The Basics : Getting Started

Welcome to the AGI UTM. This document will serve as a getting started guide for BETA users of the UTM system. This will be divided into two main sections corresponding to the usage and familiarization with the user interface web portal as well as the usage of the back-end API for developers and integrators!

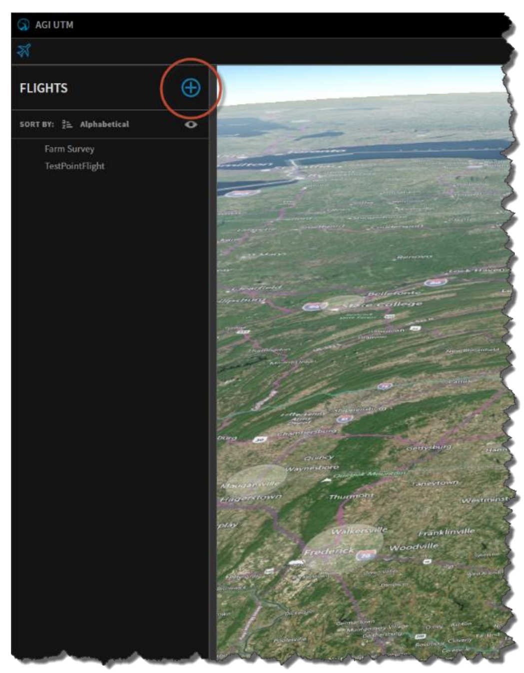
Let's Get Started!

The basic functionality of the UTM at its present state provides a user the capability to submit a flight plan into the system. This occurs with one of two different options for what a "flight plan" consists of.

<u>POINT FLIGHT</u>

The first option is referred to as a 'Point Flight' and allows a user to define an area in which the flight is going to be taking place and constrained within. This means a single central point defined with latitude, longitude, altitude as well as a maximum radius and maximum height. This will result in a cylinder representing the flight volume. Of course you will also want to define both the start and stop time of the flight.

In order to create this type of flight from the web-based user interface, you will log into the UTM system and can begin creating flights by clicking the 'PLUS' icon in the flight management panel. This will immediately open up the flight creation panel where you can enter in the relevant information pertaining to the flight.



You can use the 3D Globe Window to click and place the center point defining the location.

Enter any additional information which may be required for defining the flight and then click 'OK' when you are ready.

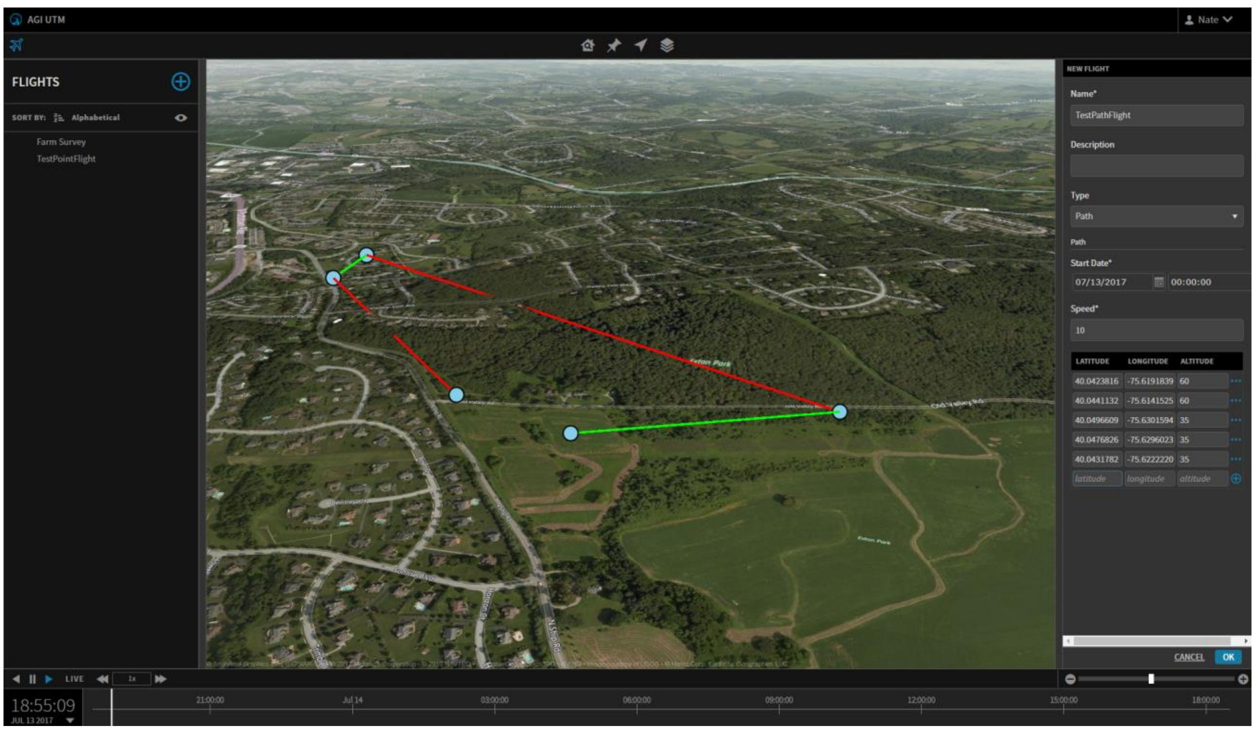
Now that you have defined the 'Point' type of flight, it will show up in the 3D Window as a large cylinder representing the volume of area that your flight will encompass.

