**Test C.1: Passthrough Mapper**

1. The rationale behind the test; i.e., what is it testing and why we care.

This is testing the creation of a passthrough mapper and assigning it to a position sensor, and then getting it’s value.

We care about this because it shows that the passthrough mapper creation is working and that it gets assigned to a sensor.

2. A general English description of the initial conditions of the test.

The initial conditions are that there is no mapper, no sensor, no actuators, and no reporters. The mapper must be made before the sensor, the actuators need to be made before the reporter, and the reporter needs to be made before the sensor.

3. The commands for (2), which must appear in a standalone form that could be directly copied into a text file to reproduce the test without manual intervention. Do not cross-reference other tests.

@CONFIGURE LOG \"a.txt\" DOT SEQUENCE \"b.txt\" NETWORK \"c.txt\" XML \"d.txt\"

CREATE MAPPER myMapper EQUATION PASSTHROUGH

CREATE SENSOR POSITION mySensor1 MAPPER myMapper

CREATE CONTROLLER FORWARDING myController1 WITH COMPONENTS mySensor1

GET SENSOR mySensor1 VALUE

SET SENSOR mySensor1 VALUE 10

GET SENSOR mySensor1 VALUE

@exit

4. A brief English narrative of the expected results of executing the test. (Proper testing discipline expects that you do this before running the test.)

The expected results are for the GET SENSOR command to retrieve the value of the sensor and output it to the screen once before we set the value to see what the default is and once after to see how the mapper changed it, if it did. The expected result from passthrough is that the mapper doesn’t change the value at all so the value should be 0 at first, then 10 after it gets set.

5. At least one representation of the actual results. The form is your choice.

Welcome to your ParserHelper

Welcome to your Startup class

PARSE> @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

SCHEDULE | @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

PARSE> CREATE MAPPER myMapper EQUATION PASSTHROUGH

SCHEDULE | CREATE MAPPER myMapper EQUATION PASSTHROUGH

PARSE> CREATE SENSOR POSITION mySensor1 MAPPER myMapper

SCHEDULE | CREATE SENSOR POSITION mySensor1 MAPPER myMapper

PARSE> CREATE CONTROLLER FORWARDING myController1 WITH COMPONENTS mySensor1

SCHEDULE | CREATE CONTROLLER FORWARDING myController1 WITH COMPONENTS mySensor1

PARSE> GET SENSOR mySensor1 VALUE

SCHEDULE | GET SENSOR mySensor1 VALUE

PARSE> SET SENSOR mySensor1 VALUE 10

SCHEDULE | SET SENSOR mySensor1 VALUE 10

PARSE> GET SENSOR mySensor1 VALUE

SCHEDULE | GET SENSOR mySensor1 VALUE

PARSE> @exit

SCHEDULE | @exit

TIME | 0.02

EXECUTE | @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

EXECUTE | CREATE MAPPER myMapper EQUATION PASSTHROUGH

EXECUTE | CREATE SENSOR POSITION mySensor1 MAPPER myMapper

EXECUTE | CREATE CONTROLLER FORWARDING myController1 WITH COMPONENTS mySensor1

EXECUTE | GET SENSOR mySensor1 VALUE

The value of Identifier{name=mySensor1} is 0.0

EXECUTE | SET SENSOR mySensor1 VALUE 10

EXECUTE | GET SENSOR mySensor1 VALUE

The value of Identifier{name=mySensor1} is 10.0

EXECUTE | @exit

EXITING |

6. A brief discussion on how the actual results differ from the expected results.

The value was not affected by the mapper at all, which is expected as the passthrough mapper wasn’t supposed to change the value.

7. A suggestion for how to extend this test to cover related aspects not required here

Reporter or Watchdog could be added in to further test how mapper alters the value.

**Task C.2: Scaled Mapper**

1. The rationale behind the test; i.e., what is it testing and why we care.

This is testing the creation of a scaled mapper and assigning it to a position sensor, and then getting it’s value.

We care about this because it shows that the scaled mapper creation is working and that it gets assigned to a sensor.

2. A general English description of the initial conditions of the test.

The initial conditions are that there is no mapper, no sensor, no actuators, and no reporters. The mapper must be made before the sensor, the actuators need to be made before the reporter, and the reporter needs to be made before the sensor.

3. The commands for (2), which must appear in a standalone form that could be directly copied into a text file to reproduce the test without manual intervention. Do not cross-reference other tests.

@CONFIGURE LOG \"a.txt\" DOT SEQUENCE \"b.txt\" NETWORK \"c.txt\" XML \"d.txt\"

CREATE MAPPER myMapper2 EQUATION SCALE 10

CREATE SENSOR POSITION mySensor2 MAPPER myMapper2

CREATE CONTROLLER FORWARDING myController2 WITH COMPONENTS mySensor2

GET SENSOR mySensor2 VALUE

SET SENSOR mySensor2 VALUE 1

GET SENSOR mySensor2 VALUE

@exit

4. A brief English narrative of the expected results of executing the test. (Proper testing discipline expects that you do this before running the test.)

The expected results are for the GET SENSOR command to retrieve the value of the sensor and output it to the screen once before we set the value to see what the default is and once after to see how the mapper changed it, if it did. The expected result is that the default value is 0, and then after we set the value to 1, it should be 10 as we are scaling it by 10. So the value should be multiplied by 10.

