



Informatica® Cloud Data Integration
February 2025

Industry Solutions for Healthcare

Informatica Cloud Data Integration Industry Solutions for Healthcare
February 2025

© Copyright Informatica LLC 2024, 2025

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, Informatica Cloud, Informatica Intelligent Cloud Services, PowerCenter, PowerExchange, and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2025-02-07

Table of Contents

Preface	5
Chapter 1: Industry solutions for healthcare	6
Industry solutions components.	6
Chapter 2: Exchange electronic health information through HL7 2.x	9
Receive patient and health information.	9
Step 1. Create the source connection.	10
Step 2. Create the target connection.	10
Step 3. Configure the mapping.	10
Step 4. Create and run a mapping task.	12
Send patient and health information.	12
Step 1. Download the XSD file for the serializer data service.	12
Step 2. Create the source connection.	12
Step 3. Create the target connection.	13
Step 4. Create the mapping.	13
Step 5. Create and run a mapping task.	14
Processing HL7 messages using MLLP.	14
Chapter 3: Exchange electronic health information through HIPAA	16
Receive patient and health information.	16
Step 1. Create the source connection.	17
Step 2. Create the target connection.	17
Step 3. Configure the mapping.	17
Step 4. Create and run a mapping task.	18
Send patient and health information.	19
Step 1. Download the XSD file for the serializer data service.	19
Step 2. Create the source connection.	19
Step 3. Create the target connection.	20
Step 4. Create the mapping.	20
Step 5. Create and run a mapping task.	20
Chapter 4: Exchange electronic health information through FHIR	21
Chapter 5: Bridge between HL7 2.x and FHIR	23
Step 1. Create the source connection.	23
Step 2. Create the target connection.	23
Step 3. Configure the mapping.	24
Step 4. Create and run a mapping task.	25

Chapter 6: Using HL7 for data analysis.....	26
Step 1. Create the source connection.	26
Step 2. Create the target connections.	26
Step 3. Configure the mapping.	27
Step 4. Create and run a mapping task.	28
Chapter 7: Using FHIR for data analysis.....	30
Step 1. Create the source connection.	30
Step 2. Create the target connections.	30
Step 3. Configure the mapping.	31
Step 4. Create and run a mapping task.	33

Preface

Use *Industry Solutions for Healthcare* to learn how to receive, process, validate, and transform data using complex healthcare data standards such as HL7, HIPAA, and FHIR.

CHAPTER 1

Industry solutions for healthcare

Industry solutions for healthcare and life sciences deliver advanced, cloud-native data services to enable and support healthcare data standards. They're designed to help receive, process, validate, and transform data using complex healthcare data standards such as HL7 and HIPAA.

Healthcare and life sciences organizations can use pre-built data services, templates, and healthcare-specific connectors to expedite common use cases in the healthcare industry. Organizations can efficiently ingest, process, manage, master, and access EMR data and claims data. The data can be used in operational or analytical use cases in batch and in real time to overcome the data and technology barriers to interoperability.

Organizations can leverage industry solutions for healthcare to handle healthcare data in the following use cases:

- Exchange electronic health information through HL7
- Exchange electronic health information through HIPAA
- Exchange electronic health information through FHIR
- Bridge between HL7 2.x and FHIR
- Use HL7 for data analysis
- Use FHIR for data analysis

Industry solutions components

Industry solutions for healthcare use multiple components in Informatica Intelligent Cloud Services.

Industry solutions use the following components:

- Data services repository
- Data services API
- Industry data service customizer
- Mapping templates
- Connections
- Data Services transformation
- Hierarchical mappers
- Intelligent structure models
- Mapping tasks

Data services repository

The data services repository is a central repository of pre-built data services that process, validate, and transform industry-standard messages. You can also upload your own services to the repository.

The name of a data service indicates the industry standard, the version of the standard, the message type, and the service usage type, in the following syntax:

```
<industry standard>_<version>_<message type>_<usage type>
```

A parser data service receives and parses messages from your partners while a serializer data service serializes and sends messages to your partners. You can also use a restricted serializer data service that uses stricter rules to validate HIPAA messages before sending them to your partners.

You can view, create, edit, and delete data services on the **Data Services Repository** page in Administrator.

For more information, see [Data Services Repository](#) in the Administrator help.

Data services API

The data services API runs data services using Informatica Intelligent Cloud Services REST API. You pass a file to the API, and a data service processes it.

For more information, see [Data services](#) in the *REST API Reference*.

Industry data service customizer

You can use the industry data service customizer to customize an industry-standard message and publish it as a custom data service to the data services repository.

A customizer provides out-of-the-box message types based on the messaging standard. You select a message type and customize the message structure by adding, editing, and removing elements.

For more information, see [Industry data service customizer](#) in *Components*.

Mapping templates

A mapping template is a pre-made mapping that contains pre-populated transformations. You can click each transformation in the mapping to see the purpose of the transformation, how the transformation is configured, and which parameters are used.

A set of mapping templates for common use cases is available under the Industry Solutions category. The templates are pre-configured and include the common logic needed to handle a specific use case using data services. Each template includes a Data Services transformation that uses a data service from the data services repository to parse incoming messages and output the data in XML format.

For more information, see [Mapping templates](#) in *Mappings*.

Connections

Connections are used to access data in cloud and on-premises applications, platforms, databases, and flat files. They specify the location of sources, lookup objects, and targets. For industry solutions, Data Integration offers several industry-specific connections such as FHIR REST-based connections and MLLP connections.

You use connectors to create connections. You can create a connection for any connector that is installed in your organization.

To create a connection, use the **Connections** page in Administrator. The **Connections** page lists all connections that have been configured in the organization.

For more information, see [Connections](#) in the Administrator help.

Data Services transformation

You can use the Data Services transformation to run a data service from the data service repository within a mapping in Data Integration. The data service parses or serializes industry-standard messages in the mapping flow. For example, you can use a data service to parse and serialize HL7 messages that you exchange with your healthcare partners.

