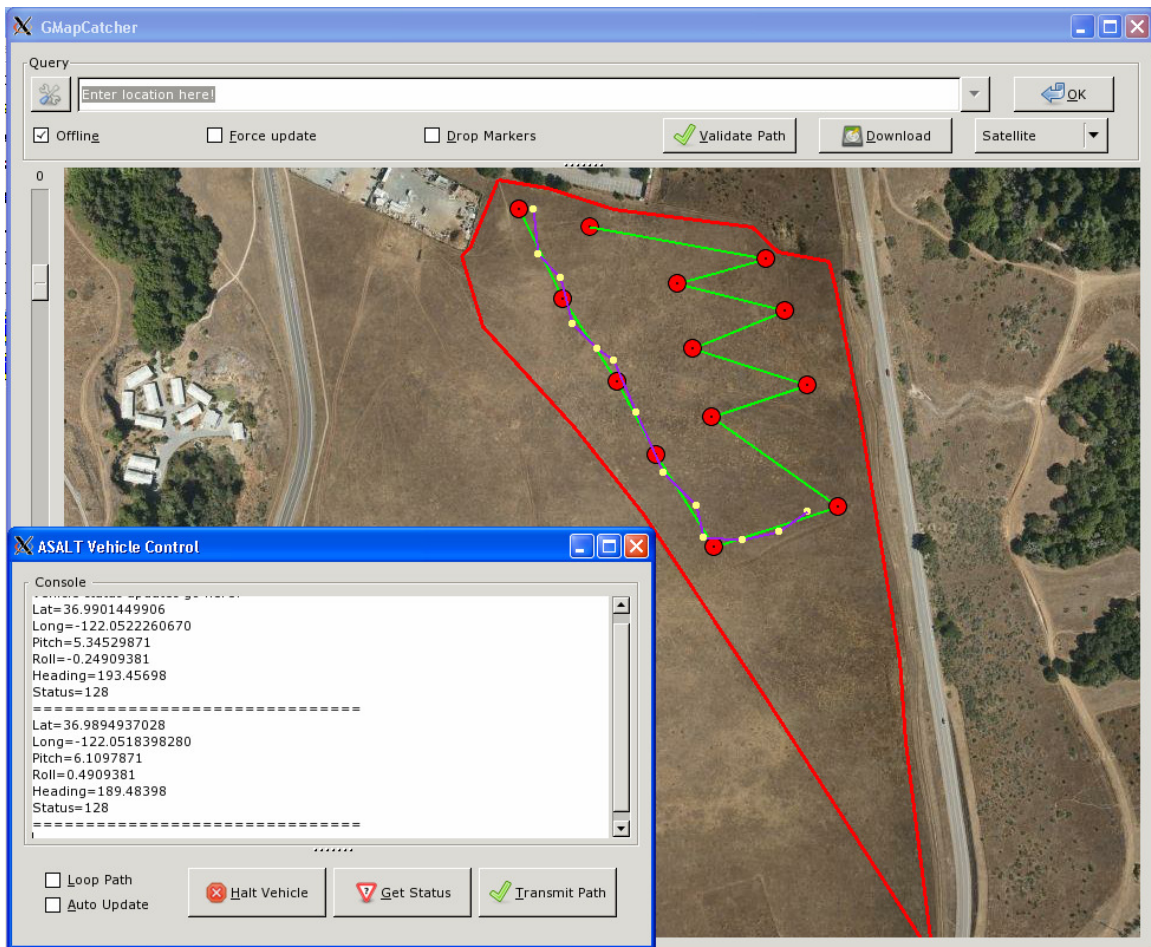


# ASALT Vehicle User Interface Manual



## **Table of Contents**

- 1 Overview**
  - 1.1 Installation**
    - 1.1.1 Linux**
    - 1.1.2 Windows**
    - 1.1.3 Mac**
  - 1.2 Running the Program**
- 2 ASALT Settings**
- 3 The Map Area**
  - 3.1 Moving Around**
  - 3.2 Dropping Markers**
  - 3.3 Deleting/Editing Markers**
  - 3.4 Setting Wait States**
- 4 Path Validation**
  - 4.1 Defining No-Go Areas**
  - 4.2 Removing No-Go Areas**
- 5 Vehicle Control Window**
  - 5.1 Transmitting Path**
  - 5.2 Receiving Updates**
    - 5.2.1 Auto-Update**
  - 5.3 Halting the Vehicle**
- 6 Configuration and Data files**
  - 6.1 gmapcatcher.conf**
  - 6.2 Markers**
  - 6.3 Statuses**
  - 6.4 No-Go**
- 7 Source Files of Interest**

