Enhanced Dig Site Information

Well Known Text (WKT) Delivery Options

Georgia 811 can deliver excavation tickets in a variety of formats including text email, html email, and xml. Each format also has the option of including the Well Known Text (WKT) information, which describes the dig site as a series of Latitude and Longitude points. These points can be used to display the dig site on a member's ticket management or GIS system.

This document describes WKT as used by Georgia 811's ticketing system, GeoCall.

Contents

Well Known Text

Georgia 811's ticketing system provides the ability to send dig site information to our members along with the regular ticket details. In addition to ticket number, excavator name and contact information, marking instructions, etc. we can transmit geographic points that describe the area to be excavated. This can then be used by a member to compare the dig site (in conjunction with the Locate Instructions for descriptions of specific particularity) to service area information to help them determine what course of action to take.

The information here is provided for members and other stakeholders that might want to implement Well Known Text (WKT) processing to add dig site visualization into their systems. The samples provided should help with implementation and demonstrates how GeoCall constructs the WKT for different dig site drawing scenarios so that you will know what to expect if you implement this for your systems. Georgia 811 implements a portion of the WKT. We only produce simple two-dimensional area features in the EPSG:426 projection.

A Basic WKT Example

When a ticket is called in to Georgia 811, or submitted online, the dig site is drawn on a map. For instance, a rectangle is a common shape to draw to indicate the area to be excavated. In GeoCall, this might look like this:

This rectangle has four points that define the corners. Those points are identified as Longitude/Latitude points in WKT and are included as part of a Polygon definition. Below is the WKT for the above dig site, highlighted with alternating color to help identify the different points:

```
POLYGON (( -84.3641541604937 33.71316821215546, -84.36414611386687
33.71303657522174, -84.36409515189553 33.71303657522174, -84.36410319852232
33.71317267442025, -84.3641541604937 33.71316821215546 ))
```

The WKT starts with the geometry type definition. In this case (and mostly all cases) it is a POLYGON. It will then be followed by an optional space. This is followed by two consecutive open parentheses. Following this is multiple (four or more) space separated Longitude/Latitude pairs enclosed within the double parentheses, separated with commas.

Note that there are 5 Longitude/Latitude points for this shape, but only 4 corners. This is because the WKT specification "closes" polygons by referencing the same point that the shape begins with as the ending point as well. So, the 5th point is the same as the first. You can see here that the first pair has the exact same longitude and latitude values as the last pair in the POLYGON definition.

The above example shows the semantics of a very simple WKT. The WKT sent by GeoCall will be more complex due to the complexities of actual drawn dig sites and buffering geocoded items in the system. In the following sections, we will show some real-world examples of WKT and discuss this further.

A note on Whitespace: The WKT format calls for whitespace delimiters between the X and Y coordinate. There also may (or may not) be whitespace between other delimiters like the commas and geometry type etc. This following WKT is just as valid:

```
POLYGON((-84.3641541604937 33.71316821215546,   -84.36414611386687
33.71303657522174,-84.36409515189553     33.71303657522174, -
84.36410319852232 33.71317267442025,       -84.3641541604937
33.71316821215546 )  )
```

Most WKT processors will work if you convert multiple whitespace runs into a single space character. There are many open-source projects WKT processing code in them. Here are just a few of the examples:

JavaScript:	OpenLayers.js (https://openlayers.org)
Dot net:	Net Topology Suite (https://github.com/NetTopologySuite/NetTopologySuite/wiki)
Java:	Java Topology Suite (https://github.com/locationtech/jts)
C++:	GDAL (https://gdal.org/api/gdaldataset_cpp.html)

