 ITI TF-2x: 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_A05.

3.10.4.1.2.2 EVN Segment

See ITI TF-2x: 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.

3.10.5 Security Considerations

3.10.5.1 Audit Record Considerations

The PIX Update Notification Transaction is ”Patient Record” event, as defined in Table 3.20.6-1. The Actors involved in the transaction shall create audit data in conformance with DICOM (Supp 95) “Patient Record”, with the following exceptions:

3.10.5.1.1 Patient Identifier Cross-reference Manager audit message:

	Field Name	Opt	Value Constraints
Event AuditMessage/ EventIdentification	EventID	M	EV(110110, DCM, “Patient Record”)
	EventActionCode	M	“R” (Read)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV(“ITI-10”, “IHE Transactions”, “PIX Update Notification”)
Source (Patient Identifier Cross-reference Manager) (1)			
Human Requestor (0..n)			
Destination (Patient Identifier Cross-reference Consumer) (1)			
Audit Source (Patient Identifier Cross-reference Manager) (1)			
Patient IDs(1..n) (represents the components of PID-3)			

Where:

Source AuditMessage/ ActiveParticipant	UserID	M	The identity of the Patient Identifer Cross-reference Manager Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*Not specialized*
	UserIsRequestor	M	“true”
	RoleIDCode	M	EV(110153, DCM, “Source”)
	NetworkAccessPointTypeCode	M	“1” for machine (DNS) name, “2” for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) AuditMessage/ ActiveParticipant	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*Not specialized*
	UserName	*U*	*Not specialized*
	UserIsRequestor	M	“true”
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifier Cross-reference Consumer facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*Not specialized*
	UserName	*U*	*Not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

1620

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*Not specialized*
	AuditSourceTypeCode	*U*	*Not specialized*

Patient IDs **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*Not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

3.10.5.1.2 Patient Identifier Cross-reference Consumer audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110110, DCM, "Patient Record")
	EventActionCode	M	"U" (update)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-10", "IHE Transactions", "PIX Update Notification")
Source (Patient Identifier Cross-reference Manager) (1)			
Destination (Patient Identifier Cross-reference Consumer) (1)			
Audit Source (Patient Identifier Cross-reference Consumer) (1)			
Patient IDs(1..n) (represents the components of PID-3)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifer Cross-reference Manager Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Identifier Cross-reference Consumer facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

1625

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient IDs **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

3.11 Retrieve Specific Information for Display

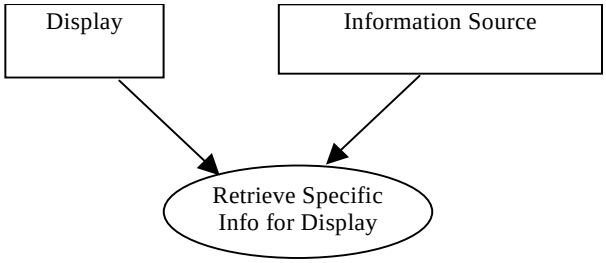
This section corresponds to Transaction ITI-11 of the IHE IT Infrastructure Technical Framework. Transaction ITI-11 is used by the Information Source and Display actors.

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.

http://www.w3.org/TR/xhtml-basicInteraction Diagram

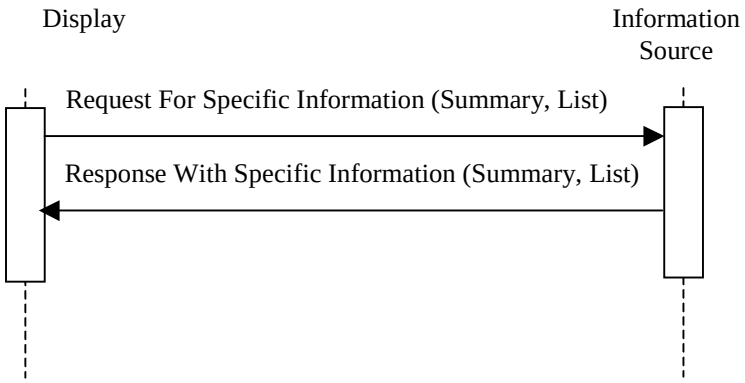


Figure 3.11-1 Request For Specific Information – Summary sequence

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 ITI TF-2a: 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.

1680

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 ITI TF-2x: Appendix A.

1685 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=99998410^^^%26www.mlhlife.com%26DNS &lowerDateTime=2003-01-01T00:00:00&upperDateTime=2003-01-01T23:59:59&mostRecentResults=1

1690 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.

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 ITI TF-2a: 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 ITI TF-2a: 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 ITI TF-2a: 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 ITI TF-2a: 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.
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 ITI TF-2a: 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 ITI TF-2x: 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 ITI TF-2a: 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 ITI TF-2a: 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.

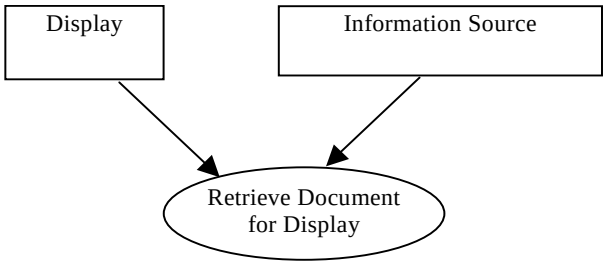
3.12 Retrieve Document for Display

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

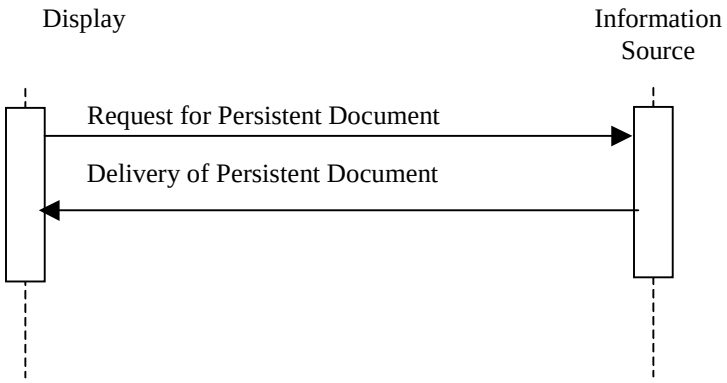


Figure 3.12-1 Request for Persistent Document Sequence

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 ITI TF-2x: 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 ITI TF-2x: 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 ITI TF-2a: 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 ITI TF-2a: 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 ITI TF-2a: 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

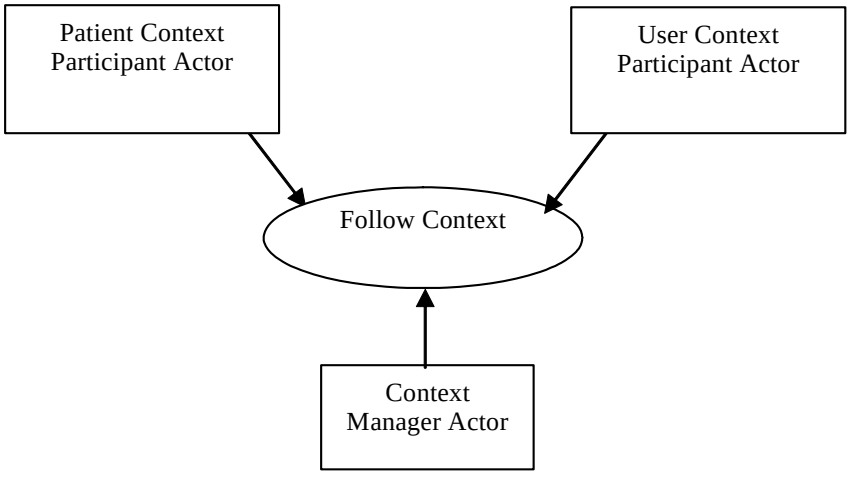
This transaction allows the Context Manager Actor to force other context participant actors to synchronize based on the new context values.

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

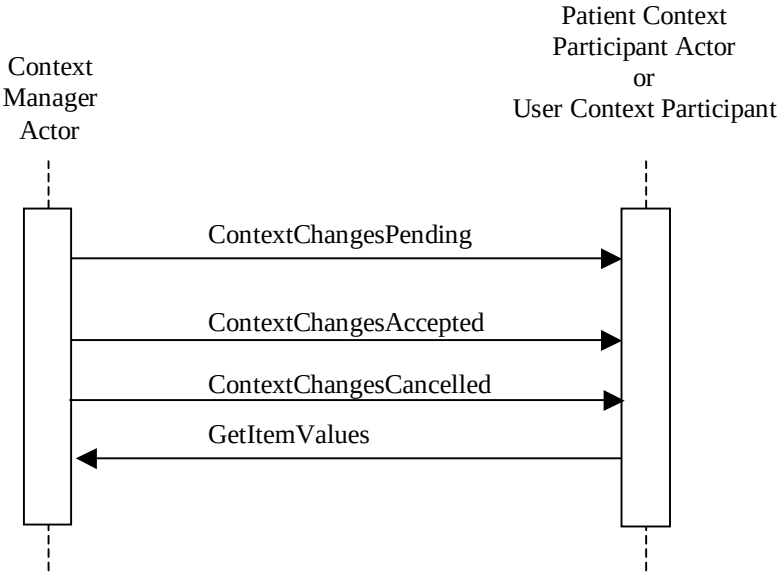


Figure 3.13-1 Follow Context – ContextChangesPending Method Sequence

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

NOTE: The XDS.a profile is deprecated starting in 2009. However, editorial changes to remove references to XDS.a, and associated changes to the relevant transactions, will not be complete until IT Infrastructure Technical Framework, Revision 7.0 is published in 2010.

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.

Note: This transaction is used by the XDS.a Integration Profile. For a discussion on XDS.a and XDS.b Integration Profiles see IHE ITI TF-1: 10 and IHE XDS.b Supplement (www.IHE.net/Technical _Frameworks).

3.14.1 Scope

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

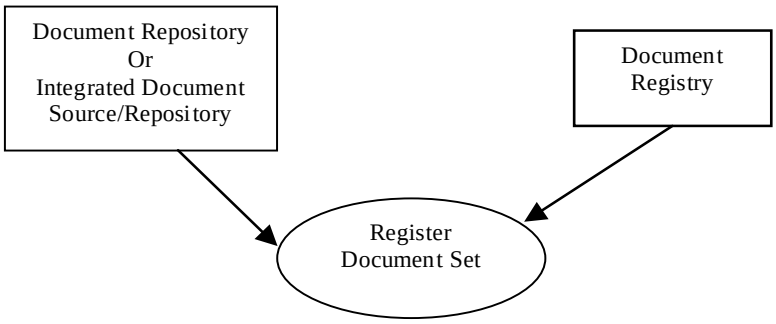
A Register Document Set transaction carries:

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 or Integrated Document Source/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.

Note: Within this transaction, the Document Repository and Integrated Document Source/Repository actors can be used interchangeably

3.14.3 Referenced Standards

ebRIM OASIS/ebXML Registry Information Model v2.1

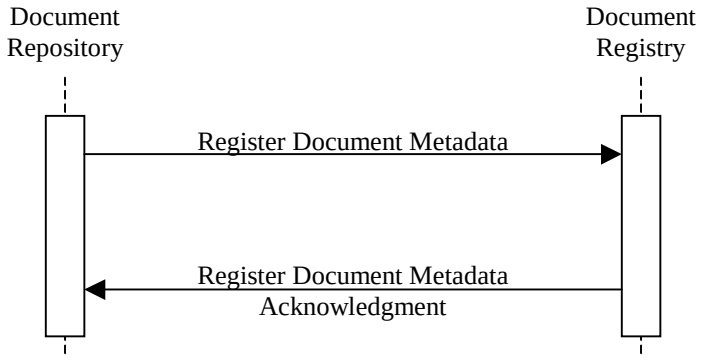
ebRS OASIS/ebXML Registry Services Specifications v2.1

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 sections in ITI TF-3: 4.1 specify the mapping of XDS concepts to ebRS and ebRIM semantics and document metadata.

The Registry actor shall store and later include in metadata returned in a query response the XDSDocumentEntry.URI attribute along with the other metadata attributes received in the Register Document Set [ITI-14] transaction as determined by profile and transaction requirements. The Registry actor may store and later include in metadata returned in a query response the XDSDocumentEntry.repositoryUniqueId attribute if it is present in the Register Document Set [ITI-14] transaction.

3.14.4.1.2.1 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-2a: 3.15.4.1.2.3.1 (On-line protocol binding).

3.14.4.1.2.2 Sequencing Requirements

The Repository actor shall:

1. 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:

2. The Document Registry actor may choose to validate URIs contained in metadata before acknowledging the Register Document Metadata transaction.
3. The Document Consumer actor may retrieve the document before the Register Document Metadata Acknowledgement is received by the Repository actor.

3.14.4.1.2.3 Intentionally Left Blank

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

3.14.4.1.3.1 Basic Patient Privacy Enforcement Option

If the Basic Patient Privacy Enforcement Option is implemented:

1. The Integrated Document Source / Repository actor shall populate the confidentialityCode in the document metadata with the list of OID values that identify the Patient Privacy Consent Policies that apply to the associated document. The confidentiality codes for different documents in the same submission may be different.

2. The Integrated Document Source / Repository actor shall be able to be configured with the Patient Privacy Consent Policies, Patient Privacy Consent Policy Identifiers (OIDs) and associated information necessary to understand and enforce the XDS Affinity Domain Policy. The details of this are product specific and not specified by IHE.
3. The Integrated Document Source / Repository actor may have a user interface or business rule capabilities to determine the appropriate confidentiality codes for each document. The details of this are product specific and not specified by IHE. However, the information about how confidentiality codes are assigned must be part of the published policy for the XDS Affinity Domain. For example, when publishing a document, the Integrated Document Source / Repository might show a list of checkboxes where a user can select which of the available consents a document is to be published.

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.

3.14.5 Security Considerations

3.14.5.1 XDS Affinity Domain Security Considerations

The XDS 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 XDS Affinity Domain Audit Repository Actor.

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

The individual actors involved are often members of different secure domains, as illustrated in Figure 3.14.5.1-2. The data transfers between different secure domains need different protection than transfers

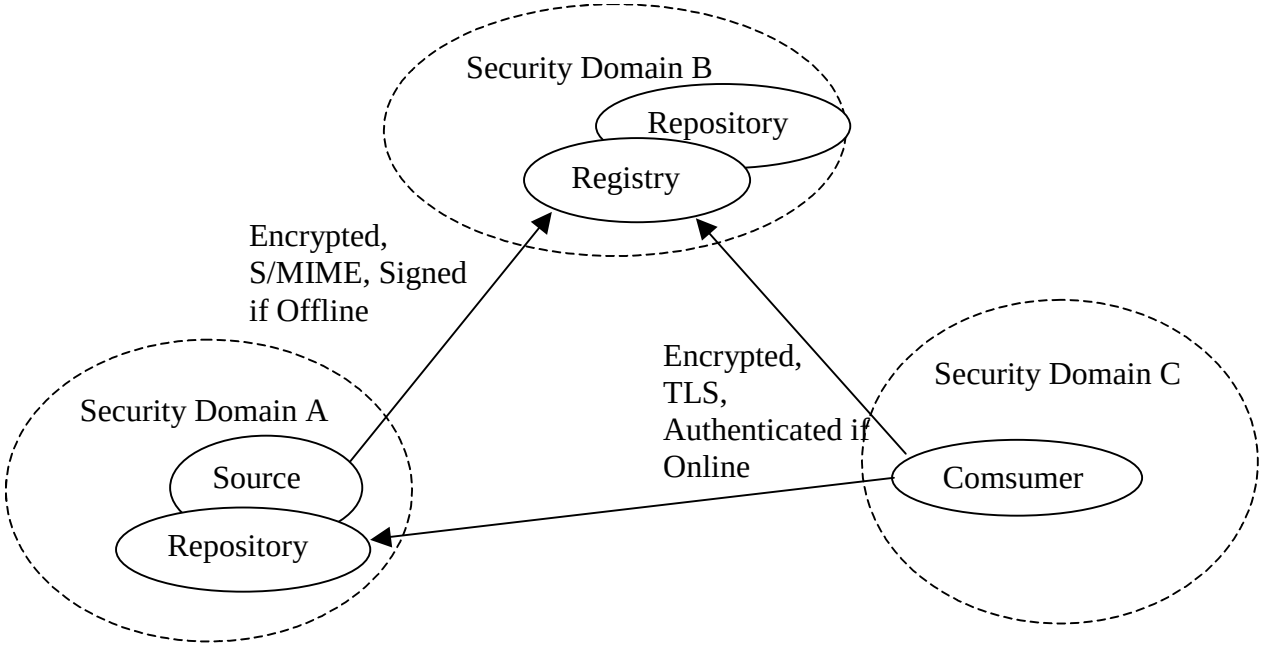
within a secure domain. The transactions used between different secure domains shall use the ATNA Encryption Option.

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 XDS 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 XDS Affinity Domain is managed and controlled by the business relationship terms of the XDS Affinity Domain. There is no automatic IHE transaction for such access.

The audit records that shall be generated (references IHE ATNA Integration Profile) by normal XDS activities are defined in the appropriate Security Considerations section of each transaction:

Figure 3.14.5-2 - Example Security Domain Relationships



All Actors are part of the same Clinical Affinity Domain

3.14.5.2 Audit Record Considerations

The Register Document Set Transaction is PHI-Export event, as defined in table 3.20.6-1. The Actors involved in the transaction shall create audit data in conformance with DICOM (Supp 95) “Data Export”, with the following exceptions.

3.14.5.2.1 Document Repository or Integrated Document Source/Repository audit message:

	Field Name	Opt	Value Constraints
Event AuditMessage/ EventIdentification	EventID	M	EV(110106, DCM, "Export")
	EventActionCode	M	"R" (Read)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-14", "IHE Transactions", "Register Document Set")
Source (Document Repository or Integrated Document Source/Repository) (1)			
Human Requestor (0..n)			
Destination (Document Registry) (1)			
Audit Source (Document Repository or Integrated Document Source/Repository) (1)			
Patient (1)			
SubmissionSet (1)			

Where:

Source AuditMessage/ ActiveParticipant	UserID	U	not specialized
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) AuditMessage/ ActiveParticipant	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

2250

Destination AuditMessage/ ActiveParticipant	UserID	M	SOAP endpoint URI.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source AuditMessage/ AuditSourceIdentification	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (person)
	ParticipantObjectTypeCodeRole	M	"1" (patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format..
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Submission Set **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (System)
	ParticipantObjectTypeCodeRole	M	"20" (job)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd", "IHE XDS Metadata", "submission set classificationNode")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The submissionSet unique ID
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*

3.14.5.2.2 Document Registry audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110107, DCM, "Import")
	EventActionCode	M	"C" (Create)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-14", "IHE Transactions", "Register Document Set")
Source (Document Repository or Integrated Document Source/Repository) (1)			
Destination (Document Registry) (1)			
Audit Source (Document Registry) (1)			
Patient (1)			
SubmissionSet (1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	*U*	*not specialized*
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

2255

Destination **AuditMessage/ ActiveParticipant**	UserID	M	SOAP endpoint URI
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

2260

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (person)
	ParticipantObjectTypeCodeRole	M	"1" (patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	the patient ID in HL7 CX format..
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Submission Set **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (System)
	ParticipantObjectTypeCodeRole	M	"20" (job)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd", "IHE XDS Metadata", "submission set classificationNode")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The submissionSet unique ID
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*

3.15 Provide and Register Document Set

NOTE: The XDS.a profile is deprecated starting in 2009. However, editorial changes to remove references to XDS.a, and associated changes to the relevant transactions, will not be complete until IT Infrastructure Technical Framework Volume 2, Revision 7.0 is published in 2010.

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.

Note: This transaction is used by the XDS.a Integration Profile. For a discussion on XDS.a and XDS.b Integration Profiles see IHE ITI TF-1: 10 and IHE XDS.b Supplement (www.IHE.net/Technical _Frameworks).

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-3: 4.1.3.2) from a Document Source to a Document Registry.

A Provider and Register Document Set transaction carries:

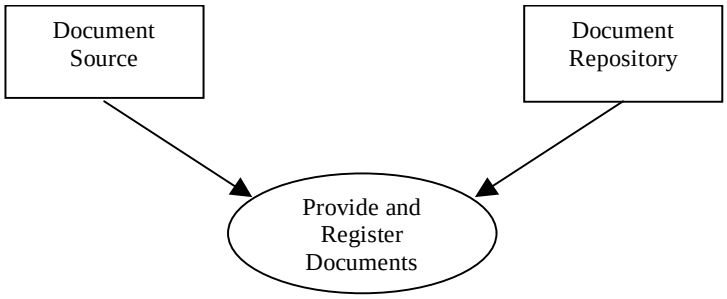
Metadata describing zero or more new documents

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 ITI-TF-2a: 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 v2.1

ebRIM OASIS/ebXML Registry Information Model v2.1

ebRS OASIS/ebXML Registry Services Specifications v2.1

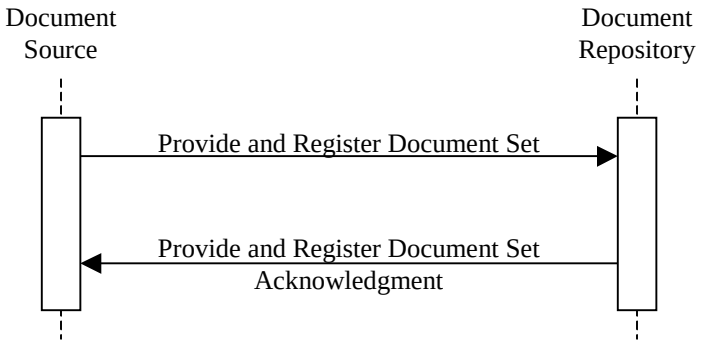
HTTP HyperText Transfer Protocol HTTP/1.1 (IETF RFC2616)

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.

The Document Repository shall, upon receipt of a Provide and Register Document Set [ITI-15] transaction send a corresponding Register Document Set [ITI-14] transaction to the Document Registry actor.

- The Document Repository actor shall create and insert the XDSDocumentEntry.URI, XDSDocumentEntry.size, and XDSDocumentEntry.hash attributes for each document received from the Provide and Register Document Set [ITI-15] transaction into the Register Document Set [ITI-14] transaction metadata. If any of these attributes are present in the Provide and Register Document Set [ITI-15] transaction they shall be replaced. The XDSDocumentEntry.URI attribute value shall later be accepted in a Retrieve Document transaction [ITI-17] for that document and the document shall be returned.
- The Document Repository actor shall also create and insert the XDSDocumentEntry.repositoryUniqueId attribute if it will support retrieval of that document via the Retrieve Document Set transaction [ITI-43]. If ITI-43 is not supported then this attribute shall not be present in ITI-14 metadata (removed by the Document Repository actor if necessary).

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 ITI TF-3: 4.1.7 through 4.1.9) 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 Intentionally Left Blank

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.

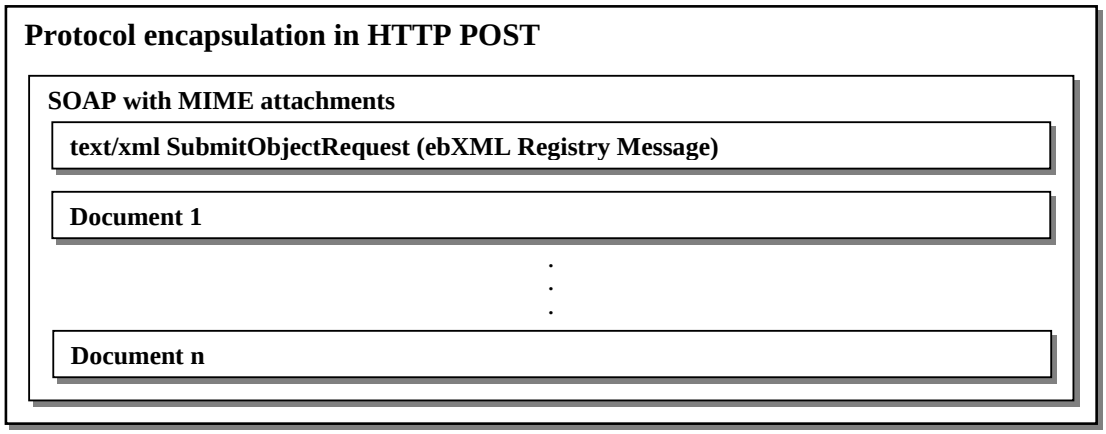


Figure 3.15.4.1-1 General Diagram of the Main message composing the On-Line Provide and Register Document Set Transaction

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 documents1.