5. At least one representation of the actual results. The form is your choice.

Welcome to your ParserHelper

Welcome to your Startup class

PARSE> @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

SCHEDULE | @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

PARSE> CREATE MAPPER myMapper2 EQUATION SCALE 10

SCHEDULE | CREATE MAPPER myMapper2 EQUATION SCALE 10

PARSE> CREATE SENSOR POSITION mySensor2 MAPPER myMapper2

SCHEDULE | CREATE SENSOR POSITION mySensor2 MAPPER myMapper2

PARSE> CREATE CONTROLLER FORWARDING myController2 WITH COMPONENTS mySensor2

SCHEDULE | CREATE CONTROLLER FORWARDING myController2 WITH COMPONENTS mySensor2

PARSE> GET SENSOR mySensor2 VALUE

SCHEDULE | GET SENSOR mySensor2 VALUE

PARSE> SET SENSOR mySensor2 VALUE 1

SCHEDULE | SET SENSOR mySensor2 VALUE 1

PARSE> GET SENSOR mySensor2 VALUE

SCHEDULE | GET SENSOR mySensor2 VALUE

PARSE> @exit

SCHEDULE | @exit

TIME | 0.02

EXECUTE | @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

EXECUTE | CREATE MAPPER myMapper2 EQUATION SCALE 10

EXECUTE | CREATE SENSOR POSITION mySensor2 MAPPER myMapper2

EXECUTE | CREATE CONTROLLER FORWARDING myController2 WITH COMPONENTS mySensor2

EXECUTE | GET SENSOR mySensor2 VALUE

The value of Identifier{name=mySensor2} is 0.0

EXECUTE | SET SENSOR mySensor2 VALUE 1

EXECUTE | GET SENSOR mySensor2 VALUE

The value of Identifier{name=mySensor2} is 10.0

EXECUTE | @exit

EXITING |

6. A brief discussion on how the actual results differ from the expected results.

The value of Sensor was unchanged at 0 as 0\*10 = 0, and when it was changed to 1 then it became 10, which was expected.

7. A suggestion for how to extend this test to cover related aspects not required here

Reporter or Watchdog could be added in to further test how mapper alters the value.

**Task C.3: Normalized Mapper**

1. The rationale behind the test; i.e., what is it testing and why we care.

This is testing the creation of a normalized mapper and assigning it to a position sensor, and then getting it’s value.

We care about this because it shows that the normalized mapper creation is working and that it gets assigned to a sensor.

2. A general English description of the initial conditions of the test.

The initial conditions are that there is no mapper, no sensor, no actuators, and no reporters. The mapper must be made before the sensor, the actuators need to be made before the reporter, and the reporter needs to be made before the sensor.

3. The commands for (2), which must appear in a standalone form that could be directly copied into a text file to reproduce the test without manual intervention. Do not cross-reference other tests.

@CONFIGURE LOG \"a.txt\" DOT SEQUENCE \"b.txt\" NETWORK \"c.txt\" XML \"d.txt\"

CREATE MAPPER myMapper3 EQUATION NORMALIZE 10 20

CREATE SENSOR POSITION mySensor3 MAPPER myMapper3

CREATE CONTROLLER FORWARDING myController3 WITH COMPONENTS mySensor3

GET SENSOR mySensor3 VALUE

SET SENSOR mySensor3 VALUE 15

GET SENSOR mySensor3 VALUE

@exit

4. A brief English narrative of the expected results of executing the test. (Proper testing discipline expects that you do this before running the test.)

The expected results are for the GET SENSOR command to retrieve the value of the sensor and output it to the screen once before we set the value to see what the default is and once after to see how the mapper changed it, if it did. The expected value at the default is 0 as 0 lies outside the bounds of 10 and 20 so it is 0. After it’s set to 15 it should return a value of 50 as 50 is halfway between 10 and 20.

5. At least one representation of the actual results. The form is your choice.

Welcome to your ParserHelper

Welcome to your Startup class

PARSE> @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

SCHEDULE | @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

PARSE> CREATE MAPPER myMapper3 EQUATION NORMALIZE 10 20

SCHEDULE | CREATE MAPPER myMapper3 EQUATION NORMALIZE 10 20

PARSE> CREATE SENSOR POSITION mySensor3 MAPPER myMapper3

SCHEDULE | CREATE SENSOR POSITION mySensor3 MAPPER myMapper3

PARSE> CREATE CONTROLLER FORWARDING myController3 WITH COMPONENTS mySensor3

SCHEDULE | CREATE CONTROLLER FORWARDING myController3 WITH COMPONENTS mySensor3

PARSE> GET SENSOR mySensor3 VALUE

SCHEDULE | GET SENSOR mySensor3 VALUE

PARSE> SET SENSOR mySensor3 VALUE 15

SCHEDULE | SET SENSOR mySensor3 VALUE 15

PARSE> GET SENSOR mySensor3 VALUE

SCHEDULE | GET SENSOR mySensor3 VALUE

PARSE> @exit

SCHEDULE | @exit

TIME | 0.02

EXECUTE | @CONFIGURE LOG "a.txt" DOT SEQUENCE "b.txt" NETWORK "c.txt" XML "d.txt"

EXECUTE | CREATE MAPPER myMapper3 EQUATION NORMALIZE 10 20

EXECUTE | CREATE SENSOR POSITION mySensor3 MAPPER myMapper3

EXECUTE | CREATE CONTROLLER FORWARDING myController3 WITH COMPONENTS mySensor3

EXECUTE | GET SENSOR mySensor3 VALUE

The value of Identifier{name=mySensor3} is 0.0

EXECUTE | SET SENSOR mySensor3 VALUE 15

EXECUTE | GET SENSOR mySensor3 VALUE

The value of Identifier{name=mySensor3} is 50.0

EXECUTE | @exit

EXITING |

6. A brief discussion on how the actual results differ from the expected results.

The actual results is that at the default it remained 0 then after we set it to 15 it turned into 50.0 which was the expected value.

7. A suggestion for how to extend this test to cover related aspects not required here.

Reporter or Watchdog could be added in to further test how mapper alters the value.