MinecraftBridge

Release 1.0.12

Dana Hughes

Jan 14, 2022

CONTENTS:

CHAPTER

ONE

LICENSE

MIT License

Copyright (c) 2020 Dana Hughes

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER

TWO

OVERVIEW

CHAPTER

THREE

INSTALLATION

3.1 Requirements

This module may be used with python version 3.4 and later. Installation is best performed using pip.

Interfacing through a MQTT broker requires the Eclipse Paho MQTT client package to be installed. This is most easily installed through *pip*:

```
pip install [--user] paho-mqtt
```

Messages generated by the bridge also leverate the *dateutil* package to parse and generate ISO formatted timestamps. This is installed easily through *pip*:

```
pip install [--user] python-dateutil
```

This module requires MinecraftElements. to correctly operate. While the *master* branch should be compatible with this version of *MinecraftBridge*, specific versions of *MinecraftElements* should be utilized to ensure compatibility. See the *MinecraftElements Compatibility Table* for compatible versions of *MinecraftElements* and *MinecraftBridge*.

MinecraftBridge Vesrion	MinecraftElements Version
0.4.0b1	0.3.2b
0.2.3	0.1.1

3.2 Installation using pip

This module can be installed as a package using *pip*. Assuming *pip* is installed, the package can be installed (for users) from the root folder by:

```
pip install --user -e .
```

Once installed, the module can be imported from arbitrary locations.

3.3 Adding as a Submodule to a Project

As an alternative, this repository could be cloned into existing code as a *submodule*. This can be done with the following command:

```
git submodule add https://gitlab.com/cmu_asist/MinecraftElements
```

If the generated folder is empty, a recursive update may need to be performed:

```
git submodule update --init --recursive
```

(See Working with Submodules)

Additionally, cloning a repo with this submodule will require the *recursive* flag:

```
git clone --recursive <project url>
```

3.4 Documentation

Documentation for this module can be built using Sphinx, which is also installed with pip. The source and build files for the documentation is in the *docs* folder.

3.4.1 Requirements

Note that build targets may require additional dependencies to be installed. For building a PDF of the documentation using Latex, basic Latex dependencies need to be installed. On Ubuntu-based systems, this can be done by installing the following packages using *aptitude*:

```
apt-get install texlive-latex-recommended texlive-latex-extra texlive-fonts-recommended
```

3.4.2 Building Documentation

To build a PDF of the *MinecraftBridge Manual*, from the *docs* folder, run the build command with the latexpdf target:

```
cd docs
make latexpdf
```

This will create a *build* subfolder, and *latex* subfolder within build. The final pdf file, *MinecraftBridge_Manual.pdf* will be written to the *latex* subfolder, which can be moved to the root directory easily. In Latex or OSX, this is done with:

```
mv build/latex/MinecraftBridge_Manual.pdf ..
```

While in Windows, this would be done by:

```
move build\latex\MinecraftBridge_Manual.pdf ..
```

CHAPTER

FOUR

ARCHITECTURE

The purpose of the *MinecraftBridge* package is to decouple messages generated from Minecraft (nominally with the ASISTMod) and consumers of the messages. The package employs a **`Bridge Pattern`_** to allow client code to interface with a consistent and stable set of message classes, as opposed to directly interacting with low-level messages and connections. The benefits to this include

1. Client code can be developed independent of the source of messages.
2. Changes to message formats can be resolved with bridge parsers, while maintaining the same message class interface, requiring no modification of client code.
3. Multiple clients can be attached to a single bridge, allowing one-to-many dependencies.
4. New bridges can be created to handle different methods to connect to or simulate Minecraft environments.

4.1 Class Diagram

The basic class diagram for MinecraftBridge is shown in Fig. 4.1. The package consists of two main categories of classes: the *Bridge* classes (*AbstractBridge*, *Bridge* and *FileBridge* in Fig. 4.1), and the *Message* classes.

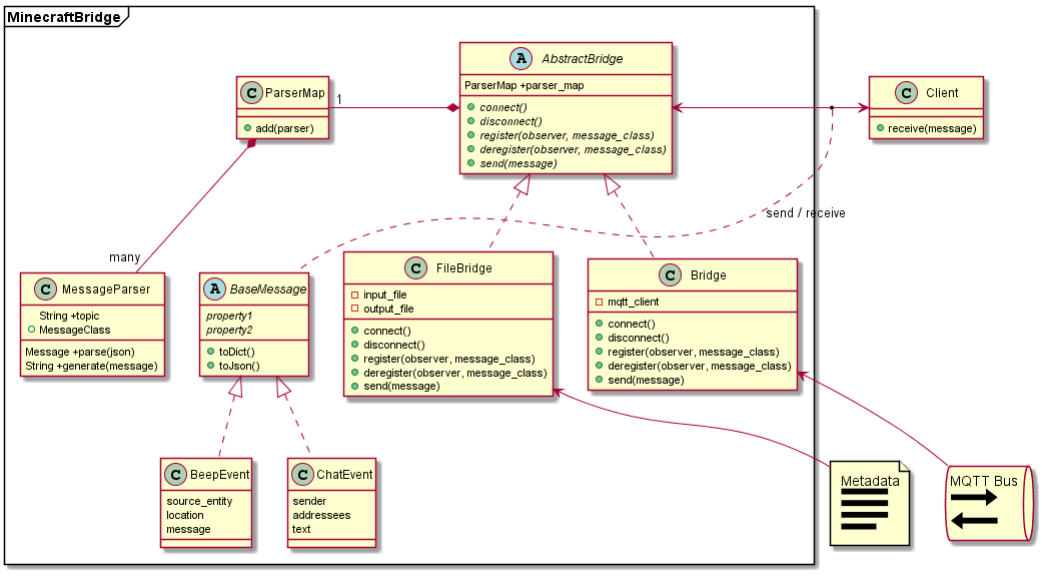

Fig. 4.1: Simplified class diagram of MinecraftBridge showing major components and concrete implementations.

The *Bridge* class is used to connect to an arbitrary Minecraft backend, and convert output of the connection to common *Message* objects. Note that *AbstractBridge* is not currently implemented, but is included to demonstrate a common

interface for *all* Bridge classes, and should be considered the means for client interaction (i.e., clients should not rely on specific concrete implementations). Bridge classes implement

All *Message* classes inherit from a *BaseMessage* class, which defines *toDict* and *toJson* to convert messages to dictionary and JSON string formats, respectively, and a *header* attribute and *addHeader* method for adding backend-specific header information. In practice client code should identify the *type* of message based on the class of the object (queried using, e.g., *message.__class__*), and handle the messages according to type.

Bridge Pattern:: https://refactoring.guru/design-patterns/bridge

CHAPTER

FIVE

USAGE

The typical use cases for MinecraftBridge is to provide a simple approach for an object to receive Minecraft messages from an arbitrary backend, without having to have the object be responsible for establishing, maintaining, or working with low-level connections, or parse backend-specific data. This typically involves the following steps:

1. Create a client object with the appropriate callback for receiving messages from the bus (i.e., *receive(self, message)*).
2. Create an instance of the Bridge suitable for connecting to the backend.
3. Register the client object with the bridge instance for each message type the client should receive.
4. Create the bridge connection.
5. (Optional) Generate messages and send to the Bridge connection.
6. Disconnect the bridge when complete.

5.1 Creating a Client Class

The only stipulation placed on a client object is that it must implement a *receive* method, which accepts a *message* argument. For example, a minimal class definition would be:

```
class ClientClass:
    """
    Minimal client class for connecting with a MinecraftBridge instance.
    """

    def receive(self, message):
        """
        Callback method for receiving messages from the Bridge instance.
        """

        # Define object's behavior here
        pass
```

In practice, the object will want to invoke different behavior based on the class of the received message. A simple pattern for managing message-specific behavior is to utilize a dictionary mapping message class to specific callback function, which can be created during object initialization. In the *receive* method, the mapping can be used to look up which method to call, and pass the message to that method. As an example, assuming the object wants to process the *BeepEvent* and *ChatEvent* messages, the *ClientClass* can be extended as follows:

```
from MinecraftBridge.messages import BeepEvent, ChatEvent

class ClientClass:
    """
    Minimal client class for connecting with a MinecraftBridge instance.
    """

    def __init__(self):

        # Message callback dictionary -- a simple approach for handling
        # which message types the client should receive, and what methods
        # to call when they are received.
        self.__message_callbacks = { BeepEvent: self.__onBeepEvent,
                                     ChatEvent: self.__onChatEvent
                                   }

    def __onBeepEvent(self, message):
        """
        Callback method for handling BeepEvent messages
        """

        # Define BeepEvent behavior here
        pass

    def __onChatEvent(self, message):
        """
        Callback method for handling ChatEvent messages
        """

        # Define ChatEvent behavior here
        pass

    def receive(self, message):
        """
        Callback method for receiving messages from the Bridge instance.
        """

        # Determine which method to call based on the message's class, and
        # pass the message to that method.

        # Note that it is possible that a message type is received that
        # isn't in the callback dictionary, so check this first.
        if not message.__class__ in self.__message_callbacks:
            return

        # Call the appropriate callback method, passing the message
        self.__message_callbacks[message.__class__](message)
```

5.2 Creating a Bridge

While specific Bridge classes are typically instantiated with specific arguments, once created, the interface should remain consistent between differen classes, allowing the client object to interact with bridges without consideration to the concrete implementation being used. Specifically, the client interacts with the bridge using the following methods:

- *register(client_object, message_class)*: Indicates that the provided client should receive message instances of the message class type. A client can register with a bridge multiple times (once for each message class of instance), as well as register with multiple bridges.
- *send(message)*: Method for allowing clients to *generate* instances of a message, and push the message to the bridge. This allows for client code to communicate back to the Minecraft backend, assuming such communication can be properly handled.

In addition, Bridge classes also implement the following methods to connect / disconnect to the corresponding Minecraft backend:

- *connect()*: Connect to the Minecraft backend, and start receiving messages.
- *disconnect()*: Disconnect from the Minecraft backend, and stop receiving messages.

As with *send*, the behavior invoked by these two methods will vary with specific Bridge classes. Additionally, the client object should *register* all desired message classes *prior* to invoking *connect* or *send*.

As an example, an instance of the *ClientClass* above can be connected to a *MinecraftBridge.mqtt.Bridge* instance in the following manner:

```
from MinecraftBridge.mqtt import Bridge as MessageBusBridge
from MinecraftBridge.messages import BeepEvent, ChatEvent

# Create instances of the ClientClass and MessageBusBridge
client = ClientClass()
bridge = MessageBusBridge()

# Register the client with the bridge, indicating that it wants to receive
# `BeepEvent` and `ChatEvent` messages
bridge.register(client, BeepEvent)
bridge.register(client, ChatEvent)

# Connect the bridge instance to the message bus, to start receiving messages
# Note that, when connecting, this bridge takes as arguments the host and
# port number of the MQTT server.
bridge.connect('localhost', 1883)
```

In typical use cases, it may make sense to allow the client to maintain an instance of the bridge as an attribute, allowing for interactions between the client and bridge (e.g., sending messages) to remain encapsulated in the client. In this case, extending the *ClientClass* initialization to take an instance of a bridge as an attribute allows for simplifying client registration for each messaage:

```
from MinecraftBridge.messages import BeepEvent, ChatEvent

class ClientClass:
    """
    Minimal client class for connecting with a MinecraftBridge instance.
    """
```

(continues on next page)

(continued from previous page)

```
    def __init__(self, bridge):

        # Keep the reference to the provided bridge as an attribute, to
        # simplify sending messages in the future
        self.__bridge = bridge

        # Message callback dictionary -- a simple approach for handling
        # which message types the client should receive, and what methods
        # to call when they are received.
        self.__message_callbacks = { BeepEvent: self.__onBeepEvent,
                                     ChatEvent: self.__onChatEvent
                                   }

        # Register each of the message classes in the callback dictionary
        # with the bridge
        for message_class in self.__message_callbacks.keys():
            self.__bridge.register(self, message_class)


    def __onBeepEvent(self, message):
        """
        Callback method for handling BeepEvent messages
        """

        # Define BeepEvent behavior here
        pass


    def __onChatEvent(self, message):
        """
        Callback method for handling ChatEvent messages
        """

        # Define ChatEvent behavior here
        pass


    def receive(self, message):
        """
        Callback method for receiving messages from the Bridge instance.
        """

        # Determine which method to call based on the message's class, and
        # pass the message to that method.

        # Note that it is possible that a message type is received that
        # isn't in the callback dictionary, so check this first.
        if not message.__class__ in self.__message_callbacks:
            return

        # Call the appropriate callback method, passing the message
        self.__message_callbacks[message.__class__](message)
```

(continues on next page)

(continued from previous page)

```
    def connect(self):
        """
        Connect the bridge to the Minecraft backend
        """

        self.__bridge.connect()


    def disconnect(self):
        """
        Disconnect the bridge from the Minecraft backend
        """

        self.__bridge.disconnect()
```

Using the above approach, a bridge can be created in conjunction with the *ClientClass*, and the addition of the *connect* and *disconnect* methods allows for interaction with a single object for both connection and message handling, while still allowing for versatility in backend connection. As an example, switching between using a *MessageBusBridge* and *FileBridge* simply involves changing the class and arguments used for the bridge; compare the code using a *MessageBusBridge*:

```
from MinecraftBridge.mqtt import Bridge as MessageBusBridge

client = ClientClass(MessageBusBridge())
client.connect()
```

and code using a *FileBridge*:

```
from MinecraftBridge.mqtt import FileBridge

client = ClientClass(FileBridge())
client.connect()
```

5.3 Handling Messages

Messages received by the client are generally lightweight objects, with properties corresponding to the contents or information contained in the events that produced the message. For many messages, properties are *aliased*, allowing, for instance, to refer to a location either as a tuple using the *location* property, or by accessing the individual x, y, z components. Additionally, messages should be considered *immutable*; attempts to set properties will generally raise an Exception.

As an example, *BeepEvent* messages contain *source_entity*, *message*, and *location* properties; *source_entity* and *location* have additional aliases. When a *BeepEvent* message is received, the client can simply access the values of these properties directly, with aliases providing equivalent behavior:

```
def __onBeepEvent(self, message):
    """
    Callback method for handling BeepEvent messages
    """
```

(continues on next page)

(continued from previous page)

```
    # These two properties are aliases, and will produce the same data
    source = message.source_entity
    source = message.sourceEntity

    message = message.message

    # Individual values in the `location` property can also be accessed
    # with `beep_x`, `beep_y`, and `beep_z`
    x, y, z = message.location
    x = message.beep_x
    y = message.beep_y
    z = message.beep_z
```

5.4 Generating and Sending Messages

In addition to receiving messages, Bridge instances also allow for sending messages. Generally speaking, messages are instantiated with a set of keyword argument specific to the message type. Additionally, messages are *immutable*; once instantiated, properties of the message instance cannot be modified. Once the message is generated, it is simply a matter of passing the message to the bridge instance using the *send* method. For example:

```
from MinecraftBridge.messages import BeepEvent
from MinecraftBridge.mqtt import Bridge as MessageBusBridge

# Create the bridge to the message bus and connect
bridge = MessageBusBridge()
bridge.connect('localhost', 1883)

# Create a BeepEvent message and send to the message bus through the bridge
message = BeepEvent(sourceEntity='Location Device',
                    message='beep beep',
                    location=(-2187,56,61))
bridge.send(message)
```

When generating a message, the following exceptions may be raised:

- *MissingMessageArgumentException*: raised when an expected keyword argument is not provided during creation.
- *MalformedMessageCreationException*: raised when a specific keyword argument cannot be coerced into the format expected by the message class. A common example is when a *location* argument is passed that isn't an tuple, list, or similar data type with three elements.

