



TGH

Making Integrations Simpler



Migrating Process Schedules Across Atoms with Atomsphere API in Boomi



Migrating Process Schedules Across Atoms with AtomSphere API in Boomi

In this blog, we will see how to migrate process schedules from one atom to another atom using Atmosphere API.

What is Atmosphere API?

It is an API in Boomi which automates the packaging, deployment of integration processes and other components. All AtomSphere API calls are authenticated by a user name or API token and an account ID that is associated with an active AtomSphere account. Deploying an AtomSphere API connector does not affect your license count.

Use Case:

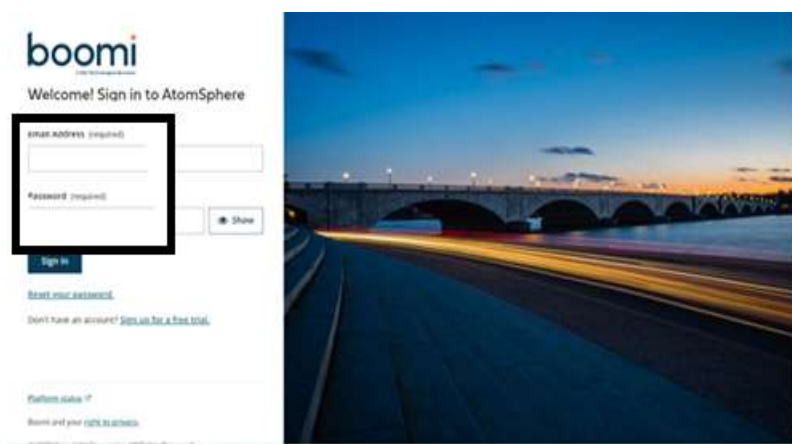
We will configure the Atmosphere API connector and apply a filter to fetch the source atom ID and transform the process schedules from one atom to another atom by setting the Dynamic process property and Dynamic Document Property. Also, we will make use of extensions to get the value of Source Atom ID which is configured in the Dynamic Process property.

In this Use case, we have deployed the “Mail Process” to both Source and Target Atoms and will copy the schedule from the Source atom ID to the Destination Atom ID using AtomSphere API.

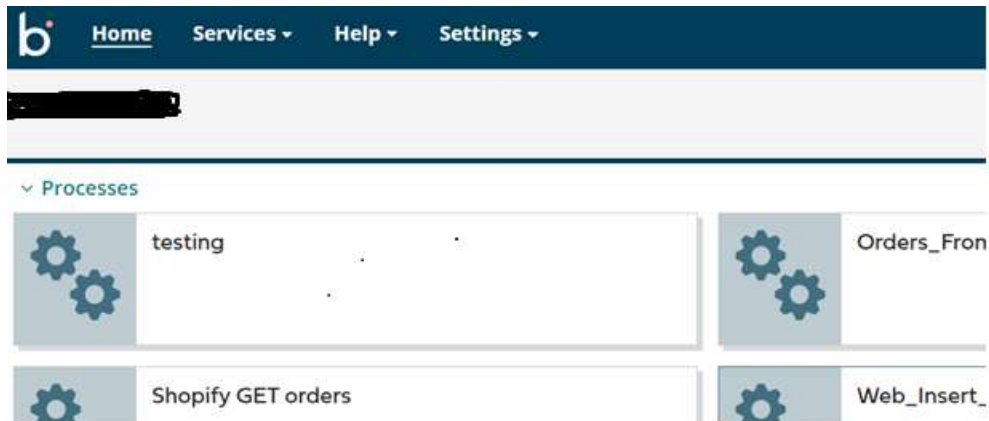
NOTE: We have to make sure to deploy the process to source and target atoms.

Let us begin with the steps to develop the process in Boomi.

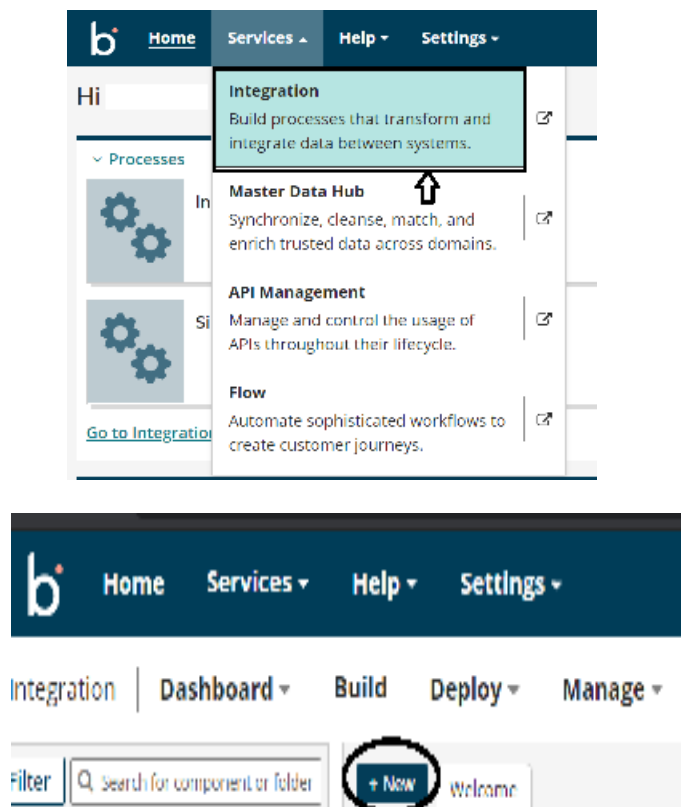
Step 1: Log on to the Boomi platform (<https://platform.boomi.com/>) with the required credentials i.e. Email Address and Password.



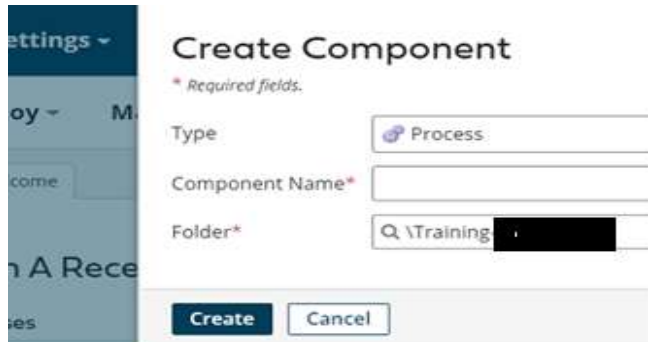
Step 2: Once logged into the Boomi platform, we will be able to view the Home page.



Step 3: Now, click on Services followed by Integration. We will see the Build page. Click on New.



Step 4: Once, click on New, we will be able to see three fields i.e. Type, Component Name and Folder.



- Select Type as process as we are building a process. Component Name and Folder can be given based on your choice (i.e. which name to be given and where do we want to create the process). Click on Create.

Step 5: We see that the process gets created with a start shape which is configured with AS2 Shared Server by default.



