

Article

[Jin Kim](#) · Mar 19, 2020 10m read[Open Exchange](#)

## Developing Operational Analytics Dashboards

Hi Developers and Interface Engineers!

I'd like to share with you how you can help your organization today obtain a better understanding of key message processing metrics (i.e. average message processing times, number of inbound messages, number of outbound messages, etc.)! Given the embedded IRIS database powering integration, you already have all the data you need -- you just need to put the data to use and present them in a user-friendly format!

Let me walk you through a sample implementation of an operational analytics dashboard that has served as a template for many organizations using these dashboards in production today. This particular sample dashboard will include the following metrics:

Namespace	Year	AvgAckTime	AvgTime	MaxTime	MinTime	NumInbound	NumOutbound
FHIRREPO	2019	0.027	0.214	0.396	0.012	22	22
SANDBOX	2019	23.781	48.071	3,621.146	11.125	38549	3884

You can find the accompanying sample code [here on Open Exchange](#). Just as a note, I used IRIS for Health 2019.1, but this can be done on Ensemble, Health Connect, and HealthShare.

Now, let's dive in!

### 1) Create a New Namespace: “ REPORTS ”

In this section, we will create a new namespace called “ REPORTS,” where we will be compiling all the messages from other namespaces of interest to compute the message processing metrics.

1. From the main screen of the Management Portal, click System Administration à Configuration à System Configuration à Namespaces
2. To create a new namespace, click the Create New Namespace button on the top left corner of the page.
3. Fill in the following information to configure the namespace:
  - a. Name of the namespace: REPORTS
  - b. The default database for Globals: Local Database
  - c. Select an existing database for Globals: Select Create New Database

This will launch the Database Wizard, which is used to create a new database

- i. Enter the name of your database: REPORTSGLOBALS
- ii. Database directory: C:\InterSystems\HealthShare\mgr\REPORTSGLOBALS

A warning will appear saying that the directory does not exist. This is okay.

- iii. Click Next. Click Finish.
- d. The default database for Routines: Local Database
- e. Select an existing database for Routines: Select Create New Database

This will launch the Database Wizard, which is used to create a new database

- i. Enter the name of your database: REPORTSROUTINES
- ii. Database directory: C:\InterSystems\HealthShare\mgr\REPORTSROUTINES

A warning will appear saying that the directory does not exist. This is okay.

- iii. Click Next. Click Finish.
  - f. Click Save. Wait for the namespace to be created – a new page will show up.
4. Scroll to the bottom of the new Enable Ensemble Namespace page and click Close.

To edit the settings of a namespace, click the namespace on the Namespaces page and edit the desired properties. Then click Save in the top left corner of the page.

## 2) Import Code via Studio

In this section, we will use Studio to code and create necessary class files to fetch messages from all namespaces specified in the ^ReportNamespaces global and store the computed metrics in the Demo.OperationalAnalytics.MetricsTable class.

1. Open Studio. Ensure you are in the REPORTS namespace. To switch namespaces, go to File à Change Namespace and select the REPORTS namespace.
2. Click File à New. Under Categories, select General. Under Templates, select Class Definition and click OK.
  - a. In the New Class Wizard, specify the following:
    - i. Enter a package name: Demo.OperationalAnalytics
    - ii. Enter a class name: Compute
    - iii. Click “ Next ”
    - iv. For Class Type, press “ Extends ” and enter %SYS.Task.Definition for the Name of super class. This is to enable scheduling via Task Manager.
    - v. Click “ Finish ”
3. Within the curly braces of the Demo.OperationalAnalytics.Compute class, copy-paste the following:
  - Retrieve the code from [Open Exchange](#)
  - Wait until we create the actual table to compile the class we just created.
4. Click File à New. Under Categories, select General. Under Templates, select Class Definition and click OK.
  - a. In the New Class Wizard, specify the following:
    - i. Enter a package name: Demo.OperationalAnalytics
    - ii. Enter a class name: MetricsTable
    - iii. Click “ Next ”
    - iv. For Class Type, select “ Persistent ”
    - v. Click “ Finish ”
5. Within the curly braces of the Demo.OperationalAnalytics.MetricsTable class, copy-paste the following (the code will also be available on Open Exchange):

```
Property RunTime As %TimeStamp;  
Property Namespace As %String;  
Property AvgTime As %Float;  
Property AvgAckTime As %Float;  
Property NumInbound As %Integer;  
Property NumOutbound As %Integer;  
Property MinTime As %Float;  
Property MaxTime As %Float;
```

