LEOWEB Version 24.0 User Manual

Kansas Geological Survey
March 2024

For nearly 50 years the Kansas Geological Survey (KGS) has been involved with developing computer programs to convert legal land descriptions to mapping coordinates. The first program was written in 1964 by Donald Good. His program used 126 land locations and an arbitrary mapcoordinate system to test the possibility of using a computer to convert section, township, and range notation to Cartesian coordinates. Good's program was written in FORTRAN II for an IBM 1620 computer. Subsequent programs, such as KANS, developed by Charles O. Morgan and Jesse M. McNellis (1969) offered significant improvements, including the ability to calculate latitude and longitude values. KANS was written in FORTRAN IV and tested on an IBM 7040 at the University of Kansas Computing Center. The program utilized 948 latitude and longitude control points and 962 township and range correction parameters.

These early programs ran in a mainframe environment, making access to the public impractical. With the advent of the PC, however, KGS employee Charles G. Ross (1989) saw an opportunity to write such a program. Ross named the program LEO, a play on words "LEGAL to GEO." In 1994 LEO II was released as a second-generation version of the program. Subsequent development work was performed by David R. Collins with the release of LEO 3.4, LEO 3.6, and finally LEO 3.9, released in 1999. The DOS versions of LEO were written in FORTRAN 77 for IBM-compatible personal computers. They remained a popular tool for industry, researchers, and government for nearly 20 years.

In 2007 work began on a GUI version of LEO that would utilize the benefits of the more powerful yet low-cost computers. LEO 7.0 was developed by Glen Gagnon as a stand-alone desktop program written in Java. Released in 2008, it offered an intuitive graphic user interface, NAD27 and NAD83 conversion options, and batch processing capabilities. A key aspect of the LEO 7.0 design was the ability to de-couple the calculation engine from the GUI front end. When implemented in this fashion, the same engine used in the desktop version could be called within database environments, such as Oracle. This feature made LEO very flexible and guaranteed the same results in all environments.

In 2024, the GUI of LEO (LEOWEB 24) was enhanced and additional functionality of bulk upload of records as a file was successfully implemented. Deployed as a web service, LEO is called thousands of times per day to perform location calculations on the KGS web site.

As of this writing, LEO 7.0 has been retired and is no longer supported by the KGS. This is primarily because web-based software and the advent of mobile computing have made it possible to implement a solution that streamlines use, runs remotely, and can update coordinates in a timely manner. This document serves as the User Manual for the LEOWEB 24 application and is intended to assist users with various features of the program.

Application Overview:

LEOWEB 24 is a web-based application that converts Kansas Public Land Survey System (PLSS) legal land descriptions to geographic coordinates and vice versa. It represents the most recent development efforts in a series of computer programs designed to perform coordinate conversions. The project was funded by the Kansas GIS Policy Board and the Kansas Geological Survey (KGS). It is not a surveying tool and only provides approximate conversion values. The program serves a broad range of users, including industry, researchers, government agencies, and the general public. The primary goal of the project is to provide a web-based version of LEO that 1) leverages existing development work, 2) simplifies use, and 3) provides updated geographical information to a larger population of end users in a timely manner.

Key features of the web version include the following:

1) An Interactive tool to convert PLSS to Geographic and Cartesian coordinates.
2) NAD27/NAD83/WGS84 coordinate conversions to Township, Range, and Section (TRS).
3) Support for Multiple Latitude/Longitude input formats.
4) Conversion of TRS and footage to latitude/longitude coordinates.
5) Conversion of TRS and Quarter Calls to latitude/longitude coordinates.
6) Conversion of Universal Transverse Mercator (UTM) coordinates to TRS.
7) A REST web service.

There are two distinct components to the project:

1. LEOWEB offers an interactive web page for performing online coordinate conversion
2. LEOWEB also provides a web service intended for accessing conversion algorithms via computer programs. This second method is referred to as a REST web service and renders data as JSON structured string.

The scope of work for the project included the design, development, and implementation of the application and support documentation. LEOWEB 11 development was initially performed by KGS Senior System Analyst/Programmer Glen Gagnon and KGS Oracle Database Administrator Kurt Look. The LEOWEB 24 development effort consisted of modern upgrades to the application and to the user interface. This work was completed by the KGS IT Development team of Sayeed Ouddin, Ramya Paladugu, and Sabrina Shaik with technical oversight by Asif Iqbal.

A link to the LEOWEB 24 application can be found on the Kansas Geological Survey's web site at the following URL: https://kgs.ku.edu/coordinate-conversion-tools

Getting Started:

A basic understanding of latitude and longitude coordinates, geodetic reference datum, and the PLSS system is required to use the program properly. Fortunately, a wealth of information on these subjects can be found on the Internet. The Survey's Public Information Circular 20, "The public land survey system in Kansas," by Daniel R. Suchy, is an excellent place to start. This paper can be found online at http://www.kgs.ku.edu/Publications/pic20/pic20_1.html

System Requirements:

It requires a web browser that supports JavaScript and adheres to W3C standards for HTML and CSS. The application has been tested on multiple versions of Microsoft Edge, Mozilla Firefox, and Google Chrome.

Running the Program:

The LEOWEB application can be found on the Kansas Geological Survey's website at the following URL: https://maps.kgs.ku.edu/leoweb/

Upon navigating to the web page the user is presented with the display panel depicted in figure 1. At this point you could convert a latitude/longitude pair to PLSS values by entering coordinates into the text fields and pressing the Submit button. There are a total of nine tabbed panels; six are for coordinate conversions and three for informational purposes. This document considers each panel, but first we should highlight some key program features.

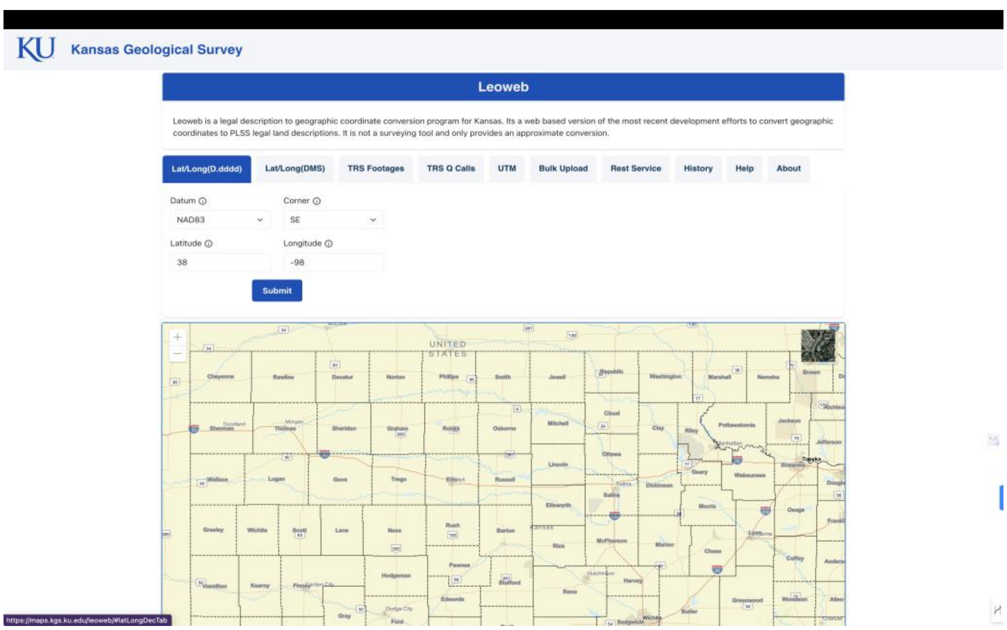


Figure 1. Default display panel for LEOWEB 24.

Screen Layout:

Each page can be divided into two key regions. One region is for navigation and the other for content. The navigation region is at the top of each page. It shows a tabbed menu bar used to select the conversion options. Once clicked, a tab is highlighted and the corresponding page is rendered.

The content region below the navigation bar is a combination of input fields and an area for showing conversion results. Common to all conversion pages is the requirement for specifying a **Datum** and reference **Corner**. A datum is a reference frame that defines the earth as a geometric model. LEOWEB datum options are NAD27, NAD83 and WGS84. The reference corner is used to determine the offset footage of the spot with respect to the section boundaries. Dropdown lists are used for selecting the Datum and reference Corner on all of the LEOWEB conversion pages.

