OHF XDS Document Consumer

Architecture & API Documentation

Version X.X.X

seknoop[AT]us[DOT]ibm[DOT]com | Sarah Knoop

Contents

1. Introduction

The Eclipse Foundation is a not-for-profit corporation formed to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services. Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software.

www.eclipse.org

The Eclipse Open Healthcare Framework (EOHF) is a project within Eclipse formed for the purpose of expediting healthcare informatics technology. The project is composed of extensible frameworks and tools which emphasize the use of existing and emerging standards in order to encourage interoperable open source infrastructure, thereby lowering integration barriers.

www.eclipse.org/ohf

The Integrating the Healthcare Enterprise (IHE) is an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information. IHE promotes the coordinated use of established standards such as DICOM and HL7 to address specific clinical needs in support of optimal patient care. Systems developed in accordance with IHE communicate with one another better, are easier to implement, and enable care providers to use information more effectively.

www.ihe.net

The IHE Technical Frameworks are a resource for users, developers and implementers of healthcare imaging and information systems. They define specific implementations of established standards to achieve effective systems integration, facilitate appropriate sharing of medical information and support optimal patient care. They are expanded annually, after a period of public review, and maintained regularly by the IHE Technical Committees through the identification and correction of errata.

http://www.ihe.net/Technical_Framework/index.cfm

This document describes the current release of the Eclipse OHF plugin implementation of the IHE ITI Technical Framework XDS Profile Document Consumer Actor. This implementation supports the following IHE Transactions: ITI-16: Query Registry, ITI-17: Retrieve Document and ITI-18: Registry Stored Query.

2. Getting Started

2.1 Platform Requirements

Verify that the following platform requirements are installed on your workstation, and if not follow the links provided to download and install.

Eclipse SDK 3.2	http://www.eclipse.org/downloads/
Java JDK 5.0	http://java.sun.com/javase/downloads/index.jsp
Eclipse Modeling Framework 2.2.0	http://www.eclipse.org/emf/

2.2 Source Files

Information on how to access the Eclipse CVS technology repository is found on the eclipse wiki:

http://wiki.eclipse.org/index.php/CVS_Howto

Download from dev.eclipse.org/technology/org.eclipse.ohf/plugins

- org.eclipse.ohf.ihe.xds.consumer

For details regarding plugin contents, see the README.txt located in the resources/doc folder of each plugin.

2.3 Dependencies

2.3.1 Other OHF Plugins

Plugin dependencies include the following from dev.eclipse.org/technology/org.eclipse.ohf/plugins

• org.apache.log4j	Debug, warning and error logging
• org.eclipse.ohf.ihe.atna.audit	Support of audit events for relevant transactions
• org.eclipse.ohf.ihe.common.ebXML._2._1 payload	Model to process and construct transaction
• org.eclipse.ohf.ihe.common.hl7v2	HL7 v2 data type support for XDS metadata
• org.eclipse.ohf.hl7v2.core	HL7 v2 processing needed by the above plugin
• org.elcipse.ohf.utilities	General utilities needed by the above plugin
• org.eclipse.ohf.ihe.xds.metadata	Model to represent XDS metadata
• org.eclipse.ohf.ihe.xds.metadata.extract	Model to render XDS metadata from other formats
• org.eclipse.ohf.ihe.xds.soap	SOAP messaging support for transactions
• org.apache.axis	Apache Axis 1.3 to support the above plugin

2.3.2 External Sources

No additional external dependencies.

2.4 Resources

2.4.1 Other OHF plugin documentation

The following OHF plugin documents are related to the OHF XDS Document Consumer:

- OHF ATNA Audit Client
- OHF XDS Metadata Model
- OHF XDS SOAP Client

2.4.2 IHE ITI Technical Framework

Nine IHE IT Infrastructure Integration Profiles are specified as Final Text in the Version 2.0 ITI Technical Framework: Cross-Enterprise Document Sharing (XDS), Patient Identifier Cross-Referencing (PIX), Patient Demographics Query (PDQ), Audit trail and Node Authentication (ATNA), Consistent Time (CT), Enterprise User Authentication (EUA), Retrieve Information for Display (RID), Patient Synchronized Applications (PSA), and Personnel White Pages (PWP).

