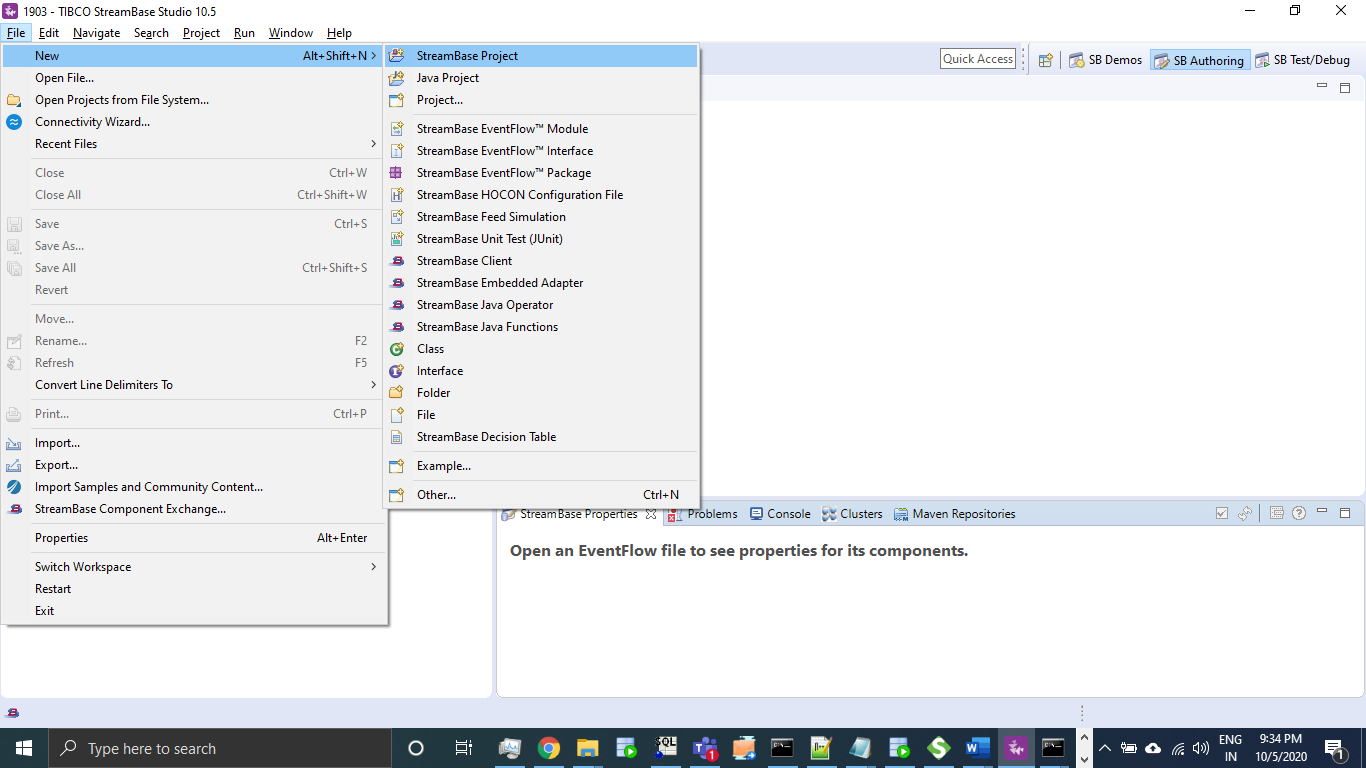
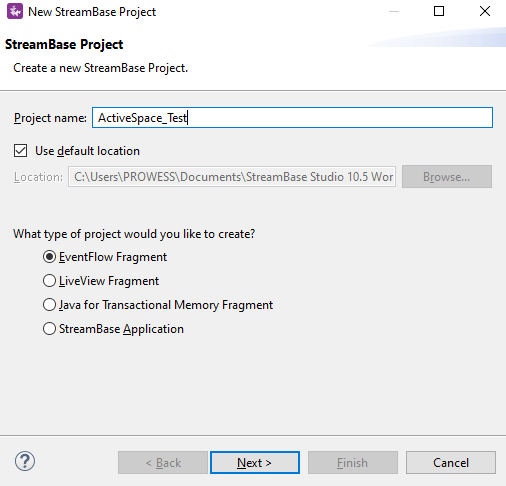
## Active Space (In memory data grid):

# 1.Create Streambase Studio Project Eventflow Fragment

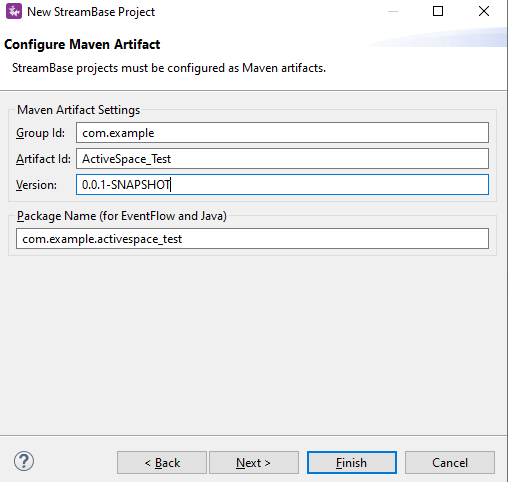
Click on File - > New - > StreamBase Project

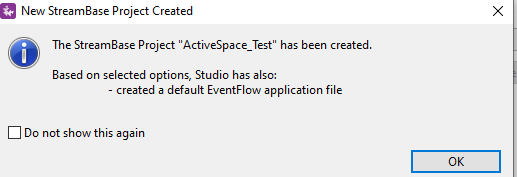


Mention Project name “ActiveSpace\_Test” and select project type as **EventFlow Fragment** and click on Next.

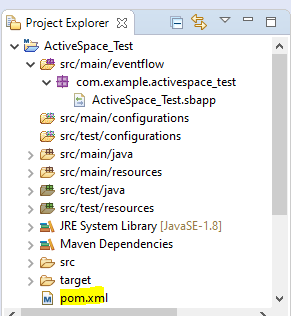


Group Id, Artifact Id, Version will generate by default for package creation change if needed and click on **Finish**.

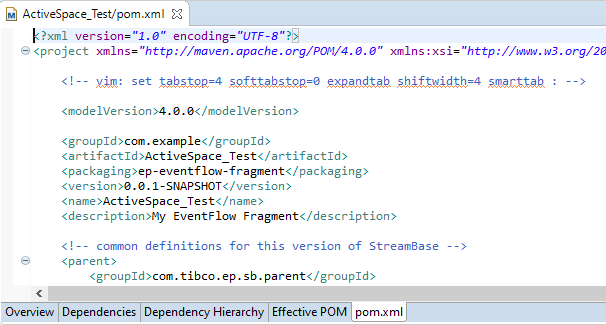




A project with package has been created having sb event flow fragment with. sbapp extension.



