InsightCloud

API/Developer Documentation

Version 0.4 Beta

June 09, 2015

Contents

Revision Table

Version Number	*Date*	*Description*	*Authors*
0.0 Beta	08/15/2014	First Draft for initial release	DigitalGlobe
0.1 Beta	02/11/2015	Second Draft updated to include new features and updated widgets.	DigitalGlobe
0.2 Beta	04/21/2015	Third Draft updated to include new features and updated widgets.	DigitalGlobe
0.3 Beta	05/27/2015	Fourth Draft updated to include new features and updated widgets.	DigitalGlobe
0.4 Beta	06/09/2015	Fifth Draft updated to include new features and updated widgets.	DigitalGlobe

1 Scope

1.1 Identification

This document applies to InsightCloud Version 0.4 Beta Dated 06/09/2015.

1.2 System Overview

The DigitalGlobe InsightCloud combines an open data processing environment with powerful geospatial analytic algorithms and easy to use tools. The result is that our customers are able to cost effectively exploit massive amounts of location data in near real-time. The InsightCloud abstracts data management, application logic, and the user experience. Our customers can fuse DigitalGlobe Geospatial Big Data™ with their native and 3rd party sources. Experts can design custom Map Algebra algorithms within our framework and our Open APIs enable our analytics to integrate into end user workflows.

Insight Explorer	A portfolio of "easy button" analytic applications that enable interactive mission planning and collaborative mapping at global scale
GBD Analytic Toolkit	A portfolio of geospatial analytic tools that enable interactive analysis of massive volumes of raster and vector data to enable mission planning, predictive modeling, and real-time situational awareness. The GBD Analytic Toolkit leverages the core algorithms of Signature Analyst™ a patented tool that discover unseen patterns in spatial data.
Geospatial Data Hub	A big data processing and exploitation environment built on the Cloudera Enterprise Data Hub that has been adapted to process and exploit high resolution imagery, terrain, map features, human geography, movement tracks, and social media at global scale.

InsightCloud can be delivered via an On-Premise or Software as a Service (SaaS) model. Our On-Premise solution operates within the emerging IC ITE and DCGS Architectures. Our SaaS model runs in the Amazon Cloud and our Classified Datacenter in Herndon, VA.

1.3 Document Overview

The purpose of this document is to:

- Identify the REST service calls made by the software.

2 Reference Documents

Document Name	Ver	Location
InsightCloud General Users Guide (GUG)	0.0 Beta	

3 CAS Authentication

The following is a sample Groovy script that walks through the authentication and access steps for users who want to access our data via direct calls through REST services. Before running the script, users may need to import the insightcloud.digitalglobe.der cert into the JRE cacerts file by downloading the cert and running this command:

```
sudo keytool -import -trustcacerts -alias insightcloud.digitalglobe.der -file
/home/<user>/insightcloud.digitalglobe.der -keystore /usr/lib/jvm/java-7-
oracle/jre/lib/security/cacerts
```

The sample Groovy script:

```
import java.io.IOException
import java.util.logging.Logger
import java.util.regex.Matcher
import java.util.regex.Pattern
import org.apache.commons.httpclient.HttpClient
import org.apache.commons.httpclient.NameValuePair
import org.apache.commons.httpclient.methods.PostMethod
import org.apache.commons.httpclient.methods.GetMethod
import org.apache.commons.httpclient.methods.DeleteMethod


@Grab("commons-httpclient:commons-httpclient:3.1")

class Client
{
  static final Logger LOG = Logger.getLogger(Client.class.getName())
  String getServiceTicket(String server, String ticketGrantingTicket, String service)
  {
    if (!ticketGrantingTicket)
      return null
    HttpClient client = new HttpClient()
    PostMethod post = new PostMethod("$server/$ticketGrantingTicket")
    post.setRequestBody([new NameValuePair("service", service)].toArray(new
NameValuePair[1]))
    try
    {
      client.executeMethod(post)
      String response = post.getResponseBodyAsString()
      switch (post.getStatusCode())
      {
        case 200:
          return response
        default:
          LOG.warning("Invalid response code ( $post.getStatusCode() ) from CAS server!")
          LOG.info("Response (1k): " + response.substring(0, Math.min(1024, response.length())))
          break
      }
    }
    catch (final IOException e)
    {
      LOG.warning(e.getMessage())
    }
    finally
    {
      post.releaseConnection()
    }
    return null
  }
  String getTicketGrantingTicket(String server, String username, String password)
  {
    HttpClient client = new HttpClient()
    PostMethod post = new PostMethod(server)
    post.setRequestBody([new NameValuePair("username", username),new
NameValuePair("password", password)].toArray(new NameValuePair[2]))
    try
    {
      client.executeMethod(post)
      String response = post.getResponseBodyAsString()
      switch (post.getStatusCode())
      {
        case 201:
          Matcher matcher = Pattern.compile(".*action=\".*/(.*?)\".*").matcher(response)
          if (matcher.matches())
            return matcher.group(1)
          LOG.warning("Successful ticket granting request, but no ticket found!")
          LOG.info("Response (1k): " + response.substring(0, Math.min(1024, response.length())))
          break
        default:
          LOG.warning("Invalid response code (${post.getStatusCode()}) from CAS server!")
          LOG.info("Response: $response")
          break
      }
    }
    catch (final IOException e)
    {
      LOG.warning(e.getMessage())
    }
    finally
  {
   post.releaseConnection()
  }
  return null
 }
 void notNull(Object object, String message)
 {
  if (object == null)
   throw new IllegalArgumentException(message)
 }
 void getServiceCall(HttpClient client, String service, String serviceTicket) {
   GetMethod method = new GetMethod(service)
   method.setQueryString([new NameValuePair("ticket", serviceTicket)].toArray(new
NameValuePair[1]))
   try
   {
    client.executeMethod(method)
    String response = method.getResponseBodyAsString()
    switch (method.getStatusCode())
    {
     case 200:
      LOG.info("Response: $response")
      break
     default:
      LOG.warning("Invalid response code (" + method.getStatusCode() + ") from CAS
server!")
      LOG.info("Response: $response")
      break
    }
   }
   catch (final IOException e)
   {
    LOG.warning(e.getMessage())
   }
   finally
   {
     method.releaseConnection()
   }
 }
 void logout(String server, String ticketGrantingTicket) {
   HttpClient client = new HttpClient()
   DeleteMethod method = new DeleteMethod("$server/$ticketGrantingTicket")
   try
   {
    client.executeMethod(method)
    switch (method.getStatusCode())
    {
     case 200:
      LOG.info("Logged out")
      break
     default:
      LOG.warning("Invalid response code (" + method.getStatusCode() + ") from CAS
server!")
```

```
      LOG.info("Response: $response")
      break
    }
   }
   catch (final IOException e)
   {
    LOG.warning(e.getMessage())
   }
   finally
   {
     method.releaseConnection()
   }
 }

 public static void main(String[] args)
 {

    String server = "https://insightcloud.digitalglobe.com/cas/v1/tickets"
    String username = "user.name"
    String password = "user.password"
    String authEndpoint = "https://insightcloud.digitalglobe.com/monocle-
3/j_spring_cas_security_check"
    String apiEndpoint = "https://insightcloud.digitalglobe.com/monocle-
3/app/broker/mrgeo/Comms/datasources"

    Client client = new Client()

    // get a ticket-granting ticket from CAS
    String ticketGrantingTicket = client.getTicketGrantingTicket(server, username, password)
    println "TicketGrantingTicket is $ticketGrantingTicket"

    // get a service ticket to access the app
    String serviceTicket = client.getServiceTicket(server, ticketGrantingTicket, authEndpoint)
    println "ServiceTicket is $serviceTicket"

    // CAS puts something in the client's store, so must reuse the client
    HttpClient http = new HttpClient()

    // authenticate with the ticket
    client.getServiceCall(http, authEndpoint, serviceTicket)

    // actually make the API call we want
    client.getServiceCall(http, apiEndpoint, serviceTicket)

    // logout from CAS
    client.logout(server, ticketGrantingTicket)
 }
}
```

