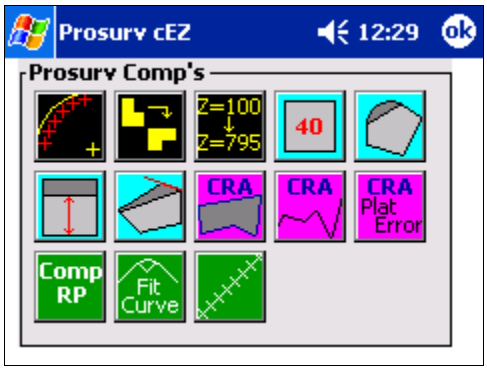
cEZ Comps



The Prosurv cEZ Comps button contains 13 powerful routines:

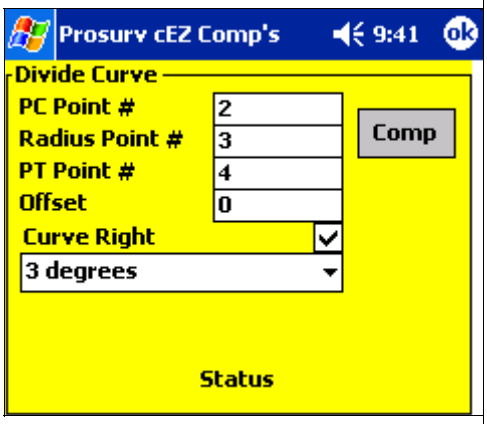
- **Divide Curve**
- **Translate and Rotate**
- **Translate Elevations**
- **Polygon Area**
- **Segment (Curve) Area**
- **Pre-determined Sliding Line Area**
- **Pre-determined Hinged Area**
- **Closed Traverse Compass Rule Adjustment**
- **Connecting Traverse Compass Rule Adjustment**
- **Distribute Plat Error (Compass Rule)**
- **Compute a Radius Point based on 3 points**
- **Fit Curve to Two Tangents (Curb Returns)**
- **Divide Line 2D/3D**

Note: The first time you use a Comps function, it may take a few seconds to load.

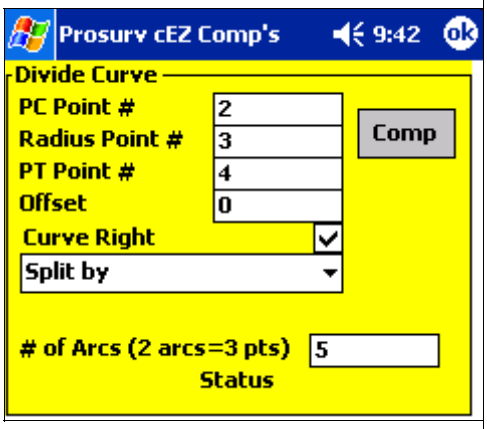
Divide Curve

The divide curve routine lets you divide a curve several different ways. It can even create Offset points (offset from the curve). Options include:

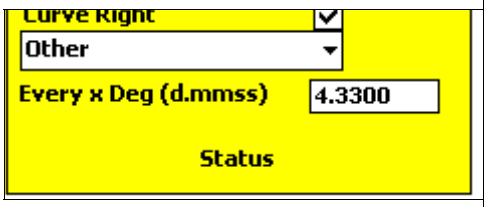
- Every 3 degrees
- 2 degrees
- 1 degree
- 1/2 degree
- 1/3 degree
- 1/4 degree
- Split by
- Other



If you select **Split by**, then you can enter a number of divisions. The example shows 5 total arcs, which creates 6 points.



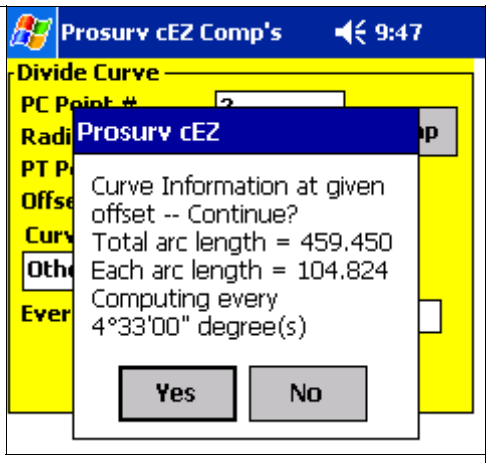
If you select **Other**, you can enter any Degree amount in D.MMSS format. In this example, a point will be computed every 4°33'00".



Using the last example, pressing the **Comp** button yields the message shown here.

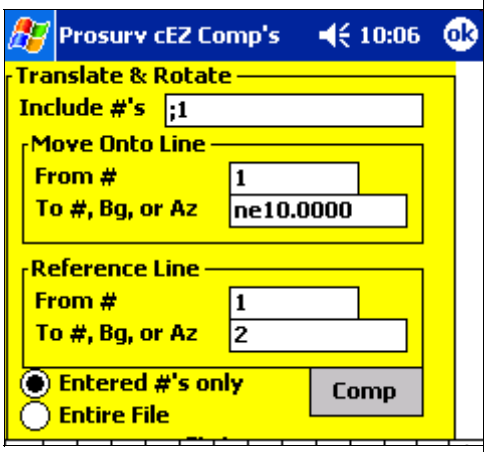
Tap **Yes** to store the computed points, or No to exit the routine.

Then tap the **OK** button in the upper right corner to exit the routine.



Translate & Rotate

The powerful translate and rotate routine gives you the ability to translate points, rotate points, or do both at the same time. Options include:



- Translate/Rotate up to 200 points at one time
- Translate/Rotate the entire job
- Translate/Rotate a list of points
- Translate/Rotate a pre-defined SET of points
- Use 4 point numbers
- Use a "From point" and an Azimuth or Bearing to define each system
- Use an asterisk to define a line, such as 8*22 which uses the Azimuth between points 8 and 22

In the example above, the points contained in SET #1 are to be translated and rotated. SET #1 contains points 1.5 (1 through 5) as previously defined, and happens to represent some horizontal geometry that was defined as tangent-curve-curve.