The IHE ITI Technical Framework can be found on the following website: http://www.ihe.net/Technical_Framework/index.cfm#IT.

Key sections relevant to the OHF XDS Document Consumer include (but are not limited to):

- Volume 1, Section 10 and Appendices A,B, E, J, K, L and M
- Volume 2, Section 1, Section 2, Section 3.16, 3.17 and Appendices J and K
- Stored Query Supplement also available at: http://www.ihe.net/Technical_Framework/index.cfm#IT

2.4.3 Newsgroup

Any unanswered technical questions may be posted to Eclipse OHF newsgroup. The newsgroup is located at news://news.eclipse.org/eclipse.technology.ohf.

You can request a password at: http://www.eclipse.org/newsgroups/main.html**.**

3. API Documentation

The XDS Document Consumer provides API for the execution of the following IHE Transactions: ITI-16: Query Registry, ITI-17: Retrieve Document and ITI-18: Registry Stored Query. The API also provides utility functions for the construction of SQL queries for ITI-16. The API currently supports automatic construction of the FindDocuments and the GetDocument query for ITI-16. Support for ITI-18 is pending implementation. Compliance with the edits to the IHE Technical Framework for 2006 is pending.

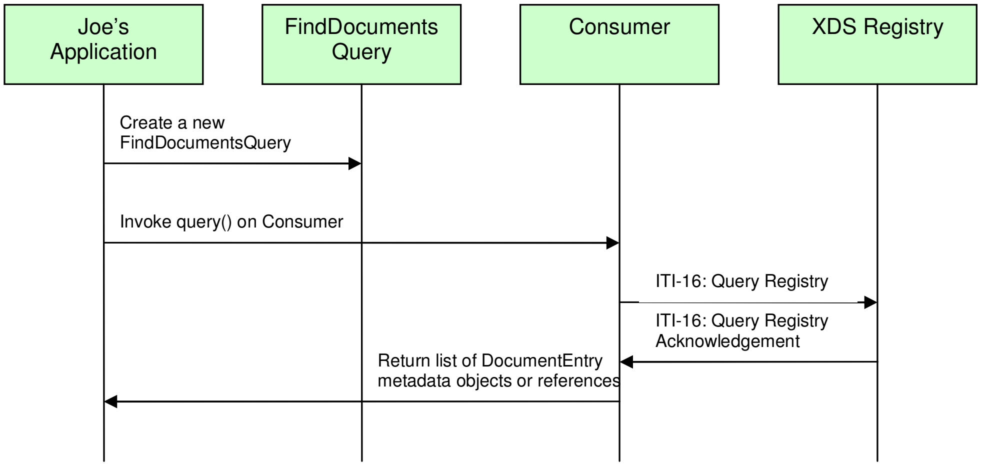
3.1 Use Case 1 – Query Registry: FindDocuments Query

Joe User, using his EMR Application, wants to find all documents where:

- patientID is “st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO”
- Only “Approved” documents are returned
- Creation time on the document is between 200412252300 and 200501010800
- Healthcare Facility Type Code is “Outpatient”

Comment: SEK – get new params that are on our registry.

3.1.1 Flow of Execution



Joe’s Application
FindDocuments Query
Consumer
XDS Registry
Create a new FindDocumentsQuery
Invoke query() on Consumer
ITI-16: Query Registry
ITI-16: Query Registry Acknowledgement
Return list of DocumentEntry metadata objects or references