4 External REST calls

The following are examples of the interaction between the UI and the external services REST calls that are triggered by each widget.

4.1 Terrain

4.1.1 Comms

The Comms widget provides the analyst with a visual output of radio frequency propagation using a set of criteria and an available DEM. It calculates radio frequency propagation loss against surrounding terrain based on Longley-Rice Irregular Terrain Model algorithm for radio frequency propagation loss. RF propagation result is filtered based on a supplied maximum path loss. The end result for each cell is the number of cells within a radius of interest below or equal to the maximum path loss.

4.1.1.1 Parameters

Parameters	Type	Default	Optional	Multiple	Description
BBOX	Float	<none>	False	False	Coordinates in decimal degrees, separated by commas, of the bounding box corners. In order, these coordinates are Lower Left Lon, Lower Left Lat, Upper Right Lon, Upper Right Lat; bbox always surrounds the origin point. BBOX=49.122,14.554,49.134,14.565 for example
Datasource	String	ASTER1	True	False	The analytic to operate on such as ASTER1
frequency	Float	300.0	False	False	The frequency of radio signal in MHz (20 MHz to 20 GHz)
outerRadius	Float	3000.0	False	False	The outer radius of the source transmission to consider for each cell (meters).
transmitterHeight	Float	10.0	False	False	The height of the transmitting antenna above the terrain

Parameters	Type	Default	Optional	Multiple	Description
					(meters).
transmitterCoord	Point2	<none>	True	True	An x/y coordinate pair for a transmitting tower's location. Repeat parameter for multiple transmitters. Can be substituted with inputPoints1 data source
receiverHeight	Float	10.0	False	False	The height of the receiving antenna above the terrain (meters).
confidence	Float	0.5	False	False	A measure used to describe the expected number of measurements be a given signal strength from multiple locations at a time t (0.0 to 1.0)
earthConductivity	Float	0.005	False	False	The Earth's conductivity for the terrain of the region of interest (Siemens per meter)
earthDialectric	Float	15.0	False	False	The Earth's dielectric constant for the terrain of the region of interest (Relative permittivity)
polarity	Integer	0	False	False	The polarity of the radio signal (0 = horizontal and 1 = vertical).
reliability	Float	0.5	False	False	A measure used to describe the expected percentage over a period of time at a target location the signal will be at least a given signal strength (0.0 to 1.0)

Parameters	Type	Default	Optional	Multiple	Description
radioClimate	Integer	5	False	False	A description of the environment of the region of interest (1 = Equatorial, 2 = Continental Subtropical, 3 = Maritime Subtropical, 4 = Desert, 5 = Continental Temperate, 6 = Maritime Temperate, Over Land, 7 = Maritime Temperate, Over Sea).
surfaceRefractivity	Float	301.0	False	False	The Atmospheric Bending Constant (N-Units).
colorType	String	COLORSCALE	True	False	Determines the type of color scheme (one of COLOR, COLORSCALE, COLORMAP)
colorString	String	204:215+48+39+148,153:252+141+89+148,102:254+224+139+148,51:145+207+96+148,0:26+152+80+148	True	False	Color values separated by '+' 1:255+0+0+148 for example
fileType	String	KMZ	True	False	The filetpye of the return, PNG, GeoTIFF, and KMZ are supported; DOWNLOAD only
fileName	String	<none>	True	False	Resulting filename; DOWNLOAD only

Table 4.1 Comms RFP Parameters

Origin Array String (transmitterCoord)

Origin Array String must be built in one of two ways. With only one Orgin point "ORIGINPOINT:-117.96122799195+38.019902697754" or with multiple points*

"ORIGINPOINT:-117.96122799195+38.019902697754,originPoint:-117.97152767457+38.044965258789,originPoint:-117.9198576001+38.027627459717,originPoint:-117.97959575928+38.033807269287"

