

Oracle® Cloud

Using Oracle Integration for Healthcare in Oracle Integration 3



F84764-16
January 2026



Oracle Cloud Using Oracle Integration for Healthcare in Oracle Integration 3,

F84764-16

Copyright © 2024, 2026, Oracle and/or its affiliates.

Primary Author: Oracle Corporation

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, software documentation, data (as defined in the Federal Acquisition Regulation), 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 embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside 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, Epyc, and the AMD 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

About This Content

1 Get Started with Healthcare

Find More Resources	1
---------------------	---

2 Introduction to Oracle Integration for Healthcare

About HL7	1
About Clinical Document Architecture	1
Oracle Integration for Healthcare Support for HL7 and FHIR	2
Oracle Integration for Healthcare Restrictions	3
Quick Tour Through Oracle Integration for Healthcare Use Cases	3
Synchronize EMR Patient Updates in a FHIR Patient Repository and a Pathology System	4
Synchronize Employee Information Between an HR Application and an EMR Application	5
Workflow to Use Oracle Integration for Healthcare	7
Provision Oracle Integration for Healthcare	8
Oracle Integration for Healthcare Videos and Live Labs	9

3 Create Oracle Integration for Healthcare Schemas and Documents

Create a New Healthcare Schema	1
Add a New Segment	3
Add a New Loop	4
Edit or Clone a Healthcare Schema	4
Create a Custom HL7 Message	6
Edit or Clone an HL7 Message	9

4 Create Clinical Document Architecture (CDA) Documents

Create a Clinical Document Architecture Document	1
Edit or Clone a Clinical Document Architecture Document	4

5	Design an Integration with a Healthcare Action	
	<hr/>	
	Best Practices for Designing a Healthcare Integration	1
	Create Trigger and Invoke Connections that Support the HL7 and FHIR Standards	5
	Handle Inbound Debatching of Large HL7 Message Payloads	6
	Convert HL7 Messages with a Healthcare Action	6
6	Import FHIR Profile Packages and Create Custom FHIR Resources	
	<hr/>	
	Import a FHIR Profile Package	1
	Browse the Contents of the FHIR Profile Package	2
	Create Custom FHIR Resources	3
	Edit a Custom FHIR Resource	6
7	Ensure Messages are Processed in the Received Order	
	<hr/>	
	About Ordered Message Processing in Oracle Integration	1
	Create and Add Message Queues to an Integration	3
	Monitor and Manage Queued Messages at Runtime	5
	Ordered Message Processing Use Cases	9
8	Design Time and Runtime Overview of an Oracle Integration for Healthcare Use Case	
	<hr/>	
	Use Case Overview	1
	Overview of the Design-Time Messages	2
	Overview of the Integration Receiving the Inbound HL7 Message During Runtime	3
	Overview of the Integration Processing the Patient Updates During Runtime	6

About This Content

This guide describes how to use Oracle Integration for Healthcare.

Audience

This guide is intended for developers who want to use Oracle Integration for Healthcare.

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 that have purchased support 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 Resources

See these Oracle resources:

- Oracle Cloud at <http://cloud.oracle.com>
- *Using Integrations in Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*
- Oracle Integration documentation on the Oracle Help Center.

Conventions

The following text conventions are used in this document.

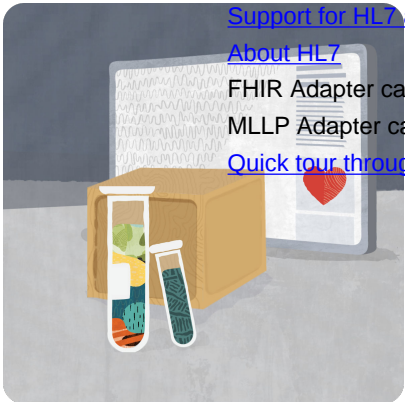
Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type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

1

Get Started with Healthcare

A healthcare integration synchronizes data in HL7 format across applications. Browse the following popular topics to learn more.

Explore Healthcare

Image	Links
	<ul style="list-style-type: none">Support for HL7 and FHIRAbout HL7FHIR Adapter capabilitiesMLLP Adapter capabilitiesQuick tour through healthcare use cases

Connect Healthcare Applications

Image	Links
	<ul style="list-style-type: none">Provision Oracle Integration for HealthcareWorkflow: Create a healthcare integrationDesign an integration with a healthcare actionCreate a healthcare schema

Find More Resources

Oracle Integration offers many ways to stay connected and grow your expertise.

Resource	Description	Link
Documentation	Find step-by-step instructions, best practices, and more.	Get Started page on the Oracle Help Center
Tutorials	Take a product tour, build an integration from start to finish, or complete a longer self-directed workshop with a LiveLab.	Tours and Tutorials in <i>Using Integrations in Oracle Integration 3</i>
Videos	Learn more about Oracle Integration by watching a short video.	Videos page on the Oracle Help Center
Live demos	Created by the product management team, live demos offer deep dives into new features, use cases, and more. Subscribe to the channel to get notified when a new live demo is available.	Oracle Integration on YouTube
Oracle Integration blog	Learn what's new from the product management team.	Oracle Integration blog
A-Team Blog	Learn from technical solution architects and software engineers.	A-Team Chronicles blog
Cloud Customer Connect	Connect with experts, ask or answer questions, and share your opinions.	Cloud Customer Connect
Newsletter	Oracle Integration sends a quarterly newsletter with product, event, and training information.	Subscribe to the Oracle Integration Newsletter
Webcasts	Oracle Integration offers the following quarterly webcasts: <ul style="list-style-type: none"> • Product Update Webcasts Get product updates and watch live demos. • Customer Success Webcasts Get inspired by success stories and best practices, or present your own wins to the community. 	Integration Resources blog post: Watch past webcasts and learn about upcoming webcasts Subscribe to the Oracle Integration Newsletter : Learn about upcoming webcasts by subscribing to the newsletter
Oracle Partners	Use the Oracle PartnerNetwork to find a partner to support your business goals. Connect to the Oracle network to view this page.	Oracle Partner Finder

2

Introduction to Oracle Integration for Healthcare

Learn about how Oracle Integration for Healthcare provides support for HL7 in Oracle Integration.

Topics:

- [About HL7](#)
- [About Clinical Document Architecture](#)
- [Oracle Integration for Healthcare Support for HL7 and FHIR](#)
- [Oracle Integration for Healthcare Restrictions](#)
- [Quick Tour Through Oracle Integration for Healthcare Use Cases](#)
- [Workflow to Use Oracle Integration for Healthcare](#)
- [Provision Oracle Integration for Healthcare](#)
- [Oracle Integration for Healthcare Videos and Live Labs](#)

About HL7

Health Level Seven (HL7) provides standards for the exchange, integration, sharing, and retrieval of health information between systems within and across healthcare providers. HL7 provides a delimited, flat file-based message structure. Messages are event-based (order blood tests, receive test results, update patient information, and others). There is a corresponding message for each event.

Health organizations typically include numerous healthcare systems to process different patient administrative or clinical tasks, including billing, medication management, patient tracking, and documentation. These systems communicate when they receive new information or need to retrieve information. HL7 provides standards, guidelines, and methodologies for these systems to communicate with each other. The standards allow for the interoperability of healthcare data as it's shared and processed by different systems. This interoperability allows clinical and non-clinical data to be shared more easily with the goal of improving patient care and health system performance.

About Clinical Document Architecture

Clinical Document Architecture (CDA) is an HL7v3, XML-based standard that provides a structure for encoding, formatting, and exchanging electronic clinical documents, such as patient and discharge summaries, progress notes, imaging and lab reports, and other medical reports. CDA enables interoperability between different healthcare IT systems.

CDA consists of the following core elements:

- The method for transporting documents is not specified.
- Documents have a consistent structure with required and optional sections (for example, patient demographics, observations, and medications).

- Codes from standard vocabularies are implemented, supporting interoperability and data exchange.
- Long-term storage of documents as part of a patient's medical record is supported.
- Accommodations of new requirements across different clinical environments are supported.

Common CDA use cases include the following:

- A continuity of care document (CCD) is exchanged between providers through a health information exchange (HIE) or direct secure exchange such as REST. An HIE enables health care providers to securely access and share a patient's medical information electronically across different healthcare organizations, such as hospitals, clinics, and labs.
- A patient receives their discharge summary from a patient portal.
- A primary care provider (PCP) refers a patient to a specialist such as a physician from oncology, podiatry, cardiology, immunology, or another department.

See [Clinical Document Architecture \(CDA\)](#).

Oracle Integration for Healthcare Support for HL7 and FHIR

Oracle Integration for Healthcare enables you to create integrations that interact with healthcare organizations that use HL7 and FHIR.

Oracle Integration for Healthcare provides the following capabilities:

- Available for use with Oracle Integration through installation of the Healthcare edition in the Oracle Cloud Console. See Oracle Integration Editions in *Provisioning and Administering Oracle Integration 3*.
- Supports HL7 version 2.
- Supports the following versions of the HL7 version 2 standard: 2.3.1, 2.4, 2.5.1, 2.5, 2.6, 2.7, 2.7.1, 2.8.1, 2.8, 2.8.2, and 2.9.
- Supports a processing mechanism to ensure that HL7 messages are delivered to target systems in the order received. For example, an ADT_A01 (Patient Admitted) message is received and routed to multiple departments in the correct order, such as lab, hospital management, and other departments. See [Ensure Messages are Processed in the Received Order](#).
- Supports HL7 applications on different versions interacting with each other through Oracle Integration for Healthcare.
- Supports importing FHIR profile packages. If you are unable to successfully import a profile package, see [this blog](#). Once imported, you can view the contents of the FHIR profile package.
- Supports custom FHIR resources. This feature enables you to customize a FHIR resource from the underlying FHIR base schema to use in integrations.
- Supports Clinical Document Architecture (CDA) documents, an XML-based standard that provides a structure for encoding, formatting, and exchanging electronic clinical documents. See [Create Clinical Document Architecture \(CDA\) Documents](#).
- Supports creating healthcare integrations in projects or in standalone environments (outside of a project). See Design, Manage, and Monitor Integrations in Projects and Create Healthcare Integrations in a Project in *Using Integrations in Oracle Integration 3*
- Supports common healthcare interoperability use cases, including:

- Transfer of care
- Inventory synchronization – employee medical record (EMR) and enterprise resource planning (ERP)
- Payer/provider
- Modernization programs:
 - * Exposure of health data using Fast Healthcare Interoperability Resources (FHIR)
 - * Mobile applications
 - * Patient portal
- Update of a state immunization registry upon patient checkout
- Supports message standards and transport protocols commonly used in the healthcare interoperability (integration) market:
 - Supports the Minimal Lower Layer Protocol (MLLP). MLLP is a TCP-based standard for sending and receiving HL7 messages. You configure the MLLP Adapter to receive inbound (trigger) HL7 messages or send outbound (invoke) HL7 messages. See MLLP Adapter Capabilities in *Using the MLLP Adapter with Oracle Integration 3*.
 - Supports Fast Healthcare Interoperability Resources (FHIR). FHIR is a set of rules and specifications for exchanging electronic healthcare data over the REST protocol in XML or JSON format. You configure the FHIR Adapter to invoke FHIR resources (for example, a patient resource, encounter resource, and others) in outbound messages in Oracle Integration. See FHIR Adapter Capabilities in *Using the FHIR Adapter with Oracle Integration 3*.
- Supports defining the connections to external systems with which to exchange HL7 messages.
- Supports designing and customizing the HL7 messages to exchange. See [Create Oracle Integration for Healthcare Schemas and Documents](#).
- Supports integrating and processing HL7 messages in integrations with the healthcare action. The healthcare action converts inbound HL7 messages to XML format and outbound messages from XML format to HL7 format. See [Design an Integration with a Healthcare Action](#).
- Supports tracking and monitoring HL7 integrations at runtime.
- Supports retaining runtime data for 184 days when selecting the **Production** tracing level during integration activation. See Activate an Integration in *Using Integration Insight in Oracle Integration 3*

Oracle Integration for Healthcare Restrictions

Be aware of the following Oracle Integration for Healthcare restrictions.

- Oracle Integration supports HL7 version 2. HL7 version 3 is not currently supported.

Quick Tour Through Oracle Integration for Healthcare Use Cases

This section provides high level overviews of several Oracle Integration for Healthcare use cases.

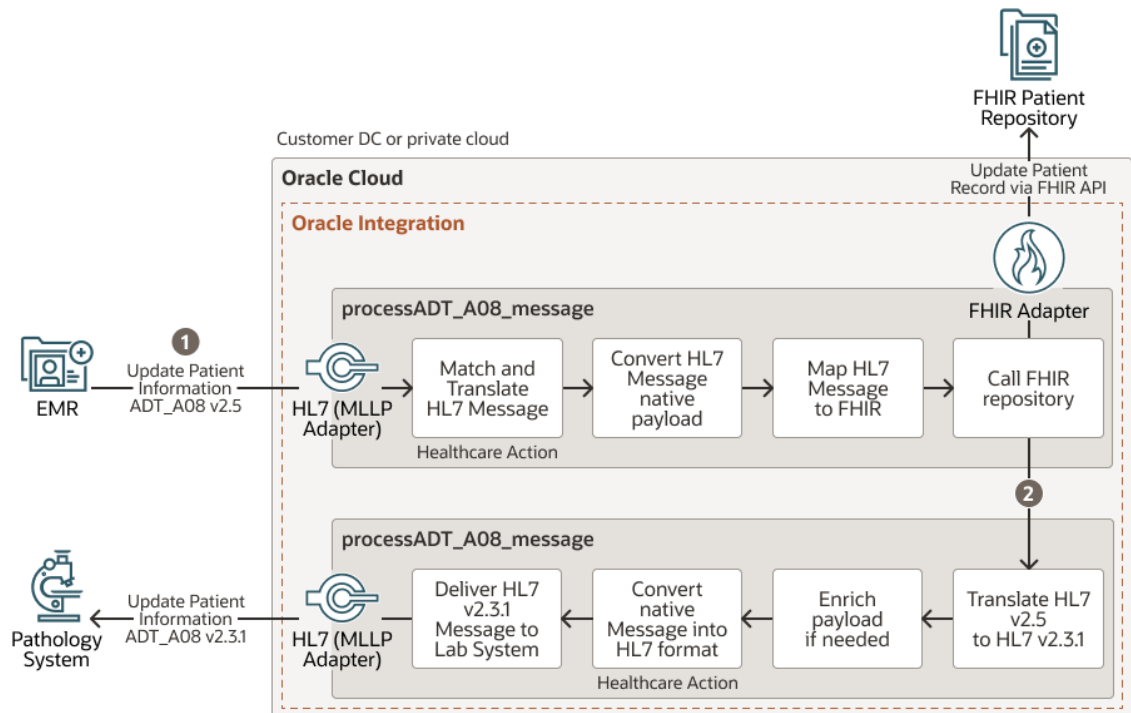
Topics:

- [Synchronize EMR Patient Updates in a FHIR Patient Repository and a Pathology System](#)

- [Synchronize Employee Information Between an HR Application and an EMR Application](#)

Synchronize EMR Patient Updates in a FHIR Patient Repository and a Pathology System

This use case describes how an update to patient details in an employee medical record (EMR) application is automatically propagated to a FHIR patient repository and a pathology system through use of Oracle Integration for Healthcare.



1. Patient details are updated in an EMR, which emits an HL7 ADT_A08 message. The message is received by Oracle Integration and converted to a FHIR patient resource for updating in a FHIR-enabled patient repository.
2. The integration continues and Oracle Integration translates an HL7 message from version 2.5 to version 2.3.1 and sends it to a pathology system for updating the patient records.

EMR Application Patient Updates

- A doctor updates details about a patient (for this example, a change in name) in an EMR application. The updates are performed through use of an HL7 Update Patient Information message (ADT_A08). The EMR application uses HL7 version 2.5.
- The EMR application emits the updated HL7 ADT_A08 message.
- An MLLP Adapter running inside the connectivity agent and configured as a trigger connection in an integration receives the incoming message for processing.
- The HL7-formatted message is converted into an XML-formatted message by a healthcare action in the integration.

The following updates are automatically made in parallel in the FHIR patient repository and the pathology system:

- [FHIR Patient Repository Updates](#)
- [Pathology System Updates](#)

FHIR Patient Repository Updates

- The XML message is converted into a JSON-formatted, FHIR patient resource in the integration.
- The FHIR patient repository is invoked with the FHIR Adapter. Based on the type of patient, the following updates are applied in the FHIR repository:
 - For an existing FHIR repository patient: The patient is updated with the name change made in the EMR application.
 - For a new patient: All patient details are added to the FHIR repository.

Pathology System Updates

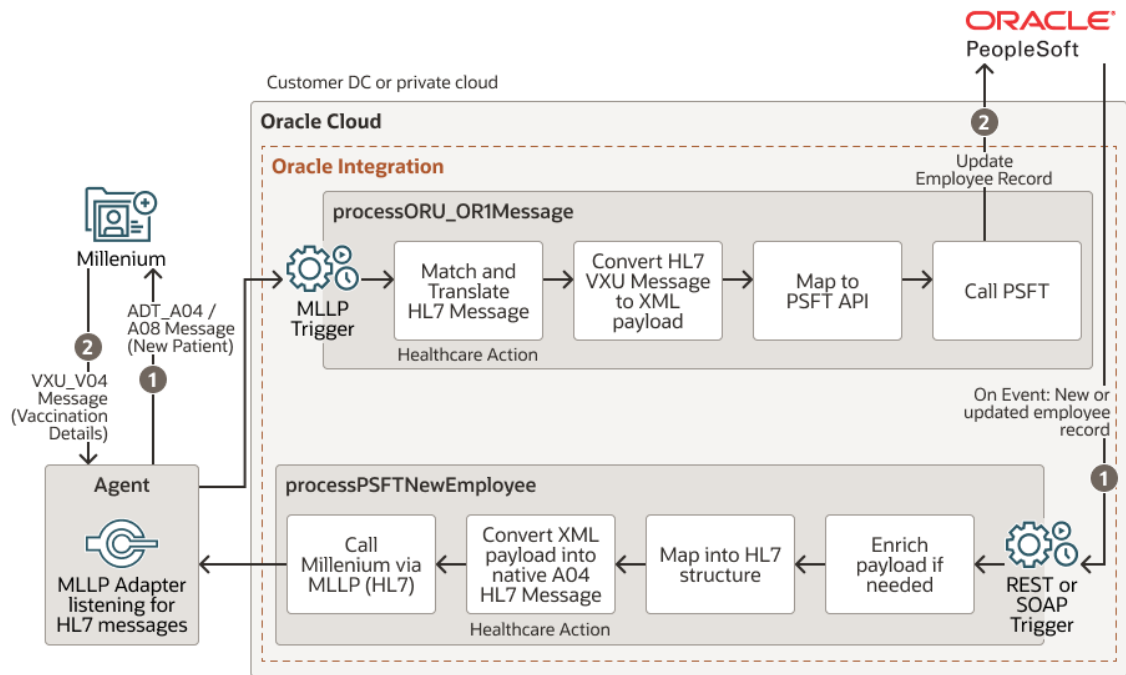
Because the pathology system uses an earlier version of HL7 (2.3.1), the message must be down-versioned from 2.5 to 2.3.1 for the message to be successfully processed.

