Integrating the Healthcare Enterprise

IHE IT Infrastructure

Technical Framework Supplement

Multi-Patient Queries (MPQ)

Trial Implementation

Date: August 10, 2010

Author: ITI Technical Committee

Email: iti@ihe.net

Foreword

This is a supplement to the IHE IT Infrastructure Technical Framework V7.0. Each supplement undergoes a process of public comment and trial implementation before being incorporated into the volumes of the Technical Frameworks.

This supplement is submitted for Trial Implementation as of August 10, 2010 and will be available for testing at subsequent IHE Connectathons. The supplement may be amended based on the results of testing. Following successful testing it will be incorporated into the IT Infrastructure Technical Framework. Comments are invited and may be submitted on the IHE forums at http://forums.rsna.org/forumdisplay.php?f=198 or by email to iti@ihe.net.

This supplement describes changes to the existing technical framework documents and where indicated amends text by addition (**bold underline**) or removal (~~**bold strikethrough**~~), as well as addition of large new sections introduced by editor's instructions to "add new text" or similar, which for readability are not bolded or underlined.

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

Replace Section X.X by the following:

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

Information about the IHE IT Infrastructure can be found at: http://www.ihe.net/Domains/index.cfm

Information about the structure of IHE Technical Frameworks and Supplements can be found at: http://www.ihe.net/About/process.cfm and http://www.ihe.net/profiles/index.cfm

The current version of the IHE Technical Framework can be found at: http://www.ihe.net/Technical_Framework/index.cfm

CONTENTS

1 Introduction

This supplement that will be added to the ITI Technical Framework provides a means to do queries on multiple patients, based on pre-established meta-data. This supplement contains a new transaction, Multi-Patient Stored Query between a Document Consumer, and a Document Registry. The new transaction is based on the Registry Stored Query transaction [ITI-18], using a new query catalog. The multi-patient queries will allow clinical research, quality accreditation institutions and public health organizations to make sound decisions in their field of activity. The data to be aggregated, queried, and retrieved are pertaining to XDS Affinity Domains.

1.1 Profile Abstract

Currently, the Stored Query transaction [ITI-18] defines a single catalog of queries, which require that either a single patient ID, a folder ID, or a submission set ID are present in each query. While the existing query catalog serves various healthcare integration workflows, there are other cases where aggregated queries, i.e. queries not constrained to a single patient, folder, or catalog, are necessary. The domain that is currently needing the aggregated queries is the QRPH (the Quality, Research and Public Health), where data needs to be combined so that a pattern can ensue. Examples in the three areas would be repurposing, secondary use, and monitoring population health.

Quality accreditation organizations need to be able to aggregate data so that they can perform measurements of how institutions perform (*author* or *healthcareFacilityTypeCode*).

Clinical Research needs to be able to combine the results of different patients in a clinical trial (*typeCode*).

Public Health needs to have the means to make aggregated queries on certain fields such as *eventCodeList* in order to identify potential outbreaks and take appropriate decisions.

1.2 Open Issues and Questions

1. Currently this profile calls for the generation of one audit message per Patient ID present in the query response. An alternative is to generate only a single audit message (independent of how many Patient IDs present) for the query response that contains multiple Patient blocks. Any opinions?

1.3 Closed Issues

1. This supplement considered adding a new MPQ Document Registry actor and a MPQ Document Consumer actor to XDS.b, as well as a new transaction, using a new query catalog complementing the Registered Stored Query ITI-18 in the XDS.b supplement, creating a new profile addressing the issue of aggregated queries**. In the end we added a new transaction and defined two new queries that restrict parameter restrictions.**