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. 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

1 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.

Document Repository e-mail address to be able to provide and register a document set.

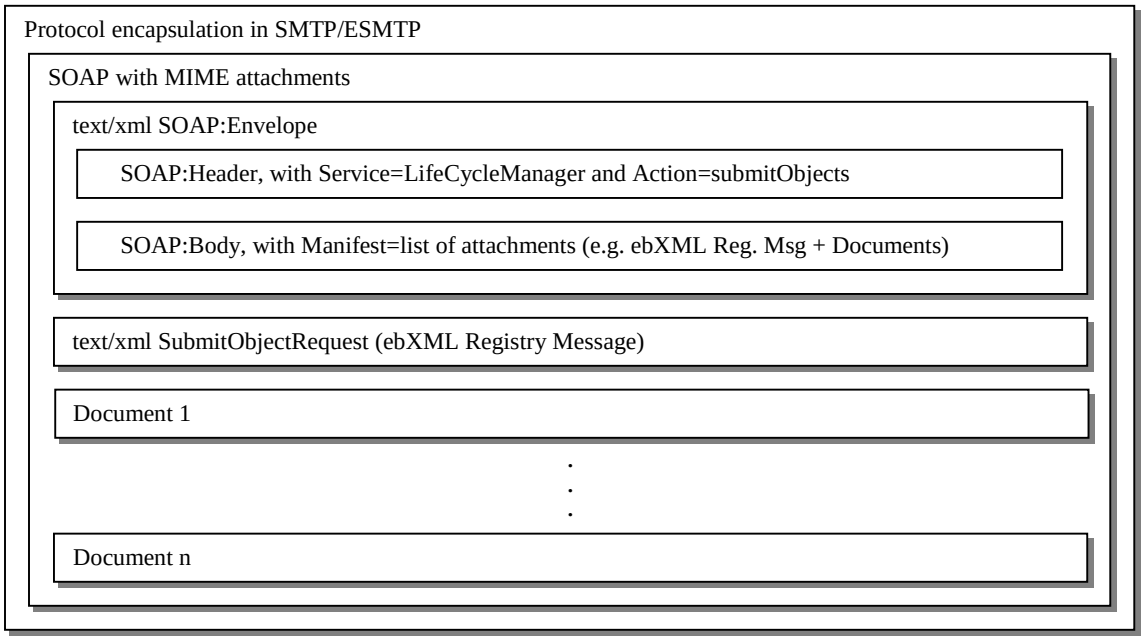


Figure 3.15.4.1-2 General Diagram of the Main message composing the Off-Line Provide and Register Document Set Transaction

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.

2430 **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="2.1" eb:signed="false"

- 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").

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-2a: 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).

If hash and/or size slots are present in the submission, the repository shall verify the value(s) with the actual value(s) of the submitted document and return an error on mismatch.
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).

3.15.4.1.3.1 Basic Patient Privacy Enforcement Option

If the Basic Patient Privacy Enforcement Option is implemented:

1. The Document Source actor shall populate the confidentialityCode in the document metadata with the list of OID values that identify the Patient Privacy Consent Policies that apply to the associated document. The confidentiality codes for different documents in the same submission may be different.
2. The Document Source actor shall be able to be configured with the Patient Privacy Consent Policies, Patient Privacy Consent Policy Identifiers (OIDs) and associated information necessary to understand and enforce the XDS Affinity Domain Policy. The details of this are product specific and not specified by IHE.
3. The Document Source actor may have user interface or business rule capabilities to determine the appropriate confidentiality codes for each document. The details of this are product specific and not specified by IHE. However, the information about how confidentiality codes are assigned must be part of the published policy for the XDS Affinity Domain. For example, when publishing a document, the Document Source, might show a list of checkboxes where a user can select which of the available consents a document is to be published.
4. The Document Recipient actor shall be able to be configured with the Patient Privacy Consent Policies, Patient Privacy Consent Policy Identifiers (OIDs) and associated information necessary to understand and enforce the policies. The meanings of the codes on the media must be provided out of band, e.g., by telephone, fax, or email. The detail of how this is done is product specific and not specified by IHE. If the documents are transferred internally within the organization or to other members of the recipient's affinity domain, appropriate internal confidentiality codes shall be applied.
5. The Document Recipient actor shall have the ability to coerce the confidentiality code in the metadata associated with the document from the codes used by the Document Source to the codes used by the Document Recipient.
6. The Document Recipient actor shall abide by the XDS Affinity Domain Policies represented by the confidentialityCode in the metadata associated with the document. The Document Recipient actor likely will have user access controls or business rule capabilities to determine the details of how confidentiality codes apply to query results. The details of this are product specific and not specified by IHE. These rules shall reduce the query results to only those that are appropriate to the current situation for that actor and user.

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 one or more documents. Whether a submission contains a single or multiple documents depends on workflows, policies, and other external factors which are outside of the scope of this profile.

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

1. **Document Replacement Option:** In this option the Document Source offers the ability to submit a document as a replacement for another document already in the registry/repository.
2. **Document Addendum Option:** In this option the Document Source shall offer the ability to submit a document as an addendum to another document already in the registry/repository.
3. **Document Transformation Option:** In this option the Document Source shall offer the ability to 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)

4. **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 ITI TF-3: 4.1.3.4 Other Properties of Submission Requests.

3.15.5.2 Document Repository

A Document Repository shall be capable of accepting submissions containing multiple documents.

Note: The Document Source may submit single documents or multiple documents depending on its needs.

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.

3.15.6 Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the Register Document transaction (see ITI TF-2a: 3.14.5.1).

3.15.6.1 Audit Record Considerations

The Provide and Register Document Set Transaction is PHI-Export event, as defined in Table 3.20.6-1. The Actors involved in the transaction shall create audit data in conformance with DICOM (Supp 95) "Data Export"/"Data Import", with the following exceptions.

3.15.6.1.1 Document Source audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110106, DCM, "Export")
	EventActionCode	M	"R" (Read)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-15", "IHE Transactions", "Provide and Register Document Set")
Source (Document Source) (1)			
Human Requestor (0..n)			
Destination (Document Repository) (1)			
Audit Source (Document Source) (1)			
Patient (1)			
SubmissionSet (1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	U	not specialized
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination **AuditMessage/ ActiveParticipant**	UserID	M	SOAP endpoint URI.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

2555

Patient **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
	ParticipantObjectTypeCode	M	"2" (System)
	ParticipantObjectTypeCodeRole	M	"20" (job)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd", "IHE XDS Metadata", "submission set classificationNode")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The submissionSet unique ID

	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*

3.15.6.1.2 Document Repository audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110107, DCM, "Import")
	EventActionCode	M	"C" (Create)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-15", "IHE Transactions", "Provide & Register Document Set")
Source (Document Source) (1)			
Destination (Document Repository) (1)			
Audit Source (Document Repository) (1)			
Patient (1)			
SubmissionSet (1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	*U*	*not specialized*
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

2560

Destination **AuditMessage/ ActiveParticipant**	UserID	M	SOAP endpoint URI
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*

	ParticipantObjectDetail	*U*	*not specialized*
Submission Set **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (System)
	ParticipantObjectTypeCodeRole	M	"20" (job)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd", "IHE XDS Metadata", "submission set classificationNode")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The submissionSet unique ID
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*

3.16 Query Registry

NOTE: The XDS.a profile is deprecated starting in 2009. However, editorial changes to remove references to XDS.a, and associated changes to the relevant transactions, will not be complete until IT Infrastructure Technical Framework Volume 2, Revision 7.0 is published in 2010.

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

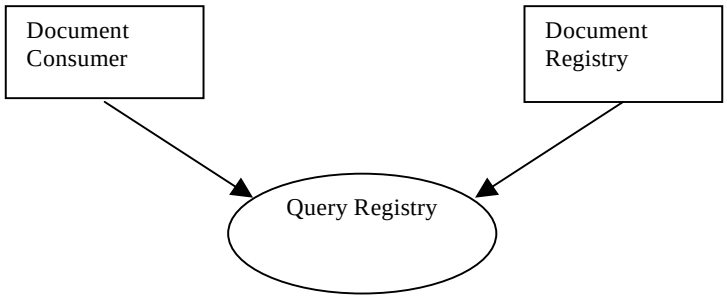
Query by time of submission

The list of XDS registry entries attributes that can be the target of a query are defined in ITI TF-3: 4.1.7 through 4.1.9. 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.

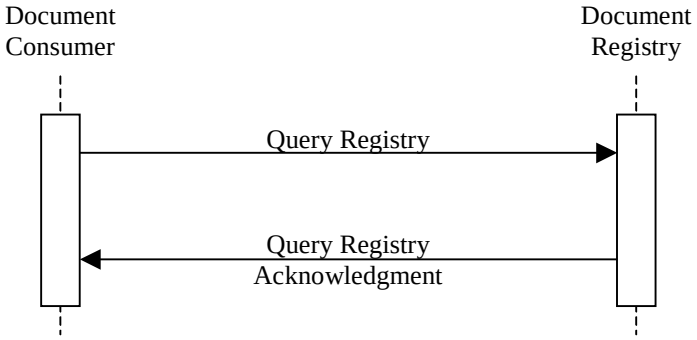
Actor: Document Registry

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 v2.1
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

The 'LeafClass' returnType is meant for returning a small amount of fully specified ebXML objects (such as a list of ExtrinsicObject (XDSDocumentEntry) elements with full contents: slots, external

identifiers, classifications etc.). This type of query result is self-contained, everything known about the object(s) is returned. The specific query documented in this section describes which object types will be included. ObjectRef elements are also returned. These represent objects not included in the returned object list that are referenced by objects in the returned object list. These ObjectRefs are required by the registry standard.

The 'ObjectRef' returnType returns references to the registry objects that match the query. This type query is recommended when the returned object list could be large. An initial query returning ObjectRefs for all objects of interest followed by secondary queries requesting full metadata (query type LeafClass) is an efficient way to query for large bodies of metadata. This strategy is particularly easy to use when querying for a single object type (XDSDocumentEntry or XDSSubmissionSet are examples) since only a single object type is involved.

An ObjectRef looks like:

```
<ObjectRef id="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"/>
```

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.

3.16.4.1.2.2.1 SQL Select Statement

All SELECT statements shall explicitly code the SELECT variable. So,

SELECT eo.id FROM ExtrinsicObject eo …

Is valid, and

SELECT * FROM ExtrinsicObject eo …

is not.

Note: This requirement is stated in ebRS verson 2.1 in section 8.3.2 Semantic Constraints on Query Syntax.

3.16.4.1.2.3 Security Requirements

Relevant security requirements are discussed in the Register Document transaction (see ITI TF-2a: 3.14.4.1.2.3) and in Security Considerations Sections ITI TF-2a: 3.16.4.1.5 and 3.16.4.2.4.

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.

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 YYYYMMDDHHMMSS (each degree of accuracy is optional, see definition of the DTM XDS Data Type in Table 3.14.2.1-3). 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 '%'.

Single values are coded as

123 - without quotes for numbers

'Approved' - in single quotes for strings.

'Children''s Hospital' – a single quote is inserted in a string by specifying two single quotes

Within the LIKE predicate

Underscore ('_') matches an arbitrary character

Percent ('%') matches an arbitrary string

Format for multiple values is

(value, value, value, …) OR

(value) if only one value is to be sepecified.

where each value is coded as described above for single values. Parameters labeled as accepting multiple values shall be coded using this format.

In the query parameter tables below, each row represents a query parameter. Optional parameters which are not included in the query invocation have no affect on the query. Queries return registry objects that match all the supplied parameters. When multiple values are included for a parameter, objects are returned that match any included value (within the context of the larger query).

In the following tables, coding schemes are represented by a pair of parameters, one representing the code value and the second representing the coding scheme name from which the code value is taken. For example, in the FindDocuments query, are found parameters

$XDSDocumentEntryClassCode – classCode values of interest

$XDSDocumentEntryClassCodeScheme – coding scheme for each class code value

While the 'codes' can be specified without the 'code schemes', if any code schemes are specified (if the code schemes parameter is specified) then the code schemes for all codes listed shall be present and in the same order as the codes.

The 'Opt' column of each query parameter table below specifies whether the parameter is required ('R') or optional ('O'). Document Registry actor implementations shall reject queries missing required parameters and shall accept but not require all optional parameters.. The 'Mult' column labels each query parameter as accepting multiple values ('M') or not accepting multiple values ('—').

The 'status' attribute (XDSDocumentEntryStatus, XDSSubmissionSetStatus, XDSFolderStatus) shall take on values in the set ('Approved', 'Deprecated'). Formatting of the Patient ID field requires special attention. The Registry actor shall perform string matching on the Patient ID field per SQL specifications and will provide no pre-processing of Patient ID values. That is, Document Registry actors do not process the Patient ID, remove unwanted components, and then process the query. Document Consumer actors are required to specify the proper format for Patient ID values (IDNumber^^^&OIDofAA&ISO) and shall include no further values in the Patient ID. Should the Document Consumer add extra values (e.g., a value in component 5 of the HL7 view of the data), the response of the Registry actor is undefined.

3.16.4.1.4.1 FindDocuments

Find documents (XDSDocumentEntry objects) in the registry for a given patientID with a matching 'status' attribute. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects matching the query parameters

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	R	--

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryClassCode	XDSDocumentEntry. classCode	O	M
$XDSDocumentEntryClassCodeScheme	XDSDocumentEntry. classCode[1]	O[2]	M[2]
$XDSDocumentEntryPracticeSettingCode	XDSDocumentEntry. practiceSettingCode	O	M
$XDSDocumentEntryPracticeSettingCodeScheme	XDSDocumentEntry. practiceSettingCode[1]	O[2]	M[2]
$XDSDocumentEntryCreationTimeFrom	Lower value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryCreationTimeTo	Upper value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryServiceStartTimeFrom	Lower value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStartTimeTo	Upper value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStopTimeFrom	Lower value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryServiceStopTimeTo	Upper value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryHealthcareFacilityTypeCode	XDSDocumentEntry. healthcareFacilityTypeCode	O	M
$XDSDocumentEntryHealthcareFacilityTypeCodeScheme	XDSDocumentEntry. healthcareFacilityTypeCode[1]	O[2]	M[2]
$XDSDocumentEntryEventCodeList	XDSDocumentEntry. eventCodeList	O	M
$XDSDocumentEntryEventCodeListScheme	XDSDocumentEntry. eventCodeList[1]	O[2]	M[2]
$XDSDocumentEntryConfidentialityCode	XDSDocumentEntry. confidentialityCode	O	M
$XDSDocumentEntryFormatCode	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryStatus	XDSDocumentEntry. status	R	M

[1]This attribute is not listed by name in ITI TF-3: Table 4.1-5 Document Metadata Attribute Definition. It is included in the documentation of the attribute that bears the same name without the 'Scheme' suffix. As an example, XDSDocumentEntry.ClassCodeScheme is not an attribute listed in attribute table for XDSDocumenEntry. It is documented as part of XDSDocumentEntry.classCode, specifically as the 'codingScheme' Slot.

[2]This parameter is optional but if included shall 1) have the same number of values as the corresponding 'code' attribute, 2) the i^{th} value of the 'code' attribute (e.g. XDSDocumentEntryClassCode) shall correspond to the i^{th} value of the 'codeScheme' attribute (e.g. XDSDocumentEntryClassCodeScheme).

Example SQL

```
SELECT doc.id
FROM ExtrinsicObject doc, ExternalIdentifier patId
, Classification clCode     # $XDSDocumentEntryClassCode
, Classification psc        # $XDSDocumentEntryPracticeSettingCode
, Classification hftc       # $XDSDocumentEntryhealthcareFacilityTypeCode
, Classification ecl        # $XDSDocumentEntryEventCodeList
, Slot clCodeScheme         # $XDSDocumentEntryClassCodeScheme
, Slot psCodeScheme         # $XDSDocumentEntryPracticeSettingCodeScheme
, Slot crTimef              # $XDSDocumentEntryCreationTimeFrom
, Slot crTimet              # $XDSDocumentEntryCreationTimeTo
, Slot serStartTimef        # $XDSDocumentEntryServiceStartTimeFrom
, Slot serStartTimet        # $XDSDocumentEntryServiceStartTimeTo
, Slot serStopTimef         # $XDSDocumentEntryServiceStopTimeFrom
, Slot serStopTimet         # $XDSDocumentEntryServiceStopTimeTo
```



```
, Slot hftcScheme           # $XDSDocumentEntryHealthcareFacilityTypeCodeScheme
, Slot eclScheme            # $XDSDocumentEntryEventCodeListScheme
, Classification conf       # $XDSDocumentEntryConfidentialityCode
, Classfication fmtCode     # $XDSDocumentEntryFormatCode
 WHERE
  doc.objectType = 'urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1'
# patientID
  AND (doc.id = patId.registryobject AND
       patId.identificationScheme='urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427' AND
       patId.value = $XDSDocumentEntryPatientId )
# classCode
  AND (clCode.classifiedobject = doc.id AND                                          # $XDSDocumentEntryClassCode
       clCode.classificationScheme = 'urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a' AND # $XDSDocumentEntryClassCode
       clCode.nodeRepresentation IN $XDSDocumentEntryClassCode )                     # $XDSDocumentEntryClassCode
# classCode Scheme
#   This coding depends on the above clause being included.
  AND (clCodeScheme.parent = clCode.id AND                                           # $XDSDocumentEntryClassCodeScheme
       clCodeScheme.name = 'codingScheme' AND                                        # $XDSDocumentEntryClassCodeScheme
       clCodeScheme.value IN $XDSDocumentEntryClassCodeScheme)                       # $XDSDocumentEntryClassCodeScheme
# practice setting code
  AND (psc.classifiedObject = doc.id AND                                             # $XDSDocumentEntryPracticeSettingCode
       psc.classificationScheme='urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead' AND  # $XDSDocumentEntryPracticeSettingCode
       psc.nodeRepresentation IN $XDSDocumentEntryPracticeSettingCode )              # $XDSDocumentEntryPracticeSettingCode
# practiceSetting Scheme
#   This coding depends on the above clause being included.
  AND (psCodeScheme.parent = psc.id AND                                              # $XDSDocumentEntryPracticeSettingCodeScheme
       psCodeScheme.name = 'codingScheme' AND                                        # $XDSDocumentEntryPracticeSettingCodeScheme
       psCodeScheme.value IN $XDSDocumentEntryPracticeSettingCodeScheme)             # $XDSDocumentEntryPracticeSettingCodeScheme
# creationTime from
  AND (crTimef.parent = doc.id AND                                                   # $XDSDocumentEntryCreationTimeFrom
       crTimef.name = 'creationTime' AND                                             # $XDSDocumentEntryCreationTimeFrom
       $XDSDocumentEntryCreationTimeFrom &lt;= crTimef.value )                       # $XDSDocumentEntryCreationTimeFrom
  # creationTime to
  AND (crTimet.parent = doc.id AND                                                   # $XDSDocumentEntryCreationTimeTo
       crTimet.name = 'creationTime' AND                                             # $XDSDocumentEntryCreationTimeTo
       crTimet.value &lt; $XDSDocumentEntryCreationTimeTo)                           # $XDSDocumentEntryCreationTimeTo
# serviceStartTime from
  AND (serStartTimef.parent = doc.id AND                                             # $XDSDocumentEntryServiceStartTimeFrom
       serStartTimef.name = 'serviceStartTime' AND                                   # $XDSDocumentEntryServiceStartTimeFrom
       $XDSDocumentEntryServiceStartTimeFrom &lt;= serStartTimef.value )             # $XDSDocumentEntryServiceStartTimeFrom
# serviceStartTime to
  AND (serStartTimet.parent = doc.id AND                                             # $XDSDocumentEntryServiceStartTimeTo
       serStartTimet.name = 'serviceStartTime' AND                                   # $XDSDocumentEntryServiceStartTimeTo
       serStartTimet.value &lt; $XDSDocumentEntryServiceStartTimeTo)                 # $XDSDocumentEntryServiceStartTimeTo
# serviceStopTime from
  AND (serStopTimef.parent = doc.id AND                                              # $XDSDocumentEntryServiceStopTimeFrom
       serStopTimef.name = 'serviceStopTime' AND                                     # $XDSDocumentEntryServiceStopTimeFrom
       $XDSDocumentEntryServiceStopTimeFrom &lt;= serStopTimef.value )               # $XDSDocumentEntryServiceStopTimeFrom
# serviceStopTime to
  AND (serStopTimet.parent = doc.id AND                                              # $XDSDocumentEntryServiceStopTimeTo
       serStopTimet.name = 'serviceStopTime' AND                                     # $XDSDocumentEntryServiceStopTimeTo
       serStopTimet.value &lt; $XDSDocumentEntryServiceStopTimeTo)                   # $XDSDocumentEntryServiceStopTimeTo
# healthcare facility type code
  AND (hftc.classifiedObject = doc.id AND                                            # $XDSDocumentEntryhealthcareFacilityTypeCode
       hftc.classificationScheme = 'urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1' AND # $XDSDocumentEntryhealthcareFacilityTypeCode
       hftc.nodeRepresentation IN $XDSDocumentEntryhealthcareFacilityTypeCode )      # $XDSDocumentEntryhealthcareFacilityTypeCode
# healthcareFacilityTypeCode Scheme
#   This coding depends on the above clause being included.
  AND (hftcScheme.parent = hftc.id AND                                  # $XDSDocumentEntryHealthcareFacilityTypeCodeScheme
       hftcScheme.name = 'codingScheme' AND                             # $XDSDocumentEntryHealthcareFacilityTypeCodeScheme
       hftcScheme.value IN $XDSDocumentEntryHealthcareFacilityTypeCodeScheme)  # $XDSDocumentEntryHealthcareFacilityTypeCodeScheme
# event code list
  AND (ecl.classifiedObject = doc.id AND                                             # $XDSDocumentEntryEventCodeList
       ecl.classificationScheme = 'urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4' AND # $XDSDocumentEntryEventCodeList
       ecl.nodeRepresentation IN $XDSDocumentEntryEventCodeList )                    # $XDSDocumentEntryEventCodeList
# eventCodeList Scheme
#   This coding depends on the above clause being included.
  AND (eclScheme.parent = ecl.id AND                                                 # $XDSDocumentEntryEventCodeListScheme
       eclScheme.name = 'codingScheme' AND                                           # $XDSDocumentEntryEventCodeListScheme
       eclScheme.value IN $XDSDocumentEntryEventCodeListScheme)                      # $XDSDocumentEntryEventCodeListScheme
# confidentialityCode
  AND (conf.classifiedObject = doc.id AND                                            # $XDSDocumentEntryConfidentialityCode
       conf.classificationScheme = 'urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f' AND # $XDSDocumentEntryConfidentialityCode
       conf.nodeRepresentation IN $XDSDocumentEntryConfidentialityCode )             # $XDSDocumentEntryConfidentialityCode
# format code
  AND (fmtCode.classifiedObject = doc.id AND                                         # $XDSDocumentEntryFormatCode
       fmtCode.classificationScheme = 'urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d' AND # $XDSDocumentEntryFormatCode
       fmtCode.nodeRepresentation IN $XDSDocumentEntryFormatCode)                     # $XDSDocumentEntryFormatCode
 # status
  AND doc.status IN $XDSDocumentEntryStatus
```


3.16.4.1.4.2 FindSubmissionSets

Find submission sets (XDSSubmissionSet objects) in the registry for a given patientID with matching 'status' attribute. The other parameters can be used to restrict the collection of XDSSubmissionSet objects returned.

