Fast DDS Visualizer Plugin Documentation

Release 1.0.0

eProsima

Mar 02, 2023

INTRODUCTION

eProsima Fast DDS Visualizer Plugin is a plugin for the *PlotJuggler* application. PlotJuggler is a graphical desktop application providing visualization features of data series, time series, X-Y plots. It also adds data management features, such as data import and export, custom and built-in data manipulation functions, data series merges, etc. Also, this software supports many different layouts, with dynamic, rich and user-friendly customization.

Fast DDS Visualizer Plugin allows users to visualize topic-related data from a DDS network. The user can select several topics from the discovered topics running in the DDS network. These topics will be divided by values using data type introspection, and each value could be visualized and managed separately, allowing the user to quickly visualize in different ways the detailed data content that is being exchanged in the network.

eProsima Fast DDS Visualizer Plugin supports the following features:

1. **Data type introspection**: by using *Fast DDS Dynamic Types*, this plugin allows to discover the data type of the topic, and to visualize the data content using the corresponding data type.
2. **DDS Configurations**: Different configurations can be set to choose the *Domain Id* and to select specific *DDS Topics* to be visualized.
3. **All PlotJuggler features**: This plugin composes nicely with all the PlotJuggler features, so that users can create rich graphs from sophisticated data manipulations.

Find more about us at eProsima's webpage.

Support available at:

- Email: support@eprosima.com
- Phone: +34 91 804 34 48

CHAPTER

ONE

CONTRIBUTING TO THE DOCUMENTATION

Fast DDS Visualizer Plugin Documentation is an open source project, and as such all contributions, both in the form of feedback and content generation, are most welcomed. To make such contributions, please refer to the Contribution Guidelines hosted in our GitHub repository.

CHAPTER

TWO

STRUCTURE OF THE DOCUMENTATION

This documentation is organized into the sections below.

- *Installation Manual*
- *Getting Started*
- *User Manual*
- *Developer Manual*
- *Release Notes*

eProsima Fast DDS Visualizer Plugin is a plugin for the *PlotJuggler* application. PlotJuggler is a graphical desktop application providing visualization features of data series, time series, X-Y plots. It also adds data management features, such as data import and export, custom and built-in data manipulation functions, data series merges, etc. Also, this software supports many different layouts, with dynamic, rich and user-friendly customization.

Fast DDS Visualizer Plugin allows users to visualize topic-related data from a DDS network. The user can select several topics from the discovered topics running in the DDS network. These topics will be divided by values using data type introspection, and each value could be visualized and managed separately, allowing the user to quickly visualize in different ways the detailed data content that is being exchanged in the network.

eProsima Fast DDS Visualizer Plugin supports the following features:

1. **Data type introspection**: by using *Fast DDS Dynamic Types*, this plugin allows to discover the data type of the topic, and to visualize the data content using the corresponding data type.
2. **DDS Configurations**: Different configurations can be set to choose the *Domain Id* and to select specific *DDS Topics* to be visualized.
3. **All PlotJuggler features**: This plugin composes nicely with all the PlotJuggler features, so that users can create rich graphs from sophisticated data manipulations.

2.1 Contacts and Commercial support

Find more about us at eProsima's webpage.

Support available at:

- Email: support@eprosima.com
- Phone: +34 91 804 34 48

2.2 Contributing to the documentation

Fast DDS Visualizer Plugin Documentation is an open source project, and as such all contributions, both in the form of feedback and content generation, are most welcomed. To make such contributions, please refer to the Contribution Guidelines hosted in our GitHub repository.

2.3 Structure of the documentation

This documentation is organized into the sections below.

- *Installation Manual*
- *Getting Started*
- *User Manual*
- *Developer Manual*
- *Release Notes*

2.4 PlotJuggler eProsima Edition on Windows

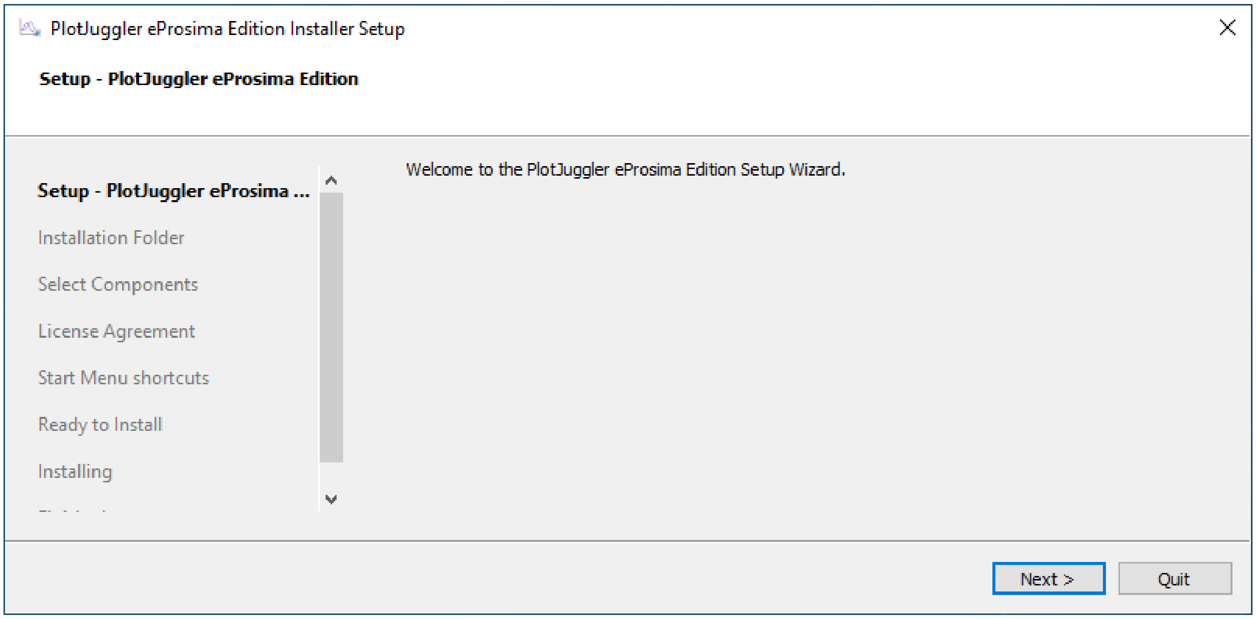
This section provides instructions on how to install the *PlotJuggler eProsima Edition* application. This is available on eProsima's downloads page.

In the list of eProsima official releases, search for *PlotJuggler eProsima Edition* and go to the available files of the latest version available. Then click on the **Download now** button of the Windows installer (`PlotJuggler eProsima Edition x.x.x - Win (32 & 64)` where `X.X.X` is the version of the application.).

Now locate the downloaded file and run the installer. The *PlotJuggler eProsima Edition* installer window should open as shown

in the following image.

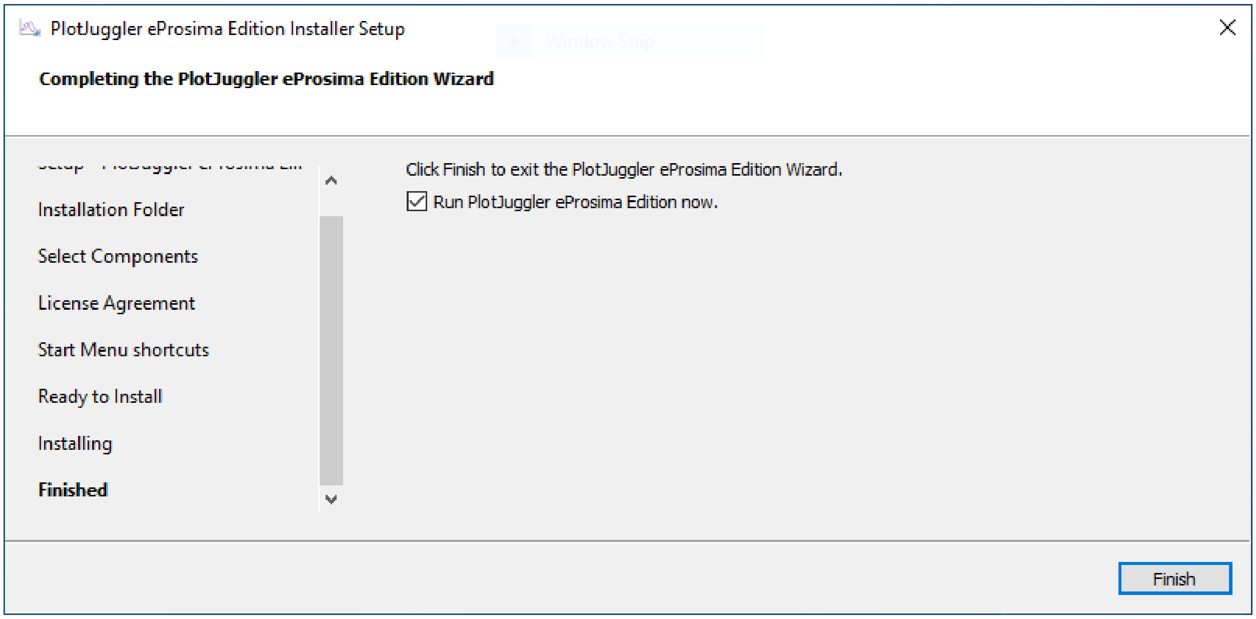
Follow the installation steps until the installation process is complete. A window as the one shown below should be visible on screen after installation, from which you may directly run the *PlotJuggler eProsima Edition* application.



PlotJuggler eProsima Edition Installer Setup

Setup - PlotJuggler eProsima Edition

Welcome to the PlotJuggler eProsima Edition Setup Wizard.