Input field text boxes lie just below the drop-down lists. These parameters will vary depending on the type of conversion that is being requested. After entering the conversion values, the user can click on the **Submit** button, and the results will be displayed in the **Result String** text area**.**

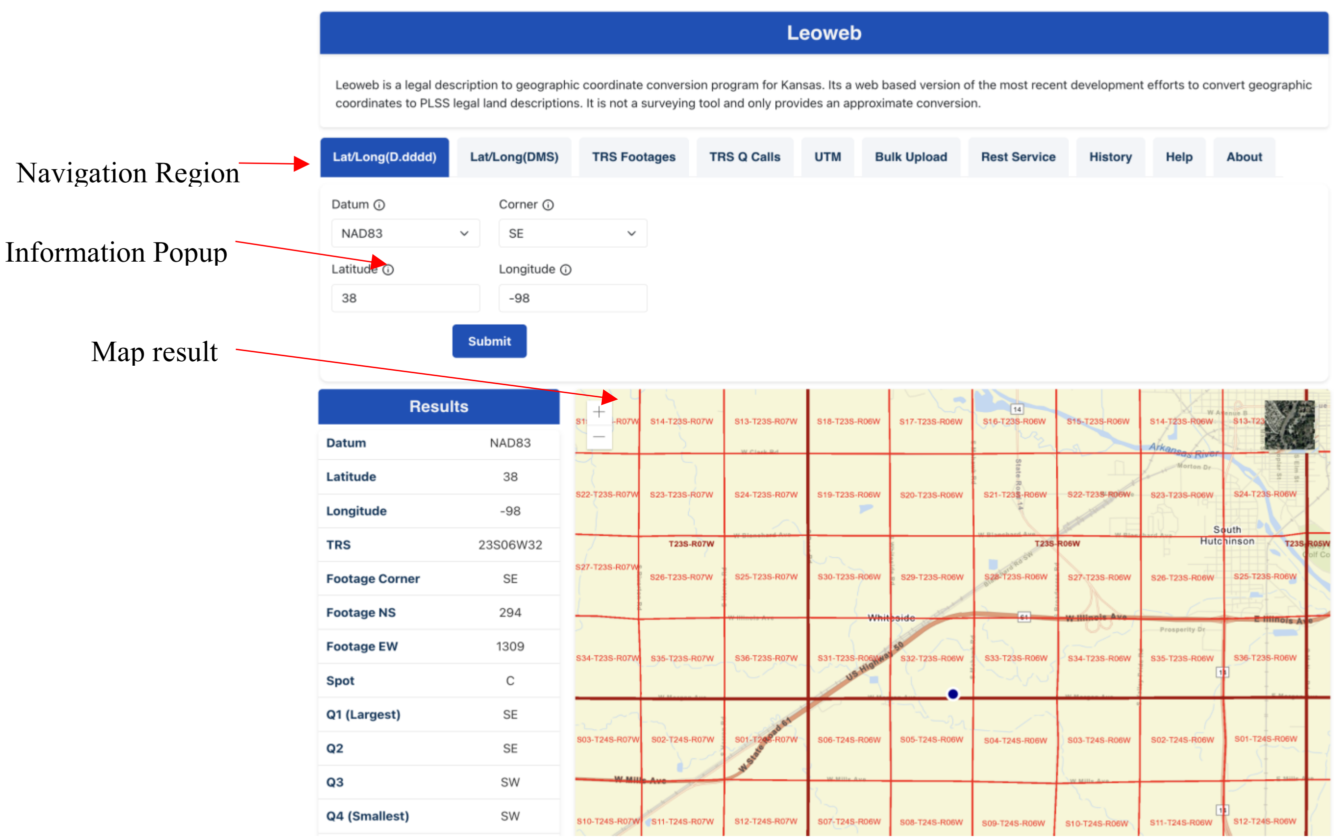


Figure 2. Navigation and Map result.

Interactive Map:

LEOWEB 24 introduces map view functionality, allowing users to visualize input coordinates as a blue dot on the map. This feature provides two types of maps: a Street view map and a high-resolution satellite image.

A notable feature of this update is the integration of the (PLSS) layer, which overlays sections, townships, and ranges onto the map as a grid. Additionally, the county boundaries of Kansas are also depicted as a grid overlay, offering a clear representation of geographic boundaries.

Conversion History:

The "Search History" feature allows you to view a log of all the searches conducted within the application. This can be useful for tracking previous searches and revisiting past results.

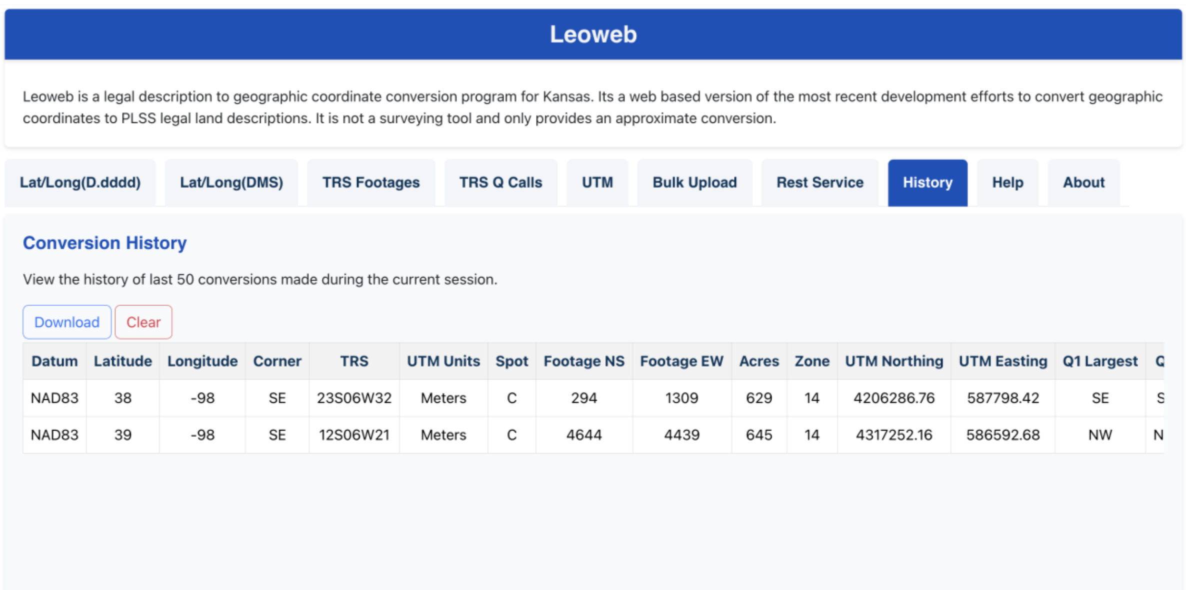


Figure 3. History Tab.

Interactive Help:

Help has been added to all of the interactive screen objects. This includes the dropdown lists, input textboxes, and radio buttons. For help with any object, simply click on the label that corresponds to the object (i.e. click on the word Latitude). A help dialog box will appear with information associated with the object, including acceptable min/max values. (See figure 3).

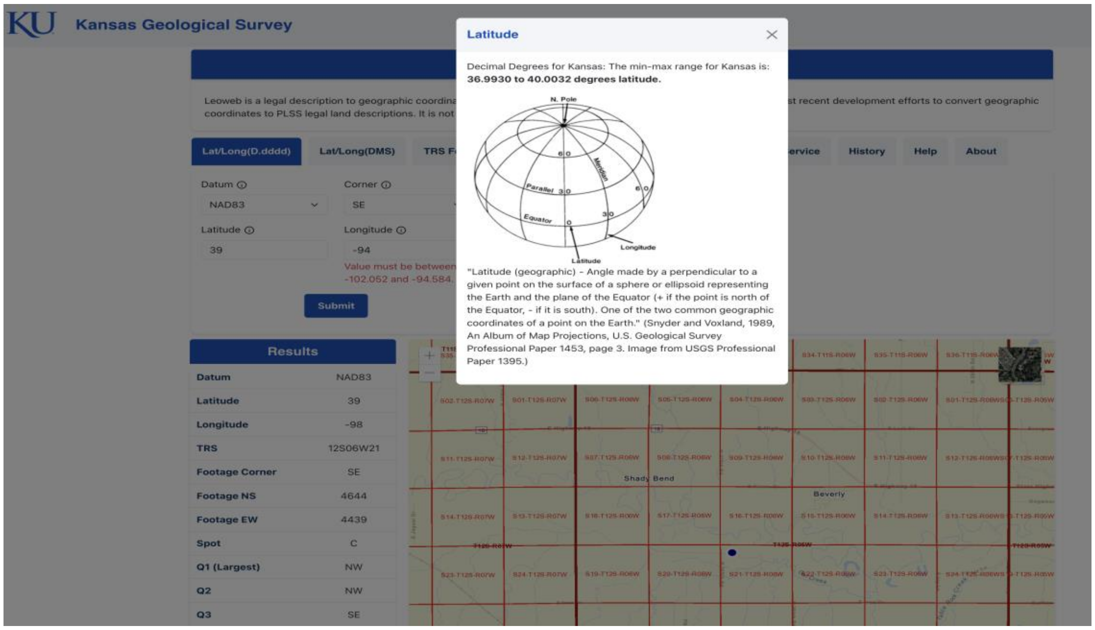


Figure 4. Help is available on all interactive objects.

Conversion Options:

LEOWEB provides six routines for coordinate conversion:

1) Convert decimal degree latitude/longitude coordinates to TRS values.
2) Convert degree, minutes, and seconds latitude/longitude coordinates to TRS values.
3) Convert TRS and footage values to latitude/longitude coordinates.
4) Convert TRS and quarter call values to latitude/longitude coordinates.
5) Convert UTM coordinates to TRS values.

Coordinated conversion options can be performed using the interactive web pages or through the REST web service. Both methods require the same input parameters; however, the output results vary slightly between the two methods (see Table 1). The difference has to do with how the data are accessed, utilized, and rendered. Interactive users generally have only a few coordinates to convert at one time. In this scenario, the latest results are displayed on the screen, and the entire history is written to the history tab. With the REST service, the data are accessed by a computer program and the results are returned as JSON (key-value paired information).

Examples of the REST service and JSON are given later in this document. However, a detailed discussion of these subjects is beyond the scope of this paper.

Table 1 - LEOWEB interactive versus REST output values.