## 1 Overview

The ASALT Vehicle user interface is an extension of a piece of open source software called GMAPCatcher. Since the ASALT UI is typically used in an area without internet connectivity, GMAPCatcher allows pre-downloaded map tiles from Google Maps to appear without requiring an internet connection. This way, you can get a bird's eye view of the vehicle's location and choose a path for it to follow. This software also supplies functionality to obtain information about the vehicle's current status, as well as communicate simple commands (such as HALT) to the vehicle remotely. This guide outlines the functionality of the ASALT extension to GMapCatcher. If you would like to read the user manual for GMAPCatcher, please refer to this link:

[http://code.google.com/p/gmapcatcher/wiki/User\\_Guide](http://code.google.com/p/gmapcatcher/wiki/User_Guide)

### 1.1 Installation

The ASALT UI is written in the Python Programming language, since Python is an interpreted language, the program does not typically require "installation" in the classic sense. All files that GMAP needs are located in its folder, with the main file called **maps.py**. Let's see how to run the ASALT UI on various OS platforms.

#### 1.1.1 Linux

Linux installation is the most straight-forward; simply copy other GMAP folder to whichever folder you choose. Then program can then be run by navigating to that folder in the terminal.

#### 1.1.2 Windows

In order to properly run the ASALT UI in Windows, the use of Cygwin is required. Cygwin is a program for Windows which emulates the UNIX environment in a standalone program.

The Cygwin setup tool can be acquired here:

<http://www.cygwin.com/setup.exe>

Once you have downloaded setup.exe, run the installer. The installer will have several steps.

1. Select 'Install From Internet'
2. Select cygwin's directories (default is usually fine)
3. Select your internet connection settings (usually direct connection)
4. Choose a mirror close to you for cygwin to download the packages to your computer
5. **Important!** Cygwin will now download and display the available packages; the following are **required** to run GMAPCatcher with the ASALT UI.
  - a. Python->Python
  - b. Python->Python-gtk2.0
  - c. X11->xinitCygwin will automatically select other packages which depend on the packages selected above
6. Select 'Next' to install Cygwin.
7. After finishing installation, be sure to run XWin Server, instead of cygwin, to allow the GTK windows to appear properly.

### 1.1.2 Mac OS X

The ASALT user interface requires the X11 windowing environment in order for it to work with Mac OS X. If you are using a version that does not include X11 by default please upgrade or download a version of X11 that will work with your version of Mac OS X. Instructions for installing X11 are outside the scope of this document.

Follow the install directions for MacPorts at <http://www.macports.org/install.php>.

Use MacPorts to install PyGTK using the following command:

```
sudo port -v install py25-gtk
```

