

[Home](#)

Exported Sep 09, 2020 at 12:58 PM

HL7 FHIR eReferral Workstream



Messages

Campfire

To-dos

Docs & Files

Check-ins

Schedule

Forwards

Added by Tim Berezny • Updated Jun 11, 2018 at 10:03 PM

/Task

60% Pilot Ready

Links:

- [/Task - Canadian eReferral Profile](#)
- [/Task - FHIR 3.2 Standard](#)

Relevant chat.fhir.org Threads

- [Closed/Complete for ServiceRequest \(or task\)](#)
- [When should a resource's _lastUpdated be set?](#) (related also to task resource)

ValueSets:

- status - [TaskStatus valueset](#)
- intent - [RequestIntent valueset](#): Always set = "proposal"
- code - Always = "Process Request", until more tasks are added to the process.

Notes:

Task State Machine:

eReferral Supported Tasks

The eReferral process supports the following task codes definition to support specific workflows.

"Process Request": Track the status of an eReferral status over its lifecycle. This task is created by the eReferral server on submission, and closed when the referral processing is complete. The task.status updates in parallel with the similar (but more broadly defined) ServiceRequest.status. In addition, Task.BusinessStatus can also be used to communicate custom deployment specific statuses (e.g., waitlisted, waiting for call-back, pending assessment completion, etc...)

"Request Communication Response" (a.k.a., "Request for Information"): Send requests and track the status of requests for information between eReferral requesters and receivers. The request is made in the form of a /Task (e.g., "please send Assessment X"), and the response comes in the form of a /Communication which may be text/html and include /DocumentReference with attachments.

/Task: "Process Request"

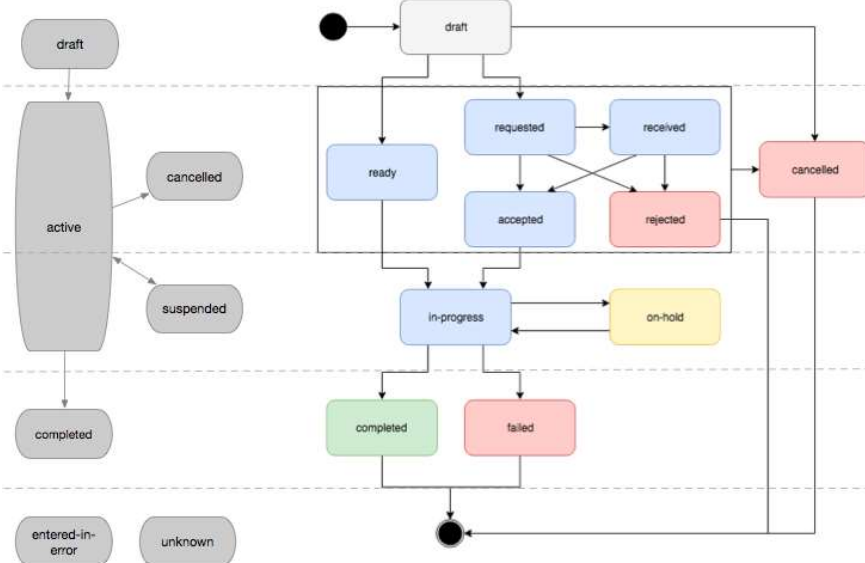
```
Task.BasedOn = reference(ReferralRequest)
Task.Code.Text = "Process Request"
Task.BusinessStatus = <custom_deployment_dependant>
Task.Status = <see table below>
```

ServiceRequest Statuses

ServiceRequest.status = _

Task State Machine

Task.status = _



ServiceRequest .status =	Task .Status =
draft	draft - when a referral has not yet been submitted
active	ready - after submitted, if provider has no "acceptance" step
active	requested - if provider has a accept/reject process
active	received - if the provider acknowledged receipt
active	accepted - if provider accepts referral for full processing
active	rejected - if provider rejects referral
cancelled	cancelled - if provider OR requester cancels (but wasn't rejected)
active	in-progress - while the provider is seeing the patient (rarely used)
suspended	on hold - while patient is in a "wait list" state
completed	completed - referral was fully processed
completed	failed - could not complete referral (e.g., no show, insufficient documentation, etc.)

2018-06-13_18-53-45.png 229 KB • [Download](#)

Proposed Decisions:

1. How should/could the **.output** element be used? Some options:

1. track the output from a ServiceRequest, such as an outcome message, or a consult note? (My preference is that outcome messages/consult notes be returned in a communication. Note that we would need different types of communications such as onSubmit, onComplete, RFI, etc... - Tim)

2. Track more ServiceRequests that were generated as a result of this referral (e.g., that is sent via central intake). Also, the **.input** element could be used to record which referral was forwarded to create THIS referral. You could then follow the input & output fields through the entire "referral-chain". (This deserves further consideration and discussion - but this decision can be delayed to a later phase).