2. The security requirements will be different since they can be defined separately from the existing XDS Document Registry Actor. A security assessment needs to be done.
3. The MPQ query should have at least one “key” parameter, but there can be other multiple parameters in order to serve the particular use cases. At least one parameter must be specified.
4. There is a difference between querying directly for aggregated data for statistical purposes, and querying for detailed data for more advanced purposes. Policies are to be put into place to restrict access to these types of queries, meaning that certain individuals or categories of individuals, or certain institutions can have access to one or to the other.
5. The aggregated queries for statistical purposes provide limited functionality. A list corresponding to the query with the UUID is provided. It is understood that the application performing the query will be responsible for counting. The implementation of this feature is left up to the implementers and it is out of scope of this profile. The ordering in the aggregated queries (for example by date) or will that be the user’s responsibility?
6. A mechanism for pseudonymization and how it should be used in the context of protecting the patient’s privacy, as well as providing a mechanism (when needed) to reverse it (traceability to the patient) is out of the scope of this profile.
7. This spec does not need to specify handling of homeCommunityId. The referenced Stored Query transaction’s documentation is adequate.
8. A time out value for retrieving the documents/values is not part of this profile but it is an implementation issue.
9. How will MPQ be updated so that it can reflect properly the future modifications brought in the future to the Stored Queries? Profile format changed to only show enhancements over Stored Query.
10. The vocabulary used to code the metadata is assumed to be the same across communities. This is not an assumption that is correct at this point in time. Answer: No cross-community vocabulary assumptions are made.
11. The multi-patient queries are applicable to XDS Affinity Domains but not to other local communities. However, this is an important case and it will be considered in future work. Answer: the extension of MPQ to Cross-Community environment has not yet been considered.
12. Only the FindDocuments and FindFolders Stored Queries have been extended for MPQ use. FindSubmissionSets and GetAll Stored Queries, the only other Stored Queries that have patient id as a parameter, are unsuited since there is no query parameter available that can be used to limit the scope of the query.
13. Integration of MPQ and Dsub – we believe MPQ is ready for use by DSUB and that the work of integrating the two will focus primarily on the security risk assessment.

Volume 1 – Integration Profiles

Glossary

Add the following terms to the Glossary:

1.7 History of Annual Changes

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

- Added the Multi-Patient Queries profile which allows aggregated queries to a registry.

2.1 Dependencies among Integration Profiles

Add the following to Table 2-1

MPQ	Audit Trail and Node Authentication	Each Document Registry actor and each Document Consumer shall be grouped with a Secure Node or a Secure Application Actor	Required to manage audit trail of exported PHI, node authentication and transport encryption
MPQ	Consistent Time	Each Document Registry actor and each Document Consumer shall be grouped with the Time Client Actor.	To ensure consistency among document and submission set dates

Add the following section to Section 2.2

2.2.25 Multi-Patient Queries Integration Profile

The Multi-Patient Queries profile defines a mechanism to enable aggregated queries to a Document Registry based on certain criteria needed by areas related to data analysis, such as quality accreditation of health care practitioners or health care facilities, clinical research trial data collection or population health monitoring.

Add Section 25

25 Multi-Patient Queries Integration Profile

The Multi-Patient Queries profile defines a mechanism to enable aggregated queries to a Document Registry based on certain criteria needed by areas related to data analysis, such as quality accreditation of health care practitioners or health care facilities, clinical research trial data collection or population health monitoring.

25.1 Actors/ Transactions

Figure 25.1-1 shows the actors directly involved in the MPQ Integration Profile in a solely XDS Affinity Domain and the relevant transactions between them. Other actors that may be indirectly involved due to their participation in other related profiles, etc. are not necessarily shown.

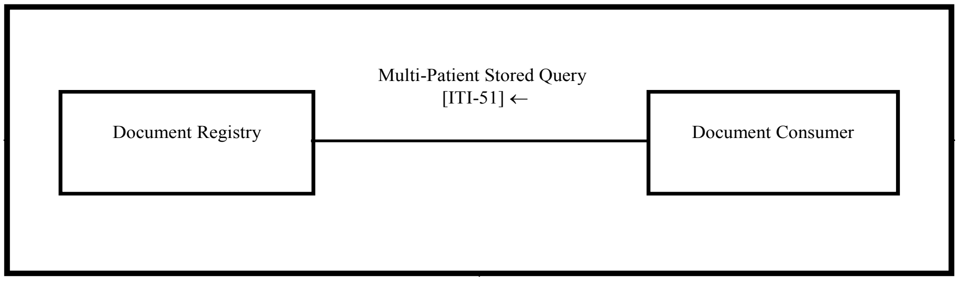


Figure 25.1-1. Multi-Patient Queries Actor Diagram

Table 25.1-1 lists the transactions for each actor directly involved in the Multi-Patient Query Profile. In order to claim support of this Integration Profile, an implementation must perform the required transactions (labeled "R"). Transactions labeled "O" are optional. A complete list of options defined by this Integration Profile and that implementations may choose to support is listed in ITI TF-1: 25.2.