NOTE: The UTM is powered by a robust Terrain Server which streams in the visual terrain data you see in the 3D Globe and also uses that terrain data for many analytical calculations. Without getting too deep into the technical details, be advised that clicking a 'Point' location when defining a Point Flight will query the terrain altitude or "ground" height automatically, so any additional height above ground can be entered in when defining the flight with the 'Altitude' value seen in the panel. The corresponding 'Max Height' will be with relation to that 'Altitude' value. You can imagine that this would be useful when operating from/around a structure that is already some altitude above ground such as a large tower being surveyed or other critical infrastructure piece.

PATH FLIGHT

The second option for defining a flight is referred to as a 'Path Flight'. Switching to this type of flight will allow a user to define time-dependent waypoints for the flight plan. There are many options for how you may be able to define these through the API for programmers and integrators, but within the web portal a user must define a start time, avg velocity, and collection of waypoints.

Waypoints can be defined by clicking interactively within the 3D Globe panel. Consecutive clicks will create new waypoints in series, while hovering over existing waypoints and left-click/holding will allow you to move and adjust the waypoint individually. While moving existing waypoints or defining new waypoints, you may notice some areas of the flight path turn RED which signifies that there is a terrain intersection between those two connecting waypoints. You should consider moving the waypoints until there are only GREEN sections within the entire route. You can manually adjust the waypoint's altitude within the definition panel.



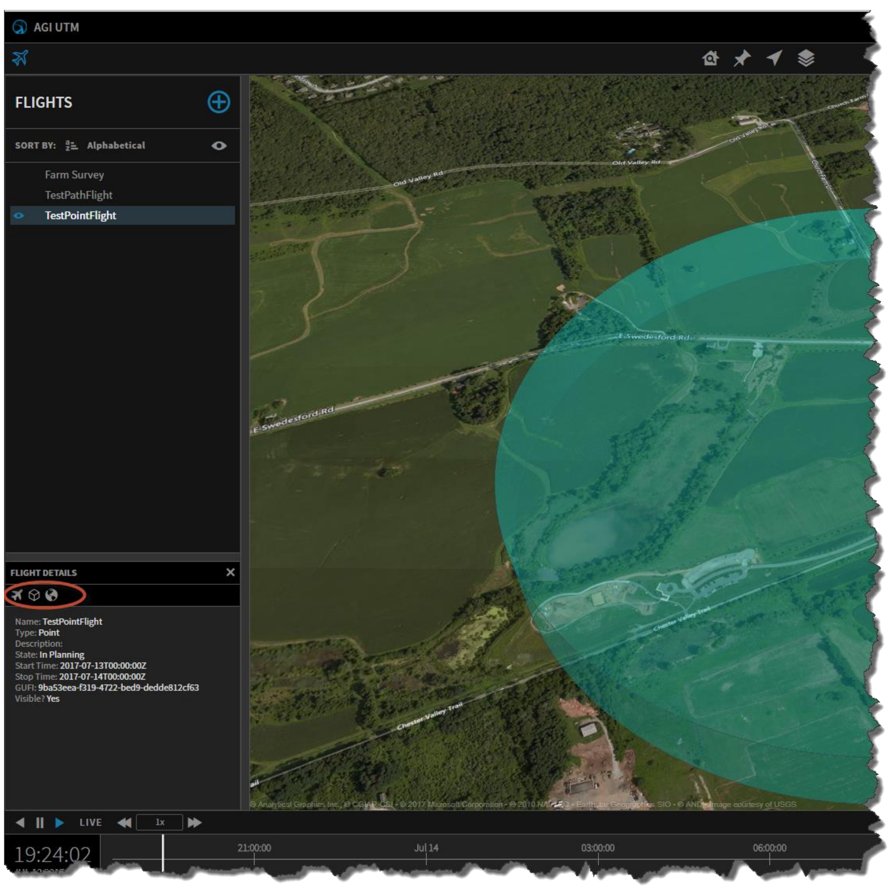
NOTE: Terrain Data is used interactively during this flight creation process which means as you move an individual waypoint whose altitude may have been defined as 100 ft AGL for example, the waypoint will remain at 100 ft above the terrain as you drag it around in the 3D Globe environment. You will also see the interconnecting route segments changing between Red/Green if positions are impacted by terrain while you move them.

OK, So I Have a Flight....Now What?

