

# Utilities Package

---

## Utilities Package

---

### Description:

This page discusses the *utilities\_package* in detail. The main purpose of this package is to process raw bag files. There are two nodes included:

- **Frame Saver Node** - subscribes to published image messages from bag files and writes each frame to a folder
- **Video Writer Node** - writes a video file from the saved frames

Using these two nodes, you can siphon image data and video data from your recorded bag files for further processing!

---

### Requirements:

- Ubuntu Focal Fossa
  - ROS2 Foxy Fitzroy
  - Arena SDK for Linux
  - C++17 or higher
- 

## Configuration and Launch Files

Before using this package, make sure ALL paths in the configuration and launch files are set correctly. This will especially cause issues when cloning the repository to a new machine as the paths cloned from the remote repo are (most likely) not valid for the new local repo.

**Note:** All paths are set globally. Paths are also rarely set outside of configuration/launch files. This allows the user to point to different files in a more automated manner. Editing the configuration file DOES NOT require you to rebuild the package, but editing the launch file DOES!

### Configuration File:

`video_writer_config.yaml` is shown below:

```
src > utilities_package > config > / video_writer_config.yaml
1  /**
2   * ros_parameters:
3   |   input_path: "/home/tahnt/T3_Repos/post_process_packages/ros2_ws/src/utilities_package/frames"
4   |   frame_name: "/frame"
5   |   extension: ".jpg"
6   |   output_path: "/home/tahnt/T3_Repos/post_process_packages/ros2_ws/src/utilities_package/output/example_video.mp4"
7   |   fps: 30.0
```

There are only configuration and launch files for the **Video Writer Node**. The **Frame Saver** has paths set within code - TODO: make frame saver a more automated process with configs/launch files!

ROS2 Parameters Configured:

- **input\_path**: Path to input (WITHOUT name of the file!) This should point into the folder that each frame will be saved. Once set, it should not need to be edited.
- **frame\_name**: Name of saved frame (each distinct frame is appended with a count)
- **extension**: Extension/format of saved frame
- **output\_path**: Path of the output video file
- **fps**: Desired FPS of the video. There is a tuning process - does not always perfectly sync to `$ ros2 topic hz` when playing back a bag file.
  - **Note**: ROS2 messages are NOT published over a constant frequency. The FPS approximation should always be considered when looking at post processed data! Data should be consider as "per frame" instead of "over time".

**Launch File:**

`video_writer_launch.py` is shown below:

```
src > utilities_package > launch > video_writer_launch.py > generate_launch_description
1  from launch import LaunchDescription
2  from launch_ros.actions import Node
3
4  def generate_launch_description():
5
6      config_path = "/home/tahnt/T3_Repos/post_process_packages/ros2_ws/src/utilities_package/config/video_writer_config.yaml"
7
8      return LaunchDescription([
9          Node(
10              package="utilities_package",
11              executable="video_writer",
12              name="video_writer",
13              output="screen",
14              emulate_tty=True,
15              parameters=[config_path]
16          ),
17      ])

```

Typically, the launch file does not need to be edited often. Make sure **config\_path** correctly points to `video_writer_config.yaml`.

## Configuring `frame_saver.cpp`

Before running `frame_saver.cpp` be sure to configure the save directory and topic name correctly!

The save directory is shown below. This should MATCH the corresponding configurations set in `video_writer_config.yaml`. Once set correctly, it should not need to be edited unless a change in path or name is desired.

```
21  //Initialize save directory
22  std::string save_frames_dir = "/home/tahnt/T3_Repos/post_process_packages/ros2_ws/src/utilities_package/frames";
23  std::string save_frames_name = "/frame";
24  std::string save_frames_ext = ".jpg";
25
```

Finally and most importantly, make sure that the topic name that is being subscribed to matches the corresponding topic name that the bag file will publish to. To check the topic name of a bag file, you can either use `$ ros2 bag info <bag_file_name>` (shown below) or play the bag using `$ ros2 bag play <bag_file_name>` followed by `$ ros2 topic list` in another sourced terminal. The first is easier if you're just trying to get the name of the topic.

The screenshot shows a terminal window with two tabs. The left tab displays the output of the command `ros2 bag info 1280x720_lat_bag1`. The right tab shows a portion of a C++ file named `frame_saver.cpp`.