**User beware*: Comms RFP multiple point Origin Array String has little testing; issues may be present.

Parameters	Limits
Longitude	Between -180 and 180 degrees inclusive
Latitude	Between -90 and 90 degrees inclusive
Datasource	One of: ASTER1, SRTM3, VRICON2M
OuterRadius	Between 1 and 50000 meters inclusive
TransmitterHeight	Between 0 and 5000 meters inclusive
ReceiverHeight	Between 0 and 5000 meters inclusive
Confidence	Between 0 and 1 inclusive
Reliability	Between 0 and 1 inclusive
Polarity	One of '0', '1' where 0=horizontal and 1=vertical
FrequencyMhz	Between 1 and 20000 inclusive
FrequencyGhz	Between 0.001 and 20 inclusive
outputType	One of 'MAX', 'SUM' (may be any case)
fileType	One of 'PNG', 'GEOTIFF', 'KMZ' (may be any case)

Table 4.2 Comms RFP Parameter Limits

4.1.1.2 Steps for REST View

1. Submit a GET request to view a Comms RFP analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/Comms;REQUEST=Execute;DataSource={source};BBOX={LLLon,LLLat,URLon,URLat};Operation=LegionRfPropagationOperation;ColorType={type};ColorString={string};originPoint={Lon+Lat};outerRadius={value};transmitterHeight={value};receiverHeight={value};confidence={value};earthConductivity={value};earthDialectric={value};polarity={value};reliability={value};radioClimate={value};surfaceRefractivity={value};frequency={value}
```

4.1.1.3 Steps for REST Download

1. Submit a GET request to download a Comms RFP analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/Comms;REQUEST=Execute;DataSource={source};BBOX={LLLon,LLLat,URLon,URLat};Operation=LegionRfPropagationOperation;ColorType={type};ColorString={string};originPoint={Lon+Lat};outerRadius={value};transmitterHeight={value};receiverHeight={value};confidence={value};earthConductivity={value};earthDialectric={value};polarity={value};reliability={value};radioClimate={value};surfaceRefractivity={value};frequency={value};FileType={type};FileName={string}
```

4.1.2 Cumulative Viewshed (Cumulative VS)

The Cumulative Viewshed analytic computes a viewshed within a given area of interest from a specified observation point and a search radius. The resulting image creates a Cumulative Exposure Grid (CEG) where each pixel represents the percentage of the surrounding area that is viewable within the specified search radius.

4.1.2.1 Parameters

Parameters	Type	Default	Optional	Multiple	Description
Left	Float	<none>	False	False	Longitude in decimal degrees of the lower left bounding box corner
Bottom	Float	<none>	False	False	Latitude in decimal degrees of the lower left bounding box corner
Right	Float	<none>	False	False	Longitude in decimal degrees of the upper right bounding box corner
Top	Float	<none>	False	False	Latitude in decimal degrees of the upper right bounding box corner
Datasource	String	ASTER1	True	False	The analytic to operate on such as ASTER1
ObserverHeight	Float	2	True	False	Height of the observer's view above the current elevation (meters)

Parameters	Type	Default	Optional	Multiple	Description
TargetHeight	Float	0.01	True	False	Maximum height of the target above the elevation of its location (meters)
InnerRadius	Float	0	True	False	Inner radius of region around observer cell to compute the viewshed (meters)
OuterRadius	Float	10000	True	False	Outer radius of region around observer cell to compute the viewshed (meters)
UpperVertAngle	Float	90	True	False	Upper angle of the observer's view above the horizon (degrees)
LowerVertAngle	Float	-90	True	False	Lower angle of the observer's view below the horizon (degrees)
sAzimuth	Float	0	True	False	Beginning azimuth angle of the observer's view in the horizontal plane with north equal to 0 (degrees)
eAzimuth	Float	360	True	False	Ending azimuth angle of the observer's view in the horizontal plane with north equal to 0 (degrees)
ColorType	String	COLORSCALE	True	False	Determines the type of color scheme (one of COLOR, COLORSCALE, COLORMAP)
Color	String	1:215+48+39+148,2:252+141+89+148,3:254+224+139+148, 4:217+239+139+148,5:145+207+96+148,6:26+152+80+148	True	False	Color values separated by '+' 1:255+0+0+148 for example

Parameters	Type	Default	Optional	Multiple	Description
fileType	String	KMZ	True	False	The filetpye of the return, PNG, GeoTIFF, and KMZ are supported; DOWNLOAD only
fileName	String	<none>	True	False	Resulting filename; DOWNLOAD only
normalize	String	RADIUS	True	False	Optional parameter to normalize results (RADIUS, or MINMAX). If not set then results are defaulted to RADIUS.
normalizeScaleValue	String	100	True	False	Optional parameter as the maximum of the linear scale range after normalizing.

Table 4.3 Cumulative Viewshed Parameters

Parameters	Limits
Left	Between -180 and 180 degrees inclusive
Right	Between -180 and 180 degrees inclusive
Bottom	Between -90 and 90 degrees inclusive
Top	Between -90 and 90 degrees inclusive
Datasource	One of: ASTER1, SRTM3, VRICON2M
ObHeight	Between 0.01 and 20000 meters inclusive
TargHeight	Between 0.01 and 20000 meters inclusive
InnerRadius	Between 0 and 50000 meters inclusive and ALSO less than OuterRadius
OuterRadius	Between 1 and 50000 meters inclusive
UVA	Between -90 and 90 degrees inclusive
LVA	Between -90 and 90 degress inclusive and ALSO is less than UpperVertAngle
sAzimuth	Between 0 and 360 degrees and ALSO is less than eAzimuth
eAzimuth	Between 0 and 360 degrees
normalize	One of 'RADIUS', 'MINMAX' (may be any case)

Parameters	Limits
normalizeScaleValue	Between 1.0 and 1000 inclusive
fileType	One of 'PNG', 'GEOTIFF', 'KMZ' (may be any case)

Table 4.4 Cumulative Viewshed Parameter Limits

4.1.2.2 Steps for REST View

1. Submit a GET request to view a Cumulative Viewshed analytic. The REST call via monocle-3 application will look similar to the following:

    https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/massvs;DataSource={source};Left={value};Bottom={value};Right={value};Top={value};ObHeight={value};TargHeight={value};InnerRadius={value};LVA={value};UVA={value};sAzimuth={value};eAzimuth={value};colorType={type};color={string};normalize={string};normalizeScaleValue={value};OuterRadius={value}

4.1.2.3 Steps for REST Download

1. Submit a GET request to download a Cumulative Viewshed analytic. The REST call via monocle-3 application will look similar to the following:

    https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/massvs;DataSource={source};Left={value};Bottom={value};Right={value};Top={value};ObHeight={value};TargHeight={value};InnerRadius={value};LVA={value};UVA={value};sAzimuth={value};eAzimuth={value};colorType={type};color={string};normalize={string};normalizeScaleValue={value};OuterRadius={value};fileType={type};fileName={string}

4.1.3 Sightline

The SightLine analytic provides the analyst with a visual line of sight output for any origin point and a given elevation model. This ability to quickly identify areas of high/low visibility provides a logistic advantage to analysts who need to ensure the success of a route where concealment or exposure becomes an issue.

