ACC, HIMSS and RSNA

Integrating the Healthcare Enterprise

IHE IT Infrastructure Technical Framework

Supplement 2007-2008

Registry Stored Query Transaction for XDS Profile

[ITI-18]

Trial Implementation Version

Publication Date: August 20, 2007

Contents

1 Foreword

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 not to define new integration standards, but rather to support the use of existing standards, HL7, DICOM, IETF, and others, as appropriate in their respective domains in an integrated manner, defining configuration choices when necessary. IHE maintain formal relationships with several standards bodies including HL7, DICOM and refers recommendations to them when clarifications or extensions to existing standards are necessary.

This initiative has numerous sponsors and supporting organizations in different medical specialty domains and geographical regions. In North America the primary sponsors are the American College of Cardiology (ACC), the Healthcare Information and Management Systems Society (HIMSS) and the Radiological Society of North America (RSNA). IHE Canada has also been formed. IHE Europe (IHE-EUR) is supported by a large coalition of organizations including the European Association of Radiology (EAR) and European Congress of Radiologists (ECR), the Coordination Committee of the Radiological and Electromedical Industries (COCIR), Deutsche Röntgengesellschaft (DRG), the EuroPACS Association, Groupement pour la Modernisation du Système d'Information Hospitalier (GMSIH), Société Francaise de Radiologie (SFR), Società Italiana di Radiologia Medica (SIRM), the European Institute for health Records (EuroRec), and the European Society of Cardiology (ESC). 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.

The IHE Technical Frameworks for the various domains (IT Infrastructure, Cardiology, Laboratory, Radiology, etc.) 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 for these Technical Frameworks may be found at www.ihe.net/Technical_Framework.

The IHE Technical Framework identifies a subset of the functional components of the healthcare enterprise, called IHE Actors, and specifies their interactions in terms of a set of coordinated, standards-based transactions. It describes this body of transactions in progressively greater depth. The volume I 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 clinical needs. The subsequent volumes provide detailed technical descriptions of each IHE transaction.

This IHE IT Infrastructure Technical Framework Supplement is issued for Trial Implementation through March 2008.

Comments and change proposals arising from Trial Implementation may be submitted to http://forums.rsna.org under the forum:

"Integrating the Healthcare Enterprise"

Select the sub-forum:

"IHE IT Infrastructure 2007 Supplement for Trial Implementation"

The IHE IT Infrastructure Technical Committee will address these comments resulting from implementation, connect-a-thon testing, and demonstrations such as HIMSS 2008. Final text is expected to be published in June 2008.

2 Introduction

This supplement adds a single transaction, Stored Query, to the XDS Profile. Stored Query is a large improvement over the existing Query Registry transaction since it removes the use of SQL. Passing SQL in the Query Registry transaction opens a registry to denial of service attacks and constrains the implemention of the Document Registry actor since the structure of the underlying database is reflected in the structure of the SQL queries.

This Stored Query Supplement introduces an optimized query that simplifies implementation both on the XDS Document Registry and XDS Document Consumer. As this Stored Query is functionally identical to the required subset of the existing Query Registry transaction [ITI-16], it has been decided to make this new Stored Query mandatory and the existing query optional both on the XDS Document Registry and

Document Consumer. The simplification and optimization benefits realized outweigh the few incompatible changes introduced by this approach:

- A few XML schema change introduced by the move from ebXML Registry 2.x to 3.0 (e.g. name space)
- Replacement of the SQL Query expression by a compact set of query attributes

The Provide and Register Document Set and Register transactions could have been upgraded to version 3.0 at the same time. They were not upgraded for the following reasons:

- Provide and Registry Document Set relies on Soap with Attachments. We anticipate upgrading this transaction to use MTOM once WS-I finishes its profiling work. When that work is done, we anticipate upgrading Provide and Register with both changes at one time.
- The Register transaction should be upgraded at the same time as Provide and Register.

2.1 Open Issues and Questions

1. Should the existing Query Registry transaction [ITI-16] be labeled as Deprecated and eventually be removed from the profile?
2. Given closed issues #8, additional review of queries selected for the Stored Query transaction and the query catalog in the Query Registry transaction is necessary.

