2020-09 DaVinci Notifications

Track overview

Short Description

FHIR version of care team notifications around a patient's admission or discharge to higher level care settings in support of ONC Final rule.

Ready to go
What did you change? Notify watchers

Provide feedback

Long Description

Current ADT exchange approaches typically use an HL7 V2 ADT message. This is a legacy EDI-style technology that uses HL7 specific protocols over ports that are not typically exposed to the internet. While HL7 V2 works well within the confines of a hospital system's intranet, it is not particularly well suited to cross-enterprise data exchange.

FHIR resources can be use to transport patient admission and discharge information to improve care coordination, in the form of ADT notifications. An ADT notification from the provider to the health plan can help track where care has been delivered and help ensure timely follow-up care is delivered as needed since healthcare stakeholders are increasingly responsible for knowing what care their patients have received and what care they need, regardless of where their patients sought care. FHIR-enabled provider systems can support functionality to send admission, discharge, and transfer information to a system with a FHIR API to receive the notification via an Encounter resource and its associated resources. The information can be sent to a pre-determined end-point or, optionally, an end-point or Direct Address that is determined by looking up the information recipients.

This is the first iteration of this IG and intent here is to have a minimally viable product to demonstrate utility for notifying care team members of important events in a patient's health.

This Guide defines a FHIR technical framework for sending unsolicited notifications to the appropriate actors when triggered by an event or request. The notification should provide enough information to understand what the notification is about and to enable the recipient to determine if and what additional steps they need to take in response to the notification, While a notification may generate workflow on the recipient's part, the notification is not a part of that workflow. As such, the sender of a notification should not expect any additional response outside of the standard FHIR functionality.

The notifications follow a simple FHIR messaging paradigm for notifications employing both the FHIR Messaging Bundle and RESTful interaction using the *$process-message* operation for exchanging data. **Note that Da Vinci Notifications allow implementers to use only these FHIR messaging components in the same manner as a FHIR RESTful operation without having to fully implement the FHIR messaging framework. But it is also compatible with FHIR messaging as implemented.** The Framework documents in detail how to:

1. Assemble the necessary resources for a particular use case into a FHIR Messaging Bundle
2. Send the notification to the receiver using the FHIR defined $process-message operation

The Admit/Discharge/Transfer Use case is also documented in the IG to comply with part 485 of the CMS 21st Century Cures Act final rule. This use case demonstrates how the Da Vinci Notifications IG framework is used to define the Da Vinci Notification Bundle for admissions, discharges, and transfers and how to send a notification between a Sender and a Recipient/Intermediary and will be the main focus of this connectathon.

Type

- Test an Implementation Guide

Submitting Work Group/Project/Accelerator/Affiliate/Implementer Group

Proposed Track Lead

Riki Merrick, Vernetzt, LLC / APHL (rikimerrick@gmail.com)

Related tracks

2020-09 subscriptions Track

FHIR Version

- FHIR R4

Specification(s) this track uses

Da Vinci Unsolicited Notifications Implementation Guide - THIS IS THE LINK TO THE CI BUILD WITH APPLIED TRACKERS

Artifacts of focus

Base Da Vinci Notification Profiles

- Da Vinci Notifications MessageHeader Profile
- Da Vinci Notifications Bundle Profile

Da Vinci Admission/Discharge Notification Profiles

- Da Vinci Admit Notification MessageHeader Profile
- Da Vinci Discharge Notification MessageHeader Profile
- Da Vinci Transfer Notification MessageHeader Profile
- Da Vinci Admit/Transfer/Discharge Notification Condition Profile
- Da Vinci Admit/Transfer/Discharge Notification Coverage Profile
- Da Vinci Admit/Transfer/Discharge Notification Encounter Profile

Clinical input requested (if any)

For each of the use case, the data elements that a care team is interested in knowing for meeting the notification purpose are needed. One of the goals of this connectathon is to determine if the admit and discharge, transfer data elements meets the needs of the majority of different care team members. Some may only be applicable to particular care team members, for example, primary care provider vs payor vs physical therapist vs "meal on wheels" support.