**Terminal Output (Left Tab):**

```
tahnt@pelican-glide: ~/Documents/bagfiles/10_3_23_cam_data$ ros2 bag info 1280x720_lat_bag1
Files: 1280x720_lat_bag1_0.db3
Bag size: 3.3 GiB
Storage id: sqlite3
Duration: 53.569s
Start: Oct 3 2023 10:56:29.824 (1696355789.824)
End: Oct 3 2023 10:57:23.394 (1696355843.394)
Messages: 1270
Topic information: Topic: /tri028s_cc/stream | Type: sensor_msgs/msg/Image | Count: 1270 | Serialization Format: cd...
```

**C++ Code (Right Tab):**

```
83 int main(int argc, char ** argv)
84 {
85     rclcpp::init(argc, argv);
86     rclcpp::NodeOptions options;
87     rclcpp::Node::SharedPtr node = rclcpp::Node::make_shared("image_listener", options);
88     cv::namedWindow("OpenCV Playback");
89     cv::startWindowThread();
90     image_transport::ImageTransport it(node);
91     image_transport::Subscriber sub = it.subscribe("tri028s_cc/stream", 1, imageCallback);
92     rclcpp::spin(node);
93     cv::destroyWindow("OpenCV Playback");
94
95     return 0;
96 }
```

The topics circled in red in the images above should match to successfully use `frame_saver.cpp`!

## Using the Package

Once all paths are set correctly for each node, running is simple. Follow the steps below to save frames from a bag file and write them to a video file (also included in `README.md`):

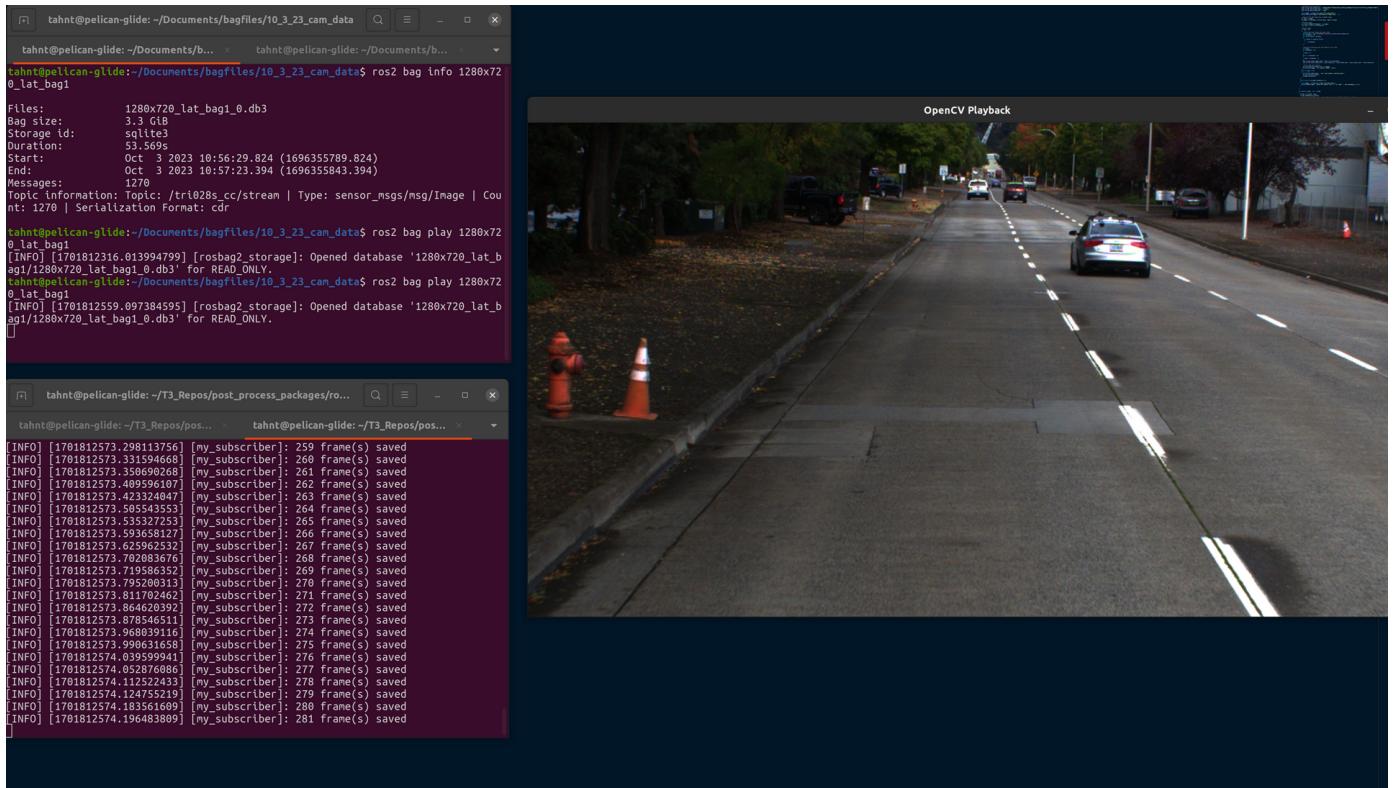
### Before Use:

- Make sure ALL PATHS ARE SET CORRECTLY in `frame_saver.cpp`
- Make sure ALL PATHS ARE SET CORRECTLY in the launch and config files
- These steps assume you have already created a workspace folder and a `/src` directory within it

### Save Stream Images Steps:

1. Navigate into the `/src` directory of your workspace and clone the repo using `git clone`
2. Navigate back into the workspace directory and source `$ source /opt/ros/foxy/setup.bash`
3. Build package `$ colcon build` or `$ colcon build --packages-select <package_name>`
4. Open a new terminal and source it `$ . install/setup.bash`
5. Run executable `$ ros2 run <package_name> <executable_name>` in this case it is `$ ros2 run utilities_package frame_saver`  
**Note:** This should be executed before playing the bag file to ensure you capture all frames.
6. In a new terminal navigate to the directory of your desired bag file and source it using `$ source /opt/ros/foxy/setup.bash`
7. Play bag file `$ ros2 bag play <bag_name>`

If executed correctly, a playback window is displayed and the terminal logs frame counts (shown below):



```
tahnt@pelican-glide: ~/Documents/bagfiles/10_3_23_cam_data$ ros2 bag info 1280x720_lat_bag1
Files:          1280x720_lat_bag1_0.db3
Bag size:      3.3 GB
Storage id:    sqlite3
Duration:      33.569s
Start:         Oct 3 2023 10:56:29.824 (169355789.824)
End:          Oct 3 2023 10:57:23.394 (169355843.394)
Messages:       1278
Topic information: Topic: /tri028s_cc/stream | Type: sensor_msgs/msg/Image | Count: 1278 | Serialization Format: cdr

tahnt@pelican-glide: ~/Documents/bagfiles/10_3_23_cam_data$ ros2 bag play 1280x720_lat_bag1
[INFO] [1701812316.013904799] [rosbag2_storage]: Opened database '1280x720_lat_bag1' for READ_ONLY.
tahnt@pelican-glide: ~/Documents/bagfiles/10_3_23_cam_data$ ros2 bag play 1280x720_lat_bag1
[INFO] [1701812559.097384595] [rosbag2_storage]: Opened database '1280x720_lat_bag1' for READ_ONLY.

[OpenCV Playback]
[INFO] [1701812573.298113756] [my_subscriber]: 259 frame(s) saved
[INFO] [1701812573.331594668] [my_subscriber]: 260 frame(s) saved
[INFO] [1701812573.350690268] [my_subscriber]: 261 frame(s) saved
[INFO] [1701812573.409596107] [my_subscriber]: 262 frame(s) saved
[INFO] [1701812573.423324047] [my_subscriber]: 263 frame(s) saved
[INFO] [1701812573.423324047] [my_subscriber]: 264 frame(s) saved
[INFO] [1701812573.423324047] [my_subscriber]: 265 frame(s) saved
[INFO] [1701812573.535327253] [my_subscriber]: 266 frame(s) saved
[INFO] [1701812573.593658127] [my_subscriber]: 267 frame(s) saved
[INFO] [1701812573.625962532] [my_subscriber]: 268 frame(s) saved
[INFO] [1701812573.702083676] [my_subscriber]: 269 frame(s) saved
[INFO] [1701812573.719586352] [my_subscriber]: 270 frame(s) saved
[INFO] [1701812573.795209313] [my_subscriber]: 271 frame(s) saved
[INFO] [1701812573.846290262] [my_subscriber]: 272 frame(s) saved
[INFO] [1701812573.870546511] [my_subscriber]: 273 frame(s) saved
[INFO] [1701812573.960839116] [my_subscriber]: 274 frame(s) saved
[INFO] [1701812573.990631658] [my_subscriber]: 275 frame(s) saved
[INFO] [1701812574.039599441] [my_subscriber]: 276 frame(s) saved
[INFO] [1701812574.052876086] [my_subscriber]: 277 frame(s) saved
[INFO] [1701812574.112522433] [my_subscriber]: 278 frame(s) saved
[INFO] [1701812574.124755219] [my_subscriber]: 279 frame(s) saved
[INFO] [1701812574.183561699] [my_subscriber]: 280 frame(s) saved
[INFO] [1701812574.196483891] [my_subscriber]: 281 frame(s) saved
```

