NBS 7.11 Data Ingestion Rhapsody Route Integration

Introduction:

Rhapsody is an integrated software tool that provides connectivity and data sharing within and among Hospitals, Information Exchanges, and Public Health Organizations. Rhapsody is the Windows based integration engine embedded in NBS which accepts HL7 and other. It reads HL7 messages and directly posts them into the Data Ingestion service which then validates and converts them to patient data in NBS system, ensuring data integrity and accuracy.

Pre-Requisites

- Rhapsody is installed.
- MS SQL Server Database is installed on the same system that has Rhapsody.
- Server name of the Database
- SQL scripts to create the database and tables. The scripts are located at NEDSS-DataIngestion/data-ingestion-service/src/main/resources/rhapsody
- Rhapsody route information
- Keycloak Client Id and Client Secret required to connect to the Data Ingestion service.

Installation

After installing the database on the same machine where rhapsody is installed, connect to the database engine using Windows Authentication. Remember to record the Server name, as it will be required during the setup of rhapsody's routing configuration.

Once you've logged in, right-click the SQL Server name and select "New Query" to open the SQL editor window.

Create Database:

1. Navigate to this location in a browser window: NEDSS-DataIngestion/data-ingestion-service/src/main/resources/rhapsody
2. Open rhapsody-create-local-database.sql, copy the code and then paste it in the new query section of the SQL server and execute it.
3. The output looks like this

```
Commands completed successfully.

Completion time: 2023-11-09T20:51:32.8524141-05:00
```

Create Tables:

1. Once the database is created, we need to create a user and tables that we need for Rhapsody routes. To create those resources, within the same path NEDSS-DataIngestion/data-ingestion-service/src/main/resources/rhapsody, open rhapsody-create-local-database-tables.sql script , copy the code and then paste it in a new query section of the SQL server.
2. Within the sql script, replace the section that says, “WITH PASSWORD='ReplaceThisPasswordWithYourPassword123$'” with your own password and execute it.
3. Your output will look something similar as below. Along with tables, it creates a new user rhapsodyuser.

```
Commands completed successfully.

Completion time: 2023-11-09T20:54:53.0018562-05:00
```

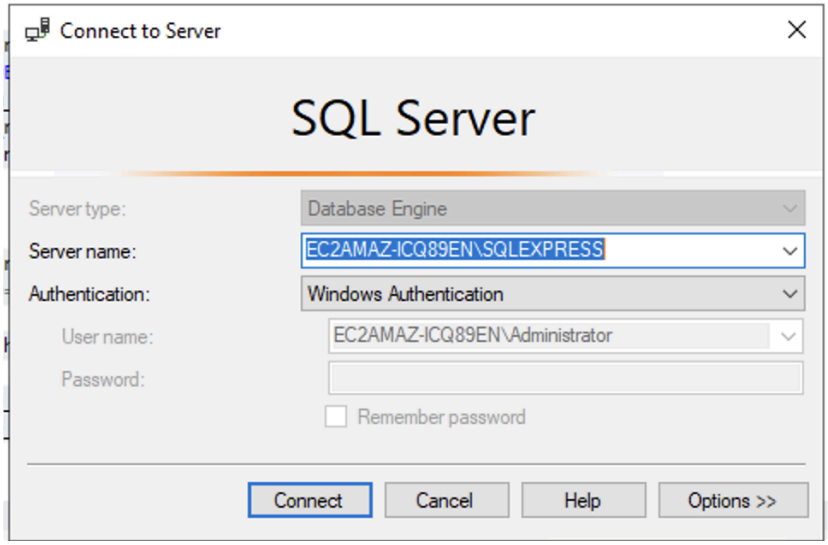
Enable TCP/IP protocol:

1. Once the tables and user are created, we need to enable TCP/IP protocol for port 1433. Here are the instructions on how to do that.
2. Click the Windows Start button.
3. Under Programs, open SQL Server Configuration Manager, and then expand SQL Server Network Configuration.
4. Click Protocols for Instance Name, and then make sure TCP/IP is enabled in the right panel and double-click TCP/IP.
5. On the Protocol tab, make sure Listen All is Yes.
6. Click the IP Addresses tab → Under IPAll, enter 1433 as TCP Port.
7. Click OK.
8. In SQL Server Configuration Manager, in the left pane, select SQL Server Services.