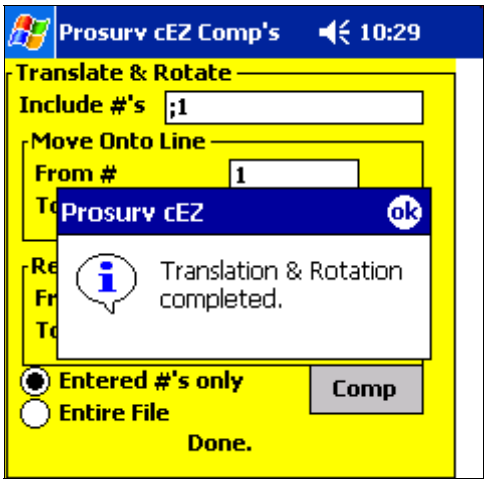
To use this routine, you define two points (or a point and a line) **to translate/rotate ONTO**. Then you define two points (or a point and a line) that represent your existing coordinate system.

In the example, there won't be any translation, since you're using the same point number "From" for each system, in this case, **Point #1.** If you entered a different "From" number for the Reference Line, such as 55, then Prosurv cEZ would "move" or translate your points based on the translation between 55 and 1.

Back to the example, the **existing bearing** from 1 to 2 is N 22°15'10" E, which you **don't need to know**, because Prosurv cEZ will compute the bearing from 1 to 2 since you entered these numbers for the reference line.

Ultimately, in the example, we will be rotating our horizontal geometry from it's current position so that points 1 to 2 will be at N 10°00'00" E.

Tap the Comp button to compute and store the points. In this case, only the points 1.5 will be changed.



If you need to translate without any rotation, you could enter **NE0.0000** for each **To #**. An example would be moving from a 1000, 1000 system to 50000,50000.

Translate Elevations

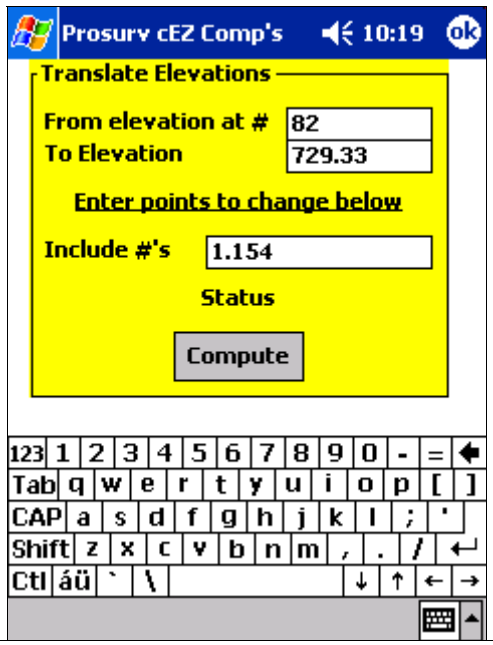
This routine lets you translate elevations from one datum to another. To translate the elevations of points in your job:

Enter the point number whose elevation is used as the "From" elevation

1. Enter the new elevation for that point
2. Enter the point numbers whose elevations will be changed, based on the difference between 1 and 2. The point being used for #1 does not have to be in the list of points being changed.

This routine is setup so that it works the way you do. For example, say you've shot a Fire Hydrant as a Benchmark and initially it was assumed as 100.00' in elevation. When you shot the BM, it was stored as shot #82. Now, you've run a level circuit and find that it's elevation is actually 729.33. So for the first text box you'd enter 82, and in the second text box, you'd enter 729.33. It's that EZ!

In the third text box, you've entered the point list 1.154. So every point from #1 to #154 will have it's elevation increased by (729.33-100) =629.33'.

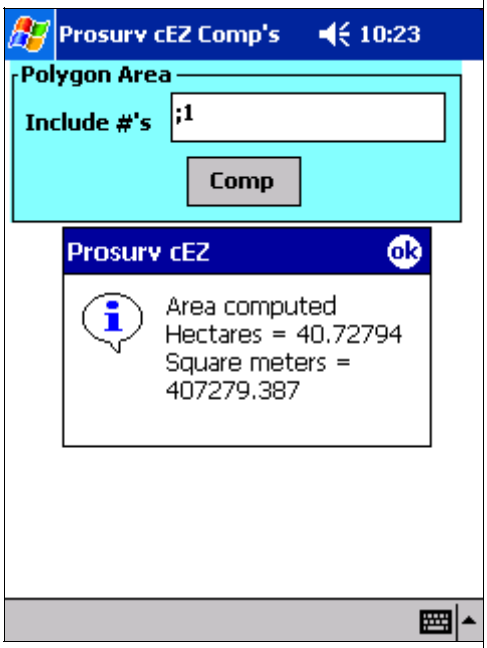


40 Polygon Area

This routine simply computes the Area within a boundary. A point list of up to 200 points can be entered, or you can enter a SET (pre-defined point list) by entering a sem-colon and the SET #, such as ;5.

Note that if your Job units have been defined as metric, the area will be displayed in Hectares and Square meters.

If your Job units are US Foot, then your area will be displayed in Acres and Square feet.