4.1.3.1 Parameters

Parameters	Type	Default	Optional	Multiple	Description
Left	Float	<none>	False	False	Longitude in decimal degrees of the lower left bounding box corner

Parameters	Type	Default	Optional	Multiple	Description
Bottom	Float	<none>	False	False	Latitude in decimal degrees of the lower left bounding box corner
Right	Float	<none>	False	False	Longitude in decimal degrees of the upper right bounding box corner
Top	Float	<none>	False	False	Latitude in decimal degrees of the upper right bounding box corner
OrgArrayString	String	<none>	False	False	Can be built in one of two ways. With only one Origin point (See below)
Datasource	String	ASTER1	True	False	The analytic to operate on such as ASTER1
ObserverHeight	Float	2	True	False	Height of the observer's view above the current elevation (meters)
TargetHeight	Float	0.01	True	False	Maximum height of the target above the elevation of its location (meters)
InnerRadius	Float	0	True	False	Inner radius of region around observer cell to compute the viewshed (meters)
OuterRadius	Float	10000	True	False	Outer radius of region around observer cell to compute the viewshed (meters)
UpperVertAngle	Float	90	True	False	Upper angle of the observer's view above the horizon (degrees)
LowerVertAngle	Float	-90	True	False	Lower angle of the observer's view below the horizon (degrees)
sAzimuth	Float	0	True	False	Beginning azimuth angle of the observer's view in the horizontal plane with north equal to 0 (degrees)

Parameters	Type	Default	Optional	Multiple	Description
eAzimuth	Float	360	True	False	Ending azimuth angle of the observer's view in the horizontal plane with north equal to 0 (degrees)
outputType	String	MAX	True	False	Defines how viewsheds from multiple sources are aggregated. The default MAX simply overlaps the viewsheds while SUM will add 1 to an output pixel for each overlap.
colorType	String	COLORMAP	True	False	Determines the type of color scheme (one of COLOR, COLORSCALE, COLORMAP)
color	String	1:0+255+0+148, 0:255+0+0+148	True	False	Color values separated by '+' 1:255+0+0+148 for example
fileType	String	KMZ	True	False	The filetpye of the return, PNG, GeoTIFF, and KMZ are supported; DOWNLOAD only
fileName	String	<none>	True	False	Resulting filename; DOWNLOAD only

Table 4.5 Sightline Parameters

Origin Array String

Origin Array String must be built in one of two ways. With only one Orgin point "ORIGINPOINT:-117.96122799195+38.019902697754" or with multiple points* "ORIGINPOINT:-117.96122799195+38.019902697754,originPoint:-117.97152767457+38.044965258789,originPoint:-117.9198576001+38.027627459717,originPoint:-117.97959575928+38.033807269287"

**User beware*: Sightline multiple point Origin Array String has little testing; issues may be present.

Parameters	Limits
Left	Between -180 and 180 degrees inclusive

Parameters	Limits
Right	Between -180 and 180 degrees inclusive
Bottom	Between -90 and 90 degrees inclusive
Top	Between -90 and 90 degrees inclusive
Datasource	One of: ASTER1, SRTM3, VRICON2M
ObserverHeight	Between 0.01 and 20000 meters inclusive
TargetHeight	Between 0.01 and 20000 meters inclusive
InnerRadius	Between 0 and 50000 meters inclusive and ALSO less than OuterRadius
OuterRadius	Between 1 and 50000 meters inclusive
UpperVertAngle	Between -90 and 90 degrees inclusive
LowerVertAngle	Between -90 and 90 degress inclusive and ALSO is less than UpperVertAngle
sAzimuth	Between 0 and 360 degrees and ALSO is less than eAzimuth
eAzimuth	Between 0 and 360 degrees
outputType	One of 'MAX', 'SUM' (may be any case)
fileType	One of 'PNG', 'GEOTIFF', 'KMZ' (may be any case)

Table 4.6 Sightline Parameter Limits

4.1.3.2 Steps for REST View

1. Submit a GET request to view a Sightline analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/sightline;
DataSource={source};Left={value};Bottom={value};Top={value};Right={value};OrgArr
ayString=ORIGINPOINT:{Lon+Lat};ObHeight={value};TargHeight={value};OuterRadius
={value};LVA={value};UVA={value};sAzimuth={value};eAzimuth={value};outputType=
{string};colorType={type};color={string}
```

4.1.3.3 Steps for REST Download

1. Submit a GET request to download a Sightline analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/sightline;
DataSource={source};Left={value};Bottom={value};Top={value};Right={value};OrgArr
ayString=ORIGINPOINT:{Lon+Lat};ObHeight={value};TargHeight={value};OuterRadius
```

={value};LVA={value};UVA={value};sAzimuth={value};eAzimuth={value};outputType={string};colorType={type};color={string};fileType={type};fileName={string}

4.1.4 Sitescan

The Sitescan analytic computes the Topographic Position Index (TPI). The TPI algorithm compares each target raster cell and an annular neighborhood defined between the inner and outer radii, and performs a basic landform classification to isolate the Ridge, Upper Slope, Middle Slope, Flat Slope, Lower Slope and Valley classes.

