eCR Now App Configuration Guide

# Introduction:

The guide is provided to help configure the eCR Now App for the Release3.0 version. For the Release 2.0 version use the eCR Now App Configuration Guide.docx.

The configuration described in the document will allow implementers to deploy the eCRNow App for public health reporting purposes. The app is designed to be used for a single clinical site (Single tenant) or multiple clinical sites (Multi-tenant). The App provides the ability to configure each EHR’s FHIR Server and appropriate parameters that are required for producing the eICRs or other public health reports. The next section shows the various screens and sample values.

# Accessing the Configuration Screen

The configuration screen can be accessed by running the [eCRNow app](https://github.com/drajer-health/eCRNow) and the [eCRNow-UI](https://github.com/drajer-health/eCRNow-UI) applications. Follow the instructions on the github links to install the app and run the applications.

**Step 1:**

* Run the eCRNow App following the instructions at [eCRNow app](https://github.com/drajer-health/eCRNow).
* Run the eCRNow-UI following the instructions at [eCRNow-UI](https://github.com/drajer-health/eCRNow-UI)

**Step 2:**

* Access the configuration screen using the local URL
  + [**http://localhost:3000/healthcareSettings**](http://localhost:3000/healthcareSettings)
  + Substitute localhost:3000 with your server details

You should be able to see the following screen:

Graphical user interface, text

Description automatically generated

**Configuration Terminology:**

* HealthCare Settings:
  + Represent the Healthcare organization for which the eCRNow App is being configured.
  + Since the app can be configured for multiple Healthcare organizations, you can view all the configured healthcare organizations by clicking on the Existing Healthcare Setttings.
* FHIR Configuration:
  + This screen is to configure the FHIR based information to interact with the EHR being used by the Healthcare organization.
* Transport Configuration:
  + This screen is to configure the options on how to send the data from the Healthcare organization to the Public Health Agency using the eCRNow App.
* App Configuration:
  + This screen contains parameters that are used by the application for filtering the amount of data being retrieved from the EHR.
* Organization Defaults:
  + This screen contains the default to be used for the healthcare organization such as name, OID, Namespace URL etc so that it can be used for FHIR/CDA payload generation.
* Response Options:
  + This screen contains the details on how to handle an incoming response from PHA to the Healthcare organization.

**Step 3: FHIR Configuration:**

The FHIR configuration is used by the eCRNow App to connect to the EHR and access data for public health reporting. The eCRNow App supports multiple ways of connecting to the EHR. These include

* System Launch – This is following the OAuth2 client\_credentials workflow.
* Multi-tenant System Launch – This is also following the OAuth2 client\_credentials workflow with the difference being there is a single authorization server that is shared across all the healthcare settings.
* Backend – This follows the [SMART on FHIR Backend Services](https://build.fhir.org/ig/HL7/smart-app-launch/backend-services.html) authorization.
* Username/Password – This uses a service account with Username / password credentials to generate an access token.

Require Aud Parameter:

* This parameter has to be turned on when the EHR Authorization process requires an aud parameter to be passed.

EHR Supports Subscriptions Parameter:

* This parameter has to be turned on when the EHR supports FHIR Subscriptions capability.

The following is a screenshot for the FHIR Configuration using System Launch:

Graphical user interface, text, application, email

Description automatically generated

**Multi Tenant System Launch**: The Multi Tenant System Launch is similar to the System Launch.

**Backend Services Authorization**: The Backend Authorization is similar to the System Launch, except that the Keystore alias has to be provided. The Keystore alias is the name of the certificate to be used from the Java Key Store for Backend Services Authorization.

**Username/Password Authorization**: The Username/Password Authorization is similar to the System Launch, except that instead of client Id and client secret , username and password would be used.

**Step 4: Transport Configuration:**

The transport configuration is used to route messages from the healthcare organization to the Public Health Agency. An intermediary (AIMS platform) is used for delivery of eICR messages. So all eICR configuration will use the AIMS platform endpoints for submitting the data.