Table 25.1-1. Multi-Patient Queries Integration Profile - Actors and Transactions

Actors	Transactions	Optionality	Section in Vol. 2
Document Registry	Multi-Patient Stored Query [ITI-51]	R	ITI TF-2b: 3.51
Document Consumer	Multi-Patient Stored Query [ITI-51]	R	ITI TF-2b: 3.51

25.2 Multi-Patient Query Integration Profile Options

Options that may be selected for this Integration Profile are listed in Table 25.2-1 along with the Actors to which they apply. Dependencies between options when applicable are specified in notes.

Table 25.2-1 MPQ - Actors and Options

Actor	Options	Vol & Section
Document Registry	*Asynchronous Web Services Exchange*	ITI TF-1: 25.2.2
Document Consumer	*Asynchronous Web Services Exchange*	ITI TF-1: 25.2.2

25.2.2 Asynchronous Web Services Exchange Option

Actors that support this option shall support the following:

- Document Consumer Actor shall support Asynchronous Web Services Exchange for the Multi-Patient Stored Query [ITI-51] transaction
- Document Registry Actor shall support Asynchronous Web Services Exchange for the Multi-Patient Stored Query [ITI-51] transaction

Use of Synchronous or Asynchronous Web Services Exchange is dictated by the individual install environment and policies. Refer to section ITI TF-2x: V.5 Synchronous and Asynchronous Web Services Exchange for an explanation of Asynchronous Web Services Exchange.

25.3 MPQ Process Flow

This section describes the process and information flow when a Document Consumer will query an Document Registry.

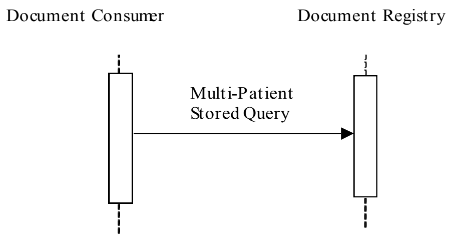


Figure 25.3-1: Basic Process Flow in Multi-Patient Queries Profile

25.4 Use Cases

25.4.1 Multi-Patient Query used in Public Health

Current Situation

The emergency department at Hospital A is treating patient B for certain symptoms, which are indicative of a reportable condition (such as A1H1), according to already established guidelines from an official public health agency. The symptoms mandate the use of a pre-determined value set for the XDS metadata *eventCodeList*. This can be a combination of the eventCodeList and observation such as "*influenza"* and "*possible A1H1"*. Hospital A sends any type of document capturing this information such as a Discharge Summary, an ED Encounter Summary (EDES), or in a larger sense any document intended for this purpose, using an XDS.b Provide and Register transaction to the local XDS repository, as well as a report to the appropriate public health agency P, using mechanisms which are outside the scope of this supplement.

After reviewing the report, the public health agency P determines that a review of recent patients' encounters with similar symptoms is necessary. Unfortunately, the XDS Document Registry only accepts patient specific queries, as currently defined in the Stored Query transaction. The public health agency P needs to obtain a list of patients with the appropriate symptoms from the healthcare providers.

Hospital A queries the local Document Registry for other Document Entries containing the same event code. Since it is not possible to query for multiple patients in one operation, a query is initiated for each patient known to the Document Registry. This is very time consuming and may not be very accurate.

