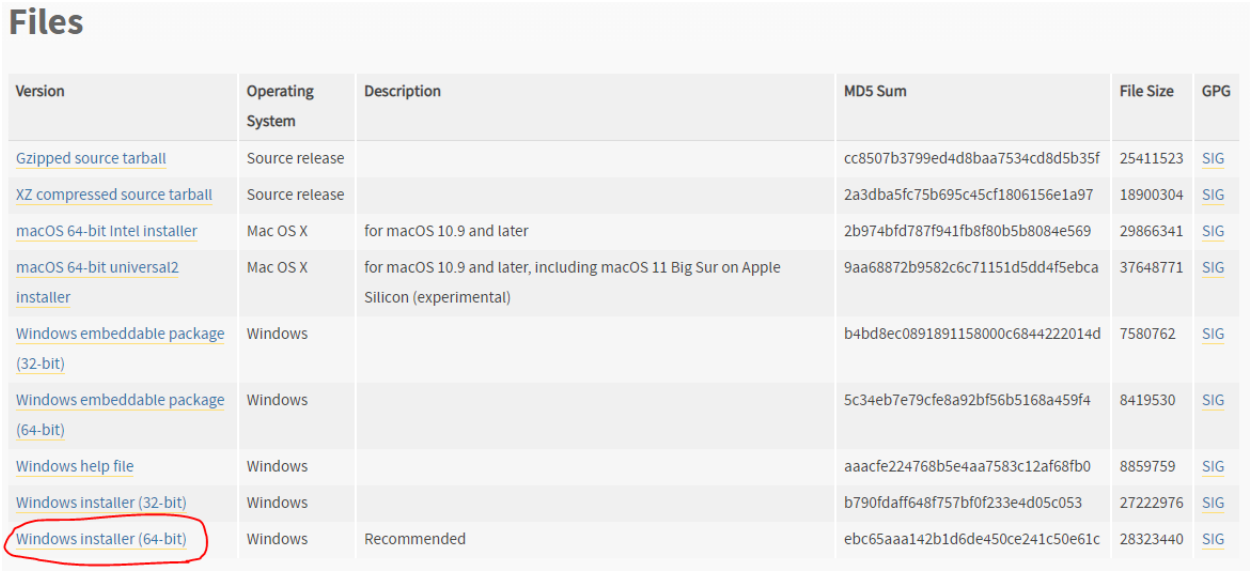
IllCAT ReadMe (IllCAT Version 1.06) (GUI Version 1.11)

Overview

IllCAT is a python script that allows for transforming between coordinates in NAD83-based Illinois legacy projections and coordinates in NAD83-based Illinois systems without NGS sanctioning or other federal backing. Support for Illinois East and West, the 2002 full-state projection, Hutson's IDOT District 8 LDPs, and all ICS83 low distortion projections (LDPs) are included by default. The Python code has been crafted to be extensible, meaning that adding further Transverse Mercator and Lambert Conformal Conic (1-parallel) projections is a relatively trivial procedure. A graphical user interface (IllcatGUI), has been developed to make interacting with the IllCAT script more intuitive for regular users. The IllCAT script should work with any operating system if Python version 3 or later is installed on that system. The IllcatGUI tool is only compatible with Microsoft Windows.

Installation and Setup

1. If Python 3 is not already installed on your computer, open your web browser and navigate to https://www.python.org/downloads/windows/. Click on the link to the latest release, download the installer executable from the website, and run it to install Python on your machine. Any version of Python 3 should be sufficient.