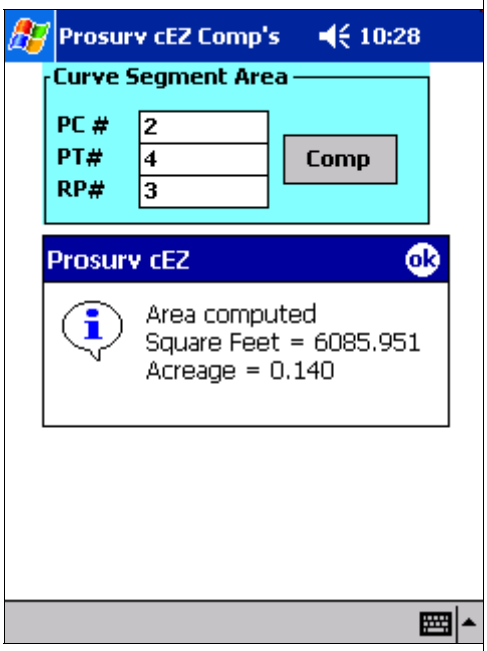


Curve (Segment) Area

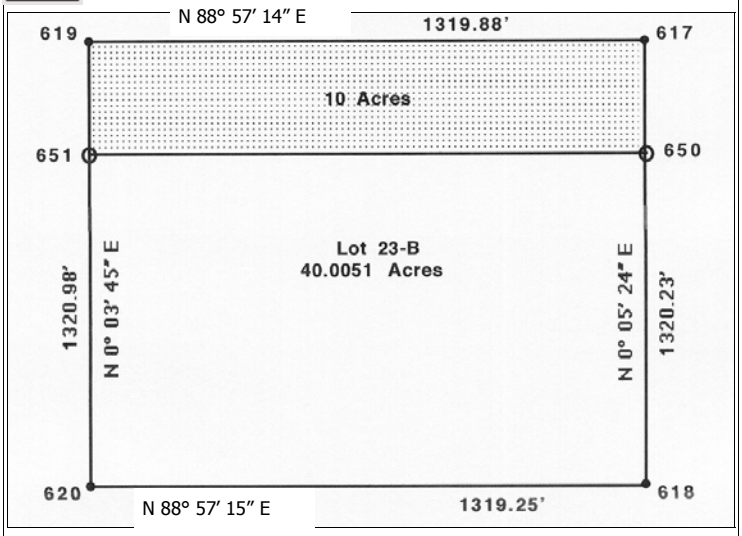
This routine computes the curve area within the curve segment. Enter the PC, PT, and Radius Point of the curve whose area you need.

Then press the Comp button to see the results.

If your Job units are metric, then Square meters and Hectares will be displayed.



Pre-determined Area—Sliding Line



You may use this routine to cut up a parcel or to expand acreage. The routine will create two intersecting points and only requires you to enter four points. Each point created will be on the lines given (i.e. Line 1 and Line 2). The two points will be at the bearing that you've chosen.

The Bearing that you choose can be **any Bearing.** It is <u>not</u> required that this bearing be parallel to any other line. To select the bearing of the new line, you may either:

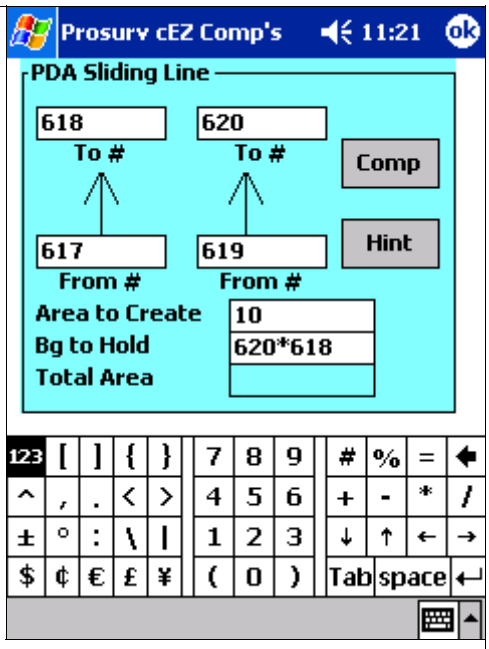
- **Enter the Bearing directly.**
- **Enter two points representing the Bearing.**

<u>Example</u>

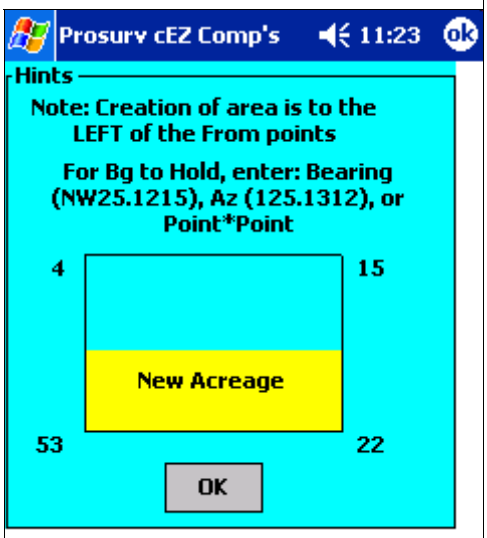
You want to create a 10 acre piece of land inside of a known 40+ acre Lot whose south line is parallel with the south line of the Lot. Line 1 is represented by points 617-618 while Line 2 is represented by points 619-620. The needed acreage therefore lies to the left of points 617 and 619 (the 'From' points of each line).

This will tell Prosurv cEZ how to compute the 10 acres. If you gave Line 1 as 619-620 and Line 2 as 617-618, then the 10 acres would be created

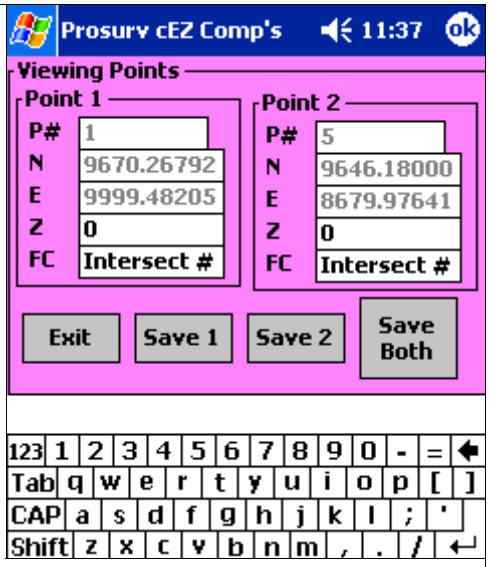
above the 40 acre Lot rather than inside of it. This is how you would *expand* acreage (simply by the way that you enter the four points).



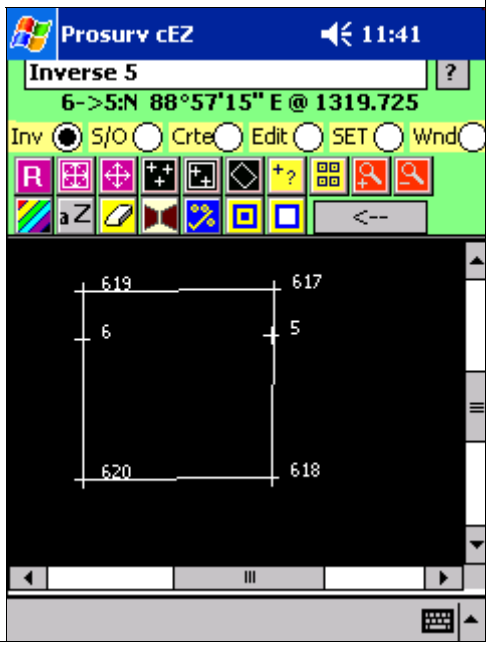
Tap the **Hint** button to view the screen shown here.