2.2 Closed Issues

1. Use v2 UUIDs or v3 identifiers for labeling queries? In v2 identifiers must be UUIDs (urn:uuid:…). In v3 it is possible to use other urns. MUST USE UUIDS for V2 compatibility.
2. Add registry v3 encoding section
3. Complete volume 1 material
4. Reconcile query documentation between this document and existing ITI-16
5. Need to add instructions on how to implement GetAll etc., queries that take multiple simple queries in 2.1
6. Optionality of this transaction (Registry Stored Query) and the original Registry Query transaction needs discussion. Early implementers of XDS need the original Registry Query transaction to be required and this transaction to be optional. Future implementers will need the opposite. The reason is that this query transaction removes SQL from the interface of XDS and therefore removes constraints from the implementation of the registry engine itself.

7. Discuss Federate bit? – No, its use is not covered by this supplement.
8. The community needs to review the Query Catalog documented in the Query Registry transaction [ITI -16] to ensure that all queries needed by the community are included. Errors in the SQL can be safely ignored since they are being repaired via Change Proposals from this last testing season. These changes will not appear until after this review is complete. Proposals for new queries should be sent to IT Infrastructure technical committee as part of this Public Comment. If they are published in XDS this season they will be part of the established Query Catalog and all registry implementations will be required to support them. If they are documented later, they will be labeled as optional by normal IHE rules. – Query catalog has been updated from review.

Volume I – Integration Profiles

Add edits below to table 10.1-1.

Table 10.1-1 XDS - Actors and Transactions

Actors	Transactions	Optionality	Section in Vol. 2
Document Consumer	Query Registry	~~R~~**O**	ITI TF-2:3.16
	Retrieve Document	R	ITI TF-2:3.17
	Registry Stored Query	**R (Note 4)**	
Document Source	Provide and Register Document Set	R (Note 1)	ITI TF-2:3.15
	~~Off-Line Transaction mode~~	~~O~~	~~ITI TF-1:10.4.7.1~~
	~~Multiple Documents Submission~~	~~O~~	~~ITI TF-2:3.15.5~~
	~~Document Life Cycle Management~~	~~O~~	~~ITI TF-2:3.15.5~~
	~~Folder Management~~	~~O~~	~~ITI TF-2:3.15.5~~
Document Repository	Provide and Register Document Set	R (Note 1)	ITI TF-2:3.15
	Register Document Set	R (Note 2)	ITI TF-2:3.14
	Retrieve Document	R	ITI TF-2:3.17
	~~Off-Line Transaction mode~~	~~O~~	~~ITI TF-1:10.4.7.1~~
Document Registry	Register Document Set	R (Note 2)	ITI TF-2:3.14
	Query Registry	~~R~~**O**	ITI TF-2:3.16
	Patient Identity Feed	R	ITI TF-2:3.8
	Registry Stored Query	**R (Note 4)**	
Patient Identity Source	Patient Identity Feed	R (Note 3)	ITI TF-2:3.8

Note 4: The Document Registry actor part of the Registry Stored Query transaction shall implement all queries defined by the Registry Stored Query transaction. No such minimum requirements are placed on the Document Consumer actor.

Introduce this section as 10.1.2.4 which requires the renumbering of several following sections.

10.1.2.4 Registry Stored Query

The Registry Stored Query transaction is issued by the Document Consumer Actor on behalf of a care provider (EHR-CR) to a Document Registry. The Document Registry Actor searches the registry to locate documents that meet the provider's specified query criteria. It will return registry metadata containing a list of document entries found to meet the specified criteria including the locations and identifier of each corresponding document in one or more Document Repositories.

This transaction differs from Query Registry [ITI-16] by storing the query in the Document Registry actor and referencing the query in the transaction instead of passing SQL.

With the Query Transaction [ITI-16], SQL language queries are transmitted to the Registry actor and results are returned. In a Stored Query, the definition of the query is stored on the Registry actor. To invoke the query, an identifier associated with the query is transmitted along with parameters defined by the query. This has the following benefits:

1. Malicious SQL transactions cannot be introduced
2. Alternate database styles and schemas can be used to implement the Document Registry actor. This is because the style of SQL query statements are directly related to the table layout in a relational database.

This profile does not define how Stored Queries are loaded into or implemented in the Document Registry actor.

2.3 1.7 History of Annual Changes

Does this remain in the Supplement or get put into volume 1?

Add the following bullet to the end of the bullet list in section 1.7

- Added the Stored Query transaction

The following diagram is updated.

