TIBCO BusinessEvents™ Loadbalancer

### Introduction & Example

Software Release 5.0

# Contents

[Contents 2](#_Toc279595289)

[What is it? 3](#_Toc279595290)

[Why? 3](#_Toc279595291)

[How? 4](#_Toc279595292)

[Deployment 7](#_Toc279595293)

[Example - SimpleDemo 8](#_Toc279595294)

[Limitations 18](#_Toc279595295)

# What is it?

It is a new feature. It is a new way to deploy BusinessEvents applications that use the Cache OM. For simple applications, it does not require any additional coding.

# Why?

Until now, BE applications that consume events from EMS had no concept of data locality or sticky sessions.

When a cluster of BE engines were consuming events from EMS, related events like Customer, Order and Shipment would be delivered to engines in an arbitrary fashion. This reduced the effectiveness of the L1Cache greatly, since it had to make version checks with the distributed cache to get the latest version modified in another engine.

This also meant that expensive distributed locks had to be acquired to prevent related events in different engines from modifying the same concepts.

Overall, this resulted in reduced throughput due to excessive network communication from cache writes, reads and locks. Distributed locks also meant that threads across multiple engines could get blocked while waiting on a lock.

The Loadbalancer alleviates this problem by routing all events related on some field/key to the same engine all the time. This is somewhat akin to what JMS Selectors offer but without the hassle of static partitioning.

# How?

Loadbalancing occurs at the Destination level. A destination has to be setup to receive and route events based on a routing key. The routing key is a field in the event(s) being sent over the Destination.

2 sets of engines are required – router and receiver. An Inference Agent Class with no rules can serve as a router. The router will only have the destinations earmarked for loadbalancing.

The receiver will have the receive-side of the loadbalanced destinations. It can also have other destinations and rules (or queries).

Any number of receivers and routers can be configured for fault tolerance and availability. The loadbalancing/partitioning will occur transparently.

### Without Loadbalancer

EMS

BE cache agent

BE inference agent

BE inference agent

Sender

### With Loadbalancer

EMS

BE cache agent

BE inference agent

BE inference agent

Sender

BE loadbalancer (inference agent)

BE loadbalancer (inference agent)

Router

Receiver

# Deployment

It is very easy to set up the cluster to use the Loadbalancer.

* You just have to create a new Inference Agent Class
* It looks just like the Inference Agent Class that you have been using so far, but without any Rules
* We will call the new loadbalancing inference agent class as the router
* The old inference agent class will be called the receiver
* The router has the all loadbalanced destinations that the receiver also has
* Each agent class has to be setup with certain properties per destination, that specify which destinations are loadbalanced

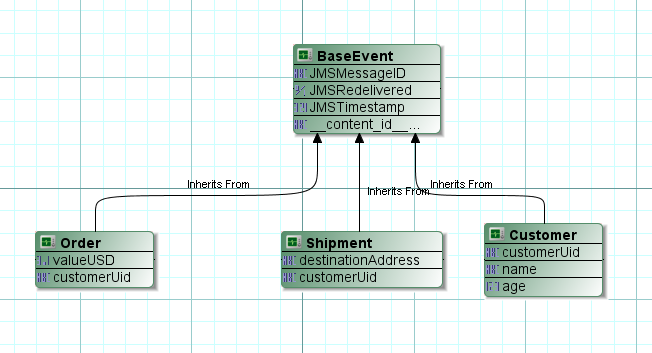
(The same applies to Query Agent Class)

# Example - SimpleDemo

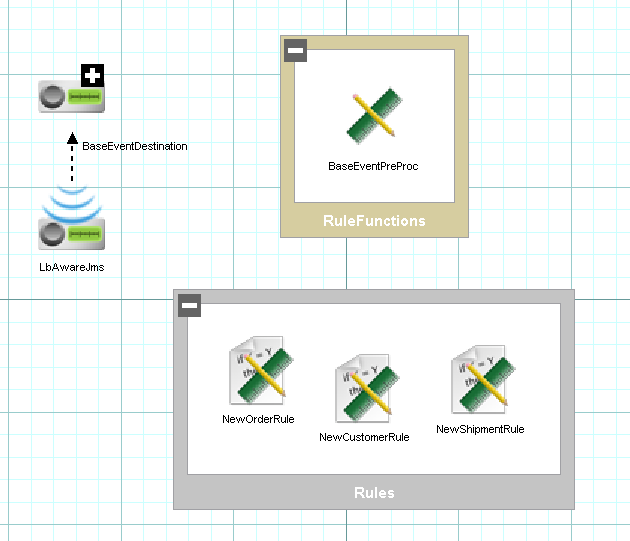
The example is a simple, pre-packaged application set up to demonstrate the loadbalancing feature.