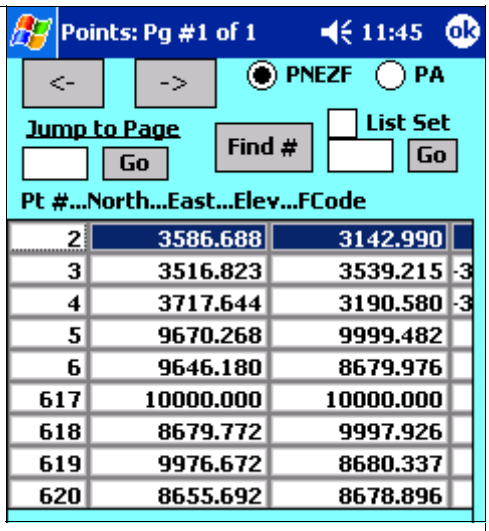
The results of the computation are shown here. You can save either point, or both points.



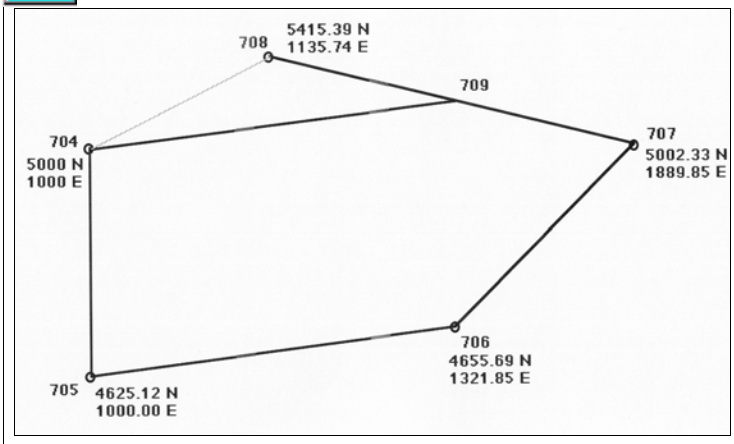
The resulting points (5 and 6) are plotted and displayed. Note the Inverse between the points is parallel to the south line of the Lot, which is what we wanted.



Computed points are shown to the right.

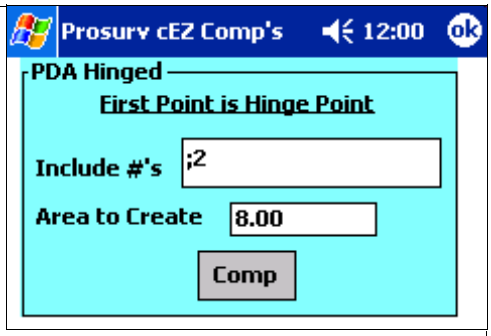


Pre-determined Area—Hinged

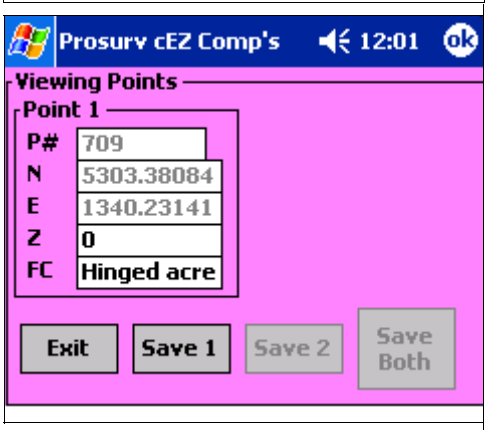


This routine can create acreages using up to 100 points to define the polygon. The first point entered will be the point used as the hinge point. The points of the polygon can be entered on the first line, or you may use a set number representing the points of the polygon simply by pressing a semi-colon ; followed by the set number.

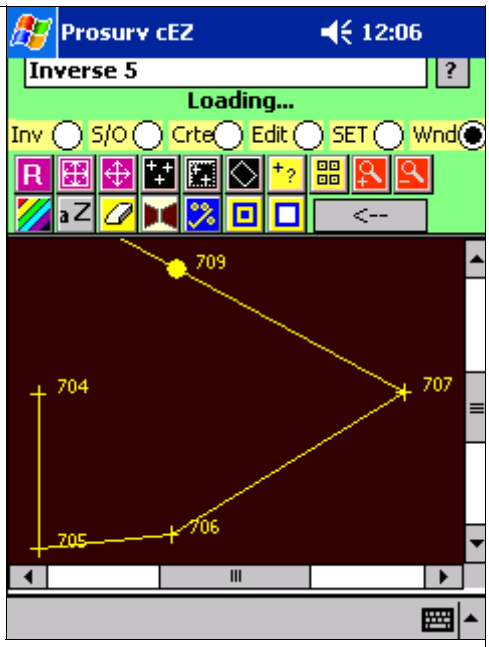
In this example, points 704.708 have been stored as SET #2, and we'd like to create 8 acres.



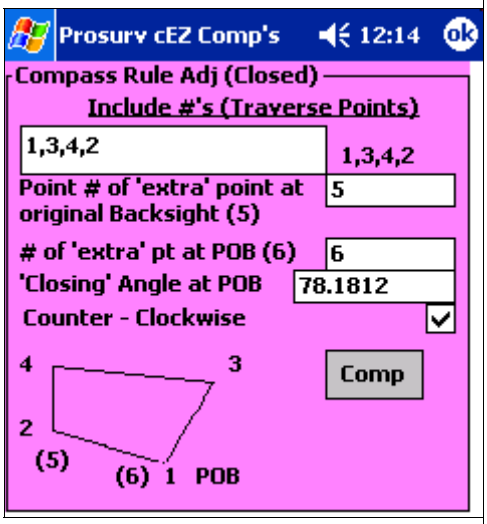
View the computed point and tap **Save 1** to store it.