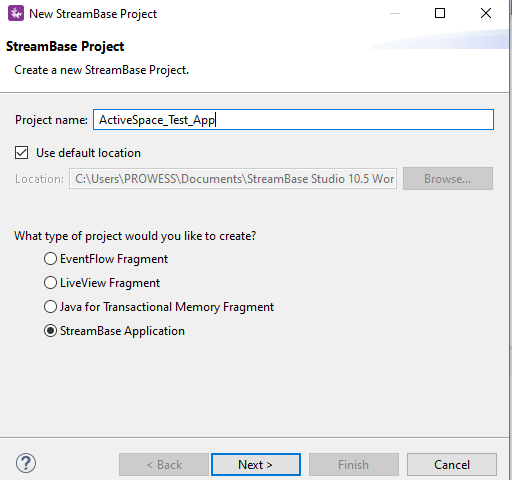
Open pom.xml having event flow fragment groupId, artifactId and version:



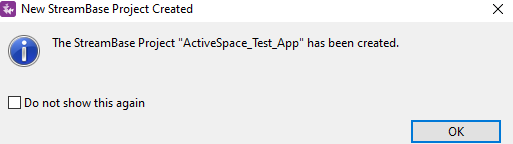
# 2.Create SB application:

Click on File - > New - > StreamBase Project

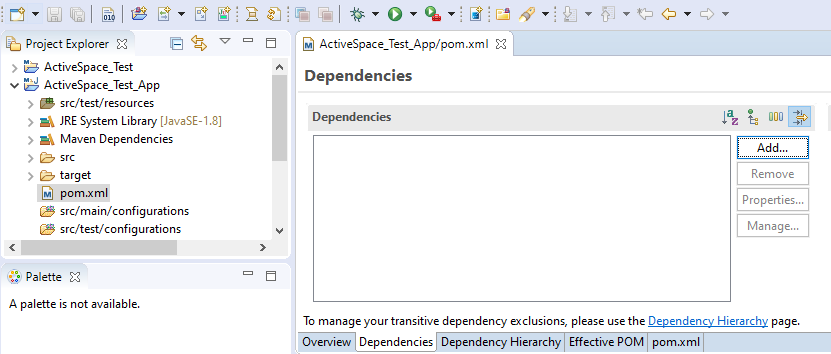
Mention Project name “ActiveSpace\_Test\_App” and select project type as **StreamBase Application** and click on Next.



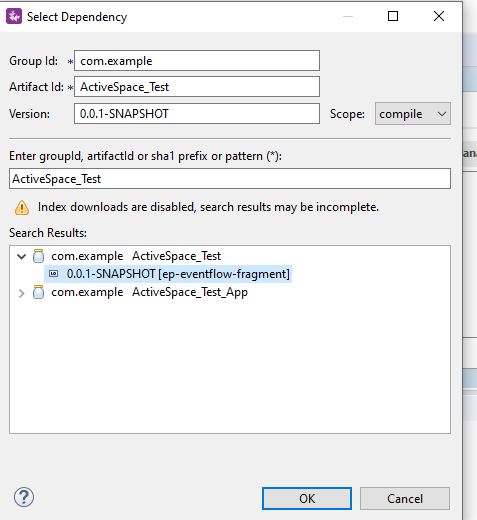
Click Next and Click on **Finish**.



Open “ActiveSpace\_Test\_App” project pom.xml and click on Dependencies Tab and click on Add to add Eventflow Fragment as a dependency created earlier in step 1.



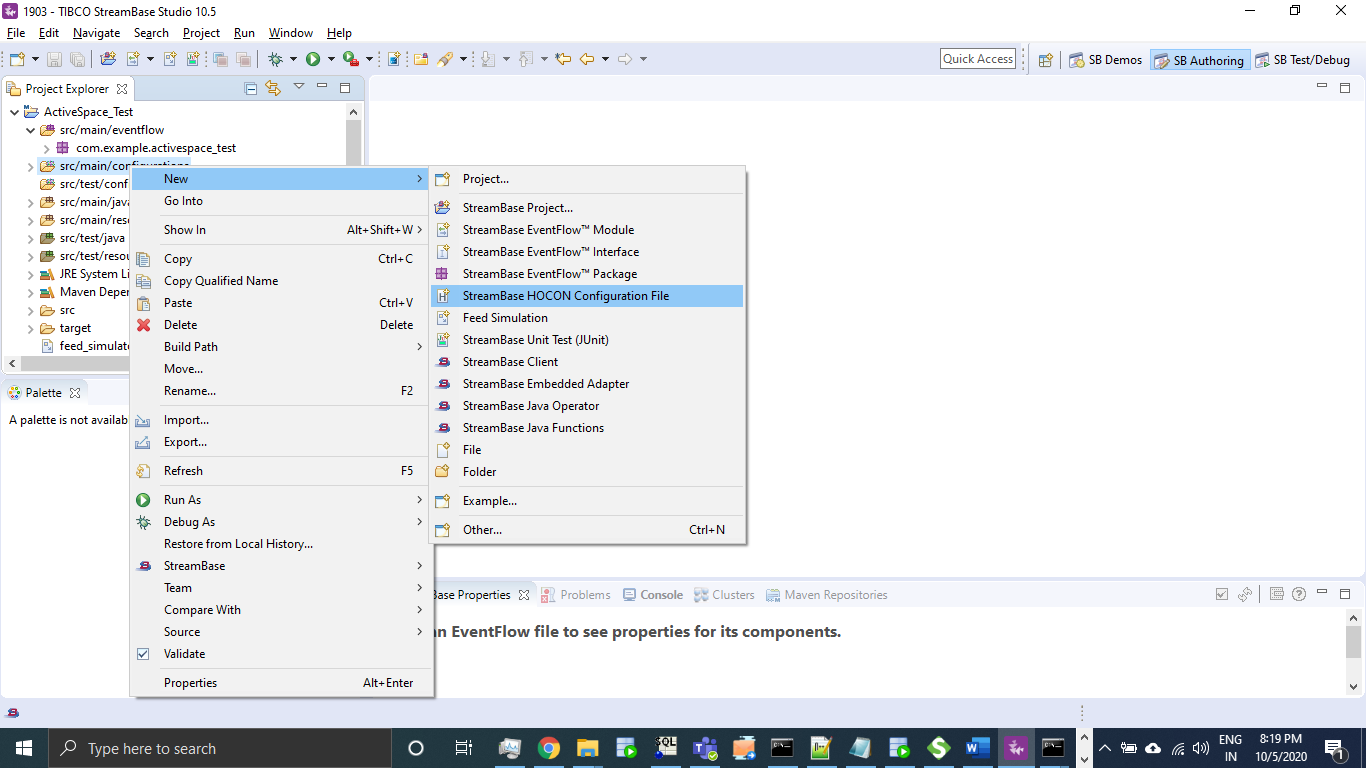
Search with aritifactId and select SNAPSHOT of ep-eventflow-fragment and click on Ok and Save.



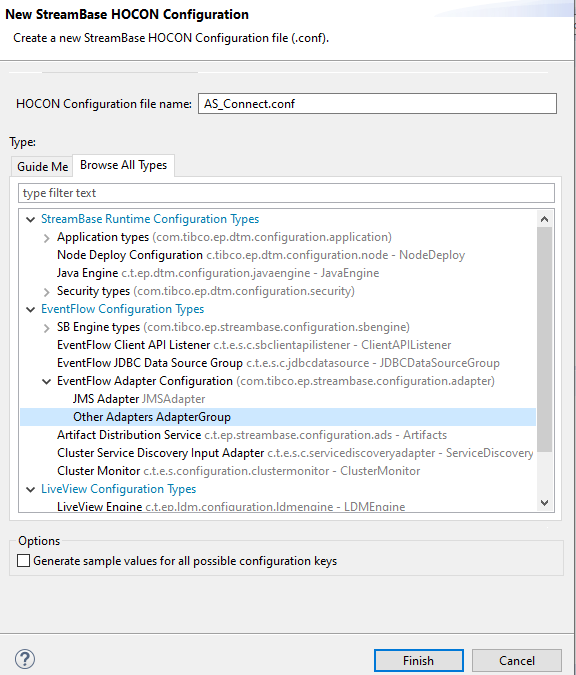
# 3. configurations:

For Activespace connection we need a configuration file in StreamBase.

In ActiveSpace\_Test project Right click on src/main/configurations - > New - > StreamBase HOCON Configuration File.



Name File as AS\_Connect Under EventFlow Configuration Types select Other Adapters AdapterGroup and click on Finish.