- The ADT_A08 message is:
 - Converted from XML format back to HL7 format by a healthcare action in the integration.
 - Down-versioned from HL7 version 2.5 to version 2.3.1.
- The MLLP Adapter running inside the connectivity agent and configured as an invoke connection sends the down-versioned, HL7-formatted message to the pathology system, where the patient details are updated.

An overview of design instructions for this use case is provided. See [Design Time and Runtime Overview of an Oracle Integration for Healthcare Use Case](#).

Synchronize Employee Information Between an HR Application and an EMR Application

This use case describes how a new employee hire added to a human resources (HR) application is also automatically added as a patient in an employee medical record (EMR) application. When this employee later receives a vaccination, the update made to the EMR application is automatically propagated back to the HR application.



1. Start of onboarding a new employee. A new employee event is sent to Oracle Integration, converted to an HL7 message, and sent to Mellennium.
2. Start of a new or updated employee vaccination record. A new HL7 vaccination message event is sent to Oracle Integration, converted to Oracle PeopleSoft format, and sent to Oracle PeopleSoft.

HR Application Employee Updates Synchronized in an EMR Application

Acme Hospital uses two applications to store information:

- An HR application (for this example, Oracle PeopleSoft) stores information about their employees (for example, date of hire, job title, compensation details, and so on).
- An EMR application (for this example, Mellennium) stores internal health records pertinent to employee jobs (for example, employee vaccination records, employee allergy records, and so on). The EMR application uses HL7.

The following actions occur:

- A new employee is hired and added to the HR application.
- The HR application update triggers a REST Adapter connection to start an integration between the HR application and the EMR application.
- The HR application message is converted to an HL7-formatted, ADT_A04 message (patient registration) by a healthcare action in the integration.
- An MLLP Adapter running inside the connectivity agent and configured as an invoke connection sends the outbound message to the EMR application at Acme Hospital.
- The ADT_A04 message (patient registration) is automatically added to the EMR application. The new employee is now available in both applications.

EMR Application Employee Updates Synchronized in the HR Application

Acme Hospital employees must periodically receive various vaccinations, allergy tests, and so on.

- An Acme Hospital nurse administers an employee's annual vaccination and updates their medical record in the EMR application.
- The EMR application emits an HL7 ORU_OR1 message (vaccination details message).
- An MLLP Adapter running inside the connectivity agent and configured as a trigger connection in a second integration receives the incoming message for processing.
- The HL7-formatted message is converted into an XML-formatted message by a healthcare action in the integration.
- A REST API call invokes the HR application and the employee's records are updated to reflect their vaccination details.

Workflow to Use Oracle Integration for Healthcare

You follow a simple workflow to create integrations in Oracle Integration for Healthcare.

Note

To follow this workflow, you must first select the Healthcare edition during Oracle Integration instance provisioning. See [Provision Oracle Integration for Healthcare](#).

Step	Description	More Information
1	Install the connectivity agent. The MLLP Adapter runs directly in the connectivity agent to send and receive HL7 messages to and from MLLP endpoints.	Download and Run the Connectivity Agent Installer in <i>Using Integrations in Oracle Integration 3</i>
2	Create adapter connections: <ul style="list-style-type: none"> • If your use case requires invoking a FHIR-based endpoint, create a FHIR Adapter connection. • If your use case requires receiving inbound (trigger) HL7 messages or sending outbound (invoke) HL7 messages, create an MLLP Adapter connection. 	<ul style="list-style-type: none"> • Create a FHIR Adapter Connection in <i>Using the FHIR Adapter with Oracle Integration 3</i> • Create an MLLP Adapter Connection in <i>Using the MLLP Adapter with Oracle Integration 3</i>
3	Create a healthcare schema and HL7 message. You create a schema and then an HL7 message based on the schema to use in your integration. Note: You cannot use the healthcare action if you have not first created at least one HL7 message (message definition).	Create a New Healthcare Schema and Create a Custom HL7 Message

Step	Description	More Information
4	Design an integration with your adapter connections and a healthcare action. You select the HL7 message created in Step 3 for conversion when you configure the healthcare action.	Create Trigger and Invoke Connections that Support the HL7 and FHIR Standards and Design an Integration with a Healthcare Action
5	Activate the integration. Once activated, you can run the integration.	Activate an Integration in <i>Using Integrations in Oracle Integration 3</i>
6	Monitor the integration.	Monitor Integrations During Runtime and Monitor Integrations in a Project in <i>Using Integrations in Oracle Integration 3</i>

Provision Oracle Integration for Healthcare

Oracle Integration for Healthcare is available *only* as part of the Healthcare edition. It is *not* included with the Standard or Enterprise editions of Oracle Integration. You can select the Healthcare edition during Oracle Integration instance installation or change an existing instance from the Standard or Enterprise edition to the Healthcare edition.

This table provides an overview of key installation, edition change, and billing information and provides references to relevant documentation.

Subject	Description	See
Edition Features	The Healthcare edition includes the same features as the Enterprise Edition, plus Oracle Integration for Healthcare.	Oracle Integration Editions in <i>Provisioning and Administering Oracle Integration 3</i>
Installation	You select the Healthcare edition during instance installation.	Create an Oracle Integration Instance in <i>Provisioning and Administering Oracle Integration 3</i>
Changing editions	If your existing Oracle Integration instance is already configured with the Standard or Enterprise edition, you can change to the Healthcare edition. Note: Once you change to the Healthcare edition, you cannot change back to Standard or Enterprise.	Edit the Edition, License Type, Message Packs, and Custom Endpoint of an Instance in <i>Provisioning and Administering Oracle Integration 3</i>
Billing	Healthcare edition message packs are billed at a higher price than Standard or Enterprise.	See Monitor Oracle Integration 3 Instances in <i>Provisioning and Administering Oracle Integration 3</i>

How Do I Know If My Instance Includes the Healthcare Edition

If your existing Oracle Integration installation does not currently consist of the Healthcare edition, you see the following message when navigating through the user interface:

- When you select **Healthcare** in the left navigation pane
- When you click **Healthcare**  inside a project

Integrate with healthcare applications

Exchange messages and events between healthcare applications.
Before you start, upgrade to Healthcare edition in Oracle Cloud Infrastructure.

[How do I upgrade to Healthcare edition?](#)

If you change this instance from the Standard or Enterprise edition to the Healthcare edition, you can edit these pages.

Oracle Integration for Healthcare Videos and Live Labs

Interested in viewing Oracle Integration for Healthcare live demo videos or completing a hands-on lab? Live demo videos show how to configure and run key Oracle Integration for Healthcare features. Oracle LiveLabs provide you with access to Oracle's technologies to run labs and workshops. See the following links for details.

- [Healthcare live demo videos](#)
- [Oracle LiveLabs - Get Started with Healthcare](#)

3

Create Oracle Integration for Healthcare Schemas and Documents

You create a healthcare schema using a standard HL7 schema definition as a starting point. All HL7 version 2 base schemas from version 2.3.1 onwards are supported and provided with Oracle Integration. You can also customize the schema according to your needs.

You then create a document definition based on the schema. This document definition represents the HL7 message to use in your integrations. You then select this document to convert when configuring the healthcare action in your integration.


Topics:

- [Create a New Healthcare Schema](#)
- [Edit or Clone a Healthcare Schema](#)
- [Create a Custom HL7 Message](#)
- [Edit or Clone an HL7 Message](#)

Create a New Healthcare Schema

You can create a new HL7 schema based on a standard out-of-the-box HL7 message definition and customize it to match your requirements.

You can create a schema in a project or a standalone environment (outside of a project).

1. To create a schema in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to create the schema.
 - c. Click **Healthcare** .
 - d. In the **Schemas** section, click **Add**.
2. To create a schema in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **Schemas**.
 - b. On the Schemas page, click **Create**.
3. Enter the following details.

Element	Description
Name	Enter a schema name.
Identifier	This field is automatically populated with a unique schema identifier based on your schema name. You can manually change this value, if needed.
Description	Enter an optional description of this schema.

Element	Description
Message standard	Select the message standard. The message standard identifies the business protocol to follow when exchanging messages between applications. <ul style="list-style-type: none"> • HL7V2. • CDA. See Create a Clinical Document Architecture Document.
Message version	Select the message version. HL7 V2 versions 2.3.1, 2.4, 2.5.1, 2.5, 2.6, 2.7, 2.7.1, 2.8.1, 2.8, 2.8.2, and 2.9 are supported.
Message type	Select the message type. The types shown are based on the HL7 V2 version you selected.

For this example, ADT_A08 (update patient information) is selected.

Create Healthcare schema

Name
MYADT_A08_SCHEMA

Identifier
MYADT_A08_SCHEMA

What does it do?

Description
Schema for ADT_A08 message

998

Message standard
HL7V2 ▼

Message version
2.9 ▼

Message type
ADT_A01 (Admit/visit notification) ▼

4. Click **Create**.

The details page for your new HL7 message is displayed. The standard segments and elements that come with the message type you selected are shown. A segment is a higher level construct, consisting of a sequence of elements and composites. An element is the smallest unit that represents a single data field of a primitive type, such as alphanumeric text, integer, decimal, date, time, or binary. If you create a schema in a project, the project name is displayed in the banner. For this example, the schema was created in a standalone environment (outside of a project).



Segment/Element	Name	Cardinality	Element type
MSH	Message Header	[1..1]	
ARV	Access Restriction	[0..*]	
SFT	Software Segment	[0..*]	
UAC	User Authentication Credential Segment	[0..1]	
EVN	Event Type	[1..1]	
PID	Patient Identification	[1..1]	
PDI	Patient Additional Demographic	[0..1]	
OH1	Person Employment Status	[0..*]	
OH2	Past or Present Job	[0..*]	

You can customize the standard HL7 message. For example, the application with which you are integrating may require another segment or loop. You can add new constructs (such as new segments and loops) to a standard healthcare schema or edit existing constructs within it.

5. If you want to create a new segment, see [Add a New Segment](#).
6. If you want to create a new loop, see [Add a New Loop](#).

Add a New Segment

According to your organization's requirements, you can add new segments to a standard healthcare schema to customize it.

1. Hover over the row of a segment/element and click **Actions** , then **New segment**. The Add new segment panel is displayed.
2. If you want to use the contents of an existing segment definition to create a new segment, click **Existing segment**. This selection is useful when the contents of an existing segment are similar to the one you want to create.
 - a. Expand **Advance Search**.
 - b. Search by the version number or schema name to use.
 - c. Begin typing the initial letters of the segment to show the available selections.
 - d. Select the segment.
 - e. Click **Add**.
 - f. Enter a name, description, and cardinality value. For example, if you enter [0..1], a value of either zero or one is expected.
 - g. Click **Add**.
3. If you want to create a completely new segment without using existing information, click **New segment**.
 - a. Enter a name, and click **Add**. The Properties panel is displayed.
 - b. Enter a name, description, and cardinality value. For example, if you enter [0..1], a value of either zero or one is expected.
 - c. Click **Save**.
4. If you want to perform additional customizations, hover over the row of the segment/element and click **Actions** , then select to perform more customizations:


- **New child element:** The smallest unit that represents a single data field of a primitive type, such as alphanumeric text, integer, decimal, date, time, or binary.
- **New child composite:** A complex data type consisting of one or more elements.
- **New segment:** The next higher level construct, consisting of a sequence of elements and composites.
- **New loop:** A container for a specific set of segments or child loops, which makes its structure nested and hierarchical.

Note

If your inbound HL7 message request contains unexpected custom segments, you can select the **Ignore non-modeled segments** option when configuring the healthcare action in the integration canvas. With this option selected, unexpected custom segments not found in the schema are ignored completely, no errors are raised, and a response is successfully generated at runtime. See Convert HL7 Messages with a Healthcare Action in *Using Oracle Integration for Healthcare in Oracle Integration 3*.

Add a New Loop



According to your organization's requirements, you can add new loops to a standard healthcare schema to customize it. A loop is a container for a specific set of segments or child loops, which make its structure nested and hierarchical.


1. Hover over the row of a segment/element and click **Actions** , then **New loop**. The Add new loop panel is displayed.
2. Enter a name for the segment, and click **Add**.
3. Enter a cardinality value. For example, if you enter [0..1], a value of either zero or one is expected.
4. Click **Save**.

Edit or Clone a Healthcare Schema

You can edit an existing schema to update the name, description, and segment/element properties. You can also clone a healthcare schema to create a copy of it.

Edit a Healthcare Schema

1. To edit a healthcare schema in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to edit the healthcare schema.
 - c. Click **Healthcare** .
 - d. Go to the **Schemas** section.
 - e. Click **Actions** , then select **Edit**.
2. To edit a healthcare schema in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **Schemas**.
 - b. Hover over a schema row to see the actions you can perform on a schema.

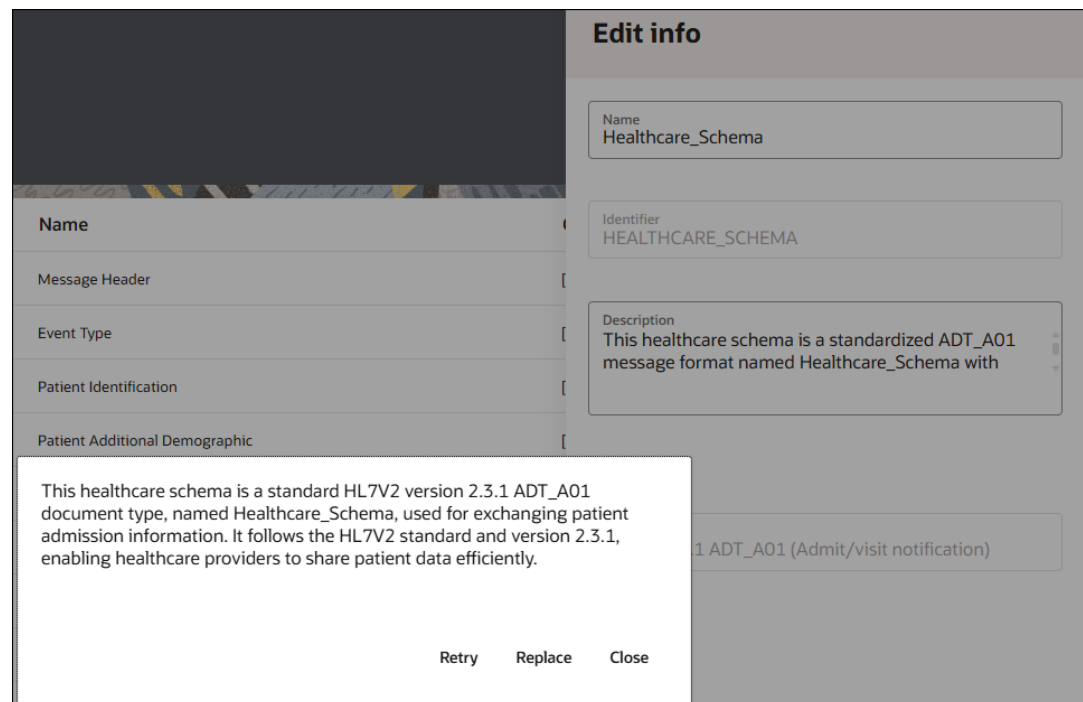
- c. Click **Edit** .
3. Click **Primary Info**.
4. Edit the schema's name and description, as necessary.
You can use AI to generate a new description or replace an existing description of a healthcare schema with a single click. Click **Generate** to create an AI-powered description.

Note

AI features are available in select regions. See *AI Feature Availability in Provisioning and Administering Oracle Integration 3*.

The **Description** field can be empty or already include a description.

- Click **Retry** to regenerate the description. If you retry, the newer description varies slightly in terms of the words used, but the meaning remains the same.
- Click **Replace** to add the generated schema description to the **Description** field. If the **Description** field is empty, the generated schema description is added to the field. If the **Description** field already includes a schema description, the current description is replaced by the generated description. For this example, a description already existed. **Replace** is only selectable if the healthcare schema is editable.



Edit info


Name
Healthcare_Schema

Identifier
HEALTHCARE_SCHEMA



Description
This healthcare schema is a standardized ADT_A01 message format named Healthcare_Schema with

This healthcare schema is a standard HL7V2 version 2.3.1 ADT_A01 document type, named Healthcare_Schema, used for exchanging patient admission information. It follows the HL7V2 standard and version 2.3.1, enabling healthcare providers to share patient data efficiently.

Retry Replace Close

5. If you want to edit the properties of a segment/element, then hover over a segment/element row to see the actions you can perform on a segment/element.
6. Click **Edit** .
7. Click **Save**.


Clone a Healthcare Schema

1. To clone a healthcare schema in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to clone the healthcare schema.
 - c. Click **Healthcare** .
2. Go to the **Schemas** section.
3. To clone a healthcare schema in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **Schemas**.
4. Hover over a healthcare schema row to see the actions you can perform on a healthcare schema.
5. Click **Actions** , then select **Clone** to clone the healthcare schema. The Clone schema panel opens.
6. Make any necessary updates, then click **Clone**.

Create a Custom HL7 Message

After you create a schema, you must create an HL7 message based on that schema for use in your integration. The message represents a specific snapshot of a schema to use in an integration. You later select the HL7 message for conversion into a supported format when you configure the healthcare action during design of your integration.

You can create an HL7 message in a project or a standalone environment (outside of a project).

1. To create a message in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to create the message.
 - c. Click **Healthcare** .
 - d. In the **HL7 messages** section, click **Add**.
2. To create a message in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **HL7 messages**.
 - b. On the HL7 messages page, click **Create**.
3. Enter the following details to create a new HL7 message definition.

Element	Description
Name	Enter a message name.
Identifier	This field is automatically populated with the message name. You can manually change this value.
Description	Enter an optional description of this message.
Message standard	View the message standard. Only HL7V2 is available and it cannot be deselected. The message standard identifies the business protocol to follow when exchanging messages between applications.