PlotJuggler eProsima Edition Installer Setup

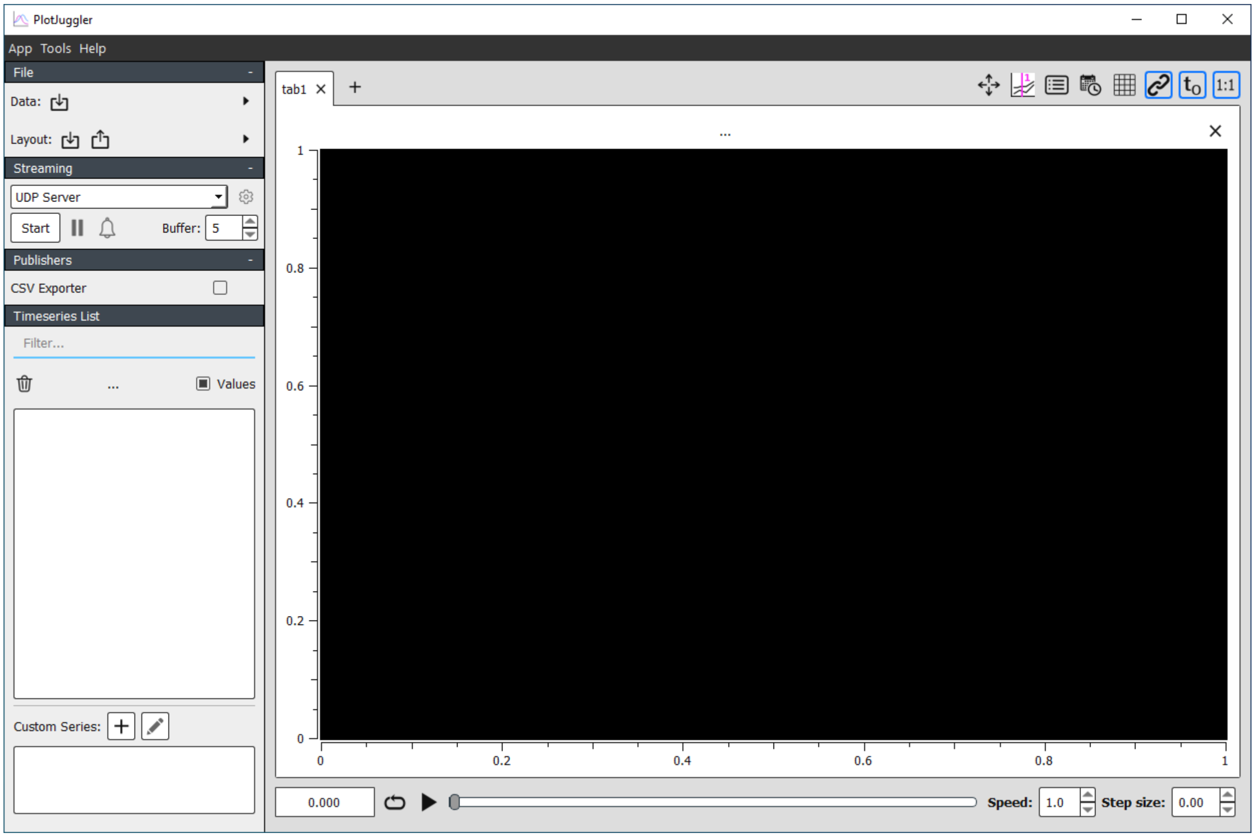
Completing the PlotJuggler eProsima Edition Wizard

Click Finish to exit the PlotJuggler eProsima Edition Wizard.

☑ Run PlotJuggler eProsima Edition now.

2.4.1 OpenGL Troubleshooting

At application startup, if a black screen like the following image appears:



it is likely because an incompatibility between the OpenGL version used in PlotJuggler and the latest versions of Windows. To solve this problem, go to Preferences:

and then disable the OpenGL checkbox:

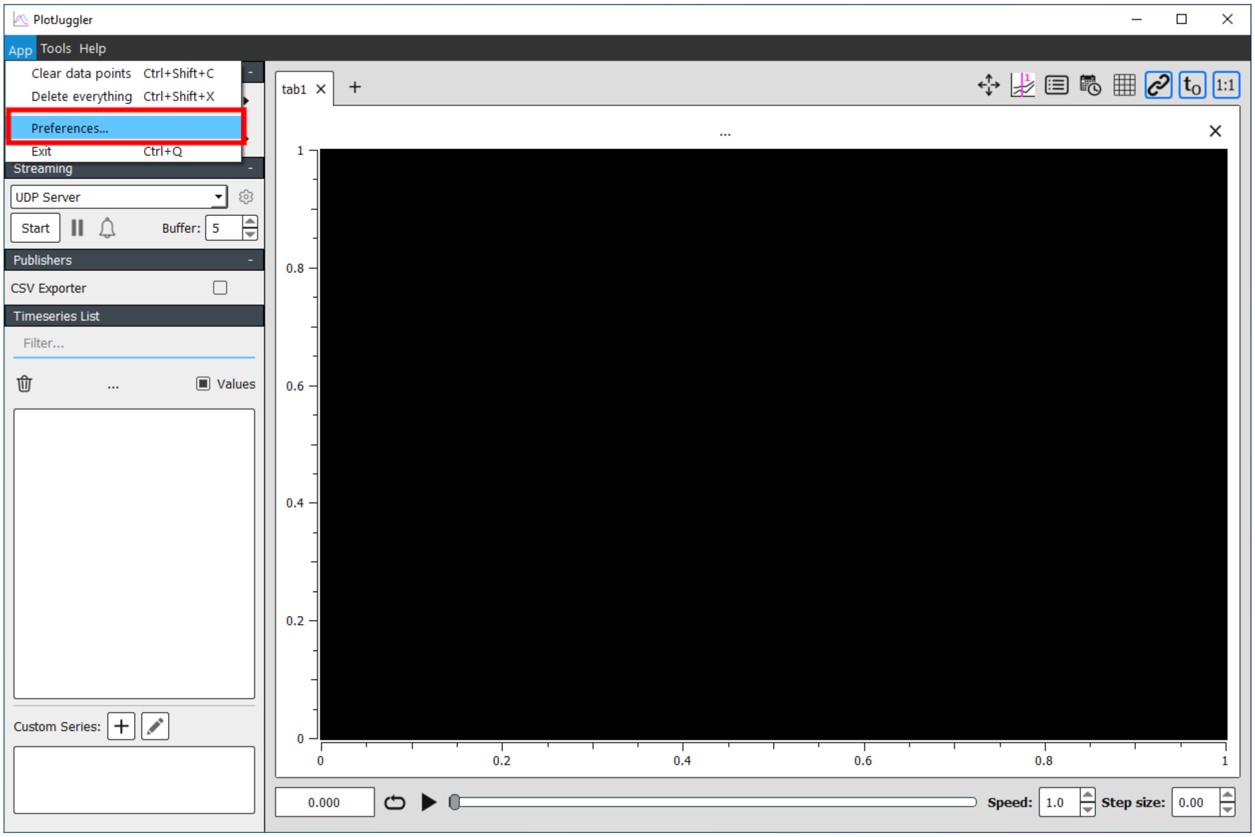
After clicking OK and **restarting** the application, the problem should now be solved.

2.5 PlotJuggler eProsima Edition on Linux

This section provides instructions on how to install the *PlotJuggler eProsima Edition*. This is available on eProsima's downloads page.

There are two mechanisms for running the application.

- Through the *PlotJuggler eProsima Edition* installer.
- Using the *AppImage* format, which is a portable format of the application software.



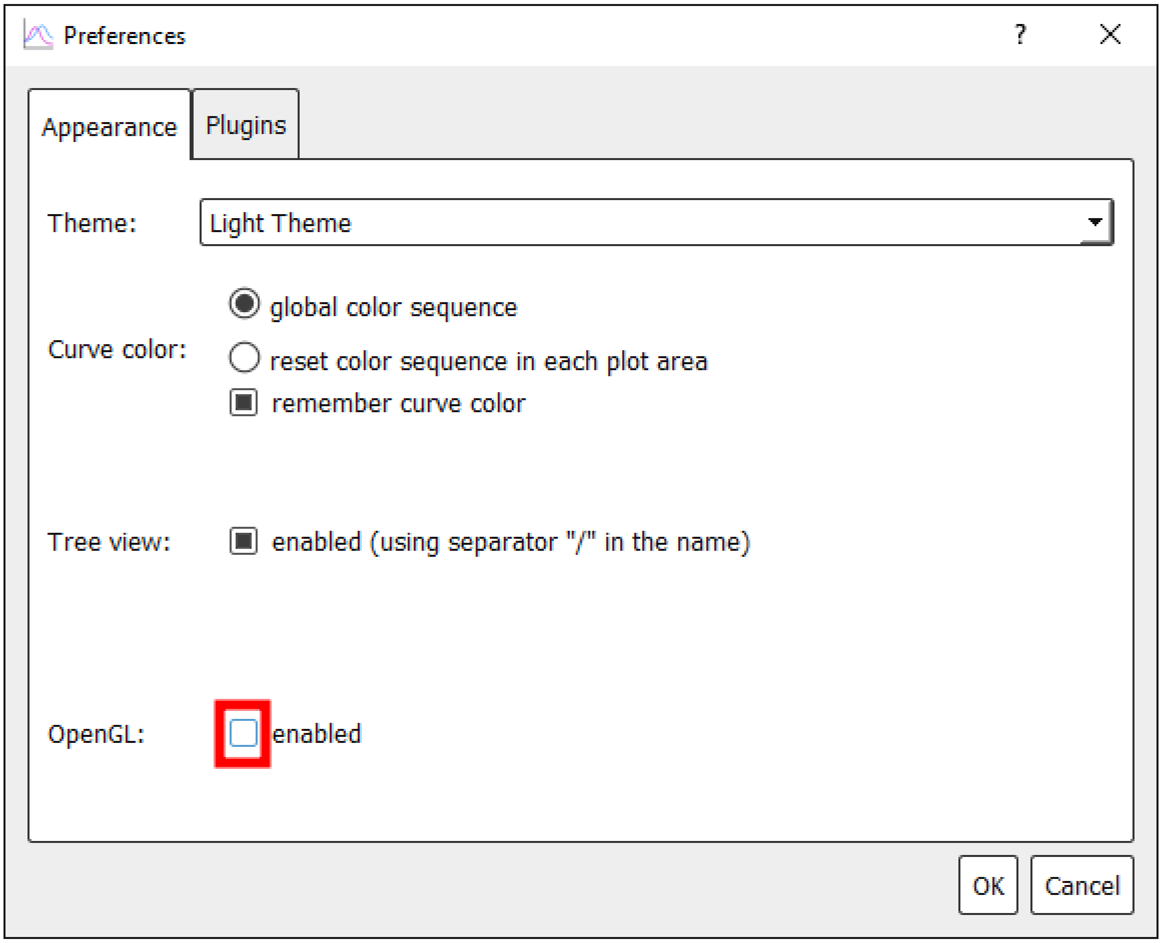
2.5.1 *Fast DDS Visualizer Plugin* installer

The first option is to install the *PlotJuggler eProsima Edition* application together with all its dependencies. To do so, first run the `eProsima_PlotJuggler-eProsima-Edition-<version>-Linux.run` executable and follow the instructions it provides to install the program in a directory on the system.