Step 6: Select the start shape and choose No Data. Click ok.



Start Shape ⓘ

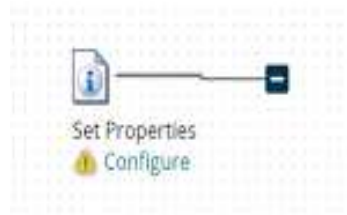
The Start shape is the main shape that begins the process flow. It is automatically added to each new process and it cannot be removed.

Process Mode: General

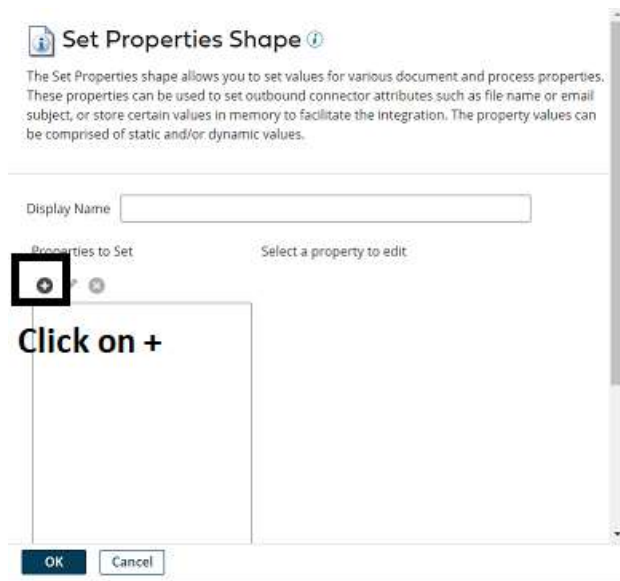
Type: ☐ Connector ☐ Trading Partner ☐ Data Passthrough ☒ No Data

Select this option if this process should not receive or retrieve data from any source. In other words, the process will be scheduled and run similar to a cron job or any other sort of job scheduler. The process will generate a single empty document.

Step 7: Drag and drop the Set Properties shape onto the process canvas to create Dynamic Process Property and Dynamic Document Property.



- Click + on properties to set the property name.



- Choose Dynamic Process Property from the drop-down.



- Give the property name “Source Atom ID” and click ok.

Choose Property

Property Type: Dynamic Process Property

Property Name*: Source Atom Id

Options: ☐ Persist this property

Step 8: Now, we have to set parameters for Dynamic Process Property. Select the Dynamic Process property and choose + on parameters.

Properties to Set: Dynamic Process Property - Source Atom Id

Parameters: +

- Here, we are selecting Type as Static from drop down and Static Value will come from extensions.

Parameter Value

Type: Static

Static Value: Connector Call
Cross Reference Lookup
Current Data
DateTime
Document Cache Lookup
Document Property
Execution Property
Dynamic Process Property
Process Property
Profile Element
Sequential Value
SQL Statement
Static
Stored Procedure
Unique Value

Parameter Value

Type: Static

Static Value: Comes from Extensions

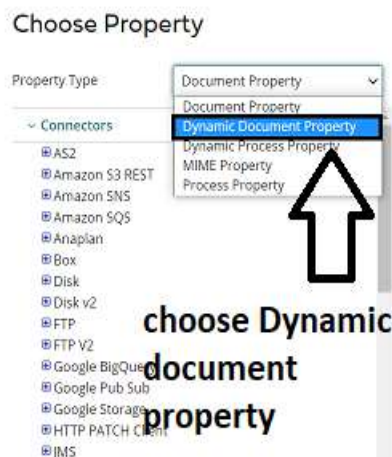
- In this Use Case, the Source Atom ID will be the local atom to which the process is deployed and scheduled.

Step 9: We will now create Dynamic Document Property in set properties shape to add target atom id.

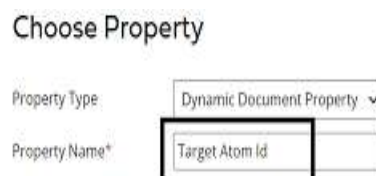
- Let us create by selecting the + symbol as shown.



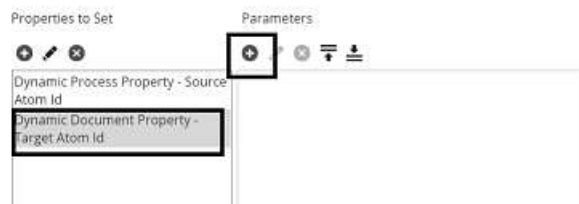
- Choose Dynamic Document Property from the drop-down.



- Give the property name “Target Atom ID” and click ok.



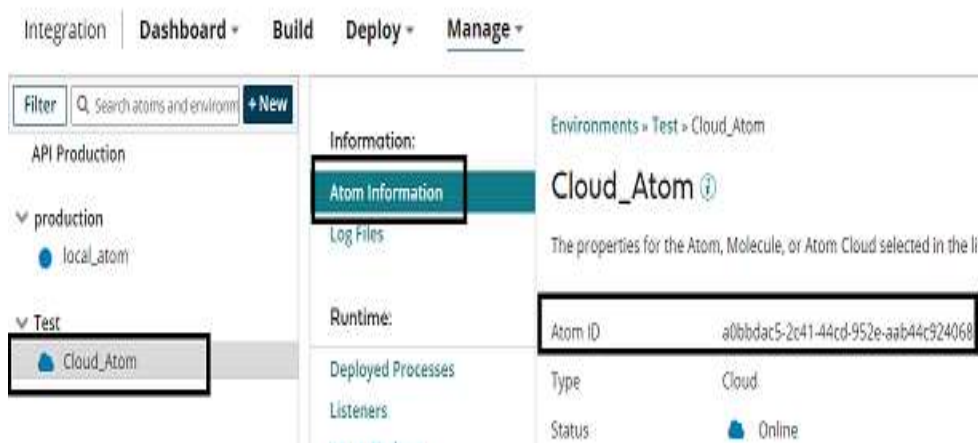
- Now, we have to set parameters for Dynamic Document Property. Select the Dynamic Document property and choose + on parameters.



- Here, we are selecting Type as Static from the drop-down and Static Value would be the target atom ID.
- To get the Atom ID, navigate to Manage > Atom Management.



- Select the atom > Atom Information and get the Atom ID.



- In this Use Case, the Target Atom ID will be the cloud atom to which the process is deployed and to be scheduled once we run the process. Click ok.