Element	Description
Message version	Select the message version. HL7 message versions 2.3.1, 2.4, 2.5.1, 2.5, 2.6, 2.7, 2.7.1, 2.8.1, 2.8, 2.8.2, and 2.9 are supported.
Message type	Select the HL7 message type. The types shown are based on the HL7 message version you selected.

Create HL7 message

Name
MyADT_A08_Doc

Identifier
MYADT_A08_DOC

What does it do?

Description
HL7 ADT_A08 Doc

Message standard
HL7V2

Message version
2.9

Message type
ADT_A08 (Update patient information)

4. Click **Create**.
The details page for your new message is displayed.

MyADT_Ao8_Doc

Identifier: MYADT_A08_DOC Updated on: Jul 31st, 2024, 01:44:59 PM PDT

Message selection

Name
MyADT_A08_Doc

Description
HL7 ADT_A08 Doc

Message standard
HL7V2

Message version
2.9

Message type
ADT_A08 (Update patient information)

Message schema
Standard

Customize

5. Note that the message version and message type values you selected previously are displayed. These cannot be changed. The **Message schema** field shows **Standard** as the schema type by default. If you had previously created custom schemas, they are also displayed for selection in the drop-down list.
6. Select the schema to use or click **Customize** to display the Clone standard schema dialog to create a new schema. For this example, the schema created in [Create a New Healthcare Schema](#) is selected.
7. Click **Save**, and return to the HL7 messages page. If you create a document in a project, the project name is displayed in the banner. For this example, the document was created in a standalone environment (outside of a project).




Name	Standard	Version	Type	Usage	Last updated
MyADT_A08_Doc HL7 ADT_A08 Doc	HL7V2	2.9	ADT_A08	0 integrations	Jul 31st, 2024 01:44:59 PM PDT

The message is now available for conversion when you configure the healthcare action in an integration. See [Design an Integration with a Healthcare Action](#).

Edit or Clone an HL7 Message

You can edit an existing message to associate a different schema. You can also clone an HL7 message to create a copy of it.

Edit an HL7 Message

1. To edit a message in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to edit the message.
 - c. Click **Healthcare** .
 - d. Go to the **HL7 messages** section.
 - e. Click **Actions** , then select **Edit**.
2. To edit a message in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **HL7 messages**.
 - b. Hover over a message row to see the actions you can perform on a message.
 - c. Click **Edit** .
3. Edit the message's name and description, as necessary.

You can use AI to generate a new description or replace an existing description of an HL7 message with a single click. Click **Generate** to create an AI-powered description.

Note

AI features are available in select regions. See *AI Feature Availability in Provisioning and Administering Oracle Integration 3*.

The **Description** field can be empty or already include a description.

- Click **Retry** to regenerate the description. If you retry, the newer description varies slightly in terms of the words used, but the meaning remains the same.
- Click **Replace** to add the generated HL7 message description to the **Description** field. If the **Description** field is empty, the generated HL7 message description is added to the field. If the **Description** field already includes a HL7 message description, the current description is replaced by the generated description. For this example, a description already existed. **Replace** is only selectable if the HL7 message is editable.

Message selection

Name
Healthcare_Document

Description
This is a healthcare document.

This healthcare document is an admit/visit notification. It provides details about a patient's admission or visit to a healthcare facility, including patient information and visit details.

Retry Replace Close

📘 **Note**

You cannot change the message standard or version.

4. Select the message schema to use. You can select the standard schema or one you created.

Message type
ADT_A08 (Update patient information)



Message schema
MYADT_A08_SCHEMA

Edit

5. Click **Save**.
This message is now available for selection when you configure the healthcare action in an integration.

Clone an HL7 message

1. To clone a message in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to clone the message.

- c. Click **Healthcare** .
 2. Go to the **HL7 messages** section.
 3. To clone a message in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **HL7 messages**.
 4. Hover over a message row to see the actions you can perform on a message.
 5. Click **Actions** , then select **Clone** to clone the message.
The Clone Healthcare message panel opens.
 6. Make any necessary updates, then click **Clone**.

4

Create Clinical Document Architecture (CDA) Documents

You create a CDA document as a starting point. You then select the document in a healthcare action in an integration.


Topics:

- [Create a Clinical Document Architecture Document](#)
- [Edit or Clone a Clinical Document Architecture Document](#)

See [About Clinical Document Architecture](#).

Create a Clinical Document Architecture Document

You can create a clinical document architecture (CDA) document in a project or a standalone environment (outside of a project). You then select the document in a healthcare action in an integration.

1. To create a CDA document in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to create the CDA document.
 - c. Click **Healthcare** .
 - d. In the **CDA documents** section, click **Add**.
2. To create a CDA document in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **Schemas**.
 - b. On the Schemas page, click **Create**.
3. Enter the following details.

Element	Description
Name	Enter a CDA document name.
Identifier	This field is automatically populated with a unique schema identifier based on your CDA document name. You can manually change this value, if needed.
Description	Enter an optional description of this CDA document.
Standard	View the HL7v3 CDA (with extensions) document standard. Only CDA is available and it cannot be deselected.

Element	Description
Version	<p>Select the CDA document version to use.</p> <ul style="list-style-type: none"> • R2 (Normative - No Extensions): The Release 2 (R2) version of the standard consisting of the officially recognized elements, data types, and structures. The standards published by the governing body are followed and there are no additional elements or customizations included. • R2 SDTC Approved Extensions: The R2 standard that includes the officially recognized elements, data types, and structures, along with a set of extensions approved by the Standards Development and Testing Committee (SDTC) for specific purposes. These are not arbitrary or custom extensions, but are officially sanctioned, providing additional functionality or compatibility (for example, to support regulatory requirements or specific workflows).
Type	<p>View the document type. Only ClinicalDocument (Base document type for CDA) is available and it cannot be deselected. This type is the root XML element for all CDA documents, and provides a standardized structure for the exchange of clinical documents such as discharge summaries, progress notes, and continuity of care documents.</p>
Schema	<p>View the schema standard. Only Standard is available and it cannot be deselected.</p>
Identification expression (XPath)	<p>If you want to distinguish between different types of CDA XML documents, use this XPath expression and provide an identification value in the Identification value field (for example, 34133-9). This value is consumed by the 'code' parameter in the XPath expression. This value distinguishes between different types of CDA documents because the structure of CDA is the same.</p> <pre>/*[local-name()='ClinicalDocument']/*[local-name()='code']/@code</pre> <p>If you do not want to distinguish between the different types, then use the default CDA document root element of <code>ClinicalDocument</code>:</p> <pre>/*[local-name()='ClinicalDocument']</pre>
Identification value	<p>Optionally enter an identification value if you want to distinguish between different types of CDA XML documents in the Identification expression (XPath).</p>
Optional Schematron [Validation] Files (zipped)	<p>You can customize the CDA document with localized extensions and upload an XSD zip file.</p>

Create CDA document

What is it called?

The name can be changed later. The identifier can be set only now and it must be unique.

Name
PatientNotes

Identifier
PATIENTNOTES

What does it do?

Description
Sends patient notes

Standard
CDA

Version
R2 SDTC Approved Extensions

Type
ClinicalDocument (Base document type for

Schema
Standard

Identification expression (XPath)
/*[local-name()='ClinicalDocument']/*[local-name()]

Identification value

Required

Optional Schematron [Validation] Files(zippped)

Drag and Drop
Select a .zip file or drop one here

4. Click **Create**.



The CDA document selection is displayed. You can edit this document, as needed. See [Edit or Clone a Clinical Document Architecture Document](#).

The created CDA document is now selectable in the healthcare action in the integration canvas. No explicit adapter configuration is required. The REST Adapter is implicitly used to pass the CDA document through the healthcare action. Because the CDA document is already in XML format, no translation is required.

Edit or Clone a Clinical Document Architecture Document

You can edit an existing CDA document to update the name, description, and other properties. You can also clone a CDA document to create a copy of it.

Edit a CDA Document

1. To edit a CDA document in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to edit the CDA document.
 - c. Click **Healthcare** .
 - d. Go to the **CDA documents** section.
 - e. Hover over the CDA document to edit.
 - f. Click **Actions** , then select **Edit**.
2. To edit a CDA document in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **Schemas**.
 - b. Click the CDA document name to edit.
3. Edit the document's name, description, schema, and identification expression and value, as necessary.

PatientNotes

<small>Identifier</small> PATIENTNOTES	<small>Updated on</small> Oct 5th, 2025, 01:47:36 PM PDT	<small>Project</small> MY_FIFO_PARENT_CHILD_TEST
---	---	---

CDA document selection

Name
PatientNotes

Description
Sends patient notes

Standard
CDA

Version
R2 SDTC Approved Extensions

Type
ClinicalDocument (User Defined)

Schema
Standard

▼ **Identification**


Identification type
XML

Identification expression (XPath)
/*[local-name()='ClinicalDocument']/*[local-name()='code']/@code

Identification value
1344

4. Click **Save**.

Clone a CDA Document

1. To edit a CDA document in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to clone the CDA document.
 - c. Click **Healthcare** .
 - d. Go to the **CDA documents** section.
 - e. Hover over the CDA document to clone.
2. To clone a CDA document in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **Schemas**.

5

Design an Integration with a Healthcare Action

Once you have created your schema and message, you can design an integration with a healthcare action.

Topics:

- [Best Practices for Designing a Healthcare Integration](#)
- [Create Trigger and Invoke Connections that Support the HL7 and FHIR Standards](#)
- [Handle Inbound Debatching of Large HL7 Message Payloads](#)
- [Convert HL7 Messages with a Healthcare Action](#)

Best Practices for Designing a Healthcare Integration

Note the following best practices for designing an optimal healthcare integration.

- [Build a Parent Routing Integration to Invoke Child Integrations that Process a Single HL7 Inbound Message](#)
- [Use a Lookup Function in the Mapper to Select the Child Integration to Invoke](#)
- [Add Fault Handling to Address Inbound Messages that are Not Selected in the Healthcare Action](#)
- [Plan for Future Oracle Integration for Healthcare Updates](#)

Build a Parent Routing Integration to Invoke Child Integrations that Process a Single HL7 Inbound Message

You may design an integration that receives many different HL7 inbound message types. For example:

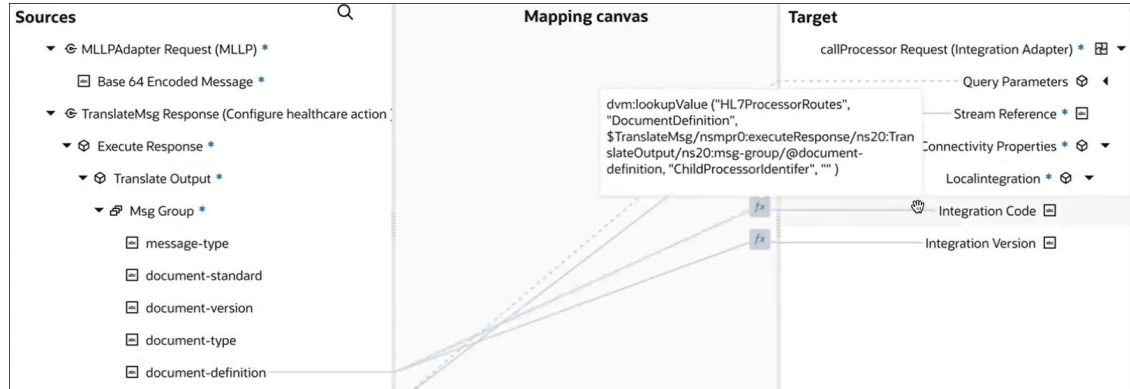
- ADT_A05 (Pre-admit a patient)
- ADT_A08 (Update patient information)
- ADT_A09 (Patient departing - tracking)

As a best practice, build a parent routing integration that invokes child integrations that each process a single HL7 inbound message type. For this example, create three child integrations. Do not attempt to build a single integration with a routing action (for example, a for-each action or switch action) to process each message.

Create your child integration with a REST Adapter trigger connection that can accept a binary format as its request payload. Select the default application/octet-stream as the mime-type. You pass the message-reference object that you get back from calling the healthcare action into your child integration. See [Overview of the Integration Receiving the Inbound HL7 Message During Runtime](#).

Use a Lookup Function in the Mapper to Select the Child Integration to Invoke

Use a lookup function in the mapper to select the child integration to invoke. Based on the message and message type, the lookup function returns the name of the integration (**Integration Code**) and the version of the integration (**Integration Version**) to invoke.



In the lookup table, based on the specific inbound message received, a specific integration is invoked.

HL7ProcessorRoutes
Used by HL7 handler integrations to lookup which child integration (processor) to call to process the incoming message.

Configured | Updated on 6 June 2023, 07:15:53 PM KST | Used in 3 integrations

	DocumentDefinition	ChildProcessorIdentifier	ChildProcessorVersion
	ADT_A08_PATIENT_UPDATE	PROCESSADT_A08	1.0
	ADT_A08_PATI_UPDA_V2_3_1	PROCESSADT_A08	1.0
	A01_ADMIT_PATIE_2_3_1	PROCESSADT_A01	1.0
	A01_ADMIT_PATIEN_2_5	PROCESSADT_A01	1.0
	A08_UPDAT_PATIE_2_3_1	PROCESSADT_A08	1.0
	A08_UPDATE_PATIEN_2_5	PROCESSADT_A08	2.0

Add Fault Handling to Address Inbound Messages that are Not Selected in the Healthcare Action

A healthcare action is not designed to throw faults. You should design fault handling logic into your integration to catch message faults. For example, assume you design the following integration to receive inbound messages from an HL7 application:



In the healthcare action, you select the following two inbound message types for translation:

- ADT_A08 (Update patient information) version 2.5

- ADT_A08 (Update patient information) version 2.3.1

View
Configure healthcare action call

Name
translateHL7

Description

Operation
Match and translate inbound message

Choose documents

Available	Selected
PATIENT_CHANGE	A08_PATIE_UPDATE_2_5
AA_TEST	A08_PATIE_UPDAT_2_3_1
AA_ADT_A01_DEMO	
AA_PATIENTADMIT28	
AA_PATIENTREGISTRAT	

Assume the healthcare action also receives an inbound ADT_A04 (Register a patient) message from the HL7 application. Because this message type has not been selected in the healthcare action, it cannot be translated into an Oracle Integration payload. The ADT_A04 message type is passed through the healthcare action, but fails in the subsequent map action with errors.

Errors

- Error Code *
- Error Description *

As a best practice, add and design a switch action immediately after the healthcare action and before the map action with the following branch logic:

- Route 1: If the message is not selected in the healthcare action, then address the fault in that branch with your own fault handling logic. For example, add a notification action that sends an email to the inbound HL7 application informing them that an unplanned message was received.
- Otherwise: If the message is selected in the healthcare action, translate it into an Oracle Integration payload, perform mapping, and call the child integration for further processing and delivery to the outbound application endpoint.

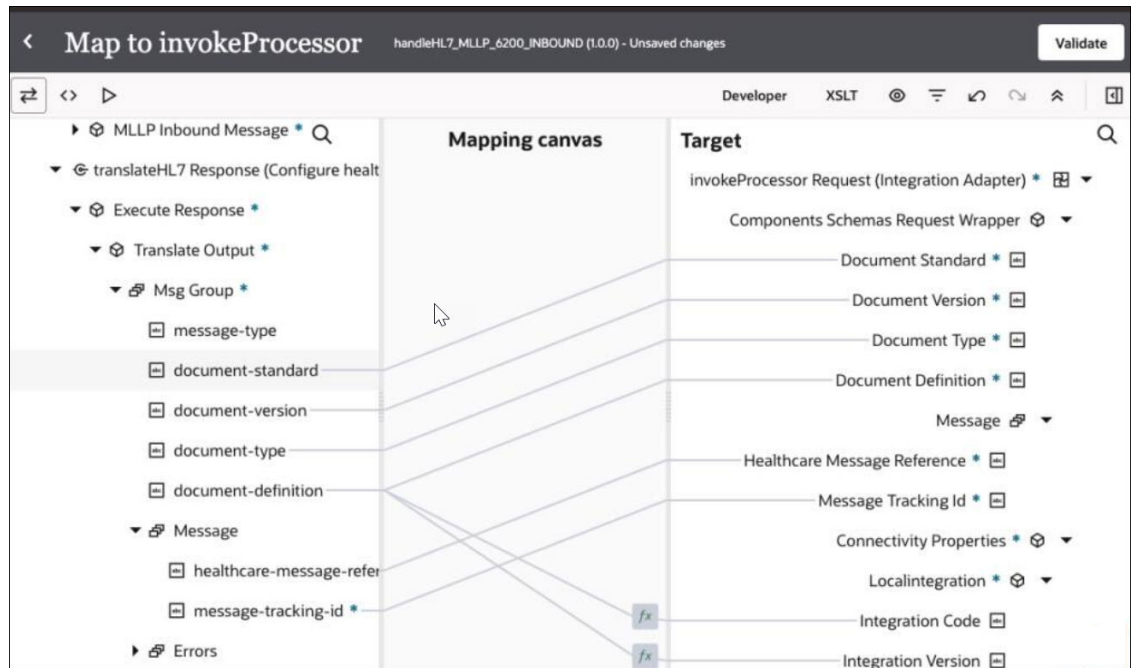
Plan for Future Oracle Integration for Healthcare Updates

A common interface payload to use for the invocation from the handler integration to the child (processor) integration is provided. This JSON document forms the contract between the handler and the processor integrations.

```
<strong>{
  "document-standard": "HL7V2",
  "document-version": "2.3.1",
  "document-type": "ADT_A01",
  "document-definition": "HL7V2_ADT_A01",
  "message": [
    {
      "healthcare-message-reference": "clm:base64_meta.base64_payload1",
      "message-tracking-id": "123"
    },
    {
      "healthcare-message-reference": "clm:base64_meta.base64_payload2",
      "message-tracking-id": "abc"
    }
  ]
}</strong>
```

Use of this JSON message in your child integrations is not required. However, use of this message enables you to take advantage of potential future features. The idea is that if you use this JSON message as the contract in your child integrations now, change are not required in the future. You do not need to create a new handler integration. The incoming HL7 message is routed to the appropriate processor integration based on your routing rules. For this to work, you need to build your integration using this JSON payload.

This payload provides metadata about the message and capability to also support batch messages. You map this data using the data returned from the healthcare action when you call the operation **Match and translate inbound message**. An example of this mapping is shown below.



