



**Patient Identifier Cross-reference Consumer
Architecture & API Documentation
Version 0.1.2**

srenly@us.ibm.com | Sondra R Renly





Contents

1.	Introduction	5
2.	Getting Started.....	6
2.1	Platform Requirements.....	6
2.2	Source Files.....	6
2.3	Dependencies.....	6
2.3.1	Other OHF Plugins	6
2.3.2	External Sources	7
2.4	Resources	7
2.4.1	Other OHF plugin documentation	7
2.4.2	HL7 Standard 2.5.....	7
2.4.3	IHE ITI Technical Framework	7
2.4.4	Newsgroup.....	7
3.	API Documentation	8
3.1	Use Case - ITI-9 PIX Query	8
3.1.1	Flow of Execution	9
3.1.2	Source Code.....	10
4.	Security.....	13
4.1	Node Authentication	13
4.2	Auditing.....	13
5.	Configuration	14
6.	Debugging Recommendations.....	15
7.	IHE Connectathon MESA Tests.....	16
7.1	Plugin Testing.....	16
7.2	Bridge Testing	16





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 (Eclipse OHF) 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 documentation addresses the beta release of the Eclipse OHF plugin implementation of the IHE ITI Technical Framework actor Patient Identifier Cross-reference Consumer for the implementation of the ITI-9 PIX Query Transaction.



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 1.4.2

<http://java.sun.com/javase/downloads/index.jsp>

Note that in early 2007, all OHF components will be allowed to use Java 1.5.

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.common.hl7v2.client
- org.eclipse.ohf.ihe.pix.consumer

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

2.3 Dependencies

The Patient Identifier Cross-reference Consumer has dependencies on other OHF plugins and external sources.

2.3.1 Other OHF Plugins

Patient Identifier Cross-reference Consumer plugins are dependent on additional org.eclipse.ohf project plugins. You also need to check-out the following:

- org.eclipse.ohf.hl7v2.core
org.eclipse.ohf.utilities
org.apache.commons
org.apache.axis
org.xmlpull.v1
HL7v2 message object plugins and dependencies
- org.eclipse.ohf.ihe.common.mllp
Minimum Lower Level Protocol
- org.eclipse.ohf.ihe.atna.audit
org.eclipse.ohf.ihe.atna.agent
org.eclipse.ohf.ihe.common.atna
Auditing for messages sent and responses received
- org.apache.log4j
Debug, warning, and error logging



2.3.2 External Sources

For the purpose of message object creation and verification, either a licensed copy of the HL7 access database (hl7_58.mdb) or a free javaStream file (hl7Definitions.javaStream) is necessary. You can obtain the latest free javaStream file from:

org.eclipse.ohf.hl7v2.ui > Resources > hl7Definitions.javaStream

If you would like to obtain a copy of the HL7 access database file, refer to <http://www.hl7.org>.

To complete message verification, an XML conformance profile is necessary. A sample Q23 Get Corresponding IDs (HL7v2.5) conformance profile is included with this plugin in the /resources/conf folder.

2.4 Resources

The following resources are recommended.

2.4.1 Other OHF plugin documentation

The following OHF plugin documents are related to the Patient Identifier Cross-reference Consumer:

- OHF ATNA Audit Client (http://wiki.eclipse.org/index.php/OHF_IHE_Client_plugins)

2.4.2 HL7 Standard 2.5

The Patient Identifier Cross-reference Consumer references standards HL7 version 2.5.

<http://www.hl7.org>.

2.4.3 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.

2.4.4 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 Patient Identifier Cross-reference Consumer client supports three formats for input. The client will accept:

- a raw HL7 message (String)
- an HL7v2 message object (org.eclipse.ohf.hl7v2.core Message)
- a ITI-9 PIX Query message supporting the message construction of event:

QBP^Q23 – Get Corresponding Identifiers

Examples for the three types of inputs are found in the org.eclipse.ohf.ihe.pix.consumer plugin.

```
org.eclipse.ohf.ihe.pix.consumer > src_tests > org.eclipse.ohf.ihe.pix.consumer.tests > HL7PixQuery.java
org.eclipse.ohf.ihe.pix.consumer > src_tests > org.eclipse.ohf.ihe.pix.consumer.tests > MSGPixQuery.java
org.eclipse.ohf.ihe.pix.consumer > src_tests > org.eclipse.ohf.ihe.pix.consumer.tests > ClientPixQuery.java
```

The files in src_tests use a TestConfiguration.java file for extracting the various file locations and MLLP connection parameters. Update this file with your settings before running the sample code.

