ASIST Testbed Experiment/Trial Walkthrough

V3.3.6 – 25-JAN-2022

Contents

[1 Introduction 1](#_Toc94006903)

[2 Experimenter 2](#_Toc94006904)

[2.1 Creating an Experiment 2](#_Toc94006905)

[2.2 Getting players into the testbed 3](#_Toc94006906)

[2.3 Define the client information 5](#_Toc94006907)

[2.4 Observer 6](#_Toc94006908)

[2.5 Starting up an ASI agent 6](#_Toc94006909)

[2.6 Creating a Trial and Starting the Mission 6](#_Toc94006910)

[2.7 Starting a mission without an observer 10](#_Toc94006911)

[2.8 Checking the testbed components are running 11](#_Toc94006912)

[2.9 Taking a survey 12](#_Toc94006913)

[2.10 Connecting to the Running Mission with Minecraft 14](#_Toc94006914)

[2.11 Ending the Trial 14](#_Toc94006915)

[2.12 Exporting the Trial Data 15](#_Toc94006916)

[3 Developer/Admin 15](#_Toc94006917)

[3.1 Checking data that is being Captured 16](#_Toc94006918)

[3.2 Checking the Container Logs 16](#_Toc94006919)

[3.3 Message Validation 17](#_Toc94006920)

[4 Basic Setup Instructions 18](#_Toc94006921)

# Introduction

This document describes how to run an experiment/trial in the ASIST testbed. The walkthrough assumes that the testbed has already been successfully installed, setup and in currently running. For basic testbed setup instructions see Basic Setup Instructions

This document has several sections which are designed to focus on different user roles.

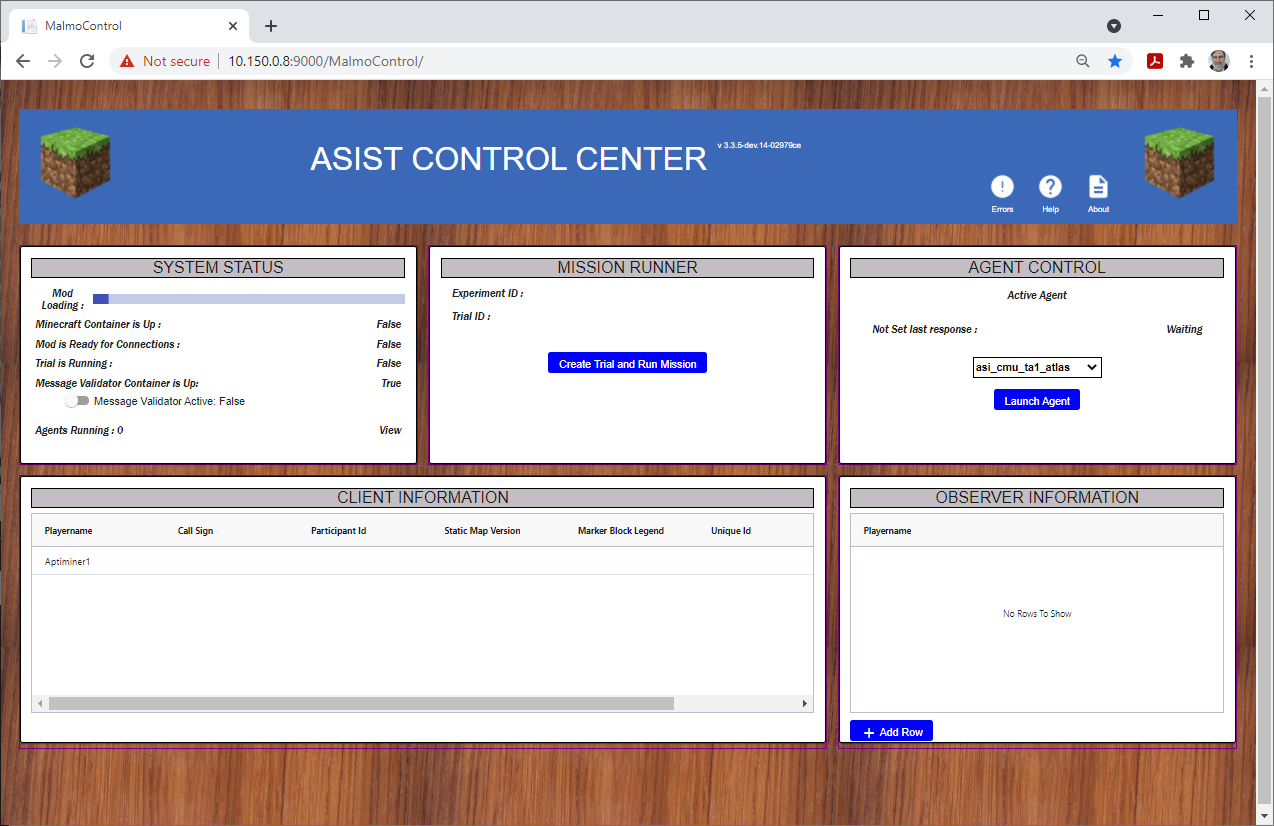
* Experimenter
* Developer/Admin

# Experimenter

The experimenter role creates experiments and runs trials. Each trial is one run of the Minecraft mission

## Creating an Experiment

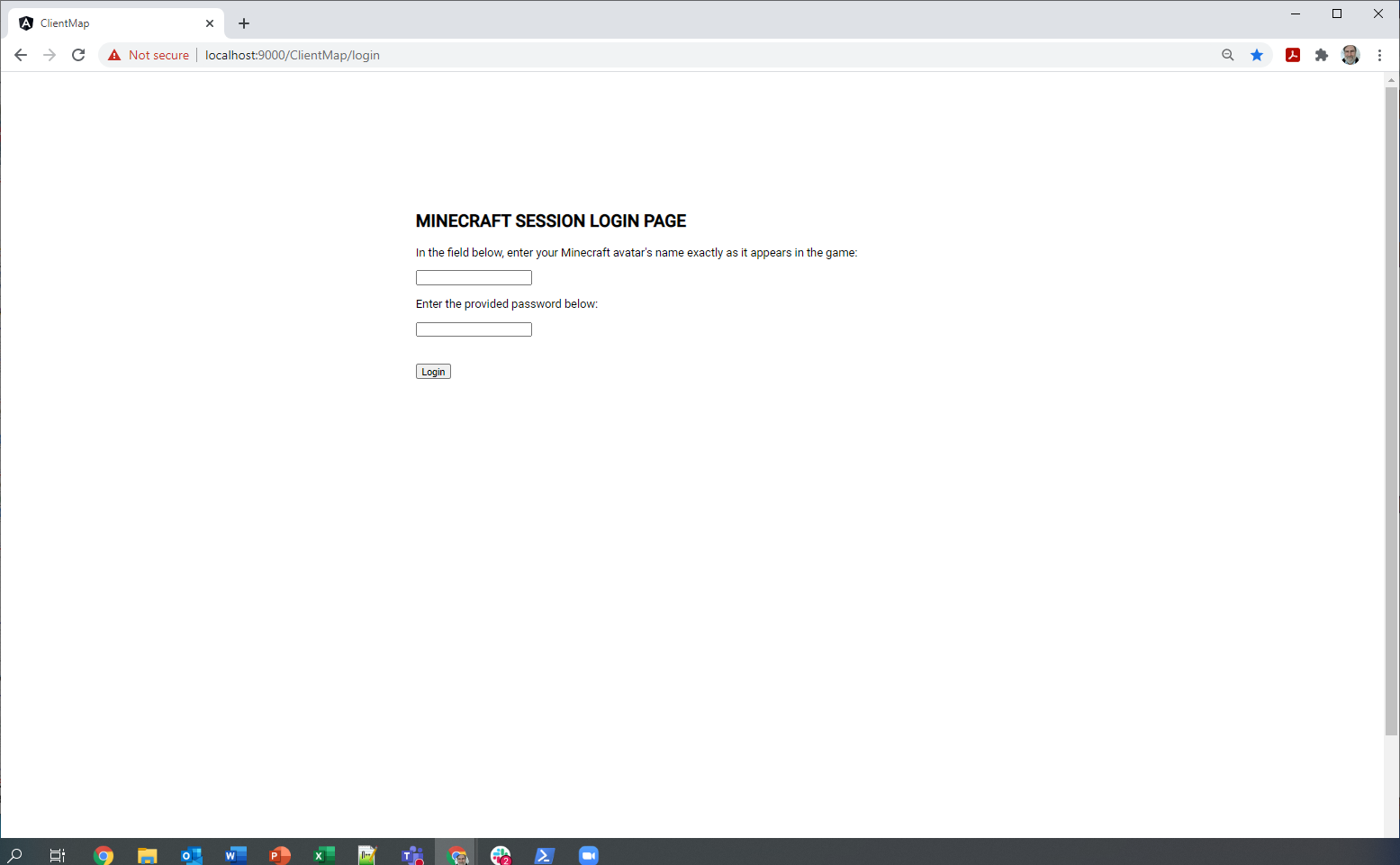
* In a browser (Chrome preferred), open up https://<testbed\_host>:9000/MalmoControl
* The ASIST Control Center looks like:



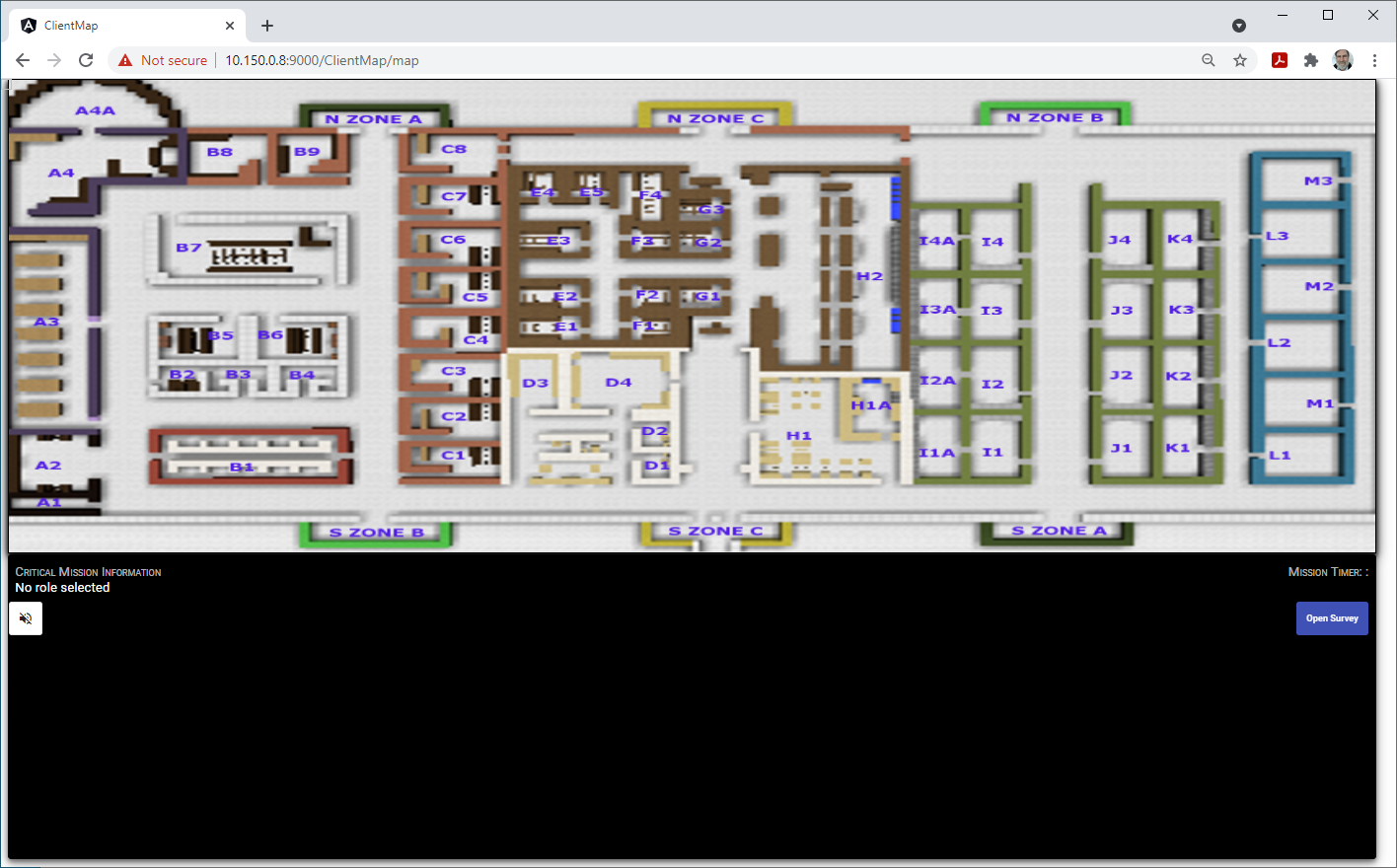
* + The version of the testbed is shown to the right of the main header.
  + The top left panel shows status of the testbed. These will change as the testbed is used. The top line shows the progress of the testbed as it is starting up a trial. The testbed may restart several times to get to a successful startup. The message validation service can be started by moving the slider to the right. The status of the agents can be displayed by clicking on the last one of this panel which will cause a pop-up panel to show agent status
  + The top center panel is used to run an experiment/trial. The assigned experiment and trial full universal unique ids are displayed when the trial is started
  + The top right panel controls the ASI agents that will be run. Select an agent from the pull down list and click the "Launch Agent" button. If the agent is running properly, the time line will update.
  + The bottom left/center panel shows client information. This should be filled in before the trial is started
  + The bottom right panel shows the observer in the trial

## Getting players into the testbed

There is a sequence of operation that needs to be followed for the various aspects of the testbed to collect the data it needs to start up a trial and configure all of the components of the testbed. The ASIST Control Center should be started up first (as above). Then each player should open the clientMap page at https://<testbed\_host>:9000/ClientMap

This page will show a login screen that looks like:  


* Each user should log in using their Minecraft player name, with the exact spelling and capitalization as their Minecraft username.
* Currently the password is always "admin". This may be changed in the future

After the player logs in the screen will look like:  


This page will be altered when the mission starts based on the parameters that the experimenter sets in the ASIST Control Center.

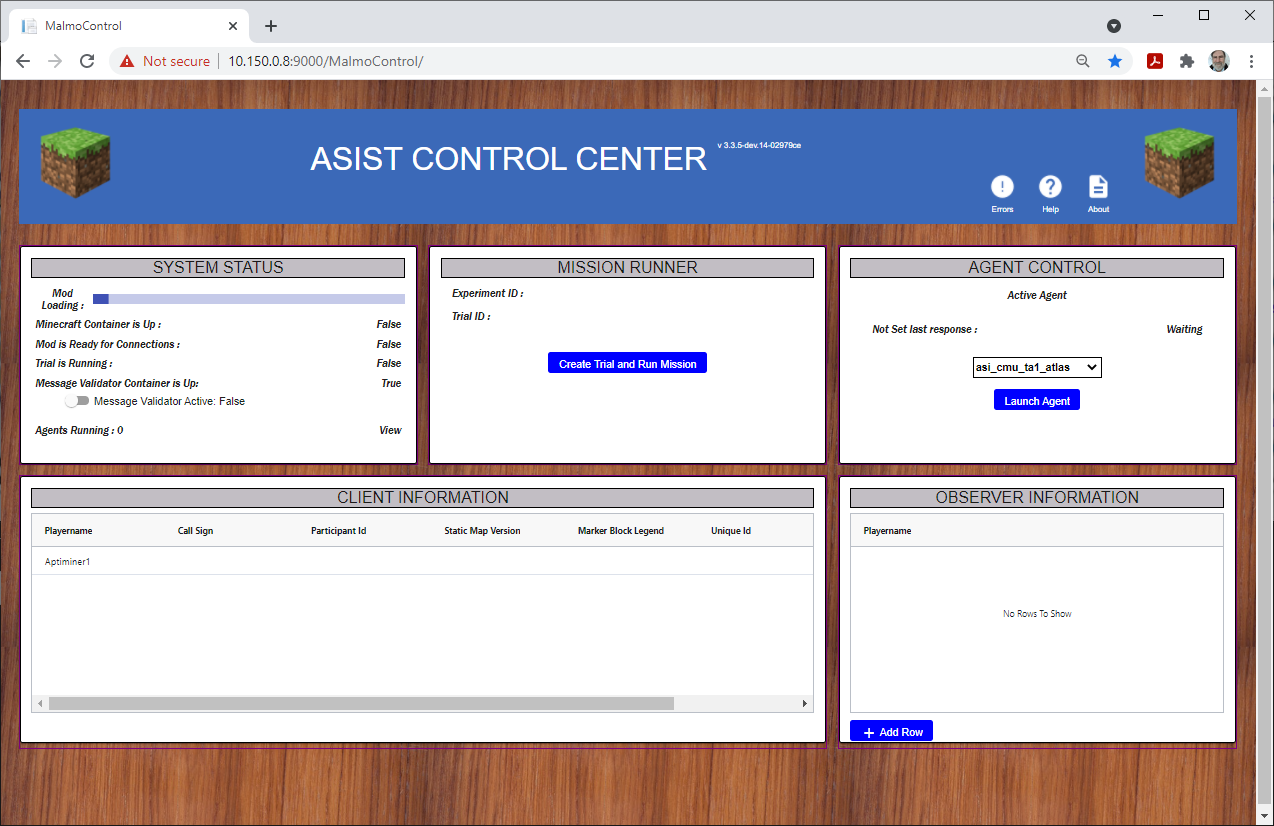
From this page, the player can:

* Turn on the microphone to collect audio data. The microphone should turn on automatically when a trial is started and turn off when the trial ends. Therefore, the player should not have to click the microphone button.
  + The testbed creates a .wav file for each player for each trial. If the player toggles the microphone button themselves, that will stop and start a new .wav file capture of that player's audio.
* Initiate the various surveys they are taken during the experiment.
  + Note that surveys are captured in the testbed along with the trial that is running. To associate a survey response with a trial, and therefore have the survey response exported with the trial data in a .metadata file, there should be a trial running when the player pushes the survey button and starts a survey. If a subsequent trial is run with the same participants (using the same unique ids) then the .metadata file will include all of the surveys taken up until that time for those participants with those unique ids. .