CHAPTER

SIX

MESSAGES

This section summarizes individual message classes used by *MinecraftBridge*. As the primary purpose of this package is for interfacing with the DARPA ASIST Minecraft Testbed, message classes primarily reflect the message specifications produced for that project.

For each message listed in this section, a table is provided which describes the properties of the message, aliases for the property, and keyword arguments accetped to assign value to the property during creation. Additionally, unless otherwise indicated, the behaviors and properties from *BaseMessage* will be inherited for each message class. For specific messages, additional notes may be provided, as well as descriptions of related classes and use cases, where applicable.

6.1 BaseMessage

Table 6.1: Properties for *BaseMessage* class

Property	Expected Datatype	Accepted Keyword Arguments
mission_timer missionTimer	tuple of integers; string as argument	mission_timer
elapsed_milliseconds	integer	elapsed_milliseconds

- The *BaseMessage* is a common abstract parent class from which all messages are derived.
- All message subclasses will also have these properties, however, they may be initialized to default values of (-1,-1) for *mission_timer* and -1 for *elapsed_milliseconds* if the properties are not relevant to the specific message class.

6.2 Agent Intervention Messages

6.2.1 AgentIntervention

Table 6.2: Properties for *AgentIntervention* class

Property	Expected Datatype	Accepted Keyword Arguments
id	string	id
agent	string	agent
created	string	created
start	tuple of ints	start
end	string	end
explanation	dictionary	explanation

6.2.2 AgentChatIntervention

Table 6.3: Properties for *AgentChatIntervention* class

Property	Expected Datatype	Accepted Keyword Arguments
content	string	content
receiver	string	receiver
type	string	type
renderer	string	renderer

6.3 Agent Prediction Messages

6.3.1 AgentPredictionBaseMessage

Table 6.4: Properties for *AgentPredictionBaseMessage* class

Property	Expected Datatype	Accepted Keyword Arguments
group	AgentPredictionGroupProperty	group
created	string	created

6.3.2 AgentActionPredictionMessage

Table 6.5: Properties for *AgentActionPredictionMessage* class

Property	Expected Datatype	Accepted Keyword Arguments
predictions	list of AgentActionPrediction	predictions

6.3.3 AgentStatePredictionMessage

Table 6.6: Properties for *AgentStatePredictionMessage* class

Property	Expected Datatype	Accepted Keyword Arguments
predictions	list of AgentStatePrediction	predictions

6.3.4 AgentActionPrediction

Table 6.7: Properties for *AgentActionPrediction* class

Property	Expected Datatype	Accepted Keyword Arguments
unique_id	string	unique_id
start	string	start
duration	string	duration
predicted_property	string	predicted_property
action	string	action
using	string	using
subject	string	subject
object	string	object
probability_type	string	probability_type
probability	string or float	probability
confidence_type	string	confidence_type
confidence	string or float	confidence
explanation	dictionary	explanation

6.3.5 AgentStatePrediction

Table 6.8: Properties for *AgentStatePrediction* class

Property	Expected Datatype	Accepted Keyword Arguments
unique_id	string	unique_id
start	string	start
duration	string	duration
subject_type	string	subject_type
subject	string	subject
predicted_property	string	predicted_property
prediction	string	prediction
probability_type	string	probability_type
probability	string or float	probability
confidence_type	string	confidence_type
confidence	string or float	confidence
explanation	dictionary	explanation

6.3.6 AgentPredictionGroupProperty

Table 6.9: Properties for *AgentPredictionGroupProperty* class

Property	Expected Datatype	Accepted Keyword Arguments
start	string	
duration	float	
explanation	dictionary	explanation

6.4 AgentVersionInfo

Table 6.10: Properties for *AgentVersionInfo* class

Property	Expected Datatype	Accepted Keyword Arguments
agent_name	string	agent_name
version	string	version
owner	string	owner
source	list of strings	source
dependencies	list of strings	dependencies
config	dictionary	config
publishes	list of tuples	publishes
subscribes	list of tuples	subscribes

- *source*, *dependencies*, *config*, *publishes*, and *subscribes* are optional keyword arguments

6.5 BeepEvent

Table 6.11: Properties for *BeepEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
sourceEntity source_entity	string	sourceEntity
message	string	message
location beep_x beep_y beep_z	tuple of floats float float float	location

- *beep_x*, *beep_y*, and *beep_z* correspond to specific elements of *location*, i.e., *location* = (*beep_x*, *beep_y*, *beep_z*).

6.6 BlockageList

Table 6.12: Properties for *BlockageList* class

Property	Expected Datatype	Accepted Keyword Arguments
mission	string	mission
blockages mission_blockage_list	list of *Blockage*	blockages (optional)

- *blockages* is an optional keyword argument; if not provided, *blockages* will be set to an empty list.
- Adding individual *Blockage* instances can be achieved with the *add(blockage)* method of the class.
- Once all blocks have been added, the *finalize()* method should be called to make the message immutable.

6.6.1 Blockage

6.7 ChatEvent

Table 6.13: Properties for *ChatEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
sender	string	sender
addressees	list of strings	addressees
text	string	text

6.8 CompetencyTaskEvent

Table 6.14: Properties for *CompetencyTaskEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
taskMessage task_message	string	taskMessage
playerName	string	playerNAme
callSign	string	callSign

6.9 DoorEvent

`door_event_message_properties` summarizes the properties for the *DoorEvent* class. Note that *door_x*, *door_y*, and *door_z* correspond to individual elements of *location* / *position*.

Table 6.15: Properties for *DoorEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
position location door_x door_y door_z	tuple of integers integer integer integer	position
opened open	boolean	opened

- *door_x*, *door_y*, and *door_z* correspond to individual elements of *location*, i.e., *location* = (*door_x*, *door_y*, *door_z*)

6.10 Experiment

Table 6.16: Properties for *Experiment* class

Property	Expected Datatype	Accepted Keyword Arguments
name	string	name
date	string	date
author	string	author
mission	string	mission

6.11 FoVSummary

Table 6.17: Properties for *FoVSummary* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
observationNumber	integer	observationNumber
blocks	list of dicts	blocks (optional)

- Providing the *blocks* keyword argument is optional; by default, the property is set to an empty list.
- Block summaries can be added to the *blocks* property through the *addBlock(summary)* method.

6.12 FoV_VersionInfo

Table 6.18: Properties for *FoV_VersionInfo* class

Property	Expected Datatype	Accepted Keyword Arguments
version	string	version
url	string	url
dependencies	list of FoV_Dependency	dependencies (optional)

- *dependencies* is an optional keyword argument; if not provided, the property will default to an empty list.
- Dependencies can be added with the *addDependency(dependency)* method.

6.12.1 FoV_Dependency

Table 6.19: Properties for *FoV_Dependency* class

Property	Expected Datatype	Accepted Keyword Arguments
package	string	package
version	string	version
url	string	url

6.13 FoV_MapMetadata

Table 6.20: Properties for *FoV_MapMetadata* class

Property	Expected Datatype	Accepted Keyword Arguments
map_name	string	map_name
world_name	string	world_name
map_url	string	map_url
world_url	string	world_url
creation_time	string	creation_time
lower_bound	tuple of floats	lower_bound
upper_bound	tuple of floats	upper_bound
ignored_blocks	list	ignored_blocks
parser_metadata	dictionary	parser_metadata

6.14 FreezeBlockList

Table 6.21: Properties for *FreezeBlockList* class

Property	Expected Datatype	Accepted Keyword Arguments
mission	string	mission
freezeblocks mission_freezeblock_list	list of Freezeblocks	freezeblocks (optional)

- *freezeblocks* is an optional argument; if not provided, the attribute will be set to an empty list.

6.14.1 FreezeBlock

Table 6.22: Properties for *FreezeBlock* class

Property	Expected Datatype	Accepted Keyword Arguments
location x y z	tuple of integers integer integer integer	location
block_type	string	block_type
room_name	string	room_name

6.15 GasLeakPlacedEvent

Table 6.23: Properties for *GasLeakPlacedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
location gasleak_x gasleak_y gasleak_z	tuple of integers integer integer integer	location

6.16 GasLeakRemovedEvent

Table 6.24: Properties for *GasLeakRemovedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
source	string	*source*
location gasleak_x gasleak_y gasleak_z	tuple of integers integer integer integer	location

6.17 ItemDropEvent

Table 6.25: Properties for *ItemDropEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
itemName itemname	string	itemName
location item_x item_y item_z	tuple of floats float float float	location

6.18 ItemEquippedEvent

Table 6.26: Properties for *ItemEquippedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
itemName equippeditemname	string	itemName

6.19 ItemPickupEvent

Table 6.27: Properties for *ItemPickupEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
itemName itemname	string	itemName
location item_x item_y item_z	tuple of floats float float float	location

6.20 LeverEvent

Table 6.28: Properties for *LeverEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
position location lever_x lever_y lever_z	tuple of integers integer integer integer	position
powered	boolean	powered

6.21 LocationEvent

Table 6.29: Properties for *LocationEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
participant_id	string	participant_id
playername	string	playername
callsign	string	callsign (optional)
corresponding_observation_number	integer	corresponding_observation_number
locations	list	locations (optional)
connections	list	connections (optional)
exited_locations	list	exited_locations (optional)
exited_connections	list	exited_connections (optional)

6.22 MarkerDestroyedEvent

Table 6.30: Properties for *MarkerDestroyedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
type marker_type	string	marker_type
location marker_x marker_y marker_z	tuple of integers integer integer integer	location

6.23 MarkerPlacedEvent

Table 6.31: Properties for *MarkerPlacedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername name	string	playername name
type marker_type	string	marker_type
location marker_x marker_y marker_z	tuple of integers integer integer integer	location

6.24 MarkerRemovedEvent

Table 6.32: Properties for *MarkerRemovedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername name	string	playername name
type marker_type	string	marker_type
location marker_x marker_y marker_z	tuple of integers integer integer integer	location

6.25 MissionStateEvent

Table 6.33: Properties for *MissionStateEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
misison	string	mission
state mission_state	enum: MissionState	state

6.26 PauseEvent

Table 6.34: Properties for *PauseEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
paused	boolean	paused

6.27 PlayerJumpedEvent

Table 6.35: Properties for *PlayerJumpedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
location	tuple of floats	location
item_x	float	
item_y	float	
item_z	float	

6.28 PlayerSprintingEvent

Table 6.36: Properties for *PlayerSprintingEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
sprinting	boolean	sprinting

6.29 PlayerState

Table 6.37: Properties for *PlayerState* class

Property	Expected Datatype	Accepted Keyword Arguments
playername name	string	playername, name
id entity_id	string	entity_id
entity_type	string	entity_type
observation_number	integer	observation_number
timestamp	string	timestamp
world_time	integer	world_time
total_time	integer	total_time
position x y z	tuple of floats float float float	position
orientation pitch yaw	tuple of floats float float	orientation
velocity motion_x motion_y motion_z	tuple of floats float float float	velocity
life	float	life

6.30 PlayerSwingingEvent

Table 6.38: Properties for *PlayerSwingingEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
swinging	boolean	swinging

6.31 RoleSelectedEvent

Table 6.39: Properties for *RoleSelectedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
new_role newRole	string	new_role
prev_role previousRole	string	prev_role

6.32 RubbleDestroyedEvent

Table 6.40: Properties for *RubbleDestroyedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername name	string	name, playername
location rubble_x rubble_y rubble_z	tuple of integers	location

6.33 RubblePlacedEvent

Table 6.41: Properties for *RubblePlacedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
from_location from_x from_y from_z	tuple of integers integer integer integer	from_location
to_location to_x to_y to_z	tuple of integers integer integer integer	to_location

6.34 ScoreboardEvent

Table 6.42: Properties for *ScoreboardEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
scoreboard	dictionary	scoreboard (optional)

6.35 StaticMapInitialized

Table 6.43: Properties for *StaticMapInitialized* class

Property	Expected Datatype	Accepted Keyword Arguments
semantic_map_name	string	semantic_map_name
semantic_map	dictionary	semantic_map

6.36 ThreatSignList

Table 6.44: Properties for *ThreatSignList* class

Property	Expected Datatype	Accepted Keyword Arguments
mission	string	mission
threat_signs mission_threatsign_list	string	threat_sign_list

6.36.1 ThreatSign

Table 6.45: Properties for *ThreatSign* class

Property	Expected Datatype	Accepted Keyword Arguments
location x y z	tuple of integers integer integer integer	location
block_type	string	block_type
room_name	string	room_name
feature_type	string	feature_type (optional)

6.37 ToolDepletedEvent

Table 6.46: Properties for *ToolDepletedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
tool_type	string	tool_type

6.38 ToolUsedEvent

Table 6.47: Properties for *ToolUsedEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
tool_type	string	tool_type
durability	integer	durability
count	integer	count
block_location target_block_x target_block_y target_block_z	tuple of integers integer integer integer	block_location
block_type target_block_type	string	block_type

6.39 TriageCount

Table 6.48: Properties for *TriageCount* class

Property	Expected Datatype	Accepted Keyword Arguments
player_name	string	player_name
triage_counts	dictionary	triage_counts

6.40 TriageEvent

Table 6.49: Properties for *TriageEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
player_name playername	string	player_name
victim_location victim_x victim_y victim_z	tuple of integers integer integer integer	victim_location
color type	string	color
victim_id	integer	

6.40.1 TriageState

TriageState is an enumeration of possible triage states:

- IN_PROGRESS
- UNSUCCESSFUL
- SUCCESSFUL

6.41 Trial

Table 6.50: Properties for *Trial* class

Property	Expected Datatype	Accepted Keyword Arguments
name	string	name
date	string	date
experimenter	string	experimenter
subjects	list of strings	subjects
trial_number	string	trial_number
group_number	string	group_number
study_number	string	study_number
condition	string	condition
notes	list of strings	notes
testbed_version	string	testbed_version
experiment_name	string	experiment_name
experiment_date	string	experiment_date
experiment_author	string	experiment_author
experiment_mission	string	experiment_mission
map_name	string	map_name
map_block_filename	string	map_block_filename
client_info	list of ClientInfo	client_info

6.41.1 ClientInfo

Table 6.51: Properties for *ClientInfo* class

Property	Expected Datatype	Accepted Keyword Arguments
playername	string	playername
callsign	string	callsign
participantid	string	participantid
staticmapversion	string	staticmapversion
markerblocklegend	string	markerblocklegend
uniqueid	string	uniqueid

6.42 VictimList

Table 6.52: Properties for *VictimList* class

Property	Expected Datatype	Accepted Keyword Arguments
mission	string	mission
victims mission_victim_list	list of Victim	victims (optional)

6.42.1 Victim

Table 6.53: Properties for *Victim* class

Property	Expected Datatype	Accepted Keyword Arguments
block_type	string	block_type
room_name	string	room_name
unique_id	integer	unique_id
location x y z	tuple of integers integer integer integer	location

6.43 VictimNoLongerSafe

Table 6.54: Properties for *VictimNoLongerSafe* class

Property	Expected Datatype	Accepted Keyword Arguments
type color		type color
location victim_x victim_y victim_z	tuple of integers integer integer integer	location

6.44 VictimPickedUp

Table 6.55: Properties for *VictimPickedUpVictimNoLongerSafe* class

Property	Expected Datatype	Accepted Keyword Arguments
playername name	string	playername
victim_id		victim_id
type color		type color
location victim_x victim_y victim_z	tuple of integers integer integer integer	location

6.45 VictimPlaced

Table 6.56: Properties for *VictimPlaced* class

Property	Expected Datatype	Accepted Keyword Arguments
playername name	string	playername
victim_id		victim_id
type color		type color
location victim_x victim_y victim_z	tuple of integers integer integer integer	location

6.46 VictimsExpired

Table 6.57: Properties for *VictimsExpired* class

Property	Expected Datatype	Accepted Keyword Arguments
message expired_message	string	message

6.47 VictimsRescued

Table 6.58: Properties for *VictimsRescued* class

Property	Expected Datatype	Accepted Keyword Arguments
message rescued_message	string	message

6.48 WoofEvent

Table 6.59: Properties for *WoofEvent* class

Property	Expected Datatype	Accepted Keyword Arguments
sourceEntity source_entity	string	sourceEntity
message	string	message
location woof_x woof_y woof_z	tuple of floats float float float	location

CHAPTER

SEVEN

BRIDGES

CHAPTER

EIGHT

MINECRAFT BRIDGE API

Package containing messages and bridges to message sources for interfacing with Minecraft:

```
    __  ____                                  ______  ____       _     __
   /  |/  (_)___  ___  _________________ _/ __/ /_/ __ )_____(_)___/ /___ ____
  / /|_/ / / __ \/ _ \/ ___/ ___/ __ `/ /_/ __/ __  / ___/ / __  / __ `/ _     / /  / / /
↪ / / /  __/ /__/ /  / /_/ / __/ /_/ /_/ / /  / / /_/ / /_/ /  __/
/_/  /_/_/_/ /_/\___/\___/_/   \__,_/_/  \__/_____/_/  /_/\__,_/\__, /\___/
                                                                /____/
```

The purpose of this module is to simplify interfacing to various sources of observations from Minecraft (e.g., Malmo, MQTT, File Readers, etc.) and provide a common representation of messages received from Minecraft, and interface to subscribe to specific observations and publish messages back to the Minecraft bridge.

