

# Execution Flow Diagram for ExampleSimulationProgram.java



main() method controls program invocation

- a. Handle command-line arguments
- b. New ExampleSimulationProgram(), initialize()
- c. runSimulationLoops()
- d. tearDownNetworkInterface()
- e. System.exit()

a. handleArguments()

Read, save new network address/port combination (if any, use default values if unspecified)  
Provide warning usage message otherwise

b. constructor new ExampleSimulationProgram()

Initialize DIS channel parameters  
Initialize simulation entities of interest  
Join DIS channel for live network connection  
Send CommentPdu as initial message

initialize()

initializeDisChannel()

Initialize network interface  
Setup PDU recorder

initializeSimulationEntities()

Create PDU factory  
Define each model of interest including names, platform IDs, network identifiers

Join DIS channel  
Send CommentPDU as initial announcement

d. disChannel.tearDownNetworkInterface()

Clears all buffers  
Closes pduRecorder  
Shuts down threads  
Releases network resources

e. System.exit() // quit



c. runSimulationLoops() runs each timestep loop of a simulation

Initialize loop counters, announce commencement on network

```
while (simulationLoopCount > MAX) // or other condition
Increment counters and timestep state variables
// =====
// * your own simulation code goes here! *****

Are you listening for DIS PDUs from networked simulation channel?

Compute, set state variable whether termination condition is met
// =====

Send outgoing DIS PDU messages to network channel for this loop

Thread sleep for real-time duration of simulation timestep

If termination condition met, break out of loop. Otherwise continue.
```