A raw HL7 message string should be used as input when the originating application is fully capable of sending and receiving HL7 messages. In this case, the Patient Identifier Cross-reference Consumer client is simply providing auditing, communication with the PIX server, and optional message verification. Server responses are returned to the caller as raw HL7v2 message strings. (HL7PixQuery)

A message object should be used as input when the originating application is directly using the OHF HL7v2 component which the Patient Identifier Cross-reference Consumer client sits on top of. In this case, the application has taken full responsibility for message creation and reading the response. The Patient Identifier Cross-reference Consumer client is simply providing conversion to raw HL7, auditing, communication with the PIX server, and optional message verification. Server responses are returned to the caller as HL7v2 message objects. (MSGPixQuery)

A ITI-9 PIX Query message should be used as input when the originating application has neither support for raw HL7 nor message objects. The Patient Identifier Cross-reference Consumer client provides a friendly interface to set and read message fields as well as auditing, communication with the PIX server, and optional message verification. Server responses are returned to the caller as PixConsumerResponse objects. (ClientPixQuery)

ITI-9 PIX Query Message Class

- PixConsumerQuery

ITI-9 PIX Query Server Response Class

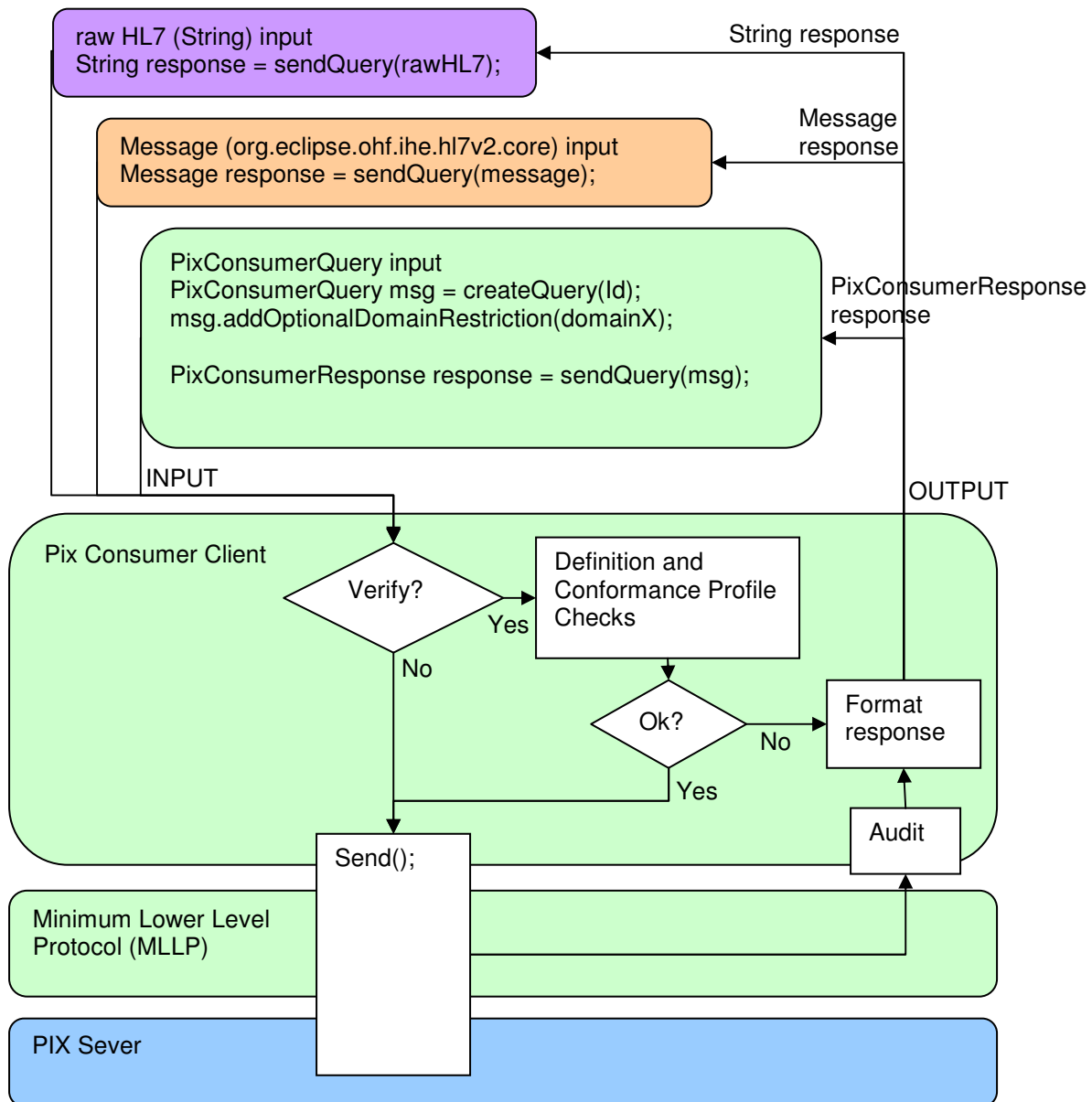
- PixConsumerResponse

3.1 Use Case - ITI-9 PIX Query

The Patient Identifier Cross-reference Consumer has one transaction. This use case demonstrates in step-by-step and with sample code the creation and use of the Patient Identifier Cross-reference Consumer Client, including the three input options. It includes example client construction of the event specific message object as input but not the creation of raw HL7 or HL7v2 Message object.



3.1.1 Flow of Execution



Create a Patient Identifier Cross-reference Consumer object:

1. Construct ITI-9 PIX Query
2. Construct and associate MLLP (Minimum Lower Level Protocol) Destination
3. Enable auditing.
4. Override the maximum level of validation error allowed before message submission is halted. The levels of error are constants in the OHF HL7v2 CPValidator.java file. The default is to allow up to the warning level.



Create a tailored HL7v2 message object:

1. Create Patient Identifier Cross-reference Consumer Message. Message field defaults are obtained first from the associated Conformance Profile. Required fields not found there default to settings for the IBM Test Dallas Server. (<http://ibmod235.dal-ebis.ihost.com:9080/IBMIHII/serverInfoOHF.htm>)
2. Change default settings.
3. Add optional field values.
4. As not all fields have a corresponding method, use the generic method to set these additional values. Use method `.setField(path, data)`.

The Patient Identifier Cross-reference Consumer supports populating data in MSH, QPD, and RCP segments. Information about the fields, components, and sub-components available in these segments is available in the HL7 Version 2.5 Standard document in Chapter 2 Section 2.15 Message Control Segments (MSH) and Chapter 5 Section 5.5 Query/Response Message Segments (QPD/RCP). Note that the ITI-9 PIX Query RCP segment is complete when the message is created and no further manipulation is required.

Send the message:

1. Send message

Read the response message:

1. Read response message fields.

3.1.2 Source Code

Create a Patient Identifier Cross-reference Consumer Query:

1. Construct ITI-9 PIX Query

There are three ways to construct the ITI-9 PIX Query client. If you have HL7 input and are not using message validation or auditing, no constructor parameters are necessary. Otherwise, you are required to provide the HL7 definition file (the access database file or javaStream file).

In this sample code, TConfig refers to the TestConfiguration.java file mentioned in the beginning of this section. See this file for example formatting of these constants.

```
//pixQuery set-up: no supporting files
pixQuery = new PixConsumer();

//pixQuery set-up: supporting access database & conformance profile
String msAccessFile = TConfig.ACCESS_DATABASE_PATH;
InputStream cpaStream = new FileInputStream(TConfig.CPROFILE_PATH);
pixQuery = new PixConsumer(msAccessFile, cpaStream);

//pixQuery set-up: javaStream setup example
Stream javaStream = new Stream(new File(TConfig.SERIALISED_PATH));
InputStream cpStream = new FileInputStream(TConfig.CPROFILE_PATH);
pixQuery = new PixConsumer(javaStream, cpStream);
```



2. Construct and associate MLLP (Minimum Lower Level Protocol) Destination

Un-Secure Connection:

```
MLLPDestination mllp = new MLLPDestination(TConfig.MLLP_URI);
pixQuery.setMLLPDestination(mllp);
```

