Patient Demographics Consumer Architecture & API Documentation Version 0.2.0

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 (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 document corresponds to the Eclipse OHF plugin implementation of the IHE ITI Technical Framework actor Patient Demographics Consumer for the implementation of the ITI-21 Patient Demographics Query Transaction that was used successfully by several OHF adopters at the 2007 IHE Connectathon.

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 or higher	http://www.eclipse.org/downloads/
Java JDK 1.4.2 or higher	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.pdq.consumer

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

2.3 Dependencies

The Patient Demographics Consumer has dependencies on both other OHF plugins and external sources.

2.3.1 Other OHF Plugins

Patient Demographics 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

Message defaults and message restrictions can be specified in an optional XML conformance profile. A sample Q22 Find Candidates (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 Demographics Consumer:

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

2.4.2 HL7 Standard 2.5

The Patient Demographics 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 Demographics Consumer client supports three formats for input. The client will accept:

- a raw HL7 message (String)
- an HL7v2 message object (org.eclipse.ohf.ihe.hl7v2.core Message)
- an ITI-21 Patient Demographics Query message supporting the construction of event:

  QBP^Q22 Patient Demographics Query

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

org.eclipse.ohf.ihe.pdq.consumer > src_tests > org.eclipse.ohf.ihe.pdq.consumer.tests > HL7PdqQuery.java
org.eclipse.ohf.ihe.pdq.consumer > src_tests > org.eclipse.ohf.ihe.pdq.consumer.tests > MSGPdqQuery.java
org.eclipse.ohf.ihe.pdq.consumer > src_tests > org.eclipse.ohf.ihe.pdq.consumer.tests > ClientPdqQuery.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 Demographics 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. (HL7PdqQuery)

A message object should be used as input when the originating application is directly using the OHF HL7v2 component which the Patient Demographics Consumer client sits on top of. In this case, the application has taken full responsibility for message creation and reading the response. The Patient Demographics 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. (MSGPdqQuery)

A ITI-21 Patient Demographics Query message should be used as input when the originating application has neither support for raw HL7 nor message objects. The Patient Demographics 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 messages are returned to the caller as PdqConsumerResponse objects. (ClientPdqQuery)