Parameter Value

Type

Static Value

Static
 Connector Call
 Cross Reference Lookup
 Current Data
 Date/Time
 Document Cache Lookup
 Document Property
 Execution Property
 Dynamic Process Property
 Process Property
 Profile Element
 Sequential Value
 SQL Statement
 Static
 Stored Procedure
 Unique Value

Parameter Value

Type

Static Value

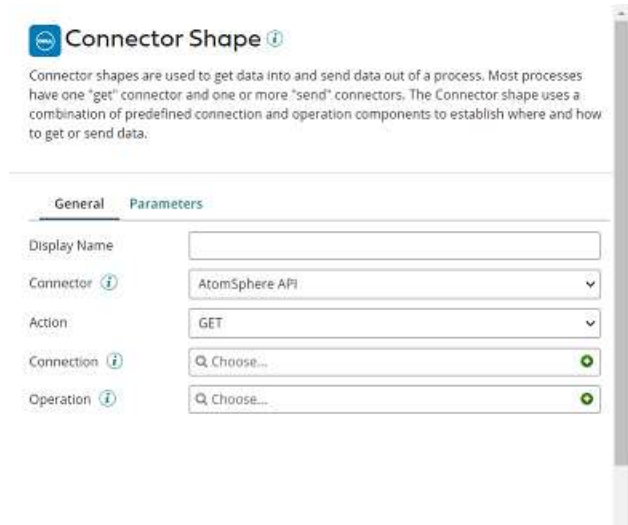
Static

a0bbdac5-2c41-44cd-952e-aab44

- After setting properties and parameters in the Set Properties shape, it looks like

Properties to Set	Parameters
+ ✎ ✕ Dynamic Process Property - Source Atom Id Dynamic Document Property - Target Atom Id	+ ✎ ✕ ⚙ ⚡ Static value of 'Comes from Extensions'

Step 10: Drag and drop the Atmosphere connector onto the process canvas and configure it.



Connector Shape

Connector shapes are used to get data into and send data out of a process. Most processes have one "get" connector and one or more "send" connectors. The Connector shape uses a combination of predefined connection and operation components to establish where and how to get or send data.

General Parameters

Display Name

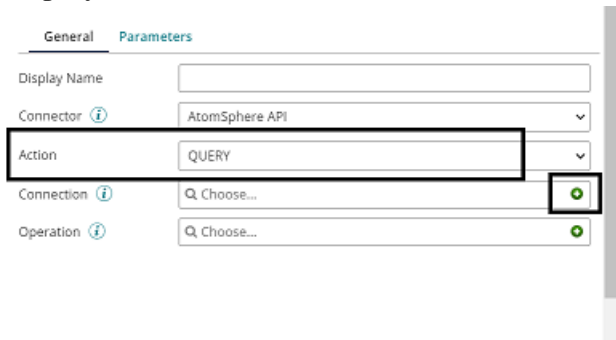
Connector

Action

Connection

Operation

- Here, we choose Action as **QUERY** as we are querying the process schedule for the existing process deployed to the atom.



General Parameters

Display Name

Connector

Action

Connection

Operation

- Name the connection and configure the details.



AtomSphere API Connection - AtomSphere API

Connection

WSDL URL

Authentication Type

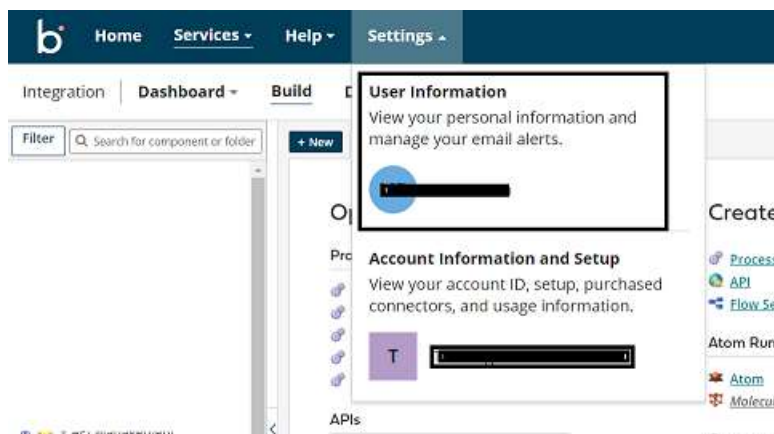
Username

Password

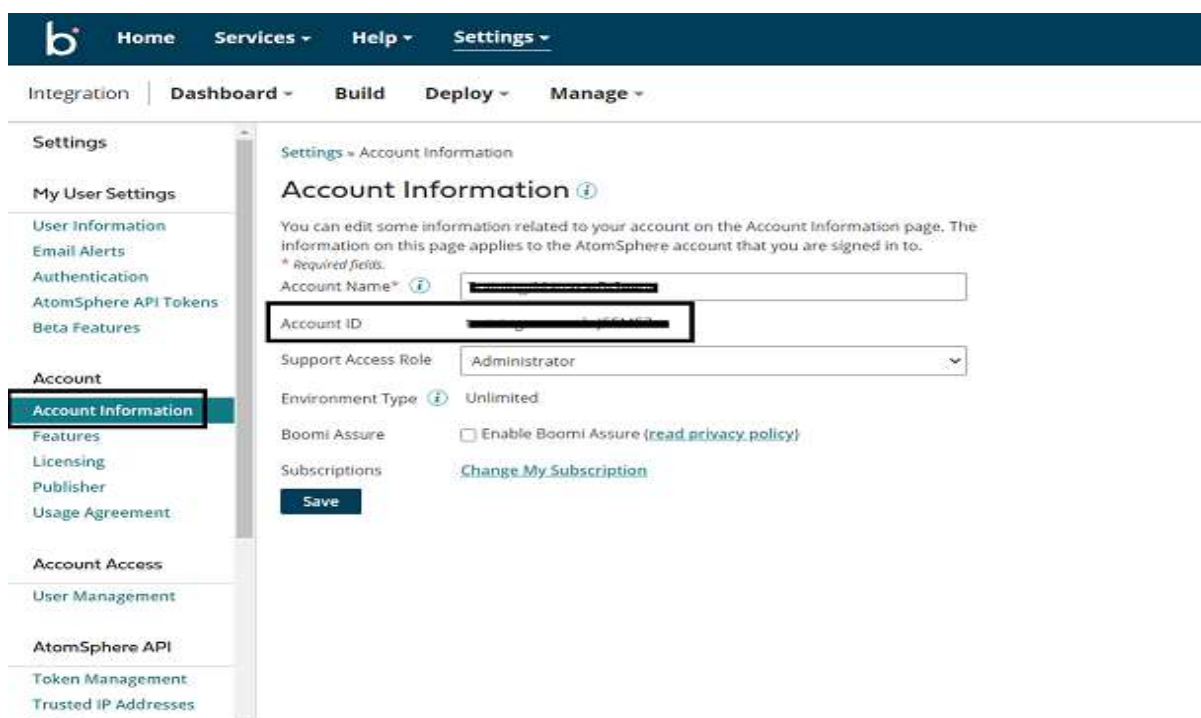
API Token

- In the WSDL URL, give your account ID instead of ACCOUNT_ID.

- To get a Boomi Account ID, go to settings and click on User Information.

