
firstSteps Documentation

Release 1.8.0

Carlos Gomez Carrasco

Sep 17, 2019

FIRST STEPTS

1	Using eProxima Shapes Demo	3
1.1	Publishing a Shape	4
1.2	Subscribing to a Shape	6
1.3	Other Options	8
1.4	Endpoints and Output tabs	9
2	Discovery and basic connectivity	11
3	History and Durability	15
4	Partition	19
5	Redundancy and Fault Tolerance	23
6	Content Based Filter	27
7	Time Based Filter	31
8	TCP Transport	33
8.1	LAN configuration	33
8.2	WAN configuration	33
9	Troubleshooting	37
9.1	Two instances in the same computer	37
10	Interoperability	39

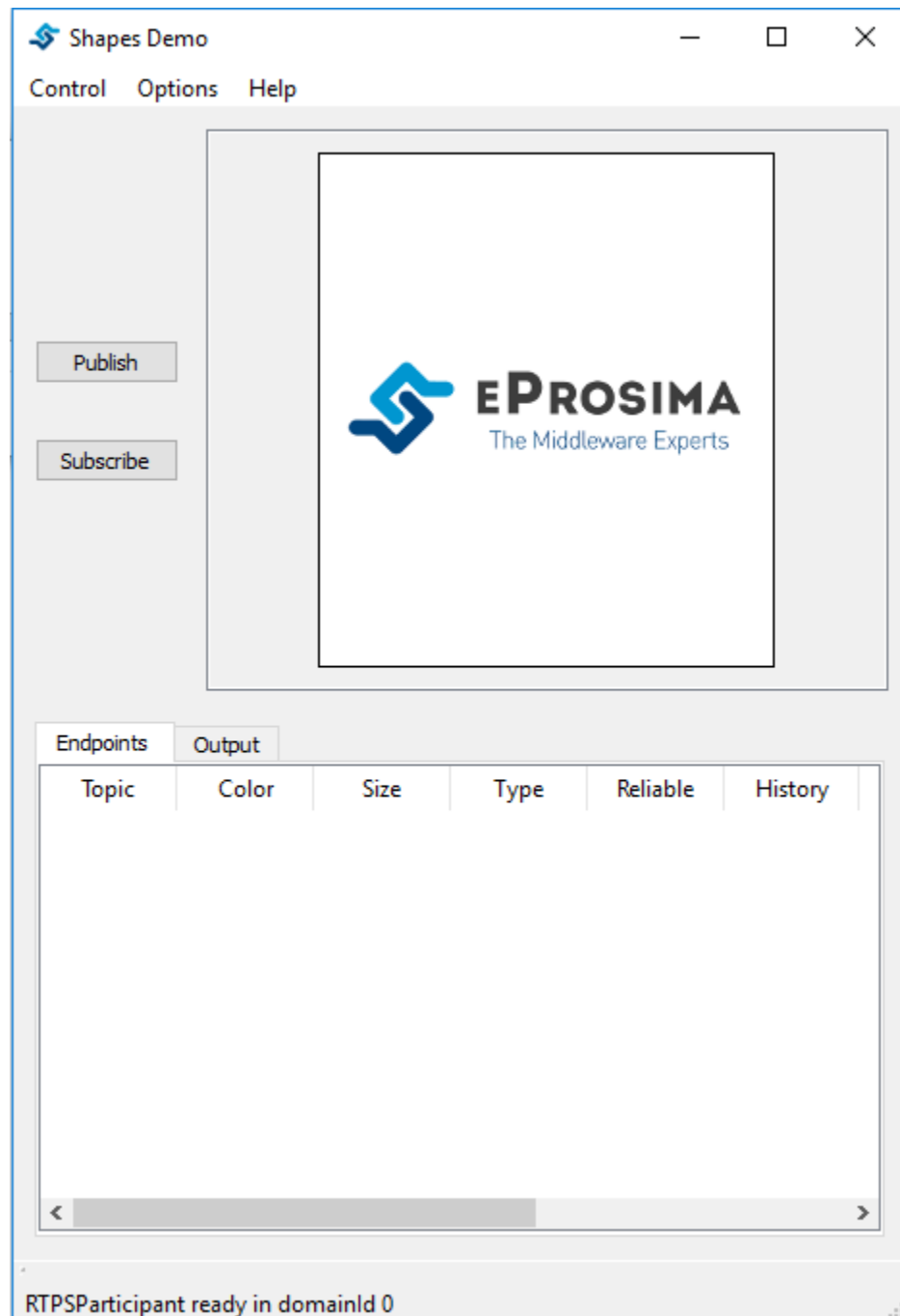
Shapes Demo is an application that Publishes and Subscribes to shapes of different colors and sizes moving on a board. Each Shape conforms its own topic: Square, Triangle or Circle. A single instance of the Shapes Demo can publish on or subscribe to several topics at a time.

It can be used to demonstrate the capabilities of eProxima Fast RTPS or as an interoperability demonstrator with other implementations of the RTPS protocol.

- *First Stepts*
- *Shapes Demo Examples*
- *Troubleshooting*

USING EPROSIMA SHAPES DEMO

This section guides you through the main menus of eProsimas Shapes Demo. After the executable is launched you should see a window similar to the one presented in the following image.



As you can see the main window is design for simplicity, with only two principal buttons situated on the left. There are additional menus that will be explained later.

1.1 Publishing a Shape

When the user presses the Publish button a new window appear to allow the user to define what type of Shape and which QoS the want to use in its publication. The following image shows an example of the Publication menu.

Publish

Shape: Square Color: RED Size: 30 Partition: ☐ A ☐ B ☐ C ☐ D ☐ *

QoS Settings

History: 1 ☒ Reliable

Durability: TRANSIENT_LOCAL

Liveliness

Kind: AUTOMATIC

Lease Duration (ms): INF

Announcement Period (ms): INF

Ownership

Kind: SHARED_OWNERSHIP

Ownership Strength: 0

Deadline

Duration (ms): INF

Lifespan

Duration (ms): INF

OK Cancel

There are multiple parameters that the user can define in this menu:

- **Shape:** This parameter defines the topic where the publication is going to occur. Three different shapes can be published: **Square**, **Circle** and **Triangle**.
- **Color:** The user can define the color of the Shape. This parameter will be used as key; that is, a way to distinguish between multiple instances of the same shape.
- **Size:** This parameter allows you to control how big the shape is. The size can vary between 1 and 99.

- **Partition:** The user can select different partitions to differentiate groups of publishers and subscribers. The user can select between four partitions (A, B, C and D). Additionally the user can select the “*” partition, that will be matched against all other partitions. The user should note, however, that using the *wildcard* (*) partition is not the same as not using any partition. A publisher that uses the wildcard partition will not be matched with a subscriber that uses no partitions.
- **Reliable:** The user can select to disable the Reliable check-box to use a Best-Effort Publisher.
- **History and Durability:** The History of the Publishers is set to **KEEP_LAST**. The user can select the number of samples that the Publisher is going to save. The user can also select whether this History is going to be **VOLATILE** or **TRANSIENT_LOCAL**. The latter will send that last stored values to subscribers joining after the Publisher has been created.
- **Liveliness:** The user can select the Liveliness Qos for the Publisher between three different values: **AUTOMATIC**, **MANUAL_BY_PARTICIPANT** and **MANUAL_BY_TOPIC**. The user can also set the Lease Duration value and in case of using **AUTOMATIC** or **MANUAL_BY_PARTICIPANT** also the Announcement Period.
- **Ownership:** The Ownership Qos determines whether the instance (color) of the Topic (Shape) is owned by a single Publisher. If the selected ownership is **EXCLUSIVE** the Publisher will use the Ownership strength value as the strength of its publication. Only the publisher with the highest strength can publish in the same Topic with the same Key.
- **Deadline:** The Deadline Qos determines the maximum expected amount of time between samples. When the deadline is missed the application will be notified and a message printed on the console.
- **Lifespan:** The Lifespan Qos determines the duration while the sample is still valid. When a sample’s lifespan expires, it will be removed from publisher and subscriber histories.

Note that using Lifespan QoS will not have any visual effect.

1.2 Subscribing to a Shape

When the user presses the Subscriber button a new window appear to allow the user to define what type of Shape and which QoS the want to use in its subscription. The following image shows an example of the Subscribe menu.