Right now, though, things are just implemented for working with the MQTT bus, since that's the priority!

(Hoped for) Usage: The purpose of the bridge is to allow client code to easily change between message sources by simply changing the import statement. For instance:

```
from MinecraftBridge import messages
if BACKEND == "MQTT":
        import MinecraftBridge.mqtt as bridge
elif BACKEND == "Malmo":
        import MinecraftBridge.malmo as bridge
```

Whether or not this is how it turns out remains to be seen...

8.1 Messages

Definition of message classes used by MinecraftBridge. Classes in this module are independent of the *source* of the messages; source-specific information and methods (e.g., parsers) are located in specific bridge modules.

class MinecraftBridge.messages.**ASR_Alternative**(*text*, *confidence*)

A simple wrapper class for alternative transcriptions. This class does not define a message itself; rather, instances of ASR_Alternative are used by ASR_Messages to represent alternative transcriptions.

text [string] Text of the alternative transcription

confidence [float] Confidence in the transcription

property confidence

Get the confidence of the alternative transcription.

Attempting to set *confidence* raises an *ImmutableAttributeException*.

property text
Get the text of the alternative transcription.

Attempting to set *text* raises an *ImmutableAttributeException*.

toDict()
Generate a dictionary representation of the alternative transcription.

dict A dictionary representation of the ASR Alternative

class MinecraftBridge.messages.ASR_Message(**kwargs)
A class encapsulating ASR messages.

text [string] The transcription returned from the ASR system

alternatives [list [ASR_Alternative], default=[]] List of alternative transcriptions

is_final [boolean] Indicates whether the transcription is intermediate or final

asr_system [string] The system used to generate the transcription

id [string, optional] A UUID associated with the transcription, autogenerated if not provided

participant_id [string] The participant id the transcription was generated from

add(*alternative*)
Add an alternative transcription to the the ASR message

alternative [ASR_Alternative] Alternative transcription to add

property alternatives
Get the list of alternative transcriptions.

Attempting to set *alternatives* raises an *ImmutableAttributeException*.

property asr_system
Get the ASR system used to generate the transcription.

Attempting to set *asr_system* raises an *ImmutableAttributeException*.

finalize()
Indicate that all alternatives have been added to the ASR message

property id
UUID of the transcription

property is_final
Get whether the transciption is intermediate or final

Attempting to set *is_final* raises an *ImmutableAttributeException*.

property participant_id
Get the id of the participant that produced the transcription.

Attempting to set *participant_id* raises an *ImmutableAttributeException*.

property text
Get the text of the transcription.

Attempting to set *text* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ASR_Message.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ASR message.

`class MinecraftBridge.messages.AgentActionPrediction(**kwargs)`
A class encapsulating the contents of a single action prediction.

unique_id [string, optional] A unique identifier for the prediction, using UUID format. Will be generated if one is not provided.

start [string, default=None] The time the prediction becomes valid. If *None*, then the prediction is assumed to be effective immediately.

duration [string, default=None] The duration in seconds that the prediction remains valid. If *None*, then the prediction is valid for the trial run

action [string] The type of action being predicted

using [string] The tool that is used to perform the action

subject [string] The name of the entity taking action

object [string] The entity / block being acted upon

probability_type [string, default="float"] Data type of the probability ("string" or "float")

probability [string or float] Probability of the action occuring

confidence_type [string, default="float"] Data type of the confidence ("string" or "float")

confidence [float or string] Confidence of the prediction

explanation [dict, default={}] Explanation of the prediction

`property action`
Get the predicted action.

Attempting to set *action* raises an *ImmutableAttributeException*.

`property confidence`
Get the confidence value of the prediction.

Attempting to set *confidence* raises an *ImmutableAttributeException*.

`property confidence_type`
Get the confidence datatype (*string* or *float*).

Attempting to set *confidence_type* raises an *ImmutableAttributeException*.

`property duration`
Get the duration that the prediction is valid, in seconds.

Attempting to set *duration* raises an *ImmutableAttributeException*.

`property explanation`
Get the explanations for the prediction.

Attempting to set *explanation* raises an *ImmutableAttributeException*.

`property object`
Get the block / entity being acted upon.

Attempting to set *object* raises an *ImmutableAttributeException*.

property predicted
Get the predicted property.

Attempting to set *predicted_property* raises an *ImmutableAttributeException*.

property predicted_property
Get the predicted property.

Attempting to set *predicted_property* raises an *ImmutableAttributeException*.

property probability
Get the probability that the action is performed.

Attempting to set *probability* raises an *ImmutableAttributeException*.

property probability_type
Get the probability datatype (*string* or *float*).

Attempting to set *probability_type* raises an *ImmutableAttributeException*.

property start
Get the time the prediction becomes valid.

Attempting to set *start* raises an *ImmutableAttributeException*.

property subject
Get the name of the entity performing the action.

Attempting to set *subject* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the prediction.

dict A dictionary representation of the action prediction.

property unique_id
Get the UUID of the prediciton.

Attempting to set *unique_id* raises an *ImmutableAttributeException*.

property using
Get the tool used in the action.

Attempting to set *using* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**AgentActionPredictionMessage**(***kwargs*)
A class encapsulating agent action prediction messages.

In addition to the list of attributes below, the message also includes the attributes inherited from *AgentPredictionBaseMessage*.

predictions [list of AgentActionPrediction, default=[]] List of action predictions

add(*prediction*)
Add a prediction to the list of predictions.

prediction [AgentActionPrediction] Prediction to add to the list.

finalize()
Indicate that all predictions have been added to the message.

property predictions
Get the list of predictions.

Attempting to set *predictions* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the AgentActionPredicitonMessage.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the AgentActionPredictionMessage message.

class MinecraftBridge.messages.**AgentChatIntervention**(***kwargs*)
A class encapsulating Chat Intervention messages. Chat Intervention messages inherit attributes of AgentIntervention messages.

content [string] Message from the agent to display

receiver [string, default="broadcast"] Name of the participant id, or "broadcast" if the message should be sent to all participants

type [enum ["json", "string", "HTML", "block"], default="string"] Type of data to be displayed

renderer [string ["Minecraft_Chat", "Minecraft_Block", "Client_Map"], default="Minecraft_Chat"] Renderer used to display the content

property content
Get the content of the intervention.

Attempting to set *content* raises an *ImmutableAttributeException*.

property receiver
Get the receiver of the intervention.

Attempting to set *receiver* raises an *ImmutableAttributeException*.

property renderer
Get the renderer to display the content

Attempting to set *renderer* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the AgentChatIntervention message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ChatIntervention message.

toJson()
Generates a JSON representation of the AgentChatIntervention message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the AgentChatIntervention message.

property type
Get the type of the intervention.

Attempting to set *type* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**AgentFeedback**(***kwargs*)
A class encapsulating Agent Feedback messages.

participant_id [string] The id of the participant providing the feedback

feedback_type [string] The type of feedback requested

feedback_text [string] The content of the feedback (if text)

property feedback_text
Text of feedback provided.

Attemting to set *feedback_text* raises an *ImmutableAttributeException*.

property feedback_type
Type of feedback provided.

Attemting to set *feedback_type* raises an *ImmutableAttributeException*.

property participant_id
ID of the participant providing feedback.

Attemting to set *participant_id* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the AgentFeedback message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the AgentFeedback message.

toJson()
Generates a JSON representation of the AgentFeedback message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the AgentFeedback message.

class MinecraftBridge.messages.**AgentPredictionGroupProperty**(*start*, *duration*, *explanation={}*)
A simple class encapsulating group properties for predictions. Properties in group apply to all predictions in an AgentPredictionMessage, unless overridden by an individual prediction

start [string] The time the prediction is effective

duration [float] The length of time, in seconds, that the prediction will remain valid.

explanation [dict (optional, default={})] Agent custom dictionary describing why the prediction was generated.

property duration
Get the length of time the prediction remains valid.

Attempting to set *duration* raises an *ImmutableAttributeException*.

property explanation
Get the explanation of why the prediction was generated.

Attempting to set *explanation* raises an *ImmutableAttributeException*.

property start
Get the time the prediction is effective

Attempting to set *start* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the group properties.

dict A dictionary representation of the group properties.

class MinecraftBridge.messages.**AgentStatePrediction**(***kwargs*)
A class encapsulating the contents of a single state prediction.

unique_id [string, optional] A unique identifier for the prediction, using UUID format. Will be generated if one is not provided.

start [string, default=None] The time the prediction becomes valid. If *None*, then the prediction is assumed to be effective immediately.

duration [string, default=None] The duration in seconds that the prediction remains valid. If *None*, then the prediction is valid for the trial run

subject [string] The subject of the prediction (e.g., player, victim, team)

subject_type [string] The type of subject (individual or team) being predicted about

predicted_property [string] Name of the discrete property being predicted

prediction [string] Actual predicted value (e.g., score, player location, etc.)

probability_type [string, default="float"] Data type of the probability ("string" or "float")

probability [string or float] Value of the prediction as a string for float

confidence_type [string, default="float"] Data type of the confidence ("string" or "float")

confidence [float or string] Confidence of the prediction

explanation [dict, default={}] Explanation of the prediction

`property confidence`
Get the confidence value of the prediction.

Attempting to set *confidence* raises an *ImmutableAttributeException*.

`property confidence_type`
Get the confidence datatype (*string* or *float*).

Attempting to set *confidence_type* raises an *ImmutableAttributeException*.

`property duration`
Get the duration that the prediction is valid, in seconds.

Attempting to set *duration* raises an *ImmutableAttributeException*.

`property explanation`
Get the explanations for the prediction.

Attempting to set *explanation* raises an *ImmutableAttributeException*.

`property prediciton`
Get the value of the prediction.

Attempting to set *prediction* raises an *ImmutableAttributeException*.

`property predicted_property`
Get the predicted property.

Attempting to set *predicted_property* raises an *ImmutableAttributeException*.

`property prediction`
Get the value of the prediction.

Attempting to set *prediction* raises an *ImmutableAttributeException*.

`property probability`
Get the probability that the action is performed.

Attempting to set *probability* raises an *ImmutableAttributeException*.

`property probability_type`
Get the probability datatype (*string* or *float*).

Attempting to set *probability_type* raises an *ImmutableAttributeException*.

property start
Get the time the prediction becomes valid.

Attempting to set *start* raises an *ImmutableAttributeException*.

property subject
Get the name of the entity whose state is being predicted.

Attempting to set *subject* raises an *ImmutableAttributeException*.

property subject_type
Get the type of the entity (individual or team) that predictions are made about.

Attempting to set *subject_type* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the prediction.

dict A dictionary representation of the action prediction.

property unique_id
Get the UUID of the prediciton.

Attempting to set *unique_id* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**AgentStatePredictionMessage**(***kwargs*)
A class encapsulating agent state prediction messages.

In addition to the list of attributes below, the message also includes the attributes inherited from *AgentPredictionBaseMessage*.

predictions [list of AgentStatePrediction, default=[]] List of state predictions

add(*prediction*)
Add a prediction to the list of predictions.

prediction [AgentStatePrediction] Prediction to add to the list.

finalize()
Indicate that all predictions have been added to the message.

property predictions
Get the list of predictions.

Attempting to set *predictions* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the AgentStatePredicitonMessage.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the AgentStatePredictionMessage message.

class MinecraftBridge.messages.**AgentVersionInfo**(***kwargs*)
A class encapsulating Agent Version Info messages.

agent_name [string] The name of the agent

agent_type [string] The type of the agent

version [string] The version number of the agent

owner [string] The name of the person / organization that supports the agent

source [list of strings] A list of URLs where the agent was obtained from

dependences [list of strings] A list of the dependent components of the agent

config [dictionary of configuration values] A list of configuration parameters

publishes [list of (topic, message_type, message_subtype) tuple] A list of messages the agent publishes

subscribes [list of (topic, message_type, message_subtype) tuple] A list of messages the agent subscribes to

`addConfig`(*name*, *value*)
Add a configuration value to the message

name [string] Name of the configuration parameter

value value of the configuration parameter (will be cast as a string)

`addDependency`(*url*)
Add a URL of a dependency of the agent

url [string] URL of the dependency

`addPublishInfo`(*topic*, *message_type*, *message_subtype*)
Add message info for a message type that the agent publishes.

topic [string] Topic the agent publishes to

message_type [string] message type of the published message

message_subtype [string] subtype of the published message

`addSource`(*url*)
Add a URL where the agent can be obtained

url [string] URL where the agent can be obtained

`addSubscribeInfo`(*topic*, *message_type*, *message_subtype*)
Add message info for a message type that the agent subscribes.

topic [string] Topic the agent publishes to

message_type [string] message type of the published message

message_subtype [string] subtype of the published message

`property agent_name`
Get the name of the agent.

Attempting to set *agent_name* raises an *ImmutableAttributeException*.

`property agent_type`
Get the type of the agent.

Attempting to set *agent_type* raises an *ImmutableAttributeException*.

`property config`
Get the dictionary of configuration values.

Attempting to set *config* raises an *ImmutableAttributeException*.

`property dependencies`
Get the list of dependencies of the agent.

Attempting to set *dependencies* raises an *ImmutableAttributeException*.

finalize()
Finalize the message instance, so no new scores can be added.

property owner
Get the person / organization that supports this agent.

Attempting to set *owner* raises an *ImmutableAttributeException*.

property publishes
Get the list of messages that the agent publishes.

Attempting to set *publishes* raises an *ImmutableAttributeException*.

property source
Get the list of URLs where the agent can be obtained.

Attempting to set *source* raises an *ImmutableAttributeException*.

property subscribes
Get the list of messages that the agent subscribes to.

Attempting to set *subscribes* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the AgentVersionInfo message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the AgentVersionInfo message.

toJson()
Generates a JSON representation of the AgentVersionInfo message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the AgentVersionInfo message.

property version
Get the version string of the agent.

Attempting to set *version* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**BeepEvent**(***kwargs*)
A class encapsulating Beep Event messages.