Parameter Name (Description)	Data Type	Web Page	Rest Service
Version	Number	No	No
TRS (Township, Range, and Section string - i.e 16S24W11)	Character	Yes	Yes
Latitude	Number	Yes	Yes
Longitude	Number	Yes	Yes
Datum	Character	Yes	Yes
Township	Number	Yes	No
Township Direction	Character	Yes	No
Range	Number	Yes	No
Range Direction	Character	Yes	No
Section	Number	Yes	No
Footage NS (North or South distance to section line in feet)	Number	Yes	Yes
Footage EW (East or West distance to section line in feet)	Number	Yes	Yes
Footage Corner (Reference corner for offset footages)	Character	Yes	Yes
Spot	Character	Yes	Yes
Q1 Largest	Character	Yes	Yes
Q2	Character	Yes	Yes
Q3	Character	Yes	Yes
Q4 Smallest	Character	Yes	Yes
Len_N_Side_Feet	Number	Yes	Yes
Len_S_Side_Feet	Number	Yes	Yes
Len_E_Side_Feet	Number	Yes	Yes
Len_W_Side_Feet	Number	Yes	Yes
Acres (Approximate number of acres in the section)	Number	Yes	Yes
UTM_Northing (Y coordinate)	Number	Yes	Yes
UTM_Easting (X coordinate)	Number	Yes	Yes
UTM_Zone	Number	Yes	Yes
UTM_UNITS (UTM coordinates are in meters)	Character	Yes	Yes
Errors (Error trapping message returned)	Character	Yes	Yes

A comparison chart of the output fields is depicted in the above table. For the most part the fields returned with the interactive page correspond to the fields returned from the web service. Yet, there are some instances where the JSON tag name slightly differs. The results can be seen in Example 1, a complete JSON returned by the REST service.

```
{
  "errorMessage": "",
  "statusCode": 1,
  "data":{
           datum: "NAD83",
           latitude: "39.202159",
           longitude: "99.170960",
           trs: "10S17W12",
           footageCorner: "SE",
           nsFootage: 4129,
           ewFootage: 3450,
           spot: "C",
           q1Largest: "NW",
           q2: "NE",
           q3: "SW",
           q4Smallest: "SE",
           acres: 638,
           lenNSideFeet: 5261,
           lenSSideFeet: 5245
           lenESideFeet: 5278,
           lenWSideFeet: 5289,
           utmZone: 14,
           utmUnits: "Meters",
           utmNorthing: "485238.529260",
           utmEasting: "4339224.726356",
        }
}
```
Example 1. JSON output from the LEOWEB REST Service.

Reporting Quarter Calls and Spot:

Quarter call subdivisions are reported left to right with the smallest first and the largest last. For a nominal section (one square mile), the smallest subdivision represents an area of 2.5 acres. For geographic to PLSS conversions, the "spot" within the smallest subdivision is estimated using a closest approximation algorithm. It represents the best fit of the actual coordinate to one of nine possible pre-designated positions. In figure 5, the geographic coordinate "X" is closest to the northwest (NW) corner of the subdivision. The nine fixed positions are 1) center, 2) northeast corner, 3) southeast corner, 4) southwest corner, 5) northwest corner, 6) center of the north line, 7) center of the east line, 8) center of the south line, and 9) center of the west line.

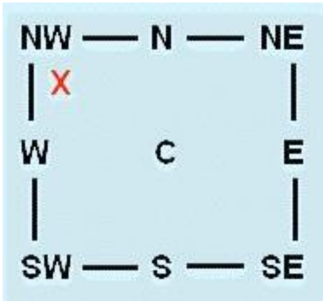


Figure 5. The spot is one of nine possible subdivision locations.

When entering TRS and Quarter calls (TRS Q Calls), the user specifies the spot (see above) and up to four levels of subdivision. A nominal section is divided as follows:

Level	Acres	Example
0. -	640 acres	(i.e. Center of Township 1 South, Range 1 East, Section 1 or 01S01E01.)
1. -	160 acres	(i.e. Center of NW quarter of 01S01E01)
2. -	40 acres	(i.e. Center of SE NW quarter of 01S01E01)
3. -	10 acres	(i.e. Center of SW SE NW quarter of 01S01E01)
4. -	2.5 acres	(i.e. Center of SW SW SE NW quarter of 01S01E01)

Figure 6 shows location "X" spotted to three levels of subdivision. In LEOWEB "quarter calls" are reported "spot first" then from smallest to largest subdivision. In this example the q-call location could be identified as being the center of the southwest quarter (smallest) of the southeast quarter of the southwest quarter (largest) of Section 7, Township 9S, Range 5E.

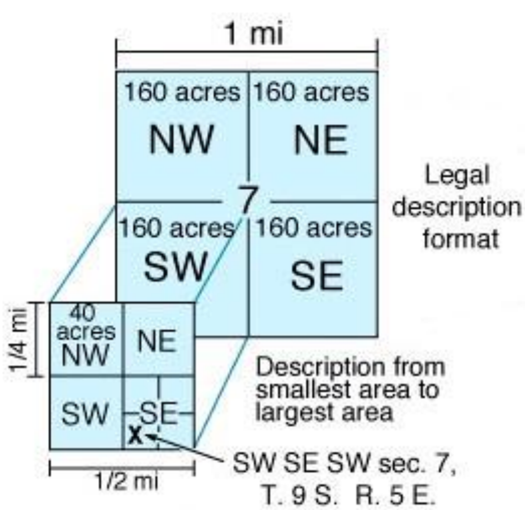


Figure 6. This image shows three levels of subdivision.
Note: LEOWEB reports on up to four levels of subdivision.

Four levels of subdivision are computed when converting geographic coordinates to TRS Calls. For example: A NAD83 lat/long coordinate of 38.41039961 and -99.720785 would approximate to a spot near the **center of the NW NE SW SE** subdivision of sec. 11, T. 19 S., R. 22 W. It should be noted that converting lat/long values to spot and quarter calls only produces a rough approximation of the actual location. In this example, the "Center" spot is approximately 40 feet away from the actual location. Accuracy is improved by reporting TRS and footage descriptions if requirements permit.

Geodetic Datum:

When working with latitude and longitude coordinates, the North American Datum (NAD) serves as the horizontal geodetic control. This program allows the user to select from NAD27, NAD83, or the WGS84 datum. It is essential to record and specify the correct geodetic datum when capturing and converting data. Guessing or specifying the wrong datum results in calculation errors on the order of 10's of feet.

Footage Corner:

Footages refer to the offset distance from the location spot to the boundary lines of the section. A footage corner is used to identify which boundary lines are used in calculating the footage offset. If the southeast corner is specified (default), then the program calculates the offset distance from the south and east lines. Likewise the northwest corner would specify the offset with respect to the north and west boundary lines. Figure 7 depicts a location spotted 1750 feet north of the south line and 2000 feet west of the east line. The LEOWEB program uses a dropdown list to allow the user to change the reference corner for offset footages.

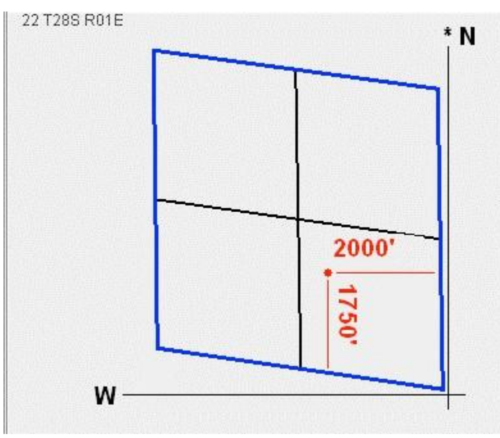


Figure 7. Spot footage 1750 ft north of south line and 2000 ft west of east line sec. 22, T. 28 S., R. 1 E.

Valid Kansas Coordinates:

LEOWEB is designed to report on locations that fall within the borders of Kansas. To aid the user, client-side data validation routines have been added that create warning messages when values fall outside a specified range. Figure 8 depicts an example of a warning message that is created by a data validation routine. The appearance of the message will vary depending on how the browser renders JavaScript messages. Google Chrome was used for this example

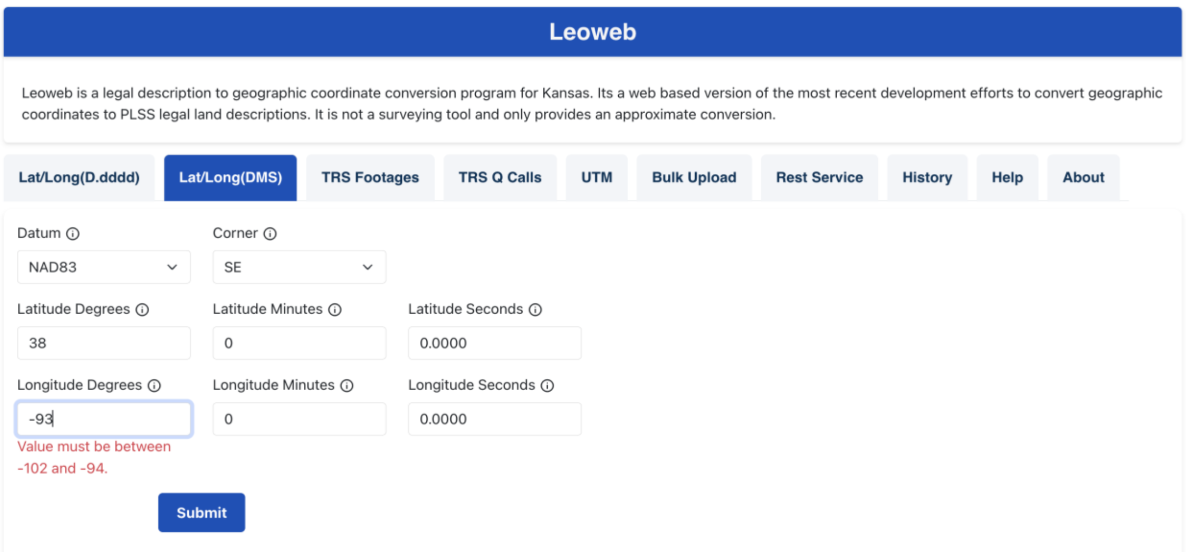


Figure 8. Data validation warning message for an invalid entry.

The minimum and maximum value input parameters are summarized in Table 2.

