
Article

[Yuri Marx](#) · Aug 25, 2022 8m read

[Open Exchange](#)

Transform custom healthcare messages to SDA

EHR (Electronic Health Record) systems are modeled in a proprietary format/structure and are not based on market models such as FHIR or HL7. Some of these systems can interoperate data in a proprietary format for FHIR and other market models, but others can not. InterSystems has two platforms that can interoperate proprietary formats for market ones: InterSystems HealthShare Connect and InterSystems IRIS for Health. The transformation functionality (DTL - Data Transformation Language) of these platforms can receive data in any format, structure, or communication channel (CSV, JSON, XML, and others via FTP, File, HTTP, etc.) and transform it into market formats directly (FHIR, CDA, HL7 and so on). However, InterSystems has an intermediate format called SDA (Summary Document Architecture) which is used by these platforms to effortlessly generate FHIR STU, R3, R4, HL7v2, HL7v3, and others. In addition, when in SDA format, health data can be persisted in the HealthShare UCR. Thus, first, the proprietary/custom format is transformed to SDA, and then the data can be automatically converted into any market format, as well as saved in HealthShare. In this article, we will show you how to transform proprietary/custom data into SDA using IRIS for Health. The sample data we used was generated by the SYNTHEA bulk data generation project (<https://synthea.mitre.org/downloads>). We will convert 1000 patients from a CSV file to SDA, using the Interoperability features of IRIS for Health.

Article support application – custom2sda

Install the sample application that will be used with this article by following the instructions:

If you want to install using ZPM:

1. Open IRIS Namespace with Interoperability Enabled.
2. Open Terminal and call: `USER>zpm "install custom2sda"`

If you want to install using Docker:

1. Git clone and pull the repo into any local directory:

```
$ git clone https://github.com/yurimarx/custom2sda.git
```

2. Open the terminal in this directory and run:

```
$ docker-compose build
```

3. Run the IRIS container with your project:

```
$ docker-compose up -d
```

4. Open the production (<http://localhost:52775/csp/healthshare/user/EnsPortal.ProductionConfig.zen?PRODUCTION=customsda.CustomT> [oSDAProduction](#)) and start it. (user `SYSTEM` and password `SYS`). It will read patients.csv and convert it to SDA.

Create a CSV Record Map to get the custom/proprietary data

In the previous step, you ran the production that read the patients.csv and turned it into SDA. Now we will do the same with patients2.csv. We could take advantage of the current production, but I would like to demonstrate how to create everything from scratch. So, stop this production, and let's do the following.

1. Go to Management Portal ([http://localhost:52775/csp/sys/%25CSP.Portal.Home.zen?\\$NAMESPACE=USER](http://localhost:52775/csp/sys/%25CSP.Portal.Home.zen?$NAMESPACE=USER), make sure you are in the User namespace).
2. Create a CSV Mapper for the patients2.csv. Interoperability > Build > CSV Record Wizard:

Interoperability > CSV Record Wizard

CSV Record Wizard

Create RecordMapper

Sample file

RecordMap name

Separator

Record Terminator ☒ CRLF ☐ CR ☐ LF Other

Character Encoding

Sample has header row ☒

Keep SQL Column order ☐

Quote-style escaping in use ☐

Note: Select CRLF for Record Terminator

3. Click Create RecordMap to open the Record Mapper UI and change the Target Classname to customsda.Patients2RecordMap.Record:

Interoperability > Record Mapper - (Patients2RecordMap)

Record Mapper

Open New Save Save As Generate Delete CSV Wizard

Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY

Select sample file Undo Hide sample Refresh sample

User.Patients2RecordMap

Field	Datatype	Required	Repeating	Ignore
1 Id	0.1 %String			
2 BIRTHDATE	0.1 %String			
3 DEATHDATE	0.1 %String			
4 SSN	0.1 %String			
5 DRIVERS	0.1 %String			
6 PASSPORT	0.1 %String			
7 PREFIX	0.1 %String			
8 FIRST	0.1 %String			
9 LAST	0.1 %String			
10 SUFFIX	0.1 %String			
11 MAIDEN	0.1 %String			
12 MARITAL	0.1 %String			
13 RACE	0.1 %String			

Record

Target Classname: customda.Patients2RecordMap.Record

Batch Class:

Type: Character Encoding: Right justify: ☐

Annotation:

Leading data:

Padding Character: None ☐ Space ☐ Tab ☐ Other

Record Terminator: None ☐ CRLF ☐ CR ☐ LF ☐ Other

Allow Complex Record Mapping: ☐

Field separator(s):

Add Separator:

Repeat separator:

4. Select BIRTHPLACE and set MAXLEN=200 in the Datatype Parameters field. By default, all %String fields store 50 characters, but BIRTHPLACE and ADDRESS need more space. Do the same for ADDRESS:

Interoperability > Record Mapper - (Patients2RecordMap)

Record Mapper

Open New Save Save As Generate Delete CSV Wizard

Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY	GENDER
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY	GENDER
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY	GENDER
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PASSPORT	PREFIX	FIRST	LAST	SUFFIX	MAIDEN	MARITAL	RACE	ETHNICITY	GENDER

Select sample file Undo Hide sample Refresh sample

User.Patients2RecordMap

Field	Datatype	Required	Repeating	Ignore
11 MAIDEN	0.1 %String			
12 MARITAL	0.1 %String			
13 RACE	0.1 %String			
14 ETHNICITY	0.1 %String			
15 GENDER	0.1 %String			
16 BIRTHPLACE	0.1 %String(MAXLEN=200)			
17 ADDRESS	0.1 %String(MAXLEN=200)			
18 CITY	0.1 %String			

Field

Make Composite

Name: BIRTHPLACE

Datatype: %String

Annotation:

Required: ☐ Repeating: ☐ Ignore: ☐

Datatype Parameters: MAXLEN=200

Note: The %String datatype has a default MAXLEN of 50. Override the default if you expect more data.

SQL Column Number: Index:

5. Click the Save button and Generate button. Accept default options and click Ok to Generate RecordMap classes.

GENERATE RECORDMAP

RecordMap Name:
User.Patients2RecordMap

Target Classname:
customsda.Patients2RecordMap.Record

Advanced persistence options

Compile generated classes
Compile generated classes ▾

Compile flags
ck

Overwrite behavior
Always overwrite existing classes ▾

Clear existing storage
☐ Clear the existing storage definition

You should only clear the storage if you do not have any data. In most cases, you should leave this option unchecked.


Generated Classes

New classes
customsda.Patients2RecordMap.Record

Cancel

OK

6. Click Interoperability to go to the next tasks:

 Management

Server ec32acc35279 Namespace USER [Switch](#)

Interoperability >

Record Mapper - (Patients2RecordMap)

Record Mapper

Open

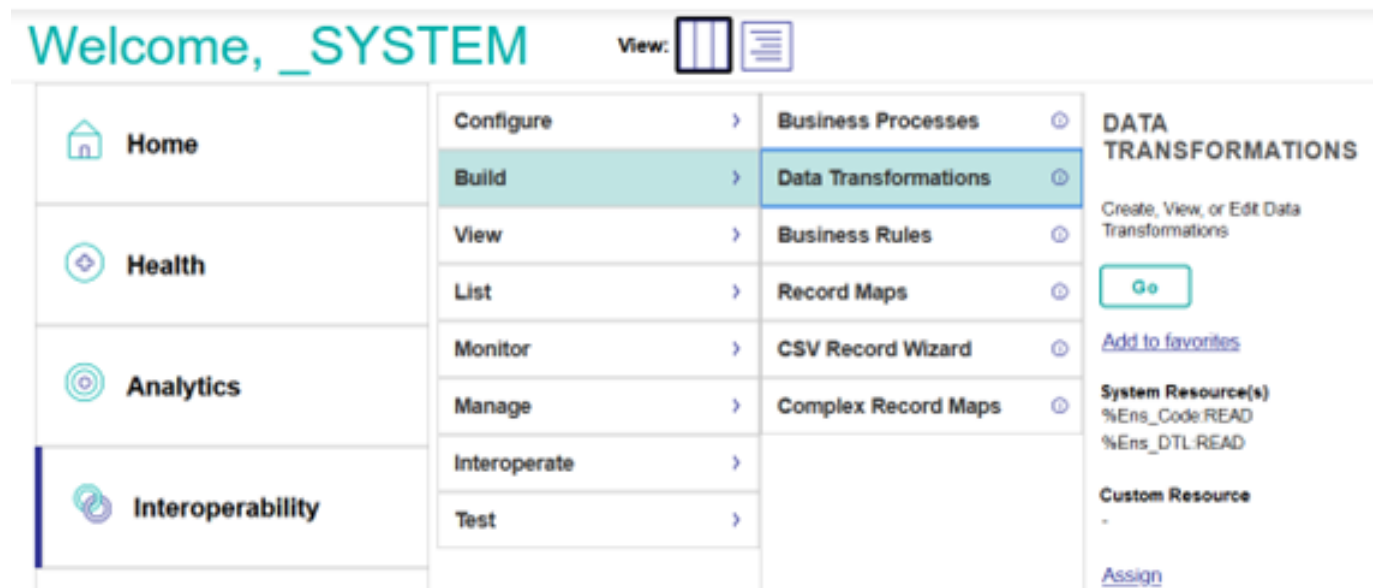
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PA
Id	BIRTHDATE	DEATHDATE	SSN	DRIVERS	PA