Desirable Situation

The emergency department at Hospital A is treating patient B for certain symptoms, which are indicative of a reportable condition (such as A1H1), according to already established guidelines from an official public health agency. The symptoms mandate the use of a pre-determined value set for the XDS metadata *eventCodeList*. This can be a combination of the eventCodeList and observation such as "*influenza*" and "*possible A1H1"*. Hospital A sends any type of document capturing this information such as a Discharge Summary, an ED Encounter Summary (EDES), or in a larger sense any document intended for this purpose, using an XDS.b Provide and Register transaction to the local XDS repository, as well as a report to the appropriate public health agency P, using mechanisms which are outside the scope of this supplement.

After reviewing the report, the public health agency P determines that a review of recent patients' encounters with similar symptoms is necessary. Using Multi-Patient Queries, the health care provider is able to provide in a timely and accurate fashion all the documents with the having the same pre-determined value in the *eventCodeLis*t XDS metadata to the public health agency P. The public health agency is able to initiate an appropriate response and hence to contain a possible outbreak of the A1H1.

25.4.1.1 Post-factual and semi-real time reporting

There are needs to aggregate data so that a pattern can emerge, but the patients' identities need not to be known. For example, CDC (The Center for Disease Control and Prevention) or the InVS in France would like to know how many case of A1H1 are present at a national level at one point in time. In this case, there is no need to identify the patient, and unless other data is necessary to establish a trend (such as age, for example); an aggregated query on the metadata *eventCodeList* is sufficient using the *ObjectRefs* query. In this case irreversible pseudonymization or annonymization can be used since the data is employed statistically to generate a trend. This is the simplest case of implementing policies regarding security and privacy.

There are other cases where statistical analysis in semi-real time is desired, such as an aggregated query at a district level to do profiling by region in times of an influenza epidemic. Again, this is a situation where the patient's identity is not needed, but the number of cases and perhaps certain parameters such as the date. In order to be able to perform the aggregated queries, there has to be a minimum data set as per HIPAA recommendations.

25.4.1.2 Detailed queries

If more scrutiny is needed, such as in patient safety (reporting to FDA a patient safety issue concerning medications, medical equipment malfunction, or surgical procedures), or population health monitoring such as the real-time control of an outbreak), detailed queries can be used.

If in the Stored Query the *LeafClass* are specified the metadata of the document or of the folder (including the document ID and Repository ID) is returned. According to policies, these metadata can be pseudonymized or not.

For the multi-patient queries for detailed use, depending on the need, the policies regarding patient's privacy are different.

25.4.2 Technical Use Cases

The output of a Multi-Patient Query can be in one of two forms: a list of opaque identifiers, each identifying a matching document (assuming that the query targets Document Entries and not Folders or Submission Sets); or full metadata where all details known in metadata are returned.

25.4.2.1 Opaque Identifiers

Opaque identifiers, known in XDS as ObjectRefs, are useful to: discover the number of matches in the registry and then possibly to later retrieve the full metadata for the matching registry content. Applications that need only statistics (counts) can count the returned identifiers. Note that these identifiers represent documents (for example) that match the query and not patients. A single patient could have multiple matching documents.

25.4.2.2 Full Metadata

A Multi-Patient Query can return full metadata, known as LeafClass in XDS. This metadata includes Patient Ids and patient demographics from potentially multiple patients so it is difficult to protect yet must be protected. Because of this sensitivity this type of return result would likely be only allowed by very highly trusted systems and thus this query is likely not to be available as widely as others.

25.5 Security Considerations

This profile applies the same ATNA grouping to protect against the typical XDS identified risks. This profile may be grouped with XUA to further provide authentication of the user of the result.

The new security and privacy considerations arise because this profile allows for a single query to result in multiple patients XDS metadata to be returned in one transaction. Although the XDS metadata is not high grade health data it is still identifiable health information and thus needs to be protected. The combination of multiple patient's protected information in the same result results in a more difficult task to assure that the intended recipient has all the authorizations necessary for the intended use. In classical XDS queries the query request/response is constrained to a single patient and therefore the access control decision can be done across the whole transaction.