6. Compile both class files. On the top toolbar, click Build à Compile to compile the class.
7. In the Output console at the bottom of Studio, store the names of all namespaces into the ^ReportNamespaces global.
  - a. For example:
    - i. set ^ReportNamespaces(1) = " SANDBOX "
    - ii. set ^ReportNamespaces(2) = " FHIRSERVER "
    - iii. Repeat until you have the names of all namespaces stored in ^ReportNamespaces. You can check which namespaces have been stored by entering " zwrite ^ReportNamespaces "

### 3) Add Appropriate Global/Package Mappings to Namespaces of Interest

In this section, we will add the necessary global and package mappings to all namespaces that will be reporting to the REPORTS namespace.

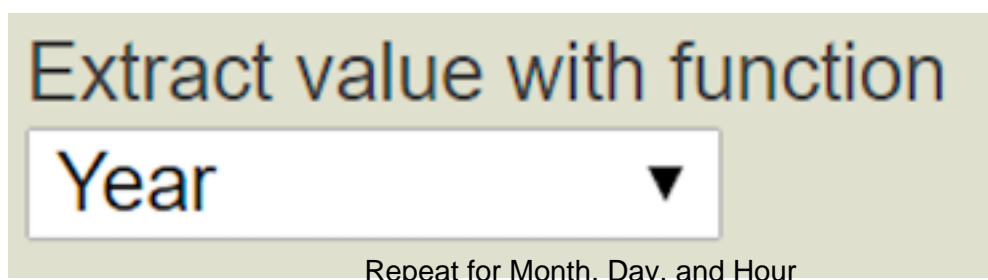
1. From the main screen of the Management Portal, click System Administration à Configuration à System Configuration à Namespaces
2. For every namespace you would like to include in the final table,
  - a. Click on Global Mappings, and then click " New. " Fill in the following information:
    - i. Global Database Location: REPORTSGLOBALS
    - ii. Global Name: (copy paste from <DataLocation> tag in Demo.OperationalAnalytics.MetricsTable file in Studio – see below)
      - i. Do not include the " ^ "
    - iii. Press " OK " to close the pop-up box, and then press " Save Changes. "
  - b. Click on Package Mappings, and then click " New. " Fill in the following information:
    - i. Package Database Location: REPORTSROUTINES
    - ii. Package Name: Demo.OperationalAnalytics
    - iii. Press " OK " to close the pop-up box, and then press " Save Changes. "

```
<DataLocation>Demo.Operatio43E6.MetricsTableD</DataLocation>
<DefaultData>MetricsTableDefaultData</DefaultData>
<IdLocation>^Demo.Operatio43E6.MetricsTableD</IdLocation>
<IndexLocation>^Demo.Operatio43E6.MetricsTableI</IndexLocation>
<StreamLocation>^Demo.Operatio43E6.MetricsTableS</StreamLocation>
<Type>%Storage.Persistent</Type>
```

## 4) Set up Cube and Pivot Table on Analytics

In this section, we will build a Cube on Analytics Architect to organize the data from Demo.OperationalAnalytics.Compute class.