Constructing a BeepEvent message requires passing the following keyword arguments:

sourceEntity message location

While aliases exist for these attributes, they are currently not accepted as constructor parameters.

sourceEntity [string] The name of the entity triggering the beep event

source_entity [string] An alias for *sourceEntity*

message [string] The message emitted by the source entity

location [tuple of floats] The (x,y,z) of the location of the entity

beep_x [float] The x location of the entity, alias of location[0]

beep_y [float] The y location of the entity, alias of location[1]

beep_z [float] The z location of the entity, alias of location[2]

property beep_x
Get the x-value of the location of the entity when the beep message was generated (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property beep_y
Get the y-value of the location of the entity when the beep message was generated (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property beep_z
Get the z-value of the location of the entity when the beep message was generated (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

property location
Get the location of the entity when the beep message was generated. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

property message
Get the message generated by the beep device. Attempting to set the value of *message* will result in an *ImmutableAttributeException* being raised.

property sourceEntity
Get the name of the entity that produced the beep message. Attempting to set the value of the *sourceEntity* will result in an *ImmutableAttributeException* being raised.

property source_entity
Alias for *sourceEntity*.

toDict()
Generates a dictionary representation of the BeepEvent message. BeepEvent information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the BeepEvent.

toJson()
Generates a JSON representation of the BeepEvent message. BeepEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the BeepEvent message.

class MinecraftBridge.messages.Blockage(***kwargs*)
A class encapsulating individual blockages.

Constructing a Blockage message requires passing the following keyword arguments:

location

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

location [tuple of ints] Location of the blockage

x [int] x location of the blockage (alias of *location[0]*)

y [int] y location of the blockage (alias of *location[1]*)

z [int] z location of the blockage (alias of *location[2]*)

block_type [string] Block type of the blockage

room_name [string] Name of the room the blockage is in

feature_type [string] type of map feature that this blockage is associated with

property block_type
Get the block type of the blockage.

Attempting to set *block_type* raises an *ImmutableAttributeException*.

property feature_type
Get the type of the blockage.

Attempting to set *feature_type* raises an *ImmutableAttributeException*.

property location
Get the location of the blockage.

Attempting to set *location* raises an *ImmutableAttributeException*.

property room_name
Get the name of the room the blockage is in.

Attempting to set *room_name* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Blockage.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Blockage.

property x
Alias of *location[0]*

property y
Alias of *location[1]*

property z
Alias of *location[2]*

class MinecraftBridge.messages.**BlockageList**(***kwargs*)
A class encapsulating BlockageList messages.

mission [string] Name of the mission

blockages [list of Blockage] List of blockages

mission_blockage_list [list of Blockage] Alias of *blockages*

add(*blockage*)
Add a blockage to the list of blockages

blockage [Blockage] Instance of a Blockage to add

property blockages
Get the list of blockages.

Attempting to set *blockages* raises an *ImmutableAttributeException*.

finalize()
Indicate that all Blockage instances have been added to the list

property mission
Get the mission name.

Attempting to set *mission* raises an *ImmutableAttributeException*.

property mission_blockage_list
Alias of *blockages*

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the BlockageList.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the BlockageList message.

`class MinecraftBridge.messages.BusHeader(message_type, timestamp=None, version=0.5)`
A lightweight wrapper of a bus header. Bus headers contain information regarding transmission of messages through a communication channel:

timestamp - Timestamp of the message publication (ISO8601 format) message_type - Type of message version - Version of the message

BusHeaders are used by the MQTT bus with the ASIST Testbed. See the message specifications in the testbed for details.

TODO: Implement header validation

`toDict()`
Convert the BusHeader to a Python dictionary whose key / value pairs match those of the testbed message format

`toJson()`
Create a JSON representation of this header, by converting attributes to their corresponding string representations.

`class MinecraftBridge.messages.ChatEvent(**kwargs)`
A class encapsulating Chat Event messages.

sender [string] The name of the sender

addressees [tuple of strings] The list of addressee names

text [string] The text of the chat message

`property addressees`
Get the list of addressee names of the chat. Attempting to set the value of the *addressees* will result in an *ImmutableAttributeException* being raised.

`property sender`
Get the name of the sender of the chat. Attempting to set the value of the *sender* will result in an *ImmutableAttributeException* being raised.

`property text`
Get the text of the chat. Attempting to set the value of the *text* will result in an *ImmutableAttributeException* being raised.

`toDict()`
Generates a dictionary representation of the ChatEvent message. ChatEvent information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ChatEvent.

`toJson()`
Generates a JSON representation of the ChatEvent message. ChatEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the BeepEvent message.

`class MinecraftBridge.messages.ClientInfo(**kwargs)`
A class encapsulating ClientInfo messages.

ClientInfo message contents are all optional (as per the message specs), and are all strings. If no value is provided, a value of <*NOT_PROVIDED*> will be assigned.

playername [string, optional] Name of the player

callsign [string, optional] Callsign of the player

participantid [string, optional] Unique ID of the participant

staticmapversion [string, optional] Static map version provided the player

markerblocklegend [string, optional] The marker block legend version provided to the player

uniqueid [string, optional] Unique identifier of the player

`property callsign`
Get the callsign of the player.

Attempting to set *callsign* raises an *ImmutableAttributeException*.

`property markerblocklegend`
Get the version of the marker block legend provided to the player.

Attempting to set *markerblocklegend* raises an *ImmutableAttributeException*.

`property participantid`
Get the participant ID of the player.

Attempting to set *participantid* raises an *ImmutableAttributeException*.

`property playername`
Get the name of the player.

Attempting to set *playername* raises an *ImmutableAttributeException*.

`property staticmapversion`
Get the version of the static map provided to the player.

Attempting to set *staticmapversion* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ClientInfo.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ClientInfo message.

`property uniqueid`
Get the unique id of the player.

Attempting to set *uniqueid* raises an *ImmutableAttributeException*.

`class MinecraftBridge.messages.CompetencyTaskEvent(**kwargs)`
A class encapsulating Competency Task Event messages.

Constructing a CompetencyTaskEvent message requires passing the following keyword argument:

taskMessage

While an alias exist for this attribute, it is currently not accepted as a constructor parameter.

Study 2 includes *playerName* and *callSign* in the messages; these are not included in the testbed repository MessageSpecs, so are included as optional.

taskMessage [string] The message indicating the competency task status

task_message [string] Alias of *taskMessage*

playerName [string, optional, default=None] Name of the player performing the competency task

callSign [string, optional, default=None] Callsign of the player performing the competency task

`property callSign`
Get the call sign of the player.

Attempting to set *callSign* raises an *ImmutableAttributeException*.

`property playerName`
Get the name of the player.

Attempting to set *playerName* raises an *ImmutableAttributeException*.

`property taskMessage`
Get the task message. Attempting to set the value of *taskMessage* will result in an *ImmutableAttributeException* being raised.

`property task_message`
Alias for 'taskMessage'.

`toDict()`
Generates a dictionary representation of the CompetencyTaskEvent message. Competency Task information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Competency Task Event.

`toJson()`
Generates a JSON representation of the CompetencyTaskEvent message. Comtetency Task information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Competency Task Event message.

`class MinecraftBridge.messages.DoorEvent(**kwargs)`
A class encapsulating Door Event messages.

participant_id: string Unique identifier of participant (e.g. "P000420")

playername [string] The name of the player that opened / closed the door. If not supplied, then used as an alias for participant_id

position [tuple of ints] Location of the door

location [tuple of ints] Location of the door; alias for *position*

opened [boolean] True if the door was opened, False otherwise

open [boolean] True if the door was opened, False if closed; alias for *opened*

door_x [int] The x location of the opened / closed door, alias of position[0]

door_y: int The y location of the opened / closed door, alias of position[1]

door_z: int The z location of the opened / closed door, alias of position[2]

`property door_x`
Get the x-value of the location of the door (i.e., *position[0]*). Attempting to set the x-value of the location will result in an

ImmutableAttributeException being raised.

property door_y
Get the y-value of the location of the door (i.e., *position[1]*). Attempting to set the y-value of the location will result in an

ImmutableAttributeException being raised.

property door_z
Get the z-value of the location of the door (i.e., *position[2]*). Attempting to set the z-value of the location will result in an

ImmutableAttributeException being raised.

property location
Alias for *position*.

property open
Alias for *opened*.

property opened
Get whether the door was opened or closed. Attempting to set the value of the *playername* will result in an *ImmutableAttributeException* being raised.

True if the door was opened, False if the door was closed

property participant_id
Get the unique identifier of the player that opened or closed the door. Attempting to set the value of the *participant_id* will result in an *ImmutableAttributeException* being raised.

property playername
Get the name of the player that opened or closed the door. Attempting to set the value of the *playername* will result in an *ImmutableAttributeException* being raised.

property position
Get the location of the door opened or closed the door. Attempting to set the value of the *position* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the DoorEvent message. DoorEvent information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the DoorEvent.

toJson()
Generates a JSON representation of the DoorEvent message. DoorEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the DoorEvent message.

class MinecraftBridge.messages.**Experiment**(***kwargs*)
A class encapsulating Experiment messages.

name [string] A user friendly name for the experiment

date [string] The date and time that the experiment was created

author [string] Name of the author of the experiment

mission [string] Name of the mission associated with the experiment

property author
Get the author of the experiment. Attempting to set the value of *author* will result in an *ImmutableAttributeException* being raised.

property date
Get the date the experiment was created. Attempting to set the value of *date* will result in an *ImmutableAttributeException* being raised.

property name
Get the name of the experiment. Attempting to set the value of *name* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the Experiment message. Experiment information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Experiment.

toJson()
Generates a JSON representation of the Experiment message. Experiment information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Experiment message.

class MinecraftBridge.messages.**FoVProfile**(***kwargs*)
A class encapsulating PyGLFoVAgent profile messages.

backend [string] GL backend used (GLFW, GLUT, etc)

vendor [string] Graphics card / library vendor

renderer [string] Graphics card model / library

sl_version [string] Shading language vendor

add_processing_statistics(*name*, *mean_time*, *std_time*, *min_time*, *max_time*, *count*)
Add statistics for a given compute acction

name [string] Name of the compute / processing action performed

mean_time [double] Average time to perform the action

std_time [double] Standard deviation of the time to perform the action

min_time [double] Minimum amount of the time action took to perform

max_time [double] Maximum amount of time the action took to perform

count [int] Number of times the process was called

property backend
GL Backend used

Attempting to set *backend* raises an *ImmutableAttributeException*.

property renderer
Attempting to set *renderer* raises an *ImmutableAttributeException*.

property sl_version
Attempting to set *sl_version* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the FoVProfile.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the FoVProfile message.

property vendor
Attempting to set *vendor* raises an *ImmutableAttributeException*.

property version
Attempting to set *version* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**FoVSummary**(***kwargs*)
A class encapsulating Field of View messages.

playername [string] The player whose FoV is being summarized

observationNumber [string] The observation number (from PlayerState) associated with the FoV summary

blocks [list of dictionaries] List of block summary information, consisting of a dictionary for each block summarized.

addBlock(*summary*)
Add a block to the list of blocks.

summary [dictionary] key-value mapping of the name and value of the block summary

property blocks
Get the list of block summaries

Attempting to set *blocks* raises an *ImmutableAttributeException*.

property observationNumber
Get the observation number (from the PlayerState message) associated with this FoV message.

Attempting to set *observationNumber* raises an *ImmutableAttributeException*.

property participant_id
Get the participant_id whose FoV is summarized.

Attempting to set *participant_id* raises an *ImmutableAttributeException*.

property playername
Get the name of the player whose FoV is summarized.

Attempting to set *playername* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the FoVSummary.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the FoVSummary message.

class MinecraftBridge.messages.**FoV_BlockLocationList**(***kwargs*)
A class encapsulating Field of View block location list messages.

playername [string] The player whose FoV is being summarized

observationNumber [string] The observation number (from PlayerState) associated with the FoV summary

locations [list of tuples] List of block locations (x,y,z).

property locations
Get the list of block locations

Attempting to set *locations* raises an *ImmutableAttributeException*.

property observation
Alias of *observationNumber*

property observationNumber
Get the observation number (from the PlayerState message) associated with this FoV message.

Attempting to set *observationNumber* raises an *ImmutableAttributeException*.

property playername
Get the name of the player whose FoV is summarized.

Attempting to set *playername* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the FoV_BlockLocationList message.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the FoV_BlockLocationList message.

class MinecraftBridge.messages.FoV_Dependency(**kwargs)
A class encoding a single depencendy of the FoV_VersionInfo

toDict()
Convert the version dependency to a Python dictionary

toJson()
Convert the version dependency to JSON format

class MinecraftBridge.messages.FoV_VersionInfo(**kwargs)
A class encapsulating Field of View version info messages.

Data in FoV version info messages consist of the following:

version - version string of the FoV agent url - url of the tag / release of the repo of the version of the agent dependencies - list of dependencys (packages) and their version information

addDependency(*dependency*)
Add a dependency to the list of dependencies.

Args: dependency - instance of a FoV_Dependency

toDict()
Convert the FoV to a Python dictionary

toJson()
Convert the FoV Message to a json message

class MinecraftBridge.messages.FreezeBlock(**kwargs)
A class encapsulating individual blockages.

Constructing a FreezeBlock message requires passing the following keyword arguments:

location

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

location [tuple of ints] Location of the freeze block

x [int] x location of the blockage (alias of *location[0]*)

y [int] y location of the blockage (alias of *location[1]*)

z [int] z location of the blockage (alias of *location[2]*)

block_type [string] Block type of the freeze block

room_name [string] Name of the room the freeze block is in

`property block_type`
Get the block type of the freeze block.

Attempting to set *block_type* raises an *ImmutableAttributeException*.

`property location`
Get the location of the freeze block.

Attempting to set *location* raises an *ImmutableAttributeException*.

`property room_name`
Get the name of the room the freeze block is in.

Attempting to set *room_name* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the FreezeBlock.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the FreezeBlock.

`property x`
Alias of *location[0]*

`property y`
Alias of *location[1]*

`property z`
Alias of *location[2]*

`class MinecraftBridge.messages.FreezeBlockList(**kwargs)`
A class encapsulating FreezeBlockList messages.

mission [string] Name of the mission

freezeblocks [list of FreezeBlock] List of blockages

mission_freezeblock_list [list of FreezeBlock] Alias of *freezeblocks*

`add(block)`
Add a freeze block to the list of blockages

block [FreezeBlock] Instance of a FreezeBlock to add

`finalize()`
Indicate that all FreezeBlock instances have been added to the list

property freezeblocks
Get the list of freezeblocks.

Attempting to set *freezeblocks* raises an *ImmutableAttributeException*.

property mission
Get the mission name.

Attempting to set *mission* raises an *ImmutableAttributeException*.

property mission_freezeblock_list
Alias of *freezeblocks*

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the FreezeBlockList.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the FreezeBlockList message.

class MinecraftBridge.messages.GasLeakPlacedEvent(**kwargs)
A class encapsulating Gas Leak Placed Event messages.

Constructing a GasLeakPlacedEvent message requires passing the following keyword argument:

location

While aliases exist for this attribute, they are currently not accepted as constructor parameters.

location [tuple of ints] The (x,y,z) location of the gas leak

gasleak_x [int] The x location of the gas leak, alias of location[0]

gasleak_y: int The y location of the gas leak, alias of location[1]

gasleak_z [int] The z location of the gas leak, alias of location[2]

property gasleak_x
Get the x-value of the location of the gas leak (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property gasleak_y
Get the y-value of the location of the gas leak (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property gasleak_z
Get the z-value of the location of the gasleak (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

property location
Get the location of the gas leak. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the GasLeakPlacedEvent message. GasLeakPlaced information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the GasLeakPlacedEvent.

toJson()
Generates a JSON representation of the GasLeakPlacedEvent message. GasLeakPlacedEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the GasLeakPlacedEvent message.

class MinecraftBridge.messages.**GasLeakRemovedEvent**(***kwargs*)
A class encapsulating Gas Leak Removed Event messages.