Subscribe

Shape: Square | Color: PURPLE | Size: 30 | Partition: ☐ A ☐ B ☐ C ☐ D ☐ *

QoS Settings

History: 6 | ☐ Reliable

Durability: VOLATILE

Liveliness

Kind: AUTOMATIC

Lease Duration (ms): INF

Ownership

Kind: SHARED_OWNERSHIP

Ownership Strength: 0

Deadline

Duration (ms): INF

Lifespan

Duration (ms): INF

Filters

☐ Time Based (ms): 0

☐ Content Based

OK Cancel

This menu is very similar to the Publication menu but the user cannot change the color and size, and it has additional elements:

- **Liveliness:** This Qos policy is applied in the same way as in the Publisher except for the Announcement Period, which doesn't apply for the Subscriber.
- **Time Based Filter:** This value can be used by the user to only specify the minimum amount of time (in

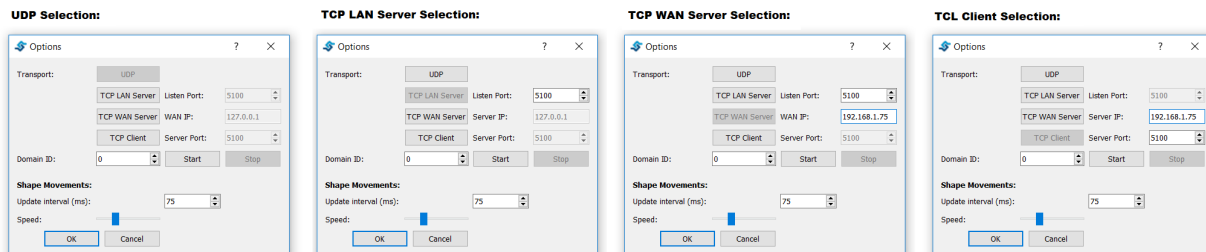
milliseconds) that the subscriber wants between updates.

- **Content Based Filter:** This filter draws a rectangle in the shapes window. Only the instances that are included in this rectangle are accepted. The rest of them are ignored. The user can dynamically resize and move this content filter.

Note that using Lifespan QoS will not have any visual effect.

1.3 Other Options

The eProsima Shapes Demo application allows the user to define additional options. To see the Options window, please click *Options->Preferences* in the main bar. The following image shows the Options Menu.



The user can customize several aspects of ShapesDemo operation:

- **Transport Protocol:** UDP is the default transport protocol for Fast RTPS but TCP protocol is available. In order to use TCP we must be aware of its point-to-point connection nature: one of the ShapesDemo instance must be a TCP server and all the others must be TCP clients. To use TCP follow the next steps:
 - Push the *Stop* button in order to end UDP use. This will automatically remove all publishers and subscribers from this instance.
 - To create a TCP LAN server push the corresponding button and fill the *Server Port* textbox with an available port where the application will be listening for incoming connections.
 - To create a TCP WAN server push the corresponding button:
 - * fill the *WAN IP* textbox with the public IPv4 router address.
 - * fill the *Server Port* textbox with an available TCP port where the application will be listening for incoming connections.

Note that the router NAT and computer firewall settings must allow external connections to the server port.

 - To create a TCP client push the corresponding button:
 - * fill the *Server IP* textbox with the IP address of the server.
 - * if client and server do not share the same net because the server is behind a NAT the WAN IP address of the server gateway must be specified.
 - * fill the *Server port* textbox with the corresponding server listening port.
 - Push the *Start* button in order to resume ShapesDemo operation.
- **Domain ID:** The user can select different Domain Ids. Shapes Demo instances using different Domain Ids will not communicate. To modify the Domain ID the user needs to stop the participant (thus removing all existing Publishers and Subscribers) and start a new one with the new Domain Id.
- **Update interval:** This value changes the publication period for all the Publishers.

- **Speed:** This scrollbar allows the user to change how much the Shape moves between two write calls.

1.4 Endpoints and Output tabs

A table including all created endpoints is also provided. An example of this legend is shown in the following image.

Endpoints		Output								
	Topic	Color	Size	Type	Reliable	History	Partitions	Ownership	Durability	Liveliness
1	Square	RED	30	Pub	True	6	-	SHARED	TRANSIENT	AUTOMATIC
2	Triangle	CYAN	30	Pub	True	6	-	SHARED	TRANSIENT	AUTOMATIC
3	Circle	GRAY	30	Pub	True	6	-	SHARED	VOLATILE	AUTOMATIC
4	Circle	---	---	Sub	False	6	-	SHARED	VOLATILE	AUTOMATIC

The user can use this table to remove endpoints. Two methods are provided:

- Right click in an endpoint: An option to remove the endpoint is shown.
- Pressing the delete button when the endpoint is selected.

An example of the output tab is shown in the next figure.

Endpoints	Output
<pre>[11:49:13.997]: RTPSParticipant ready in domainId 0 [11:49:46.266]: Publisher created in topic: RED Square [11:50:38.268]: Publisher created in topic: CYAN Triangle [11:50:57.571]: Publisher created in topic: GRAY Circle [11:53:25.364]: Subscriber created in topic: Circle</pre>	

DISCOVERY AND BASIC CONNECTIVITY

In Fast RTPS, the discovery task is automatic. Fast RTPS performs the task of finding the relevant information and distributing it to its destination. It means that new nodes are automatically discovered by any other in the network.

In this test we have to launch three Publishers and three Subscribers. At the end, you will see two additional squares in each instance, mirroring the movements of the original square in real-time.

Step-by-Step

First, you have to create three publishers:

1 - Create a red square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance1)
- Click on Publish.
- Select SQUARE option for Shape and RED for Color.

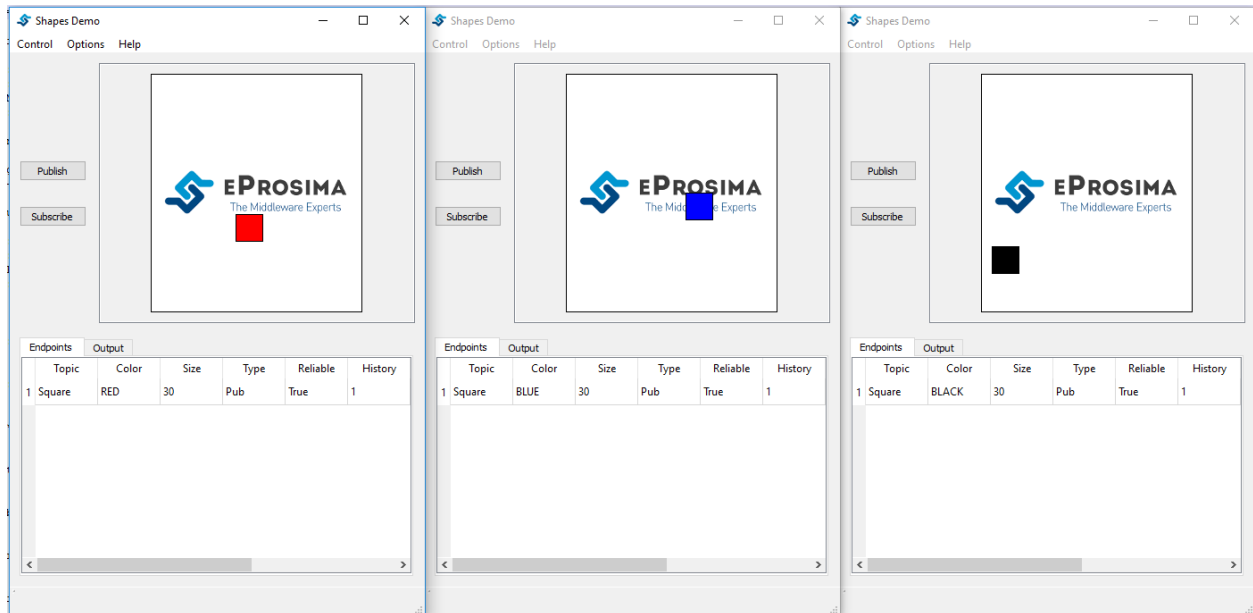
2 - Create a blue square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance2)
- Click on Publish.
- Select SQUARE option for Shape and BLUE for Color.