## Studio Project

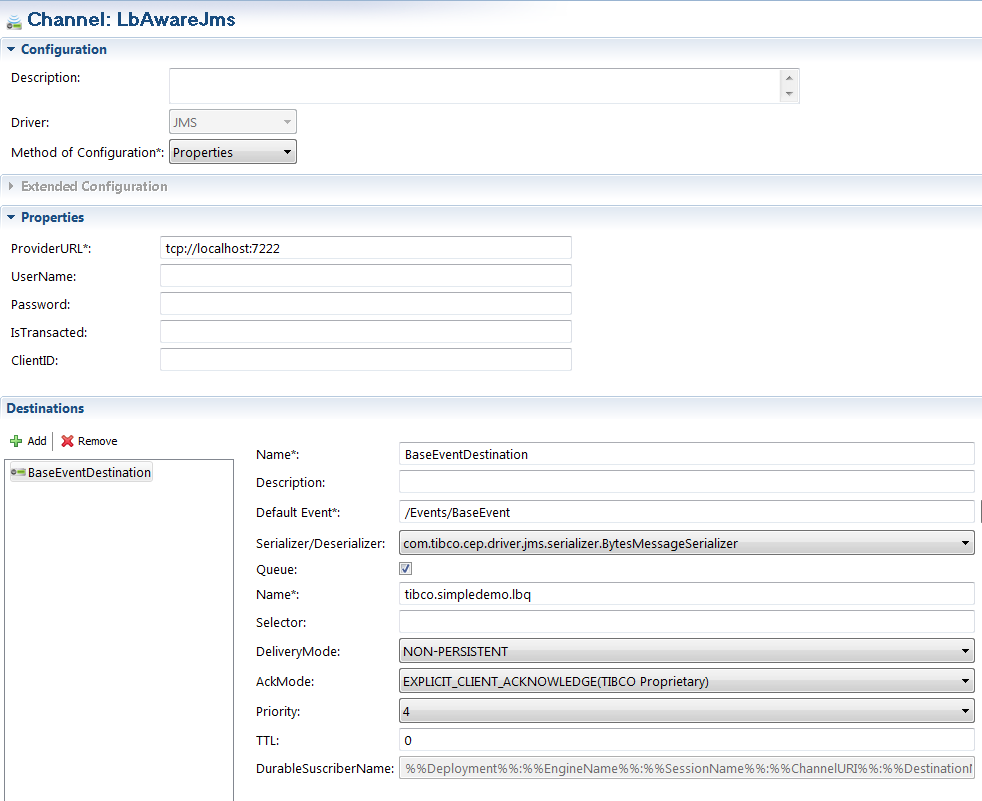
### Events



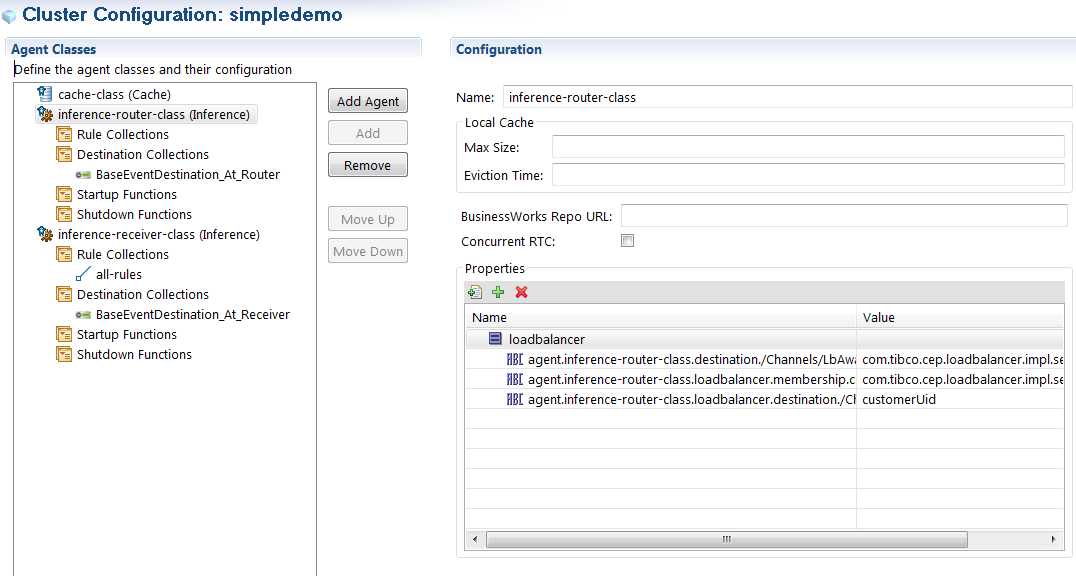
### Business Logic



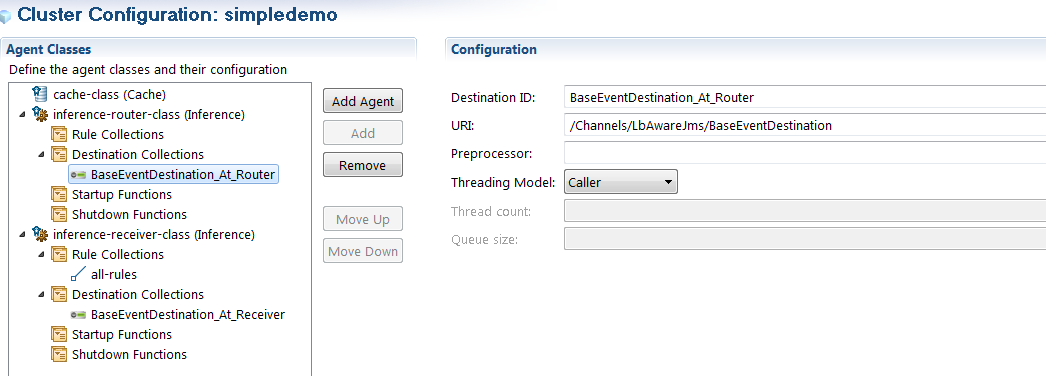