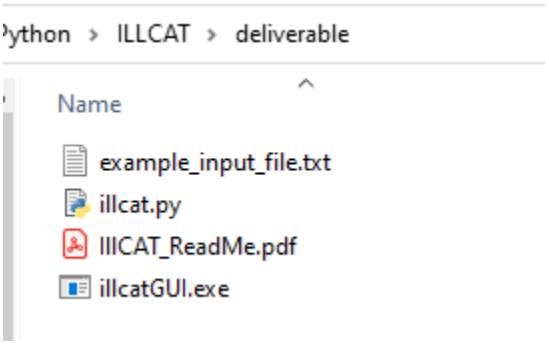


Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		cc8507b3799ed4d8baa7534cd8d5b35f	25411523	SIG
XZ compressed source tarball	Source release		2a3dba5fc75b695c45cf1806156e1a97	18900304	SIG
macOS 64-bit Intel installer	Mac OS X	for macOS 10.9 and later	2b974bfd787f941fb8f80b5b8084e569	29866341	SIG
macOS 64-bit universal2 installer	Mac OS X	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	9aa68872b9582c6c71151d5dd4f5ebca	37648771	SIG
Windows embeddable package (32-bit)	Windows		b4bd8ec0891891158000c6844222014d	7580762	SIG
Windows embeddable package (64-bit)	Windows		5c34eb7e79cfe8a92bf56b5168a459f4	8419530	SIG
Windows help file	Windows		aaacfe224768b5e4aa7583c12af68fb0	8859759	SIG
Windows installer (32-bit)	Windows		b790fdaff648f757bf0f233e4d05c053	27222976	SIG
Windows installer (64-bit)	Windows	Recommended	ebc65aaa142b1d6de450ce241c50e61c	28323440	SIG

2. To get IllCAT working, add Python3 to your Windows environment PATH variable, install the "py" python launcher, or do both.
3. You should have received a zip file containing the following files:
    a. "illcat.py" : This is the heart of the application, the ILLCAT command line tool
    b. "illcatGUI.exe" : The windows executable for the graphical user interface that interfaces with illcat.py

c. “IllCAT_ReadMe.pdf” : This readme document



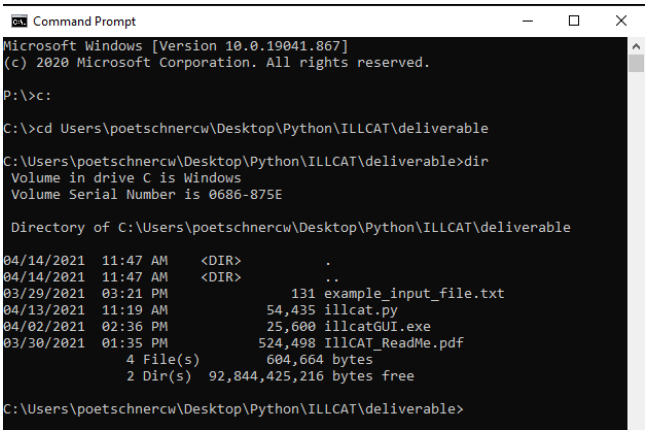
d. Several example files

4. Unzip these files to a directory of your choice on your computer. Note that “illcatGUI.exe” and “illcat.py” must share the same containing folder for the former to function.
5. IllcatGUI can be run by double-clicking it like any other application. IllCAT (the Python script) should be run from the command line.

Using illcat.py

If you have no interest in learning how to use the IllCAT Python script directly, skip ahead to the IllcatGUI usage section.

1. Getting started
    a. On Microsoft Windows open your preferred terminal emulator (e.g. “Command Prompt”). One way to do this is to open the Windows start menu, type “cmd”, and press enter.
    b. Open the drive containing the ILLCAT directory. Nearly everyone has the C drive as their main Windows drive where they store everything. In that case simply type “c:” (without the quotes) into the command prompt and press enter.
    c. Navigate to the directory where you unzipped all the IllCAT files by entering the “cd” command like “cd Path\To\IllCAT\Directory”.
    d. Confirm that you have navigated to the correct folder by entering the “dir” command. A list of files should pop-up that includes the IllCAT files.
    e. It is possible to set up easier ways to open command prompts pre-navigated to directories that are already open in Windows Explorer, but that is beyond the scope of this readme.