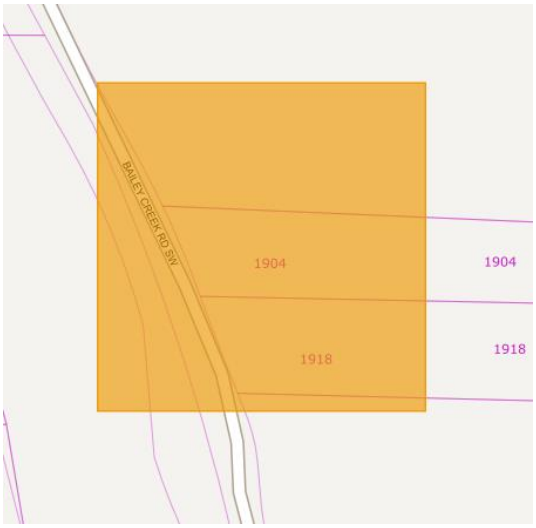
eRequest

Georgia 811's eRequest system is a popular way for homeowners to submit tickets. This system is designed to help the user walk through the process of filling out a simple excavation ticket request. Because a CSR is not involved and the user is not required to complete training as other excavators are, the eRequest system limits the dig site drawing to placing a box on the map and getting confirmation from the user that the excavation will be taking place within the box. So, tickets submitted through this system will always be a box defined in WKT such as this:

eRequest Sample

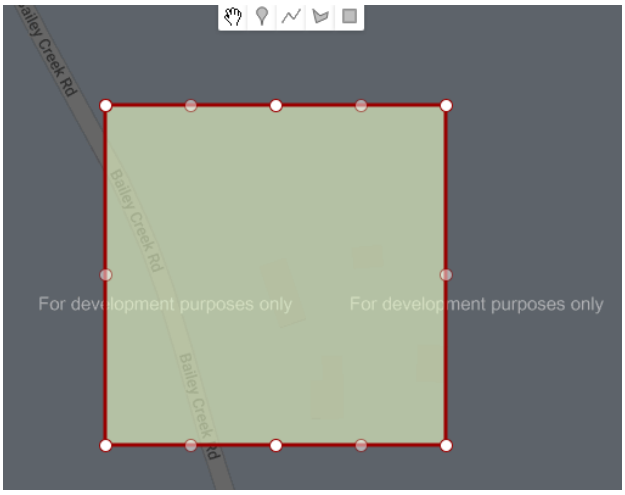
POLYGON((-84.034 33.53583,-84.034 33.53458,-84.03325 33.53458,-84.0325 33.53458,-84.0325 33.53583,-84.03325 33.53583,-84.034 33.53583))

This is an example of what this eRequest dig site looks like in GeoCall:



Also notice in the WKT sample above that there are seven points instead of 5 points from the sample WKT we started with. If we put the WKT from eRequest into Wicket, a WKT javascript library with a demo site located at:

https://arthur-e.github.io/Wicket/sandbox-gmaps3.html, we can see why:



Wicket adds in some additional "handle" points that allows you to alter the shape. Those handles are transparent circles. The WKT points are the opaque white points on the map. Notice that there are two extra points, one in the middle of the top line of the box, and one in the middle of the bottom line of the box. This is simply how eRequest creates the box, where it measures from the center point of an address match north and south to these points, and then extends east and west until it forms a box.

These extra points are really reference points used in this process. But, by connecting the points together, you can see the entire box.

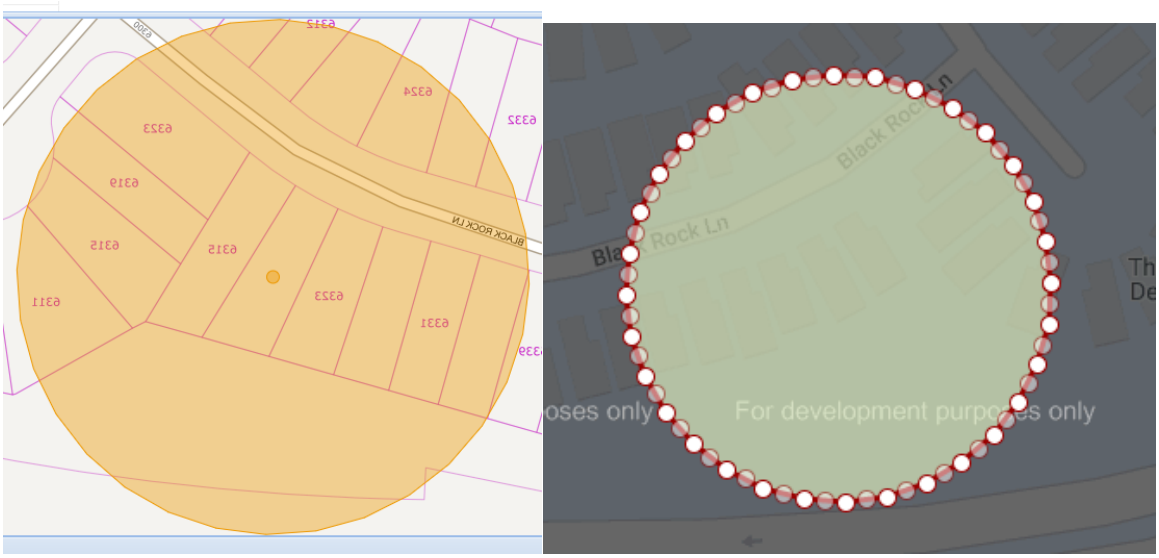
As mentioned above, WKT can add additional whitespace, and some parsers have difficulty with it. If you are trying some of these on your own and Wicket is failing, try an alternative viewer that is more forgiving of whitespace, such as the OpenStreetMap WKT Playground: https://clydedacruz.github.io/openstreetmap-wkt-playground/