When you define the data service in the transformation, you can use one of the following options:

- Use a specific data service name to process a single message type for a specific standard and usage type.
- Use a dynamic service name to process messages for multiple message types from different standards and usage types.

The Data Services transformation passes the processed message data to the downstream transformation in a file or as a buffer. It can create additional output fields based on the status tracing level and the data service usage type and send different output types to different targets. For example, you can send message data to one target and send status tracing reports to another target.

For more information, see [Data Services transformation](#) in *Transformations*.

Hierarchical mappers

A hierarchical mapper maps a hierarchical schema, such as the schema for an industry-standard message, to another hierarchical schema. For example, you can map the XML output of a parser data service to a target XSD file.

You can use a hierarchical mapper in a Data Services transformation to process hierarchical input and convert it to hierarchical output that uses a different hierarchical structure.

For more information, see [Hierarchical mappers](#) in *Components*.

Intelligent structure models

A CLAIRE® intelligent structure model is an asset that determines the underlying patterns and structures of the input that you provide, and it can be used to parse long, complex files with little or no structure.

Intelligent structure models can provide pre-built mappers that convert XML data to relational format. For example, you can find pre-built mappers that convert HL7 and FHIR data to relational structures.

For more information, see [Intelligent structure models](#) in *Components*.

Mapping task

A mapping task runs the data flow logic defined in a mapping. After you create a mapping, you create and run a mapping task to process the data.

For more information, see [Mapping tasks](#) in *Tasks*.

CHAPTER 2

Exchange electronic health information through HL7 2.x

Industry solutions for healthcare include a set of pre-built data services to help healthcare and life science organizations exchange data through the HL7 2.x messaging standard. The pre-built data services simplify and expedite the handling of HL7 data for operational and analytical use cases through Informatica's data integration and data ingestion capabilities.

You can use the pre-built data services to process and exchange data through HL7 data standards by using the pre-built parsers and serializers for receiving and sending data. For example, you can receive data from different EMRs, healthcare providers, and other sources using the MLLP client and server connectivity. You can also process and transform patient and other data received through HL7 data services to XML format and write the data to any target application, data warehouse, or data lake.

This section addresses the following scenarios:

- Receiving patient and health information from HL7. You can receive HL7 messages and convert the data to XML format.
- Sending patient and health information. You can provide message data in XML format and translate it to an HL7 message.

Receive patient and health information

To receive patient and health information through the HL7 messaging standard, you can use a mapping template that's available in Data Integration. The mapping template requires source and target connections to read the HL7 messages. To run the mapping, you'll need to run a mapping task.

Perform the following tasks:

1. Create a source connection.
2. Create a target connection.
3. Create a mapping using a mapping template. In Data Integration, use the mapping template called "Process incoming HL7 messages."
4. Create and run a mapping task.

Step 1. Create the source connection

To receive HL7 messages, configure a source connection to access the HL7 data.

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- A Kafka connection through an MLLP server to read HL7 data from a messaging queue

To enable and configure an MLLP server, use the **File Servers** page in Administrator. Set the target type to write the HL7 data to a Kafka queue. For more information, see *File Transfer* in the Administrator help.

Step 2. Create the target connection

To write the XML data to a target, configure a target connection.

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- A relational connection if the relational database can store large objects that can hold XML data
- An MLLP connection to upload the XML data to a messaging queue

If you use an MLLP connection, you need to enable and configure an MLLP server before you create the connection.

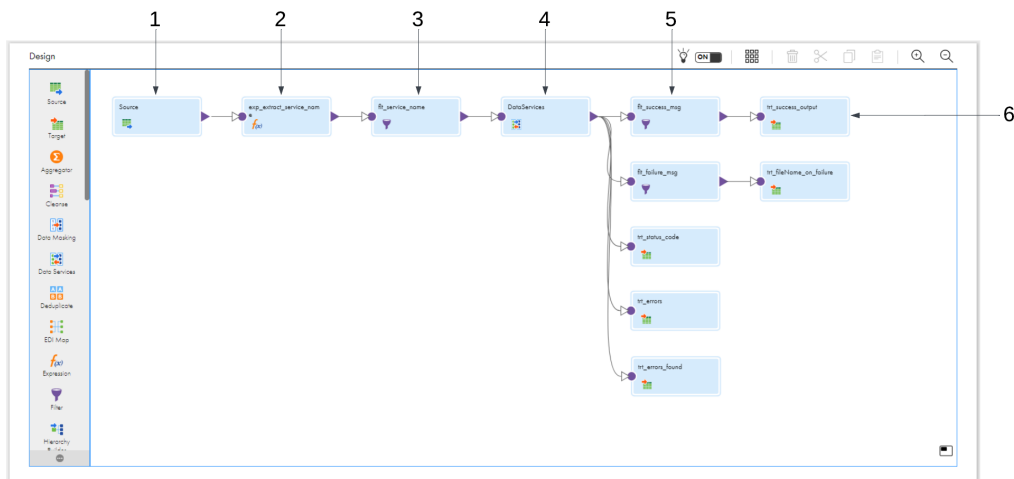
To enable and configure the MLLP server, use the **File Servers** page in Administrator. You can set the target type to write HL7 messages to a flat file or to a Kafka queue. For more information, see *File Transfer* in the Administrator help.

Step 3. Configure the mapping

In Data Integration, use the mapping template called "Process incoming HL7 messages" and configure the mapping.

The mapping template uses a Data Services transformation that accesses a pre-built data service from the data services repository to parse HL7 messages. You can populate the mapping template with source and target data using the source and target connections that you created.

The following image shows the transformations in the mapping template:



The mapping template contains the following transformations:

1. Source transformation

The Source transformation reads an HL7 file as a source. You can use a flat file connection, an SFTP connection to a remote server, or a Kafka connection to access the HL7 data.