2. IllCAT Usage Basics
    a. Once you are in the correct folder in your terminal emulator, you should be able to run IllCAT by entering “python illcat.py” or “py illcat.py”. This readme will assume that you invoke your Python 3 installation with the “python” command.

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>python illcat.py
usage: ILLCAT [-h] [--version | -l] {coord,file} ...

C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

    b. As you can see, IllCAT needs to know more about what you want it to do before it can run. Try getting some help by appending the “-h” or “—help” flag to the command.

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>py illcat.py -h
usage: ILLCAT [-h] [--version | -l] {coord,file} ...

Transform between NAD83(2011) lat/longs, Illinois State Plane coordinates,
IDOT District 8 LDP coordinates, and ICS83 LDP coordinates. The datum for all
geospatial data should be NAD83(2011). Transformations are accurate up to 50
decimal places, thanks to Python's Decimal module and this precision is
adjustable by the user.

positional arguments:
  {coord,file}          ILLCAT has two sub-commands, "coord" and "file"
                        representing its two operational modes: single
                        coordinate pair transformation and coordinate file
                        transformation respectively. Enter "ILLCAT coord -h"
                        or "ILLCAT file -h" for help with each sub-command.

optional arguments:
  -h, --help            show this help message and exit
  --version             print the version of this IllCAT script and exit.
  -l, --listprojections
                        print a list of the IllCAT coordinate system IDs of
                        the available projections and exit.

This script was coded by Christopher Poetschner under contract to IDOT, Anno
Domini MMXX-MMXXI

C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

    c. We can see from the above help text, that the functionality of IllCAT is divided into two subcommands: “coord” which is used to input a single point for transformation and “file” which is used to input a file containing multiple points for transformation. Explore the single point subcommand by entering “python illcat.py coord -h”.

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>py illcat.py coord -h
usage: ILLCAT coord [-h] -c <y,x> [-v] [-w] [-p <5-50>] [-r <0-50>] [-i <coordsys ID
>] [-o <coordsys ID>] [-iu <meters|intlft|ussurveyft>] [-ou <all|meters|intlft|ussur
veyft>]

arguments:
  -c COORDS, --coords COORDS, --yx COORDS
                        input horizontal coordinates to be converted in the
                        following format "lat,long" or "northing,easting".
                        For lat/long input, decimal degree or degree-minute
                        -second formats are acceptable. In degree-minute
                        -second format, latitudes should be formatted as
                        DD-MM-SS.ssssss... and longitudes as
                        DDD-MM-SS.ssssss..., with leading zeroes as
                        necessary. There should be no spaces.
  -v, --verbose         enable verbose output
  -w, --positivewest    enable positive westing decimal degree I/O mode
  -p {5-50}, --precision {5-50}
                        set decimal precision for the conversion calculations.
                        Maximum allowed precision is 50 and the minimum is 5.
                        (Default: 50)
  -r {0-50}, --rounding {0-50}
                        set the rounding for output display. Maximum
                        allowed rounding is 50 and the minimum is 0.
                        (Default: 8)
  -i COORDSYSID, --incoordsystem COORDSYSID
                        specify the coordinate system of the input coordinates
                        (Default: NAD83_2011_Lat_Long)
  -o COORDSYSID, --outcoordsystem COORDSYSID
                        specify the coordinate system of the output
                        coordinates (Default: NAD83_2011_Lat_Long)
  -iu {meters,intlft,ussurveyft}, --inputunits {meters,intlft,ussurveyft}
                        specify the units of the input values. Lat/long input
                        ignores this. (Default: meters)
  -ou {all,meters,intlft,ussurveyft}, --outputunits {all,meters,intlft,ussurveyft}
                        specify the units of the output values. Lat/long
                        output ignores this. (Default: all)

optional arguments:
  -h, --help  show this help message and exit

C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

d. The help output explains the different flags that control the operation of IllCAT's single point mode. Now look at the help output for the file subcommand.

