COLORADO DEPARTMENT OF TRANSPORTATION

Information Systems / Engineering Customer Support Unit

COGO User Guide and Command Reference

Version 2.0.4

ENGINEERING CUSTOMER SUPPORT UNIT

COGO User Guide and Command Reference

© 2001 – 2002 Colorado Department of Transportation
4201 E. Arkansas
Denver, Colorado 80222
December 16, 2002
COGO Version 2.0.4

Table of Contents

Table of Contents ii
Preface vi
Intended Audience vi
Document Structure vi
Manual Conventions vi
Introduction to COGO 1
What Is COGO? 1
COGO Item Definitions 1
Points 1
Lines and Line Segments 2
Curves 3
Simple Curves 3
Compound Curves 4
Reversing Curves 4
Transition Spirals 5
Complex Curves 6
Arcs 6
Curve Points 7
Alignments 7
Traverses 8
Parcels 9
Installing COGO 10
Recommended Computer Configuration 10
Installation Instructions 10
Installing from the CDOT network 10
Installing COGO via the Internet 12
Setting up COGO in AutoCAD 14
Executing COGO in Microsoft Windows 16
Executing COGO Inside of AutoCAD 17
Entering Data in COGO 19
Entering Numbers 19
Entering Characters 20
Item Names 20
File Names 20
Dates 20
Entering Distances 21
Entering Directions 21
Azimuths 21
Bearings 22
Zenith Angles 23
Entering Stations 24
Creating Points 25
The Point Menu 25
Storing Points 26
Copying Points 26
Locating Points Using Existing Information 27
Using a Distance and Direction to Locate a Point 27
Using a Direction from a Line Segment to Locate a Point 27

Deflection Method 28
Angle Method 28
Using a Station to Locate a Point 29
Locating an Offset point 30
Offsetting Points from a Curve 30
Projecting Points on a COGO Item 31
Creating Points by Intersecting Two COGO Objects 33
Creating Points by Turning Angles from the Ends of a Line Segment 34
Locating a new point using two distances from two points 34
Locating a new point on a line, a distance from a point 35
Locating a point of tangency to a curve, using a direction 36
Locating a Point of Tangency to a Curve, Using a Point Near the Curve 36
Other Functions That Create Points 37
Creating and Manipulating Lines 38
The Line Menu 38
Using Temporary Line Segments 39
Storing Line Segments 39
Storing Lines or Rays 39
Dividing Lines to Create Points 40
Best Fit Line 41
Creating Curves 42
The Curve Menu 42
Creating a Curve 43
Dividing a Curve to Create New Points 45
Creating and Adjusting Traverses 46
The Traverse menu 46
Storing a Traverse 46
Creating a Traverse Using Distance / Direction Pairs 47
Creating a Traverse Using the Angle Right Survey 49
Adjusting a Traverse 50
Creating Alignments 51
The Alignment Menu 51
Storing an Alignment 51
Creating an Alignment 52
Creating Curves as Part of an Alignment 54
Creating Parcels 55
The Parcel Menu 55
Creating a Parcel 55
Listing, Reporting, and Deleting Items 59
Listing Items in the Model 59
Item Reports 60
Deleting Items 61
Other COGO Functions 62
The Misc Menu 62
Angle, Azimuth, Bearing, and Distance 62
Executing COGO Commands 63
Deleting all COGO Items 63
Resetting the Total Area Register 64
Setting the Working Units 64
Station Offset Function 64
Rotating and Translating COGO data in Windows and AutoCAD 65
Transferring Information to Other Engineering Applications 66
AutoCAD 66

Layer Names in AutoCAD 67
MX..... 68
SDR Data Collectors and PICS 68
RoadCalc Version 13..... 70
COGO Input, Output, Log, Saved Tables, and Archive Files..... 71
COGO Input from a File..... 71
The COGO Log File..... 72
COGO Output to a File 73
COGO Saved Tables..... 73
Archiving COGO Data for Future Use 74
COGO in Windows Functions..... 75
The View Menu..... 75
Zooming..... 76
Panning..... 76
Refreshing the Display Area..... 77
Setting Item Colors 77
COGO in AutoCAD Functions..... 78
Displaying COGO Data in AutoCAD..... 78
Graphically Selecting COGO Points..... 79
Graphically Moving COGO Points 80
COGO Command Reference 81
Adjust Traverse Command..... 81
Alignment Command 82
Angle Command 86
Archive Command 86
Area Command..... 87
Azimuth Command 88
Bearing Command..... 88
Deleting Points, Lines, Curves, Traverses, Alignments, Parcels and Text ... 89
Deleting All Items in the COGO Model 89
Distance Command 89
Divide Line Command 90
Divide Line Segment Command..... 90
Divide Curve Command..... 91
Eject Command 91
Finish Command..... 91
Layout Ties Command..... 92
Layout Curve Command..... 93
Layout Alignment Command 93
List Points, Lines, or Curves Command 94
List Alignments, Traverses, Parcels, or Text Command 94
List Alignment, Traverse, or Parcel Contents Command 94
List All Items Command..... 95
List Table Size Command..... 95
Locate Command Common Parameters and Options 95
Locate a Point Given a Distance and Direction From Another Point 98
Locating a Point Given a Distance and Deflection from the End of a Line Segment 98
Locating a Point Using a Distance and Angle from the Beginning of a Line Segment 99
Locating a Point Using a Station on an Alignment 100
Locating a Point Which Is Offset from a COGO Item 101
Locating Points Which Are Offset from a Curve 101
Locating a Point by Projecting Another Point Onto a COGO Item 102
Locating the Intersection of Two COGO Items..... 103
Locating a Point Using Deflections From the Ends of a Line Segment..... 104

Locating a Point Using Two Distances from Two Different Points 105
Locating a Point Using a Distance from a Point and an Angle Turned from a Line Segment 106
Locating a Point on a Curve Where the Tangent to the Curve Is a Specified Direction 107
Locating a Point Where the Tangent to the Curve Passes Through Another Point 108
Parcel Command 109
Print Alignment Book Command 112
Reporting the Geometry of Points, Lines, Curves, Alignments, Parcels, and Text 113
Print All Command 115
Reset Area Command 115
Restation Alignment Command 115
Restore Command 116
Rotate Translate Command 117
Save Command 117
Set No Redefine Command 118
Set Date Command 118
Set File Commands 119
Set Log File Command 119
Set File Input 119
Set File Input TTY Command 120
Set File Output Command 120
Set MOSS File Output Command (COGO to MX Interface) 121
Set Roadcalc File Output Command (COGO to Roadcalc Interface) 123
Set SDR File Output Command (Leitz data collector interface) 124
Set Three Dimensions Command 124
Set Units Command 125
Station Alignment Command 125
Station Offset Alignment Command 125
Store Alignment Command 126
Store Curve Command 127
Store Line Command (direction form) 132
Store Line Command (point to point form) 132
Store Point Command 132
Copy Point Command 133
Store Spiral Command 133
Store Text Command 134
Store Traverse Command 135
Traverse Command (distance/direction) 135
Traverse Command (angle right) 137
An Example of a COGO Project 140
Appendix A – COGO command word abbreviations 153
Appendix B - COGO Error Messages 155
Appendix C - COGO Command Quick Reference 158

Preface

This manual is a user guide and command reference for the Colorado Department of Transportation COGO program.

Intended Audience

This manual is intended someone who is familiar with their discipline and has an understanding of geometry.

Document Structure

The first part of this manual is a guide to understanding CDOT COGO. If you have never used COGO before it is advisable to start with this section.

The second part of this manual is devoted to the individual commands. The commands are in alphabetical order.

The appendices contain command word abbreviations, a listing of the COGO error messages, and a COGO quick reference.

Manual Conventions

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

Each chapter in this manual begins with an icon key. These icons point out COGO command information, Windows information or tips, AutoCAD information, and important notes.

Just as COGO has conventions for entering commands, this manual has conventions which indicate how to enter the command, the sequence of items in the command and what data is supplied by the user to complete the command. A COGO command is composed of three major elements

- Command words
- User supplied data
- Command parameters

Command words may be required or optional. All command words appear in the manuals as upper case (capital) letters. Optional commands words are surrounded by square brackets ([ ]). Please note that the examples are the exception to this convention. Required command words **cannot** be omitted from the command. If the required command words are omitted COGO will not recognize the command and may execute a command which is similar in structure. This may cause unexpected results. Optional command words may be omitted from the command. Optional command words serve two purposes. Optional words reduce the amount of data input and allow the COGO user to customize the command to be executed. Omitting command words results in less input for the user but may decrease the legibility of the COGO input. If a question arises about whether or not an optional command word should be used, let the legibility issue decide it. Another way of reducing the input to COGO is to use command word abbreviations. Most command words may be abbreviated. The abbreviations are contained in the COGO command reference in Appendix A. The following is an example of a command in the reference manual.

```
STORE [POINT] p *id* [NORTH] n [EAST] e [STATION] s [ELEV] el
```

The required command word is STORE. The optional command words are; POINT, NORTH, EAST, STATION, and ELEV. The user supplied data is; p, n, e, s, and el.

Command parameters are parts of the command that are not required for proper execution of the command. All command parameters are surrounded by square brackets ([ ]). Some command words in the command parameters are optional and they too will be surrounded by square brackets.

If a series of command words or phrases are surrounded by parenthesis ( ) and separated by slashes (/), one of the command words or phrases **must** be selected from the list for the command to execute properly.

Chapter 1

Introduction to COGO

This chapter covers information about COGO, the basic items used in COGO, and how to install and execute COGO. This chapter is essential reading for the new COGO user.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

This chapter is divided into four sections: What is COGO; COGO item definitions; Installing COGO; and How to execute COGO in Windows and in AutoCAD.

What Is COGO?

COGO (COordinate GeOmetry) is an engineering tool used to determine geometric relationships. There are a number of ways to create a geometric model. There is a command language that allows the user to define and solve geometric problems in engineering terms. There are Windows forms that you fill in to create the geometry. And you can execute COGO inside of AutoCAD using the Windows forms and native AutoCAD drafting tools to create, display, and manipulate the COGO model.

The result is a model that describes the geometric relationships for a particular project. The information in the COGO model is used in many disciplines such as surveying, highway design, right of way, land acquisition, and construction.

COGO Item Definitions

The definitions in this section are used throughout this manual and in the COGO command reference.

Points

A point in COGO is a specific location and has the following properties.

- A northing (north or Y)
- An easting (east or X)
- An elevation (elev or Z)
- A point number
- An identification (PID), maximum of 20 characters

The northing, easting, and elevation make up the coordinate part of the point. The point is referenced by the point number. The identification is used for comments about the point or identifies the special nature of the point. Points are stored in the COGO model.

Another type of point is available it is the curve point. Please refer to the section on curves for further information.

! You can influence the speed at which COGO executes by using smaller point numbers and by not leaving large gaps in the point numbers used. This is because COGO must have a way to determine when a point is defined or undefined. All undefined points must be cleared such that when point number 1000 is the first point defined in the COGO execution, point numbers 1 to 999 must be cleared before point 1000 can be created.

Lines and Line Segments

There are three types of lines in COGO, a line and a line segment. The method used to create the line determines the line type.

A line in COGO is comprised of a starting point and a direction. The line starts at the point and continues in the direction infinitely. In mathematics this type of line is commonly called a ray. A line has the following properties.

- Line number
- Starting point
- Direction

A line segment is a line that starts at a point and ends at another point. A line segment is stored in the model. A line segment has the following properties.

- Line number
- Starting point
- Ending Point

A temporary line segment is a line that starts at a point and ends at another point. A temporary line segment is **not** stored in the model. The temporary line segment is only available during the function where it is created. A line segment has the following properties.

- Starting point
- Ending Point

! You can influence the speed at which COGO executes by using smaller line numbers and by not leaving large gaps in the line numbers used. This is because COGO must have a way to determine when a line is defined or undefined. All undefined lines must be cleared such that when line number 100 is the first line defined in the COGO execution, line numbers 1 to 99 must be cleared before line 100 can be created.

Curves

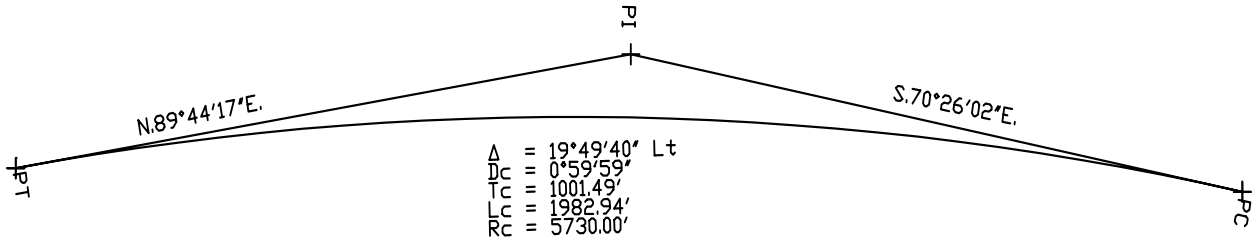
COGO has many types of curves. All curves are stored in the COGO model independent of the type. The possible types of curves are:

- Simple curves
- Compound curves
- Complex curves
- Reversing curves
- Transition spirals
- Arcs (Parcels only)

! You can influence the speed at which COGO executes by using smaller curve numbers and by not leaving large gaps in the curve numbers used. This is because COGO must have a way to determine when a curve is defined or undefined. All undefined curves must be cleared such that when curve number 500 is the first curve defined in the COGO execution, curve numbers 1 to 499 must be cleared before curve 500 can be created.