Returns: XDSSubmissionSet objects matching the query parameters

Parameter Name	Attribute	Opt	Mult
$XDSSubmissionSetPatientId	XDSSubmissionSet. patientId	R	--
$XDSSubmissionSetSourceId	XDSSubmissionSet. sourceId	O	M
$XDSSubmissionSetSubmissionTimeFrom	XDSSubmissionSet. submissionTime Lower value	O	--
$XDSSubmissionSetSubmissionTimeTo	XDSSubmissionSet. submissionTime Upper value	O	--
$XDSSubmissionSetAuthorPerson[1]	XDSSubmissionSet. authorPerson	O	--
$XDSSubmissionSetContentType	XDSSubmissionSet. contentTypeCode	O	M
$XDSSubmissionSetStatus	XDSSubmissionSet. status	R	M

[1]The value for this parameter is a pattern compatible with the SQL keyword LIKE.

Example SQL

```
SELECT ss.id
FROM RegistryPackage ss, ExternalIdentifier patId
 , Slot subTimeFrom       # $XDSSubmissionSetSubmissionTimeFrom
 , Slot subTimeTo         # $XDSSubmissionSetSubmissionTimeTo
 , Slot ap                # $XDSSubmissionSetAuthorPerson
 , ExternalIdentifier sid # $XDSSubmissionSetSourceId
 , Classification ctc     # $XDSSubmissionSetContentType
WHERE
# patient ID
      ( ss.id = patId.registryobject AND
      patId.identificationScheme= 'urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446' AND
      patId.value = $XDSSubmissionSetPatientId )
# source ID
  AND ( sid.registryobject = ss.id AND                                                  # $XDSSubmissionSetSourceId
      sid.identificationScheme = 'urn:uuid:554ac39e-e3fe-47fe-b233-965d2a147832' AND    # $XDSSubmissionSetSourceId
      sid.value IN $XDSSubmissionSetSourceId )                                          # $XDSSubmissionSetSourceId
# submission set time from
  AND ( subTimeFrom.parent = ss.id AND                                                  # $XDSSubmissionSetSubmissionTimeFrom
      subTimeFrom.name = 'submissionTime' AND                                           # $XDSSubmissionSetSubmissionTimeFrom
      subTimeFrom.value >= $XDSSubmissionSetSubmissionTimeFrom )                        # $XDSSubmissionSetSubmissionTimeFrom
# submission set time to
  AND ( subTimeTo.parent = ss.id AND                                                    # $XDSSubmissionSetSubmissionTimeTo
      subTimeTo.name = 'submissionTime' AND                                             # $XDSSubmissionSetSubmissionTimeTo
      subTimeTo.value &lt; $XDSSubmissionSetSubmissionTimeTo )                          # $XDSSubmissionSetSubmissionTimeTo
# author person
  AND ( ap.parent = ss.id AND                                                           # $XDSSubmissionSetAuthorPerson
      ap.name = 'authorPerson' AND                                                      # $XDSSubmissionSetAuthorPerson
      ap.value LIKE $XDSSubmissionSetAuthorPerson )                                     # $XDSSubmissionSetAuthorPerson
# content type codes
  AND ( ctc.classifiedObject = ss.id AND                                                # $XDSSubmissionSetContentType
      ctc.classificationScheme = 'urn:uuid:aa543740-bdda-424e-8c96-df4873be8500' AND    # $XDSSubmissionSetContentType
      ctc.nodeRepresentation IN $XDSSubmissionSetContentType  )                         # $XDSSubmissionSetContentType
# status
  AND ss.status = 'Approved'
```

3.16.4.1.4.3 FindFolders

Find folders (XDSFolder objects) in the registry for a given patientID with matching 'status' attribute. The other parameters can be used to restrict the collection of XDSFolder objects returned.

Returns: XDSFolder objects matching the query parameters

Parameter Name	Attribute	Opt	Mult

Parameter Name	Attribute	Opt	Mult
$XDSFolderPatientId	XDSFolder.patientId	R	--
$XDSFolderLastUpdateTimeFrom	XDSFolder. lastUpdateTime lower value	O	--
$XDSFolderLastUpdateTimeTo	XDSFolder. lastUpdateTime upper bound	O	--
$XDSFolderCodeList	XDSFolder. codeList	O	M
$XDSFolderCodeListScheme	XDSFolder. codeList1	O2	M2
$XDSFolderStatus	XDSFolder.status	R	M

[1]This attribute is not listed by name in Table 3.14.4.1-5 Document Metadata Attribute Definition. It is included in the documentation of the attribute that bears the same name without the 'Scheme' suffix. As an example, XDSDocumentEntry.ClassCodeScheme is not an attribute listed in attribute table for XDSDocumenEntry. It is documented as part of XDSDocumentEntry.classCode, specifically as the 'codingScheme' Slot.

[2]This parameter is optional but if included shall 1) have the same number of values as the corresponding 'code' attribute, 2) the i^{th} value of the 'code' attribute (e.g. XDSDocumentEntryClassCode) shall correspond to the i^{th} value of the 'codeScheme' attribute (e.g. XDSDocumentEntryClassCodeScheme).

Example SQL

```
SELECT fol.id
FROM RegistryPackage fol, ExternalIdentifier patId
 , Slot lupdateTimef      # $XDSFolderLastUpdateTimeFrom
 , Slot lupdateTimet      # $XDSFolderLastUpdateTimeTo
 , Classification cl      # $XDSFolderCodeList
 , Slot clScheme          # $XDSFolderCodeListScheme
     WHERE
# patientID
     ( patId.registryobject = fol.id  AND
     patId.identificationScheme = 'urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a' AND
     patId.value = $XDSFolderPatientId )
# last update time from
 AND ( lupdateTimef.parent = fol.id AND                                          # $XDSFolderLastUpdateTimeFrom
     lupdateTimef.name = 'lastUpdateTime' AND                                    # $XDSFolderLastUpdateTimeFrom
     lupdateTimef.value >= $XDSFolderLastUpdateTimeFrom )                        # $XDSFolderLastUpdateTimeFrom
# last update time to
 AND ( lupdateTimet.parent = fol.id AND                                          # $XDSFolderLastUpdateTimeTo
     lupdateTimet.name = 'lastUpdateTime' AND                                    # $XDSFolderLastUpdateTimeTo
     lupdateTimet.value &lt; $XDSFolderLastUpdateTimeTo )                        # $XDSFolderLastUpdateTimeTo
# code list
 AND ( cl.classifiedObject = fol.id AND                                          # $XDSFolderCodeList
     cl.classificationScheme = 'urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5' AND  # $XDSFolderCodeList
     cl.nodeRepresentation IN $XDSFolderCodeList )                               # $XDSFolderCodeList
# code list Scheme
#   This coding depends on the above clause being included.
  AND (clScheme.parent = cl.id AND                                               # $XDSFolderCodeListScheme
      clScheme.name = 'codingScheme' AND                                         # $XDSFolderCodeListScheme
      clScheme.value IN $XDSFolderCodeListScheme )                               # $XDSFolderCodeListScheme
# status
AND fol.status = 'Approved'
```

3.16.4.1.4.4 GetAll

Get all registry content for a patient given the indicated status, format codes, and confidentiality codes.

Returns:

- XDSSubmissionSet, XDSDocumentEntry, and XDSFolder objects with patientId attribute matching $patientId parameter
- Association objects with sourceObject or targetObject attribute matching one of the above objects

Parameter Name	Attribute	Opt	Mult

Parameter Name	Attribute	Opt	Mult
$patientId	XDSFolder. patientId, XDSSubmissionSet. patientId, XDSDocumentEntry. patientId	R	--
$XDSDocumentEntryStatus	XDSDocumentEntry. status	R	M
$XDSSubmissionSetStatus	XDSSubmissionSet. status	R	M
$XDSFolderStatus	XDSFolder. status	R	M
$XDSDocumentEntryFormatCode	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryConfidentialityCode	XDSDocumentEntry. confidentialityCode	O	M

Because of the limits placed on the usage of the SQL query language by ebXML Registry version 2.1, this query cannot be reasonably implemented as a single query. The following piece-wise queries return

XDSDocumentEntry objects

XDSSubmissionSet and XDSFolder objects

Association objects

Example SQL Part 1

```
SELECT eo.id
FROM ExtrinsicObject eo, ExternalIdentifier patId
 , Classification cCode         # $XDSDocumentEntryConfidentialityCode
 , Classification fmtCode         # $XDSDocumentEntryFormatCode
WHERE
  eo.status IN $XDSDocumentEntryStatus AND
  eo.objectType = 'urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1' AND
  patId.registryObject = eo.id AND
  patId.identificationScheme = 'urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427' AND
  patId.value = $patientId
 AND ( cCode.classifiedObject = eo.id AND                                    #
$XDSDocumentEntryConfidentialityCode
      cCode.classificationScheme = 'urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f' AND    #
$XDSDocumentEntryConfidentialityCode
      cCode.nodeRepresentation IN $XDSDocumentEntryConfidentialityCode  )                #
$XDSDocumentEntryConfidentialityCode
# format code
  AND (fmtCode.classifiedObject = doc.id AND                                  #
$XDSDocumentEntryFormatCode
      fmtCode.classificationScheme = 'urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d' AND  #
$XDSDocumentEntryFormatCode
      fmtCode.nodeRepresentation IN $XDSDocumentEntryFormatCode)                  #
$XDSDocumentEntryFormatCode
```

Example SQL Part 2

```
SELECT rp.id FROM RegistryPackage rp, Classification cl, ExternalIdentifier patId
WHERE
 (
  rp.status IN $XDSSubmissionSetStatus AND
  cl.classifiedObject = rp.id AND
  cl.classificationNode = 'urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd' AND
  patId.registryObject = rp.id AND
  patId.identificationScheme = 'urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446' AND
  patId.value = $patientId
 )
OR
 (
  rp.status IN $XDSFolderStatus AND
  cl.classifiedObject = rp.id AND
  cl.classificationNode = 'urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2' AND
  patId.registryObject = rp.id AND
```



```
  patId.identificationScheme = 'urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a' AND
  patId.value = $patientId
 )
```

Example SQL Part 3

```
SELECT DISTINCT ass.id FROM Association ass, ExtrinsicObject eo, RegistryPackage ss, RegistryPackage fol
WHERE
 (
       (ass.sourceObject = ss.id AND ass.targetObject = fol.id) OR
       (ass.sourceObject = ss.id AND ass.targetObject = eo.id) OR
       (ass.sourceObject = fol.id AND ass.targetObject = eo.id)


 ) AND
  eo.id IN
       (SELECT eo.id FROM ExtrinsicObject eo, ExternalIdentifier patId
              WHERE
                eo.status IN $XDSDocumentEntryStatus AND
                patId.registryObject = eo.id AND
                patId.identificationScheme = 'urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427' AND
                patId.value = $patientId) AND

  ss.id IN
        (SELECT ss.id FROM RegistryPackage ss, ExternalIdentifier patId
          WHERE
            ss.status IN $XDSSubmissionSetStatus AND
            patId.registryObject = ss.id AND
            patId.identificationScheme = 'urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446' AND
            patId.value = $patientId
        ) AND
   fol.id IN
        (SELECT fol.id FROM RegistryPackage fol, ExternalIdentifier patId
        WHERE
            fol.status IN $XDSFolderStatus AND
            patId.registryObject = fol.id AND
            patId.identificationScheme = 'urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a' AND
            patId.value = $patientId
       )
```

3.16.4.1.4.5 GetDocument

Retrieve a collection of XDSDocumentEntry objects. XDSDocumentEntry objects are selected either by their entryUUID or uniqueId attribute.

Returns: XDSDocumentEntry objects requested

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M

1 - Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

Example SQL

```
# UUID param
```



```
SELECT doc.id                                                                    #
$XDSDocumentEntryEntryUUID
FROM ExtrinsicObject doc                                                         #
$XDSDocumentEntryEntryUUID
WHERE  doc.id IN $XDSDocumentEntryEntryUUID                                      #
$XDSDocumentEntryEntryUUID
# uniquid param
   SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier uniqId             #
$XDSDocumentEntryUniqueId
   WHERE                                                                         #
$XDSDocumentEntryUniqueId
      uniqId.registryobject = doc.id AND                                         #
$XDSDocumentEntryUniqueId
      uniqId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND   #
$XDSDocumentEntryUniqueId
      uniqId.value IN $XDSDocumentEntryUniqueId                                  #
$XDSDocumentEntryUniqueId
```

3.16.4.1.4.6 GetSubmissionSetContentsAndContents

Retrieve an XDSSubmissionSet object along with its contents. XDSSubmissionSet objects are selected either by their entryUUID or uniqueId attribute. The XDSDocumentEntry objects returned shall match one of the confidentiality

codes listed if that parameter is included.

Returns:

- XDSSubmissionSet object specified in the query
- Association objects with type HasMember whose sourceObject attribute references the above XDSSubmissionSet object
- XDSDocumentEntry and XDSFolder objects referenced by the targetObject attribute of one of the above Associations

Parameter Name	Attribute	Opt	Mult
$XDSSubmissionSetEntryUUID	XDSSubmissionSet. entryUUID	O[1]	--
$XDSSubmissionSetUniqueId	XDSSubmissionSet. uniqueId	O[1]	--
$XDSDocumentEntryFormatCode	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryConfidentialityCode	XDSDocumentEntry. confidentialityCode	O	M

1 - Either $XDSSubmissionSetEntryUUID or $XDSSubmissionSetUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

This query is specified as 4 component queries: submission set, documents in the submission set, folders, and associations. The following special instructions apply:

- The variable $ssuuid in parts 2 and 3 and 4 must be replaced with the UUID of the registry package returned in part 1.
- The variable $docuuids in part 4 must be replaced with the UUIDs, in list format, of the documents returned in part 2
- The variable $foluuids in part 4 must be replaced with the UUIDs, in list format, of the folders returned in part 3

Example SQL Part 1

```
# get submission set by uuid
SELECT ss.id FROM RegistryPackage ss                                                    #
$XDSSubmissionSetEntryUUID
WHERE                                                                                   #
$XDSSubmissionSetEntryUUID
       ss.id = $XDSSubmissionSetEntryUUID                                              #
$XDSSubmissionSetEntryUUID
# get submission set by uniqueId
SELECT ss.id FROM RegistryPackage ss, ExternalIdentifier uniq                           #
$XDSSubmissionSetUniqueId
WHERE                                                                                   #
$XDSSubmissionSetUniqueId
  uniq.registryObject = ss.id AND                                                       #
$XDSSubmissionSetUniqueId
  uniq.identificationScheme = 'urn:uuid:96fdda7c-d067-4183-912e-bf5ee74998a8' AND    #
$XDSSubmissionSetUniqueId
  uniq.value = $XDSSubmissionSetUniqueId                                                #
$XDSSubmissionSetUniqueId
```

Example SQL Part 2

```
# get docs based on submission set uuid
SELECT doc.id FROM ExtrinsicObject doc, Association a
, Classification conf       # $XDSDocumentEntryConfidentialityCode
, Classfication fmtCode     # $XDSDocumentEntryFormatCode
WHERE
  a.sourceObject = $ssuuid AND
  a.associationType = 'HasMember' AND
  a.targetObject = doc.id
 AND (
  conf.classificationScheme = 'urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f' AND        #
$XDSDocumentEntryConfidentialityCode
  conf.classifiedObject = doc.id AND                                                    #
$XDSDocumentEntryConfidentialityCode
  conf.nodeRepresentation IN $XDSDocumentEntryConfidentialityCode)                      #
$XDSDocumentEntryConfidentialityCode
  AND (fmtCode.classifiedObject = doc.id AND                                            #
$XDSDocumentEntryFormatCode
      fmtCode.classificationScheme = 'urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d' AND #
$XDSDocumentEntryFormatCode
      fmtCode.nodeRepresentation IN $XDSDocumentEntryFormatCode)                        #
$XDSDocumentEntryFormatCode
```

Example SQL Part 3

```
SELECT fol.id FROM RegistryPackage fol, Association a
WHERE
  a.associationType = 'HasMember' AND
  a.sourceObject = $ssuuid AND
  a.targetObject = fol.id
```

Example SQL Part 4

```
SELECT ass.id FROM Association ass
WHERE
```



```
  ass.associationType = 'HasMember' AND
  ass.sourceObject = $ssuuid AND
  (
    ass.targetObject IN $docuuids OR
    ass.targetObject IN $foluuids
  )
```

3.16.4.1.4.7 GetFolderAndContents

Retrieve an XDSFolder object and its contents. XDSFolder objects are selected either by their entryUUID or uniqueId attribute. The XDSDocumentEntry objects returned shall match one of the confidentiality codes listed if that parameter is included.

Returns:

- XDSFolder object specified in the query
- Association objects of type HasMember that have a sourceObject attribute referencing the XDSFolder object specified in the query
- XDSDocumentEntry objects referenced by the targetObject attribute of one of the Association objects specified above

Parameter Name	Attribute	Opt	Mult
$XDSFolderEntryUUID	XDSFolder. entryUUID	O[1]	--
$XDSFolderUniqueId	XDSFolder. uniqueId	O[1]	--
$XDSDocumentEntryFormatCode	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryConfidentialityCode	XDSDocumentEntry. confidentialityCode	O	M

1 - Either $XDSFolderEntryUUID or $XDSFolderUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

This query is specified as 3 component queries: folder, documents in the folder, and associations linking the folder and documents. The following special instructions apply:

- The variable $foluuid in parts 2 and 3 must be replaced with the UUID of the registry package returned in part 1.
- The variable $docuuids in part 3 must be replaced with the UUIDs, in list format, of the documents returned in part 2

Example SQL Part 1

```
# get folder by uuid
SELECT fol.id FROM RegistryPackage fol                                    #
$XDSFolderEntryUUID
WHERE                                                                     #
$XDSFolderEntryUUID
       fol.id = $XDSFolderEntryUUID                                      #
$XDSFolderEntryUUID
# get folder by uniqueId
SELECT fol.id FROM RegistryPackage fol, ExternalIdentifier uniq           #
$XDSFolderUniqueId
```



```
WHERE                                                                          #
$XDSFolderUniqueId
  uniq.registryObject = fol.id AND                                             #
$XDSFolderUniqueId
  uniq.identificationScheme = 'urn:uuid:75df8f67-9973-4fbe-a900-df66cefecc5a' AND      #
$XDSFolderUniqueId
  uniq.value = $XDSFolderUniqueId                                              #
$XDSFolderUniqueId
```

Example SQL Part 2

```
# get docs based on folder uuid
SELECT doc.id FROM ExtrinsicObject doc, Association a
, Classification conf         # $XDSDocumentEntryConfidentialityCode
, Classfication fmtCode     # $XDSDocumentEntryFormatCode
WHERE
  a.sourceObject = $foluuid AND
  a.associationType = 'HasMember' AND
  a.targetObject = doc.id
 AND (                                                                         #
$XDSDocumentEntryConfidentialityCode
  conf.classificationScheme = 'urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f' AND        #
$XDSDocumentEntryConfidentialityCode
  conf.classifiedObject = doc.id AND                                           #
$XDSDocumentEntryConfidentialityCode
  conf.nodeRepresentation IN $XDSDocumentEntryConfidentialityCode)             #
$XDSDocumentEntryConfidentialityCode
# format code
  AND (fmtCode.classifiedObject = doc.id AND                                   #
$XDSDocumentEntryFormatCode
      fmtCode.classificationScheme = 'urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d' AND #
$XDSDocumentEntryFormatCode
      fmtCode.nodeRepresentation IN $XDSDocumentEntryFormatCode)               #
$XDSDocumentEntryFormatCode
```

Example SQL Part 3

```
SELECT ass.id FROM Association ass
WHERE
  ass.associationType = 'HasMember' AND
  ass.sourceObject = $foluuid AND
  ass.targetObject IN $docuuids
```

3.16.4.1.4.8 GetFoldersForDocument

Retrieve XDSFolder objects that contain the XDSDocumentEntry object provided with the query. XDSDocumentEntry objects are selected either by their entryUUID or uniqueId attribute.

Returns: XDSFolder objects that contain specified XDSDocumentEntry object. More specifically, for each Association object of type HasMember that has a targetObject attribute referencing the target XDSDocumentEntry object, return the object referenced by its sourceObject if it is of type XDSFolder.

Parameter Name	Attribute	Opt	Mult

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	--
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	--

1 - Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

Example SQL

```
SELECT fol.id FROM RegistryPackage fol, Association a, ExtrinsicObject doc, Classification c
WHERE
  doc.id IN
 (
# UUID param
SELECT doc.id                                                                    #
$XDSDocumentEntryEntryUUID
FROM ExtrinsicObject doc                                                         #
$XDSDocumentEntryEntryUUID
WHERE  doc.id = $XDSDocumentEntryEntryUUID                                       #
$XDSDocumentEntryEntryUUID
# uniquid param
   SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier uniqId             #
$XDSDocumentEntryUniqueId
   WHERE                                                                         #
$XDSDocumentEntryUniqueId
      uniqId.registryobject = doc.id AND                                         #
$XDSDocumentEntryUniqueId
      uniqId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND   #
$XDSDocumentEntryUniqueId
      uniqId.value = $XDSDocumentEntryUniqueId                                   #
$XDSDocumentEntryUniqueId
 ) AND
  a.targetObject = doc.id AND
  a.associationType = 'HasMember' AND
  a.sourceObject = fol.id AND
  c.classifiedObject = fol.id AND
  c.classificationNode = 'urn:uuid:d9d542f3-6cc4-48b6-8870-ea235fbc94c2'
```

3.16.4.1.4.9 GetRelatedDocuments

Retrieve XDSDocumentEntry objects that are related to the specified document via an Association object. The specified document is designated by UUID or uniqueId. The query shall return

- Association objects in which the sourceObject attribute OR the targetObject attribute references the specified document and the associationType attribute matches a value included in the $AssociationTypes parameter
- XDSDocumentEntry objects referenced by the targetObject attribute OR the sourceObject attribute of an Association object matched above

Note: A side effect of the query is that the specified document is returned in the results.

See ITI TF-2a: 3.14.4.1.2.6 Document Relationships and Associations for background.

Returns: XDSDocumentEntry objects and related Association objects

Given : An XDSDocumentEntry object and a collection of association types.

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	--
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	--
$AssociationTypes	Not a named attribute	R	M

1 - Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

This query is specified as 2 component queries: associations and documents linked by the associations. The following special instructions apply:

- The variable $assuuids in part 2 must be replaced with the UUIDs, in list format, of the association objects returned in part 1.

Example SQL Part 1

```
SELECT a.id FROM Association a, ExtrinsicObject doc
WHERE
  doc.id IN
 (
# UUID param
SELECT doc.id                                                                    #
$XDSDocumentEntryEntryUUID
FROM ExtrinsicObject doc                                                         #
$XDSDocumentEntryEntryUUID
WHERE  doc.id = $XDSDocumentEntryEntryUUID                                       #
$XDSDocumentEntryEntryUUID
# uniquid param
   SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier uniqId             #
$XDSDocumentEntryUniqueId
   WHERE                                                                         #
$XDSDocumentEntryUniqueId
      uniqId.registryobject = doc.id AND                                         #
$XDSDocumentEntryUniqueId
      uniqId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND   #
$XDSDocumentEntryUniqueId
      uniqId.value = $XDSDocumentEntryUniqueId                                   #
$XDSDocumentEntryUniqueId
 ) AND
  a.associationType IN $AssociationTypes AND
 (
  a.sourceObject = doc.id OR
  a.targetObject = doc.id
 )
```

Example SQL Part 2

```
SELECT doc.id FROM ExtrinsicObject doc, Association a
WHERE
  a.id IN $assuuids AND
 (
  doc.id = a.sourceObject OR
  doc.id = a.targetObject
 )
```


3.16.4.1.4.10 GetFolders

Retrieve a collection of XDSFolder objects. XDSFolder objects are selected either by their entryUUID or uniqueId attribute.

Returns: XDSFolder objects requested.

Parameter Name	Attribute	Opt	Mult
$XDSFolderEntryUUID	XDSFolder. entryUUID	O[1]	M
$XDSFolderUniqueId	XDSFolder. uniqueId	O[1]	M

1 - Either $XDSFolderEntryUUID or $XDSFolderUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

Example SQL

```
# by UUID
SELECT fol.id FROM RegistryPackage fol  # $XDSFolderEntryUUID
WHERE                                   # $XDSFolderEntryUUID
  fol.id IN $XDSFolderEntryUUID         # $XDSFolderEntryUUID
# by uniqueID
SELECT fol.id from RegistryPackage fol, ExternalIdentifier uniq                    #
$XDSFolderUniqueId
WHERE                                                                              #
$XDSFolderUniqueId
  uniq.registryObject = fol.id AND                                                 #
$XDSFolderUniqueId
  uniq.identificationScheme = 'urn:uuid:75df8f67-9973-4fbe-a900-df66cefecc5a' AND  #
$XDSFolderUniqueId
  uniq.value IN $XDSFolderUniqueId                                                 #
$XDSFolderUniqueId
```

Insert this new section. This section documents a new query.