4.1.4.1 Parameters

Parameters	Type	Default	Optional	Multiple	Description
Left	Float	<none>	False	False	Longitude in decimal degrees of the lower left bounding box corner
Bottom	Float	<none>	False	False	Latitude in decimal degrees of the lower left bounding box corner
Right	Float	<none>	False	False	Longitude in decimal degrees of the upper right bounding box corner
Top	Float	<none>	False	False	Latitude in decimal degrees of the upper right bounding box corner
Datasource	String	ASTER1	True	False	The analytic to operate on such as ASTER1
InnerRadius	Float	0	True	False	The inner radius of the annular neighborhood around each cell (meters).
outerRadius	Float	1000	True	False	The outer radius of the annular neighborhood around each cell (meters).
colorType	String	COLORMAP	True	False	Determines the type of color scheme (one of COLOR,

Parameters	Type	Default	Optional	Multiple	Description
					COLORSCALE, COLORMAP)
Color	String	1:215+48+39+148,2:252+141+89+148,3:254+224+139+148, 4:217+239+139+148,5:145+207+96+148,6:26+152+80+148	True	False	Color values separated by '+' 1:255+0+0+148 for example
fileType	String	KMZ	True	False	The filetpye of the return, PNG, GeoTIFF, and KMZ are supported; DOWNLOAD only
fileName	String	<none>	True	False	Resulting filename; DOWNLOAD only

Table 4.7 Sitescan Parameters

Parameters	Limits
Left	Between -180 and 180 degrees inclusive
Right	Between -180 and 180 degrees inclusive
Bottom	Between -90 and 90 degrees inclusive
Top	Between -90 and 90 degrees inclusive
DataSource	One of: ASTER1, SRTM3
OuterRadius	Between 1 and 25000 inclusive
InnerRadius	Between 0 and 25000 inclusive and ALSO less than OuterRadius
fileType	One of 'PNG', 'GEOTIFF', 'KMZ' (may be any case)

Table 4.8 Sitescan Parameter Limits

4.1.4.2 Steps for REST View

1. Submit a GET request to view a Cumulative Viewshed analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/sitescan;DataSource={source};Left={value};Bottom={value};Right={value};Top={value};ColorType={type};Color={string};OuterRadius={value};InnerRadius={value}
```

4.1.4.3 Steps for REST Download

1. Submit a GET request to download a Sitescan analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/sitescan;DataSource={source};Left={value};Bottom={value};Right={value};Top={value};ColorType={type};Color={string};OuterRadius={value};InnerRadius={value};FileType={type};FileName={string}
```

4.1.5 Touchdown (HLZ)

The Touchdown (HLZ) identifies suitable landing pad locations for different helicopters based on an origin location (dropped on the map in the viewport), LiDAR/DEM elevation source, Helicopter Type, Time of Day and Environmental Conditions.

4.1.5.1 Parameters

Parameters	Type	Default	Optional	Multiple	Description
Left	Float	<none>	False	False	Longitude in decimal degrees of the lower left bounding box corner
Bottom	Float	<none>	False	False	Latitude in decimal degrees of the lower left bounding box corner
Right	Float	<none>	False	False	Longitude in decimal degrees of the upper right bounding box corner
Top	Float	<none>	False	False	Latitude in decimal degrees of the upper right bounding box corner
Datasource	String	<none>	False	False	The elevation dataset to operate on such as BUCKEYE
Color	String	0,255,0	True	False	RGB color to render the HLZ

Parameters	Type	Default	Optional	Multiple	Description
HlzDiameter	Float	25	True	False	Diameter of the HLZ touchdown area (meters)
MaxHlzSlope	Float	7	True	False	Maximum slope of the HLZ touchdown area (percent)
ObstacleRadius	Float	125	True	False	Radius of obstacle avoidance (meters)
ApproachAngle	Float	10	True	False	Angle of approach to the HLZ
HoverHeight	Float	0	True	False	Hover height above HLZ (meters)
fileType	String	KMZ	True	False	The filetpye of the return, PNG, GeoTIFF, and KMZ are supported; DOWNLOAD only
fileName	String	<none>	True	False	Resulting filename; DOWNLOAD only

Table 4.9 HLZ/Touchdown Parameters

Parameters	Limits
Left	Between -180 and 180 degrees inclusive
Right	Between -180 and 180 degrees inclusive
Bottom	Between -90 and 90 degrees inclusive
Top	Between -90 and 90 degrees inclusive
Datasource	One of: VRICON2m
hlzDiameter	Between 0 and 250 meters inclusive
maxHlzSlope	Between 0 and 90 degrees inclusive
obstacleRadius	Between 0 and 500 meters inclusive
approachAngle	Between 0 and 90 degrees inclusive
hoverHeight	Between 0 and 100 meters inclusive
fileType	One of 'PNG', 'GEOTIFF', 'KMZ' (may be any case)

Table 4.10 HLZ/Touchdown Parameter Limits

There are three pseudo-parameters in Touchdown (HLZ) that are used in the UI to quickly set the parameters for HLZ Diameter, Maximum HLZ Slope, Obstacle Radius, Approach

Angle, and Hover Height. These pseudo-parameters are not involved in the REST calls to retrieve HLZ rasters.

Pseudo-Parameter	UI Name	REST Call Name
Helicopter Type	MH-6 / AH-6	MH-6%2520%252F%2520AH-6
Helicopter Type	UH-72A / OH-58D	UH-72A%2520%252F%2520OH-58D
Helicopter Type	AH-1W/Z / AH-64 / UH-1Y/N	AH-1W%252FZ%2520%252F%2520AH-64%2520%252F%2520UH-1Y%252FN
Helicopter Type	UH-60A/L/M / SH-60	UH-60A%252FL%252FM%2520%252F%2520SH-60
Helicopter Type	MV-22B / CV-22B	MV-22B%2520%252F%2520CV-22B
Helicopter Type	CH-47(D/F)	CH-47(D%252FF)
Helicopter Type	CH-53(E/K)	CH-53(E%252FK)
Helicopter Type	Sling Load Aircraft	Sling%2520Load%2520Aircraft
Helicopter Type	Sling Load long lines	Sling%2520Load%2520long%2520lines
Time of Day	Day	DAY
Time of Day	Night	NIGHT
Environment	Desert/Snow	DESERT%252FSNOW
Environment	Mountain	MOUNTAIN
Environment	Jungle	JUNGLE
Environment	Urban	URBAN