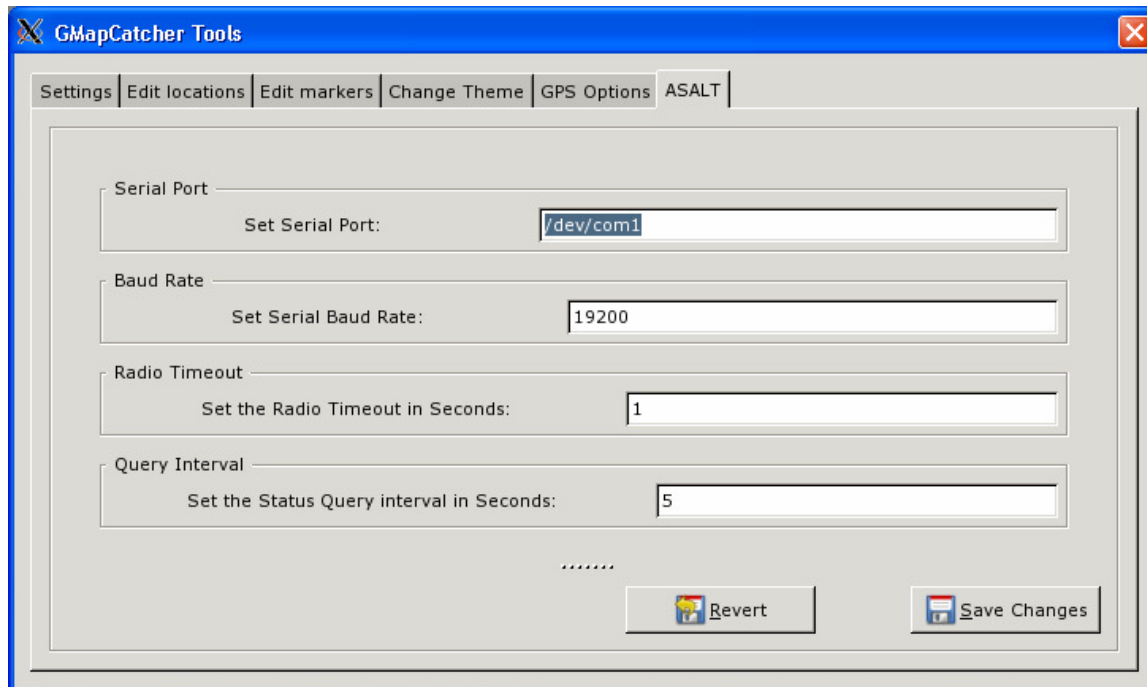
Ensure you start the ASALT user interface using the version of python installed by MacPorts by executing the following command:  
/opt/local/bin/python2.5 /path/to/gmapcatcherfolder/maps.py

## 1.2 Running The Program

To run the program, type this command:

```
> $ python gmapcatcherfolder/maps.py
```

## 2 ASALT Settings



The ASALT settings can be accessed by selecting 'ASALT' from the settings drop down menu on the upper left corner of the main window. There are four main settings for the ASALT vehicle.

**Serial Port:** Sets the serial port for the ASALT radio link. (ex. COM1, /dev/ttyS0)

**Baud Rate:** Sets the baud rate used by the radio. The default is 19200. If changing this setting, make sure that the radio baud has been changed first.

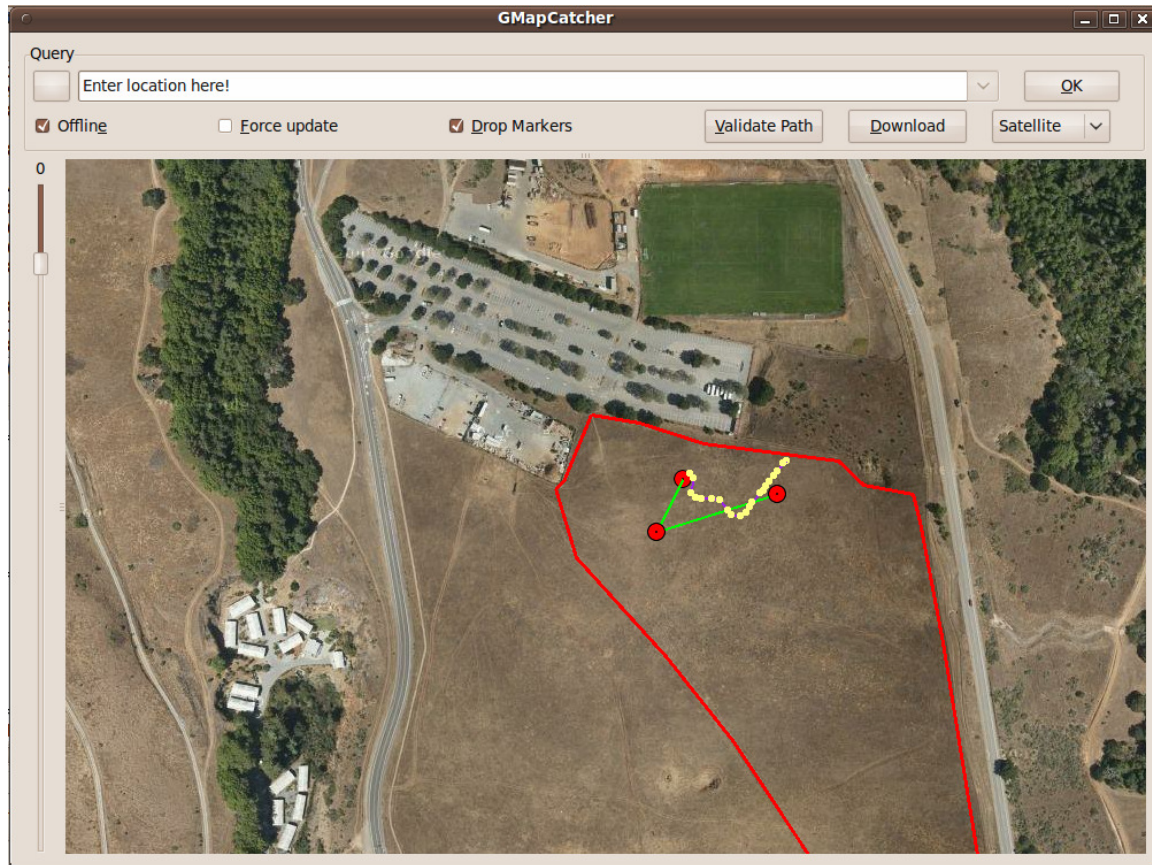
**Radio Timeout:** The amount of seconds before a status query times out. This typically means the radio transmission was not successful.