### Channel and Destination



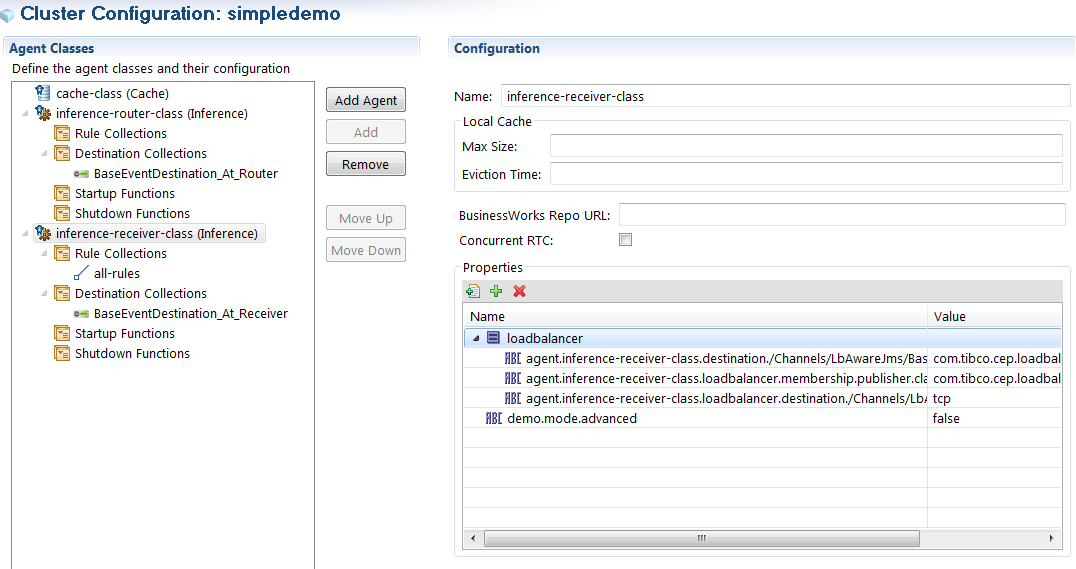
### Router Agent Class



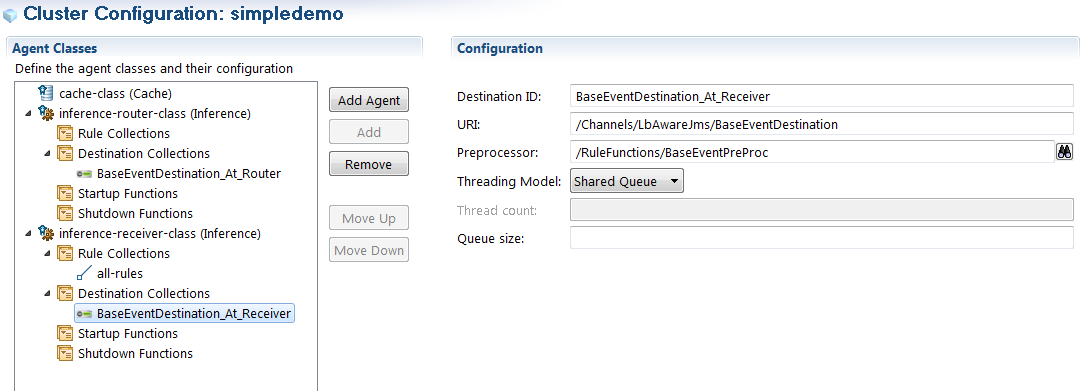