The source can be one of the following types:

- Single object
- File list to process multiple files
- Command if the files are stored on the same Linux machine that runs the Secure Agent
- Parameter if you want to specify the source in the mapping task

The source is parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameter with a specific connection and source object in the Source transformation.
- Leave the parameter in the Source transformation and specify a source in the mapping task or use a parameter file.

2. Expression transformation

The Expression transformation uses the data in the HL7 file to construct the name of the data service.

Data services use the following naming convention:

```
<industry standard>_<version>_<message type>_<usage type>
```

For example, hl7_2_6_adt_A01_parser.

3. Filter transformation

The Filter transformation filters the incoming data from the Expression transformation to output only the final name of the data service that the Expression transformation constructs.

4. Data Services transformation

The Data Services transformation uses the name of the HL7 parser data service to access the data service in the data services repository. The HL7 parser reads the incoming patient and health information in the HL7 file and converts the data to XML format.

The mapping template sets the output type to buffer by default. If the output data is larger than the output precision, change the output type from buffer to file to avoid truncating the data.

5. Filter transformations

The Filter transformations filter the upstream fields to pass only the relevant fields to each target.

6. Target transformations

The Target transformations write the successful messages, failure messages, status codes, errors, and error flags to the target files. To write the data to the target files, you can use any target connection that writes XML data, such as a flat file, relational, or MLLP connection.

The target is parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameter with a specific connection and target object in the Target transformation.
- Leave the parameter in the Target transformation and specify a target in the mapping task or use a parameter file.

Step 4. Create and run a mapping task

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

If you left the source and target parameters in the mapping template, you can configure the source and target in the mapping task.

In the mapping task, the **Sources** page appears if the mapping includes source parameters. You can add a single source object or multiple source objects based on the connection type and the mapping configuration. You can also configure a source filter. Similarly, the **Targets** page appears if the mapping includes target parameters.

Send patient and health information

Industry solutions for healthcare include pre-built data services that allow you to provide message data in XML format and translate it to an HL7 message. These data services are called serializers and you can find the XSD file that each serializer requires in the data services repository.

To send patient data or any other healthcare data through HL7, you create a mapping that generates the required XML structure and passes it to the Data Services transformation. The Data Services transformation uses the appropriate serializer data service, such as HL7_2_1_ADT_A01_Serializer, to format and output the HL7 message. Then, you write the message to a target.

Perform the following tasks:

1. Download the XSD file for the serializer data service.
2. Create a source connection.
3. Create a target connection.
4. Create a mapping.
5. Create and run a mapping task.

Step 1. Download the XSD file for the serializer data service

To identify the XML structure that the HL7 serializer data service requires to convert XML data to an HL7 message, download the XSD file for the serializer data service from the data services repository in Administrator.

To determine the name of the serializer data service, you can find the information in a sample HL7 file. The data service name uses the following format:

```
<industry standard>_<version>_<message type>_<usage type>
```

For example, HL7_2_1_ADT_A01_Serializer.

When you download the serializer data service, use the XSD file to determine the XML structure that the data service requires.

Step 2. Create the source connection

To read the XML data, create a source connection that accesses the XML files.

If the data is already formatted in the expected XML structure, you can create any source connection that reads an XML file as a source, such as a flat file connection or an SFTP connection to a remote server.

If the data is in relational format, create a source connection that reads a relational source. In this case, you also need to create a mapping that uses a Hierarchy Builder transformation to map the relational data to the XML structure that the serializer data service requires.

Step 3. Create the target connection

To send the HL7 messages, create a target connection to write the HL7 messages to a target.

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- An MLLP connection that uses an MLLP server to write the HL7 messages to a flat file

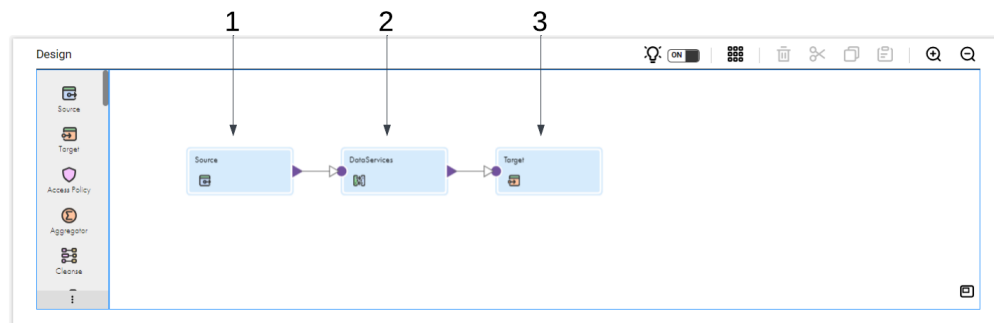
If you use an MLLP connection, you need to enable and configure an MLLP server before you create the connection.

To enable and configure the MLLP server, use the **File Servers** page in Administrator. You can set the target type to write HL7 messages to a flat file or to a Kafka queue. For more information, see *File Transfer* in the Administrator help.

Step 4. Create the mapping

In Data Integration, create a mapping to read the XML data and write the data as HL7 messages to a target.

The following image shows the data flow in the mapping:



The mapping contains the following transformations:

1. Source transformation

The Source transformation reads an XML file as a source. You can use a flat file connection or an SFTP connection to a remote server.

2. Data Services transformation

The Data Services transformation uses an HL7 serializer data service from the data services repository. The HL7 serializer reads the incoming patient and health information in the XML file and converts the data to an HL7 message.

You can get the name of the HL7 serializer data service by using the following format:

```
<industry standard>_<version>_<message type>_<usage type>
```

For example, HL7_2_6_ADT_A12_Serializer.

3. Target transformation

The Target transformation writes the HL7 messages to a target. You can use a flat file connection, an SFTP connection to a remote server, or an MLLP connection to write the HL7 messages to a flat file.