3.16.4.1.4.11 GetAssociations

Retrieve Association objects whose sourceObject or targetObject attribute match $uuid.

Returns: Association objects

Parameter Name	Attribute	Opt	Mult
$uuid	None	O	M

Example SQL

```
SELECT DISTINCT ass.id FROM Association ass
WHERE
  ass.sourceObject IN $uuid OR
  ass.targetObject IN $uuid
```


3.16.4.1.4.12 GetDocumentsAndAssociations

Retrieve a collection of XDSDocumentEntry objects and the Association objects surrounding them. XDSDocumentEntry objects are selected either by their entryUUID or uniqueId attribute. This is the GetDocuments query and GetAssociations query combined into a single query.

Returns:

- XDSDocumentEntry objects
- Association objects whose sourceObject or targetObject attribute matches one of the above objects

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M

1 - Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

This query is specified in two parts: a query of XDSDocumentEntry objects and a separate query of Association objects.

Example SQL Part 1

```
# UUID param
SELECT doc.id                                                                    #
$XDSDocumentEntryEntryUUID
FROM ExtrinsicObject doc                                                         #
$XDSDocumentEntryEntryUUID
WHERE  doc.id IN $XDSDocumentEntryEntryUUID                                      #
$XDSDocumentEntryEntryUUID
# uniquid param
   SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier uniqId             #
$XDSDocumentEntryUniqueId
   WHERE                                                                         #
$XDSDocumentEntryUniqueId
      uniqId.registryobject = doc.id AND                                         #
$XDSDocumentEntryUniqueId
      uniqId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND   #
$XDSDocumentEntryUniqueId
      uniqId.value IN $XDSDocumentEntryUniqueId                                  #
$XDSDocumentEntryUniqueId
```

Example SQL Part 2

```
SELECT DISTINCT ass.id FROM Association ass                                      #
$XDSDocumentEntryEntryUUID
WHERE                                                                            #
$XDSDocumentEntryEntryUUID
  ass.sourceObject IN $XDSDocumentEntryEntryUUID OR                              #
$XDSDocumentEntryEntryUUID
  ass.targetObject IN $XDSDocumentEntryEntryUUID                                 #
$XDSDocumentEntryEntryUUID

SELECT DISTINCT ass.id FROM Association ass, ExtrinsicObject doc, ExternalIdentifier uniqId   #
$XDSDocumentEntryUniqueId
WHERE                                                                            #
$XDSDocumentEntryUniqueId
```



```
  uniqId.registryobject = doc.id AND                                                    #
$XDSDocumentEntryUniqueId
  uniqId.identificationScheme = 'urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab' AND     #
$XDSDocumentEntryUniqueId
  uniqId.value IN $XDSDocumentEntryUniqueId AND                                         #
$XDSDocumentEntryUniqueId
  (ass.sourceObject = doc.id OR                                                         #
$XDSDocumentEntryUniqueId
   ass.targetObject = doc.id)                                                           #
$XDSDocumentEntryUniqueId
```

3.16.4.1.4.13 GetSubmissionSets

Retrieve the XDSSubmissionSet objects used to submit a collection of XDSDocumentEntry and XDSFolder objects. The XDSDocumentEntry and XDSFolder objects of interest are identified by their UUIDs in the $uuid parameter.

Selection: XDSSubmissionSet objects are selected because Association objects exist that have:

- Type HasMember
- targetObject attribute containing one of the UUIDs provided in the $uuid parameter

sourceObject attribute referencing an XDSSubmissionSet object

Returns:

- XDSSubmissionSet objects described above

Association objects described in the Selection section above

Parameter Name	Attribute	Opt	Mult
$uuid	XDSDocumentEntry. entryUUID and XDSFolder. entryUUID	R	M

Example SQL

```
SELECT ss.id FROM RegistryPackage ss, Classification c, Association a
WHERE
  c.classifiedObject = ss.id AND
  c.classificationNode = 'urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd' AND
  a.sourceObject = ss.id AND
  a.associationType = 'HasMember' AND
  a.targetObject IN $uuid
```

3.16.4.1.5 Security considerations

The transaction shall be audited by the Document Consumer as follows

	Field Name	Opt.	Value Constraints
Event	EventID	M	EV (110112, DCM, “Query”)

	EventActionCode	M	EV "E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	*M*	*EV("ITI-16", "IHE Transactions", "Registry SQL Query") or EV("ITI-18", "IHE Transactions", "Registry Stored Query")*
Audit Source (Document Consumer) (1)			
Source (Document Consumer) (1)			
Destination (Document Registry) (1)			
Human Requestor (0..1)			
Patient (1)			
Query (1) (AudittMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	EV 2 (system object)
	ParticipantObjectTypeCodeRole	M	EV 24 (query)
	ParticipantObjectDataLifeCycle	*NA*	
	ParticipantObjectIDTypeCode	M	*EV("ITI-16", "IHE Transactions", "Registry SQL Query")*
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	(empty)
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	M	Shall hold the AdhocQueryRequest of the query, base64 encoded.
	ParticipantObjectDetail	*NA*	
	ParticipantObjectDescription	*U*	*not further specialized*

Note The ParticipantObjectQuery is encoded in base64 to simplify generic library processing of these audit records. Some other query sources generate queries in binary form. Defective query sources could generate queries that contain improper XML that would cause processing problems for this audit message. By encoding the contents, the generic audit libraries do not need to worry about correctness or format of the query request. Any request can be encoded as base64 without causing problems for the audit capture library.

It is furthermore expected that the analysis processing of audit records use the ParticipantObjectIDTypeCode to determine how to analyze the encoded query.

Using the following common blocks

Audit Source AuditMessage/ AuditSourceIdentification	AuditSourceID	M	The identity of the process issuing the audit message.
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Source AuditMessage/ ActiveParticipant	UserID	M	The identity of the Source. If the source logs, suggested format <process id>@<machine name>. If the destination logs, leave empty.
	AlternateUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	EV TRUE
	RoleIDCode	M	EV (110153, DCM, "Source")
	NetworkAccessPointTypeCode	MC	This field shall be present with values selected from those specified in RFC 3881 where applicable.
	NetworkAccessPointID	MC	Shall be present if Net Access Point Type Code is present. Shall use fields as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Destination process. For SOAP-based services, this is the SOAP endpoint URI.
	AlternateUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	EV FALSE
	RoleIDCode	M	EV (110152, DCM, "Destination")
	NetworkAccessPointTypeCode	MC	This field shall be present with values selected from those specified in RFC 3881 where applicable.
	NetworkAccessPointID	MC	Shall be present if Net Access Point Type Code is present. Shall use fields as specified in RFC 3881.

3420

Human Requestor **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternateUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	EV TRUE
	RoleIDCode	*U*	*not specialized*
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Source and destination shall always be specified and refer to the systems executing the transaction. If known, human requestor must be used to additionally identify the human that initiated the transaction.

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	EV 1 (person)
	ParticipantObjectTypeCodeRole	M	EV 1 (patient)
	ParticipantObjectDataLifeCycle	*NA*	
	ParticipantObjectIDTypeCode	M	EV 2 (patient number)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	XDS patientId, in CX format as constrained by XDS
	ParticipantObjectName	*U*	*The patient's name. Discouraged for privacy reasons.*
	ParticipantObjectQuery	*NA*	
	ParticipantObjectDetail	*NA*	
	ParticipantObjectDescription	*U*	*not further specialized*

3425

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.

3430 #### 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

3.16.4.2.4 Security considerations

The transaction shall be audited by the Document Registry as follows

	Field Name	Opt.	Value Constraints
Event AuditMessage/ EventIdentificration	EventID	M	EV (110112, DCM, "Query")
	EventActionCode	M	EV "E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	*M*	*EV("ITI-16", "IHE Transactions", "Registry SQL Query") or EV("ITI-18", "IHE Transactions", "Registry Stored Query")*
Audit Source (Document Registry) (1)			
Source (Document Source) (1)			
Destination (Document Registry) (1)			
Human Requestor (0..1)			
Patient (1)			
Query (1) (AudittMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	EV 2 (system object)
	ParticipantObjectTypeCodeRole	M	EV 24 (query)
	ParticipantObjectDataLifeCycle	*NA*	
	ParticipantObjectIDTypeCode	M	*EV("ITI-16", "IHE Transactions", "Registry SQL Query")*
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	(empty)
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	M	Shall hold the AdhocQueryRequest of the query, base64 encoded.
	ParticipantObjectDetail	*NA*	
	ParticipantObjectDescription	*U*	*not further specialized*

For queries that do not specify the patient ID in the query request, this ID must be filled from the query response.

Using the following common blocks

Audit Source AuditMessage/ AuditSourceIdentification	AuditSourceID	M	The identity of the process issuing the audit message.
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Source. If the source logs, suggested format <process id>@<machine name>. If the destination logs, leave empty.
	AlternateUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	EV TRUE
	RoleIDCode	M	EV (110153, DCM, "Source")
	NetworkAccessPointTypeCode	MC	This field shall be present with values selected from those specified in RFC 3881 where applicable.
	NetworkAccessPointID	MC	Shall be present if Net Access Point Type Code is present. Shall use fields as specified in RFC 3881.

3445

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Destination process. For SOAP-based services, this is the SOAP endpoint URI.
	AlternateUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	EV FALSE
	RoleIDCode	M	EV (110152, DCM, "Destination")
	NetworkAccessPointTypeCode	MC	This field shall be present with values selected from those specified in RFC 3881 where applicable.
	NetworkAccessPointID	MC	Shall be present if Net Access Point Type Code is present. Shall use fields as specified in RFC 3881.

Human Requestor **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternateUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	EV TRUE
	RoleIDCode	*U*	*not specialized*
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Source and destination shall always be specified and refer to the systems executing the transaction. If known, human requestor must be used to additionally identify the human that initiated the transaction.

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	EV 1 (person)
	ParticipantObjectTypeCodeRole	M	EV 1 (patient)
	ParticipantObjectDataLifeCycle	*NA*	
	ParticipantObjectIDTypeCode	M	EV 2 (patient number)
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	XDS patientId, in CX format as constrained by XDS
	ParticipantObjectName	*U*	*The patient's name. Discouraged for privacy reasons.*
	ParticipantObjectQuery	*NA*	
	ParticipantObjectDetail	*NA*	
	ParticipantObjectDescription	*U*	*not further specialized*

3.17 Retrieve Document

NOTE: The XDS.a profile is deprecated starting in 2009. However, editorial changes to remove references to XDS.a, and associated changes to the relevant transactions, will not be complete until IT Infrastructure Technical Framework Volume 2, Revision 7.0 is published in 2010.

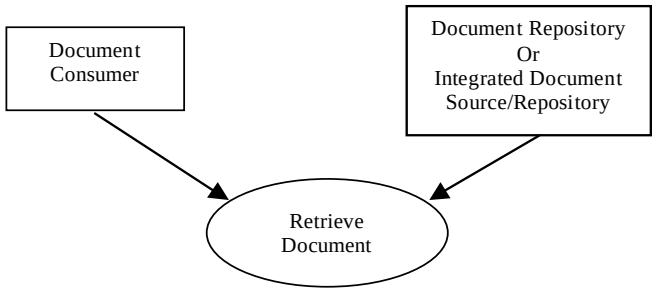
This section corresponds to Transaction ITI-17 of the IHE Technical Framework. The Document Consumer and Document Repository actors use transaction ITI-17.

Note: This transaction is used by the XDS.a Integration Profile. For a discussion on XDS.a and XDS.b Integration Profiles see IHE ITI TF-1: 10 (www.IHE.net/Technical _Frameworks).

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 or Integrated Document Source/Repository

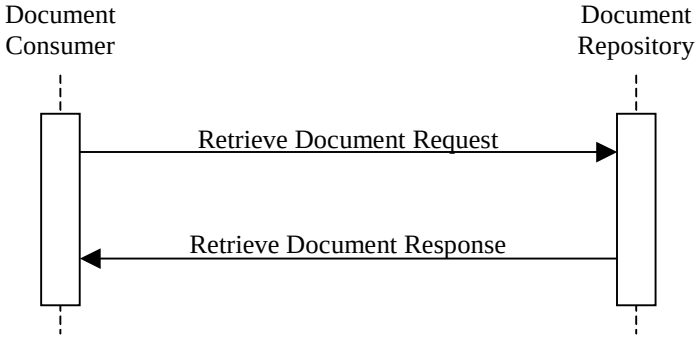
Role: Provides documents.

Note: Within this transaction, the Document Repository and Integrated Document Source/Repository actors can be used interchangeably.

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 Protocol2. 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 Support	Comments
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].

2 Ed Note: To allow common web browsers to be used without restriction.

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.

3.17.4.1.3 Expected Actions

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

3.17.4.1.3.1 Basic Patient Privacy Enforcement Option

If the Basic Patient Privacy Enforcement Option is implemented:

1. The Document Consumer actor shall abide by the XDS Affinity Domain Policies represented by the confidentialityCode in the metadata associated with the document. The Document Consumer actor likely will have user access controls or business rule capabilities to determine the details of how confidentiality codes apply to query results. The details of this are product specific and not specified by IHE. These rules shall reduce the query results to only those that are appropriate to the current situation for that actor and user.
2. The Document Consumer actor shall be able to be configured with Patient Privacy Consent Policies, Patient Privacy Consent Policy Identifiers (OIDs) and associated information necessary to understand and enforce the XDS Affinity Domain Policy. The details of this are product specific and not specified by IHE.

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:

3515

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 specification], and the document has not been modified since the requested modification date.	O
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:

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 XDSDocumentEntry. Content-Type is required in the response[3]. The other two are optional, but if present, must be the same as the values provided to the registry.
Content-Language Content-Length	O	
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.

3.17.4.2.3 Expected Actions

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

3.17.5 Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the Register Document transaction (see ITI TF-2a: 3.14.5.1).

3.17.5.1 Audit Record Considerations

The Retrieve Document Transaction is PHI-Export event, as defined in table 3.20.6-1. The Actors involved in the transaction shall create audit data in conformance with DICOM (Supp 95) “Data Export”/”Data Import”, with the following exceptions.

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

3 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.

transactions. Combination is permitted when the active participants and patient are the same, and the time difference is considered insignificant.

3.17.5.1.1 Document Consumer audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110107, DCM, "Import")
	EventActionCode	M	"C" (Create)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-17", "IHE Transactions", "Retrieve Document")
Source (Document Repository) (1)			
Destination (Document Consumer) (1)			
Human Requestor (0..n)			
Audit Source (Document Consumer) (1)			
Patient (0..1)			
Document URI (1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	*U*	*not specialized*
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	*U*	*not specialized*
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient (if-known) **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Document URI **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (System)
	ParticipantObjectTypeCodeRole	M	"3" (report)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(12, RFC-3881, "URI")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	Document URI
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	MC	Type=XDSDocumentEntry.uniqueId (the literal string), Value=the value of the Document Unique ID (from the XDS metadata)

3.17.5.1.2 Document Repository audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110106, DCM, "Export")
	EventActionCode	M	"R" (Read)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-17", "IHE Transactions", "Retrieve Document")
Source (Document Repository) (1)			
Destination (Document Consumer) (1)			
Audit Source (Document Repository) (1)			
Document URI(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	*U*	*not specialized*
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	*U*	*not specialized*
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

3550

Document URI **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (System)
	ParticipantObjectTypeCodeRole	M	"3" (report)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(12, RFC-3881, "URI")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	Document URI
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	MC	Type=XDSDocumentEntry.uniqueId (the literal string), Value=the value of the Document Unique ID (from the XDS metadata)

3.18 Registry Stored Query

NOTE: The XDS.a profile is deprecated starting in 2009. However, editorial changes to remove references to XDS.a, and associated changes to the relevant transactions, will not be complete until IT Infrastructure Technical Framework Volume 2, Revision 7.0 is published in 2010.

This section corresponds to Transaction 18 of the IHE Technical Framework. Transaction 18 is used by the Document Registry and Document Consumer actors.

Note: This transaction is used by the XDS.a and the XDS.b Integration Profile. For a discussion on XDS.a and XDS.b Integration Profiles see IHE ITI TF-1: 10 and IHE XDS.b Supplement (www.IHE.net/Technical _Frameworks).

3.18.1 Scope

The Registry Stored Query 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

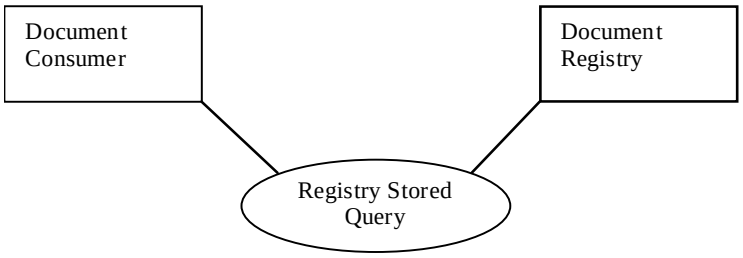
Query by time of submission

The list of XDS registry entries attributes that can be the target of a query are defined in ITI TF-3: 4.1.7 through 4.1.9. 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).

3.18.2 Use Case Roles



Actor: Document Consumer

Role: Requests a query by identifier (UUID), and passes parameters to the query. A parameter controlling the format of the returned data is passed, it selects either object references or full objects.

Actor: Document Registry

Role: Services the query using its stored definitions of the queries defined for XDS.

3.18.3 Referenced Standards

Implementors of this transaction shall comply with all requirements described in ITI TF-2x: Appendix V: Web Services for IHE Transactions except for the level of SOAP supported. The SOAP level support depends on the profile conformance of the implementing actor. The following table specifies the level of conformance to SOAP:

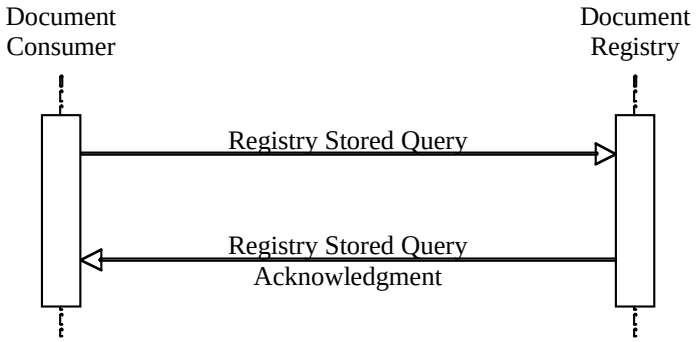
Profile implemented by actor	SOAP level required of actor
XDS.a	SOAP 1.1
XDS.b	SOAP 1.2
XDS.a & XDS.b	SOAP 1.1 and SOAP 1.2 for all Registry Stored Query requests without reconfiguration or restart

ebRIM OASIS/ebXML Registry Information Model v3.0

ebRS OASIS/ebXML Registry Services Specifications v3.0

Note: The ebRIM and ebRS standards referenced here are version 3.0. Other transactions of XDS.a reference version 2.1.

3.18.4 Interaction Diagram



3.18.4.1 Registry Stored Query

This is a query request to the Document Registry from a Document Consumer. The query request contains:

- A reference to a pre-defined query stored on the Document Registry actor.
- Parameters to the query. The query parameters are matched up with the query variables defined in the query definition on the Document Registry actor.

3.18.4.1.1 Trigger Events

This message is initiated when the Document Consumer wants to query/retrieve document metadata.

3.18.4.1.2 Message Semantics

The semantics of Stored Query are defined in section 6.3. *Stored Query Support* of ebRS version 3.0. This transaction corresponds to section 6.3.2 *Invoking a Stored Query* and 6.3.3 *Response to a Stored Query Invocation.* This profile does not specify how the queries come to be stored in the Registry actor nor how they are to be translated for other database architectures.

3.18.4.1.2.1 Version 3.0 ebXML Registry Standard

This transaction uses ebXML Registry version 3.0. The Invoke Stored Query message and the Invoke Stored Query Acknowledgement message shall be in version 3.0 format and be consistent with version 3.0 ebRIM and ebRS standards.

Version 3.0 ebXML Registry XML Schemas shall be used to validate the messages of this transaction. The major differences between version 2.1 and 3.0 of the Schema are:

- Different XML namespaces
- LeafRegistryObjectList element becomes RegistryObjectList
- ObjectType attribute changes format, changing from a text name to a UUID. For example, RegistryPackage becomes urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:RegistryPackage
- Status attribute value format changes from Approved to urn:oasis:names:tc:ebxml-regrep:StatusType:Approved
- Order of elements changes – Name, Description, Slot, Classification, ExternalIdentifier ordering becomes Slot, Name, Description, Classification, ExternalIdentifier.
- Id attribute is required for Classification, ExternalIdentifier, and Association
- The registryObject attribute is required on the ExternalIdentifier element.
- Association Types must be namespace qualified. For details see ITI TF-3: 4.1.6.3 Association type formatting.

It is the responsibility of the Document Registry actor to translate between version 2.1 and version 3.0 formats when returning v2.1 objects in v3.0 query responses.

3.18.4.1.2.2 Sample Query Request

The sample query is included under the ITI TF-2a: 3.18.4.1.3 Expected Actions.

3.18.4.1.2.3 Query Request Parameters – Coding Style

The ebXML Registry stored query facility (Invoke Stored Query transaction) accepts the following parameters:

- returnType – ‘LeafClass’ or ‘ObjectRef’
- Query ID – a UUID from the Stored Query IDs section (ITI TF-2a: 3.18.4.1.2.4) below
- Query Parameters – as defined in the Query Parameters section (ITI TF-2a: 3.18.4.1.2.3.7) below

3.18.4.1.2.3.1 Parameter returnType

Registry Stored Query 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

The 'LeafClass' returnType is meant for returning a small amount of fully specified ebXML objects (such as a list of ExtrinsicObject (XDSDocumentEntry) elements with full contents: slots, external identifiers, classifications etc.). This type of query result is self-contained, everything known about the object(s) is returned. The specific query documented in this section describes which object types will be included. ObjectRef elements are also returned. These represent objects not included in the returned object list that are referenced by objects in the returned object list. These ObjectRefs are optional by the registry standard version 3.0.

The 'ObjectRef' returnType returns references to the registry objects that match the query. This type query is recommended when the returned object list could be large. An initial query returning ObjectRefs for all objects of interest followed by secondary queries requesting full metadata (query type LeafClass) is an efficient way to query for large bodies of metadata. This strategy is particularly easy to use when querying for a single object type (XDSDocumentEntry or XDSSubmissionSet are examples) since only a single object type is involved.

An ObjectRef looks like:

```
<ObjectRef id="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"/>
```

3.18.4.1.2.3.2 Parameter Query ID

This parameter holds the UUID assigned to the query to be invoked. UUIDs are assigned by this profile (see ITI TF-2a: 3.18.4.1.2.4) to each of the queries defined in ITI TF-2a: 3.18.4.1.2.3.7.

3.18.4.1.2.3.3 Date/Time Coding

All Date/time values are to be inclusive, interpreted as:

$XDSDocumentEntryCreationTimeFrom <= XDSDocumentEntry.creationTime <
$XDSDocumentEntryCreationTimeTo

for example. The 'From' time or the 'To' time may be omitted.

3.18.4.1.2.3.4 Coding of Code/Code-Scheme

When specifying a coded value parameter, an abbreviated form of the HL7 V2.5 CE format shall be used. Only the first (identifier) and third (coding scheme) elements shall be specified. Both are required. The second element shall be empty. The HL7 V2.5 length limits shall not apply. The ebRIM limit on Slot Value size does apply. An example of this format is:

code^^coding-scheme

This style parameter always accepts multiple values so example codings in context look like:

<Value>('code1^^coding-scheme1')</Value>

or

<Value>('code1^^coding-scheme1','code2^^coding-scheme2')</Value>

within the parameter Slot.