Plotted points are shown to the right.



Compass Rule Adjustment—Closed Traverse



This routine will solve a correctly closed traverse, and requires the point numbers of the actual traverse points, along with the "extra" points shot when closing your traverse.

The illustration demonstrates the use of the "extra points".

Note that #5 is the point that you shot while occupying #4, backsighting #3 and "closing" to #2, which already existed since it was your original backsight when you started the traverse.

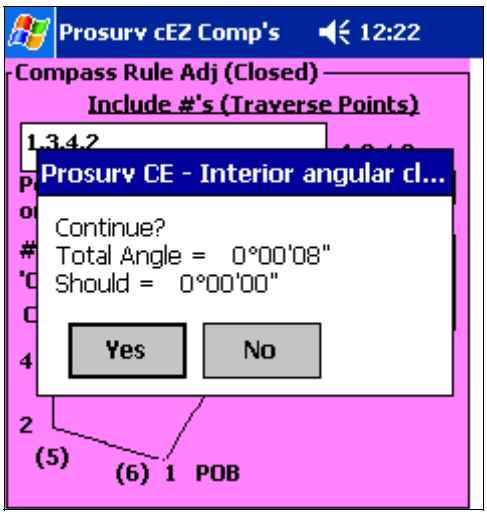
Point #6 is the shot you took while occupying #5, backsighting #4 and turning to #1 (the original POB).

Finally, you set up on #6, backsighted #5, and turned a closing angle to #3. This is the 78.1812 (D.MMSS) angle shown above.

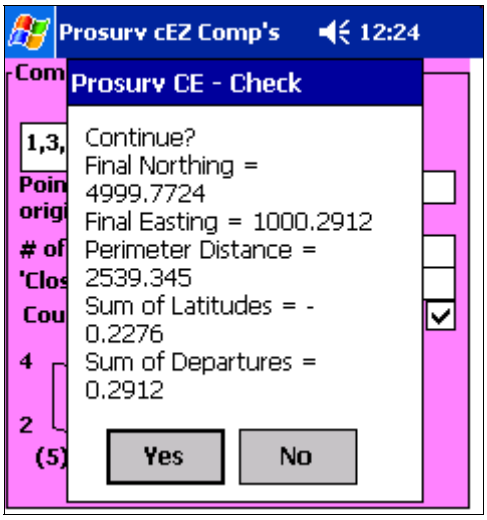
1	5000.0000	1000.0000
2	5438.2487	561.7513
3	5350.9003	1230.4849
4	6031.4494	639.9355
5	5438.3502	561.6466
6	4999.7825	1000.0827

Coordinates for the traverse points are shown here.

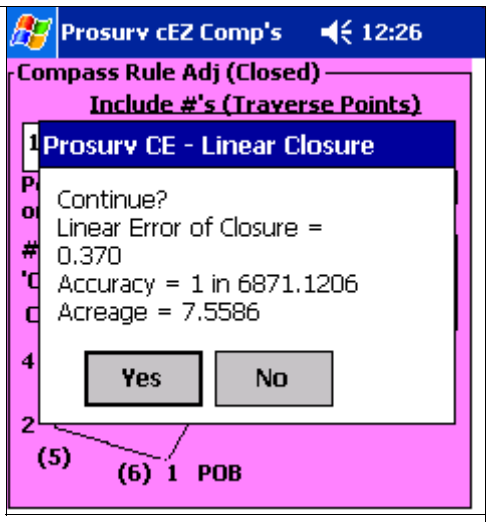
First, Prosurv cEZ shows you the Angular Error (in this case, the total should be exactly 360° which = 0°).



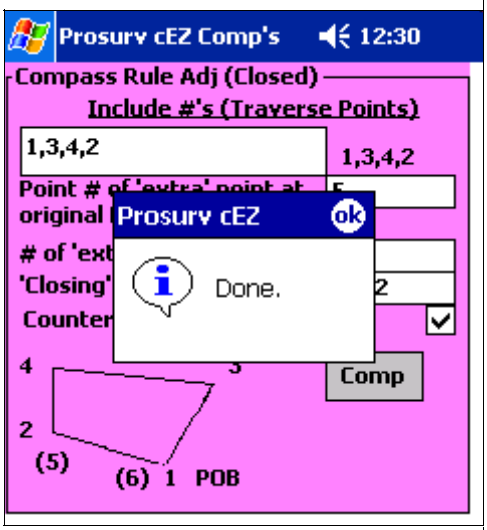
The next screen shows you the Sum of Latitudes and Departures, along with other necessary closure information.



Finally, the Linear Error of Closure, Accuracy, and total Area is shown.



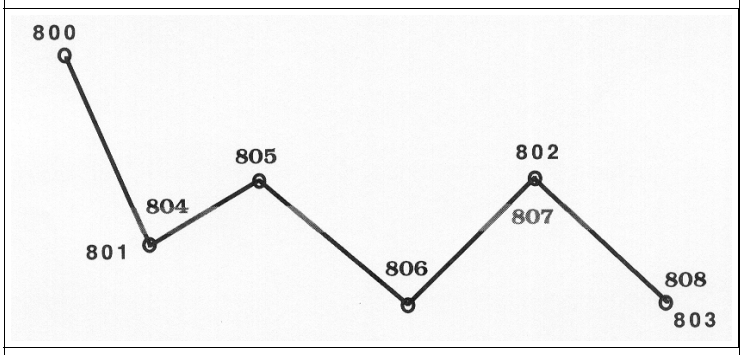
When you have completed viewing the closure information, Prosurv cEZ allows you to *Overwrite* the current traverse point's coordinates with the adjusted coordinates. If you wish to save the original coordinates, you should **export your points as an asc file** *before* computing the closure and overwriting the points. This way, you could bring them back into the job (and overwrite the adjusted points, or offset the point numbers) if needed.



The adjusted coordinates are shown here.