Create Trigger and Invoke Connections that Support the HL7 and FHIR Standards

You must create trigger (inbound) and invoke (outbound) connections to communicate with applications that support the HL7 or FHIR standards.

- [Create MLLP Adapter Connections](#)
- [Create a FHIR Connection](#)

Create MLLP Adapter Connections

You can create MLLP Adapter trigger and invoke connections for integrations to communicate with health care systems that use HL7 version 2. Separate adapter connections are required for a trigger (incoming) and an invoke (outgoing).

- Trigger connection: The MLLP Adapter listens for and receives inbound HL7 messages from an application.
- Invoke connection: The MLLP Adapter establishes a connection and sends outbound HL7 messages to an application.

See [Create an MLLP Adapter Connection](#) in *Using the MLLP Adapter with Oracle Integration 3*.

Create a FHIR Connection

You can exchange health care data using the FHIR standard in XML or JSON. You configure the FHIR Adapter to allow FHIR resources (for example, a patient resource, encounter resource, and others) to be used as outbound (invoke) messages in Oracle Integration. The is similar to the REST Adapter, but with a FHIR-specific configuration experience.

As an alternative, you can also configure the REST Adapter to allow FHIR resources to be used as inbound (trigger) and outbound (invoke) messages.

- See Create a FHIR Adapter Connection in *Using the FHIR Adapter with Oracle Integration 3*.
- See Create a REST Adapter Connection in *Using the REST Adapter with Oracle Integration 3*.

Handle Inbound Debatching of Large HL7 Message Payloads

If your inbound HL7 message includes a large payload, you can debatch (split) the payload into manageable messages if you select the **Split payload into chunks** operation in the healthcare action. This operation enables your inbound message to be successfully converted to XML format without errors in the healthcare action.

Because the native payload size can be large, the **Split payload into chunks** operation enables you to split the payload into smaller messages for processing. You can specify the number of messages into which to split the payload and the size for each message. The split into smaller messages occurs before translation from the native payload into XML format. For example, assume you have a payload that is 4 MB in size. You can specify to split the message into 4 messages, each of 1 MB in size.

The following types of inbound message format are supported for debatching of large payloads.

Supported Message Format	How Debatching Works
<p>A payload that contains multiple batch headers within a file header and multiple message header segments within a batch header:</p> <pre>[File Header] { [Batch Header] { [Message Header Segment] } [Batch Trailer] }</pre>	<ul style="list-style-type: none"> • Elements in a file header segment and batch header segment are gathered and each individual message header segment is debatched for the transaction. • The information (file header, batch header, and message header segment transaction) is passed to the healthcare action for conversion. • The HL7 (message header segment transaction) format is then converted into XML format and the values from the file header, batch header, file trailer, and batch trailer are set in headers and trailers when the transformed XML is returned from the healthcare action.
<p>A batch payload that contains multiple message header segments:</p> <pre>{ [Message Header Segment for HL7 Message] }</pre>	<ul style="list-style-type: none"> • Each message header segment for the HL7 message transaction is passed to the healthcare action. • The healthcare action returns HL7 in XML format without header and trailer information because the incoming batch does not have file header, batch header, file trailer, and batch trailer segment information.

See Convert HL7 Messages with a Healthcare Action.


Convert HL7 Messages with a Healthcare Action

Convert a message to or from HL7 format with a healthcare action in an integration. The healthcare action converts native inbound HL7 messages into Oracle Integration XML payloads and generates native HL7 messages from Oracle Integration XML payloads to send

to external applications that support HL7. XML-formatted CDA messages can also be passed through the healthcare action.

Note

If your Oracle Integration instance does not include the Healthcare edition, you cannot drag the healthcare action into an integration. See [Create an Oracle Integration Instance in *Provisioning and Administering Oracle Integration 3*](#).

1. To create an integration in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to create the integration.
 - c. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist.
 - d. Click **Create**, then select **Application** or **Schedule**.
2. To create an integration in a standalone environment:
 - a. In the navigation pane, click **Design**, then **Integrations**.
 - b. Click **Create**, then select **Application** or **Schedule**.
3. Design your integration, including adding and configuring your connections. See [Create Trigger and Invoke Connections that Support the HL7 and FHIR Standards](#).
4. Add a healthcare action to the integration in either of the following ways:
 - On the right side of the canvas, click **Actions**  and drag the **Healthcare** action to the appropriate location.
 - Click **+** at the location where you want to add the healthcare action, then select **Healthcare**.

The Configure healthcare action call wizard is displayed.

5. Enter the following details:

Element	Description
Name	Enter a name.
Description	Enter an optional descriptions for what this action does.

Element	Description
Operation	<p>Select an operation to perform:</p> <ul style="list-style-type: none"> • Convert message reference to document: Use this operation to identify the incoming message type and version. This operation returns a reference to a translated message and metadata needed to route the integration to an appropriate child integration to process the message. When you call the operation, you need to specify a list of HL7 or CDA documents with which you want to match. • Match and translate inbound message: Use this operation to convert inbound HL7-formatted messages into XML format or pass through XML-formatted CDA documents. Specify the HL7 message or CDA document you want this action to receive. These are the documents that you configured during design time. This operation takes the following: <ul style="list-style-type: none"> – A raw HL7 message off the wire, parses it, and matches it with the document definition you select in the Choose documents section. The message is then converted to an Oracle Integration XML payload. – An XML-formatted CDA document and passes it through the healthcare action. • Translate to outbound message: Use this operation to convert an XML payload message to an outbound HL7 message or pass through an XML-formatted CDA document. • Split payload into chunks: Use this operation if your inbound HL7 message includes a large payload and you need to debatch (split) the payload into manageable chunks. See Handle Inbound Debatching of Large HL7 Message Payloads.
Ignore non-modeled segments	<p>Select this checkbox for incoming HL7 messages that contain unexpected custom segments to be successfully processed. With this option selected, unexpected custom segments not found in the schema are ignored completely, no errors are raised, and a response is successfully generated at runtime.</p>
Choose documents This field is displayed if you select the Match and translate inbound message operation.	<p>Select <i>all</i> the HL7 message definitions that you expect to receive on the port number you configured for your MLLP Adapter connection. You can also select CDA documents. This integration receives and routes all messages sent to that port number. Documents that are not selected are ignored.</p> <p>If you created the integration in a project, only documents available in that project are visible for selection.</p>
Document This field is displayed if you select the Convert message reference to document or Translate to outbound message operation.	<p>Select the document to use.</p> <p>If you created the integration in a project, only documents available in that project are visible for selection.</p>
Split message count This field is displayed if you select the Split payload into chunks operation.	<p>Enter the number of messages into which to split the payload. A value between 10 and 10000 messages is supported. The default size is 1000 chunks.</p>
Split batch size This field is displayed if you select the Split payload into chunks operation.	<p>Enter the size for each message. A value between 1 and 100 MB is supported. The default is 1 MB.</p>

For example, if you select an ADT_A08 HL7 message:

- If you select **Match and translate inbound message**, the selected message document ADT_A08 message document is converted from HL7 format to an Oracle Integration XML-formatted message.

The screenshot shows the 'Configure' interface for a healthcare action call. The title is 'Configure' with a subtitle 'Configure healthcare action call'. The interface includes several input fields and a document selection area.

Name: patientUpdates

Description: Update patient details

Operation: Match and translate inbound message


Choose documents: Search available documents

Available documents: VLANTIGU_DOC

Selected documents: MYADT_A08_DOC

Navigation buttons between the document lists include: >, >|, <, and <|.

- If you select **Translate to outbound message**, the selected ADT_A08 message is converted from an Oracle Integration XML message to an HL7-formatted message for delivery to an external application that supports HL7.



Configure

Configure healthcare action call

Name
patientUpdates


Description
Update patient details

Operation
Translate to outbound message

Document
MyADT_A08_Doc

Selected Document Definition has Standard 'HL7V2', Version '2.9', Type 'ADT_A08 (Update patient information)'.

- If you select **Convert message reference to document**, the message reference is converted into a mappable payload. This operation returns the contents of the HL7 message as a variable in your integration that is ready to use in the business logic and mapper of your integration. The variable appears on the source side of the mapper.



Configure

Configure healthcare action call

Name
patientupdates

Description

Operation
Convert message reference to document

Document
MyADT_A08_Doc

Selected Document Definition has Standard 'HL7V2', Version '2.9', Type 'ADT_A08 (Update patient information)'.

- If you select **Split payload into chunks**, the payload of the inbound HL7 message is debatched (split) into manageable chunks.

Configure

Configure healthcare action call

Name Required

Description

Ignore non-modeled segments for HL7 messages

Ignore non-modeled segments

Operation
Split payload into chunks

Split Message Count
1000

Split Batch Size
1

6. Click **Finish**.
7. Continue designing the remaining parts of your integration, including adding and configuring your invoke connections.

Examples of using all three operations in a use case are provided. See Design Time and Runtime Overview of an Oracle Integration for Healthcare Use Case.

6

Import FHIR Profile Packages and Create Custom FHIR Resources

You can import FHIR profile packages and create custom FHIR resources in Oracle Integration. You then select the profile package or custom FHIR resource to use when configuring the FHIR Adapter in an integration.

Topics:

- [Import a FHIR Profile Package](#)
- [Browse the Contents of the FHIR Profile Package](#)
- [Create Custom FHIR Resources](#)
- [Edit a Custom FHIR Resource](#)

See [Profile Definitions and Documentation](#).


Watch a video to learn more:



Import a FHIR Profile Package

You can import a FHIR profile package into Oracle Integration and use the resource definitions in that profile with the FHIR Adapter.

The HL7 FHIR standard defines a set of base resources (for example, patient and observation). The standard base resources have generic definitions. FHIR profile packages enables you to create customized resource definitions by specifying a set of constraints and/or extensions on a base resource.

1. Import a FHIR profile into a project.
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to import the FHIR profile package.
 - c. Click **Healthcare** .
 - d. In the **FHIR profiles** section, click **Add**.
2. Import a FHIR profile into a standalone environment.
 - In the navigation pane, click **Healthcare**, then **FHIR profiles**.
3. Click **Import**.

The Import FHIR profile panel opens.
4. Enter the following details.

Element	Description
Drag and Drop	Click to select a TGZ file from your host or drag and drop a file into the field. You cannot import the same profile package more than once.
Description	Enter an optional description of this profile package.

5. Click **Import**.


Import of the FHIR profile package occurs in the background and can take up to five minutes.


Once imported, the state changes to **Configured**. You can browse the contents of the profile package. See [Browse the Contents of the FHIR Profile Package](#).

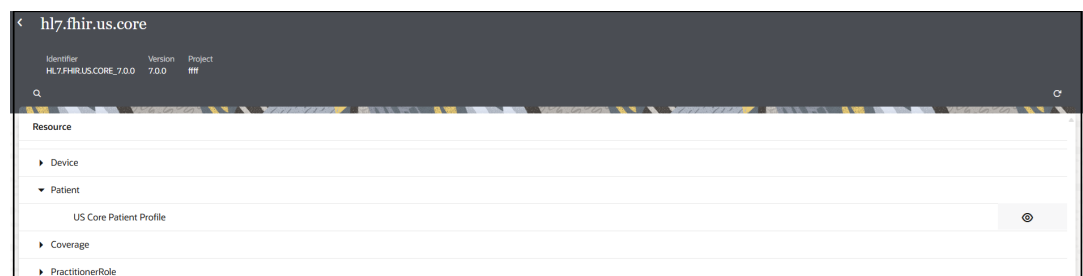
The profile package is now available for selection when you configure the FHIR Adapter in an integration. See Add the FHIR Adapter Connection to an Integration in *Using the FHIR Adapter with Oracle Integration 3*.

Browse the Contents of the FHIR Profile Package

You can browse the contents of the imported FHIR profile package.


1. Browse a FHIR profile package in a project.
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to browse the FHIR profile package.
 - c. Click **Healthcare** .
 - d. Go to the **FHIR profiles** section.
2. Browse a FHIR profile package in a standalone environment.
 - In the navigation pane, click **Healthcare**, then **FHIR profiles**.
3. Click the FHIR profile package to browse.

The supported FHIR resources are shown.
4. Expand a resource and click **View** .




The segments and elements for the resource are displayed. This structure matches the Snapshot Table of the US core patient profile in the [FHIR specification](#). Elements that include asterisks (for example, **race** and **ethnicity**) are known as extensions. Extensions provide a way to add user-defined data elements to FHIR resources and data types when the core FHIR standard doesn't provide a suitable field. Extensions allow you to customize and extend the FHIR standard without modifying the core structure.

Segment/Element	Name	Cardinality	Element type
▼ Patient	Patient	[1..1]	
□ resourceType	resourceType	[0..1]	
□ id	id	[0..1]	id
▶ meta	meta	[0..1]	Meta
□ implicitRules	implicitRules	[0..1]	uri
□ language	language	[0..1]	code
▶ text	text	[0..1]	Narrative
▶ * race	race	[0..>+1]	Extension
▶ * ethnicity	ethnicity	[0..>+1]	Extension
▶ * tribalAffiliation	tribalAffiliation	[0..*]	Extension

5. Expand an extension to view the associated data types.
6. Hover your cursor over a row and click **View**  to view property details.

Create Custom FHIR Resources

You can create your own custom FHIR resources. This feature enables you to customize a FHIR resource from the underlying FHIR base schema to use in integrations. You must first import a FHIR profile package to create a custom FHIR resource. A FHIR resource is analogous to an HL7 version 2 message.

1. Create a custom FHIR resource in a project.
 - a. In the navigation pane, click **Projects**.
 - b. Click **Healthcare** .
 - c. In the **Custom FHIR resources** section, click **Add**.
2. Create a custom FHIR resource in a standalone environment.
 - a. In the navigation pane, click **Healthcare**, then **Custom FHIR resources**.
 - b. Click **Create**.
3. Enter the following details, then click **Create**.

Element	Description
Name	Enter a custom FHIR resource name.
Identifier	This field is automatically populated with a unique identifier based on your name. You can manually change this value, if needed.
Description	Enter an optional description of this custom FHIR resource.
FHIR schema source	Select a schema source. <ul style="list-style-type: none"> • Standard schema definition • FHIR profile
FHIR resource	Select the resource within the standard schema definition or FHIR profile.
FHIR profile package Appears only if you select FHIR profile .	Select the profile package you imported.

Element	Description
Profile Appears only if you select FHIR profile .	Select the profile.

The **Contained resource selection** section is displayed. In the FHIR resource structure, there is a field called `contained` used to include other resources within the same document, effectively embedding them. This is useful when the referenced resources don't have independent existence and are logically part of the containing resource.

You can add another resource to be contained in the FHIR resource you selected in Step [3](#).

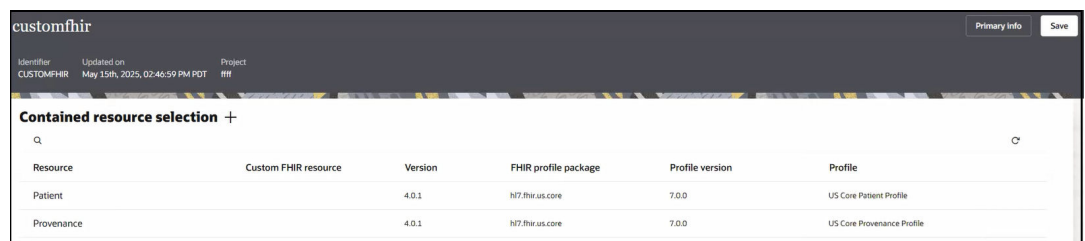
- Click **Add** .

The Add resource panel is displayed.

- Enter the following details.

Element	Description
FHIR schema source	Select a schema source. <ul style="list-style-type: none"> Standard schema definition FHIR profile Custom FHIR resource
FHIR profile package Appears only if you select FHIR profile .	Select the profile package you imported.
FHIR resource	Select the resource within the standard schema definition or FHIR profile.
Profile Appears only if you select FHIR profile .	Select the profile.
Custom FHIR resource Appears only if you select Custom FHIR resource .	Select the resource. The Description and FHIR resource fields are displayed in read-only mode.

- Click **Add**, then click **Done**. Add more resources, as required by your business needs. For this example, **Patient** and **Provenance** were added. You cannot add the same resource type (for example, **Patient**) more than once.



Contained resource selection +					
Resource	Custom FHIR resource	Version	FHIR profile package	Profile version	Profile
Patient		4.0.1	hl7.fhir.us.core	7.0.0	US Core Patient Profile
Provenance		4.0.1	hl7.fhir.us.core	7.0.0	US Core Provenance Profile

- Click **Save**.

The custom FHIR resource is now available for selection when you configure the FHIR Adapter in an integration. See [Add the FHIR Adapter Connection to an Integration in Using the FHIR Adapter with Oracle Integration 3](#).

You can also add heterogeneous bundles to custom FHIR resources. A FHIR heterogeneous bundle is a resource that groups multiple entries, each of which can contain any type of FHIR resource (such as patient, bundle, observation, or medication).

8. Return to the Custom FHIR resources page, and create another custom FHIR resource.
9. Create a heterogeneous bundle resource. For this example, the following selections are made:
 - Select **Standard schema definition** from the **FHIR schema source** list.
 - Select **Bundle** from the **FHIR resource** list.
 - Select **Searchset** from the **Bundle type** list.

Create custom FHIR resource

Name
custom_bundle

Identifier
CUSTOM_BUNDLE

What does it do?

Description

FHIR schema source
Standard schema definition ▼

FHIR resource
Bundle ▼

Bundle type
searchset ▼

The **Bundle entry resource selection** section is displayed.

10. Click **Add** .

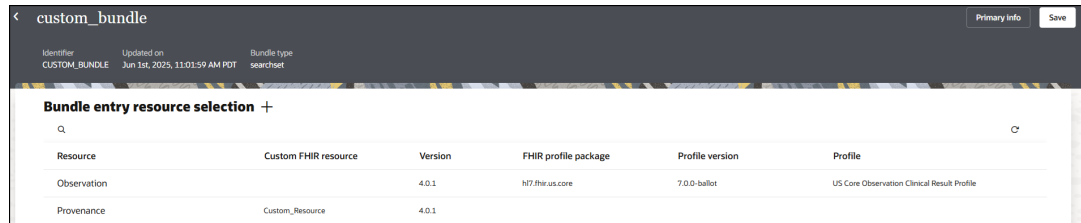
The Add resource panel is displayed.