Simple Curves

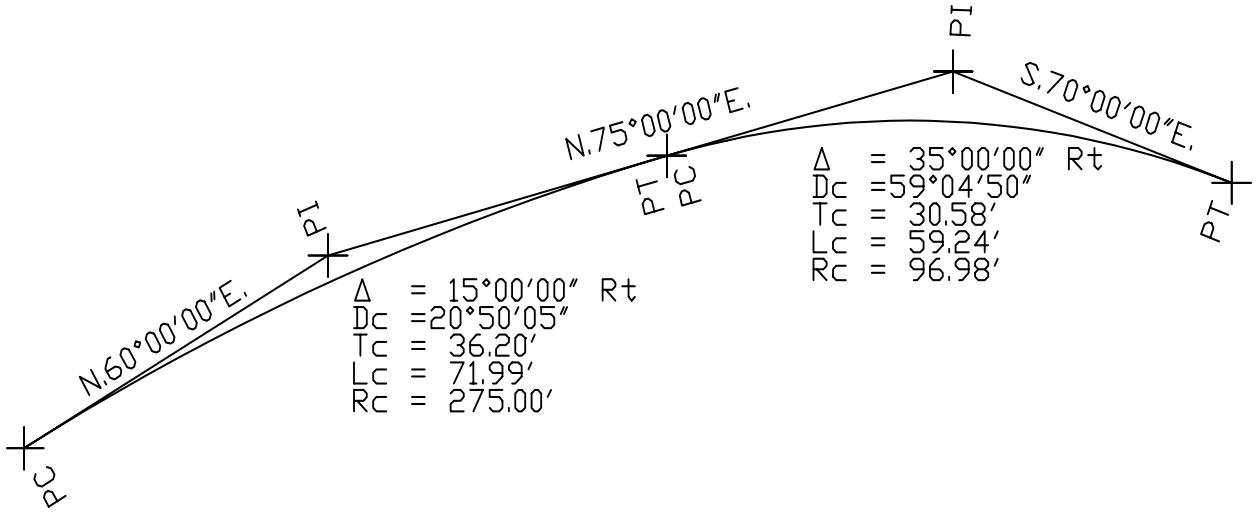
A simple curve is a circular arc in the horizontal plane that has a central angle of less than 180 degrees (PI radians). A simple curve starts at the PC (Point of Curvature), has a CC (Circle Center) and a PI (Point of Intersection for the tangents to the curve), and ends at the PT (Point of Tangency, the ending point of the curve).



Simple Curve

Compound Curves

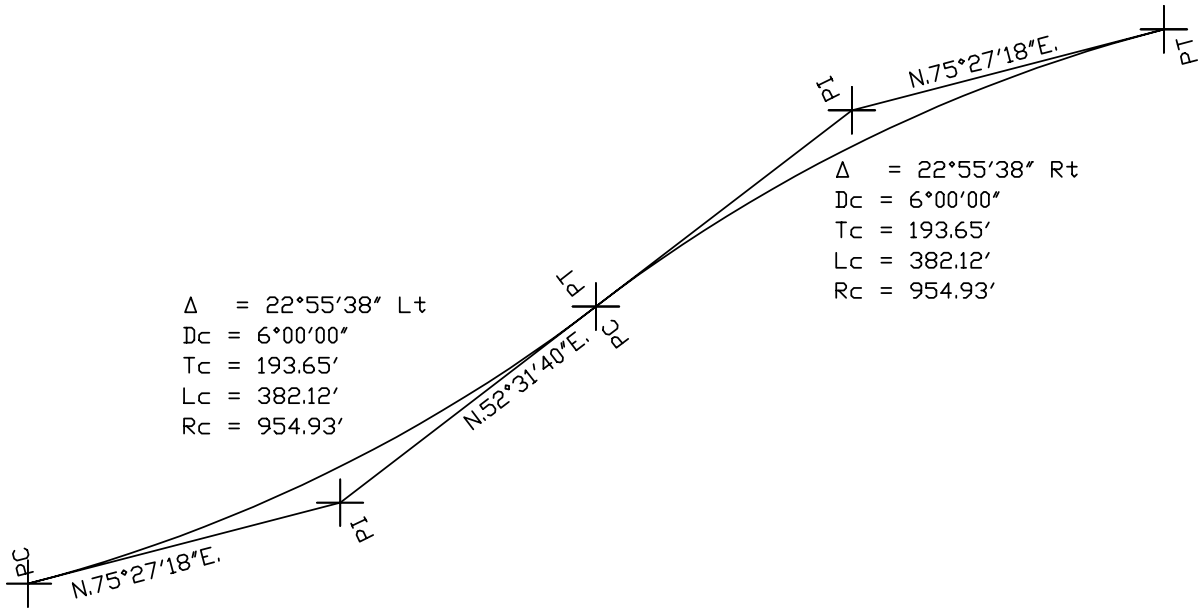
A compound curve is a curve that consists of two or more simple curves with different radii, which curve in the same direction. Normally the adjacent simple curves will have a common point such as the PT of one curve is the PC of the next. If a short tangent is between the simple curves, the compound curve is usually referred to as a broken back compound curve. Compound curves are used as a transition from large to small radius curves. Each simple curve in the compound curve is stored separately in the COGO model.



Compound Curve

Reversing Curves

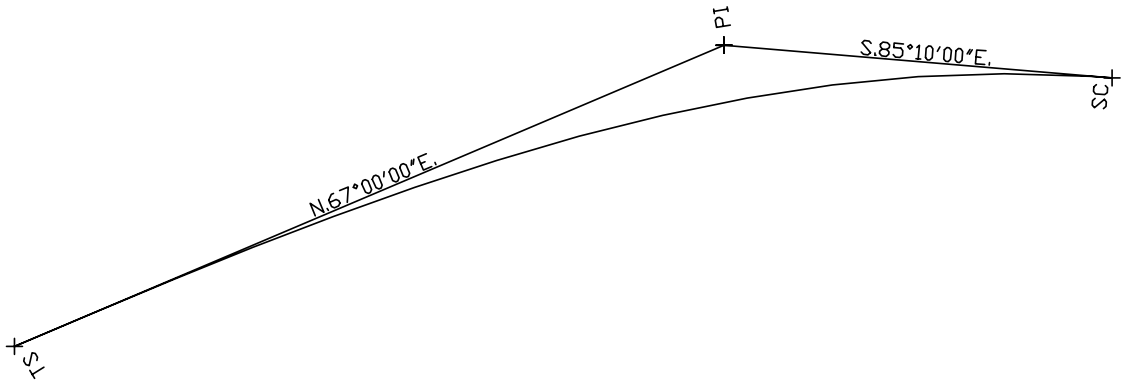
Reversing curves are two simple curves which curve in opposite directions. The curves may have the same or different length radii and have a zero length tangent between them. The primary use for reversing curves are temporary detours and are usually used in pairs known as double reversing curves. The simple curves for the reversing curves are stored in the COGO model separately.



Reversing Curve

Transition Spirals

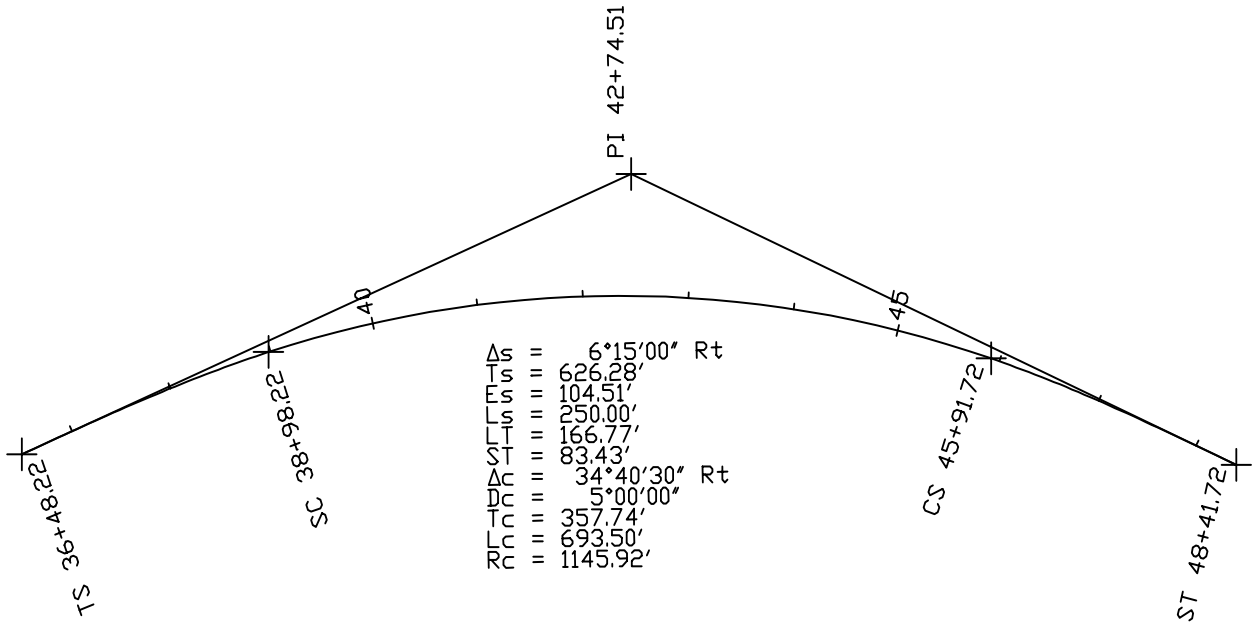
A transition spiral is a curve that provides a gradual change in curvature from a straight line to a circular curve or a transition between two curves of different radii. As the length of the spiral increases the radius of the curve decreases. The spiral starts at the TS (Tangent to Spiral) or CS (Curve to Spiral) point and ends at the ST (Spiral to Tangent) or at the SC (Spiral to Curve) point. A spiral that is a transition from a tangent line to a curve is commonly called a spiral in, a spiral from a curve to a tangent line is commonly called a spiral out. A spiral can be stored in the COGO model by itself or as a part of a complex curve.



Transition Spiral

Complex Curves

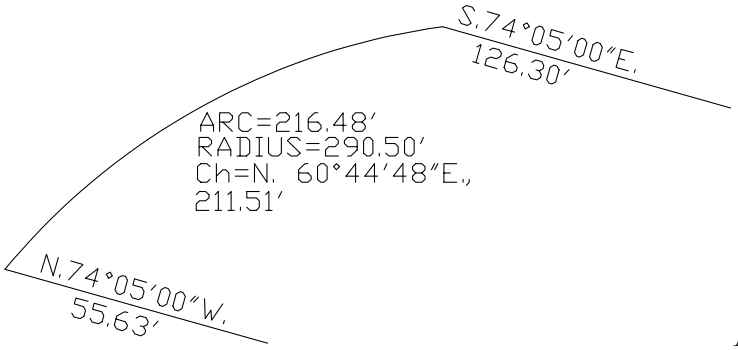
Complex Curves are curves with a simple curve and transition spiral(s). Complex curves fall into three categories; a spiral/curve, a curve/spiral, and a spiral/curve/spiral. Complex curves can be stored in the COGO model as a single curve or each part of the complex curve can be stored separately (not recommended).



Complex Curve With Equal Length Spirals

Arcs

An arc is a part of a circle. Arcs are primarily used in the geometric description of a parcel of land. Arcs are not limited to 180 degrees or less because defining an arc is not dependent on tangent lines. Arcs may or may not be stored as a part of the COGO model depending on which COGO command is used. Please refer to the AREA command and the PARCEL command for more information on arcs.



An Arc in a Parcel

Curve Points

All of the curves in COGO have associated points that may be referenced as independent points by some COGO commands. The rules for referencing curve points is that the reference to the curve point **must** start with the command word POINT followed by the curve point type and the curve number which the point belongs to. The type of curve dictates which points can be referenced. The following is a list of the curve points that may be referenced.

PC - Point of curvature

PI - Point of intersection for the curve tangents

PT - Point of tangency

TS - Tangent to spiral point

SC - Spiral to curve point

ST - Spiral to tangent point

CS - Curve to spiral point

CC - The center of the curve

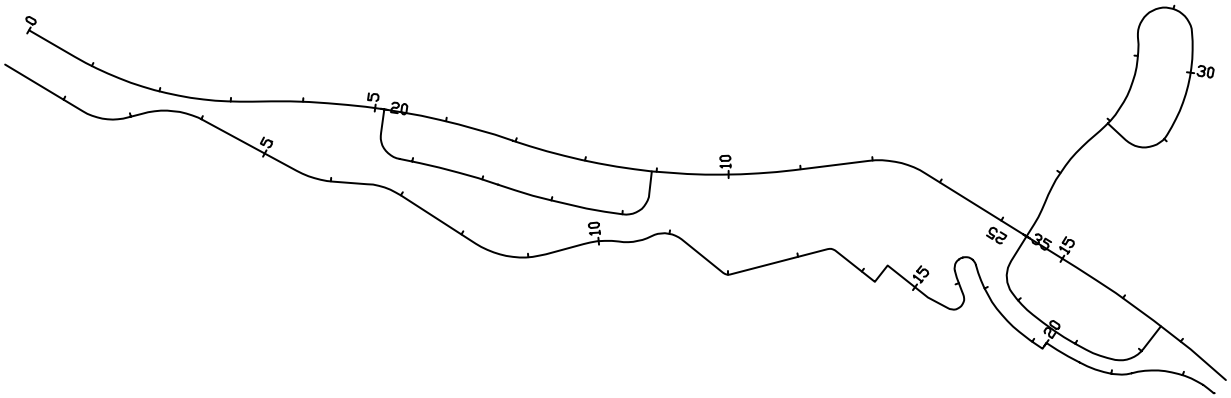
More information on curve points is available in the sections on specific curve types.

Alignments

An alignment is a collection of points, and curves. Alignments are typically used to geometrically describe the center line of a road. Alignments are stored in the COGO model. An alignment has the following properties.

- A name
- A starting point
- An ending point
- Automatic stationing

- Curves (Optional)
- Points (Optional)
- Station equations (Optional)

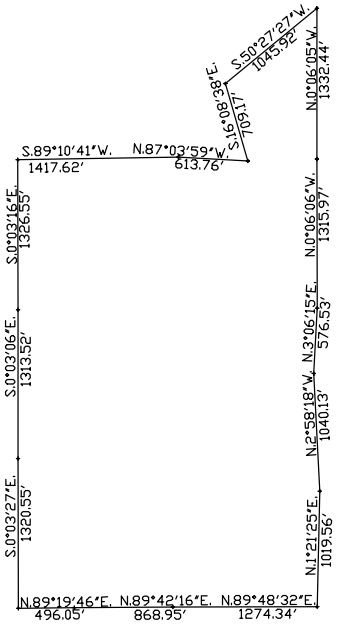


COGO Alignments

Traverses

A traverse is a collection of points. The points are linked by a distance and direction from the previous point in the traverse. The distance/direction from point to point is commonly called a course in the traverse. A traverse may be open (the starting point does not equal the end point), or closed (the starting point equals the end point). Traverses are stored in the COGO model. A traverse has the following properties.