**Query Interval:** The amount of time in seconds between each status query when 'Auto-Update' is enabled.

Once settings have been adjusted, they must be saved by clicking 'Save Changes.' After saving the settings, you may exit the window by clicking the X in the upper left hand corner.

### 3 The Map Area

The map area displays the area that the vehicle may traverse. From here, you can select a sequence of waypoints, validate them, and later transmit them using the ASALT vehicle control window.



#### 3.1 Moving Around

Moving around the map works much like Google maps works online. Holding down on the left mouse button and dragging will move the map in the desired direction. Double clicking the map will zoom in. You may also choose to zoom using the zoom bar on the left hand side of the window, or by right clicking and selecting 'Zoom Out' or 'Zoom In.'

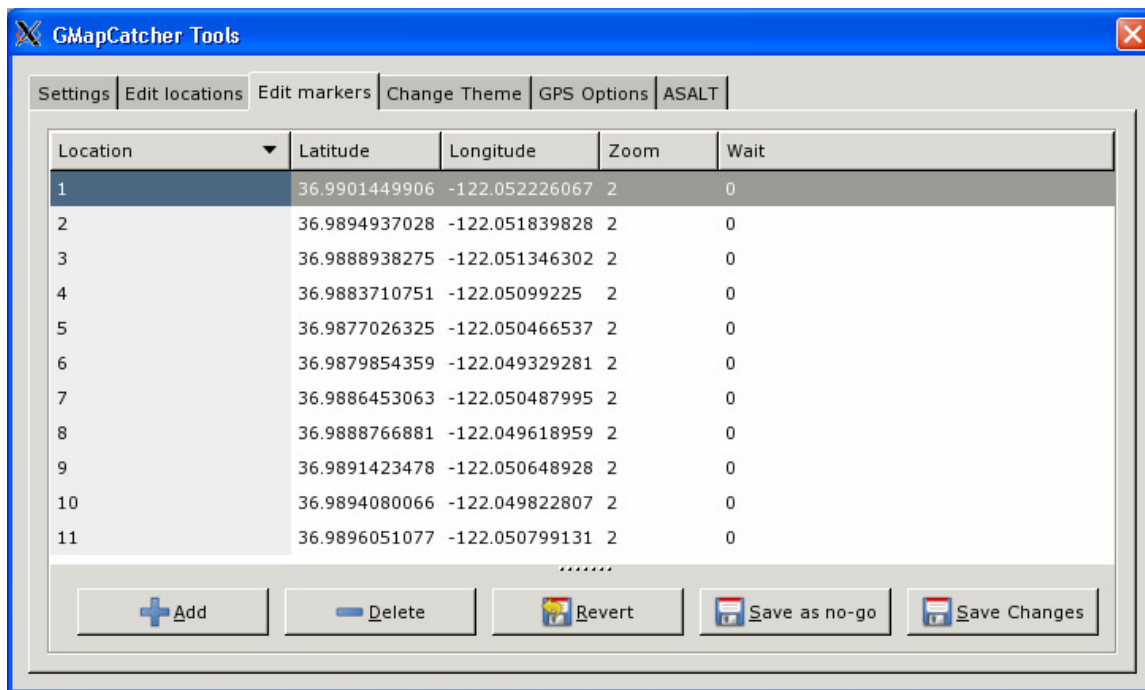


### 3.2 Dropping Markers

There are two ways to drop markers. First, you may drop a marker by right clicking on the desired location on the map, and selecting 'Add Marker.' If you would like to drop markers in a quicker manner, check the 'Drop Marker' checkbox above the map area. Checking this option allow you