11. Enter the following details.

Element	Description
FHIR schema source	Select Custom FHIR resource .
Custom FHIR resource	Select a different custom FHIR resource. The Description and FHIR resource fields are in read-only mode.

12. Click **Add**, then click **Done**.

The contents of the custom FHIR resource heterogeneous bundle are displayed. You can add more resources to the heterogeneous bundle, based on your business requirements. For this example, **Observation** and **Provenance** were added.






Resource	Custom FHIR resource	Version	FHIR profile package	Profile version	Profile
Observation		4.0.1	hl7.fhir.us.core	7.0.0-ballot	US Core Observation Clinical Result Profile
Provenance	Custom_Resource	4.0.1			

The custom FHIR resource heterogeneous bundle is now available for selection when you configure the FHIR Adapter in an integration. See *Add the FHIR Adapter Connection to an Integration in Using the FHIR Adapter with Oracle Integration 3*.

Edit a Custom FHIR Resource

You can edit a custom FHIR resource to update the name and description.

1. To edit a custom FHIR resource in a project:
 - a. In the navigation pane, click **Projects**.
 - b. Click the project in which to edit the custom FHIR resource.
 - c. Click **Healthcare** .
 - d. Go to the **Custom FHIR resources** section.
 - e. Click **Actions** , then select **Edit**.
2. To edit a custom FHIR resource in a standalone environment:
 - a. In the navigation pane, click **Healthcare**, then **Custom FHIR resources**.
 - b. Hover over a custom FHIR resource row to see the actions you can perform on it.
 - c. Click **Edit** .
3. Click **Primary Info**.
4. Edit the custom FHIR resource's name and description, as necessary.

You can use AI to generate a new description or replace an existing description of a custom FHIR resource with a single click. Click **Generate** to create an AI-powered description.

Note

AI features are available in select regions. See *AI Feature Availability in Provisioning and Administering Oracle Integration 3*.

The **Description** field can be empty or already include a description.

- If the **Description** field is empty, the generated custom FHIR resource description is added to the field.

- Click **Replace** to add the generated custom FHIR resource description to the **Description** field. If the **Description** field is empty, the generated resource description is added to the field. If the **Description** field already includes a description, the current description is replaced by the generated description. For this example, a description already existed. **Replace** is only selectable if the custom FHIR resource is editable.

Edit info

Name
Custom_Resource

Identifier
CUSTOM_RESOURCE

sion

This healthcare custom FHIR resource is Custom_Resource based on Provenance from hl7.fhir.us.core (7.0.0-ballot) US Core Provenance Profile. This includes various resources such as AllergyIntolerance from hl7.fhir.us.core (7.0.0-ballot) US Core AllergyIntolerance Profile and Provenance from hl7.fhir.us.core (7.0.0-ballot) US Core Provenance Profile.

Retry Replace Close

Note

You cannot change the resource standard or version.

5. Click **Save**.

7

Ensure Messages are Processed in the Received Order

Oracle Integration provides a processing mechanism to ensure that HL7 messages are delivered to target systems in the order received. For example, an ADT_A01 (Patient Admitted) message is received and routed to multiple departments in the correct order, first to a lab, then to hospital management, and then to other departments.

Topics:

- [About Ordered Message Processing in Oracle Integration](#)
- [Create and Add Message Queues to an Integration](#)
- [Monitor and Manage Queued Messages at Runtime](#)
- [Ordered Message Processing Use Cases](#)

About Ordered Message Processing in Oracle Integration

To ensure that HL7 messages are delivered to target endpoints in the order in which they are received, Oracle Integration implements ordered message processing. Messages are stored in queues to ensure that they are processed in the order received.

- [Capabilities](#)
- [Use Case](#)
- [Restrictions](#)

Capabilities

It's critical for healthcare messages to be delivered in the correct order to ensure patient safety. Out-of-order messages can result in serious consequences, such as the incorrect administration of medication or treatment. For example, receiving a medication order message out of order with related allergy information may lead to a serious adverse reaction. To prevent out-of-order messaging from occurring, Oracle Integration uses ordered message processing to deliver messages in the correct order.

Ordered message processing utilizes the following key components:

Component	Description
Queues	A processing mechanism that ensures that messages (specifically, HL7 messages in Oracle Integration) are processed in the order received.
Policy	Attached to MLLP Adapter trigger and invoke connections to enforce guaranteed ordering of messages.
MLLP Adapter trigger connection	Receives and stages HL7 messages in a queue in the order received. Upon receiving the message, the MLLP Adapter also sends an acknowledgment back to the client application.
MLLP Adapter invoke connection	Attached to the queue to receive and route HL7 messages to one or more target endpoints in the order staged in the queue.

Ordered message processing in Oracle Integration provides the following capabilities:

- Message order preservation
 - Delivers each message to the target endpoint in the same order as sent by the client application.
 - If one message being processed runs into an error, subsequent messages are blocked from being processed to preserve the order.
 - If a message recovery occurs on a message after the first failure, the message sequence is maintained. Subsequent messages cannot go through until the first message is successfully processed. Once the failed message is successfully processed, messaging resumes in the same order.
- Delivery guarantees
Every message is ensured to be delivered to the target system at least once.
- Duplicate message and error handling
Duplicate messages adhere to the queue order. Duplicate messages may occur under the following scenarios:
 - A message is sent and stored in a queue, and an acknowledgment is sent to the client application, but never received.
 - A message is sent and stored in a queue, but a failure occurs when sending the acknowledgment to the client application.
 - A message arrives, but fails to be added to the queue. The client never receives the acknowledgment.
 - A message fails transformation or translation. In this case, you must resubmit the failed message after making the necessary changes.
 - A message is sent to one queue, but fails to be sent to the second outbound queue. This scenario results in a fault and must be resubmitted.
- Support for ordered message processing for parent-to-child integration invocations (local invocations)
- Runtime monitoring
Monitoring of queues is supported, including the current message delivery status and the number of messages in the queue. You can pause delivery for a queue. You can also perform management tasks such as deleting messages from queues, resubmitting failed messages in a queue, and clearing all messages from a queue.

Use Case

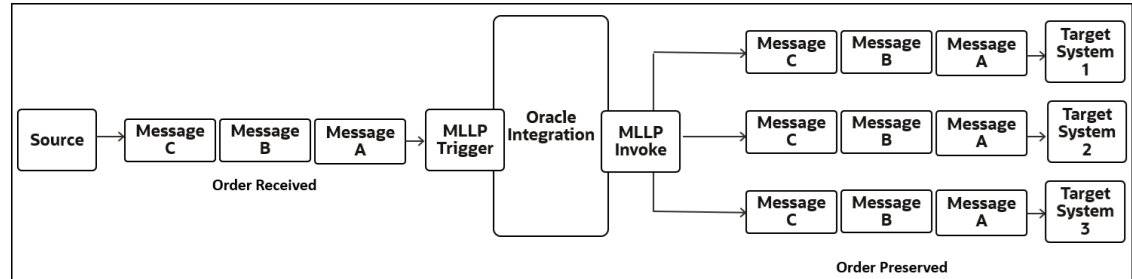
Let's take a quick look at a use case. A client application must send three messages in the correct order to three separate target system endpoints.

- Target system 1
- Target system 2
- Target system 3

Ordered message processing works as follows:

- The MLLP Adapter trigger connection receives and captures HL7 messages C, B, and A in that order in a queue and sends an acknowledgment back to the client application.
- The messages are processed in Oracle Integration.
- If the message must be routed to multiple destinations in the queued order, each destination MLLP Adapter invoke connection is associated with the policy.

- When consuming messages from the endpoint-specific queues, the system coordinates with the order of the messages in the queue upon arrival to maintain end-to-end order.
- The MLLP Adapter invoke connection delivers HL7 messages C, B, and A in the queued order to targets 1, 2, and 3, respectively. Ordering is enforced at endpoint processing from the outbound queues.



Restrictions

- Queues are only available for use with the Oracle Integration for Healthcare component of Oracle Integration.
- Queues are only supported with MLLP Adapter trigger and invoke connections. Note that:
 - If you configure an MLLP Adapter trigger connection to use a queue, but don't configure an MLLP Adapter invoke connection to use that queue, there is no end-to-end ordered message ordering.
 - You can attach a policy to an MLLP Adapter invoke connection, but if the invoke connection happens in an integration in which there is no MLLP Adapter trigger connection with a policy, the messages won't be placed in queues for re-ordering.
- You can only create and use queues in projects.
- Queues cannot be shared across projects.
- Partitioning of incoming messages is not supported. Each message is treated as an opaque entity. Message ordering is based solely on the time of arrival in Oracle Integration.
- Message processing can fail and be blocked (for example, if the target endpoints are not operational). The system tries to send the message again, which may be blocked again. Messages that get blocked because of retry exhaustion must be manually resubmitted.
- Editing of messages that fail due to missing data is not supported. Messages that cannot be processed due to such failures must be edited with the correct changes and resubmitted manually.

Create and Add Message Queues to an Integration

You create queues that you configure in the MLLP Adapter trigger and invoke connections during design time. At runtime, messages are staged in the queues in the order in which they are received before delivery to target endpoints.

The design process for creating and including queues in an integration is as follows:

- [Create MLLP Adapter Trigger and Invoke Connections](#)
- [Create a Queue](#)
- [Add the Queue to an Integration](#)

- [Manage Queues](#)

Create MLLP Adapter Trigger and Invoke Connections

Only the MLLP Adapter is supported for using queues. Create an MLLP Adapter with the **Trigger** role and an MLLP Adapter with the **Invoke** role. See *Create a Connection in Using the MLLP Adapter with Oracle Integration 3*.

Connections				+
	mlp anh_mllp_trigger	Trigger	Configured	...
	mlp anh_mllp_invoke	Invoke	Configured	...

Create a Queue

1. In the navigation pane, click **Projects**.
2. Select the project name.
3. Click **Integration**
4. Scroll down to the **Queues** section, and click **Add** if no queues currently exist or **+** if queues already exist. The Create queue panel opens.
5. Enter a name and optional description, then click **Create**. The queue is displayed in the **Queues** section.

Add the Queue to an Integration

1. In the navigation pane, click **Projects**.
2. Select the project name.
3. Click **Integration**
4. In the **Integrations** section, click **Add** if no integrations currently exist or **+** if integrations already exist. The Add integration panel opens.
5. Click **Create**, then select **Application**. The integration canvas is displayed.
6. Add the MLLP Adapter trigger connection to the integration.
7. Enter the following details, then click **Continue**.

Element	Description
Name	Enter a name.
Always show acknowledgment in Activity Stream	Select to show details about the acknowledge message that is sent to the client application in the activity stream.


Element	Description
Enable FIFO Queue	Select to use the queue created in Create a Queue .




- Click **Finish**.
- Add the MLLP Adapter invoke connection to the integration to consume the queue.
- Enter the following details, then click **Continue**.

Element	Description
Name	Enter a name.
Always show acknowledgment in Activity Stream	Select to show details about the acknowledge message in the activity stream.
Select a FIFO Queue	Select the queue to use.

- Click **Finish**.
- Continue designing your integration.
- Add a business identifier, then activate the integration.

Manage Queues

You can manage queues in integrations from the **Queues** section under **Integration**  in the project.

Task	How
Edit queues	Click Actions  , then Edit to update the name.
Delete queues	Click Actions  , then Delete . You cannot delete a queue currently being used by an integration.
View queue usage	Click Actions  , then View usage to see which integrations are using the queue. This option is only available if a queue is used by an integration.
Export a project that contains queues	You can export a project that includes a queue. See Export a Project in Using Integrations in Oracle Integration 3 .
Import a project that contains queues	You can import a project that includes a queue. See Create or Import a Project in Using Integrations in Oracle Integration 3 .
Clone a project that contains queues	You can clone a project that contains queues. The queue is retained in the cloned version. See Clone a Project in Using Integrations in Oracle Integration 3 .

You monitor and manage the messages in queues during runtime from the **Observe** tab. See [Monitor and Manage Queued Messages at Runtime](#).


Monitor and Manage Queued Messages at Runtime

You monitor the status of queued messages at runtime from the **Queues** tab of the **Observe** page in a project. You can also perform management tasks such as deleting messages from queues, resubmitting failed messages in a queue, and clearing all messages from a queue.

- In the navigation pane, click **Projects**.




2. Click the project that contains the queues to monitor.
3. Click **Observe**, then click **Queues**.


The **Queues** tab provides a centralized location for monitoring and managing queues and their associated messages. The queues, the messages in the queues currently being processed, and the delivery status of the queues are displayed.


4. Hover your cursor over the queue to view, and click **View Messages**  to display the associated messages and their current status. For this example, the status is **Pending**. If you refresh the page, the status should eventually change to **Sending**.

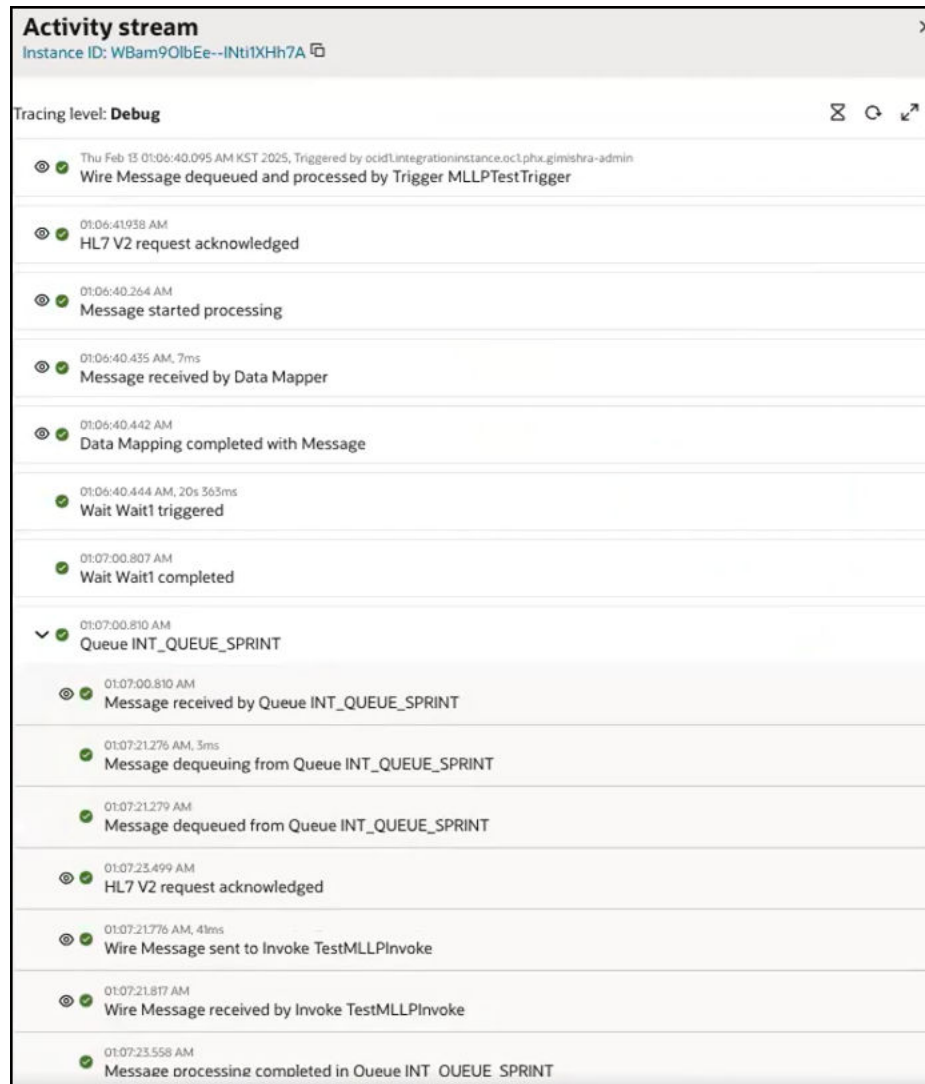
2 messages				
Message enqueued on	Instance created on	Integration	Instance ID	Status
Sep 30th, 2025 08:18:37 PM IST	Sep 30th, 2025 08:18:27 PM IST	SIMPLE_FLOW_1 01.00.0000	hgoYg54MEfCXXLO6Pmm5xw	Pending
Sep 30th, 2025 08:18:32 PM IST	Sep 30th, 2025 08:18:30 PM IST	SIMPLE_FLOW_1 01.00.0000	IA8QR54MEfCXXLO6Pmm5xw	Pending


During processing of queued messages, the delivery status of the queue can vary. You can also perform tasks such as resubmitting failed queued messages from inside a queue or at the instance level, deleting messages from a queue, or clearing all messages in a queue.


Delivery Status	Description
Active	<p>The queue is ready to accept messages. This is the default status of a queue. Only a user with ServiceAdministrator privileges can pause or resume a queue.</p> <p>When all the expected messages have been delivered to the queue, click View Messages  in the queue row to view the messages. Click Pause  if you need to pause the queue.</p>
Waiting	<p>Messages are in the process of being delivered to the queue. If you configured a wait action in your integration, the message waits for the specified time to complete before being delivered. The queue can remain in this state if other messages are waiting to be delivered to the queue (for example, if a second wait action is defined in the integration to wait 60 seconds before delivering the second message to the queue).</p> <p>If you click the Instances tab, the Status column shows a value of In progress for the message instance.</p> <p>After all messages in the integration are delivered to the queue, the queue status changes back to Active.</p>
Paused	<p>The flow of messages <i>out</i> of the queue has been stopped. However, the flow of messages <i>into</i> the queue has not stopped.</p> <p>If you click the Instances tab, the Status column shows a value of Pending Delivery for the inbound message instances.</p> <p>Click Resume  to reactivate the queue.</p>


Delivery Status	Description
Blocked	<p>The queue cannot process additional messages because there was an error with the last message received (for example, the connectivity agent was down or the target destination was unreachable). This status depends on the runtime value of the last message sent to the queue.</p> <p>Click View Messages  in the queue row to view the message in error. You can perform more granular management of messages from inside a queue, such as deleting a message or resubmitting a specific message, which causes all messages after that message to be resubmitted.</p> <ul style="list-style-type: none"> • To delete failed messages, see Step 9. • To resubmit failed queued messages from inside a queue or at the instance level, see Step 10.



5. Click the **Instances** tab to view the messages being processed through the queue. Every message in the queue is represented by a corresponding instance ID. The name of the queue used is displayed in the **Primary identifier** column. The **Filter**  provides a queue name filter to show all instances (messages) that were processed by a particular queue.
6. Expand the queue milestones in the activity stream. The activity stream shows the sequence of instance execution inside queues, including tracking milestones to indicate when the message was received, when message dequeuing began, and when message dequeuing completed. For this example, the **INT_QUEUE_SPRINT** queue is shown.