Patient input requested (if any)

Expected participant

Signup Sheet

Reference Implementations

HealthLX Notifications RI:

Login to HSPCs Logica Sandbox at: https://sandbox.logicahealth.org/DaVinciAlertsEHR/apps

App: FHIR Alert sender
Receiver URL: https://davinci-alerts-receiver.logicahealth.org

quick video demo

Contact Viet Nguyen (vietnguyen@stratametrics.com) for login credentials

Da Vinci Notification Sender Simulation: (Client Facade to simulate Sender) at: http://ehaas.pythonanywhere.com/

Zulip stream

https://chat.fhir.org/#narrow/stream/205917-Da-Vinci.20Alerts

for the connectathon: https://chat.fhir.org/#narrow/stream/179207-connectathon-mgmt/topic/DaVinci.20Notifications.20track.20participants

Track Orientation

(Copy of Orientation Slide)

The recording of Track Orientation session given on August 19, 2020

Kick-off Thursday 11AM - 12 PM EDT

Check in Thursday 3 PM EDT on the Whova Notifications channel

Check in Friday 11 AM - 12 PM EDT on the Whova Notifications channel

Track details

System Roles

The following actors have been defined:

- **Sender** - the system responsible for sending the notification, typically operated by the facility or organization where the event occurred
- **Recipient** – the system responsible for receiving generated notifications from Notification Senders
- **Intermediary** (e.g. ClearingHouse or HIE/HIN)– a system that can act as a central point to receive notifications from multiple Notification Senders and distribute/forward notifications to Notification Recipients based on previously defined policies

They correspond to these roles in the notification transactions:

Role	**server-mode**	**client-mode**
Sender	Da Vinci Notification Query Responder	Da Vinci Notification Sender
Intermediary	Da Vinci Notification Receiver,Da Vinci Notification Query Responder	Da Vinci Notification QueryRequester, Da Vinci Notification Forwarder
Recipient	Da Vinci Notification Receiver	Da Vinci Notification Query Requester

Preconditions

Please review the **Da Vinci Notifications Implementation Guide** and Capability Statements for the supported profiles, operation as well as the extensive guidance surrounding the transaction workflow.

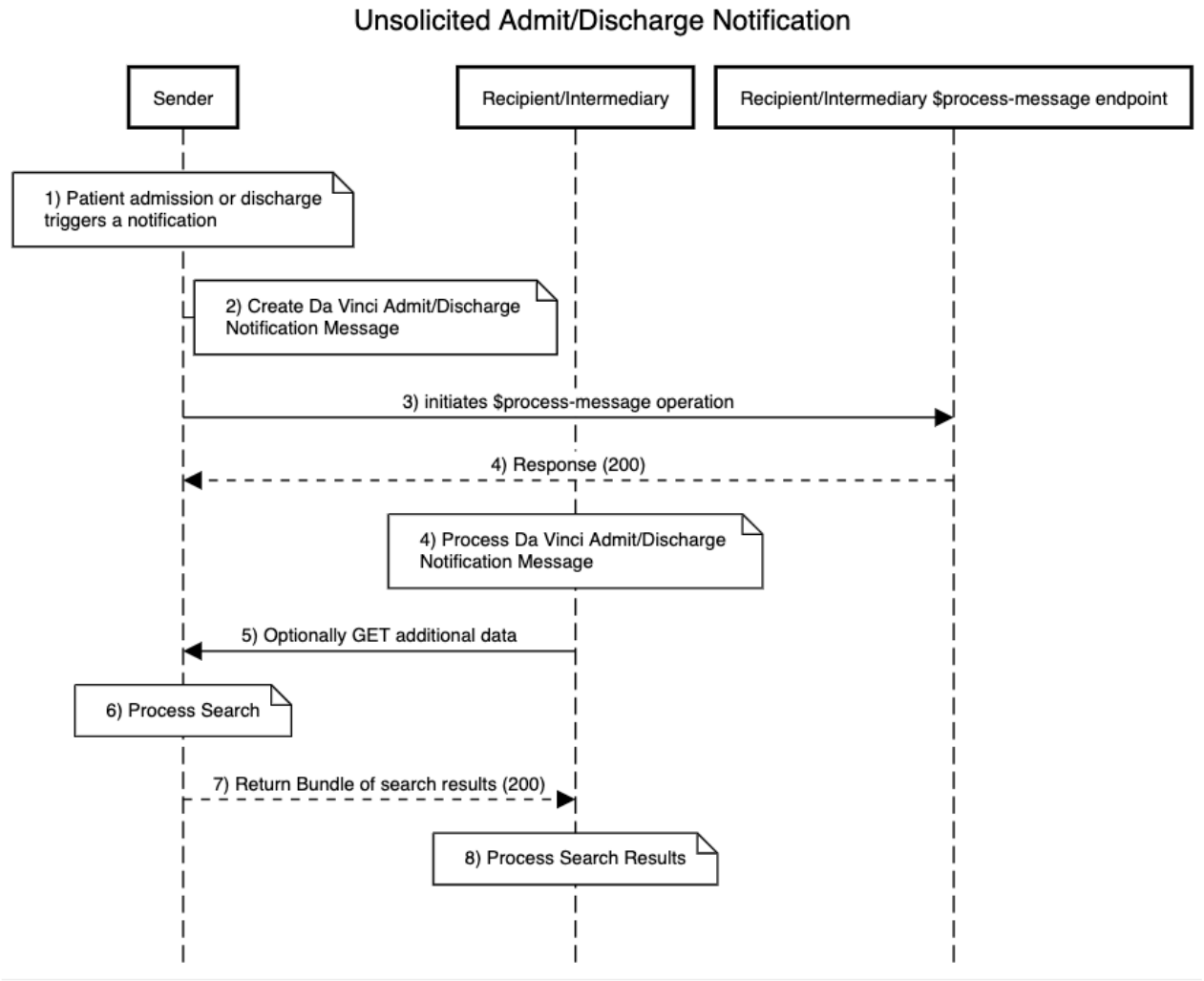
- Receiver and Intermediary must be able to support the synchronous $process-message operation on their system endpoint
  - 200,202,204 +/- OperationOutcome all ok for successful transactions
    - **PREFERRED 200 OK**: Indicates that the message **has been fully processed**. If an application-level response is expected for the submitted message, that response SHALL be returned as the body of the 200 response.
    - **202 Accepted**: Indicates that the receiving system **has accepted custody of the message**
    - **204 No Content**: Indicates that the message has been fully processed and would normally have had an application-level response, but because of instructions from the sender (e.g. the messageheader-response-request extension), no response is being provided - NOTE NO response is expected for Da Vinci Notificiations
  - 401,404 +/- OperationOutcome
  - 429 +/- OperationOutcome
  - 500+ +/- OperationOutcome
  - including these http responses:

- Although not required an OperationOutcome Message is preferred for ease of debugging.
- Optionally the notification contents may be echoed back to the Sender.

Please review the DaVinci Notification FHIR IG for preconditions and assumptions.

General workflow :

Figure 9



Scenarios

Admit and Discharge Use Case Scenarios

(Examples using this approach are available in the Implementation Guide and the Reference Implementation (see below). Test data generated from is also provided for each scenario in the links provided.

1 Emergency and Inpatient Admissions

Storyboard: Patient Y has been admitted to Provider X 's facility. Acting in the role of Notification Sender Provider X notifies Payer Z who is acting in the role Notification Recipient of the admission event.

event code = notification-admit

Encounter.class code = EMER or IMP

(Encounter.type may not be known when the notification is sent. If data is not available use the appropriate "unknown" concept code from the value set SNOMED SCTID: 261665006 | Unknown (qualifier value))

Encounter status = "in-progress"

2 Discharges/Visit ends

Storyboard: Patient Y has been discharged from Provider X 's facility. Acting in the role of Notification Sender Provider X notifies Payer Z who is acting in the role Notification Recipient of the discharge event.

event code = notification-discharge

encounter.hospitalization.dischargeDisposition = any from http://terminology.hl7.org/CodeSystem/discharge-disposition

3 Transfer from the ED to inpatient ICU

Storyboard: Patient Y has been transfered from Provider X 's ED to their inpatient ICU. Acting in the role of Notification Sender Provider X notifies Payer Z who is acting in the role Notification Recipient of the transfer event.

event code = notification-transfer

encounter.hospitalization.dischargeDisposition = any from http://terminology.hl7.org/CodeSystem/discharge-disposition

4 Other types of admissions:

4a Admission for Observation

Storyboard: Patient Q has been admitted to Provider R 's facility for Observation. Acting in the role of Notification Sender Provider R notifies ClearingHouse C who is acting in the role Notification Recipient of the admission event.

event code = notification-admit

encounter code = OBSENC

(Encounter.type may not be known when the notification is sent. If data is not available use the appropriate "unknown" concept code from the value set SNOMED SCTID: 261665006 | Unknown (qualifier value))

4b Admission for special services, such as outpatient surgery

Storyboard: Patient M has been admitted to Provider N 's facility for outpatient surgery to repair a broken collarbone. Acting in the role of Notification Sender Provider N notifies HIE I who is acting in the role Notification Recipient of the outpatient surgery.

event code = notification-admit

encounter code = AMB or SS

(Encounter.type may not be known when the notification is sent. If data is not available use the appropriate "unknown" concept code from the value set SNOMED SCTID: 261665006 | Unknown (qualifier value))