- A name
- A starting point
- An ending point
- Points (Optional)

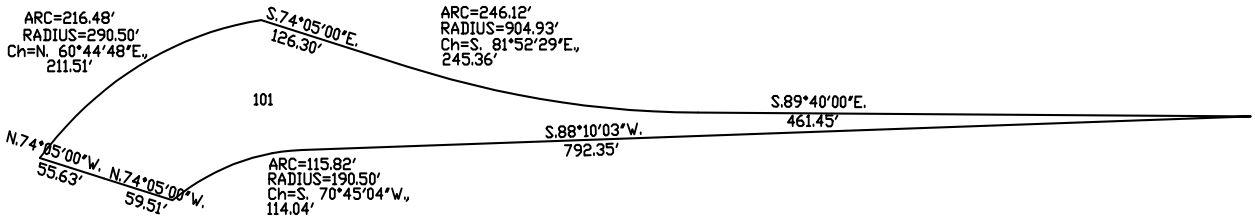


Closed Traverse

Parcels

A parcel is a collection of points and arcs that geometrically describe a parcel of land. Parcels may be open (the starting point does not equal the end point) or closed (the starting point equals the end point). An open parcel is commonly called an access. The beginning of the parcel may be tied to other points in the COGO model. Parcels are stored in the COGO model. A parcel has the following properties.

- A name
- Tie points (optional)
- A starting point
- An ending point
- Arcs (Optional)
- An area to be added or subtracted from the Total Area Register (if the parcel is closed)



Closed Parcel

Installing COGO

Recommended Computer Configuration

- Windows NT with service pack 5, or Windows 2000.
- AutoCAD 2000 to 2002 (required for use in AutoCAD)
- 64MB memory, minimum (128MB preferred for AutoCAD)
- 20MB free disk space (maximum)

Installation Instructions

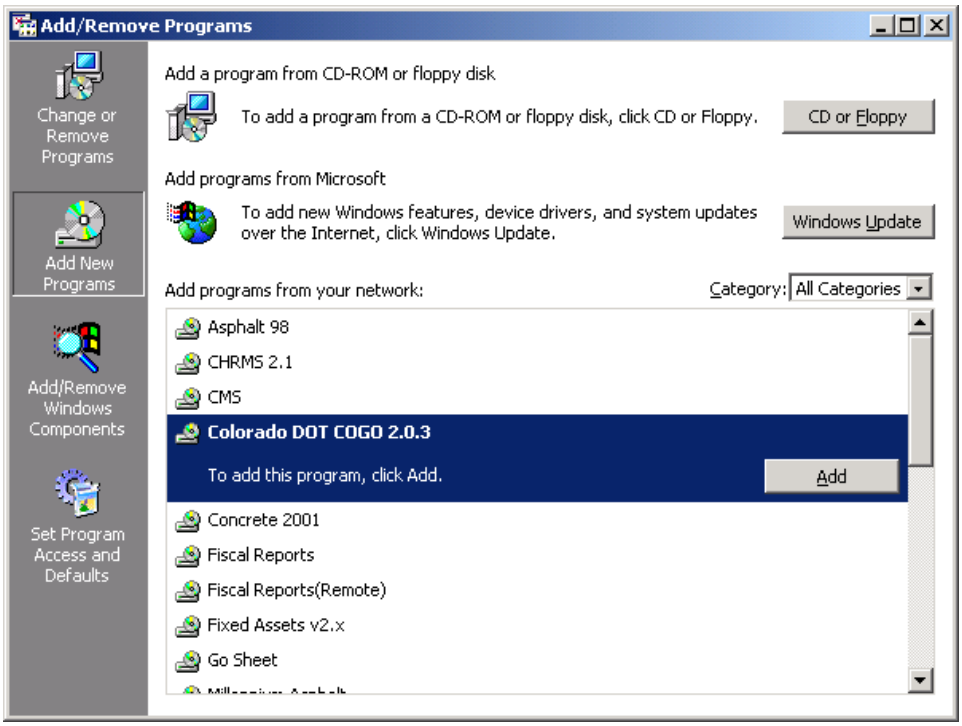
! If you have a previous version of COGO or WinCOGO installed on your computer, **remove it** before proceeding with these installation instructions.

To remove an older version of COGO; click on the Start button, click on Settings, then Control Panel, then Add Remove Programs. Select COGO from the list and then click on the Remove button.

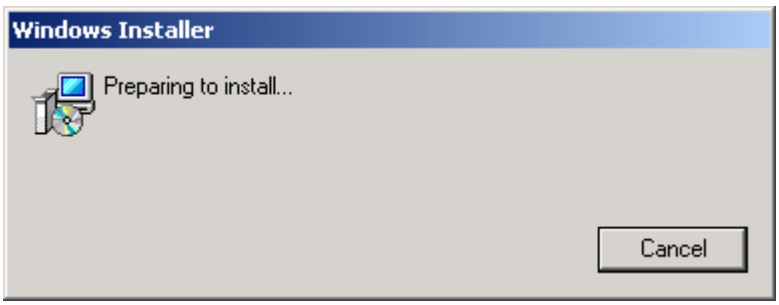
COGO may be installed in one of two ways. The first is available only if your computer is part of the CDOT network. The second is available to anyone who can connect to the Internet.

Installing from the CDOT network

Installing COGO when your computer is connected to the CDOT Network is done via Start => Settings => Control Panel => Add/Remove Programs. Click on the Add New Programs box on the left. Click on Colorado DOT COGO. The window should appear as follows.



Click on the "Add" button and the following window will appear.

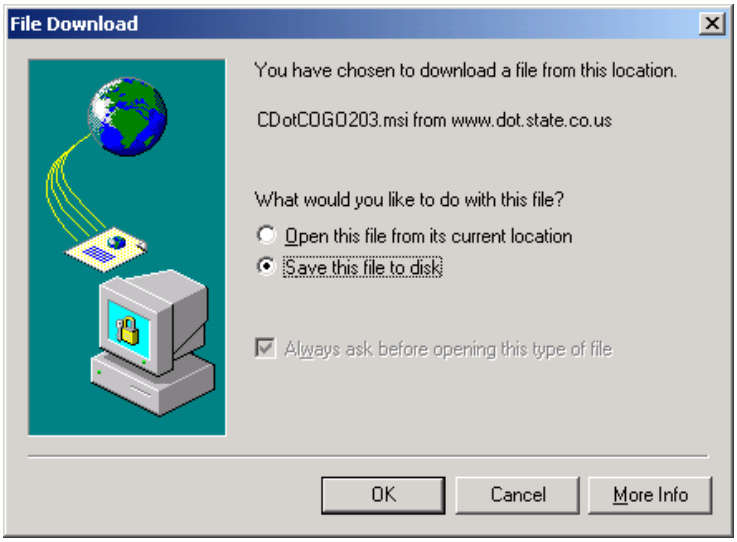


As installation progresses this windows will show the status of the install. Once the installation is complete the next window is displayed.

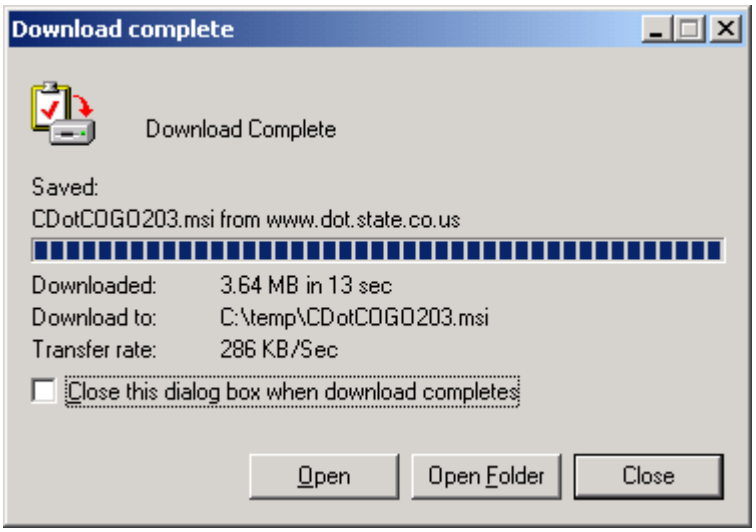
Click on the "OK" button to finish the installation.

Installing COGO via the Internet

COGO is available on the Colorado DOT web site at http://www.dot.state.co.us/. When this manual was published you select Planning/Construction, then Design and Construction Project Support, then Engineering Customer Support Unit (toward the bottom of the page), and then Download Area on the right. Click on CDOT COGO, then Download COGO. The following window will be displayed.

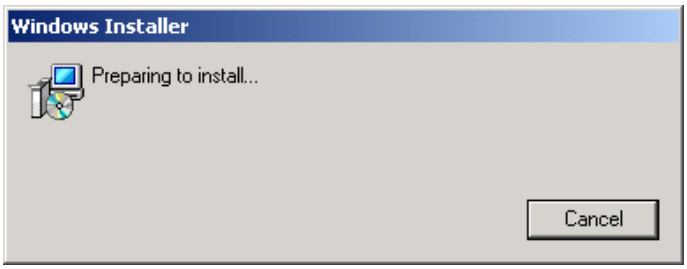


Click on the "OK" button and indicate where you would like the COGO ".MSI" file to reside. **Note:** you want to keep this ".MSI" file since this allows COGO to be self-repairing. Once the download is complete the following window will appear.



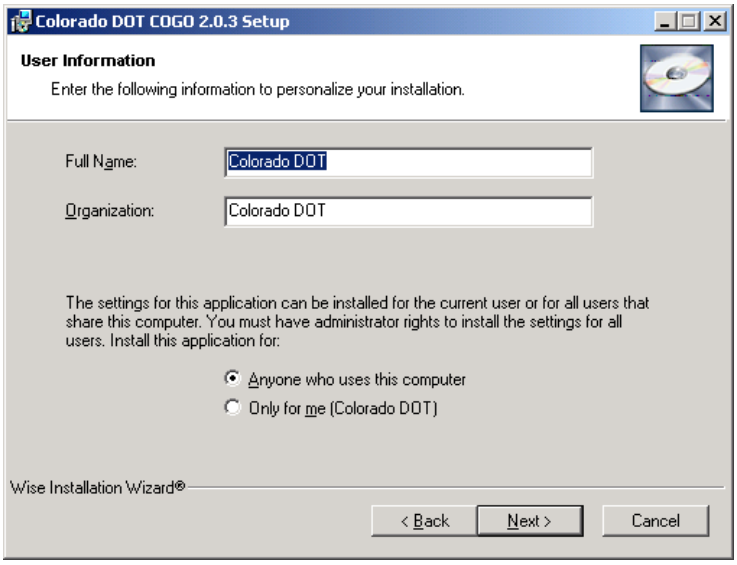
Click on the "Close" button.

Execute the ".MSI" file by double clicking on it. The following window will be displayed.

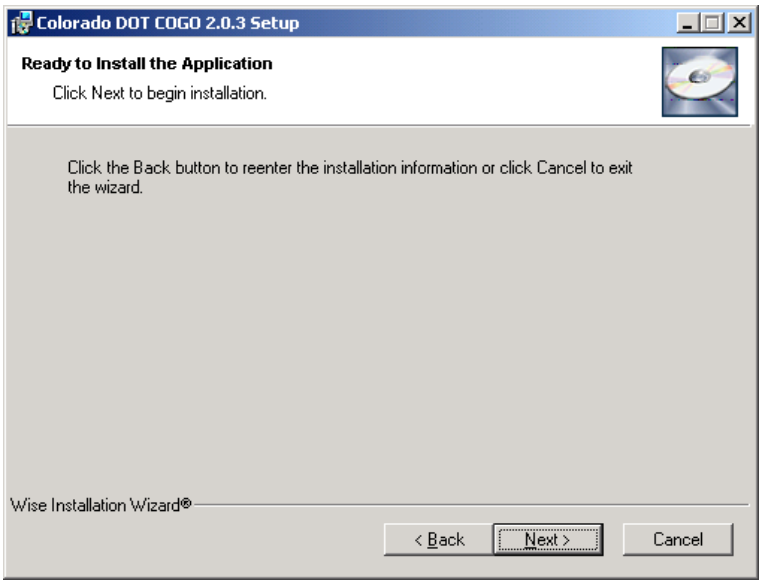


After finishing installation preparations, the following window appears.

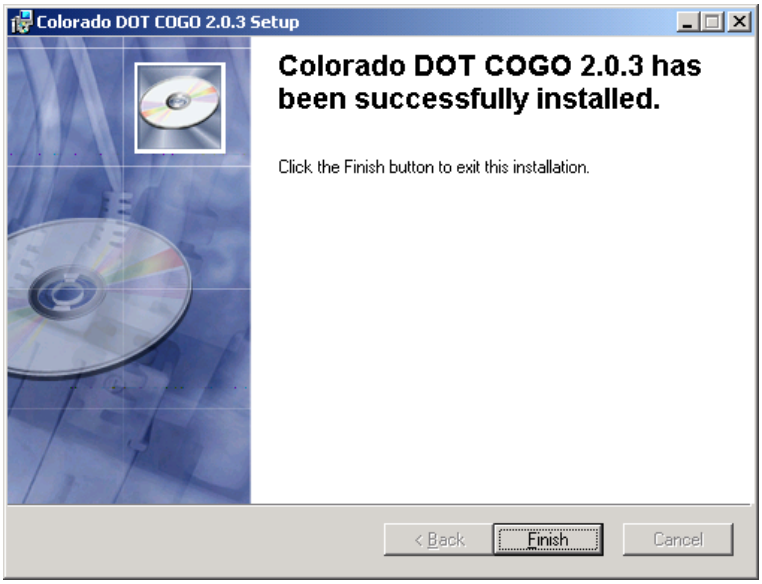
Click on the "Next" button and the next installation window appears.



Fill in the information requested. The radio buttons asking how you want to install COGO appear in this window only if you are installing COGO with an Administrative account. If you do see this, click on the "Anyone" radio button and then click on the "Next" button. The following window will appear.