ITI-21 Patient Demographics Query Message Class

- PdqConsumerQuery

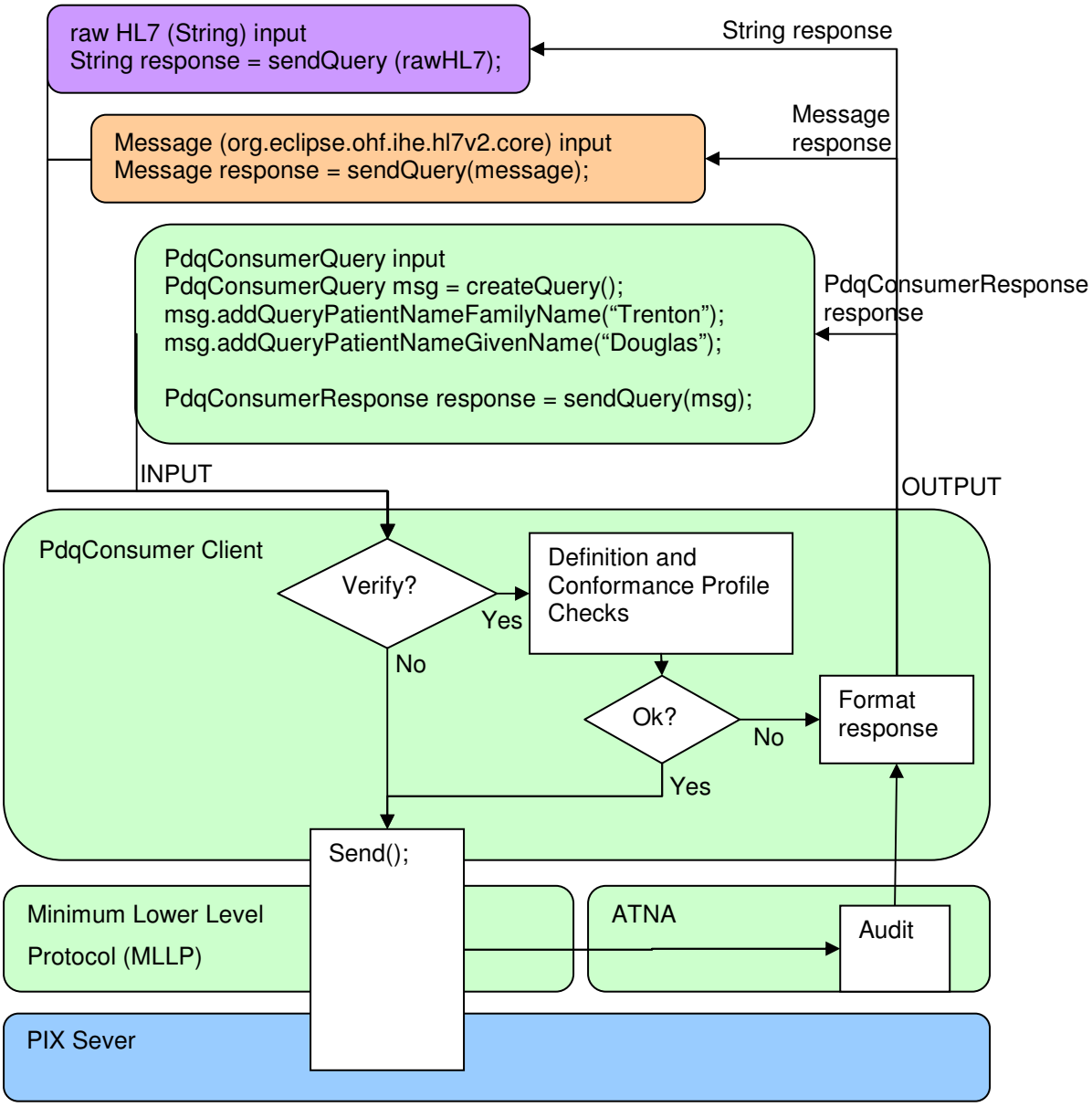
ITI-21 Patient Demographics Query Server Response Class

- PdqConsumerResponse

3.1 Use Case - ITI-21 Patient Demographics Query

The Patient Demographics Consumer has one transaction. This use case demonstrates in step-by-step and with sample code the creation and use of the Patient Demographics 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 an HL7v2 Message object.

3.1.1 Flow of Execution



Create a Patient Demographics Consumer object:

1. Construct ITI-21 Patient Demographics Query
2. Construct and associate MLLP (Minimum Lower Level Protocol) Destination
3. Disable auditing if desired.
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 Demographics Consumer Message. Message field defaults are obtained first from the associated Conformance Profile. Required fields not found there default to settings for the IBM Dallas Server. (http://ibmod235.dal-ebis.ihost.com:9080/IBMIHII/serverInfoOHF.htm)
2. Change default settings.
3. Add optional query values.
4. As not all fields have a corresponding method, use the generic method to set these additional values. Use method .addOptionalDemographicSearch(alias, data).
5. Add optional domain restrictions.
6. Add optional number of patients to return restriction.

The Patient Demographics Consumer supports querying data from PID and PD1 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 3 Section 3.4 Message Segments.

Send the message:

1. Send message

Read the response message:

1. Read response message fields.

3.1.2 Sample Code

Create a Patient Demographics Consumer Query:

1. Construct ITI-21 Patient Demographics Query

   There are two ways to construct the ITI-21 Patient Demographics Query client. At client creation, HL7 definitions are now automatically loaded for you from the HL7v2 toolkit. You can optionally provide an XML conformance profile for providing message defaults and additional message validation restrictions. The conformance profile can be added at a later time as well.