3.1.2 API Details

3.1.2.1 org.eclipse.ohf.ihe.xds.consumer.Consumer

Constructor Summary

Consumer(java.lang.String regURL, boolean doAudit) Constructor

Method Summary

boolean	**getHackMetadataFlag**()
java.lang.String	**getRegURL**()
boolean	**isDoAudit**()
java.util.List	**query**(Query q, boolean returnReferencesOnly, java.lang.String initiatingUser) Transaction ITI-16: Query Registry Query for and retrieve a list of docoment metadata or metadata references based on the input attributes.
java.util.List	**query**(java.lang.String sqlQueryString, boolean returnReferencesOnly, java.lang.String initiatingUser) Transaction ITI-16: Query Registry Query for and retrieve a list of docoment metadata or metadata references based on the input attributes.
java.io.InputStream	**retrieveDocument**(java.lang.String uri, java.lang.String initiatingUser) Transaction ITI-17: Retrieve Document Retrieve an input stream from which the document content can be read.
void	**setDoAudit**(boolean doAudit)
void	**setRegURL**(java.lang.String regURL)

3.1.2.2 org.eclipse.ohf.ihe.xds.consumer.query.FindDocumentsQuery

Constructor Summary

FindDocumentsQuery(java.lang.String patientID, org.eclipse.ohf.ihe.xds.metadata.AvailabilityStatusType[] status)

Constructor, with minimal parameters.
FindDocumentsQuery(java.lang.String patientID, java.lang.String[] classCodes, DateTimeRange[] dateTimeRanges, java.lang.String[] practiceSettingCodes, java.lang.String[] healthCareFacilityCodes, java.lang.String[] eventCodeList, org.eclipse.ohf.ihe.xds.metadata.AvailabilityStatusType[] status) Constructor witf full set of parameters.

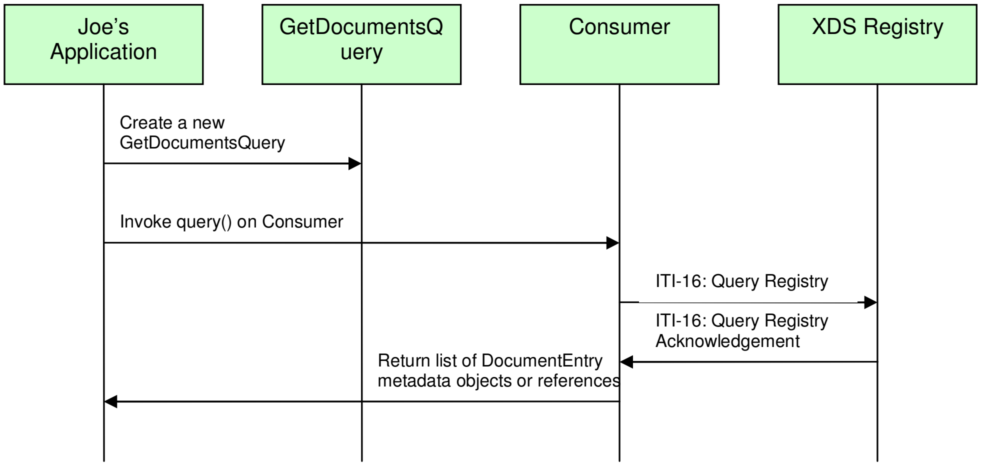
3.2 Use Case 2 – Query Registry: GetDocument Query

Joe User, using his EMR Application, wants to find all documents where:

- The document uniqueId is "1.2.3.4.55.6"

Comment: SEK – get new params that are on our registry.

3.2.1 Flow of Execution


Joe's Application
GetDocumentsQuery
Consumer
XDS Registry
Create a new GetDocumentsQuery
Invoke query() on Consumer
ITI-16: Query Registry
ITI-16: Query Registry Acknowledgement
Return list of DocumentEntry metadata objects or references

3.2.2 API Details

NOTE: Refer back to 3.1.2.1 for OHF XDS Document Consumer API details.

3.2.2.1 org.eclipse.ohf.ihe.xds.consumer.query.GetDocumentQuery

Constructor Summary
GetDocumentQuery(java.lang.String docID, boolean isUUID)

Constructor.