1. From the main screen of the Management Portal, click Analytics àArchitect. Ensure you are in the REPORTS namespace.
2. Click “ New ” and fill in the following information:
  - a. Definition Type: Cube
  - b. Cube Name: Operational Analytics
  - c. Cube Source: Class
  - d. Source Class: Demo.OperationalAnalytics.MetricsTable
    - i. Use the “ Browse... ” button to select the class
  - e. Class Name for the Cube: Leave as is.
  - f. Hit “ OK ” on the bottom right.
3. From Source Class on the left-hand column, drag and drop Namespace onto the Dimensions category under the “ Operational Analytics ” .
  - a. To create a roll-up for the date/time (year, month, day, hour):
    - i. Click “ Add Element ” next to “ Model Elements ” at the top.
    - ii. In the pop-up window, enter the following:
      - i. Enter New Element Name: DateRollUp
      - ii. Select Time Dimension
      - iii. Click “ OK ”
    - iii. Click on “ DateRollUp ” under Dimensions and select “ Property ” as “ RunTime ” in the Details pane. Then, click on the “ H1 ” listed under “ DateRollUp ” , and click “ Add Element ” at the top again.
      - i. Enter New Element Name: Year
      - ii. Select Level
      - iii. Click “ OK ”
      - iv. Click the new level that was just created and select the appropriate “ Extract value with function ” on the right-hand side under “ Details ” . See table below for corresponding functions.



Element Name	Year	Month	Day	Hour
Function	Year	MonthYear	DayMonthYear	HourNumber

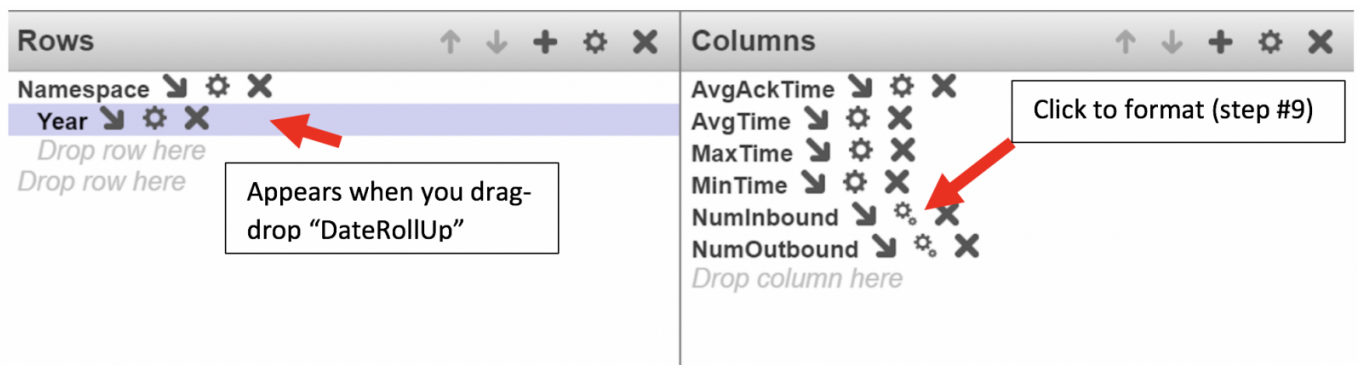
4. Drag and drop the remaining metrics except %ID (AvgAckTime, AvgTime, MaxTime, MinTime, NumInbound, and NumOutbound) onto the Measures category.
  - a. The following only applies to the time metrics (AvgAckTime, AvgTime, MaxTime, MinTime). Leave NumInbound and NumOutbound as is:

Under Details on the right-hand side, set the “ Precision ” to 3, as shown below.