Create the Data Transformation from Custom to SDA

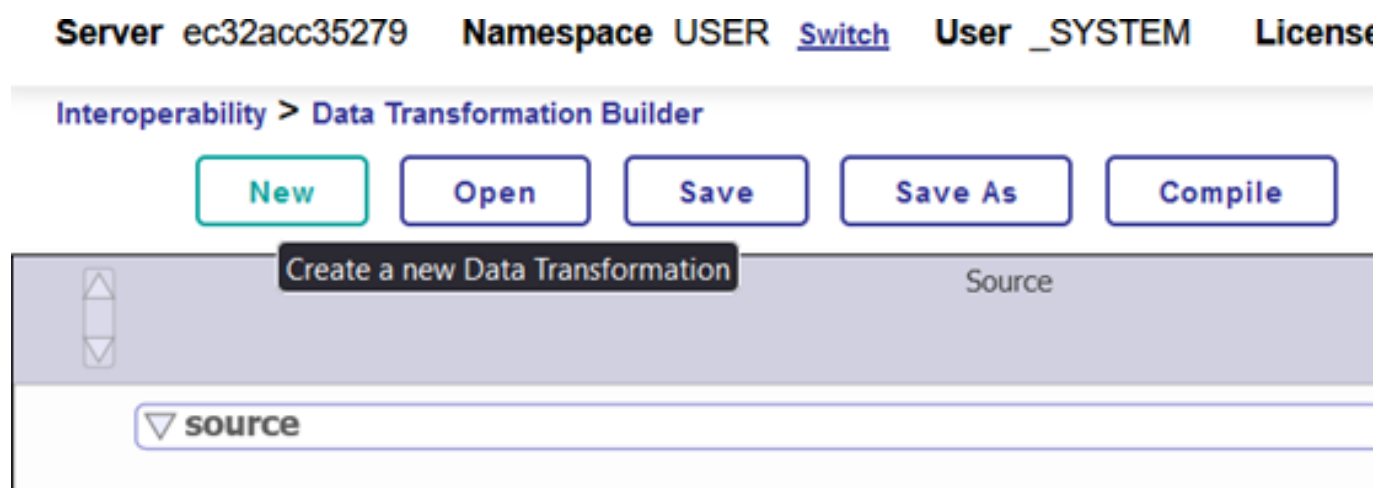
It is time to use the DTL to visually build the transformation map from custom to SDA.

Page 4 of 22

1. Click Interoperability > Build > Data Transformations > Go button:



2. You can see the Data Transformation Builder here. Click the New button:



3. Change the Data Transformation Wizard set Package to customsda, and the Name to PatientDTL2:

DATA TRANSFORMATION WIZARD

Create a new Data Transformation definition.

Package



Class package containing this Data Transformation

Name



Name of this Data Transformation

Description

Source Type
☒ All Messages ☐ HL7 ☐ X12 ☐ ASTM ☐ EDIFACT ☐ XML


Source Class 
Source Document Type 

Target Type
☒ All Messages ☐ HL7 ☐ X12 ☐ ASTM ☐ EDIFACT ☐ XML

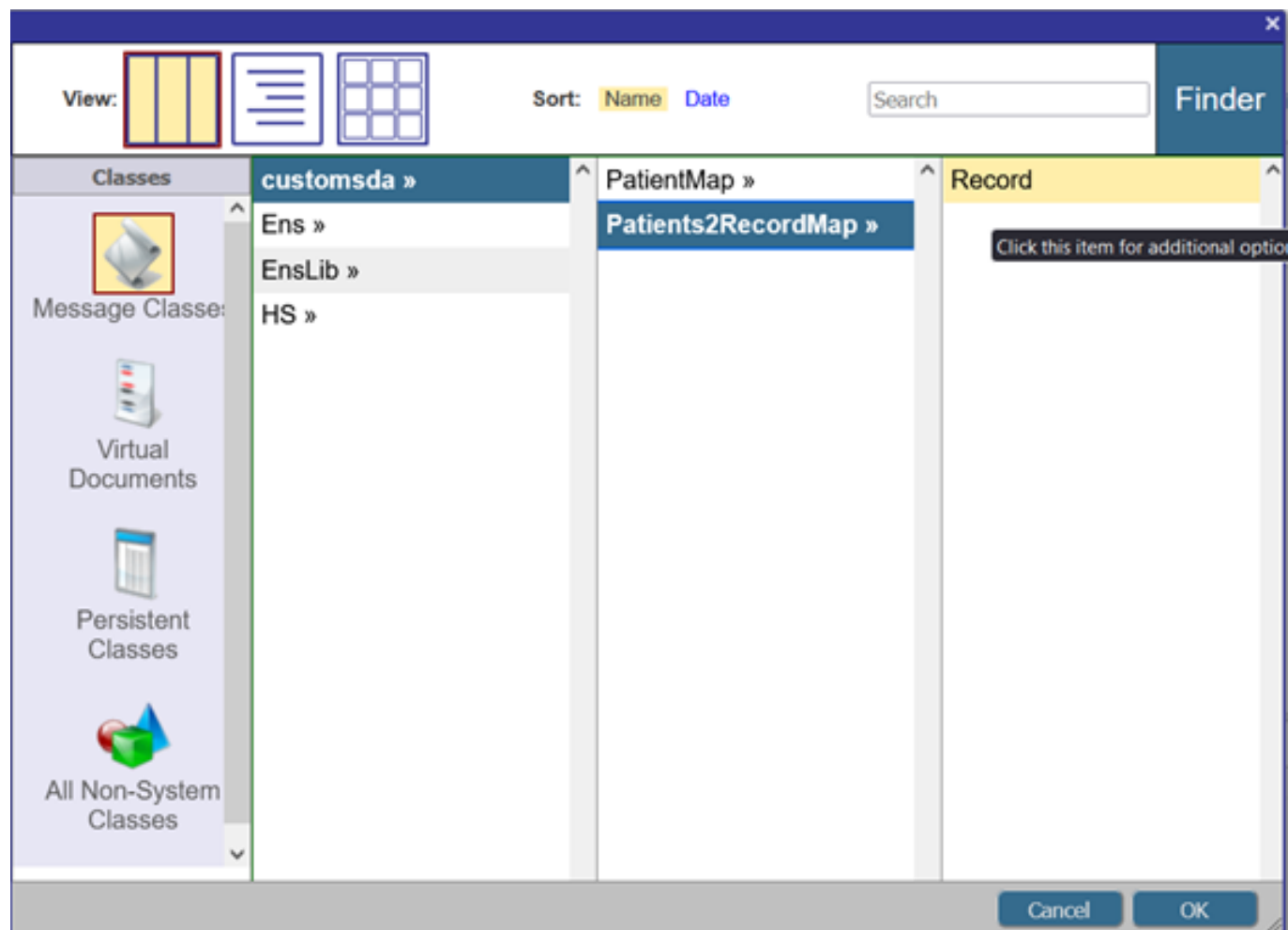
Target Class 
Target Document Type 

- At this point, we will set Source Class. Click the magnifier icon near the Source Class field:

Source Type
☒ All Messages ☐ HL7 ☐ X12 ☐ ASTM

Source Class
 

- Click Message Classes > customsda > Patients2RecordMap > Record:



6. At this stage, the Source Class should look like this:

Source Type

☒ All Messages ☐ HL7 ☐ X12 ☐ ASTM

Source Class

🔍

7. In the Target Type section, select XML and accept Target Class with the value EnsLib.EDI.XML.Document:

Target Type

☐ All Messages ☐ HL7 ☐ X12 ☐ ASTM ☐ EDIFACT ☒ XML

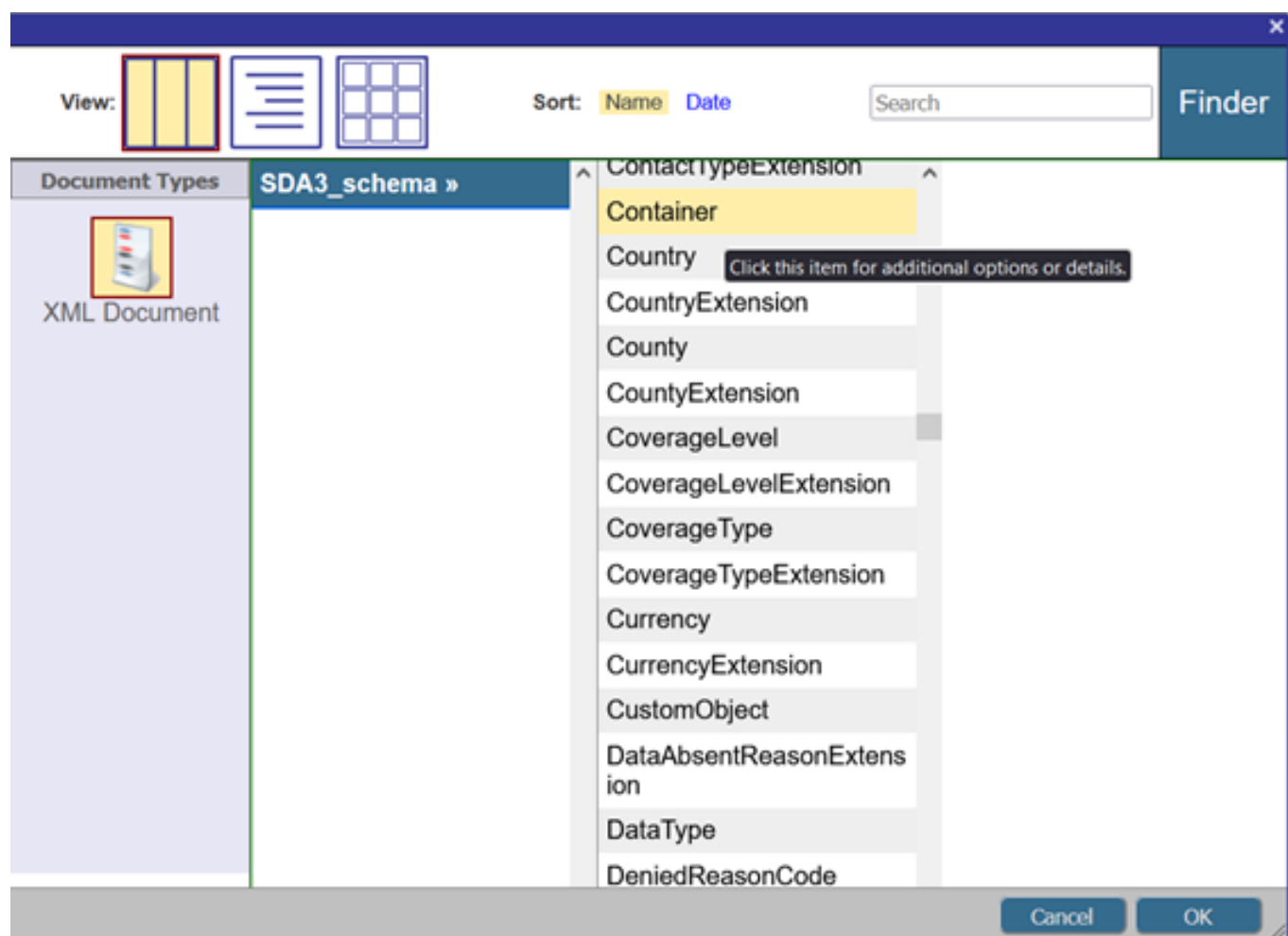
Target Class

🔍

Target Document Type

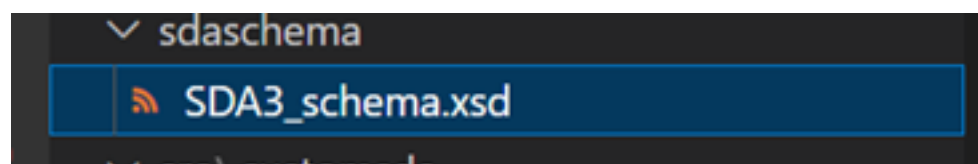
🔍

8. Select the Magnifier icon near the Target Document Type and click XML Document > SDA3schema > Container:



Note 1: Container is the root element for all SDA elements, like Patient.

Note 2: for SDA3schema to be available the following actions are needed:
Copy SDA3schema.xsd to your local file system:



Import the SDA3 XSD Schema:

```
zn "USER"
do $System.OBJ.LoadDir("/opt/user/src","ck",,1)
Do ##class(EnsLib.EDI.XML.SchemaXSD).Import("/opt/user/sdaschema/SDA3_schema.xsd")
zpm "load /opt/user/ -v"
```

9. Now, when the source and target are configured, click the OK button:

DATA TRANSFORMATION WIZARD
Create a new Data Transformation definition.

Package
customsda
Class package containing this Data Transformation

Name
PatientDTL2
Name of this Data Transformation

Description

Source Type
☒ All Messages ☐ HL7 ☐ X12 ☐ ASTM ☐ EDIFACT ☐ XML

Source Class
customsda.Patients2RecordMap.Record

Source Document Type

Target Type
☐ All Messages ☐ HL7 ☐ X12 ☐ ASTM ☐ EDIFACT ☒ XML

Target Class
EnsLib.EDI.XML.Document

Target Document Type
SDA3_schema.Container

Cancel OK

10. Now we have the source and target fields available for visual mapping:

Interoperability > Data Transformation Builder - (customsda.PatientDTL2)

New Open Save Save As Compile 100% -Add Action- View: [Icons]

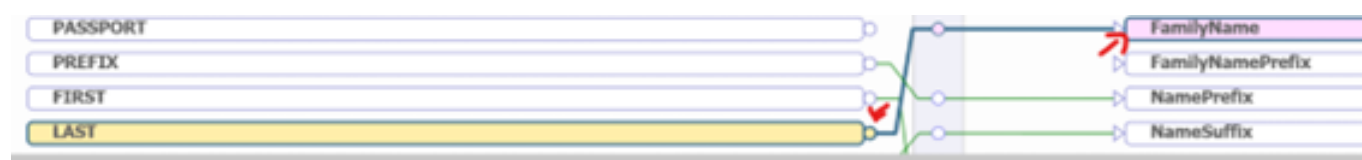
Source
customsda.Patients2RecordMap.Record

- source
 - %Source
 - Id
 - BIRTHDATE
 - DEATHDATE
 - SSN
 - DRIVERS
 - PASSPORT
 - PREFIX
 - FIRST
 - LAST

Target
EnsLib.EDI.XML.Document
SDA3_schema.Container

- target
 - Patient
 - Encounters()
 - Alerts()
 - AdvanceDirectives()
 - Allergies()
 - IllnessHistories()
 - SocialHistories()
 - FamilyHistories()
 - Guarantors()
 - Diagnoses()

11. To create a transformation, you need to drag the circle of the source field and drop it into the arrow of the target field, line by line.



12. In the Actions section, you can see the set results:

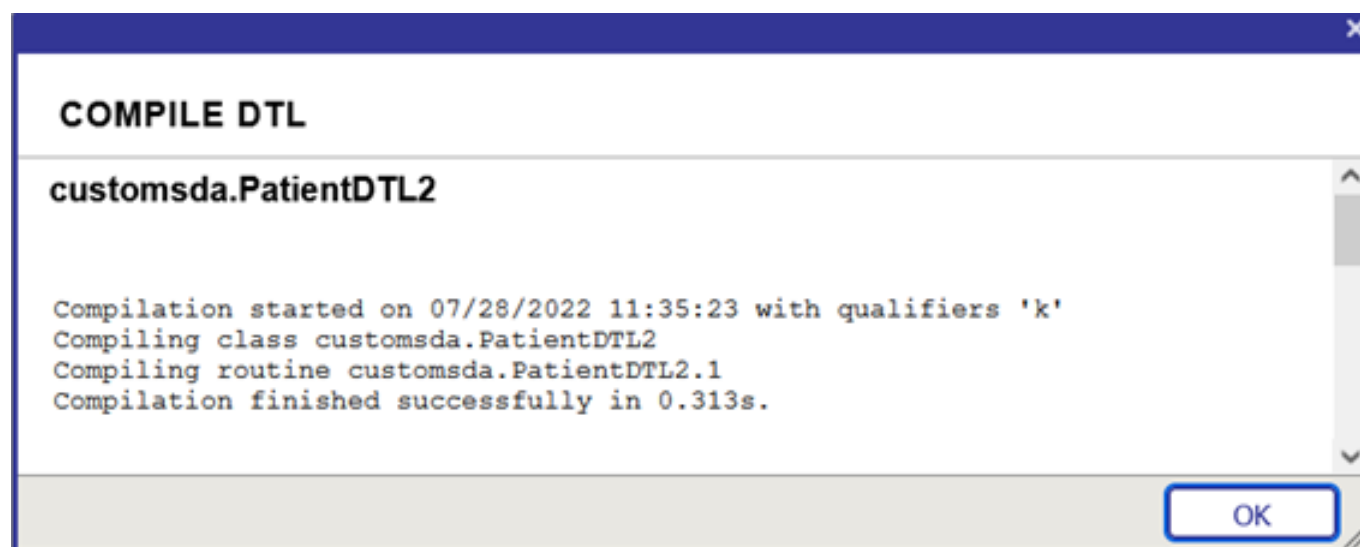
# Action	Condition	Property	Value	Key / Transform
1 set		target.(Patient.Name.GivenName)	source.FIRST	--
2 set		target.(Patient.Name.FamilyName)	source.LAST	--
3 set		target.(Patient.Name.MiddleName)	source.MAIDEN	--
4 set		target.(Patient.Name.NameSuffix)	source.SUFFIX	--
5 set		target.(Patient.Name.NamePrefix)	source.PREFIX	--

13. After mapping all the fields, your actions list will look like this:

# Action	Condition	Property	Value
1 set		target.(Patient.Name.GivenName)	source.FIRST
2 set		target.(Patient.Name.FamilyName)	source.LAST
3 set		target.(Patient.Name.MiddleName)	source.MAIDEN
4 set		target.(Patient.Name.NameSuffix)	source.SUFFIX
5 set		target.(Patient.Name.NamePrefix)	source.PREFIX
6 set		target.(Patient.MaritalStatus.Code)	source.MARITAL
7 set		target.(Patient.Race.Code)	source.RACE
8 set		target.(Patient.EthnicGroup.Code)	source.ETHNICITY
9 set		target.(Patient.Gender.Code)	source.GENDER
10 set		target.(Patient.BirthPlace.Street)	source.BIRTHPLACE
11 set		target.(Patient.Addresses(1).Street)	source.ADDRESS
12 set		target.(Patient.Addresses(1).City.Code)	source.CITY
13 set		target.(Patient.Addresses(1).State.Code)	source.STATE
14 set		target.(Patient.Addresses(1).County.Code)	source.COUNTY
15 set		target.(Patient.Addresses(1).Zip.Code)	source.ZIP

Note: for properties with (), you need to set the index because those properties can have more than one item. In this sample we have only one address, so target.{Patient.Addresses(1)...} is configured with 1.