### Write Video File Steps:

1. Open a new terminal in your working directory and source using `$ . install/setup.bash`
2. Run launch file `$ ros2 launch <package_name> <launch_file_name>` in this case it is `$ ros2 launch utilities_package video_writer_launch.py`

If executed correctly, a very fast playback will display as the frames are written into a video file shown below. The terminal will log the write count and display a 'write successful' message indicating that the process is complete and pointing to the path of the output video (shown below):

The image shows a ROS2 workspace window titled "tahnt@pelican-glide: ~/T3\_Re..." containing three terminal tabs. The top tab displays a "CV Image" of a street scene with a car in the center. The bottom-left tab shows the terminal command "tahnt@pelican-glide: ~/T3\_Re..." and the bottom-right tab shows "tahnt@pelican-glide: ~/T3\_Re...". Both tabs show the same log output from the "Frame\_Saver.cpp" file, which tracks the number of frames written by a "video\_writer". The log shows approximately 1195 frames being written, followed by a "Write Successful" message and the path to the output video file: "/home/tahnt/T3\_Repos/post\_process\_packages/ros2\_ws/src/utilities\_package/output/example\_video.mp4".

```
[video_writer-1] [INFO] [1701812696.401680187] [video_writer]: 1195 frame(s) written
[video_writer-1] [INFO] [1701812696.423475700] [video_writer]: 1196 frame(s) written
[video_writer-1] [INFO] [1701812696.445426758] [video_writer]: 1197 frame(s) written
[video_writer-1] [INFO] [1701812696.468606509] [video_writer]: 1198 frame(s) written
[video_writer-1] [INFO] [1701812696.490830673] [video_writer]: 1199 frame(s) written
[video_writer-1] [INFO] [1701812696.512316267] [video_writer]: 1200 frame(s) written
[video_writer-1] [INFO] [1701812696.534411095] [video_writer]: 1201 frame(s) written
[video_writer-1] [INFO] [1701812696.557024380] [video_writer]: 1202 frame(s) written
[video_writer-1] [INFO] [1701812696.578550649] [video_writer]: 1203 frame(s) written
[video_writer-1] [INFO] [1701812696.601451682] [video_writer]: 1204 frame(s) written
[video_writer-1] [INFO] [1701812696.622957058] [video_writer]: 1205 frame(s) written
[video_writer-1] [INFO] [1701812696.645344535] [video_writer]: 1206 frame(s) written
[video_writer-1] [INFO] [1701812696.667595914] [video_writer]: 1207 frame(s) written
[video_writer-1] [INFO] [1701812696.689396036] [video_writer]: 1208 frame(s) written
[video_writer-1] [INFO] [1701812696.710981008] [video_writer]: 1209 frame(s) written
[video_writer-1] [INFO] [1701812696.733090871] [video_writer]: 1210 frame(s) written
[video_writer-1] [INFO] [1701812696.755086728] [video_writer]: 1211 frame(s) written
[video_writer-1] [INFO] [1701812696.776913699] [video_writer]: 1212 frame(s) written
[video_writer-1] [INFO] [1701812696.800783286] [video_writer]: 1213 frame(s) written
[video_writer-1] [INFO] [1701812696.822493293] [video_writer]: 1214 frame(s) written
[video_writer-1] [INFO] [1701812696.844283308] [video_writer]: 1215 frame(s) written
[video_writer-1] [INFO] [1701812696.866671760] [video_writer]: 1216 frame(s) written
[video_writer-1] [INFO] [1701812696.869247020] [video_writer]: Write Successful
[video_writer-1] [INFO] [1701812696.869294896] [video_writer]: Video file written to: /home/tahnt/T3_Repos/post_process_packages/ros2_ws/src/utilities_package/output/example_video.mp4
```