3 - Create a black square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance3)
- Click on Publish.
- Select SQUARE option for Shape and BLACK for Color.

Your windows should look similar to the following image.



Second, you have to create three subscribers:

1 - Click Subscribe on Instance1.

- Select SQUARE option for Shape.
- Change the History field from 6 to 1.

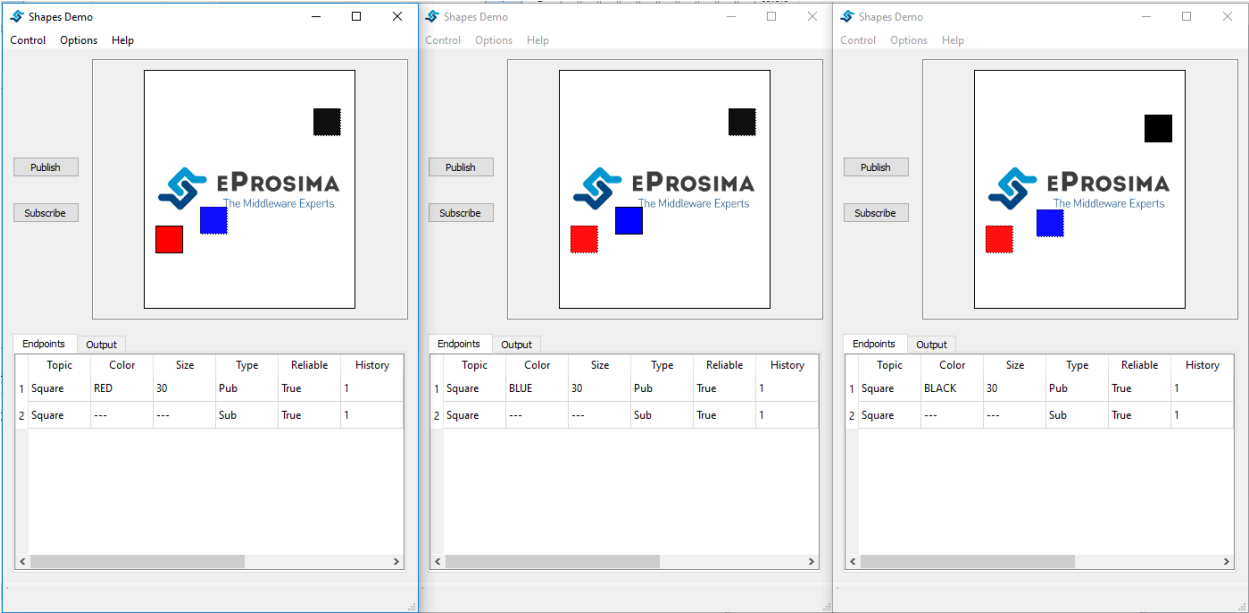
2 - Click Subscribe on Instance2.

- Select SQUARE option for Shape.
- Change the History field from 6 to 1.

3 - Click Subscribe on Instance3.

- Select SQUARE option for Shape.
- Change the History field from 6 to 1.

Your windows should look similar to the following image.



HISTORY AND DURABILITY

Fast RTPS allows Publishers to send data before Subscriber comes. All these publications are stored in the history of each Publisher, therefore, when a subscriber arrives it has two options, to start from scratch or to request all previous publications.

The history configuration determines the number of data that will keep in the send queue. The History of the Publishers is set to `KEEP_LAST` in this example and we have two options to the “Durability” concept, `VOLATILE` and `TRANSIENT_LOCAL`. If you select `VOLATILE`, the previous data samples will not be sent, on the contrary, if you select `TRANSIENT_LOCAL`, the n previous data samples will be sent to the late-joining subscriber.

In this test we have to launch three Publisher and three Subscriber in `TRANSIENT_LOCAL` mode.

Finally, you will see 100 red squares on Instance2 and Instance3, mirroring the movements of the red square in the publisher of Instance1. The leading square indicates the current position of the published square.

Step-by-Step

First, we have to launch three instances and create a Publisher in each of them:

1 - Create a red square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance1)
- Click on Publish.
- Select `SQUARE` option for Shape and `RED` for Color.
- Change the History field from 6 to 100.
- Select `TRANSIENT_LOCAL`.

2 - Create an orange square publisher:

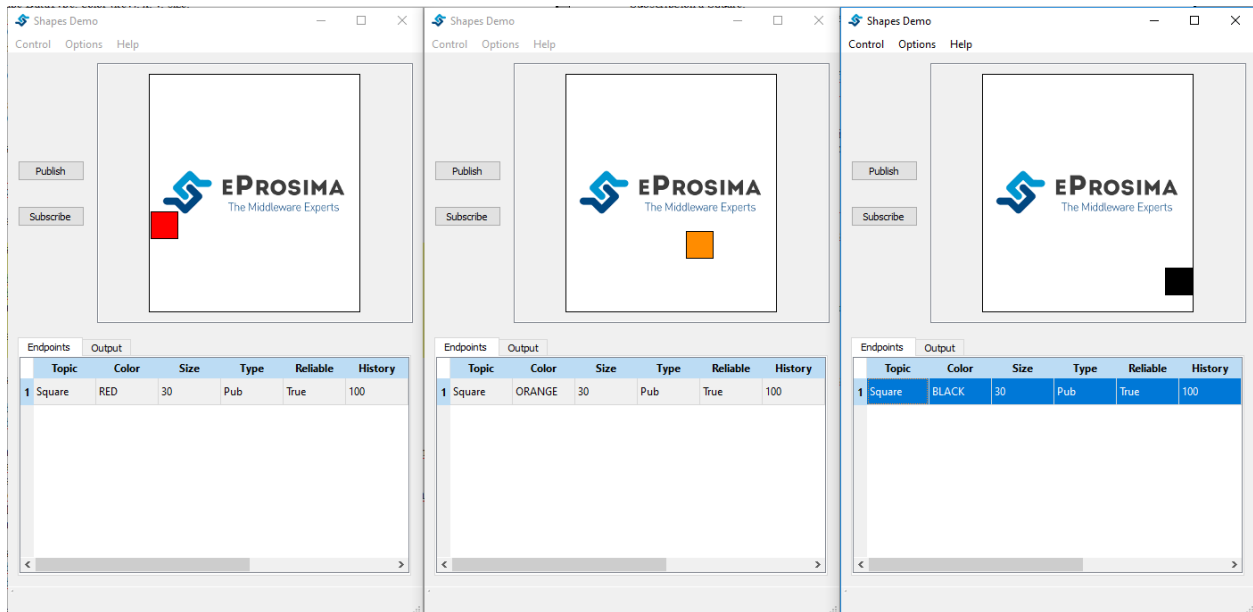
- Start eProsima Shapes-Demo. (We will refer to this instance as Instance2)
- Click on Publish.
- Select `SQUARE` option for Shape and `ORANGE` for Color.
- Change the History field from 6 to 100.
- Select `TRANSIENT_LOCAL`.

3 - Create a black square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance3)
- Click on Publish.
- Select `SQUARE` option for Shape and `BLACK` for Color.
- Change the History field from 6 to 100.

- Select TRANSIENT_LOCAL.

Your windows should look similar to the following image.



Second, we create a Subscriber at each instance.

4 - Click Subscribe on Instance1.

- Select SQUARE option for Shape.
- Change the History field from 6 to 100.