The methods supported by the app include:

* Direct Transport
  + The app will use Direct transport (SMTP/IMAP) protocols to send and receive messages.
  + In order to use this option, the healthcare organization has to setup a dedicated account for the eCRNow App to use for sending and receiving messages.
  + The Healthcare organization can use their exiting Direct HISP and create an account within their existing HISP for transmission.
  + If the Direct HISP implementer has dedicated SMTP and IMAP URLs they should be configured as shown below.
  + POP3 support currently does not exist but will be added in the future.
* REST Api
  + The app will handoff the messages to a RESTful endpoint hosted by the healthcare organization.
  + The Healthcare organization is then responsible to take the payload and submit it to the Intermediary/PHA as applicable.
* FHIR
  + The app will handoff the messages to a FHIR endpoint hosted by the PHA or an Intermediary.
  + This options is not currently supported for eICRs, but is used for other Medmorph piloting and testing.
* XDR
  + The app will handoff the messages to a XDR endpoint hosted by the healthcare organization.
  + The XDR endpoint is expected to be a RESTful endpoint. The Healthcare organization is then responsible to take the payload and submit it to the Intermediary/PHA as applicable using the XDR protocols.

Graphical user interface, text, application, email

Description automatically generated

**Step 5: App Configuration:**

This section contains parameters to be used to filter/limit data based on the encounter times. The Encounter Start Time threshold will look back from the encounter start time for the configured number of hours to load the data. (For e.g as shown in the figure below, the app will look back 72 hours to limit the amount of data being loaded for the purposes of loading the data where applicable.)

Similarly the Encounter End Time threshold will look at data associated with the encounter or the patient for upto 72 hours after the encounter has been closed.

Application, Teams

Description automatically generated with medium confidence

**Step 6: Organization Configuration:**

This screen contains information about the organization. The following are descriptions of the fields:

* Name – name of the Organization
* Namespace URL – This is the URL that can be used to generate unique ids for the healthcare organization. This can be set to the FHIR Server URL itself or an organization url.
* Organization Id: An Id that can be used to identify the healthcare organization. This could be used when the same eCRNow App instance is used for multiple healthcare organizations.
* Assigning Authority OID: This is the registered Object Identifier that is used as a namespace for creating Ids and is used in the CDA generation as the root attribute for identifiers such as MRNs etc.
* Provider Id : This is a default provider Id reference to a FHIR Resource that can be set when POSTing response data (e.g reportability responses) to the EHR as document references. The value gets overridden with any value that is supplied as part of the Access tokens.

Graphical user interface, text, application, email

Description automatically generated

**Step 6: Response Options Configuration:**

This section contains the information that is needed to create a Document Reference when a response is received from the PHA/Intermediary. For eICRs, DocumentReference objects are created when a Reportability Response is received from the AIMS platform.

The options are:

* Create a Document Reference: This option creates a document reference in the EHR. The document references are expected to be surfaced to the Provider as part of the regular handling of external documents.
* Invoke REST API: This options provides the response received to a RESTful API hosted by the Healthcare organization for further processing and surfacing it to the Provider.

**Graphical user interface, text, application, email, Teams

Description automatically generated**

**Step 7: eCR Specifications/KAR Configuration:**

This option is enabled only after finishing Step 6 and saving the data. Once the data is saved, the healthcare setting can be edited by going to the Existing Healthcare Settings list.

Graphical user interface, text, application, email, Teams

Description automatically generated

Once you edit the healthcare setting then you will see the following:

Graphical user interface, text, application, Teams

Description automatically generated

The Select FHIR Server URL will be populated with the information on the local file system. These local file system location is configured based on the “kar.directory=//users//nbashyam//Downloads//kars” property.

Each subfolder in the {{kar.directory}} will be called a local-repo-[folder name].

If there are no sub-folders, then the list of ERSD files / Knowledge Artifacts will be displayed directly as local-repo.

In the above example, I have the following directory structure within the {{kar.directory}}.

Graphical user interface, text, application

Description automatically generated

Within these directories, you can place specifications related to the various different public health use cases as shown below.

Graphical user interface, application, website

Description automatically generated

Once the specifications are in place, they can be activated for processing. Without the specifications in place, you cannot activate them. If they are not activated, they are not processed when the app gets notified of potential events.

The following is a figure of activating the ERSD specification for processing and reporting COVID-19 cases.

Graphical user interface, application, email

Description automatically generated

“As of 2022, all eCR FHIR App implementers are expected to work towards support of using the eCR workflow for all reportable conditions. In alignment with this guidance, ensure you set the “Only Covid” flag is not selected. It is present only for backwards compatibility.

Once all the above changes are saved, the configuration for a single Healthcare Setting is completed.

**Additional Details:**

For FHIR R4, here is an example of scopes that can be used.