5. Press “ Compile ” and hit “ Done ” once finished. The Cube should look like the following:

» Operational Analytics	Element Type	Details	
▼ Measures			
AvgAckTime	number measure	SUM AvgAckTime	×
AvgTime	number measure	SUM AvgTime	×
MaxTime	number measure	SUM MaxTime	×
MinTime	number measure	SUM MinTime	×
NumInbound	number measure	SUM NumInbound	×
NumOutbound	number measure	SUM NumOutbound	×
▼ Dimensions			
▼ Namespace			
H1	data dimension		×
Namespace	hierarchy		×
	level 1	Namespace	×
▼ DateRollUp			
H1	time dimension	Run Time	×
	hierarchy		×
Year	level 1	Year	×
Month	level 2	MonthYear	×
Day	level 3	DayMonthYear	×
Hour	level 4	HourNumber	×

6. Back in the Management Portal, go to Analytics àAnalyzer. In the Analyzer, press the cube on the top left and select the “ Operational Analytics ” cube we just created.
  - a. The Analyzer may already load the “ Operational Analytics ” cube. Move onto step 7 if so
7. Drag and drop all the measures (except “ Count ” ) under Columns. As for the dimensions, drag and drop them onto rows in the following manner:



After dragging Namespace into Rows, drag DateRollUp onto the diagonal arrow of Namespace.

8. To format the numbers that will show up in the table, click on the “ wheel ” symbol next to each measure you added in the column (see image above).
9. When the “ Level Options ” window pops out, press “ Format and Style ” to expand the section.
  - a. Enter “ # ” in the Format section for NumInbound and NumOutbound.
  - b. Hit “ OK ” to exit.





**Level Options**  
Change options for a Pivot Table level.

[Measures].[AvgTime]

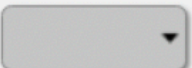


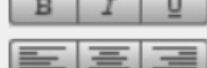

**Show**  
☒ Measure

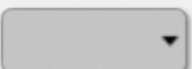


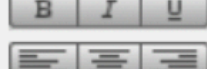

**Caption**  
  
For example, Year \*

**Format and Style**

**Format**  
    
For example, #,###.#

**Total Override**  
  
Override aggregate used for pivot total

**Cell Style**  
    

**Header Style**  
    

Modify the measure in this axis

**Measure Aggregate**  
  
Override the default aggregate applied to this measure

**Cancel** **OK**

10. Press “ Save ” at the top, and fill in the following information for “ Save Pivot ” :
  - a. Folder: Demo (up to user discretion)
  - b. Pivot Name: Operational Analytics Pivot
  - c. Press “ OK ”

## 5) Create a Dashboard on Analytics – User Portal

In this section, we will set up a dashboard on Analytics User Portal to display the resulting Operational Analytics pivot table.

1. From the main screen of the Management Portal, click Analytics à User Portal. Ensure you are in the REPORTS namespace.
2. Press the large “ + ” sign at the top, and then press “ Add Dashboard ” under “ Add New Item ” on the left-hand column.
  - a. Fill in the following information to create a new dashboard.
    - i. Folder: Demo (up to user discretion)
    - ii. Dashboard Name: Operational Analytics Dashboard
    - iii. Dashboard Title: Operational Analytics Dashboard
    - iv. Press “ OK ” at the bottom right.
3. An empty dashboard will have appeared on the page. Click the right-hand arrow on the left-side of the page. In the Dashboard Editor that will slide out, click “ Widgets ” and then the “ + ” sign to add a widget.
4. In the Widget Wizard, click on “ Pivots and Charts ” on the left-hand column, and then select “ Table ”. See example image below.
  - a. Data source: Demo/Operational Analytics Pivot.pivot (use the magnifying glass button to navigate)
  - b. Widget Name: Operational Analytics Widget
  - c. Press “ OK ” on the bottom right to exit the wizard.

The screenshot shows the 'Widget Wizard' window with the following configuration:

- Step 1:** Select the type of widget you wish to create using the menu on the left. The 'Pivots and Charts' category is selected, and 'Table' is chosen.
- Step 2:** Select a data source. The data source is 'Demo/Operational Analytics Pivot.pivot'. The 'Link to' dropdown is set to 'Use same data source as another widget on the dashboard'.
- Step 3:** Give this widget a unique name to identify it. The 'Widget Name' is 'Operational Analytics V'.

Buttons for 'Cancel' and 'OK' are visible at the bottom right.