Constructing a GasLeakRemovedEvent message requires passing the following keyword arguments:

source location

While aliases exist for the *location* attribute, they are currently not accepted as constructor parameters.

source [string] The name of the entity (playername / "system") that removed the gasleak

location [tuple of ints] The (x,y,z) location of the gas leak

gasleak_x [int] The x location of the gas leak, alias of location[0]

gasleak_y [int] The y location of the gas leak, alias of location[1]

gasleak_z [int] The z location of the gas leak, alias of location[2]

property gasleak_x
Get the x-value of the location of the gas leak (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property gasleak_y
Get the y-value of the location of the gas leak (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property gasleak_z
Get the z-value of the location of the gasleak (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

property location
Get the location of the gas leak. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

property source
Get the name of the entity (playername or "system") that removed the gasleak. Attempting to set the value of *source* will result in an *ImmutableAttributeException* being raised.

toDict()
Convert the BeepEvent to a Python dictionary whose key / value pairs match those of the testbed message format

toJson()
Convert the BeepEvent to a JSON message

class MinecraftBridge.messages.**InterventionStatistics**(***kwargs*)
This class encapsualtes intervention statistics. Fundamentally, this message reports the number of issued, resolved and discarded interventions.

issued [int] Total number of issued interventions

resolved [int] Number of interventions that have been qued for resolution

discarded [int] Number of interventions that have been discarded and never needed to be resolved

property active
Get the total number of created interventions

Attempting to set *actuve* raises an *ImmutableAttributeException*.

property discarded
Get the total number of interventions that have been discarded before needing to be resolved

Attempting to set *discarded* raises an *ImmutableAttributeException*.

property resolved
Get the total number of interventions that were resolved

Attempting to set *resolved* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ItemPickupEvent.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ItemPickupEvent message.

class MinecraftBridge.messages.ItemDropEvent(**kwargs)
A class encapsulating ItemDropEvent messages.

Constructing an ItemDropEvent message requires passing the following keyword arguments:

itemName location

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

playername [string] Name of the player who dropped the item

itemName [string] Name of the item dropped

itemname [string] Alias of *itemName*

location [tuple of floats] Location where the item was dropped

item_x [float] x location of the dropped item

item_y [float] y location of the dropped item

item_z [float] z location of the dropped item

property itemName
Get the name of the item dropped.

Attempting to set *itemName* raises an *ImmutableAttributeException*.

property item_x
Get the location where the item was dropped.

Attempting to set *location* raises an *ImmutableAttributeException*.

property item_y
Get the location where the item was dropped.

Attempting to set *location* raises an *ImmutableAttributeException*.

property item_z
x location of the dropped item (i.e., alias of *location[2]*)

property itemname
Alias of *itemName*

property location
Get the location where the item was dropped.

Attempting to set *location* raises an *ImmutableAttributeException*.

property playername
Get the name of the player who dropped the item.

Attempting to set *playername* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ItemDroppedEvent.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ItemDroppedEvent message.

class MinecraftBridge.messages.**ItemEquippedEvent**(***kwargs*)
A class encapsulating ItemDropEvent messages.

playername [string] Name of the player who dropped the item

itemName [string] Name of the item dropped

equippeditemname [string] Alias of *itemName*

property equippeditemname
Alias of *itemName*

property itemName
Get the name of the item equipped.

Attempting to set *itemName* raises an *ImmutableAttributeException*.

property playername
Get the name of the player who equipped the item.

Attempting to set *playername* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ItemEquippedEvent.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ItemEquippedEvent message.

class MinecraftBridge.messages.**ItemUsedEvent**(***kwargs*)
A class encapsulating ItemUsedEvent messages.

Constructing an ItemUsedEvent message requires passing the following keyword arguments:

itemName location

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

playername [string] Name of the player who dropped the item

itemName [string] Name of the item dropped

itemname [string] Alias of *itemName*

location [tuple of floats] Location where the item was dropped

item_x [float] x location of the dropped item

item_y [float] y location of the dropped item

item_z [float] z location of the dropped item

`property itemName`
Get the name of the item used.

Attempting to set *itemName* raises an *ImmutableAttributeException*.

`property item_x`
Get the location where the item was used.

Attempting to set *location* raises an *ImmutableAttributeException*.

`property item_y`
Get the location where the item was used.

Attempting to set *location* raises an *ImmutableAttributeException*.

`property item_z`
x location of the used item (i.e., alias of *location[2]*)

`property itemname`
Alias of *itemName*

`property location`
Get the location where the item was used.

Attempting to set *location* raises an *ImmutableAttributeException*.

`property playername`
Get the name of the player who used the item.

Attempting to set *playername* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ItemUsedEvent.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ItemUsedEvent message.

`class MinecraftBridge.messages.LeverEvent(**kwargs)`
A class encapsulating Lever Event messages.

Constructing a LeverEvent message requires passing the following keyword arguments:

playername position

While aliases exist for these attributes, they are currently not accepted as constructor parameters.

playername [string] The name of the player that operated the lever

position [tuple of ints] Location of the lever

location [tuple of ints] Location of the lever; alias for *position*

powered [boolean] True if the lever was turned on, False otherwise

lever_x [int] The x location of the lever, alias of position[0]

lever_y: int The y location of the lever, alias of position[1]

lever_z: int The z location of the lever, alias of position[2]

`property lever_x`
Get the x-value of the location of the lever (i.e., *position[0]*). Attempting to set the x-value of the location will result in an

ImmutableAttributeException being raised.

`property lever_y`
Get the y-value of the location of the lever (i.e., *position[1]*). Attempting to set the y-value of the location will result in an

ImmutableAttributeException being raised.

`property lever_z`
Get the z-value of the location of the lever (i.e., *position[2]*). Attempting to set the z-value of the location will result in an

ImmutableAttributeException being raised.

`property location`
Alias for *position*.

`property playername`
Get the name of the player that operated the lever. Attempting to set the value of the *playername* will result in an *ImmutableAttributeException* being raised.

`property position`
Get the location of the lever. Attempting to set the value of the *position* will result in an *ImmutableAttributeException* being raised.

`property powered`
Get whether the lever was turned on or off. Attempting to set the value of the *powered* will result in an *ImmutableAttributeException* being raised.

True if the lever was turned on, False if the lever was turned off

`toDict()`
Generates a dictionary representation of the LeverEvent message. LeverEvent information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the LeverEvent.

`toJson()`
Generates a JSON representation of the LeverEvent message. LeverEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the LeverEvent message.

`class MinecraftBridge.messages.LocationEvent(**kwargs)`
A class encapsulating LocationEvent messages.

Future iterations will need to objectify the sub-attributes

participant_id [string] ID of the participant whose location changed

playername [string (optional)] Name of the player

callsign [string (optional)] Callsign of the player

corresponding_observation_number [int] observation number from PlayerState message

locations [list] A list of locations the player is currently in

connections [list] A list of connections the player is currently in

exited_locations [list] A list of locations the player just left

exited_connections [list] A list of connections the player just left

`toDict()`
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the SemanticMapInitialized message.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the SemanticMapInitialized message.

`class MinecraftBridge.messages.MarkerBlockType(value)`
Enumeration of the marker block types used in the ASIST mod of Minecraft.

`class MinecraftBridge.messages.MarkerDestroyedEvent(**kwargs)`
A class encapsulating Marker Destroyed Event messages.

Constructing a MarkerDestroyedEvent message requires passing the following keyword arguments:

location

While aliases exist for this attribute, they are currently not accepted as constructor parameters.

type [string] The type of marker placed

marker_type [string] Alias of *type*

location [tuple of ints] The (x,y,z) location of the marker

marker_x [int] The x location of the marker, alias of location[0]

marker_y: int The y location of the marker, alias of location[1]

marker_z [int] The z location of the marker, alias of location[2]

`property location`
Get the location of the marker. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

`property marker_type`
Get the type of marker destroyed. Attempting to set the value of *marker_type* will result in an *ImmutableAttributeException* being raised.

`property marker_x`
Get the x-value of the location of the marker (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property marker_y
Get the y-value of the location of the marker (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property marker_z
Get the z-value of the location of the marker (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the MarkerDestroyedEvent message. MarkerDestroyed information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the MarkerDestroyedEvent.

toJson()
Generates a JSON representation of the MarkerDestroyedEvent message. MarkerDestroyedEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the MarkerDestroyededEvent message.

property type
Alias of *marker_type*

class MinecraftBridge.messages.**MarkerPlacedEvent**(***kwargs*)
A class encapsulating Marker Placed Event messages.

playername [string] The name of the player that placed the marker

name [string] Alias of *playername*

type [string] The type of marker placed

marker_type [string] Alias of *type*

location [tuple of ints] The (x,y,z) location of the marker

marker_x [int] The x location of the marker, alias of location[0]

marker_y: int The y location of the marker, alias of location[1]

marker_z [int] The z location of the marker, alias of location[2]

property location
Get the location of the marker. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

property marker_type
Get the type of marker placed. Attempting to set the value of *marker_type* will result in an *ImmutableAttributeException* being raised.

property marker_x
Get the x-value of the location of the marker (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property marker_y
Get the y-value of the location of the marker (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property marker_z
Get the z-value of the location of the marker (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

property name
Alias for *playername*

property participant_id
Get the participant_id of the player who placed the marker.

Attempting to set *participant_id* raises an *ImmutableAttributeException*.

property playername
Get the name of the player who placed the marker. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the MarkerPlacedEvent message. MarkerPlaced information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the MarkerPlacedEvent.

toJson()
Generates a JSON representation of the MarkerPlacedEvent message. MarkerPlacedEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the MarkerPlacedEvent message.

property type
Alias of *marker_type*

class MinecraftBridge.messages.**MarkerRemovedEvent**(***kwargs*)
A class encapsulating Marker Removed Event messages. Marker Removed Events occur when a player removes an existing marker in the environment.

playername [string] The name of the player that placed the marker

name [string] Alias of *playername*

type [string] The type of marker placed

marker_type [string] Alias of *type*

location [tuple of ints] The (x,y,z) location of the marker

marker_x [int] The x location of the marker, alias of location[0]

marker_y: int The y location of the marker, alias of location[1]

marker_z [int] The z location of the marker, alias of location[2]

property location
Get the location of the marker. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

property marker_type
Get the type of marker removed. Attempting to set the value of *marker_type* will result in an *ImmutableAttributeException* being raised.

property marker_x
Get the x-value of the location of the marker (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property marker_y
Get the y-value of the location of the marker (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property marker_z
Get the z-value of the location of the marker (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

property name
Alias for *playername*

property participant_id
Get the participant_id of the player who removed the marker. Attempting to set the value of *participant_id* will raise an *ImmutableAttributeException*.

property playername
Get the name of the player who removed the marker. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the MarkerRemovedEvent message. MarkerRemoved information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the MarkerRemovedEvent.

toJson()
Generates a JSON representation of the MarkerRemovedEvent message. MarkerRemovedEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the MarkerRemovedEvent message.

property type
Alias of *marker_type*

class MinecraftBridge.messages.**MessageHeader**(*sub_type*, *experiment_id*, *trial_id*, *source*, *timestamp=None*, *version=0.5*, *replay_id=None*)
A lightweight wrapper of a message header. Message headers contain common information about the generation of the message:

timestamp - Timestamp of the creation of the data (ISO8601) experiment_id - String identifier of an experiment trial_id - String identifier of an experiment trial replay_id (optional) - UUID of a replay of original trial, if replayed source - name of testbed component that generated the data sub_type - Subtype of the data version - Version of the subtype

See the message specifications in the testbed

TODO: Implement message validation

toDict()
Convert the MessageHeader to a Python dictionary whose key / value pairs match those of the testbed message format

toJson()
Create a JSON representation of this header, by converting attributes to their corresponding string representations.

class MinecraftBridge.messages.**MissionStateEvent**(***kwargs*)
A class encapsulating Mission State Event messages.

Constructing a MissionStateEvent message requires passing the following keyword argument:

state

While an alias exists for this attribute, it is currently not accepted as constructor parameters.

mission [string] The name of the mission, referencing the map name and variant

state [instance of MissionState] State of the mission ("Start / Stop")

mission_state [instance of MissionState] Alias of *state*

class MissionState(*value*)
Enumeration of possible mission states.

property mission
Get the name of the mission. Attempting to set the value of *mission* will result in an *ImmutableAttributeException* being raised.

property mission_state
Alias of *state*

property state
Get the state of the mission (START / STOP). Attempting to set the value of *state* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the MissionStateEvent.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the MissionStateEvent message.

class MinecraftBridge.messages.**PauseEvent**(***kwargs*)
A class encapsulating PauseEvent messages.

paused [boolean] Indicate if the game is paused (True) or unpaused (False)

property paused
Get whether the message is referring to a pause or unpause event. Attempting to set the value of the *paused* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the PauseEvent message. PauseEvent information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the PauseEvent.

toJson()
Generates a JSON representation of the PauseEvent message. PauseEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the PauseEvent message.

class MinecraftBridge.messages.**PerturbationEvent**(***kwargs*)
A class encapsulating ##### messages.

type [string] Name of perturbation type ["blackout", "rubble"]

state [string] state of the perturbation ["start", "stop"]

property state
State of the perturbation (start / stop)

Attemting to set *state* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the PerturbationEvent message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the PerturbationEvent message.

`toJson()`
Generates a JSON representation of the PerturbationEvent message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ##### message.

`property type`
Tpe of perturbation (blackout / rubble)

Attemting to set *type* raises an *ImmutableAttributeException*.

`class MinecraftBridge.messages.PlayerJumpedEvent(**kwargs)`
A class encapsulating PlayerJumpedEvent messages.

Constructing a PlayerJumpedEvent message requires passing the following keyword arguments:

location

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

PlayerJumpedEvent messages contain attributes *item_x*, *item_y*, and *item_z*. While the semantics of these are unusual, they are included to conform to the ASIST testbed MessageSpecs.

playername [string] Name of the player who jumped

location [tuple of floats] Location where the player jumped

item_x [float] x position of the location of the player (alias of *location[0]*)

item_y [float] y position of the location of the player (alias of *location[1]*)

item_z [float] z position of the location of the player (alias of *location[2]*)

`property item_x`
Alias of *location[0]*

`property item_y`
Alias of *location[1]*

`property item_z`
Alias of *location[2]*

`property location`
Get the location where the player jumped.

Attempting to set *location* raises an *ImmutableAttributeException*.

`property playername`
Get the name of the player who jumped.

Attempting to set *playername* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the PlayerJumpedEvent message.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the PlayerJumpedEvent message.

`class MinecraftBridge.messages.PlayerSprintingEvent(**kwargs)`
A class encapsulating PlayerSprintingEvent messages.

playername [string] Name of the player who jumped

sprinting [boolean] Indicate if the player is sprinting (True if sprinting, False if not)

`property playername`
Get the name of the player who sprinted.

Attempting to set *playername* raises an *ImmutableAttributeException*.

`property sprinting`
Get whether the player is sprinting (True) or not (False)

Attempting to set *sprinting* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the PlayerSprintingEvent message.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the PlayerSprintingEvent message.

`class MinecraftBridge.messages.PlayerState(**kwargs)`
A class encapsulating PlayerState messages.

participant_id: string Unique identifier of participant (e.g. "P000420")

playername [string] Name of the entity

name [string] Alias of *playername*

id [string] A UUID unique identifier for the entity

entity_id [string] Alias of *id*

entity_type [string] The type of the entity (e.g., "human")

observation_number [int] Sequence number of the observation, since beginning of trial run

timestamp [string] Timestamp when the data was captured, in ISO 8601 format

world_time [int] Current time in ticks

total_time [int] Total world time (in ticks) independent of server

position [tuple of floats] (x,y,z) position of the entity

x [float] x position of the entity (alias of *position[0]*)

y [float] y position of the entity (alias of *position[1]*)

z [float] z position of the entity (alias of *position[2]*)

orientation [tuple of floats] (pitch, yaw) of the entity

pitch [float] Pitch of the entity (alias of *orientation[0]*)

yaw [float] Yaw of the entity (alias of *orientation[1]*)

velocity [tuple of floats] (x,y,z) of the velocity of the entity

motion_x [float] x value of the entity's velocity (alias of *velocity[0]*)

motion_y [float] y value of the entity's velocity (alias of *velocity[1]*)

motion_z [float] z value of the entity's velocity (alias of *velocity[2]*)

life [float] Current life value of the entity

`property entity_id`
Get the unique id of the entity.