Step 5. Create and run a mapping task

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

Processing HL7 messages using MLLP

The Minimal Lower Layer Protocol (MLLP) protocol is used to transfer HL7 messages. To receive HL7 messages from external sources using MLLP, you enable the MLLP server in your organization. In the MLLP server, you can define how to transfer the message: either through a flat file or through a Kafka queue.

When you receive an HL7 message through the MLLP server, you can write the message to a Kafka queue or to a flat file for further processing. Create the appropriate flat file or Kafka connection and set it as the target type for the MLLP server. Then, to send HL7 messages through MLLP, you can create an MLLP connection and use it as a target in a mapping as part of an outbound flow.

To use MLLP, complete the following tasks:

1. Enable and configure an MLLP server.

To transfer HL7 messages through the MLLP protocol, configure an MLLP server so that partners can connect the server, upload files to the server, and download files from the server.

To enable and configure an MLLP server, use the **File Servers** page in Administrator. The target type can be a flat file or a Kafka queue. For more information, see *File Transfer* in the Administrator help.

Note: An MLLP server can transfer only HL7 messages, so the flat files or the Kafka queue must contain HL7 messages.

2. Optionally, create a Kafka connection

Create a Kafka connection in the following scenarios:

- If the MLLP server transfers HL7 messages using a Kafka queue, create a Kafka connection and specify it in the MLLP server properties.
- You want to process HL7 messages using a Kafka queue in a mapping.

3. Optionally, create a flat file connection.

Create a flat file connection if you want to process HL7 messages using a flat file in a mapping.

4. Create an MLLP connection.

An MLLP connection passes the HL7 messages it receives from a third party to the MLLP server.

To create an MLLP connection, use the **Connections** page in Administrator. For more information about the connection properties, see *Connections* in the Administrator help.

You can use an MLLP connection only to write HL7 messages to a target in a mapping.

5. Create a mapping.

To process HL7 messages through the MLLP server, create a mapping. In the Source transformation, use the connection that you specified as the target type in the MLLP server. For example, if the target

type for the MLLP server is a Kafka queue, then use the Kafka connection in the Source transformation. This mapping will read the HL7 messages that the MLLP server wrote to the Kafka queue and process the messages according to the mapping logic.

Note: When a mapping uses a Kafka queue as a source, the mapping runs in real time and it'll continue running until you stop the job.

To send an HL7 message using MLLP, use the MLLP connection in the Target transformation.

HL7 messages that are transferred through an MLLP server can also generate acknowledgements.

CHAPTER 3

Exchange electronic health information through HIPAA

Industry solutions for healthcare include a set of pre-built data services to help healthcare and life science organizations exchange data through the HIPAA messaging standard. The pre-built data services simplify and expedite the handling of HIPAA data for operational and analytical use cases through Informatica's data integration and data ingestion capabilities.

You can use the pre-built data services to process and exchange data through HIPAA data standards by using the pre-built parsers and serializers for receiving and sending data. For example, you can receive data from different EMRs, healthcare providers, and other sources. You can also process and transform patient and other data received through HIPAA data services to XML format and write the data to any target application, data warehouse, or data lake.

This section addresses the following scenarios:

- Receiving patient and health information from HIPAA. You can receive HIPAA messages and convert the data to XML format.
- Sending patient and health information. You can provide message data in XML format and translate it to a HIPAA message.

Receive patient and health information

To receive patient and health information through the HIPAA messaging standard, you can use a mapping template that's available in Data Integration. The mapping template requires source and target connections to read the HIPAA messages. To run the mapping, you'll need to run a mapping task.

Perform the following tasks:

1. Create a source connection.
2. Create a target connection.
3. Create a mapping using a mapping template. In Data Integration, use the mapping template called "Process incoming X12 HIPAA messages."
4. Create and run a mapping task.

Step 1. Create the source connection

To receive HIPAA messages, configure a source connection to access the HIPAA data.

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- A Kafka connection to read HIPAA data from a messaging queue

Step 2. Create the target connection

To write the XML data to a target, configure a target connection.

You can use one of the following types of connections:

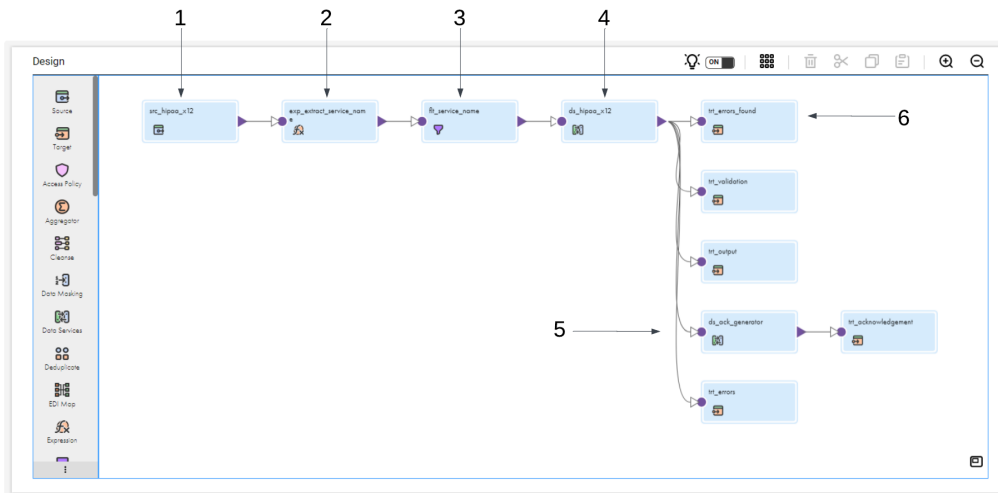
- A flat file or an SFTP connection to a remote server
- A relational connection if the relational database can store large objects that can hold XML data

Step 3. Configure the mapping

In Data Integration, use the mapping template called "Process incoming X12 HIPAA messages" and configure the mapping.

