

How to add a new drone wrapper

Drone wrappers are the connection point between the multi-drone platform and a drone's native API. Hence for the platform to interact with a drone type a wrapper is created to translate messages between the platform and the drone.

A new drone wrapper can be created by following these steps:

1. Copy the wrapper template file under */wrappers/object.cpp*
2. Rename both the file name and the first parameter of the `DRONE_WRAPPER(...)` line to the new drone's identifier. E.g. if the drone identifier is 'foo' the file will be renamed to *'/wrappers/foo.cpp'* and the first param '`DRONE_WRAPPER(foo, ...)`'.
3. Any required parameters on drone initialization can be declared by adding parameters to the `DRONE_WRAPPER` line. E.g. say the drone 'foo' requires an IP address and port number on initialisation, then the `DRONE_WRAPPER` line will be '`DRONE_WRAPPER(foo, ip_address, port_number)`'. When a user declares a drone for a mdp session using the 'add_drone' program, the user will be required to fill in these parameters. The values for these parameters will then be given back to the drone as a vector of strings in the '`on_init(std::vector<std::string>)`' function in the order they were provided in the `DRONE_WRAPPER` line. For drone 'foo' this means that the first element of the vector corresponds to the given 'ip_address' and the second the 'port_number'.
4. Next fill out the virtual functions so that they perform the given task using the drone's native API. Positions given are in absolute terms relative to the world's origin point in meters. Velocities are given in meters per second relative to world coordinates. All durations are given in seconds. Yaw is given in degrees, yaw rate in degrees per second.
5. Additional function details can be found in the doxygen documentation.
6. Once completed the wrapper will automatically be included in the platform's next compilation given it lies in the */wrappers/* folder and is named correctly as described above.