Add below configuration in Hocon File.

name = "sample-AdapterGroup-document"

type = "com.tibco.ep.streambase.configuration.adapter"

version = "1.0.0"

configuration = {

AdapterGroup = {

adapters = {

activespaces = {

sections = [

{

name = "datagrid-definition"

settings = {

binding-strategy = "-1"

connect-numresponses = "-1"

connect-proxynames = ""

connect-retries = "-1"

connect-wait-time-seconds = "-1"

connection-timeout-seconds = "500"

datagrid-id = "\_default"

datagrid-secondary-url = “http://localhost:8080"

datagrid-url = http://localhost:8080"

id = "\_default"

transacted-session = "true"

}

}

]

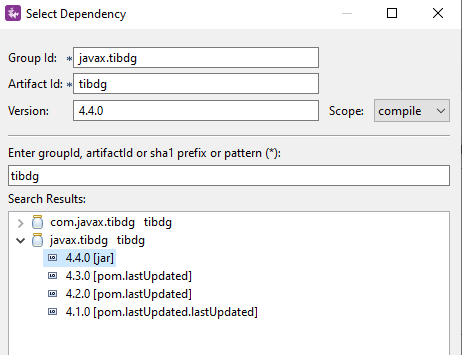
}

}

}

}

Add tibdg jar as a dependency for ActiveSpace\_Test project:



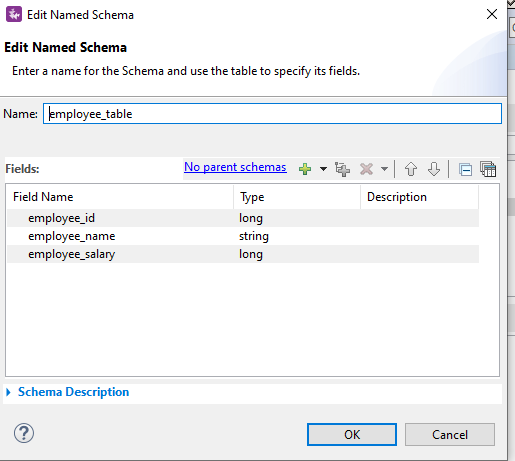
Configure tibco Activespace grid in local machine.

Create a Table in local grid, using below table script.

tibdg table create row\_counts=exact employee employee\_id long

tibdg column create employee employee\_name string employee\_salary long

Create a Named space schema in StreamBase with below elements.



Configure Process as below by using Tibco ActiveSpace Pallets as below.

1) ActiveSpaces Control (To establish a connection)

2) ActiveSpaces Delete (To Delete the existing record)

3) ActiveSpaces Put (To Insert record)

4) ActiveSpaces Get (To Query record)

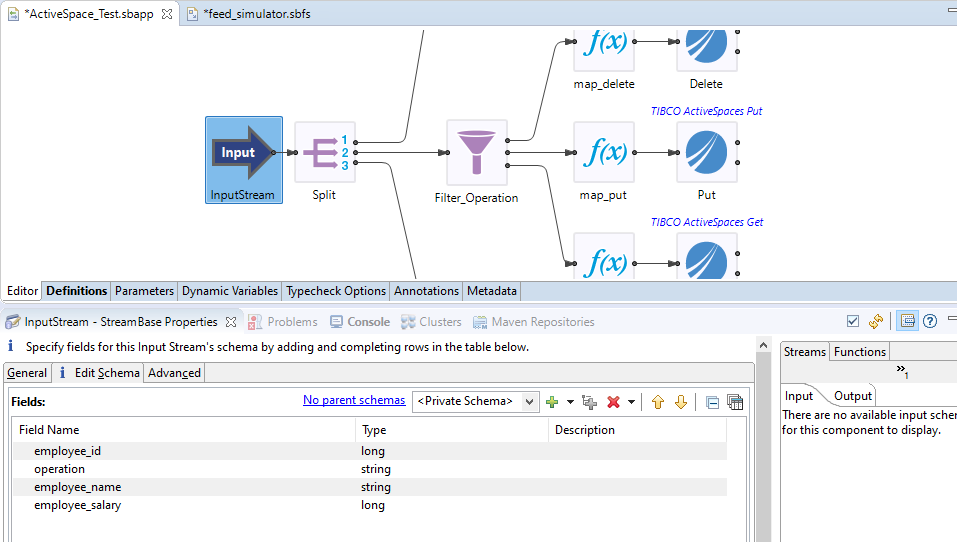
5) ActiveSpaces Transaction (To Commit Transaction)

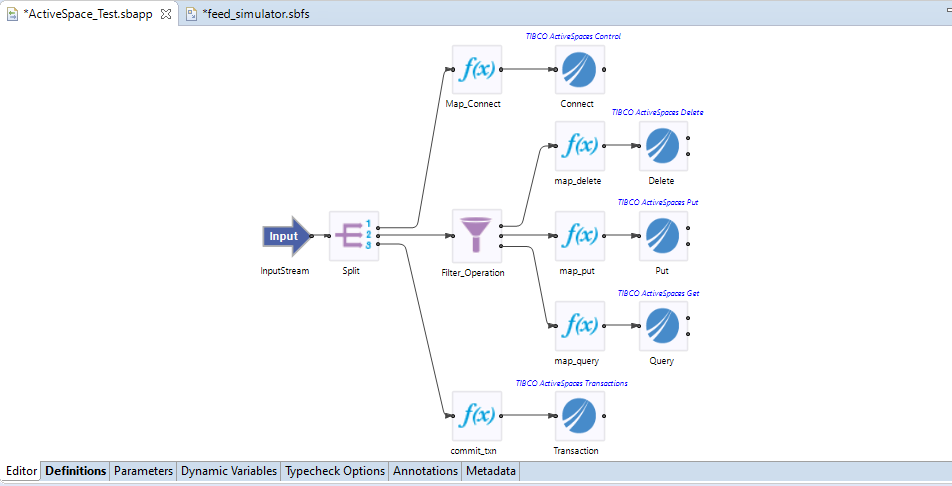
Design a Eventflow fragment using operation as a filter from the input as mentioned below to define the path either Delete,Insert,Query.

Operators:

Tuple Flow comes from InputStream, wherein further split operator executes flows in sequential order

In First path it establishes connection, secondly, executes either Delete,Put or Query based on the input Operation provided in Filter operator where input.Operation = ‘D - Delete’ or ‘I - Insert’ or ‘Q - Query’ further mapper is used to map the required elements employee\_id, employee\_name and employee\_salary, in final split Transaction will get committed for the operation performed.