Buffered Point Example

Another type of dig site drawing you may see from GeoCall is a buffered point. A point can be added to a map to indicate where digging will occur. For instance, an address match might result in a single point which GeoCall then buffers by 200 feet. Here is a sample WKT of such a buffered point:

POLYGON((-83.8451708127808 34.1149763074013,-83.8451799080436 34.1148688885822,-83.8452140243387 34.1147650096172,-83.8452718505113 34.1146686625008,-83.8453511642722 34.1145835497704,-83.8454489176089 34.1145129422239,-83.845561353923 34.1144595532305,-83.8456841523933 34.1144254344632,-83.8458125940159 34.1144118970597,-83.8459417429418 34.1144194612406,-83.8460666361451 34.1144478363192,-83.8461824741346 34.114495931873,-83.8462848053831 34.114561899645,-83.8463696973874 34.1146432045675,-83.8464338877879 34.1147367221769,-83.84647490974 34.1148388586802,-83.8464911867209 34.1149456890561,-83.8464820931246 34.1150531078876,-83.8464479783183 34.1151569871284,-83.8463901532299 34.115253334742,-83.8463108399837 34.1153384481152,-83.8462130865137 34.1154090563523,-83.8461006494387 34.1154624459791,-83.8459778496957 34.115496565226,-83.8458494064824 34.1155101028824,-83.8457202558899 34.1155025386891,-83.8455953611979 34.1154741633346,-83.8454795221242 34.1154260672837,-83.845377190361 34.1153600988688,-83.8452922984899 34.1152787932557,-83.8452281088504 34.1151852750129,-83.8451870881709 34.11508313803,-83.8451708127808 34.1149763074013))

Notice that this is identified as a POLYGON again. This is because the buffered point is created as a multi-point polygon that resembles a circle from far enough away but is a 32-point polygon. Again, the final point is the same as the first point just as with our simple boxes. This is what this looks like in both GeoCall and in the Wicket viewer:



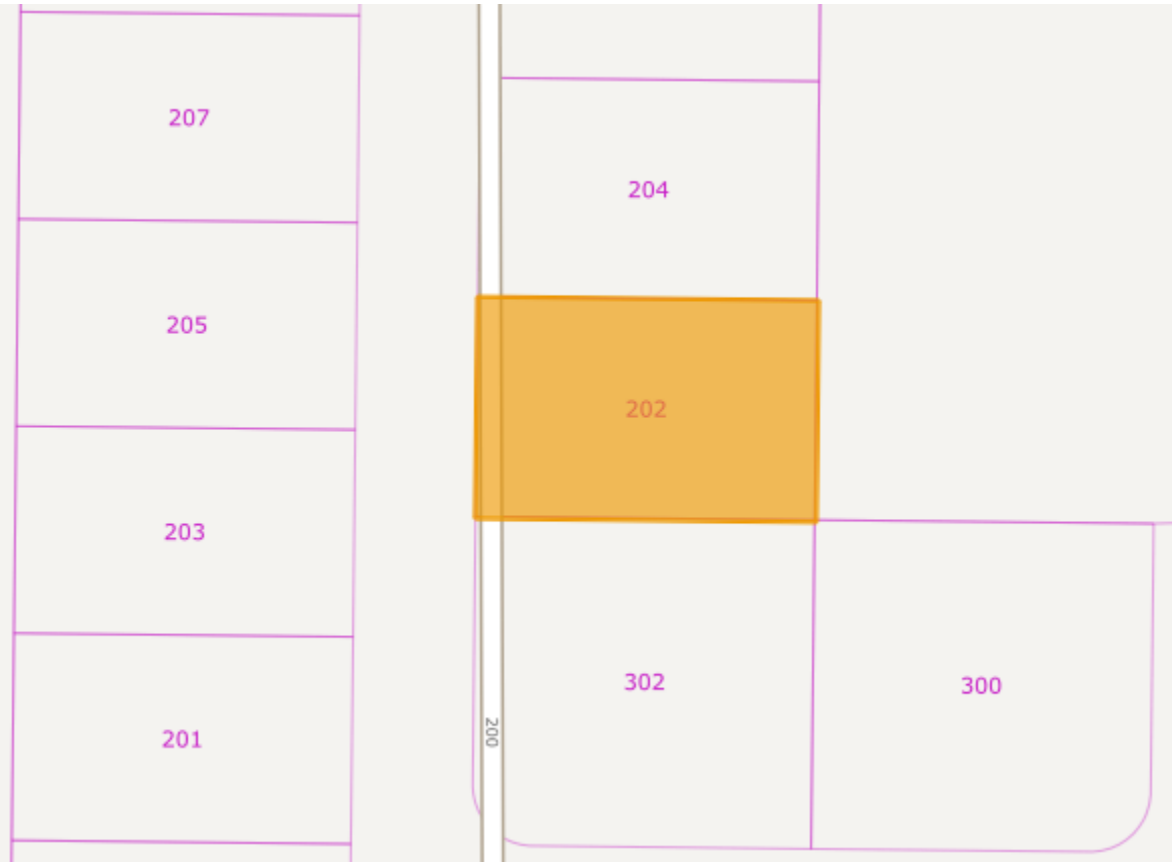
GeoCall's buffering extends and rounds the edges of the dig site, whereas eRequest creates an enclosing box.

Parcel Matches

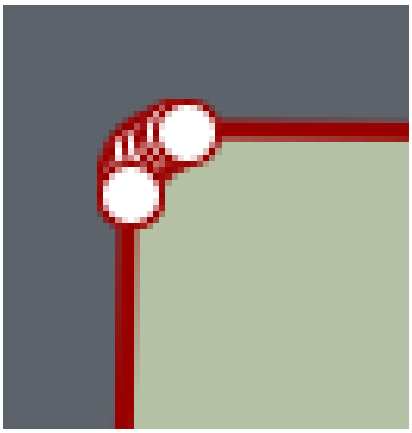
GeoCall also contains land parcel information, and when an address entered matches a parcel record, these can be selected as the dig site. The following is an example of a buffered parcel match:

POLYGON((-83.6380035570491 32.64304204858241,-83.6380034879091 32.6430425839991,-83.6380032966392 32.643043097715626,-83.63800299058981 32.64304356999018,-83.6380025815221 32.64304398267352,-83.6380020851565 32.6430443199063,-83.63800152056791 32.64304456872911,-83.6380009094533 32.643044719579585,-83.6380002752973 32.64304476666091,-83.6375454774971 32.6430406678548,-83.6375448446704 32.64304060935511,-83.6375442374923 32.643040447524896,-83.6375436792964 32.64304018858309,-83.6375431915339 32.64303984248091,-83.6375427929491 32.64303942251879,-83.6375424988595 32.6430389448355,-83.63754232056678 32.643038427788305,-83.6375422649226 32.64303789124689,-83.63754537987619 32.64279052539819,-83.6375454490188 32.64278998998158,-83.6375456402909 32.64278947626569,-83.63754594634199 32.6427890039922,-83.63754635541069 32.64278859131039,-83.63754685177672 32.642788254079306,-83.637547416365 32.64278800525849,-83.63754802747881 32.642787854410194,-83.6375486616332 32.642787807331324,-83.6380034581543 32.642791906114404,-83.63800409097949 32.6427919646118,-83.63800469815659 32.6427921264399,-83.6380052563522 32.642792385379494,-83.6380057441151 32.642792731479894,-83.6380061427008 32.64279315144059,-83.63800643679201 32.64279362912281,-83.6380066150868 32.64279414616932,-83.6380066707335 32.642794682710395,-83.6380035570491 32.64304204858241))