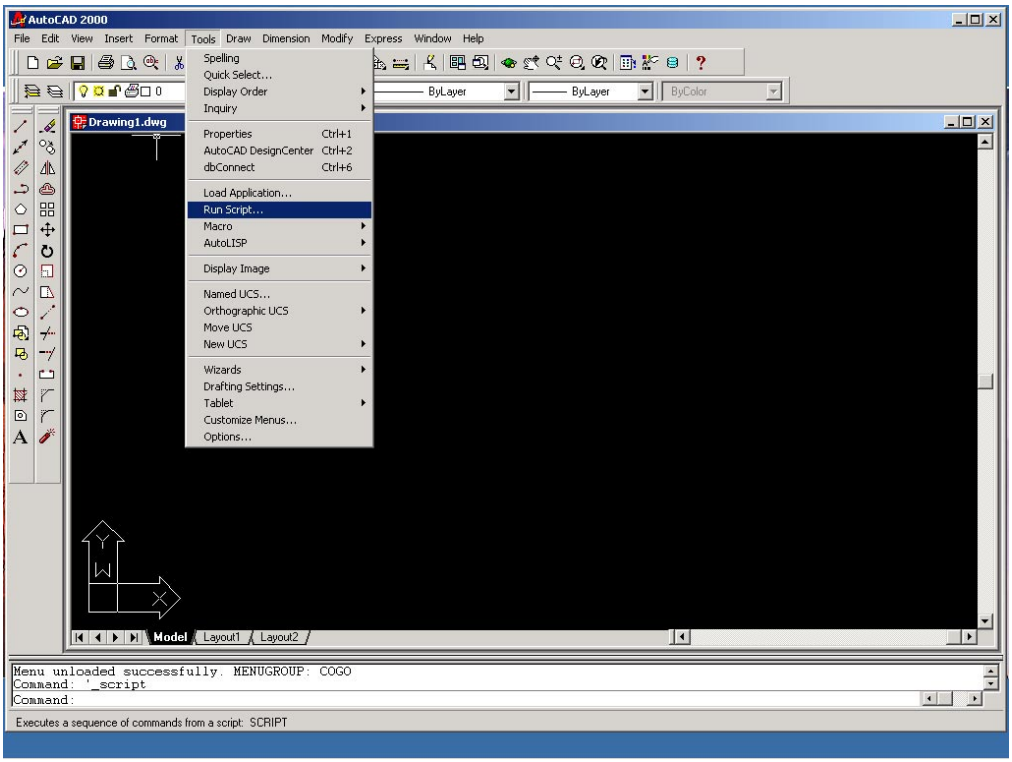
Click on the "Next" button. When the installation is complete the following window appears.



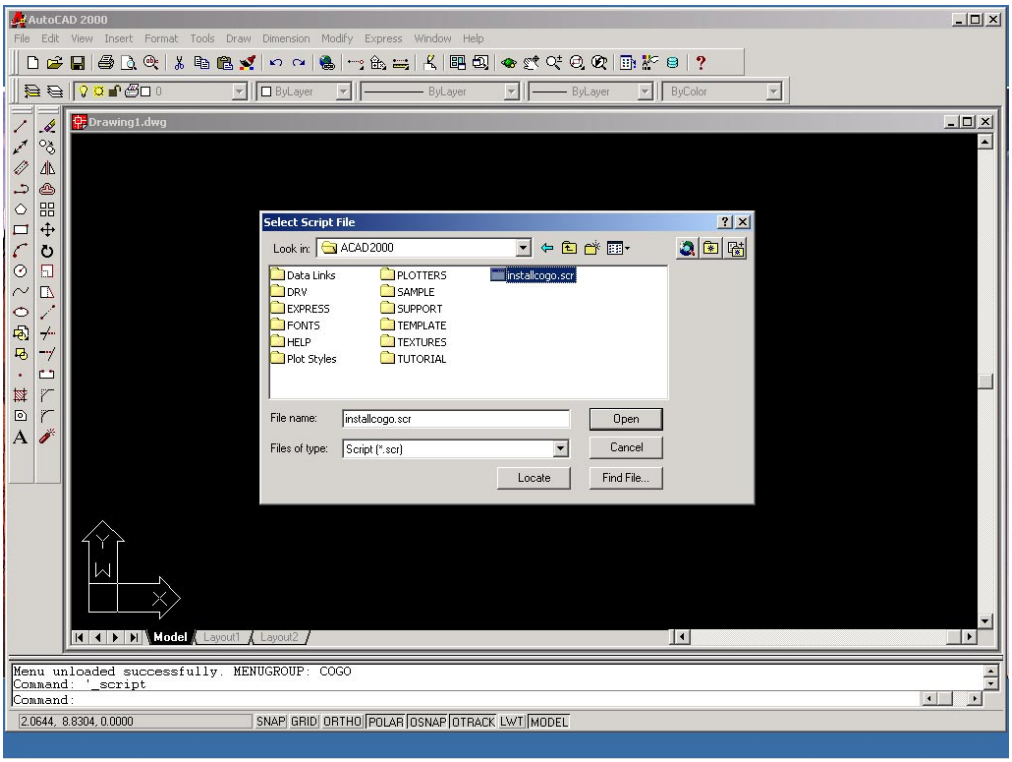
Click on the "Finish" button.

Setting up COGO in AutoCAD

In order to execute COGO in AutoCAD, one more step is needed. The COGO menus need to be loaded in AutoCAD. To do this, start AutoCAD and follow the instructions.



In AutoCAD, select "Run Script …" in the Tools menu.



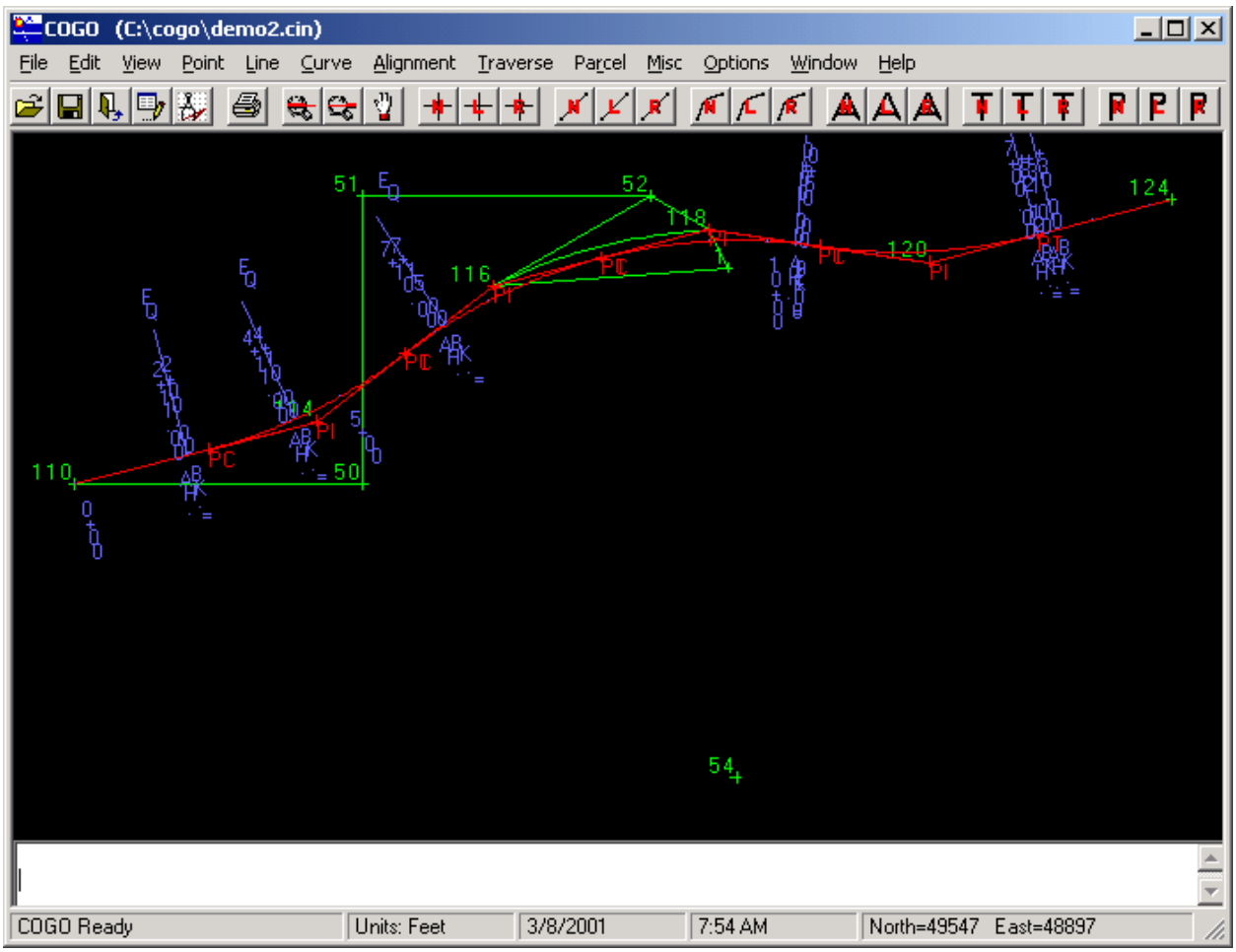
Select the script file "installcogo.scr" in the "C:\Program Files\AutoCAD 2002" folder and click on the Open button. Then close AutoCAD. This script will set up the COGO menus in AutoCAD 2002.

Executing COGO in Microsoft Windows

After COGO has been successfully installed, the folder C:\Program Files\CDOTCOGO will contain the file CDOTCOGO.exe. This is the executable file that is needed for executing COGO in Windows. If this file is missing the installation of COGO was not successful and COGO needs to be re-installed.

COGO may be executed in Windows via the Start | Programs | COGO menu selection.

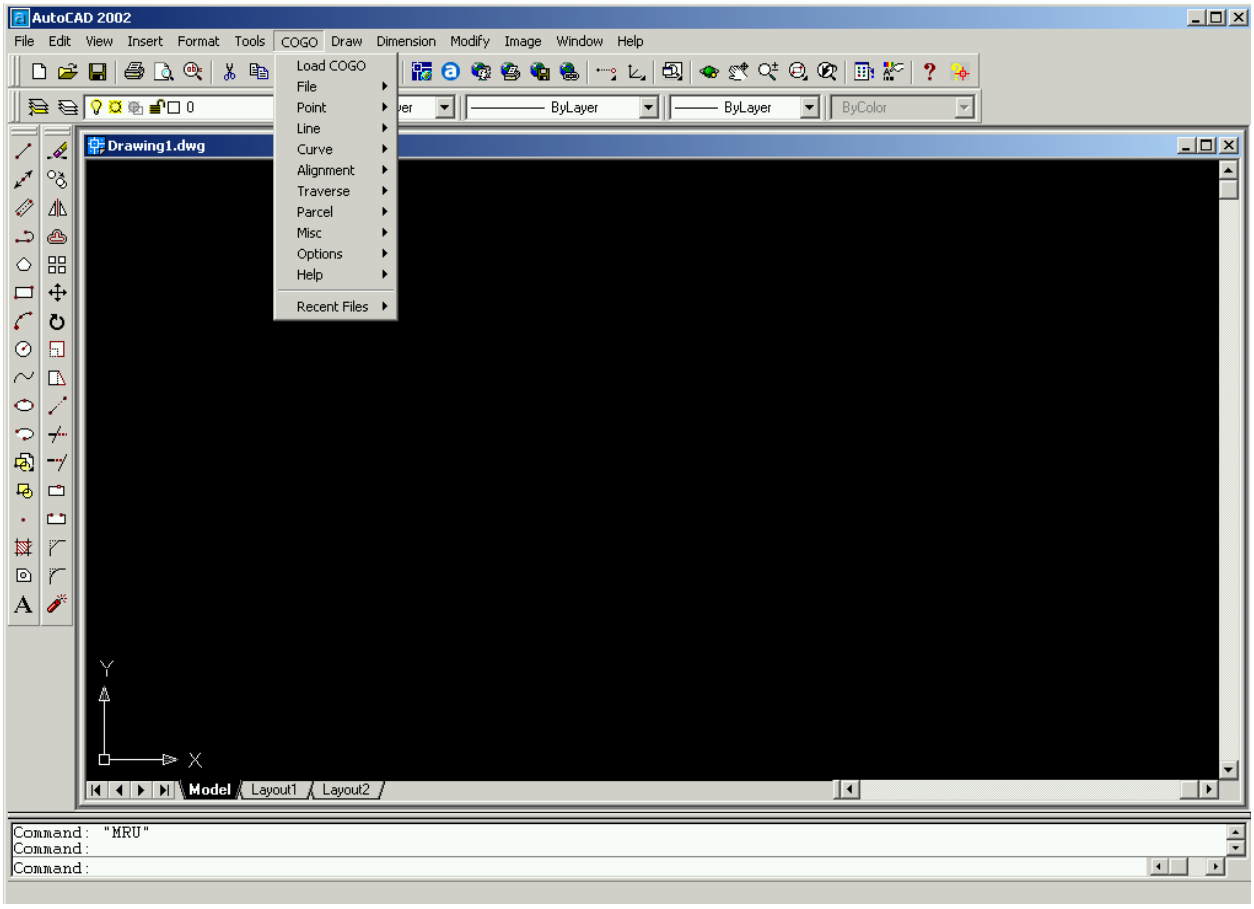
The following figure is how COGO appears in windows.



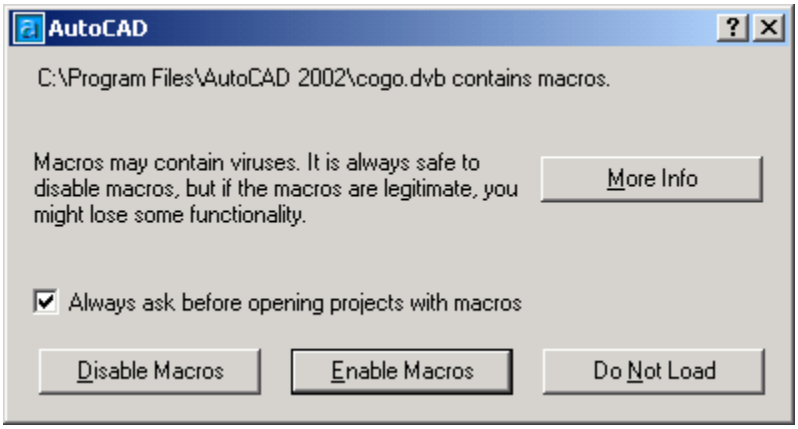
It is recommended that the settings for COGO accuracy (Options | Accuracy) be set or checked when COGO is executed in Windows.

Executing COGO Inside of AutoCAD

After successfully installing COGO, a COGO menu will appear in AutoCAD. Select the COGO menu.

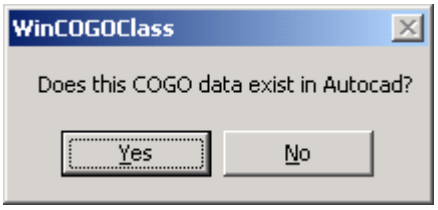


This is the COGO menu in AutoCAD. To Execute COGO in AutoCAD, click on the COGO menu item Load COGO in order to load the necessary macros for proper execution of COGO. AutoCAD may display a form that asks if you want to run macros, respond that you want to enable the running of macros.