7. Scroll down further to view the MLLP Adapter invoke connection that received the wire message (for this example, named **TestMLLPInvoke**). A final message indicates that message processing has completed in **INT_QUEUE_SPRINT**.
8. If you want to clear all the messages in a queue:
 - a. Go to the **Queues** tab.
 - b. Find the queue to clear. This task is performed at the queue level.
 - c. Click **Clear** .
 - d. Click **Confirm** when prompted to clear the queue. This action deletes all messages in the queue.
 - e. Click the **Instances** tab.

Note that the status for each of the deleted queue messages is displayed as **Partially Aborted**.
9. If you want to delete a message in a queue:
 - a. Go to the **Queues** tab.
 - b. Hover your cursor over the queue, and click **View Messages** .

- c. Find the message to delete.
 - d. Click **Delete** .

The message is deleted from the selected queue. If that same message also appears in a different queue, it is considered a separate message and is not deleted.
10. If you want to resubmit a failed message, you can do so from inside a queue or at the instance level.
- From inside the queue:
- a. Go to the **Queues** tab.
 - b. Hover your cursor over the queue in which to resubmit a message, and click **View Messages** .
 - c. Click **Resubmit** .
 - d. Select the trace level when prompted, then click **Resubmit**.

This action resubmits the message, plus all messages that come after that message in the selected queue. Any messages that come before the resubmitted message are not resubmitted. For example, if there are four messages in a queue and you resubmitted the second message, then the third and fourth messages are also resubmitted, but the first message is not resubmitted.
- From the instance level:
- a. On the **Instances** tab, find the message instance to resubmit.
 - b. Select **Actions** , then **Resubmit** to resubmit the failed message. While this action is occurring, any messages delivered after the failed message are paused in the queue. Pausing ensures that messages are delivered and processed in the correct order. If the resubmit is successful, all messages resume delivery and processing through the queue in the correct order.

Ordered Message Processing Use Cases

This section provides a high-level walkthrough of several ordered message processing use cases.

- [Queue and Dequeue a Single Message](#)
- [Maintain Message Order When Calling a Child Integration from a Parent Integration](#)

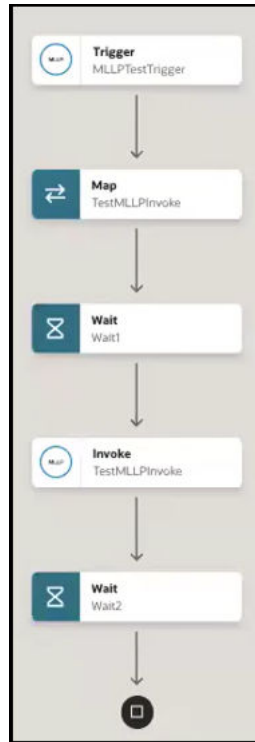
Queue and Dequeue a Single Message

This use case provides an overview of a single HL7 message being successfully queued and dequeued.

- A queue named **INT_QUEUE_SPRINT** was created in the **Queues** section under **Integration**  in a project.



- The design-time integration looks as follows:



- The MLLP Adapter trigger connection is configured with **Enable FIFO Queue** selected.
 - The first wait action is configured with a time delay of several seconds to simulate HL7 message processing time.
 - The MLLP Adapter invoke connection is configured with the **INT_QUEUE_SPRINT** queue selected from the **Select a FIFO Queue** list to queue messages in order.
 - The second wait action is configured with a time delay of several seconds to simulate HL7 message processing time.
- A tool such as HL7 inspector simulates a client application sending an HL7 message to Oracle Integration. The MLLP Adapter trigger connection listens for incoming messages.
- A message instance is created and visible on the **Instances** tab with a status of **In Progress**.

Primary Identifier	Instance ID	Business Identifiers	Status	Duration
message-control-id: A01 FIFO_SPRINT_DEMO_INTEGRATION 1.0.0	WBamP0BEe-iNiDh7A		In progress	Received In progress Just now

- The **Queues** tab shows there is one message in the expanded queue in a **Pending** state.

The screenshot shows the Oracle Integration Cloud interface with the 'Queues' tab selected. The main table displays the following data:

Name	Messages in queue	Delivery status
INT_QUEUE_SPRINT	1	Pending

Below the table, a detailed view of the message instance is shown:

Message created on	Instance ID	Status	Action
2025-02-12T16:06:40.195+0000	FIFO_SPRINT_DEMO_INTEGRAT 01.00.0000	Pending	

- Once the time delay in the wait action is reached, the message status changes to **Sending**.

The screenshot shows the Oracle Integration Cloud interface with the 'Queues' tab selected. The main table displays the following data:

Name	Messages in queue	Delivery status
INT_QUEUE_SPRINT	1	Sending

Below the table, a detailed view of the message instance is shown:

Message created on	Instance ID	Status	Action
2025-02-12T16:06:40.195+0000	FIFO_SPRINT_DEMO_INTEGRAT 01.00.0000	Sending	

- The **Instances** tab shows that the message instance completed successfully. The activity stream shows that the message was successfully received and dequeued in the queue.

Activity stream
Instance ID: WBam90lbEe--INti1XHh7A

Tracing level: **Debug**

- Thu Feb 13 01:06:40.095 AM KST 2025, Triggered by ocid1Integrationinstance.oc1.phx.gimishra-admin
Wire Message dequeued and processed by Trigger MLLPTestTrigger
- 01:06:41.958 AM
HL7 V2 request acknowledged
- 01:06:40.264 AM
Message started processing
- 01:06:40.435 AM, 7ms
Message received by Data Mapper
- 01:06:40.442 AM
Data Mapping completed with Message
- 01:06:40.444 AM, 20s 363ms
Wait Wait1 triggered
- 01:07:00.807 AM
Wait Wait1 completed
- 01:07:00.810 AM
Queue INT_QUEUE_SPRINT
- 01:07:00.810 AM
Message received by Queue INT_QUEUE_SPRINT
- 01:07:21.276 AM, 3ms
Message dequeuing from Queue INT_QUEUE_SPRINT
- 01:07:21.279 AM
Message dequeued from Queue INT_QUEUE_SPRINT
- 01:07:23.499 AM
HL7 V2 request acknowledged
- 01:07:21.776 AM, 4ms
Wire Message sent to Invoke TestMLLPInvoke
- 01:07:21.817 AM
Wire Message received by Invoke TestMLLPInvoke
- 01:07:23.558 AM
Message processing completed in Queue INT_QUEUE_SPRINT

Maintain Message Order When Calling a Child Integration from a Parent Integration

This use case provides an overview of how ordered message processing is maintained when a parent integration invokes a child integration (local invocation).

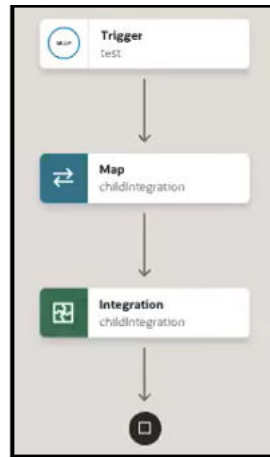
- A queue named **QUEUE_TEST_DEMO** was created in the **Queues** section under **Integration** in the project.

Queue Name	Queue Type	Status
QUEUE_TEST_DEMO	FIFO	Configured

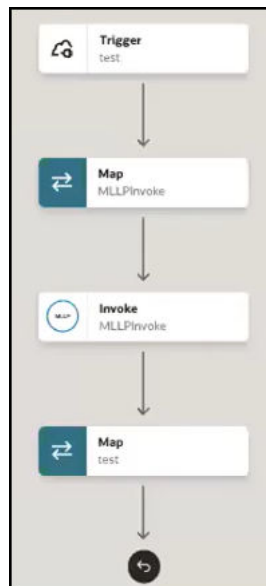
- The Integration section shows parent (**Parent_TEST_DEMO**) and child (**RestChild_TEST_DEMO**) integrations.



- The parent integration consists of an MLLP Adapter trigger connection configured with **Enable FIFO Queue** selected and an **Integration** action that calls a child integration (local invocation).



- The child integration includes an MLLP Adapter invoke connection is configured with the **QUEUE_TEST_DEMO** queue selected from the **Select a FIFO Queue** list to queue messages in order.



- A tool such as HL7 inspector simulates a client application sending an HL7 message to Oracle Integration. The MLLP Adapter trigger connection in the parent integration listens for incoming messages.
- The **Instances** tab shows that the parent and child integrations completed successfully.

The screenshot shows the Oracle Integration Cloud 'Instances' page. At the top, there are tabs for 'Deploy' and 'Observe'. Below the tabs, there are navigation links for 'Integrations', 'Instances', 'Queues', 'Subscriptions', 'Future runs', and 'Audit'. A search bar and filters are present, including 'Time window: Last 1 hour', 'Display instances: Retained', and 'Sort by: Last updated time'. The page title is 'Feb 13, 2025, 01:14:47 AM KST'. Below the filters, it says '4 instances'. The main content is a table with the following data:

Primary Identifier	Instance ID	Business Identifiers	Status	Duration
httprequestmethod: POST RestChild_TEST_DEMO 1.0.0	cjpldekEe-T6lBBEqs6A		Succeeded	Received Succeeded Duration Just now Just now 1.48 seconds
message-control-id: A01 Parent_TEST_DEMO 1.0.0	ce-q5ukEe-INiD0H7A		Succeeded	Received Succeeded Duration Just now Just now 749 milliseconds

- The activity streams for both instances shows that messages were successfully received and dequeued in the queue.

8

Design Time and Runtime Overview of an Oracle Integration for Healthcare Use Case

This section provides an overview of key design-time and runtime component capabilities in the healthcare use case that synchronizes EMR application patient updates with a FHIR patient repository and a pathology system.

Topics:

- [Use Case Overview](#)
- [Overview of the Design-Time Messages](#)
- [Overview of the Integration Receiving the Inbound HL7 Message During Runtime](#)
- [Overview of the Integration Processing the Patient Updates During Runtime](#)

A quick tour of this use case is provided. See [Synchronize EMR Patient Updates in a FHIR Patient Repository and a Pathology System](#).

Use Case Overview

This use case provides a design-time and runtime overview of two integrations that perform the following tasks.

The integrations are referred to as *handlers* and *processors*. The handler integration handles all incoming HL7 messages for a given TCP/IP connection. The processor integration processes each specific type of HL7 message.

- `handleHL7_MLLP_6200_INBOUND`: A parent integration that receives inbound HL7 messages from an EMR application. In this example, the EMR sends ADT_A08 (update patient information) messages.
- `processADT_A08`: A child integration called by `handleHL7_MLLP_6200_INBOUND` to process the inbound HL7 message and synchronize a FHIR patient repository and pathology system with the same patient updates.

Both integrations are shown as successfully completed on the Instances page under the **Observe** tab in the project.

Primary Identifier	Instance ID	Business Identifiers	Status	Duration
message-tracking-id: 0002501 processADT_A08 3.4.0	5BT6FGBDEe-Hfj...	NAME Mary Anderson FHIR ID 11419985	Succeeded	Received Thursday at 02:03 PM KST Aborted Thursday at 02:06 PM KST Duration 3.69 seconds
message-control-id: 0002501 handleHL7_MLLP_6200_INBOUND 1.0.0	4gyAAWBDEe-sv...	HL7 Msg ADT_A08 Msg Ve... 2.5	Succeeded	Received Thursday at 02:03 PM KST Succeeded Thursday at 02:03 PM KST Duration 3.89 seconds

Overview of the Design-Time Messages

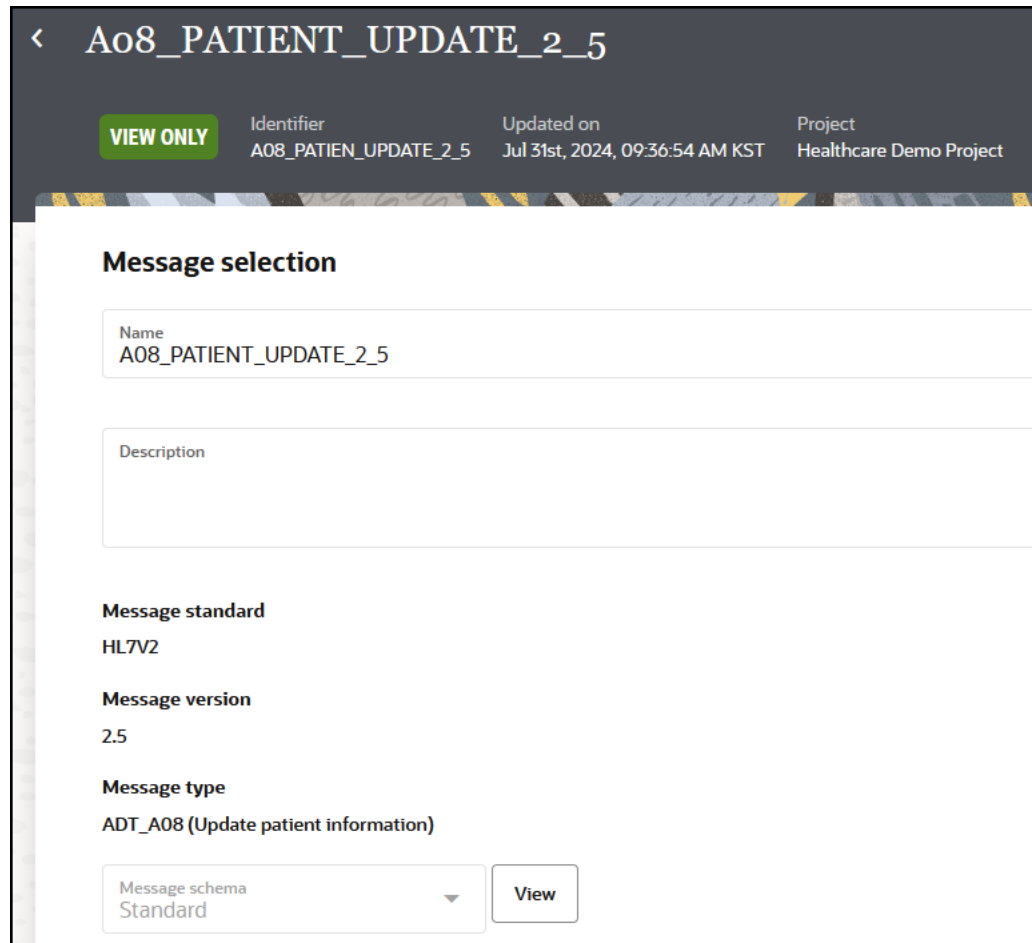
This section provides a design-time overview of the HL7 messages used in this use case. This use case is created inside a project.

1. The **Healthcare** tab for the project shows the available messages. A message is based on an associated HL7 schema. The workflow is as follows:
 - You create the schema and select the HL7 version, message version, and message type to use.
 - You create the message based on the schema.
 - The message that you create is the HL7 message definition for the message that your integration processes at runtime.
 - You select the messages to use when configuring the healthcare action during integration design.

HL7 messages	Schemas
A08_PATIENT_UPDATE_2_5 HL7V2 2.5 ADT_A08	CUSTOM_ADT_A08 HL7V2 2.5 ADT_A08
A08_PATIENT_UPDATE_2_3_1 HL7V2 2.31 ADT_A08	

As an example, the contents of the **A08 Patient Update 2_5** message are as follows:

- HL7 version 2
- Message version 2.5
- Message type ADT_A08 (Update patient information)



Overview of the Integration Receiving the Inbound HL7 Message During Runtime

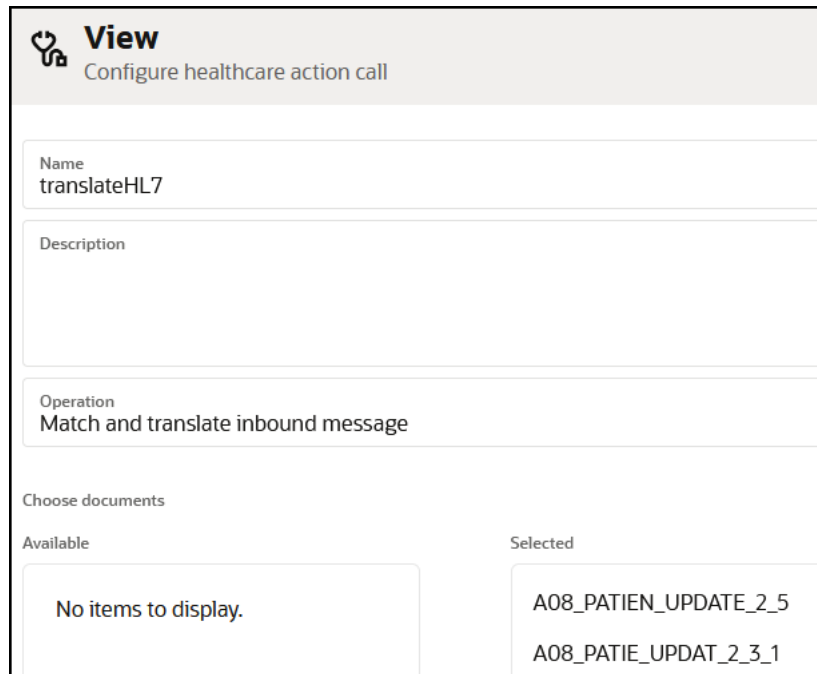
This section provides a runtime overview of key component responsibilities in the handleHL7_MLLP_6200_INBOUND parent integration that receives the initial HL7 inbound message from the EMR application. The EMR application was updated with patient details that must also be automatically updated in a FHIR patient repository and pathology system.

- [Receive and Convert the Inbound HL7 Message](#)
- [Invoke Child Integration](#)

Receive and Convert the Inbound HL7 Message

1. The MLLP Adapter trigger connection (**MLLPAdapter**) listens for messages from an EMR application that uses HL7. For this use case, the MLLP Adapter receives an HL7 Update Patient Information message (ADT_A08) from the EMR application and starts the integration.