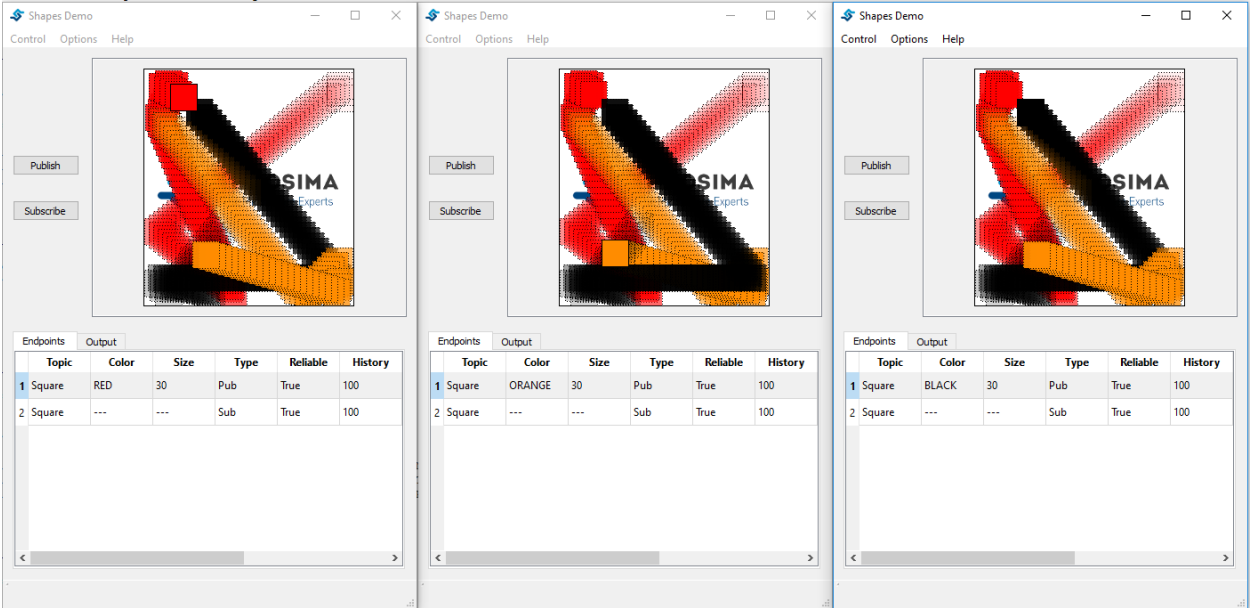
5 - Click Subscribe on Instance2.

- Select SQUARE option for Shape.
- Change the History field from 6 to 100.

6 - Click Subscribe on Instance3.

- Select SQUARE option for Shape.
- Change the History field from 6 to 100.

Your windows should look similar to the following image.



PARTITION

In Fast RTPS, you can use *Partitions* to group Subscribers and Publishers. If you deploy a Publisher with a partition, only the subscriber with the same partition will receive data from it. In this demo, there are four partitions (A, B, C and D). Additionally, you can select the “*” partition, that will be matched against all other partitions.

In this test, we are going to create three Publishers (Square in Partition A, Circle in Partition B, and Triangle in Partition “*”), and three Subscribers per instance, all of them in Partition A.

Finally, we have red squares and black triangles in partition A. Because of this, all instances are able to find triangles and squares. On the contrary, orange circles are published in partition B and they are only visible in the Instance2.

Step-by-Step

First, we must create three Publishers.

1 - Create a red square publisher:

- Start eProsimas Shapes-Demo. (We will refer to this instance as Instance1)
- Click on Publish.
- Select SQUARE option for Shape and RED for Color.
- Change the History field from 6 to 1.
- Check Partition A.

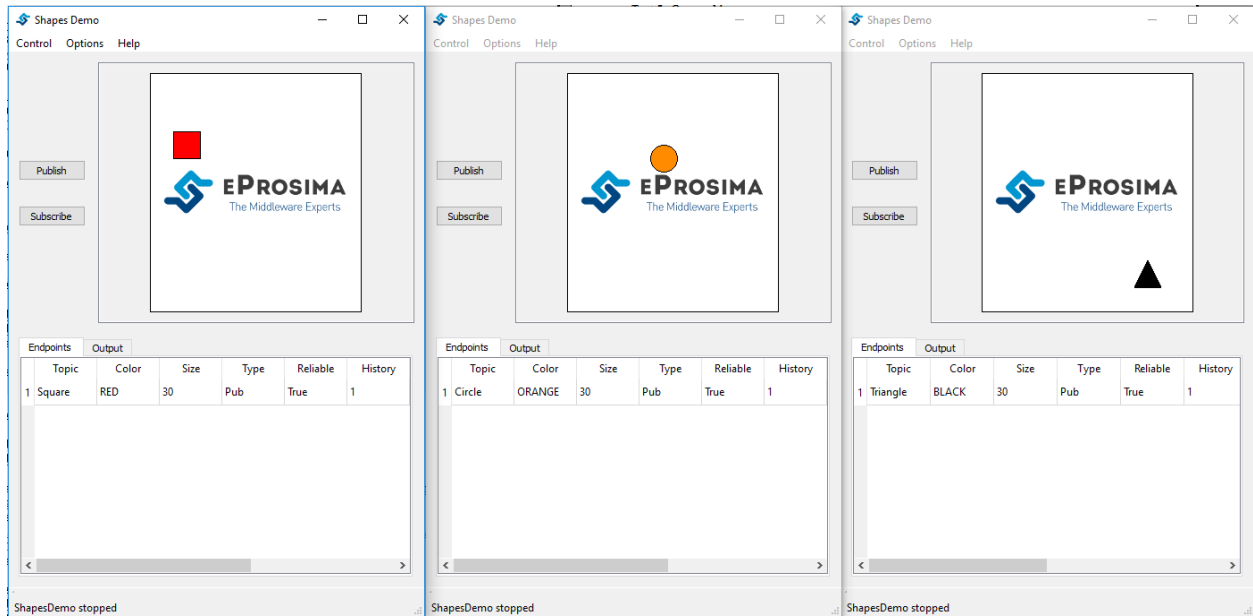
2 - Create an orange circle publisher:

- Start eProsimas Shapes-Demo. (We will refer to this instance as Instance2)
- Click on Publish.
- Select CIRCLE option for Shape and ORANGE for Color.
- Change the History field from 6 to 1.
- Check Partition B.

3 - Create a black triangle publisher:

- Start eProsimas Shapes-Demo. (We will refer to this instance as Instance3)
- Click on Publish.
- Select TRIANGLE option for Shape and BLACK for Color.
- Change the History field from 6 to 1.
- Check Partition “*”.

Instance1 will publish red squares in the partition A, Instance2 will publish orange circles in the partition B, and Instance3 will publish black triangles in all partitions. Your windows should look similar to the following image.



Second, create three Subscribers per Instance with the following characteristics.

Instance1:

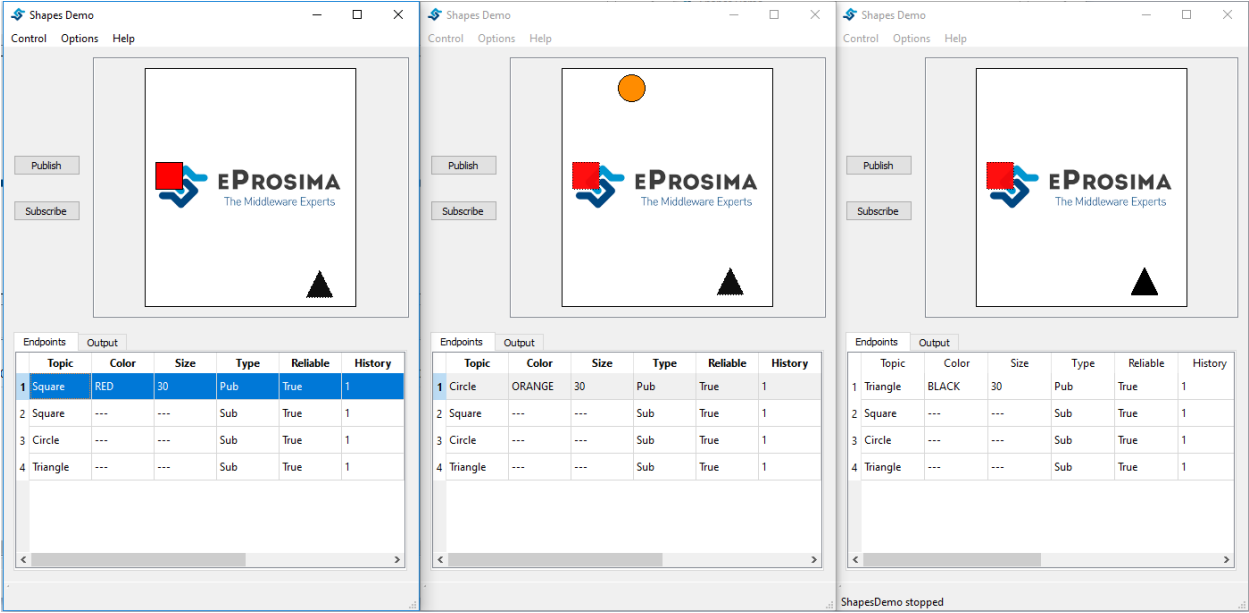
	Shape	Partition	History (Reliable)	Durability	Ownership
	Square	A	1 (ON)	VOLATILE	SHARED
I1	Circle	A	1 (ON)	VOLATILE	SHARED
	Triangle	A	1 (ON)	VOLATILE	SHARED

Instance2:

	Shape	Partition	History (Reliable)	Durability	Ownership
	Square	A	1 (ON)	VOLATILE	SHARED
I2	Circle	A	1 (ON)	VOLATILE	SHARED
	Triangle	A	1 (ON)	VOLATILE	SHARED

Instance3:

	Shape	Partition	History (Reliable)	Durability	Ownership
	Square	A	1 (ON)	VOLATILE	SHARED
I3	Circle	A	1 (ON)	VOLATILE	SHARED
	Triangle	A	1 (ON)	VOLATILE	SHARED



REDUNDANCY AND FAULT TOLERANCE

Fast RTPS allows more than one Publisher to write data on the same Topic. You can allow all Publishers to have the same relevance, or you can set one as the primary Publisher and keep the rest as secondary Publisher. In that case, only the main Publisher can send data to the subscribers.