to add new markers by simply left clicking in the desired location. Note that if you would like to move the map, it is recommended that you uncheck 'Drop Marker' before clicking and dragging.

### 3.3 Deleting/Editing Markers

By opening the 'Edit Markers' tab in the settings window, you may add, delete, or edit previously dropped markers.



This window displays all markers currently placed on the screen in the order they were placed. To edit a field, simply double click on the desired cell, edit the value, and press enter. Like the ASALT settings tab, you must click 'Save Changes' to save you changes.

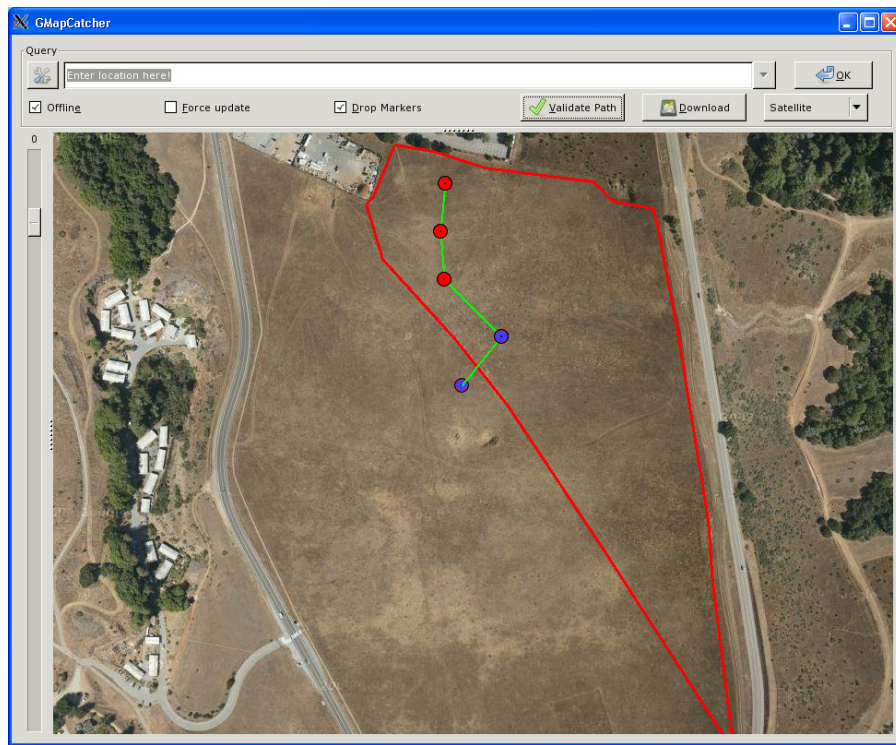
To delete a marker, select the row/marker that you wish to delete, and click the 'Delete' button. You may notice that the marker has not yet disappeared from the map area in the main window. This is because you must 'Save Changes' before the map area will be updated.

### 3.4 Setting Wait States

Wait states tell the vehicle how long it should wait when it has reached a waypoint marker. To set a wait state, simply change the desired marker's wait field to a numerical value. The value represents the number of seconds the vehicle will wait when it reaches that position.