4c Encounter/Visit Notification for ambulatory services

Storyboard: Patient K has arrives at to Provider A 's facility for an office visit. Acting in the role of Notification Sender Provider A notifies ClearingHouse C who is acting in the role Notification Recipient of the admission.

event code = notification-admit

encounter code = AMB

(Encounter.type may not be known when the notification is sent. If data is not available use the appropriate "unknown" concept code from the value set SNOMED SCTID: 261665006 | Unknown (qualifier value))

Scenario Step 1 Assemble Notification Bundle

Description:

An event such as an inpatient admission or discharge triggers the Da Vinci Notifications Technical Workflow (step 1 above). Note that the actual trigger details is out of scope for the Notifications Guide. See the Subscription's track for more information on the triggering workflows.

***Action*:**

Based on the type of event, the Notification Sender assembles the Notification Message Bundle and all the included FHIR resources. (step 2 in the workflow diagram)

***Success Criteria*:**

For Notification Sender

1. Creation of a *conformant* Message Bundle containing the following resources and references. Refer to the Implementation Guide for graphical, tabular and formal definitions of the Bundle contents.

Link	Source Profile	Path	Target Profile	Min	Max	Must Support
—	—	—	—	—	—	—
1	Da Vinci Notification MessageHeader Profile	MessageHeader.focus	US Core Encounter Profile	1	1	true
2	Da Vinci Notification Encounter Profile	Encounter.location	US Core Location Profile	1	*	true
3	Da Vinci Notification Encounter Profile	Encounter.participant.individual	US Core Practitioner Profile	0	*	true
4	Da Vinci Notification Encounter Profile	Encounter.subject	US Core Patient Profile	1	1	true
5	Da Vinci Notification Coverage Profile	Coverage.beneficary	US Core Patient Profile	0	1	true
6	Da Vinci Notification Condition Profile	Condition.encounter	US Core Encounter Profile	0	*	true
7	Da Vinci Notification MessageHeader Profile	MessageHeader.sender	US Core Practitioner Profile|US Core PractitionerRole Profile|US Core Organization Profile	0	1	true
8	Da Vinci Notification MessageHeader Profile	MessageHeader.responsible	US Core Practitioner Profile|US Core PractitionerRole Profile|US Core Organization Profile	0	1	true
9	Da Vinci Notification MessageHeader Profile	MessageHeader.author	US Core Practitioner Profile|US Core PractitionerRole Profile	0	1	true

Scenario Step 2 Notification Sender Initiates FHIR Operation

Description:

Notification Sender initiates the $process-message operation to the notification recipient/Intermediary's $process-message endpoint (step 3 in worflow diagram)

Action:

Notification Sender POSTs Da Vinci Notification Message payload body to the known notification recipient/Intermediary's $process-message endpoint

Success Criteria:

For Notification Sender

1. POSTS to correct [base]/$process=message endpoint

Scenario Step 3 Notification Recipient/Intermediary Response to FHIR Operation

Description:

Notification Recipient/Intermediary responds to transaction with http response (step 4 above)

Action:

Upon successful delivery of Da Vinci Notification Message payload body to the recipient/Intermediary's $process-message endpoint, it returns a response code of 200.

- 200 PREFERRED ,202,204 +/- OperationOutcome all ok for successful transaction

Success Criteria:

Notification Recipient/Intermediary

Note: Since these actors are considered a 'Black box' in the context of processing an operation there is no prescribed behavior. However for the purpose of the Connectathon, the Notification Recipient/Intermediary should display the contents of the Notification FHIR message bundle to demonstrate the transaction was successful.

1. Notification Recipient/Intermediary responds to transaction with http response
2. Notification Recipient/Intermediary display the contents of the Notification FHIR message bundle to demonstrate the transaction was successful.

Bonus 1: Error Conditions:

Technical errors are typically handled by lower level protocols or manual processes. Typically the Sender would simply resubmit the Notification after correcting the issue.

The Guide only covers the "happy path". In addition to the basic guidance in FHIR specification for $process-message operation:

- 400,401,404+/- OperationOutcome no point in retry
- 429 +/- OperationOutcome retry but slow down traffic
- 500+ +/- OperationOutcome - server issues may retry a few times

Other errors may need to be communicated back to the Notification Sender and SHOULD be transmitted in the OperationOutcome

400 Error

- Sender/Intermediary: Create an invalid message
- Receiver: return a 400 error +OperationOutcome detailing the error

401 Error (see Authorization and Authentication below**)**

- Sender/Intermediary: Send message without proper authorization
- Receiver: return a 401 error -/+ OperationOutcome

404/429/500 Error out of scope for this connectathon

Bonus 2: Authorization and Authentication:

Following the FHIR Bulk Data IG's SMART Backend Services:Authorization Guide authorization and authentication details to access the Intermediary /Recipient FHIR Endpoint for POSTING the Notifications.