2.4 Actors/Transactions

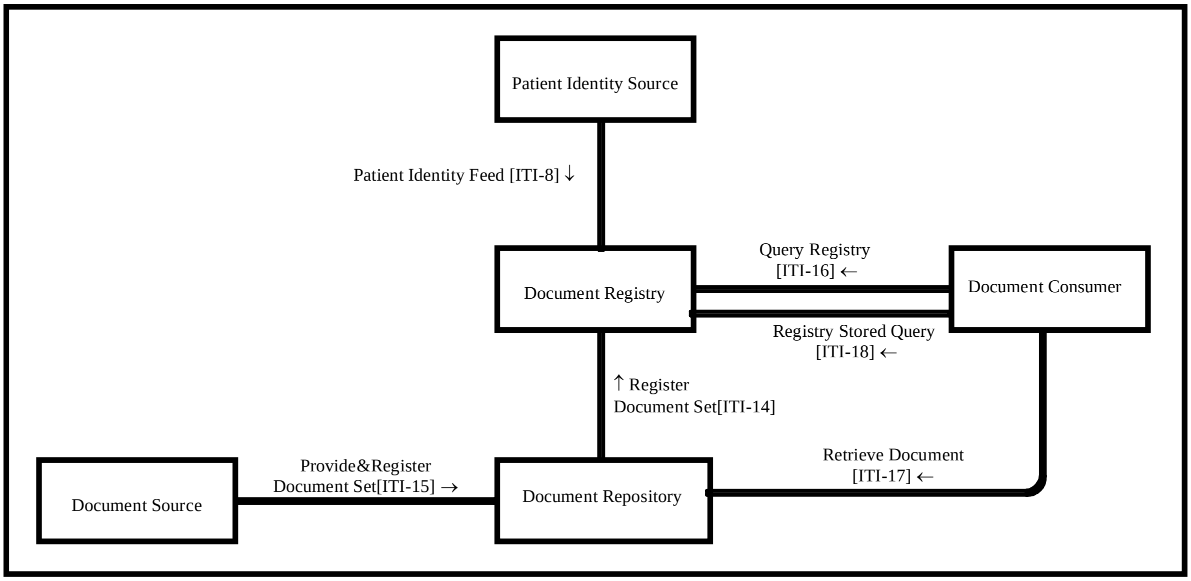


Figure 10.1-1 Cross-Enterprise Document Sharing Diagram

Volume 2 - Transactions

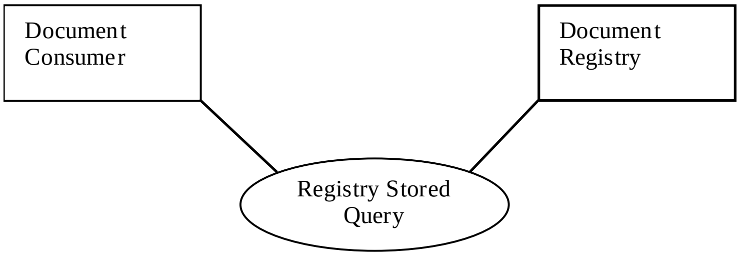
3.18 Registry Stored Query

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

3.18.1 Scope

The scope of the Registry Stored Query transaction is the same as the Query Registry transaction [ITI-16]. The same queries are required to be supported and the options controlling what kind of data is returned are the same.

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

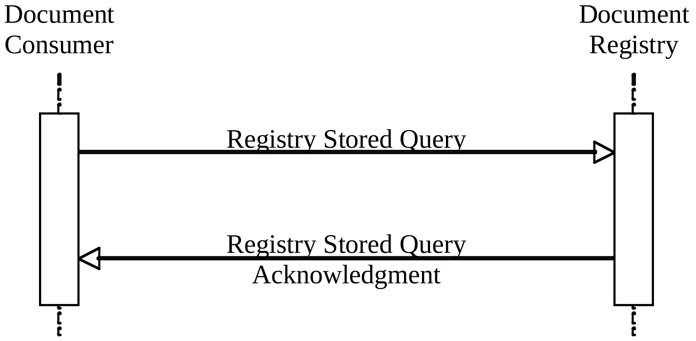
Role: Services the query using its stored definitions of the queries defined for XDS.

3.18.3 Referenced Standards

ebRIM	OASIS/ebXML Registry Information Model v3.0
ebRS	OASIS/ebXML Registry Services Specifications v3.0
SOAP	SOAP v1.1
SOAP12	SOAP 1.2 Recommendation http://www.w3.org/TR/soap/

Note: The ebRIM and ebRS standards referenced here are version 3.0. Most of XDS references 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.

The Cross Community Access (XCA) profile updates the Message Semantics of this transaction to include use of the ebRIM “home” attribute. This attribute is used to hold the homeCommunityId on some queries. Refer to the XCA profile for more details.