2.5.2 *PlotJuggler eProsima Edition* portable format

eProsima also distributes a portable version of the *PlotJuggler eProsima Edition* for Linux in AppImage format. In this case, download this version from the eProsima’s downloads page and run the downloaded file to launch the monitor. The name of this file is `eProsima_PlotJuggler-eProsima-Edition-<X.X.X>-Linux.AppImage`, where `X.X.X` is the version of the application.

Warning: In case these files are not executed, check that they have executable permissions.



Preferences

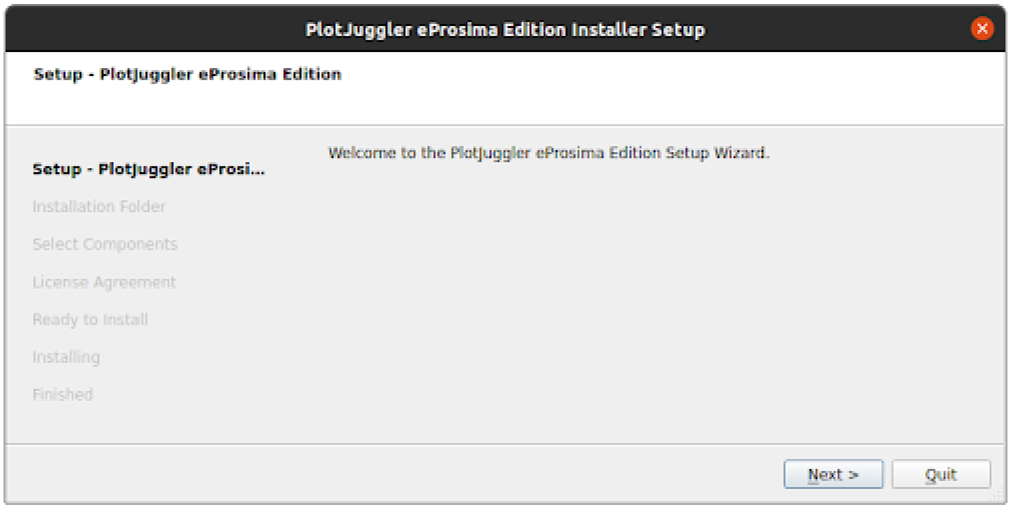
Appearance | Plugins

Theme: Light Theme

Curve color:
● global color sequence
○ reset color sequence in each plot area
☒ remember curve color

Tree view: ☒ enabled (using separator "/" in the name)

OpenGL: □ enabled



PlotJuggler eProsima Edition Installer Setup

Setup - PlotJuggler eProsima Edition

Welcome to the PlotJuggler eProsima Edition Setup Wizard.

2.6 PlotJuggler eProsima Edition Docker image

eProsima distributes a Docker image of *PlotJuggler eProsima Edition* with Ubuntu 20.04 as base image. This image launches the *PlotJuggler* application with the *Fast DDS Visualizer Plugin* already installed and sourced.

1. This Docker image was built for Ubuntu 20.04 (Focal Fossa).

   To run this container you need Docker installed. From a terminal run

   ```
   sudo apt install docker.io
   ```

2. Download the compressed Docker image in `.tar` format from the eProsima's downloads page. It is strongly recommended to download the image corresponding to the latest version of *Fast DDS Visualizer Plugin*.

3. Extract the image by executing the following command:

   ```
   docker load -i ubuntu-plotjuggler-eprosima-edition:<version>.tar
   ```

   where `version` is the downloaded version of *Fast DDS Visualizer Plugin*.

4. Run the Docker container executing the following command:

   ```
   xhost local:root
   docker \
       --net=host \
       --privileged -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix \
       ubuntu-plotjuggler-eprosima-edition:<version>
   ```

   After executing the previous command you should be able to see the application running. Check the following section *Start Fast DDS Visualizer Plugin* to learn how to start the *Fast DDS Visualizer Plugin*.

2.6.1 Fast DDS Suite

eProsima distributes a Docker image with several features and applications regarding dds, including the *PlotJuggler eProsima Edition*. Check section *PlotJuggler eProsima Edition within Fast DDS Suite* to learn how to download and use the *Fast DDS Suite*.

2.7 Example of usage

This example will show how to monitor a DDS network using *eProsima Fast DDS Visualizer Plugin*, by leveraging some of the features and configuration options this add-on has to offer.

We will make use of ShapesDemo, a tool primarily utilized for testing interoperability between different DDS implementations, as well as to experiment with different QoS configurations. Shapes of different types and colors move around the screen, while publishing their coordinates in real time over DDS.

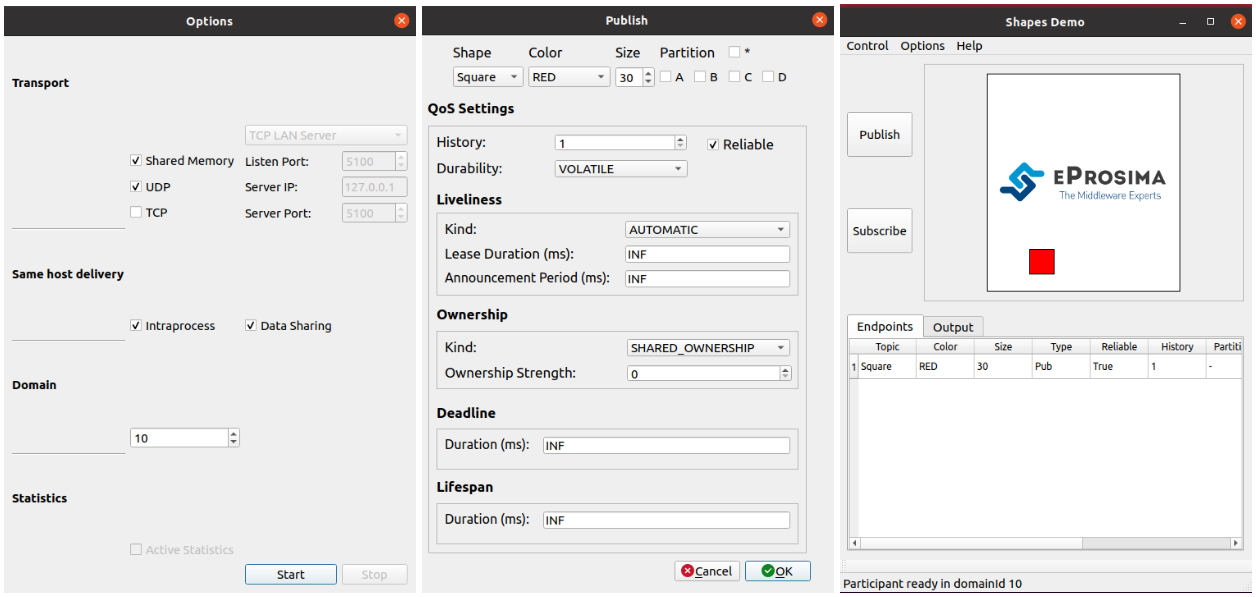
2.7.1 Prerequisites

First of all, you need to follow the steps outlined in the *Installation Manual* for the installation of *PlotJuggler eProsima Edition* and all its dependencies. In addition, please refer to *ShapesDemo* Installation Manual and install this application before getting started. Version `2.7.0` or above is required in this tutorial

Both of them, along with other *eProsima* tools and products, are preinstalled in **Fast DDS Suite**, so consider using *PlotJuggler eProsima Edition within Fast DDS Suite* as it already presents all the elements required to follow this example. It may also be useful, though not required, to have a look at *Plotjuggler*'s introductory tutorials available in this section *PlotJuggler Tutorials*.

2.7.2 Launch ShapesDemo application

Let us run an instance of *ShapesDemo*, and start publishing under the `Square` topic. Choose the DDS domain on which data will be published (Options->Participant Configuration), and create a publisher with default settings on this topic. See Using eProsima Shapes Demo for more details.



Transport
TCP LAN Server
☑ Shared Memory Listen Port: 5100
☑ UDP Server IP: 127.0.0.1
☐ TCP Server Port: 5100

Same host delivery
☑ Intraprocess ☑ Data Sharing

Domain
10

Statistics
☐ Active Statistics

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

QoS Settings
History: 1 ☑ Reliable
Durability: VOLATILE

Liveliness
Kind: AUTOMATIC
Lease Duration (ms): INF
Announcement Period (ms): INF