The mapping template uses a Data Services transformation that accesses a pre-built data service from the data services repository to parse HIPAA messages. You can populate the mapping template with source and target data using the source and target connections that you created.

The following image shows the transformations in the mapping template:



The mapping template contains the following transformations:

1. Source transformation

The Source transformation reads an HIPAA file as a source. You can use a flat file connection, an SFTP connection to a remote server, or a Kafka connection to access the HIPAA data.

The source can be one of the following types:

- Single object

- File list to process multiple files
- Command if the files are stored on the same Linux machine that runs the Secure Agent
- Parameter if you want to specify the source in the mapping task

The source is parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameter with a specific connection and source object in the Source transformation.
- Leave the parameter in the Source transformation and specify a source in the mapping task or use a parameter file.

2. Expression transformation

The Expression transformation uses the data in the HIPAA file to construct the name of the data service.

Data services use the following naming convention:

```
<industry standard>_<version>_<message type>_<usage type>
```

For example, HIPAA_5010A1_270_Parser.

3. Filter transformation

The Filter transformation filters the incoming data from the Expression transformation to output only the final name of the data service that the Expression transformation constructs.

4. Data Services transformation to invoke a parser data service

The Data Services transformation uses the name of the HIPAA parser data service to access the data service in the data services repository. The HIPAA parser reads the incoming patient and health information in the HIPAA file and converts the data to XML format.

The mapping template sets the output type to buffer by default. If the output data is larger than the output precision, change the output type from buffer to file to avoid truncating the data.

The input fields include L1_Validation and L2_Validation. You can set the value of these fields to true to generate the corresponding validation reports.

5. Data Services transformation to generate acknowledgements

The Data Services transformation invokes the ACK_999_Generator data service from the data service repository. The acknowledgement generator parses the XML structure that the previous Data Services transformation generated, validates it, and create an acknowledgement message.

6. Target transformations

The Target transformations write the output, validation report, errors, and error flags to the target files. To write the data to the target files, you can use any target connection that writes XML data, such as a flat file or relational connection.

The target is parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameter with a specific connection and target object in the Target transformation.
- Leave the parameter in the Target transformation and specify a target in the mapping task or use a parameter file.

Step 4. Create and run a mapping task

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

If you left the source and target parameters in the mapping template, you can configure the source and target in the mapping task.

In the mapping task, the **Sources** page appears if the mapping includes source parameters. You can add a single source object or multiple source objects based on the connection type and the mapping configuration. You can also configure a source filter. Similarly, the **Targets** page appears if the mapping includes target parameters.

Send patient and health information

Industry solutions for healthcare include pre-built data services that allow you to provide message data in XML format and translate it to a HIPAA message. These data services are called serializers and you can find the XSD file that each serializer requires in the data services repository.

To send patient data or any other healthcare data through HIPAA, you create a mapping that generates the required XML structure and passes it to the Data Services transformation. The Data Services transformation uses the appropriate serializer data service, such as HIPAA_5010A1_270_Serializer, to format and output the HIPAA message. Then, you write the message to a target.

Perform the following tasks:

1. Download the XSD file for the serializer data service.
2. Create a source connection.
3. Create a target connection.
4. Create a mapping.
5. Create and run a mapping task

Step 1. Download the XSD file for the serializer data service

To identify the XML structure that the HIPAA serializer data service requires to convert XML data to a HIPAA message, download the XSD file for the serializer data service from the data services repository in Administrator.

To determine the name of the serializer data service, you can find the information in a sample HIPAA file. The data service name uses the following format:

```
<industry standard>_<version>_<message type>_<usage type>
```

For example, HIPAA_5010A1_270_Serializer.

When you download the serializer data service, use the XSD file to determine the XML structure that the data service requires.

Step 2. Create the source connection

To read the XML data, create a source connection that accesses the XML files.

If the data is already formatted in the expected XML structure, you can create any source connection that reads an XML file as a source, such as a flat file connection or an SFTP connection to a remote server.

If the data is in relational format, create a source connection that reads a relational source. In this case, you also need to create a mapping that uses a Hierarchy Builder transformation to map the relational data to the XML structure that the serializer data service requires.

Step 3. Create the target connection

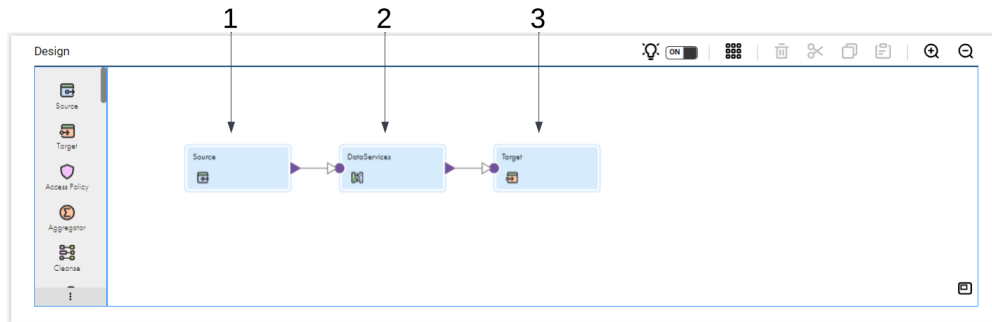
To send the HIPAA messages, create a target connection to write the HIPAA messages to a target.

You can use a flat file or an SFTP connection to a remote server.

Step 4. Create the mapping

In Data Integration, create a mapping to read the XML data and write the data as HIPAA messages to a target.

The following image shows the data flow in the mapping:



The mapping contains the following transformations:

1. Source transformation

The Source transformation reads an XML file as a source. You can use a flat file connection or an SFTP connection to a remote server.