The Ownership QoS determines if the Topic (Shape) is owned by a single Publisher or not.

You can select SHARE or EXCLUSIVE ownership, if you select EXCLUSIVE for a publisher, you have to indicate the value of strength for it. Only the Publisher with the highest strength can send data on this Topic. If the primary Publisher has a problem and cannot write data, the secondary Publisher with the highest strength is promoted to primary Publisher. If you select SHARE, all the publishers can write data at the same time.

In this test, we are going to create two Publishers with EXCLUSIVE ownership in SQUARE Shape, and one Subscriber with EXCLUSIVE ownership at the same Shape.

Step-by-Step

First, you have to launch two instances and create a Publisher in each of them:

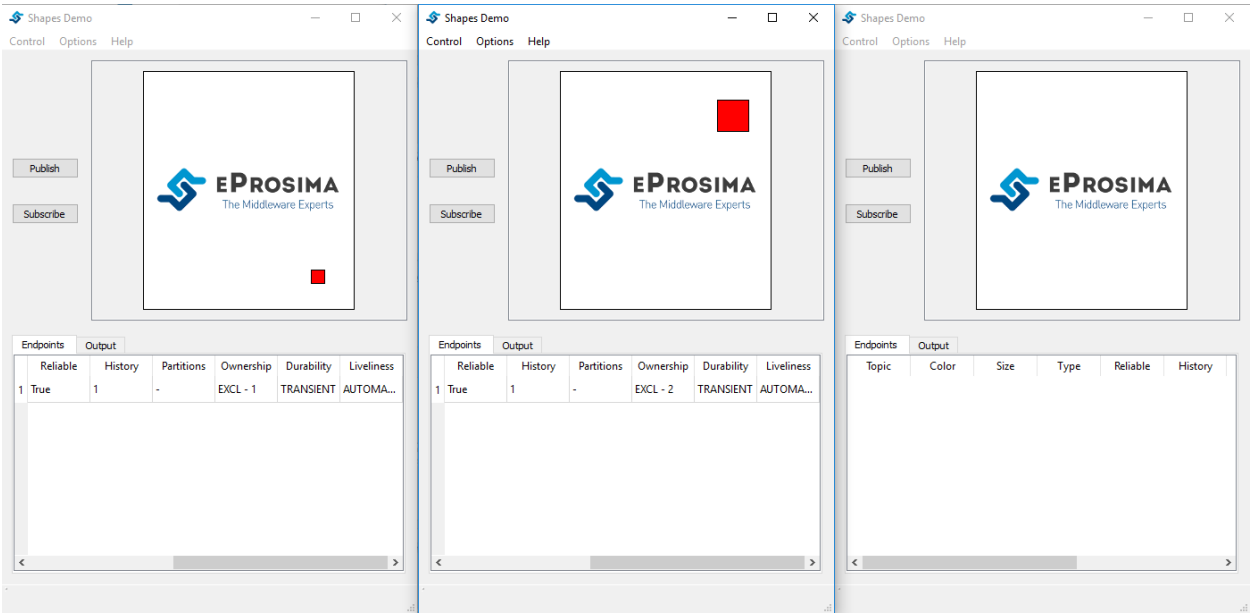
1 - Create a red square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance1)
- Click on Publish.
- Select SQUARE option for Shape and RED for Color.
- Change the History field from 6 to 1.
- Select EXCLUSIVE.
- Set Strength to 1.
- Set Size to 15.

2 - Create a red square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance2)
- Click on Publish.
- Select SQUARE option for Shape and RED for Color.
- Change the History field from 6 to 1.
- Select EXCLUSIVE.
- Set Strength 2.
- Set Size to 30.

You should see a small red square on Instance1 and a big red square on Instance2.

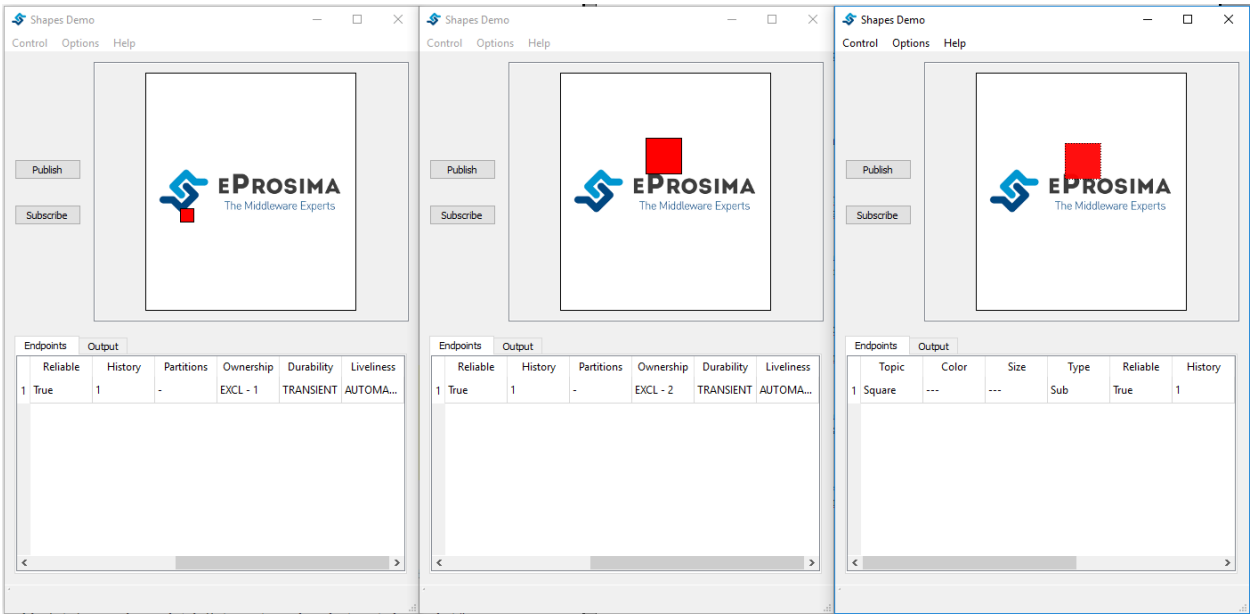


Now, create a Subscriber.