Ownership
Kind: SHARED_OWNERSHIP
Ownership Strength: 0

Deadline
Duration (ms): INF

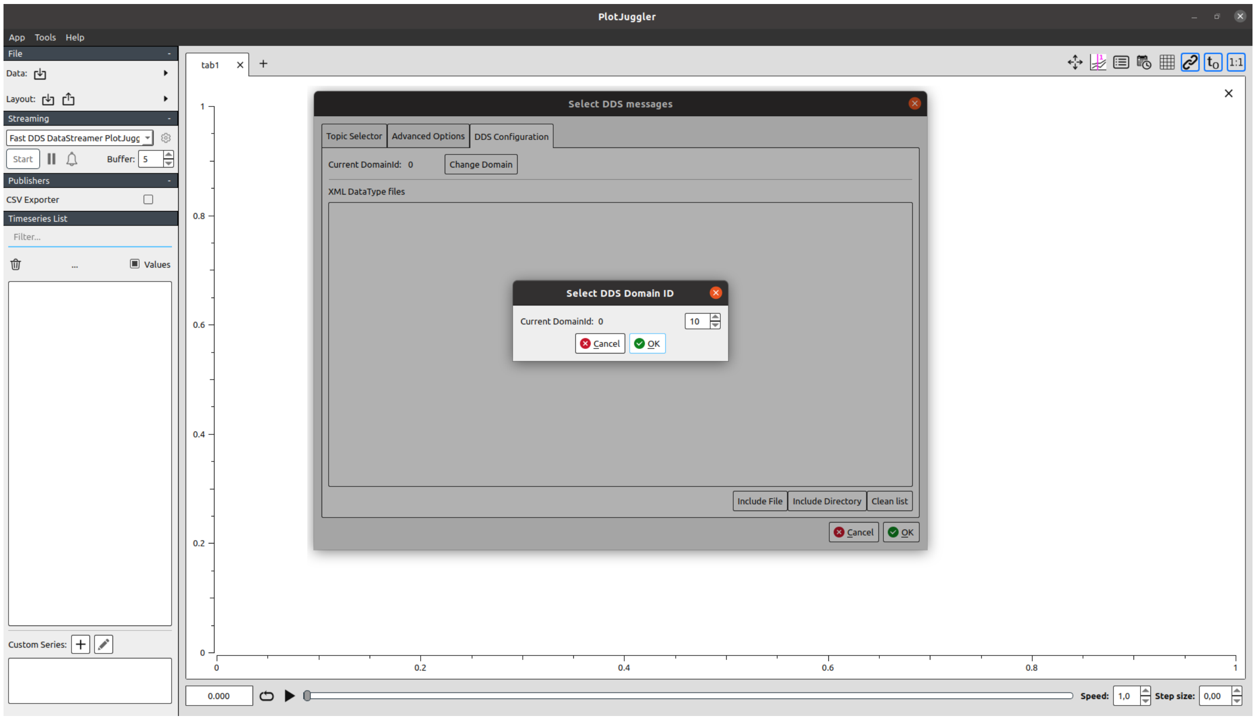
Lifespan
Duration (ms): INF

	Topic	Color	Size	Type	Reliable	History	Partiti
1	Square	RED	30	Pub	True	1	-

Participant ready in domainId 10

2.7.3 Visualize DDS data

Run now *Plotjuggler* application, and *start streaming* data through the *Fast DDS Visualizer Plugin*. In the DDS *configuration* settings tab, select the same DDS domain as the one previously chosen in *ShapesDemo*.



Current DomainId: 0

Select DDS Domain ID — Current DomainId: 0 — 10

A row with the `Square` topic should be visible under the *Topic Selector* tab. Select it and hit the `OK` button to proceed.

Plotting time series

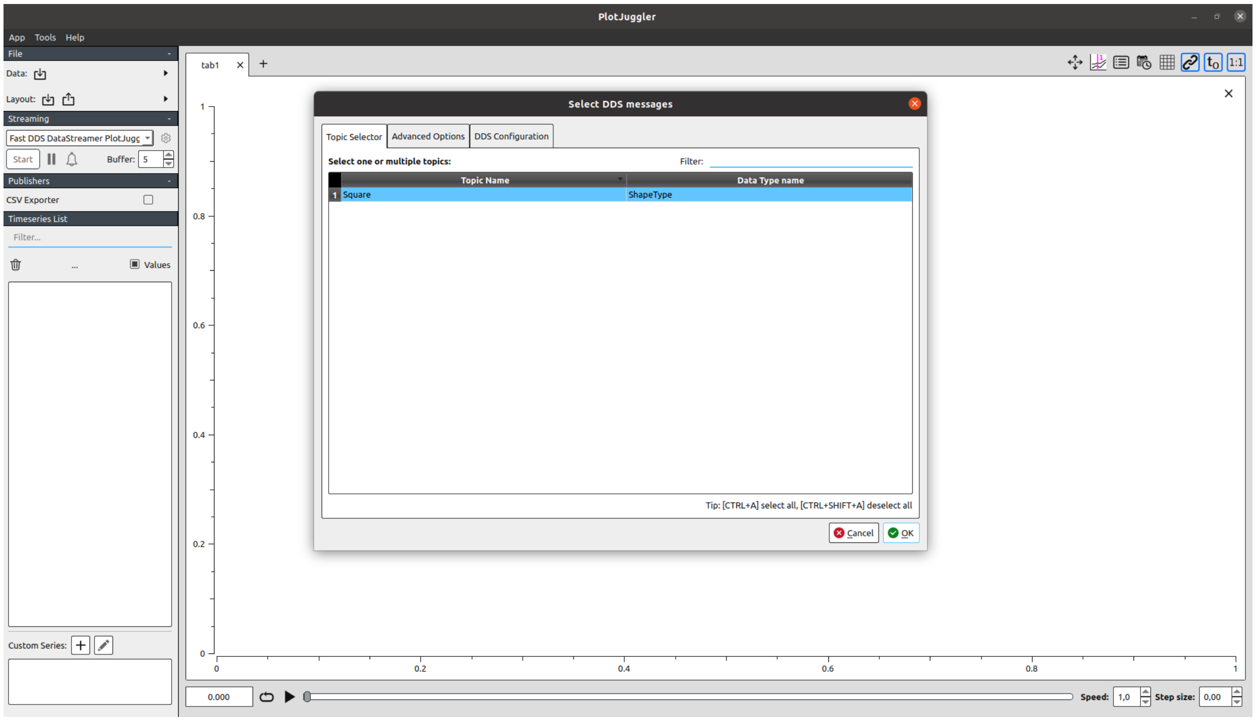
One can now plot, for example, the evolution in time of the `y` coordinate. Just drag and drop to the main panel the corresponding entry appearing on the left column, and data should start being displayed right away.

It is rather confusing to interpret this particular time series, as in *ShapesDemo* the origin for this axis corresponds to the top of the grid, while in *Plotjuggler* it is depicted the other way around. This can be easily solved by flipping the Y axis (option available when right clicking over a plot).

X-Y curves

Still, it is not the most intuitive to interpret both series separately. Let us exploit another useful feature *Plotjuggler* brings to the table: x-y curves.

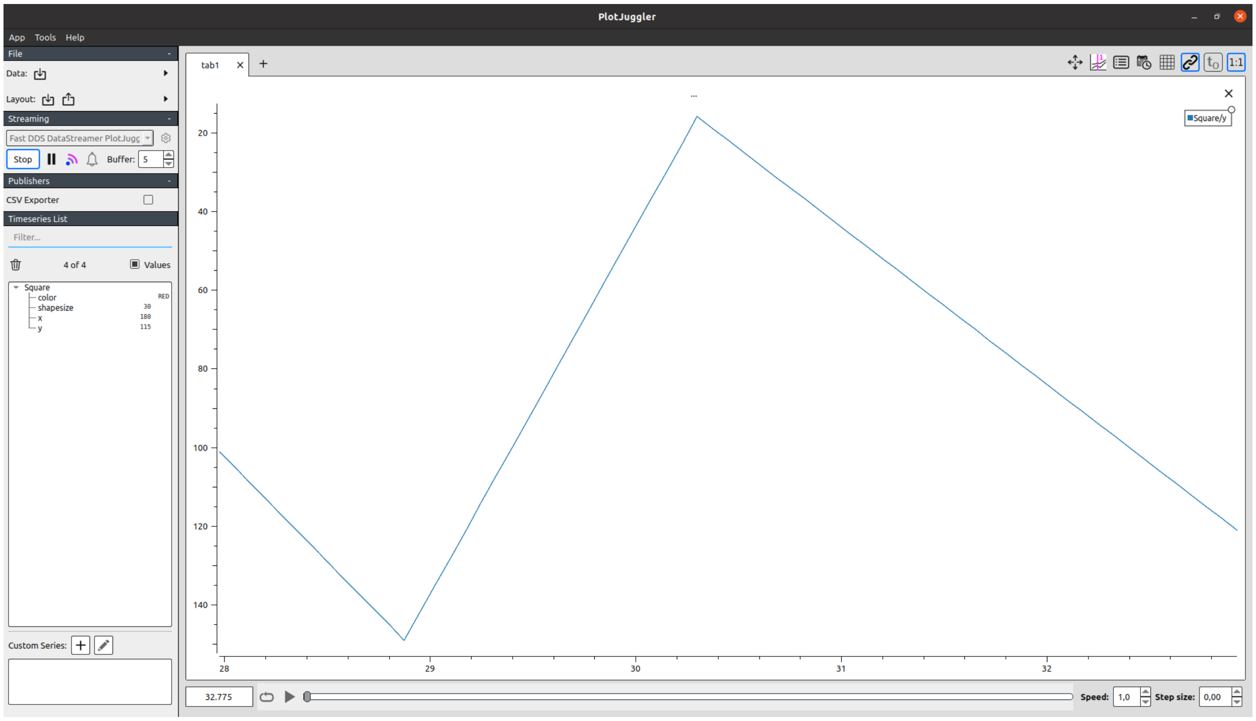
Create a new tab, select both `x` and `y` items from the *Timeseries List*, and drag them to the empty canvas, this time using the right mouse button. Again, flip the y axis to visualize the movement of the square shape, now closely resembling the one observed at *ShapesDemo* application.



Select one or multiple topics:

	Topic Name	Data Type name
1	Square	ShapeType

Tip: [CTRL+A] select all, [CTRL+SHIFT+A] deselect all



Square	
color	RED
shapesize	30
x	188
y	115

2.7.4 Next steps

We hope this example served as a useful resource for getting a grasp on the basics of *eProsima Fast DDS Visualizer Plugin*. Feel free to explore the *User Manual* section to learn more about its various features and configuration options.