Attempting to set *entity_id* raises an *ImmutableAttributeException*.

`property entity_type`
Get the type of the entity.

Attempting to set *entity_type* raises an *ImmutableAttributeException*.

`property id`
Alias of *entity_id*.

`property life`
Get the life of the entity.

Attempting to set *life* raises an *ImmutableAttributeException*.

`property motion_x`
Alias of *velocity[0]*.

`property motion_y`
Alias of *velocity[1]*.

`property motion_z`
Alias of *velocity[2]*.

`property name`
Alias of *playername*.

`property observation_number`
Get the sequence number of the observation.

Attempting to set *observation_number* raises an *ImmutableAttributeException*.

`property orientation`
Get the orientation of the entity.

Attempting to set *orientation* raises an *ImmutableAttributeException*.

`property participant_id`
Get the unique identifier for the participant.

Attempting to set *participant_id* raises an *ImmutableAttributeException*.

`property pitch`
Alias of *orientation[0]*.

`property playername`
Get the name of the entity.

Attempting to set *playername* raises an *ImmutableAttributeException*.

property position
Get the position of the entity.

Attempting to set *position* raises an *ImmutableAttributeException*.

property timestamp
Get the timestamp of when the observation was collected.

Attempting to set *timestamp* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the PlayerState.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the PlayerState message.

property total_time
Get the total time passed, independent of the server properties.

Attempting to set *total_time* raises an *ImmutableAttributeException*.

property velocity
Get the velocity of the entity.

Attempting to set *velocity* raises an *ImmutableAttributeException*.

property world_time
Get the world time, in ticks, since the start of the trial.

Attempting to set *world_time* raises an *ImmutableAttributeException*.

property x
Alias of *position[0]*.

property y
Alias of *position[1]*.

property yaw
Alias of *orientation[1]*.

property z
Alias of *position[2]*.

class MinecraftBridge.messages.**PlayerSwingingEvent**(***kwargs*)
A class encapsulating PlayerSwingingEvent messages.

playername [string] Name of the player who jumped

swinging [boolean] Indicate if the player is swinging (True if swinging, False if not)

property playername
Get the name of the player who swinged.

Attempting to set *playername* raises an *ImmutableAttributeException*.

property swinging
Get whether the player is swinging (True) or stopped (False)

Attempting to set *sprinting* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the PlayerSwingingEvent message.

`toJson()`
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the PlayerSwingingEvent message.

`class MinecraftBridge.messages.PlayerUtility(**kwargs)`
A class encapsulating PlayerUtility messages.

participant_id [string] ID of the participant whose utility is being reported

utility [string] Utility of the participant

`property participant_id`
ID of the participant whose utility is being reported

Attemting to set *participant_id* raises an *ImmutableAttributeException*.

`toDict()`
Generates a dictionary representation of the PlayerUtility message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the PlayerUtility message.

`toJson()`
Generates a JSON representation of the PlayerUtility message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the PlayerUtility message.

`property utility`
Utility of the participant

Attemting to set *utility* raises an *ImmutableAttributeException*.

`class MinecraftBridge.messages.RoleSelectedEvent(**kwargs)`
A class encapsulating RoleSelectedEvent messages.

Constructing a RoleSelectedEvent message requires passing the following keyword arguments:

participant_id new_role prev_role

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

participant_id: string Unique identifier of participant (e.g. "P000420")

playername [string] Name of the player changing roles

new_role [string] New role the player is adopting

newRole [string] Alias of *new_role*

prev_role [string] Previous role of the player

previousRole [string] Alias of *prev_role*

`property newRole`
Alias of *new_role*

property new_role
Get the new role adoped by the player.

Attempting to set *new_role* raises an *ImmutableAttributeException*

property participant_id
Get the unique identifier for the participant changing role.

Attempting to set *participant_id* raises an *ImmutableAttributeException*.

property playername
Get the name of the player changing role. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

property prev_role
Get the previous role of the player.

Attempting to set *prev_role* raises an *ImmutableAttributeException*.

property previousRole
Alias of *prev_role*

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the RoleSelectedEvent message.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the RoleSelectedEvent message.

class MinecraftBridge.messages.RoleText(**kwargs)
A class encapsulating RoleText messages.

missionName [string] Name of the mission

transport_specialist_text [list of strings] Text provided to the Transport Specialist

medical_specialist_text [list of strings] Text provided to the Medical Specialist

engineering_specialist_text [list of strings] Text provided to the Engineering Specialist

property engineering_specialist_text
List of messages given to the engineering specialist

Attemting to set *engineering_specialist_text* raises an *ImmutableAttributeException*.

property medical_specialist_text
List of messages given to the medical specialist

Attemting to set *medical_specialist_text* raises an *ImmutableAttributeException*.

property missionName
Name of the mission

Attemting to set *missionName* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the RoleText message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the RoleText message.

toJson()
Generates a JSON representation of the RoleText message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the RoleText message.

property transport_specialist_text
List of messages given to the transport specialist

Attemting to set *transport_specialist_text* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.RollcallRequest(**kwargs)
A class encapsulating Rollcall Request messages.

rollcall_id [string] The ID of teh rollcall request

property rollcall_id
Get the ID of the rollcall request

Attempting to set *rollcall_id* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the Rollcall Request message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Rollcall Request message.

toJson()
Generates a JSON representation of the Rollcall Request message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Rollcall Request message.

class MinecraftBridge.messages.RollcallResponse(**kwargs)
A class encapsulating Rollcall Response messages.

rollcall_id [string] The ID of the rollcall request

version [string] The version of the responding component

status [enum] Status of the component

uptime [int] number of seconds that the component has been up

property agent_type
Get the type of agent generating this response

Attempting to set *agent_type* raises and *ImmutableAttributeException*.

property rollcall_id
Get the ID of the rollcall response

Attempting to set *rollcall_id* raises an *ImmutableAttributeException*.

property status
Get the status of the component generating the response

Attempting to set *status* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the Rollcall Request message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Rollcall Request message.

toJson()
Generates a JSON representation of the Rollcall Request message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Rollcall Request message.

property uptime
Get the number of seconds the component has been running

Attempting to set *uptime* raises an *ImmutableAttributeException*.

property version
Get the version of the component generating the response

Attempting to set *version* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**RubbleCollapse**(***kwargs*)
A class encapsulating RubbleCollapse messages.

participant_id [string] The id of the player being reported

playername [string] The name of the player

triggerLocation [tuple of integers] The location of the trigger block

triggerLocation_x [integer] Alias of *triggerLocation[0]*

triggerLocation_y [integer] Alias of *triggerLocation[1]*

triggerLocation_z [integer] Alias of *triggerLocation[2]*

fromBlock [tuple of integers] the corner defining a rectangle of rubble

fromBlock_x [integer] Alias of *fromBlock[0]*

fromBlock_y [integer] Alias of *fromBlock[1]*

fromBlock_z [integer] Alias of *fromBlock[2]*

toBlock [tuple of integers] other corner defining a rectangle of rubble

toBlock_x [integer] Alias of *toBlock[0]*

toBlock_y [integer] Alias of *toBlock[1]*

toBlock_z [integer] Alias of *toBlock[2]*

property fromBlock
Attempting to set *fromBlock* raises an *ImmutableAttributeException*

property fromBlock_x
Attempting to set *fromBlock_x* raises an *ImmutableAttributeException*

property fromBlock_y
Attempting to set *fromBlock_y* raises an *ImmutableAttributeException*

property fromBlock_z
Attempting to set *fromBlock_z* raises an *ImmutableAttributeException*

property participant_id
Attempting to set *participant_id* raises an *ImmutableAttributeException*

property playername
Attempting to set *playername* raises an *ImmutableAttributeException*

property toBlock
Attempting to set *toBlock* raises an *ImmutableAttributeException*

property toBlock_x
Attempting to set *toBlock_x* raises an *ImmutableAttributeException*

property toBlock_y
Attempting to set *toBlock_y* raises an *ImmutableAttributeException*

property toBlock_z
Attempting to set *toBlock_z* raises an *ImmutableAttributeException*

toDict()
Generates a dictionary representation of the RubbleCollapse message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the RubbleCollapse message.

toJson()
Generates a JSON representation of the RubbleCollapse message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ##### message.

property triggerLocation
Attempting to set *triggerLocation* raises an *ImmutableAttributeException*

property triggerLocation_x
Attempting to set *triggerLocation_x* raises an *ImmutableAttributeException*

property triggerLocation_y
Attempting to set *triggerLocation_y* raises an *ImmutableAttributeException*

property triggerLocation_z
Attempting to set *triggerLocation_z* raises an *ImmutableAttributeException*

class MinecraftBridge.messages.**RubbleDestroyedEvent**(***kwargs*)
A class encapsulating Rubble Destroyed Event messages.

Constructing a RubbleDestroyedEvent message requires passing the following keyword arguments:

location

While aliases exist for this attribute, they are currently not accepted as constructor parameters.

participant_id: string Unique identifier of participant that destroyed the rubble (e.g. "P000420")

playername [string] The name of the player that destroyed the rubble

name [string] Alias of *playername*

location [tuple of ints] The (x,y,z) location of the rubble

rubble_x [int] The x location of the rubble, alias of location[0]

rubble_y: int The y location of the rubble, alias of location[1]

rubble_z [int] The z location of the rubble, alias of location[2]

property location
Get the location of the rubble. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

property name
Alias for *playername*

property participant_id
Get the unique identifier of the participant who destroyed the rubble. Attempting to set the value of *participant_id* will result in an *ImmutableAttributeException* being raised.

property playername
Get the name of the player who destroyed the rubble. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

property rubble_x
Get the x-value of the location of the rubble (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property rubble_y
Get the y-value of the location of the rubble (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property rubble_z
Get the z-value of the location of the rubble (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the RubbleDestroyedEvent message. RubbleDestroyed information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the RubbleDestroyedEvent.

toJson()
Generates a JSON representation of the RubbleDestroyedEvent message. RubbleDestoryedEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the RubbleDestroyedEvent message.

class MinecraftBridge.messages.**RubblePlacedEvent**(***kwargs*)
A class encapsulating Rubble Placed Event messages.

from_location [tuple of ints] The original (x,y,z) location of the rubble

to_location [tuple of ints] The new (x,y,z) location of the rubble

from_x [int] The original x location of the rubble, alias of from_location[0]

from_y: int The original y location of the rubble, alias of from_location[1]

from_z [int] The original z location of the rubble, alias of from_location[2]

to_x [int] The new x location of the rubble, alias of to_location[0]

to_y: int The new y location of the rubble, alias of to_location[1]

to_z [int] The new z location of the rubble, alias of to_location[2]

property from_location
Get the original location of the rubble. Attempting to set the value of *from_location* will result in an *ImmutableAttributeException* being raised.

property from_x
Get the x-value of the original location of the rubble (i.e., *from_location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property from_y
Get the y-value of the original location of the rubble (i.e., *from_location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property from_z
Get the z-value of the original location of the rubble (i.e., *from_location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the RubblePlacedEvent message. RubblePlaced information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the RubblePlacedEvent.

toJson()
Generates a JSON representation of the RubblePlacedEvent message. RubblePlacedEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the RubblePlacedEvent message.

property to_location
Get the new location of the rubble. Attempting to set the value of *to_location* will result in an *ImmutableAttributeException* being raised.

property to_x
Get the x-value of the new location of the rubble (i.e., *to_location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property to_y
Get the y-value of the new location of the rubble (i.e., *to_location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property to_z
Get the z-value of the new location of the rubble (i.e., *to_location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

class MinecraftBridge.messages.**ScoreboardEvent**(***kwargs*)
A class encapsulating ScoreboardEvent messages.

Creating the ScoreboardEvent requires no parameters to be given. Rather, the *addScore* method should be used to add scores to the scoreboard, and once all scores are added, the *finalize* method should be used to make the scoreboard immutable.

Adding scoreboard events involves the following steps:

1. Creating a new instance of a ScoreboardEvent

```
>>> scoreboard = ScoreboardEvent()
```

2. Adding scores of individual players

```
>>> scoreboard.addScore("ArthurDent", 100)
>>> scoreboard.addScore("ZaphodBeeblebrox", 80)
>>> scoreboard.addScore("FordPrefect", 200)
```

3. Finalizing the message, to disallow any further adding of scores

```
>>> scoreboard.finalize()
```

scoreboard [dict of ints] Dictionary mapping playernames to current scores.

addScore(*name*, *score*)
Add a score to the scoreboard, overwritting the score if it exists

name [string] Name of the player to add the score to

score [int] score of the added player

finalize()
Finalize the message instance, so no new scores can be added.

property scoreboard
Get the scoreboard dictionary. While attempting to set the value of *scoreboard* will result in an *ImmutableAttributeException* being raised, it is still possible to modify the *contents* of the scoreboard dictionary (which should not be done).

toDict()
Generates a dictionary representation of the ScoreboardEvent message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ScoreboardEvent message.

toJson()
Generates a JSON representation of the ScoreboardEvent message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ScoreboardEvent message.

class MinecraftBridge.messages.SemanticMapInitialized(**kwargs)
A class encapsulating Semantic Map Initialization messages.

Future iterations will need to objectify the semantic_map

semantic_map_name [string] Filename of the semantic map

semantic_map [dict] Dictionary representation of the semantic map

property name
Get the filename of the semantic map.

Attempting to set *semantic_map_name* raises an *ImmutableAttributeException*.

property semantic_map
Get the dictionary defining the semantic map

Attempting to set *semantic_map* raises an *ImmutableAttributeException*.

property semantic_map_name
Get the filename of the semantic map.

Attempting to set *semantic_map_name* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the SemanticMapInitialized message.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the SemanticMapInitialized message.

class MinecraftBridge.messages.Status(**kwargs)
A class encapsulating status (heartbeat) messages.

state [Status.State] state of the agent

status [string, optional] message providing detail about the status of the component

active [boolean, optional] message indicating if the component is actively functioning

class State(*value*)
Enumeration of Status.State

property active
Get whether the status is active. Attempting to set *active* will raise an *ImmutableAttributeException*.

property state
Get the name of the state of the status. Attempting to set the value of *state* will raise an *ImmutableAttributeException*.

property status
Get the status message. Attempting to set *status* will raise an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the Status message. Status information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Status.

toJson()
Generates a JSON representation of the Status message. Status information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Status message.

class MinecraftBridge.messages.ThreatSign(***kwargs*)
A class encapsulating individual threat signs.

Constructing a ThreatSign message requires passing the following keyword arguments:

location

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

location [tuple of ints] Location of the threat sign

x [int] x location of the threat sign (alias of *location[0]*)

y [int] y location of the threat sign (alias of *location[1]*)

z [int] z location of the threat sign (alias of *location[2]*)

block_type [string] Block type of the threat sign (i.e., redstone)

room_name [string] Name of the room the threat sign is in

feature_type [string] Type of map feature that this threat sign is associated with

property block_type
Get the block type of the threat sign.

Attempting to set *block_type* raises an *ImmutableAttributeException*.

property feature_type
Get the type of the threat sign.

Attempting to set *feature_type* raises an *ImmutableAttributeException*.

property location
Get the location of the threat sign.

Attempting to set *location* raises an *ImmutableAttributeException*.

property room_name
Get the name of the room the threat sign is in.

Attempting to set *room_name* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ThreatSign.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Threat Sign.

property x
Alias of *location[0]*

property y
Alias of *location[1]*

property z
Alias of *location[2]*

class MinecraftBridge.messages.ThreatSignList(**kwargs)
A class encapsulating BlockageList messages.

mission [string] Name of the mission

threat_signs [list of ThreatSign] List of blockages

mission_threatsign_list [list of ThreatSign] Alias of *blockages*

add(*sign*)
Add a threat sign to the list of threat signs

sign [ThreatSign] Instance of a ThreatSign to add

finalize()
Indicate that all Threat Signs instances have been added to the list

property mission
Get the mission name.