Consider using the “custom" system scope: system/process-message.write

Also using dynamic registration as described in Da Vinci HREX : http://build.fhir.org/ig/HL7/davinci-ehrx/smart-app-reg.html

Bonus 3: Intermediary Forwarding Messages (Acting as Sender with Provenance)

Following the updated guidance in the *Framework* section of the guide, forward a message bundle to a Recipient/Intermediary.

- Create a new message bundle with a new `Bundle.id` and new `MessageHeader.id`
- Update the `MessageHeader.sender` to reflect the Intermediary as the new Sender
- Replace the resource in the Bundle with the resource referenced by the updated `MessageHeader.sender` element.
- Update the `MessageHeader.destination` to reflect the new Recipient/Intermediary.
- Add the appropriate US Core Provenance Resource to the message Bundle as outlined in the guide.

(this is demonstrated in the Da Vinci Notification Sender Simulation)

Bonus 4: Follow up Queries for more information:

Notification Recipient/IntermediaryQueries Notification Sender to fetch more information. (e.g. medications). For additional guidance refer to the US Core IG

How do they identify the proper endpoint and launch the proper oauth 2.0 credentials to get access to the patient data. (see the **SMART Application Launch Framework Implementation Guide Release 1.0.0)**

- Registering a SMART App with an EHR
- FHIR endpoint
- auth endpoint
- scopes

Notification Sender.

1. Returns an appropriate http response with the appropriate status code
2. returns conformant resources in the HTTP response as a searchset bundle

Test Data:

"Examplitis": 3 admits/3 transfers/3 discharges

These examples follow the lifecycle of a three patient encounters and are the source for the Da Vinci Notification Sender

format: FHIR Transaction Bundle

- Admit Bundle (PUT resources to FHIR Server)
- Transfer Bundle (Updates Encounter and adds Condition)
- Discharge Bundle (Updates Encounter)

format: CSV

admit-notify-2

"Examplitis": 100 admits/100 transfers/100 discharges

- Admit Bundle (PUT resources to FHIR Server)
- Transfer Bundle (Updates Encounter and adds Condition)
- Discharge Bundle (Updates Encounter)

format: CSV

admit-notify-100

TestScript(s)

Touchstone Tests for Connectathon 25 use cases are found at: Touchstone_DaVinciNotification_Scripts

Please create a Touchstone user account (free) associated to the DaVinci organization if you have not already in order to run your systems through the test scripts. Please feel free to contact Touchstone_Support@aegis.net for any questions about testing using Touchstone.

2020-01 Da Vinci Notification (aka Alerts)

Security and Privacy Considerations

Report out Summary:

*1 paragraph summary: what was the track trying to achieve: Test senders or receiver capabilities for notification message bundles for the admit/transfer /discharge use case

*1 list of participants (with logos if you have time and energy) Tell Health Inc. (Ivan Daykov) - receiver; needed to use **Da Vinci Notification Sender Simulation:** (Client Facade to simulate Sender) at: http://ehaas.pythonanywhere.com/ for testing

*systems which have implemented the IG, Profile, or Resource, and approximate percentage covered: Tell Health Inc. (Ivan Daykov):

- We tested the receive part and are passing 100% of the touchstone tests on that.
- We're not passing the touchstone tests for sending/forwarding messages because we haven't implemented those (we may not have a need to do them for these for our use scenario).
- I'm not sure whether $process-message, which is what we implement/test, is a Resource or a Profile. I believe it is termed an 'Operation' and may be Da Vinci Notifications is a profile on $process-message because it specifies all the details of the Bundle.
- I guess we have the receiving part of it but not the sending part, so depending on the 'counting profile' it's either 100% of what we need for our use case, 50% of the 'specification subdivision' (send and receive), or 33% of the details (1/3 to assemble a bundle, 1/3 to send it, 1/3 to receive it)

*1 paragraph: notable achievements Tell Health Inc. was able to receive the simulated notificaiton bundles; worked through several of the test scripts and was able to pass 8/12 attempted; will be running some more after we figured out how to set up additional test scrpts

*1-2 screenshots if relevant and interesting and/or links to further information about implementations/achievements

*1 paragraph: discovered issues / questions (if there are any) was helpful to experiment with the technology

*1 paragraph: now what? will provide feedback, if more comes up during the follow up testing