2. Data Services transformation

The Data Services transformation uses a HIPAA serializer data service from the data services repository. The HIPAA serializer reads the incoming patient and health information in the XML file and converts the data to an HIPAA message.

You can get the name of the HIPAA serializer data service by using the following format:

```
<industry standard>_<version>_<message type>_<usage type>
```

For example, HIPAA_5010A1_270_Serializer.

3. Target transformations

The Target transformation writes the HIPAA messages to a target. You can use a flat file connection or an SFTP connection to a remote server to write the HIPAA messages to a flat file.

Step 5. Create and run a mapping task

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

CHAPTER 4

Exchange electronic health information through FHIR

Industry solutions for healthcare include a set of pre-built data services to help healthcare and life science organizations exchange data through the FHIR messaging standard. The pre-built data services simplify and expedite the handling of FHIR data for operational and analytical use cases through Informatica's data integration and data ingestion capabilities.

To process and exchange data through the FHIR data standard, you can use the FHIR connector and the pre-built FHIR validation service. You can process and transform patient and other data received through FHIR and write the data to any target application, data warehouse, or data lake. You can also write data to a FHIR server.

For example, to send patient and health information through the FHIR data standard, complete the following tasks:

1. Create a source connection.

You can create a flat file or SFTP connection to a remote server to use XML data as a source.

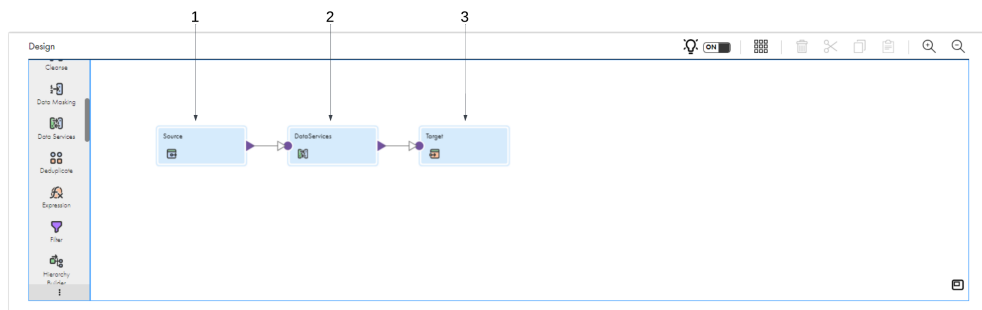
2. Create a target connection.

To write the data as FHIR bundles to a target, create a FHIR connection and configure the FHIR server properties.

3. Create a mapping.

In Data Integration, create a mapping to read the XML data as a source and write the data as FHIR bundles to a FHIR server.

The following image shows the data flow in the mapping:



The mapping contains the following transformations:

1. The Source transformation reads XML data using the flat file or SFTP connection that you created in step 1.
2. The Data Services transformation uses the FHIR validation service in the data services repository to validate the XML data based on the FHIR data standard.

You can configure the FHIR validation service on the **Data Services** tab. Select **FHIR Validation** as the service type. Then select the FHIR version and optionally enter the URL for the terminology server.

3. The Target transformation posts the XML data as FHIR bundles to the FHIR server using the FHIR connection that you created in step 2.

4. Create and run a mapping task.

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

To read data on a FHIR server, you can create a similar mapping that uses a FHIR connection in the Source transformation to read FHIR bundles on the FHIR server and a flat file connection in the Target transformation to write the XML data to a flat file.

CHAPTER 5

Bridge between HL7 2.x and FHIR

Industry solutions for healthcare include a set of pre-built data services to help healthcare and life science organizations exchange data through the HL7 2.x and FHIR messaging standards. The pre-built data services simplify and expedite the handling of HL7 and FHIR data for operational and analytical use cases through Informatica's data integration and data ingestion capabilities.

To process and exchange data through the HL7 and FHIR standards, you can use the pre-built HL7 parser to receive HL7 data and the FHIR mapper and validation services to map the data to the FHIR standard and validate it.

To create the mapping, you can use a mapping template that's available in Data Integration. The mapping template requires source and target connections to read the HL7 messages and write the data to a FHIR server. To run the mapping, you'll need to run a mapping task.

Perform the following tasks:

1. Create a source connection.
2. Create a target connection.
3. Create a mapping using a mapping template. In Data Integration, use the mapping template called "Process incoming HL7 messages to an FHIR bundle."
4. Create and run a mapping task.

Step 1. Create the source connection

To receive HL7 messages, configure a source connection to access the HL7 data.

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- A Kafka connection through an MLLP server to read HL7 data from a messaging queue

To enable and configure an MLLP server, use the **File Servers** page in Administrator. Set the target type to write the HL7 data to a Kafka queue. For more information, see *File Transfer* in the Administrator help.

Step 2. Create the target connection

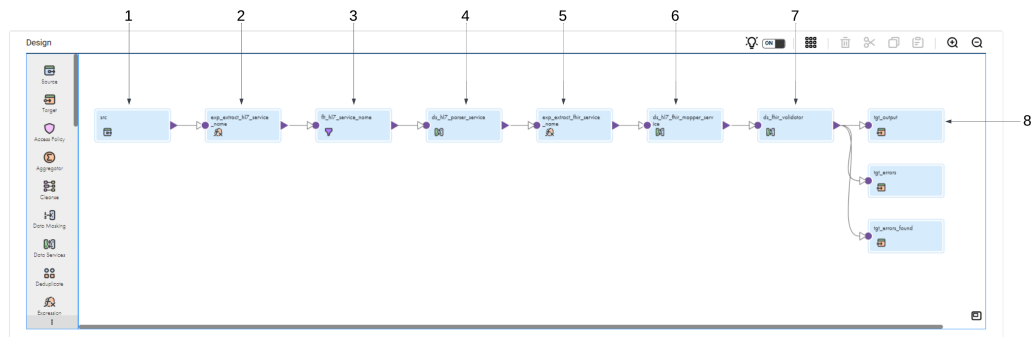
To write the FHIR bundles to a target, create a FHIR connection and configure the FHIR server properties.