Table 2

Conversion Option	**Parameter**	**Minimum Value**	**Maximum Value**
Decimal Degree Lat/Long to TRS NAD83/WGS84	Latitude	36.99300	39.9890
Decimal Degree Lat/Lon to TRS NAD83/WGS84	Longitude	-102.052000	-94.584000
Decimal Degree Lat/Lon to TRS NAD27	Latitude	36.9930 000	40.003200
Decimal Degree Lat/Lon to TRS NAD27	Longitude	-102.052000	-94.584000
Degree Minute Second Lat/Lon to TRS	Latitude Degrees	36	40
Degree Minute Second Lat/Lon to TRS	Latitude Minutes	0	59
Degree Minute Second Lat/Lon to TRS	Latitude Seconds	0.00	59.99
Degree Minute Second Lat/Lon to TRS	Longitude Degrees	-94	-102
Degree Minute Second Lat/Lon to TRS	Longitude Minutes	0	59
Degree Minute Second Lat/Lon to TRS	Longitude Seconds	0.00	59.99

Degree Decimal Minutes Lat/Lon to TRS	Latitude Degrees	36	40
Degree Decimal Minutes Lat/Lon to TRS	Latitude Minutes	0.0000	59.9999
Degree Decimal Minutes Lat/Lon to TRS	Longitude Degrees	-94	-102
Degree Decimal Minutes Lat/Lon to TRS	Longitude Minutes	0.0000	59.9999
Conversion Option	**Parameter**	**Minimum Value**	**Maximum Value**
TRS Conversions to Lat/Lon	Township	1	35
TRS Conversions to Lat/Lon	Range East	1	25
TRS Conversions to Lat/Lon	Range West	1	43
TRS Conversions to Lat/Lon	Section	1	36
TRS and Offset Footage to Lat/Lon	N-S Footage	0	5280*
TRS and Offset Footage to Lat/Lon	E-W Footage	0	5280*
UTM Coordinates to Lat/Lon	Northing	Not Validated	Not Validated
UTM Coordinates to Lat/Lon	Easting	Not Validated	Not Validated
UTM Coordinates to Lat/Lon	Zone	13	15

* Allows input greater than 5280 feet but throws a warning message.

Irregular Sections:

A section is an area 1/36th of a township bounded by 1 square mile and covering 640 acres. Many sections in Kansas are smaller or larger than the prescribed nominal section. In fact very few, if any, sections are true square miles and less than 7 percent of Kansas sections actually cover an area equaling 640 acres. Figure 9 shows sec. 6, T. 35 S., R 8 E., encompassing 1616 acres. This irregular section is extremely large having east and west boundary lines of 5,284 and 5,285 feet, respectively, and north and south boundary lines of 13,316 and 13,327 feet in length. Fractional sections are frequently found along the borders and in areas with natural obstacles such as rivers. Users should be aware of irregular sections and problems associated with converting lat/long values to legal land descriptions.

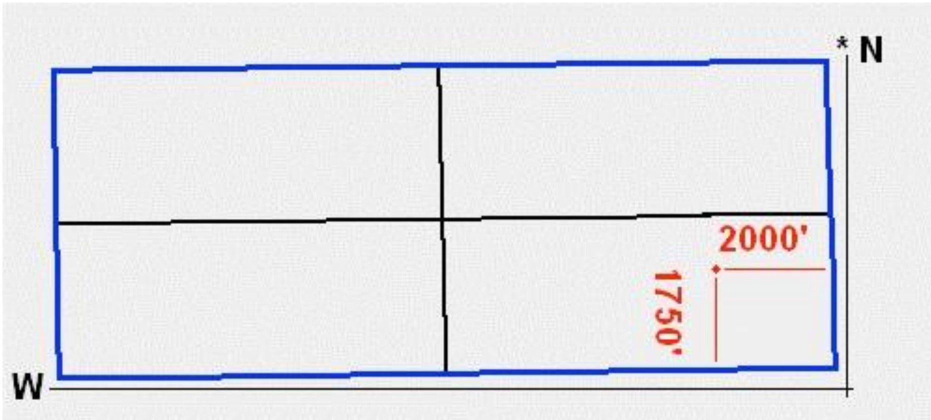


Figure 9. Irregular sec. 6, T. 35 S., R. 8 E. covers over 2 square miles of land.
Note: Compare the footage spot to figure 7.

Unfortunately LEOWEB does not handle irregular sections very well. The southern boundary of sec. 36, T. 26 S., R. 20 E. (depicted in figure 10) is a good example of a section boundary that is not a straight line. In this example, land below the dashed line is still inside the section but not considered in the conversion calculation. This is because the calculation engine only considers the corner points of the section. Since LEOWEB uses an equal subdivision method, the program also has problems when subdividing irregular sections. The region around Fort Riley, Kansas, is a good example of an area with some extremely irregular sections. (See figure 11).

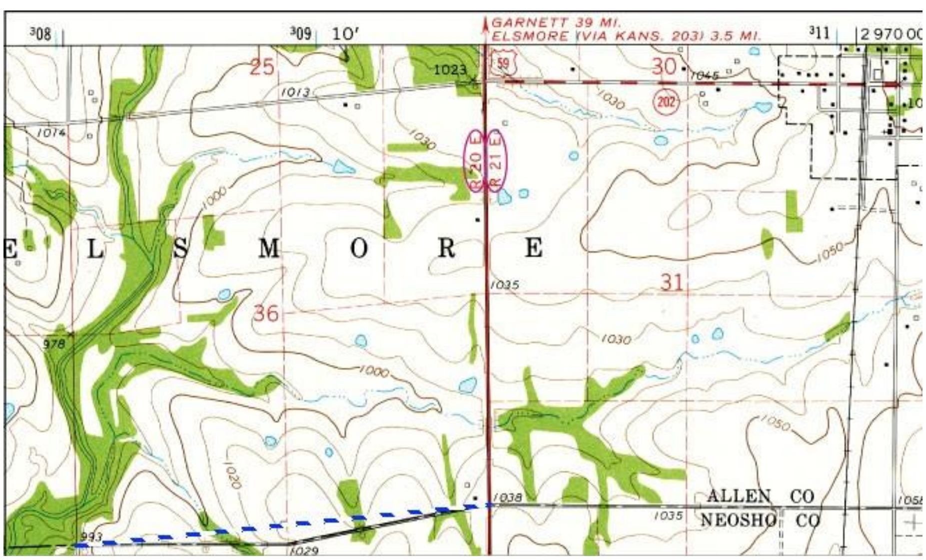


Figure 10. Irregular sec. 36, T. 26 S., R. 20 E. Note: the area below the dashed line is not considered in the LEOWEB conversion.

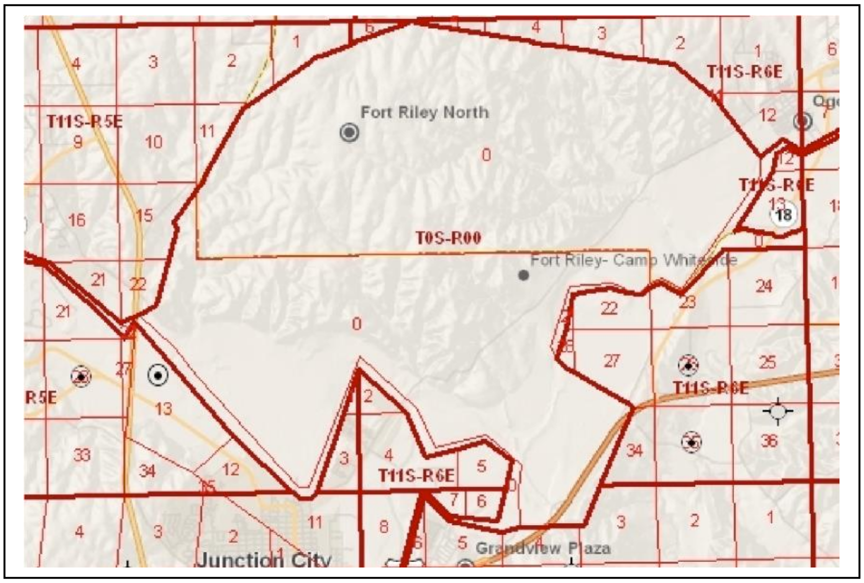


Figure 11. Irregular sections around the Fort Riley area.

Conversion Screens:

The following is a brief discussion on each of the six interactive web pages.

Enter Latitude and Longitude as Decimal Degree:

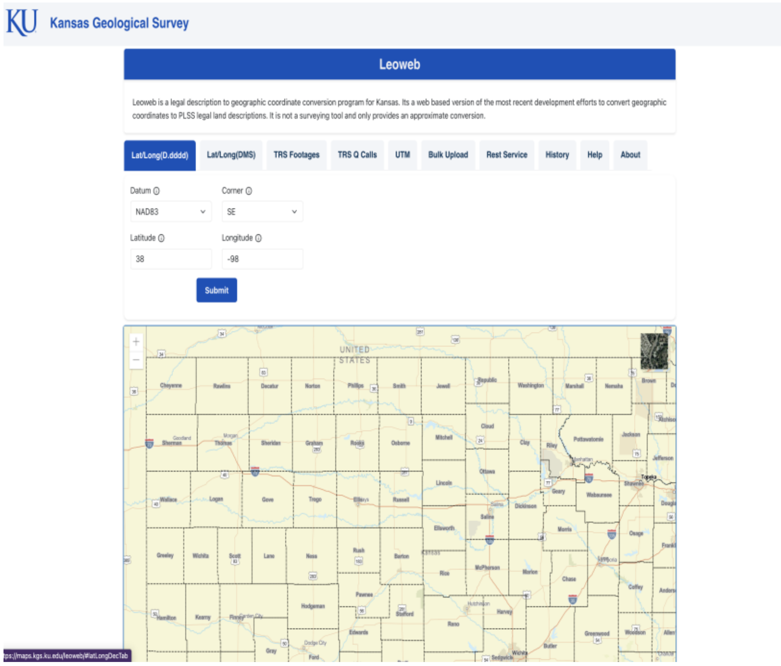


Figure 12. Decimal Degree Lat/Long to TRS.