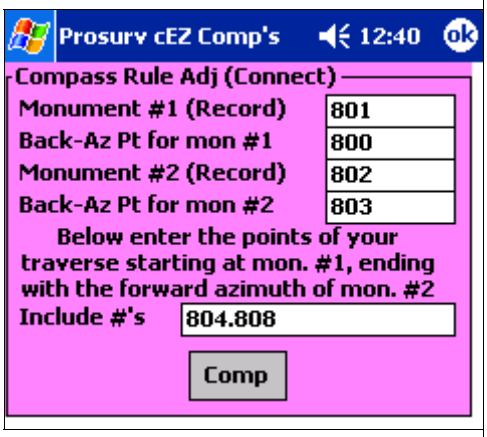
1	5000.0000	1000.0000
2	5438.4133	561.5362
3	5350.9379	1230.4368
4	6031.5620	639.7773
5	5438.3502	561.6466
6	4999.7825	1000.0827

Compass Rule Adjustment—Connecting Traverse



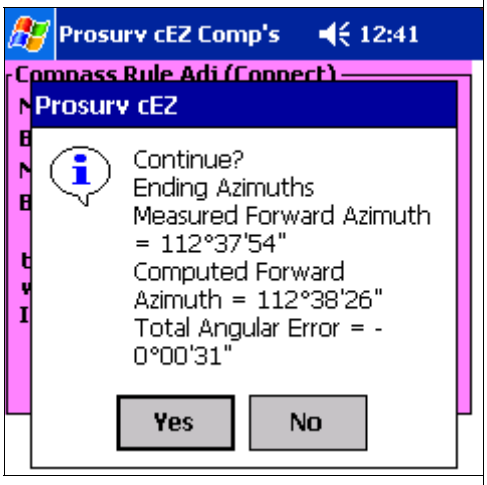
800 and 803 are concrete monuments. The coordinates shown below are from data given to you by your client (820-823 are the results of this routine).

Your traverse points between the monuments were shot as shown above. Also, as shown above, the coordinate of your first traverse point should be the same as the monument's coordinate.

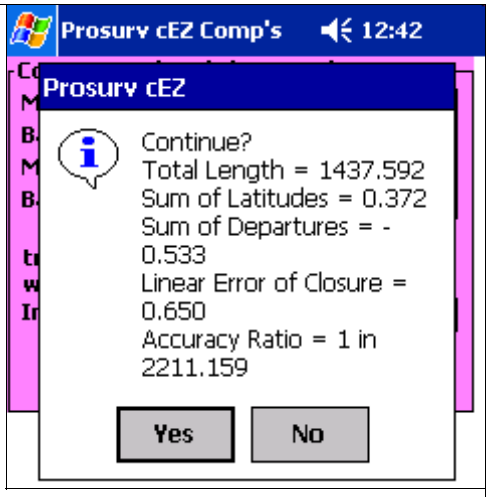


Point #	Northing	Easting
800	6002.85	800.77
801	5053.77	1029.55
802	5218.65	2129.33
803	4897.22	2899.98
804	5053.77	1029.55
805	5100.35	1492.87
806	4795.04	1819.33
807	5219.10	2128.77
808	4897.52	2900.11
820	5053.77	1029.55
821	5100.21	1493.04
822	4794.76	1819.65
823	5218.65	2129.33

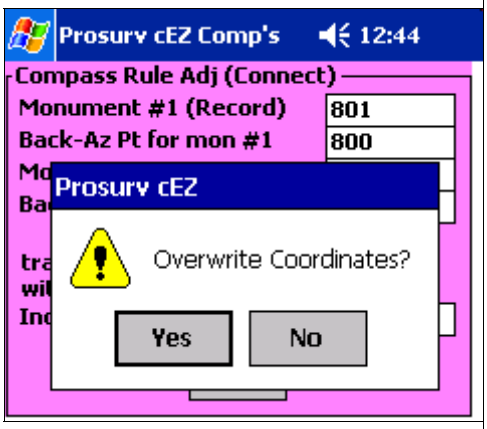
The forward azimuth to the final monument is displayed and compared to the computed forward azimuth. This angular error will then be applied to each bearing along your traverse.



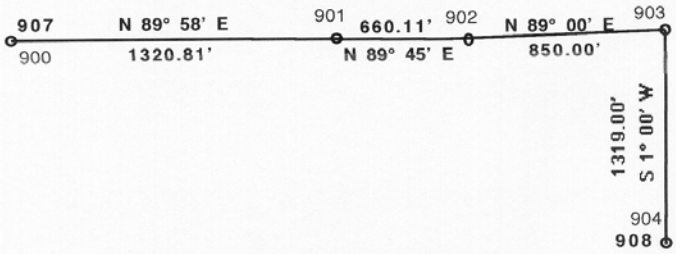
The closure information is shown here. If the closure is acceptable to you, press **Yes**. You will now be able to store the fully adjusted coordinates.



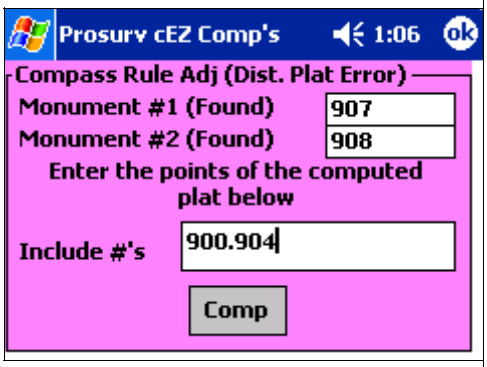
This routine overwrites the given points with the adjusted coordinates. Because of this, you should export your points as a backup prior to using this routine (in case of any mistakes).



Compass Rule Adjustment—Distribute Plat Error

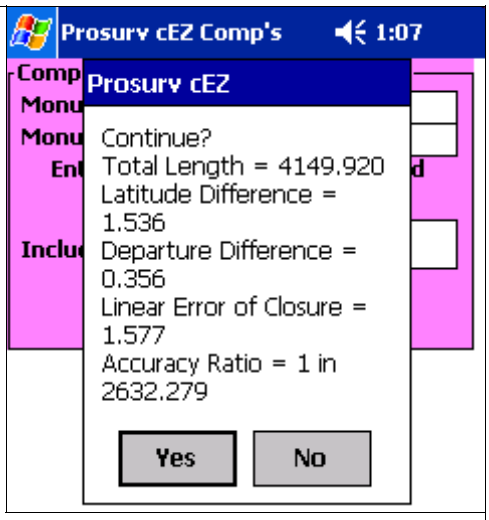


This routine allows you to adjust pre-computed plat coordinates to better fit what you've located in the field.