3 - Create a square subscriber:

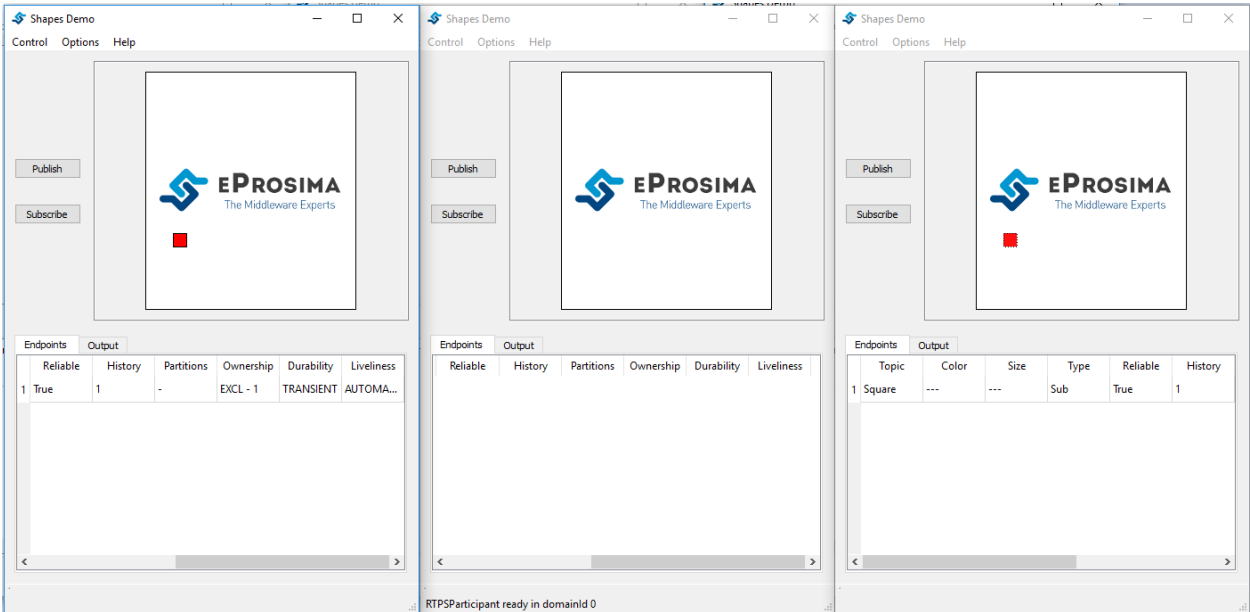
- Start eProsima Shapes-Demo. (We will refer to this instance as Instance3)
- Click on Subscribe.
- Select SQUARE option for Shape.
- Select EXCLUSIVE.

You should see a big square on Instance3, because Instance2 has a higher strength than Instance1.



Failure

Now you have to stop Instance2. Initially, Instance2 had higher strength and you saw a big red square on Instance2, but you should see a small red square because Instance1 has higher strength now.



If you repeat the second step, you should see a big red square mirroring the big red square movements in Instance2 again.

CONTENT BASED FILTER

In Fast RTPS, you can restrict the data that will be available to the subscriber to control network and CPU usage. When you deploy a new Subscriber, you can check the Content Based Filter. This filter draws a shaded region in the shapes window. Only the samples that are inside will be made available to the subscriber. You can resize and move this region dynamically.

In this test, we are going to create two Publishers and two Subscriber, one with Content Based.

Step-by-Step

First, you have to launch two instances and create a Publisher in each of them:

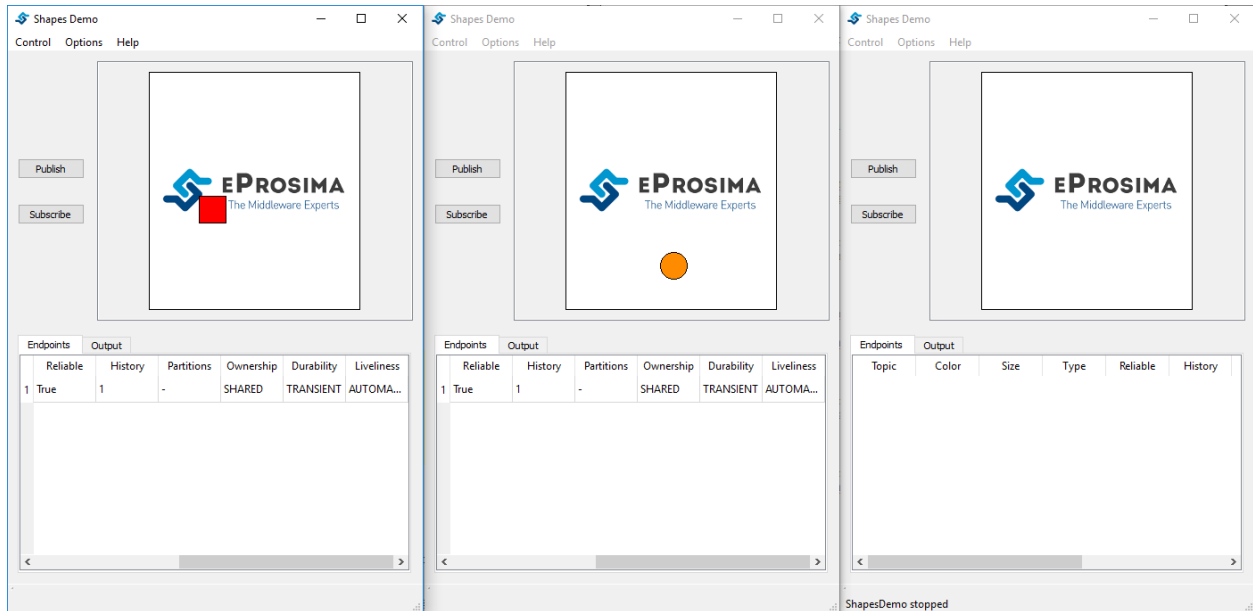
1 - Create a red square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance1)
- Click on Publish.
- Select SQUARE option for Shape and RED for Color.
- Change the History field from 6 to 1.

2 - Create an orange circle publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance2)
- Click on Publish.
- Select CIRCLE option for Shape and ORANGE for Color.
- Change the History field from 6 to 1.

Your windows should look similar to the following image.



Now, create two subscriber:

3 - Create a circle subscriber:

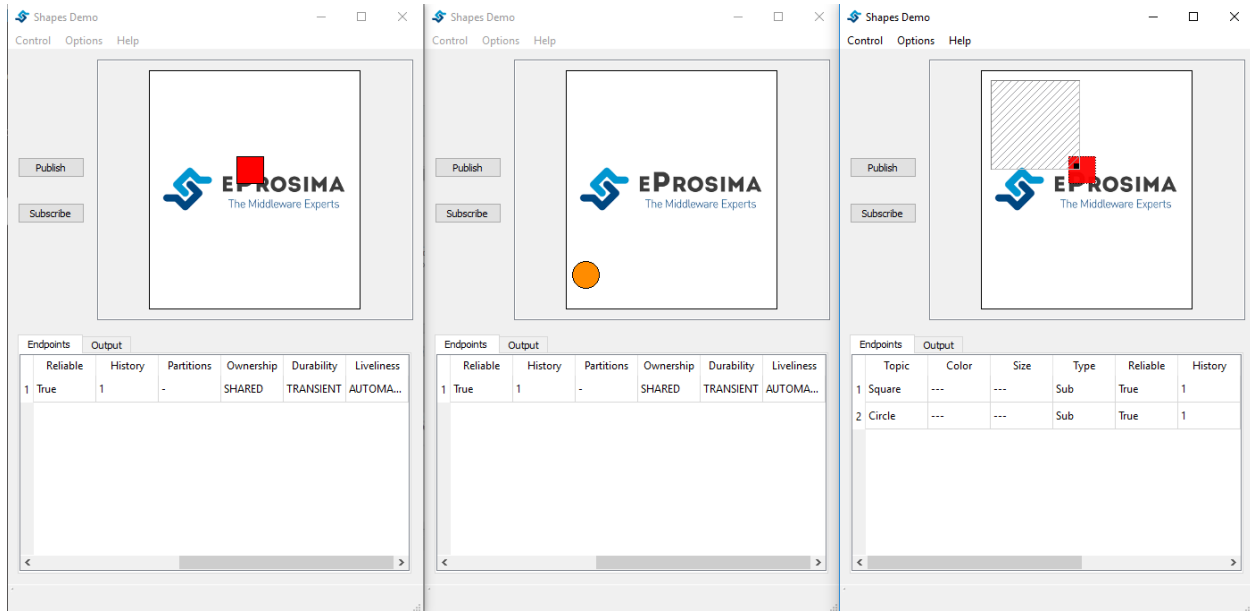
- Start eProsimas Shapes-Demo. (We will refer to this instance as Instance3)
- Click on Subscribe.
- Select CIRCLE option for Shape.
- Change the History field from 6 to 1.
- Check Content Based.