14. Click the Save button and Compile button.



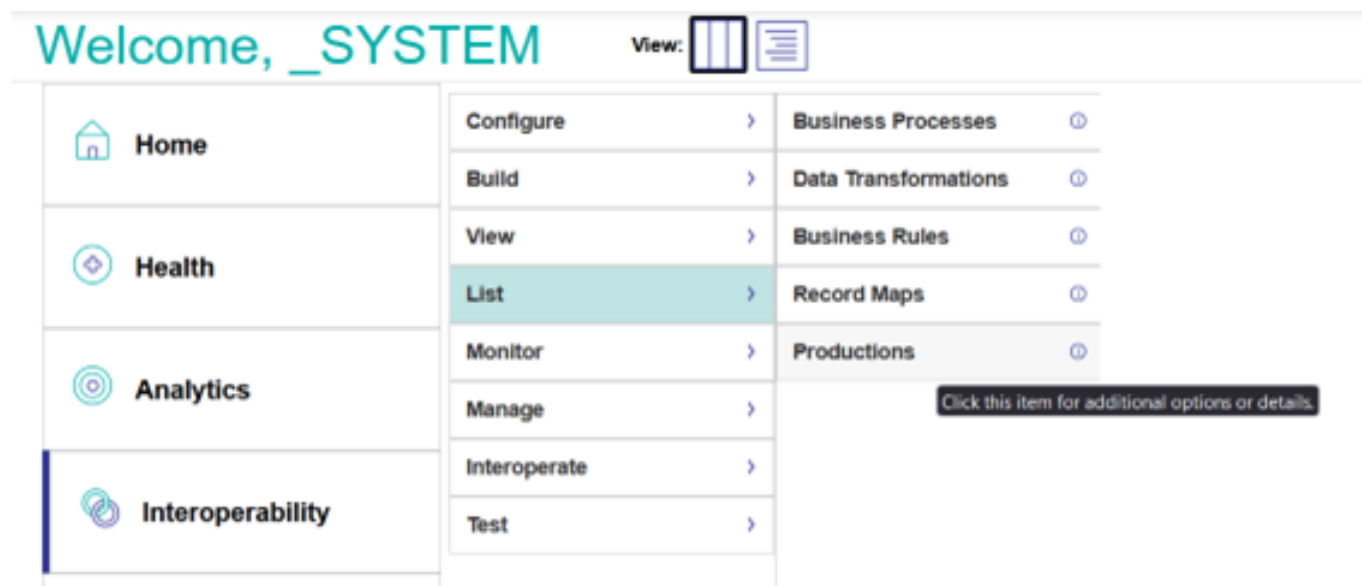
15. Finally, click the Interoperability shortcut to go to the Interoperability menu:

Interoperability > Data Transformation Builder - (customsda.PatientDTL2)

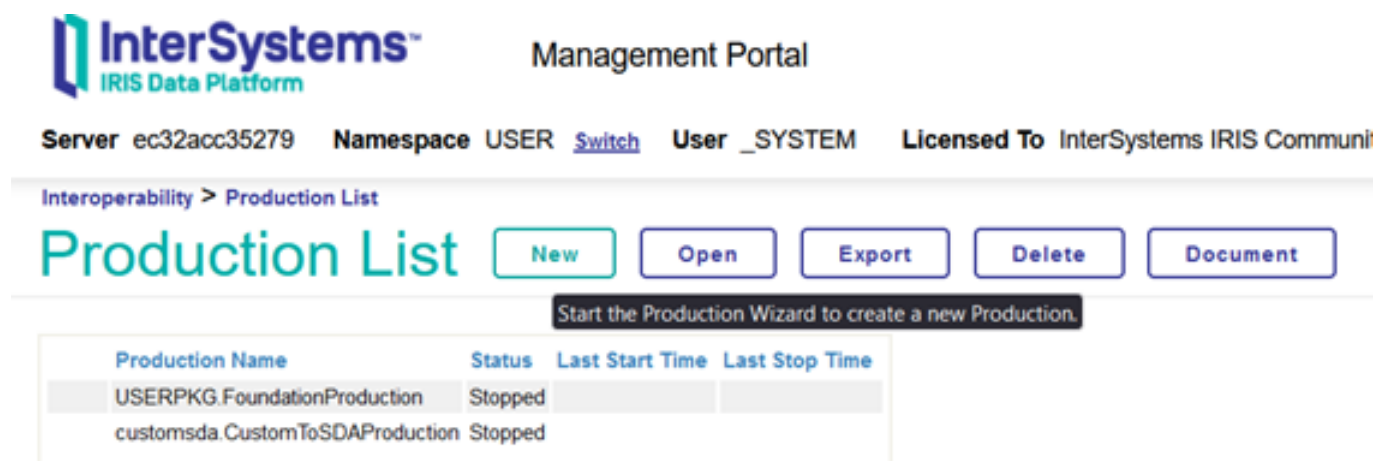
Create the Interoperability Production, the last artifact to complete our work!

Productions are the mechanisms used to effectively automate integration flows. It ' s time to create our output to transform the patient data in the patients2.csv file to SDA.

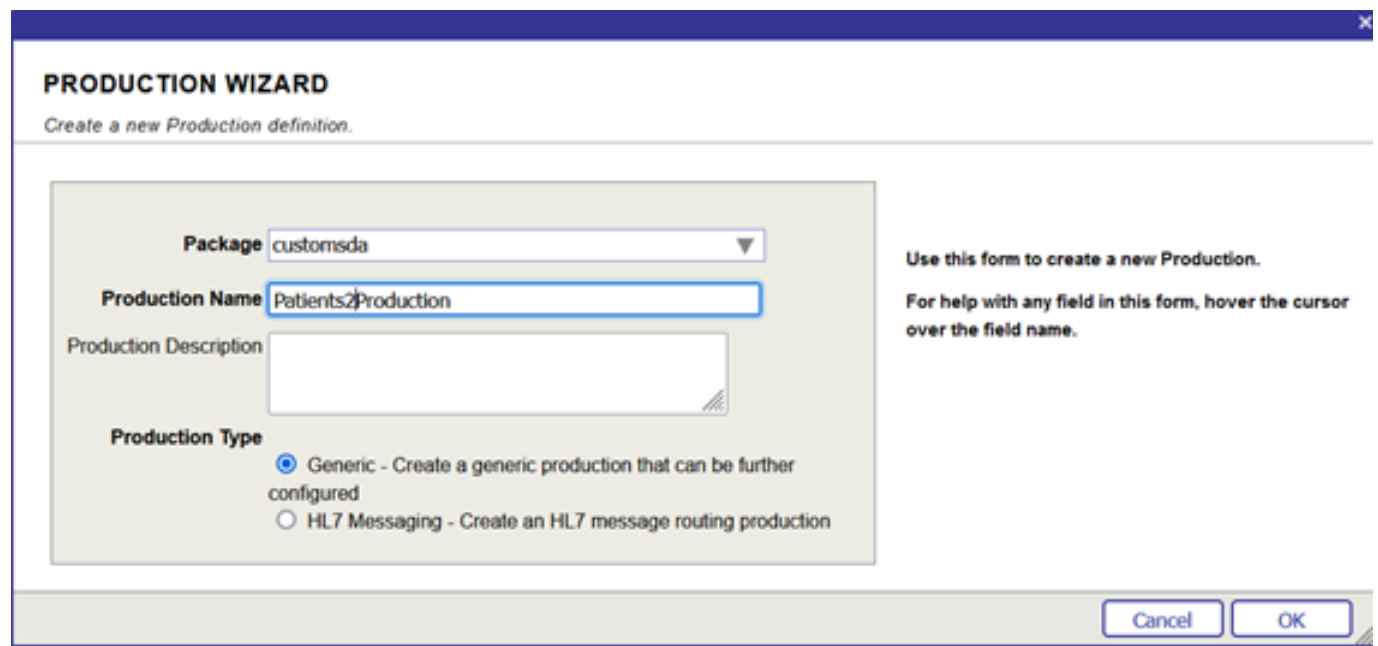
1. Click Interoperability > List > Productions:



2. Click the New button:



3. Set customsda to Package, Patients2Production to Production Name, and Production Type to Generic. Click Ok:



PRODUCTION WIZARD
Create a new Production definition.

Package: customsda

Production Name: Patients2Production

Production Description:

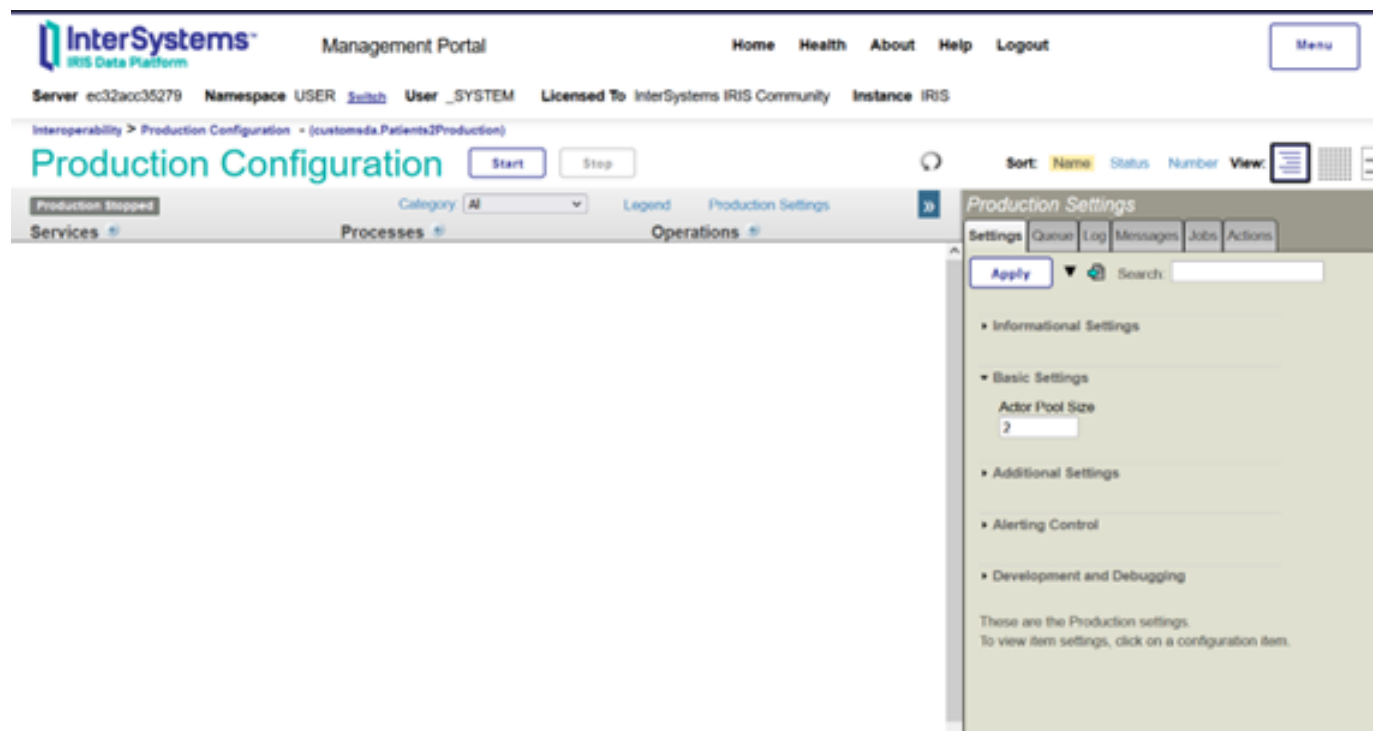
Production Type:

- ☒ Generic - Create a generic production that can be further configured
- ☐ HL7 Messaging - Create an HL7 message routing production

Use this form to create a new Production.
For help with any field in this form, hover the cursor over the field name.

