



**OHF XDS SOAP Client**  
**Architecture & API Documentation**  
**Version X.X.X**

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



---

# Contents

1.	Introduction.....	3
2.	Getting Started .....	4
2.1	Platform Requirements .....	4
2.2	Source Files .....	4
2.3	Dependencies .....	4
2.3.1	Other OHF Plugins .....	4
2.3.2	External Sources .....	4
2.4	Resources .....	5
2.4.1	Apache Axis 1.3.....	5
2.4.2	IHE ITI Technical Framework .....	5
2.4.3	Newsgroup.....	5
3.	API Documentation.....	6
3.1	Use Case 1 – Query Registry .....	6
3.1.1	Flow of Execution .....	6
3.1.2	API Details .....	6
3.1.2.1	org.eclipse.ohf.ihe.xds.soap.XDSSoapClient.....	6
3.2	Use Case 2 – Provide and Register Document Set.....	6
3.2.1	Flow of Execution .....	6
3.2.2	API Details .....	6
3.2.2.1	org.eclipse.ohf.ihe.xds.soap.SOAPAttachment.....	6
4.	Sample Code.....	9
4.1	Example for Use Case 1 – Query Registry .....	9
4.1.1	Description .....	9
4.1.2	Code .....	9
4.2	Example for Use Case 2 – Provide And Register Document Set.....	9
4.2.1	Description .....	9
4.2.2	Code .....	9
5.	Additional Sections – repeat as necessary .....	11
6.	Glossary .....	12



---

# 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](http://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](http://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](http://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](http://www.ihe.net/Technical_Framework/index.cfm)

This document describes the current release of the Eclipse OHF plugin implementation of the client side of the SOAP based transactions used in XDS; chiefly ITI-15: Provide and Register Document Set and ITI-16: Query Registry.



---

## 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>

---

### 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](http://wiki.eclipse.org/index.php/CVS_Howto)

Download from [dev.eclipse.org/technology/org.eclipse.ohf/plugins](http://dev.eclipse.org/technology/org.eclipse.ohf/plugins)

- org.eclipse.ohf.ihe.xds.soap

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

---

### 2.3 Dependencies

Document dependencies the source has (feel free to extend this section)

#### 2.3.1 Other OHF Plugins

Plugin dependencies include the following from [dev.eclipse.org/technology/org.eclipse.ohf/plugins](http://dev.eclipse.org/technology/org.eclipse.ohf/plugins)

- org.apache.axis Apache Axis 1.3 to support SOAP messaging
- org.apache.log4j Debug, warning and error logging

#### 2.3.2 External Sources

At this time, the XDS Soap Client is dependant on the Axis 1.3 API. This API does not provide an implementation of the `javax.activation.DataHandler` nor the `javax.mail.internet.MimeMultipart` packages needed for compilation and SOAP with attachment support. Java Mail 1.3.3 (`mailapi.jar`) and Java Activation Framework 1.0.2 (`activation.jar`) must be downloaded separately and incorporated into the build path or plugin dependencies list for this plugin. Take note of exact versions of these jars. These .jars can be found at

JAF v1.0.2:

<http://java.sun.com/products/archive/javabeans/jaf102.html>

JAVA MAIL v1.3.3:

[http://java.sun.com/products/javamail/javamail-1\\_3\\_3.htmlResources](http://java.sun.com/products/javamail/javamail-1_3_3.htmlResources)

---



**\*\* NOTE: At the present time we are still experiencing bugs related to the use of SOAP attachments. We are working on the solution to this problem and will update this document upon resolution.**

---

## 2.4 Resources

### 2.4.1 Apache Axis 1.3

Further documentation and source code for Apache Axis 1.3 can be found at:

<http://ws.apache.org/axis/>

### 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](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>.



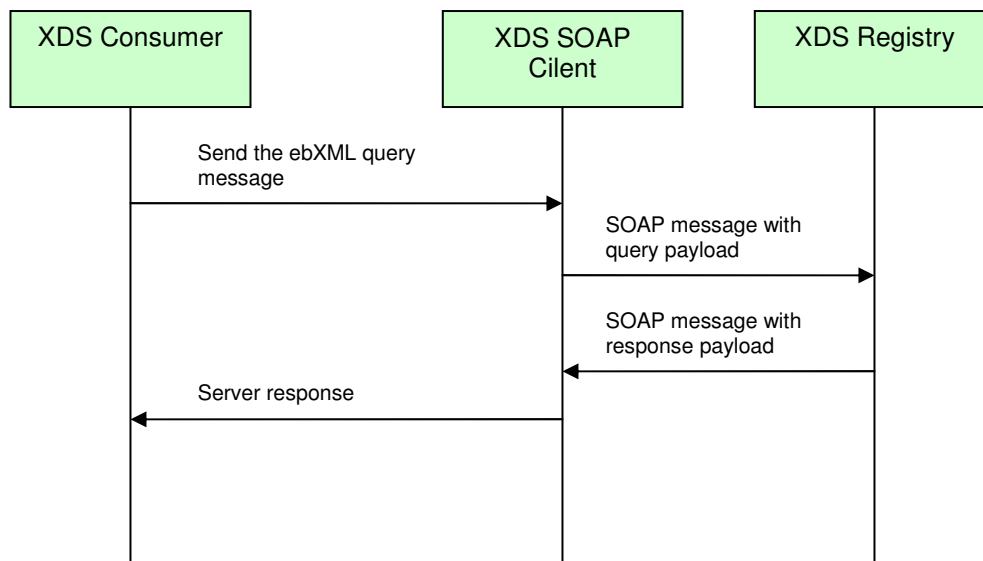
## 3. API Documentation

The OHF XDS SOAP Client is intended to provide a simple and common interface for the XDS client actors (Document Source and Document Consumer) to execute their respective transactions using SOAP. The SOAP Client constructs and SOAP message, adds any attachments and sends the message. Upon receiving a response from the server, the SOAP client unwraps the SOAP Envelope from the response, returning to the XDS client only the contents of the SOAP Body as a DOM element. We note that the API for this plugin is not stable due to the pending integration with other OHF components.

### 3.1 Use Case 1 – Query Registry

The OHF XDS SOAP Client constructs the SOAP message for the XDS Consumer Query Registry Transaction (ITI-16).

#### 3.1.1 Flow of Execution



#### 3.1.2 API Details

##### 3.1.2.1 org.eclipse.ohf.ihe.xds.soap.XDSSoapClient

### Method Summary

org.w3c.dom.Element	<a href="#">send</a> (java.lang.String request, <a href="#">SoapAttachment</a> [] attachments, java.lang.String url) Constructs the SOAPMessage from the parameters, sends the messages
---------------------	--

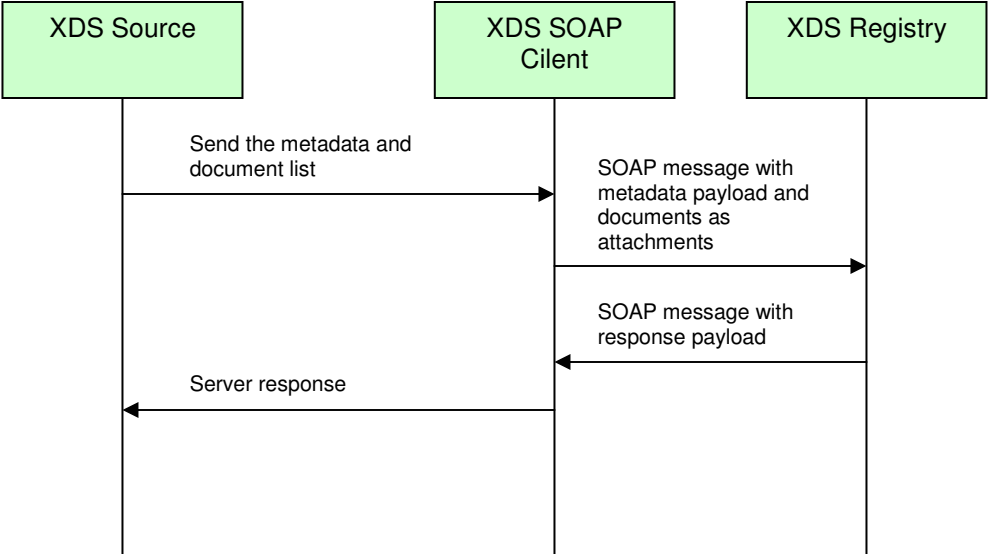


to the designated url and processes the response SOAPMessage.