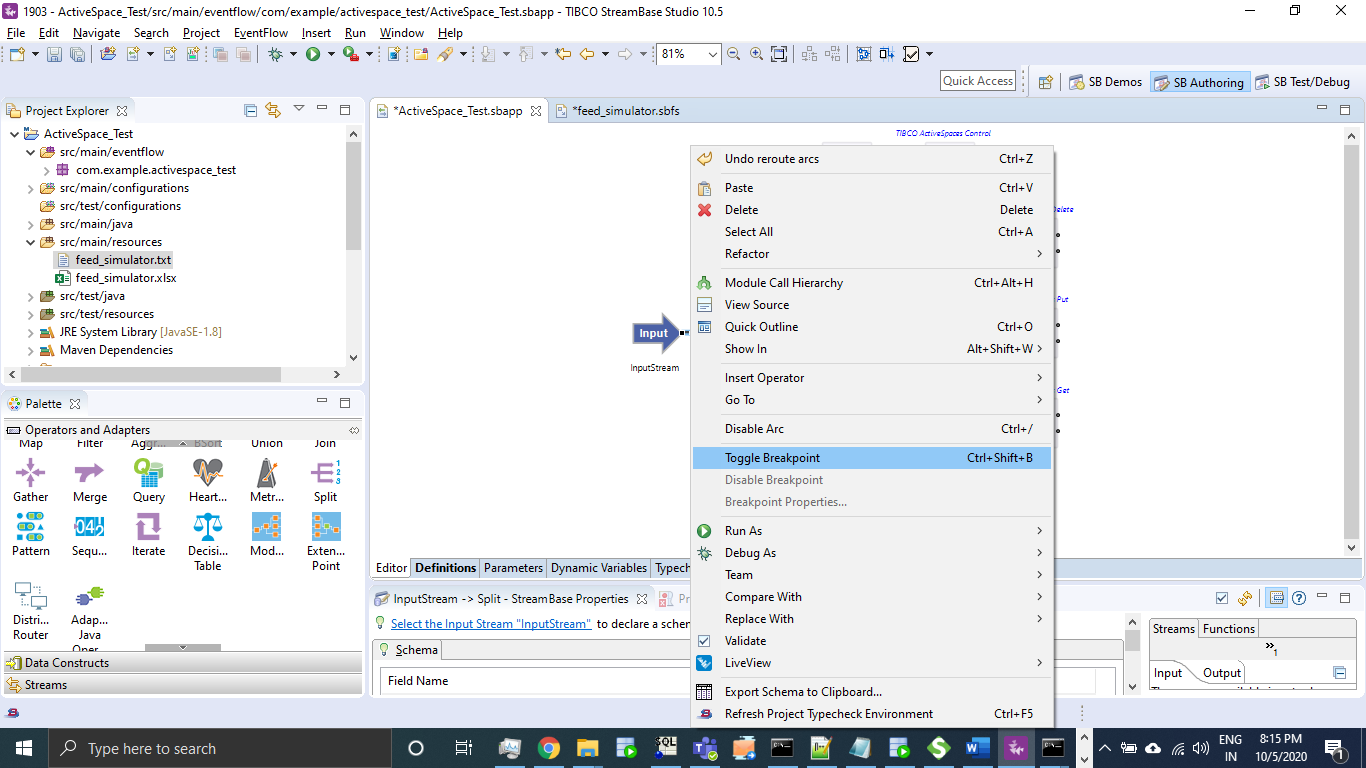




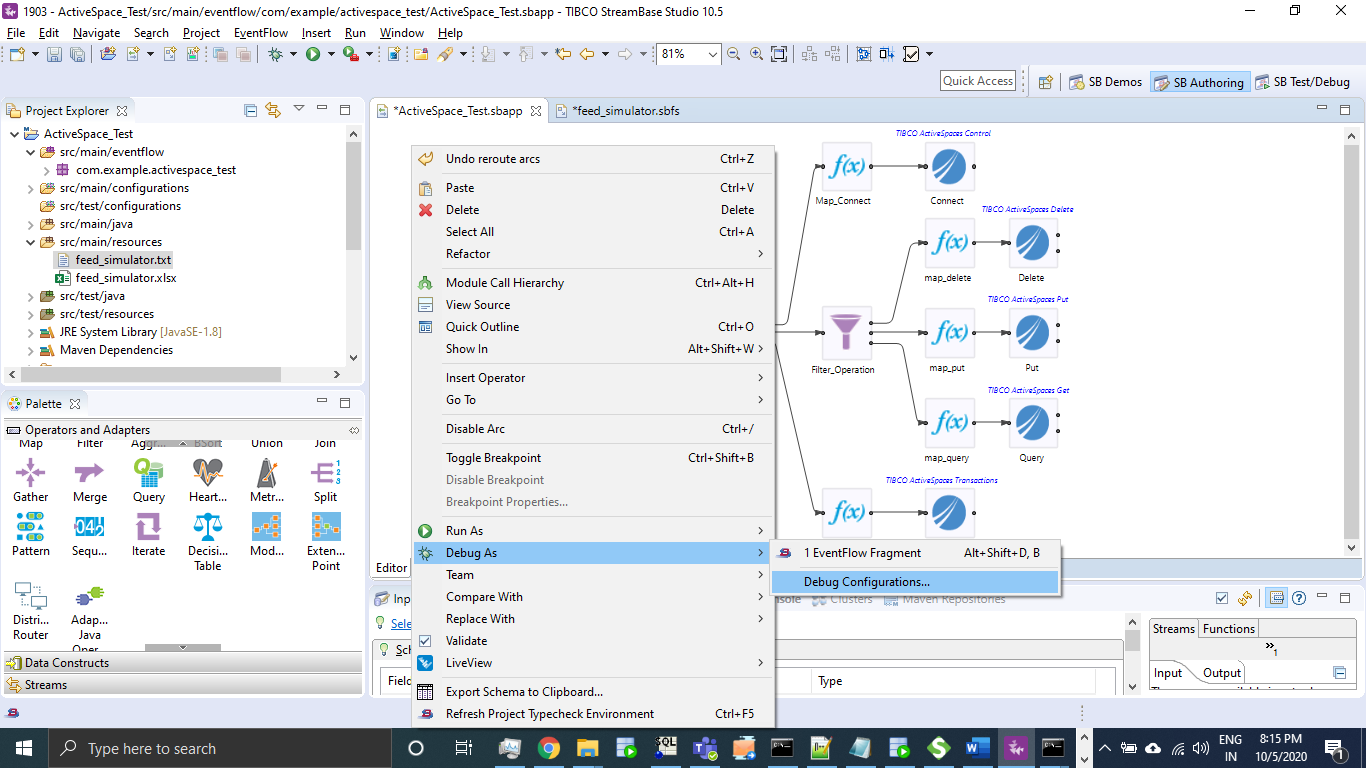
# 4.Run the application (Studio)

Using Debug Mode:

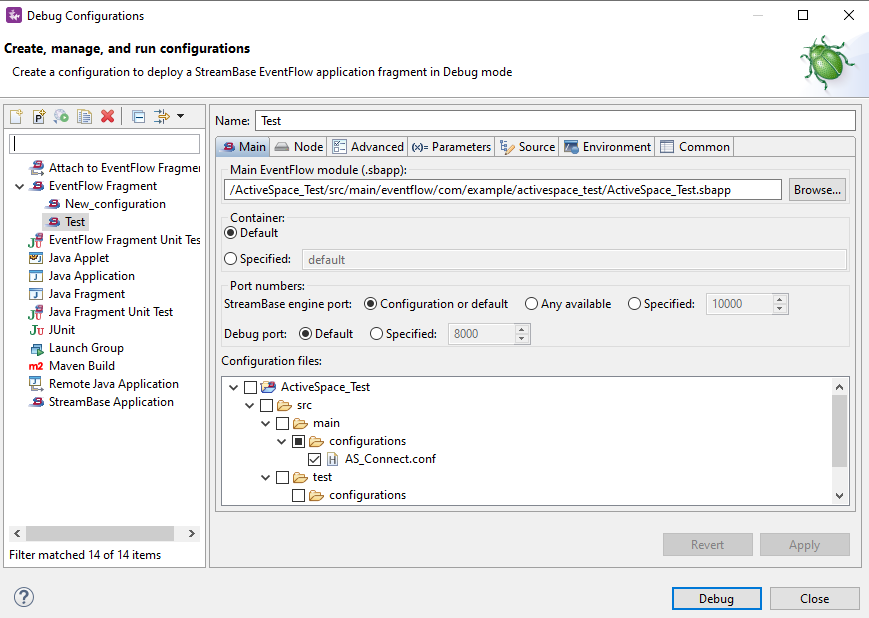
Right click on Transition and click on Toggle Breakpoint as below to control the flow.



Right click on canvas and select Debug As -> Debug configurations as below

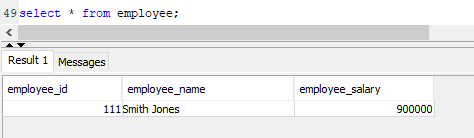


Add new config **Test** Browse sbapp and select AS\_Connect.conf and click on **Debug**.

