

Oracle® Healthcare Master Person Index

Working With IHE Profiles User's Guide

Release 3.0

E62314-03

July 2016

E62314-03

Copyright © 2011, 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Finding Information and Patches on My Oracle Support	viii
Finding Oracle Documentation	x
Conventions	x
1 Overview of the IHE Profiles Application	
Introducing the IHE Profiles Application	1-1
Understanding the Layers of the IHE Profiles Application	1-1
IHE Profiles Application Integration Layer	1-2
Native HL7 Message Handling	1-2
Master Person Index Service Layer	1-3
Master Person Index Core	1-3
Infrastructure Services	1-3
Linking to IHE Profiles Application Tasks	1-3
For Running Test Data	1-3
For a Production Environment	1-3
Additional Steps to Set Up the IHE Profiles Application	1-4
2 Creating an IHE Profiles Application Project	
Creating an IHE Profiles Application Project	2-1
To Create an IHE Profiles Application Project	2-2
Configuring the Master Person Index Project	2-2
Using the IHE Project Actions Menu	2-3
Configuring an IHE Profiles Application Project	2-3
To Configure hl7message.xml	2-4
To Configure hl7service.xml	2-6
To Configure Secure PIXv2 Manager and PDQv2 Supplier	2-8
To Configure services.properties	2-9
Building an IHE Profiles Application Project	2-13
To Build an IHE Profiles Application Project	2-13

3 Creating MPI and IHE Databases and Tables

Creating the IHE Profiles Application Databases and Tables	3-1
Creating Master Person Index Databases and Tables for Oracle Manually	3-1
To Create the Master Person Index Database	3-1
To Create the IHE Profiles Application Database Tables	3-2
Creating the Database and Tables Automatically	3-3
Maintaining Subscriptions to Patient Updates	3-3

4 Getting Started With Oracle WebLogic Server

Creating an Instance of an Oracle WebLogic Server	4-1
Related Documentation	4-1
Oracle WebLogic Server Requirements	4-1
Creating a Project and Starting the OHMPI Wizard	4-1
To Create a Project and Start the OHMPI Wizard	4-2
Adding an Oracle WebLogic Server	4-2
To Add an Oracle WebLogic Server	4-2
Creating, Configuring, and Building a Master Person Index Application	4-3

5 Configuring an Application Server

Configuring the Oracle WebLogic Server	5-1
Configuring the HL7 v2 Server	5-1
Creating JDBC Data Resources for an IHE Profile Application Project	5-2
To Create JDBC Data Resources for an IHE Profiles Application Project for Oracle	5-2
Creating JMS Resources for an IHE Profile Application Project	5-3
To Create JMS Server	5-4
To Create JMS Module	5-4
To Create JMS Connection Factory	5-5
To Create JMS Topic	5-5
To Create PixUpdateNotificationTopic	5-6
Setting Up the User	5-6

6 Configuring and Using Audit Record Repository

Installing the Audit Record Repository	6-2
Running the Audit Record Repository Server	6-2
Configuring the Audit Client	6-2

7 Deploying the IHE Profiles Application

Deploying an IHE Profiles Application	7-1
To Deploy an EAR File from an Admin Console	7-1
Running Data Through the IHE Profiles Application	7-2
To Add Required Systems to the Database	7-2
To Run HL7 V2 Sample Data	7-3
To Run HL7 V3 Sample Data	7-3

8 Monitoring and Maintaining Master Person Index Data

Using the Master Index Data Manager to Monitor and Maintain Master Person Index Data..	8-1
Logging in to the Master Index Data Manager	8-1
To Log in to the MIDM	8-2

A HL7 Message Types Supported by IHE Within OHMPI

Supported HL7 v2.3.1, v2.5, and v3 Message Types in OHMPI	A-1
---	-----

B Adding IHE Profile Support to User-defined MPI Object Model

Creating IHE Project with User-defined MPI Project	B-1
Implementing Customized Mapper Project.....	B-2
Mapper Project Overview	B-2
Mapper Example	B-2
Value Mapping	B-5
Required Fields.....	B-5
Mapping Context	B-6
Code System Support	B-6
Configuring IHE Project	B-7

Preface

This guide introduces users to Integrating the Healthcare Enterprise (IHE) and the Oracle Healthcare Master Person Index (OHMPI) IHE Profiles Application. IHE standards and profiles (PIX, PDQ, PAM, and ARR) help create, process, and manage electronic health records in secure patient cross-reference applications. They work in conjunction with native Health Level 7 (HL7) v2 and v3 messaging and transport standards which define how information is packaged and shared between systems.

This guide provides instructions on how to configure PIX, PDQ, PAM, and ARR components, create databases and runtime components, how to build and deploy an IHE Profiles Application project, and how to prepare the IHE Profiles Application to run data through it.

Audience

This guide is intended for users of the OHMPI that want to create IHE Profiles Application projects.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information and instructions for implementing and using a Master Person Index application, see the following documents in the Oracle Healthcare Master Person Index documentation set:

- *Oracle Healthcare Master Person Index Working With IHE Profiles User's Guide* [This document]
- *Oracle Healthcare Master Person Index Analyzing and Cleansing Data User's Guide*
- *Oracle Healthcare Master Person Index Australia Patient Solution User's Guide*

- *Oracle Healthcare Master Person Index Command Line Reports and Database Management User's Guide*
- *Oracle Healthcare Master Person Index Configuration Guide*
- *Oracle Healthcare Master Person Index Configuration Reference*
- *Oracle Healthcare Master Person Index Data Manager User's Guide*
- *Oracle Healthcare Master Person Index Installation Guide*
- *Oracle Healthcare Master Person Index Loading the Initial Data Set User's Guide*
- *Oracle Healthcare Master Person Index Match Engine Reference*
- *Oracle Healthcare Master Person Index Message Processing Reference*
- *Oracle Healthcare Master Person Index Provider Index User's Guide*
- *Oracle Healthcare Master Person Index Real-time Loader User's Guide*
- *Oracle Healthcare Master Person Index Release Notes*
- *Oracle Healthcare Master Person Index Security Guide*
- *Oracle Healthcare Master Person Index Standardization Engine Reference*
- *Oracle Healthcare Master Person Index United Kingdom Patient Solution User's Guide*
- *Oracle Healthcare Master Person Index United States Patient Solution User's Guide*
- *Oracle Healthcare Master Person Index User's Guide*
- *Oracle Healthcare Master Person Index Working With HPD Profile Application User's Guide*

Note: These documents are designed to be used together when implementing a master index application.

Finding Information and Patches on My Oracle Support

Your source for the latest information about Oracle Healthcare Master Person Index is Oracle Support's self-service website My Oracle Support (formerly MetaLink).

Before you install and use Oracle Healthcare Master Person Index, always visit the My Oracle Support website for the latest information, including alerts, White Papers, installation verification (smoke) tests, bulletins, and patches.

Creating a My Oracle Support Account

You must register at My Oracle Support to obtain a user name and password account before you can enter the website.

To register for My Oracle Support:

1. Open a web browser to <https://support.oracle.com>.
2. Click the **Register here** link to create a My Oracle Support account. The registration page appears.
3. Follow the instructions on the registration page.

Signing In to My Oracle Support

To sign in to My Oracle Support:

1. Open a web browser to <https://support.oracle.com>.
2. Click **Sign In**.
3. Enter your user name and password.
4. Click **Go** to open the My Oracle Support home page.

Finding Information on My Oracle Support

There are many ways to find information on My Oracle Support.

Searching by Article ID

The fastest way to search for information, including alerts, White Papers, installation verification (smoke) tests, and bulletins is by the article ID number, if you know it.

To search by article ID:

1. Sign in to My Oracle Support at <https://support.oracle.com>.
2. Locate the Search box in the upper right corner of the My Oracle Support page.
3. Click the sources icon to the left of the search box, and then select **Article ID** from the list.
4. Enter the article ID number in the text box.
5. Click the magnifying glass icon to the right of the search box (or press the Enter key) to execute your search.

The Knowledge page displays the results of your search. If the article is found, click the link to view the abstract, text, attachments, and related products.

Searching by Product and Topic

You can use the following My Oracle Support tools to browse and search the knowledge base:

- **Product Focus** — On the Knowledge page under Select Product, type part of the product name and the system immediately filters the product list by the letters you have typed. (You do not need to type "Oracle.") Select the product you want from the filtered list and then use other search or browse tools to find the information you need.
- **Advanced Search** — You can specify one or more search criteria, such as source, exact phrase, and related product, to find information. This option is available from the **Advanced** link on almost all pages.

Finding Patches on My Oracle Support

Be sure to check My Oracle Support for the latest patches, if any, for your product. You can search for patches by patch ID or number, or by product or family.

To locate and download a patch:

1. Sign in to My Oracle Support at <https://support.oracle.com>.
2. Click the **Patches & Updates** tab.

The Patches & Updates page appears and displays the Patch Search region. You have the following options:

- In the **Patch ID or Number** field, enter the number of the patch you want. (This number is the same as the primary bug number fixed by the patch.) This option is useful if you already know the patch number.

- To find a patch by product name, release, and platform, click the **Product or Family** link to enter one or more search criteria.
- 3. Click **Search** to execute your query.
The Patch Search Results page appears.
- 4. Click the patch ID number. The system displays details about the patch. In addition, you can view the Read Me file before downloading the patch.
- 5. Click **Download**. Follow the instructions on the screen to download, save, and install the patch files.

Finding Oracle Documentation

The Oracle website contains links to all Oracle user and reference documentation. You can view or download a single document or an entire product library.

Finding Oracle Health Sciences Documentation

To get user documentation for Oracle Health Sciences applications, go to the Oracle Health Sciences Documentation page at:

<http://www.oracle.com/technetwork/documentation/hsgbu-154445.html>

Note: Always check the Oracle Health Sciences Documentation page to ensure that you have the latest updates to the documentation.

Finding Other Oracle Documentation

To get user documentation for other Oracle products:

1. Go to the following web page:
<http://www.oracle.com/technology/documentation/index.html>
Alternatively, you can go to <http://www.oracle.com>, point to the Support tab, and then click **Documentation**.
2. Scroll to the product you need and click the link.
3. Click the link for the documentation you need.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Overview of the IHE Profiles Application

This chapter provides an overview of Oracle Healthcare Master Person Index (OHMPI) and the Integrating the Healthcare Enterprise (IHE) Profiles Application. It also provides links to instructions in the following chapters that help you understand and use OHMPI and the IHE Profiles Application.

This chapter includes the following sections:

- [Introducing the IHE Profiles Application](#) on page 1-1
- [Linking to IHE Profiles Application Tasks](#) on page 1-3

Introducing the IHE Profiles Application

OHMPI's IHE Profiles Application provides a flexible solution to healthcare integration needs. It uses processing logic based on the guidelines and standards put forth by IHE to assure compatibility with other vendors and healthcare organizations.

The IHE Profiles Application forms a specialized implementation of HL7 messaging that facilitates patient health information exchange, and supports both native HL7 v2 and HL7 v3 messaging. HL7 v2 is supported through the HL7 v2 server over the Minimal Lower Layer Protocol (MLLPV1 or MLLPV2). HL7 v3 is supported through SOAP 1.2 over HTTP.

The IHE Profiles Application leverages the advanced standardization and matching algorithms of Master Person Index to cross-reference and uniquely identify the patients in your healthcare organization. Master Person Index provides a single complete view of the participants in your healthcare system and is able to quickly reconcile which information is associated with which patient.

The IHE Profiles Application processes messages based on the IHE IT infrastructure technical frameworks, which define how to process HL7 messages using existing standards when available. In compliance with these frameworks, the IHE Profiles Application generates and maintains an audit repository of all events processed by the manager. Information about the state of the IHE Profiles Application components is provided by a common logging, alerting, error handling, and reporting mechanism.