4 - Create a square subscriber:

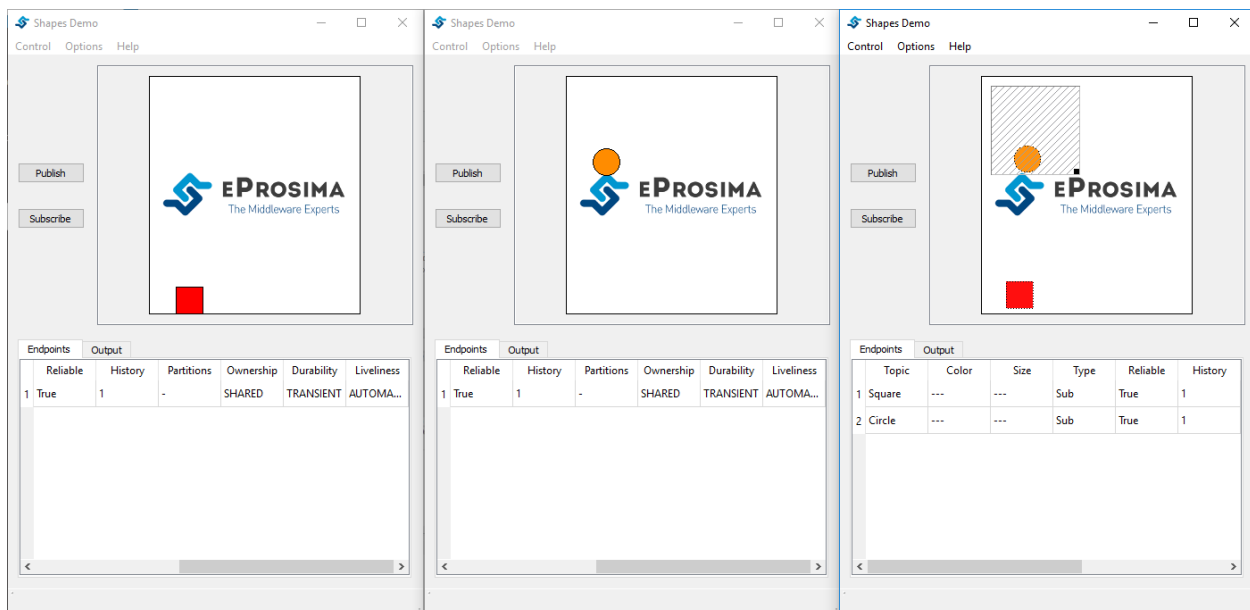
- Click on Subscribe in Instance3.
- Select SQUARE option for Shape.
- Change the History field from 6 to 1.

You will see a shaded rectangle in Instance3. This is the filter for the samples of the Circle Shape.

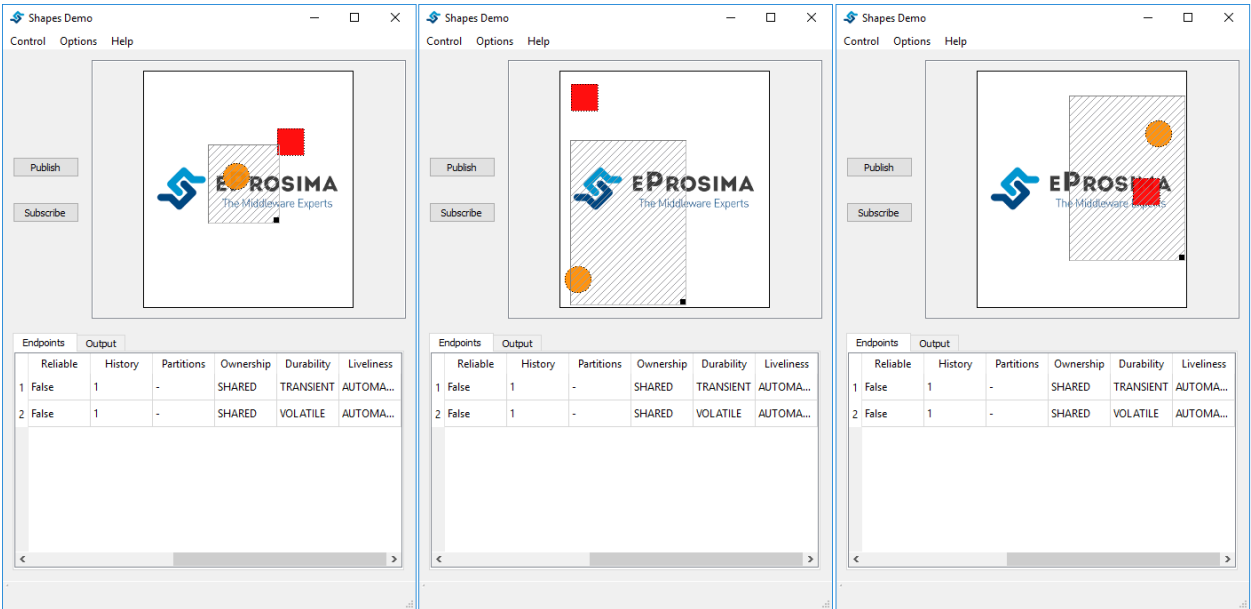
If the circle is out of the rectangle, it is not available for the subscriber.



On the contrary, if the instance is in the rectangle, it is available.



The rectangle is configurable, you can resize and move it dynamically. The following images show examples of the filter.



TIME BASED FILTER

When you deploy a Subscriber in Fast RTPS, you can select a time based filter. You can restrict the number of data updates in the Subscriber with this filter. You can specify the minimum separation time (in milliseconds) between updates. Any data arriving during this interval will be discarded.

In this test, we are going to create two Publishers and two Subscribers, one with a time based filter of 2000ms. You will see that one updates its position continuously, but the other jumps every two seconds.

Step-by-Step

First, you have to launch two instances and create a Publisher in each of them:

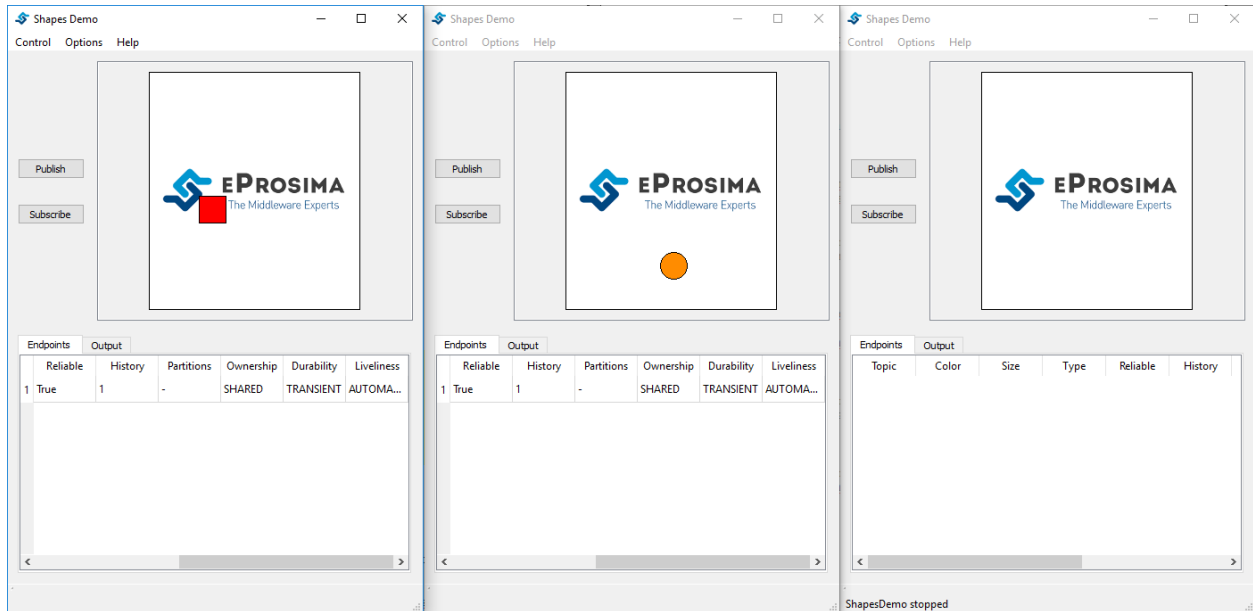
1 - Create a red square publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance1)
- Click on Publish.
- Select SQUARE option for Shape and RED for Color.