Table 4.11 HLZ/Touchdown Pseudo-Parameters

4.1.5.2 Steps for Parameter REST call

1. Submit a GET request to see the set parameter values for the combined pseudo-parameters. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/hlz/parameters;HelicopterType={string};TimeOfDay={string};Environment={string}
```

4.1.5.3 Steps for REST View

1. Submit a GET request to view an HLZ/Touchdown analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/hlz;DataSource={source};Left={value};Bottom={value};Top={value};Right={value};Color={string};HlzDiameter={value};MaxHlzSlope={value};ObstacleRadius={value};ApproachAngle={value};HoverHeight={value}
```

4.1.5.4 Steps for REST Download

1. Submit a GET request to download an HLZ/Touchdown analytic. The REST call via monocle-3 application will look similar to the following:

```
https://iipbeta.digitalglobe.com/monocle-3/app/broker/mrgeo/hlz;DataSource={source};Left={value};Bottom={value};Top={value};Right={value};Color={string};HlzDiameter={value};MaxHlzSlope={value};ObstacleRadius={value};ApproachAngle={value};HoverHeight={value};FileType={type};FileName={string}
```

4.2 Search

SMA submits different calls for each of the sub-widgets, depending on the active sub-widgets.

4.2.1 Social Media Analytics

The InsightCloud Social Media widget provides the analyst with a visual output of up-to-the-second geolocated social media from Twitter and RSS feeds, based on user-defined queries.

General note: the GET call returns an application/json response that changes depending on if and what dimension is used. Below, sections have been divided by the dimensions to provide examples of each type of resulting response.

4.2.1.1 Parameters

Parameters	Type	Default	Optional	Multiple	Description
BBOX	Float	<none>	True	False	Coordinates in decimal degrees, separated by commas, of the bounding box corners. In order, these coordinates are Lower Left Lon, Lower Left Lat, Upper Right Lon, Upper Right Lat; bbox always surrounds the origin point. BBOX=49.122,14.554,4

Parameters	Type	Default	Optional	Multiple	Description
					9.134,14.565 for example
datetimerange	String	<none>	True	False	Date and time restriction on the range of data returned by the request. datetimerange=2015-02-11T08:33:46.812Z,2015-02-11T14:33:46.812Z for example
poly	String	<none>	True	True*	Coordinates in decimal degrees poly=%5B%5B%5B%5B-77.116721,39.002453%5D,%5B-77.514975,38.77156%5D,%5B-77.185385,38.587165%5D,%5B-76.82009,38.72872%5D,%5B-76.905234,39.01099%5D,%5B-77.116721,39.002453%5D%5D%5D%5D for example *Note: multiple polygons can be specified under poly
dimension	String	<none>	True	False	Additional filters for the analytic, displaying specific data in specific fashion. dimension=datetime, for example Possible dimensions are: For Timeline: datetime For Word Cloud values: term

Parameters	Type	Default	Optional	Multiple	Description
					sigterm hashtag For Top Ten values: topten For Statistics: stats For Map Aggregations: geohash
query	String	<none>	True	True	Used for specifying search terms to narrow results. Query follows basic Elastisearch query logic. query=+dmvmusic+AND+positiveSentiment:%3E0.96, for example.
format	String	<none>	True	False	The filetype of the return, SHP packaged in ZIP is supported; DOWNLOAD only format=shp for example
stream-shape-file	String	<none>	True	False	Method of downloading the return, streaming is supported; DOWNLOAD only stream-shape-file=true for example

Table 4.12 Social Media Parameters

Parameters	Limits
Longitude	Between -180 and 180 degrees inclusive
Latitude	Between -90 and 90 degrees inclusive

Table 4.13 Social Media Parameter Limits

4.2.1.2 Differences in REST View - Map – Cluster vs Kernel Density

Note: Sections detailing REST View for maps are capable of using Cluster or KD calls; the examples in later sections are given using Cluster calls.

4.2.1.2.1 Cluster, Cluster Sentiment

Note: Both Cluster and Cluster Sentiment are triggered via the same call; Sentiment makes use of the pos_sentiment_avg and neg_sentiment_avg.

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=geohash

   OR

1. The REST call via monocle-3 application will look similar to the following for **Twitter**:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=geohash

4.2.1.2.2 Kernel Density, KD Sentiment

Note: Both Kernel Density and Kernel Density Sentiment are triggered via the same call; KD Sentiment makes use of the positiveSentiment and negativeSentiment.
Note: For any KD call be sure to include the pageSize={value} in the call; maximum pageSize=100000.

2. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&pageSize={value}

   OR

2. The REST call via monocle-3 application will look similar to the following for **Twitter**:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&pageSize={value}

4.2.1.3 Steps for REST View - Polygon

4.2.1.3.1 RSS

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions={string}&poly={string}

4.2.1.3.2 Twitter

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions={string}&poly={string}

4.2.1.4 Steps for REST View - Map – Viewport

4.2.1.4.1 RSS

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=geohash

4.2.1.4.2 Twitter

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=geohash

4.2.1.5 Steps for REST View - Query – Simple Stats

4.2.1.5.1 RSS

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=stats

4.2.1.5.2 Twitter

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=stats

4.2.1.6 Steps for REST View - Query – Stats with Query

4.2.1.6.1 RSS

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:

https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox ={LLLon,LLLat,URLon,URLat}}&datetimerange={datetime}&dimensions=stats&query={string}

4.2.1.6.2 Twitter

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox ={LLLon,LLLat,URLon,URLat}}&datetimerange={datetime}&dimensions=stats&query={string}

4.2.1.7 Steps for REST Stored Query (Twitter)

4.2.1.7.1 Retrieve List of Stored Queries

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/userprofile/api/workspace

4.2.1.7.2 Save New Query

1. Submit a POST request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/userprofile/api/workspace

   With a source body detailing parameters.