### Router Agent Destination



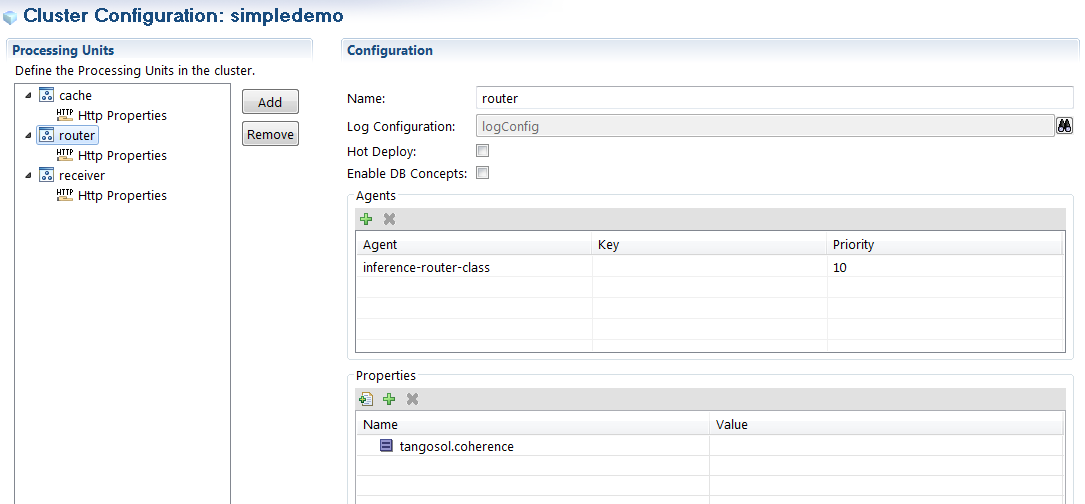
### Receiver Agent Class



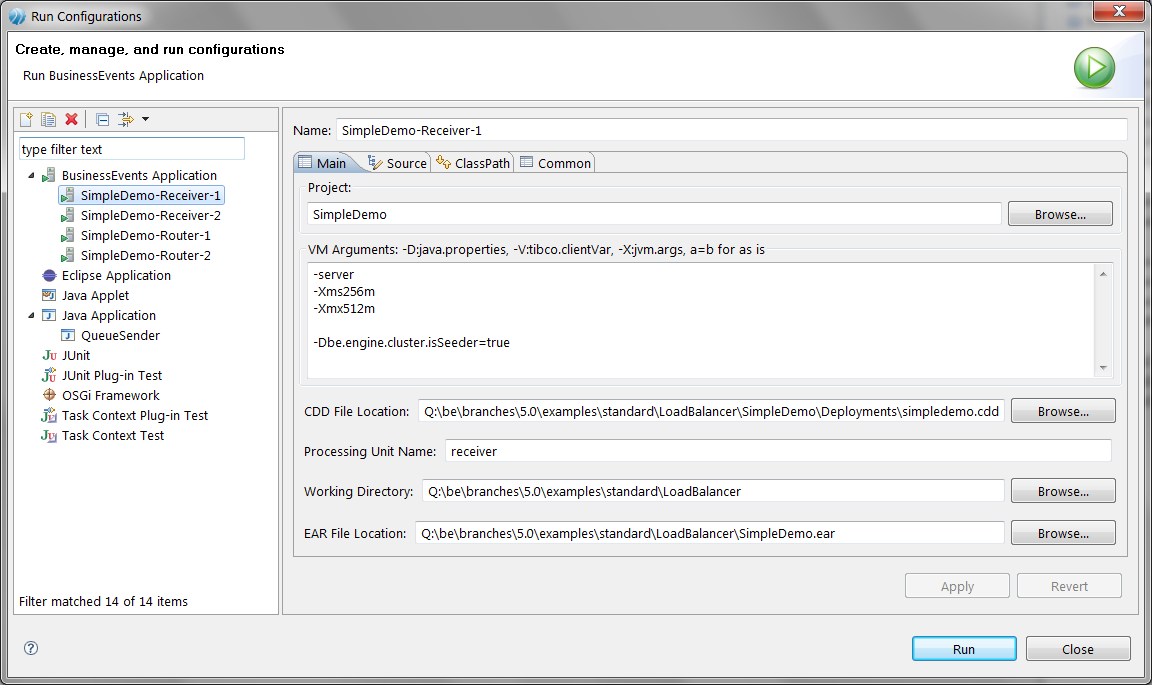
### Receiver Agent Destination & Pre-Processor