After loading the COGO macros all other menu selections are active and may be used. The Load COGO must be done before COGO will execute properly in AutoCAD, and needs to be done only once per execution of AutoCAD.

After the COGO functions of File | Open or File | Restore Tables, COGO asks if the data already exists in the AutoCAD drawing.



If you answer Yes, none of the data just added to COGO will be added to the AutoCAD drawing. If you answer No, the data in COGO will be added to the AutoCAD drawing.

If you are short on disk space, or your disk is heavily fragmented, COGO may not execute well inside of AutoCAD.

A COGO works in only one of the AutoCAD windows (drawings). Using more than one instance of COGO in AutoCAD may have disastrous results. It is recommended that a number of settings in COGO and AutoCAD be checked or set before using COGO in AutoCAD. They are: the text size in COGO; the PDMODE and PDSIZE in AutoCAD; the Options | Accuracy settings for points and distances in COGO, and the object snap settings in AutoCAD.

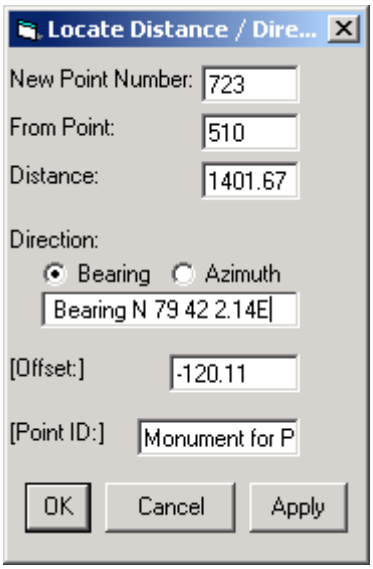
Chapter 2

Entering Data in COGO

This chapter covers the basic data types that you will use to create the geometry model, and the rules for entering the different data types. This chapter is essential reading.

As with all engineering applications, you need to supply data to COGO to create the model. This chapter covers the types of data you will enter, and the rules for entering the data. The basic types of data you will supply are numbers, characters, distances, directions, and alignment stations.

Entering Numbers



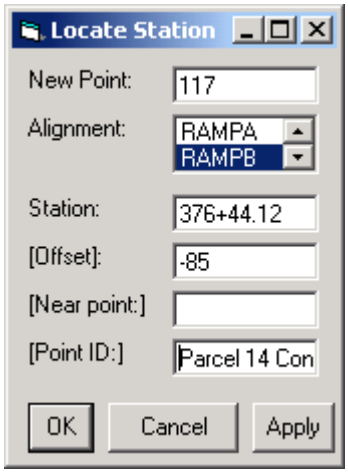
Numbers are used for point numbers, line numbers, curve numbers, coordinate data, distances, directions, stations, deflections, offsets, and a variety of other geometric data. Numeric data may be entered in many forms. The number 1 for example may be entered in any of the following ways.

1 +1 1. 1.0 01 01.0 etc.

If the decimal is omitted it is assumed to be after the last digit in the number. If the number is negative, the minus must be immediately before the first digit of the number. The characters which are legal in a number are 0 through 9, . (period), +, and -. If any other characters are present COGO will issue an error.

When the number of characters in the number exceeds the available space in the data entry area, as in the distance example (1401.6732), the ⇦ (left arrow) and ⇨ (right arrow) on the keyboard are used to move through the number.

Entering Characters



Character data is used in COGO for names, dates, file names, and comments. Character data may be composed of characters, characters and numbers, or numbers. Except for comments, all character data in COGO commands is surrounded by asterisks (*). COGO uses whatever is between the asterisks as characters. The asterisks are not necessary for most Windows forms and in cases where an item name is needed and the item type is known, the names of the items of that type will be listed so that you can scroll through the list and select the item.

Item names are limited to eight characters and may have blank spaces in them. Dates are entered in an eight-character format such as 11/26/01. File names are limited to eighty characters if entered in a command; otherwise the Windows rules apply for file names.

When the number of characters exceeds the available space in the data entry area, as in the Point ID example (Parcel 14 Control), the ⇦ (left arrow) and ⇨ (right arrow) on the keyboard are used to move through the characters.

Item Names

(8 characters maximum)

PROJECT *SURVEY* *TEST* *WB ML*

File Names

C:\project 1\cogo\survey.cin

D:\I70 and I25\COGO\I-70-WB

89022_JUNK_TEST.COT

Dates

(8 characters maximum)

01/12/25 (yy/mm/dd)

12/25/01 (mm/dd/yy)

25/02/01 (dd/mm/yy)

Entering Distances

All distances in COGO are expressed in feet and decimal feet or meters. COGO does not accept input of feet and inches. The rules for entering numbers apply to entering distances.

Point-to-point distances need not be calculated. Instead, COGO can calculate the point-to-point distance needed. This is done by the command word USE. The form for the USE option is

```
DISTANCE USE [POINT] n TO [POINT] m
```

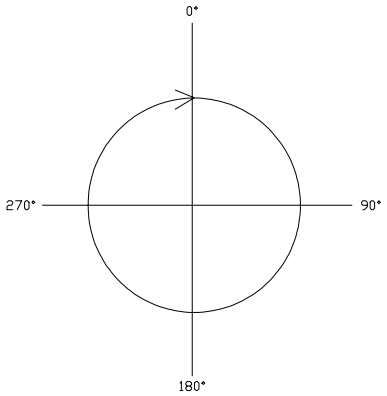
The above command option indicates to COGO that it should use the distance from point n to point m as the distance needed by the command COGO is executing.

Entering Directions

Directions may be expressed in azimuths, bearings, or zenith angles.

Azimuths

An azimuth is a horizontal measurement of angles between 0 and 360 degrees, north being 0 degrees and the azimuth increasing in a clockwise direction.



Azimuth

Azimuths may be expressed in terms of degrees and decimal degrees or as degrees minutes and seconds. If the minutes and seconds are zero they may be omitted. If the minutes are zero and the seconds are not, both must be entered. If the seconds are zero and the minutes are not, only the minutes need be entered. All of the following are valid azimuths.

0	0 0	0 0 0.0
10.5	10 30	10 30 0.0
270	270 00	270 0 0.0

45.255555 45 15 2. 45 15 2.0

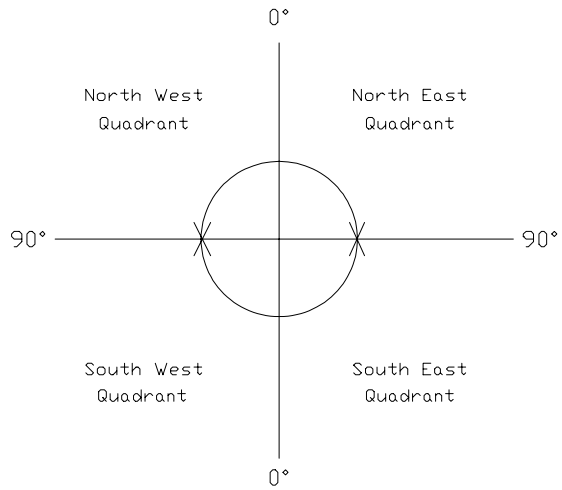
As with distances, COGO provides a function by which a point-to-point azimuth can be used with any COGO command or form without actually having to determine the azimuth between the points. This is done with the AZIMUTH USE option. The form for the AZIMUTH USE option is

```
AZIMUTH USE [POINT] n TO [POINT] m
```

This command option may be used in most areas that require a direction (azimuth or bearing).

Bearings

Bearings are horizontal measurements of angles based on quadrants. The four quadrants may be expressed in terms of the North East, North West, South West, and South East. Bearings are entered in terms of degrees minutes and seconds with the North or South in front of the bearing and the East or West after the bearing. The quadrants are indicated by the first letter in the word. For example, North is expressed as N.



Bearing

COGO will accept many forms of bearing input. The minimum requirements are that the North/South and East/West of the bearing and at least the degrees part of the bearing must be entered. If the minutes and seconds are zero they may be omitted. If the minutes are zero and the seconds are not, both must be entered. If the seconds are zero and the minutes are not, only the minutes need be entered. The proper form for a bearing may be expressed as

N/S dd mm ss.ss E/W

Where N/S indicates north or south, dd are the degrees, mm are the minutes, ss.ss are the seconds, and E /W indicates east or west. A space between the N/S and the degrees are not required, neither is a space required

between the ss.ss and the E/W. Spaces are required between the degrees, minutes, and seconds. COGO will accept all of the following examples of the same bearings.

N 00 E	S15 02W
N00 00 E	S15 02 00 W
N00 00 00.0E	S 15 02 00.0 W

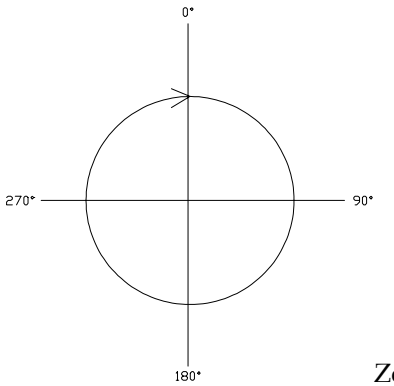
As with distances, COGO provides a function by which a point to point bearing can be used with any COGO command or form without actually having to determine the bearing between the points. This is done with the BEARING USE option. The form for the BEARING USE option is

```
BEARING USE [POINT] n TO [POINT] m
```

This command option may be used in most areas that require a direction (or bearing).

Zenith Angles

Zenith angles are vertical measurements of angles that range from 0° to 360°. 0 or 360 degrees being at the Zenith (straight up) and 90 or 270 degrees being the horizon.



Zenith Angles

Zenith angles may be expressed in terms of degrees and decimal degrees or as degrees minutes and seconds. If the minutes and seconds are zero they may be omitted. If the minutes are zero and the seconds are not, both must be entered. If the seconds are zero and the minutes are not, only the minutes need be entered. All of the following are valid zenith angles.

0	0 0	0 0 0.0
10.5	10 30	10 30 0.0
270	270 00	270 0 0.0
45.255555	45 15 2.	45 15 2.0

At this time zenith angles are only used for an angle right traverses.

Entering Stations

All stations in COGO are expressed in feet and decimal feet or in meters. COGO does not accept input of feet and inches. Stations are entered and represented in COGO as follows 1234+56.78 for feet, and 123456.78 for metric. There is no limit to the number of decimals. Blank spaces cannot precede or follow the plus for the feet stations. The following are valid examples.

Feet	Metric
10+46.893	1046.893
1120+15.27	112015.27
451+22.11	45122.11

Chapter 3

Creating Points

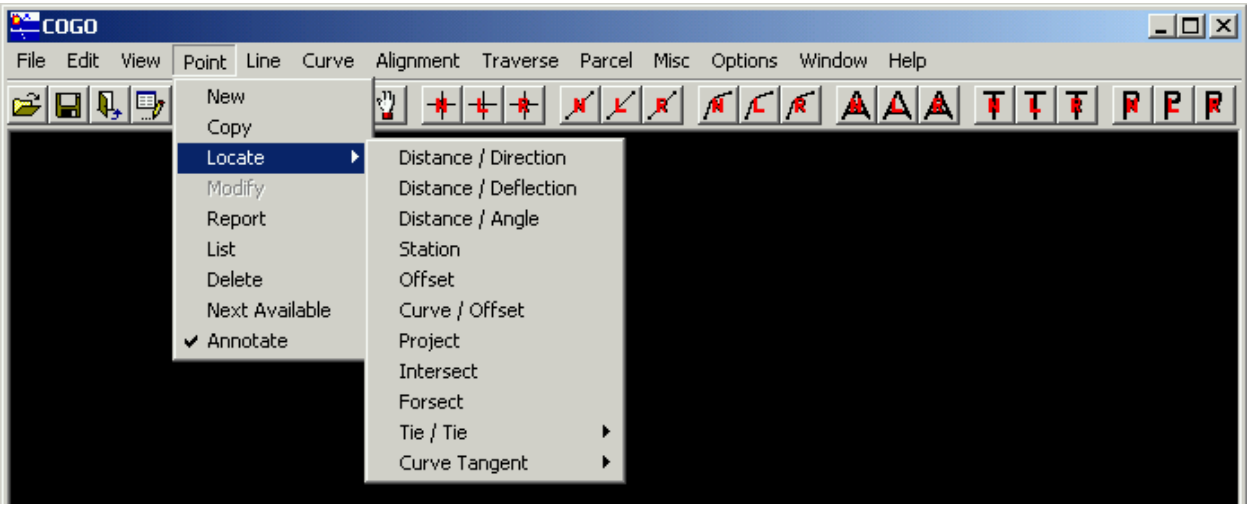
This chapter covers how to create points in COGO. The primary emphasis of this chapter is the use of the Windows forms and includes the equivalent syntax for the COGO command (refer to the COGO command reference for a complete description).

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

This chapter covers storing points, copying points, and creating points based on existing information.

Points are the basic building block for all work done in COGO. In order to use COGO effectively, a good understanding of how to create points is necessary.

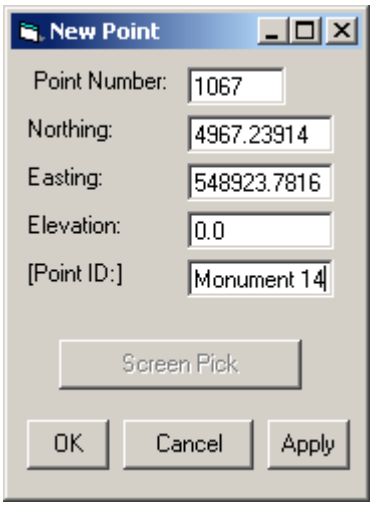
The Point Menu



The point menu in Windows has functions for creating new points by entering coordinates, copying a point, locating new points using existing items in the geometry model, displaying point reports, listing the point numbers in the model, deleting points, listing the next available unused point number, and if the points should be annotated with their point numbers.

Storing Points

The minimum information needed to store a point is the point number, northing and easting. Optional information for a point is the elevation, and point identifier (PID).