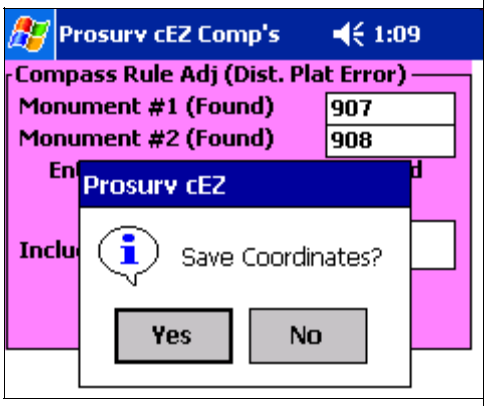
Points 900.904 are pre-computed coordinates based on a General Land Office Plat of 1917. A traverse was run during which monuments were found at points 900 and 904. These monuments were shot as points **907 and 908** respectively. The coordinates of these points are shown below. When using this routine, the coordinate of the first monument **must** be the same coordinate as the first point of your pre-computed traverse.

The closure information is shown here. If the closure is acceptable to you, tap **Yes**. You will now be able to store the fully adjusted **plat** coordinates.

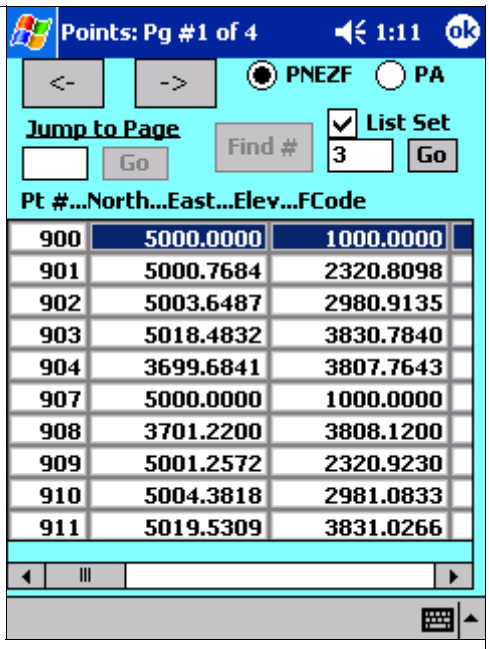


You may enter a new point number to begin storing your corrected coordinates (of the traverse).

The adjusted coordinates are stored automatically, using the Auto #.

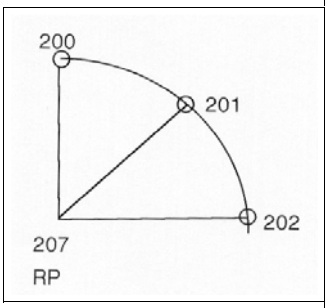


The entered, computed, and adjusted coordinates are shown here.



3-Point Radius Point

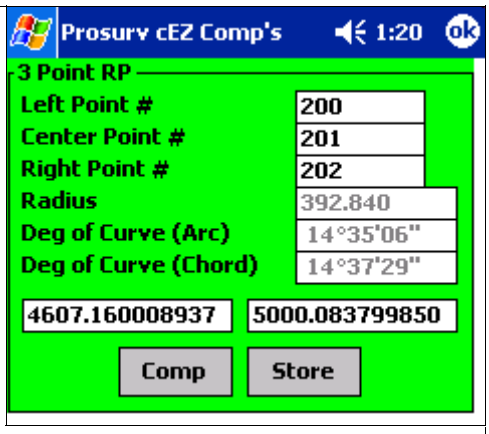
This routine will compute a radius point given 3 points on a curve. You must enter the points from left to right. The resulting coordinate will have an equal radius to all three points on the curve.



Prosurv cEZ lets you see the results of the computed point before you store the point. This lets you analyze several different points that you might want to use (for example, if you shot 10 points of a CL of a curve).

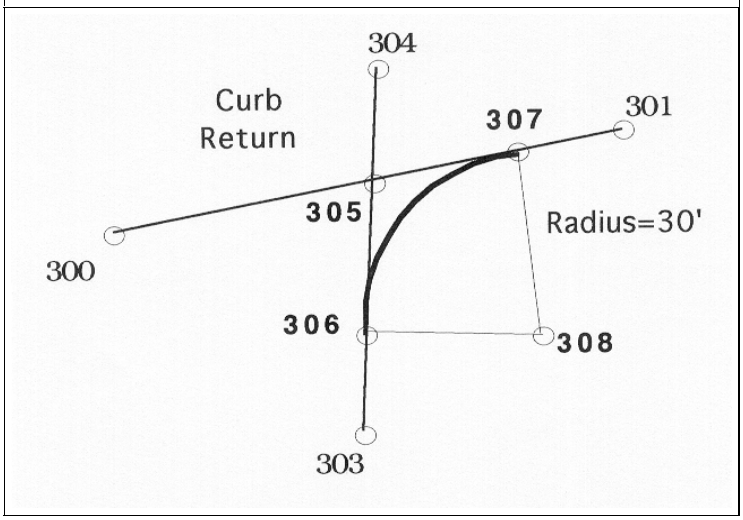
The coordinates of the computed RP are also displayed.

Tap the **Store** button to store the point.