4.2.1.7.3 Load Query

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/userprofile/api/workspace

   FOLLOWING THIS in the UI:
   Various calls made to return the viewport, zoom level, etc to the values stored in the saved query (See: Map call, Query call, Timeline call, Top Ten call, Tweets call, Word Cloud call)

4.2.1.7.4 Delete Query from list

1. Submit a DELETE request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/userprofile/api/workspace?name={string}

4.2.1.8 Steps for REST View - Timeline

4.2.1.8.1 RSS

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox={LLLon,LLLat,URLon,URLat}&dimensions=datetime

4.2.1.8.2 Twitter

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox={LLLon,LLLat,URLon,URLat}&dimensions=datetime

4.2.1.9 Steps for REST View - Top Ten

4.2.1.9.1 RSS

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=topten

4.2.1.9.2 Twitter

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions=topten

4.2.1.10 Steps for REST View - Tweets/Articles Panel

4.2.1.10.1 RSS – Articles

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}

4.2.1.10.2 Twitter – Tweets

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}

4.2.1.11 Steps for REST View - Word Cloud

The Word Cloud has two primary options for dimensions settings, plus an extra dimension option for Twitter. The three dimension options are: *term*, *sigterm*, and the Twitter-only option of *hashtag*.

4.2.1.11.1 RSS

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **RSS**:

   ```
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox
   ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions={string}
   ```