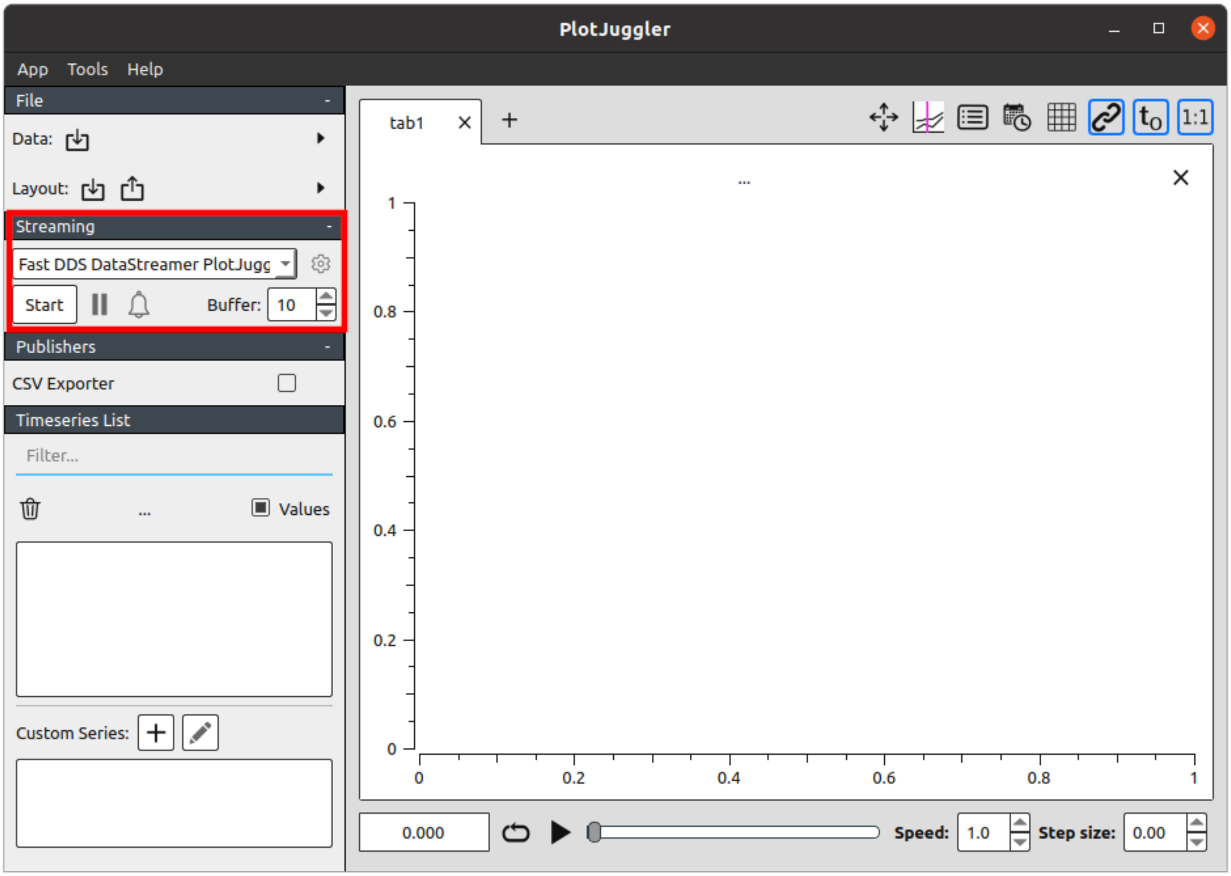
2.8 PlotJuggler Tutorials

The *PlotJuggler* is an application that support several features to interact with data series, with many graphical interfaces, layouts, configurations and aids. To learn how to use *PlotJuggler* application and know about all its features, please refer to the following tutorials:

- PlotJuggler Tutorial: learn the basics
- PlotJuggler Tutorial: data sources
- PlotJuggler Tutorial: custom plots, filters and transformations

2.9 Start Fast DDS Visualizer Plugin

In order to start the *Fast DDS Visualizer Plugin* in the *PlotJuggler* application, select in combo box in *Streaming* section in left panel the plugin *Fast DDS DataStreamer Plugin*.



Fast DDS DataStreamer PlotJugg

Buffer: 10

0.000

Speed: 1.0

Step size: 0.00

Note: If this plugin does not automatically appear in the combo box, it means the plugin has not been correctly loaded. Check section *Source Plugin* to learn how to add the plugin to the application. Be sure that *Fast DDS* libraries are sourced.

Once this plugin is selected, press *Start* button and a new Dialog will appear asking for plugin configurations. To learn more about these configurations, check section *Fast DDS Visualizer Plugin Configuration*.

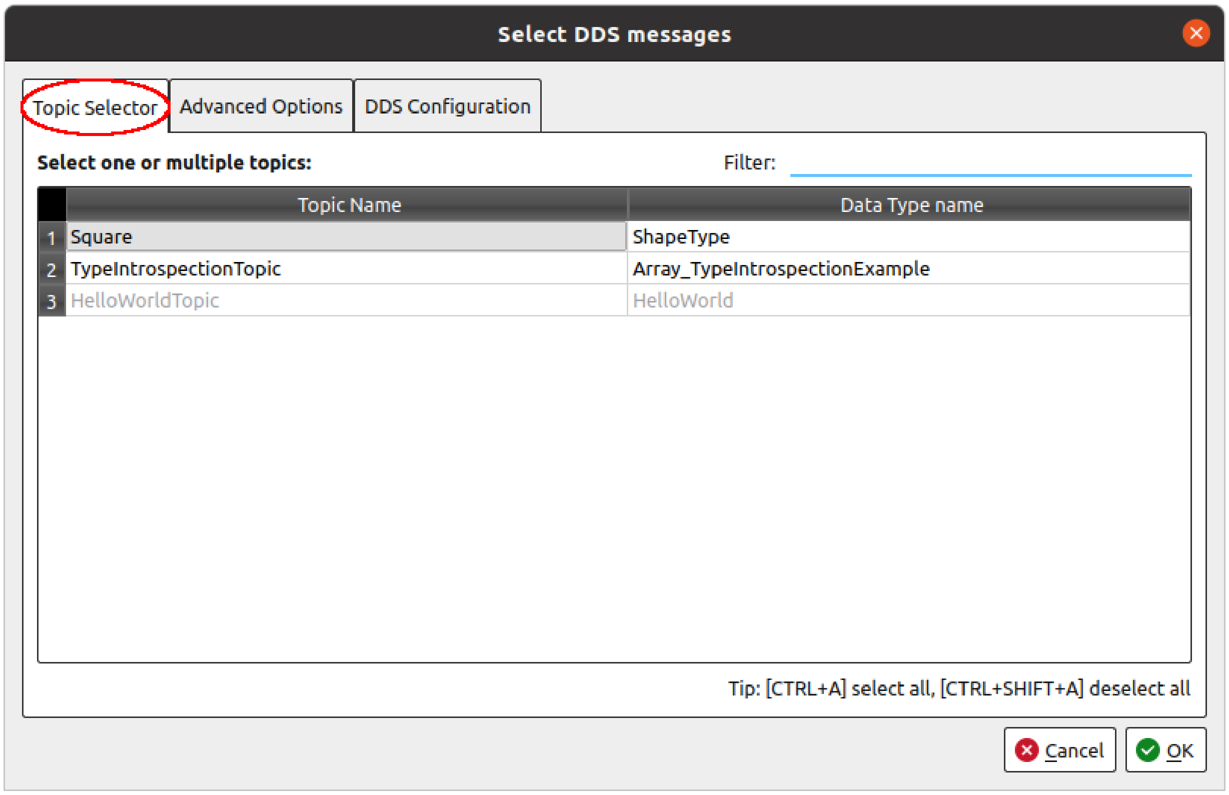
2.9.1 Stop plugin

User can press *Stop* button in *Streaming* section at any time to stop the plugin and destroy every DDS entity internally created.

2.10 Fast DDS Visualizer Plugin Configuration

Once the plugin is started, a new menu appears to configure the internal *Fast DDS Domain Participant*. This section explains the different configuration options. Check section *Start Fast DDS Visualizer Plugin* to learn how to start the plugin.

2.10.1 Topic Selector



Select one or multiple topics:

Filter:

	Topic Name	Data Type name
1	Square	ShapeType
2	TypeIntrospectionTopic	Array_TypeIntrospectionExample
3	HelloWorldTopic	HelloWorld

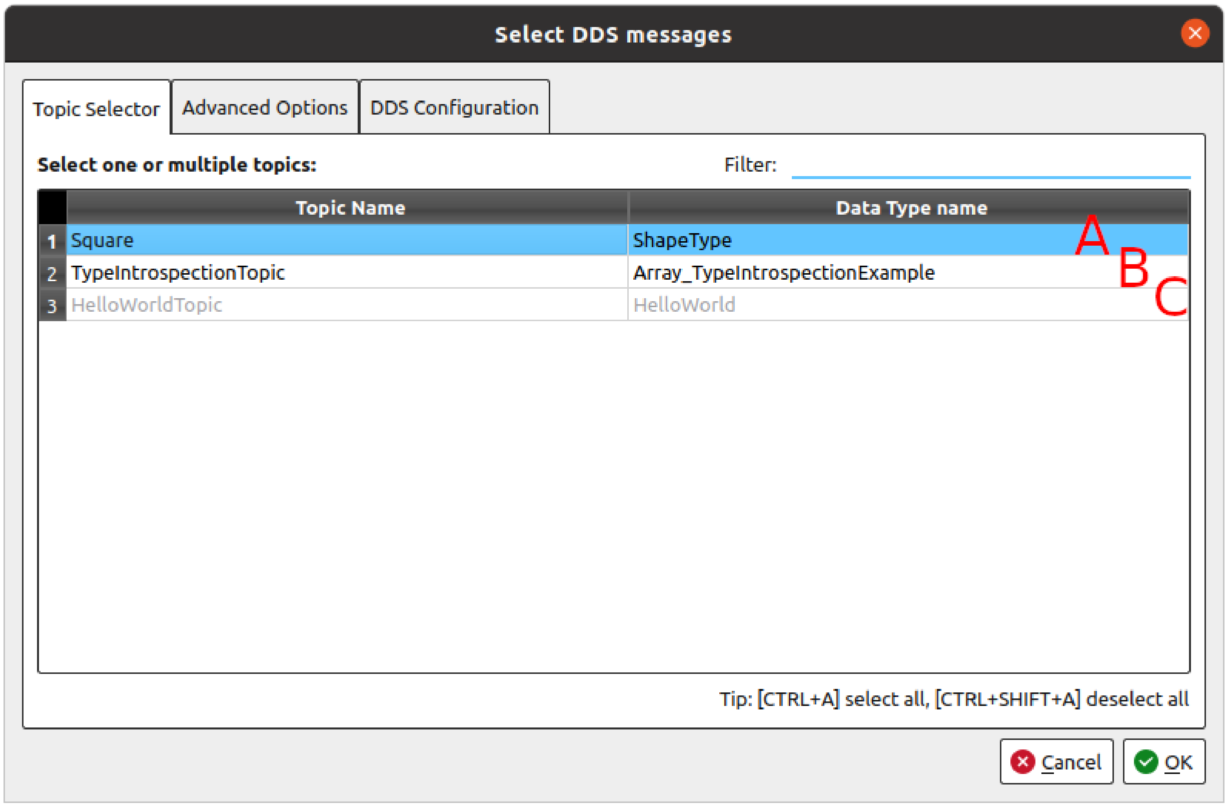
Tip: [CTRL+A] select all, [CTRL+SHIFT+A] deselect all

In order for *Fast DDS Visualizer Plugin* to receive data from a DDS topic, it is required to select the DDS topics. In each of the topics selected, the *Fast DDS Visualizer Plugin* will create a *DataReader* and receive the data, allowing to plot it afterwards. The following capture shows the *topic selector* dialog.