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>py illcat.py file -h
usage: ILLCAT file [-h] -f <path to coordinates file> [-of <output file path>] [-v]
 [-w] [-p <5-50>] [-r <0-50>] [-i <coordsys ID>] [-o <coordsys ID>] [-iu <meters|in
tlft|ussurveyft>] [-ou <meters|intlft|ussurveyft>]

arguments:
  -f COORDSFILE, --coordsfile COORDSFILE, --xyfile COORDSFILE
                        path to a plain text file, where each line represents
                        a coordinate pair to be transformed. The acceptable
                        format for a line is the same as for an NCAT input
                        file: "<Any kind of ID>,<lat or northing>,<lon or
                        easting>"
  -of OUTFILE, --outfile OUTFILE
                        path to the output file to be created, where each line
                        represents a transformed line from the input file
  -v, --verbose         enable verbose output
  -w, --positivewest    enable positive westing decimal degree I/O mode
  -s, --stdout          redirect output to stdout. Overrides -of switch.
  -p {5-50}, --precision {5-50}
                        set decimal precision for the conversion calculations.
                        Maximum allowed precision is 50 and the minimum is 5.
                        (Default: 50)
  -r {0-50}, --rounding {0-50}
                        set the rounding for file output. Maximum
                        allowed rounding is 50 and the minimum is 0.
                        (Default: 8)
  -i COORDSYSID, --incoordsystem COORDSYSID
                        specify the coordinate system of the input coordinates
                        (Default: NAD83_2011_Lat_Long)
  -o COORDSYSID, --outcoordsystem COORDSYSID
                        specify the coordinate system of the output
                        coordinates (Default: NAD83_2011_Lat_Long)
  -iu {meters,intlft,ussurveyft}, --inputunits {meters,intlft,ussurveyft}
                        specify the units of the input values. Lat/long input
                        ignores this. (Default: meters)
  -ou {meters,intlft,ussurveyft}, --outputunits {meters,intlft,ussurveyft}
                        specify the units of the output values. Lat/long
                        output ignores this. (Default: meters)

optional arguments:
  -h, --help  show this help message and exit

C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

e. As you can see the functionality is almost the same, with nearly all the same control flags as the coord subcommand. However, instead of inputting a single coordinate pair with the "-c" flag, you input the path to the multi-point input file with the "-f" flag. You can also optionally specify the exact location of the converted output file with the "-of" flag.

3. IllCAT Practical Example (Single Point Transformation)
   a. Pretend that you have a point in IL State Plane West (NAD832011, US Survey Foot) that you know is in Sangamon county and you want to transform this point to the ICS83 Sangamon County projection (Springfield) in meters. Input northing is 1081584.0396661535 usft and easting is 2419587.72738858 usft.

   b. The following command will tell IllCAT to perform the transformation: "python illcat.py coord -c 1081584.0396661535,2419587.72738858 -i NAD83_2011_1202_IL_West -iu ussurveyft -o ICS83_22_Springfield -ou meters".

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>python illcat.py coord -c 1081584.0396661535,2419587.72738858 -i NAD83_2011_1202
_IL_West -iu ussurveyft -o ICS83_22_Springfield -ou meters
Output (ICS83_22_Springfield) (Meters):
867306.33247771 N , 703020.88326589 E

C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

c. One can get the IDs associated with all the projections that IllCAT knows by entering the command “python illcat.py -l” or “python illcat.py --listprojections”.