This profile allows for two different types of return result. The ObjectRef result can be used to limit the exposure as this result will return only opaque identifiers of the matching documents. It is expected that this result would be more widely allowed. The Document Consumer can still obtain the full metadata but must use the classic XDS queries on an object-by-object basis thus allowing for transactions that are constrained to a single patient. This additional set of transactions to retrieve the metadata may be unnecessary when the system doing the query is authorized to use the LeafClass response. For example when the querying system is known as a system that will protect the information to the same degree. Where it is known that this querying system will apply the appropriate access control prior to ultimate use or disclosure.

Appendix A Actor Summary Definitions

Appendix B Transaction Summary Definitions

Multi-Patient Stored Query - A Document Consumer actor issues a Multi-Patient Stored Query to a Document Registry to locate documents that meet the user’s specified query criteria. The Document Registry returns a list of document entries pertaining to multiple patients found to meet the specified criteria, including the locations and identifier of each corresponding document in one or more Document Repository.

Volume 2b - Transactions

Add Section 3.51

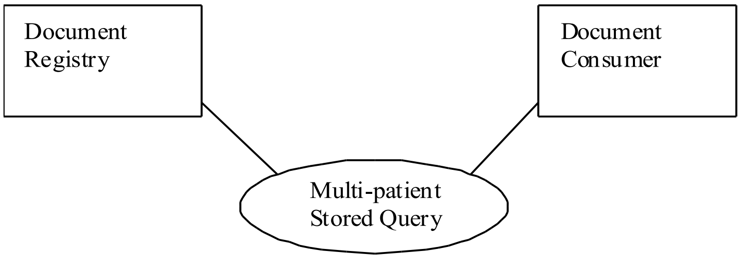
3.51 Multi-Patient Stored Query

This section corresponds to Transaction ITI-51 of the IHE Technical Framework. Transaction ITI-51 is used by the Document Consumer and Document Registry actors.

3.51.1 Scope

The Multi-Patient Stored Query supports a variety of queries for multiple patients. It is based on the Registry Stored Query transaction [ITI-18]. The main difference is the set of queries, which is specified in this transaction.

3.51.2 Use Case Roles



Actor: Document Consumer

Role: Issues a Multi-Patient Stored Query to retrieve metadata based on criteria common to multiple patients

Actor: Document Registry

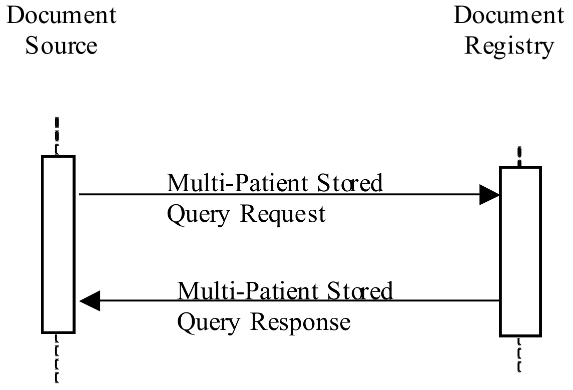
Role: Responds to a Multi-Patient Stored Query by providing the metadata or object references of registry objects which satisfy the query parameters

3.51.3 Referenced Standard

Implementers of this transaction shall comply with all requirements described in ITI TF-2x: Appendix V Web Services for IHE Transactions.

ebRIM	OASIS ebXML Registry Information Model v3.0
ebRS	OASIS ebXML Registry Services Specifications v3.0
ITI-18	ITI TF-2a: 3.18: Registry Stored Query
Appendix V	ITI TF-2x: Appendix V: Web Services for IHE Transactions Contains references to all Web Services standards and requirements of use

3.51.4 Interaction Diagram


Document Source
Document Registry
Multi-Patient Stored Query Request
Multi-Patient Stored Query Response

3.51.4.1 Multi-Patient Stored Query Request

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

- A reference to a pre-defined query stored on the Document Registry actor
- Parameters to the query

3.51.4.1.1 Trigger Events

The message is initiated when a Document Consumer wants to query for metadata based on criteria spanning multiple patients (multiple Patient IDs).

3.51.4.1.2 Message Semantics

The message semantics are identical to those documented for the Registry Stored Query [ITI-18] transaction except where noted below. The following sections document the differences.