3.18.4.1.2.1 Version 3.0 ebXML Registry Standard

This transaction is taken from ebXML Registry version 3.0 while the rest of XDS comes from version 2.1. 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 in 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.

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 section 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 (3.18.4.1.2.4) below
- Query Parameters – as defined in the Query Parameters section (3.18.4.1.2.3.7) below

3.18.4.1.2.3.1 Parameter returnType

This parameter is defined in section 3.16.4.1.2.1 Parameter returnType.

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 section 3.18.4.1.2.4) to the standard XDS queries defined in the Query Registry transaction [ITI-16].

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 Optional/Multivalued Parameters

In the following table, optional (Opt) refers to a parameter being optional in the query. Multivalued (Mult) indicates that multiple values are acceptable in the query.

3.18.4.1.2.3.5 Coding of Single/Multiple Values

Single values are coded as *(need reference to SQL standard or include all sql encoding rules for this datatype)*

- 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 sepecified.

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

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

See section 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.

Note: Transactions ITI-16 and ITI-18 must be maintained together to keep the query catalog and the parameters to queries consistent. The Required Queries in this transaction are identical in content to the Query Catalog in ITI-16.

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.

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	--

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
$XDSDocumentEntryHealthcareFacilityTypeCodeSchem e	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
$XDSDocumentEntryConfidentialityCodeScheme	XDSDocumentEntry. confidentialityCode	O[2]	M[2]
$XDSDocumentEntryFormatCode	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryStatus	XDSDocumentEntry. status	R	M

[1] This attribute is not listed by name in 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 ith value of the ‘code’ attribute (e.g. XDSDocumentEntryClassCode) shall correspond to the ith value of the ‘codeScheme’ attribute (e.g. XDSDocumentEntryClassCodeScheme).

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	XDSSubmissionSet. contentTypeCode	O	M
$XDSSubmissionSetStatus	XDSSubmissionSet. status	R	M

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

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	XDSFolder. codeList	O	M
$XDSFolderCodeListScheme	XDSFolder. codeList[1]	O[2]	M[2]
$XDSFolderStatus	XDSFolder.status	R	M

[1] This attribute is not listed by name in 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 ith value of the 'code' attribute (e.g. XDSDocumentEntryClassCode) shall correspond to the ith value of the 'codeScheme' attribute (e.g. XDSDocumentEntryClassCodeScheme).

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	XDSDocumentEntry. formatCode	O	M
$XDSDocumentEntryConfidentialityCode	XDSDocumentEntry. confidentialityCode	O	M

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

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

Note: A query for a single XDSDocumentEntry.uniqueId can return multiple results. See section 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

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

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

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

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

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

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.

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.

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	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.

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]	--

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

Note: A query for a single XDSDocumentEntry.uniqueId can return multiple results. See section 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 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 section 4.1.6 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.

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

3.18.4.1.2.4 Stored Query IDs

The standard XDS queries, documented in transaction ITI-16 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-`

Query Name	Query ID
	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 Query Scope

Within the Query Registry transaction [ITI-16], several queries, such as GetAll, are composite queries requiring multiple simple SQL queries to implement within the context of version 2.1 ebXML Registry standard. The Stored Query transaction shall be implemented such that each defined query is a single request/response message pair.

Note: This requirement is a reminder that version 2.1 Registry standard specifies a subset of the SQL standard that forces multiple simple queries to be used where a single more complicated query might have been used. Stored Query forces this implementation detail to be hidden from the Document Consumer. If the Document Registry actor is implemented with a version 2.1-registry engine then the Registry Adaptor must translate the single high-level Stored Query request into multiple lower level SQL requests and consolidate the results.

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 Appendix V. The specific values for the WSDL describing the Stored Query Service are described in this section.

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:

Note to the editor: please keep the following format for the sample text – courier new, 8pt, no spacing before and after the paragraph, tab stops every 1/8 of an inch for the first inch.

```
<?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="XDSRegistry_PortType">
    <operation name="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 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 W3C WS-Addressing standard. The body of the SOAP message is omitted for brevity; in a real scenario the empty element will be populated with the appropriate metadata.

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>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">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"/>
  </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>
```

3.18.4.1.2.8 Security considerations

The transaction shall be audited by the Document Consumer as follows

	Field Name	Opt.	Value Constraints
Event AuditMessage/ EventIdentification	EventID	M	EV (110112, DCM, “Query”)
	EventActionCode	M	EV “E” (Execute)
	EventDateTime	*M*	*not specialized*
	EventOutcomeIndicator	*M*	*not specialized*

	Field Name	Opt.	Value Constraints
	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-18", "IHE Transactions", "Registry Stored Query")
	ParticipantObjectSensitivity	U	not specialized
	ParticipantObjectID	M	The stored query UUID
	ParticipantObjectName	U	not specialized
	ParticipantObjectQuery	M	Shall hold the AdhocQueryRequest of the query, base64 encoded.
	ParticipantObjectDetail	NA	
	ParticipantObjectDescription	U	not further specialized

620 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.