The option to convert decimal degree lat/lon values to TRS information (figure 12) is the default screen that appears on start-up. Data-validation routines force the values to conform to the minimum/maximum range specified in table 2.

Select the appropriate Datum and reference Corner from the dropdown lists and enter the desired latitude and longitude values prior to “clicking” on the **Submit** button. The conversion will be calculated and the results displayed in a table below.

Enter Latitude and Longitude as Degrees Minute Seconds:

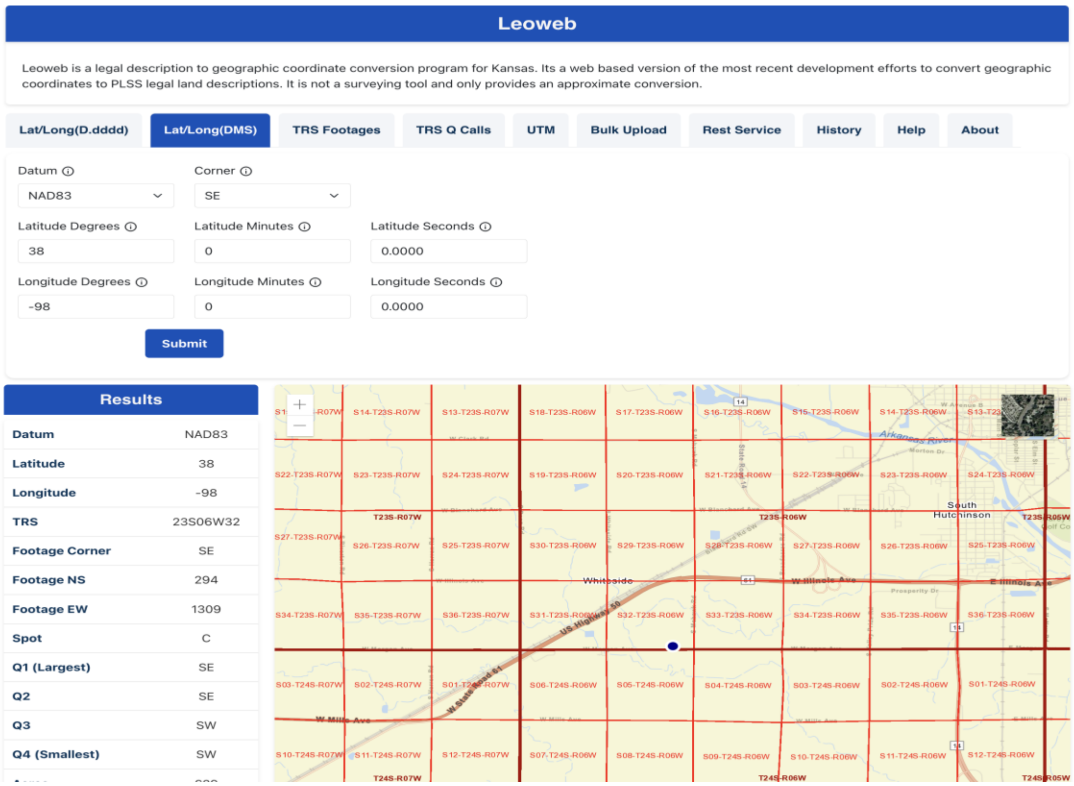


Figure 13. Degree Minute Second Lat/Long to TRS.

The second option converts lat/long values specified in degrees, minutes, and seconds to TRS. Figure 13 shows the data input fields and an example of the calculated results. Notice the datum has been set to calculate NAD27 values, and the reference corner has been set to report using the northeast reference corner.

Enter Township Range Section and Footage values:

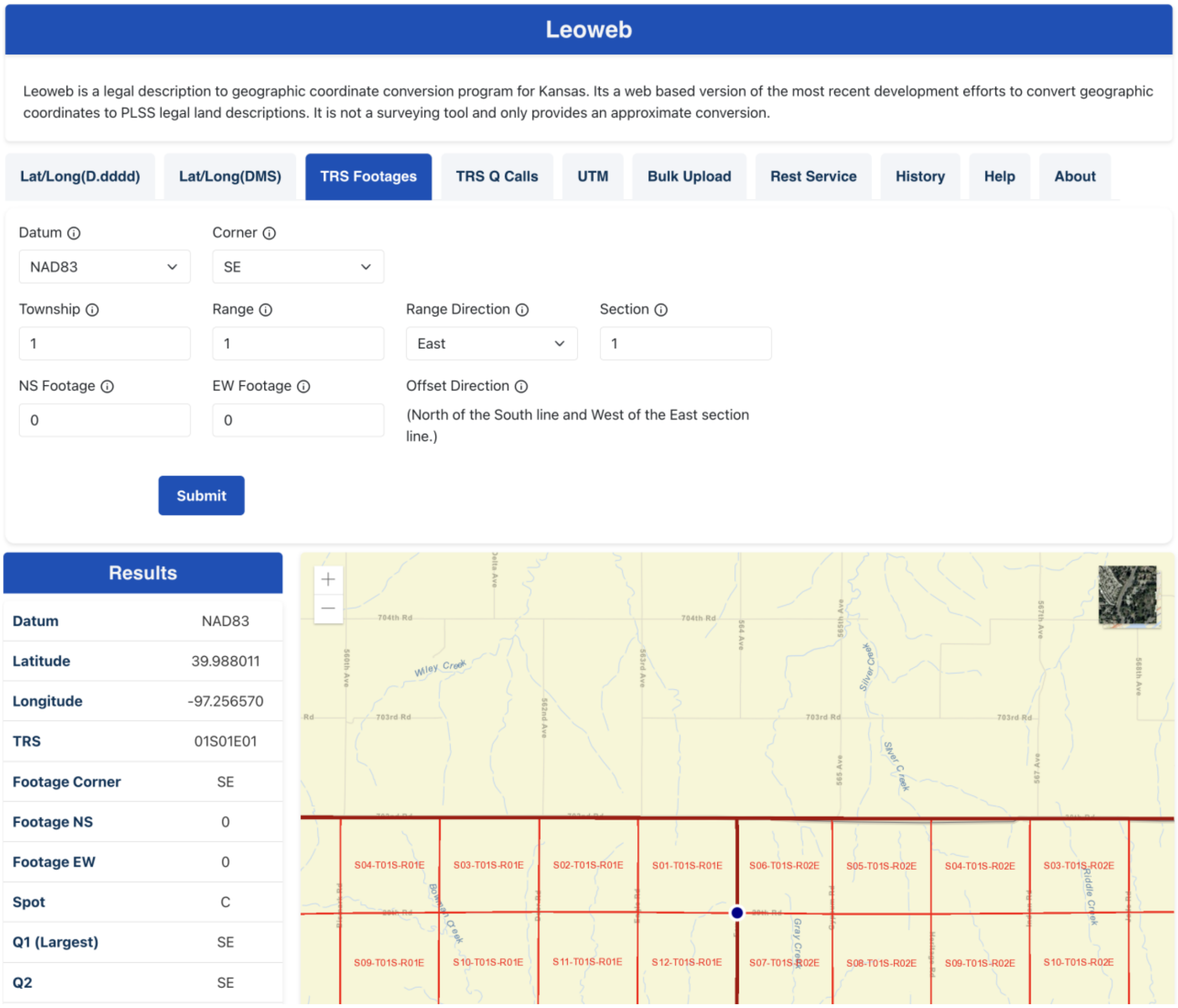


Figure 14. TRS and Offset Footage to Lat/Long.

Option 4 is selected when the user wants to convert TRS and footage information to latitude and longitude coordinates. The results are returned using the datum and reference corner specified in the dialog screen.

Enter Township Range Section and Quarter Calls:

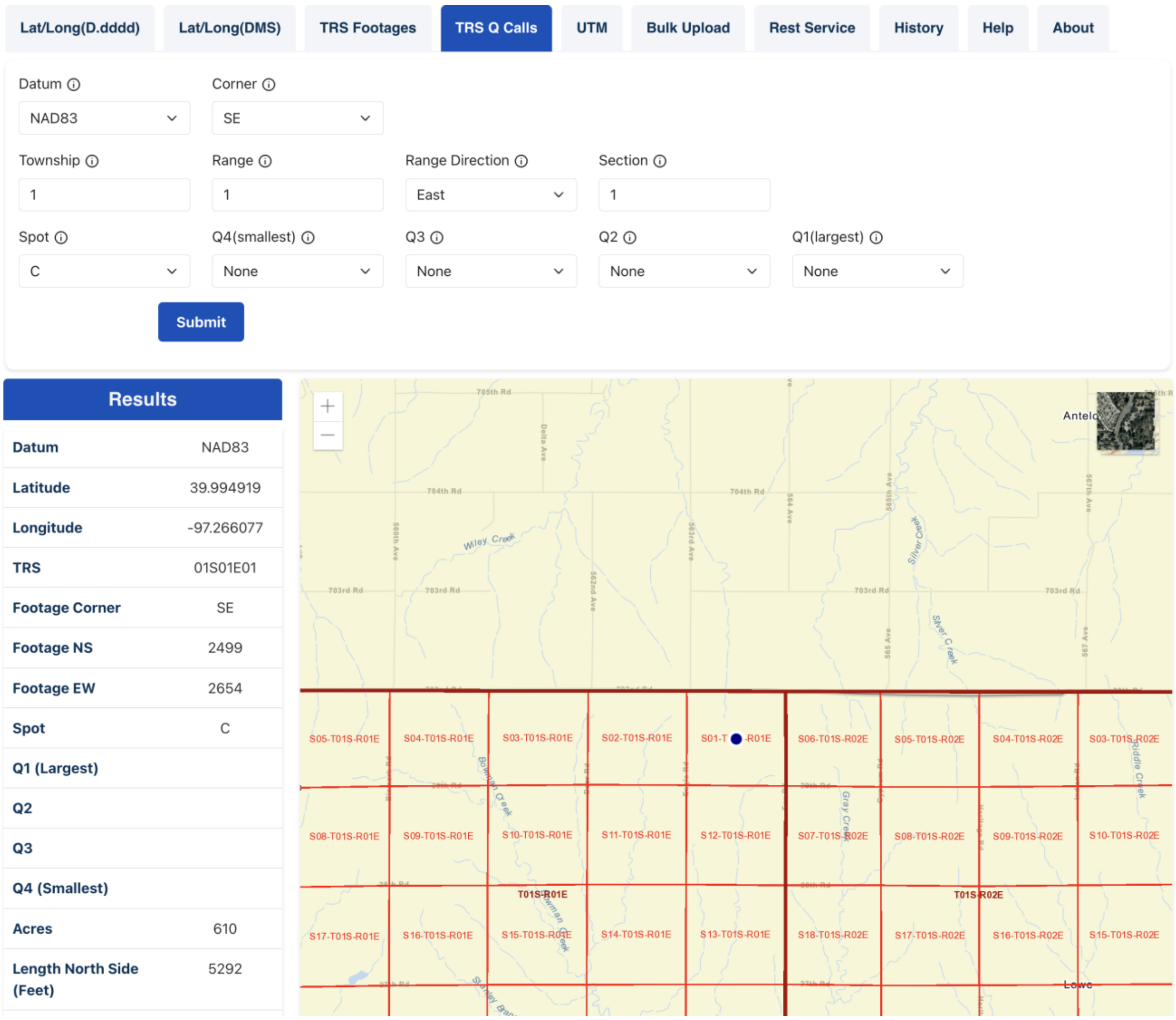


Figure 15. TRS and Quarter Calls to Lat/Long.

The input panel for converting TRS and quarter call data to latitude and longitude coordinates offers more selection tools than any other interactive option. The user has the ability to specify the quarter call subdivision level and which "nine point spot" to calculate. When using the "TRS and Quarter Calls to Lat/Long" option, the user may find selecting the quarter calls from the dropdown lists a bit awkward. In order to enforce data validation rules, the screen was intentionally designed to require the user to enter the largest subdivision first, using the most right list (Q1). The user then works "left to right," entering Q2, Q3, and Q4 as desired.

Enter UTM Northing Easting and Zone:

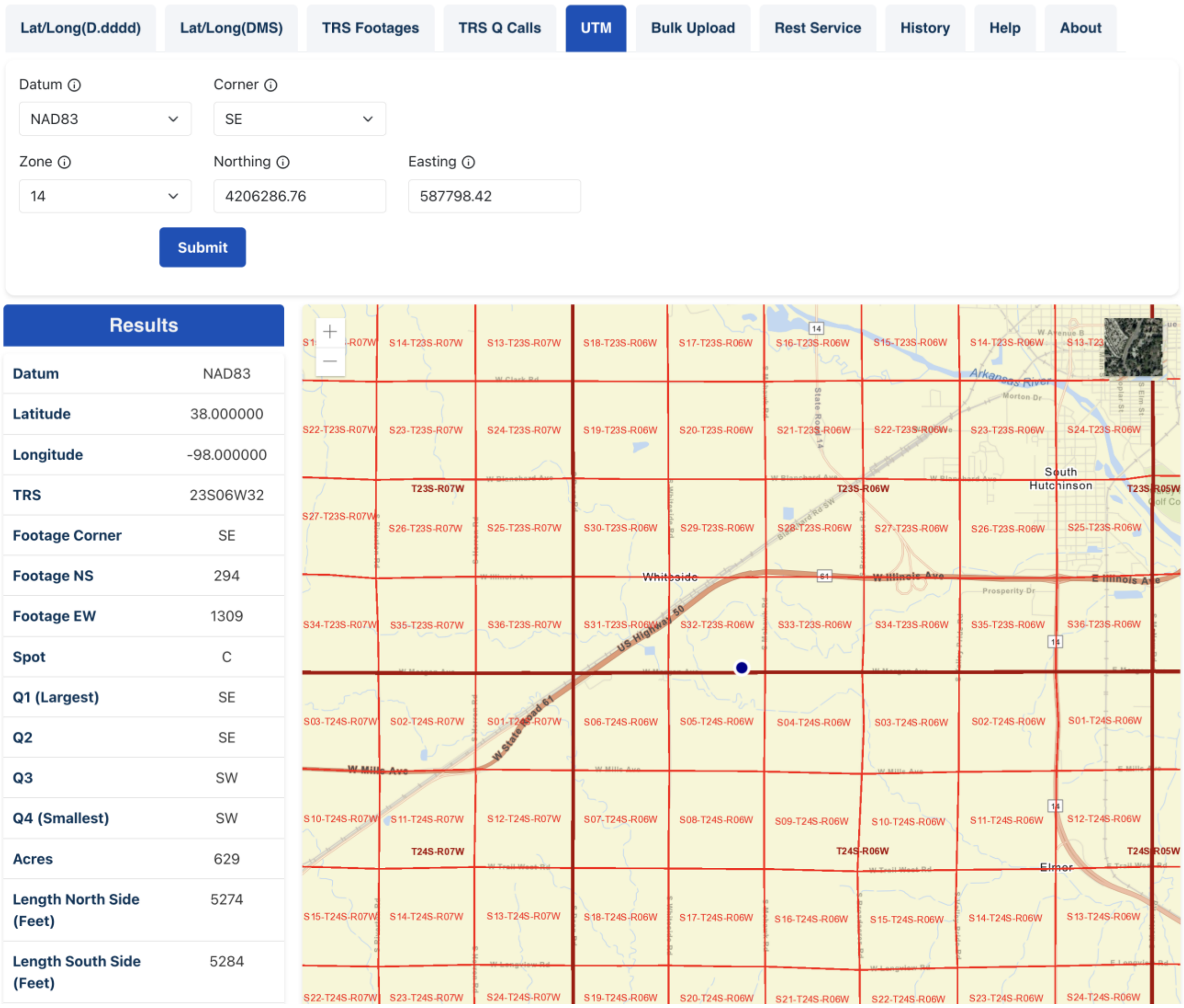


Figure 16. UTM Coordinates to Lat/Long.

When converting UTM coordinates, specifying the correct geodetic datum and UTM zone is important. Zone 14 is the default option, but the state of Kansas includes zones 13, 14, and 15. In this example the help feature for the "Zone" select dropdown has been activated by clicking on the word "Zone." (Help is available on all LEOWEB objects.)

Upload CSV file for bulk conversion:

Upload your CSV file, choose the appropriate conversion type, and enter your email, Datum, and Corner. The tool will process your file and email you the results. Sample CSV formats are available for download.