Document Source and Document Registry actors that support the Asynchronous Web Services Exchange option shall support Asynchronous Web Services requirements as defined in ITI TF-2x: V.5.

3.51.4.1.2.1 Query Definitions

This profile defines the following Stored Queries that may query for multiple Patient Ids.

3.51.4.1.2.1.1 FindDocumentsForMultiplePatients

This Multi-Patient Query is semantically identical to the FindDocuments Stored Query (see ITI TF-2a: 3.18.4.1.2.3.7.1) except:

– $XDSDocumentEntryPatientId is optional (may have zero values).

- $XDSDocumentEntryPatientId may contain multiple values.
- At least one of the ClassCode, EventCodeList, or HealthcareFacilityTypeCode shall be specified in the provided set of parameters.

Returns: XDSDocumentEntry or ObjectRef objects matching the query parameters

Parameter Name	Attribute	Opt	Mult
$XDSDocumentEntryPatientId	XDSDocumentEntry. patientId	O	M
$XDSDocumentEntryClassCode[1] [2]	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. serviceStartTime	O	--
$XDSDocumentEntryServiceStartTimeTo	Upper value of XDSDocumentEntry. serviceStartTime	O	--
$XDSDocumentEntryServiceStopTimeFrom	Lower value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryServiceStopTimeTo	Upper value of XDSDocumentEntry. serviceStopTime	O	--
$XDSDocumentEntryHealthcareFacilityTypeCode[1] [2]	XDSDocumentEntry. healthcareFacilityTypeCode	O	M
$XDSDocumentEntryEventCodeList[1] [2]	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.

[2] At least one of $XDSDocumentEntryClassCode, $XDSDocumentEntryEventCodeList, or $XDSDocumentEntryHealthcareFacilityTypeCode shall be specified.

[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)

3.51.4.1.2.1.2 FindFoldersForMultiplePatients

This Multi-Patient Query is semantically identical to the FindFolders Stored Query (see ITI TF-2a: 3.18.4.1.2.3.7.3) except:

- $XDSFolderPatientId is optional (may have zero values).
- $XDSFolderPatientId may contain multiple values.
- $XDSFolderCodeList shall be a required parameter.

Returns: XDSFolder or ObjectRef objects matching the query parameters

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

3.51.4.1.2.2 Multi-Patient Stored Query IDs

The following Query Ids shall be used to represent these queries.

Query Name	Query ID
FindDocumentsForMultiplePatients	`urn:uuid:3d1bdb10-39a2-11de-89c2-2f44d94eaa9f`
FindFoldersForMultiplePatients	`urn:uuid:50d3f5ac-39a2-11de-a1ca-b366239e58df`

3.51.4.1.2.3 Web Services Transport

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

The Document Registry actor shall accept a Multi-Patient Stored Query Request formatted as a SIMPLE SOAP message and respond with a Multi-Patient Stored Query Response formatted as a SIMPLE SOAP message. The Document Consumer actor shall generate the Multi-Patient Stored Query Request formatted as a SIMPLE SOAP message and accept a Multi-Patient 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    -> "MultiPatientStoredQuery_Message"
      query response   -> "MultiPatientStoredQueryResponse_Message"
      portType         -> "DocumentRegistry_PortType"
      operation        -> "DocumentRegistry_MultiPatientStoredQuery"
      SOAP 1.2 binding -> "DocumentRegistry_Binding_Soap12"
      SOAP 1.2 port    -> "DocumentRegistry_Port_Soap12"
```

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

These are the requirements for the Multi-Patient 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 Multi-Patient Stored Query Request message shall be defined as “query:AdhocQueryRequest”
- The /definitions/message/part/@element attribute of the Multi-Patient Stored Query Response message shall be defined as “query:AdhocQueryResponse”
- The /definitions/portType/operation/input/@wsaw:Action attribute for the Multi-Patient Stored Query Request message shall be defined as “urn:ihe:iti:2009:MultiPatientStoredQuery”
- The /definitions/portType/operation/output/@wsaw:Action attribute for the Multi-Patient Stored Query Response message shall be defined as “urn:ihe:iti:2009:MultiPatientStoredQueryResponse”
- The /definitions/binding/operation/soap12:operation/@soapAction attribute should be defined as “urn:ihe:iti:2009:MultiPatientStoredQuery”

The following WSDL fragment shows an example of Multi-Patient 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>Multi-Patient Stored Query</documentation>
    <part name="body" element="query:AdhocQueryRequest"/>
  </message>
  <message name="RegistryStoredQueryResponse_Message">
    <documentation>Multi-Patient Stored Query Response</documentation>
    <part name="body" element="query:AdhocQueryResponse"/>
  </message>
  ...
  <portType name="MPQRegistry_PortType">
    <operation name="MultiPatientStoredQuery">
      <input message="ihe:RegistryStoredQuery_Message"
          wsaw:Action="urn:ihe:iti:2009:MultiPatientStoredQuery"/>
      <output message="ihe:RegistryStoredQueryResponse_Message"
          wsaw:Action="urn:ihe:iti:2009:MultiPatientStoredQueryResponse"/>
    </operation>
    ...
  </portType>
  ...
</definitions>
```