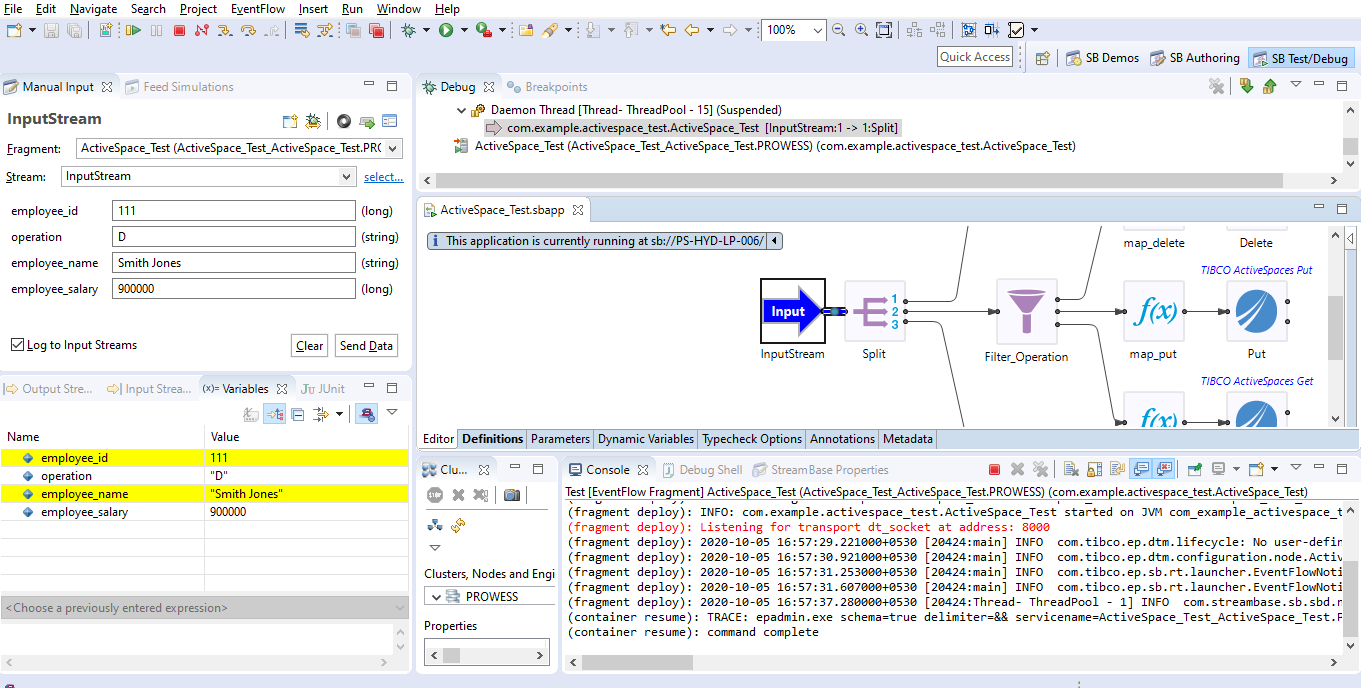


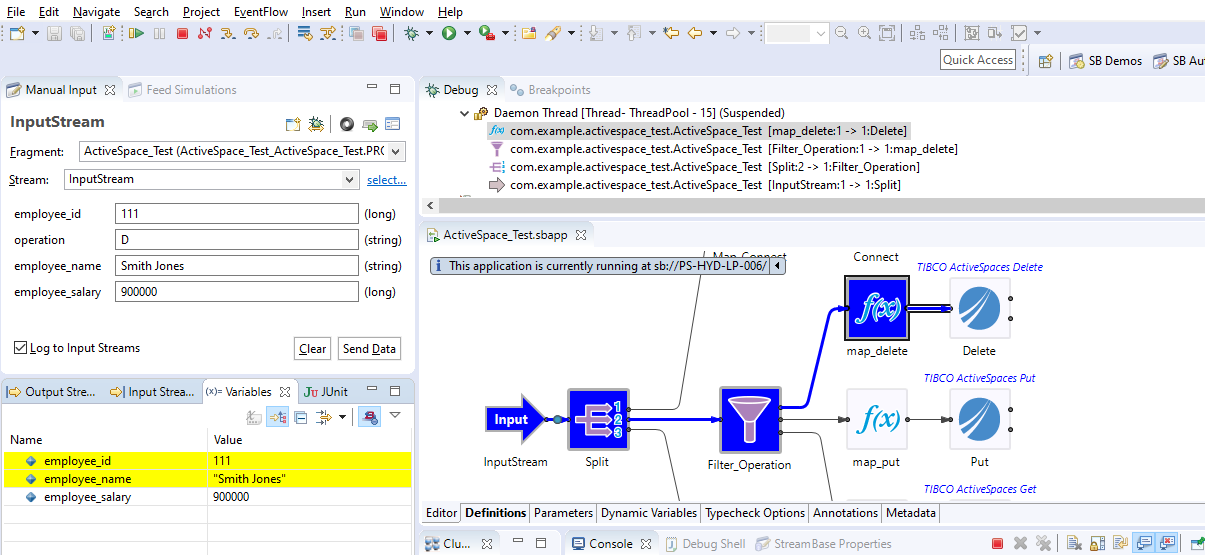
1.Test Delete

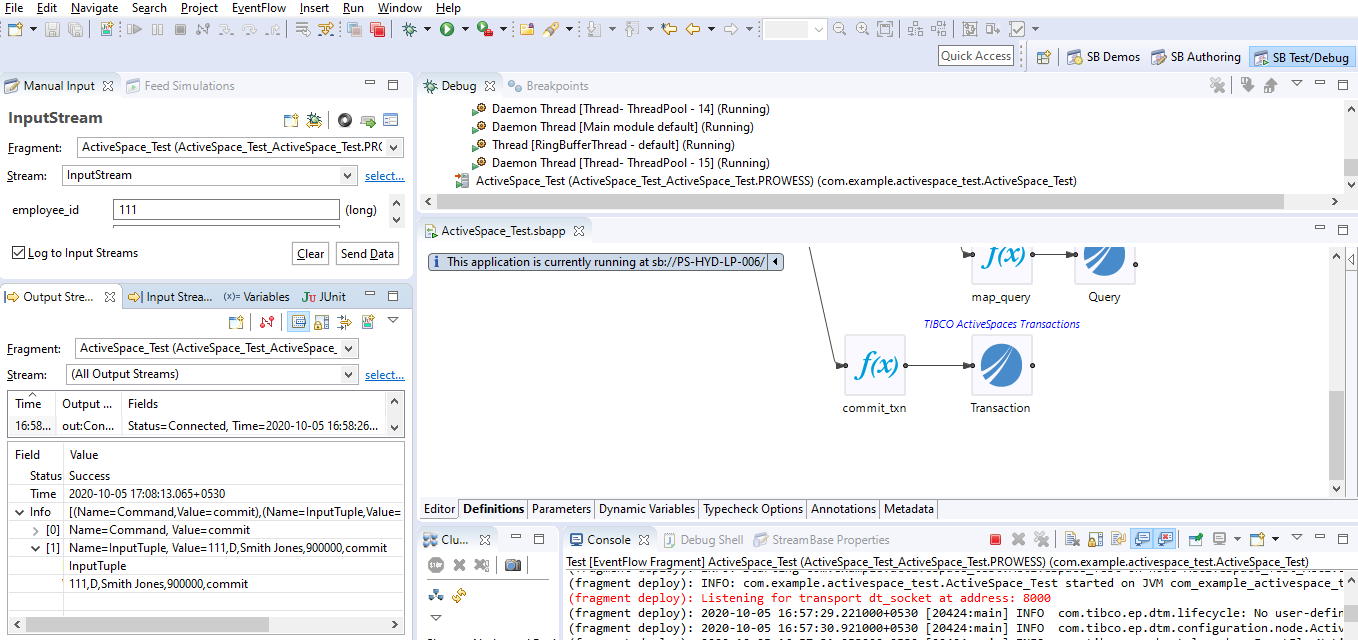
Insert a pre-existing record in Activespace local grid to delete from StreamBase.



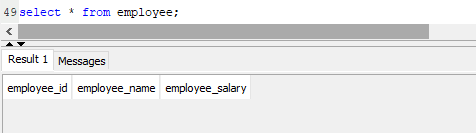
Pass manual input as below with Operation as ‘D’ ro delete the existing record.