The UTM system is built on top of some of AGI's deep analytical heritage which means we are able to take advantage of an ever-evolving set of analytical capabilities pertaining to dynamic platforms. Since the entire system and back-end analytical engine is time-dynamic, we can easily evaluate time-based analytics against any object within the system.

Let's take a look at some of these analytical features!

When you have a flight, or a collection of flights, in the UTM system, these will be listed in the 'Flights' panel as seen here. You can click on any individual flight listed here to open some additional information about the flight, or double-click to focus on and see the flight plan in the 3D Globe panel. You will also notice the included flight analytics icons on the 'Analytics Ribbon':



Clicking these analysis icons will launch the various analytic results.

Navigation Accuracy – GPS Quality/Availability Predictions

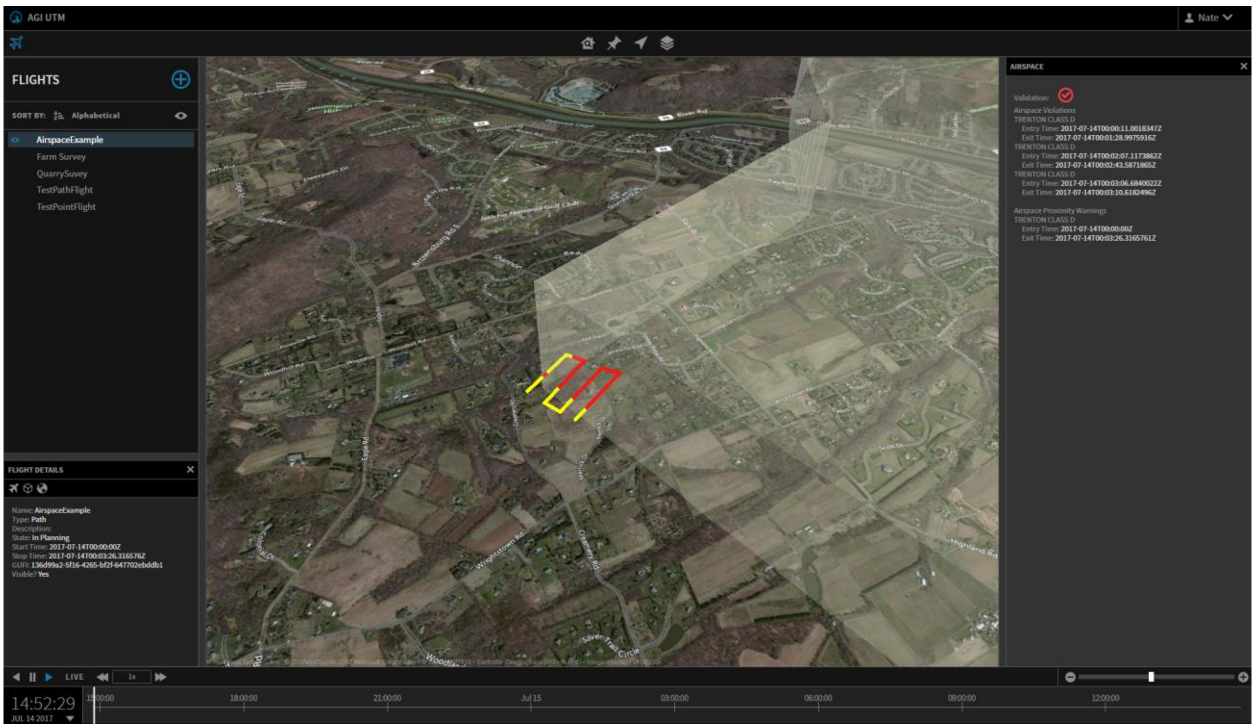
Let's look at the first icon which will show GPS results pertaining to the flight selected. This will show various metrics related to the predicted quality and availability of the GPS satellite constellation such as the anticipated number of satellites available as well as various Dilution of Precision (DOP) values. HDOP or 'Horizontal Dilution of Precision' is an important metric to consider since most unmanned systems rely heavily on the ability of the GPS solution onboard the aircraft to communicate accurate position information to the flight controller to support various flight modes. Many factors are at play when determining the accuracy of a GPS solution such as the positions of all the satellites with relation to the receiver, inherent clock errors/biases, individual satellite outages, etc. These are taken into consideration when producing the GPS/Navigation Accuracy predictions you see here:

These results are just a sample of the data that can also be returned through the developer API for this particular service. The UI seen here simply displays some of the analytical results that are returned and also gives some visual cues about the predicted "health" with a GREEN/YELLOW 'check mark' or RED 'x' icon indicating the overall status of HDOP in this case.

Airspace Intersections / Proximity

Another analytical flight safety check includes consideration of the national airspace. UAS operations are required to stay clear of specific airspaces in the U.S. as defined by the FAA's Part 107 regulations. As part of the adherence to those rules, a flight plan is checked for both intersection as well as proximity to specified airspace information. This includes Class B,C,D,E airspaces as well as Natl. Security Prohibited airspaces. Newer LAANC datasets are also being integrated into the system which will provide additional information and ultimately would tie into a quicker FAA authorization ability in the future.