## 4 Path Validation

The ASALT UI will validate the selected path to see if it crosses over pre-defined “no-go” boundaries. The default no-go boundary is the fence line of the field which the vehicle is defined for. You will also have the option to define their own no-go areas for use in different areas besides the East Field at UCSC.



After you have placed the desired markers on the map area, click the ‘Validate’ button located above the map area. This button has two purposes: first, it runs the validation algorithm and highlights the problematic markers. It also launches the ASALT vehicle control window. If the path is invalid, this note will appear in the ASALT console

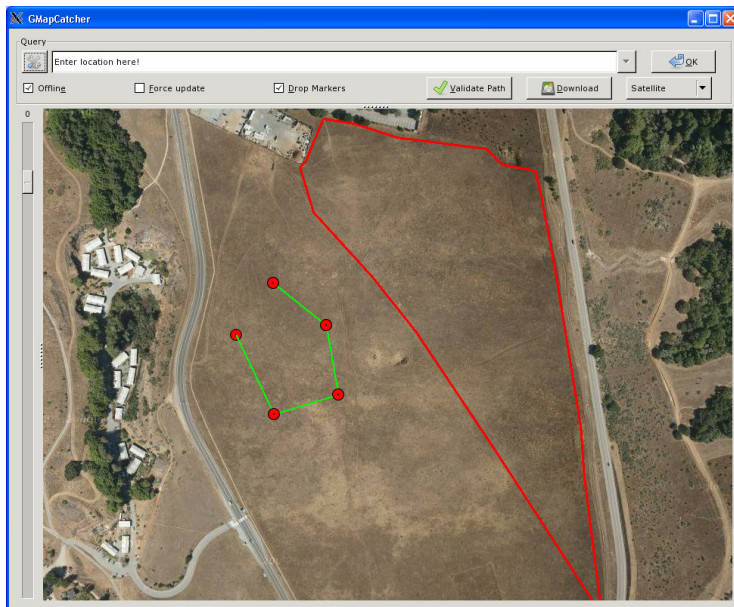
**“Your selected path appears invalid, please double check the markers and adjust as necessary”**



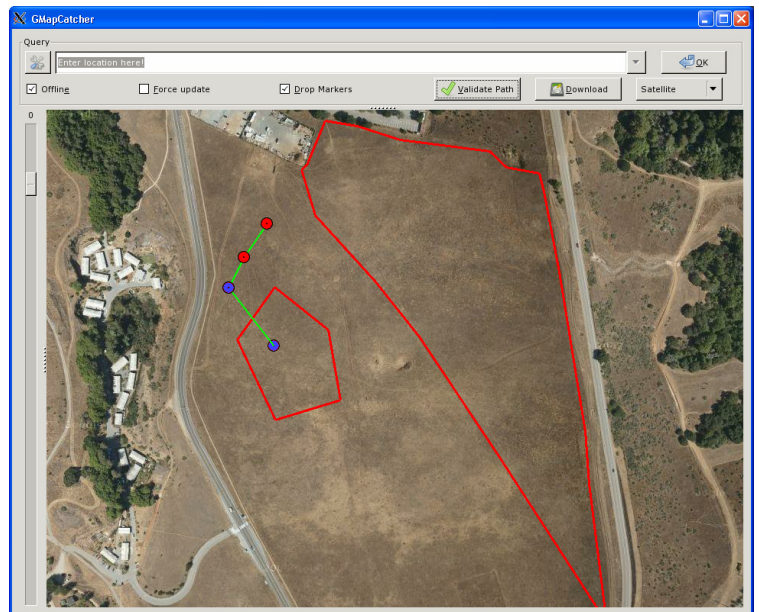
Note that you can still transmit an invalid path, so make sure you have properly corrected your path (if necessary) before transmitting.

#### 4.1 Defining No-Go Areas

The ASALT UI allows you to easily define no-go areas as you go.