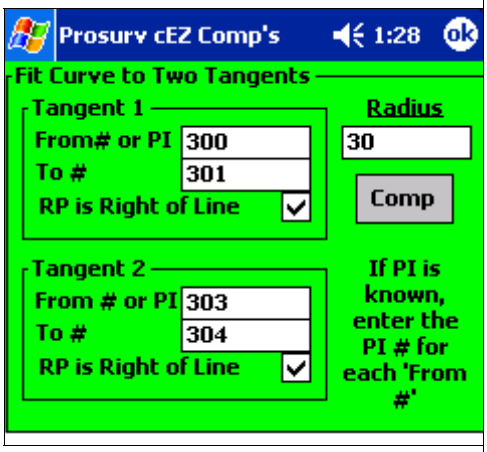
Fit a Curve to Two Tangents (Curb Returns)

The **Fit Curve** routine computes the Point of Intersection (PI), Point of Curvature (PC), Point of Tangency (PT), and Radius Point (RP) of a simple curve when given two points on each line and a radius amount. The PC and PT are computed so that they are at 90° angles to/from the Radius



point. This routine is great for computing the following:

- **Curb returns where a new road intersects an existing one**
- **Curb returns where two new roads will intersect**
- **Determining the PC, PT, RP, and PI when designing a new road in the field (Fit a Curve to Two Tangents)**



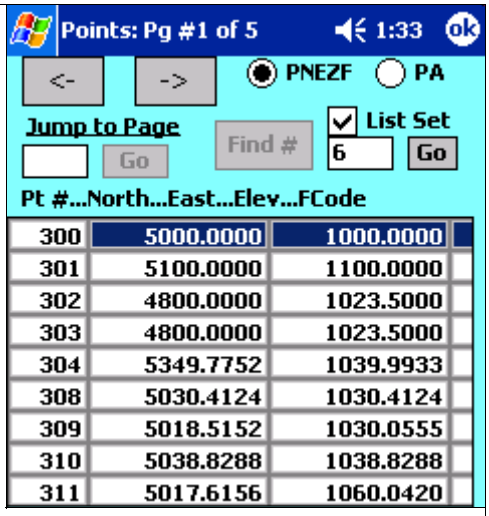
The PC, PI, PT, and Radius Point will be computed. You will have to know 2 points representing the existing curb (as shot in the field) and 2 points on the future curb line which will meet the existing. **Or, if you know the PI, you can enter the PI and one point from each line.**

If you don't know the PI, then the four points will have to intersect coordinately-i.e. a 'cross' is created if the points are connected. If, however, (as often happens) the points do not cross, simply create a point 500' up each line for example, and use these coordinates instead.

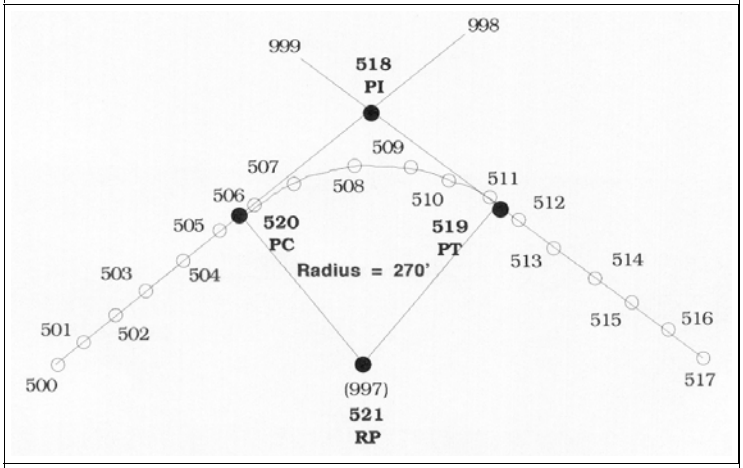
In the following example, point #304 was created at 500' from point #302 and 180 degrees from line 302-303 in order that the two lines would cross. The reason for this is that a total of four radius points could be computed from the two lines.

You will have to determine which is your PC and PT by inversing or plotting etc... The points are given the descriptors PC, PI, PT, and RP automatically.

Points used in this example are shown here.



Another great reason to use this routine is when designing a new roadway in the field. Refer to the diagram below for this example:

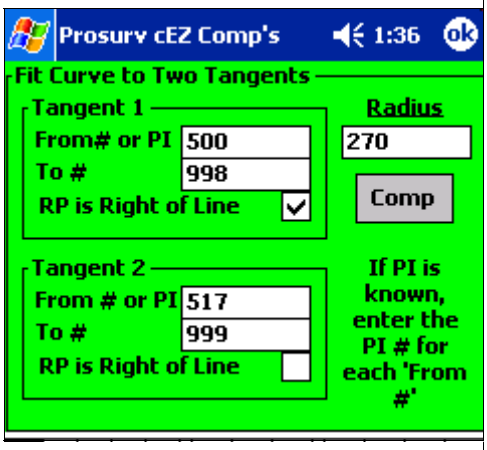


Shots 500-517 were taken at the centerline of an existing two-track dirt road. It is desired that the incoming tangent (Backtangent) should be between points 500 and 505, and that the outgoing tangent (Foretangent) be between points 512 and 517. To create the true curve (with the PC and PT

at 90° to the tangents), follow these simple steps:

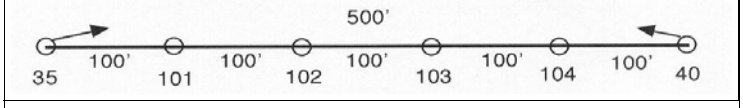
- **Compute a bogus radius point #997 to determine what radius to use. To do this, use the Compute Radius Point routine (Ctrl-*) and enter 506 as the left point, 508 as the center point, and 511 as the right point. Store this point as some bogus # i.e. 997.**
- **Inverse 997 to points 506 through 511 just to check that the radius to all the points is about the same amount (i.e. 270'). This is the radius you'll use to create the curve.**
- **Be sure to create bogus points (i.e. 998 and 999) so that the two tangent lines actually intersect (a requirement of this routine). To do this, you could use the Field Note reduction routine (Ctrl-F). I.E. BS = 500, Gun = 505, angle = 180°, distance = 1000' to create #998.**

Once you've got 998 and 999, press Shift-Ctrl-Q (this routine) and enter the data shown here.



Divide a Line (2D or 3D)

This routine will divide a line into *n* equal parts. All you need to supply are two points (representing the beginning and end of the line), the number of equal parts, and the starting point number to save the points. **This routine can also compute the points "3-D".** In other words, the slope of the line can be computed and the appropriate computed elevations will be stored with each point. If you need the elevations computed, simply check the "Compute Elevations" check box.



The example above shows a distance of 500' between points 35 and 40. It has 5 divisions which will create 4 points with 100' intervals automatically. The first point will be saved as point #101. The default number of divisions is 2, which would create one point. There will always be one less point created than there are divisions.