Select topics

In this tab it is shown every DDS Topic that is discovered by the internal *Domain Participant* in the current *Domain Id*. Each topic name has its related data type name. This is the data type of the data that will be published within this topic. Be aware that some data types may not be known by the internal *Fast DDS* application, and must be discovered via Dynamic Types. Refer to the following section *Configure Dynamic Data Types in Fast DDS* for more information.

- In order to select a topic to be visualized, click on the topic name or data type name *(A)*.
- In order to deselect a topic, click it again and it will be unselected *(B)*.
- Those topics which data types have not been discovered by the application could not be selected *(C)*. This is because the internal *Fast DDS* application could not create a *DataReader* in those topics, because it will not be able to know the data type it is reading.



	Topic Name	Data Type name
1	Square	ShapeType
2	TypeIntrospectionTopic	Array_TypeIntrospectionExample
3	HelloWorldTopic	HelloWorld

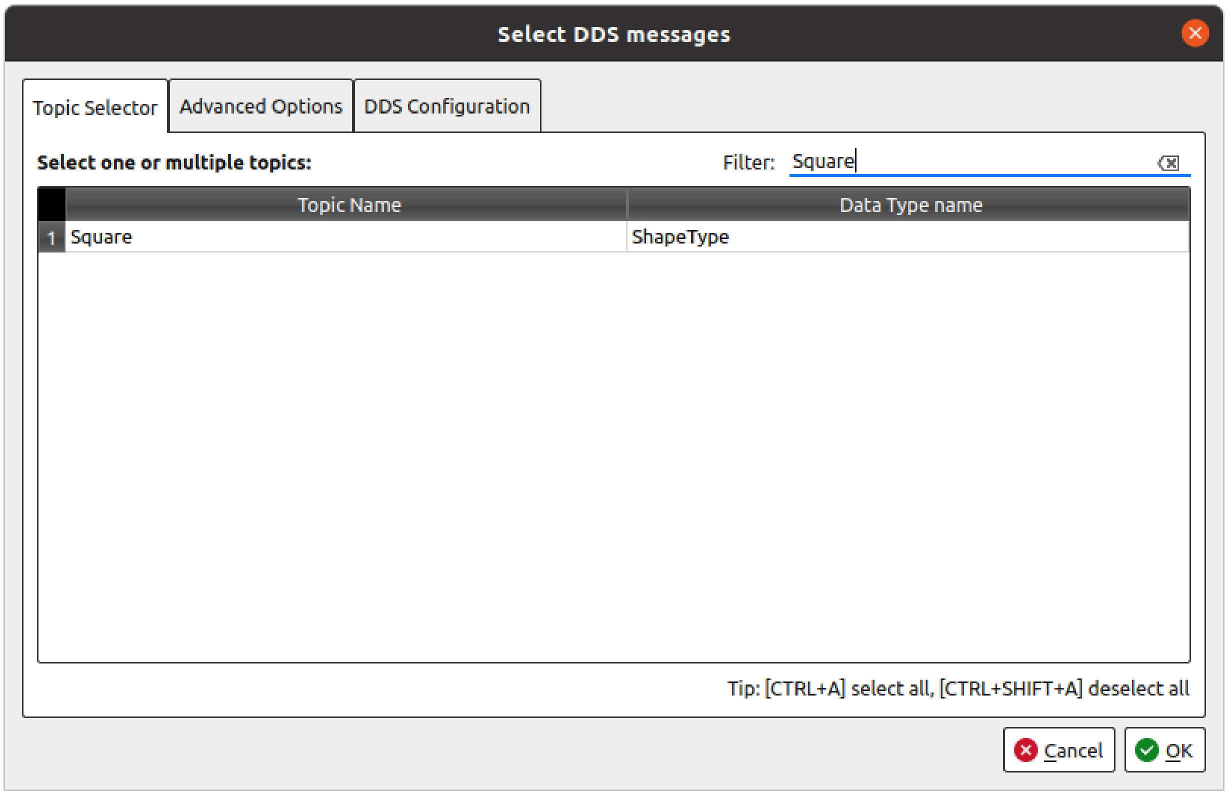
Tip: [CTRL+A] select all, [CTRL+SHIFT+A] deselect all

Select all topics

In order to select all possible topics discovered, press `Ctrl+A`. In order to deselect all topics, press `Ctrl+Shift+A`.

Filter topics

There is a filter in the upper right corner of the tab that allows to filter topics by name. Writing plain text in this text box will hide those topics that do not contain such text in their names. Be aware that those topics selected before filtering applies will still be selected even if hidden.



Filter: Square

	Topic Name	Data Type name
1	Square	ShapeType

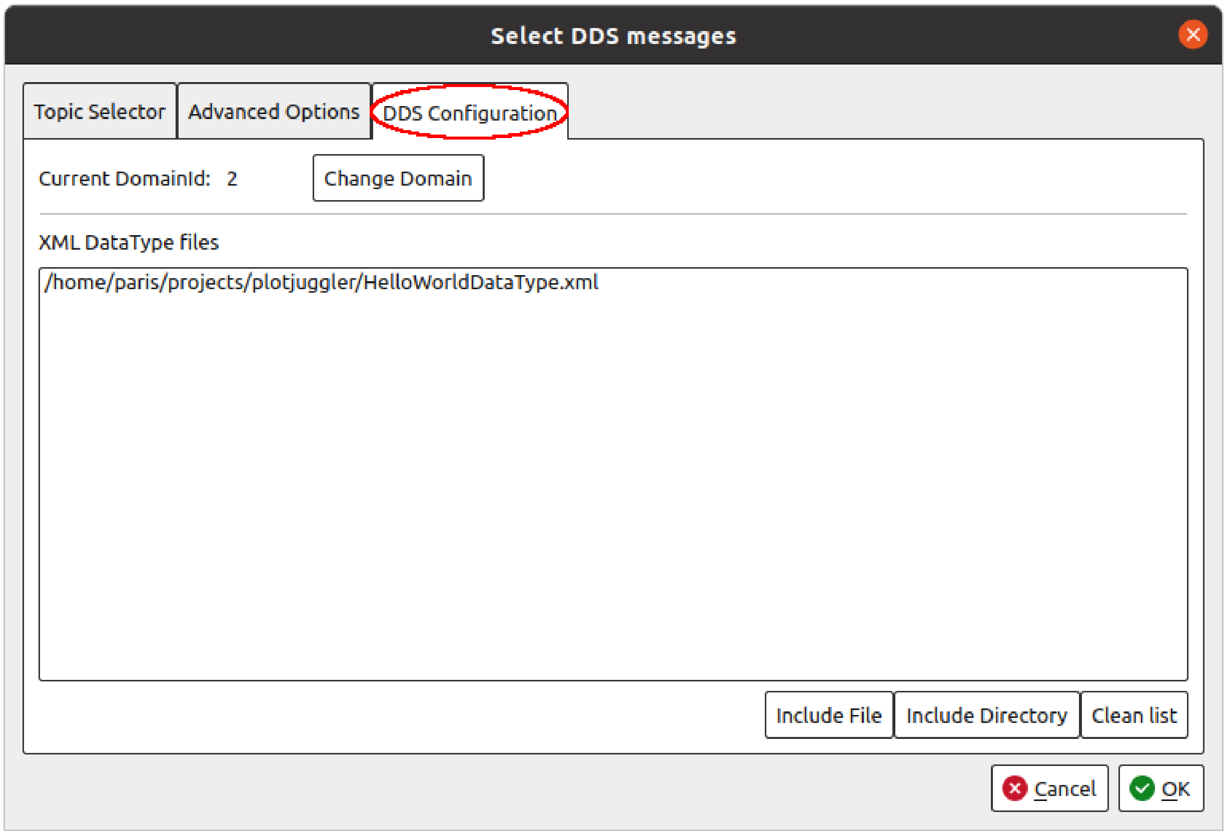
Tip: [CTRL+A] select all, [CTRL+SHIFT+A] deselect all

2.10.2 DDS Configuration

In this tab is configured the internal *Domain Participant* of the *Fast DDS Visualizer Plugin*.

DDS Domain Id

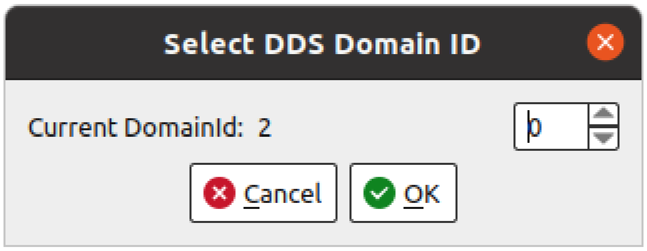
To select the DDS Domain Id of the internal *Fast DDS Domain Participant*, press `Change Domain` button. This pops up a new Dialog to choose a new *Domain Id*.