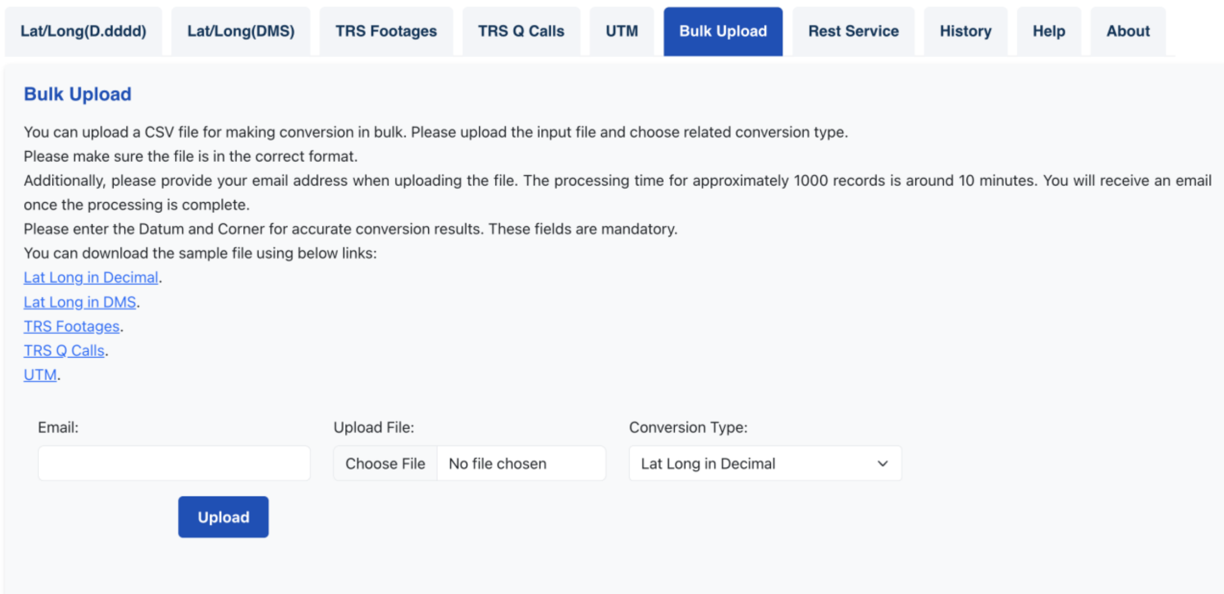
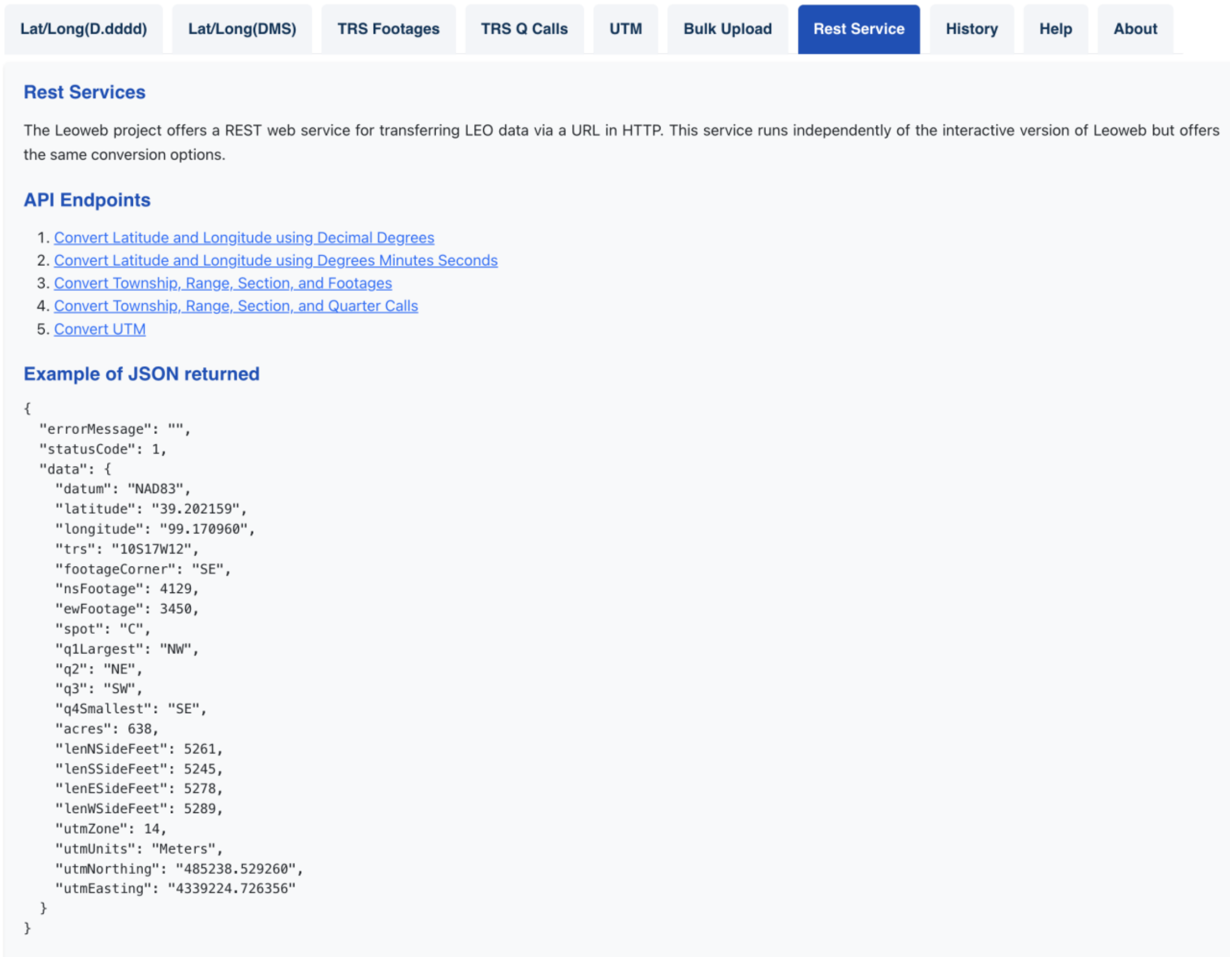


Figure 17. Bulk upload conversion service.



Lat/Long(D.dddd) | Lat/Long(DMS) | TRS Footages | TRS Q Calls | UTM | Bulk Upload | Rest Service | History | Help | About

Rest Services

The Leoweb project offers a REST web service for transferring LEO data via a URL in HTTP. This service runs independently of the interactive version of Leoweb but offers the same conversion options.

API Endpoints

1. Convert Latitude and Longitude using Decimal Degrees
2. Convert Latitude and Longitude using Degrees Minutes Seconds
3. Convert Township, Range, Section, and Footages
4. Convert Township, Range, Section, and Quarter Calls
5. Convert UTM

Example of JSON returned


```
{
  "errorMessage": "",
  "statusCode": 1,
  "data": {
    "datum": "NAD83",
    "latitude": "39.202159",
    "longitude": "99.170960",
    "trs": "10S17W12",
    "footageCorner": "SE",
    "nsFootage": 4129,
    "ewFootage": 3450,
    "spot": "C",
    "q1Largest": "NW",
    "q2": "NE",
    "q3": "SW",
    "q4Smallest": "SE",
    "acres": 638,
    "lenNSideFeet": 5261,
    "lenSSideFeet": 5245,
    "lenESideFeet": 5278,
    "lenWSideFeet": 5289,
    "utmZone": 14,
    "utmUnits": "Meters",
    "utmNorthing": "485238.529260",
    "utmEasting": "4339224.726356"
  }
}
```


Figure 18. Leoweb help page for REST service.

REST is an acronym for Representational State Transfer. It is a software architecture designed to permit distributed computing over networks. Because of its simplicity, it has become a popular method for running programs over the World Wide Web. Generally speaking, resources are accessed by making HTTP requests to a web service through the use of a URL (uniform resource locator). Resources are then made available or acted upon in a format that is appropriate for the resource. It could be a downloaded image, the delivery of an XML document, a JSON string or perhaps some other action, such as the execution of an automated process.

The LEOWEB offers a READ-ONLY service that performs the same coordinate conversions as the interactive tool. Each conversion method has its own URL format consisting of a resource name along with query parameters. The result of each API endpoint is a nested JSON derived from the LEO calculation engine.

All Conversion URLs:

The question mark (?) is used to separate the resource name from the query variables. For any conversion, use the resource name of "https://maps.kgs.ku.edu/leoweb/24_convertLLDEG" along with the query variables: Datum, Latitude, Longitude, and Corner etc. Values are assigned to each variable using the equal sign (=). The ampersand (&) is used to separate variables. i.e., Datum=NAD83&Latitude=38.1&Longitude=-98.1&Corner=SE

1. – Example of URL for converting decimal degree latitude and longitude.

https://maps.kgs.ku.edu/leoweb/24_convertLLDEG?Datum=wgs84&Latitude=+39&Longitude=-101.1&Corner=SE

- Be sure to include a negative sign (-) for longitude value

2. – Example of URL for converting Degrees Minutes Seconds latitude and longitude.

https://maps.kgs.ku.edu/leoweb/24_convertLLDMS?Datum=NAD27&LatDeg=38&LatMin=0&LatSeconds=&LonDeg=-96&LonMin=56&LonSeconds=&Corner=SESE

- Be sure to add -ve sign for LonDeg value
- LatSeconds &LonSeconds can be left empty and will be considered as a default 0 E.g., LatSeconds=&LonSeconds= or by not mentioning the LatSeconds and LonSeconds variables in the URL at all. E.g., http://127.0.0.1:58696/convertLLDMS?Datum=NAD27&LatDeg=38&LatMin=0&LonDeg=-96&LonMin=56&Corner=SE

3. – Example of URL for converting Township, Range, Section, and Footage Offsets.

https://maps.kgs.ku.edu/leoweb/24_convertTRSFT?Datum=NAD83&TRS=10S17W12&Corner=SE&FootageNS=0&FootageEW=0

- The QCalls accepts a string of Q1, Q2, Q3 and Q4 values all together. Any of the individual QCalls will be considered 'None' if left empty
- E.g., QCalls=&Corner=SE
- QCalls=SE&Corner=SE
- QCalls=SENW&Corner=SE
- QCalls=SENESW&Corner=SE

4. – Example of URL for converting Township, Range, Section, and Quarter Calls.

https://maps.kgs.ku.edu/leoweb/24_convertTRSQC?Datum=NAD83&TRS=10S17W12&QCall=NWNESWSE&Spot=C&Corner=SE

5. – Example of URL for converting UTM coordinates.

https://maps.kgs.ku.edu/leoweb/24_convertUTM?Datum=NAD83&Northing=4206286.76&Easting=587798.42&Corner=SE&zone=14

Appendix A

LEOWEB REST Service:

Example of JSON returned


```
{
  "errorMessage": "",
  "statusCode": 1,
  "data":{
           datum: "NAD83",
           latitude: "39.202159",
           longitude: "99.170960",
           trs: "10S17W12",
           footageCorner: "SE",
           nsFootage: 4129,
           ewFootage: 3450,
           spot: "C",
           q1Largest: "NW",
           q2: "NE",
           q3: "SW",
           q4Smallest: "SE",
           acres: 638,
           lenNSideFeet: 5261,
           lenSSideFeet: 5245
           lenESideFeet: 5278,
           lenWSideFeet: 5289,
           utmZone: 14,
           utmUnits: "Meters",
           utmNorthing: "485238.529260",
           utmEasting: "4339224.726356",
        }
}
```


Example of JSON returned when an error occurs.


```
{
  "errorMessage": "Corner should be NW, NE, SE or SW only. Given corner value is "S"",
  "statusCode": 3,
  "data": {}
}
```

Appendix B

Breakdown of JSON response format

1. **Status Code:** A number representing the status of the API request. Each scenario is assigned a unique status code such as every time the request is successful and the ‘data field contains a JSON output, the status code is 1.

Status Code	Type of Error
1	Successful request
2	Missing input variable in URL
3	An error occured
4	Datum or Corner invalid
5	Latitude or Longitude invalid
6	Invalid LatDeg/LatMin /LatSec or LonDeg/LonMin/LonSec
7	TRS value invalid
8	Footages invalid
9	QCalls invalid
10	Spot invalid
11	Zone invalid
12	Northing or Easting invalid

2. **Error Message:** This provides a descriptive message about the status of the API request. If the request is successful the “ErrorMessage” field is empty in the JSON response. Every time the API request fails due to incorrect values passed this field displays a descriptive information about the parameter causing the error.