Attempting to set *mission* raises an *ImmutableAttributeException*.

property mission_threatsign_list
Alias of *threat_signs*

property threat_signs
Get the list of threat_signs.

Attempting to set *threat_signs* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ThreatSignList.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ThreatSignList message.

class MinecraftBridge.messages.ToolDepletedEvent(**kwargs)
A class encapsulating ToolDepletedEvent messages.

playername [string] The name of the player whose tool was depleted

tool_type [string] The type of tool depleted

property playername
Get the name of the player using the tool. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ToolDepletedEvent message.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ToolDepletedEvent message.

property tool_type
Get the type of tool used by the player. Attempting to set the value of *tool_type* will result in an *ImmutableAttributeException* being raised.

class MinecraftBridge.messages.**ToolUsedEvent**(***kwargs*)
A class encapsulating ToolUsedEvent messages.

playername [string] The name of the entity that used the tool

tool_type [string] The type of tool being used

durability [int] The number of uses left in the tool

count [int] The number of tools of this type in the player's inventory

block_location [tuple of ints] The location of the block being hit

target_block_x [int] The x location of the block being hit (i.e., alias of block_location[0])

target_block_y [int] The y location of the block being hit (i.e., alias of block_location[1])

target_block_z [int] The z location of the block being hit (i.e., alias of block_location[2])

block_type [string] Block type of the block being hit

target_block_type [string] Alias of block_type

property block_location
Get the location of the block the tool was used on. Attempting to set the value of *block_location* will result in an *ImmutableAttributeException* being raised.

property block_type
Get the type of block the tool was used on. Attempting to set the value of *block_type* will result in an *ImmutableAttributeException* being raised.

property count
Get the number of tools of the given type in the player's inventory. Attempting to set the value of *count* will result in an *ImmutableAttributeException* being raised.

property durability
Get the remaining durability of the tool. Attempting to set the value of *durability* will result in an *ImmutableAttributeException* being raised.

property playername
Get the name of the player using the tool. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

property target_block_type
Alias of *block_type*

property target_block_x
Alias of *block_location[0]*

property target_block_y
Alias of *block_location[1]*

property target_block_z
Alias of *block_location[2]*

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the ToolUsedEvent message.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ToolUsedEvent message.

property tool_type
Get the type of tool used by the player. Attempting to set the value of *tool_type* will result in an *ImmutableAttributeException* being raised.

class MinecraftBridge.messages.TriageEvent(**kwargs)
A class encapsulating TriageEvent messages.

Constructing a Triage message requires passing the following keyword arguments:

participant_id color

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

participant_id: string Unique identifier of participant performing the triage (e.g. "P000420")

player_name [string] Name of the player performing the triage

playername [string] Alias of *player_name*

victim_location [tuple of ints] Location of the victim being triaged

victim_x [int] x location of the victim being triaged (i.e., alias of victim_location[0])

victim_y [int] y location of the victim being triaged (i.e., alias of victim_location[1])

victim_z [int] z location of the victim being triaged (i.e., alias of victim_location[2])

triage_state [string] State of the triage (e.g., IN_PROGRESS)

color [string] The color of the victim being triaged

type [string] The type of the victim being triaged (pseudo-alias of *color*)

victim_id [int] Unique identifier for the victim being triaged

class TriageState(*value*)
An enumeration of possible triage states

property color
Get the type / color of the victim being triaged.

Attempting to set *color* raises an *ImmutableAttributeException*.

property participant_id
Get the unique identifier for the participant who performed the triage.

Attempting to set *participant_id* raises an *ImmutableAttributeException*.

property player_name
Get the name of the player who performed the triage.

Attempting to set *player_name* raises an *ImmutableAttributeException*.

property playername
Alias of *playername*

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the TriageEvent.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the TriageEvent message.

property triage_state
Get the state of the triage attemp (e.g., IN_PROGRESS).

Attempting to set *triage_state* raises an *ImmutableAttributeException*.

property type
Alias of *color*.

property vicitm_x
Alias of *victim_location[0]*

property vicitm_y
Alias of *victim_location[1]*

property vicitm_z
Alias of *victim_location[2]*

property victim_id
Get the victim's unique ID.

Attempting to set *victim_id* raises an *ImmutableAttributeException*.

property victim_location
Get the location of the victim being triaged.

Attempting to set *victim_location* raises an *ImmutableAttributeException*.

property victim_x
Alias of *victim_location[0]*

property victim_y
Alias of *victim_location[1]*

property victim_z
Alias of *victim_location[2]*

class MinecraftBridge.messages.**Trial**(***kwargs*)
A class encapsulating Trial messages.

name [string] Human-readable name for the trial

date [string] Date and time when the trial was run

experimenter [string] Name of the experimenter performing the trial

subjects [list of strings] List of the names or ids of the subjects in the trial

trial_number [string] Sequentially numbered trial

group_number [string] Data organization identifier

study_number [string] Study identifier

condition [string] Experimental condition used for the trial

notes [list of strings] List of experimenter notes for the trial

testbed_version [string] Testbed version used for the trial

experiment_name [string] Human-readable name of the experiment

experiment_date [string] The date and time the experiment was created

experiment_author [string] The name of the author of the experiment

experiment_mission [string] The mission associated with the experiment

map_name [string] The name of the map used

map_block_filename [string] The map block filename used during the trial

client_info [List of ClientInfo] Client info parameters assigned by the epxerimenter

class **TrialState**(*value*)
Enumeration of possible trial states

property **client_info**
Get the list of client info parameters assigned by the experimenter.

Attempting to set *client_info* raises an *ImmutableAttributeException*.

property **condition**
Get the experimental condition used for the trial.

Attempting to set *condition* raises an *ImmutableAttributeException*.

property **date**
Get the date the trial was run

Attempting to set *date* raises an *ImmutableAttributeException*.

property **experiment_author**
Get the name of the author of the experiment.

Attempting to set *experiment_author* raises an *ImmutableAttributeException*.

property **experiment_date**
Get the date and time the experiment was created.

Attempting to set *experiment_date* raises an *ImmutableAttributeException*.

property **experiment_mission**
Get the mission associated with the experiment.

Attempting to set *experiment_mission* raises an *ImmutableAttributeException*.

property experiment_name
Get the human-readable name of the experiment.

Attempting to set *experiment_name* raises an *ImmutableAttributeException*.

property experimenter
Get the name of the experimenter performing the trial.

Attempting to set *experimenter* raises an *ImmutableAttributeException*.

property group_number
Get the group number data organization identifier.

Attempting to set *group_number* raises an *ImmutableAttributeException*.

property map_block_filename
Get the name of the map block file used during the trial.

Attempting to set *map_block_filename* raises an *ImmutableAttributeException*.

property map_name
Get the name of the map used for the mission.

Attempting to set *map_name* raises an *ImmutableAttributeException*.

property name
Get the human-readable name for the trial.

Attempting to set *name* raises an *ImmutableAttributeException*.

property notes
Get the list of notes for the trial.

Attempting to set *notes* raises an *ImmutableAttributeException*.

property study_number
Get the study identifier.

Attempting to set *study_number* raises an *ImmutableAttributeException*.

property subjects
Get the list of names or ids of the subjects of the trial.

Attempting to set *subjects* raises an *ImmutableAttributeException*.

property testbed_version
Get the version of the testbed used for the trial.

Attempting to set *testbed_version* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Trial.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Trial message.

property trial_number
Get the sequence number of the trial.

Attempting to set *trial_number* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**Victim**(***kwargs*)

A class encapsulating individual victims.

Constructing a Victim message requires passing the following keyword arguments:

location

While aliases exists for these attribute, they are currently not accepted as constructor parameters.

location [tuple of ints] Location of the victim

x [int] x location of the victim (alias of *location[0]*)

y [int] y location of the victim (alias of *location[1]*)

z [int] z location of the victim (alias of *location[2]*)

block_type [string] Block type of the victim

room_name [string] Name of the room the victim is in

unique_id [int] Unique identifier of the victim

property **block_type**

Get the block type of the victim.

Attempting to set *block_type* raises an *ImmutableAttributeException*.

property **location**

Get the location of the victim.

Attempting to set *location* raises an *ImmutableAttributeException*.

property **room_name**

Get the name of the room the victim is in.

Attempting to set *room_name* raises an *ImmutableAttributeException*.

toDict()

Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the Victim.

toJson()

Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the Victim.

property **unique_id**

Get the id of the victim.

Attempting to set *unique_id* raises an *ImmutableAttributeException*.

property **x**

Alias of *location[0]*

property **y**

Alias of *location[1]*

property **z**

Alias of *location[2]*

class MinecraftBridge.messages.**VictimEvacuated**(***kwargs*)

A class encapsulating VictimEvacuated messages.

participant_id [string] The id of the player being reported

playername [string] The name of the player

victim_location [tuple of integers] Location of the victim

victim_x [integer] Alias of *victim_location[0]*

victim_y [integer] Alias of *victim_location[1]*

victim_z [integer] Alias of *victim_location[2]*

type [string] Type of victim being evacuated

victim_id [unique id identifying the victim] Unique ID identifying the victim

success [boolean] whether or not the victim was evacuated to the area corresponding with its letter identifier (a,b,c)

`property participant_id`
Attempting to set *participant_id* raises an *ImmutableAttrubteException*

`property playername`
Attempting to set *playername* raises an *ImmutableAttrubteException*

`property success`
Attempting to set *success* raises an *ImmutableAttrubteException*

`toDict()`
Generates a dictionary representation of the VictimEvacuated message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the VictimEvacuated message.

`toJson()`
Generates a JSON representation of the VictimEvacuated message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the ##### message.

`property type`
Attempting to set *type* raises an *ImmutableAttrubteException*

`property victim_id`
Attempting to set *victim_id* raises an *ImmutableAttrubteException*

`property victim_location`
Attempting to set *victim_location* raises an *ImmutableAttrubteException*

`property victim_x`
Attempting to set *victim_x* raises an *ImmutableAttrubteException*

`property victim_y`
Attempting to set *victim_y* raises an *ImmutableAttrubteException*

`property victim_z`
Attempting to set *victim_z* raises an *ImmutableAttrubteException*

`class MinecraftBridge.messages.VictimList(**kwargs)`
A class encapsulating FreezeBlockList messages.

mission [string] Name of the mission

victims [list of Victim] List of victims

mission_victim_list [list of Victim] Alias of *victims*

add(*block*)
Add a victim to the list of victims

victim [Victim] Instance of a Victim to add

finalize()
Indicate that all Victim instances have been added to the list

property mission
Get the mission name.

Attempting to set *mission* raises an *ImmutableAttributeException*.

property mission_victim_list
Alias of *victims*

toDict()
Generates a dictionary representation of the message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the VictimList.

toJson()
Generates a JSON representation of the message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the VictimList message.

property victims
Get the list of victims.

Attempting to set *victims* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**VictimNoLongerSafe**(***kwargs*)
A class encapsulating VictimNoLongerSafe messages.

Constructing a VictimPickedUp message requires passing the following keyword arguments:

location

While aliases exist for these attributes, they are currently not accepted as constructor parameters.

location [tuple of ints] The (x,y,z) location of the victim

type [string] Type of victim being triaged [REGULAR, CRITICAL]

color [string] Color of the victim being triaged [GREEN, YELLOW]. This should be considered a (pseudo-)alias of *type*

victim_x [int] The x location of the victim, alias of from_location[0]

victim_y: int The y location of the victim, alias of from_location[1]

victim_z [int] The z location of the victim, alias of from_location[2]

property color
Alias of *type*

property location
Get the location of the victim. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the VictimNoLongerSafe message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the VictimNoLongerSafe message.

`toJson()`
Generates a JSON representation of the VictimNoLongerSafe message. VictimNoLongerSafe information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the VictimNoLongerSafe message.

`property type`
Get the type of victim. Attempting to set the value of *type* will result in an *ImmutableAttributeException* being raised.

`property victim_x`
Get the x-value of the location of the victim (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

`property victim_y`
Get the y-value of the location of the victim (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

`property victim_z`
Get the z-value of the location of the victim (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

`class MinecraftBridge.messages.VictimPickedUp(**kwargs)`
A class encapsulating VictimPickedUp messages.

Constructing a VictimPickedUp message requires passing the following keyword arguments:

participant_id location type

While aliases exist for these attributes, they are currently not accepted as constructor parameters.

location [tuple of ints] The (x,y,z) location of the victim

participant_id: string Unique identifier of participant who picked up the victim (e.g. "P000420")

playername [string] Name of the player who picked up the victim

name [string] Alias of *playername*

victim_id [int] Unique identifier of the victim

type [string] Type of victim being triaged [REGULAR, CRITICAL]

color [string] Color of the victim being triaged [GREEN, YELLOW]. This should be considered a (pseudo-)alias of *type*

victim_x [int] The x location of the victim, alias of from_location[0]

victim_y: int The y location of the victim, alias of from_location[1]

victim_z [int] The z location of the victim, alias of from_location[2]

`property color`
Alias of *type*

`property location`
Get the location of the victim. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

`property name`
Alias for *playername*

property participant_id
Get the unique identifier for the participant who picked up the victim. Attempting to set the value of *participant_id* will result in an *ImmutableAttributeException* being raised.

property playername
Get the name of the player who picked up the victim. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

toDict()
Generates a dictionary representation of the VictimPickedUp message. VictimPickedUp information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the VictimPickedUp message.

toJson()
Generates a JSON representation of the VictimPickedUp message. VictimPickedUp information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the VictimPickedUp message.

property type
Get the type of victim picked. Attempting to set the value of *type* will result in an *ImmutableAttributeException* being raised.

property victim_id
Get the id of victim picked up. Attempting to set the value of *victim_id* will result in an *ImmutableAttributeException* being raised.

property victim_x
Get the x-value of the location of the victim (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property victim_y
Get the y-value of the location of the victim (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property victim_z
Get the z-value of the location of the victim (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

class MinecraftBridge.messages.**VictimPlaced**(***kwargs*)
A class encapsulating VictimPlaced messages.

location [tuple of ints] The (x,y,z) location of the victim

participant_id: string Unique identifier of participant who placed the victim (e.g. "P000420")

playername [string] Name of the player who placed the victim

name [string] Alias of *playername*

victim_id [int] Unique identifier of the victim

type [string] Type of victim being triaged [REGULAR, CRITICAL]

color [string] Color of the victim being triaged [GREEN, YELLOW]. This should be considered a (pseudo-)alias of *type*

victim_x [int] The x location of the victim, alias of from_location[0]

victim_y: int The y location of the victim, alias of from_location[1]

victim_z [int] The z location of the victim, alias of from_location[2]

property color
Psuedo-alias of *type*; REGULAR is replaced with GREEN, CRITICAL is replaced with YELLOW.

property location
Get the location of the victim. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

property name
Alias for *playername*

property participant_id
Get the unique identifier for the participant who placed the victim. Attempting to set the value of *participant_id* will result in an *ImmutableAttributeException* being raised.

property playername
Get the name of the player who placed the victim. Attempting to set the value of *playername* will result in an *ImmutableAttributeException* being raised.

toJson()
Generates a JSON representation of the VictimPlaced message. VictimPlaced information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the VictimPlaced message.

property type
Get the type of victim placed. Attempting to set the value of *type* will result in an *ImmutableAttributeException* being raised.

property victim_id
Get the id of victim placed. Attempting to set the value of *victim_id* will result in an *ImmutableAttributeException* being raised.

property victim_x
Get the x-value of the location of the victim (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

property victim_y
Get the y-value of the location of the victim (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

property victim_z
Get the z-value of the location of the victim (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

class MinecraftBridge.messages.**VictimSignal**(***kwargs*)
A class encapsulating VictimSignal messages.