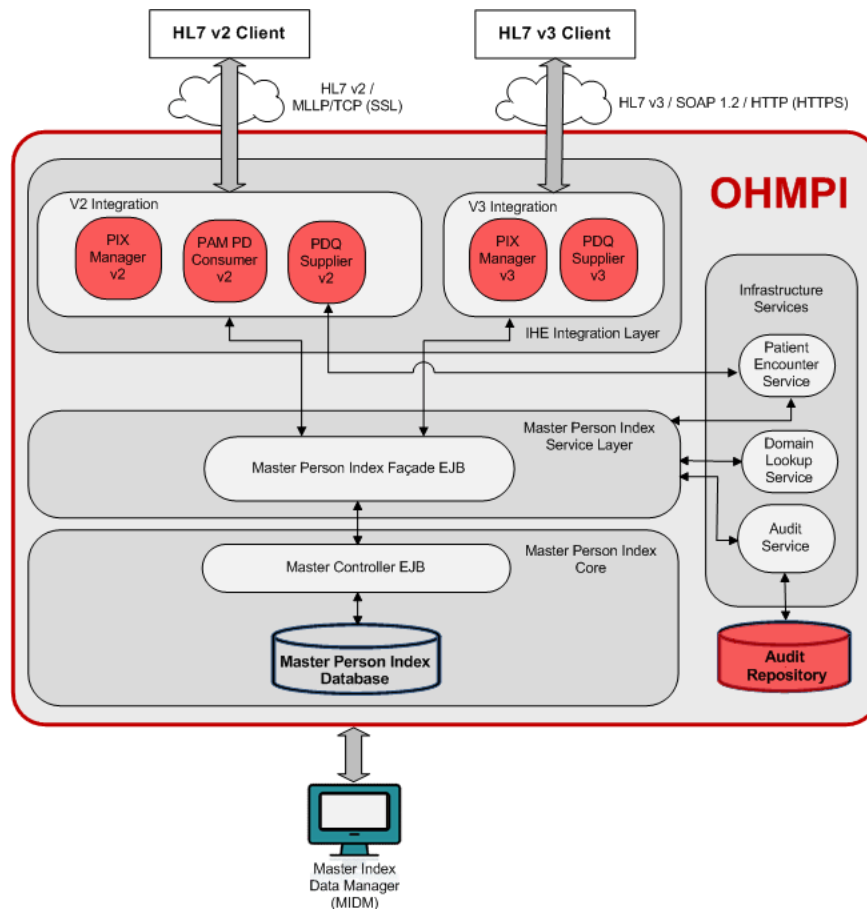
Understanding the Layers of the IHE Profiles Application

The IHE Profiles Application has a number of layers that function inside of OHMPI, which are illustrated in [Figure 1-1](#), and include:

- [IHE Profiles Application Integration Layer](#) on page 1-2
- [Master Person Index Service Layer](#) on page 1-3
- [Master Person Index Core](#) on page 1-3

- [Infrastructure Services](#) on page 1-3

Figure 1–1 HL7 v2 and v3 Integration with OHMPI



IHE Profiles Application Integration Layer

The IHE Integration layer adapts HL7 v2 and HL7 v3 messages to the simplified Java data model and service layer exceptions and vice versa. It also ensures that the message header properties (such as sending facility, receiving facility) are populated according to the specification.

The V3 Integration exposes OHMPI’s functionality through SOAP version 1.2 over HTTP/HTTPS to IHE PIX/PDQ HL7 v3 clients. Its interface is defined by the IHE specification and the XSD/WSDL artifacts are provided by IHE.

The V2 Integration exposes OHMPI’s functionality through TCP/IP-MLLP and HL7v2 based protocol to HL7v2 clients.

Native HL7 Message Handling The IHE Integration layer is HL7 v2 and v3 message-aware; that is, it can receive and validate incoming HL7 messages and handle standards-defined acknowledgements. HL7 v2 is supported through an HL7 v2 server that uses secured TCP/IP and MLLP. HL7 v3 is supported through SOAP messages transported over HTTP/HTTPS. Support for IHE Profiles in OHMPI is designed with the aim to implement the solution with minimal configuration level changes. This layer works seamlessly in tandem with the Master Person Index layer to provide the appropriate patient identity resolution and demographic information responses.

Master Person Index Service Layer

The Master Person Index Service Layer consists of the core set of services mapping PIX/PDQ/PAM operations and data into OHMPI. The interface to this layer is IHE aligned. This means that the data structures are simplified versions of the HL7 message formats. The operations are analogous to the operations defined in IHE.

This layer also implements the audit requirements and domain resolution (namespace ID and universal ID mapping) service.

Note: There is no protocol specific functionality in this layer. It implements the response behavior of PIX/PDQ/PAM from a functional point of view.

Master Person Index Core

This layer is the core patient identity resolution component that leverages OHMPI's powerful matching and standardization engines to uniquely identify and cross-reference patient objects based on a canonical patient object model. OHMPI's powerful, extensible, and configurable Match Engine compares complex data records containing a multitude of datatypes and calculates a global composite weight that reflects how closely the records match. From this, OHMPI builds a cross-index that is used to provide a single view of the patient information from multiple source systems in real time.

Infrastructure Services

As related to OHMPI and the IHE Profiles Application, the Audit Record Repository (ARR) is an audit server and repository which functions with a Domain Lookup Service that provides cross-referencing to identify a domain by domain ID (also known as namespace ID) and/or universal ID, which in turn provides secure transmission, auditing, logging, alerting, error handling, and reporting for healthcare application systems. A supporting component is an audit service that generates audit messages and implements an IHE audit trail that is defined in the Audit Trail and Node Authentication (ATNA) Integration Profile.

Linking to IHE Profiles Application Tasks

This document includes several tasks that are optional and some that are required for a production environment. The tasks are listed by category as follows.

For Running Test Data

If you have performed a complete installation of OHMPI, you are ready to work with the IHE Profiles Application and can start running test data and using the Master Index Data Manager (MIDM). See [Running Data Through the IHE Profiles Application](#) on page 7-2 for instructions on running test data and monitoring the processed data.

For a Production Environment

For a production environment, you need to configure certain variables unless you specified the correct values during installation, and each domain needs to be added to the IHE Profiles Application. Optionally, you can configure Master Person Index processing.

Additional Steps to Set Up the IHE Profiles Application

You should perform the following additional steps before you begin using the IHE Profiles Application:

- [Creating the IHE Profiles Application Databases and Tables](#) on page 3-1
- Configure application server resources. For information, see [Configuring the Oracle WebLogic Server](#) on page 5-1.

Creating an IHE Profiles Application Project

This chapter provides procedures on how to create, configure, and build an IHE Profiles Application project. It also discusses the features of an IHE Profiles Application project Actions menu.

This chapter includes the following sections:

- [Creating an IHE Profiles Application Project](#) on page 2-1
- [Configuring the Master Person Index Project](#) on page 2-2
- [Using the IHE Project Actions Menu](#) on page 2-3
- [Configuring an IHE Profiles Application Project](#) on page 2-3
- [Building an IHE Profiles Application Project](#) on page 2-13

Creating an IHE Profiles Application Project

IHE design-time is a NetBeans IDE plug-in. The IHE wizard takes your input, such as database type, application server, and the V2 Service Port, and creates an IHE Profiles Application project and pre-configured Master Person Index (MPI) project. You are able to fine-tune the matching service of the Master Person Index, and configure V2 Service Port of the IHE project. When the IHE project builds, the database scripts and three artifacts are generated. The IHE Create Database dialog box lets you create a database (if needed). Once the database is created you can deploy the IHE ear to the target server using the application server administration console. The steps to create the IHE Profiles Application project and deploy it are:

- Create an IHE Profiles Application project
- Configure the IHE Profiles Application project
- Build the IHE Profiles Application project
- Run the database script
- Configure the application server
- Deploy the IHE Profiles Application project on an application server

The IHE wizard requires the following input to create an IHE Profiles Application project and pre-configured MPI project. The wizard also sets the database type and application server information for both the IHE project and the pre-configured MPI project.

- Project name
- Project location

- Application server
- Database
- V2 Service Port

To Create an IHE Profiles Application Project

1. From the NetBeans IDE, click **File** and then select **New Project**.
The New Project dialog box appears.
2. In the Categories section of the Choose Project panel, select **OHMPI** and in the Projects section, select **IHE Profiles Application**.
3. Click **Next**.
The New IHE Profiles Application dialog box appears.
4. In the Name and Location panel, perform the following:
 - a. In the **Project Name** field, enter a name for your project (for example, IHEProject1).
 - b. In the **Project Location** field, browse to the path where you want to create the IHE project.
The **Project Folder** field populates with the path and name of your project.
 - c. From the **Server** drop-down list, select the application server.
 - d. From the **Database** drop-down list, select a database (for example, **Oracle**).
 - e. In the **V2 Service Port** field, enter the port number (for example, 4447).
 - f. Click **Finish**.

Configuring the Master Person Index Project

You can configure several features of the Master Person Index to customize how data is matched and processed through the IHE Profiles Application. The type and quality of data stored for a patient varies from system to system and from organization to organization. The Master Person Index takes this into account and lets you tailor the processing for your specific data requirements. For example, you may know that some systems contain more accurate data than others. You can weight those systems higher when determining which values will populate the single best record (SBR). Also, some fields might contain more accurate data than others and would thus be given a higher match weight than other, less accurate fields.

You can configure the following components of the Master Person Index:

- Patient demographic queries, matching queries, and queries performed from the Master Index Data manager (MIDM).
- Standardization and matching rules, including which fields to standardize, which fields to match, a range of match weights to assign to each field, and the type of algorithm to use to match each field.
- Filters for match results. For example, you can configure the Master Person Index to ignore default values.
- The single best record and how it is formed.
- The appearance of the MIDM

- Field validations

In addition, you can customize logic for the underlying process of the Master Person Index. For complete information on configuring the Master Person Index, see *Oracle Healthcare Master Person Index Configuration Guide*.

If you modify the configuration of the Master Person Index, you need to rebuild and redeploy the IHE Profiles Application Project. See [Creating an IHE Profiles Application Project](#) on page 2-1.

Note: You cannot modify an object model in an IHE-MPI application.

Using the IHE Project Actions Menu

When you right-click an IHE Profiles Application project in the left panel of NetBeans IDE, the Action menu appears. It provides the necessary tools to manage a project.

- **Build** - Builds an IHE Profiles Application project and generates its artifacts.
- **Clean and Build** - Removes generated artifacts from an IHE Profiles Application project and then rebuilds the project and generates new artifacts.
- **Clean** - Removes generated artifacts from an IHE Profiles Application project.
- **Create Database** - Opens the Create Database dialog box (see [Creating the Database and Tables Automatically](#) on page 3-3).
- **Set as Main Project** - Sets a selected IHE Profiles Application project as the main project.
- **Open Required Project** - Opens a pre-configured Master Person Index project.
- **Close** - Closes an IHE Profiles Application project.
- **Properties** - Opens the Project Properties - *<Project_Name>* dialog box, where you can change the HL7 V2 Service Port and Web Service Security Setting.

Configuring an IHE Profiles Application Project

Before building an IHE Profiles Application project you must configure it. This includes:

- Editing the following configuration files to change IHE service properties:
 - hl7message.xml
 - hl7service.xml
 - services.properties
- Using the Master Person Index editor to fine-tune matching configuration. See *Oracle Healthcare Master Person Index Configuration Guide* for information about using the editor.
- If the web service endpoints of the IHE Profiles Application need to be secured with TLS, check the **Secure Application** check box. This disables the plain HTTP access to the web service endpoints, and only allows HTTP over TLS access with mutual authentication. See [To Configure Secure PIXv2 Manager and PDQv2 Supplier](#) on page 2-8 for additional information on secure application configuration.

To Configure hl7message.xml

hl7message.xml defines HL7 v2 native message properties according to the HL7 specification.