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>py illcat.py -l
NAD83_2011_Lat_Long
NAD83_2011_1201_IL_East
NAD83_2011_1202_IL_West
ICS83_01_Freeport
ICS83_02_Rockford
ICS83_03_Aurora
ICS83_04_Chicago
ICS83_05_Moline
ICS83_06_Sterling
ICS83_07_Ottawa
ICS83_08_Joliet
ICS83_09_Monmouth
ICS83_10_Galesburg
ICS83_11_Peoria
ICS83_12_Eureka
ICS83_13_Bloomington
ICS83_14_Pontiac
ICS83_15_Watseka
ICS83_16_Quincy
ICS83_17_Macomb
ICS83_18_Lincoln
ICS83_19_Decatur
ICS83_20_Champaign
ICS83_21_Jacksonville
ICS83_22_Springfield
ICS83_23_Charleston
ICS83_24_Jerseyville
ICS83_25_Carlinville
ICS83_26_Taylorville
ICS83_27_Effingham
ICS83_28_Robinson
ICS83_29_Belleville
ICS83_30_Mount_Vernon
```

d. The rounding of the input coordinates is 10 decimal places. By default, IllCAT rounds output to 8 decimal places. Let’s increase the rounding by adding in the “-r” flag to get a result that better corresponds to the input: “python illcat.py coord -c 1081584.0396661535,2419587.72738858 -i NAD83_2011_1202_IL_West -iu ussurveyft -o ICS83_22_Springfield -ou meters -r 10”.

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>python illcat.py coord -c 1081584.0396661535,2419587.72738858 -i NAD83_201
1_1202_IL_West -iu ussurveyft -o ICS83_22_Springfield -ou meters -r 10
Output (ICS83_22_Springfield) (Meters):
867306.3324777131 N , 703020.8832658935 E

C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

e. Let us now use these results to reverse the transformation and see if we get our initial input in step 3b. The command for this is “python illcat.py coord -c 867306.3324777131,703020.8832658935 -o NAD83_2011_1202_IL_West -ou ussurveyft -i ICS83_22_Springfield -iu meters -r 10”:

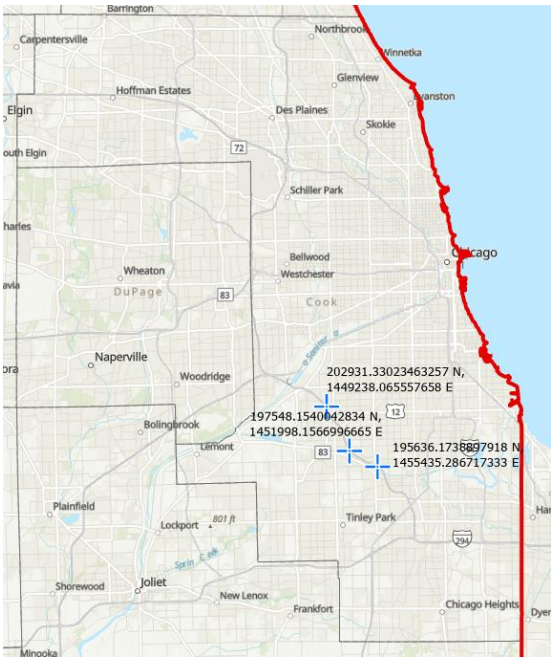
```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>python illcat.py coord -c 867306.3324777131,703020.8832658935 -o NAD83_201
1_1202_IL_West -ou ussurveyft -i ICS83_22_Springfield -iu meters -r 10
Output (NAD83_2011_1202_IL_West) (US Survey Ft):
1081584.0396649566 N , 2419587.7273885806 E

C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

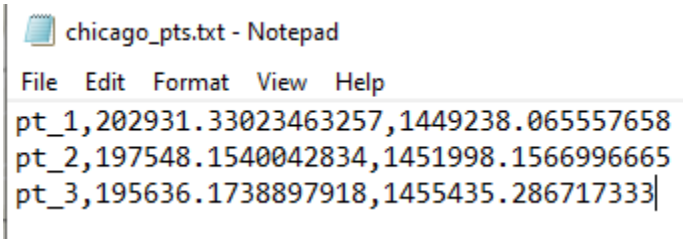
f. As you can see there is an infinitesimal change in the northing, a loss of accuracy, but the easting stays the same as far as we can see with this rounding. This was a transverse Mercator to Lambert Conformal Conic transformation. Notice how the

order of the flags does not matter; they need only be present and not all the flags must necessarily be used all the time. There are default settings for rounding, calculation precision, output verbosity, etc.