Players should not refresh the browser page once they are connected to the testbed.

The next step in the process is for the experimenter to specify various parameters of the trial.

## Define the client information

After all the players have logged into the clientMap page, each player will automatically be added to the Client Information table at the bottom of the ASIST Control Center page. After one player has logged in, the page will look like:  


The experimenter can now fill in the Call Sign, Participant ID and Unique ID for each player in the table. The Status Map Version and Marker Block Legend fields are not being used currently.

If you refresh the ASIST Control Center page, the client information table will be cleared and they will have to log back in again. If there are players that you don't want included in the mission, you can use the delete row button at the right end of the table row.

The information in the client information table will be published on the message bus and will be used to customize the trial as well as the client map page for each player. After the mission is started, the client map page will automatically be updated based on the values in the client information table.

Each trial can include an observer. This is a Minecraft player that starts the mission and can observe the activity of all of the players in the game from an overhead observation point. The activity data for the observer is not published on the message bus. The observer player is added to the trial by clicking the Add Row button at the bottom right of the page and typing in the exact Minecraft player name who will act as the observer.

NOTE: the testbed now checks to make sure when a player joins a Minecraft mission that they have logged into the client map web site first. If the player hasn't logged into the client map web page, they will not be allowed to enter the Minecraft mission. This restriction was added because participants from other sessions were connecting to the Minecraft missions when they weren't authorized to do so.

## Observer

The testbed allows an observer to enter the game and provide a view of the entire building and all of the activity in it. To set up a Minecraft player as the observer do the following before you start the mission:

1. Click the "+ Add Row" button at the lower right of the ASIST Control Center page.
2. Fill in the exact name of the Minecraft player who will be acting as the observer.
3. Start the trial

When the mission starts and the observer player enters the game, they will be automatically teleported to the location where they can start the mission and be correctly positioned to observe the entire build and movement of the other players.