Step 3. Configure the mapping

In Data Integration, use the mapping template called "Process incoming HL7 messages to an FHIR bundle" and configure the mapping.

The mapping template uses Data Services transformations that access pre-built data services from the data services repository to parse HL7 messages, map the XML data to FHIR bundles, and validate the FHIR bundles. You can populate the mapping template with source and target data using the source and target connections that you created.

The following image shows the transformations in the mapping template:



The mapping template contains the following transformations:

1. Source transformation

The Source transformation reads an HL7 file as a source. You can use a flat file connection, an SFTP connection to a remote server, or a Kafka connection to access the HL7 data.

The source can be one of the following types:

- Single object
- File list to process multiple files
- Command if the files are stored on the same Linux machine that runs the Secure Agent
- Parameter if you want to specify the source in the mapping task

The source is parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameter with a specific connection and source object in the Source transformation.
- Leave the parameter in the Source transformation and specify a source in the mapping task or use a parameter file.

2. Expression transformation

The Expression transformation uses the data in the HL7 file to construct the name of the data service.

Data services use the following naming convention:

```
<industry standard>_<version>_<message type>_<usage type>
```

For example, hl7_2_6_adt_A01_parser.

3. Filter transformation

The Filter transformation filters the incoming data from the Expression transformation to output only the final name of the data service that the Expression transformation constructs.

4. Data Services transformation (HL7 parser)

The Data Services transformation uses the name of the HL7 parser data service to access the data service in the data services repository. The HL7 parser reads the incoming patient and health information in the HL7 file and converts the data to XML format.

The mapping template sets the output type to buffer by default. If the output data is larger than the output precision, change the output type from buffer to file to avoid truncating the data.

5. Expression transformation

The Expression transformation constructs the name of the FHIR mapper service. It also creates fields for the base URL, bundle type, and event type to construct a FHIR bundle.

6. Data Services transformation (FHIR mapper service)

The Data Services transformation uses the FHIR mapper service in the data services repository to map the HL7 messages in XML format to FHIR bundles in XML format.

7. Data Services transformation (FHIR validation service)

The Data Services transformation uses the FHIR validation service in the data services repository to validate the FHIR bundles.

8. Target transformations

The Target transformations post the FHIR bundles to the FHIR server and include any errors and error flags. To write the data to the FHIR server, you use a FHIR connection.

The target is parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameter with a specific connection and target object in the Target transformation.
- Leave the parameter in the Target transformation and specify a target in the mapping task or use a parameter file.

Step 4. Create and run a mapping task

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

If you left the source and target parameters in the mapping template, you can configure the source and target in the mapping task.

In the mapping task, the **Sources** page appears if the mapping includes source parameters. You can add a single source object or multiple source objects based on the connection type and the mapping configuration. You can also configure a source filter. Similarly, the **Targets** page appears if the mapping includes target parameters.

CHAPTER 6

Using HL7 for data analysis

To use patient and health information through the HL7 messaging standard for data analysis, you can import a mapping template in Data Integration. The mapping template requires source and target connections to read and write the HL7 data. To run the mapping, you'll need to run a mapping task.

Perform the following tasks:

1. Create a source connection.
2. Create target connections.
3. Configure a mapping using a mapping template that you download from the Knowledge Base.
4. Create and run a mapping task.

Step 1. Create the source connection

To receive HL7 messages, configure a source connection to access the HL7 data.

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- A Kafka connection through an MLLP server to read HL7 data from a messaging queue

To enable and configure an MLLP server, use the **File Servers** page in Administrator. Set the target type to write the HL7 data to a Kafka queue. For more information, see *File Transfer* in the Administrator help.

Step 2. Create the target connections

To write the HL7 data in XML format to an intermediary target and the HL7 data in relational format to a relational target, create and configure the target connections.

Create the following target connections:

Target connection to write the HL7 data in XML format to an intermediary target

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- A relational connection if the relational database can store large objects that can hold XML data
- An MLLP connection to upload the XML data to a messaging queue

If you use an MLLP connection, you need to enable and configure an MLLP server before you create the connection.

To enable and configure the MLLP server, use the **File Servers** page in Administrator. Set the target type to write HL7 messages to a flat file or to a Kafka queue. For more information, see *File Transfer* in the Administrator help.

Target connection to write the HL7 data in relational format to a relational target

To write the HL7 data to a relational target, configure a connection to a relational database.

Step 3. Configure the mapping

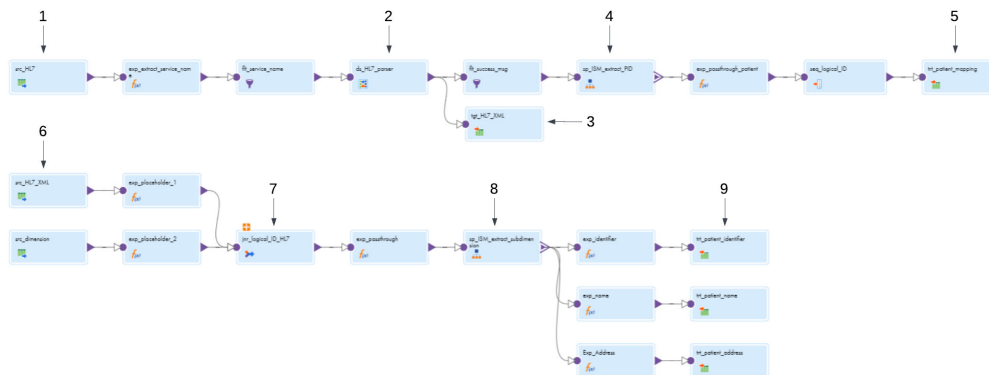
In Data Integration, import the mapping template and configure it.