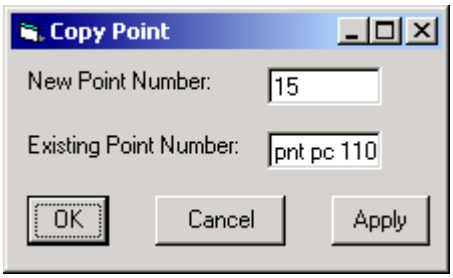


This is the Windows form for storing a new COGO point. The same form is used in Windows and in AutoCAD. In this example the point number is 1067, the northing is 4967.23914, the easting is 548923.786, the elevation is 0.0, and the point identifier (ID) is Monument 14.

There are four buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may store another point. The Screen Pick button is only active when executing COGO in AutoCAD. If you use the Screen Pick button in AutoCAD, the New Point form will be hidden, you will be prompted to select a point in AutoCAD, COGO will capture the coordinates of the point selected in AutoCAD, and then the form will be displayed so that you can enter the point number, elevation, and point ID.

STORE [POINT] p [NORTH] n [EAST] e [ELEV el] [*id*]

Copying Points



You use the copy point form to copy the northing, easting, elevation, and point ID to another point. The required information is the existing point number and the point number to copy the information to. In this example, the coordinates for the PC of curve 110 are copied to point 15. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may copy another point.

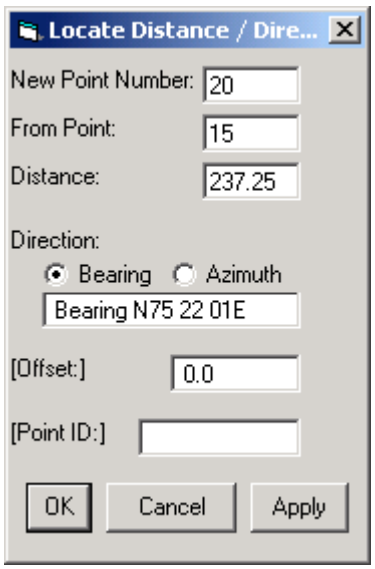
STORE [POINT] p POINT k [*id*]

Locating Points Using Existing Information

COGO has many commands that locate new points based on existing information. Based on the information available different commands are used to locate points. The following sections cover the commands that locate new points.

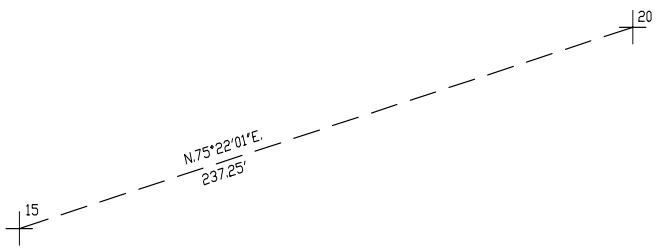
Using a Distance and Direction to Locate a Point

One of the most common ways to define a new point is to locate the new point based on a distance and direction from another point.



Point 20 will be 237.25 feet from point 15 at a bearing of N75 22 01E.

The required information is the existing point number, the new point number, the distance and direction from the existing point. Optional information is the point ID and an offset. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

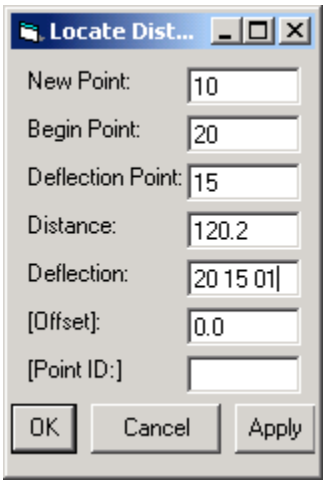



```
LOCATE [POINT] n [*id*] FROM [POINT] m distance direction [offset]
    e.g. LOCATE pnt 20 FROM 15 DISTANCE 237.25 BEARING N75 22 01E
```

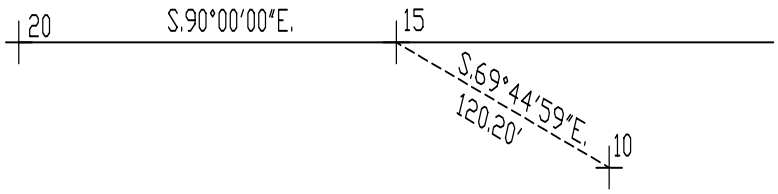
Using a Direction from a Line Segment to Locate a Point

Two commands in COGO will locate a new point based on a line segment and a direction. The first uses a deflection from the end of the line segment to locate a point. The second uses an angle from the beginning of the line segment to locate a new point.

Deflection Method

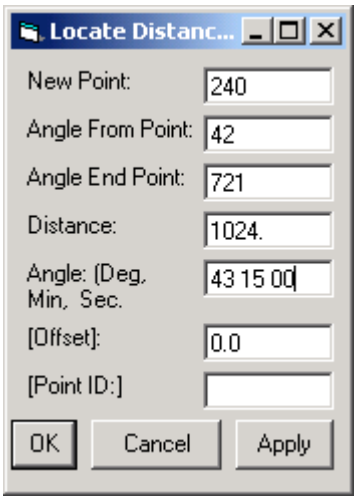


The minimum data required for this command are two points that define a line segment, a distance, and a deflection angle from the direction of the line segment. The direction of the line segment is from the first point in the segment to the second. The deflection is applied at end point of the line segment. If the deflection is positive the new point will be on the right, a negative deflection results in the point being on the left. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

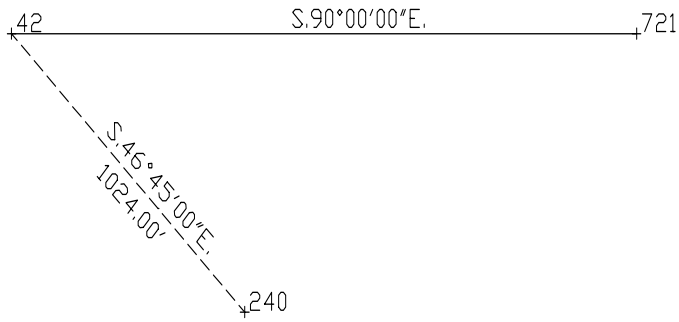


LOCATE [POINT] n *id* FROM [POINT] m TO [POINT] p distance deflection [offset]
e.g. LOCATE 10 FROM pnt 20 TO 15 DIST 120.2 DEFL 20 15 01
(This command will locate point 10 120.2 feet from point 15 at a deflection of 20 degrees 15 minutes and 1 second to the right from the angle of the line segment from point 20 to 15.)

Angle Method

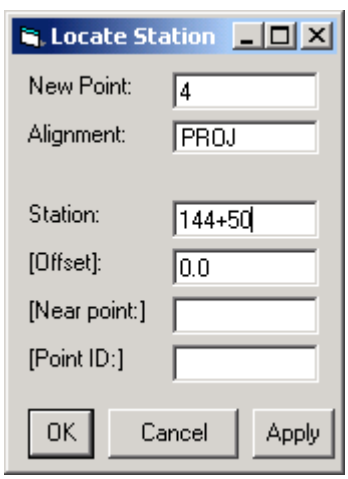


The minimum data required for this command are two points that define a line segment, a distance, and an angle from the direction of the line segment. The direction of the line segment is from the first point in the segment to the second. The angle is applied at the starting point of the line segment. If the angle is positive the new point will be on the right, a negative angle results in the point being on the left. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

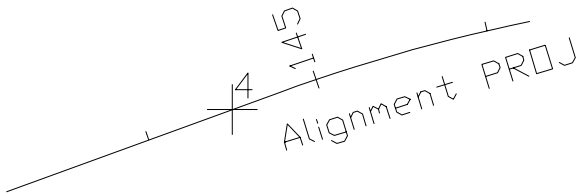


LOCATE [POINT] n *id* FROM [POINT] m distance angle FROM [POINT] p [offset]
e.g. LOC 240 FROM pnt 42 1024.0 angle 43 15 00 FROM 721
(This command will locate point 240 1024 feet from point 42 at an angle of 43 degrees 15 minutes from the direction of the line segment from point 42 to 721.)

Using a Station to Locate a Point

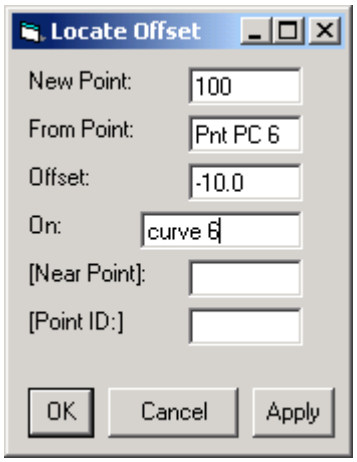


The minimum information needed is the new point number and a station on the alignment. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

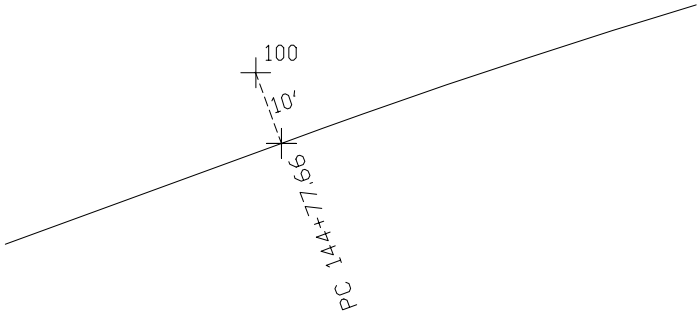


LOCATE [POINT] n *id* ON ALIGNMENT *name* [offset] AT [STA] station [NEAR [POINT] p]
e.g. LOC 4 *STA 144+50* ON ALI *PROJ* AT STA 144+50.0
(Point 4 is created with coordinates corresponding to station 144+50.0 on alignment PROJ.)

Locating an Offset point

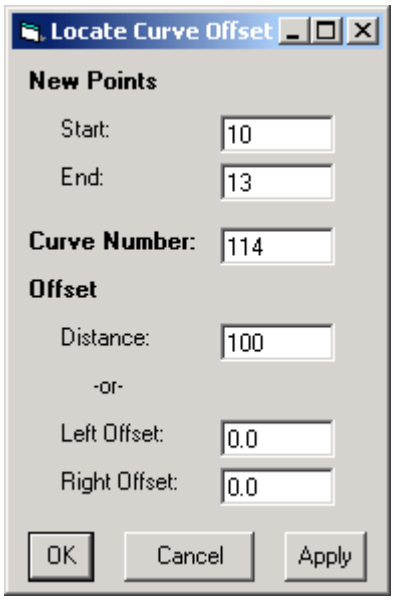


This function is for finding a point offset 90 degrees from an item. The minimum information necessary is a point on the item, the offset distance from that point and the item. Items that this command will accept are line segments, curves, and alignments. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

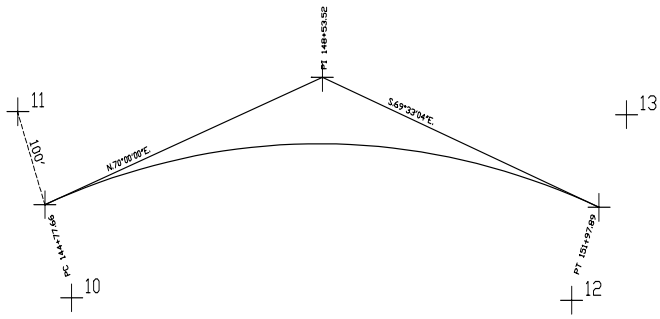


LOCATE [POINT] a *id* OFFSET distance FROM [POINT] p ON item [near]
e.g. LOC pnt 100 OFFSET -10 FROM PNT PC 6 ON CURVE 6
(Point 100 will be 10 feet to the left of the point of curvature (PC) on curve 6.)

Offsetting Points from a Curve

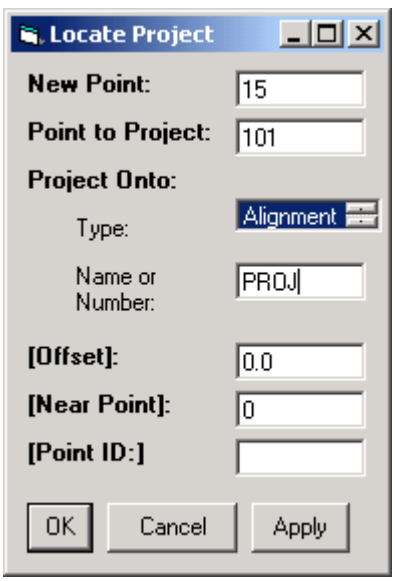


This function creates points offset 90 degrees (normal) to the left and right from all curve points except the PI and CC. The minimum information required is a curve, the offset distance to the right and left, and a group of consecutive point numbers that are used to store the new offset points. The number of new points is dependent on the type of curve. A simple curve and stand alone spiral generates 4 points, a complex curve generates 6 to 8 points depending on the number of spirals in the complex curve. This command uses the point identification to indicate which curve point the new point is offset from and if it is to the left or right of the curve point. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

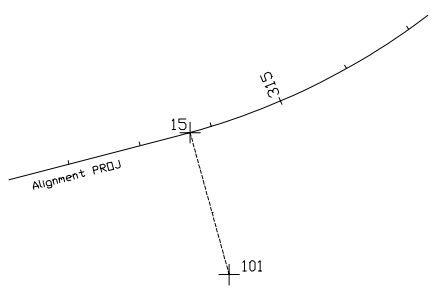


LOCATE [POINT] n TO m OFFSET (d / r RIGHT 1 LEFT / 1 LEFT r RIGHT) FROM CURVE c
e.g. LOC POINTS 10 TO 13 OFFSET 100 FROM CURVE 114
(Curve 114 is a simple curve so 4 new points (10, 11, 12, and 13) will be created. Points 10 and 11 are offset 100' to the left and right of the PC and points 12 and 13 are offset 100' to the left and right of the PT.)

Projecting Points on a COGO Item

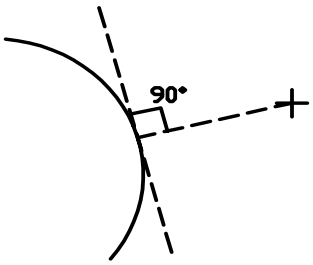


This function projects a point onto a COGO object. This means that the line segment from the new point to the projected point will be perpendicular (normal) to the COGO object. The minimum information required is a point near a COGO object and the object. Some projections have more than one solution, others have no solution. A good example of a projection that has more than one solution is the projection of the center of a curve on to the curve, a situation that has infinite solutions. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point. The diagram on the following page portrays possible and impossible projections.