- **validateMSH**

An indicator of whether MSH segment validation is enabled.
- **seqNumEnabled**

An indicator of whether sequence number protocol is enabled.
- **processingId**

The processing Id value of the MSH-11 segment in the received message is validated when validateMSH is enabled. MSH-11 is used to indicate whether a message is processed as defined in the HL7 processing rules. The processing Id options include the following:

 - **D** - Debugging
 - **T** - Training
 - **P** - Production
- **sendingApplicationNamespaceId**

The sending application (MSH-03 segment) is used to create an acknowledgement or response message. This is used to help identify the application from other participating applications within the network enterprise.
- **sendingApplicationUniversalId**
- **sendingApplicationUniversalIdType**
- **sendingFacilityNamespaceId**

The sending facility (MSH-04 segment) is responsible for creating an acknowledgement or response message. This is used to help further identify participating facilities within the network enterprise.
- **sendingFacilityUniversalId**
- **sendingFacilityUniversalIdType**
- **encodingCharacters**

The four encoding characters used to create an acknowledgement or response message. This attribute contains the four characters in the following order:

 - **Component separator**
 - **Repetition separator**
 - **Escape character**
 - **Subcomponent separator**

The recommended value is ^~\& (that is, ASCII 94, 126, 92, and 38, respectively).
- **fieldSeparator**

The separator is used to separate between the segment Id and the first real field. This value defines the character that is used as a separator for the rest of the message. Enter the value as a decimal ASCII number ranging from 1 to 127. The default value is 124 which is the character "|".
- **softwareVendorOrganization**

The Software Vendor Organization field (SFT-1-Software Vendor Organization) identifies the vendor who is responsible for maintaining the application. This property only applies to HL7 versions 2.5 relevant IHE integration profile messages.

- **softwareCertifiedVersionOrReleaseNumber**

The Software Certified Version or Release Number is the value of the HL7 segment SFT-02. The current software version number or release number for the sending system helps to provide a more complete profile of the application that is sending or receiving the HL7 messages. This property only applies to HL7 versions 2.5 relevant IHE integration profile messages.

- **softwareProductName**

The name of the software product that submitted the transaction is the value of the HL7 segment SFT-03. The software product name is a key component for identifying the sending application. This property only applies to HL7 versions 2.5 relevant IHE integration profile messages.

- **softwareBinaryId**

The Software Binary Id is the value of HL7 segment SFT-04. This property is available starting with HL7 version 2.5. Software Binary Ids are issued by a vendor for each unique software version instance. These Ids are used to differentiate between differing versions of the same software. Identical Primary Ids indicate that the software is identical at the binary level, but configuration settings may differ. This property only applies to HL7 versions 2.5 relevant IHE integration profile messages.

- **softwareProductInformation**

The software product identification information is the value of the HL7 segment SFT-05. This may include a description of the software application, configuration settings, modifications made to the software, and so forth. This information is used for diagnostic purposes to help identify the application software. This property only applies to HL7 versions 2.5 relevant IHE integration profile messages.

- **softwareInstallDate**

The Software InstallDate is the value of HL7 segment SFT-06. This is the date on which the submitting software was installed at the sending site. The install date can provide key information in regard to the behavior of an application. The date format is YYYYMMDDHHSS. This property only applies to HL7 versions 2.5 relevant IHE integration profile messages.

The following properties are related to the XDS Document Registry. See the `enable_xad_pid_link_change_notification` property in `services.properties` on how to enable XAD-PID Link Change Notification from OHMPI HL7 v2 service to XDS Document Registry.

- `registryApplicationNamespaceId`
- `registryApplicationUniversalId`
- `registryApplicationUniversalIdType`

The receiving application is used to identify the participating application for the XDS Document Registry within the network enterprise.

- `registryFacilityNamespaceId`
- `registryFacilityUniversalId`

- **registryFacilityUniversalIdType**
The receiving facility is used to identify the participating facility for the XDS Document Registry within the network enterprise.
- **registryEndpointAddress**
This is the HL7 v2 endpoint address for the XDS Document Registry.

To Configure hl7service.xml

hl7service.xml defines hl7 v2 server runtime communication protocol properties.

The runtime communication protocol properties of the OHMPI HL7 v2 server specifies the protocol information for the external HL7 system to connect to the OHMPI HL7 v2 server and vice versa.

- **url**
A URL used for an external system to connect to the OHMPI HL7 v2 server. The format of the URL is: hl7://<host>:<port>.
- **threadCount**
The maximum number of processing threads.
- **connectionTimeout**
The idle time in seconds for a connection timeout.
- **maxConnectRetries**
The maximum number of connection retries the outbound connector attempts before executing the recourse action.
 - **Value:** The value is presented in the format: n1;n2, n1;n2, ..., where n1 is the number of retry attempts to connect, and n2 is the interval in seconds between retry attempts. If the value of n1 is -1, it indicates that the number of retry attempts is indefinite.
 - **Action:** Suspend and Error.
- **TimeToWaitForAResponse**
The amount of time the outbound connector waits for a response from the external system before executing the configured recourse action.
 - **Value:** An integer indicating the time interval in seconds.
 - **Action:** Resend, Reset, and Suspend.
- **maxNakSent**
The maximum number of negative acknowledgements (NAKs) the server sends to the client before executing the recourse action. A NAK is sent from the HL7 inbound connection when the received message from client is invalid. An invalid message, according to the HL7 specification, is one that fails MSH segment validation or has an invalid sequence number.
 - **Enabled:** true or false
 - **Value:** An integer indicating the appropriate maximum number
 - **Action:** Suspend and Reset
- **maxCannedNakSent**

The maximum number of Canned NAKs the server sends to the client before executing the configured recourse action. This communication control deals with the case where the message received from the client is invalid as per the HL7 specification rules for that particular message. A Canned HL7 NAK is created when a read error occurs, or when a message cannot be identified as an HL7 message.

- **Enabled:** true or false
- **Value:** An integer indicating the appropriate maximum number
- **Action:** Suspend and Reset

- **mllpv1**

The MLLP Version 1 properties configure the Minimal Lower Layer Protocol version 1 property.

- **IlpType:** MLLPv1 indicates Minimal Lower Layer Protocol version 1.0.
- **StartBlockCharacter:** The start block character value in a decimal ASCII number from 1 to 127 specifies a block start. Unless there is a conflict, the value should be ASCII VT which is decimal 11.
- **EndBlockCharacter:** The end block character value in decimal ASCII number from 1 to 127 specifies a block end. To be strictly compliant with the HL7 standard, this parameter must be set to a carriage return which is decimal 13.
- **EndDataCharacter:** The end data character value in decimal ASCII number from 1 to 127 specifies a data end. To be strictly compliant with the HL7 standard, the value should be a carriage return indicated by decimal 28.

- **mllpv2**

The MLLP Version 2 properties configure the Minimal Lower Layer Protocol version 2 property.

- **IlpType:** MLLPv2 indicates Minimal Lower Layer Protocol version 2.0.
- **StartBlockCharacter:** The start block character value in a decimal ASCII number from 1 to 127 specifies a block start. Unless there is a conflict, the value should be ASCII VT which is decimal 11.
- **EndBlockCharacter:** The end block character value in decimal ASCII number from 1 to 127 specifies a block end. To be strictly compliant with the HL7 standard, this parameter must be set to a carriage return which is decimal 13.
- **EndDataCharacter:** The end data character value in decimal ASCII number from 1 to 127 specifies a data end. To be strictly compliant with the HL7 standard, the value should be a carriage return indicated by decimal 28.
- **RetryCountOnNak:** The maximum configured number of attempts to send a message after receiving a negative acknowledgement.
- **RetryInterval:** The duration in milliseconds to wait between each retry attempt.
- **timeToWaitForAckNak:** The duration in milliseconds to wait for a commit positive or negative acknowledgement.
- **hllp**
 - **IlpType:** HLLP indicates Hybrid Lower Layer Protocol.
 - **StartBlockCharacter:** The start block character value in a decimal ASCII number from 1 to 127 specifies a block start. Unless there is a conflict, the

value should be ASCII VT which is decimal 11.

- **EndBlockCharacter:** The end block character value in decimal ASCII number from 1 to 127 specifies a block end. To be strictly compliant with the HL7 standard, this parameter must be set to a carriage return which is decimal 13.
- **EndDataCharacter:** The end data character value in decimal ASCII number from 1 to 127 specifies a data end. To be strictly compliant with the HL7 standard, the value should be a carriage return indicated by decimal 28.
- **checksumEnabled:** An indicator of whether the HLLP Checksum feature is enabled. This can only be enabled if the LLP Type is set to HLLP.

The actions supported by the IHE OHMPI HL7 v2 server can include the following values:

- **Reset**
The server closes the connection with the HL7 external system and throws an alert.
- **Resend**
The server component resends the last sent message to the HL7 external system.
- **Suspend**
The server closes the connection with the HL7 external system, suspends the connection from processing the messages, and throws an alert.

To Configure Secure PIXv2 Manager and PDQv2 Supplier

To secure IHE Profiles Application you need to enable SSL support, including Mutual authentication.

Note: This configuration step is optional.

To configure secure PIXv2 Manager and PDQv2 Supplier you need to configure `hl7service.xml`:

- **url**
Define a URL used for secured connection. The formation of the URL is:
`hl7s://<host>:<port>`.
- **keyStore**
Define a keyStore file full path where private keys in JKS format are stored.
- **keyStorePassword**
Define a password to access to the keyStore.
- **trustStore**
Define a trustStore file full path where trusted certificates in JKS format are stored.
- **trustStorePassword**
Define a password to access to the trustStore.
- **keyManagerKeyStorePassword**
Define a password for KeyManager.

- **clientAuthentication**

Set **true** for requiring client authentication, or mutual authentication. Set **false** for only requiring server authentication.

- **protocols**

Define supported SSL protocols. You can define a list of different protocols delimited by "," (that is, you can define "SSLv3,TLSv1").

- **cipherSuites**

Define supported SSL cryptographic algorithms. You can define a list of different cipher suites delimited by "," (that is, you can define "TLS_RSA_WITH_AES_128_CBC_SHA,SSL_RSA_WITH_NULL_SHA").

To import keys and certificates into KeyStore or TrustStore, perform the following:

- If your certificate or key is X509 format, transform X509 certificates and keys to pkcs#12 format using openssl tool:

```
openssl pkcs12 -export -out my_keystore.pkcs12 -in my_server_cert.pem
-inkey my_server_key.pem
```

- Import the pkcs#12 certificate into a JKS keystore using keytool:

```
keytool -importkeystore -deststorepass changeit -destkeystore my_
keystore.jks -srckeystore my_keystore.pkcs12 -srcstoretype PKCS12
-srcstorepass changeit
```

- Import the public keys into the trust store using keytool:

```
keytool -import -noprompt -trustcacerts -file my_client_cert.pem
-keystore my_truststore.jks -alias client
```

To Configure services.properties

The following is a list of user-configurable properties for IHE profiles and HL7 v3 web services. You need to manually edit the services.properties file before building the IHE project.

Note: If you have to change any of these configurations after you build and/or deploy the IHE project, you can always rebuild and redeploy the IHE project after updating the services.properties file. Alternatively, you can create a runtime property file called ohmpi-ihe.properties under the <app_server_domain>/config/ohmpi directory. The properties defined in the runtime ohmpi-ihe.properties overwrite the properties defined in services.properties at build time.

Open the services.properties file from within NetBeans IDE and make the appropriate edits to the following.