Before: Defining the no-go area



After: The no-go area appears on screen

Defining no-go areas is simple and straightforward. First, define a set of markers that you wish to represent a no-go area. Don't worry about connecting the first and last markers, the ASALT UI will do it automatically for you. After you are satisfied with your selection, open the 'Edit Markers' setting window as described in section 3.3. Simply click the 'Save as no-go' button to save the area as a no-go

**Note:** Make sure to only hit 'Save as no-go' once, pressing this button more than once will cause the area to be added to the no-go file multiple times. Although this does not affect functionality of path validation, it may result in performance issues.

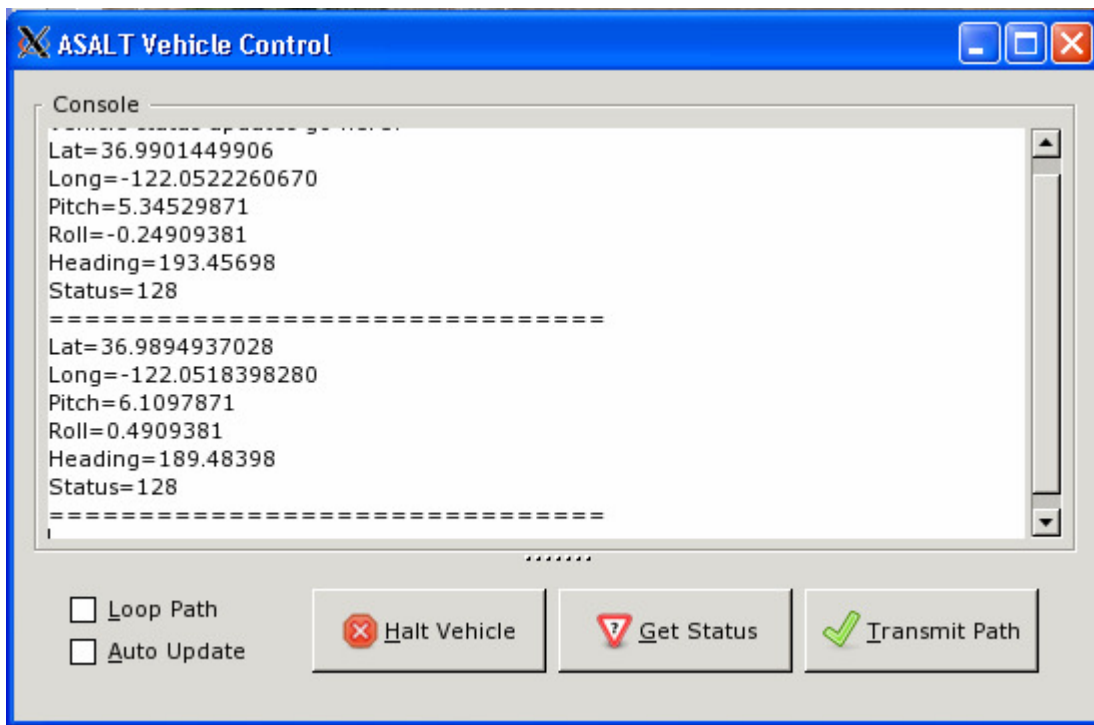
Currently, the ASALT UI will not render the no-go area until it is restarted, so in order to see the newly defined no-go area appear in the map area, simply delete the markers, and restart the program.

## 4.2 Deleting No-Go Areas

To delete a no-go area, you must delete it from the no-go data file. For more information about the ASALT UI's data files, please refer to section 6. Note that the default no-go area will reappear if the no-go data file has been completely deleted.

## 5 The Vehicle Control Window

After you have selected and validated a path, you are almost ready to start the vehicle on its path. From here, you will be able to transmit the markers to the vehicle, obtain status updates from the vehicle, and halt the vehicle remotely.



### 5.1 Transmitting the Path

To transmit the path, just click the "Transmit Path" button. Before transmitting the path, make sure you have exited the vehicle's command mode first.

**Note:** that clicking "Transmit Path" in command mode or during the vehicle traverses its path, unwanted errors may occur and you will need to restart the vehicle's main board.

## 5.2 Receiving Status Updates

In order to receive status updates, just click on the ‘Get Status’ button. Depending on the distance between the vehicle and the base-station running the ASALT UI, the time to receive the update may change. It is recommended that you adjust the timeout in the ASALT settings window accordingly.