3.18.4.1.2.3.5 Coding of Single/Multiple Values

Single values are coded as

- 123 - without quotes for numbers
- 'urn:oasis:names:tc:ebxml-regrep:StatusType:Approved' - in single quotes for strings.
- 'Children's Hospital' – a single quote is inserted in a string by specifying two single quotes

Within the LIKE predicate

- Underscore ('_') matches an arbitrary character
- Percent ('%') matches an arbitrary string

Format for multiple values is

- (value, value, value, …) OR
- (value) if only one value is to be specified.

where each value is coded as described above for single values.

When coding multiple values there is a potential conflict between needing to code a long list of values and the length restriction imposed by Schema on the size of the value of the <Value/> element. Slot values shall never exceed the Schema-enforced limit. Therefore, the use of multiple Value elements within the Slot shall be acceptable. Splits may occur only between values, where each Value element is surrounded by parentheses. The following example shows multiple values, split across multiple Value elements:

```
<Slot name="$uuid">
  <ValueList>
    <Value>('urn:uuid:a96d7361-6617-488a-891c-ee3f37d1f218','urn:uuid: 5655a680-1b6a-11dd-bd0b-0800200c9a66')</Value>
    <Value>('urn:uuid:ae315e81-2056-4829-a5b4-cf9531941f96')</Value>
  </ValueList>
</Slot>
```

This example shall be treated as equivalent to:

```
<Slot name="$uuid">
  <ValueList>
    <Value>('urn:uuid:a96d7361-6617-488a-891c-ee3f37d1f218','urn:uuid: 5655a680-1b6a-11dd-bd0b-0800200c9a66','urn:uuid:ae315e81-2056-4829-a5b4-cf9531941f96')</Value>
```



```
    </ValueList>
</Slot>
```

Character comparisons shall be performed in accordance with the rules in 4.2.

And/or semantics for the coding of parameters shall be available only on parameters for multi-valued metadata elements (such as $XDSDocumentEntryEventCodeList). . Multi-valued parameters shall be coded in two ways with different interpretations.

A parameter specified as a Slot with multiple values shall be interpreted as disjunction (OR semantics). For example:

```
<rim:Slot name="$XDSDocumentEntryEventCodeList">
  <rim:ValueList>
     <rim:Value>('a')</rim:Value>
     <rim:Value>('b')</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

shall match an XDSDocumentEntry object with an eventCodeList attribute containing either 'a' or 'b'. The following coding of the parameter shall yield the same results:

```
<rim:Slot name="$XDSDocumentEntryEventCodeList">
  <rim:ValueList>
     <rim:Value>('a','b')</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

A parameter specified as multiple Slots shall be interpreted as conjunction (AND semantics). For example:

```
<rim:Slot name="$XDSDocumentEntryEventCodeList">
  <rim:ValueList>
     <rim:Value>('a')</rim:Value>
  </rim:ValueList>
</rim:Slot>
<rim:Slot name="$XDSDocumentEntryEventCodeList">
  <rim:ValueList>
      <rim:Value>('b')</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

shall match an XDSDocumentEntry object with an eventCodeList attribute containing both 'a' and 'b'.

Furthermore, the following specification of the $XDSDocumentEntryEventCodeList parameter:

```
<rim:Slot name="$XDSDocumentEntryEventCodeList">
  <rim:ValueList>
     <rim:Value>('a','b')</rim:Value>
  </rim:ValueList>
</rim:Slot>
<rim:Slot name="$XDSDocumentEntryEventCodeList">
  <rim:ValueList>
     <rim:Value>('c')</rim:Value>
  </rim:ValueList>
</rim:Slot>
```

shall be interpreted as matching a document having eventCode (a OR b) AND c.

3.18.4.1.2.3.6 Valid Document Status Values

The Registry Object status values, in ebRIM v 3.0 format, used by XDS are:

```
urn:oasis:names:tc:ebxml-regrep:StatusType:Submitted
urn:oasis:names:tc:ebxml-regrep:StatusType:Approved
```


urn:oasis:names:tc:ebxml-regrep:StatusType:Deprecated

3.18.4.1.2.3.6.1 Valid AdhocQueryResponse Status Values

The status attribute of AdhocQueryResponse shall contain one of the following values:

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

See ITI TF-3: 4.1.13 Error Reporting for the interpretation of these values.

3.18.4.1.2.3.7 Parameters for Required Queries

The sections below document the queries defined in the Query Registry transaction [ITI-16]. These sections document a collection of Stored Queries. Document Registry actors implementing this transaction shall support all queries in this collection and all parameters defined for each query. Document Consumer actors implementing this transaction shall implement one or more of these queries as needed to support the use cases it implements.

Note that dollar sign ($) prefix on query parameters is required by ebRS 3.0.

In the query parameter tables below, each row represents a query parameter. Optional parameters which are not included in the query invocation have no affect on the query. Queries return registry objects that match all the supplied parameters. When multiple values are included for a parameter, objects are returned that match any included value (within the context of the larger query).

3.18.4.1.2.3.7.1 FindDocuments

Find documents (XDSDocumentEntry objects) in the registry for a given patientID with a matching 'status' attribute. The other parameters can be used to restrict the set of XDSDocumentEntry objects returned.

Returns: XDSDocumentEntry objects matching the query parameters

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	R	--
$XDSDocumentEntryClassCode[1]	XDSDocumentEntry. classCode	O	M
$XDSDocumentEntryTypeCode[1]	XDSDocumentEntry.typeCode	O	M
$XDSDocumentEntryPracticeSettingCode[1]	XDSDocumentEntry. practiceSettingCode	O	M
$XDSDocumentEntryCreationTimeFrom	Lower value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryCreationTimeTo	Upper value of XDSDocumentEntry. creationTime	O	--
$XDSDocumentEntryServiceStartTimeFrom	Lower value of XDSDocumentEntry.	O	--

Parameter Name	Attribute	Opt	Mult
	serviceStartTime		
$XDSDocumentEntryServiceStartTimeTo	Upper value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStopTimeFrom	Lower value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryServiceStopTimeTo	Upper value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryHealthcareFacilityTypeCode[1]	XDSDocumentEntry. healthcareFacilityTypeCode	O	M
$XDSDocumentEntryEventCodeList[1]	XDSDocumentEntry. eventCodeList[3]	O	M
$XDSDocumentEntryConfidentialityCode[1]	XDSDocumentEntry. confidentialityCode[3]	O	M
$XDSDocumentEntryAuthorPerson[4]	XDSDocumentEntry. author	O	M
$XDSDocumentEntryFormatCode[1]	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryStatus	XDSDocumentEntry. status	R	M

[1]Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

[3]Supports AND/OR semantics as specified in ITI TF-2a: 3.18.4.1.2.3.5.

[4]The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute)

3.18.4.1.2.3.7.2 FindSubmissionSets

Find submission sets (XDSSubmissionSet objects) in the registry for a given patientID with matching 'status' attribute. The other parameters can be used to restrict the collection of XDSSubmissionSet objects returned.

Returns: XDSSubmissionSet objects matching the query parameters

Parameter Name	Attribute	Opt	Mult
$XDSSubmissionSetPatientId	XDSSubmissionSet. patientId	R	--
$XDSSubmissionSetSourceId	XDSSubmissionSet. sourceId	O	M
$XDSSubmissionSetSubmissionTimeFrom	XDSSubmissionSet. submissionTime Lower value	O	--
$XDSSubmissionSetSubmissionTimeTo	XDSSubmissionSet. submissionTime Upper value	O	--
$XDSSubmissionSetAuthorPerson[1]	XDSSubmissionSet. authorPerson	O	--
$XDSSubmissionSetContentType[2]	XDSSubmissionSet. contentTypeCode	O	M
$XDSSubmissionSetStatus	XDSSubmissionSet. status	R	M

[1]The value for this parameter is a pattern compatible with the SQL keyword LIKE which allows the use of the following wildcard characters: % to match any (or no) characters and _ to match a single character. The match shall be applied to the text contained in the Value elements of the authorPerson Slot on the author Classification (value strings of the authorPerson sub-attribute).

[2]Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

3.18.4.1.2.3.7.3 FindFolders

Find folders (XDSFolder objects) in the registry for a given patientID with matching 'status' attribute. The other parameters can be used to restrict the collection of XDSFolder objects returned.

Returns: XDSFolder objects matching the query parameters

Parameter Name	Attribute	Opt	Mult
$XDSFolderPatientId	XDSFolder.patientId	R	--
$XDSFolderLastUpdateTimeFrom	XDSFolder. lastUpdateTime lower value	O	--
$XDSFolderLastUpdateTimeTo	XDSFolder. lastUpdateTime upper bound	O	--
$XDSFolderCodeList[1,3]	XDSFolder. codeList	O	M
$XDSFolderStatus	XDSFolder.status	R	M

[1]Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

[3]Supports AND/OR semantics as specified in ITI TF-2a: 3.18.4.1.2.3.5.

3.18.4.1.2.3.7.4 GetAll

Get all registry content for a patient given the indicated status, format codes, and confidentiality codes.

Returns:

- XDSSubmissionSet, XDSDocumentEntry, and XDSFolder objects with patientId attribute matching $patientId parameter
- Association objects with sourceObject or targetObject attribute matching one of the above objects

Parameter Name	Attribute	Opt	Mult
$patientId	XDSFolder. patientId, XDSSubmissionSet. patientId, XDSDocumentEntry. patientId	R	--
$XDSDocumentEntryStatus	XDSDocumentEntry. status	R	M
$XDSSubmissionSetStatus	XDSSubmissionSet. status	R	M
$XDSFolderStatus	XDSFolder. status	R	M
$XDSDocumentEntryFormatCode[2]	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryConfidentialityCode[1, 2]	XDSDocumentEntry. confidentialityCode[1]	O	M

[1]Supports AND/OR semantics as specified in ITI TF-2a: 3.18.4.1.2.3.5.

[2]Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme

3.18.4.1.2.3.7.5 GetDocuments

Retrieve a collection of XDSDocumentEntry objects. XDSDocumentEntry objects are selected either by their entryUUID or uniqueId attribute.

Returns: XDSDocumentEntry objects requested

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M
$homeCommunityId	None	O[2]	--

[1]Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[2]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: <AdhocQuery id="…" home="urn:oid:1.2.3" … >. Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

Note: A query for a single XDSDocumentEntry.uniqueId can return multiple results. See ITI TF-3: 4.1.4 under the topic of Document metadata duplication for explanation.

3.18.4.1.2.3.7.6 GetFolders

Retrieve a collection of XDSFolder objects. XDSFolder objects are selected either by their entryUUID or uniqueId attribute.

Returns: XDSFolder objects requested.

Parameter Name	Attribute	Opt	Mult
$XDSFolderEntryUUID	XDSFolder. entryUUID	O[1]	M
$XDSFolderUniqueId	XDSFolder. uniqueId	O[1]	M
$homeCommunityId	None	O[2]	--

[1]Either $XDSFolderEntryUUID or $XDSFolderUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[2]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: <AdhocQuery id="…" home="urn:oid:1.2.3" … >. Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

3.18.4.1.2.3.7.7 GetAssociations

Retrieve Association objects whose sourceObject or targetObject attribute match $uuid.

Returns: Association objects

Parameter Name	Attribute	Opt	Mult
$uuid	None	R	M

Parameter Name	Attribute	Opt	Mult
$homeCommunityId	None	O[1]	-

[1]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: `<AdhocQuery id="…" home="urn:oid:1.2.3" … >.` Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

3.18.4.1.2.3.7.8 GetDocumentsAndAssociations

Retrieve a collection of XDSDocumentEntry objects and the Association objects surrounding them. XDSDocumentEntry objects are selected either by their entryUUID or uniqueId attribute. This is the GetDocuments query and GetAssociations query combined into a single query.

Returns:

- XDSDocumentEntry objects
- Association objects whose sourceObject or targetObject attribute matches one of the above objects

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	M
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	M
$homeCommunityId	None	O[2]	--

[1]Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[2]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: <AdhocQuery id="…" home="urn:oid:1.2.3" … >. Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

3.18.4.1.2.3.7.9 GetSubmissionSets

Retrieve the XDSSubmissionSet objects used to submit a collection of XDSDocumentEntry and XDSFolder objects. The XDSDocumentEntry and XDSFolder objects of interest are identified by their UUIDs in the $uuid parameter.

Selection: XDSSubmissionSet objects are selected because Association objects exist that have:

- Type HasMember
- targetObject attribute containing one of the UUIDs provided in the $uuid parameter
- sourceObject attribute referencing an XDSSubmissionSet object

Returns:

- XDSSubmissionSet objects described above

- Association objects described in the Selection section above

Parameter Name	Attribute	Opt	Mult
$uuid	XDSDocumentEntry. entryUUID and XDSFolder. entryUUID	R	M
$homeCommunityId	None	O[1]	--

[1]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: `<AdhocQuery id="…" home="urn:oid:1.2.3" … >.` Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

3.18.4.1.2.3.7.10 GetSubmissionSetAndContents

Retrieve an XDSSubmissionSet object along with its contents. XDSSubmissionSet objects are selected either by their entryUUID or uniqueId attribute. The XDSDocumentEntry objects returned shall match:

- One of the confidentiality codes listed if that parameter is included and
- One of the format codes listed if that parameter is included

Returns:

- XDSSubmissionSet object specified in the query
- Association objects with type HasMember whose sourceObject attribute references the above XDSSubmissionSet object
- XDSDocumentEntry and XDSFolder objects referenced by the targetObject attribute of one of the above Associations
- Association object with type HasMember whose sourceObject attribute references an XDSFolder object targeted by the preceding rule and whose targetObject attribute references an XDSDocumentEntry object also targeted by the preceding rule.

Parameter Name	Attribute	Opt	Mult
$XDSSubmissionSetEntryUUID	XDSSubmissionSet. entryUUID	O[1]	--
$XDSSubmissionSetUniqueId	XDSSubmissionSet. uniqueId	O[1]	--
$XDSDocumentEntryFormatCode[4]	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryConfidentialityCode[4]	XDSDocumentEntry. confidentialityCode[2]	O	M
$homeCommunityId	None	O[3]	--

[1]Either $XDSSubmissionSetEntryUUID or $XDSSubmissionSetUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[2]Supports AND/OR semantics as specified in ITI TF-2a: 3.18.4.1.2.3.5.

[3]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: <AdhocQuery id="…" home="urn:oid:1.2.3" … >. Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the

previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

[4]Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

3.18.4.1.2.3.7.11 GetFolderAndContents

Retrieve an XDSFolder object and its contents. XDSFolder objects are selected either by their entryUUID or uniqueId attribute. The XDSDocumentEntry objects returned shall match one of the confidentiality codes listed if that parameter is included.

Returns:

- XDSFolder object specified in the query
- Association objects of type HasMember that have a sourceObject attribute referencing the XDSFolder object specified in the query
- XDSDocumentEntry objects referenced by the targetObject attribute of one of the Association objects specified above

Parameter Name	Attribute	Opt	Mult
$XDSFolderEntryUUID	XDSFolder. entryUUID	O[1]	--
$XDSFolderUniqueId	XDSFolder. uniqueId	O[1]	--
$XDSDocumentEntryFormatCode[4]	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryConfidentialityCode[4]	XDSDocumentEntry. confidentialityCode[2]	O	M
$homeCommunityId	None	O[3]	--

[1]Either $XDSFolderEntryUUID or $XDSFolderUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[2]Supports AND/OR semantics as specified in ITI TF-2a: 3.18.4.1.2.3.5.

[3]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: `<AdhocQuery id="…" home="urn:oid:1.2.3" … >.` Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

[4]Shall be coded according to specification in ITI TF-2a: 3.18.4.1.2.3.4 Coding of Code/Code-Scheme.

3.18.4.1.2.3.7.12 GetFoldersForDocument

Retrieve XDSFolder objects that contain the XDSDocumentEntry object provided with the query. XDSDocumentEntry objects are selected either by their entryUUID or uniqueId attribute.

Returns: XDSFolder objects that contain specified XDSDocumentEntry object. More specifically, for each Association object of type HasMember that has a targetObject attribute referencing the target XDSDocumentEntry object, return the object referenced by its sourceObject if it is of type XDSFolder.

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	--
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	--
$homeCommunityId	None	O[2]	--

[1]Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[2]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: <AdhocQuery id="…" home="urn:oid:1.2.3" … >. Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

Note: A query for a single XDSDocumentEntry.uniqueId can return multiple results. See ITI TF-3: 4.1.4 under the topic of Document metadata duplication for explanation.

3.18.4.1.2.3.7.13 GetRelatedDocuments

Retrieve XDSDocumentEntry objects that are related to the specified document via Association objects. Also return the Association objects. The specified document is designated by UUID or uniqueId. The query shall return

- Association objects where:
    - The sourceObject attribute OR the targetObject attribute references the specified document AND
    - Both sourceObject attribute and targetObject attribute reference documents AND
    - The associationType attribute matches a value included in the $AssociationTypes parameter
- XDSDocumentEntry objects referenced by the targetObject attribute OR the sourceObject attribute of an Association object matched above.

Note: A side effect of the query is that the specified document is returned in the results if at least one Association is returned.

Note: A side effect of this query is that if the document specified by the $XDSDocumentEntryUUID or $XDSDocumentEntryUniqueId parameters has no associations linking it to other documents, then no documents and no associations are returned.

See ITI TF-3: 4.1.6 Document Relationships and Associations for background.

Returns: Association objects and related XDSDocumentEntry objects

Given : An XDSDocumentEntry object and a collection of association types.

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryEntryUUID	XDSDocumentEntry. entryUUID	O[1]	--
$XDSDocumentEntryUniqueId	XDSDocumentEntry. uniqueId	O[1]	--
$AssociationTypes	Not a named attribute	R	M
$homeCommunityId	None	O[2]	--

[1]Either $XDSDocumentEntryEntryUUID or $XDSDocumentEntryUniqueId shall be specified. This transaction shall return an error if both parameters are specified.

[2]The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in: <AdhocQuery id="…" home="urn:oid:1.2.3" … >. Document Consumer actors shall specify the homeCommunityId value if they received a value for this attribute as part of the previous Registry Stored Query response entry which contained the specified EntryUUID or UniqueID. See ITI TF-2a: 3.18.4.1.2.3.8 for more details.

Note: A query for a single XDSDocumentEntry.uniqueId can return multiple results. See ITI TF-3: 4.1.4 under the topic of Document metadata duplication for explanation.

3.18.4.1.2.3.8 Use of homeCommunityId

The Registry Stored Query makes use of the homeCommunityId which is a globally unique identifier for a community and is used to obtain the Web Services endpoint of services that provide access to data in that community. homeCommunityId is structured as an OID limited to 64 characters and specified in URI syntax, for example the homeCommunityId of 1.2.3 would be formatted as `urn:oid:1.2.3`.

Its use is as follows:

- It is returned within the response to Registry Stored Query transactions to indicate the association of a response element with a community. It is specified as the ebRIM 'home' attribute within the ExtrinsicObject, RegistryPackage and ObjectRef elements. Document Consumers process the value as an opaque unique identifier.
- It is an optional parameter to Registry Stored Query requests, not requiring a patient id parameter, and Retrieve Document Set requests to indicate which community to direct the request.

For stored queries which do not require the patient id as a parameter , meaning query by EntryUUID or UniqueID:

- If the Document Consumer received the EntryUUID or uniqueID in a previous Registry Stored Query response which contained a homeCommunityId, then the Document Consumer shall specify the homeCommunityId parameter.
- The homeCommunityId value is specified as the home attribute on the AdhocQuery element of the query request, as in:
  `<AdhocQuery id="…" home="urn:oid:1.2.3" … >`
- Each query request can have at most one homeCommunityId value. If the Document Consumer specifies multiple entryUUID or uniqueID values they must all be associated with the same homeCommunityId value. Multiple individual query requests can be used to retrieve data associated with different homeCommunityIds.

3.18.4.1.2.4 Stored Query IDs

The standard XDS queries are assigned the following Query IDs. These IDs are used in the AdhocQueryRequest to reference queries stored on the Document registry actor. Query IDs are in UUID format (RFC4122). An error shall be returned when an unsupported stored query ID is received.

Note: This query mechanism can be extended by adding a query by allocating a Query ID, defining query parameters, and implementing the query in the Document Registry.

Query Name	Query ID
FindDocuments	urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d
FindSubmissionSets	urn:uuid:f26abbcb-ac74-4422-8a30-edb644bbc1a9
FindFolders	urn:uuid:958f3006-baad-4929-a4de-ff1114824431
GetAll	urn:uuid:10b545ea-725c-446d-9b95-8aeb444eddf3
GetDocuments	urn:uuid:5c4f972b-d56b-40ac-a5fc-c8ca9b40b9d4
GetFolders	urn:uuid:5737b14c-8a1a-4539-b659-e03a34a5e1e4
GetAssociations	urn:uuid:a7ae438b-4bc2-4642-93e9-be891f7bb155
GetDocumentsAndAssociations	urn:uuid:bab9529a-4a10-40b3-a01f-f68a615d247a
GetSubmissionSets	urn:uuid:51224314-5390-4169-9b91-b1980040715a
GetSubmissionSetAndContents	urn:uuid:e8e3cb2c-e39c-46b9-99e4-c12f57260b83
GetFolderAndContents	urn:uuid:b909a503-523d-4517-8acf-8e5834dfc4c7
GetFoldersForDocument	urn:uuid:10cae35a-c7f9-4cf5-b61e-fc3278ffb578
GetRelatedDocuments	urn:uuid:d90e5407-b356-4d91-a89f-873917b4b0e6

3.18.4.1.2.5 Intentionally Left Blank

3.18.4.1.2.6 Managing Large Query Responses

EbXML version 3.0 supports query results pagination (ebRS version 3.0 chapter 6.2). The interactions between the stored query capability and the query results pagination capability within the standard have never been reconciled and are not recommended for use together. It is recommended instead that query pagination be implemented within the Document Consumer actor.

This can be accomplished by specifying returnType=”ObjectRef” on all large queries. This returns a list of references (UUIDs) instead of full objects (large XML structures). This is practical for queries returning thousands of objects. To construct a page for display, a small number of objects can be retrieved through a second query. This is repeated for each page. As an example, the following sequence of queries could be used to list a large number of documents:

- FindDocuments query with returnType=”ObjectRef” which returns a large collections of ObjectRefs (UUIDs)
- GetDocuments query with returnType=”LeafClass” issued with a subset of the above returned UUIDs which returns the details to construct one page of listing

OR

GetDocumentsAndAssocations query with returnType=”LeafClass” issued with a subset of the above returned UUIDs which returns the details to construct one page of listing. By retrieving the Association objects, the existence of document replacement, transformation, and ammedment can be included into the display.

3.18.4.1.2.7 Web Services Transport

The query request and response will be transmitted using Web Services, according to the requirements specified in ITI TF-2x: Appendix V. The specific values for the WSDL describing the Stored Query Service are described in this section.

The Document Registery actor shall accept a Registry Stored Query Request formatted as a SIMPLE SOAP message and respond with a Registry Stored Query Response formatted as a SIMPLE SOAP message. The Document Consumer actor shall generate the Registry Stored Query Request formatted as a SIMPLE SOAP message and accept a Registry Stored Query Response formatted as a SIMPLE SOAP message.

IHE-WSP201) The attribute /wsdl:definitions/@name shall be “DocumentRegistry”.

The following WSDL naming conventions shall apply:

```
wsdl:definitions/@name="DocumentRegistry":
query message    -> "RegistryStoredQuery_Message"
query response   -> "RegistryStoredQuery_Response_Message"
portType         -> "DocumentRegistry_PortType"
operation        -> "RegistryStoredQuery"
SOAP 1.2 binding -> "DocumentRegistry_Binding_Soap12"
SOAP 1.2 port    -> "DocumentRegistry_Port_Soap12"
SOAP 1.1 binding -> "DocumentRegistry_Binding_Soap11"
SOAP 1.1 port    -> "DocumentRegistry_Port_Soap11"
```

IHE-WSP202) The targetNamespace of the WSDL shall be “urn:ihe:iti:xds-b:2007”

These are the requirements for the Registry Stored Query transaction presented in the order in which they would appear in the WSDL definition:

- The following types shall be imported (xsd:import) in the /definitions/types section:
  - namespace=" urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0", schemaLocation="query.xsd"
- The /definitions/message/part/@element attribute of the Registry Stored Query Request message shall be defined as “query:AdhocQueryRequest”
- The /definitions/message/part/@element attribute of the Registry Stored Query Response message shall be defined as “query:AdhocQueryResponse”

- The /definitions/portType/operation/input/@wsaw:Action attribute for the Registry Stored Query Request message shall be defined as “urn:ihe:iti:2007:RegistryStoredQuery”
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Registry Stored Query Response message shall be defined as “urn:ihe:iti:2007:RegistryStoredQueryResponse”
- The /definitions/binding/operation/soap12:operation/@soapAction attribute should be defined as “urn:ihe:iti:2007:RegistryStoredQuery”

The following WSDL fragment shows an example of Registry Stored Query transaction definition:

```
<?xml version="1.0" encoding="utf-8"?>
<definitions ...>
  ...
  <types>
    <xsd:schema elementFormDefault="qualified" targetNamespace="urn:ihe:iti:xds-b:2007">
      <xsd:import
        namespace="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
        schemaLocation="schema\query.xsd"/>
      ...
    </xsd:schema>
  </types>
  <message name="RegistryStoredQuery_Message">
    <documentation>Registry Stored Query</documentation>
    <part name="body" element="query:AdhocQueryRequest"/>
  </message>
  <message name="RegistryStoredQueryResponse_Message">
    <documentation>Registry Stored Query Response</documentation>
    <part name="body" element="query:AdhocQueryResponse"/>
  </message>
  ...
  <portType name="DocumentRegistry_PortType">
    <operation name="DocumentRegistry_RegistryStoredQuery">
      <input message="ihe:RegistryStoredQuery_Message"
          wsaw:Action="urn:ihe:iti:2007:RegistryStoredQuery"/>
      <output message="ihe:RegistryStoredQueryResponse_Message"
          wsaw:Action="urn:ihe:iti:2007:RegistryStoredQueryResponse"/>
    </operation>
    ...
  </portType>
  ...
</definitions>
```

A full WSDL for the Document Repository and Document Registry actors is found in ITI TF-2x: Appendix W.

3.18.4.1.2.7.1 Sample SOAP Messages

The samples in the following two sections show a typical SOAP request and its relative SOAP response. The sample messages also show the WS-Addressing headers <a:Action/>, <a:MessageID/>, <a:ReplyTo/>…; these WS-Addressing headers are populated according to the IHE ITI TF-2x: Appendix V: Web Services for IHE Transactions. The body of the SOAP message is omitted for brevity; in a real scenario the empty element will be populated with the appropriate metadata.

Samples presented in this section are also available online on the IHE FTP site, see ITI TF-2x: Appendix W.

3.18.4.1.2.7.1.1 Sample Registry Stored Query SOAP Request

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
```



```
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">urn:ihe:iti:2007:RegistryStoredQuery</a:Action>
    <a:MessageID>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:MessageID>
    <a:ReplyTo s:mustUnderstand="1">>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To>http://localhost/service/IHEXDSRegistry.svc</a:To>
  </s:Header>
  <s:Body>
        <query:AdhocQueryRequest
                        xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
                        xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
                        xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">
                <query:ResponseOption returnComposedObjects="true" returnType="LeafClass"/>
                <rim:AdhocQuery id=" urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d ">
                        <rim:Slot name="$XDSDocumentEntryPatientId">
                                <rim:ValueList>

        <rim:Value>'st3498702^^^&amp;amp;1.3.6.1.4.1.21367.2005.3.7&amp;amp;ISO'</rim:Value>
                                </rim:ValueList>
                        </rim:Slot>
                        <rim:Slot name="$XDSDocumentEntryStatus">
                                <rim:ValueList>
                                        <rim:Value>('urn:oasis:names:tc:ebxml-
regrep:ResponseStatusType:Approved')</rim:Value>
                                </rim:ValueList>
                        </rim:Slot>
                        <rim:Slot name="$XDSDocumentEntryCreationTimeFrom">
                                <rim:ValueList>
                                        <rim:Value>200412252300</rim:Value>
                                </rim:ValueList>
                        </rim:Slot>
                        <rim:Slot name="$XDSDocumentEntryCreationTimeTo">
                                <rim:ValueList>
                                        <rim:Value>200501010800</rim:Value>
                                </rim:ValueList>
                        </rim:Slot>
                        <rim:Slot name="$XDSDocumentEntryHealthcareFacilityTypeCode">
                                <rim:ValueList>
                                        <rim:Value>('Emergency Department')</rim:Value>
                                </rim:ValueList>
                        </rim:Slot>
                </rim:AdhocQuery>
        </query:AdhocQueryRequest>
  </s:Body>
</s:Envelope>
```

3.18.4.1.2.7.1.2 Sample Registry Stored Query SOAP Response

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">urn:ihe:iti:2007:RegistryStoredQueryResponse</a:Action>
    <a:RelatesTo>urn:uuid:def119ad-dc13-49c1-a3c7-e3742531f9b3</a:RelatesTo>
  </s:Header>
  <s:Body>
    <query:AdhocQueryResponse xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"/>
  </s:Body>
</s:Envelope>
```

3.18.4.1.3 Expected Actions

The Document Registry actor shall

1. Accept a parameterized query in an AdhocQueryRequest message

2. Verify the required parameters are included in the request. Additionally, special rules documented in the above section 'Parameters for Required Queries' shall be verified.
3. Errors shall be returned for the following conditions:
   - Unknown query ID (error code XDSUnknownStoredQuery)
   - Required parameter missing (error code XDSStoredQueryParamNumber)

   See ITI TF-3: 4.1.13 Error Reporting for additional error codes and general information on formating error responses.
4. Retrieve the internal implementation template of the query based on the Query ID supplied in the query request. Substitute appropriate parameters as indicated in ITI TF-2a: 3.18.4.1.2.3.7 Parameters for Required Queries and execute the query.
   - The Document Registry shall accept the homeCommunityId value if it is specified in a Registry Stored Query request.
   - If a patient identifier specified as a parameter to the query is unknown to the Document Registry it shall return a sucessful response with no elements.
5. Return XML formatted metadata in an AdhocQueryResponse message.
   - The Document Registry may specify the homeCommunityID attribute on any appropriate elements The homeCommunityId attribute corresponds to the 'home' attribute specified in the ebRIM standard. For more information on homeCommunityId see ITI TF-2a: 3.18.4.1.2.3.8 . The elements that may include the home attribute are:
     - If returntype="LeafClass" the ExtrinsicObject and RegistryPackage elements shall contain the home attribute.
     - If returnType="ObjectRef" the ObjectRef element shall contain the home attribute

This transaction may return both errors and results in an AdhocQueryResponse message. To do this, the returned AdhocQueryResponse message would contain both a RegistryObjectList element and a RegistryErrorList element. See ITI TF-3: 4.1.13 for additional details on formatting of error responses.

3.18.4.1.3.1 Sample Query Request

This example query specifies:

- The FindDocuments query (id attribute of AdhocQuery element)
- patientID st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO
- Return Approved documents only
- Time range (creation time) 200412252300 to 200501010800
- Healthcare Facility Type Code of Emergency Department

Note that ebRS 3.0 specifies the use of Slot to specify name/value(s) pairs as parameters to a Stored Query.

Note: query parameter names are highlighted for readability.

```
<query:AdhocQueryRequest
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:query="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
    xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
    xmlns:rs="urn:oasis:names:tc:ebxml-regrep:xsd:rs:3.0">
  <query:ResponseOption returnComposedObjects="true" returnType="LeafClass"/>
  <rim:AdhocQuery id="urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d">
    <rim:Slot name="$XDSDocumentEntryPatientId">
      <rim:ValueList>
        <rim:Value>'st3498702^^^&amp;amp;1.3.6.1.4.1.21367.2005.3.7&amp;amp;ISO'</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryStatus">
      <rim:ValueList>
        <rim:Value>('urn:oasis:names:tc:ebxml-regrep:StatusType:Approved')</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryCreationTimeFrom">
      <rim:ValueList>
        <rim:Value>200412252300</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryCreationTimeTo">
      <rim:ValueList>
        <rim:Value>200501010800</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$XDSDocumentEntryHealthcareFacilityTypeCode">
      <rim:ValueList>
        <rim:Value>('Emergency Department')</rim:Value>
      </rim:ValueList>
    </rim:Slot>
  </rim:AdhocQuery>
</query:AdhocQueryRequest>
```

The following example shows a get documents query for XDSDocumentEntry objects for a specified list of entryUUIDs (urn:uuid:aff99222-18e3-4812-bc71-c410b2860e18, urn:uuid:aff99222-18e3-4812-bc71-c410b2860e19, urn:uuid:aff99222-18e3-4812-bc71-c410b2860e20) and corresponding homeCommunityId value (urn:oid:1.2.3):

```
<query:AdhocQueryRequest … >
    <query:ResponseOption returnComposedObjects="true"
returnType="LeafClass"/>
    <rim:AdhocQuery id="urn:uuid:5c4f972b-d56b-40ac-a5fc-c8ca9b40b9d4"
home="urn:oid:1.2.3">
        <rim:Slot name="$XDSDocumentEntryEntryUUID">
            <rim:ValueList>
              <rim:Value>
              ("urn:uuid:aff99222-18e3-4812-bc71-c410b2860e18",
               "urn:uuid:aff99222-18e3-4812-bc71-c410b2860e19",
               "urn:uuid:aff99222-18e3-4812-bc71-c410b2860e20")
               </rim:Value>
            </rim:ValueList>
        </rim:Slot>
    </rim:AdhocQuery>
</query:AdhocQueryRequest>
```


3.18.4.1.3.2 Intentionally Left Blank

3.18.4.1.3.3 Sample Query Response

This sample query response corresponds to the above query. Note that the query response message is coded in version 3.0 ebRIM and ebRS. This sample response and the ebXML Registry version 3.0 schema files are available online. The Implementation Guide found at http://wiki.ihe.net/index.php?title=ITI_Implementation_Guide contains such supplemental material.

```
<?xml version="1.0" encoding="UTF-8"?>
<AdhocQueryResponse
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0
      file:/Users/bill/RegSchema/V3.0/query.xsd"
    xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
    xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
    status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
  <rim:RegistryObjectList>
    <rim:ExtrinsicObject
        xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
        xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
        id="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
        isOpaque="false"
        mimeType="text/xml"
        objectType="urn:uuid:7edca82f-054d-47f2-a032-9b2a5b5186c1"
        status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved">
      <rim:Slot name="URI">
        <rim:ValueList>
          <rim:Value>http://localhost:8080/XDS/Repository/08a15a6f-5b4a-42de-8f95-
89474f83abdf.xml</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="authorInstitution">
        <rim:ValueList>
          <rim:Value>Some Hospital^^^^^^^^^1.2.3.4.5.6.7.8.9.1789.45</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="creationTime">
        <rim:ValueList>
          <rim:Value>200412261119</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="hash">
        <rim:ValueList>
          <rim:Value>4cf4f82d78b5e2aac35c31bca8cb79fe6bd6a41e</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="languageCode">
        <rim:ValueList>
          <rim:Value>en-us</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="serviceStartTime">
        <rim:ValueList>
          <rim:Value>200412230800</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="serviceStopTime">
        <rim:ValueList>
          <rim:Value>200412230801</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Slot name="size">
        <rim:ValueList>
```



```
        <rim:Value>54449</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="sourcePatientId">
      <rim:ValueList>
        <rim:Value>jd12323^^^wsh</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="sourcePatientInfo">
      <rim:ValueList>
        <rim:Value>PID-3|pid1^^^domain</rim:Value>
        <rim:Value>PID-5|Doe^John^^^</rim:Value>
        <rim:Value>PID-7|19560527</rim:Value>
        <rim:Value>PID-8|M</rim:Value>
        <rim:Value>PID-11|100 Main St^^Metropolis^Il^44130^USA</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Name>
      <rim:LocalizedString charset="UTF-8" value="Sample document 1" xml:lang="en-us"/>
    </rim:Name>
    <rim:Description/>
    <rim:Classification
        classificationScheme="urn:uuid:41a5887f-8865-4c09-adf7-e362475b143a"
        classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
        id="urn:uuid:ac872fc0-1c6e-439f-84d1-f76770a0ccdf"
        nodeRepresentation="Education"
        objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
      <rim:Slot name="codingScheme">
        <rim:ValueList>
          <rim:Value>Connect-a-thon classCodes</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Education" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
    </rim:Classification>
    <rim:Classification
        classificationScheme="urn:uuid:f4f85eac-e6cb-4883-b524-f2705394840f"
        classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
        id="urn:uuid:f1a8c8e4-3593-4777-b7e0-8b0773378705"
        nodeRepresentation="C"
        objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
      <rim:Slot name="codingScheme">
        <rim:ValueList>
          <rim:Value>Connect-a-thon confidentialityCodes</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Celebrity" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
    </rim:Classification>
    <rim:Classification
        classificationScheme="urn:uuid:a09d5840-386c-46f2-b5ad-9c3699a4309d"
        classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
        id="urn:uuid:b6e49c73-96c8-4058-8c95-914d83bd262a"
        nodeRepresentation="CDAR2/IHE 1.0"
        objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
      <rim:Slot name="codingScheme">
        <rim:ValueList>
          <rim:Value>Connect-a-thon formatCodes</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="CDAR2/IHE 1.0" xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
    </rim:Classification>
```



```
<rim:Classification
    classificationScheme="urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1"
    classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
    id="urn:uuid:61e2b376-d74a-4984-ac21-dcd0b8890f9d"
    nodeRepresentation="Emergency Department"
    objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>Connect-a-thon healthcareFacilityTypeCodes</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Assisted Living" xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:Classification>
<rim:Classification
    classificationScheme="urn:uuid:cccf5598-8b07-4b77-a05e-ae952c785ead"
    classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
    id="urn:uuid:fb7677c5-c42f-485d-9010-dce0f3cd4ad5"
    nodeRepresentation="Cardiology"
    objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>Connect-a-thon practiceSettingCodes</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Cardiology" xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:Classification>
<rim:Classification
    classificationScheme="urn:uuid:f0306f51-975f-434e-a61c-c59651d33983"
    classifiedObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
    id="urn:uuid:0a8a8ed9-8be5-4a63-9b68-a511adee8ed5"
    nodeRepresentation="34098-4"
    objectType="Urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Classification">
  <rim:Slot name="codingScheme">
    <rim:ValueList>
      <rim:Value>LOINC</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString
        charset="UTF-8"
        value="Conference Evaluation Note" xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:Classification>
<rim:ExternalIdentifier
    id="urn:uuid:db9f4438-ffff-435f-9d34-d76190728637"
    registryObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
    identificationScheme="urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427"
    objectType="ExternalIdentifier"
    value="st3498702^^^&amp;amp;1.3.6.1.4.1.21367.2005.3.7&amp;amp;ISO">
  <rim:Name>
    <rim:LocalizedString
        charset="UTF-8"
        value="XDSDocumentEntry.patientId"
        xml:lang="en-us"/>
  </rim:Name>
  <rim:Description/>
</rim:ExternalIdentifier>
<rim:ExternalIdentifier
    id="urn:uuid:c3fcbf0e-9765-4f5b-abaa-b37ac8ff05a5"
    registryObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
    identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"
    objectType="ExternalIdentifier"
```



```
        value="1.3.6.1.4.1.21367.2005.3.99.1.1010">
      <rim:Name>
        <rim:LocalizedString
            charset="UTF-8"
            value="XDSDocumentEntry.uniqueId"
            xml:lang="en-us"/>
      </rim:Name>
      <rim:Description/>
    </rim:ExternalIdentifier>
  </rim:ExtrinsicObject>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:41a5887f-8865-4c09-adf7-
e362475b143a"/>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:f4f85eac-e6cb-4883-b524-
f2705394840f"/>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:a09d5840-386c-46f2-b5ad-
9c3699a4309d"/>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:f33fb8ac-18af-42cc-ae0e-
ed0b0bdb91e1"/>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:cccf5598-8b07-4b77-a05e-
ae952c785ead"/>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:f0306f51-975f-434e-a61c-
c59651d33983"/>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:58a6f841-87b3-4a3e-92fd-
a8ffeff98427"/>
  <rim:ObjectRef xmlns:q="urn:oasis:names:tc:ebxml-regrep:xsd:query:3.0"
xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0" id="urn:uuid:2e82c1f6-a085-4c72-9da3-
8640a32e42ab"/>
 </rim:RegistryObjectList>
</AdhocQueryResponse>
```

The following query response is the same as above (repeated sections replaced with …) with the homeCommunityId attribute specified, in bold for readability. Subsequent requests specifying entryUUID of `urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf` or uniqueID of `1.3.6.1.4.1.21367.2005.3.99.1.1010` shall include the homeCommunityId value of urn:oid:1.2.3 in the query.

```
<?xml version="1.0" encoding="UTF-8"?>
<AdhocQueryResponse ... status="Success">
      <rim:RegistryObjectList>
         <rim:ExtrinsicObject … id="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
isOpaque="false"  mimeType="text/xml" objectType="urn:uuid:7edca82f-054d-47f2-a032-
9b2a5b5186c1" status="urn:oasis:names:tc:ebxml-regrep:StatusType:Approved"
home=”urn:oid:1.2.3”>

…

          <rim:ExternalIdentifier id="urn:uuid:c3fcbf0e-9765-4f5b-abaa-
b37ac8ff05a5" registryObject="urn:uuid:08a15a6f-5b4a-42de-8f95-89474f83abdf"
identificationScheme="urn:uuid:2e82c1f6-a085-4c72-9da3-8640a32e42ab"
objectType="ExternalIdentifier" value="1.3.6.1.4.1.21367.2005.3.99.1.1010">
               <rim:Name>
                  <rim:LocalizedString charset="UTF-8"
value="XDSDocumentEntry.uniqueId" xml:lang="en-us"/>
               </rim:Name>
```