4. IllCAT Practical Example (Multiple Points Transformation)
    a. Let us now see how to transform multiple points at once. Imagine that we have three points projected in ICS83 Chicago LDP (NAD832011, Meters) in Cook county and we want to keep these points in ICS83 Chicago LDP but convert their units from meters to international feet:
        i. 202931.33023463257 N, 1449238.065557658 E
        ii. 197548.1540042834 N, 1451998.1566996665 E
        iii. 195636.1738897918 N, 1455435.286717333 E



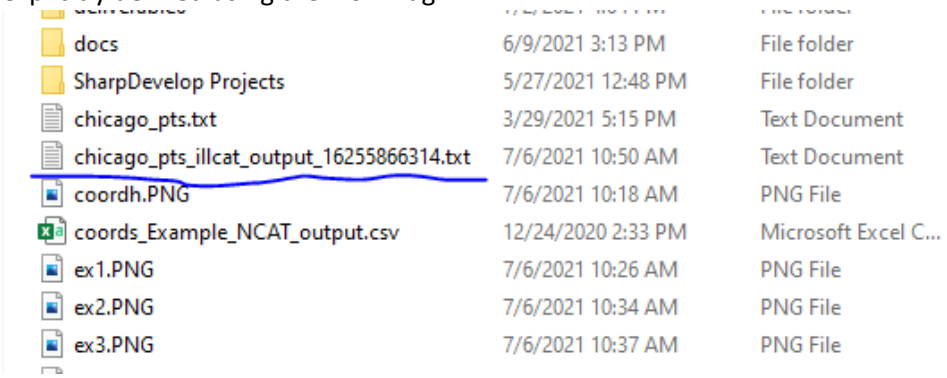
    b. Open notepad or a different text editor of your choice, enter its contents as below, and save it to a file called “chicago_pts.txt” in your IllCAT directory:



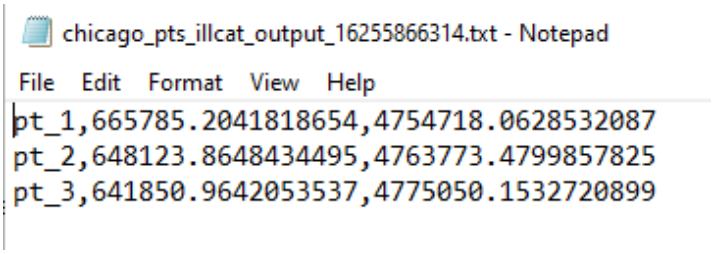
    c. The format for each line of an IllCAT input file is <label>,<northing>,<easting>,<any number of additional, comma-separated columns that affect nothing but are preserved>.
    d. Transform the points in the file you just created by invoking the following command: “python illcat.py file -f chicago_pts.txt -i ICS83_04_Chicago -o ICS83_04_Chicago -ou intlft -r 10”.

```
C:\Users\poetschnercw\Desktop\Python\ILLCAT>python illcat.py file -f chicago_pts.txt -i ICS83_04_Chicago -o ICS83_04_C
hicago -ou intlft -r 10
Coordinates transformed: (3/3)
C:\Users\poetschnercw\Desktop\Python\ILLCAT>
```

e. Since we did not explicitly designate a path to an output file, IllCAT automatically generates the output file in the script's local folder, with a name an extension based on those of the input file. Note that the output file's location and name can be explicitly defined using the "-of" flag.