To check any flight for Airspace violations, simply click the appropriate icon in the Analytics Ribbon for that particular flight and a panel with airspace intersection and proximity information will be revealed.

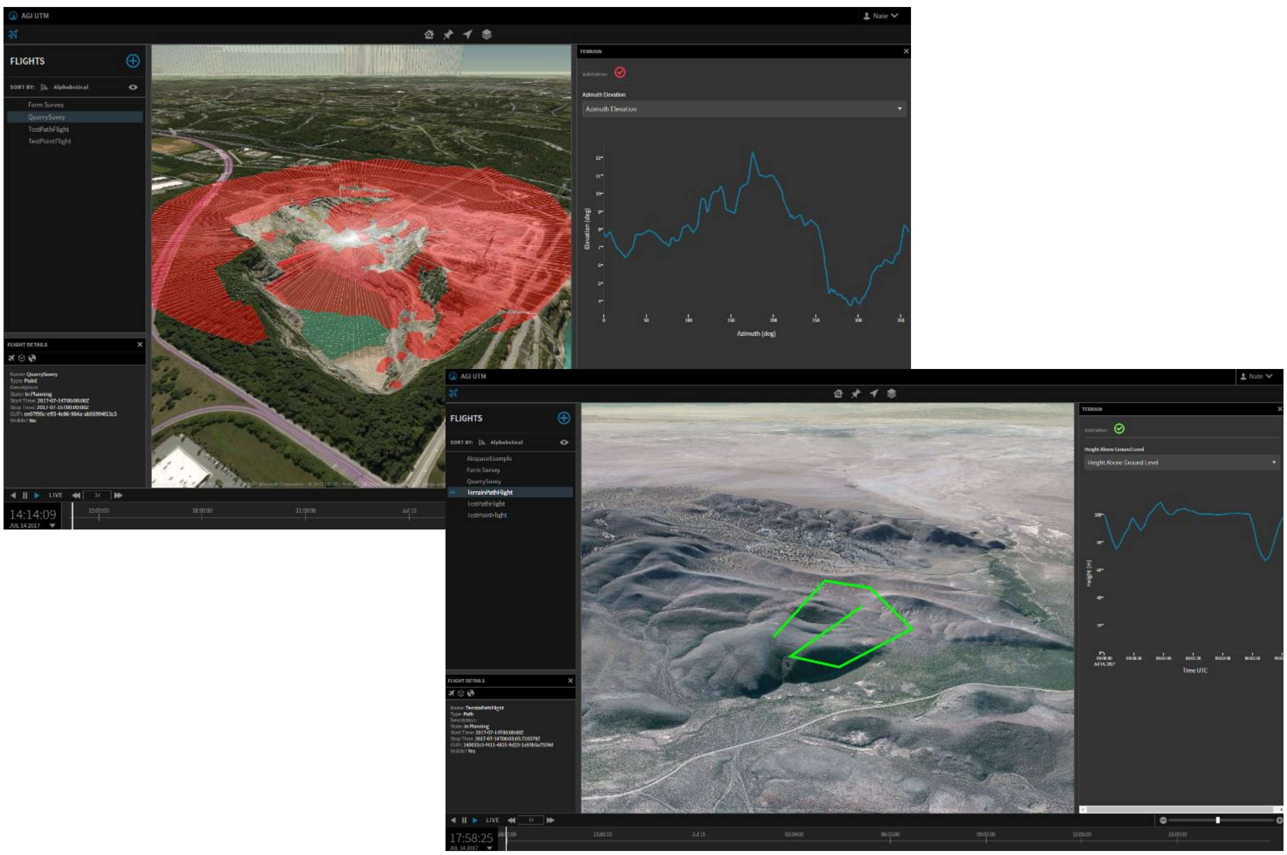


NOTE: The Airspace API is well documented and contains many additional features that may be of interest to integrators looking to be able to provide similar features within their own projects. Airspace data (currently) is sourced from the FAA published data sets for now and includes all airspaces within the Class B,C,D,E data (only airspaces which extend down to "surface" are visualized within the web browser for clarity, but analytics includes all airspace data. Proximity calculations can be further customized with threshold values supplied for both vertical and lateral distances to airspaces to support proximity warnings for airspaces which may be above a particular flight or position location.

Terrain Analysis & Line of Sight Considerations

Terrain information is used to analyze each flight in terms of line of sight considerations for Point Flight areas as well as Height Above Ground for Path Flights. This information is largely useful for maintaining a well-planned flight path in the presence of terrain as well as understand visual line of sight limitations from a command/control perspective.