Cancel OK

4. Now we have the Production Configuration:



InterSystems®
IRIS Data Platform

Management Portal

Home Health About Help Logout

Menu

Server: ec32acc35279 Namespace: USER Switch User: _SYSTEM Licensed To: InterSystems IRIS Community Instance: IRIS

Interoperability > Production Configuration - (customsda.Patients2Production)

Production Configuration Start Stop

Production Stopped

Category: All Legend Production Settings

Services Processes Operations

Sort: Name Status Number View:

Production Settings

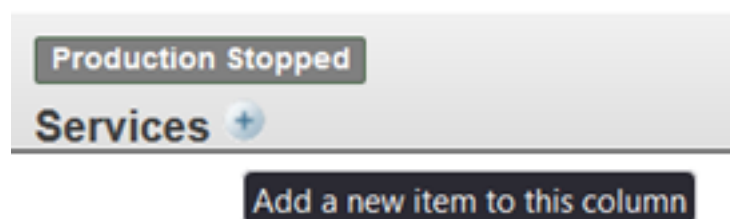
Settings Queue Log Messages Jobs Actions

Apply Search:

- Informational Settings
- Basic Settings
 - Actor Pool Size: 2
- Additional Settings
- Alerting Control
- Development and Debugging

These are the Production settings.
To view item settings, click on a configuration item.

5. Click the Plus button near Services:



Production Stopped

Services +

Add a new item to this column

Note: Services are the components used by productions to get source data.

6. Configure the Service Class with the value EnsLib.RecordMap.Service.FileService, and Service Name as

PatientCSVService, and check Enable Now:

The screenshot shows the 'BUSINESS SERVICE WIZARD' dialog box with the title 'Add a new Business Service to this Production.' It has four tabs: 'All Services', 'HL7 Input', 'X12 Input', and 'Business Metric'. The 'All Services' tab is selected. Inside the form, 'Service Class' is set to 'EnsLib.RecordMap.Service.FileService', 'Service Name' is 'PatientCSVService', 'Display Category' is empty, and 'Comment' is empty. The 'Enable Now' checkbox is checked. On the right, there is instructional text: 'Use this form to add a new business service to the production. For help with any setting in this form, hover the cursor over the setting name.' At the bottom right are 'Cancel' and 'OK' buttons.

7. Select PatientCSVService and configure the Settings tab with

the values and click the Apply button:

File Path: /opt/user/data/

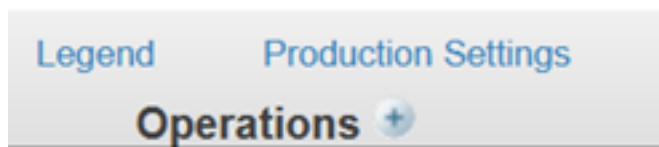
File Spec: patients2.csv

RecordMap: Patients2RecordMap

Target Config Names: PatientProcess

The screenshot shows the 'InterSystems Management Portal' interface. The 'Production Configuration' section is active, showing a list of services with 'PatientCSVService' selected. On the right, the 'Settings' tab for 'PatientCSVService' is open. It shows various configuration fields: 'Enabled' (checked), 'External Registry ID' (empty), 'File Path' (/opt/user/data/), 'File Spec' (patients2.csv), 'Archive Path' (empty), 'Work Path' (empty), 'Call Interval' (5), 'RecordMap' (Patients2RecordMap), and 'Target Config Names' (PatientProcess). An 'Apply' button is at the top left of the settings panel.

8. Click the Plus button located near Operations:



Note: Operations are production components used to write/persist data on a target (databases, systems, API, Web Services, FTP, File, etc.)

9. Configure the Operation with these values:
Operation Class: EnsLib.EDI.XML.Operation.FileOperation
Operation Name: PatientSDAOperation
Check Enable Now

A screenshot of a 'BUSINESS OPERATION WIZARD' dialog box. The title bar is dark blue with a close button. The main area has a light gray background. At the top, there's a subtitle 'Add a new Business Operation to this Production.' Below it are four tabs: 'All Operations' (selected), 'HL7 Output', 'X12 Output', and 'Workflow'. The main form area contains several fields: 'Operation Class' is a dropdown menu showing 'EnsLib.EDI.XML.Operation.FileOperation'; 'Operation Name' is a text box with 'PatientSDAOperation'; 'Display Category' is a dropdown menu; 'Comment' is a text area; and 'Enable Now' is a checkbox that is checked. To the right of the form is a help text area that says 'Use this form to add a new business operation to the production. For help with any setting in this form, hover the cursor over the setting name.' At the bottom right are 'Cancel' and 'OK' buttons.

10. Select the PatientSDAOperation and set the value /opt/user/data/ to the File Path and click the Apply button:

The screenshot shows the 'PatientSDAOperation' settings window. On the left, a sidebar contains 'Legend' and 'Production Settings' tabs. Under 'Production Settings', there is a section for 'Operations' with a plus icon, and a list item 'PatientSDAOperation' with a green circle icon. The main window has a title bar 'PatientSDAOperation' and a tabbed interface with 'Settings', 'Queue', 'Log', 'Messages', 'Jobs', and 'Actions'. The 'Settings' tab is active, showing an 'Apply' button, a search bar, and two expandable sections: 'Informational Settings' and 'Basic Settings'. The 'Basic Settings' section is expanded, showing 'Enabled' with a checked checkbox, 'External Registry ID' with a dropdown menu, 'File Path' with a text field containing '/opt/user/data/' and a search icon, and 'Format' with a text field.

11. Click the Plus button next to Processes:

The screenshot shows the 'Processes' section in the InterSystems Developer interface. It features a header bar with the word 'Processes' and a plus icon. Below the header, there is a button labeled 'Add a new item to this column'.

Note: Processes are the production component to coordinate the data flow.

12. Configure the Business Process with these values:
- Business Process Class: EnsLib.MsgRouter.RoutingEngine
 - Routing Rule Name: customsda.PatientRouterRule2
 - Business Process Name: PatientProcess
 - Check Enable Now

BUSINESS PROCESS WIZARD
Add a new Business Process to this Production.

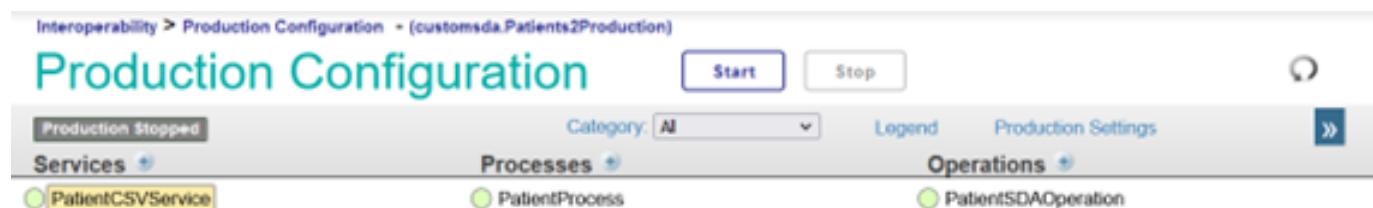
☒ All Processes
 ☐ HL7 Router
 ☐ X12 Router
 ☐ Component

Business Process Class: EnsLib.MsgRouter.RoutingEngine
 Auto-Create Rule: ☐
 Routing Rule Name: customsda.PatientRouterRule2
 Business Process Name: PatientProcess
 Display Category:
 Comment:
 Enable Now: ☒
 Pool Size: 1