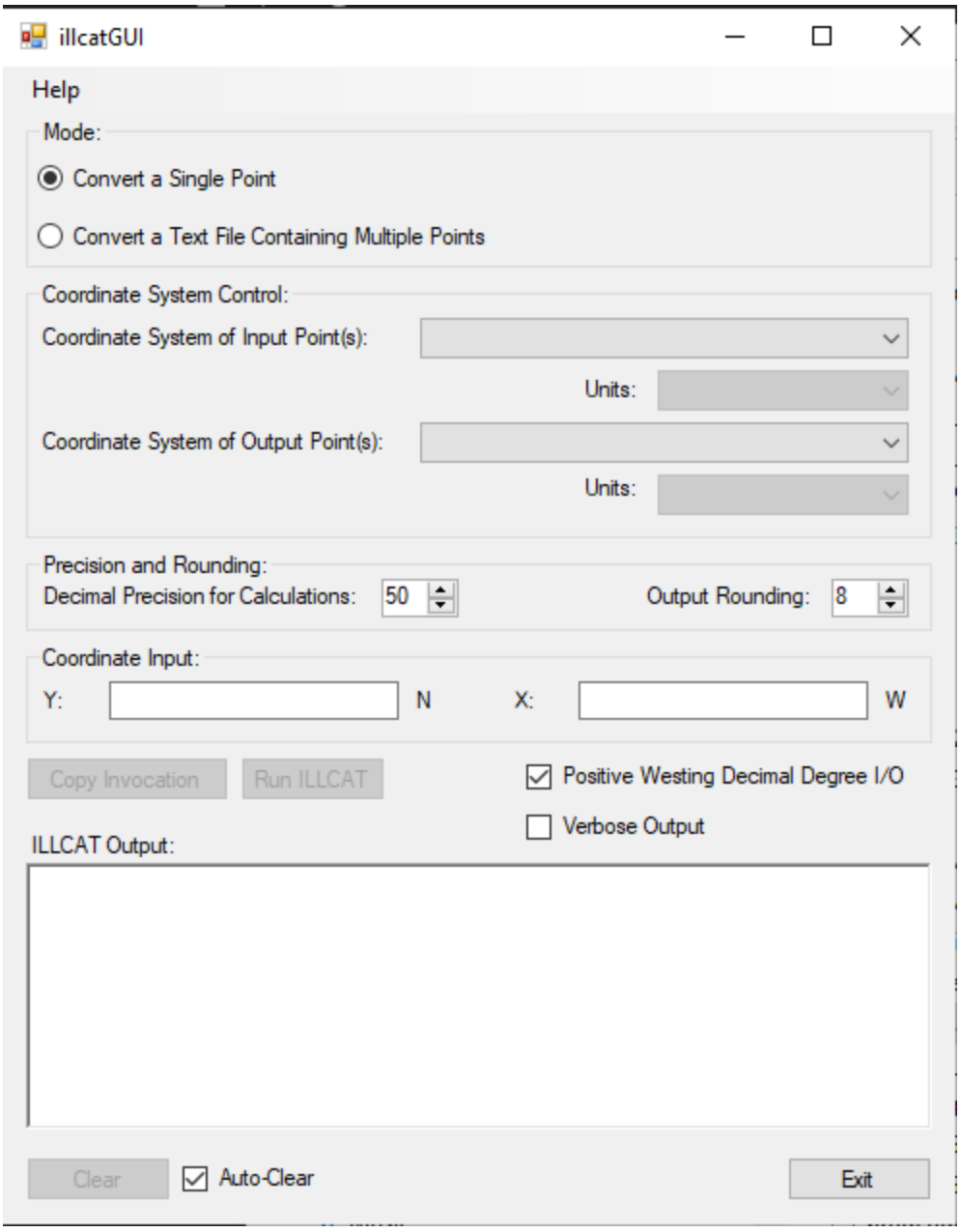


f. This file contains the three points, with labels preserved, converted to international feet, but still in the Chicago LDP projection.