## Starting up an ASI agent

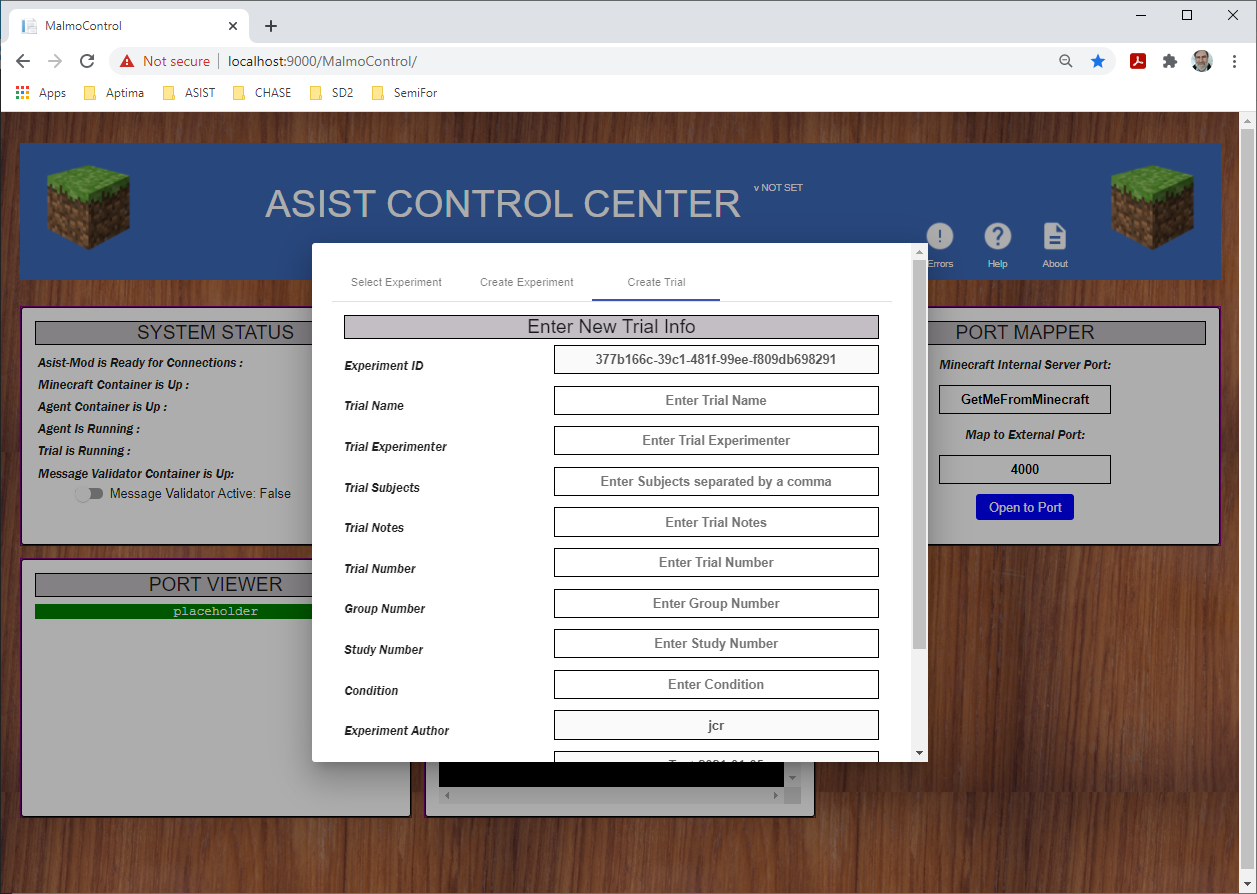
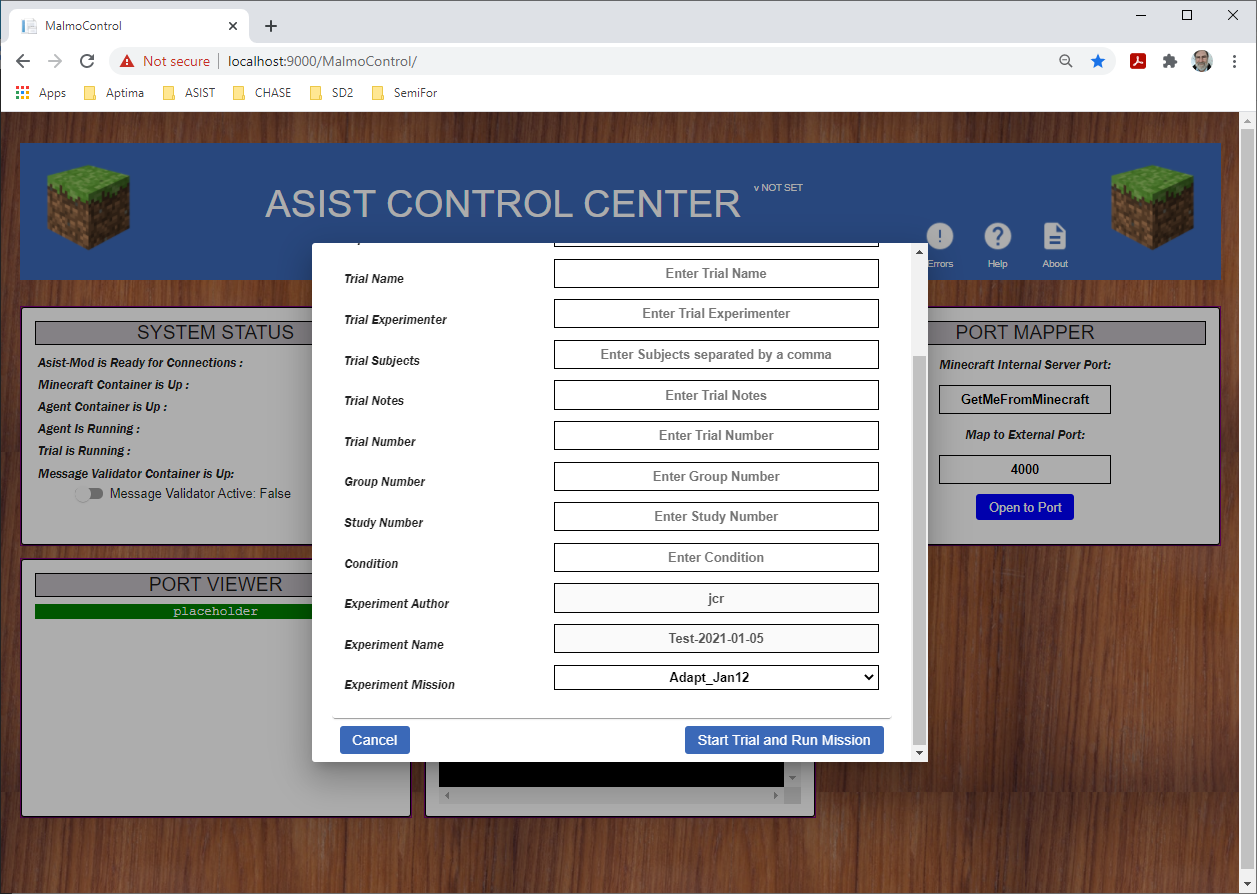
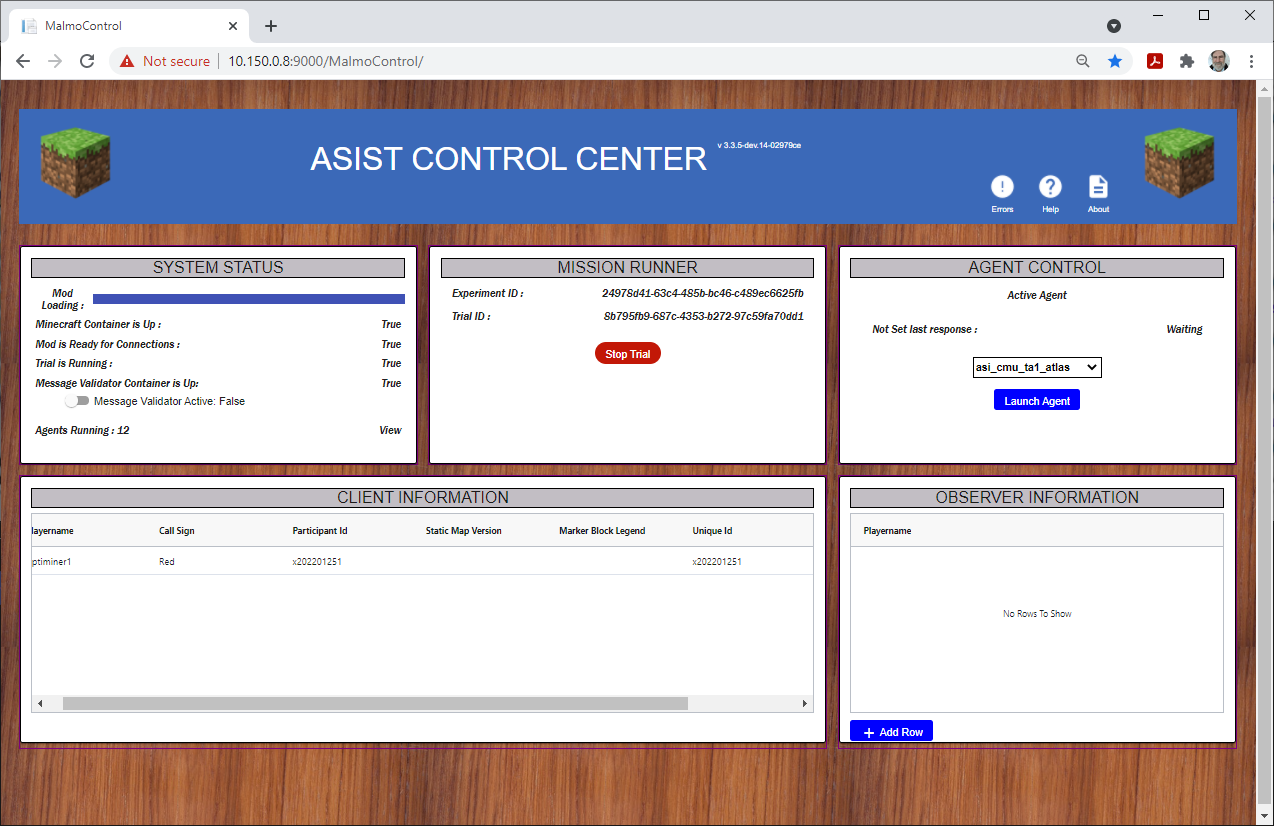
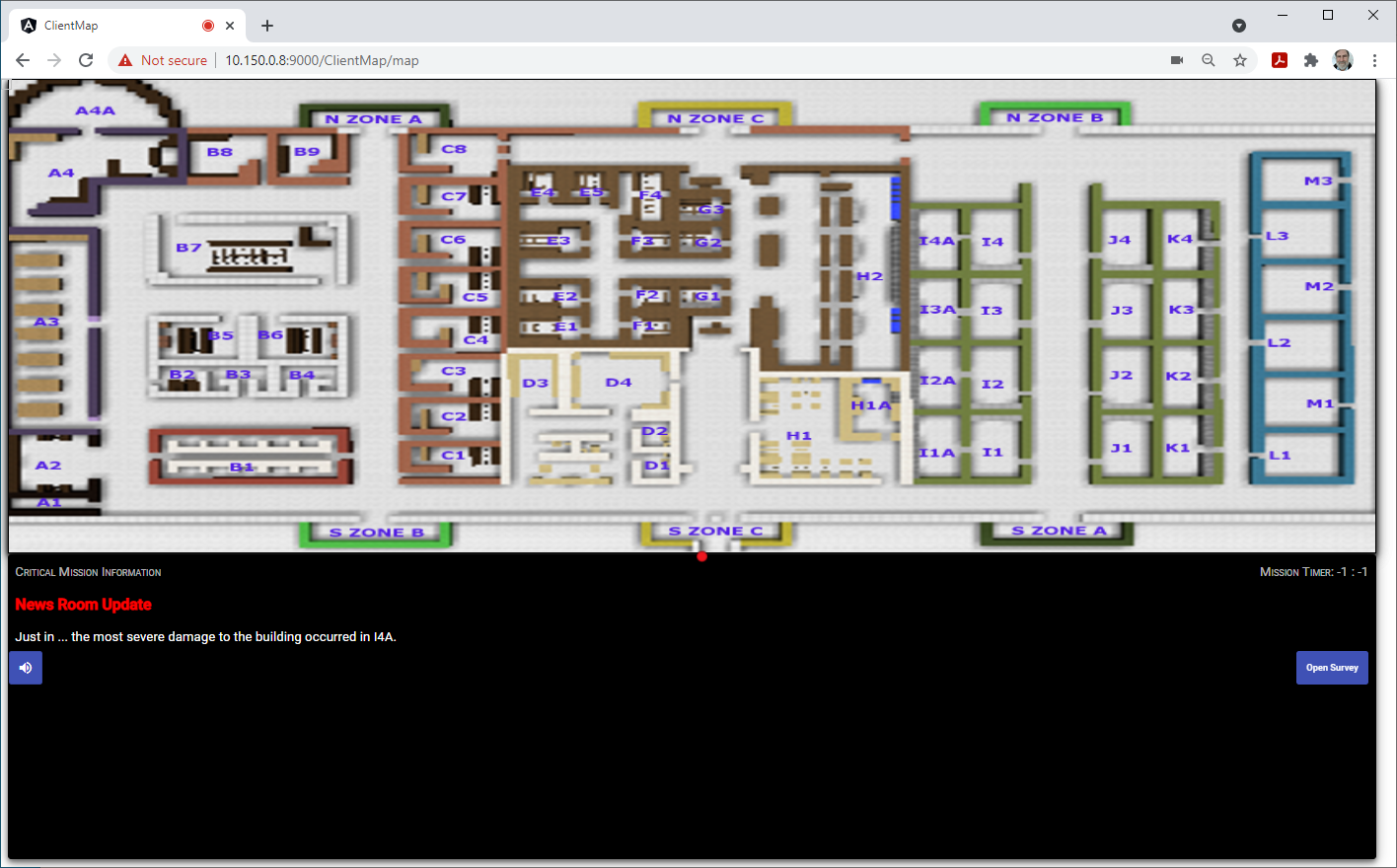
The testbed has the ability to run a number of ASI agents that generate interventions to help the participants. Generally, only one ASI agent is run at a time during a testbed trial. The ASI agents can be dynamically started and stopped using the upper right panel in the ASIST Control Center screen. Starting an agent should be done before the trial starts so the agent receives all of the initial trial information messages that are published when the trial starts.