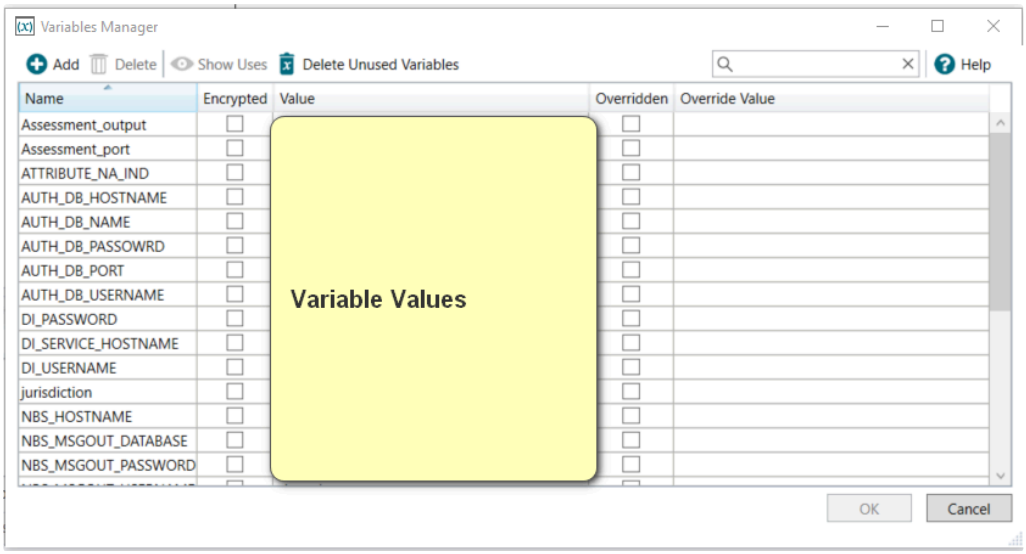
9. In the results pane, right-click SQL Server (MSSQLServer) or a named instance, and then select Restart. It'll take couple of minutes for the SQL Server to restart.

Rhapsody Route Configuration

1. Open a browser and paste the URL as NEDSS-DataIngestion/data-ingestion-service/src/main/resources/rhapsody/NBS-Data-Ingestion-ELR-Route.rlc at main · CDCgov/NEDSS-DataIngestion
2. Download this route to the computer where Rhapsody and MS SQL Server are installed.
   a. Go to Rhapsody IDE and click File → Open. Select the .rlc file that you downloaded and click Next → Finish.
   b. On the Import Configuration Passwords dialog box, click OK.
   c. Once the route is installed, click on View → Variables Manager. Below are the variables that you need to update.

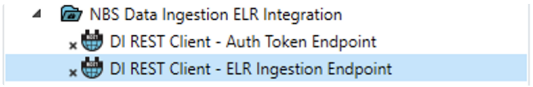


- AUTH_DB_HOSTNAME → MS SQL Server Hostname (the name that you noted during installation, or you can find this in the server connection window.
- AUTH_DB_PORT → 1433
- AUTH_DB_NAME → NBS_Data_Ingestion_Local
- AUTH_DB_USERNAME → rhapsodyuser
- AUTH_DB_PASSWORD → Password that you set while creating database user and tables.
- DI_SERVICE_HOSTNAME → DI service host name. Ex: dataingestion.datateam-testnbs.eqsandbox.com
- DI_USERNAME → Username you provided to system admin during onboarding.
- DI_PASSWORD → Password you provided to system admin during onboarding.

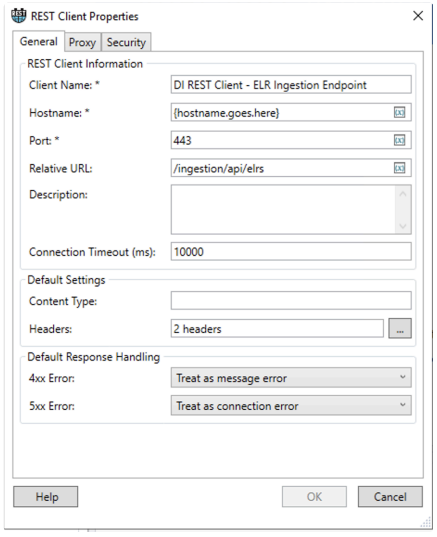


3. Once you have updated the variables, select the route, and click Check in selected components.
4. To setup **Keycloak authentication**, you need to click on Rest Clients Tab on Workspace.