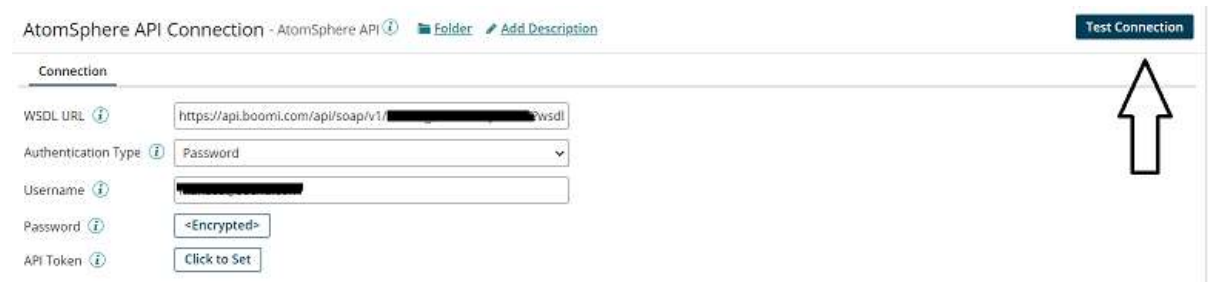


- Select Account Information on the left and We will find Account ID



- Authentication Type will be Password.
- Give the name and password of your Boomi account.

Step 11: Now, test the connection as shown in the screenshot.



AtomSphere API Connection - AtomSphere API ⓘ Folder Add Description

Test Connection

Connection

WSDL URL ⓘ

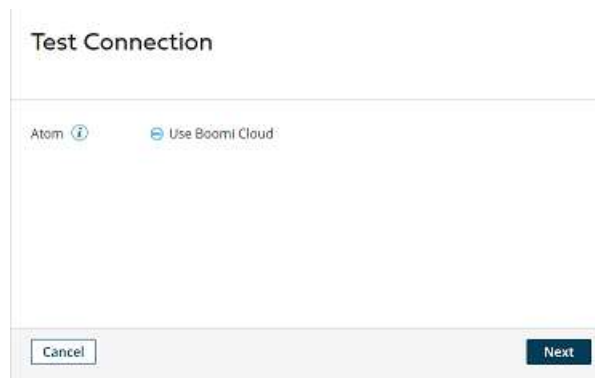
Authentication Type ⓘ Password

Username ⓘ

Password ⓘ <Encrypted>

API Token ⓘ Click to Set

- Click on Next.

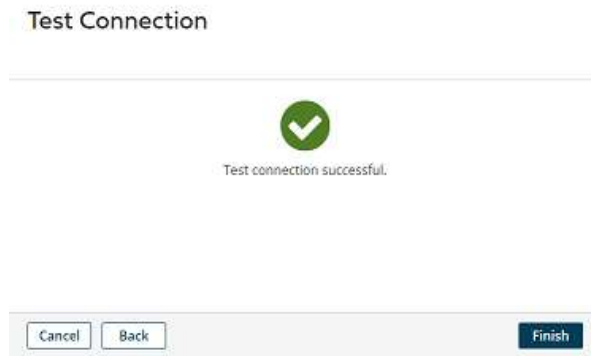


Test Connection

Atom ⓘ Use Boomi Cloud

Cancel Next

- We see that the Test connection is successful. Click Finish, Save and Close.

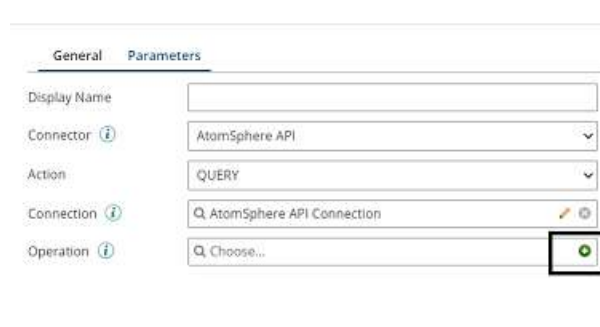


Test Connection

Test connection successful.

Cancel Back Finish

Step 12: Click + on operation and name it.




General Parameters

Display Name

Connector ⓘ AtomSphere API

Action QUERY

Connection ⓘ Q. AtomSphere API Connection

Operation ⓘ Q. Choose... 

Package Schedule Operation AtomSphere API Operation ⓘ

Options Archiving Tracking Caching

Connector Action: QUERY

Object

Tracking Direction ⓘ ☐ Input Documents ☒ Output Documents

Error Behavior ☐ Return Application Error Responses ⓘ

- Click on Import.

Package Schedule Operation - AtomSphere API Operation ⓘ Folder Add Description

Options Archiving Tracking Caching


Connector Action: QUERY

Object

Tracking Direction ⓘ ☐ Input Documents ☒ Output Documents

Error Behavior ☐ Return Application Error Responses ⓘ

Import



- Choose the connection which we have configured in Atmosphere API Connection. Click next.

AtomSphere API Import Wizard

* Required fields

Connection*

Filter ⓘ

- Select the Object type as Process Schedules. Click next.

AtomSphere API Import Wizard

Object Type

- Account
- HL7ConnectorRecord
- IntegrationPack
- IntegrationPackAtomAttachment (DEPRECATED)
- IntegrationPackEnvironmentAttachment
- IntegrationPackInstance
- ListenerStatus
- PackagedComponent
- Process
- ProcessAtomAttachment (DEPRECATED)
- ProcessEnvironmentAttachment (DEPRECATED)
- ProcessSchedules**
- ProcessScheduleStatus
- Role
- RosettaNetConnectorRecord
- ThroughputAccount
- ThroughputAccountGroup
- TradacomsConnectorRecord

- We see that the Process Schedules Object Profile gets imported. Click finish, save and close.

AtomSphere API Import Wizard

Operation Loaded

Object Name: ProcessSchedules

Response Profile: AtomSphere API ProcessSchedules QUERY Response

Recommended Filters ⓘ

Popular filters for ProcessSchedules Query on AtomSphere API. Select one to apply to your operation.

☒ No thanks, I will set up my own filters.

☐ atomId EQUALS ?

☐ atomId EQUALS ? AND processId EQUALS ?

☐ processId EQUALS ?

☐ atomId EQUALS ?

Cancel

Finish

Package Schedule Operation - AtomSphere API Operation ⓘ Folder Add

Options Archiving Tracking Caching

Connector Action: QUERY

Object: ProcessSchedules

Response Profile: AtomSphere API ProcessSchedules QUERY Response ⓘ

Tracking Direction ⓘ ☐ Input Documents ☒ Output Documents

Error Behavior ☐ Return Application Error Responses ⓘ

Objects: ProcessSchedules Filters: Filter

Step 13: Now go to filters and add a filter by clicking on Add Expression.