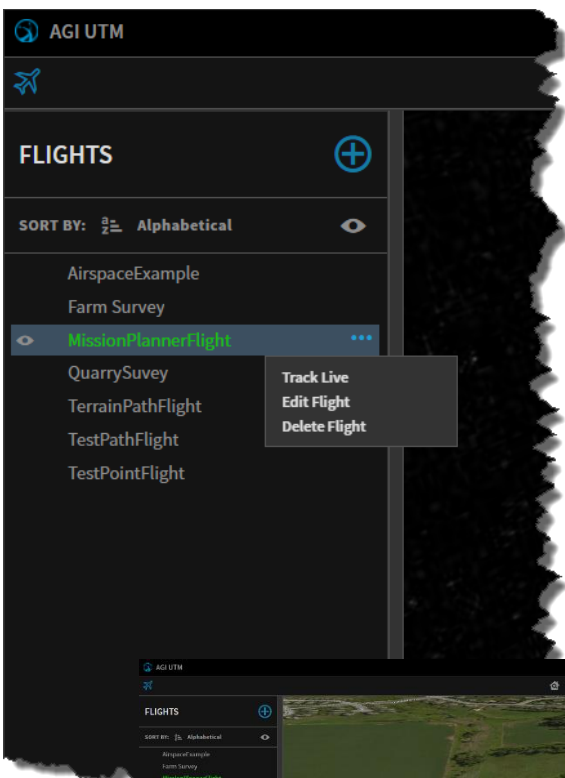
Clicking this analysis has different behaviors based on the type of flight plan, so feel free to experiment with this by looking at the analytics for Point Flights versus Path Flights. Both show interesting analytical results related to the consideration of the terrain data hosted in the UTM system.



Supporting RealTime Telemetry Data

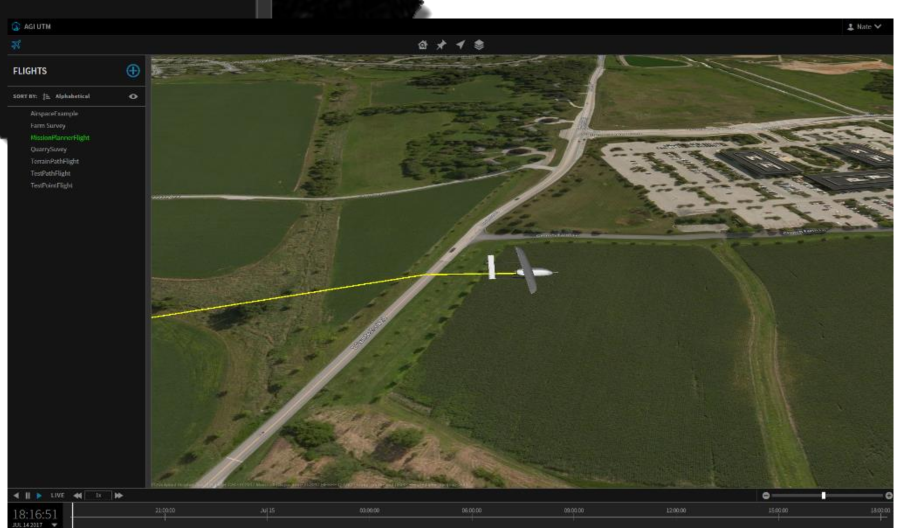
Another important feature of the UTM system is the ability to push telemetry position in during live flight operations. The API provides endpoints for easily scripting and integrating with common ground control station software or even custom written applications (more instruction and usage on how to do this in a later section).

Once realtime position information is being supplied to a flight within the UTM system, that particular flight will show up in the Flight Management panel as GREEN which allows for live tracking the flight by clicking the flight's options and selecting 'Track Live':



Once you click 'Track Live' during a realtime flight, the application will switch to a realtime clock mode and you will be actively monitoring all incoming position information being supplied to the UTM for that flight.

This will fix the camera view on the aircraft itself in the 3D Globe window and will follow the flight as new positions are fed into the system.