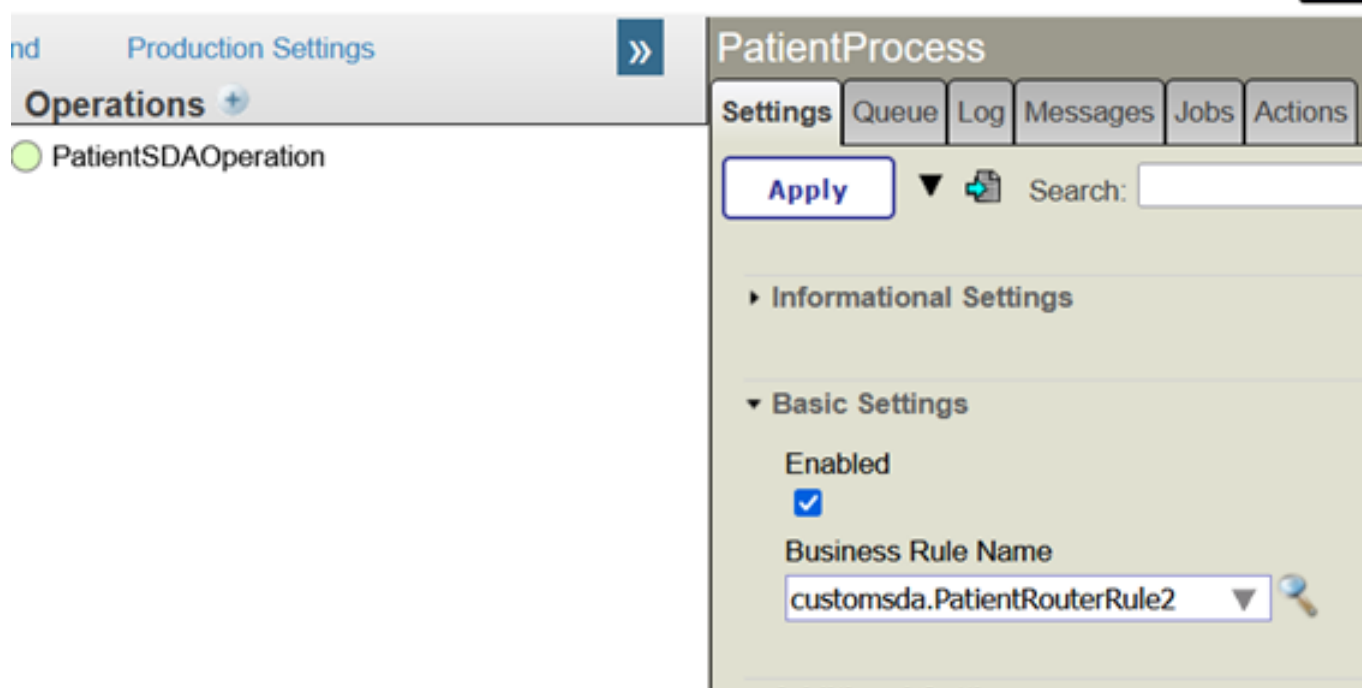
Use this form to add a new business process to the production.
 For help with any setting in this form, hover the cursor over the setting name.

Cancel OK

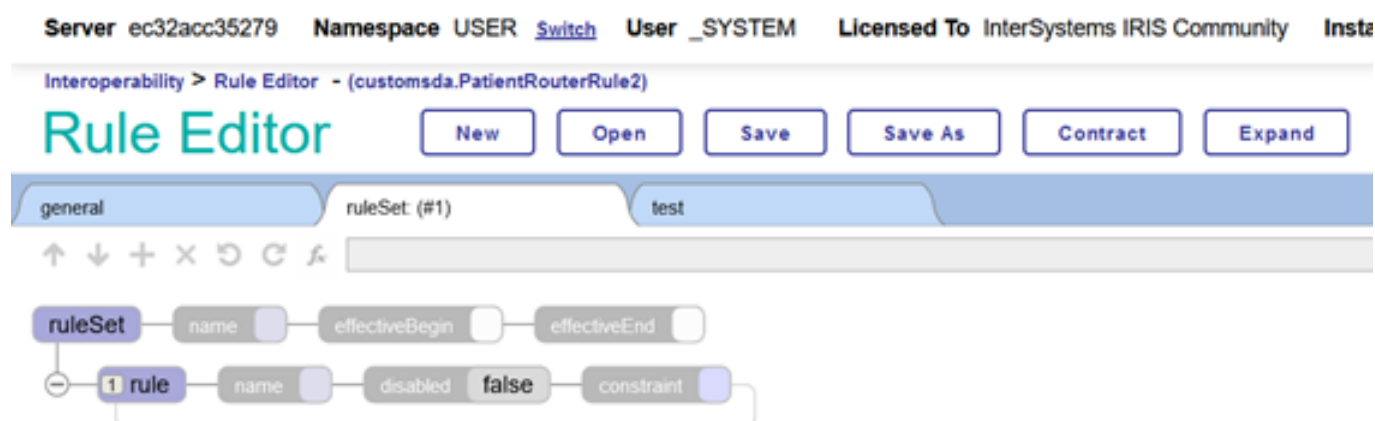
13. So far, we have got all components created:



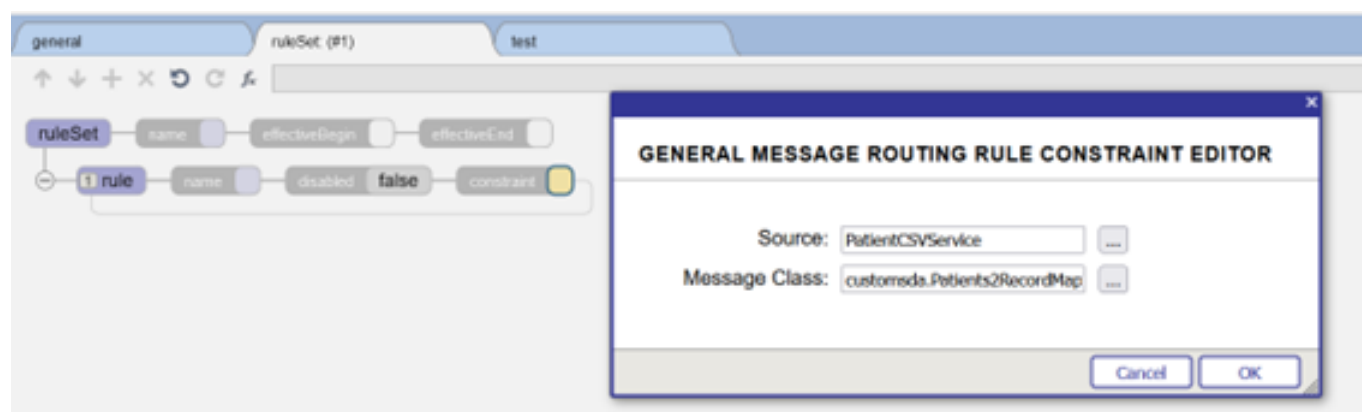
14. Select PatientProcess and go to Settings tab > Magnifier icon near Business Rule Name:



15. Now we will configure the Routing Rule into Rule Editor:



16. Do a double-click on the constraint component and configure Source with PatientCSVService and Message Class with customsda.Patients2RecordMap.Record:



17. Currently, we have source and target configured:



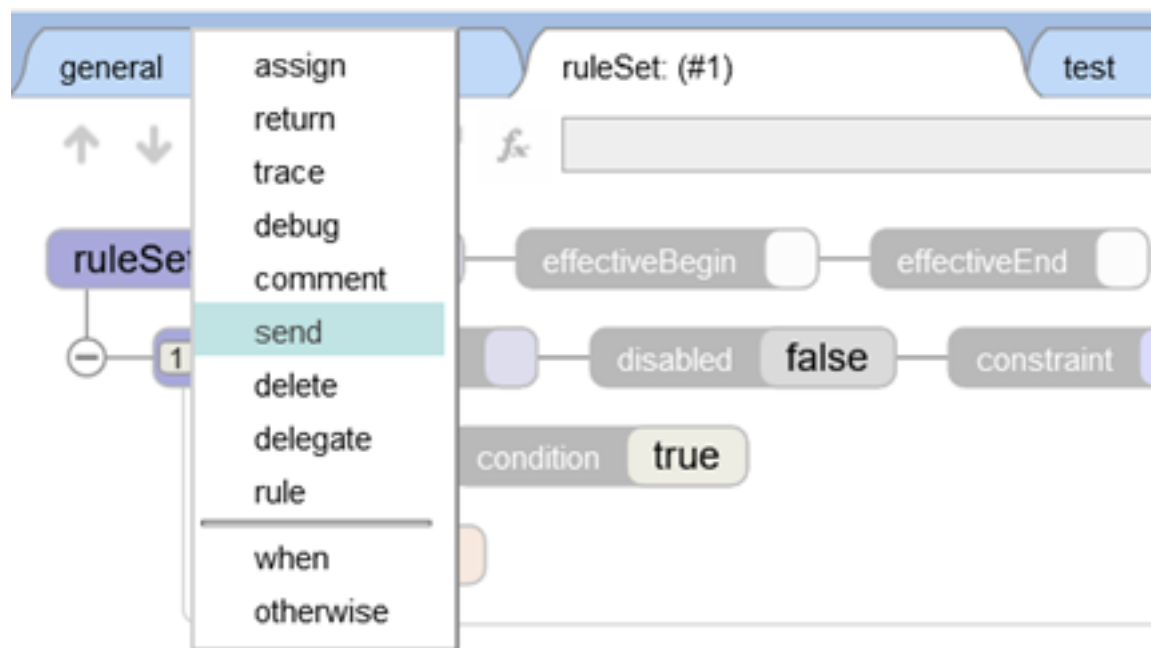
18. Select rule component and click the Green Plus button:



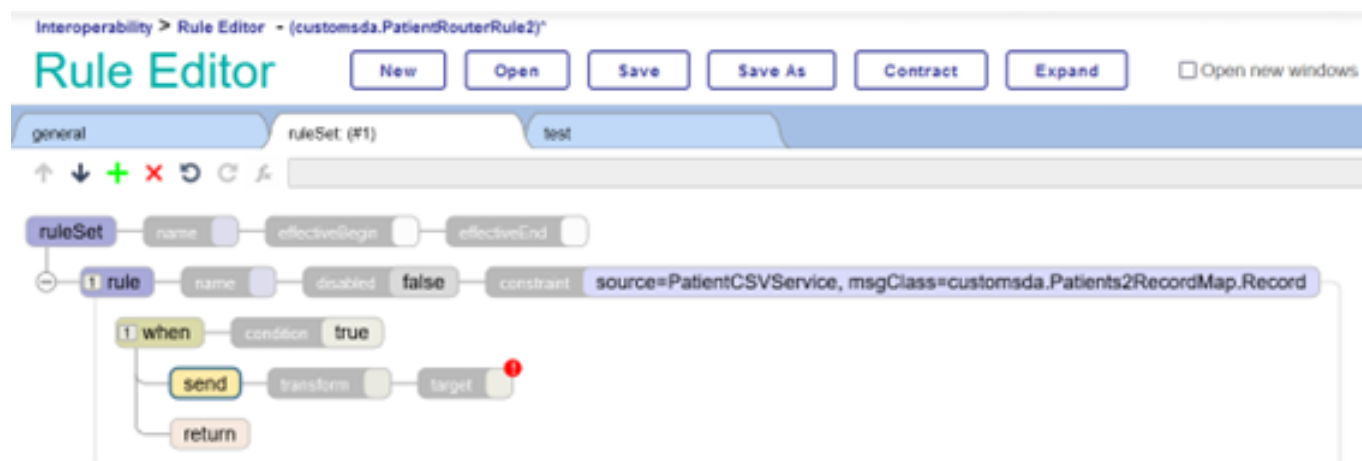
19. Select Send to create a When component:



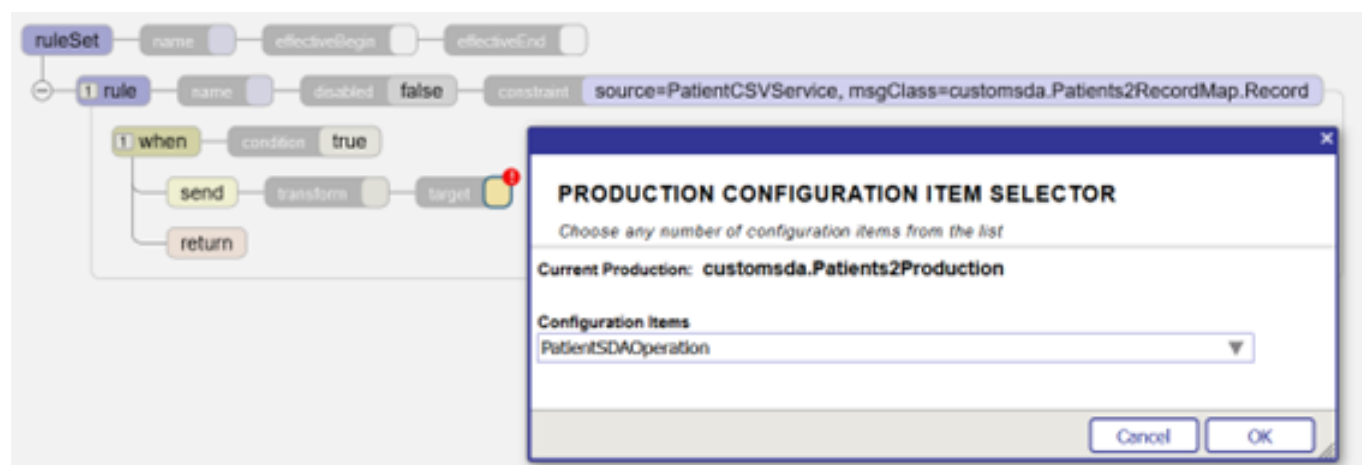
20. Select When component, click the Green Plus button, and select Send component:



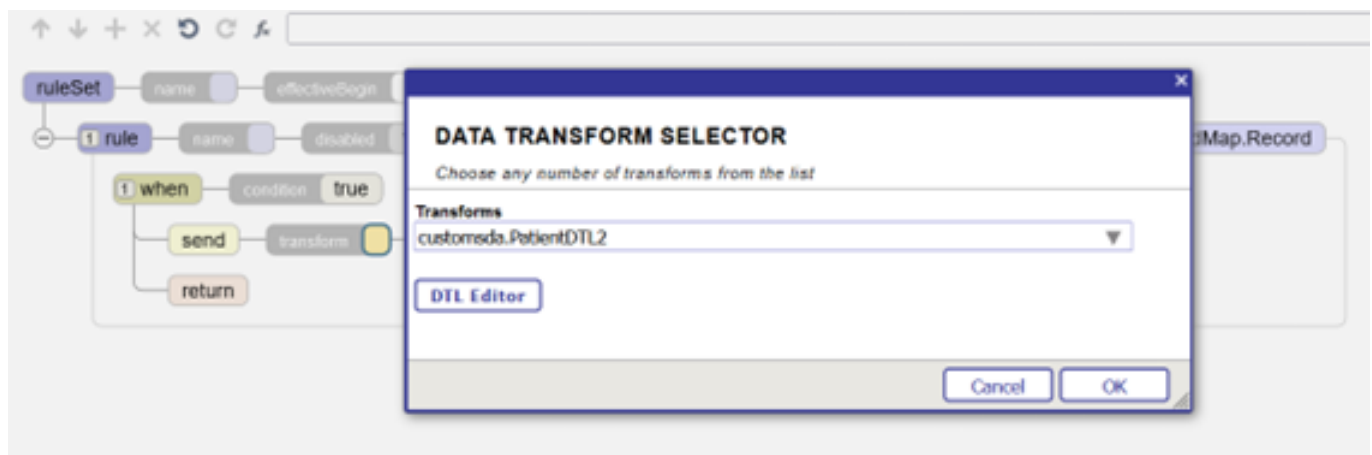
21. You are supposed to see the picture displayed below on your screen:



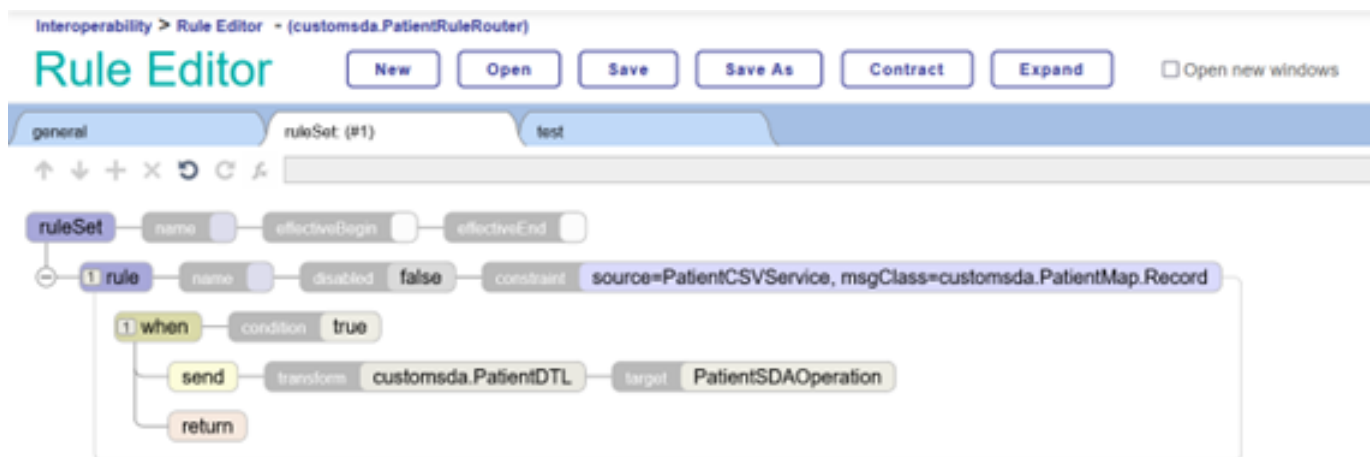
22. Do a double-click on the target component and set Configuration Items to PatientSDAOperation, then click the OK button:



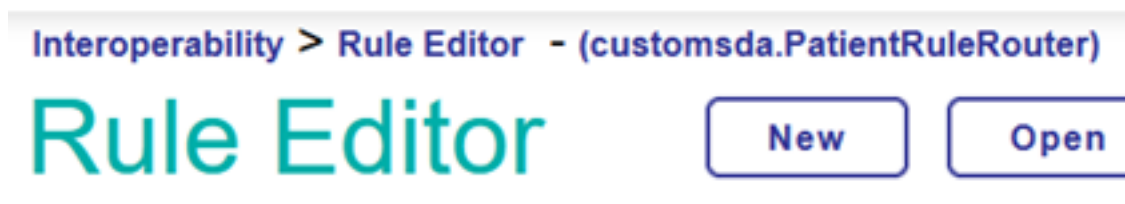
23. Do a double-click on the transform component and set Transforms to customsd.PatientDTL2, after that click OK:



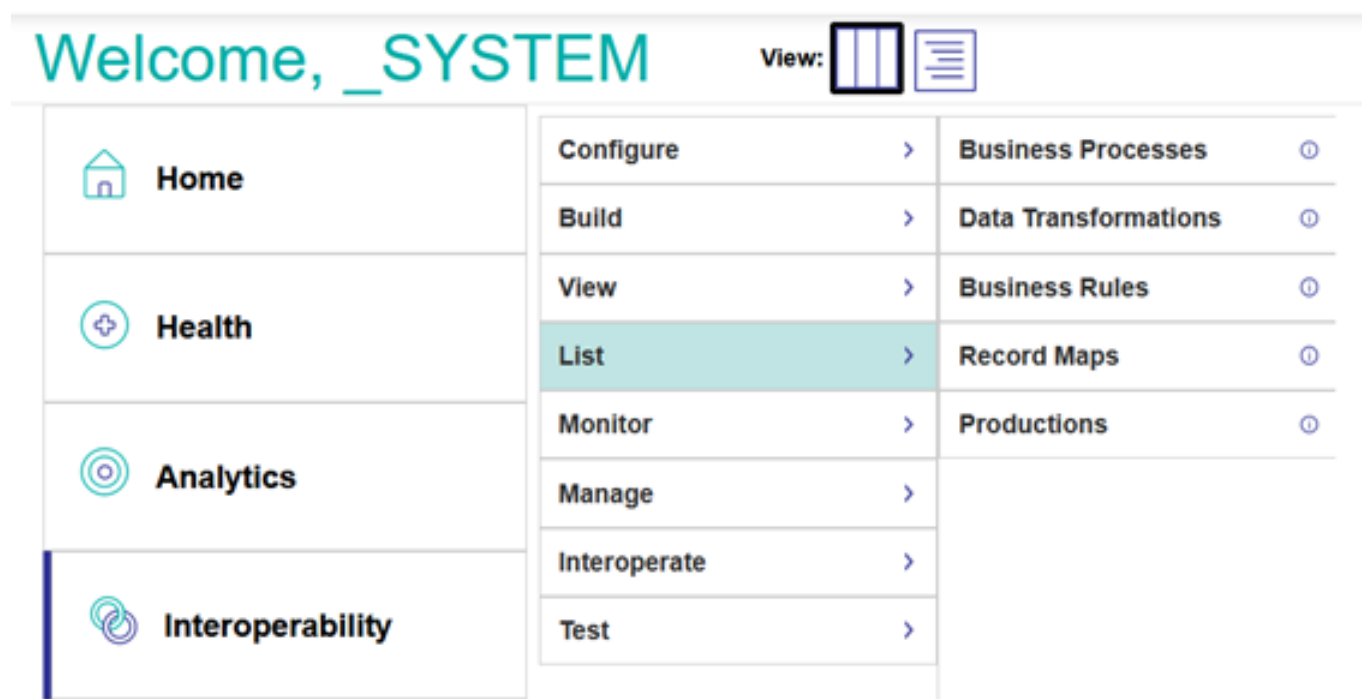
24. Now, you have your Rule definitions ready:



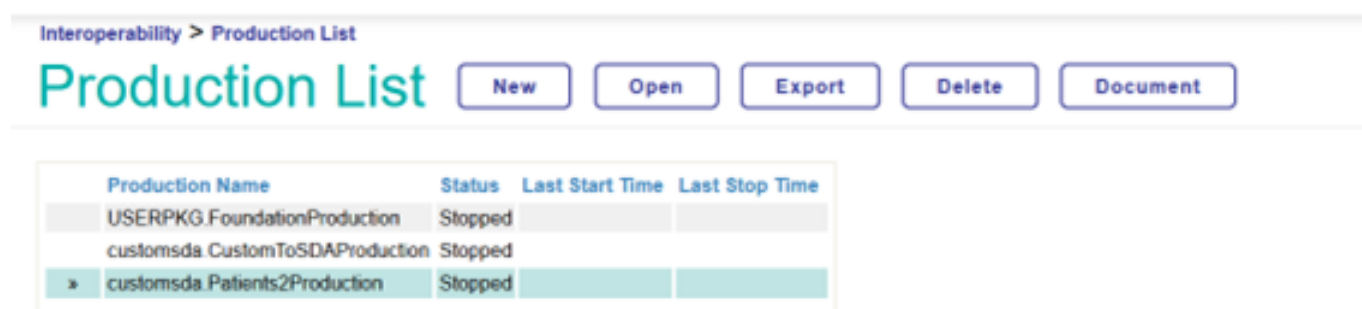
25. Click the Save button and Go to the Interoperability menu:



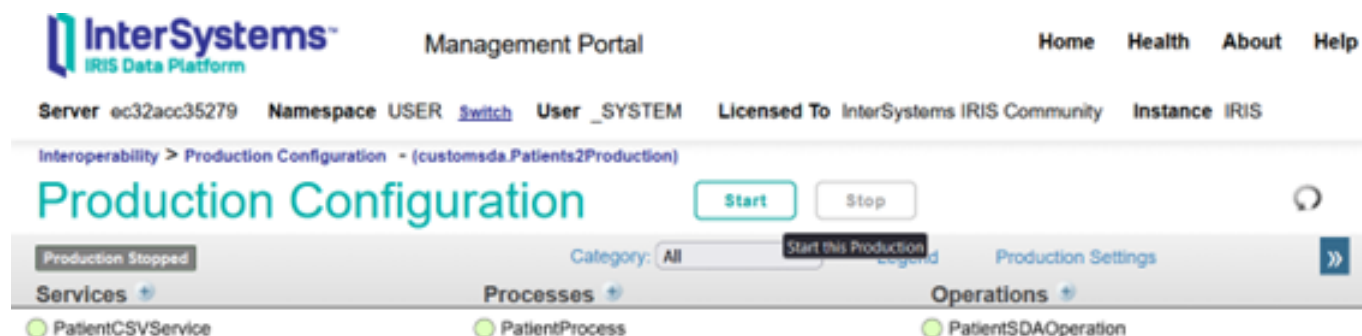
26. Go to Interoperability > List > Productions:



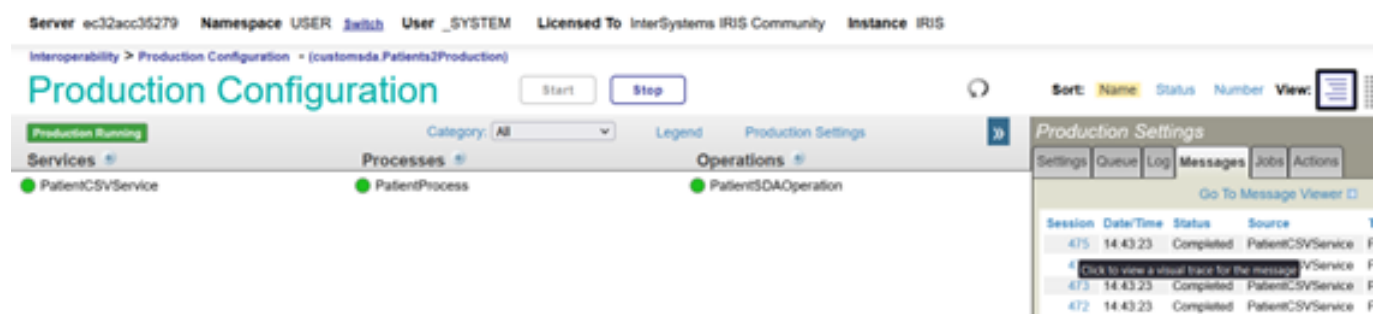
27. Select Patients2Production and click the Open button:



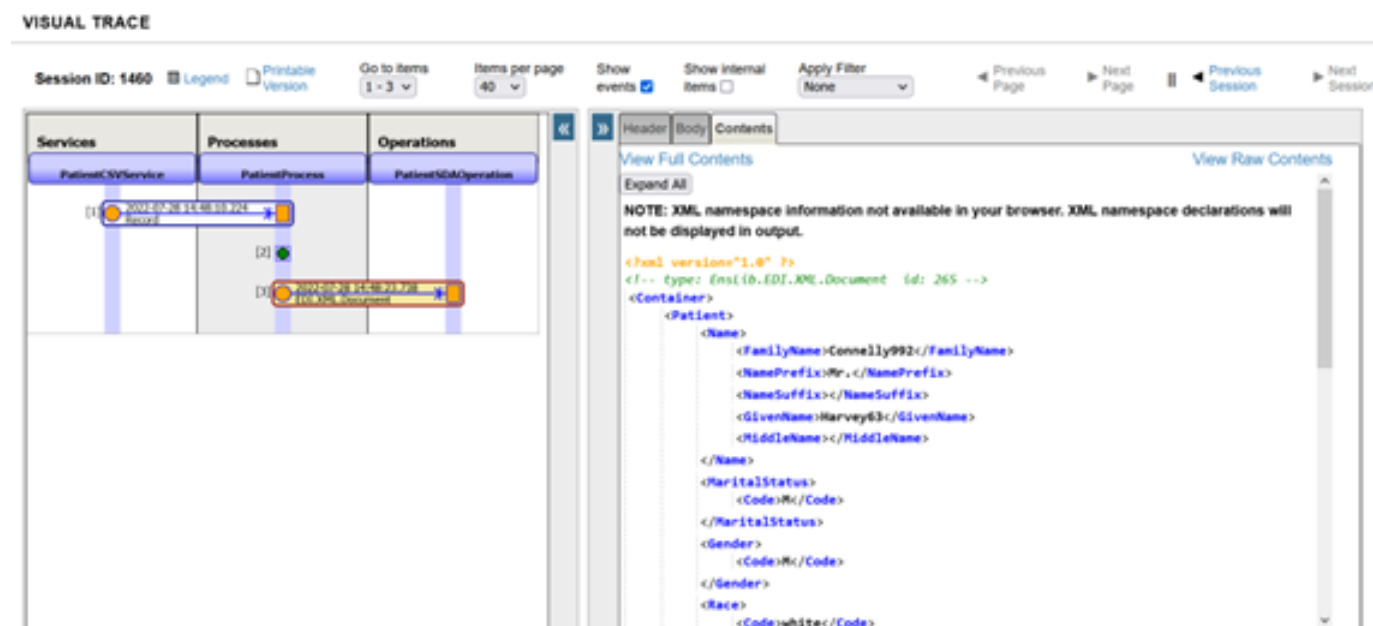
28. We will run our new Production! Click the Start button:



29. Select PatientProcess and Go to the Messages tab to see the results (messages):



30. Click on a message to see a Sequence Diagram with the transformation results:



As you can see, it is an easy and fully visual drag-and-drop process, which transforms custom messages to SDA or other formats. Learn more about it by checking these links:

1. Building Basic FHIR Integrations with InterSystems IRIS for Health:
<https://learning.intersystems.com/course/view.php?id=1959&ssoPass=1>
2. Learn HealthShare for Developers and System Integrators:
<https://learning.intersystems.com/course/view.php?id=26&ssoPass=1>
3. Building Business Integrations with InterSystems IRIS:
<https://learning.intersystems.com/course/view.php?id=1437&ssoPass=1>
4. Building Basic HL7 Integrations with InterSystems:
<https://learning.intersystems.com/course/view.php?id=1350&ssoPass=1>

[#Business Operation](#) [#Business Service](#) [#DTL](#) [#Interoperability](#) [#InterSystems IRIS for Health](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/transform-custom-healthcare-messages-sda>