- Alternately, consider an extension to ServiceRequest called **.forwardedTo** that would indicate where the referral was sent to. This is the opposite side of the "basedOn" field which is on the forwarded ServiceRequest.

Important Decisions & Rationale:

1. On receipt of a new ServiceRequest at the eReferral server, the eReferral server is responsible for attaching a task resource titled "Process Request". This task is responsible for tracking the status of the referral. When the eReferral server has completed its work with the eReferral, the task will be completed (or failed or cancelled). Correspondingly the ServiceRequest.status will also become completed (or cancelled).
2. Tasks may be used for other referral processes (such as communication), so create a valueset for **Task.code** for future-proofing purposes. Initially there will only be one value, "Process Request".
3. It is assumed that businessStatus is highly variable by deployment, thus just text in **businessStatus.text** is used, and not a code system.
4. Use **Task.lastModified** to indicate when any change of interest was done to to the serviceRequest and related resources (i.e., don't restrict to last time the Task resource itself was updated, which can be handled via the meta._lastupdated). This field can then be used in GET searches to find all updated referrals over specified period of time.
5. Track the history of a referral through **Task.relevantHistory** instead of ServiceRequest.relevantHistory. This is because (like Task.lastModified) the changes can happen across multiple resources connected to the ServiceRequest.
6. Track when a referral was closed with **executionPeriod.end**. For systems that track it, can use executionPeriod.start to indicate when work started on the referral.
7. **Task.requester** will not be used for now because it can be found in ServiceRequest.requester. Re-evaluate when reviewing the Communication process.

Comments:

Use the comments on this page to discuss the development of this resource profile. Conclusions made from the discussion will be added into this document, and the profile where applicable.

Comments & Events



Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group

I have advanced the /Task resource to be ready for broad discussion. If possible, we will include the task resource in today's Canadian eReferral WG discussion.

Jul 04, 2018 at 8:12 AM · No one was notified



Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group

Meeting discussion:

1. code: could find other uses for task with reviewing communications, changes to triage processes, etc... so set up a valueset for code, even though to begin with, we may only have one value (future proof the process)
2. Agreed that tracking when ANYTHING about the referral has changed is important to

the process. Post this question on task.fhir.org about WHERE to track this.

3. Agreed for end. also use .start to indicate when the provider started WORKING on the referral (which could be different than when it was submitted). End will be when it was closed.

4. pause on relevant history, for a future stage.

5. Question: use OUTPUT be used to FORWARD refer to new referrals? using output for referral communication outputs, re-evaluate when looking at communication resource.

6. Probably leave on task.requester, because we might have other of tasks.

Jul 10, 2018 at 12:04 PM · No one was notified



Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group

I'm going to recommend that we use **task.lastModified** to track when ANYTHING about the referral was updated.

Jul 30, 2018 at 4:15 PM · Notified 92 people



radhika verma

Hello!

We've a question about Task.code - is there a use case to support this element and if yes, should it be MS=true?

Thanks!

Nov 18, 2019 at 4:19 PM · Notified 88 people



Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group

From: <https://simplifier.net/guide/eReferralOntario/ProfileTask>

"For ServiceRequests, the eReferral server set the .coding.code="process-request" and .text to ="Process Request" when a new referral is submitted to the eReferral server. This code stays fixed for the life of this task. future work - if have other types of tasks may use alternate codes, e.g., for communications"

Really, it should be not only be mustSupport but it should be mandatory - at least insofar as we defined its use for eReferral.

Nov 18, 2019 at 4:28 PM · Notified 88 people



ion moraru

I don't know if the code attribute, with a fixed value, will bring any value in addition to the existing (more important, in my mind) attributes status, intent and businessStatus.

Instead of making Task.code mandatory have a rule related to the presence of the Task

resource in the bundle, i.e. If a Task resource is included in the bundle everyone must support it.

Nov 20, 2019 at 10:41 AM · Notified 88 people



Tim Berezny, CTO Caredove, Chair FHIR eReferral Specification Working Group

The code as it is isn't technically relevant in the very limited scope by which we have thus far used task. However, future looking if the use of the Task resource were to be broadened at all to other process flows (e.g., Process eConsult, Process Request for Information, etc...) then it would become critically important.

Nov 20, 2019 at 5:17 PM · Notified 88 people