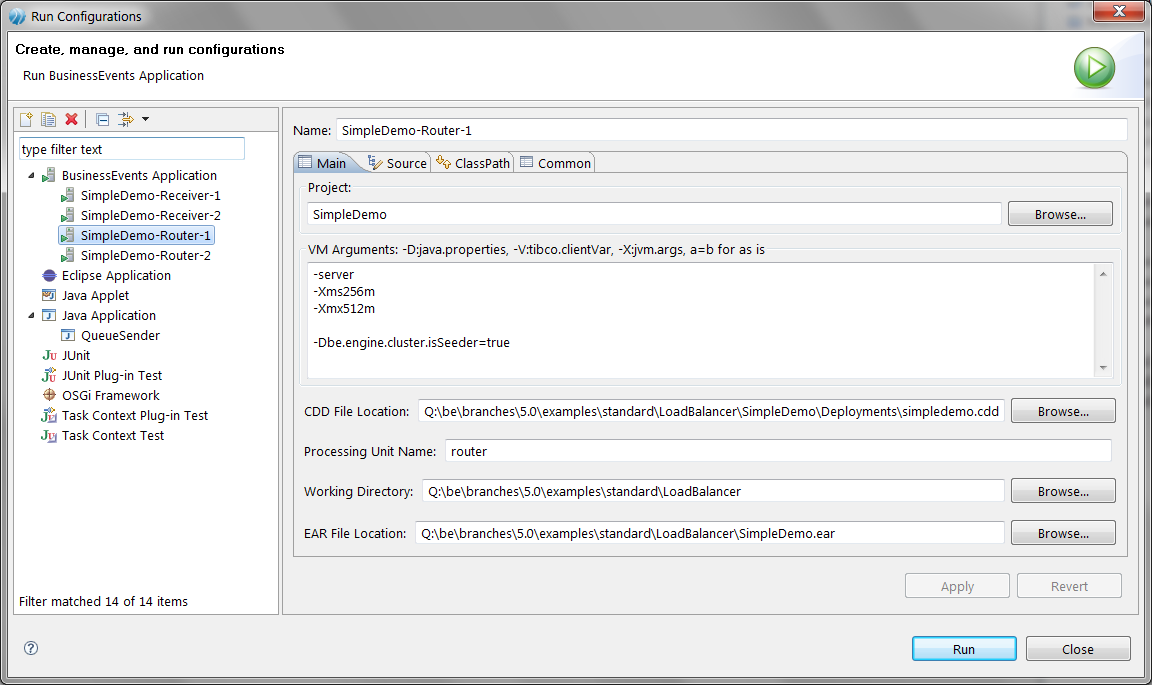


### Processing Units



## Running the agents

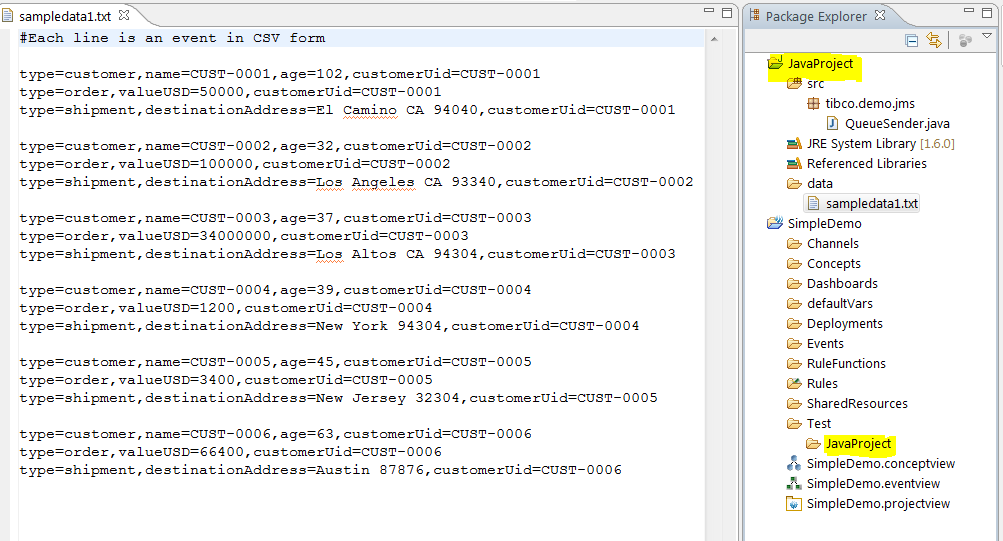




## Sending sample events

The example comes with a Java project that can be