Current DomainId: 2

XML DataType files

/home/paris/projects/plotjuggler/HelloWorldDataType.xml



Current DomainId: 2

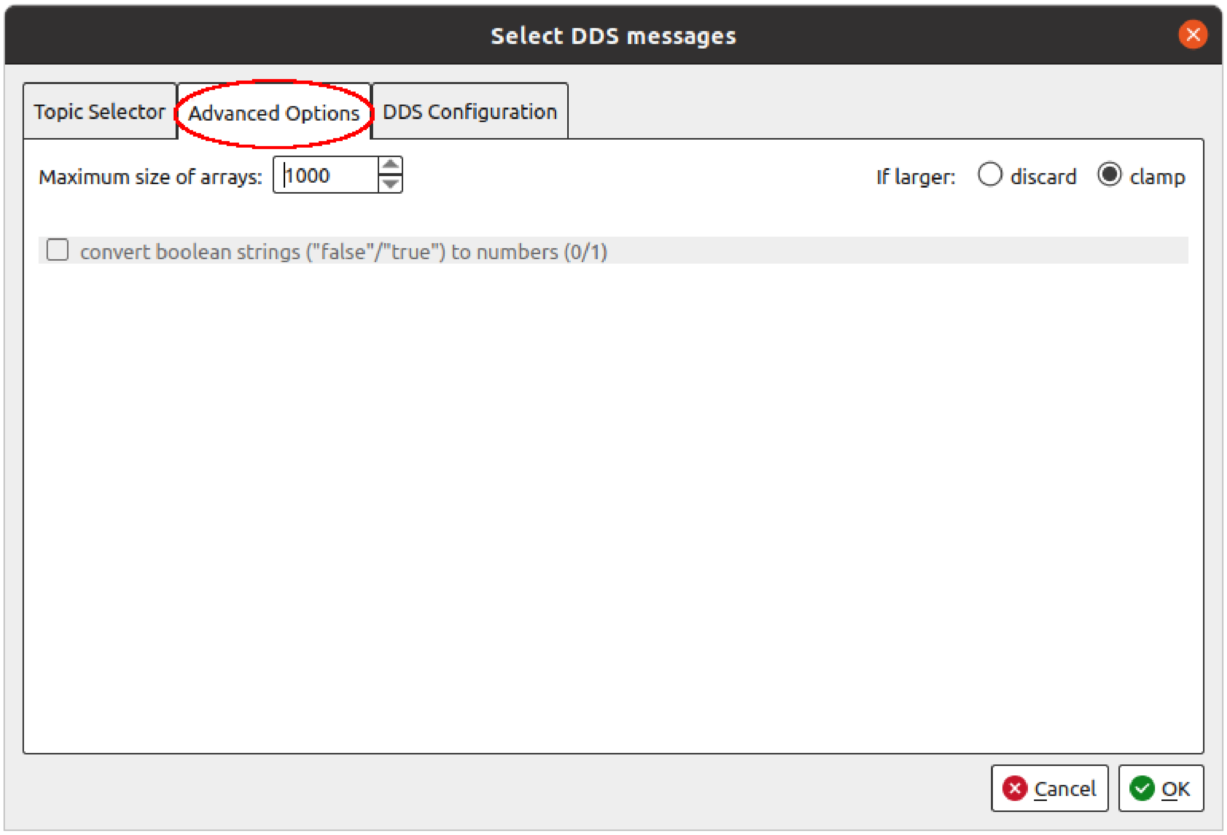
0

XML Data Types

In order to explicitly specify a Data Type for a topic, XML files with data type descriptions could be imported. Using the buttons `Include File` or `Include Directory` the user can select an XML file or all the XML files in a directory, and they will be parsed by *Fast DDS* and their data types will be known for the application hereinafter. Check section *XML Data Type* for more information.

2.10.3 Advance Options

This section is meant to configure advance options regarding internal values of data series.



Maximum size of arrays: 1000

If larger: ○ discard ● clamp

□ convert boolean strings ("false"/"true") to numbers (0/1)

Maximum array size

In some use cases, an image could be tried to display as a time series of many integer values, which could make the application unresponsive when working with big images. Due to very big amount of values, this could make the application highly unresponsive. In order to avoid this, an array size limit can be set in this section. When array data is bigger than this limit, two options can be chosen:

- `discard`: this data will be discarded.
- `clamp`: there will only be shown the first *N* values of the array, being *N* the maximum array size.

2.11 Configure Dynamic Data Types in Fast DDS

Fast DDS Visualizer Plugin uses Dynamic Types in order to be able to introspect the data type and read the internal data. If the application does not know the data type, it will not be able to connect to such topic. There are two ways of retrieving the data type information to the application:

2.11.1 Automatically discover Data Type

In order for *Fast DDS Visualizer Plugin* to discover automatically the data type of a topic, the *Fast DDS* publisher that may interact with the application must be configured to send this information. There are two main configurations that have to be considered:

Generate DataType

In order for *Fast DDS* to be able to retrieve the data type description, it must have the TypeInformation. In case that the *Fast DDS* application is already using the *Dynamic Types* API, this TypeInformation will be already available for the `DomainParticipant`.

Otherwise, if the data type is generated via Fast DDS Gen, it is required to generate it with option `-typeobject`. This generates two auxiliary `cpp` files that include the type information of the data type. Then, the data must be registered in the *Fast DDS* application before creating the publisher, using the function provided in `<DataTypeName>TypeObject.hpp` file `register<DataTypeName>Types()`.

```
// In case the DataType is HelloWorld, the HelloWorldTypeObject.hpp will define a␣
↪function
// called registerHelloWorldTypes() that registers the data type in the whole the␣
↪process.
registerHelloWorldTypes();
```

Publish Type Information

In order for the *DomainPartipant* in the *Fast DDS* application to be able to retrieve the data type information, it requires to be configured as a TypeLookup Server. To do so, add the following configuration to the *Domain Participant QoS*:

```
// Set DomainParticipantQoS as a TypeLookup Server
DomainParticipantQos qos;
qos.wire_protocol().builtin.typelookup_config.use_server = true;
```

Even when it is configured by default, make sure the `auto_fill_type_information` value of the `TypeSupport` is `true`.

2.11.2 XML Data Type

If the automatic discovery of the data type is not possible, an XML file with the data type description could be loaded, so the internal *Fast DDS Visualizer Plugin* is able to know the data type to subscribe to the topic. This XML should be added as a DDS configuration in the following tab when starting the plugin.

XML Data Type format

The format and types supported by *Fast DDS* are described in the following section of the Fast DDS documentation.

XML Data Type example

In the following snippet, there is an example of an XML file with a data type description. This example file will allow the *Fast DDS Visualizer Plugin* to discover the data type of the topic used in the *Fast DDS* HelloWorld Example.

```xml
<types>
    <type>
        <struct name="HelloWorld">
            <member name="index" type="uint32"/>
            <member name="message" type="string"/>
        </struct>
    </type>
</types>
```

Some XML files with data type description examples could be found in the plugin directory.

Warning: XML Data Type Description does not support keyed topics.

Warning: Dynamic Types in Fast DDS do not support complex data structures of kind: `sequence<struct>`.

2.12 Linux installation from sources

The instructions for installing the *PlotJuggler eProsima Edition* from sources and the required *Qt* installation are provided in this page. It is organized as follows:

2.12.1 Dependencies installation

PlotJuggler eProsima Edition depends on *eProsima Fast DDS* library, Qt, *PlotJuggler*, and certain Debian packages. This section describes the instructions for installing *PlotJuggler eProsima Edition* dependencies and requirements in a Linux environment from sources. The following packages will be installed:

- `foonathan_memory_vendor`, an STL compatible C++ memory allocator library.
- `fastcdr`, a C++ library that serializes according to the standard CDR serialization mechanism.
- `fastrtps`, the core library of eProsima Fast DDS library.

First of all, the *Requirements* and *Dependencies* detailed below need to be met. Afterwards, the user can choose whether to follow either the *colcon* or the *CMake* installation instructions.

Requirements

The installation of *PlotJuggler eProsima Edition* in a Linux environment from binaries requires the following tools to be installed in the system:

- *CMake, g++, pip, wget and git*
- *Colcon* [optional]
- *Gtest* [for test only]

CMake, g++, pip, wget and git

These packages provide the tools required to install *PlotJuggler eProsima Edition* and its dependencies from command line. Install CMake, g++, pip, wget and git using the package manager of the appropriate Linux distribution. For example, on Ubuntu use the command:

```
sudo apt install cmake g++ pip wget git
```

Colcon

colcon is a command line tool based on CMake aimed at building sets of software packages. Install the ROS 2 development tools (colcon and vcstool) by executing the following command:

```
pip3 install -U colcon-common-extensions vcstool
```

Note: If this fails due to an Environment Error, add the `--user` flag to the `pip3` installation command.

Gtest

Gtest is a unit testing library for C++. By default, *PlotJuggler eProsima Edition* does not compile tests. It is possible to activate them with the opportune CMake options when calling colcon or CMake. For more details, please refer to the *CMake installation* section. For a detailed description of the Gtest installation process, please refer to the Gtest Installation Guide.

Dependencies

PlotJuggler eProsima Edition has the following dependencies, when installed from sources in a Linux environment:

- *Asio and TinyXML2 libraries*
- *OpenSSL*
- *eProsima dependencies*
- *Qt 5.15*

Asio and TinyXML2 libraries

Asio is a cross-platform C++ library for network and low-level I/O programming, which provides a consistent asynchronous model. TinyXML2 is a simple, small and efficient C++ XML parser. Install these libraries using the package manager of the appropriate Linux distribution. For example, on Ubuntu use the command:

```
sudo apt install libasio-dev libtinyxml2-dev
```

OpenSSL

OpenSSL is a robust toolkit for the TLS and SSL protocols and a general-purpose cryptography library. Install OpenSSL using the package manager of the appropriate Linux distribution. For example, on Ubuntu use the command:

```
sudo apt install libssl-dev
```

eProsima dependencies

If it already exists in the system an installation of *Fast DDS* library, just source this library when building the *PlotJuggler eProsima Edition* by using the command:

```
source <fastdds-installation-path>/install/setup.bash
```

In other case, just download *Fast DDS* project from sources and build it together with *PlotJuggler eProsima Edition* using colcon as it is explained in section *Colcon installation*.

Qt 5.15

Qt 5.15 is needed in order to build *PlotJuggler eProsima Edition*. To install this Qt version, refer to Qt Downloads website.

Note: When going through the installation steps, make sure the box of component *Qt Charts* is checked.