- pixv3mgr/receiver/device/namespace=OTHER_ORACLE_HIE
- pixv3mgr/receiver/device/oid=1.3.6.1.4.1.21367.13.30.92
- pixv3mgr/receiver/device/org/namespace=ORACLE
- pixv3mgr/receiver/device/org/oid=1.3.6.1.4.1.21367.13.50.99
- pixv3mgr/sender/device/namespace=OTHER_ORACLE_HIE
- pixv3mgr/sender/device/oid=1.3.6.1.4.1.21367.13.10.92

- pixv3mgr/sender/device/org/namespace=ORACLE
- pixv3mgr/sender/device/org/oid=1.3.6.1.4.1.21367.13.50.99
- pdq/timeout=60
- pdq/batch_size=20
- pdq/date_format=yyyyMMddHHmmss
- pdq/default_search_algorithm=ALPHA-SEARCH
- enable_pix_update_notification=true
- enable_euid_query=false
- aa/oid=1.3.6.1.4.1.21367.13.20.165
- aa/namespace=ORO6
- mpi/application/name=Patient
- mpi/primary_object/name=Patient
- mpi/primary_object/class_name=com.sun.mdm.index.objects.patient.PatientObject
- service/mapper/class_name=oracle.hsgbu.ohmpi.ihe.mapper.DefaultMapperService
- enable_patient_feed=false
- registry/endpoint=http://<RLS_HOST>:<RLS_PORT>/hrl/regsvc
- psv3/device/namespace=OTHER_ORACLE_HIE2015
- psv3/device/oid=1.3.6.1.4.1.21367.13.10.175
- psv3/device/org/namespace=ORACLE
- psv3/device/org/oid=1.3.6.1.4.1.21367.13.50.300088
- registry/device/namespace=OTHER_ORACLE_HIE2015
- registry/device/oid=1.3.6.1.4.1.21367.13.30.181
- registry/device/org/namespace=ORACLE
- registry/device/org/oid=1.3.6.1.4.1.21367.13.50.300088
- enable_xad_pid_link_change_notification=false
- xad/oid=
- xad/namespace=

The following properties identify the PIXv3 Manager (as well as the PDQv3 Supplier) as a receiver:

Property Name	Property Description
pixv3mgr/receiver/device/namespace	PIXv3 Manager (or PDQv3 Supplier) receiver device namespace
pixv3mgr/receiver/device/oid	PIXv3 Manager (or PDQv3 Supplier) receiver device OID
pixv3mgr/receiver/device/org/namespace	PIXv3 Manager (or PDQv3 Supplier) receiver device organization namespace
pixv3mgr/receiver/device/org/oid	PIXv3 Manager (or PDQv3 Supplier) receiver device organization OID

The following properties identify the PIXv3 Manager as a sender in PIXv3 Update Notification:

Property Name	Property Description
pixv3mgr/sender/device/namespace	PIXv3 Manager sender device namespace
pixv3mgr/sender/device/oid	PIXv3 Manager sender device OID
pixv3mgr/sender/device/org/namespace	PIXv3 Manager sender device organization namespace
pixv3mgr/sender/device/org/oid	PIXv3 Manager sender device organization OID

The following properties configure general PIX/PDQ behavior:

Property Name	Property Description
pdq/timeout	Patient Demographics Query Continuation timeout (in seconds)
pdq/batch_size	Patient Demographics Query Continuation batch size
pdq/date_format	Date format in Patient Demographics Query response. Refer to Java's SimpleDateFormat for format at http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html .
pdq/default_search_algorithm	Patient Demographics Query default search algorithm. Valid values are "ALPHA-SEARCH", "PHONETIC-SEARCH", "BLOCKER-SEARCH", or any user-defined query builder names.
enable_pix_update_notification	Whether to enable PIX v2/v3 Update Notification. Valid values are "true" and "false". By default, this is enabled. When PIX Update Notification is enabled, PIX Consumer Endpoints need to be added into the IHE_PIXCONSUMER_ENDPOINTS table, and PIX Update Notification Subscriptions from PIX Consumers to Source Domains need to be added into the IHE_PIXUPDATE_SUBSCRIPTIONS table.
enable_euid_query	Whether to enable PIX/PDQ Query by EUID and PIX/PDQ Query for EUID. Valid values are "true" and "false". By default, this is disabled. See aa/oid and aa/namespace for the EUID domain identification.
aa/namespace	OHMPI Assigning Authority (i.e., EUID Domain) namespace Note: Both aa/namespace and aa/oid need to be properly defined together when PIX/PDQ Query by/for EUID (enable_euid_query) is enabled. The OHMPI Assigning Authority Domain, that is, the EUID Domain, also needs to be added into the IHE_DOMAINS table in this case. Note: This namespace is not necessarily the same namespace for the XDS Affinity Domain. See xad/namespace.
aa/oid	OHMPI Assigning Authority (i.e., EUID Domain) OID Note: Both aa/namespace and aa/oid need to be properly defined together when PIX/PDQ Query by/for EUID (enable_euid_query) is enabled. The OHMPI Assigning Authority Domain, that is, the EUID Domain, also needs to be added into the IHE_DOMAINS table in this case. Note: This OID is not necessarily the same OID for the XDS Affinity Domain. See xad/oid.

The following properties need to be updated if user-defined MPI Object Model is used along with IHE Project:

Property Name	Property Description
mpi/application/name	Application name in the user-defined MPI project
mpi/primary_object/name	Primary object name in the user-defined MPI project

Property Name	Property Description
mpi/primary_object/class_name	Fully qualified Java class name for the user-defined MPI project's primary object.
service/mapper/class_name	Fully qualified name of a Java class that implements the pre-defined oracle.hsgbu.ohmpi.services.IMapperService interface. This concrete service implementation class provides two-way object mapping between the fixed IHE model and the user-defined MPI model.

The following properties configure outbound patient feed from OHMPI to XDS Document Registry:

Property Name	Property Description
enable_patient_feed	<p>Whether to enable outbound patient feed from OHMPI to XDS Document Registry. Valid values are "true" and "false". By default, this patient feed is disabled.</p> <p>When outbound patient feed is disabled ("false"), you can ignore the rest of the properties in this section.</p> <p>When outbound patient feed is enabled ("true"), you must properly define the Registry's web service endpoint URL. Also, you must configure the identity of Registry (receiver) and the identity of OHMPI as a Patient Identity Source (sender).</p> <p>Note: OHMPI can have a different identity as a sender from its identity as a receiver.</p>
registry/endpoint	<p>XDS Document Registry's Web Service Endpoint URL</p> <p>For example, the Health Record Locator (HRL) service contained in Oracle Health Sciences Information Manager is an XDS Document Registry. It is also known as Record Locator Service. Its endpoint URL looks like the following: http://<RLS_HOST>:<RLS_PORT>/hrl/regsvc</p>
pisv3/device/namespace	<p>Patient Identity Source v3 (Sender) Device Namespace.</p> <p>This, along with the next three properties, identifies OHMPI as a sender in an outbound patient feed.</p>
pisv3/device/oid	Patient Identity Source v3 (Sender) Device OID
pisv3/device/org/namespace	Patient Identity Source v3 (Sender) Device Organization Namespace
pisv3/device/org/oid	Patient Identity Source v3 (Sender) Device Organization OID
registry/device/namespace	XDS Document Registry (Receiver) Device Namespace
registry/device/oid	XDS Document Registry (Receiver) Device OID
registry/device/org/namespace	XDS Document Registry (Receiver) Device Organization Namespace
registry/device/org/oid	XDS Document Registry (Receiver) Device Organization OID

The following properties configure XAD-PID Link Change Notification from OHMPI to XDS Document Registry:

Property Name	Property Description
enable_xad_pid_link_change_notification	<p>Whether to enable XAD-PID Link Change Notification from OHMPI to XDS Document Registry. Valid values are "true" and "false". By default, this notification is disabled.</p> <p>Since the notification listener is defined by IHE in the XAD-PID Change Management (XPID) Profile as an HL7 v2 service on the XDS Document Registry, when XAD-PID Link Change Notification from OHMPI to XDS Document Registry is enabled. The following HL7 v2 properties must be properly defined in hl7message.xml:</p> <ul style="list-style-type: none"> ▪ registryApplicationNamespaceId ▪ registryApplicationUniversalId ▪ registryApplicationUniversalIdType ▪ registryFacilityNamespaceId ▪ registryFacilityUniversalId ▪ registryFacilityUniversalIdType ▪ registryEndpointAddress
xad/oid	<p>XDS Affinity Domain OID</p> <p>If the value of this property is left empty, it is assumed to have the same value as aa/oid, in other words, OHMPI's EUID Domain is treated as the XDS Affinity Domain.</p>
xad/namespace	<p>XDS Affinity Domain Namespace</p> <p>If the value of this property is left empty, it is assumed to have the same value as aa/namespace. OHMPI's EUID Domain is treated as the XDS Affinity Domain.</p>

Building an IHE Profiles Application Project

This section provides the steps to build a project.

To Build an IHE Profiles Application Project

1. In the left panel of NetBeans IDE, right-click the project icon (for example, **IHEProject1**).

The Actions menu appears.

2. To build an IHE Profiles Application project, click **Build** on Actions menu.

When you build the IHE Profiles Application project for WebLogic, five artifacts are created:

- *Project_Name*.ear: The IHE Profiles Application.
- hl7v2.zip: The HL7 v2 lifecycle module.

The hl7v2.zip is an HL7 v2 server package that includes all artifacts required to run the IHE OHMPI HL7 v2 server and HL7 v2 lifecycle model which manages the server. You unzip it to the location where you run the server. The unzipped folder includes:

hl7v2

- config

The config sub folder includes all the configuration files.

- lib

The lib sub folder includes all the required jar files.

- `audit-repo-syslog-client.jar`: The JAR file that contains the functionality of the audit repository syslog client.
- `ihe-atna-audit-client.jar`: The JAR file that contains the functionality of the IHE ATNA audit client.
- `ohmpi_audit_client.properties`: The property file used to configure where the audit server is located.

After building a WebLogic-targeted IHE Profiles Application project (and before starting WebLogic Application Server), manually copy the following three files from the IHE Profiles Application project's `dist` directory to the appropriate WebLogic domain's `lib` directory (for example, `user_projects\domains\domain_name\lib`):

- `audit-repo-syslog-client.jar`
- `ihe-atna-audit-client.jar`
- `ohmpi_audit_client.properties`

Creating MPI and IHE Databases and Tables

This chapter provides procedures for creating Master Person Index and IHE Profiles Application databases and tables.

This chapter includes the following section:

- [Creating the IHE Profiles Application Databases and Tables](#) on page 3-1

Creating the IHE Profiles Application Databases and Tables

After installing Oracle Healthcare Master Person Index (OHMPI), you need to create the Master Person Index tables and IHE Profiles Application database tables. You must create these tables before you can work with the IHE Profiles Application. You can either create the databases and tables through OHMPI design-time (which is recommended), or create them manually.

Note: Creating a database is optional; if desired you can create your own database.

To create databases and table manually, see [Creating Master Person Index Databases and Tables for Oracle Manually](#) on page 3-1.

To create databases and tables using OHMPI design-time, see [Creating the Database and Tables Automatically](#) on page 3-3.

To configure optional connecting information, see [Maintaining Subscriptions to Patient Updates](#) on page 3-3.

Creating Master Person Index Databases and Tables for Oracle Manually

Before you begin, ensure that you have Oracle installed as a service with remote host access enabled. Perform the following to create all databases:

To Create the Master Person Index Database

1. Create an Oracle environment and user for the Master Person Index tables by running the following command:

```
grant CREATE SESSION, CREATE TABLE, CREATE TRIGGER, CREATE PROCEDURE,  
UNLIMITED TABLESPACE, CREATE TYPE to <USER_NAME> identified by <PASSWORD>
```

2. Open `systems.sql` in a text editor, and add the HL7 systems with which you will be sharing data.

Note: See Step 5 for the location of this file.

For more information about modifying this file, see *Oracle Healthcare Master Person Index User's Guide*.

3. Open `codelist.sql` in a text editor, and add any common code information you need for the Master Person Index system.

Note: See Step 5 for the location of this file.

For more information about code lists and modifying this file, see *Oracle Healthcare Master Person Index User's Guide*.

Note: This step is optional. The default configuration does not use common code tables. Adding this feature causes additional validations to be performed against incoming data.

4. Log in to the **mpi** schema as the user you created earlier.
5. Run the following SQL files against the **mpi** schema (`<project_name>\ihe-mpi\src\DatabaseScript`). The `create.sql` file must be run first.
 - `create.sql`
 - `systems.sql`
 - `codelist.sql`
 - `create-index.sql`

Note: To drop MPI tables, run the following SQL files against the **mpi** schema:

- `drop-index.sql`
 - `drop.sql`
-
-

To Create the IHE Profiles Application Database Tables

Before you begin, complete [To Create the Master Person Index Database](#) on page 3-1.

1. Navigate to the directory to which you created the IHE Profiles `<project_name>` Application projects, and then navigate to the `\src\DatabaseScripts` folder.
 - `create_ihe_ohmpi_tables.sql` creates all the IHE Profiles Application tables.
 - `create_ihe_ohmpi_sample_data.sql` creates sample domain configuration data.
 - `clean_ihe_ohmpi_tables.sql` cleans all the records from IHE Profiles Application tables and MPI tables.
2. Open `create_ihe_ohmpi_sample_data.sql` in a text editor, comment out all existing insert statements, and add any systems you added to the `systems.sql` file in [To Create the Master Person Index Database](#) on page 3-1.