4.2.1.11.2 Twitter

1. Submit a GET request to retrieve InsightCloud Social Media information. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   ```
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox
   ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&dimensions={string}
   ```

4.2.1.12 Steps for REST Download

4.2.1.12.1 RSS

1. Submit a GET request to download the results of an InsightCloud Social Media query. The REST call via monocle-3 application will look similar to the following for **RSS**:

   ```
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/rss/sentences?bbox
   ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&pageSize={value}&format=shp&stream-shape-file=true
   ```

4.2.1.12.2 Twitter

1. Submit a GET request to download the results of an InsightCloud Social Media query. The REST call via monocle-3 application will look similar to the following for **Twitter**:

   ```
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/sma/sma/twitter/tweets?bbox
   ={LLLon,LLLat,URLon,URLat}&datetimerange={datetime}&pageSize={value}&format=shp&stream-shape-file=true
   ```

4.2.2 Unified Vector Index

The Unified Vector Index (UVI) provides the analyst with a tool for visually mapping all available vectors within a given aoi. Analysts may then refine the resulting vectors into a desired selection. These quickly polled and returned arrays of points, polylines, and polygons may then be used in further analytical studies of the area.

4.2.2.1 Aggregation Queries

4.2.2.1.1 Parameters

Parameters	Data Type	Parameter Type	Description
left	String	Query	Longitude in decimal degrees of the lower left bounding box corner
lower	String	Query	Latitude in decimal degrees of the lower left bounding box corner
right	String	Query	Longitude in decimal degrees of the upper right bounding box corner
upper	String	Query	Latitude in decimal degrees of the upper right bounding box corner
binCountX	Integer	Query	The number of horizontal partitions of the bounding box; total bins limited to 20 *Note: total bins = binCountX x binCountY*
binCountY	Integer	Query	The number of vertical partitions of the bounding box; total bins limited to 20 *Note: total bins = binCountX x binCountY*

Table 4.14 Aggregation UVI Parameters

Parameters	Limits
Left	Between -180 and 180 degrees inclusive
Right	Between -180 and 180 degrees inclusive
Lower	Between -90 and 90 degrees inclusive
Upper	Between -90 and 90 degrees inclusive
Source	Current source options are: ACLED, Anthrometer, Gazetteer, GDB, GDELT, HGIS, IIP-SA-Raster, ImageMining, InsightCloudMR, MASS Service, OSM, SETD, Tomnod *Note: Less in Staging

Table 4.15 Aggregation UVI Parameter Limits

4.2.2.1.2 List Available Sources

1. Submit a GET request to retrieve InsightCloud Anthrometer information. The REST call via monocle-3 application will look similar to the following:

https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/aggs/sources?left={value}&right={value}&upper={value}&lower={value}

4.2.2.1.3 List Available Sources (binned)

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/aggs/sources/binned?north={value}&east={value}&south={value}&west={value}&binCountX={value}&binCountY={value}

4.2.2.2 UVI in ESRI ArcMap Add-In

Note: Sources are subject to rapid change, and both the list of available sources as well as the vector count within each listed source may be different than the given examples. Basic principles still apply.

4.2.2.2.1 Parameters

Parameters	Data Type	Parameter Type	Description
left	String	Query	Longitude in decimal degrees of the lower left bounding box corner
lower	String	Query	Latitude in decimal degrees of the lower left bounding box corner
right	String	Query	Longitude in decimal degrees of the upper right bounding box corner
upper	String	Query	Latitude in decimal degrees of the upper right bounding box corner
geometry	String	Path	The geometry type for which to list items or types (e.g. “Point”)
type	String	Path	The vector item type for which to list items (e.g. “Road” or “Media Outlet”)
ttl	String	Query	The time to live for the Elasticsearch paging session.
count	Integer	Query	The number of records to return per shard per page request.
fields	String	Path	The comma-separated list of fields to return for the items.
source	String	Path	The source for which to list items or types.

Parameters	Data Type	Parameter Type	Description
pagingId	String	Form	The paging session ID for which to retrieve a page.

Table 4.16 ESRI UVI Parameters

Parameters	Limits
Left	Between -180 and 180 degrees inclusive
Right	Between -180 and 180 degrees inclusive
Lower	Between -90 and 90 degrees inclusive
Upper	Between -90 and 90 degrees inclusive
Fields	Possible field options are: id, itemDate, ingestDate, ingestSource, name, itemType, format, source, geomType, geom, originalCrs, text, attributes, ingestAttributes
Source	Current source options are: ACLED, Anthrometer, Gazetteer, GDB, GDELT, HGIS, IIP-SA-Raster, ImageMining, InsightCloudMR, MASS Service, OSM, SETD, Tomnod *Note: Less in Staging

Table 4.17 ESRI UVI Parameter Limits

4.2.2.2.2 List Available Sources

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/sources?left={value}&right={value}&upper={value}&lower={value}

4.2.2.2.3 List Available Geometries for a Given Source

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/{source}/geometries?left={value}&right={value}&upper={value}&lower={value}

4.2.2.2.4 List Available Item Types for a Given Geometry and Source

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/{source}/{geometry}/types?left={value}&right={value}&upper={value}&lower={value}

4.2.2.2.5 List Available Vector Items for Given Item Type, Geometry, and Source (Returns Default Fields)

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/{source}/{geometry}/{type}?left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.2.6 List Available Vector Items for Given Item Type, Geometry, and Source (Returns Selected Fields)

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/{source}/{geometry}/{type}/{fields}?left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.2.7 List Available Geometries Matching A Given Query

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/query/geometries?q={query}&left={value}&right={value}&upper={value}&lower={value}

4.2.2.2.8 List Available Vector Item Types Matching Given Geometry and Query

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/query/{geometry}/types?q={query}&left={value}&right={value}&upper={value}&lower={value}

4.2.2.2.9 List Available Vector Items Matching Given Query, Geometry Type, and Item Type

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/query/{geometry}/{type}/{fields}?q={query}&left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.2.10 Retrieve Page of Vector Items by Type and Query (GET)

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/query/{geometry}/{type}/paging?q={query}&left={value}&right={value}&upper={value}&lower={value}&ttl={value}&count={value}

4.2.2.2.11 Retrieve Page of Vector Items by Type (GET)

1. Submit a GET request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/{source}/{geometry}/{type}/paging?left={value}&right={value}&upper={value}&lower={value}&ttl={value}&count={value}

4.2.2.2.12 Retrieve Page of Vector Items (POST)

1. Submit a POST request to retrieve InsightCloud ESRI UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/esri/paging

With a header Content-type:application/x-www-form-urlencoded and a body of the paging ID generated in Section: Retrieve Page of Vector Items by Type (GET) or in Section: Retrieve Page of Vector Items by Type and Query (GET)

4.2.2.3 UVI in Web application

4.2.2.3.1 Parameters

Parameters	Data Type	Parameter Type	Description
left	String	Query	Longitude in decimal degrees of the lower left bounding box corner
lower	String	Query	Latitude in decimal degrees of the lower left bounding box corner
right	String	Query	Longitude in decimal degrees of the upper right bounding box corner

Parameters	Data Type	Parameter Type	Description
upper	String	Query	Latitude in decimal degrees of the upper right bounding box corner
type	String	Path	The vector item type for which to list items (e.g. “Road” or “Media Outlet”)
ttl	String	Query	The time to live for the Elasticsearch paging session.
count	Integer	Query	The number of records to return per shard per page request.
fields	String	Path	The comma-separated list of fields to return for the items.
source	String	Path	The source for which to list items or types.
pagingId	String	Form	The paging session ID for which to retrieve a page.

Table 4.18 Web UVI Parameters

Parameters	Limits
Left	Between -180 and 180 degrees inclusive
Right	Between -180 and 180 degrees inclusive
Lower	Between -90 and 90 degrees inclusive
Upper	Between -90 and 90 degrees inclusive
Fields	Possible field options are: id, itemDate, ingestDate, ingestSource, name, itemType, format, source, geomType, geom*, originalCrs, text, attributes, ingestAttributes *Note: geom will always display, even if not specified
Source	Current source options are: ACLED, Anthrometer, Gazetteer, GDB, GDELT, HGIS, IIP-SA-Raster, ImageMining, InsightCloudMR, MASS Service, OSM, SETD, Tomnod *Note: less in Staging

Table 4.19 Web UVI Parameter Limits

4.2.2.3.2 List Available Sources

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/sources?left={value}&right={value}&upper={value}&lower={value}

4.2.2.3.3 List Available Vector Types for a Given Source

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/{source}/types?left={value}&right={value}&upper={value}&lower={value}

4.2.2.3.4 List Available Vector Items For Given Source and Item type

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/{source}/{type}?left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.3.5 List Available Vector Item Types Matching A Query

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/types?left={value}&right={value}&upper={value}&lower={value}

4.2.2.3.6 List Available Vector Items for Given Item Type and Query

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/{type}?q={query}&left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.3.7 List Available Vector Items by Source, Type, and Matching A Query, Returning Only Specified Fields

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/{type}/{fields}?q={query}&left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.3.8 List Available Vector Items Matching A Query

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/items?q={query}&left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.3.9 List Available Vector Items Matching A Query, Returning Only Specified Fields

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/items/{fields}?q={query}&left={value}&right={value}&upper={value}&lower={value}&count={value}

4.2.2.3.10 Retrieve Page of Vector Items by Type and Query (GET)

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/{type}/paging?q={query}&left={value}&right={value}&upper={value}&lower={value}&ttl={value}&count={value}

4.2.2.3.11 Retrieve Page of Vector Items by Query (GET)

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/paging?q={query}&left={value}&right={value}&upper={value}&lower={value}&ttl={value}&count={value}

4.2.2.3.12 Retrieve Page of Vector Items by Type (GET)

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:
   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/{source}/{type}/paging?left={value}&right={value}&upper={value}&lower={value}&ttl={value}&count={value}

4.2.2.3.13 Retrieve Page of Vector Items (POST)

1. Submit a POST request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/paging

   With a header Content-type:application/x-www-form-urlencoded and a body of the paging ID generated in Section: Retrieve Page of Vector Items by Type (GET) or in Section: Retrieve Page of Vector Items by Type and Query (GET)

4.2.2.3.14 Download Vector Items for a Given Source and Item Type

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/{source}/{type}/zip?left={value}&right={value}&upper={value}&lower={value}

4.2.2.3.15 Download Vector Items for a Given Item Type and Matching a Query

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/{type}/zip?q={string}&left={value}&right={value}&upper={value}&lower={value}

4.2.2.3.16 Download Vector Items Matching a Query

1. Submit a GET request to retrieve InsightCloud UVI information. The REST call via monocle-3 application will look similar to the following:

   https://iipbeta.digitalglobe.com/monocle-3/app/broker/vector/api/vectors/query/items/zip?q={string}&left={value}&right={value}&upper={value}&lower={value}

5 Appendix A - Acronyms

Acronym	Description
AOI	Area of Interest
CEG	Cumulative Exposure Grid
DEM	Digital Elevation Model

Acronym	Description
GIS	Geographic Information System
GPU	Graphics Processing Unit
HLZ	Helicopter Landing Zone
IED	Improvised Explosive Device
Lat/Lon	Latitude/Longitude
TPI	Topographic Position Index
UI	User Interface
UVI	Unified Vector Index
VS	Viewshed