2.12.2 Colcon installation

1. Create a `DDS-Visualizer-Plugin` directory and download the `.repos` file that will be used to install *PlotJuggler eProsima Edition* and its dependencies:

```
mkdir -p ~/DDS-Visualizer-Plugin/src
cd ~/DDS-Visualizer-Plugin
wget https://raw.githubusercontent.com/eProsima/fastdds-visualizer-plugin/main/
↪fastdds_visualizer_plugin.repos
vcs import src < fastdds_visualizer_plugin.repos
```

Note: In case there is an already *Fast DDS* and/or *PlotJuggler* installation in the system, it is not required to download and build every dependency in the `.repos` file. It is just needed to download and build the *PlotJuggler eProsima Edition* project having sourced its dependencies. Refer to section *eProsima dependencies* in order to check how to source *Fast DDS* and *PlotJuggler*, respectively.

2. Build the packages:

```
colcon build --cmake-args
```

3. Then run post-installation script `setup.bash`, so that `plotjuggler` is visible:

```
source ~/DDS-Visualizer-Plugin/install/setup.bash
```

Note: Being based on CMake, it is possible to pass the CMake configuration options to the `colcon build` command. For more information on the specific syntax, please refer to the CMake specific arguments page of the colcon manual.

2.12.3 CMake installation

This section explains how to compile *PlotJuggler eProsima Edition* with CMake, either *locally* or *globally*.

Local installation

1. Create a `Fast-DDS` directory where to download and build *PlotJuggler eProsima Edition* and its dependencies:

```
mkdir ~/DDS-Visualizer-Plugin
```

2. Clone the following dependencies and compile them using CMake.

   - Foonathan memory

```
cd ~/DDS-Visualizer-Plugin
git clone https://github.com/eProsima/foonathan_memory_vendor.git
mkdir foonathan_memory_vendor/build
cd foonathan_memory_vendor/build
cmake .. -DCMAKE_INSTALL_PREFIX=~/DDS-Visualizer-Plugin/install -DBUILD_
↪SHARED_LIBS=ON
cmake --build . --target install
```

   - Fast CDR

```
cd ~/DDS-Visualizer-Plugin
git clone https://github.com/eProsima/Fast-CDR.git
mkdir Fast-CDR/build
cd Fast-CDR/build
cmake .. -DCMAKE_INSTALL_PREFIX=~/DDS-Visualizer-Plugin/install
cmake --build . --target install
```

   - Fast DDS

```
cd ~/DDS-Visualizer-Plugin
git clone https://github.com/eProsima/Fast-DDS.git
mkdir Fast-DDS/build
cd Fast-DDS/build
cmake .. -DCMAKE_INSTALL_PREFIX=~/DDS-Visualizer-Plugin/install -DCMAKE_
↪PREFIX_PATH=~/DDS-Visualizer-Plugin/install
cmake --build . --target install
```

   - PlotJuggler

```
cd ~/DDS-Visualizer-Plugin
git clone https://github.com/facontidavide/PlotJuggler.git
mkdir PlotJuggler/build
cd PlotJuggler/build
cmake .. -DCMAKE_INSTALL_PREFIX=~/DDS-Visualizer-Plugin/install -DCMAKE_
↪PREFIX_PATH=~/DDS-Visualizer-Plugin/install
cmake --build . --target install
```

3. Once all dependencies are installed, install *PlotJuggler eProsima Edition*:

```
cd ~/DDS-Visualizer-Plugin
git clone https://github.com/eProsima/fastdds-visualizer-plugin.git
mkdir fastdds-visualizer-plugin/build
cd fastdds-visualizer-plugin/build
cmake .. \
    -DCMAKE_INSTALL_PREFIX=~/DDS-Visualizer-Plugin/install \
```

(continues on next page)

(continued from previous page)

```
    -DCMAKE_PREFIX_PATH=~/DDS-Visualizer-Plugin/install
cmake --build . --target install
```

Note: By default, *PlotJuggler eProsima Edition* does not compile tests. However, they can be activated by downloading and installing Gtest and building with CMake option -DBUILD_TESTS=ON.

4. Finally, extend LD_LIBRARY_PATH and PATH environment variables:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/DDS-Visualizer-Plugin/install/lib
export PATH=$PATH:~/DDS-Visualizer-Plugin/install/bin
```

Note: Consider adding these lines to your ~/.bashrc for permanent effect.

Global installation

To install *PlotJuggler eProsima Edition* system-wide instead of locally, remove all the flags that appear in the configuration steps of Fast-CDR, Fast-DDS, and DDS-Visualizer-Plugin, and change the first in the configuration step of foonathan_memory_vendor to the following:

```
-DCMAKE_INSTALL_PREFIX=/usr/local/ -DBUILD_SHARED_LIBS=ON
```

2.12.4 Run an application

To run the *PlotJuggler eProsima Edition* application, do not forget to source the *Fast DDS* libraries if installed from *colcon*, or extend LD_LIBRARY_PATH and PATH environment variables if *CMake* installation was used. Then execute *plotjuggler*:

```
# If built has been done using colcon, all projects could be sourced as follows
source install/setup.bash

# If built has been done using cmake directly, extend following environment variables
# with your installation lib/ and bin/ paths:
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/DDS-Visualizer-Plugin/install/lib
export PATH=$PATH:~/DDS-Visualizer-Plugin/install/bin

# Finally run plotjuggler
plotjuggler
```

Source Plugin

In case the *PlotJuggler eProsima Edition* binary compiled `libfastdds_visualizer_plugin_datastream_plugin.so` is not in the same folder than the *PlotJuggler eProsima Edition* executable `plotjuggler`, the plugin will not be automatically loaded. For this purpose, the user must add the binary path in the application. Once running the *PlotJuggler eProsima Edition* executable, go to *App -> Preferences -> Plugins* and add the folder path that contains the binary of the plugin `libfastdds_visualizer_plugin_datastream_plugin.so`. In case of colcon installation, this folder is `<workspace_path>/install/fastdds_visualizer_plugin/bin`.

To load new plugins added, the application must be restarted. New plugins added will be stored within the Qt configuration, and will be automatically loaded every time the application is started.

Note: If the plugin does not load properly or app crashes, check that the *Fast DDS* libraries have been correctly exported to the path. Check *Run an application* steps.

2.12.5 Useful Links

2.13 CMake options

Fast DDS Visualizer Plugin provides several CMake options to build the documentation and tests.

Warning: These options are only for developers who installed the *Fast DDS Visualizer Plugin* following the compilation steps described in *Linux installation from sources*.

Option	Description	Possible values	Default
`CMAKE_BUILD_TYPE`	Type of build (optional). Setting `CMAKE_BUILD_TYPE` to `Debug` enables debugging and console logging.	`Debug` `Release`	`Release`
`BUILD_TESTS`	Build the *Fast DDS Visualizer Plugin* application and documentation tests. Setting `BUILD_TESTS` to `ON` sets `BUILD_DOCUMENTATION_TESTS` to `ON`.	`OFF ON`	`OFF`
`BUILD_DOCUMENTATION_TESTS`	Build the *Fast DDS Visualizer Plugin* documentation tests. It is set to `ON` if `BUILD_TESTS` is set to `ON`.	`OFF ON`	`OFF`
`BUILD_DOCUMENTATION`	Build the *Fast DDS Visualizer Plugin* documentation. It is set to `ON` if `BUILD_TESTS_DOCUMENTATION` is set to `ON`.	`OFF ON`	`OFF`

2.14 PlotJuggler eProsima Edition within Fast DDS Suite

eProsima provides the *eProsima Fast DDS Suite* Docker image for those who want a quick demonstration of Fast DDS running on an Ubuntu platform. It can be downloaded from eProsima's downloads page.

1. This Docker image was built for Ubuntu 20.04 (Focal Fossa).

   To run this container you need Docker installed. From a terminal run

   ```
   $ sudo apt install docker.io
   ```

2. Download the compressed Docker image in `.tar` format from the eProsima's downloads page. It is strongly recommended to download the image corresponding to the latest version of *Fast DDS Suite*.

3. Extract the image by executing the following command:

   ```
   docker load -i ubuntu-fastdds-suite:<version>.tar
   ```

   where `version` is the downloaded version of *Fast DDS Suite*.

4. Run the Docker container executing the following command:

   ```
   xhost local:root
   docker \
       -it \
       --net=host \
       --privileged -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix \
       ubuntu-fastdds-suite:<version>
   ```

   After executing the previous command you should be able to enter a terminal shell within a docker with *Fast DDS* already installed and sourced, and several *eProsima* applications.

In the following Documentation you can find a detailed description of the Docker image and the features and applications available.

2.14.1 PlotJuggler eProsima Edition

To launch the *Fast DDS Visualizer Plugin*, from a terminal run

```
$ plotjuggler
```

2.14.2 Other eProsima applications

These are some of the features added in the *Fast DDS Suite* that may be useful for users of the *Fast DDS Visualizer Plugin*.

Shapes Demo

To launch the Shapes Demo, from a terminal run

```
$ ShapesDemo
```

eProsima Shapes Demo usage information can be found on the Shapes Demo First Steps.

Fast DDS Monitor

To launch the Fast DDS Monitor, from a terminal run

```
$ fastdds_monitor
```

eProsima Fast DDS Monitor usage information can be located on the Fast DDS Monitor User Manual.

Fast DDS Examples

Included in this Docker container is a set of binary examples that showcase several functionalities of the Fast DDS libraries. These examples' path can be accessed from a terminal by typing

```
$ goToExamples
```

From this folder you can access all *Fast DDS* examples.

2.15 Version v1.0.0

This is the first release of eProsima *Fast DDS Visualizer Plugin*.

This release includes the *Fast DDS Visualizer Plugin*, a **new Plugin** for the *PlotJuggler* application, supporting the configuration and visualization of DDS data powered by *Fast DDS*.

This release includes the following **User Interface features**:

- New *Fast DDS Visualizer Plugin DataStream* plugin for *PlotJuggler*.
- Dialogs and menus to configure *Fast DDS Visualizer Plugin*.

This release includes the following **Configuration features**:

- Select a Domain Id to discover and subscribe to topics.
- Load XML files with data type descriptions.
- Array clamp or discard with respect to its size.
- Select DDS topics whenever the data type is known.

This release includes the following **Discover features**:

- Discover every topic in the Domain Id.
- Automatically discover data types when the external *Domain Participant* is configured as a *Type Lookup Server*.

This release includes the following **DDS features**:

- Allow UDP, TCP, SHM and data sharing transport communication.
- Allow dynamic discovery of new topics and data types.

This release includes the following **Documentation features**:

- This same documentation.