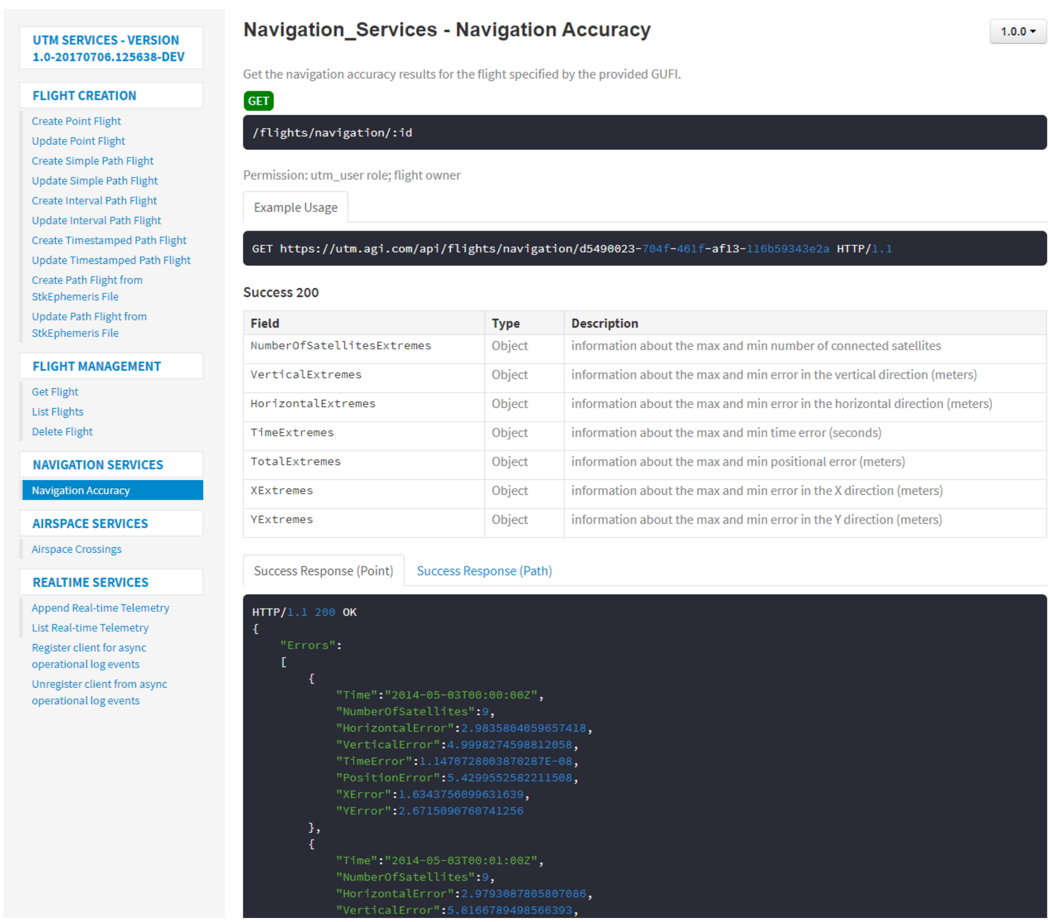
NOTE: In its current form, the UTM system provides the realtime visualization as seen here, but will soon add realtime analytics to provide alert/warning notifications with regard to various flight safety monitoring services you may wish to subscribe to.

Integration Details: Scripting & API Usage

The UTM systems provides a full back-end API for developers and integrators to take advantage of for various data interoperability needs and further automation or integration with other systems and applications. This section of the document will address common ways in which to interface with the UTM from outside via API integration and interactions.

Read The Manual!

For those interested in knowing all the available features of the UTM's API, the documentation can be accessed here (https://utm.agi.com/doc/). This web-based API documentation will outline all of the REST endpoints with examples of formatted requests and corresponding responses you can expect from the server.



Through this API, flight management procedures allow you to create flights, retrieve flights that belong to you, update flights, and of course delete flights. One important aspect of this process to notice is that each flight receives a Globally Unique Flight Identification number (GUFI) which is used to subsequently call most all of the other UTM API services such as checking a flight's Airspace Crossings/Proximity or Navigation Accuracy.

For example, once a flight has been entered into the system and assigned a corresponding GUFI number, that number should be used directly for service calls as seen here:

Airspace_Services - Airspace Crossings

1.0.0

Checks for intersections with class B, C, and D airspaces for the flight defined by the provided GUFI. For more information, visit the FAA site.

GET

```
/flights/airspaces/:id
```

Permission: utm_user role; flight owner

Example Usage

```
GET https://utm.agi.com/api/flights/airspaces/d5490023-704f-461f-af13-116b59343e2a?
horizontalBuffer=500&verticalBuffer=50
HTTP/1.1
```

Parameter

Field	Type	Description
horizontalBuffer optional	Double	An optional horizontal proximity threshold around the flight path or flight area to use while checking for possible airspace proximity violations (meters). Size range: >=0
verticalBuffer optional	Double	An optional vertical proximity threshold around the flight path or flight area to use while checking for possible airspace proximity violations (meters). Size range: >=0

Success 200

Field	Type	Description
N/A	AirspaceViolation[]	List of airspaces violated

AirspaceViolation

Field	Type	Description
AirspaceId	String	id of the violated airspace
Name	String	name of the violated airspace

Success Response

```
HTTP/1.1 200 OK
[
  {
    "AirspaceId": "ASHEVILLE CLASS C1",
    "Name": "ASHEVILLE CLASS C"
  },
```

This concept makes the usage of the included analytical services very easy to implement as well as future analytical services added to the system.

I'm Ready to Fly!

If you are simply interested in using the existing web-based GUI provided and would like to interface your aircraft with it, then you will probably be interested in the "Realtime Services" portion of the API. At its most basic form, the 'Append' function will allow the submission of position data in realtime. Anyone interested in using this API endpoint will benefit from understanding the implementation details for this particular POST endpoint. A quick-look at the documentation will show that the only information you necessarily need to provide is a time-stamped Lat/Lon/Alt data point!