You need to insert the corresponding namespace ID, universal ID, universal ID type and a description for each new system; for example:

```
insert into IHE_DOMAINS (NAMESPACEID, UNIVERSALID, UNIVERSALIDTYPE,
DESCRIPTION) values ('HOSPITAL1', '1.4.5.2.6.2.23455', 'ISO',
'HOSPITAL1 DESCRIPTION');
```

3. (Optional) To configure patient update notification, see [Maintaining Subscriptions to Patient Updates](#) on page 3-3.
4. Save and close the file.
5. Log in to the **mpi** schema as the user you created earlier.
6. Run the following scripts against the **mpi** schema.
 - `create_ihe_ohmpi_tables.sql`
 - `create_ihe_ohmpi_sample_data.sql`

Creating the Database and Tables Automatically

1. Right-click the name of your IHE Profiles Application project in the left panel of the NetBeans IDE (for example, **IHEProject1**) and select **Create Database**.
The Create Database dialog box appears. Here you provide the credentials for an existing Oracle database.
2. Click the **Root Connection** tab and perform the following to configure the database:
 - **Server Host:** The name of the server where the database is located.
 - **Root User** - The login name to the database.
 - **Root Password:** The password used to log in to the database.
 - **Server Port:** The port number of the server where the database is located.
 - **SQL URL:** The URL for the database.
3. After the Connection requirements for the database are set, click **Test Connection**.
4. When satisfied with the Connection settings, click the **User-Schema** tab.
5. Click the **User-Schema** tab and perform the following to configure the database user:
 - View the user for the database from the list.
 - Accept the default password or enter a new password.
 - Click **Apply**.
6. Click the **Database Scripts** tab to view the scripts and then perform the following:
 - To run the scripts, click **Run**.
 - After the scripts complete running, click **Close**.

Maintaining Subscriptions to Patient Updates

The IHE Profiles Application was designed to be implemented with minimal required customizations. You can customize and extend the functionality of OHMPI's connectivity information for a product environment.

In order for a PIX Consumer to receive PIX Update Notifications from a source domain, add a subscription record to the IHE_PIXUPDATE_SUBSCRIPTIONS table. For example,

```
insert into IHE_PIXUPDATE_SUBSCRIPTIONS (CONSUMERID, SOURCENAMEID,
ACTIVE) values ('CONSUMER1', 'IHEGHCADT', 'T');
```

where CONSUMER1 identifies a PIX Consumer that receives updates and IHEGHCADT is the source domain that sends updates.

For each PIX Consumer that is interested in receiving PIX Update Notifications, a record describing the PIX Consumer's endpoint should be added to the IHE_PIXCONSUMER_ENDPOINTS table first.

Ensure that the following are correct when configuring PIX Consumer Endpoints:

- The identifier of the PIX Consumer. This needs to match the CONSUMERID in the IHE_PIXUPDATE_SUBSCRIPTIONS table.
- The endpoint type (either HL7V2 or HL7V3).
- The LLPTYPE (applicable only if the endpoint type is HL7V2).
- The receiving URL for the PIX Consumer.
- The name of the HL7 application or device.
- The name of the facility or organization (organization is optional).

The following is an example describing a PIXv2 Consumer:

```
insert into IHE_PIXCONSUMER_ENDPOINTS (CONSUMERID, ENDPOINTTYPE, LLPTYPE,
ADDRESS, APPL_DEV, FACL_ORG, DESCRIPTION) values ('CONSUMER1', 'HL7V2',
'MLLPv1', 'hl7://localhost:9906', 'SUNPIXPDQRAD', NULL, 'Test domain');
```

The following is an example describing a PIXv3 Consumer:

```
insert into IHE_PIXCONSUMER_ENDPOINTS (CONSUMERID, ENDPOINTTYPE, LLPTYPE,
ADDRESS, APPL_DEV, FACL_ORG, DESCRIPTION) values ('CONSUMER2', 'HL7V3',
NULL, 'http://localhost:8080/PIXConsumer2_Service/PIXConsumer2',
'1.2.840.114350.1.13.99997.2.3412', NULL, 'Test domain');
```

Getting Started With Oracle WebLogic Server

This chapter provides the procedures for creating an instance of an Oracle WebLogic Server. It includes creating an Oracle Healthcare Master Person Index (OHMPI) project, starting the OHMPI wizard, adding the WebLogic Server, and a listing where instructions for creating, configuring, and building a Master Person Index application are located.

This chapter includes the following section:

- [Creating an Instance of an Oracle WebLogic Server](#) on page 4-1

Creating an Instance of an Oracle WebLogic Server

Oracle recommends hosting OHMPI on a dedicated WebLogic Server instance.

The OHMPI wizard provides a simple and transparent method for you to create your master person index application on the Oracle WebLogic Server. You have to add an Oracle WebLogic Server Instance or select an Oracle WebLogic Server Instance if you already have one or more instances. This process is divided into three individual steps, listed below.

- [Creating a Project and Starting the OHMPI Wizard](#) on page 4-1
- [Adding an Oracle WebLogic Server](#) on page 4-2
- [Creating, Configuring, and Building a Master Person Index Application](#) on page 4-3

Related Documentation

For additional information on the Oracle WebLogic Server, see Oracle WebLogic Server 12c Release 3 documentation.

Oracle WebLogic Server Requirements

The Oracle WebLogic Server requires:

- NetBeans IDE 8.0
- JDK 1.7.0_75 and later
- WebLogic 12cR2 (12.1.2) or above

Creating a Project and Starting the OHMPI Wizard

This section documents the steps to create an OHMPI project and start the OHMPI wizard.

To Create a Project and Start the OHMPI Wizard

1. On the NetBeans toolbar, click **New Project**.
2. Under Categories, select **OHMPI**.
3. Under Projects, select **Master Person Index Application** and then click **Next**.
The New Master Person Index Application page appears.
4. Type the project name and the path where you want to store the project files in the upper portion of the window.

Adding an Oracle WebLogic Server

This section provides the steps for adding an instance of an Oracle WebLogic Server.

To Add an Oracle WebLogic Server

1. On the New Master Person Index Application page, if an Oracle WebLogic Server instance is not listed in the **Server** drop-down list, click **Manage** to add it.

Note: If an Oracle WebLogic Server instance is listed in the **Server** field, or if you can select it from the drop-down list, click **Next** to continue to [Creating, Configuring, and Building a Master Person Index Application](#) on page 4-3.

The Servers page appears without the WebLogic Server listed under Servers.

Note: If there is a server listed, its properties display on the right side of the page.

2. Click **Add Server**.
The Add Server Instance page appears.
3. In the Server list, select **Oracle WebLogic Server**.
4. In the **Name** field, enter the instance name of your Oracle WebLogic Server (for example, Oracle WebLogic Server 12cR3).
5. Click **Next**.
The Add Server Instance page moves to the second step.
6. Under Server Location, browse to the root location of your Oracle WebLogic Server instance.

Note: Optionally, you can type the root location of your Oracle WebLogic Server instance (for example, C:\Oracle\Weblogic12cR3\wlserver).

7. Click **Next**.
The Add Server Instance page moves to the third step, Instance Properties. The fields have been populated except for the password.
8. Type a password for your Oracle WebLogic Server Instance.

Note: Do not change the Instance Properties that were populated for you.

9. Click Finish.

The Servers page reappears with the WebLogic Server you created listed on the left side of the page. Its properties are displayed on the right side of the page.

Note: If there is more than one server in the list, select the server you created to display its properties.

10. Click Close.

The name of the Oracle WebLogic Server instance you defined should appear in the **Server** field under Name and Location on the right side of the New Master Index Application page.

11. Do one of the following:

- If the Oracle WebLogic Server instance is listed in the **Server** drop-down list, click **Next**.
- If the Oracle WebLogic Server instance is not listed in the **Server** drop-down list, select it from the drop-down list and click **Next**.

You are now ready to move on to the third step of this process, [Creating, Configuring, and Building a Master Person Index Application](#) on page 4-3.

Creating, Configuring, and Building a Master Person Index Application

The generic instructions to create, configure, and build a master person index application are documented elsewhere. To perform these tasks see:

- *Oracle Healthcare Master Person Index User's Guide*
- *Oracle Healthcare Master Person Index Configuration Guide*
- *Oracle Healthcare Master Person Index User's Guide*

There is nothing for you to do as there are no specific procedures that are relevant to the Oracle WebLogic Server that you need to perform. The OHMPI wizard automatically generates and packages the artifacts of Oracle WebLogic Server for you.

Configuring an Application Server

This chapter provides procedures for configuring an Oracle WebLogic Server, including installing a database driver, creating JDBC data resources, creating JMS resources, and setting up a user.

This chapter includes the following section:

- [Configuring the Oracle WebLogic Server](#) on page 5-1

Configuring the Oracle WebLogic Server

Before using your Oracle WebLogic Server, you must prepare it to host your Master Person Index (MPI) Application or your IHE Profiles Application, using the following procedures:

- [Configuring the HL7 v2 Server](#) on page 5-1
- Configure the audit client. For information, see [To Build an IHE Profiles Application Project](#) on page 2-13.
- [Creating JDBC Data Resources for an IHE Profile Application Project](#) on page 5-2
- [Creating JMS Resources for an IHE Profile Application Project](#) on page 5-3
- [Setting Up the User](#) on page 5-6

Configuring the HL7 v2 Server

1. After building an IHE Profiles Application project, unzip **hl7v2.zip** to a directory of your choice (for example, your OHMPI installation directory).
2. Set the **OHMPI_IHE_HL7V2_HOME** environment variable to the unzipped hl7v2 directory from the previous step.

To set the **OHMPI_IHE_HL7V2_HOME** environment variable, update the *setDomainenv* script file.

- For Windows, open the `<domain_home>/bin/setDomainenv.bat` file and add the following command line:

```
set OHMPI_IHE_HL7V2_HOME=<your HL7v2 Server root path>
```

- For Linux, open the `<domain_home>/bin/setDomainenv.sh` file and add the following command line:

```
export OHMPI_IHE_HL7V2_HOME=<your HL7v2 Server root path>
```

Note: If you have used a colon in the property, you need to escape it with \\.

For example:

<property_ABC>=C:\\hl7v2Client

rather than

<property_ABC>=C:\hl7v2Client

3. Follow the existing instructions to configure and deploy the IHE Profiles Application to the WebLogic Server.

Note: For rest of the HL7 v2 server configuration and the SSL configuration for the application server, see [Configuring an IHE Profiles Application Project](#) on page 2-3.

Creating JDBC Data Resources for an IHE Profile Application Project

This section provides instructions for creating the JDBC data resources and defining the JDBC connections for an IHE Profile Application Project.

To Create JDBC Data Resources for an IHE Profiles Application Project for Oracle

1. Start and stop the Oracle WebLogic Server. For instructions, see *Oracle Fusion Middleware Online Documentation 12c Release 1 (12.1.2)*.
2. Launch the **Oracle WebLogic Server Administration Console**.
3. Log in using the default user Name (**weblogic**) and Password (**welcome1**).
The Oracle WebLogic Administration Console appears.
4. On the left panel, under Domain Structure, expand **Services**, click **Data Sources**.
A Summary of JDBC Data Sources appears in the right panel.
5. To create a new JDBC Data Source click **New** at the bottom of the right panel and select **Generic Data Source** from the drop-down list.
Settings for a new JDBC Data Source appear. It is here that you will create a new JDBC Data Source.
6. In the **Name** field, type *mpi_application_nameDataSource*.
The name you enter here will propagate elsewhere, and it must be the mpi application name that you have already created (for example, PatientDataSource).
7. In the **JNDI Name** field, type *jdbc/mpi_application_nameDataSource*.
Use the name you entered in step 6 here (for example, **jdbc/PatientDataSource**).
8. From the **Database Type** drop-down list, select the appropriate database type. For example, **Oracle**.
9. Click **Next**.
10. From the **Database Driver** drop-down list, select the appropriate driver. For example, **Oracle's Driver (Thin XA) for Instance Connections; Versions: 9.0.1 and later**.

11. Click Next.

Transaction Options appear on the page.

12. Click Next.

Connection Properties appears on the Create a New JDBC Data Source panel. Use it to define the connection properties.