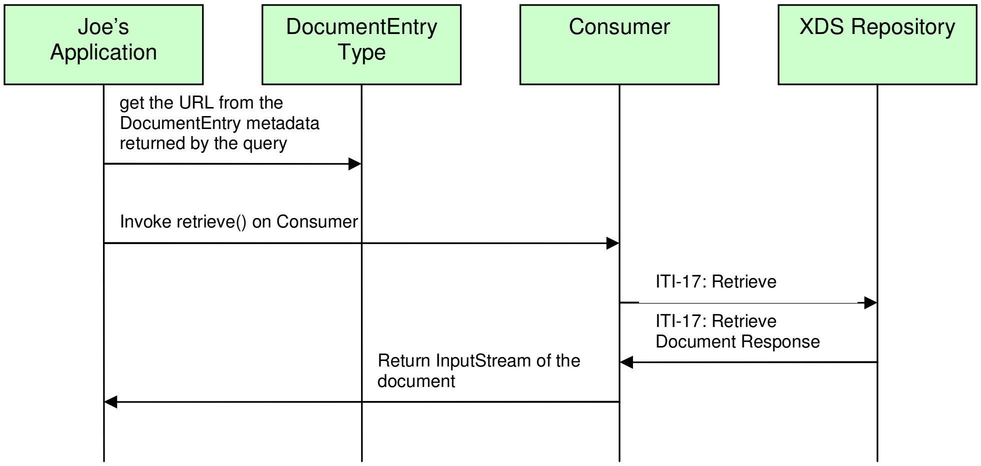
3.3 Use Case 3 – Retrieve

Joe User, using his EMR Application, wants to retrieve a document he has just queried and found where:

- The document URL is “https://some.uri.com”

Comment: SEK need doc url on our repository.

3.3.1 Flow of Execution


Joe’s Application
DocumentEntry Type
Consumer
XDS Repository
get the URL from the DocumentEntry metadata returned by the query
Invoke retrieve() on Consumer
ITI-17: Retrieve
ITI-17: Retrieve Document Response
Return InputStream of the document

3.3.2 API Details

NOTE: Refer back to 3.1.2.1 for OHF XDS Document Consumer API details.

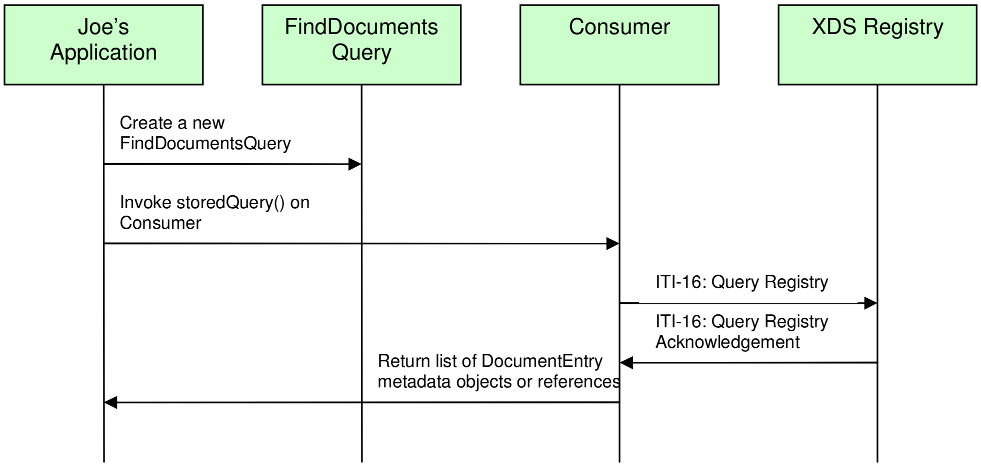
3.4 Use Case 4 – Registry Stored Query: FindDocuments Query

Joe User, using his EMR Application, wants to find all documents where:

- patientID is “st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO”
- Only “Approved” documents are returned
- Creation time on the document is between 200412252300 and 200501010800
- Healthcare Facility Type Code is “Outpatient”

Comment: SEK – get new params that are on our registry.

3.4.1 Flow of Execution


Joe's Application
FindDocuments Query
Consumer
XDS Registry
Create a new FindDocumentsQuery
Invoke storedQuery() on Consumer
ITI-16: Query Registry
ITI-16: Query Registry Acknowledgement
Return list of DocumentEntry metadata objects or references

3.4.2 API Details

NOTE 1: Refer back to 3.1.2 for OHF XDS Document Consumer and FindDocumentsQuery API details.