To start an ASI agent select the agent from the pull-down list and then click the "Launch Agent" button. While the agent is running, the status line in the panel should update every 10 seconds with a time. This indicates that the agent is running.

When the trial is over, the agent can be stopped by pushing the "Stop Agent" button.

## Creating a Trial and Starting the Mission

After all the client information table fields are filled in

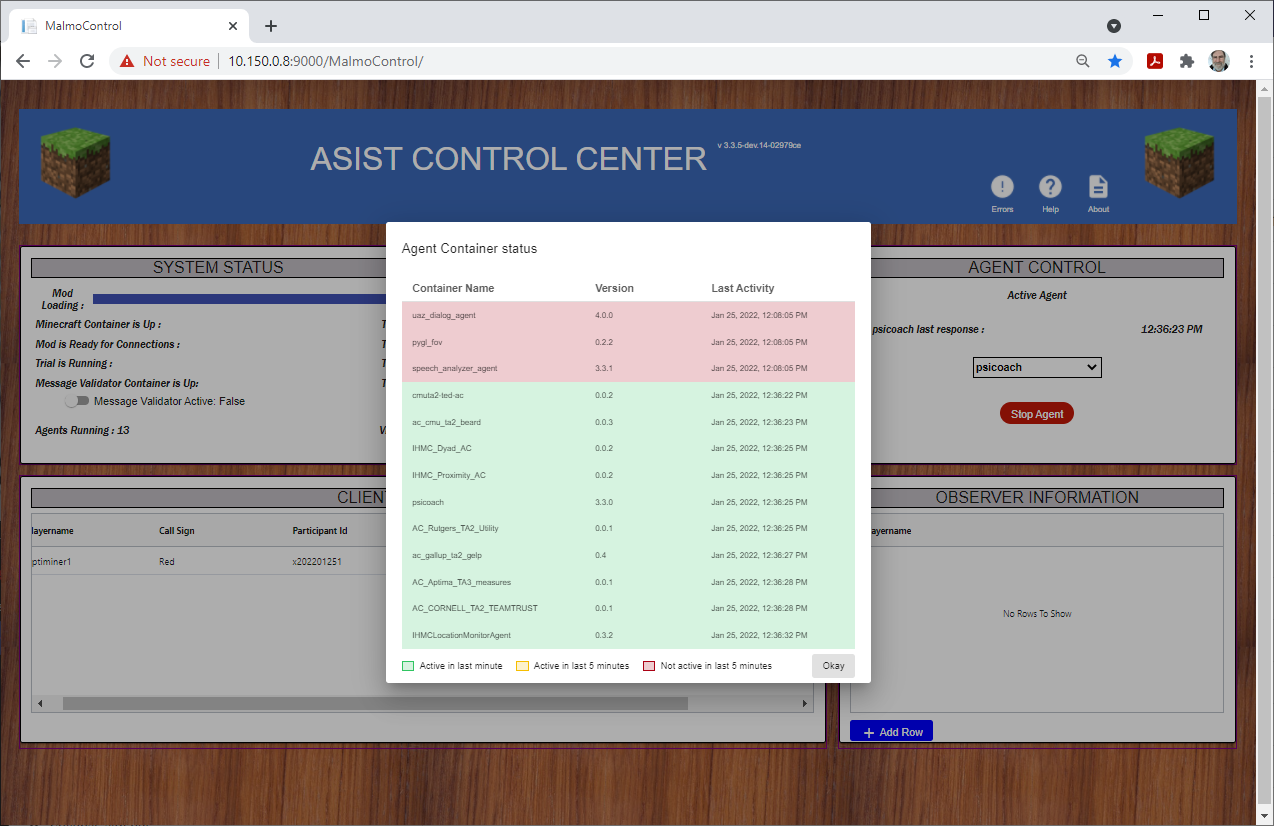
* Using the top center panel, click on the "Create Trial and Run Mission" button
* A list of existing Experiment should be displayed (if there are any). Either select one of the existing experiments by clicking on the Experiment in the list or click the "Create Experiment" tab on the top of the panel.
* If you need to create an experiment, click the "Create Experiment" tab and fill in the fields about the experiment then Click "Save to Experiment Store" at the bottom left of the panel.
* If you want to run another trial on an existing experiment, then select the experiment from the list and click "Create Trial" tab at the top of the panel.
* When you get to the "Create Trial" tab, the Experiment ID should be filled in. Fill in all of the information for the trial.   
  
* In the "Experiment Mission" field select the mission that you want to run for this trial.
* When all the fields are filled in click the "Start Trial and Run Mission" button at the lower right of the panel.  
  
* The mission will now be launched in the testbed. The "Trial Stop" button will appear under the "Create Trial and Run Mission" button. When the mission is ready to be used, the Mod Locading progress bar will be all the way to the right and the "Minecraft Container is up" and "Mod is Ready for Connections" status in the top left panel will change from False to True. At this point the ASIST Control Center screen should look like:  
  + When the "MISSION RUNNER" panel in the top center is populated with information, copy the value of the trial\_id. This will be used later to export the trial metadata to a file.
* After the mission is started and the participants have selected their roles, the ClientMap page for each player will be updated. They will look something like:  
  

## Starting a mission without an observer

If you don't have an observer in the trial and need to start one of the Saturn missions, you can use this procedure:

1. Start the mission as above
2. Have at least one player join the Minecraft game
3. In the Minecraft client type the following command  
   /tp <Minecraft player name> -2155 144 25
4. Replace <Minecraft player name> with the exact name of the player you want to teleport to this location.
5. Turn the player around and you should see the following  
   
6. Right click the "M" block right in front of you and the rubble wall just above should turn into a QR code.
7. The mission has now started.
8. To get back to the mission equipment room, issue the following command in the Minecraft client  
   /tp <Minecraft player name> -2152.5 60 120.5
9. You should teleport back to the equipment room and are ready to play the mission.