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

3.51.4.1.2.4 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 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.

3.51.4.1.2.4.1 Sample Multi-Patient 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:2009:MultiPatientStoredQuery</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/IHEMPQRegistry.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"/>

            <!-- FindDocumentsForMultiplePatients -->
            <rim:AdhocQuery id="urn:uuid:3d1bdb10-39a2-11de-89c2-2f44d94eaa9f">
                <rim:Slot name="$XDSDocumentEntryStatus">
                    <rim:ValueList>
                        <rim:Value>('urn:oasis:names:tc:ebxml-
regrep:ResponseStatusType:Approved')</rim:Value>
                    </rim:ValueList>
                </rim:Slot>
                <rim:Slot name="$XDSDocumentEntryClassCode">
                    <rim:ValueList>
                        <rim:Value>'26436-6'</rim:Value>
                    </rim:ValueList>
                </rim:Slot>
                <rim:Slot name="$XDSDocumentEntryClassCodeScheme">
                    <rim:ValueList>
                        <rim:Value>'LOINC'</rim:Value>
                    </rim:ValueList>
                </rim:Slot>


                <!-- Note the lack of a specification of the $XDSDocumentEntryPatientId parameter
-->



            </rim:AdhocQuery>
        </query:AdhocQueryRequest>
    </s:Body>
</s:Envelope>
```

3.51.4.1.2.4.2 Sample Multi-Patient 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:2009:MultiPatientStoredQueryResponse</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"
            status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
            <rim:RegistryObjectList xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">

                <!-- Internal details of ExtrinsicObjects are not shown -->

                <rim:ExtrinsicObject/>
                <rim:ExtrinsicObject/>
                <rim:ExtrinsicObject/>
                <rim:ExtrinsicObject/>
                <rim:ExtrinsicObject/>
                <rim:ExtrinsicObject/>
            </rim:RegistryObjectList>
        </query:AdhocQueryResponse>
    </s:Body>
</s:Envelope>
```

3.51.4.1.3 Expected Actions

See Registry Stored Query [ITI-18] for Expected Actions.

3.51.5 Security Considerations

All of the Security Considerations found in ITI-18 apply with the following further profiling.

It is expected that the ATNA Secure Node authentication would be used to restrict access to the MPQ transaction. It is expected that few systems would be allowed to request the LeafClass return result.

3.51.5.1 Security Audit Considerations

The Actors involved shall record audit events for each patient identity that has been included in the result according to the following. It is important for security auditing that the audit message contain one patient identity to better handle these messages in the Audit Record Repository:

3.51.5.1.1 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-51”, “IHE Transactions”, “Multi-Patient Query”0*
Source (Document Consumer) (1)			
Human Requestor (0..n)			
Destination (Document Registry) (1)			
Audit Source (Document Consumer) (1)			
Patient (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*	

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 **(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-51", "IHE Transactions", "Multi-Patient 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*

3.51.5.1.2 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-51", "IHE Transactions", "Multi-Patient 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*

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-51", "IHE Transactions", "Multi-Patient 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*