2 - Create an orange circle publisher:

- Start eProsima Shapes-Demo. (We will refer to this instance as Instance2)
- Click on Publish.
- Select CIRCLE option for Shape and ORANGE for Color.
- Change the History field from 6 to 1.

Your windows should look similar to the following image.



Now, create two subscriber:

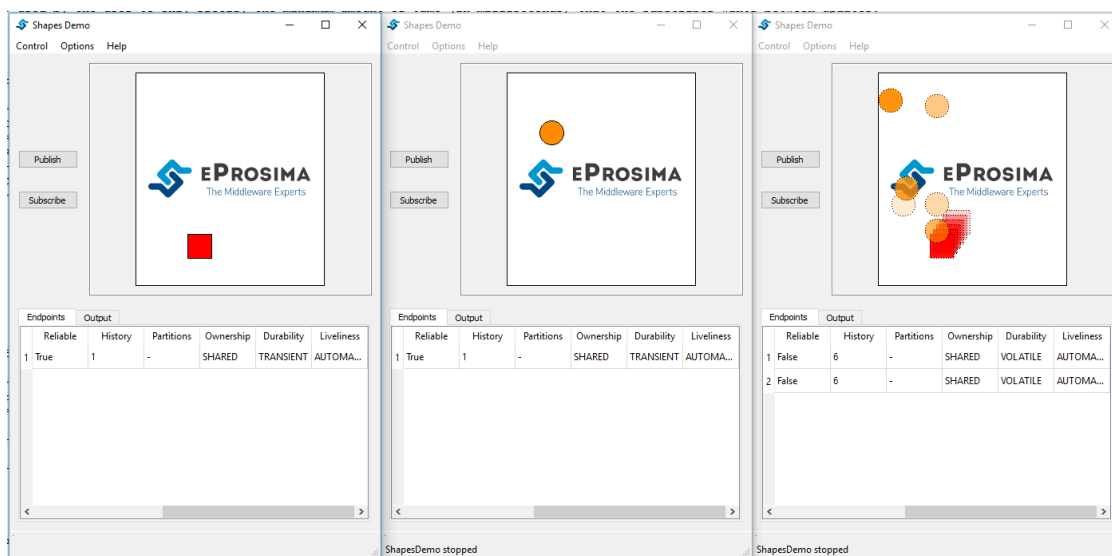
3 - Create a circle subscriber:

- Start eProsimas Shapes-Demo. (We will refer to this instance as Instance3)
- Click on Subscribe.
- Select CIRCLE option for Shape.
- Check Time Based an set 2000ms.

4 - Create a square subscriber:

- Click on Subscribe in Instance3.
- Select SQUARE option for Shape.

Your windows should look similar to the following image.



TCP TRANSPORT

Fast RTPS transport layer could be adapted to specific network requirements. UDP transport is optimal for local network applications but in order to interact on a Wide Area Network (WAN), where one or several NAT mappings may be enforced, a TCP transport would be more suitable. Here we explain how Shapes Demo should be set up in order to fit some network specific layouts.

8.1 LAN configuration

TCP over LAN can be tested in a scenario where several computers share the same LAN network; nevertheless, UDP performs better and is the advisable choice in practical situations. In this case one of the computers should set up Shapes Demo as a server and all the others as clients. Let's suppose that the server computer LAN IP address is 192.168.2.49 then all clients instances of Shapes Demo in the other computers must specify this IP Server address. In this case we select as TCP port 5100 but any other available TCP port is valid. The server computer firewall must allow inbound traffic on the selected port.

Server Instance settings

Options

Transport:

Listen Port: 5100

Server IP: 192.168.1.75

Server Port: 5100

Domain ID: 0

Shape Movements:

Update interval (ms): 75

Speed:

Client Instances settings

Options

Transport:

Listen Port: 5100

Server IP: 192.168.2.49

Server Port: 5100

Domain ID: 0

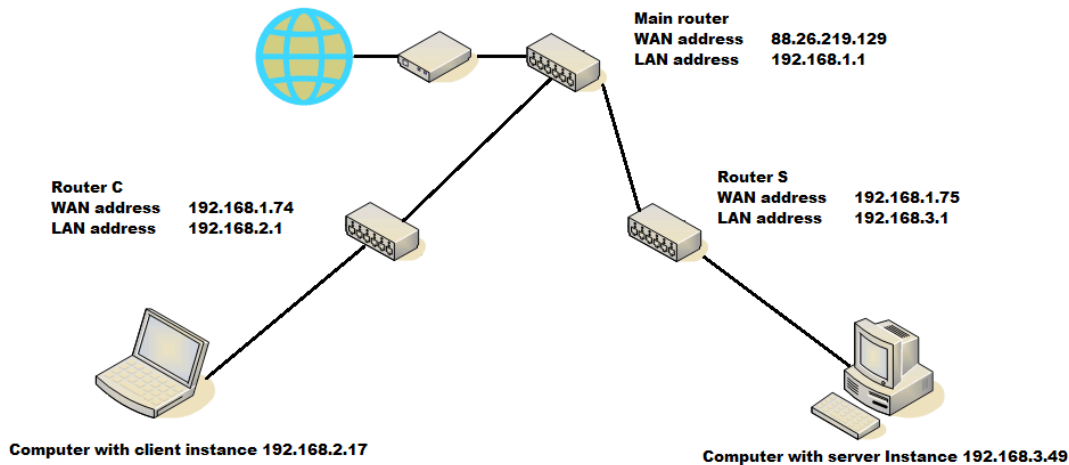
Shape Movements:

Update interval (ms): 75

Speed:

8.2 WAN configuration

This is the typical TCP scenario where the server computer does not share network with its clients but its reachable through a WAN IP address. This may happen if the server and clients are in a different LANs of the same WAN like WWW. In order to test this scenario we used the following network layout:



In the above diagram we can see:

- a client LAN network managed by router C. In this network a client Shapes Demo instance runs in computer address 192.168.2.17.
- a server LAN network managed by router S. In this network the server Shapes Demo instance runs in computer address 192.168.3.49. Router S NAT settings relay any inbound traffic to TCP port 5100 towards Server computer. TCP port 5100 was arbitrarily chosen, any available port will do. The Server computer firewall settings allow inbound traffic to TCP port 5100.
- a Main router which simulates the WAN network. In this network router C has address 192.168.1.74 and router S has address 192.168.1.75.

The previous configuration (see [LAN configuration](#)) does not work in this network scenario because server and client are behind a NAT.

Server Instance settings

Options

Transport:

Listen Port:

WAN IP:

Server Port:

Domain ID:

Shape Movements:

Update interval (ms):

Speed:

Client Instance settings

Options

Transport:

Listen Port:

Server IP:

Server Port:

Domain ID:

Shape Movements:

Update interval (ms):

Speed:

The above image shows server and client settings:

- The client *server IP* is no longer the server IP address (192.168.3.49) because is meaningless in client context. The server's router IP address must be used (router S 192.168.1.75) because the client's router can understand this address and properly lead the outbound traffic.

- The server *WAN IP* is the server's router IP address (router S 192.168.1.75). Router S NAT settings relay inbound traffic to server's TCP port towards Server computer.

TROUBLESHOOTING

This document compiles the known compatibility issues with other Shapes Demo implementations and the possible solutions.

9.1 Two instances in the same computer

When using eProsima ShapesDemo and RTI Shapes Demo in the same computer the file *RTISHAPESDEMOQOSPROFILES.xml* needs to be modified to force RTI to only use UDPv4 as transport. You should add the following lines inside the ShapesDefaultProfile qos_profile tag.

```
<participant_qos>
  <transport_builtin><mask>UDPv4</mask></transport_builtin>
  <discovery_config>
    <builtin_discovery_plugins>SDP</builtin_discovery_plugins>
    <participant_liveliness_lease_duration>
      <sec>30</sec>
      <nanosec>0</nanosec>
    </participant_liveliness_lease_duration>
  </discovery_config>
</participant_qos>
```

This setting is necessary to avoid the use of Shared Memory as transport (disabling the loopback), a feature that is not yet implemented in Fast RTPS.

INTEROPERABILITY

You can see an example of the interoperability between eProxima Shapes-Demo and other implementations in the following video:

https://www.youtube.com/watch?v=e9_oAJDh-tY