3. **Warning:** This field along with its warning message occurs only when the input value of FootageNS or FootageEW field exceeds 1 mile of distance i.e 5280 feet.

4. **Data:** This field displays a JSON formatted string containing all the output attributes associated to the Leoweb’s conversion type. This field returns a JSON output only when the API request is successful. Every time there is an error in the API request this field is returned empty.

<table>
<tr><th>STATUS CODE</th><th colspan="2">ERROR MESSAGE</th><th>DATA</th></tr>
<tr><td>1</td><td colspan="2">-</td><td>"data":{
datum: "NAD83",
latitude: "39.202159",
longitude: "99.170960",
trs: "10S17W12",
footageCorner: "SE",
nsFootage: 4129,
ewFootage: 3450,
spot: "C",
q1Largest: "NW",
q2: "NE",
q3: "SW",
q4Smallest: "SE",
acres: 638,
lenNSideFeet: 5261,
lenSSideFeet: 5245
lenESideFeet: 5278,
lenWSideFeet: 5289,
utmZone: 14,
utmUnits: "Meters",
utmNorthing: "485238.52",
utmEasting: "4339224.72",
}</td></tr>
<tr><td>2</td><td colspan="2">Missing input parameters
Expecting parameter1, parameter2 and so on based on conversion type</td><td>-</td></tr>
<tr><td>3</td><td colspan="2">An error occurred</td><td></td></tr>
<tr><td rowspan="2">4</td><td colspan="2">Datum should be NAD27, NAD83 or WGS84 only
Given datum value is "NAD40"</td><td rowspan="2">-</td></tr>
<tr><td colspan="2">Corner should be NW, NE, SE or SW only
Given corner value is "SQ"</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">NAD27</td><td>For NAD27 latitude should be between the range 36.999 to 40.0032 for Kansas
Given latitude value is " 59"</td><td rowspan="2">-</td></tr>
<tr><td>For NAD27 longitude should be between the range -102.052 to -94.584 for Kansas
Given longitude value is " 103"</td></tr>
</table>

	NAD83/WGS84	For NAD83 and WGS84 latitude should be between the range **36.9930** to **39.9890** for Kansas Given latitude value is "-35 "	
		For NAD83 and WGS84 longitude should be between the range **-102.052** to **-94.584** for Kansas Given longitude value is "101.1"	
6	NAD27	For NAD27 latitude degree i.e. **LatDeg** should be between the range **36** to **40** for Kansas Given LatDeg value is "308"	-
		For NAD27 longitude degree i.e. **LonDeg** should be between the range **-102** to **-94** for Kansas Given LonDeg value is "96"	
	NAD83/WGS84	For NAD83 and WGS84 latitude degree i.e. **LatDeg** should be between the range **36** to **39** for Kansas Given LatDeg value is "18"	
		For NAD83 and WGS84 longitude degree i.e. **LonDeg** should be between the range **-102** to **-94** for Kansas Given LonDeg value is "96"	
	Latitude Minutes i.e. **LatMin** should be within the range **0** to **59** Given LatMin value is "-58"		
	Latitude Seconds i.e. **LatSeconds** should be within the range **0** to **59** Given LatSeconds value is "-3"		
	Longitude Minutes i.e. **LonMin** should be within the range **0** to **59** Given LonMin value is "-50"		
	Longitude Seconds i.e. **LonSeconds** should be within the range **0** to **59** Given LonSeconds value is "-300"		
7	Length of **TRS** should be **8** characters eg:01S02E03 Given length of TRS is "7"		-
	Township should be within the range **1** to **35** Given township value is "50"		

	Township Direction should be **S** only Given township direction value is "w" W — For Range Direction **W** the value should be between the limit **1** to **43** Given range value is "97" E — For Range Direction **E** the value should be between the limit **1** to **25** Given range value is "97" **Range Direction** should be **E** or **W** only Given range direction value is "r" **Section** should be within the range **1** to **36** Given section value is "62"	
8	**FootageNS** should be within the range **0** to **5280** Given footageNS value is "7000" **FootageEW** should be within the range **0** to **5280** Given footageEW value is "6000"	-
9	**Q Calls** should be 2-character values each i.e. **NE**, **NW**, **SE**, **SW**, **E2**, **W2**, **N2**, **S2** only Given Qcall value is "S2NWN" **Q2** should be **NE**, **NW**, **SE**, **SW**, **E2**, **W2**, **N2**, **S2** only Given Q2 value is "N3" If Q1 is 1/2 subdivision then **Q2** must also be 1/2 subdivision i.e. **E2**, **W2**, **N2**, **S2** only Given Q2 value is "NW" **Q3** should be **NE**, **NW**, **SE**, **SW**, **E2**, **W2**, **N2**, **S2** only Given Q3 value is "NQ" If Q2 is 1/2 subdivision then **Q3** must also be 1/2 subdivision i.e. **E2**, **W2**, **N2**, **S2** only Given Q3 value is "NW" **Q4** should be **NE**, **NW**, **SE**, **SW**, **E2**, **W2**, **N2**, **S2** only Given Q4 value is "SA"	-

	If Q3 is 1/2 subdivision then **Q4** must also be 1/2 subdivision i.e. **E2**, **W2**, **N2**, **S2** only Given Q4 value is "SE"	
10	**Spot** should be **NE**, **NW**, **SE**, **SW**, **E**, **W**, **N**, **S** or **C** only Given spot value is "v"	-
11	**Zone** should be **13**, **14** or **15** only Given zone value is "16"	-
12	**Northing** should be between the range **4094999.9987** to **4435000.0067** Given northing value is "7190000"	-
	Easting should be between the range **747972.4826** to **767090.7706** Given easting value is "958910.96"	
	Northing should be between the range **4090000** to **4435000.0064** Given Northing value is "7190000	
	Easting should be between the range **232999.9949** to **767000.0051** Given easting value is "958910.96"	
	Northing should be between the range **4094999.9984** to **4435000.0063** Given Northing value is "7190000"	
	Easting should be between the range **232909.0649** to **364000** Given easting value is "958910.96"	

Python example of a program that calls the LEOWEB REST Service.

```
import requests

def get_data_from_api(latitude, longitude, corner, datum):
    # Endpoint URL
    endpoint_url = 'https://maps.kgs.ku.edu/leoweb/24_convertLLDEG'

    # Query parameters
    params = {
        'Datum': datum,
        'Latitude': latitude,
```

```
        'Longitude': longitude,
        'Corner': corner
    }

    # Make GET request to the API endpoint
    response = requests.get(endpoint_url, params=params)

    # Check if the request was successful (statusCode = 1)
    if response.status_code == 200:
        # Parse the JSON response
        data = response.json()

        # Check if the statusCode is 1 for a successful request
        if data.get('statusCode') == 1:
            nested_data = data.get('data', {})
           # Extract and print the data
            nested_data = data.get('data', {})
            for key, value in nested_data.items():
                print('{}: {}'.format(key, value))

        else:
            print('Error:', data.get('errorMessage'))

    else:
        print('Error:', response.text)

# Example usage
latitude = '39'
longitude = '-101.1'
corner = 'sE'
datum = 'wgs84'
get_data_from_api(latitude, longitude, corner, datum)



import requests

def get_data_from_api(datum, trs, corner, footage_ns, footage_ew):
    # Endpoint URL
    endpoint_url = 'https://maps.kgs.ku.edu/leoweb/24_convertTRSFT'

    # Query parameters
    params = {
        'Datum': datum,
        'TRS': trs,
        'Corner': corner,
        'FootageNS': footage_ns,
        'FootageEW': footage_ew
    }

    # Make GET request to the API endpoint
```

```
    response = requests.get(endpoint_url, params=params)

    # Check if the request was successful (statusCode = 1)
    if response.status_code == 200:
        # Parse the JSON response
        data = response.json()

        # Check if the statusCode is 1 for a successful request
        if data.get('statusCode') == 1:
            # Extract and print the data
            nested_data = data.get('data', {})
            for key, value in nested_data.items():
                print('{}: {}'.format(key, value))

            if 'Warning' in data:
                # Warning field exists only for the TRSFT conversion type
                print('{}: {}'.format('Warning', data.get('Warning')))

        else:
            print('{}: {}'.format('Error', data.get('errorMessage')) )

    else:
        print('Error:', response.text)

# Example usage
datum = 'NAD83'
trs = '1017W2'
corner = 'E'
footage_ns = '5944'
footage_ew = '0'
get_data_from_api(datum, trs, corner, footage_ns, footage_ew)
```

References:

Good, D. I., 1964, Mathematical conversion of section, township, and range notation to Cartesian coordinates: State Geological Survey of Kansas, Bulletin 170, pt. 3, 30 p.

Morgan, C. O., and McNellis, M. J., 1969, FORTRAN IV Program, KANS, for the Conversion of General Land Office Locations to Latitude and Longitude Coordinates: State Geological Survey of Kansas, Special Distribution Pub. 42, 24 p.

Thompson,M. M., 1979 Maps for America: U.S. Geological Survey, p. 252.

Ross, C. G., 1989, LEO-conversion between legal and geographic reference systems in Kansas: Kansas Geological Survey, Open-file Report 89-10, 9 p.

Suchy, D. R., 2002, The public land survey system in Kansas: Kansas Geological Survey, Public Information Circular 20, 4 p.; available online at http://www.kgs.ku.edu/Publications/pic20/pic20_1.html (accessed 06/27/2012).

Gagnon, G. F., 2008, LEO Version 7.0 User Manual: Kansas Geological Survey, Open-file Report 2008-24, 20 p.; available online at http://www.kgs.ku.edu/software/LEO/LEO_07_UserManual_OFR_2008-24.pdf (accessed 06/27/2012).

Oracle® Application Express Installation Guide, Release 4.1, Part Number E21673-04 ,http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21673/pre_require.htm#autoId2 , (accessed 06/27/2012).