If you have a script, for example, which interfaces with your Ground Control software, you would only need a GUFI for the flight you wish to send realtime position data against, and use the GUFI in the POST call you make to the service address such as:

https://utm.agi.com/api/flights/telemetry/append/d979098f-0ba9-4d48-992a-25b82c7b18bd

<u>USER TOKENS</u>

You will need to have a method for sending this information of course (such as a script, which most common Ground Control software packages provide such as Mission Planner which we will discuss as an implementation example in a moment), but another important piece is having access to the API from a security/authorized user standpoint which means you will need to supply a TOKEN identification number each time you make a call to the service. AGI can provide you with a valid token for your user account which you can use for these types of integration efforts.

Here is an example of a routine in Python for creating a "WebRequest" properly formatted for the AGI UTM:

```
def OneSkyRealTimeUpdate(uri, inputData, AccessToken):
    request = WebRequest.Create(uri)
    request.ContentType = "application/json"
    request.Method = "POST"
    request.Headers.Add("Authorization", "Bearer " + AccessToken);
    bytes = Encoding.ASCII.GetBytes(inputData)
    request.ContentLength = bytes.Length
    reqStream = request.GetRequestStream()
    reqStream.Write(bytes, 0, bytes.Length)
    reqStream.Close()

    response = request.GetResponse()
    result = StreamReader(response.GetResponseStream()).ReadToEnd()
    return result
```

Mission Planner Users

Probably one of the most common GCS software platforms for command/control and mission planning is Mission Planner which supports any APM-based flight controller. If you are using Mission Planner for your GCS solution, you can quickly take advantage of direct/in-line scripting from inside of Mission Planner:

This option utilized an embedded Python interpreter which allows you to write simple Python scripts which interface directly with data inside of Mission Planner as well as extend the full power and flexibility of Python as a scripting language!

We can easily use this feature to interface between the realtime data inside of Mission Planner and send the necessary data properly formatted to the AGI UTM.

One thing to understand is the ability to assign Mission Planner data to a script's variables for further data manipulation and integration, and fortunately Mission Planner makes this very simple. You only have to use the 'cs' namespace from within your script to have access to any data element that Mission Planner provides. For example, to access the current LATITUDE value from within your script, you would only need to declare something like this:

curLat = cs.lat

This same pattern exists for many data elements from within Mission Planner of course, so it is worth becoming familiar with the data elements, but if you are merely interested in basic position data, you could perform a query of the vehicles position from Mission Planner like this:

curLat = str(cs.lat)

curLon = str(cs.lng)

curAlt = float(cs.alt)

homeAlt = float(cs.HomeAlt)

AltASL = float(cs.altasl)

pressure = float(cs.press_abs)

etc…..

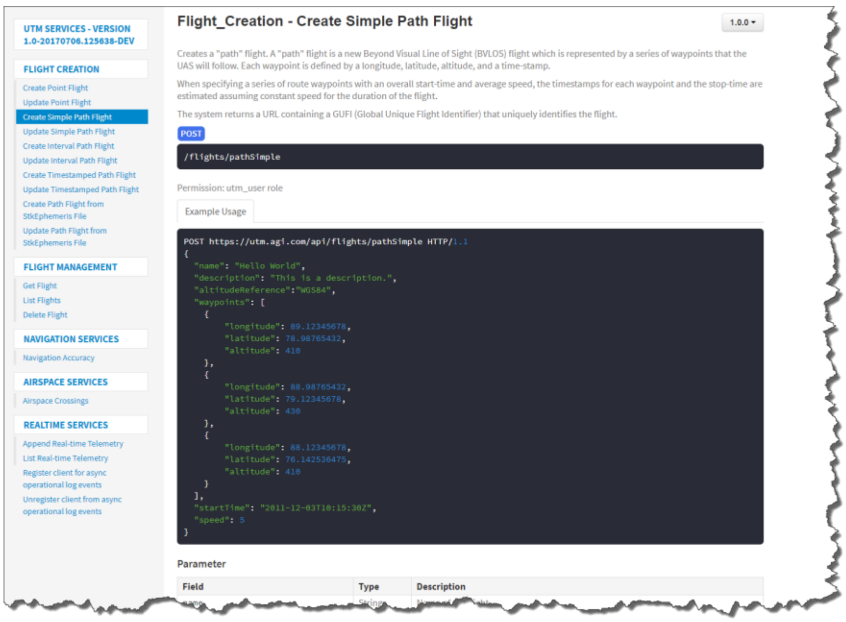
These values can be used within your script to create a POST to the AGI UTM, for example:

```
while (startRunning):
    curLat = str(cs.lat)
    curLon = str(cs.lng)
    curAlt = float(cs.alt)
    homeAlt = float(cs.HomeAlt)
    AltASL = float(cs.altasl)
    pressure = float(cs.press_abs)
    AltAdjust = 3.0
    finalAlt = str(AltASL - homeAlt - AltAdjust)
    curTime = datetime.datetime.utcnow().isoformat()
    curTimeZulu = curTime + 'Z'

    pressAlt = str(((1013.25/pressure)**(1/5.257)-1)*(15+273.15)/.0065)

    print 'Current Lat:' + curLat + ' Current Lon:' + curLon + ' Current Alt:' + finalAlt + ' Time Zulu:' + curTimeZulu

    ##Make REST call to OneSky API and push telemetry position data
    uri = "https://utm.agi.com/api/flights/telemetry/append/b3819c51-f7e8-4d68-b1fe-f8bf0ea999b8"

    ##TokenData
    AccessToken = [illegible]

    inputData= '{"eventType": "REALTIME_WAYPOINT","timestamp": "' + curTimeZulu + '","referenceLocation":{"longitude": ' +

    response = OneSkyRealTimeUpdate(uri, inputData, AccessToken)
    print response

    Script.Sleep(600)
```