Process Schedules Operation - AtomSphere API Operation ⓘ Folder Add Description

Options Archiving Tracking Caching

Connector Action: QUERY

Object: ProcessSchedules

Response Profile: AtomSphere API ProcessSchedules QUERY Response ⓘ

Tracking Direction ⓘ ☐ Input Documents ☒ Output Documents

Error Behavior ☐ Return Application Error Responses ⓘ

Objects: ProcessSchedules Filters: Filter

Filter

Add Logical Sub-Group

Add Expression

- Add expression name as Source Atom Id.

Process Schedules Operation - AtomSphere API Operation [Folder](#) [Add Description](#)

Options Archiving Tracking Caching

Connector Action: QUERY

Object: ProcessSchedules

Response Profile: Q AtomSphere API ProcessSchedules QUERY Response

Tracking Direction: ☐ Input Documents ☒ Output Documents

Error Behavior: ☐ Return Application Error Responses

Objects: ProcessSchedules

Filters: Filter

Filter Name: Source Atom Id

Field: Q Choose...

- Choose atom id from the drop-down as we are fetching the schedule of process based on atom id. Click ok and we will set the value of the source ID in the parameter section of Atmosphere API Connector.

Object: ProcessSchedules

Response Profile: Q AtomSphere API ProcessSchedules QUERY Response

Tracking Direction: ☐ Input Documents ☒ Output Documents

Error Behavior: ☐ Return Application Error Responses

Objects: ProcessSchedules

Filters: Filter

Filter Name: Source Atom Id

Field: Q Choose...

Choose a field

processId

atomId

OK Cancel

Step 14: Go to the parameters section of the Atmosphere connector and click on +

General Parameters

Display Name:

Connector: AtomSphere API

Action: QUERY

Connection: Q AtomSphere API Connection

Operation: Q Process Schedules Operation

You have set 0 of 1 parameter values. You may need to set all 1 values for your process!

No parameters have been set. Would you like to [import all](#)?

- Set the input type as Source Atom ID, Type as Dynamic Process Property and give the property name as Source Atom ID and the value for it will come from the extension. Click ok.

Parameter Value

Input	<input type="text" value="Q Source Atom Id"/>
Type	<input type="text" value="Dynamic Process Property"/>
Property Name	<input type="text" value="Source Atom Id"/>
Default Value	<input type="text"/>

Step 15: Drag and drop Atmosphere Api onto process canvas to update the process schedule.

Connector Shape

Connector shapes are used to get data into and send data out of a process. Most processes have one "get" connector and one or more "send" connectors. The Connector shape uses a combination of predefined connection and operation components to establish where and how to get or send data.

General	Parameters
Display Name	<input type="text"/>
Connector	<input type="text" value="AtomSphere API"/>
Action	<input type="text" value="GET"/>
Connection	<input type="text" value="Q Choose..."/>
Operation	<input type="text" value="Q Choose..."/>

- Here, the Action would be an UPDATE as we are updating the Process Schedule from one atom to another atom. Choose the same connection i.e. (Atmosphere API Connection) which we used for the querying process Schedule in the above steps.

General	Parameters
Display Name	<input type="text"/>
Connector	<input type="text" value="AtomSphere API"/>
Action	<input type="text" value="UPDATE"/>
Connection	<input type="text" value="Q AtomSphere API Connection"/>
Operation	<input type="text" value="Q Choose..."/>

Step 16: Click + on operation to create a new one, name it and choose import.

Process Schedule Update Operation - AtomSphere API Operation

Options Archiving Tracking Caching Import

Connector Action:

Object

Tracking Direction: ☒ Input Documents ☐ Output Documents

Error Behavior: ☐ Return Application Error Responses




- Choose the connection which we have configured in Atmosphere API Connection. Click next.

AtomSphere API Import Wizard

* Required fields

Connection*

Filter 

- Select the Object type as Process Schedules. Click next.

AtomSphere API Import Wizard

Object Type

- AccountSSOConfig
- AccountUserFederation
- Atom
- AtomExtensions (DEPRECATED)
- AtomMapExtension (DEPRECATED)
- Environment
- EnvironmentExtensions
- EnvironmentMapExtension
- EnvironmentRole
- ProcessSchedules**
- ProcessSchedulesStatus
- Role
- SharedServerInformation
- TradingPartnerComponent
- TradingPartnerProcessingGroup

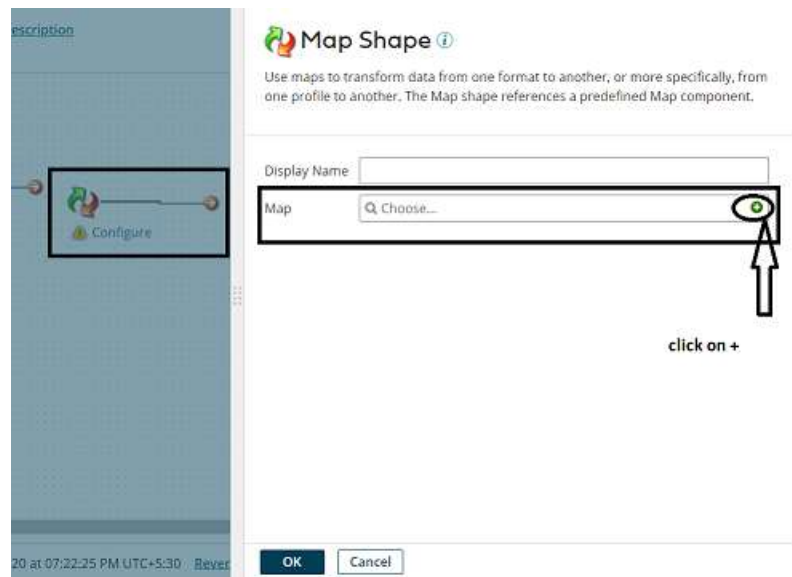
- We see that the Process Schedules Object Profile gets imported. Click finish, save and close.

AtomSphere API Import Wizard

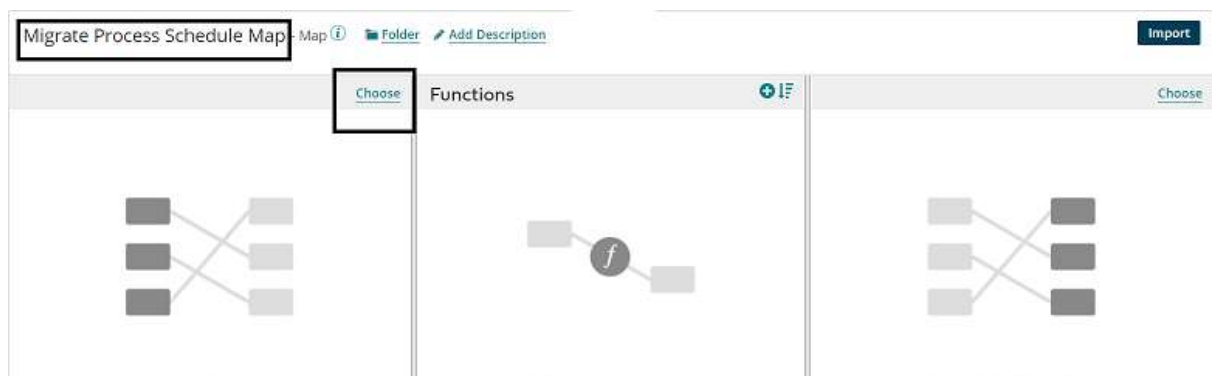
Operation Loaded

Object Name	ProcessSchedules
Request Profile	AtomSphere API ProcessSchedules UPDATE Request
Response Profile	AtomSphere API ProcessSchedules UPDATE Response