**Scopes for R4:**

launch,online\_access,offline\_access,user/Patient.read,user/Condition.read,user/Encounter.read,user/MedicationAdministration.read,user/MedicationOrder.read,user/MedicationStatement.read,user/Observation.read,user/Immunization.read,user/DiagnosticReport.read,user/Practitioner.read,user/ServiceRequest.read,patient/Patient.read,patient/Condition.read,patient/Encounter.read,patient/MedicationAdministration.read,patient/MedicationOrder.read,patient/MedicationStatement.read,patient/Observation.read,patient/Immunization.read,patient/DiagnosticReport.read,patient/ServiceRequest.read

**Assigning Authority Id**

* 1. This is being used in the CDA eICR produced as the root element for ids which do not have the System URL populated in the FHIR Resources.

**Encounter Start and End Threshold**

* 1. This is a time parameter in terms of hours. This is used to identify the resources relevant for eICR creation when the encounter context is not provided for identifying the specific resources.
     1. For example: A start threshold of 3 means the App will subtract 3 hours from the time the patient context was received by the app.
     2. Similarly an end threshold of 30 means the App will add 30 hours from the time the patient context was received by the app.
     3. During eICR creation, if the encounter context is not provided the App will use the window of time
        1. Patient Launch – 3 hours to Patient Launch + 30 hours to identify the data to be used for eICR reporting.

**Step 5: Save the App Configuration**

You should receive a success notification and you are ready to start using the app for the specific clinical site.

# Launching the Patient instance in the App.

The section outlines how to use System Accounts and specific APIs to launch a Patient instance for processing within the app. Launching a Patient instance amounts to the app processing all the different specifications based on the notification received.

**Patient Launch API:**

Method: POST

URL: http://localhost:8081/api/launchPatient

RequestBody: {

"fhirServerURL":"https://www.drajer.com/fhir/r4/ec2458f2-1e24-41c8-b71b-0e701af7583d",

"patientId":"12742571",

"encounterId":"97953900"

}

Header: Content-Type: application/json

Response:

StatusCode: 200

Message: “Patient Instance launched for processing successfully.”

# Updating ERSD Releases

1. Download the ERSD release from the ERSD website when you receive notifications of new ERSD files being available.
   1. NOTE: You will receive notifications only if you subscribed to receive notifications of changes from the ERSD website.
2. Place the new ERSD file in the {{kar.directory}} folder or its sub-folder where you had previously placed the Knowledge Artifact.
   1. NOTE: It is better to remove the old ERSD once you put in the NEW ERSD file after a period of time. (Say a period of one month, this is to ensure that any old transactions that were started using the older ERSD can be completed).
      1. However the Old ERSD should be DISABLED using the eCRNow-UI or an equivalent API on healthcare settings immediately as soon as the NEW ERSD is put in place.
3. Configure / Activate the newly downloaded ERSD so that it can take effect for all future transactions.
4. If you have a custom-queries file, you have to potentially rename the custom-queries file to use the new ERSD file’s PlanDefinition Id as per custom-query configuration.

# Externalizing Configuration Files:

There are a few configuration files that the eCRNow App uses which are best deployed to a shared file system or an AWS S3 bucket or a Azure Blob storage as per the deployment configuration.

1. ecrNow Log file (Configured in the application.properties using the logging.file.name properties.)
2. ersd file for Release 2.0 of the Application (Configured in the application.properties using the ersd.file.location.)
3. eICR schematron file (Configured in the application.properties using schematron.file.location)
4. XSD Schema file (Configured in the application.properties using xsd.schemas.location)
5. Knowledge Artifact (ERSD) directory for the Release 3.0 of the app (Configured in the application.properties using kar.directory)
6. Output directory where files get written for debugging and analysis such as JSON and XML files (Configured in the application.properties using the bsa.output.directory)
7. Custom Query directory (Configured in the application.properties using custom-query.directory)

# Re-Launching the Patient instance in the App.

The Re-Launch Patient API is to be used when a patient/encounter instance that is already launched in the app is either suspended and/or completed. However the data associated with the patient and the encounter may change after the encounter is either suspended or completed. In these cases there is a reason to potentially create a eICR and notify the PHA. To accommodate this workflow, the re-launch API of the eCRNow app can be used.

The following are example scenarios when the re-launch scenario can be used:

* An encounter is designated as “finished”, however after 5 days a new lab result is added to the same encounter information. In this case the best way to notify the app to check for a potential eICR is to use the relaunch patient api.

**Re-Launch Patient API:**

Method: POST

URL: http://localhost:8081/api/relaunchPatient

RequestBody: {

"fhirServerURL":"https://www.drajer.com/fhir/r4/ec2458f2-1e24-41c8-b71b-0e701af7583d",

"patientId":"12742571",

"encounterId":"97953900"

}

Header: Content-Type: application/json

Response:

StatusCode: 200

Message: “Patient Instance launched for processing successfully.”