   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.

```
//pdqQuery set-up
```

```
PdqConsumer pdqQuery = new PdqConsumer();

//pdqQuery set-up with conformance profile
InputStream cpStream = new FileInputStream(TConfig.CPROFILE_PATH);
pdqQuery = new PdqConsumer(cpStream);
cpStream.close();
```

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

   Un-Secure Connection:

```
MLLPDestination mllp = new MLLPDestination(TConfig.MLLP_URI);
MLLPDestination.setUseATNA(true);
pdqQuery.setMLLPDestination(mllp);
```

   Secure Connection:

```
MLLPDestination mllps = new MLLPDestination(TConfig.MLLPS_URI);
MLLPDestination.setUseATNA(true);
pdqQuery.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. Disable auditing if desired.

```
//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;
ITEM_TYPE_ERROR = 3;
ITEM_TYPE_FATAL = 4;

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

Create a tailored HL7v2 message object:

1. Create Patient Demographics Consumer Message.

```
PdqConsumerQuery msg = pdqQuery.createQuery();
msg.addQueryPatientNameFamilyName("TRENTON");
msg.addQueryPatientNameGivenName("DOUGLAS");
```

2. Change default settings.

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

3. Add optional query values.

```
msg.addOptionalQuerySex("F");
```

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

```
msg.addOptionalQuerySex("F");
msg.addOptionalDemographicSearch("PID-8", "F");
```

   For this example, the two statements are identical to show that the methods are equivalent.

5. Add optional domain restrictions.

```
msg.changeDefaultWhatDomainsReturned("HIMSS2005", "", "");
```

6. Add optional number of patients to return restriction. By default this value is set to 10 in an attempt to ensure the audit message does not exceed the maximum size limit.

```
msg.addOptionalQuantityLimit(5);
```

Send the message:

1. Send message

```
String response = pdqQuery.sendQuery(msg, isValidateOn, auditUser);
Message response = ppdQuery.sendQuery(msg, isValidateOn, auditUser);
PdqConsumerResponse response =
pdqQuery.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

//PdqConsumerResponse object
response.getResponseAckCode(isExpandCodeToString);
```

```
response.getQueryStatus(isExpandCodeToString);
response.getPatientCount();
for (int i=1; i <= response.getPatientCount(); i++) {
      String patID[] = response.getPatientIdentifier(i, 0);
      String patName[] = response.getPatientName(i);
      logger.debug("  Patient: " + patID[0]
      + "-" + response.getPatientNameFamilyName(i)
      + ", " + patName[1]);
}
```

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 Demographics 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 Demographics 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.

5.1 Conformance Profile

Create message default field values, such as message header fields, can now be read from the conformance profile field ConstantValue attribute. This is now supported at all levels: field, component, and sub-component.

Field example:

```
<Field Name="MSH-1 Field Separator" Usage="R" Min="1" Max="1" Datatype="ST"
Length="1" ItemNo="00001" ConstantValue="|"></Field>
```

Component example (in this case only the namespaceId is defaulted):

```
<Field Name="MSH-3 Sending Application" Usage="R" Min="0" Max="1" Datatype="HD"
Length="227" Table="0361" ItemNo="00003">

      <Component Name="MSH-3-1 sending application: namespace ID" Usage="R"
      Datatype="IS" ConstantValue="OHFConsumer1"></Component>

      <Component Name="MSH-3-2 sending application: universal ID" Usage="O"
      Datatype="ST"></Component>

      <Component Name="MSH-3-3 sending application: universal ID type" Usage="O"
      Datatype="ID"></Component>

</Field>
```