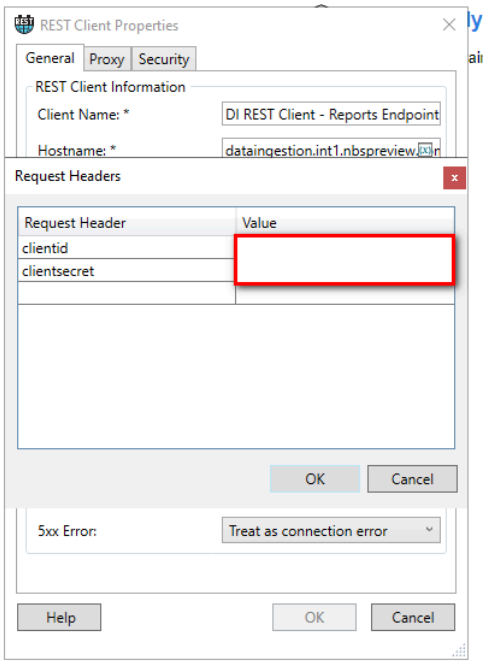
- Select and update the existing rest client or you can create new for different environment with the configuration based on the existing client.
- For the Route to work, you will update both Token endpoint and Ingestion endpoint with valid client id and client secret.



5. Once the dialog is opened, click on Headers.

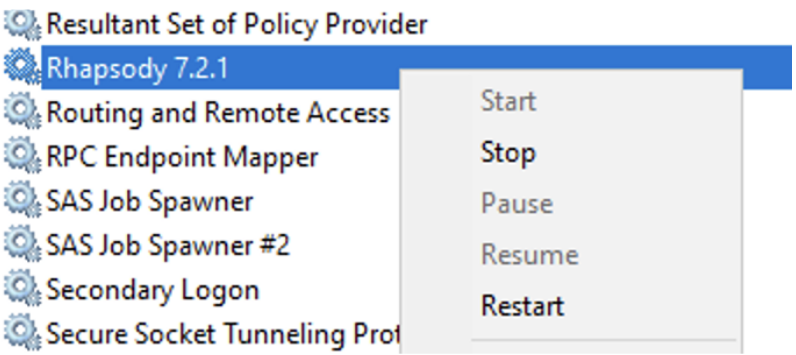


6. Update or add clientid and clientsecret accordingly depending on your environment. You can get this information from the system admins.



7. Then go to Rhapsody Installation directory and find the below file.

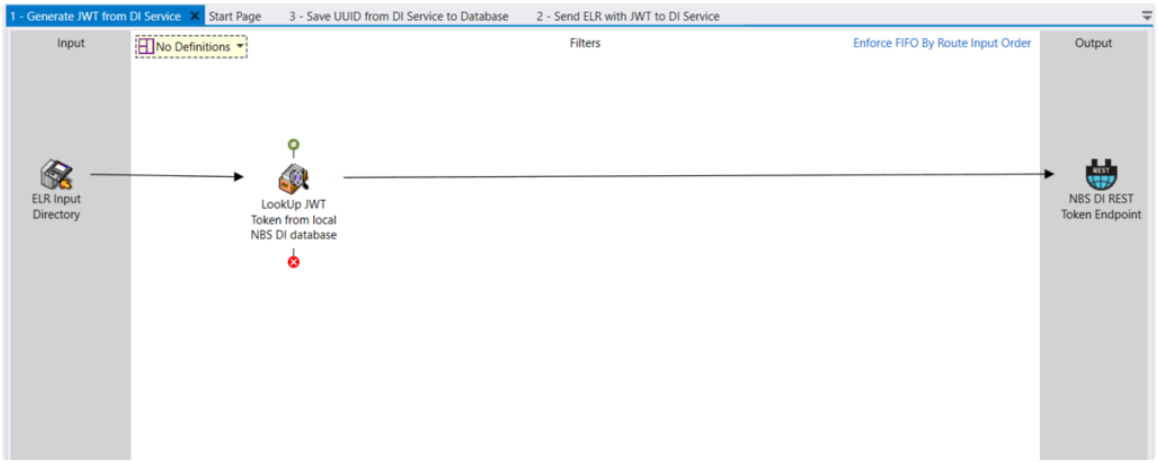
- <RhapsodyInstallationDir>/rhapsody/rhapsody.properties
- Search for this variable in the properties file SslProvider.CipherSuites.Strong and uncomment all the lines associated with it. This is needed to talk to Data Ingestion Service that is running in cloud environment.
- Once uncommented, save the file.
- Click the Windows start button and type in services. In the Services window, find Rhapsody x.x.
- Right click and hit restart.
- This will take few minutes to restart and you're good to send data to the Data Ingestion Service