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*

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

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 section 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
5. Substitute appropriate parameters as indicated in section 3.18.4.1.2.3.7 Parameters for Required Queries.
6. Execute the query
7. Return XML formatted metadata in an AdhocQueryResponse message.
   - The returned metadata is semantically identical to that returned from an SQL query in transaction ITI-16, Query Registry (ebRIM and ebRS version 2.1). The syntax of the metadata returned by this query shall be in ebRIM and ebRS version 3.0.

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 section 4.1.13 of the Technical Framework for additional details on formatting of error responses.

The Cross Community Access (XCA) profile updates the Expected Actions of this transaction to include use of the ebRIM “home” attribute. This attribute is used to hold the homeCommunityId in query responses. Refer to the XCA profile for more details.

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

See transaction ITI-16 Query Registry for more details on the FindDocuments query.

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="$XDSDocumentEntryPatientId">
      <rim:ValueList>
        <rim:Value>’st3498702^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO’</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="$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>
```

3.18.4.1.3.2 Generating SQL

Each parameter supplied implies a predicate in a query. Using SQL as an example and the following formatting of the example parameter values:

Parameter	Value	Implied Predicate
$XDSDocumentEntryPatientId	st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO	$XDSDocumentEntryPatientId = ‘st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO’
$XDSDocumentEntryCreationTimeFrom	200412252300	‘200412252300’ <= $XDSDocumentEntryCreationTimeFrom
$XDSDocumentEntryCreationTimeTo	200501010800	$XDSDocumentEntryCreationTimeTo < ‘200501010800’
$XDSDocumentEntryHealthcareFacilityTypeCode	(‘Emergency Department’)	$XDSDocumentEntryPracticeSettingCode IN (‘Emergency Department’)
$XDSDocumentEntryStatus	`urn:oasis:names:tc:ebxml-regrep:StatusType:Approved`	$XDSDocumentEntryStatus = `‘urn:oasis:names:tc:ebxml-regrep:StatusType:Approved’`

Optional parameters which are not included in the transaction do not contribute predicates to the query. The resulting SQL from the parameters shown above, coded for the database scheme used in ebRIM 2.1 results in the following SQL. Note that the $XDSDocumentEntryStatus attribute has been translated to ebRIM 2.1 format.

```
SELECT doc.id FROM ExtrinsicObject doc, ExternalIdentifier patid, Slot
creat, Slot to, Classification cl
WHERE
```

patid.id = 'st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO' AND
patid.registryobject = doc.id AND
patid.identificationScheme = 'urn:uuid:58a6f841-87b3-4a3e-92fd-a8ffeff98427' AND

'200412252300' <= creat.value AND
creat.name = 'creationTime' AND
creat.parent = doc.id AND

creat.value < '200501010800' AND
cl.nodeRepresentation IN ('Emergency Department') AND cl.classificationScheme = 'urn:uuid:f33fb8ac-18af-42cc-ae0e-ed0b0bdb91e1' AND
eo.status = 'Approved'

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 Notes’ section on http://www.ihe.net/IT_infra/committees/index.cfm contains a link to the IT Infrastructure Wiki where such supplemental content can be found.

```
<?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>Fairview Hospital</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;1.3.6.1.4.1.21367.2005.3.7&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>
```

3.18.4.1.3.4 Security considerations

The transaction shall be audited by the Document Registry as follows

	Field Name	**O pt.**	**Value Constraints**
Event AuditMessage/ EventIdentification	EventID	M	EV (110112, DCM, “Query”)
	EventActionCode	M	EV “E” (Execute)
	EventDateTime	*M*	*not specialized*

	Field Name	Opt.	Value Constraints
	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-18", "IHE Transactions", "Registry Stored Query")*
	ParticipantObjectSensitivity	*U*	*not specialized*
	ParticipantObjectID	M	The stored query UUID
	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 shall 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.

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*	

970 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.

	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*