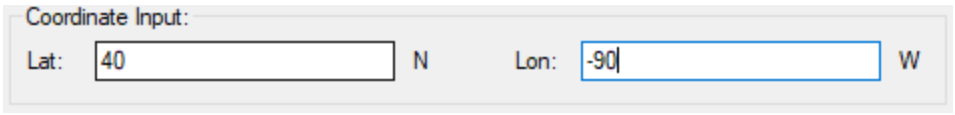


Using IllcatGUI

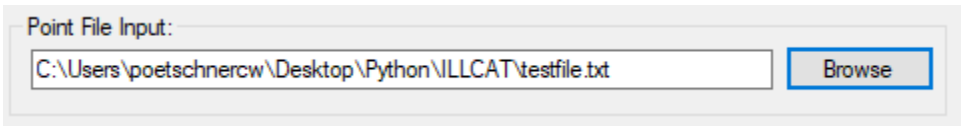
As stated earlier, the IllcatGUI program is a simple front-end for the IllCAT Python script that gives the user access to nearly every feature of the IllCAT script, without having to deal with manually constructing terminal commands. Currently, the only features it does not exploit are the automation and command piping possibilities of the script. The picture below shows the IllcatGUI as it appears before the user has adjusted any of the controls.



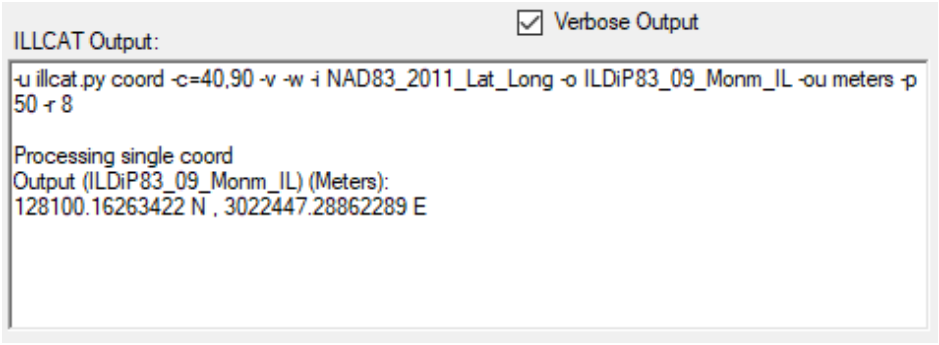
1. On the very top of the interface in the “Mode” section the user can select between single point and multi-point file input modes.
2. Below that in the “Coordinate System Control” section the user can select the input and output coordinate systems and units.
3. Further down in the “Precision and Rounding” section, the user may adjust calculation precision and output rounding. It’s fine to leave these as-is. Might want to change rounding to match your needs.
4. The next section changes depending on the operational mode.
    a. In single point conversion mode, input the northing and easting of the point you are converting.



b. In file conversion mode, click the “Browse” button and use the file browser to designate the file containing the points you wish to convert.



5. The large blank output area displays output from the IllCAT script. The user may type in this region to notate results. Right-clicking this region pops a menu with some alternate text-editing functions.



6. Clicking the “Copy Invocation” button will copy the command equivalent to the current configuration of the IllcatGUI to the clipboard. This can be useful for learning command-line interface usage of IllCAT.
7. Clicking “Run ILLCAT” will run the conversion you have set up in the interface. The button is disabled unless all required input parameters have been specified.
8. The “Clear” button clears the output area.
9. The “Positive Westing Decimal Degree I/O” checkbox is enabled by default and maps to the “-w” flag of the IllCAT script. When enabled, inputted positive longitude values are assumed to be westing longitudes and all outputted longitudes will also be in positive west.
10. The “Verbose Output” checkbox signals that you want to increase the verbosity of the output from the IllCAT script and render the invocation in the output area.
11. Checking the “Auto-Clear” checkbox tells the program to clear the output box between IllCAT runs. Otherwise, output accumulates, and you can scroll through it with the scrollbar on the right-hand side.
12. IllcatGUI has some basic instructions built-in, accessible by navigating to “Help>About”.
13. The interface is resizable.