participant_id [string] The ID of the participant that triggered the signal

playername [string] The name of the player triggering the signal

message [string] The message emitted by the signal device

location [tuple of integers] The (x,y,z) location of the signal

x [integer] The x locaiton of the entity (alias of *location[0]*)

y [integer] The y location of the entity (alias of *location[1]*)

z [integer] The z location of the entity (alias of *location[2]*)

roomname [string] The room name associated with the signal

property location
Location (x,y,z) of where the signal was generated

Attempting to set *location* raises an *ImmutableAttributeException*.

property message
Message emitted by the signaling device

Attempting to set *message* raises an *ImmutableAttributeException*.

property participant_id
ID of the participant that generated the signal

Attempting to set *participant_id* raises an *ImmutableAttributeException*.

property playername
Name of the participant that generated the signal

Attempting to set *playername* raises an *ImmutableAttributeException*.

property roomname
Name of the room associated with the signal

Attempting to set *roomname* raises an *ImmutableAttributeException*.

toDict()
Generates a dictionary representation of the VictimSignal message. Message information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the VictimSignal message.

toJson()
Generates a JSON representation of the VictimSignal message. Message information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the VictimSignal message.

property x
Alias of *location[0]*

Attempting to set *x* raises an *ImmutableAttributeException*.

property y
Alias of *location[1]*

Attempting to set *y* raises an *ImmutableAttributeException*.

property z
Alias of *location[2]*

Attempting to set *z* raises an *ImmutableAttributeException*.

class MinecraftBridge.messages.**VictimsExpired**(***kwargs*)
A class encapsulating VictimExpired messages.

message [string] Message content indicating victims that have expired

expired_message [string] Alias of *message*

property expired_message
Alias of *message*

property message
Get the message indicating the victims expired. Attempting to set the value of *message* will result in an *ImmutableAttributeException* being raised.

toJson()
Generates a JSON representation of the VictimsExpired message. VictimsExpired information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the VictimsExpired message.

class MinecraftBridge.messages.**VictimsRescued**(***kwargs*)
A class encapsulating VictimsRescued messages.

message [string] Message content indicating victims that have been rescued

rescued_message [string] Alias of *message*

property expired_message
Alias of *message*

property message
Get the message indicating the victims were rescued. Attempting to set the value of *message* will result in an *ImmutableAttributeException* being raised.

property rescued_message
Alias of *message*

toJson()
Generates a JSON representation of the VictimsRescued message. VictimsRescued information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the VictimsRescued message.

class MinecraftBridge.messages.**WoofEvent**(***kwargs*)
A class encapsulating Woof Event messages.

Constructing a WoofEvent message requires passing the following keyword arguments:

sourceEntity message location

While aliases exist for these attributes, they are currently not accepted as constructor parameters.

sourceEntity [string] The name of the entity triggering the woof event

source_entity [string] An alias for *sourceEntity*

message [string] The message emitted by the source entity

location [tuple of floats] The (x,y,z) of the location of the entity

woof_x [float] The x location of the entity, alias of location[0]

woof_y [float] The y location of the entity, alias of location[1]

woof_z [float] The z location of the entity, alias of location[2]

property location
Get the location of the entity when the woof message was generated. Attempting to set the value of *location* will result in an *ImmutableAttributeException* being raised.

property message
Get the message generated by the entity. Attempting to set the value of *message* will result in an *ImmutableAttributeException* being raised.

property sourceEntity
Get the name of the entity that produced the woof message. Attempting to set the value of the *sourceEntity* will result in an *ImmutableAttributeException* being raised.

property source_entity
Alias for *sourceEntity*.

`toDict()`
Generates a dictionary representation of the WoofEvent message. WoofEvent information is contained in a dictionary under the key "data". Additional named headers may also be present.

dict A dictionary representation of the WoofEvent.

`toJson()`
Generates a JSON representation of the WoofEvent message. WoofEvent information is contained in a JSON object under the key "data". Additional named headers may also be present.

string A JSON string mapping header names to a JSON representation of the WoofEvent message.

`property woof_x`
Get the x-value of the location of the entity when the woof message was generated (i.e., *location[0]*). Attempting to set the x-value of the location will result in an *ImmutableAttributeException* being raised.

`property woof_y`
Get the y-value of the location of the entity when the woof message was generated (i.e., *location[1]*). Attempting to set the y-value of the location will result in an *ImmutableAttributeException* being raised.

`property woof_z`
Get the z-value of the location of the entity when the woof message was generated (i.e., *location[2]*). Attempting to set the z-value of the location will result in an *ImmutableAttributeException* being raised.

8.2 MQTT Bridge

The mqtt module contains classes to process messages received from Minecraft through an MQTT broker. This module contains two bridge classes:

- *Bridge*: a class for connecting to Minecraft through an MQTT broker
- *FileBridge*: a class for processing Minecraft messages stored in a file

Both classes can also send messages—*Bridge* by publishing messages to the MQTT broker, and *FileBridge* by writing messages to a separate file.

8.2.1 Usage

Typical bridge usage will involve the following steps:

1. Create an instance of a bridge, providing a name for the pridge, as well as additional class-specific parameters:

```
>>> bridge = mqtt.Bridge("MinecraftBridge", ...)
```

2. Create one or more client objects to receive messages:

```
>>> client1 = MinecraftClient(...)
>>> client2 = MinecraftClient(...)
```

3. Register the clients to receive specific messages types:

```
>>> bridge.register(client1, ScoreboardEvent)
>>> bridge.register(client1, PlayerState)
>>> bridge.register(client2, PlayerState)
```

4. Connect to the MQTT broker or input file. This will automatically start a callback loop, which will send message instances to registered clients:

```
>>> bridge.connect()
```

5. Messages can be constructed and sent to be MQTT broker or output file:

```
>>> message = ChatEvent(...)
>>> bridge.send(message)
```

6. Once complete, disconnect from the MQTT broker or input / output files:

```
>>> bridge.disconnect()
```

class MinecraftBridge.mqtt.**Bridge**(*name*, *host='127.0.0.1'*, *port=1883*, *keepalive=60*, *bind_address=''*, *include_name_in_string=False*, ***kwargs*)

A bridge class for handling connections to Minecraft through a MQTT client.

This bridge is used to connect to a running Minecraft instance through a MQTT broker. This class assumes that the Minecraft instance is using the ASISTMod, though this is technically not a _strict_ requirement. Messages are received in JSON format, and are parsed using parsers defined in the mqtt.parsers subpackage.

Typically bridge usage will involve the following steps:

1. Create an instance of the mqtt.Bridge:

```
>>> bridge = mqtt.Bridge("MinecraftBridge")
```

2. Create one or more client objects to receive messages:

```
>>> client1 = MinecraftClient(...)
>>> client2 = MinecraftClient(...)
```

3. Register the clients to receive specific messages types:

```
>>> bridge.register(client1, ScoreboardEvent)
>>> bridge.register(client1, PlayerState)
>>> bridge.register(client2, PlayerState)
```

4. Connect to the MQTT broker. This will automatically start a callback loop, which will send message instances to registered clients:

```
>>> bridge.connect("localhost", 1883)
```

5. Messages can be constructed and sent to be MQTT broker:

```
>>> message = ChatEvent(...)
>>> bridge.send(message)
```

6. Once complete, disconnect from the MQTT broker:

```
>>> bridge.disconnect()
```

name [string] name of the bridge instance, used by the MQTT Client to identify the connection

logger [logging.Logger] instance of the logger than log messages are sent to

isConnected [boolean] indication of whether the bridge is connected to the MQTT broker or not

observers [dictionary] dictionary mapping message classes to objects that wish to receive messages of the given class

register(observer, message_class) Register an observer to receive messages of a given class

subscribe(topic) Subscribe to received messages of a given topic from the MQTT broker

send(message, qos, retain) Send a message to the MQTT broker

start() Begin recieving messages from the MQTT broker

connect(host, port, start_loop) Connect to the MQTT broker

disconnect() Disconnect from the MQTT broker

`connect`(*start_loop=True*)
Connect to the MQTT broker.

Upon connecting, the background callback loop will be started if the *start_loop* parameter is set to *True*.

host [string] IP address of the MQTT broker

port [integer] Port number the MQTT broker is listening to

start_loop [boolean] Flag indicating if the background callback loop should be started

`disconnect`()
Disconnect from the MQTT broker.

`on_message`(*client*, *userdata*, *msg*)
Callback when bridge receives a message from the broker.

Upon receiving a message, attempts to parse the message, and sends the parsed message instance to observers that are registered to receive messages of the class.

client [] the client instance for this callback

userdata [] private user data set in the MQTT client instance

msg [MQTT message] the received MQTT message

`send`(*message*, *qos=0*, *retain=False*)
Send the message to the MQTT broker.

This method assumes the message is an instance of a message class defined in MinecraftBridge.messages. Converting to JSON and topic publication is performed by the message instance and corresponding parser.

message : MinecraftBridge.message object

`start`()
Start the background callback loop to receive messages from the MQTT broker.

`subscribe`(*topic*)
Subscribe to a topic published by the MQTT broker.

topic [string] Topic to subscribe to

`unsubscribe`(*topic*)
Unsubscribe from a topic published by the MQTT broker.

topic [string] Topic to unsubscribe from

`class MinecraftBridge.mqtt.DummyBridge`(*name=None*, *include_name_in_string=False*)
A dummy bridge class where all messages are manually sent via the bridge.

Typically bridge usage will involve the following steps:

1. Create an instance of the mqtt.DummyBridge:

```
>>> bridge = mqtt.DummyBridge()
```

2. Create one or more client objects to receive messages:

```
>>> client1 = MinecraftClient(...)
>>> client2 = MinecraftClient(...)
```

3. Register the clients to receive specific messages types:

```
>>> bridge.register(client1, ScoreboardEvent)
>>> bridge.register(client1, PlayerState)
>>> bridge.register(client2, PlayerState)
```

4. Messages can be constructed and sent to be MQTT broker:

```
>>> message = ChatEvent(...)
>>> bridge.send(message)
```

name [string] name of the bridge instance

logger [logging.Logger] instance of the logger than log messages are sent to

observers [dictionary] dictionary mapping message classes to objects that wish to receive messages of the given class

send(message) Send a message, which will be forwarded to all registered relevant observers

connect()
Connect to the bus source.

disconnect()
Disconnect from the bus source.

send(*message*)
Manually send the message that will be forwarded to relevant observers.

This method assumes the message is an instance of a message class defined in MinecraftBridge.messages. Converting to JSON and topic publication is performed by the message instance and corresponding parser.

message : MinecraftBridge.message object

start()
Start receiving messages from the bus source.

subscribe(*topic*)
Subscribe to a topic published by the bus source.

topic [string] Topic to subscribe to

unsubscribe(*topic*)
Unsubscribe to a topic published by the bus source.

topic [string] Topic to unsubscribe to

class MinecraftBridge.mqtt.**FileBridge**(*name*, *input_filename*, *output_filename*, *include_name_in_string=False*, *sort_by_time=True*, ***kwargs*)
A bridge class for handling connections to Minecraft through a file.

This bridge is used to connect to a running Minecraft instance through a file. This class assumes that messages are stored in JSON format. Messages are parsed using parsers defined in the mqtt.parsers subpackage.

Typically bridge usage will involve the following steps:

1. Create an instance of the mqtt.FileBridge:

```
>>> bridge = mqtt.FileBridge("MinecraftBridge", "input.metadata", "output.
↪metadata")
```

2. Create one or more client objects to receive messages:

```
>>> client1 = MinecraftClient(...)
>>> client2 = MinecraftClient(...)
```

3. Register the clients to receive specific messages types:

```
>>> bridge.register(client1, ScoreboardEvent)
>>> bridge.register(client1, PlayerState)
>>> bridge.register(client2, PlayerState)
```

4. Connect to the input and output files. This will automatically start a callback loop, which will send message instances to registered clients:

```
>>> bridge.connect()
```

5. Messages can be constructed and sent via the bridge:

```
>>> message = ChatEvent(...)
>>> bridge.send(message)
```

6. Once complete, disconnect:

```
>>> bridge.disconnect()
```

name [string] name of the bridge instance

logger [logging.Logger] instance of the logger than log messages are sent to

input_filename [string] path of the file to read messages in from

output_filename [string] path of the file to write messages to

input_file [File instance] file to read messages from

output_file [File instance] file to write messages to

subscriptions [set of strings] set of topics subscribed to by the bridge

isConnected [boolean] indication of whether the bridge is connected to its source or not

observers [dictionary] dictionary mapping message classes to objects that wish to receive messages of the given class

register(observer, message_class) Register an observer to receive messages of a given class

subscribe(topic) Subscribe to received messages of a given topic

send(message) Send a message

start() Begin recieving messages

connect(host, start_loop) Connect to the bridge source

disconnect() Disconnect from the bridge source

connect(*start_loop=True*)
Connect to the input and output files

Upon connecting, the background callback loop will be started if the *start_loop* parameter is set to *True*.

start_loop [boolean] Flag indicating if the background callback loop should be started

disconnect()
Disconnect from the input and output files.

send(*message*)
Send the message by writing it to the output file and forwarding to relevant observers.

This method assumes the message is an instance of a message class defined in MinecraftBridge.messages. Converting to JSON and topic publication is performed by the message instance and corresponding parser.

message : MinecraftBridge.message object

send_messages_from_json(*json_lines*)
Convert a list of json lines to message objects, and send them in order as if they were coming off the bus. Aligns FoV and player state messages by observation number.

json_lines [list] List of json lines sorted by @timestamp

start()
Read in the contents of the input file, converting JSON to dictionary format. After sorting by @timestamp,

subscribe(*topic*)
Subscribe to a topic in the input file's message data.

topic [string] Topic to subscribe to

unsubscribe(*topic*)
Unsubscribe from a topic in the input file's message data.

topic [string] Topic to unsubscribe from

8.3 MQTT Parsers

Package containing parsers for creating Message instances from JSON messages recieved from an MQTT broker.

This module also maintains a *ParserMap*, which maps the Message class to the corresponding parser. This ensures that a client does not need to be aware of the actual parser classes.

class MinecraftBridge.mqtt.parsers.**MessageSubtype**(*value*)
Enumeration of subtypes of messages. These enumerations reflect the message types available from the MQTT bus in the ASIST Testbed, see message specs for more details.

class MinecraftBridge.mqtt.parsers.**MessageType**(*value*)
Enumeration of main type of messages. These enumerations reflect the message types available from the MQTT bus in the ASIST Testbed, see message specs for more details.

class MinecraftBridge.mqtt.parsers.**ParserMap**(*agent_name*)
A ParserMap maintains a set of MessageParsers and means to access the parsers.

The ParserMap accepts an *agent_name* argument during initialization. This argument is used to populate templated topic fields, and therefore should be ammenable to message topic formats (e.g., using snake_case).

add(*parser*, *replace=True*)

parser [MessageParser] Parser to add to the map

replace [boolean, default=True] Indicate if the parser should replace a currently existing one in the map, if present

`getMessageClassByTopic`(*topic*)
Return the MessageClass associated with a topic, if it's int eh ParserMap.

topic [string] The topic to search by

Message Class referenced by topic, or None if not in map

`getTopicByTypeAndSubtype`(*msg_type*, *msg_subtype*)
Return the MessageClass associated with a type and subtype, if it's in the ParserMap.

msg_type [string] Message Type

msg_subtype [string] Message Subtype

CHAPTER

NINE

CHANGELOG

- 24 August 2021
  - Create unittests for BeepEvent, ChatEvent, and CompetencyTaskEvent messages
  - Added Sphinx Documentation

CHAPTER

TEN

INDICES AND TABLES

PYTHON MODULE INDEX

m

INDEX

A

B

C

D

E

F

G

I

L

M

N

O

P

R

S

T

U

V

W

X

Y

Z