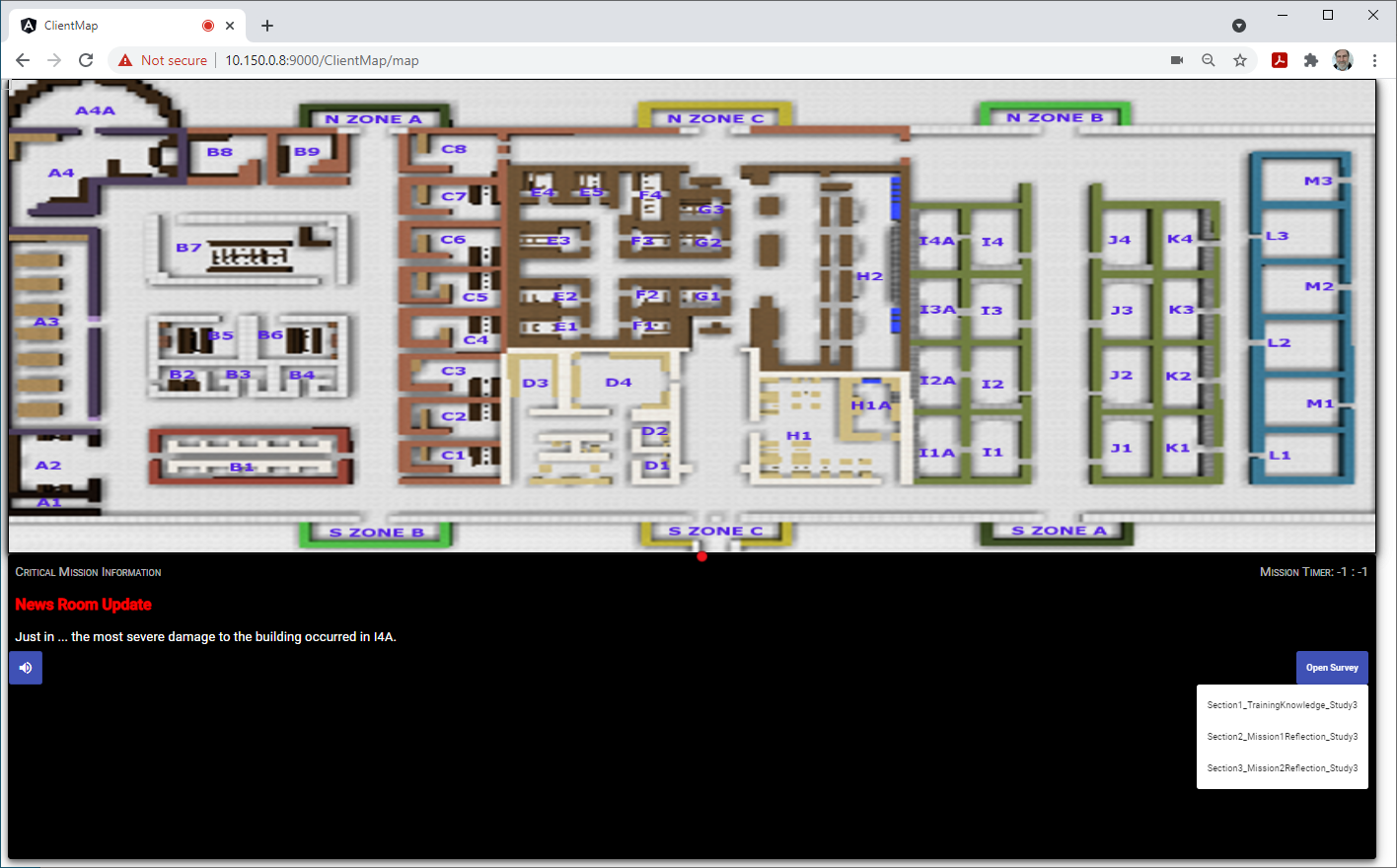
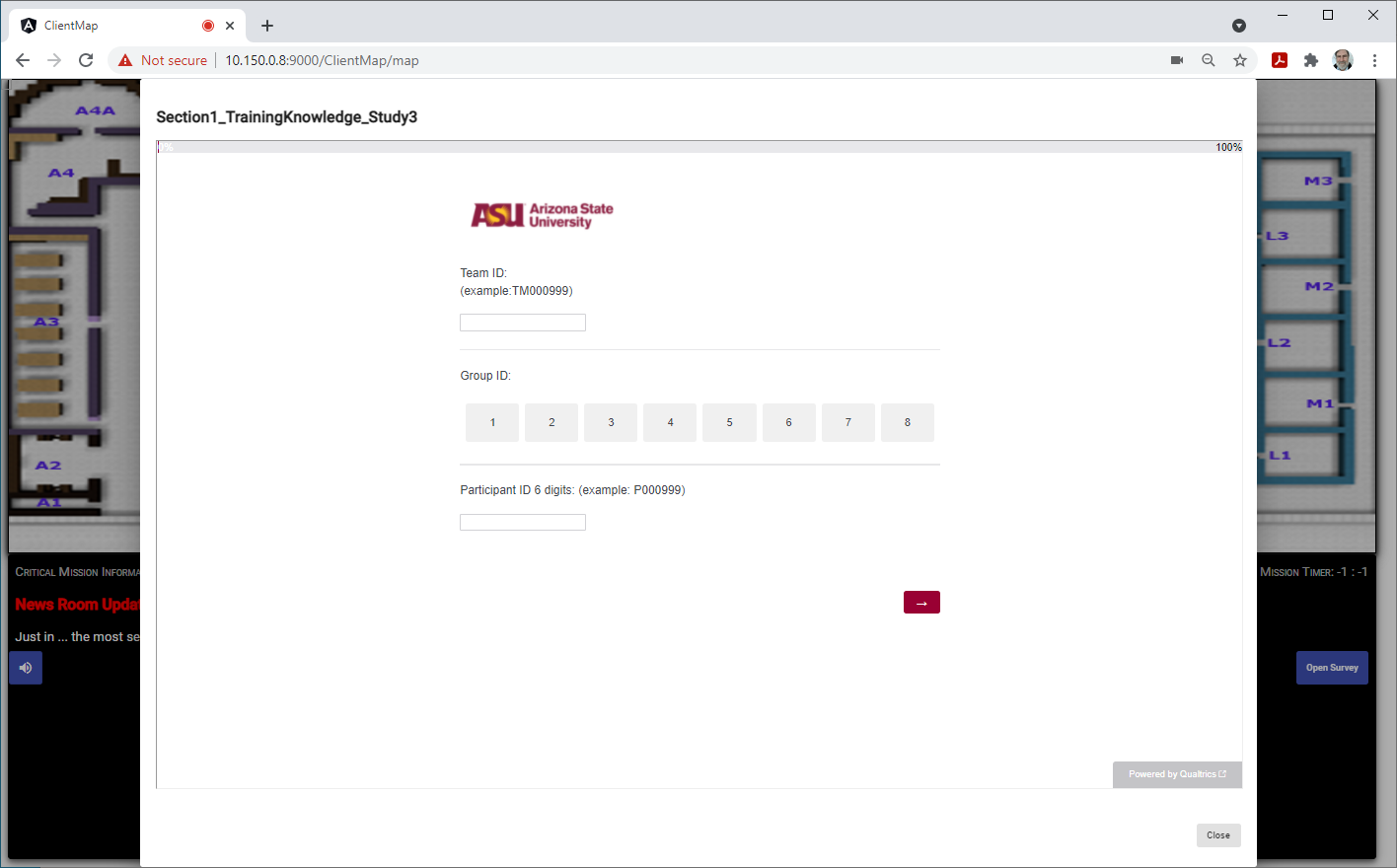
## Checking the testbed components are running

Each agent publishes heartbeat messages and responds to rollcall messages. The status of all the agents can be viewed by clicking on the "Agents Running" line in the "System Status" panel (upper left) of the ASIST Control Center. The Agent Container status panel looks something like:  


Any agent that hasn't sent a heartbeat message within the last 5 minutes will be shown in red at the top of the list. Any agents which haven't sent a heartbeat message for more than 1 minute will be shown in yellow and the agents which have sent more recent heartbeat message will be down in green. The panel can be closed by clicking the "Okay" button at the lower right of the panel.

## Taking a survey

After the mission is started, but before the players join Minecraft and push the start mission button in the game, is a good time to take the survey. It is important to have an active trial when taking a survey so that the survey is tagged with a trial\_id and will get export with a trial.

* Click on the Open Survey button on the ClientMap page and a menu of surveys will appear. The page will look like this:  
  
* Select the survey to be taken
* A pop up panel will appear which contains the survey. After responding to all of the survey questions, Close the pop-up panel. When the survey panel is open the screen should look like:  
  
* After completing the survey (there might be multiple pages), make sure to push the Close button at the lower right corner of the panel.

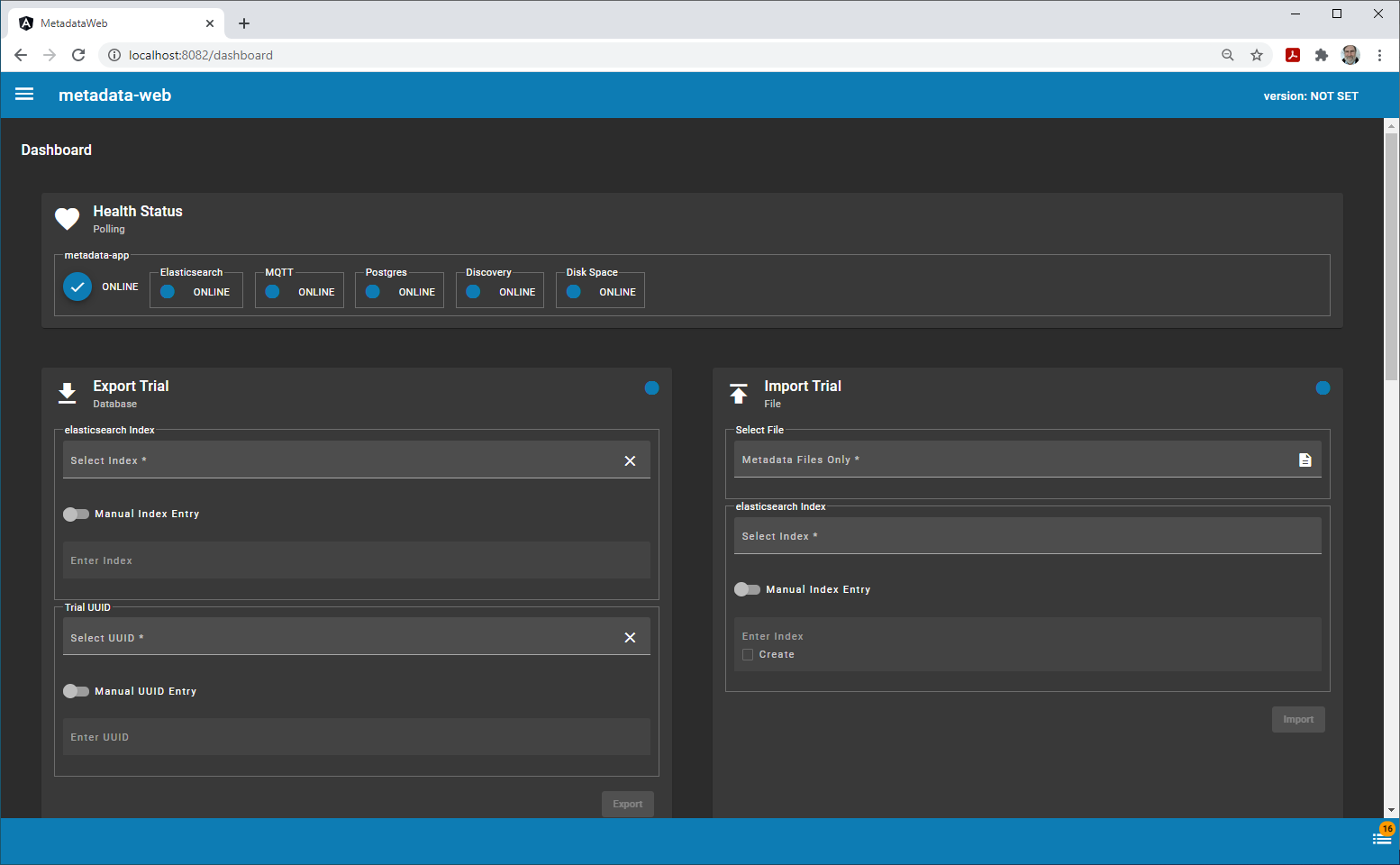
## Connecting to the Running Mission with Minecraft