This example Python script will be provided to any new user so that it can be customized and used as needed for their own usage with Mission Planner (or any other integration effort that may benefit from this example script).

Uploading Mission Planner Flight Plans

Mission Planner remains a very popular platform for creating autonomous flight plans for uploading to the drone. The simplicity of the flight plan makes it easy to send the necessary data to the AGI UTM. In looking at the API, you will notice that there are several options for sending "Path Flight" waypoints into the UTM to designate a flight plan. One of the simplest is the aptly named 'Simple Flight Path' type.



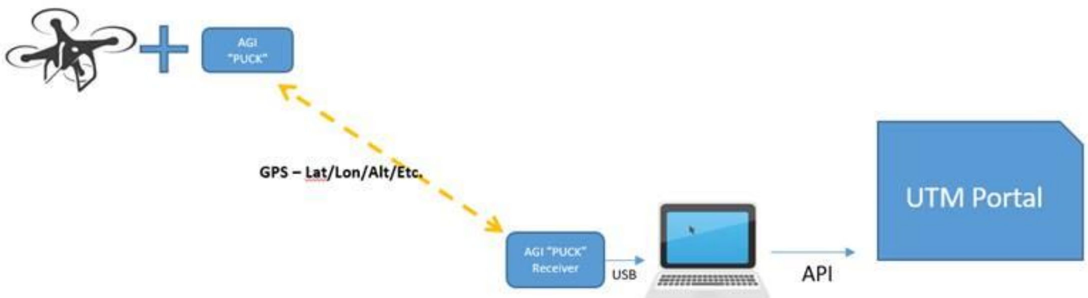
With this option, you simply declare a 'Start Time' for the flight, the corresponding waypoints with Latitude, Longitude, and Altitude, and an average speed at which the vehicle will traverse the waypoints.

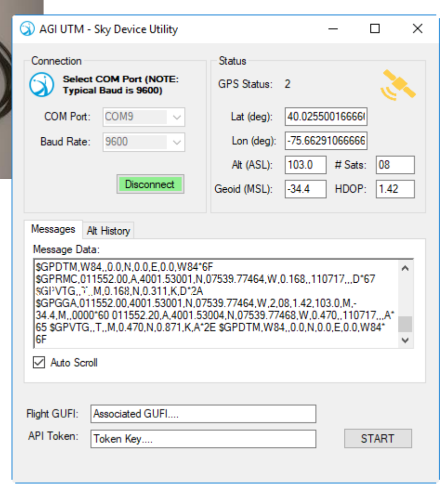
As with the previous Realtime example, there is also an example Python script available with this guide which will allow the exporting of a Mission Planner .waypoints file directly into the UTM. Feel free to explore that as a resource if interested.

NOTE: When supplying waypoints via the API, there will be options for altitude referenced to either WGS84 (ellipsoid) or Above Ground Level (AGL). When the AGL option is used, the altitudes will be referenced to the UTM's back-end terrain data. Since Mission Planner altitudes are typically chosen "relative" to where the drone is first armed, this is a convenient way to supply those altitude directly to the UTM and let the system's handle ground referencing and conversions to WGS84 or any other datum as needed.

Hardware Options – AGI SkyDevice

In some cases, there may not be an easy option to integrate via software with the platform you are using, and if that is the case then an easy solution will be to fix a small device on the aircraft which can be used to provide position/tracking information directly into the AGI UTM.





This approach will allow you to fix the remote device to your aircraft and use the ground receiver to interface with the AGI UTM. This requires a laptop and the SkyDevice utility which allows you to designate the flight GUFI and your API Token and begin pushing the realtime position data into the UTM system.

If you are interested in utilizing this sort of approach for supporting flight testing of aircraft that are not otherwise easy to integrate from a software/GCS perspective, then please let us know and we can work on getting a set of small devices for you to use. Simply emailing the UTM team (utm@agi.com) is the best way to contact us to talk about further needs.