The resulting widget will display the data stored in the Demo.OperationalAnalytics.MetricsTable. As of right now, there is no data stored and you will see an empty table.

By adding a filter on Namespace or Year/Month/Day/Hour, we can focus on particular dates and exclude “ null ” data.

1. Click the right-hand arrow on the left-side of the page. In the Dashboard Editor that will slide out, click “ Widgets ” and then select the “ Operational Analytics Widget ” widget.
2. Click on “ Controls ” and then press the “ + ” sign to prompt the Control Wizard



- a. Target: \*
  - b. Action: Apply Filter
  - c. Filter: (put in Namespace, Month, etc.)
  - d. Press “ OK ” to exit the wizard.
3. Press “ Save ” on the top to preserve the newly added filter.
  4. More filters can be added by following the above procedure.

## 6) Schedule to Run Hourly using Task Manager

In this section, we will use the Task Manager to schedule the Operational Analytics Dashboard to refresh every hour to evaluate the messages from the preceding hour.

1. From the main screen of the Management Portal, ensure you are in the REPORTS namespace.  
To navigate to Task Manager, click System Operation à Task Manager à New Task
2. Fill out the Task Scheduler Wizard:
  - a. Task name: Refresh Operational Analytics Dashboard
  - b. Namespace to run task in: REPORTS
  - c. Task type: Demo.OperationalAnalytics.Compute
  - d. Click “ Next> ” on the bottom of the screen.
  - e. How often do you want the Task Manager to execute this task? Daily
    - i. Every “ 1 ” day(s)
    - ii. Specify Start Date (End Date is not necessary)
    - iii. Select the option to “ Run every \_ ” and modify to “ Run every 1 Hours ”
    - iv. First time to run: 00:00:00
    - v. Last time to run: 23:00:00
  - f. Press “ Finish ” to exit the wizard.

## 7) Verifying the Implementation

In this section, we will examine how to verify the above implementation of the operational analytics table displaying key metrics of processing messages across specified namespaces.

To observe the contents of the Demo.OperationalAnalytics.MetricsTable, we can take advantage of the built-in SQL feature.

1. From the main screen of the Management Portal, ensure you are in the REPORTS namespace.
2. Navigate to System Explorer à SQL
3. Click “ Execute Query ” and insert the following SQL statement:
  - a. select \* from DemoOperationalAnalytics.MetricsTable
4. Press “ Execute ” and the contents will be displayed via SQL.

## 8) Resulting Analytics Dashboard: Sample

By following the above implementation, we are able to achieve a table listing the metrics for messages from specified namespace(s).

Namespace	Year	AvgAckTime	AvgTime	MaxTime	MinTime	NumInbound	NumOutbound
FHIRREPO	2019	0.027	0.214	0.396	0.012	22	22
SANDBOX	2019	23.781	48.071	3,621.146	11.125	38549	3884

\*Please note that the data has been generated for display purposes.

Double click on "2019" next to SANDBOX to take advantage of the DateRollUp we created!

		AvgAckTime	AvgTime	MaxTime	MinTime	NumInbound	NumOutbound
SANDBOX	◀ Mar-2019	23.781	40.371	129.196	3.425	48	34
	◀ Jun-2019	0.000	7.780	3,528.230	7.780	38900	3890
	◀ Jul-2019	0.000	0.000	0.000	0.000	0	0
	◀ Sep-2019	0.000	0.000	0.000	0.000	1	0

Continue clicking down into "Mar-2019" or any of the other months to drill down into days and then hours.

... and that's it! Please feel free to adapt the sample code to meet the needs of your organization!

If you have any questions, please reach out to your account's Sales Engineer!

[#Analytics](#) [#Cubes](#) [#Dashboards](#) [#Health Connect](#) [#InterSystems IRIS BI \(DeepSee\)](#) [#InterSystems IRIS for Health](#)  
[Check the related application on InterSystems Open Exchange](#)

---

Source URL: <https://community.intersystems.com/post/developing-operational-analytics-dashboards>