

Appendix: Example for part 1

Example Code – How to run?

For your benefit, the assignment also includes an example application code in the *bgu.spl.mics.example* package. You can use the example code to check your framework in part 1 and even as a reference for part 2. This code is only for testing - it deliberately does not follow this work guideline. You may read the code yourself and understand what it does (google is your friend). No questions about how this code works will be answered.

To run the example code once you complete implementing the framework, run the class `ExampleManager` which waits for one of the following commands:

- `start <service-type> <service-name> <service-args>...`
starts a service of type `<service-type>`, supplies it with the name `<service-name>` and the arguments `<service-args>....`. The supported services are:
 - `ev-handler` - Expects a single numeric argument which we will refer to as its "mbt" (messages before termination). The command will start the `ExampleEventHandlerService` which subscribes to the `ExampleEvent` message via the Message-Bus and completes delivered events. After completing "mbt" events the Micro-Service will terminate itself.
 - `brod-listener` - Expects a single numeric argument which we will refer to as its "mbt" (messages before termination). The command will start the `ExampleBroadcastListenerService` which subscribe to the `ExampleBroadcast` message. After receiving "mbt" messages the Micro-Service will terminate itself.
 - `sender` - Expects a single string argument "mode" which can either be "broadcast" or "event". The command will start the `ExampleMessageSenderService` which. In case the "mode" selected was "event", it will publish a single `ExampleEvent`, wait for its completion and then terminate. In the case where the "mode" selected was "broadcast", the sender will publish a single `ExampleBroadcast` and then terminate.
- `quit`
will ungracefully terminate the program.

Example of a simple input-output test using the example micro-services and messages:

Example manager is started - supported commands are: start,quit

Supporting services: [sender, ev-handler, brod-listener]

```
➤ start sender sender1 event
Sender sender1 started
Time has elapsed, no services has resolved the event – terminating
➤ start brod-listener brod1 2
Listener brod1 started
➤ start ev-handler hand1 1
Event Handler hand1 started
➤ start ev-handler hand2 2
Event Handler hand2 started
➤ start sender sender2 broadcast
Sender sender2 started
Sender sender2 publish a broadcast and terminated
```

Listener brod1 got a new message from sender2! (mbt: 1)

➤ start sender sender2 event

Sender sender2 started

Event Handler hand1 got a new event from sender2! (mbt: 0)

Event Handler hand1 terminating.

Completed processing the event, its result is "Hello from hand1" – success

➤ start sender sender3 event

Sender sender3 started

Event Handler hand2 got a new event from sender3! (mbt: 1)

Completed processing the event, its result is "Hello from hand2" – success

➤ start sender sender4 broadcast

Sender sender4 started

Sender sender4 publish a broadcast and terminated

Listener brod1 got a new message from sender4! (mbt: 0)

Listener brod1 terminating.

➤ start sender sender5 event

Sender sender5 started

Event Handler hand2 got a new event from sender5! (mbt: 0)

Event Handler hand2 terminating.

Completed processing the event, its result is "Hello from hand2" – success

➤ quit

Manager Terminating - UNGRACEFULLY!