13. In the Database Name field, type a name for the database to which you want to connect (for example: Patient).**14. In the Host Name field, type the name or the IP address of the database server (for example: localhost).****15. In the Port field, type the port on the database server that is used to connect to the database (for example: 1521).****16. In the Database User Name field, type the database account user name you want to use to create database connections (for example: patient).****17. In the Password field, type a password for your database account to use to create database connections.****18. In the Confirm Password field, re-type the password to confirm it.****19. Click Next.**

The Settings for PatientDataSource page appears in the right panel.

20. Click Test Configuration, and if successful click Next.

Note: If it fails, check the above steps.

Select Targets appear on the Create a New JDBC Data Source page in the right panel. Here you select one or more targets to deploy the new JDBC data source.

21. In the Servers check list, select one or more target servers and click Finish.

Note: If you do not select a target, the data source will be created but not deployed. You will need to deploy the data source at a later time.

22. Repeat the above steps to create jdbc/PatientSequenceDataSource.**23. Repeat the above steps to create jdbc/PIXDomainLUDataSource.****24. Repeat the above steps to create jdbc/PIMPendingLinksDataSource.**

Creating JMS Resources for an IHE Profile Application Project

JMS servers act as management containers for the queues and topics in the JMS modules that are targeted to them.

The following procedure includes instructions for creating JMS resources, which includes a:

- JMS Server
- JMS Module
- JMS Connection Factory in the specific JMS Module
- JMS Topic in the specific JMS Module

To Create JMS Server

1. On the left panel, under Domain Structure, expand **Services**, click **Messaging**, and then select **JMS Servers**.

A Summary of JMS Servers appears in the right panel. It includes a table that summarizes the JMS servers that have been created in the current WebLogic Server domain.

2. In the table of previously created JMS Servers, click **New**.

The Create a New JMS Server panel appears.

3. In the **Name** field, type the name for your new JMS Server.

Note: This name already exists in the table of previously created JMS Servers (in the example, **PatientJMSServer**).

4. Click **Next**.

Select targets appear in the right panel under Create a New JMS Server.

5. From the **Target** drop-down list, select a target server instance or migratable target on which you want to deploy the JMS Server.

If you are deploying on a Clustered environment, select the appropriate target server with the cluster environment.

Note: The default server instance is exampleServer.

6. Click **Finish**.

To Create JMS Module

1. On the left panel, under Domain Structure, expand **Services**, click **Messaging**, and then select **JMS Modules**.

The JMS Modules panel appears.

2. Click **New** to add a new JMS Module.

The Create JMS System Module panel appears.

3. In the **Name** field, type the new JMS Module name (for example, *mpi_applicationName*JMSModule).

Note: Again, remain consistent to the name chosen for the JDBC Data Source and the JMS Server (in the previous examples the key word was "Patient," making this name **PatientJMSModule**).

4. Click **Next**.

Targets appear in the right panel under Create a New JMS System Module.

5. In the Servers area, select the server or cluster on which you want to deploy this JMS system module.

Note: Retain the default, exampleServer.

6. Click **Next**.
7. Click **Finish**.

To Create JMS Connection Factory

1. On the left panel, under Domain Structure, expand **Services**, click **Messaging**, and then select **JMS Modules**.
2. Select the JMS Module (in the example, **PatientJMSModule**) from the table of JMS Modules.

The Settings for PatientJMSModule page appears in the right panel.

3. In the Summary of Resources table, click **New**.
4. Under the Type column in the Summary of Resources table, select **Connection Factory** and click **Next**.

Another panel of Create a New JMS System Module Resource appears.

5. In the **Name** field, type `PatientOutBoundSender`.
6. In the **JNDI Name** field, type `jms/PatientOutBoundSender`.
7. Make sure that the **XA Connection Factory Enabled** check box is selected.
8. Click **Next**.
9. In the **Target** field, retain the default server instance, which is `exampleServer`, and click **Finish**.

If you are deploying on a Clustered environment, select the appropriate target server with the cluster environment.

To Create JMS Topic

1. On the left panel, under Domain Structure, expand **Services**, click **Messaging**, and then select **JMS Modules**.
2. In the right panel, select the JMS Module you created (in the example, **PatientJMSModule**) from the table of JMS Modules.

Settings for PatientJMSModule appear in the right panel with a Summary of Resources table.

3. Click **New**, select **Topic**, and then click **Next**.

If you are deploying on a Clustered environment, select **DistributedTopic**.

The Create a New JMS System Module Resource panel appears on the right side of the window. Use this panel to set the properties that identify the new topic.

4. In the **Name** field, under JMS Destination Properties, type `mpi_application_nameTopic` (for example, `PatientTopic`).
5. Set `jms/PatientTopic` as the JNDI Name and click **Next**.

The Create a New JMS System Module Resource page appears in the right panel. Use this page to set the properties that will be used to target your new JMS system module resource.

6. In the **Subdeployments** drop-down list, select the topic name you just created (for example, `Patient Topic`) and click **Create a New Subdeployment**.
7. In the **Subdeployment Name** field, type `mpi_application_nameTopic` (for example, `PatientTopic`), and click **OK**.

8. In the Targets table of JMS Servers, select *mpi_application_name*JMS**Server** (for example, **Patient**JMS**Server**).
9. Click **Finish**.

To Create PixUpdateNotificationTopic

1. In the Summary of Resources table, click **New**, select **Topic**, and then click **Next**.
The Create a New JMS System Module Resource panel appears on the right side of the window. Use this panel to set the properties that identify the new topic.
2. In the **Name** field, under JMS Destination Properties, type *name*Topic (for example, PixUpdateNotificationTopic).
3. Set **jms/PixUpdateNotificationTopic** as the JNDI Name and click **Next**.
The Create a New JMS System Module Resource page appears in the right panel. Use this page to set the properties that will be used to target your new JMS system module resource.
4. In the **Subdeployments** drop-down list, select the topic name you just created (for example, PixUpdateNotification Topic) and click **Create a New Subdeployment**.
5. In the **Subdeployment Name** field, type *name*Topic (for example, PixUpdateNotificationTopic), and click **OK**.
6. In the Targets table of JMS Servers, select *name*JMS**Server** (for example, **Patient**JMS**Server**).
7. Click **Finish**.

Setting Up the User

In this step you create the MasterIndex.Admin and Administrator groups, and then create a new user within the two groups.

1. On the left panel, under Domain Structure, expand **Services**, and then select **Security Realms**.
2. In the table on the Summary of Security Realms panel, click **myrealm**, which is the name of the realm.
The Settings for myrealm panel appears.
3. Click the **Users and Groups** tab and then click **Groups**.
4. In the Groups table, click **New**.
5. In the **Name** field, type *MasterIndex.Admin* and click **OK**.
6. In the Groups table, click **New**.
7. In the **Name** field, type *Administrator* and click **OK**.
8. On the Settings for myrealm panel, select **Users and Groups** and then **Users**.
9. In the Users table, click **New**.
10. Type a name and a password for the new user you are creating and click **OK**.
11. Select **User Group**.
12. To add the two groups you created to the user you created, from the Available list, drag **MasterIndex.Admin** to the **Chosen** list, and then drag **Administrator** to the **Chosen** list.

13. Add the user **MasterIndex.WSUser** and then add the **MasterIndex.Admin** group for it.

Note: See *Oracle Healthcare Master Person Index User's Guide*.

Configuring and Using Audit Record Repository

The Audit Record Repository (ARR), which includes an audit server and an audit repository, is part of the Internet Protocol Suite that deals with the transmission of data. Specifically related to Oracle Healthcare Master Person Index (OHMPI) and the IHE Profiles Application, ARR provides secure transmission and auditing for healthcare application systems. The major components of the ARR include:

- Audit Trail and Node Authentication (ATNA) Integration Profile
 - <http://wiki.ihe.net/index.php?title=ATNA>which is built on top of the following:
- Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications
 - <http://tools.ietf.org/html/rfc3881>
- The Syslog Protocol
 - <http://tools.ietf.org/html/rfc5424>
- Transmission of Syslog Messages over Transport Layer Security (TLS)
 - <http://tools.ietf.org/search/rfc5425>
- Transmission of Syslog Messages over User Datagram Protocol (UDP)
 - <https://tools.ietf.org/html/rfc5426>

Note: The preceding links open documents that deal with the Internet Protocol Suite, specifically Internet Official Protocol Standards (STD1) as related to ARR. They provide critical technical information about secure transmission of data over the internet, including node authentication and an audit trail. Oracle recommends that you read them.

This chapter includes the following sections:

- [Installing the Audit Record Repository](#) on page 6-2
- [Running the Audit Record Repository Server](#) on page 6-2
- [Configuring the Audit Client](#) on page 6-2

Installing the Audit Record Repository

Running the Audit Record Repository Server

Configuring the Audit Client

In order for the IHE Profiles Application to send audit records to the Audit Record Repository Server, you have to properly configure an OHMPI audit client. The OHMPI audit client's configuration file is named by `ohmpi-audit-client.properties`. For the WebLogic application server, this property file is under `domains\<domain_name>\lib`.

There are two properties defined in `ohmpi-audit-client.properties`: `auditHost` and `auditPort`. They are the host name and port number of the ARR server, respectively. The default value for `auditHost` is **localhost**, and it needs to match wherever the ARR server is running. The default value for `auditPort` is **514**, and it needs to match the value of `arr.port` defined in `arr\bin\arr.properties`.

Deploying the IHE Profiles Application

This chapter provides procedures on how to deploy the IHE Profiles Application to the WebLogic application server and how to run HL7 v2 and HL7 v3 sample data through the IHE Profiles Application.

This chapter includes the following section:

- [Deploying an IHE Profiles Application](#) on page 7-1
- [Running Data Through the IHE Profiles Application](#) on page 7-2

Deploying an IHE Profiles Application

Deploy an IHE Profiles Application project on the application server through the server's web administrator console. This starts the IHE service.

To Deploy an EAR File from an Admin Console

1. On the left panel of the WebLogic Server Administration Console, under Domain Structure, select **Environment** and then select **Deployments**.

The Summary of Deployments panel appears.

2. On the right side of the panel under Deployments, click **Install**.

A Summary of Deployments panel with a Deployments table containing a list of EAR files appears.

3. Locate your application EAR and click **Next**.

The Install Application Assistant page appears in the right panel.

4. Locate the deployment you want to install and prepare for deployment.

Tip: Select the file path that represent the application root directory, archive file, exploded archive directory, or application module descriptor that you want to install. You can also enter the path of the application directory or file in the Path field.

Note: Only valid file paths are displayed. If you cannot find your deployment files, upload your file(s) and/or confirm that your application contains the required deployment descriptors.

5. Click **Next**.

Note: When deploying an MPI EAR file through the WebLogic Admin Console, under Security make sure that you select **DD Only**. If you select one of the other options, you will not be able to log into the MIDM.

6. Click **Finish**.
7. Launch **Master Index Data Manager** (MIDM).
8. From a web browser, enter the following:
 - For IHE Profiles Application: <http://localhost:7001/PatientMIDM>
 - For MPI Application: <http://localhost:7001/PersonMIDM>

Note: The "Patient" portion of PatientMIDM or the "Person" portion of PersonMIDM is from the object name that you previously set. "Patient" is mandatory for an IHE Profiles while "Person" could be another object name.

9. Log in using your user name and password.

Running Data Through the IHE Profiles Application

Oracle provides both HL7 v2 and HL7 v3 sample data for you to run through the PIX/PDQ system to help you get started with the applications. To enter the HL7 v2 sample messages, you need an HL7 simulator. There are several available on the internet. The HL7 v3 data is provided in the form of test cases that are run from the soapUI project.

The sample data illustrates several scenarios for data processing. You add new records, update existing records, find assumed matches in the Master Person Index, perform PIX queries for associated local identifiers, perform demographic queries, and perform merges. The data was designed to be run in a specific order to get the expected results from PIX and PDQ queries.

Before you can run these tests, you need to run a database script to add system information to the Master Index Data Manager (MIDM) database, as described in the following section. Without this data, the sample tests will fail.

Note: You can modify the HL7 v2 sample files and the HL7 v3 test input files to insert additional records into the Master Person Index database. If you modify the HL7 v2 sample files, ensure that you use an editor, such as UltraEdit, that does not automatically convert \r to \r\n.