### 3.2 Use Case 2 – Provide and Register Document Set

The OHF XDS SOAP Client constructs the SOAP message for the XDS Document Source Provide and Register Document Set Transaction (ITI-15).

#### 3.2.1 Flow of Execution



#### 3.2.2 API Details

NOTE 1: Refer back to 3.1.2 for OHF XDS Client API details.

##### 3.2.2.1 org.eclipse.ohf.ihe.xds.soap.SOAPAttachment

### Constructor Summary

[SoapAttachment](#)(java.lang.String data, java.lang.String mimeType, java.lang.String contentId)



## Method Summary

java.lang.String	<a href="#">getContentId()</a>
java.lang.String	<a href="#">getData()</a>
java.lang.String	<a href="#">getMimeType()</a>
void	<a href="#">setContentId</a> (java.lang.String contentId)
void	<a href="#">setData</a> (java.lang.String data)
void	<a href="#">setMimeType</a> (java.lang.String mimeType)





---

## 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

#### 4.1.1 Description

The OHF XDS SOAP Client constructs the SOAP message for the XDS Consumer Query Registry Transaction (ITI-16).

#### 4.1.2 Code

```
// form the query string
String ebXMLQuery = null;
try {
    ebXMLQuery = formatEBXMLQueryString(sqlQueryString, returnType);
} catch (IOException e) {
    logger.fatal("Could not format ebXML for the query string.", e);
    throw e;
}
Element reply = this.sender.send(query, null, regURL);
if (reply == null) {
    throw new ConsumerException("No data returned from registry.");
}
return loadFromStream(new ByteArrayInputStream(reply.toString().getBytes()));
```

---

### 4.2 Example for Use Case 2 – Provide And Register Document Set

#### 4.2.1 Description

The OHF XDS SOAP Client constructs the SOAP message for the XDS Document Source Provide and Register Document Set Transaction (ITI-15).

#### 4.2.2 Code

```
// form the ebXML metada payload for the message
EbXML_2_1ProvideAndRegisterDocumentSetTransformer setTransformer = new
EbXML_2_1ProvideAndRegisterDocumentSetTransformer();
ProvideAndRegisterDocumentSetType meta = txnData.getMetadata();
```



```
setTransformer.transform(meta);

String ebXMLMetadataString =
unpackSubmitObjectsRequest(setTransformer.getSubmitReq());

// form the attachments
SoapAttachment[] attachments = new SoapAttachment[txnData.getDocList().size()];
Iterator i = txnData.getDocList().iterator();
int count = 0;
while(i.hasNext()){
    Document docI = (Document)i.next();
    attachments[count] = new SoapAttachment(docI.getDocumentData(),
        docI.getDescriptor().getMimeType(), docI.getDocumentEntryUUID());
    count++;
}

// send documents
int eventOutcome = ATNAAuditClient.SUCCESS_EVENT_OUTCOME;
Element reply = null;
try {
    logger.info("Submitting Provide and Register Document Set transaction to"
        + "Repository: " + repositoryURL);
    reply = this.sender.send(ebXMLMetadataString, attachments,
        this.repositoryURL);
} catch (Exception e) {
    eventOutcome = ATNAAuditClient.SERIOUS_FAILURE_EVENT_OUTCOME;
    logger.error("Error sending SOAP message to the repository. Error was " +
        e.getMessage(), e);
    if (isDoAudit()) {
        auditor.audit(eventOutcome, initiatingUser,
            this.formatPHIDData(txnData));
    }
}
```



---

## **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.*