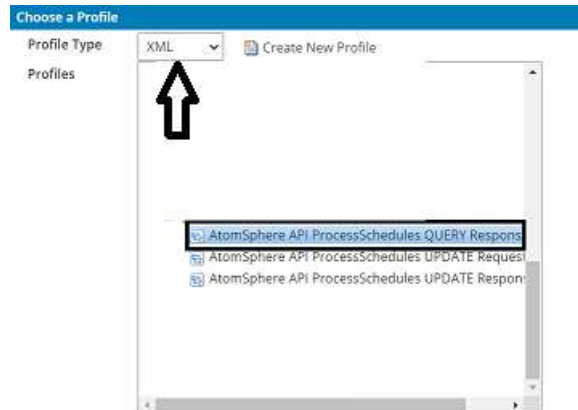
Step 17: Drag and drop the map shape onto the process canvas. Click + on the map to create one as shown below.



- Name the map and add the profiles by selecting the choose button on both sides. The left side will be the source profile which we have imported i.e.(Process Schedule Query response Profile) and the right side will be the destination profile which we have imported in the update Operation (i.e. Process Schedule Update Request Profile).
- Now, let us add the profiles. First, we will add the profile on the source side and Click on choose.



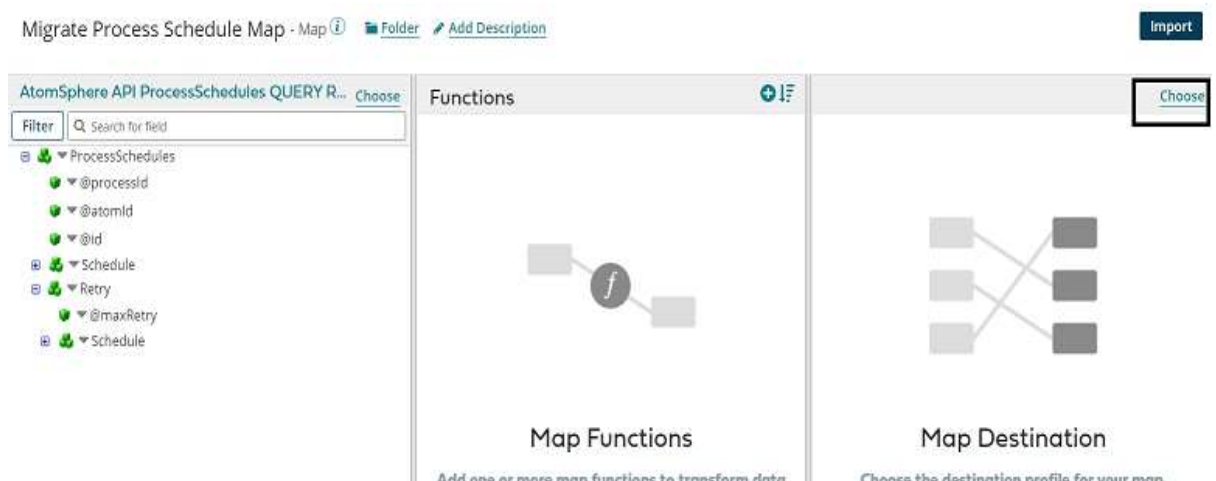
- Choose the profile type as XML and import the profile. Click ok.



- We see that the profile was added to the map.



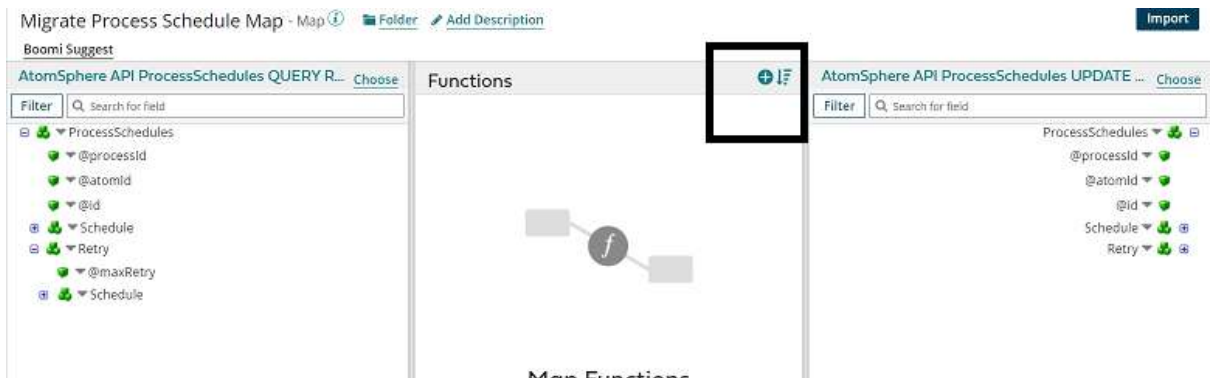
Step 18: Select Choose on the right-hand side and follow the same steps as mentioned above.



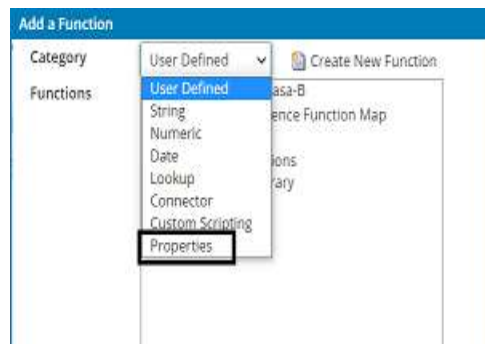
- Choose the input type as XML and select the requested profile of the process schedule object and we see both profiles have been added to the map.

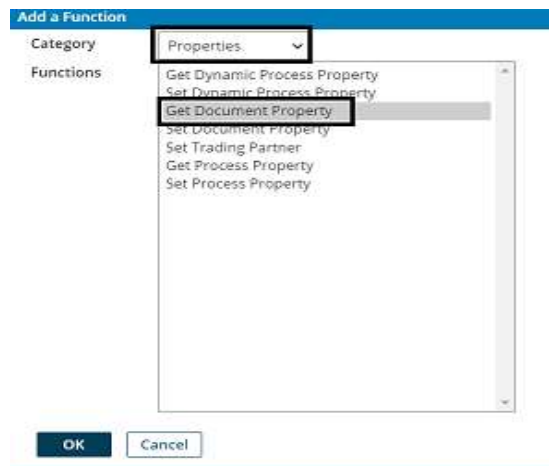


- Now, we will get dynamic document property from functions that hold the target atom ID. Click + on Functions.