Sub-component example (specifies a limit of 5 records):

```
<Field Name="RCP-2 Quantity Limited Request" Usage="O" Min="0" Max="1"
Datatype="CQ" Length="10" Table="0126" ItemNo="00031">

      <Component Name="RCP-2-1 quantity limited request: quantity" Usage="O"
      Datatype="NM" ConstantValue="5"></Component>

      <Component Name="RCP-2-2 quantity limited request: units" Usage="O"
      Datatype="CE">

            <SubComponent Name="RCP-2-2-1 quantity limited request: units:
            identifer" Usage="O" Datatype="ST"
            ConstantValue="RD"></SubComponent>

            <SubComponent Name="RCP-2-2-2 quantity limited request: units: text"
            Usage="O" Datatype="ST"></SubComponent>

            <SubComponent Name="RCP-2-2-3 quantity limited request: units: name
            of coding system" Usage="O" Datatype="ID"></SubComponent>

            <SubComponent Name="RCP-2-2-4 quantity limited request: units:
            alternate identifier" Usage="O" Datatype="ST"></SubComponent>

            <SubComponent Name="RCP-2-2-5 quantity limited request: units:
            alternate text" Usage="O" Datatype="ST"></SubComponent>

      </Component>

</Field>
```

5.2 Test Configuration

The files in src_tests now 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. 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

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

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

//TLS: Secure connection parameters
public static final String MLLP_KEYSTORE_NAME
public static final String MLLP_KEYSTORE_PASSWORD
public static final String MLLP_TRUSTSTORE_NAME
public static final String MLLP_TRUSTSTORE_PASSWORD
```

6. Debugging Recommendations

Log statements have been entered throughout the Patient Demographics 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/pdqconsumer_log4j.xml. The default configuration writes to a 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 Demographics Consumer plugin completed testing with the following junits and test scripts.

org.eclipse.ohf.ihe.pdq.consumer > src_tests > org.eclipse.ohf.ihe.pdq.consumer.test.mesa >

MESA2007Tests.java	Test11311 – PDQ Exact Name Search
	Test11312 – PDQ Name Search no Match
	Test11315 – PDQ Partial Name Search
	Test11320 – PDQ Complete ID Search Unspecified Domain
	Test11325 – PDQ Complete ID Search Single Domain
	Test11330 – PDQ Complete ID Search Multiple Domains
	Test11335 – PDQ Partial ID Search Single Domain
	Test11350 – PDQ Multi-Key Search 1
	Test11365 – PDQ Continuation Test 1
MESA2007.txt	Sample script for starting the mesa server and running all 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 >

TestMESA11311.java	PDQ Exact Name Search
TestMESA11312.java	PDQ Name Search no Match
TestMESA11315.java	PDQ Partial Name Search
TestMESA11320.java	PDQ Complete ID Search Unspecified Domain
TestMESA11325.java	PDQ Complete ID Search Single Domain
TestMESA11335.java	PDQ Partial ID Search Single Domain
TestMESA11350.java	PDQ Multi-Key Search 1
TestMESA11365.java	PDQ Continuation Test 1

org.eclipse.ohf.ihe.pdq.consumer > src_tests > org.eclipse.ohf.ihe.pdq.consumer.test.mesa >

MESA2007.txt	Sample script for starting the mesa server and running all tests.

8. IHE Connectathon Tests

8.1 Plugin Testing

The OHF Patient Demographics Consumer plugin completed testing with the following junits.

org.eclipse.ohf.ihe.pdq.consumer > src_tests > org.eclipse.ohf.ihe.pdq.consumer.test.connectathon >

ConnectathonPDQLoadTest.java	2.1 PDQ_Load The initial patient load for PDQ requires use of the Patient Identity Source plugin. This file is a placeholder for the actual junit which is located within the Patient Identity Source connectathon junits - org.eclipse.ohf.ihe.pix.source > src_tests > org.eclipse.ohf.ihe.pix.source.test.connectathon
ConnectathonPDQTest.java	2.2 PDQ_Exact_Name 2.3 PDQ_Multiple_Query 2.5 PDQ_Continuation_Test