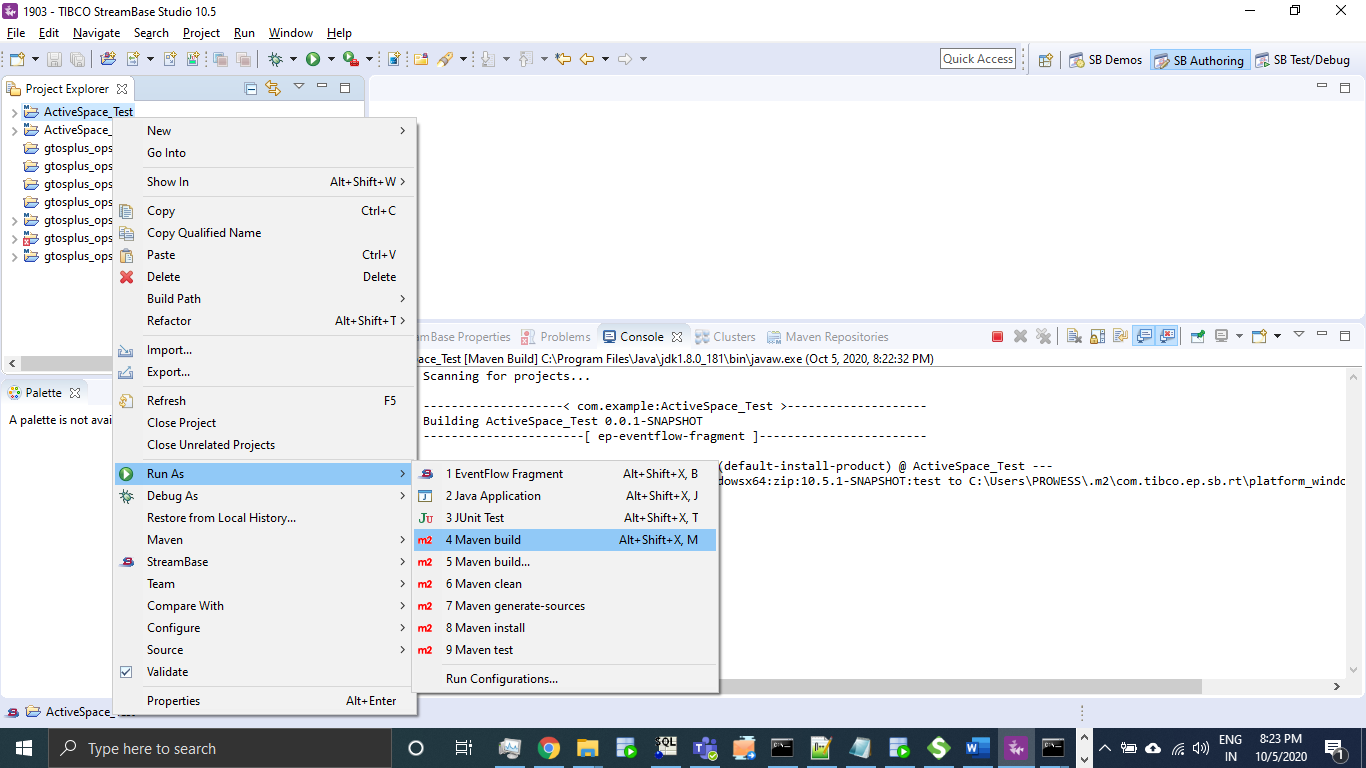
The record got deleted:



In the same manner based on operation provided Put and Insert will work.