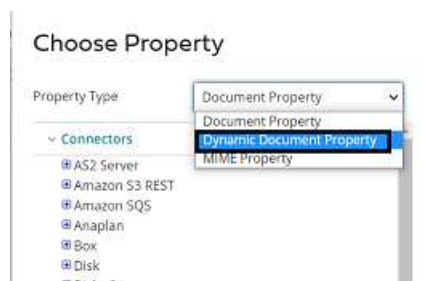


- We can see a few categories in Function i.e. String, Numeric, Date, Look up and so on. Here, we will select Category as properties and function as Get Document Property. Click ok.





- Choose Dynamic Document Property from the drop-down as shown.



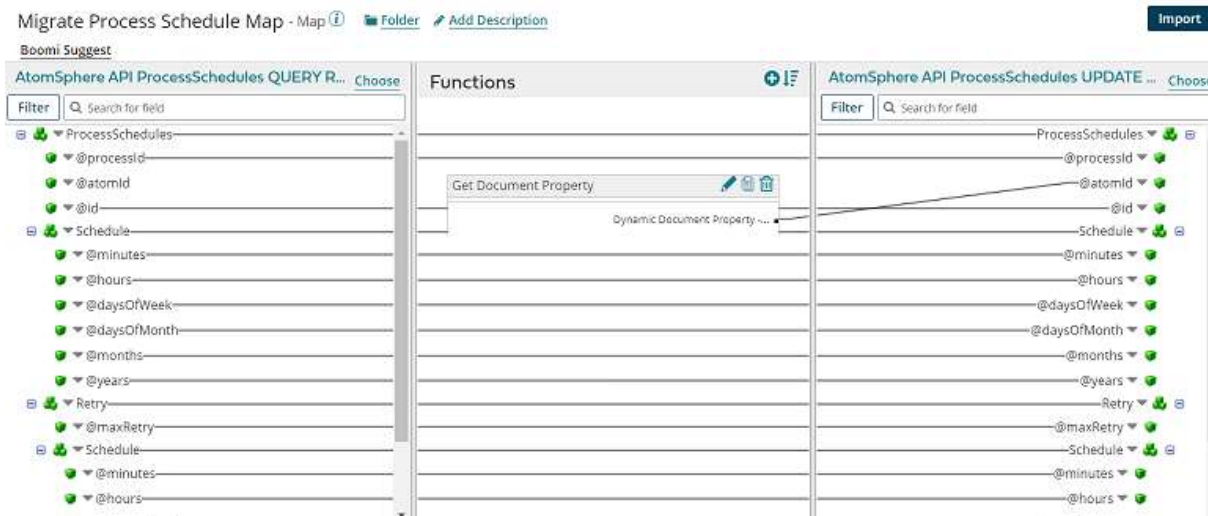
- Here, the Property Type would be Dynamic Document Property and the Property Name would be Target Atom Id which we have set in Set Properties shape. Click ok.



- We see that we can retrieve dynamic document Properties in the map. Now, set the property to the atom ID of the destination profile as shown in the screenshot.



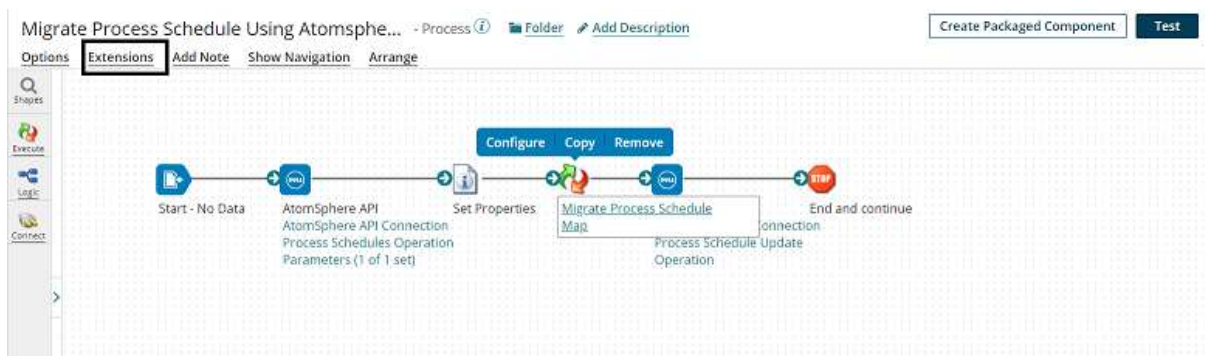
Step 19: We provide one-to-one mapping for the rest of the fields. Click save and close.



Step 20: Drag and drop the stop shape onto the process canvas to indicate the end of the flow.



Step 21: Arrange all shapes in the order and click on extensions in the build tab.



NOTE: We have to set the property as extensible, only then we will be able to set values for extensible properties at runtime.

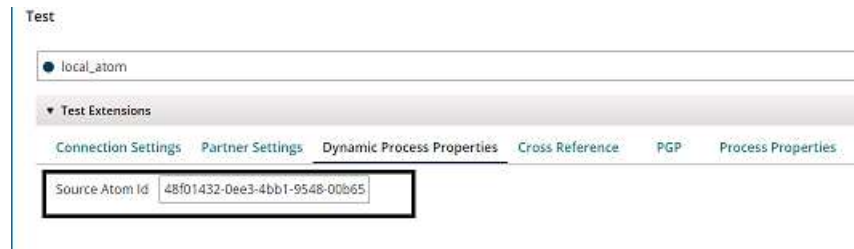
Step 22: We see Dynamic Process Properties in extensions and once we select it, click + and add the property name as “Source Atom Id” which we have configured in set properties shape and will set the value for this property via Extensions at runtime. Click ok.