```
                <rim:Description/>
            </rim:ExternalIdentifier>
         </rim:ExtrinsicObject>
      </rim:RegistryObjectList>
   </AdhocQueryResponse>
```

3.18.4.1.3.4 Intentionally Left Blank

3.18.4.1.3.5 Basic Patient Privacy Enforcement Option

If the Basic Patient Privacy Enforcement Option is implemented:

1. All Document Consumer Actors may provide a list of confidentialityCode in XDS Registry Stored Query Transaction and the XDS Registry will return only document that have at least one matching confidentialityCode. In this way documents without at least one of the requested codes will not be returned.
2. The Document Consumer actor shall be able to be configured with the Patient Privacy Consent Policies, Patient Privacy Consent Policy Identifiers (OIDs) and associated information necessary to understand and enforce the XDS Affinity Domain Policy. The details of this are product specific and not specified by IHE.
3. The Document Consumer shall not allow access to documents for which the Document Consumer does not understand at least one of the confidentialityCode returned. This assures that a Document Consumer will not improperly handle documents with confidentialityCode that may be more restrictive than the Document Consumer is configured to support.
4. The Document Consumer actor shall abide by the XDS Affinity Domain Policies represented by the confidentialityCode in the metadata associated with the document. The Document Consumer actor likely will have user access controls or business rule capabilities to determine the details of how confidentiality codes apply to query results. The details of this are product specific and not specified by IHE. These rules shall reduce the query results to only those that are appropriate to the current situation for that actor and user.
5. Note: The Registry is already required to return only documents that match the requested confidentialityCode (filter) indicated in the Registry Stored Query.
6. Note: Products implementing the Registry Actor may be able to further filter Registry Stored Query results through looking at all the Patient Privacy Acknowledgement Documents registered for the patient that have the availabilityStatus of Approved and for which have not expired.

3.18.4.1.3.6 Basic Patient Privacy Proof Option

If the Basic Patient Privacy Consents Proof Option is implemented:

1. The Document Consumer actor shall be capable of querying for ‘Approved’ Patient Privacy Consent Acknowledgement Documents in the XDS Affinity Domain. This query should be done by document class so as to catch both formats of document (Consent). The Document Consumer actor shall be capable of recognizing the eventCodeList from the resulting XDS Metadata. There is no required handling of Patient Privacy Consent Acknowledgement Document XDS Metadata.

There is no requirement for the Document Consumer to retrieve the Patient Privacy Consent Acknowledgement Document content.

3.18.5 Security Considerations

Relevant XDS Affinity Domain Security background is discussed in the Register Document transaction (see ITI TF-2a: 3.14.5.1).

3.18.5.1 Audit Record Considerations

The Registry Stored Query Transaction is a Query Information event as defined in Table 3.20.6-1. The Actors involved shall record audit events according to the following:

3.18.5.1.1 Document Consumer audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-18", "IHE Transactions", "Registry Stored Query")
Source (Document Consumer) (1)			
Human Requestor (0..n)			
Destination (Document Registry) (1)			
Audit Source (Document Consumer) (1)			
Patient (0..1)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	C	When WS-Addressing is used: <ReplyTo/>
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

	UserID	M	SOAP endpoint URI.

	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

4565

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (system object)
	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-18", "IHE Transactions", "Registry Stored Query")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	Stored Query ID (UUID)
	ParticipantObjectName	C	If known the value of <ihe:HomeCommunityId/>
	ParticipantObjectQuery	M	the AdhocQueryRequest, base64 encoded.
	ParticipantObjectDetail	*U*	*not specialized*

3.18.5.1.2 Document Registry audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-18", "IHE Transactions", "Registry Stored Query")
Source (Document Consumer) (1)			
Destination (Document Registry) (1)			
Audit Source (Document Registry) (1)			
Patient (0..1)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	C	When WS-Addressing is used: <ReplyTo/>
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	M	SOAP endpoint URI.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

4570

Patient **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters **(AuditMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (system object)
	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-18", "IHE Transactions", "Registry Stored Query")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	Stored Query ID (UUID)
	ParticipantObjectName	C	If known the value of <ihe:HomeCommunityId/>
	ParticipantObjectQuery	M	the AdhocQueryRequest, base64 encoded.
	ParticipantObjectDetail	*U*	*not specialized*

3.19 Authenticate Node

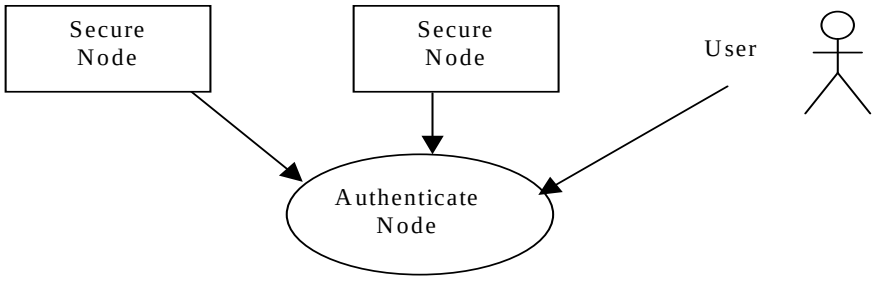
This section corresponds to Transaction 19 of the IHE ITI Technical Framework. Transaction 19 is used by the Secure Node actors

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

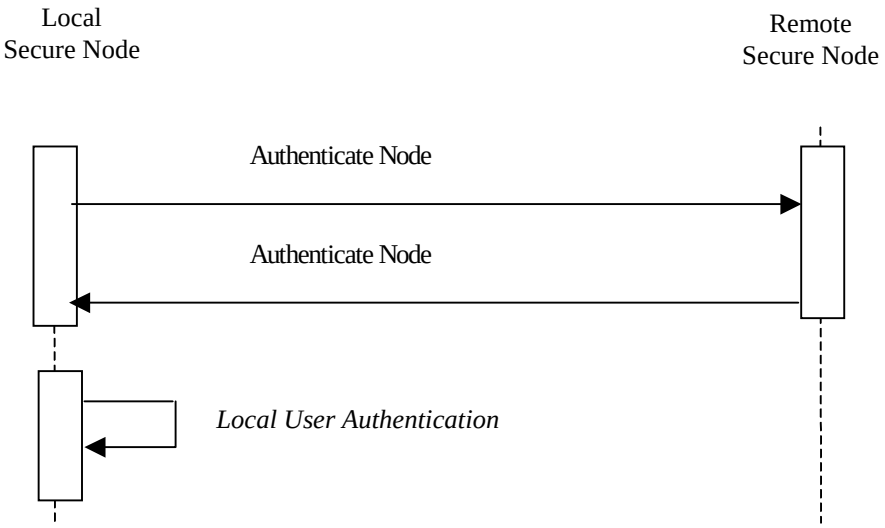
DICOM 2003 PS 3.15:
Security Profiles. Annex B1: The Basic TLS Secure Transport Connection profile.

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

The Authenticate node transaction involves the exchange of certificates representing the identities of the nodes. These identities are used to authenticate the nodes, to inform authorization, and audit logging.

3.19.6.1 Certificate Validation

The local organization (e.g., XDS Affinity Domain) will make the choice of what mixture of chain of trust and direct comparison is used to authenticate communications. This may be entirely based on chaining trust to selected CAs, entirely based upon provision of node certificates for direct comparison, or a mixture of both.

Note: The CAs used for ATNA chain of trust will be different than the default browser trusted list of CAs used for authenticating internet web servers. A worldwide CA, such as Verisign, is not generally trusted to determine which individual nodes within an organization should and should not communicate patient identifiable information.

When Authenticating the Remote Secure Node, the Local Secure Node:

- Shall be able to perform certificate validation based on signature by a trusted CA (See ITI TF-2a: 3.19.6.1.1) and
- Shall be able to perform direct certificate validation to a set of trusted certificates (See ITI TF-2a: 3.19.6.1.2)

It may reject communications when the certificate validation fails, or may restrict communications to only that which is appropriate for an unidentified other party.

3.19.6.1.1 Chain to a trusted certificate authority

The Secure Node or Secure Application:

- Shall provide the means for configuring which CAs are trusted to authenticate node certificates for use in a chain of trust. These CAs shall be identified by means of the public signing certificate for the signing CA.
- Shall support digital certificates encoded using both Deterministic Encoding Rules (DER) and Basic Encoding Rules (BER).
- Shall accept communications for which there is a certificate that is signed by a CA that is listed as a trusted signing authority.

3.19.6.1.2 Direct certificate validation

The Secure Node or Secure Application:

- Shall provide means for installing of the required certificates, for example, via removable media or network interchange (where the set of trusted certificates can be a mixture of CA signed certificates and self-signed certificates).
- Shall support digital certificates encoded using both Deterministic Encoding Rules (DER) and Basic Encoding Rules (BER).
- Shall accept communications for which there is a certificate configured as acceptable for direct certificate validation.

3.19.6.1.3 Other Certificate requirements

The Secure Node shall not require any specific certificate attribute contents, nor shall it reject certificates that contain unknown attributes or other parameters. Note that for node certificates the CN often is a hostname, attempting to use this hostname provides no additional security and will introduce a new failure mode (e.g., DNS failure).

The certificates used for mutual authentication shall be X509 certificates based on RSA key with key length in the range of 1024-4096, where the key length chosen is based on local site policy. Maximum expiration time acceptable for certificates should be defined in the applicable security policy. The IHE Technical Framework recommends a maximum expiration time of 2 years.

The method used to determine of the authenticated nodes, which ones are authorized to communicate is not specified. This may be a dual use of the set of trusted certificates, some attribute in the certificates, access control lists, or some other method. Using a certificate chain back to an external trusted certificate authority to determine authorizations is strongly discouraged.

3.19.6.2 DICOM and HL7 Connections

HL7 and DICOM transactions are required to adhere to the specifications in this section.

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.

All Secure Nodes shall be configurable for use on a physically secured network or not on a physically secured network. If Secure Node is configured for physical security, then it may use the non-TLS DICOM port and protocol.

3.19.6.3 HTTP Connections

3.19.6.3.1 Expected Actions

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.

HTTP communications shall require the encryption-option.

Note: IHE permits DICOM and HL7 connections to use encryption none because these connections are often internal to the organization, protected by other security mechanisms, and in the case of DICOM convey very large quantitites of data. The mandatory imposition of additional encryption on that traffic has significant performance impact. HTTP communications are much less voluminous, and many existing browser implementations do not support encryption none.

If Secure Node is configured for physical security, then it shall use the normal HTTP protocol.

3.19.6.4 Web-Services

3.19.6.4.1 Expected Actions

A trusted association shall be established between the two nodes utilizing WS-I Basic Security Profile Version 1.1. This association will be used for all secure transactions between the IHE actors in the two nodes. Note that IHE ITI TF-2a: 3.19.6.2 “HTTP Connections” and WS-I Basic Security Profile – section 3 “Transport Layer Mechanisms” (i.e. http://ws-i.org/profiles/basic-security/1.1/transport) are identical and interoperable.

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.

The audit log supports identification of the active participant. This is often defined as one key member of the team. Other means are used to track the entry and exit of various members of the team. IHE does not specify any specific team identification process.

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.

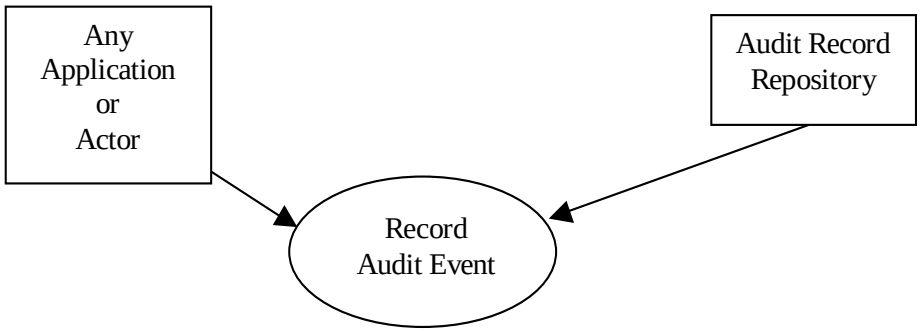
3.20 Record Audit Event

This section corresponds to Transaction 20 of the IHE IT Infrastructure 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.

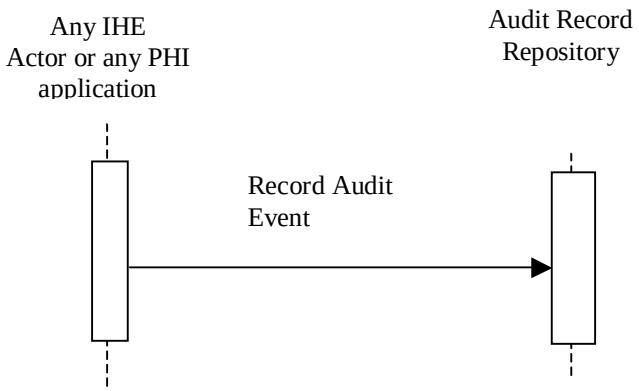
Actor: Audit Record Repository

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.
NIST:	SP 800-92 Guide to Computer Security Log Management.
W3C:	Recommendation: Extensible Markup Language (XML) 1.0

3.20.4 Interaction Diagram



4750

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

4755 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
4760 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 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 (ITI TF-2a: 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"

Trigger Event	Description	Source Vocabulary
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 (ITI TF-2a: 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 (ITI TF-2a: 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 (ITI TF-2a: 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 (ITI TF-2a: 3.20.7.3) "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) "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 3) PIX, PDQ, or XDS Query Notes: The general guideance is to log the query event with the query parameters and not the result of the query. The result of a query may be very large and is likely to be of limited value vs the overhead. The query parameters can be used effectively to detect bad behavor and the expectation is that given the query parameters the result could be	DICOM (Sup 95) "Query"

Trigger Event	Description	Source Vocabulary
	regenerated if necessary.	
Security Alert	Security Administrative actions create, modify, delete, query, and display the following: 1. 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. 2. Security attributes and auditable events for the application functions used for patient management, clinical processes, registry of business objects and methods **(e.g. WSDL, UDDI)**, 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 authentication revocation. 11. Emergency Mode Access (aka Break-Glass) Security administration events should always be audited.	DICOM (Sup 95) "Security Alert"
User Authentication	This message describes the event of a user attempting to log on or log off, whether successful or not. No Participant Objects are needed for this message.	DICOM (Sup 95) "User Authentication"
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"
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. (Future) Transport utilizing the Reliable Syslog protocol in "cooked" mode as defined in RFC-3195, subject to such changes as result from current IETF activity.4
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:

1. 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.
2. 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.

4 The Reliable Syslog implementation in the field has progressed very slowly. The IHE is waiting for the outcome of ongoing IETF activities. These may result in confirming the original IETF decision to promulgate Reliable Syslog "cooked", may result in modifications to that protocol, or may result in its replacement. IHE implementations may chose to proceed with this protocol to establish evidence for the IETF that it does perform as needed, but should be aware that it may be changed.

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.
- The MSG portion of the syslog packet shall only contain the audit XML message. The TAG portion shall not be included in the message.
- The MSG shall be in 'minimal length' UTF-8.
- The XML message contained in the MSG portion of the syslog packet shall not begin with a unicode BOM (byte order mark).

The PRI field shall be set using the facility value of 10 (security/authorization messages). Most messages should have the severity value of 5 (normal but significant), although applications may choose values of 4 (Warning condition) if that is appropriate to the more detailed information in the audit message. This means that for most audit messages the PRI field will contain the value "<85>". Audit repositories shall be prepared to deal appropriately with any incoming PRI value.

3.20.7 Audit Message Formats

3.20.7.1 RFC-3881 format

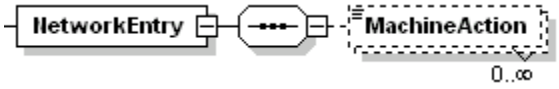
A common XML schema was defined based upon joint work by IHE, HL7, DICOM, ASTM E31, and the Joint NEMA/COCIR/JIRA Security and Privacy Committee. The IHE IT Infrastructure technical framework prefers use of this schema for audit records generated by all IHE actors. The schema can be found at: http://www.xml.org/xml/schema/7f0d86bd/healthcare-security-audit.xsd

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.

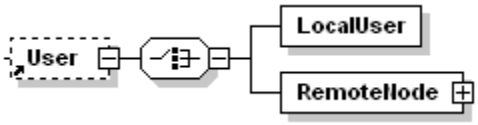
For reference, the schema elements are diagrammed below. The diagrams are read from left to right: elements to the right are part of the lefthandside element.

Required single element. A NetworkEntry element consists of exactly one MachineAction element.

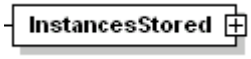
Optional single element. A NetworkEntry element consists of zero or one MachineAction element.



Optional multiple elements. A NetworkEntry element consists of zero or any number of MachineAction elements.

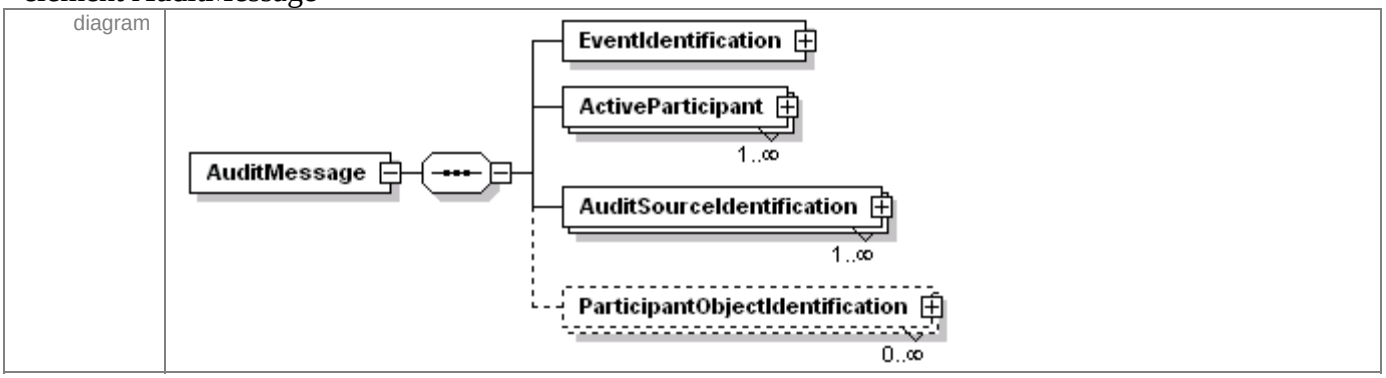


Selections of one out of several elements. A user consists either of a LocalUser element or of a RemoteNode element.

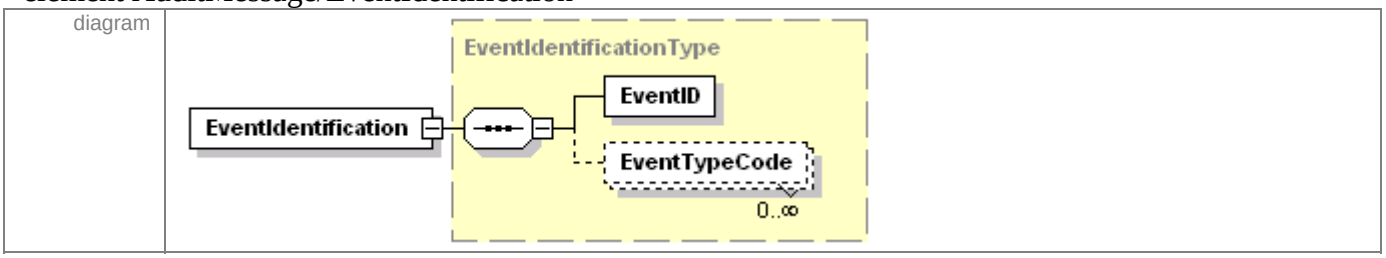


Compound element: The "+" in an element box means that the element consists of further elements. If these expansion elements have not occurred up to this point in the document, can be expected to follow below in the document.

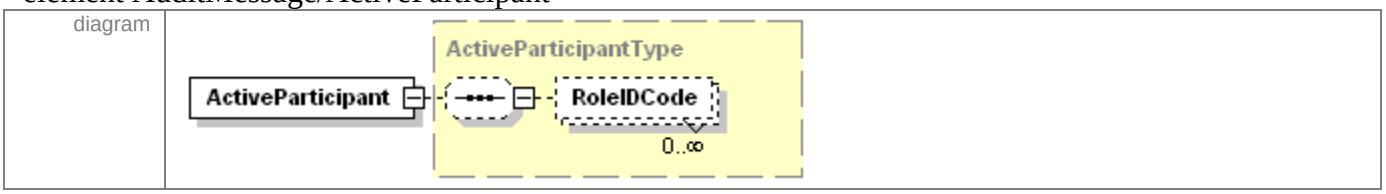
element AuditMessage



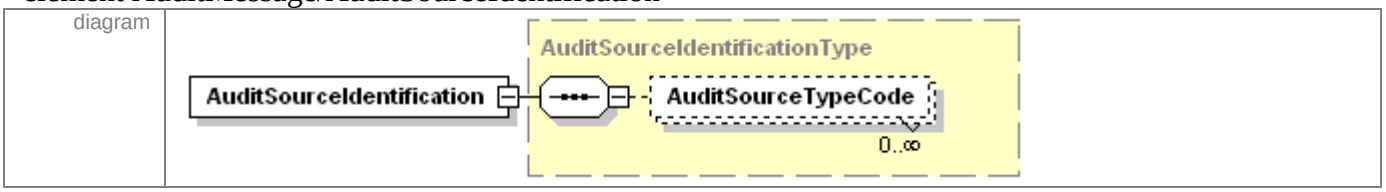
element AuditMessage/EventIdentification



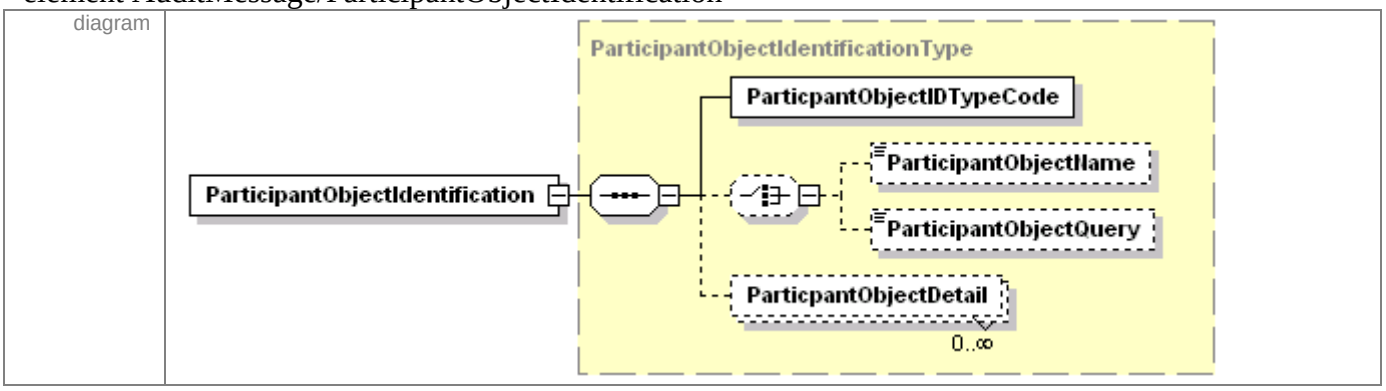
element AuditMessage/ActiveParticipant



element AuditMessage/AuditSourceIdentification



element AuditMessage/ParticipantObjectIdentification



Note: ParticipantObjectDetail should not include unnecessary detail such as duplication of the attributes otherwise encoded in the audit message.

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 and RFC-3881 do not address all the kinds of security and privacy events that can take place in the healthcare environment. The additional IHE defined events enumerated in ITI TF-2a: 3.20.7.5 shall be used for their defined purpose.

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. In cases where there is an event that applies to more than one patient, there shall be a separate audit message for each patient.

3.20.7.4 Other event reports

Events that do not correspond to DICOM events or IHE Extension events can be reported. They shall comply with RFC-3881. Neither ATNA profile, DICOM, nor RFC-3881 restrict private extensions to the RFC-3881 schema however any private extensions shall comply with the W3C XML encoding rules for the use of schemas, namespaces, etc.

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-2x: Appendix F.

3.20.7.7 RoleIDCode with access control roles

RoleIDCode is a CodedValueType. When describing a human users participation in an event, this value should represent the access control roles/permissions that authorized the event/trans. Use of standards based roles/permissions is preferable to site or application specific. As RFC-3881 indicates Many security systems are unable to produce this data, hence it is optional.

For example: at a site "St Fraser" they have defined a functional role code "NURSEA" for attending nurse. This can be represented as

EV("NURSEA", "St Fraser", "Attending Nurse")

Candidate standards based structual/functional role codes can be found at ISO, HL7, ASTM, and various other sources.

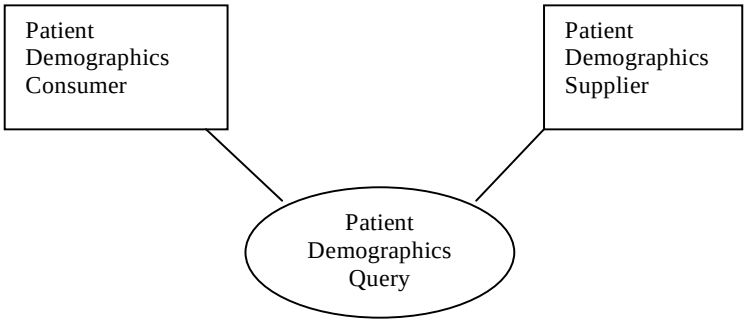
3.21 Patient Demographics Query

This section corresponds to Transaction ITI-21 of the IHE IT Infrastructure 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.

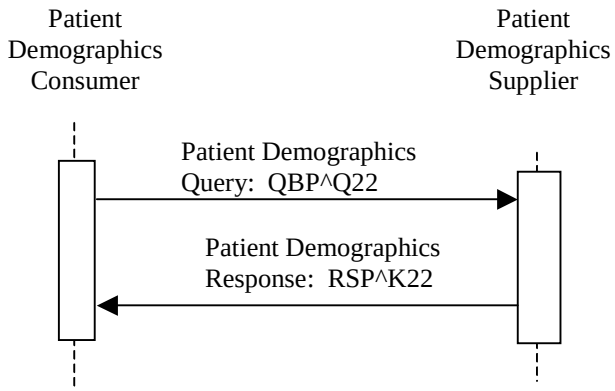
3.21.3 Referenced Standards

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
[DSC]	Continuation Pointer	2

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.

Each Patient Demographics Query request specifies two distinct concepts. 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***). A Patient Demographics Supplier may have knowledge of more than one source of demographics. A Patient Demographics Supplier shall support at least one source of patient demographics and may support multiple sources of demographics. ITI TF-2a: 3.21.4.1.2.1 describes how the the Patient Demographics Consumer specifies which source of

demographics are requested by the query. Each query response shall return demographics from a single patient information source.

The second concept present in the query is the set of patient identifier domains referenced by the query. These patient identifier domains may or may not be associated with the patient information source. A Patient Demographics Supplier shall support at least one patient identifier domain and may support multiple identifier domains. ITI TF-2a: 3.21.4.1.2.2 describes how the Patient Demographics Consumer requests identifiers from one or more patient identifier domains. Query responses may return patient identifiers from 0, 1 or multiple patient identifier domains.

3.21.4.1.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2x: C.1.2).

The Patient Demographics Supplier is able to obtain demographics from at least one and possibly multiple patient information sources. 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 RSP^K22 response. The value specified in MSH-5 is not related to the value requested 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 one and only one source of patient demographic information.

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 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

The Consumer shall specify "IHE PDQ Query" for QPD-1 Message Query Name.

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.

The Patient Demographics Consumer may specify, and the Patient Demographics Supplier shall support, the fields in the following table.

The Patient Demographics Supplier shall return demographic records that reflect the best fit to all of the search criteria.

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.5.1.1^SMITH~@PID.8^F
```


requests all patients whose family name (first subcomponent (data type ST) of the first component (data type FN) of PID-5-Patient Name (data type XPN)) matches the value 'SMITH' and whose sex (PID-8-Sex (data type IS)) matches the value 'female'.

3.21.4.1.2.2.2 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:

1. 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*.

   If QPD-8 is empty, the Patient Demographics Supplier shall return all Patient IDs known by the Patient Demographics Supplier for each patient that matches the search criteria. See Case 1 in ITI TF-2a: 3.21.4.2.2.8 for details on how this information is returned.

   If QPD-8 is specified and the domains are recognized, the Patient Demographics Supplier shall return the Patient IDs for each patient that matches the search criteria. See Case 2 in ITI TF-2a: 3.21.4.2.2.8 for details on how this information is returned.

   Any domain not recognized by the Patient Demographics Supplier is an error condition. See Case 3 in ITI TF-2a: 3.21.4.2.2.8 how to handle this condition.

2. In a single-domain environment, QPD-8 may be ignored by the Patient Demographics Supplier. The Supplier shall always return the identifier from the Patient ID Domain known by the Patient Demographics Supplier.

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-2x: Appendix M).

The Patient Demographics Consumer shall be able to support at least one of the following mechanisms for specifying QPD-8:

1. Transmit an empty value and receive all identifiers in all domains known by the Patient Demographics Supplier (one or more domains), or
2. Transmit a single value and receive zero or more identifiers in a single domain, or
3. Transmit multiple values and receive mutiple identifiers in those multiple domains.

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 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 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, always expressed as an integer greater than 0, 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" (ITI TF-2a: 3.21.4.1.3.3) and the "Expected Actions" section of the Patient Demographics Query Response message (ITI TF-2a: 3.21.4.2.3) for more information on the implementation of the continuation protocol.

3.21.4.1.2.4 DSC Segment

The Patient Demographics Consumer Actor may request additional increments of data by specifying this segment on the query request. This segment should be omitted on the initial query request. Its purpose is to request additional increments of the data from the Patient Demographic Supplier Actor.

Table 3.21-9. IHE Profile - DSC segment

SEQ	LEN	DT	OPT	TBL#	ITEM #	ELEMENT NAME
1	180	ST	O		00014	Continuation Pointer
2	1	ID	O	0398	01354	Continuation Style

3.21.4.1.2.4.1 Populating DSC-1 Continuation Pointer

To request additional increments of data, DSC-1 (Continuation Pointer) shall echo the value from RSP^K22 DSC-1.

3.21.4.1.2.4.2 Populating DSC-2 Continuation Style

DSC-2 (Continuation Style) shall always contain I, signifying that this is part of an interactive continuation message.

3.21.4.1.3 Expected Actions

3.21.4.1.3.1 Immediate Acknowledgement

The Patient Demographics Supplier shall immediately return an RSP^K22 response message as specified below in ITI TF-2a: 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-2x: 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 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 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 ITI TF-2a: 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 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2x: 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 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-2x: C.1.3) for the list of all required and optional fields within the MSA segment.

