Gazebo Ionic Command Line Interface Reference

This document provides a reference for selected Gazebo CLI subcommands, with color-coded commands and short descriptions for their options as equivalents to the ROS 2 cheatsheet.

gz sim

Launches a Gazebo simulation world.

- `gz sim <world.sdf>` : Launches the simulation with the specified SDF world file.
- `gz sim -v 4 <world.sdf>` : Launches the simulation with verbose output (error, warning, informational, and debugging messages).
- `gz sim -s <world.sdf>` : Runs the server only (headless mode, without GUI).
- `gz sim -g` : Runs the GUI only, connecting to a running server.
- `gz sim -r <world.sdf>` : Launches the simulation and immediately starts running (unpaused).

gz topic

Interacts with Gazebo topics.

- `gz topic -h` : Displays help information for the topic command.
- `gz topic -l` : Lists all active Gazebo topics.
- `gz topic -t` : Name of a topic. Required by: {-e -i -f -p}
- `gz topic -e -t <topic_name>` : Echoes data published to a specific topic.
- `gz topic -i -t <topic_name>` : Provides detailed information about a specific topic.
- `gz topic -f -t <topic_name>` : Calculates the frequency of a topic.
- `gz topic -m` : The type of the message to publish. Required by -p.
- `gz topic -d` : Duration to run, in seconds.
- `gz topic -p -t <topic_name> -m <msg_type> -d <float>` : Publishes data to a topic.

gz model

Manages Gazebo models (entities similar to nodes in ROS).

- `gz model -h` : Displays help information for the model command.
- `gz model --list` : Lists all active models in the simulation.
- `gz model -m <model_name> -p` : Outputs the pose of a specific model.

gz msg

Inspects Gazebo message types (interfaces).

- `gz msg -h` : Displays help information for the msg command.
- `gz msg -l` : Lists all available message types.
- `gz msg -i <msg_type>` : Displays the definition of a specific message type.

gz service

Interacts with Gazebo services.

- `gz service -h` : Displays help information for the service command.
- `gz service -l` : Lists all active Gazebo services.
- `gz service -s` : Name of a service. Required by: {-i, -r}
- `gz service -i -s <service_name>` : Provides detailed information about a specific service.
- `gz service --reqtype <request_type>` : Type of a request.
- `gz service --reptype <response_type>` : Type of a response.
- `gz service --timeout <integer>` : Timeout in milliseconds.
- `gz service --req <input_data>` : Requests a service. Can alternatively specify ”-r” instead of ”–req”.

  For example:

  ```
  gz service -s /echo \
      --reqtype gz.msgs.StringMsg \
      --reptype gz.msgs.StringMsg \
      --timeout 2000 \
      --req 'data: "Hello"'
  ```

gz param

List, set or get parameters To find a namespace, execute:
`gz service -l | grep '/list_parameters$'`

- `gz param -h` : Displays help information for the param command.
- `gz param -r <namespace_path>` : Namespace of the parameter registry to be queried. Required by all other flags. To find a namespace, execute

- `gz param -r <namespace_path> -l` : Get a list of the available parameters of the namespace.
- `gz param -r <namespace_path> -n <param_name>` : The parameter name.
- `gz param -r <namespace_path> -t <param_type>` : The parameter type.
- `gz param -r <namespace_path> -m <param_value>` : The parameter value.
- `gz param -r <namespace_path> -n <param_name> -g` : Gets a parameter. Requires the -n flag.
- `gz param -r <namespace_path> -s -n <param_name> -t <param_type> -m <param_dict>` : Sets a parameter with the specified type and value. Requires the -n, -t, -m flags.