Secure Connection:

```
MLLPDestination mllps = new MLLPDestination(TConfig.MLLPS_URI);
pixQuery.setMLLPDestination(mllps);
```

```
Properties props = new Properties();
props.setProperty(SecurityDomain.JAVAX_NET_SSL_KEYSTORE, "/x.jks");
props.setProperty(SecurityDomain.JAVAX_NET_SSL_KEYSTORE_PASSWORD, "pswd");
props.setProperty(SecurityDomain.JAVAX_NET_SSL_TRUSTSTORE, "/y.jks");
props.setProperty(SecurityDomain.JAVAX_NET_SSL_TRUSTSTORE_PASSWORD, "pswd");
SecurityDomain domain = new SecurityDomain("domainXY", props);
ConfigurationManager.registerDefaultSecurityDomain(domain);
```

3. Enable auditing.

```
//audit ON is the default
AtnaAgentFactory.getAtnaAgent().setDoAudit(false); //disable for all actors
```

4. Override the maximum level of validation error allowed before message submission is halted. The levels of error are constants in the OHF HL7v2 CPValidator.java file. The default is to allow up to the warning level.

```
ITEM_TYPE_INFORMATION = 1;
ITEM_TYPE_WARNING = 2; <<< Default
ITEM_TYPE_ERROR = 3;
ITEM_TYPE_FATAL = 4;

pixQuery.setMaxVerifyEvent(CPValidator.ITEM_TYPE_INFORMATION);
```

Create a tailored HL7v2 message object:

1. Create Patient Identifier Cross-reference Consumer Message.

It is highly recommended to use the Patient Identifier Cross-reference Consumer to create the query rather than use the PixConsumerQuery constructor as the underlying manager is taken care of. The patient ID parameter is the QPD-3-1 field and the assigningAuthority parameters are within the QPD-3-4 field.

```
PixConsumerQuery msg = pixQuery.createQuery(patientId);
or
PixConsumerQuery msg = pixQuery.createQuery(patientId,
                                             namespaceId, universalId, universalIdType);
```

2. Change default settings.

```
msg.changeDefaultCharacterSet("UNICODE");
```



3. Add optional query values.
You may add multiple domain restrictions by calling this method for each individual domain.

```
msg.addOptionalDomainRestriction("OHF", "", "");
```

4. As not all fields have a corresponding method, use the generic method to set these additional values.
Use method `.setField(alias, data)`.

```
msg.setField("MSH-12-1", version);
```

Note that this method does not handle repetitions, such as required by the Domain Restriction QPD-4-4 field. If the additional values require repeat support, use the methods exposed through the Eclipse OHF HL7v2 Message object. See the `PixConsumerQuery` `addOptionalDomainRestriction` method for sample code.

Send the message:

1. Send message

```
String response = pixQuery.sendQuery(msg, isValidateOn, auditUser);  
Message response = pixQuery.sendQuery(msg, isValidateOn, auditUser);  
PixConsumerResponse  
response = pixQuery.sendQuery(msg, isValidateOn, auditUser);
```

Read the response message:

1. Read response

```
//HL7v2 message object  
msg.getElement("MSA-1").getAsTableCode(); //message AckCode  
msg.getElement("QAK-2").getAsTableDescription(); //message QueryStatus  
  
//PixConsumerResponse object  
response.getResponseAckCode(isExpandCodeToString); //message AckCode  
response.getQueryStatus(isExpandCodeToString); //message QueryStatus
```



4. Security

4.1 Node Authentication

Transport Layer Security Protocol (TLS) is supported by creating a secure MLLP connection. Information required to instantiate a secure connection to one of the IBM Dallas Servers is available in the sample code configuration file. For more information and use terms on the IBM Dallas Servers, see

<http://ibmod235.dal-ebis.ihost.com:9080/IBMIHII/serverInfoOHF.htm>

http://lswin10.dfw.ibm.com:9080/IBMIHII/serverInfoIHE_ConnectathonHIMSS2007.htm

4.2 Auditing

Auditing to an Audit Record Repository is automatically enabled through the ATNA Agent. The Patient Identifier Cross-reference Consumer automatically generates the following audit messages:

- EventID 110100 - Actor Start audit message (EventTypeCode 110120)
- EventID 110112 - Query audit message
- EventID 110107 - Import audit message
- EventID 110100 - Actor Stop audit message (EventTypeCode 110121)

Auditing is no longer enabled/disabled with the doAudit boolean variable within the Patient Identifier Cross-reference Consumer client. To disable all auditing, use the ATNA Agent.

```
AtnaAgentFactory.getAtnaAgent().setDoAudit(false);
```



5. Configuration

There are two types of configuration in this release.