This buffered parcel match looks like this on GeoCall's map:



Notice that there are many more points in the WKT than in the simple example, and in the eRequest example. This has to do with GeoCall's rounding of edges. In this case, if you zoom in very close to the points, you will see that GeoCall has added in several points to round off the corners. This is what one of those corners looks like in the Wicket viewer:



The point count is not much different from the buffered point example above. This is due to how the underlying buffering is applied in GeoCall, and how that application affects different shapes.

Street Segments and Multipolygons

This next example demonstrates two concepts, Street Segments, and Multipolygons. GeoCall can match addresses based on the street name within a city. This will happen if, for instance, the address number is not known or does not yet exist (for new construction). When a street is selected in the search results, this normally results in a line drawn along the street and then buffered on the map.

However, streets can weave in and out of "official" city boundaries. And, because GeoCall matches on the street name as well as the city or locality name, you may sometimes see only parts of a street match *within* the bounds of the city or locality you searched for.

When this happens, GeoCall will draw the sections of the road that it found within the search parameters. This is what a street looks like with a chunk of it outside city boundaries:

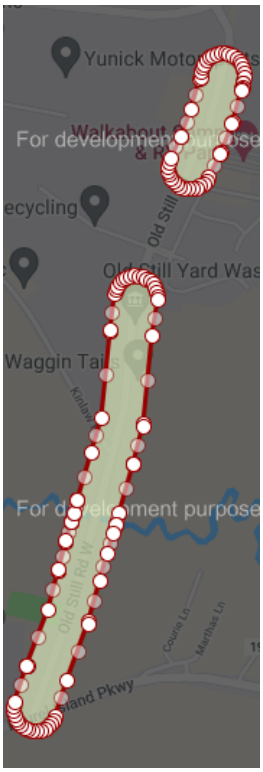
What you are seeing here is two lines on a map that have been buffered (and therefore converted into polygons). The corresponding WKT becomes more complex than what we have seen up until now. Since there are two polygons here, this is no longer a POLYGON, but a MULTIPOLYGON containing the

two polygons. Each polygon within the MULTIPOLYGON is simply a collection of points contained within double parentheses. The polygons are then separated by a comma, and themselves enclosed within the parentheses of the MULTIPOLYGON. Here is the WKT, with each polygon set to a different color for clarity:

MULTIPOLYGON(((-81.6774247398325 30.84101040063159,-81.677114197183 30.8417067785954,-81.6768519129639 30.842474910780993,-81.6768408035828 30.842530459275295,-81.6768226859763 30.842657288889487,-81.6767953153372 30.84276249768159,-81.6767446825325 30.84286107744379,-81.67667273351022 30.842949239992592,-81.67658223305251 30.84302359700623,-81.6764766591394 30.84308129114809,-81.6763600689165 30.843120105141185,-81.67623694284622 30.843138547405907,-81.6761120128614 30.843135909176283,-81.6759900798508 30.84311229189279,-81.6758758298764 30.84306860301921,-81.6757736533172 30.843006521702893,-81.67568747708232 30.842928433596306,-81.6756206126461 30.84283733969862,-81.6755756296449 30.84273674069958,-81.6755542566397 30.842630502505408,-81.67555731493958 30.842522707968406,-81.6755773157552 30.842382708071007,-81.6755818127049 30.842356460826313,-81.6756018133494 30.842256460889704,-81.6756116142496 30.84222579330981,-81.6756185174382 30.842194545173804,-81.6758985222951 30.841374546458585,-81.6759150590113 30.84133246760781,-81.6762450629577 30.840592469039517,-81.6762578177803 30.840570559423494,-81.6762677797911 30.840547597602807,-81.6765977824291 30.839967599005707,-81.6766635833517 30.839875931434324,-81.6767488453541 30.8397971018836,-81.6768502918189 30.8397341397262,-81.676964024392 30.8396894645547,-81.6770856723853 30.839664793190526,-81.6772105608723 30.839661073663297,-81.6773338907004 30.839678448992508,-81.67745092248062 30.839716251353778,-81.6775571586054 30.8397730281943,-81.67764851669078 30.839846597536194,-81.6777214858672 30.839934132150987,-81.67777326191668 30.8400322682968,-81.6778018550555 30.840137234630507,-81.677806166464 30.840244997255404,-81.6777860302796 30.840351415057302,-81.6777422203754 30.840452398476017,-81.6774247398325 30.84101040063159)),((-81.6820045333848 30.827474278823686,-81.6819968022491 30.82749310655479,-81.68137878852082 30.828915539842594,-81.6810772942744 30.829927025665405,-81.6810757913127 30.8299305794379,-81.6810750612608 30.829934310160805,-81.6808450656548 30.83066431118142,-81.680843503884 30.830669179623012,-81.6806835066998 30.831159180370598,-81.680682873565 30.8311605772851,-81.6806825592406 30.83116205191569,-81.68060256095852 30.831402051341595,-81.6806025609479 30.831402051391592,-81.68050256280668 30.831702051766786,-81.68050231210471 30.83170259616961,-81.6805021879055 30.83170317197601,-81.68008391290022 30.83294806177662,-81.6798087727928 30.833830426275387,-81.6795440400143 30.836036652657597,-81.67954121397561 30.836056628858703,-81.6793912198683 30.836976629526085,-81.67936176652202 30.837081418436803,-81.67930918741162 30.83717923552621,-81.679235503143 30.8372663217746,-81.6791435453631 30.837339330411595,-81.6790368479805 30.83739545589241,-81.6789195112483 30.83743254109048,-81.67879604455209 30.83744916095981,-81.6786711926732 30.837444676776908,-81.6785497535806 30.837419260918903,-81.6784363943257 30.83737388997809,-81.67833547122332 30.837310307622907,-81.67825086278971 30.83723095734378,-81.67818582027641

30.837138888576902,-81.6781428434558 30.837037639415207,-81.6781235836794
30.8369311009511,-81.67812878103578 30.83682336736559,-81.67827716205561
30.835913333701978,-81.6785459754152 30.833673345229297,-81.6785644807476
30.833587073642793,-81.6788544863058 30.83265707498019,-81.6788566735696
30.832652076246006,-81.678857806725 30.8326468232451,-81.6792776241479 30.83139739030059,-
81.6793774408331 30.831097945758803,-81.6793774410465 30.831097945295408,-
81.6793774411404 30.831097944854914,-81.6794569609488 30.830859390008115,-
81.67961570420628 30.83037324849978,-81.6798437961229 30.8296493190448,-81.6801527137564
30.8286129727083,-81.6801731923477 30.828556887939296,-81.6807991364362
30.827116243960898,-81.68115547402698 30.826195719878104,-81.6812064881862
30.8260972898367,-81.6812787734737 30.826009339850906,-81.6813695517179
30.825935249866205,-81.68147533442999 30.825877867045897,-81.6815920565954
30.825839396525403,-81.6817152325899 30.825821316682415,-81.68184012891192
30.825824322289797,-81.6819619460122 30.825848297934602,-81.682076002536
30.825892322101904,-81.6821779155039 30.825954703218297,-81.6822637683599
30.826033043819777,-81.6823302618688 30.826124333411297,-81.682374840715
30.826225063921896,-81.6823957916981 30.826331364258,-81.68239230962298
30.826439149376004,-81.6823645281187 30.826544277210317,-81.6820045333848
30.827474278823686))**)**

Just as with other POLYGONs, the buffering rounds out the ends of each segment with many points, which can be seen in the Wicket Viewer:



MULTIPOLYGONS represent a very small portion of tickets, but they do occur. WKT supports more objects than POLYGONS and MULTIPOLYGONS but for the full buffered dig site, GeoCall will only send these two types of objects.

The WKT Specification

The full specification document for Well Known Text is available at https://www.ogc.org/standards/sfa. Within this document, the relevant sections are 7.2.1 and 7.2.2, which describes how to construct the two-dimensional geometry with WKT. This two-dimensional geometry is what we use, and the x, y coordinates are Longitude and Latitude points, with an implied WGS 84 projection (https://epsg.io/4326).

This information is provided as a starting point for further investigation for Members and other stakeholders that are interested in receiving WKT to enhance their systems.