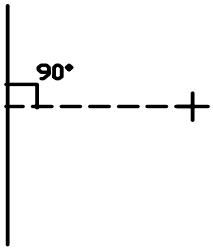


LOCATE [POINT] n *id* PROJECT [POINT] m ON item [offset] [near]
e.g. LOCATE 15 PROJECT PNT 101 ON ALI *PROJ*
(Point 101 will be projected on to alignment PROJ, the resulting point stored in point 15)

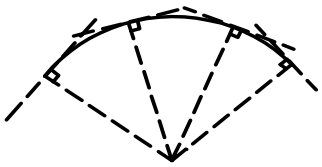
Possible Projections



One Solution

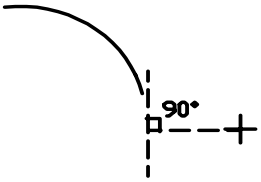


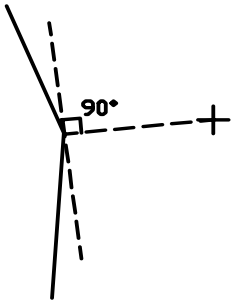
One Solution

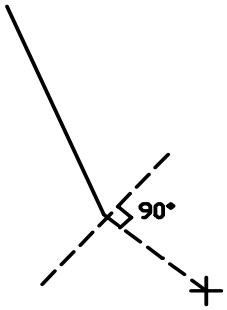


Infinite Solutions

Projection Not Possible

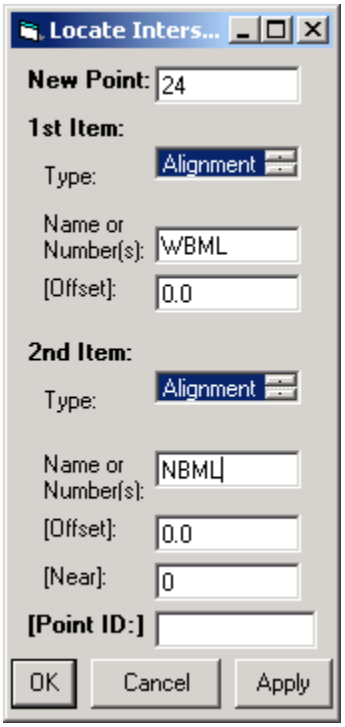




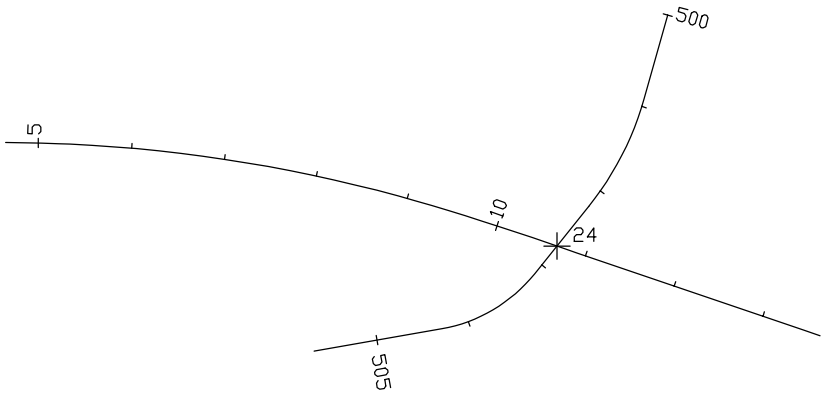


COGO has an option which specifies which projection point is desired. It is the NEAR POINT option. The command syntax for the near option is **`NEAR [point] n`**. The solution to the projection will include the point, the distance to the COGO object and the bearing of the line of projection. If the point is projected on an item that has stationing, the station will be reported.

Creating Points by Intersecting Two COGO Objects

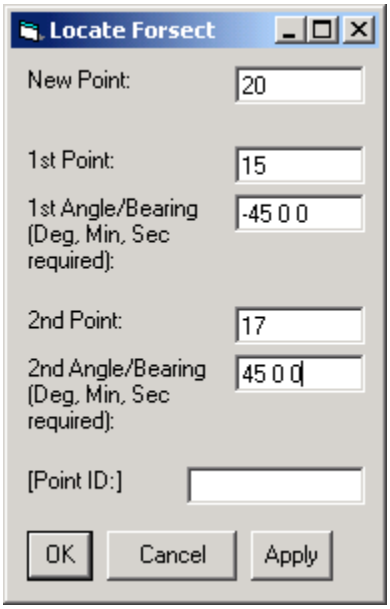


This function intersects two COGO items. The COGO items which can be intersected are lines, line segments, curves, and alignments. The minimum information required is the two objects that intersect. Some intersections have more than one solution, others have no solution. A good example of an intersection which has more than one solution is the intersection of a line which passes through the beginning and end points of a curve, a situation which has two solutions. COGO has a command option which allows the user to specify which intersection point is desired. It is the NEAR POINT option. The command syntax for the near option is **`NEAR [point] n`**. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

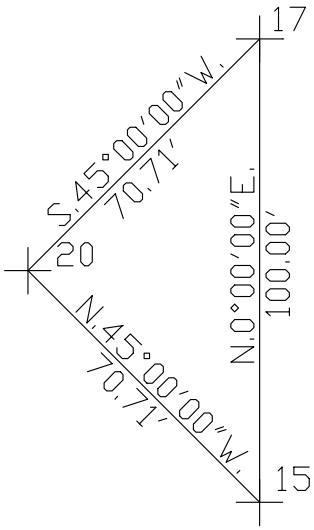


`LOCATE [POINT] n *id* INTERSECT item1 [offset] WITH item2 [offset] [near]`
`e.g. LOCATE 24 INTERSECT ALI *WBML* WITH ALI *NBML*`
(The coordinates of the intersection of alignment WBML and NBML will be stored in point 24.)

Creating Points by Turning Angles from the Ends of a Line Segment

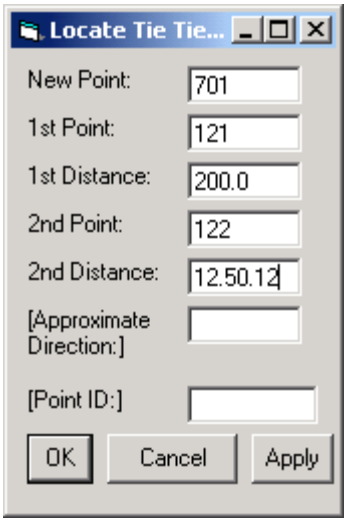


This function uses a line segment and deflections turned from the end points of the line segment to locate a new point. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

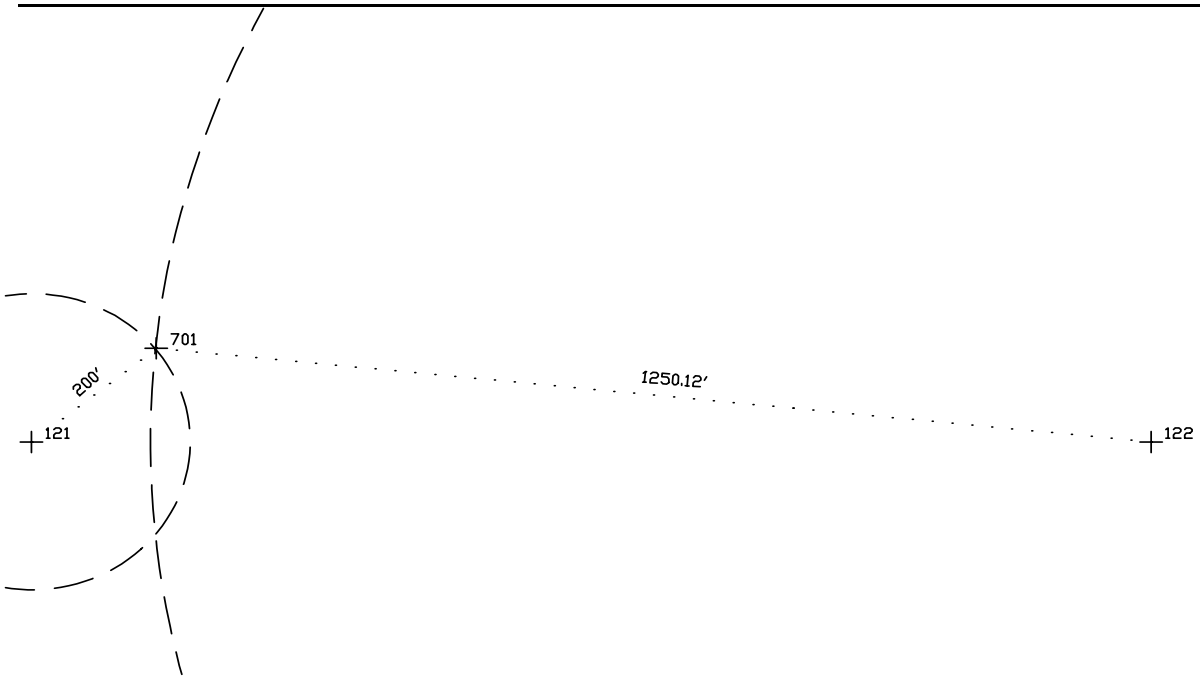


LOCATE [POINT] n [*id*] FORSECT [POINT] m (ANGLE1/BEARING1) AND [POINT] p (ANGLE2/BEARING2)
e.g. LOCATE 20 FORSECT PNT 15 MINUS 45 0 0 AND PNT 17 45 0 0
(The direction of the line segment from point 15 to point 17 is calculated, and then the deflections are applied to locate point 20.)

Locating a new point using two distances from two points

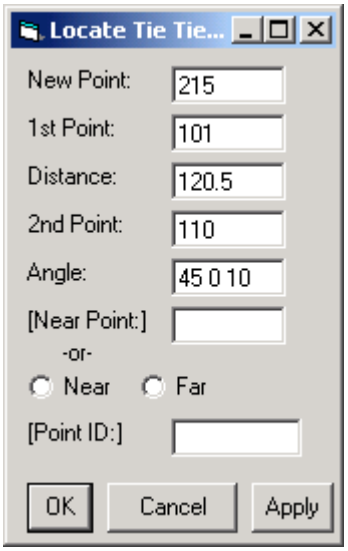


This function finds the point that is a distance from one point and another distance from another point. It is possible that there is more than one solution. If more than one solution is found, two options are provided to indicate which solution is desired. One of the options allows you to specify an approximate direction where the solution lies. The other option provides the ability to indicate a point that is near the solution. The required information is two points and distances from those points. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

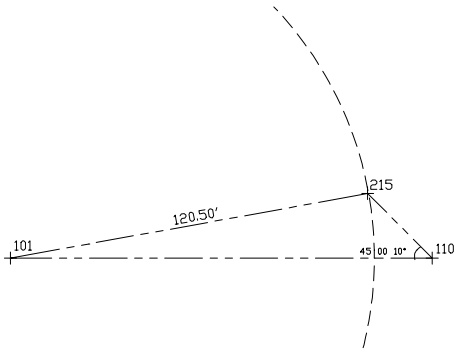


LOCATE [POINT] n [*id*] TIE [POINT] m DISTANCE d1 TIE [POINT] p DISTANCE d2 [(APPROX direction / near)]
e.g. LOCATE PNT 701 TIE PNT 121 DIST 200.0 TIE PNT 122 DIST 1250.12
(Point 701 is an intersection of the circle with the center at point 121 and a radius of 200', and the circle with the center at point 122 and a radius of 1250.12')

Locating a new point on a line, a distance from a point

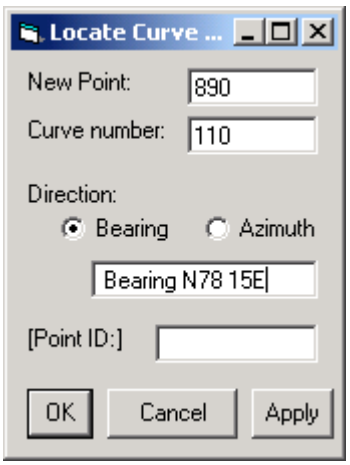


This command finds a point on a line that is a specific distance from an existing point. The user can control which solution is selected by three command options. The options allow the user to specify the solution should be near a specified point or near the starting point of the line segment or the solution farthest away from the starting point of the line. The required information is a point and the distance from the point, the starting point of the line and the direction of the line. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

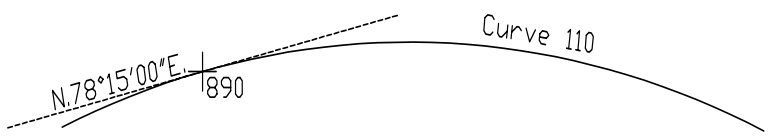


LOCATE (POINT) n *id* TIE [POINT] m distance TIE (POINT] p direction (NEAR FAR / NEAR [POINT] t)
e.g. LOC 215 TIE 101 DIST 120.5 TIE 110 ANGLE 45 00 10
(Point 215 is the intersection of the line that starts at point 110 with an azimuth of 45 degrees 00 minutes and 10 seconds, and the circle with the center at point 101 and a radius of 120.5)

Locating a point of tangency to a curve, using a direction

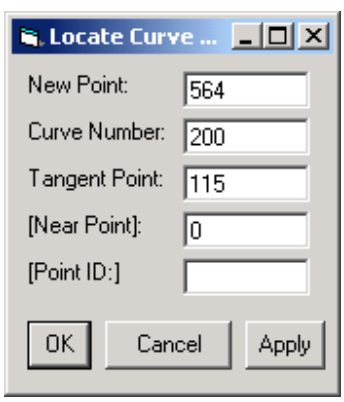


This function finds a point where a line of a specific direction is tangent to a curve. The minimum information is the direction of the line and a curve. Since curves are defined as a circular arc of less than 180 degrees there will be one solution, if there is a solution. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

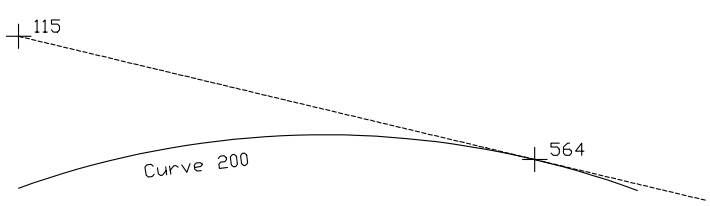


LOCATE [POINT] n [*id*] ON CURVE c TANGENT direction
e.g. LOC PNT 890 ON CUR 110 TANGENT N 78 15 E
(Point 890 will be located at the point where the tangent line to the curve has the direction N 78 15 E.)