Create message default field values, such as message header fields, can be read from the conformance profile field ConstantValue attribute. Overloading the string with component or sub-component delimited data is not supported.

```
<Field Name="MSH-3 Sending Application" ... ConstantValue="OTHER_KIOSK">
<Field Name="MSH-4 Sending Facility" ... ConstantValue="HIMSSSANDIEGO">

<Field Name="QPD-3 Person Identifier" ...
  <Component Name="QPD-3-4 ...
    <SubComponent Name="QPD-3-4-1 namespace ID"... ConstantValue="XREF2005">
      <SubComponent Name="QPD-3-4-2 person universal ID" ...
        ConstantValue="1.3.6.1.4.1.21367.2005.1.2">
          <SubComponent Name="QPD-3-4-3 universal ID type"... ConstantValue="ISO">
        </SubComponent>
      </Component>
    </Field>
```

The sample code files in src_tests use a TestConfiguration.java file for extracting the various file locations and MLLP connection parameters. Update the default values in this file as needed with your settings before running the code. Here are the fields that are configured in this file:

```
//basics
public static final String DATA_PATH
public static final String LOG4J_PATH

//HL7PixQuery - run from file
public static final String HL7FILE_PATH

//Enable examples using the Access DB File
//This file is licensed through HL7 and not provided as part of this
//plugin
public static final boolean HAS_ACCESSFILE
public static final String ACCESS_DATABASE_PATH

//Enable examples using the free javaStream file
public static final boolean HAS_JAVASTREAM
public static final String SERIALISED_PATH

//Conformance profile for second level HL7 verification
public static final String CPROFILE_PATH

//MLLP Connectivity:
//Default IBM Dallas IHII Server - more connection info available at:
//http://ibmod235.dal-ebis.ihost.com:9080/IBMIHII/serverInfoOHF.htm
//http://lswin10.dfw.ibm.com:9080/IHMIHII/serverInfoIHE_ConnectathonHIMSS2
007.htm
public static URI MLLP_URI
public static URI MLLPS_URI
```



6. Debugging Recommendations

Log statements have been entered throughout the Patient Identifier Cross-reference Consumer plugin source code for assistance in monitoring the progress of the running client. To enable logging, there is a Log4j configuration file.

An example log4j configuration file is in the file `/resources/conf/pixconsumer_log4j.xml`. The default configuration creates the log file in the folder `/resources/log`.



7. IHE Connectathon MESA Tests

For current information, please refer to the wiki:

http://wiki.eclipse.org/index.php/IHE_Connectathon_2007#Detailed_MESA_Test_Info

7.1 Plugin Testing

The OHF Patient Identifier Cross-reference Consumer plugin completed testing with the following junits and test scripts.

```
org.eclipse.ohf.ihe.pix.consumer > src_tests > org.eclipse.ohf.ihe.pix.consumer.tests.mesa >
MESA10501Test.java           Pix Query Case 1 and Case 2
MESA10502Test.java           Pix Query Case 3
MESA10503Test.java           Pix Query Case 4
MESA2007.txt                  Sample script for starting the mesa server and running all 3 tests.
```

7.2 Bridge Testing

The OHF Bridge completed testing with the following junits and test scripts.

```
org.eclipse.ohf.bridge.ws > src_tests > org.eclipse.ohf.bridge.ws.tests.mesa >
TestMESA10501a.java          Pix Query Case 1 and Case 2
TestMESA10501b.java          Pix Query Case 1 and Case 2
TestMESA10502.java           Pix Query Case 3
TestMESA10503.java           Pix Query Case 4
```

```
org.eclipse.ohf.ihe.pix.consumer > src_tests > org.eclipse.ohf.ihe.pix.consumer.tests.mesa >
MESA2007.txt                  Sample script for starting the mesa server and running all 3 tests.
```