Chapter 1 Code Example

This folder contains the full source for the example shown in the first chapter of the tutorial.

The example shows how to create a simple application that produces TempSensorType information and another that consumes it.

Compiling the example

To compile the example you need:

- g++ v4.6 or higher (clang++ would also be fine)
- CMake v2.6 or higher
- OpenSplice DDS v6.4 or higher

Once you have installed the required tools simply do:

```
 $ cmake .
 $ make
```

Running the Examples

First start one or more temperature sensors (e.g. producers):

```
 $ ./tspub 0
```

This will start producing data.

Then start the consumer:

```
 $ ./tssub
 USAGE:
tssub <filter-expression>
```

Where the `filter-expression` should have the same syntax of a SQL WHERE clause. For instance, if you were interested in getting the data only when the temperature was higher than 28.0 C and the humidity higher than 0.65 then you could run the application as follows:

```
./tssub "temp > 28.0 AND hum > 0.65"
(id = 0, temp = 29.2647, hum = 0.669657, scale = C)
(id = 0, temp = 28.5219, hum = 0.742885, scale = C)
(id = 0, temp = 28.6268, hum = 0.665124, scale = C)
(id = 0, temp = 28.7411, hum = 0.664452, scale = C)
(id = 0, temp = 28.1886, hum = 0.670146, scale = C)
(id = 0, temp = 29.9059, hum = 0.726814, scale = C)
(id = 0, temp = 29.9059, hum = 0.726814, scale = C)
(id = 0, temp = 29.9059, hum = 0.726814, scale = C)
(id = 0, temp = 29.7783, hum = 0.730949, scale = C)
(id = 0, temp = 29.7783, hum = 0.730949, scale = C)
```

As you can see only the data matching the filter is being recevied in this case. You can try different filters to experiment and refine your understanding. Recall that the syntax that you can use is the same as the one of SQL WHERE statements.