* Start up Minecraft. The Minecraft program on the client needs to be configured for use with the ASIST testbed including the correct version of the ASIST mod (the same as is running in the testbed) and configured with the right version of Forge.
* In Minecrft click on "Multiplayer"
* Click "Direct Connect"
* In the "Server Address" field type in the address of the system running the testbed and the port number 25565. For example  
  my.testbed.com:25565  
  Then click Join Server
* If you get an error connecting make sure you have network connectivity to the testbed system and the correct version of the ASIST mod copied to the correct directory so that Minecraft has access to it. Also make sure that the Minecraft username is list in the ASIST Control Center Client information table.

## Ending the Trial

* When the trial is over in the ASIST CONTROL CENTER web page, click on the "Stop Trial" button in the top middle panel

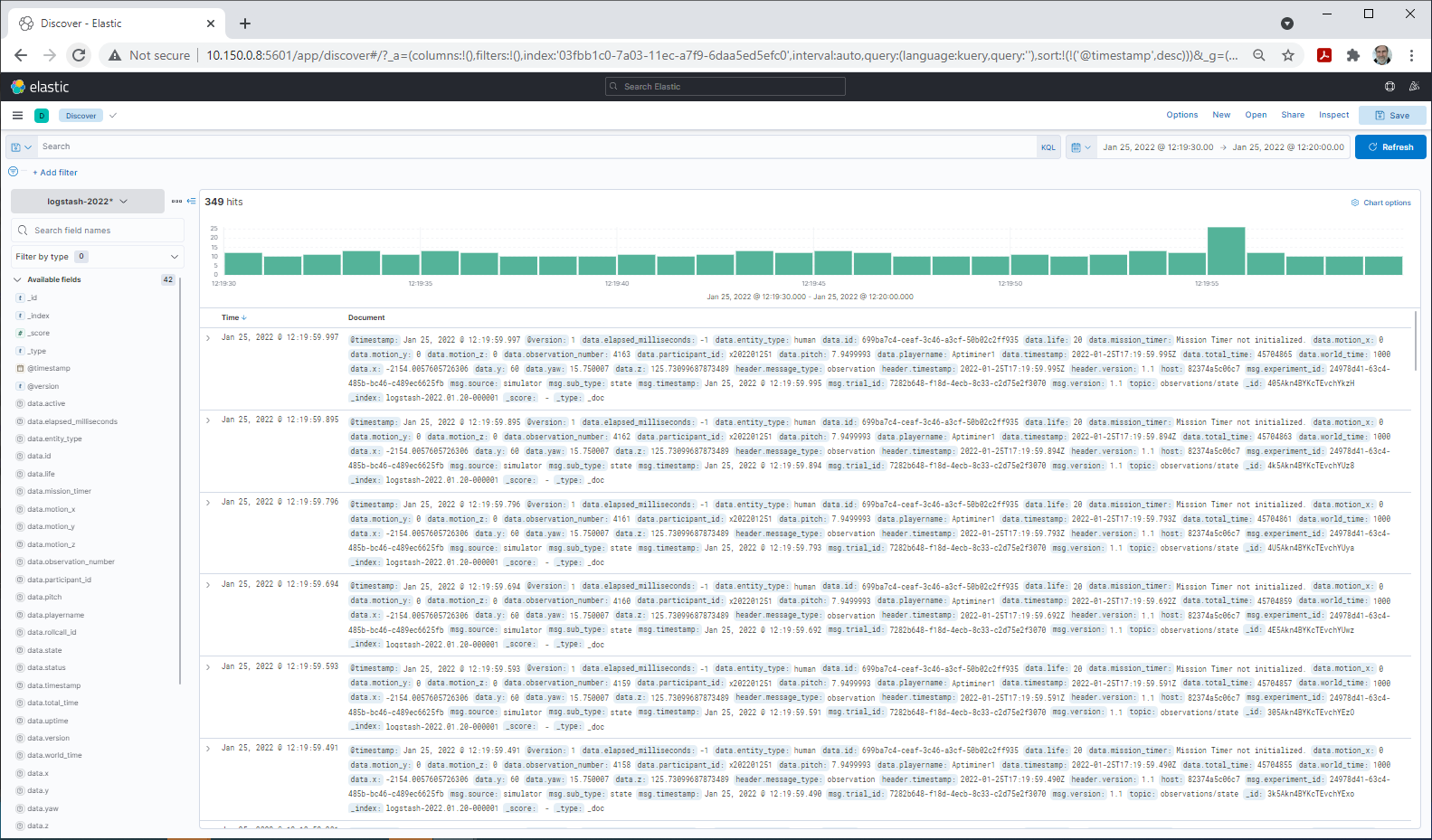
## Exporting the Trial Data

* Data can be exported from the testbed using the Input/Export Dashboard. To get to the dashboard, open a browser (use a Chrome browser) and go to: http://<testbed\_host>:8082/dashboard  
  
  + The Import/Export dashboard is connected correctly if the Health Status panel on the top bar has all circles blue and online checked.
  + In the Export Trial panel, click on the "elasticsearch index/Select Index" pulldown and look for the elasticsearch index item that contains the trial. Elasticsearch can be configured differently but this should be an entry that starts with "logstash" and could have a date after it. Select that item in the list
  + Click on the "Trial UUID/Select UUID" field and a list of trials in elasticsearch will be displayed. Search the list for the trial\_id that you copied when you start the trial above.
  + Click the "Export" button to begin the export.
  + Export files can be large and sometimes can take 5 to 10 minutes to complete. When the export is complete, the file containing the messages from the trial will be downloaded to your local system. You may need to ok a pop-up panel or two to complete the download.
  + This screen can also be used to run replays, import metadata files into the testbed instance, import replays and export replays.