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-2b: 3.21.4.2.2.8).

QAK-1 (Query Tag) shall echo the same value of QPD-2 (Query Tag) of the QBP^Q22 message, to allow the Patient Demographics Query Consumer to match the response to the corresponding query request.

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 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 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-2a: 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 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 DSC Segment

If the 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 DSC of the QBP^Q22 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 sends a cancel query (QCN^J01) message (or when there are no more increments to return).

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 set of Patient ID Domains known by the Patient Demograpics Supplier.

If an incremental number of records are 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.

The consumer will specify the value of the continuation pointer in the DSC segment on the subsequent query request to request the next increment of responses.

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. If an identifier does not exist for a domain that was specified on QPD-8, nothing is returned in the list.

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.

The consumer will specify the value of the continuation pointer in the DSC segment on the subsequent query request to request the next increment of responses.

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

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.

If the Supplier has sent a DSC segment containing a continuation pointer value, additional increments of data are available upon request by the Consumer. After receiving each increment of data that includes a DSC segment containing a continuation pointer value, the Consumer should take one of the following actions.

- If the Consumer wishes to receive another increment of the data, the Consumer reissues the query message using a new unique value in *MSH-10-message control ID* and adding the DSC segment after the RCP segment. DSC-1 shall echo the continuation pointer returned in RSP^K22 DSC-1 segment.
- If the Consumer does not wish to receive another increment of the data, the Consumer issues a cancel query (QCN^J01) message. The consumer shall echo the query tag from QAK-1 in QID-1 and the query message name from QPD-1 in QID-2.
- If the Consumer does not reissue the query or send a cancel query message, the query will eventually terminate.

If the Supplier has not sent a DSC segment containing a continuation pointer value, no more increments of data are available and no further action by the Consumer is required.

3.21.4.3 Canceling a query

The Patient Demographic Consumer can send a cancel trigger to notify the Patient Demographic Supplier that no more incremental responses will be requested, and the interactive query can be terminated. This cancellation trigger is optional. How long the Patient Demographic Supplier retains query results (for incremental response) is an implementation decision and therefore beyond the scope of IHE.

3.21.4.3.1 Trigger Events

The Patient Demographic Consumer which received a RSP^K22 response message indicating there are more incremental responses data available, can terminate the interactive query with the following HL7 trigger event:
J01 – Cancel query status

3.21.4.3.2 Message Semantics

Canceling a query is conducted by the QCN^J01 message. The Patient Demographic Consumer can generate this message to notify the Patient Demographic Supplier that no more data is desired. The segments of the message listed below are required, and their details descriptions are provided in the following subsections.

Table 3.21.10 QCN Cancel query

QCN	Cancel query	Chapter in HL7 2.5
MSH	Message Header	2
QID	Query identification Segment	5

The receiver shall acknowledge this cancel by the HL7 ACK message. See ITI TF-2x: C.1.3, "Acknowledgement Modes", for definition and discussion of the ACK message.

3.21.4.3.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2x: C.1.2).

MSH-9 (Message Type) shall have three components. The first component shall have the value of QCN; the second component shall have a value of J01. The third component shall have the value of QCN_J01.

3.21.4.3.2.2 QID Segment

The QID segment contains the information necessary to uniquely identify the query being cancelled.

Table 3.21-9. IHE Profile - QID segment

SEQ	LEN	DT	OPT	TBL #	ITEM #	ELEMENT NAME
1	32	ST	R		00696	Query Tag
2	250	CE	R	0471	01375	Message Query Name

3.21.4.3.2.2.1 Populating QID-1 Query Tag

QID-1 (Query Tag) uniquely identifies the query to be canceled. This field shall contain the same value specified in QPD-2.

3.21.4.3.2.2.2 Populating QID-2 Message Query Name

QID-2 (Message Query Name) identifies the name of the query. It is an identifier of the conformance statement for this query. This field shall contain the same value specified in QPD-1.

3.21.5 Security Considerations

3.21.5.1 Audit Record Considerations

The Patient Demographics Query Transaction is a Query Information event as defined in Table 3.20.6-1. The Actors involved shall record audit events according to the following:

3.21.5.1.1 Patient Demographics Consumer audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-21", "IHE Transactions", "Patient Demographics Query")
Source (Patient Demograhics Consumer) (1)			
Human Requestor (0..n)			
Destination (Patient Demographics Supplier) (1)			
Audit Source (Patient Demographics Consumer) (1)			
Patient (0..n)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Consumer Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Source facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
	ParticipantObjectTypeCode	M	"2" (system object)
	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-21", "IHE Transactions", "Patient Demographics Query")

	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	*U*	*not specialized*
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	M	the QPD segment of the query - Base64 encoded
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

5330

3.21.5.1.2 Patient Demographics Source audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-21", "IHE Transactions", "Patient Demographics Query")
Source (Patient Demographics Consumer) (1)			
Destination (Patient Demographics Supplier) (1)			
Audit Source (Patient Demographics Supplier) (1)			
Patient (0..n)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Consumer Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Supplier facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

5335

Patient (AudittMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters (AudittMessage/ ParticipantObjectIdentification)	ParticipantObjectTypeCode	M	"2" (system object)
	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-21", "IHE Transactions", "Patient Demographics Query")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	*U*	*not specialized*
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	M	the QPD segment of the query - Base64 encoded
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

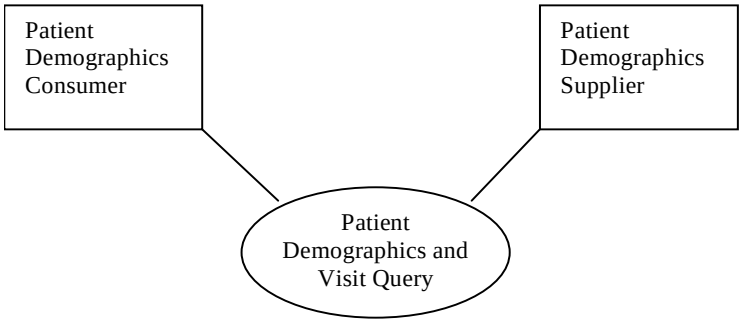
3.22 Patient Demographics and Visit Query

This section corresponds to Transaction ITI-22 of the IHE IT Infrastructure 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.

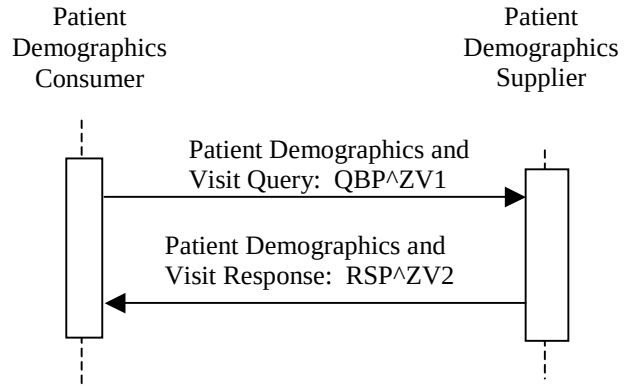
3.22.3 Referenced Standards

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
[DSC]	Continuation Pointer	2

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.

Each Patient Demographics and Visit Query request specifies two distinct concepts. 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***). A Patient Demographics Supplier

may have knowledge of more than one source of demographics. A Patient Demographics Supplier shall support at least one source of patient demographics and may support multiple sources of demographics. ITI TF-2a: 3.21.4.1.2.1 describes how the the Patient Demographics Consumer specifies which source of demographics are requested by the query. Each query response shall return demographics from a single patient information source.

The second concept present in the query is the set of patient identifier domains referenced by the query. These patient identifier domains may or may not be associated with the patient information source. A Patient Demographics Supplier shall support at least one patient identifier domain and may support multiple identifier domains. ITI TF-2a: 3.21.4.1.2.2 describes how the Patient Demographics Consumer requests identifiers from one or more patient identifier domains. Query responses may return patient identifiers from 0, 1 or multiple patient identifier domains.

3.22.4.1.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2x: C.1.2).

The Patient Demographics Supplier is able to obtain demographics from at least one and possibly multiple patient information sources. 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 RSP^ZV2 response. The value specified in MSH-5 is not related to the value requested 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 one and only one source of patient demographic information.

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

The Consumer shall specify "IHE PDVQ Query" for QPD-1 Message Query Name.

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> shall 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> 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.

The Patient Demographics Consumer may specify, and the Patient Demographics Supplier shall support, the fields in the following table.

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, the Patient Demographics Supplier should support the fields in the following table, and it shall support at least one of them. 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

The Patient Demographics Supplier shall return demographic records that reflect the best fit to all of the search criteria.

Examples of parameter expressions in QPD-3:

```
@PID.5.1.1^SMITH~@PID.8^F
```

requests all patients whose family name (first subcomponent (data type ST) of the first component (data type FN) of PID-5-Patient Name (data type XPN)) matches the value ‘SMITH’ and whose sex (PID-8-Sex (data type IS)) matches the value ‘female’.

```
@PV1.3.2^389~@PV1.3.3^2
```

requests all patients whose room number (second component (data type IS) of PV1-3-Assigned Patient Location (data type PL)) matches the value 389 and whose bed number (third component (data type IS) of PV1-3-Assigned Patient Location (data type PL)) 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:

1. 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*.

   If QPD-8 is empty, the Patient Demographics Supplier shall return all Patient IDs known by the Patient Demographics Supplier for each patient that matches the search criteria. See Case 1 in ITI TF-2a: 3.21.4.2.2.8 for details on how this information is returned.

If QPD-8 is specified and the domains are recognized, the Patient Demographics Supplier shall return the Patient IDs for each patient that matches the search criteria. See Case 2 in ITI TF-2a: 3.21.4.2.2.8 for details on how this information is returned.

Any domain not recognized by the Patient Demographics Supplier is an error condition. See Case 3 in ITI TF-2a: 3.21.4.2.2.8 how to handle this condition.

2. In a single-domain environment, QPD-8 may be ignored by the Patient Demographics Supplier. The Supplier shall always return the identifier from the Patient ID Domain known by the Patient Demographics Supplier.

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-2x: Appendix M).

The Patient Demographics Consumer shall be able to support at least one of the following mechanisms for specifying QPD-8:

1. Transmit an empty value and receive all identifiers in all domains known by the Patient Demographics Supplier (one or more domains), or

2. Transmit a single value and receive zero or more identifiers in a single domain, or

3. Transmit multiple values and receive mutiple identifiers in those multiple domains.

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, always expressed as an integer greater than 0, 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-2a: 3.22.4.1.3.3) and the "Expected Actions" section of the Patient Demographics Query Response message (ITI TF-2a: 3.22.4.2.3) for more information on the implementation of the continuation protocol.

3.22.4.1.2.4 DSC Segment

The Patient Demographics Consumer Actor may request additional increments of data by specifying this segment on the query request. This segment should be omitted on the initial query request. Its purpose is to request additional increments of the data from the Patient Demographic Supplier Actor.

Table 3.22-9. IHE Profile - DSC segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	180	ST	O		00014	Continuation Pointer
2	1	ID	O	0398	01354	Continuation Style

3.22.4.1.2.4.1 Populating DSC-1 Continuation Pointer

To request additional increments of data, DSC-1 (Continuation Pointer) shall echo the value from RSP^K22 DSC-1.

3.22.4.1.2.4.2 Populating DSC-2 Continuation Style

DSC-2 (Continuation Style) shall always contain "I", signifying that this is part of an interactive continuation message.

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 ITI TF-2a: 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-2x: 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 ITI TF-2a: 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-2x: 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-2x: 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-2a: 3.22.4.2.2.11).

QAK-1 (Query Tag) shall echo the same value of QPD-2 (Query Tag) of the QBP^Q22 message, to allow the Patient Demographics Query Consumer to match the response to the corresponding query request.

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-2a: 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 DSC segment of the QBP^ZV1 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 sends a cancel query (QCN^J01) message (or when there are no more increments to return). The Supplier shall signal no more increments by omitting the DSC segment.

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 set of Patient ID Domains known by the Patient Demograpics Supplier.

If an incremental number of records are 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.

The consumer will specify the value of the continuation pointer in the DSC segment on the subsequent query request to request the next increment of responses.

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. If an identifier does not exist for a domain that was specified on QPD-8, nothing is returned in the list.

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.

The consumer will specify the value of the continuation pointer in the DSC segment on the subsequent query request to request the next increment of responses.

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*.

3.22.4.2.3 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.

If the Supplier has sent a DSC segment containing a continuation pointer value, additional increments of data are available upon request by the Consumer. After receiving each increment of data that includes a DSC segment containing a continuation pointer value, the Consumer should take one of the following actions.

- If the Consumer wishes to receive another increment of the data, the Consumer reissues the query message using a new unique value in *MSH-10-message control ID* and adding the DSC segment after the RCP segment. DSC-1 shall echo the continuation pointer returned in RSP^K22 DSC-1 segment.

- If the Consumer does not wish to receive another increment of the data, the Consumer issues a cancel query (QCN^J01) message.
- If the Consumer does not reissue the query or send a cancel query message, the query will eventually terminate.

If the Supplier has not sent a DSC segment containing a continuation pointer value, no more increments of data are available and no further action by the Consumer is required.

3.22.4.3 Canceling a query

The Patient Demographic Consumer can send a cancel trigger to notify the Patient Demographic Supplier that no more incremental response will be requested, and interactive query can be terminated. This cancellation trigger is optional. How long the Patient Demographic Supplier retains query results (for incremental response) is an implementation decision and therefore beyond the scope of IHE.

3.22.4.3.1 Trigger Events

The Patient Demographic Consumer which received a RSP^K22 response message indicating there more incremental response data available, can terminate the interactive query with the following HL7 trigger event:

J01 – Cancel query status

3.22.4.3.2 Message Semantics

Canceling a query is conducted by the QCN^J01 message. The Patient Demographic Consumer can generate this message to notify the Patient Demographic Supplier that no more data is desired. The segments of the message listed below are required, and their details descriptions are provided in the following subsections.

Table 3.22.10 QCN Cancel query

QCN	Cancel query	Chapter in HL7 2.5
MSH	Message Header	2
QID	Query identification Segment	5

The receiver shall acknowledge this cancel by the HL7 ACK message. See ITI TF-2x: C.1.3, "Acknowledgement Modes", for definition and discussion of the ACK message.

3.22.4.3.2.1 MSH Segment

The MSH segment shall be constructed as defined in the "Message Control" section (ITI TF-2x: C.1.2).

MSH-9 (Message Type) shall have three components. The first component shall have the value of QCN; the second component shall have a value of J01. The third component shall have the value of QCN_J01.

3.22.4.3.2.2 QID Segment

The QID segment contains the information necessary to uniquely identify the query being cancelled.

Table 3.22-9. IHE Profile - QID segment

SEQ	LEN	DT	OPT	TBL#	ITEM#	ELEMENT NAME
1	32	ST	R		00696	Query Tag

2	250	CE	R	0471	01375	Message Query Name

3.22.4.3.2.2.1 Populating QID-1 Query Tag

QID-1 (Query Tag) uniquely identifies the query to be canceled. This field shall contain the same value specified in QPD-2.

3.22.4.3.2.2.2 Populating QID-2 Message Query Name

QID-2 (Message Query Name) identifies the name of the query. It is an identifier of the conformance statement for this query. This field shall contain the same value specified in QPD-1.

3.22.5 Security Considerations

3.22.5.1 Audit Record Considerations

The Patient Demographics Query Transaction is a Query Information event as defined in Table 3.20.6-1. The Actors involved shall record audit events according to the following:

3.22.5.1.1 Patient Demographics Consumer audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-22", "IHE Transactions", "Patient Demographics and Visit Query")
Source (Patient Demograhics Consumer) (1)			
Human Requestor (0..n)			
Destination (Patient Demographics Supplier) (1)			
Audit Source (Patient Demographics Consumer) (1)			
Patient (0..n)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Consumer Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.
Human Requestor (if known) **AuditMessage/ ActiveParticipant**	UserID	M	Identity of the human that initiated the transaction.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	U	Access Control role(s) the user holds that allows this transaction.
	NetworkAccessPointTypeCode	*NA*	
	NetworkAccessPointID	*NA*	

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Source facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

5770

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
	ParticipantObjectTypeCode	M	"2" (system object)

	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-22", "IHE Transactions", "Patient Demographics and Visit Query")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	*U*	*not specialized*
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	M	the QPD segment of the query - Base64 encoded
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

3.22.5.1.2 Patient Demographics Source audit message:

	Field Name	Opt	Value Constraints
Event **AuditMessage/ EventIdentification**	EventID	M	EV(110112, DCM, "Query")
	EventActionCode	M	"E" (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*
	EventTypeCode	M	EV("ITI-22", "IHE Transactions", "Patient Demographics and Visit Query")
Source (Patient Demographics Consumer) (1)			
Destination (Patient Demographics Supplier) (1)			
Audit Source (Patient Demographics Supplier) (1)			
Patient (0..n)			
Query Parameters(1)			

Where:

Source **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Consumer Actor facility and sending application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	*U*	*not specialized*
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"true"
	RoleIDCode	M	EV(110153, DCM, "Source")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

Destination **AuditMessage/ ActiveParticipant**	UserID	M	The identity of the Patient Demographics Supplier facility and receiving application from the HL7 message; concatenated together, separated by the | character.
	AlternativeUserID	M	the process ID as used within the local operating system in the local system logs.
	UserName	*U*	*not specialized*
	UserIsRequestor	M	"false"
	RoleIDCode	M	EV(110152, DCM, "Destination")
	NetworkAccessPointTypeCode	M	"1" for machine (DNS) name, "2" for IP address
	NetworkAccessPointID	M	The machine name or IP address, as specified in RFC 3881.

5775

Audit Source **AuditMessage/ AuditSourceIdentification**	*AuditSourceID*	*U*	*Not specialized.*
	AuditEnterpriseSiteID	*U*	*not specialized*
	AuditSourceTypeCode	*U*	*not specialized*

Patient **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"1" (Person)
	ParticipantObjectTypeCodeRole	M	"1" (Patient)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV(2, RFC-3881, "Patient Number")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The patient ID in HL7 CX format.
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	*U*	*not specialized*
	ParticipantObjectDetail	*U*	*not specialized*
Query Parameters **(AudittMessage/ ParticipantObjectIdentification)**	ParticipantObjectTypeCode	M	"2" (system object)
	ParticipantObjectTypeCodeRole	M	"24" (query)
	ParticipantObjectDataLifeCycle	*U*	*not specialized*
	ParticipantObjectIDTypeCode	M	EV("ITI-22", "IHE Transactions", "Patient Demographics and Visit Query")
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	*U*	*not specialized*
	ParticipantObjectName	*U*	*not specialized*
	ParticipantObjectQuery	M	the QPD segment of the query - Base64 encoded
	ParticipantObjectDetail	M	Type=MSH-10 (the literal string), Value=the value of MSH-10 (from the message content, base64 encoded)

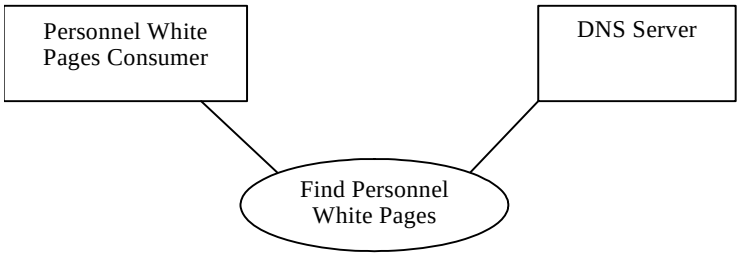
3.23 Find Personnel White Pages

This section corresponds to Transaction ITI-23 of the IHE IT InfrastructureTechnical Framework.
5780 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



5785 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

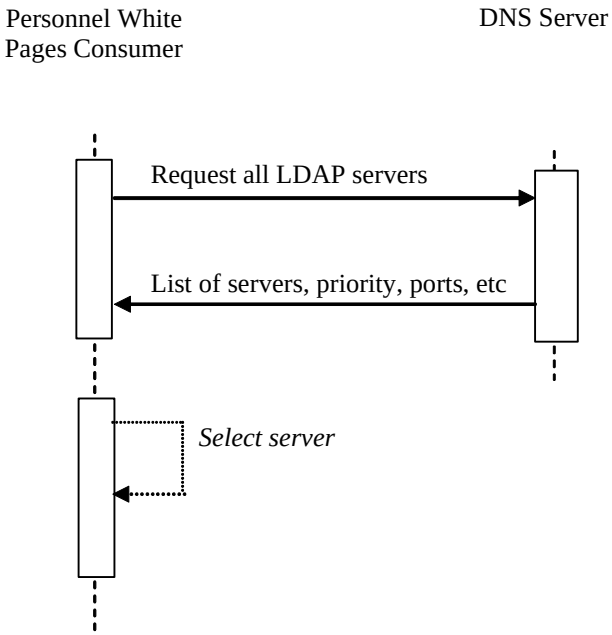
RFC-2219 Use of DNS Aliases for Network Services

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-2a: 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

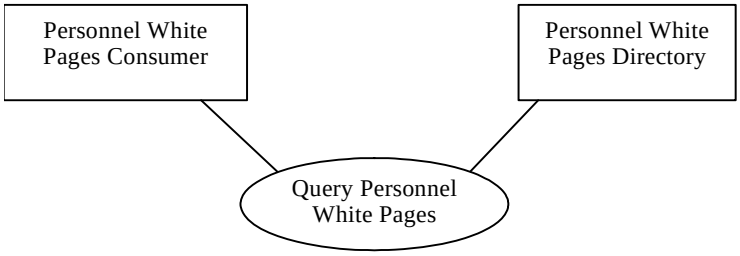
This section corresponds to Transaction ITI-24 of the IHE IT Infrastructure Technical Framework. Transaction ITI-24 is used by the Personnel White Pages Consumer and the Personnel White Pages Directory Actors.

3.24.1 Scope

This Transaction is used to retrieve information from the Personnel White Pages directory.

The RFC-3377 "Lightweight Directory Access Protocol (v3) : Technical Specification" 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

RFC 3377 - Lightweight Directory Access Protocol (v3): Technical Specification

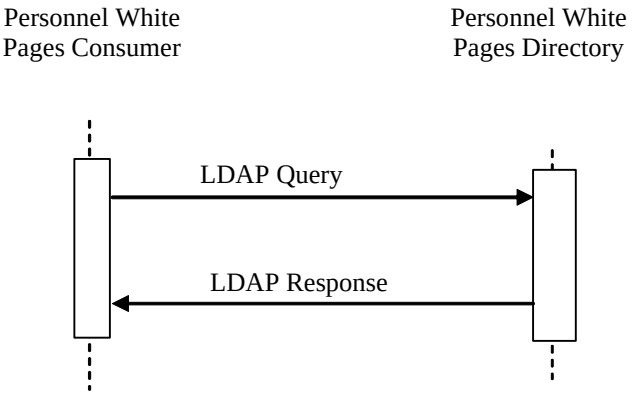
ISO: ISO/TS 17090 directory standard for healthcare identity management

CRU: Projet de schémas d'annuaires et de schémas de registres de resources numériques interopérables pour les administrations Document technique – v1, novembre 2002

ITU-T: 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
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 ITI TF-2a: 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 ITI TF-2a: 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.
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 ITI TF-2a: 3.24.4.1.2.3.3 Phone Numbers

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
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 ITI TF-2a: 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.
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	

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
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	
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	

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
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 ITI TF-2a: 3.24.4.1.2.3.3 Phone Numbers
teletexTerminalIdentifier	RFC 2798	• Teletex Terminal Identifier • Optional	D	
telexNumber	RFC 2798	• Telex Number • Optional	D	
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 • Optional • The user ID use for system login. • (e.g. bjensen)	O	See ITI TF-2a: 3.24.5.2.3.2 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.

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
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.
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	

Attribute Name	Source	• Definition • Standard defined Optionality • Description	IHE REQ	IHE Comment
X500uniqueIdentifier	RFC 2798	• Unique identifier • Optional • 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.	O	

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.

3.25 Intentionally Left Blank

3.26 Intentionally Left Blank

3.27 Intentionally Left Blank

3.28 Intentionally Left Blank