Use the steps in the following procedures to add the required systems to the database, and then to run HL7 V2 and V3 data.

To Add Required Systems to the Database

1. Navigate to the location where you installed OHMPI, and then to `samples\ihe\databasescripts\`.

2. Log in to the mpi database from a SQL prompt or editor as the mpi user.
3. Run the following files against the database:
 - a. `clean_ihe_ohmpi_sample_data.sql`
 - b. `create_ihe_ohmpi_sample_data.sql`

To Run HL7 V2 Sample Data

Before you begin, ensure that:

- Your application server is running, and your OHMPI IHE enterprise application is configured and deployed correctly.
- Your HL7 V2 service lifecycle model is deployed and started correctly.
- Your HL7 v2 client simulator is configured and starts correctly.

You can use any HL7 v2 compliant client to send and receive HL7 v2.5 and v2.3.1 sample data. Configuring the HL7 v2 client simulator is specific to the type of the simulator you select. Generally, you should only need to configure its environment to use HL7 v2 listening URL, which is defined in the `oracle.hsgbu.ohmpi.ihe.hl7v2.url` property for the OHMPI IHE HL7 V2 lifecycle module (by default `hl7://localhost:4447`), or in the `hl7service.xml` configuration file.

Note: Each group of sample data files has a readme file that you should review to learn how each sample data file should be processed and what expected results should be.

1. Navigate to the OHMPI installation directory and then to `samples\ihe\hl7v2`.
This folder contains a complete set of HL7 v2 sample data files. It also contains subdirectories, that each containing a use case. A use case contains a subset of data files that perform specific functions. It contains database scripts for preparing the database to run HL7 v2 sample data. A readme file in the use case directory explains what the use case is and how to run sample data files and the expected results.
2. To process additional records into the system, first backup your sample data files and modify the message data as needed, then use the **HL7 v2 simulator** to complete the procedure.
3. After processing sample data files from a use case, you can:
 - View the response message or the acknowledge message.
 - Review records from an OHMPI Master Person Index using the Master Index Data Manager (MIDM).
 - Review audit records and message tracing, content, and extensions.

To Run HL7 V3 Sample Data

Under the `samples\ihe\hl7v3` directory of your OHMPI installation, there is a soapUI sample project that you can use to test your PIXv3 and PDQv3 web services. A readme file explains the test case scenarios.

Before you begin, ensure that the application server is running, your OHMPI IHE Profile Application has been deployed, and soapUI has been installed properly.

Note: To install soapUI, go to <http://soapui.org>, click **Download Free Open Source**, download the latest version (currently 3.6.1) for your platform (for example, soapui-3.6.1-windows-bin.zip). Unzip the file after downloading it.

For best results, run the tests in the order they are displayed in the project. This shows you a full spectrum of PIX v3/PDQ v3 processing. If the tests are run out of order additional errors will occur because the response from the Master Person Index will be different.

1. Adjust web service URLs in the sample soapUI project, if necessary.

The soapUI sample project provided is based on an IHE Profile Application deployed on WebLogic with default configuration, that is, the PIX Manager v3 and PDQ Supplier v3 web services are assumed to be running at `http://localhost:7001/PIXManager_Soap12/PIXManager_Service` and `http://localhost:7001/PDQSupplier_Soap12/PDQSupplier_Service`, respectively. If your IHE Profile Application is running on a different host or a different port on WebLogic, then you need to adjust the preceding two URLs in the soapUI sample project accordingly. To adjust the URLs, open the `samples\ihe\hl7v3\pixpdqv3-samples-soapui-project.xml` in your favorite editor and replace all instances of the preceding two URLs based on the following instruction:

- The web service URLs must be in the following format (the default port number on WebLogic is **7001**):
 - `http://<host_name>:<port_number>/PIXManager_Soap12/PIXManager_Service`
 - `http://<host_name>:<port_number>/PDQSupplier_Soap12/PDQSupplier_Service`
2. To start soapUI, go to the `${SOAPUI_HOME}\bin` directory and load `soapui.bat`.
 3. To import the PIXV3/PDQV3 soapUI project, click **File**, select **Import Project**, and then select `pixpdqv3-samples-soapui-project.xml` from the `samples\ihe\hl7v3` directory.
 4. Expand the **PIX and PDQ v3 Samples** node in the project tree.
 5. Double click **TestSuite-PIXv3** or **TestSuite-PDQv3** node to show the Test Suite Editor.
 6. Click **Run** to run all the test cases in the test suite.
 7. After running a test suite, you can:
 - Check whether the tests passed or failed within soapUI.
 - Review records using the Master Index Data Manager (MIDM).
 - Review audit records and message tracing.

Monitoring and Maintaining Master Person Index Data

Use the Master Index Data Manager (MIDM) to monitor and maintain your Master Person Index data.

This chapter includes the following section:

- [Using the Master Index Data Manager to Monitor and Maintain Master Person Index Data](#) on page 8-1

Using the Master Index Data Manager to Monitor and Maintain Master Person Index Data

The MIDM provides the following functions so you can maintain the patient records in the Master Person Index database:

- View patient records.
- Compare patient records
- Merge and unmerge patient records.
- Compare records that are flagged as possible duplicates.
- Resolve or merge potential duplicate records.
- Review and reverse automatic record matches made by the Master Person Index.
- Create reports on Master Person Index data and statistics, including records added or updated, potential duplicates, merged and unmerged records, and so on.

Logging in to the Master Index Data Manager

The MIDM is accessed through a web browser and requires you to log in before you can access any patient information. The URL for the MIDM is:

- `http://host:http_port/app_nameMIDM`

where

- *host* is the name of the server machine.
- *http_port* is the HTTP port number of the application server.
- *app_name* is the name of the Master Person Index application.

The default URL for MIDM is <http://localhost:7001/PatientMIDM>.

To Log in to the MIDM

Before you begin, ensure that:

- You have a login ID and password.
 - The application server is running.
1. Launch a web browser.
 2. In the **Address** field, enter the appropriate URL.

The Login page appears.

3. In the top right-hand side of the page, select the language for the MIDM display.
4. Enter your user ID and password in the appropriate fields.
5. Click **Login**.

The initial page appears. (By default, the initial page is the Record Details page, but this is configurable.)

HL7 Message Types Supported by IHE Within OHMPI

This appendix provides a summary of all HL7 message type transactions for PIX v2.3.1, PDQ v2.5, PAM v2.5, PEM v2.5, PDVQ v2.5, PIX v3, and PDQ v3 that Oracle Healthcare Master Person Index (OHMPI) supports.

Supported HL7 v2.3.1, v2.5, and v3 Message Types in OHMPI

The tables in this appendix summarize all the message types of PIX v2.3.1, PDQ v2.5, PAM v2.5, PEM v2.5, PDVQ v2.5, PIX v3, and PDQ v3 transactions that are supported by IHE within OHMPI. See the following tables for information about the IHE Profile name, HL7 message type, and the action performed by the message transactions:

- [PIX v2.3.1 Supported Message Types](#) on page A-1
- [PDQ v2.5 Supported Message Types](#) on page A-2
- [PAM v2.5 Supported Message Types](#) on page A-2
- [PEM v2.5 Supported Message Types](#) on page A-2
- [PDVQ v2.5 Supported Message Types](#) on page A-3
- [PIX v3 Supported Message Types](#) on page A-3
- [PDQ v3 Supported Message Types](#) on page A-3

Table A-1 PIX v2.3.1 Supported Message Types

IHE Profile Name	HL7 Message Type	Action
ITI-8: Patient Identity Feed (v2.3.1)	ADT A01 ADT_A01	Add Patient
	ADT A04 ADT_A01	
	ADT A05 ADT_A05	
	ADT A08 ADT_A01	Update Patient
	ADT A40 ADT_A39	Merge Patients
	ACK	Acknowledgement
ITI-9: PIX Query (v2.5)	QBP Q23 QBP_Q21	Query Message
	RSP K23 RSP_K23	Query Response
ITI-10: PIX Update Notification (v2.5)	ADT A31 ADT_A05	Update Notification
	ACK	Acknowledgement

Table A–2 PDQ v2.5 Supported Message Types

IHE Profile Name	HL7 Message Type	Action
ITI-21: Patient Demographics Query (v2.5)	QBP Q22 (QBP_Q21)	Find Candidates Query
	RSP K22 (RSP_K22)	Query Response
	QBP Q22 (QBP_Q21)	Query Continue
	QCN J01 QCN_J01 ¹	Query Cancel
	ACK	Cancel Acknowledgement

¹ The specification requires three components, while the sample data only contains the first two.

Table A–3 PAM v2.5 Supported Message Types

IHE Profile Name	HL7 Message Type	Action
ITI-30: Patient Identity Management (v2.5)	ADT A28 ADT_A05	Create Patient
	ADT A31 ADT_A05	Update Patient
	ADT A40 ADT_A39	Merge Patients
	ADT A47 ADT_A30	Change Patient Identifier List
	ADT A24 ADT_A24	Link Patient
	ADT A37 ADT_A37	Unlink Patient

Table A–4 PEM v2.5 Supported Message Types

IHE Profile Name	HL7 Message Type	Action
ITI-31: Patient Encounter Management (v2.5)	ADT A01 ADT_A01	Admit Inpatient
	ADT A04 ADT_A01	Register Outpatient
	ADT A03 ADT_A03	Discharge Patient
	ADT A08 ADT_A01	Update Patient Information
	ADT A40 ADT_A39	Merge Patient Identifier List
	ADT A11 ADT_A09	Cancel Admit/Visit
	ADT A13 ADT_A01	Cancel Discharge
	ADT A05 ADT_A05	Pre-admit Patient
	ADT A38 ADT_A38	Cancel Pre-admit
	ADT A06 ADT_A06	Change Patient Class to Inpatient
	ADT A07 ADT_A06	Change Patient Class to Outpatient
	ADT A02 ADT_A02	Transfer Patient
	ADT A12 ADT_A12	Cancel Transfer

Unusual Situation with Outpatient and Inpatient Encounters

The recording of an outpatient encounter when an inpatient encounter is already open (and not ended) for the same patient is not usual. The OHMPI implementation is made on the following lines:

- Discard the outpatient encounter if inpatient encounter is already open.
- Record the outpatient encounter if the inpatient encounter has ended. (that is, the patient has been discharged).

Table A-5 PDVQ v2.5 Supported Message Types

IHE Profile Name	HL7 Message Type	Action
ITI-22: Patient Demographics And Visit Query (v2.5)	QBP_ZV1 (QBP_ZV1)	Find Candidates From Visit Query
	RSP ZV2 (RSP_ZV2)	Visit Query Response
	QCN J01 QCN_J01 ¹	Query Cancel
	ACK	Cancel Acknowledgement

¹ The specification requires three components, while the sample data only contains the first two.

Table A-6 PIX v3 Supported Message Types

IHE Profile Name	HL7 Message Type	Action
ITI-44: Patient Identity Feed (v3)	PRPA_IN201301UV02	Add Patient
	PRPA_IN201302UV02	Update Patient
	PRPA_IN201304UV02	Merge Patients
	MCCI_IN000002UV01	Acknowledgement
ITI-45: PIX Query (v3)	PRPA_IN201309UV02	Query Message
	PRPA_IN201310UV02	Query Response
ITI-46: PIX Update Notification (v3)	PRPA_IN201302UV02	Update Notification
	MCCI_IN000002UV01	Acknowledgement

Table A-7 PDQ v3 Supported Message Types

IHE Profile Name	HL7 Message Type	Action
ITI-47: Patient Demographics Query (v3)	PRPA_IN201305UV02	Find Candidates Query
	PRPA_IN201306UV02	Query Response
	QUQI_IN000003UV01	Query Continue
	QUQI_IN000003UV01	Query Cancel
	MCCI_IN000002UV01	Cancel Acknowledgement

Adding IHE Profile Support to User-defined MPI Object Model

When creating a new IHE Project in OHMPI 3.0, you have an option to use either the out-of-box MPI Project or use a user-defined MPI Project. For the out-of-box MPI Project, OHMPI provides default mapping between the out-of-box MPI Object Model and the IHE Patient Object Model. For user-defined MPI Project, it is your responsibility to provide a customized mapping between the user-defined MPI Object Model and the IHE Patient Object Model.