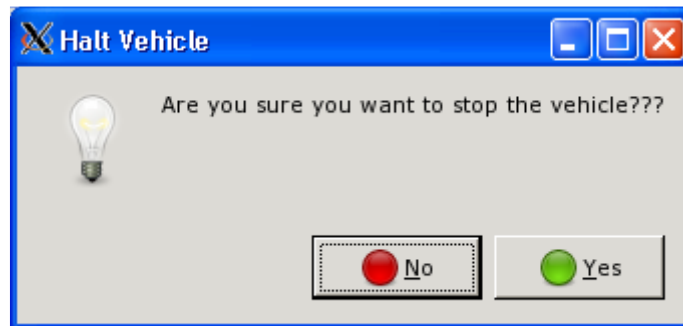
Once a status update has been received, the vehicle’s location will appear on the map, denoted by a yellow dot-shaped marker. Each status update adds a status marker to the map area. Each subsequent status marker will be connected with a purple line, showing the vehicle general direction.

## 5.3 Auto-Update

By checking the ‘Auto Update’ checkbox in the ASALT vehicle control window, the UI will automatically poll for status updates. The auto-update interval is configured in the ASALT settings window.

## 5.4 Halting the Vehicle

If you would like to halt the vehicle while it is traversing its path, simply click on the “Halt Vehicle” button. This will cause a verification alert window to appear to double check your decision to halt the vehicle.



Click ‘No’ if you do not wish to stop the vehicle, and ‘Yes’ to proceed with transmitting the halt message to the vehicle. Once the vehicle has been halted, it waits for new coordinates to be sent from the UI. At this point you may choose to send it more coordinates, or reset the main board on the vehicle to re-activate command mode.

## 6 Configuration and Data Files

By default, GMAPCatcher stores map tiles and marker data in a specific folder. For ASALT vehicle purposes, additional files have been added to this folder. The folder is configurable in the GMAPCatcher settings window, but by default the folder is:

**\$HOME/.googlemaps/**

If this folder does not exist (typically during the first time loading the program), GMAPCatcher will automatically create the folder and necessary files within. As you use the program, these files will be appended and deleted as necessary.

### 6.1 gmapcatcher.conf

**\$HOME/.googlemaps/gmapcatcher.conf**

This file holds all the configuration data for the program. If you change a parameter in this file prior to starting the program, it will be loaded into the settings. It is recommended that you adjust settings using GMAPCatcher's setting window rather than editing this file.

### 6.2 Markers

**\$HOME/.googlemaps/markers**

Format: marker id, latitude, longitude, zoom, id, wait value

This file holds the markers that have been most recently been placed on the map area. This file is read by the program at startup and the markers are placed on the screen. This means you can set a path, close the program, and the path will remain the next time you load up the ASALT UI. You may also choose to copy this file if you wish to save a path for a later time.

### 6.3 Statuses

**\$HOME/.googlemaps/asalt**

Format: latitude, longitude, pitch, roll, heading, status int

This file stores the status updates received while the vehicle is traversing a path. Since this file is cleared during program startup, it is recommended that you copy this file to another location if you wish to store the status updates for a later time.

### 6.4 No-GO

**\$HOME/.googlemaps/nogo**

Format: no-go id, lat, long

This file stores the no-go areas defined by the user. By default, the East Field fence is appended to this file with index 1, each subsequent no-go area will be added with an id = last\_id+1

## 7 Source Files of Interest

### **maps.py**

This is the main file, from here, all other windows and functions are called. This file contains functions that validate the path, and launch the ASALT vehicle control window

### **ASALTWindow.py**

This file contains the code for the ASALT vehicle control window. The main text console is implemented as a TextView area, with a function to append data to the widget.

### **ASALTradio.py**

This file contains functions for communicating with the ASALT vehicle via the ConnexLink 4490 radio. This is where you will find functions that query/parse for status updates, send coordinate data, and halt the vehicle.

### **fileUtils.py**

This file contains functions that read/write/append data to the files listed in section 6.