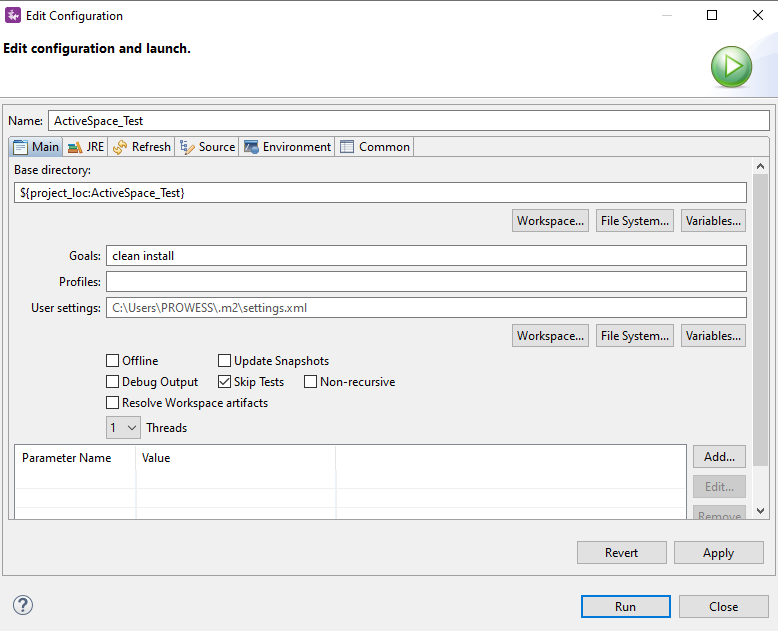
# 5.Create the application archive

Right click on project ActiveSpace\_Test and click on Run As - > Maven Build

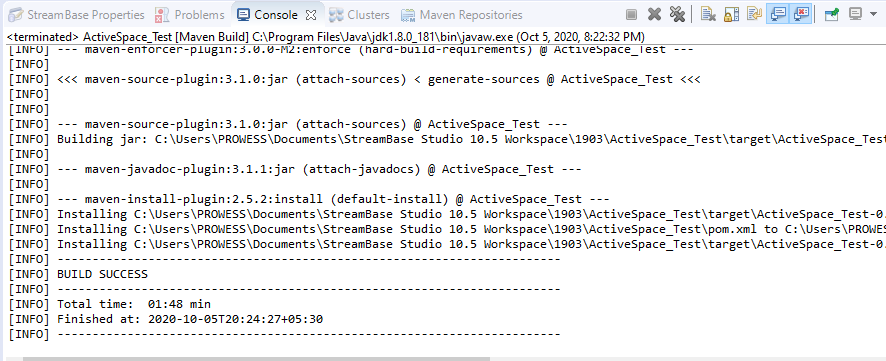


Set goal as **clean install** and check the option **skip Tests**

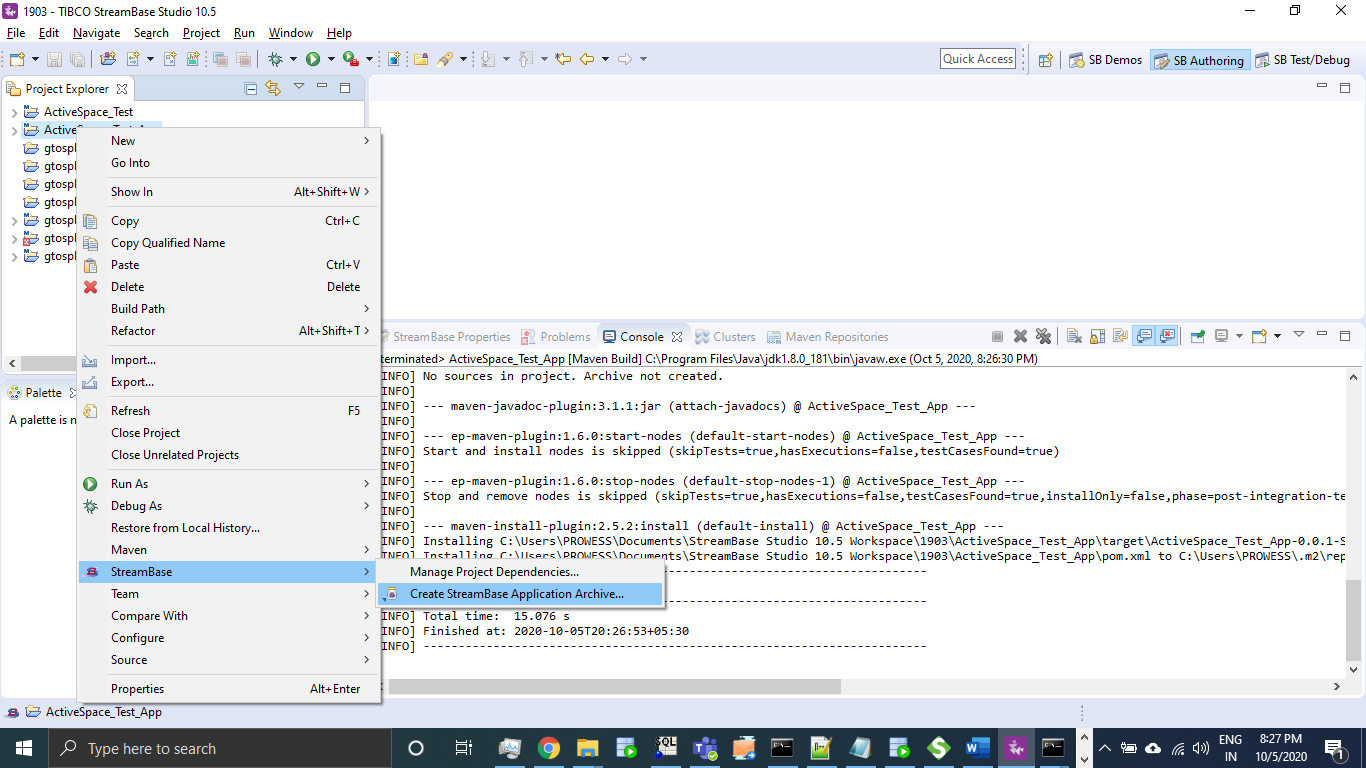
Click on Apply and Run.



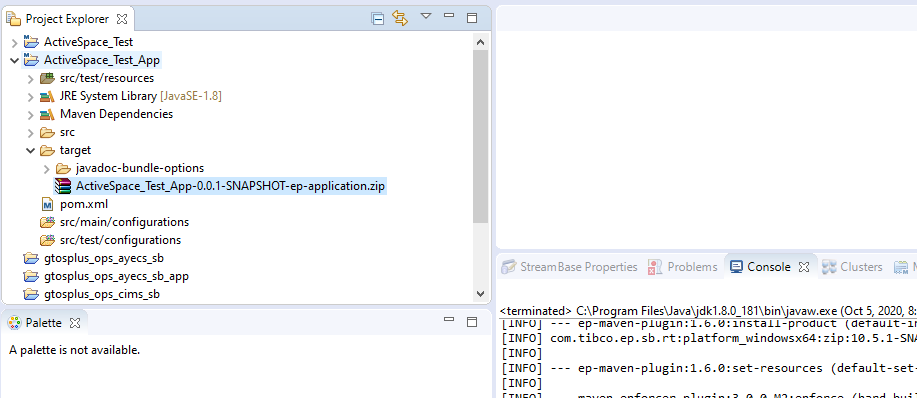
We will see as BUILD SUCCESS in Console.



Right Click on streambase application which is created in step 2: StreamBase -> Create StreamBase Application Archieve.



Under target folder for ActiveSpace\_Test\_App we can see a zip file (Archieve) got created.



# 6.Deploy and Run the Application:

Open the Streambase Command Prompt to Deploy and Run the application locally as shown in below screenshot.

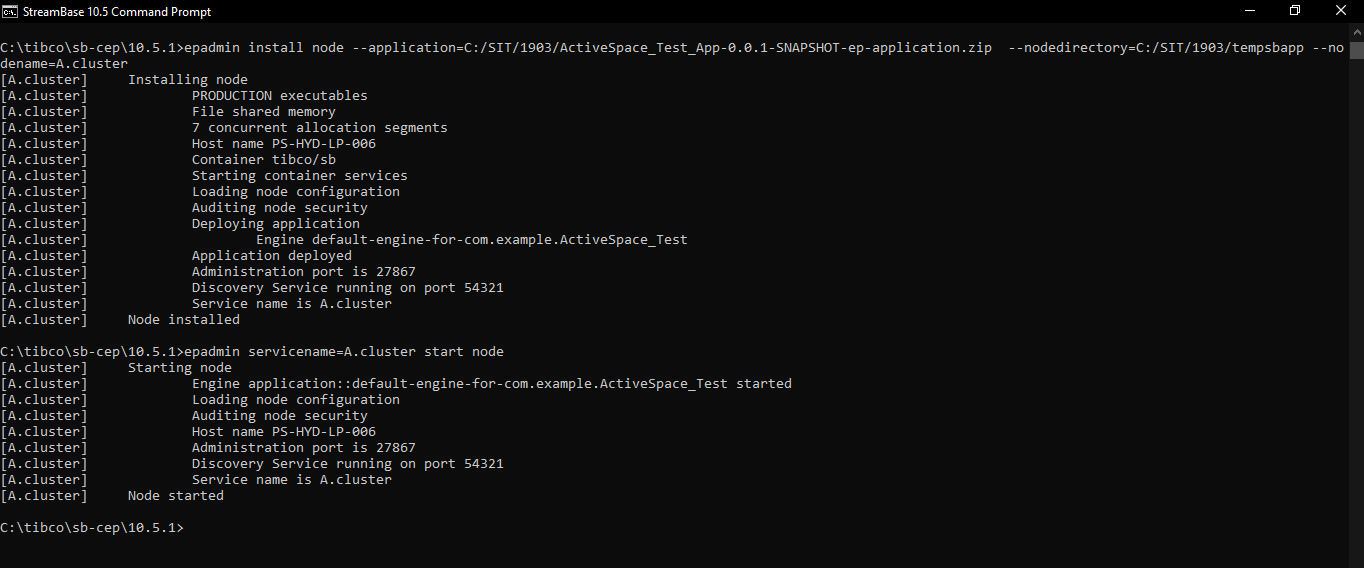


Open the Streambase Command Prompt to Deploy and Run the application locally as shown in below screenshot and run the below command where --application= " C:/SIT/1903/ActiveSpace\_Test\_App-0.0.1-SNAPSHOT-ep-application.zip” is the path of the zip file nodedirectory=C:/ SIT/1903/tempsbapp is any temp directory created in C drive and nodename=A.cluster is the any random name assigned to the node which will be running.

epadmin install node --application=C:/SIT/1903/ActiveSpace\_Test\_App-0.0.1-SNAPSHOT-ep-application.zip --nodedirectory=C:/SIT/1903/tempsbapp --nodename=A.cluster

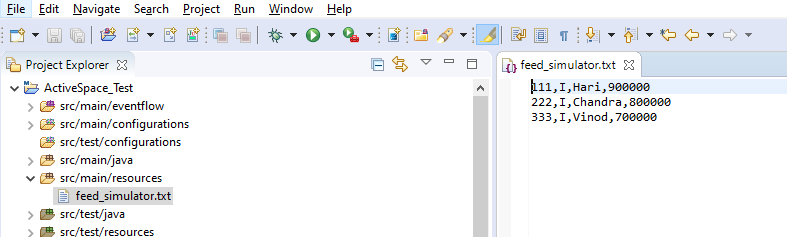
Now after your Node is installed, we can start, stop or remove node using below commands and once you have started the node you can test your application