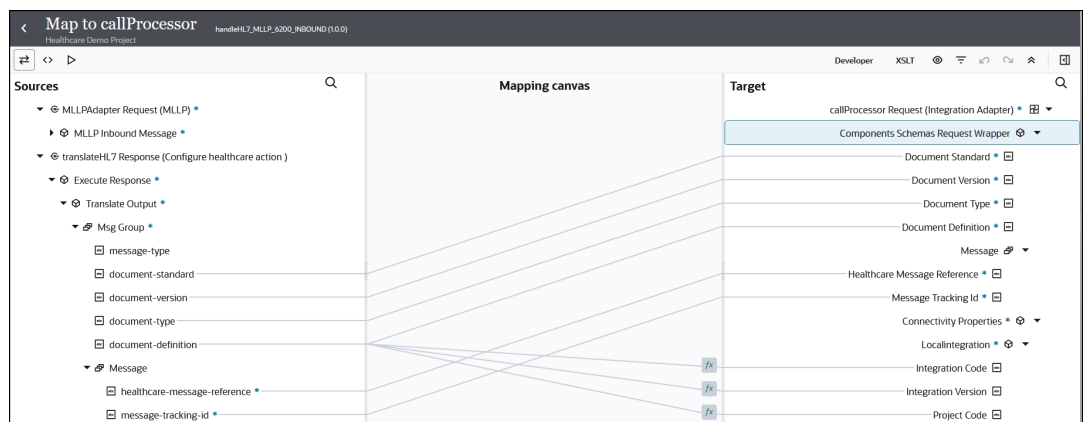


See [Create Oracle Integration for Healthcare Schemas and Documents](#).

Invoke Child Integration



1. The map action (**callProcessor**) shows the source **healthcare-message-reference** element of the **translateHL7** healthcare action. This element contains the contents of the healthcare action message in XML, but in an opaque message reference format. This output is mapped as a binary object into the integration action that invokes the **processADT_MLLP_A08** child integration.



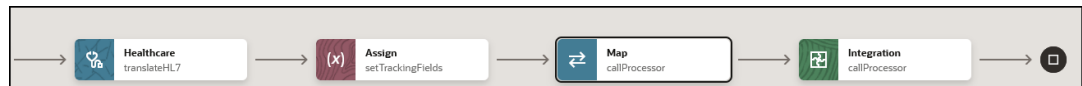
The target **Connectivity Properties** section shows the integration name, integration version, and project code being called. This information is obtained from a lookup function. For example, for **Integration Code**, if the message definition value is **A08 Update Patient**

2_5, it is routed to the **processADT_A08** child integration. The use of lookup functions offers a way to enable specific child integrations to process specific HL7 messages.

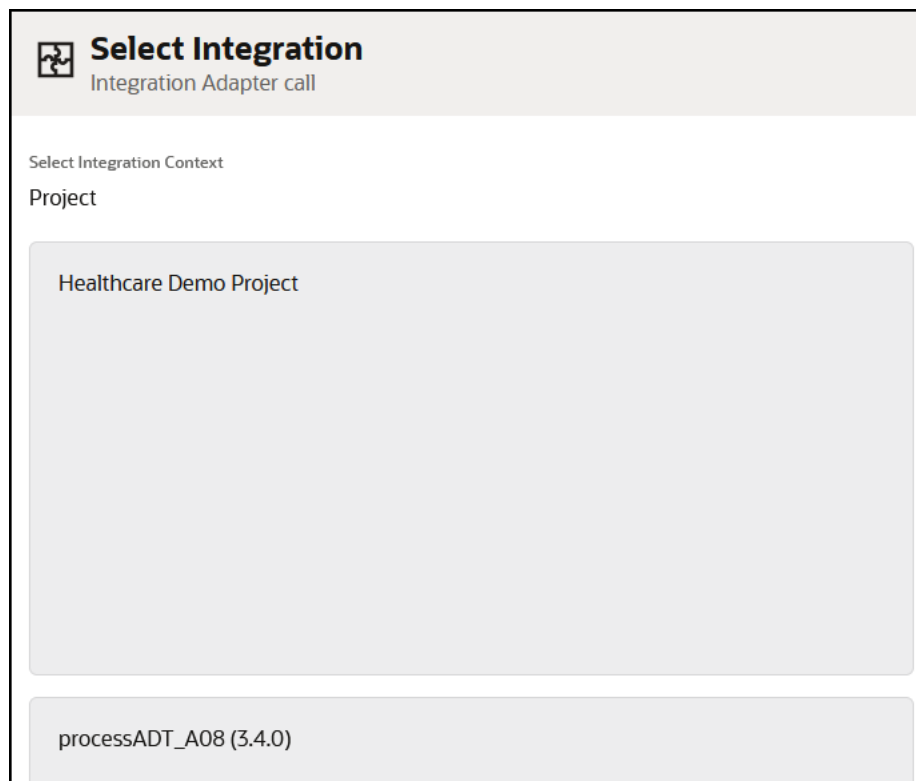
- The integration action (**callProcessor**) invokes the processADT_A08 child integration to process the ADT_A08 message.

✓ **Tip**

As a best practice, build a parent routing integration that can invoke a different child integration to process a different HL7 inbound message type (one child integration per message). Do not build a single integration with routing logic (for example, a switch or for-each action) to process each message.



- A view of the integration action configuration shows the **processADT_A08** child integration is selected to be invoked.



Overview of the Integration Processing the Patient Updates During Runtime

This section provides a runtime overview of key component responsibilities in the processADT_A08 child integration that is invoked by the handleHL7_MLLP_6200_INBOUND

parent integration to process the patient updates. The processADT_A08 child integration automatically updates the FHIR patient repository and pathology system with the patient details initially updated in the EMR application.

- [Process the Integration](#)
- [Update the FHIR Patient Repository](#)
- [Update the Pathology System](#)

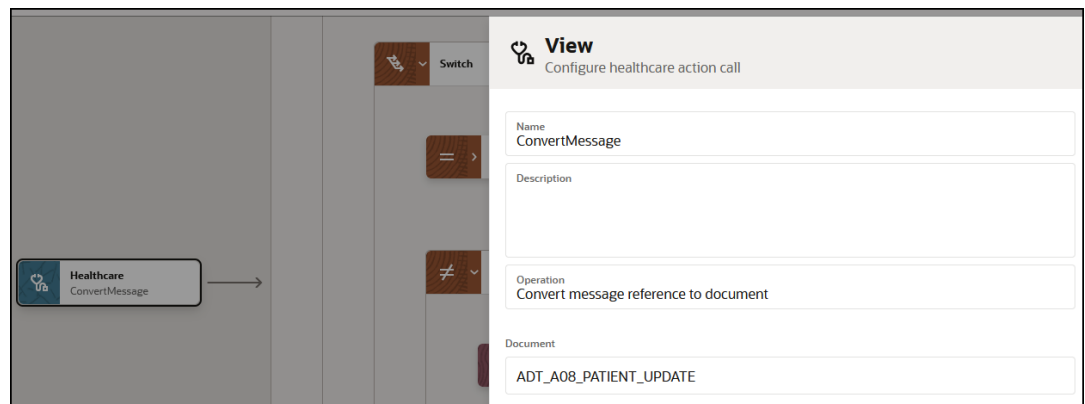
Process the Integration

1. A REST Adapter trigger connection (**ReceiveMessage**) receives the inbound message from the handleHL7_MLLP_6200_INBOUND parent integration.

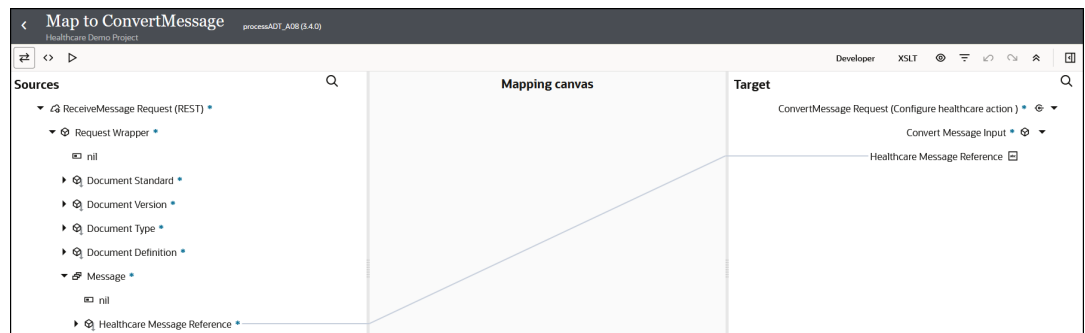


The child integration includes a healthcare action (**ConvertMessage**) after the map action of the same name.

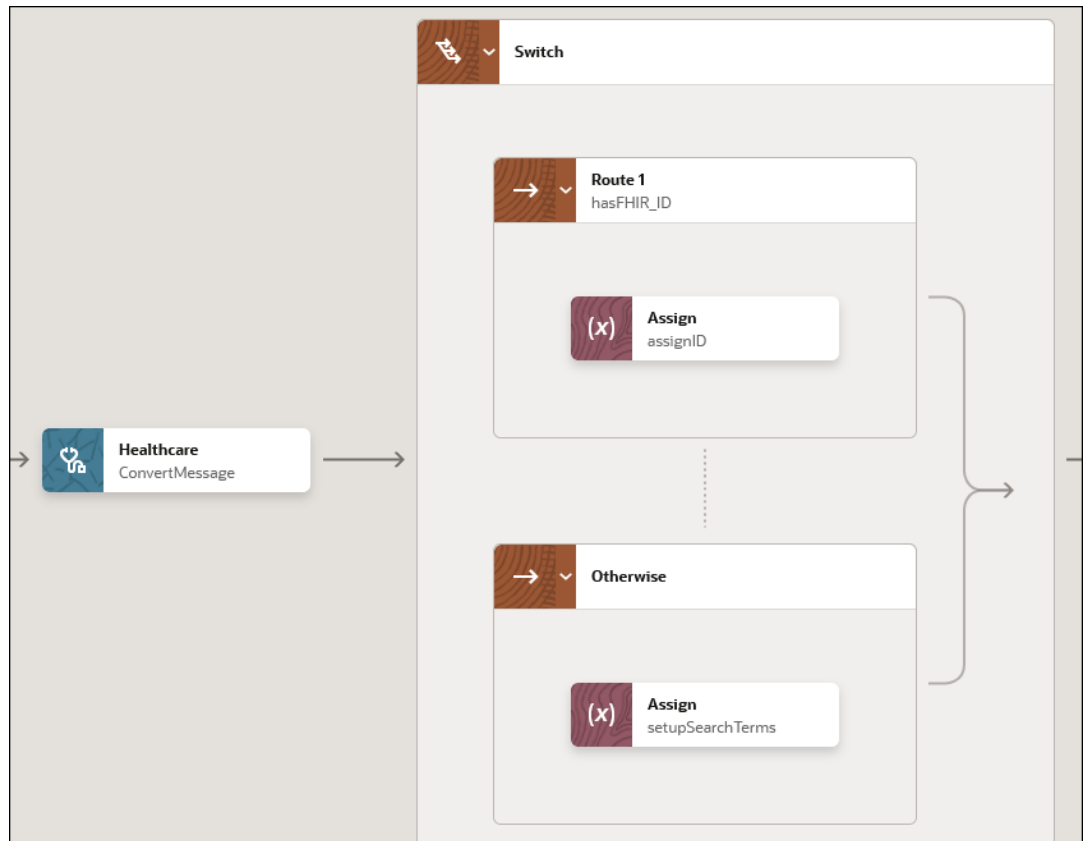
2. A view of the healthcare action configuration shows that the **Convert message reference to document** operation is selected. This operation converts the inbound HL7 document into an Oracle Integration-mappable format.



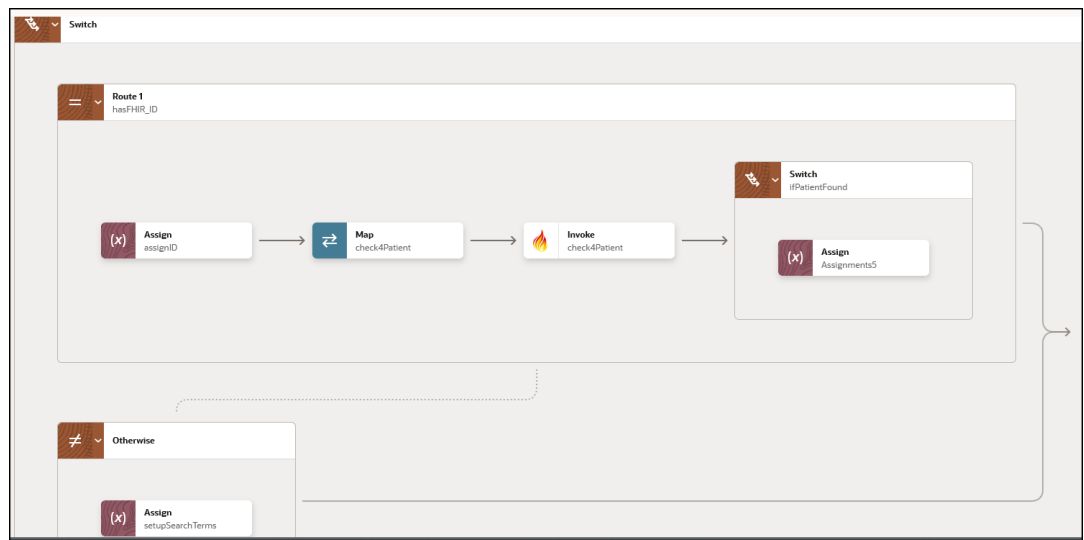
3. The map action (**ConvertMessage**) shows the mapping.



4. The assign action (**assignID**) in the **Route 1** branch of the switch action checks for a FHIR patient ID in the message. If there is an ID, it is extracted from the message.

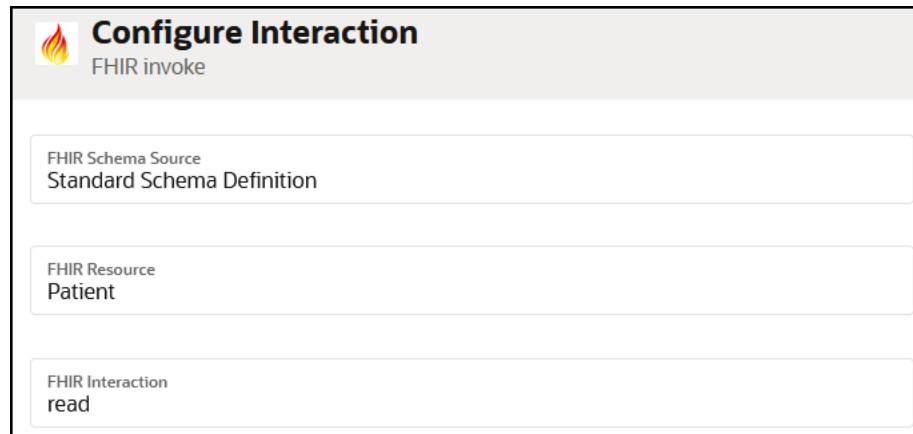


5. A check is made with the extracted ID to ensure that the patient exists in the FHIR patient repository.

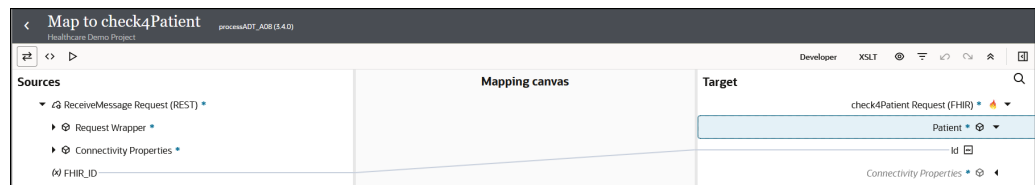


The following actions are performed.

- a. The FHIR Adapter (**check4Patient**) is invoked to check if the patient exists in the FHIR server repository.

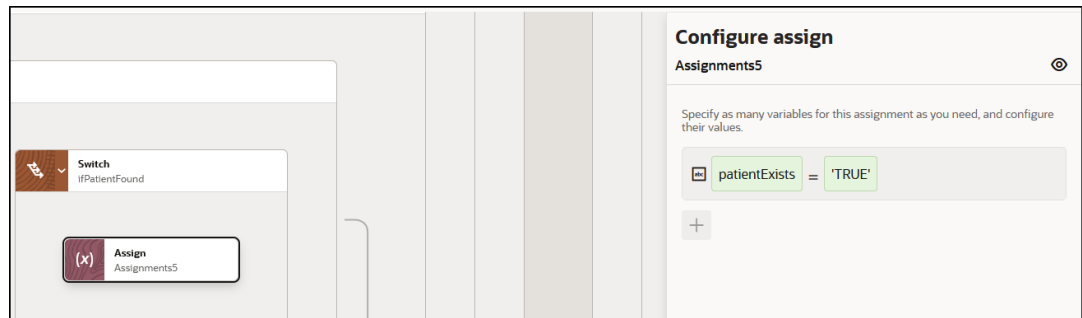


- b. The map action (**check4Patient**) passes the FHIR patient ID.



6. If the patient ID is found in the repository, a variable is assigned in the assign action (**Assignment5**).

```
patientExists = 'TRUE'
```

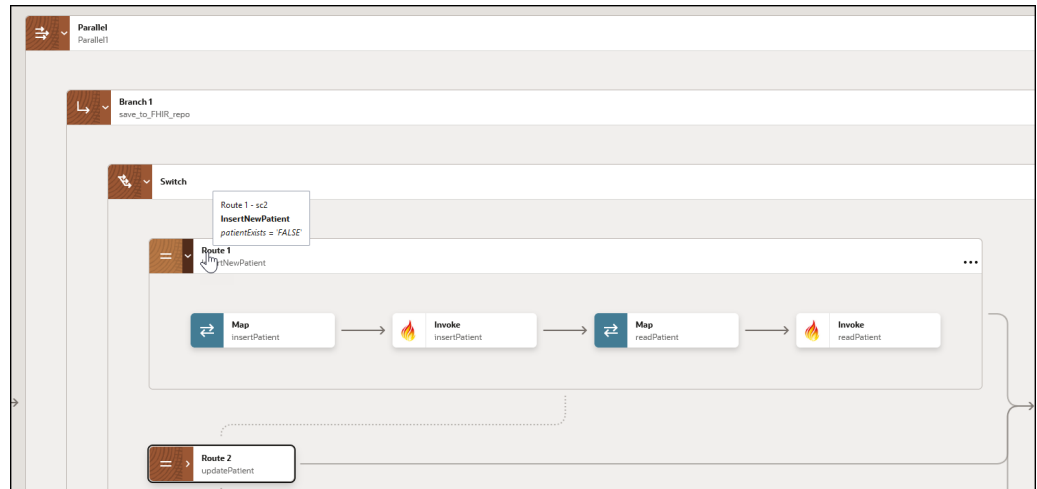


7. The FHIR ID is passed to a parallel action (**Parallel**) with two branches. A parallel action allows the path of an integration to be split into multiple branches. Each branch is processed in parallel due to their independence from each other. For this use case, a parallel branch is provided to update both external applications:
- FHIR patient repository (branch 1)
 - Pathology system (branch 2)

Update the FHIR Patient Repository

1. The first branch in the parallel action includes a switch action with two routes.
 - The first branch in the switch action identifies whether to add a new patient in the FHIR patient repository. The popup message on the first branch indicates that if the

assignment of **patientExists** equals **false**, insert the patient into the FHIR patient repository.