used to send sample events. It reads sample events from a text file which are then sent over JMS to the Inference Agents.

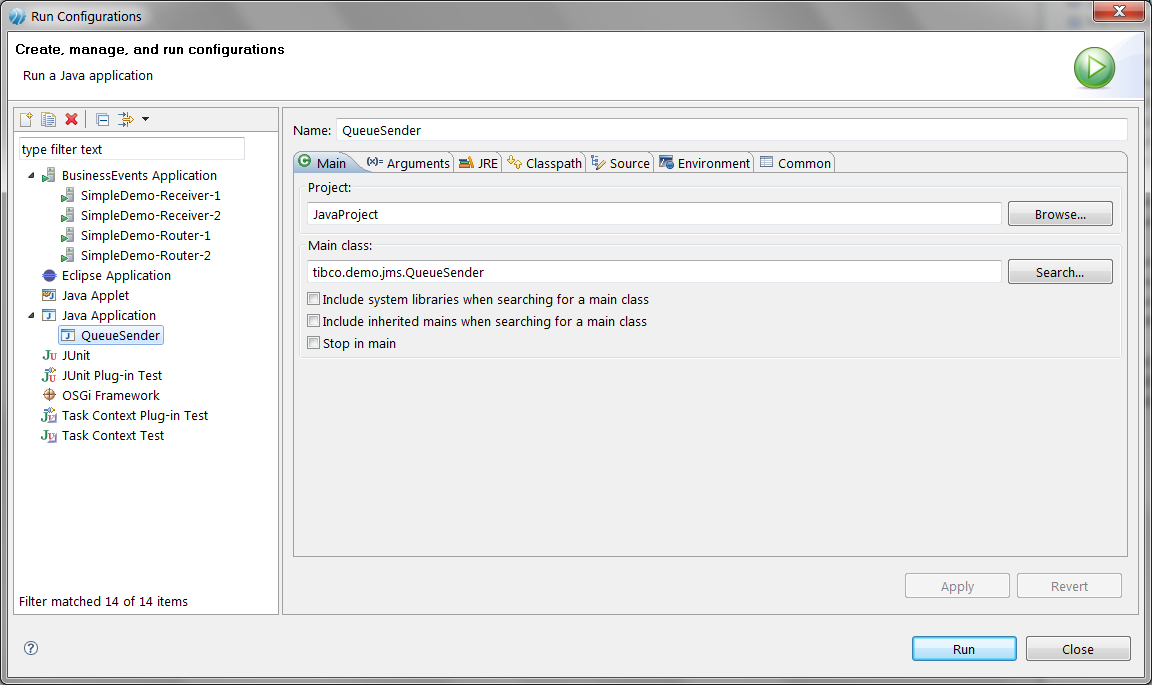
The package can be opened in the Java perspective.