NOTE 2: This API is not implemented yet because the IHE Profile Supplement for Stored Query is still in the editing process and not yet ready for trial implementation.

4. Sample Code

Below we provide sample code corresponding to the use cases outlined in the previous section

4.1 Example for Use Case 1 – Query Registry: FindDocuments Query

4.1.1 Description

Joe User, using his EMR Application, wants to find all documents where:

- patientID is “st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO”
- Only “Approved” documents are returned
- Creation time on the document is between 200412252300 and 200501010800
- Healthcare Facility Type Code is “Outpatient”

4.1.2 Code

```
Consumer c = null;

String registryURL = “http://my.registry.url:8080”;

try {

      c = new Consumer(registryURL, true); // true = turn on auditing

} catch (Exception e1) {

      logger.error("Exception", e1);

      return ERROR;

}


// form query

DateTimeRange[] ranges = {new
DateTimeRange(DocumentEntryConstants.CREATION_TIME, “200412252300”,
“200501010800”};

String[] hcfCodes = {“Outpatient”};

AvailabilityStatusType status = {AvailabilityStatusType.Approved};


FindDocumentsQuery query = new
FindDocumentsQuery(“st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO”,null, ranges,
null, hcfCodes, null, status);


// execute query

List docList = null;

try {
      docList = c.query(query,false,"JOE USER");
} catch (Exception e){
      logger.error(e, e);
      return ERROR;
}
if(docList == null){
      logger.fatal("NO DOCUMENT FOUND");
      return ERROR;
}
```

4.2 Example for Use Case 2 – Query Registry: GetDocument Query

4.2.1 Description

Joe User, using his EMR Application, wants to find all documents where:

- The document uniqueId is “1.2.3.4.55.6”

Comment: SEK – get new params that are on our registry.

4.2.2 Code

```
// we assume a Consumer c has already been appropriately constructed as in
//4.1.2


//execute query
DocumentEntryType doc = null;
List docList = null;
try {
      docList = c.query(new GetDocumentQuery(this.documentId, true),false,"SOME
      USER");
} catch (Exception e){
      logger.error(e, e);
      return ERROR;
}
if(docList == null){
      logger.fatal("NO DOCUMENT FOUND");
      return ERROR;
}
doc = (DocumentEntryType)docList.get(0);
if(doc == null){
      logger.fatal("NO DOCUMENT FOUND");
      return ERROR;

}
```

4.3 Example for Use Case 3 - Retrieve

4.3.1 Description

Joe User, using his EMR Application, wants to retrieve a document he has just queried and found where:

- The document URL is “https://some.uri.com”

4.3.2 Code

```
// we assume a Consumer c has already been appropriately constructed as in
//4.1.2


// we assume a DocumentEntryType doc filled with XDS metadata for the document
// we want has be has already obtained from the query as in 4.2.2. we assume
//that the URI in this metadata is “http://some.url.com”


// do the retrieve
InputStream is;
try{
      is = c.retrieveDocument(doc.getUri()/*, null, null*/, "SOME USER");
}catch(Exception e){
      logger.fatal("Error when attempting to retrieve from: " + doc.getUri(),
      e);
      return ERROR;
}
logger.debug("DONE RETRIEVE: " + doc.getUri());
```

4.4 Example for Use Case 4 – Registry Stored Query: FindDocuments Query

4.4.1 Description

Joe User, using his EMR Application, wants to find all documents where:

- patientID is “st3498702^^^&1.3.6.1.4.1.21367.2005.3.7&ISO”

- Only “Approved” documents are returned
- Creation time on the document is between 200412252300 and 200501010800
- Healthcare Facility Type Code is “Outpatient”

4.4.2 Code

```
code here
```

```
NOTE 2: This API is not implemented yet because the IHE Profile Supplement for
Stored Query is still in the editing process and not yet ready for trial
implementation.
```

5. Additional Sections – repeat as necessary

Any additional sections needed are added at this point.

6. Glossary

Define any non-common knowledge terms or acronyms here. Provide web-site reference if applicable.