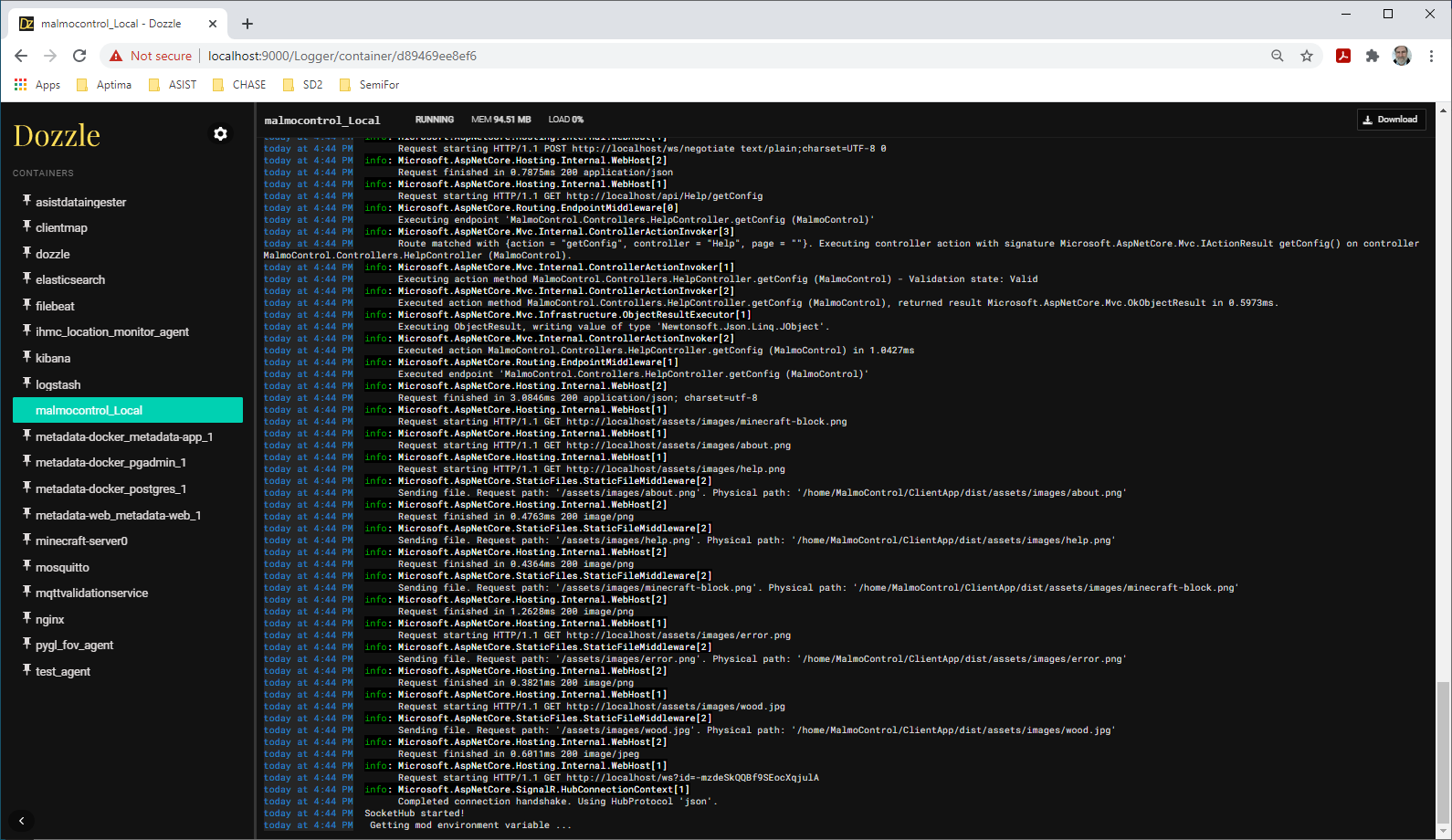
# Developer/Admin

This role is for people who are working on the testbed to enhance it or are maintaining or administering it.

## Checking data that is being Captured

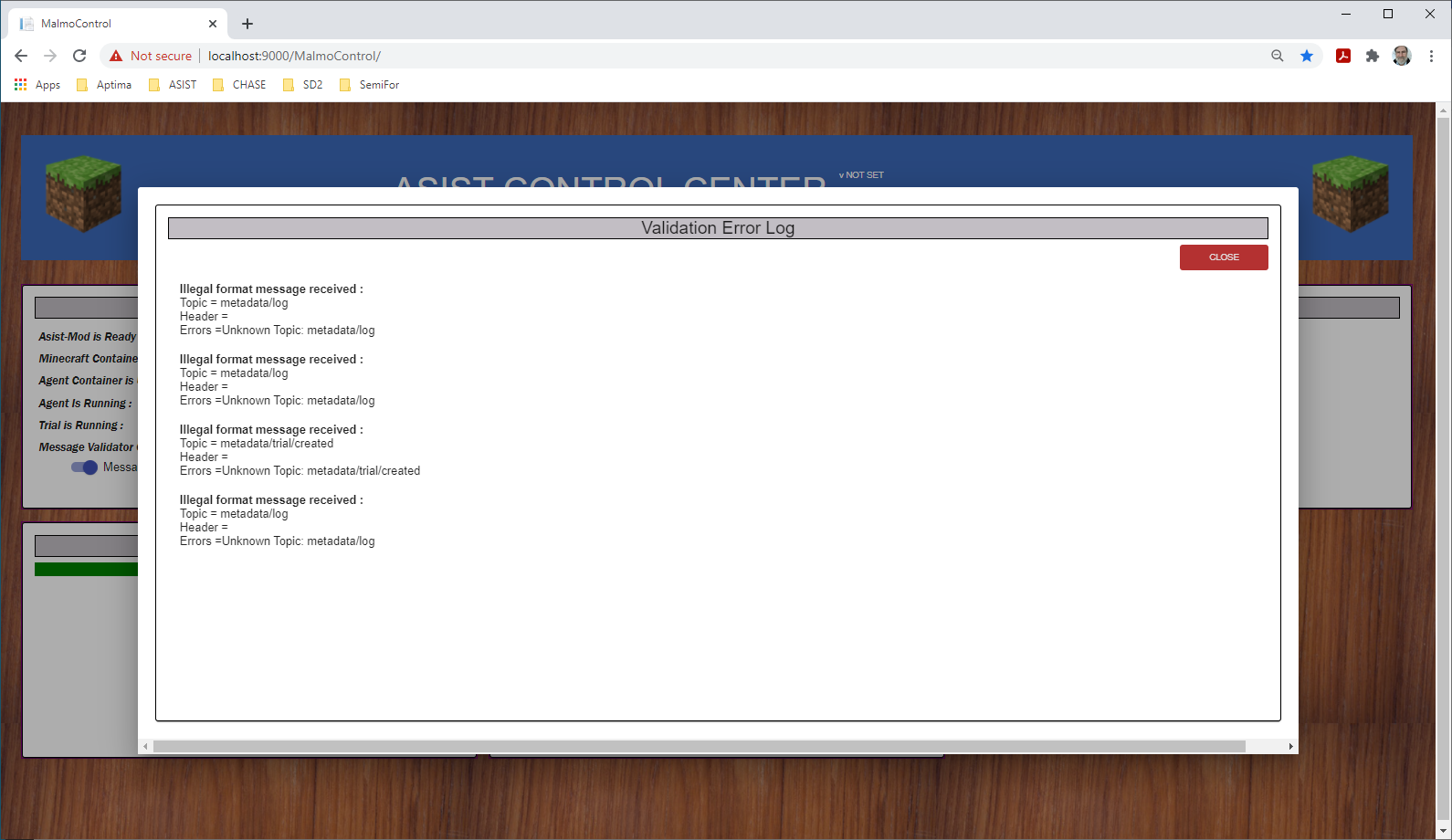
* There is an installation of Kibana in the testbed. This can be reached via the browser at: https://<testbed\_host>:5601
* When you start up a new instance of the testbed and enter kibana, you will need to create an index pattern for kibana to use to find the data.
* In Kibana click on the "Discover" app which is usually the top icon on the left column of apps.
* The Discover page should look something like:  
  
  + The Kibana index pattern should be selected just below the Search line in the upper left. Read the documentation to learn how to create Kibana Index patterns if you don't already have one.
  + In Kibana you can set the time frame that you want to query data for and you can filter data using the Search field. An example of a search query is msg.sub\_type:Event\*

## Checking the Container Logs

* The testbed contains a system wide logging capability called Dozzle.
* To view logs in a browser go to: https://<testbed\_host>:9000/Logger  
  
* A list of the containers are shown on the left. You can select one of the containers and the log for that container will appear on the right.
* The logs are only saved for the lifetime of the container (or until the container is restarted).
* This is a script file in Tools/docker-logs/get\_all\_logs.sh which can be used to capture all of the docker logs and write to a directory.

## Message Validation

The testbed contains a capability that can validate all published messages against a json schema. The schemas are all located in the repo under the MessageSpecs directory. Each messages has some documentation on the fields and the json schema definition.

* Message validation can be toggled in the ASIST CONTROL CENTER by shifting the slider in the "SYSTEM STATUS" panel in the top left panel. The "Message Validator Container is Up:" status should be True.
* To see any message validation errors click on the Errors button in the upper right of the ASIST CONTROL CENTER  
  

# Basic Setup Instructions

This section provides some basic instructions for setting up the testbed. Some setup steps may vary depending on the system being used.

1. Clone the testbed repository
2. Make configuration changes:
   1. Edit Local/Nginx/nginx.conf
      1. change rewrite statements and replace localhost with the DNS or IP of the server system.
   2. If you want to use speech recognition, make sure there is a Google access token file in Local/AsistDataIngester
   3. Edit metadata/metadata-web/metadata-web.env
      1. Replace the host metadata-app, elasticsearch and mosquitto with the DNS or IP of the server systems
   4. Edit metadata/metadata-docker/metadata-app.env
      1. Edit: MQTT\_BROKER\_URL and ELASTICSEARCH\_HTTP\_HOSTS with the DNS or IP of the server system
   5. Make sure there is a username and password that can access the container registry in Local/MalmoControl/appsettings.Production.json for the RegistryCredentials property
3. run the build script from the Local directory:  
   testbed\_build.sh (.cmd if running on Windows)
4. Run the up core script to bring up the core containers of the testbed. If you want to run all of analytic components of the testbed include the -I option with the script. Run the script from the Local directory:  
   testbed\_up\_core.sh or  
   testbed\_up\_core.sh -i
5. Check Dozzle to make sure all of the containers have started and are functioning.