The following actions are performed.

- a. The FHIR Adapter (**insertPatient**) inserts the patient into the FHIR server repository.

Configure Interaction

FHIR invoke

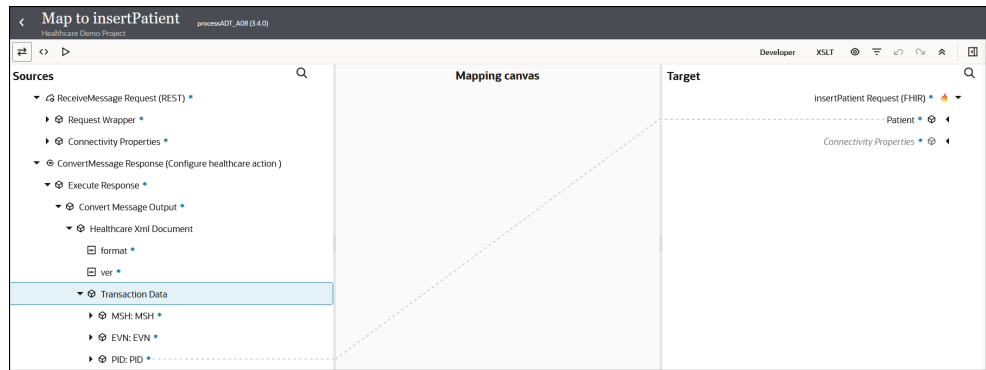
FHIR Schema Source
Standard Schema Definition

FHIR Resource
Patient

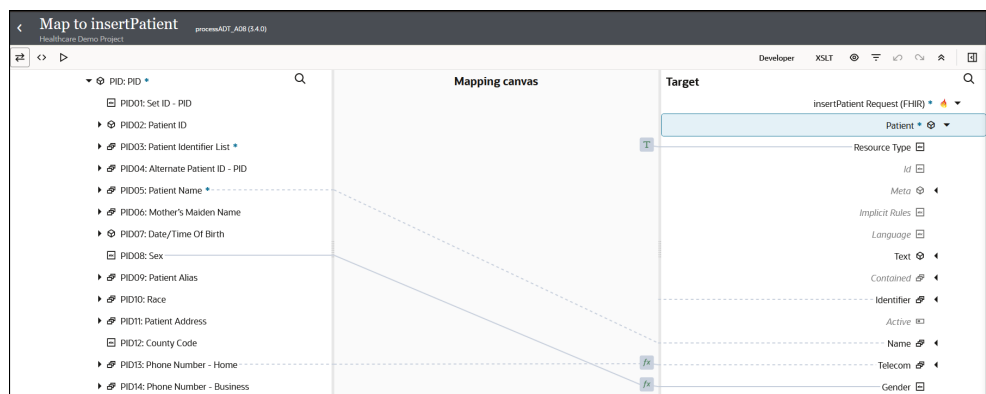
FHIR Interaction
create

Configure Response
representation

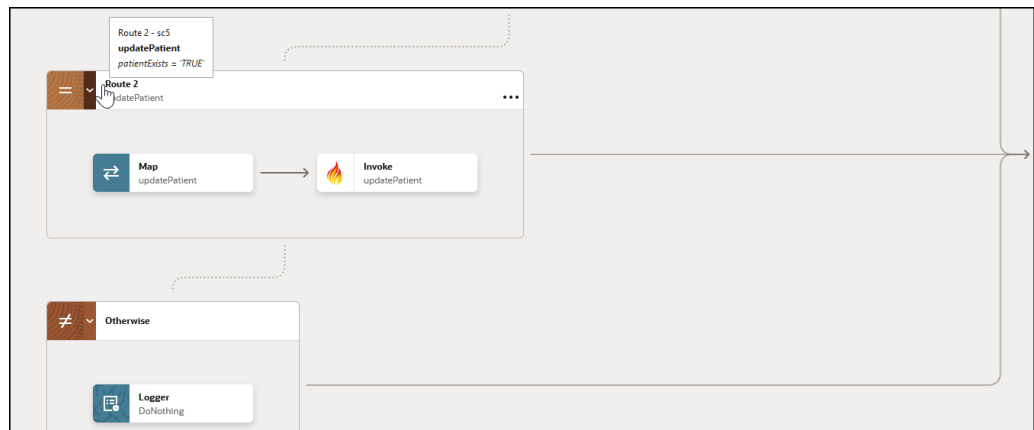
- b. The map action (**insertPatient**) shows the source HL7 message payload under **Transaction Data** that is mapped to the target **Patient**.



- c. The expanded **PID:PID** element shows the specific patient mappings, such as patient name, gender, and home phone number.

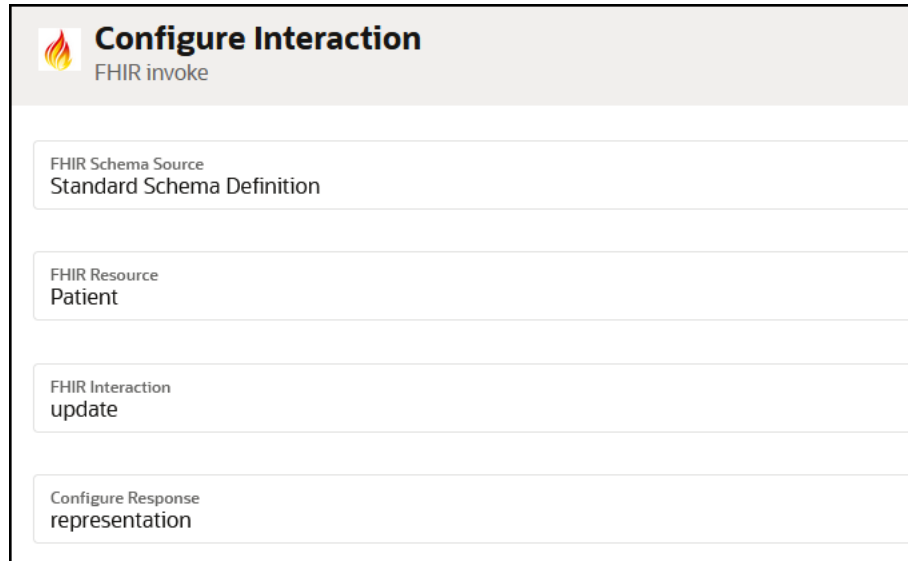


- The second branch in the switch action identifies whether to update an existing patient. The popup message on the first branch indicates that if the assignment of **patientExists** equals **true**, update the patient in the FHIR patient repository.

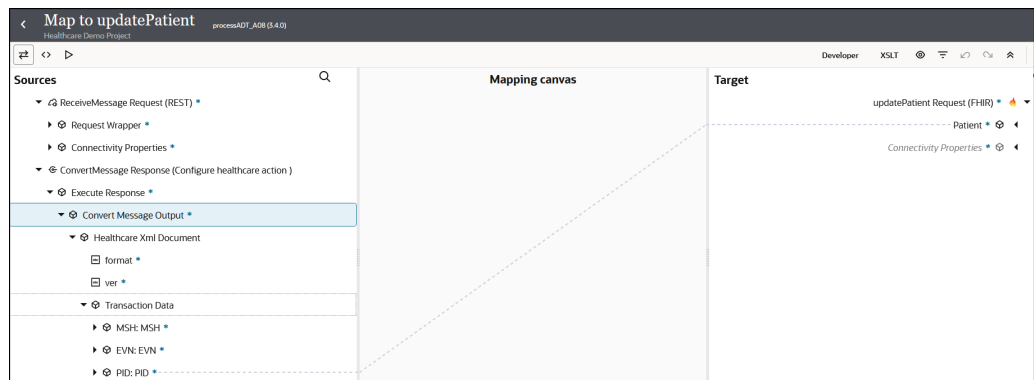


The following actions are performed.

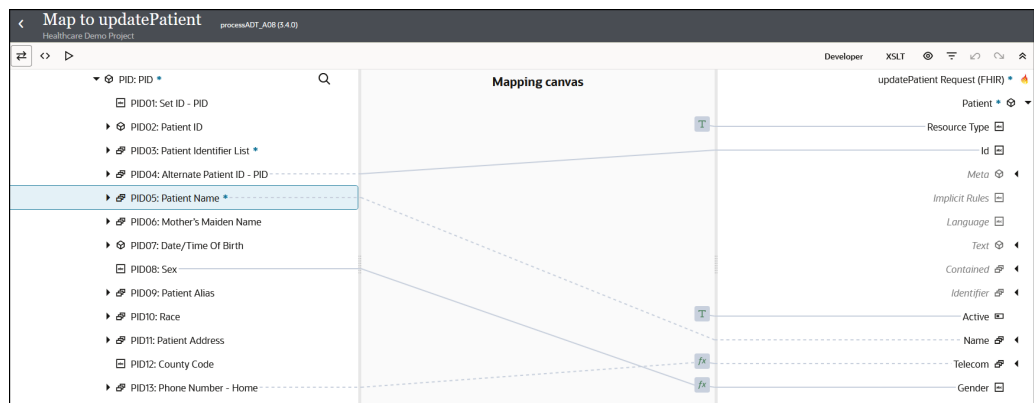
- a. The FHIR Adapter (**updatePatient**) updates the patient in the FHIR server repository.



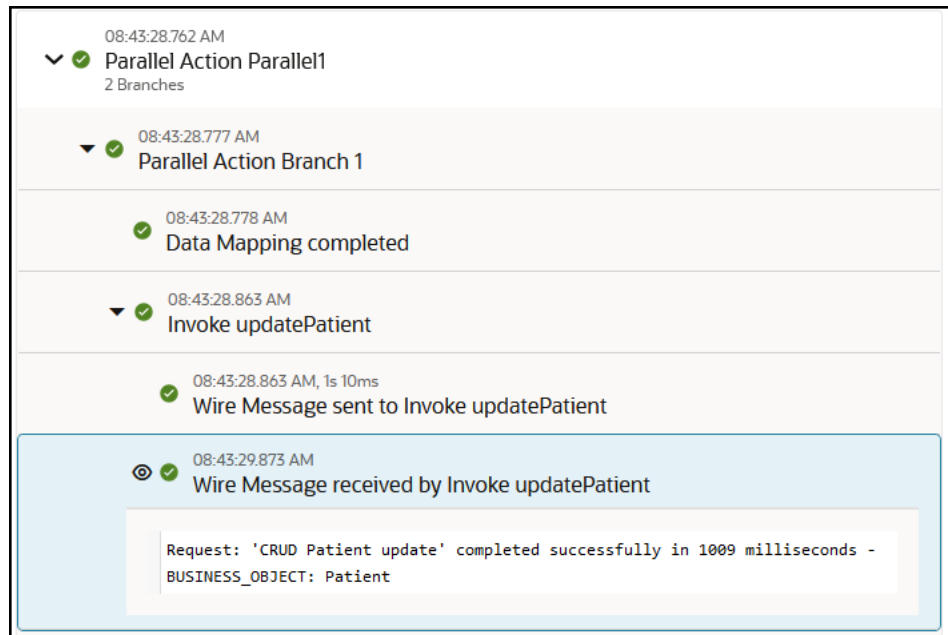
- b. The map action (**updatePatient**) shows the source HL7 message payload under **Transaction Data** that is mapped to the target **Patient**.



- c. The expanded **PID:PID** element shows the specific patient mappings, such as patient ID, name, gender, and home phone number.

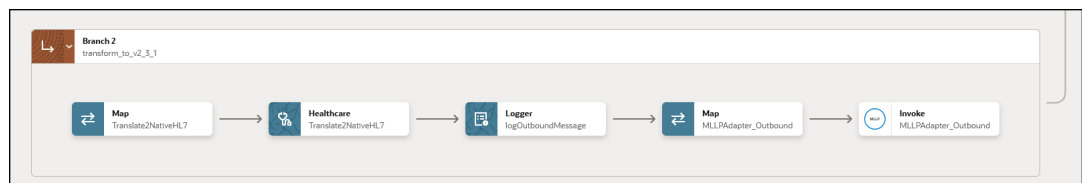


- 2. The expanded activity stream shows that the patient already existed and was updated in the FHIR server repository.

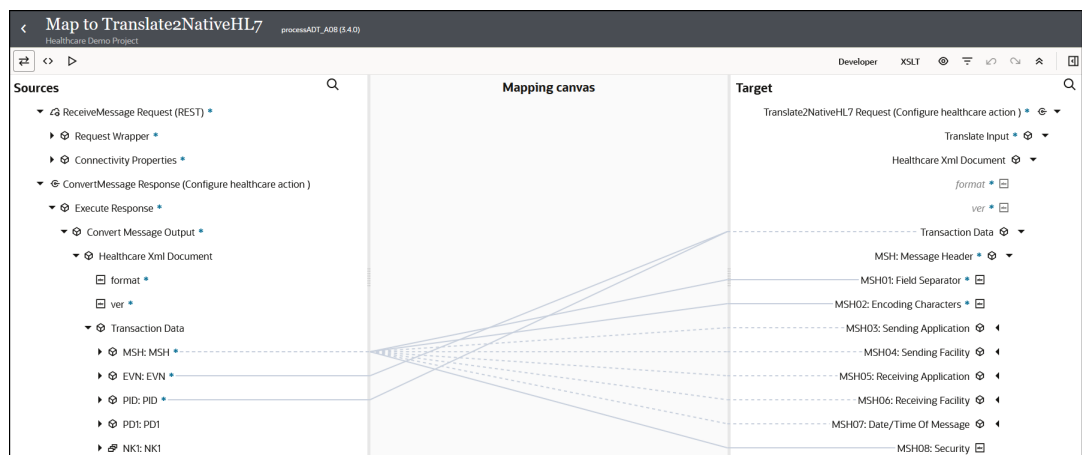


Update the Pathology System

- The second branch in the parallel action updates the pathology system. Because the EMR application uses HL7 version 2.5 and the pathology system uses HL7 version 2.3.1, a message conversion to the earlier version must be performed.



- The expanded mapper (**Translate2NativeHL7**) shows the HL7 version 2.5 source and HL7 version 2.3.1 target element mappings performed by the user at design time (patient ID, patient name, date of birth, and more).



Before sending the document to the pathology system, it must be converted from Oracle Integration XML format back to HL7 format in a healthcare action.

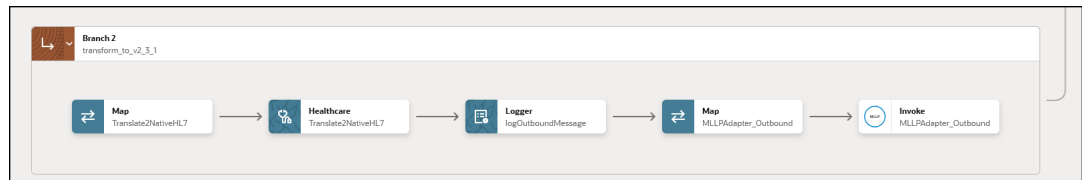
- A view of the healthcare action (**Translate2NativeHL7**) configuration shows that the **Translate to outbound message** operation is selected. This operation converts the outbound XML-formatted payload message to the HL7 version 2.3.1-formatted payload message selected in the **Choose documents** field. The healthcare action produces a healthcare message reference to pass directly to the MLLP Adapter invoke connection.

View
Configure healthcare action call

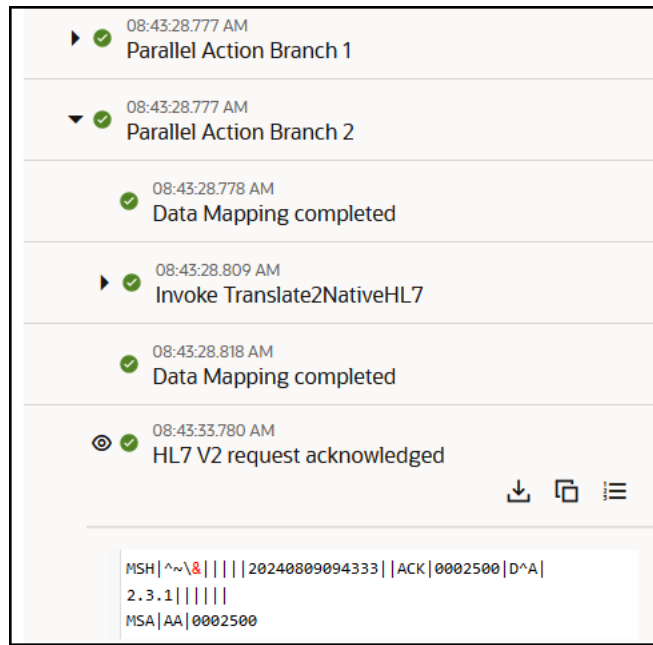
Name	Translate2NativeHL7
Description	
Operation	Translate to outbound message
Document	A08_PATIENT_UPDATE_2_3_1

Selected Document Definition has Standard 'HL7V2', Version '2.3.1', Type 'ADT_A08 (Update patient information)'.

- An MLLP Adapter invoke connection (**MLLPAdapter_Outbound**) sends the HL7 version 2.3.1 message to the pathology system for updating.



- The logger action (**logOutboundMessage**) in the expanded activity stream shows that the HL7 2.3.1 message was sent.



The screenshot displays a log window with the following entries:

- 08:43:28.777 AM Parallel Action Branch 1
- 08:43:28.777 AM Parallel Action Branch 2
- 08:43:28.778 AM Data Mapping completed
- 08:43:28.809 AM Invoke Translate2NativeHL7
- 08:43:28.818 AM Data Mapping completed
- 08:43:33.780 AM HL7 V2 request acknowledged

Below the log entries, there are three icons: a download arrow, a copy icon, and a list icon. A text box contains the following HL7 V2 message:

```
MSH|^~\&|||||20240809094333||ACK|0002500|D^A|
2.3.1|||||
MSA|AA|0002500
```

The integration is now complete. The EMR application patient detail updates have been automatically synchronized in the FHIR patient repository and pathology system.