epadmin servicename=A.cluster start node

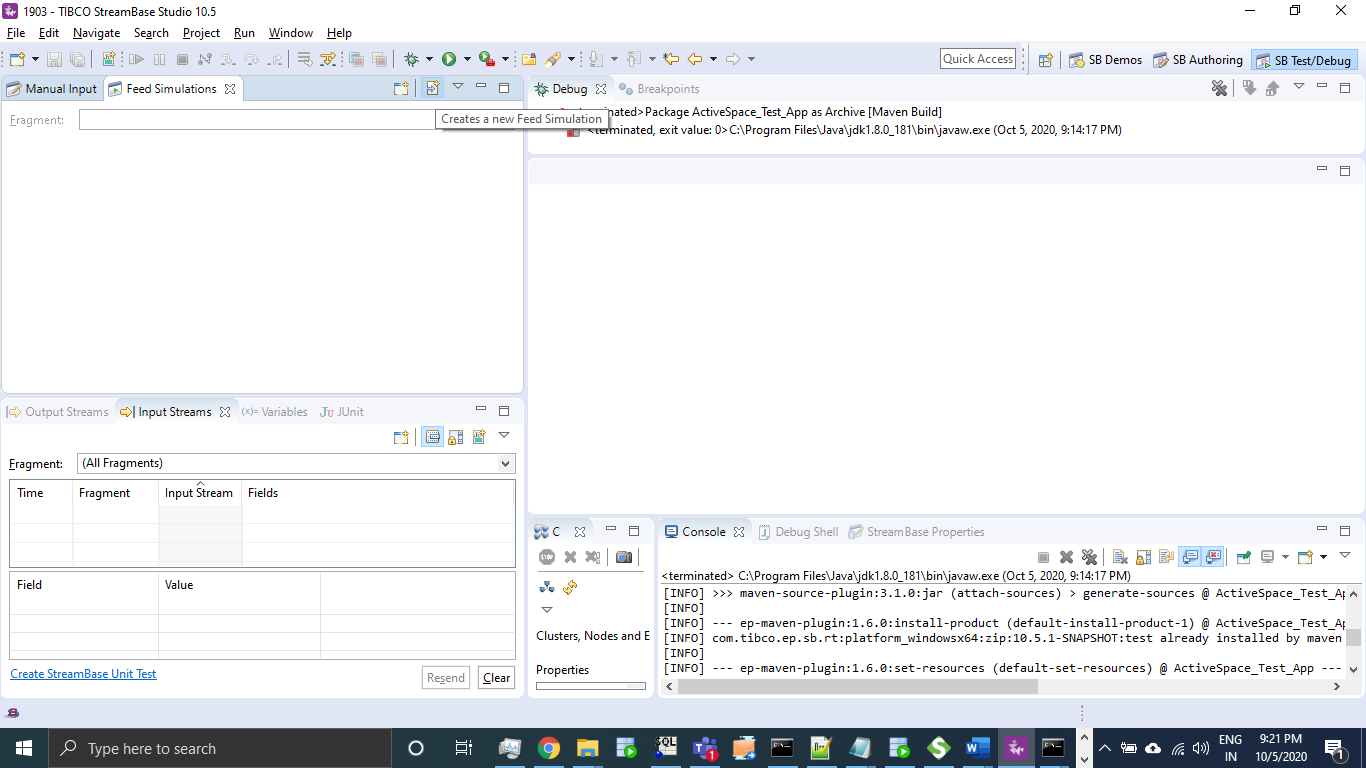


# 7. Test the application (Using Feed Simulator file from command line)

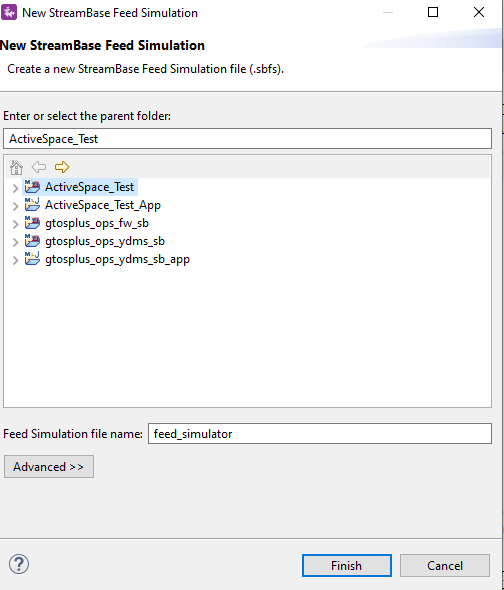
Import a feed\_simulator.txt file with below entries under src/main/resources.



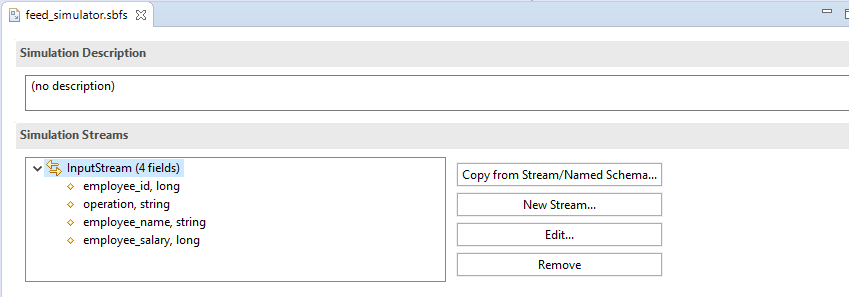
To Configure feed simulator file Navigate to Test/Debug mode, click on Feed Simulation Crate a new feed simulation file as below.



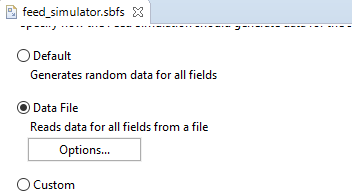
Select Project and mention file name as below.



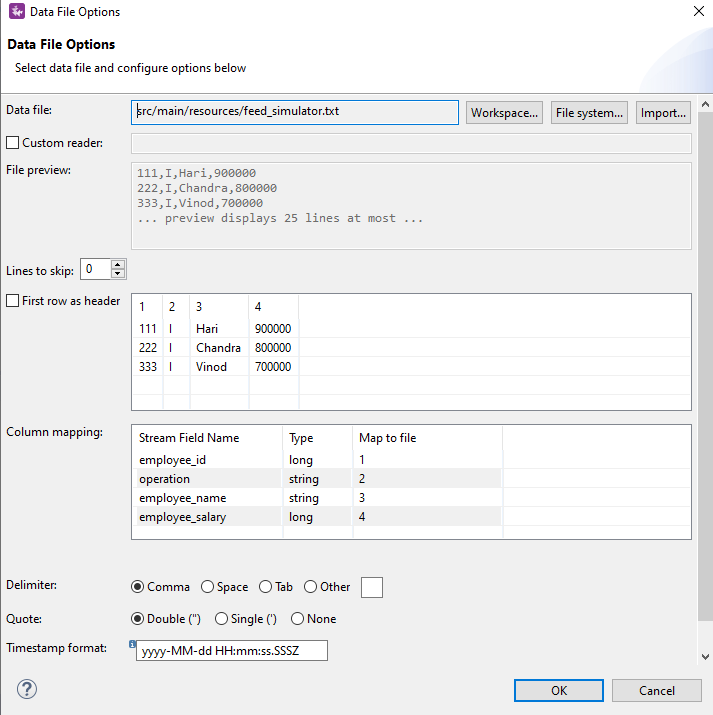
Create Input Stream and elements as mentioned below.



Select Data File option



Browse Text file imported to workspace and click on Ok.



Create Archive and Deploy the node as mentioned in steps 5 and step 6:

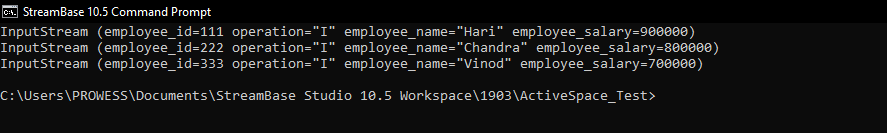
In StreamBase command prompt navigate to ActiveSpace\_Test project workspace location.

Cd C:\Users\PROWESS\Documents\StreamBase Studio 10.5 Workspace\1903

Use sbfeedsim utility command and enter as below as

**sbfeedsim** feed\_simulator.sbfs

We can see Data from feed simulator file got processed and records got inserted as mentioned operation is ‘I’



Result: Records from feed simulator file was successfully got inserted into ActiveSpace (In memory data grid)