To get the mapping template, download the attached file in the following Knowledge Base article and import it in your organization:

[HOW TO: Use a mapping template to process incoming HL7 2.x messages to load dimension and subdimension tables](#)

The mapping template extracts patient information from the PID segment in HL7 messages and converts it to relational format, and the attached file includes intelligent structure models that contain the schemas for the conversion. To create a different intelligent structure model, you can use a different HL7 file as input to the intelligent structure model and regenerate the schema. For more information, see the following Knowledge Base article: [FAQs on Using Intelligent Structure Discovery to Parse HL7 Messages](#).

The following image shows the transformations in the mapping template:



The sources and targets are parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameters with specific connections and objects in the Source and Target transformations.
- Leave the parameters in the Source and Target transformations and specify sources and targets in the mapping task or use a parameter file.

The mapping template contains the following notable transformations:

1. Source transformation (first flow)

The Source transformation reads an HL7 file as a source. You can use a flat file connection, an SFTP connection to a remote server, or a Kafka connection to access the HL7 data.

The source can be one of the following types:

- Single object
- File list to process multiple files
- Command if the files are stored on the same Linux machine that runs the Secure Agent
- Parameter if you want to specify the source in the mapping task

2. Data Services transformation (first flow)

The Data Services transformation uses the name of the HL7 parser data service to access the data service in the data services repository. The HL7 parser reads the incoming patient and health information in the HL7 file and converts the data to XML format.

3. Target transformation for XML output (first flow)

The target transformation writes the XML output to an intermediary target. To write the data to the target files, you can use any target connection that writes XML data, such as a flat file, relational, or MLLP connection.

4. Structure Parser transformation (first flow)

The Structure Parser transformation uses an intelligent structure model to convert the XML data to relational format.

5. Target transformation for relational output (first flow)

The Target transformation writes the relational output to a relational target to load the dimension table. To write the data to a relational target, use a relational connection.

6. Source transformations (second flow)

The Source transformations read the intermediary XML output and the relational output from the first flow.

7. Joiner transformation (second flow)

The Joiner transformation combines the data from both sources.

8. Structure Parser transformation (second flow)

The Structure Parser transformation uses an intelligent structure model to convert the data to relational format.

9. Target transformations (second flow)

The Target transformations write the relational output to relational targets to load subdimension tables. To write the data to a relational target, use a relational connection.

Step 4. Create and run a mapping task

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

If you left the source and target parameters in the mapping template, you can configure the source and target in the mapping task.

In the mapping task, the **Sources** page appears if the mapping includes source parameters. You can add a single source object or multiple source objects based on the connection type and the mapping configuration.

You can also configure a source filter. Similarly, the **Targets** page appears if the mapping includes target parameters.

CHAPTER 7

Using FHIR for data analysis

To use data through the FHIR data standard for data analysis, you can import a mapping template in Data Integration. The mapping template requires source and target connections to read and write the FHIR data. To run the mapping, you'll need to run a mapping task.

Perform the following tasks:

1. Create a source connection.
2. Create target connections.
3. Configure a mapping using a mapping template that you download from the Knowledge Base.
4. Create and run a mapping task.

Step 1. Create the source connection

To get FHIR bundles, create a FHIR connection and configure the FHIR server properties.

Step 2. Create the target connections

To write the errors from the FHIR validation service and the FHIR resource in relational format to a relational target, create and configure the target connections.

Create the following target connections:

Target connection to write errors in XML format from the FHIR validation service

You can use one of the following types of connections:

- A flat file or an SFTP connection to a remote server
- A relational connection if the relational database can store large objects that can hold XML data

Target connection to write the FHIR resource in relational format to a relational target

To write the FHIR resource to a relational target, configure a connection to a relational database.

Step 3. Configure the mapping

In Data Integration, import the mapping template and configure it.

To get the mapping template, download the attached file in the following Knowledge Base article and import it in your organization:

[HOW TO: Use a mapping template to process an incoming FHIR bundle in XML format and convert it to relational format](#)

The mapping template extracts the patient FHIR resource from a FHIR bundle and converts it to relational format, and the attached file includes an intelligent structure model that contains the schema for the conversion. To create a different intelligent structure model, you can use a different XML file as input to the intelligent structure model and regenerate the schema.

The following image shows the transformations in the mapping template:



The targets are parameterized in the mapping template, so you can perform one of the following tasks:

- Replace the parameters with specific connections and target objects in the Target transformations.
- Leave the parameters in the Target transformations and specify targets in the mapping task or use a parameter file.

The mapping template contains the following transformations:

1. Source transformation

The Source transformation uses a FHIR connection to submit a GET request for a patient resource by querying patient details. The sample query references a patient's given name.

2. Expression transformation

The Expression transformation removes special characters in the XML data that can cause validation issues, such as converting ampersands to `&`.

3. Data Services transformation

The Data Services transformation uses the FHIR validation service from the data services repository to validate the FHIR resource.

4. Target transformations to troubleshoot validation errors

The Target transformations that are downstream of the Data Services transformation write errors and error flags to target files. If the FHIR validation service finds errors in the incoming FHIR resource, you can use the error messages to troubleshoot the upstream data flow. To write the data to the target files, you can use any target connection that writes XML data, such as a flat file or relational connection that can hold XML data.

5. Structure Parser transformation

The Structure Parser transformation uses the intelligent structure model to convert the XML data to relational format.

6. Target transformations to write relational data

The Target transformations write the patient elements to a relational target, such as the patient, patient identifier, and patient address elements. To write the data to a relational target, use a relational connection.

Step 4. Create and run a mapping task

After you configure the mapping, create and run a mapping task to process the data flow logic in the mapping.

If you left the target parameters in the mapping template, you can configure the targets in the mapping task. The **Target** page appears if the mapping includes target parameters. You can add a single target object or multiple target objects based on the connection type and the mapping configuration.