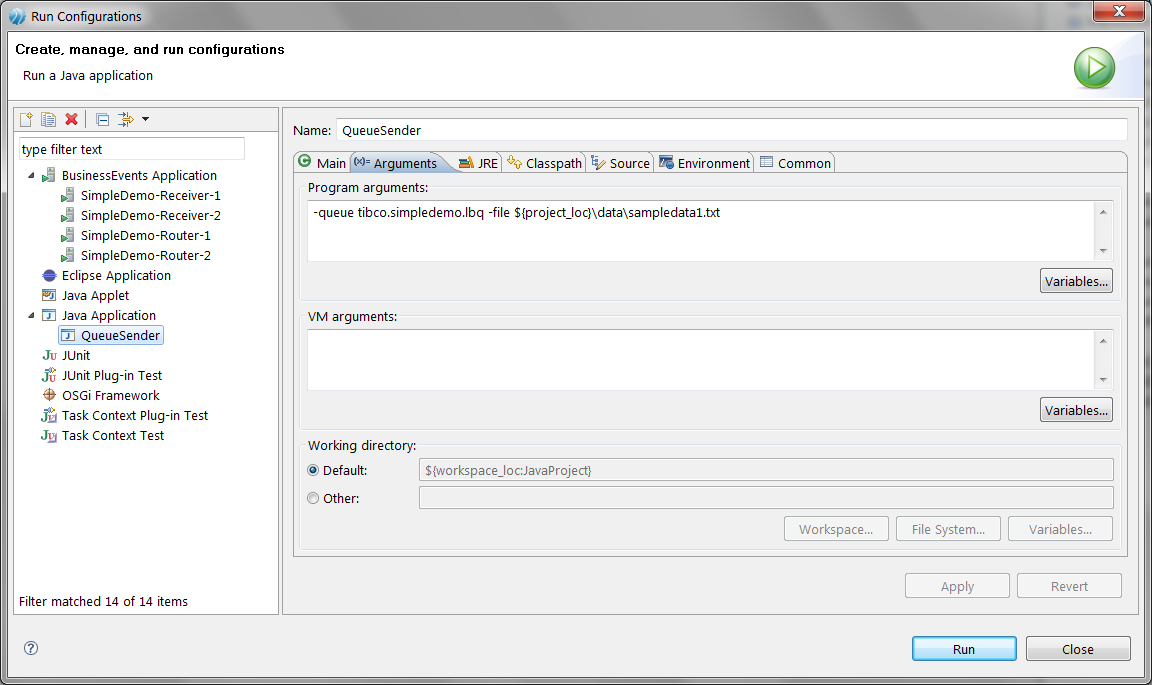


### QueueSender

By running this program the sample data in the text file will be sent as events to the EMS server.

### 





## Results

Once all the receivers and routers have started, the QueueSender program can be used to send triads of Customer, Order and Shipment events with the same customerUid. The routing key is the customerUid. As a result all the related events will be delivered to the same receiver regardless of how many routers are present.

At least 2 receivers should be running to see entire triads being sent to one or the other engine but never both.