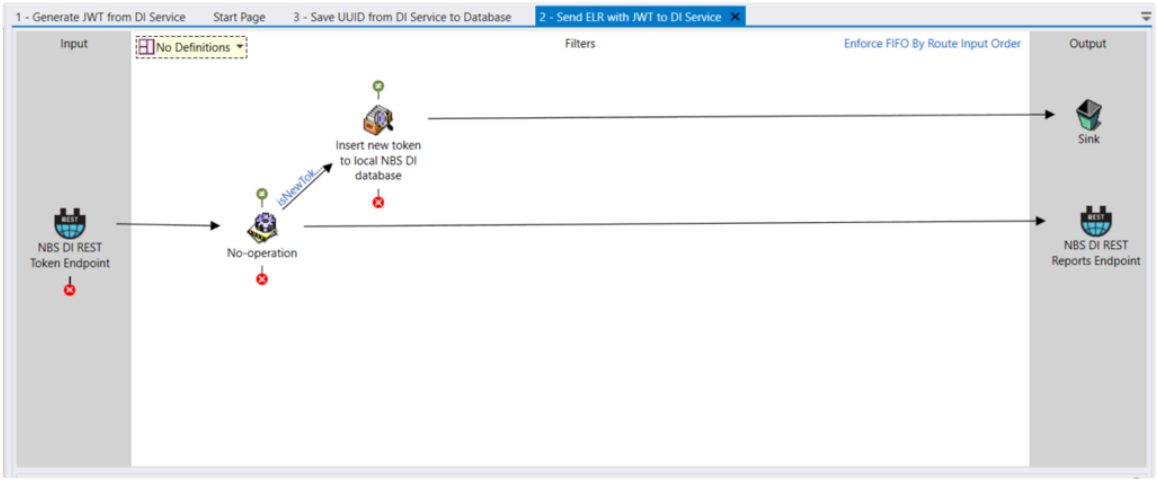
Send ELR via Rhapsody

Once the Rhapsody is installed and configured as above, the 3 routes should look like below. Within the ELR Input directory, you can paste either single or multiple HL7 files with the .txt extension.

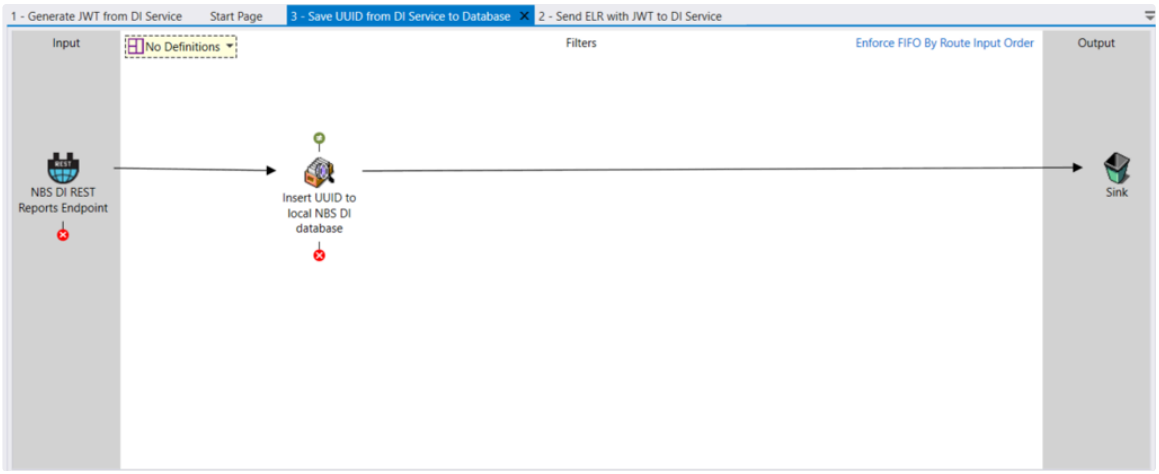
Generate JWT from DI Service:



Send ELR with JWT to DI Service



Save UUID from DI Service to Database



Verify ELR Submission via Rhapsody

1. Once Data is submitted, we can validate it within the local database that was created earlier. Navigate to the SQL Server installed earlier and click on new query. Copy and paste the below query and execute it.

```
Select * from [NBS_Data_Ingestion_Local].dbo.[report_uuid]
order by created_on desc;
```

2. Capture the UUID for the ingested data from here. Now we can use Data Ingestion CLI or Rest API to check the status of each UUID, that is whether it made it to the NBS end system and passed validation over there or it failed validation within the Data Ingestion system.