Creating IHE Project with User-defined MPI Project

This section describes how to add IHE Profile support to user-defined MPI Project.

1. From the NetBeans IDE, click **File** and then select **New Project**.

The New Project dialog box appears.

2. In the Projects section of the Choose Project panel, select **IHE Project** and click **Next**.

The New IHE Project dialog box appears.

3. In the Name and Location panel, perform the following:

- a. In the **Project Name** field, enter a name for your project (for example, IHEProject1).
- b. In the **Project Location** field, browse to the path where you want to create the IHE project.

The Project Folder field populates with the path and name of your project.

- c. From the **Server** drop-down list, select the application server.
- d. From the **Database** drop-down list, select a database (for example, Oracle).
- e. In the **V2 Service Port** field, enter the port number (for example, 4447).
- f. Click **Next**.

The Select MPI Project screen appears.

4. Select the **Select User Defined MPI Project** option and then click **Browse** to select a path of an existing MPI Project.

5. Click **Next**.

The Select Mapper Project screen appears.

6. Select **Create Mapper Project**.

7. Enter the project name in the **Mapper Project Name** field.
8. In the **Mapper Project Location** field, browse to the path where you want to create the Mapper project.
9. Click **Finish**.

When the IHE Project is created, a Java Mapper Project is also created with an empty implementation. The IHE Project is linked to the user-defined MPI Project and the Java Mapper Project, so building the IHE Project should also build the MPI Project and the Java Mapper Project automatically.

Note: You must to run "Generate Master Index Files" on the user-defined MPI Project explicitly. In the case of MPI Project never been generated while creating the IHE Project, the Mapper Project will contain unresolved reference (*mpi-client.jar*) upon creation. That unresolved reference warning is fixed when the "Generate Master Index Files" action is invoked on the MPI Project.

Implementing Customized Mapper Project

Mapper Project Overview

The purpose of the Mapper Project is to provide two-way mapping between the user-defined MPI Object Model and OHMPI's internal IHE Patient Object Model.

Mapping from the IHE Patient Object Model to the MPI Patient Object Model is required by the following IHE transactions:

- Patient Identity Feed
- Patient Demographics Query (Request)

Mapping from the MPI Patient Object Model to the IHE Patient Object Model is required by the following IHE transactions:

- Patient Demographics Query (Response)
- PIX Update Notification

Java programming knowledge is required to implement the mapping.

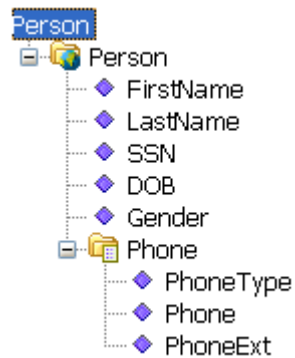
For OHMPI's internal IHE Patient Object Model, see *samples\ihe\javadoc\index.html* under your OHMPI installation directory for the Javadoc. Package *oracle.hsgbu.ohmpi.model* contains the IHE Object Model that you must map from and to your MPI Object Model.

Also, see Javadoc for the *oracle.hsgbu.ohmpi.services.IMapperService* Interface and understand the three methods you must implement in your mapper implementation class.

Mapper Example

In this section, we use a simplified version of the Person Template Project as an example to illustrate the mapping process.

The following figure shows the Object Model of a simplified Person MPI Project.

Figure B-1 Object Model

The following table shows the field mapping that we choose to implement: (Please note that *unformattedTelephoneNumber* in the IHE Patient Object Model only gets populated from HL7v3 Patient Identity Feed. To support data coming from HL7 v2 Patient Identity Feed, you need to check the other fields under the IHE Model's Telecom object as well when doing the mapping. We skip that part here for simplicity.)

User-defined MPI Object Model	IHE Patient Object Model
<i>FirstName</i>	<i>givenName</i>
<i>LastName</i>	<i>familyName</i>
<i>SSN</i>	<i>SSN</i>
<i>DOB</i>	<i>dateOfBirth</i>
<i>Gender</i>	<i>gender</i>
<i>Phone.PhoneType</i>	<i>Telecom.telecomUseCode</i>
<i>Phone.Phone</i>	<i>Telecom.unformattedTelephoneNumber</i>
<i>Phone.PhoneExt</i>	<i>Telecom.extension</i>

The following is a sample implementation of the *IMapperService* interface based on the field mapping we identified above:

```

package com.example;

import com.sun.mdm.index.objects.exception.ObjectException;
import com.sun.mdm.index.objects.person.PersonObject;
import com.sun.mdm.index.objects.person.PhoneObject;
import java.util.Collection;
import java.util.logging.Logger;
import oracle.hsgbu.ohmpi.commons.PatientName;
import oracle.hsgbu.ohmpi.model.IHEPatientObject;
import oracle.hsgbu.ohmpi.model.IHETelecomObject;
import oracle.hsgbu.ohmpi.services.IMapperService;

/**
 * Customized mapper implementation to bridge the standard IHE Patient Object
 * Model and customized MPI Object Model.
 */
public class MyMapperImpl implements IMapperService {
  
```

```

private static final Logger logger =
Logger.getLogger(MyMapperImpl.class.getName());

public IHEPatientObject getIHEPatientObject(Object mpiPatientObject) throws
Exception {
    IHEPatientObject ihePO = new IHEPatientObject();

    PersonObject mpiPO = (PersonObject) mpiPatientObject;

    ihePO.setGivenName(mpiPO.getFirstName());
    ihePO.setFamilyName(mpiPO.getLastName());
    ihePO.setGender(mpiPO.getGender());
    ihePO.setSSN(mpiPO.getSSN());
    ihePO.setDateOfBirth(mpiPO.getDOB());

    if (mpiPO.getPhone() != null) {
        for (PhoneObject mpiPhone : (Collection<PhoneObject>) mpiPO.getPhone()) {
            IHETelecomObject ihePhone = new IHETelecomObject();
            ihePhone.setExtension(mpiPhone.getPhoneExt());
            ihePhone.setTelecomUseCode(mpiPhone.getPhoneType());
            ihePhone.setUnformattedTelephoneNumber(mpiPhone.getPhone());
            ihePO.addTelecom(ihePhone);
        }
    }

    return ihePO;
}

public Object getMPIPatientObject(IHEPatientObject ihePO) throws Exception {
    PersonObject mpiPO = null;

    mpiPO = new PersonObject();
    mpiPO.setFirstName(ihePO.getGivenName());
    mpiPO.setLastName(ihePO.getFamilyName());
    mpiPO.setGender(ihePO.getGender());
    mpiPO.setSSN(ihePO.getSSN());
    mpiPO.setDOB(ihePO.getDateOfBirth());

    if (ihePO.getTelecoms() != null) {
        for (IHETelecomObject ihePhone : ihePO.getTelecoms()) {
            PhoneObject mpiPhone = new PhoneObject();

            mpiPhone.setPhoneExt(ihePhone.getExtension());
            mpiPhone.setPhoneType(ihePhone.getTelecomUseCode());
            mpiPhone.setPhone(ihePhone.getUnformattedTelephoneNumber());

            mpiPO.addPhone(mpiPhone);
        }
    }

    return mpiPO;
}

// Convenience method to avoid a full-blown mapping from MPI PO to IHE PO

```

```

public PatientName getPatientName(Object mpiPatientObject) throws Exception {

    PersonObject po = (PersonObject) mpiPatientObject;
    return new PatientName(po.getLastName(), po.getFirstName(), null, null);

}
}

```

Value Mapping

The sample mapper implementation in [Mapper Example](#) on page B-2 only shows field mapping. However, if any code list is defined in the MPI Object Model, then the mapper implementation also needs to take care of value mapping. For example, if the user-defined MPI Object Model's *Gender* field has a code list of "1" / "2" / "0", where "1" represents Male, "2" represents Female and "0" represents Unknown, these values must be properly mapped to HL7 code list ("M" / "F" / "UN") for IHE's Patient Demographics Query, and vice versa for IHE's Patient Identity Feed.

MPI's Gender Codes	IHE HL7's Sex Codes	Comment
1	M	Male
2	F	Female
0	UN	Unknown

To support this value mapping, in the *getIHEPatientObject()* implementation, replace *ihePO.setGender(mpiPO.getGender());*

by the following code snippet:

```

String mpiGender = mpiPO.getGender();
if ("1".equals(mpiGender)) {
    ihePO.setGender("M");
} else if ("2".equals(mpiGender)) {
    ihePO.setGender("F");
} else {
    ihePO.setGender("UN");
}

```

Also, in *getMPIPatientObject()* implementation, replace

mpiPO.setGender(ihePO.getGender());

by the following code snippet:

```

String iheGender = ihePO.getGender();
if ("M".equals(iheGender)) {
    mpiPO.setGender("1");
} else if ("F".equals(iheGender)) {
    mpiPO.setGender("2");
} else {
    mpiPO.setGender("0");
}

```

Required Fields

In the user-defined MPI Object Model, some fields might be defined as required fields. When a required MPI field does not have corresponding value in the IHE Patient Feed, your mapper implementation may choose to throw *oracle.hsgbu.ohmpi.exceptions.ValidationException* to reject the Patient Feed, or it may

choose to provide some default value in *getMPIPatientObject()* implementation in order to successfully process the IHE Patient Identity Feed and persist the patient record into the MPI database.

Mapping Context

In some scenario, your mapper implementation might need to have a better understanding of the mapping context or need to get more information about the underlying MPI system. For this purpose, a new *oracle.hsgbu.ohmpi.services.AbstractMapperImpl* class is introduced to provide such support.

For example, the mapping from the IHE Patient Object to the MPI Object (*getMPIPatientObject()*) is used for both Patient Identity Feed and PDQ. (It is used in PDQ because a MPI template Patient Object instance needs to be constructed based on the search criteria specified in the PDQ request.). There might be a need to determine whether the current mapping is performed inside a Patient Identity Feed transaction or not. For example, if some field (say, *gender*) is required by the MPI Object Model but is missing in the Patient Identity Feed, the mapper implementation might choose to provide some default value (for example, Unknown). On the other hand, if the current transaction is PDQ, then no default value should be provided in the mapping, otherwise, it changes the search criteria.

To differentiate Patient Identity Feed and PDQ in the user-defined mapper implementation, instead of implementing the *oracle.hsgbu.ohmpi.services.IMapperService* interface, change your mapper implementation class to extend *oracle.hsgbu.ohmpi.services.AbstractMapperImpl* instead, then call the helper method *isInPatientIdentityFeed()* to determine whether the current transaction is a Patient Identity Feed or not.

Code System Support

The IHE Patient Object Model provides code system support. For example, *IHEPatientObject* provides the following 4 APIs to get/set patient's gender:

- *String getGender()*
- *setGender(String gender)*
- *IHECodeObject getGenderCode()*
- *setGenderCode(IHECodeObject genderCode)*

The first two APIs are provided for convenience. If code system is not important in your application, your mapper implementation can simply use the first two APIs to deal with gender values (for example, M, F). On the other hand, if the code system is important in your application, your mapper implementation can have complete code system access using the last two APIs. An *IHECodeObject* simply represents a code value under some code system, for example, "M" under code system "2.16.840.1.113883.5.1".

As a side note, for simplicity, codes are not scoped by namespaces in the out-of-box MPI Object Model. For example, *gender* is a simple attribute (instead of coded value with namespace) on the *Patient* Object, and *Race* is a simple Child Object with only one attribute (instead of coded value with namespace).

Configuring IHE Project

Edit the following properties in IHE Project's *services.properties* under "Configuration Files" before building the IHE Project.

```
# Application Name of the User-defined MPI Project
mpi/application/name=Person
# Primary Object Name in the Object Model of the User-defined MPI Project
mpi/primary_object/name=Person
# Primary Object's Fully-Qualified Class Name in the User-defined MPI Project
mpi/primary_object/class_name=com.sun.mdm.index.objects.person.PersonObject
# Fully Qualified Class Name of the IHE-MPI Mapper Service Implementation
service/mapper/class_name=com.example.MyMapperImpl
```

Adjust the above property values accordingly based on your own MPI Project and Mapper Project's implementation.