Routing groups (here – Customer, Order, shipment triad) may break when a receiver goes down or a new one comes up. Once the topology stabilizes, the routing will also stabilize.

|  |
| --- |
| Receiver 1 logs |
| Info [$default.be.mt$.Thread.1] - [user] [inference-receiver-class] Received new customer:  ExtId /events/customer/1291844365177.261059767893065  CustomerUid CUST-0001  Name CUST-0001  Age 102  Info [$default.be.mt$.Thread.2] - [user] [inference-receiver-class] Received new order:  ExtId /events/order/1291844365187.261059778363700  CustomerUid CUST-0001  ValueUSD 50000.0  Info [$default.be.mt$.Thread.3] - [user] [inference-receiver-class] Received new shipment:  ExtId /events/shipment/1291844365189.261059779833326  CustomerUid CUST-0001  DestinationAddress El Camino CA 94040  Info [$default.be.mt$.Thread.4] - [user] [inference-receiver-class] Received new customer:  ExtId /events/customer/1291844365190.261059780855339  CustomerUid CUST-0002  Name CUST-0002  Age 32  Info [$default.be.mt$.Thread.6] - [user] [inference-receiver-class] Received new shipment:  ExtId /events/shipment/1291844365192.261059783213003  CustomerUid CUST-0002  DestinationAddress Los Angeles CA 93340  Info [$default.be.mt$.Thread.5] - [user] [inference-receiver-class] Received new order:  ExtId /events/order/1291844365191.261059781878893  CustomerUid CUST-0002  ValueUSD 100000.0  Info [$default.be.mt$.Thread.7] - [user] [inference-receiver-class] Received new customer:  ExtId /events/customer/1291844365196.261059786671731  CustomerUid CUST-0004  Name CUST-0004  Age 39  Info [$default.be.mt$.Thread.8] - [user] [inference-receiver-class] Received new order:  ExtId /events/order/1291844365197.261059787583895  CustomerUid CUST-0004  ValueUSD 1200.0  Info [$default.be.mt$.Thread.9] - [user] [inference-receiver-class] Received new shipment:  ExtId /events/shipment/1291844365197.261059788265579  CustomerUid CUST-0004  DestinationAddress New York 94304 |
| Receiver 2 logs |
| Info [$default.be.mt$.Thread.1] - [user] [inference-receiver-class] Received new customer:  ExtId /events/customer/1291844365193.261059784100527  CustomerUid CUST-0003  Name CUST-0003  Age 37  Info [$default.be.mt$.Thread.2] - [user] [inference-receiver-class] Received new order:  ExtId /events/order/1291844365194.261059784894628  CustomerUid CUST-0003  ValueUSD 3.4E7  Info [$default.be.mt$.Thread.3] - [user] [inference-receiver-class] Received new shipment:  ExtId /events/shipment/1291844365195.261059785789853  CustomerUid CUST-0003  DestinationAddress Los Altos CA 94304  Info [$default.be.mt$.Thread.5] - [user] [inference-receiver-class] Received new order:  ExtId /events/order/1291844365199.261059789834789  CustomerUid CUST-0005  ValueUSD 3400.0  Info [$default.be.mt$.Thread.6] - [user] [inference-receiver-class] Received new shipment:  ExtId /events/shipment/1291844365200.261059790501074  CustomerUid CUST-0005  DestinationAddress New Jersey 32304  Info [$default.be.mt$.Thread.7] - [user] [inference-receiver-class] Received new customer:  ExtId /events/customer/1291844365201.261059792021518  CustomerUid CUST-0006  Name CUST-0006  Age 63  Info [$default.be.mt$.Thread.8] - [user] [inference-receiver-class] Received new order:  ExtId /events/order/1291844365202.261059792836151  CustomerUid CUST-0006  ValueUSD 66400.0  Info [$default.be.mt$.Thread.9] - [user] [inference-receiver-class] Received new shipment:  ExtId /events/shipment/1291844365203.261059793527589  CustomerUid CUST-0006  DestinationAddress Austin 87876  Info [$default.be.mt$.Thread.4] - [user] [inference-receiver-class] Received new customer:  ExtId /events/customer/1291844365198.261059789107932  CustomerUid CUST-0005  Name CUST-0005  Age 45 |

# Limitations

* Only TIBCO EMS is supported
* Queues only
* EXPLICIT\_CLIENT\_ACKNOWLEDGE or EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE is strongly advised
* Events with attachments/payloads are not supported

**∞∞∞**