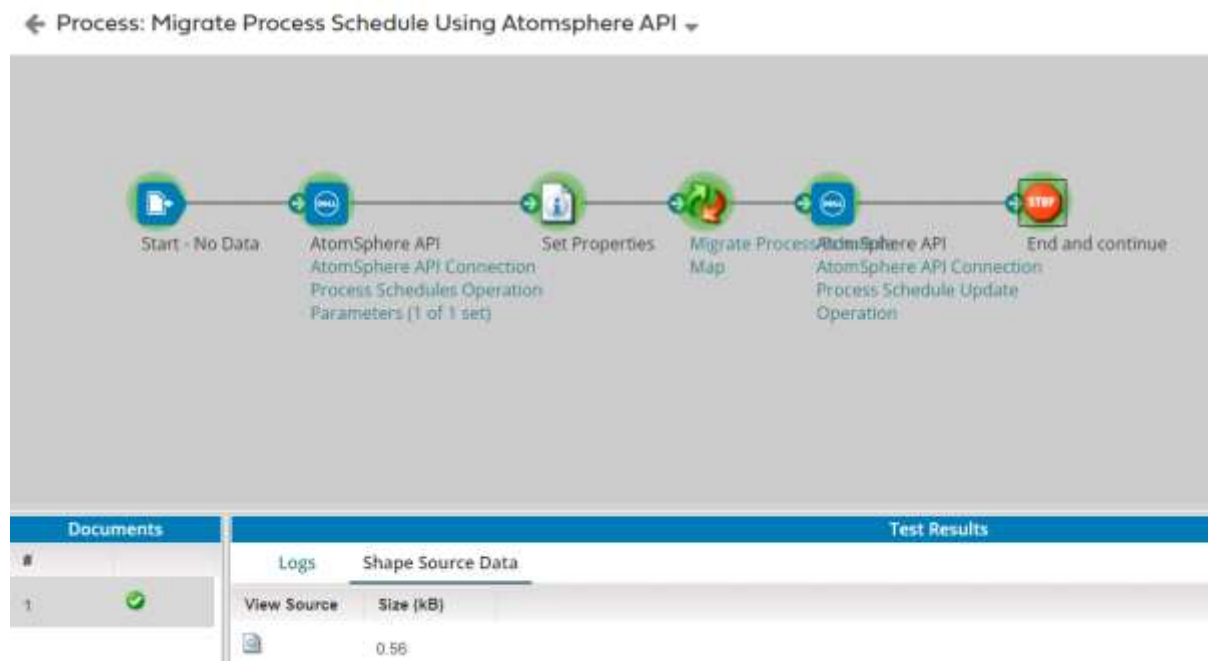
Step 23: Now, test the process by selecting the atom. Below the atom, we see there is a drop-down called **Run Test** where we can provide runtime values.



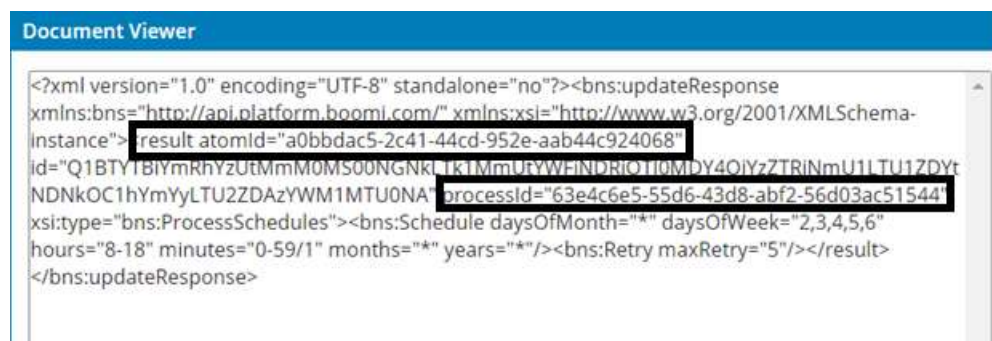
Step 24: Now, we will set the value for Dynamic Process Property by choosing Dynamic Process properties and we get to see the property name as “Source Atom Id” for which we will give the value of the local atom ID to which process is deployed and scheduled.



Step 25: Test the process and we see that the process has been executed. Click on view source to see the output.



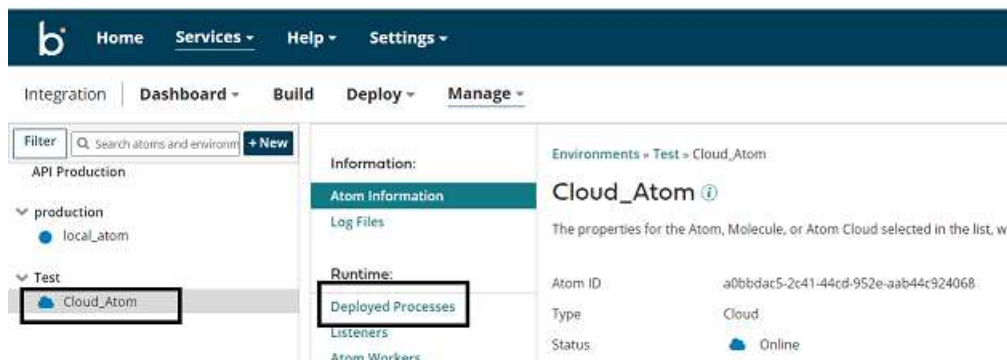
Step 26: We see that the process schedule from the local atom has been migrated to the Cloud atom.



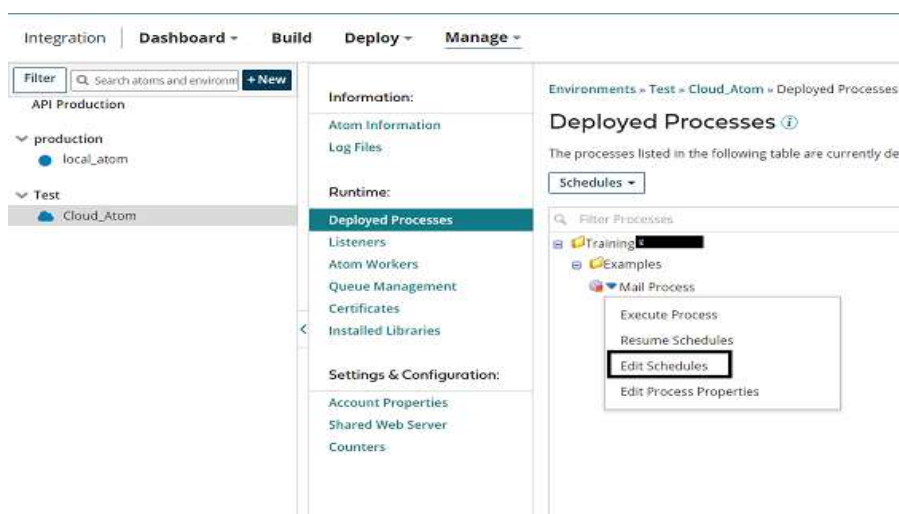
Step 27: To check if the schedule is added to the process on the target atom, close the document viewer and go to manage>atom management.



Step 28: Click on the target atom ID. Here, it is a cloud atom that selects deployed Processes.



Step 29: Select Edit Schedule and we see that the schedules have been copied from the local atom to the cloud atom of the Mail process.



Scheduling

Schedule when this process runs on the Atom. The times scheduled will run based on the Atom's time zone, which is Etc/UTC, which may not be the same as the local time zone where the schedule was created.

Execution [Retry](#)

Add

Delete

Advanced

Type

Advanced

Minutes (0-59)

0-59/1

Hours (0-23);

8-18

Days of Week (1-7)

2,3,4,5,6

Days of Month (1-31)

*

Months (1-12)

*

Years

*



TGH

Making Integrations Simpler



TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.



Email address

connect@techygeekhub.com



Phone number

+ 011-40071137
+ 91-8810610395



Our offices

Noida Office

iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office

Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081