Locating a Point of Tangency to a Curve, Using a Point Near the Curve



This function locates a point on a curve using a tangent that passes through another point. There may be two solutions to this problem. Specifying the desired solution is accomplished by indicating a point that is closer to the desired solution. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.



LOCATE [POINT] n [*id*] ON CURVE c TANGENT THRU [POINT] m [near]
e.g. LOC 564 ON CUR 200 TANGENT THRU PNT 115
(A line which passes through point 115 and is tangent to curve 200 is found, then the coordinates of the point at which that line is tangent are placed in point 564)

Other Functions That Create Points

In addition to the point functions, new points can be created by dividing lines, line segments, or curves into smaller parts, creating new points (see the section on Divide Lines and Divide Curve).

Creating new points can also be done with the two traverse functions. The first traverse function uses distances and directions from a point to create a new point. The second traverse function creates new points using the angle right traverse survey method.

Chapter 4

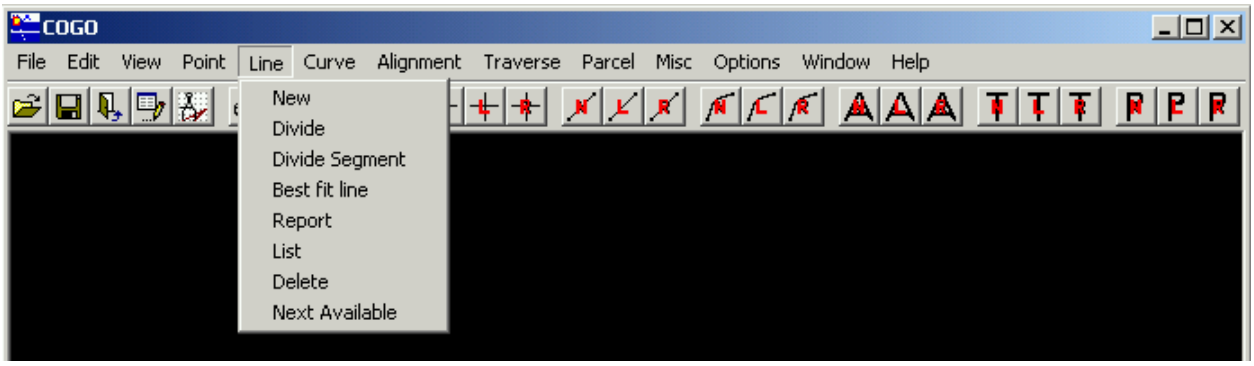
Creating and Manipulating Lines

This chapter covers how to create and manipulate lines in COGO.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

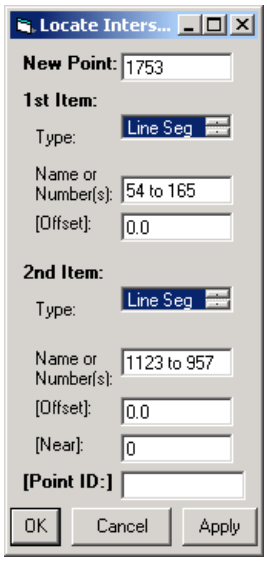
This chapter covers how to create the three different line types in COGO, divide lines to create points, and apply linear regression to a collection of points that represent a line. As mentioned in the first chapter there are three different line types, and the type is dependant on the way you create the line.

The Line Menu



The line menu in Windows has functions for creating new lines, dividing a line into parts to create new points, adjusting a line of points by linear regression, displaying line reports, listing the line numbers in the model, deleting lines, and listing the next available unused line number.

Using Temporary Line Segments

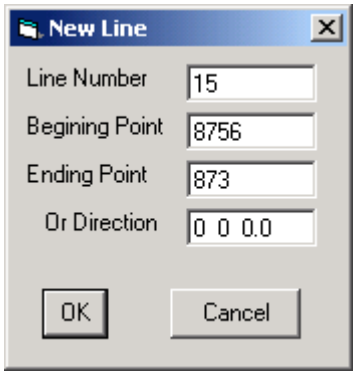


Temporary line segments exist only as long as the function is being performed. You may use the temporary line segment wherever you would use a line segment or line. For example, you can intersect two temporary line segments as in the form to the left. A soon as the calculations are completed and the point is stored, the temporary line segments are gone.

In a COGO command the temporary line segment is created with the following syntax.

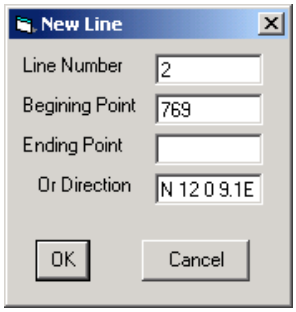
```
LINE SEG [POINT] n TO [POINT] m
```

Storing Line Segments



Line segments are stored in the geometry model and referred to via their line number. The required information to create a line segment is the line number, the beginning point, and the ending point. There are two buttons at the bottom of the form. The OK button saves the line segment and closes the form. The Cancel button closes the form without saving any information. In this example line 15 starts with point 8756, ends at point 873 and is a line segment.

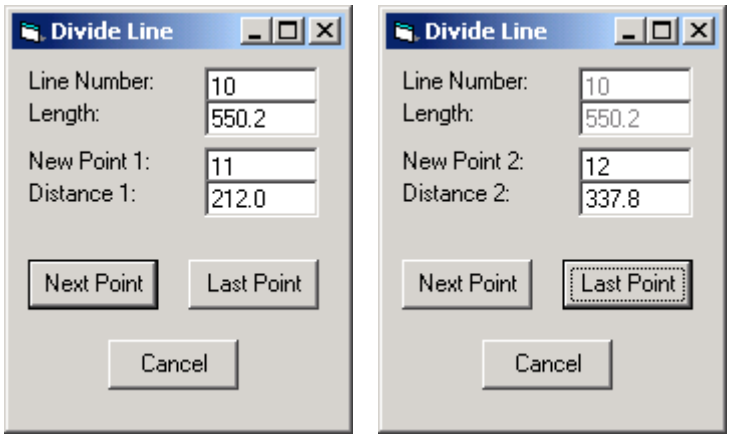
Storing Lines or Rays



Lines are stored in the geometry model and referred to via their line number. The required information to create a line is the line number, the starting point and the direction of the line. There are two buttons at the bottom of the form. The OK button saves the line and closes the form. The Cancel button closes the form without saving any information. In this example line 2 starts at point 769 and goes infinitely at a bearing of N 12 00 9.1 E.

Dividing Lines to Create Points

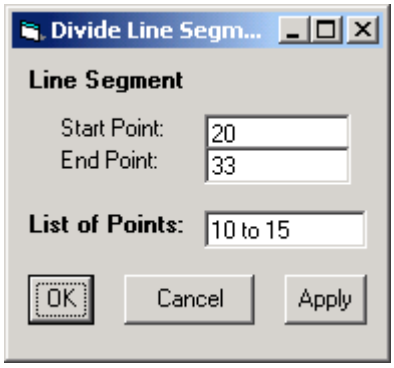
The dividing of a line is done via the Divide Line function; the dividing of a line segment is done via the Divide Line Segment function.



The required information for the Divide Line function is a line number, a length for the line (this may be different from the actual length of a line segment), and a list of points and distances. The line number, length, and information for the 1st new point is entered, then the Next Point button is clicked. Then the next point's information is entered and the next point button is clicked, until the last point information is entered and the Last Point button is clicked.

! After the first point is entered the line number and the line length cannot be changed.

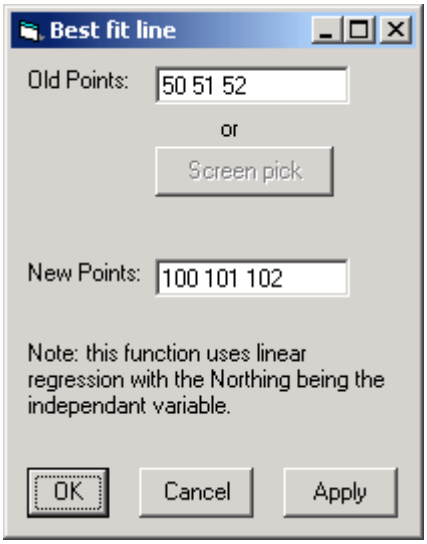
DIVIDE LINE 1 length [POINT] p1 [DISTANCE] d1 [POINT] p2 [DISTANCE] d2 etc.
e.g. DIVIDE LINE 10 550.2 PNT 11 212.0 PNT 12 337.8
(Line 10 is 550.2 feet long. Starting at the beginning point in line 10, the coordinates for point 11 will be 212.0 feet in the direction of the line. Point 12's coordinates will be 337.8 feet from point 11 in the direction of line 10.)



The required information for the Divide Line Segment function are two points which define the line segment, and a list of points. The line segment will be divided into equal length segments. The number of segments is one more than the number of points in the list of points. There are three buttons at the bottom of the form. The OK button saves the points and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the point but leaves the form up so that you may locate another point.

DIVIDE LINE SEGMENT [POINT] n TO [POINT] m list of points
e.g. DIVIDE LINE SEGMENT 20 TO 33 10 TO 15
(The line segment from point 20 to 33 is divided into 7 parts of equal length. The coordinates of the division are stored in points 10 to 15 (six points).)

Best Fit Line



The Best Fit Line function uses linear regression to adjust a series of points. The linear regression algorithm uses the northings to adjust the eastings. You enter the point numbers for the points to be adjusted as the Old Points, and COGO will adjust the points and store the results in the New Points that you specify. The Screen Pick button is only active in AutoCAD and allows you to pick points in AutoCAD rather than having to type in the point numbers. There are three buttons at the bottom of the form. The OK button saves the point and closes the form. The Cancel button closes the form without saving any information. The Apply button saves the adjusted points but leaves the form up so that you may adjust more points.

This function has no equivalent COGO command.

Chapter 5

Creating Curves

This chapter covers how to create and manipulate curves.

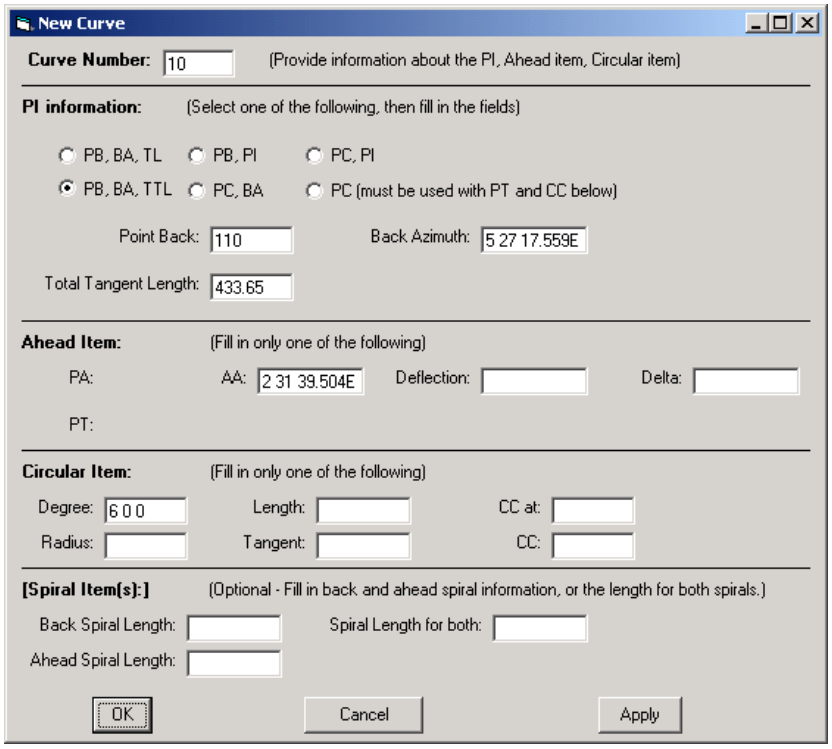
ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

Curves may be created in a number of ways. Curves may be created separately or during the creation of an alignment. This section covers the methods of creating and using curves by themselves and not as part of an alignment. Please refer to the section on alignments for more information on creating and using curves within an alignment.

The Curve Menu

The curve menu has functions for storing new curves, dividing a curve into parts to create new points, displaying curve reports, listing the curve numbers in the model, deleting curves, and listing the next available unused curve number.

Creating a Curve



The required information for creating a new curve is the curve number, information about the Point of Intersection (PI) for the curve, ahead information, circular information, and optional spiral information. You must first select the PI information you want to enter, then fill in the rest of the fields as necessary. Note that the PI selection controls which fields are displayed in the New Curve form. There are three buttons at the bottom of the form. The OK button creates the curve and closes the form. The Cancel button closes the form without saving any information. The Apply button creates the curve and leaves the New Curve form on the screen.

In the above example, curve 10 is created using the Point Back, Back Azimuth, and Total Tangent Length for the PI information. The ahead item is the Ahead Azimuth of N52 31 39.504E, and the Circular Item is the 6 0 0 Degree of Curvature. Note that in the Ahead Item section the PA and PT are not available due to the selection of the PB, BA, TTL in the PI Information section. See the next section for further information.

The following is a description of the fields in the New Curve form.

PI Information

BA Is the back azimuth or bearing

PB Is the point on the back tangent. It may be a point number, curve point, or north east station.

PI Is the point of intersection of the tangents. It may be the same form as the PB without STA.

TS Is the tangent to spiral point. It may be the same form as the PB.

PC Is the tangent to circular curve point (point of curvature). It may be the same form as the PB. Note The PC n form may only be used in conjunction with the ahead ref item PT n and the circular ref. item CC n.

TTL Is the total tangent length, PB to PI.

TL Is the tangent length, PB to TS/PC.

Ahead Item

PA Is the point on the ahead tangent. It may be the same form as the PB. This point may be on either side of the end of the curve. Note PA can only be used if a PI is given.

AA Is the ahead azimuth or bearing.

DEFL Is the total deflection at the PI. + indicates right, - indicates left.

DELTA Is the circular curve delta. + indicates right, - indicates left.

PT Is the point of tangency. Note: this option can only be used with a PC n and a CC n.

Circular Item

RADIUS Is the radius of the curve.

DEGREE Is the degree of curvature.

LENGTH Is the length of the curve. (only if delta or deflection given, non spiraled)

TANGENT Is the tangent length (TS/PC to PI or PI to ST/PC). This may only be used with a DELTA ahead ref. (only if delta or deflection given, non spiraled)

CC AT n Is the circular curve center. It can be the same form as the PB. (only if non-spiraled)

CC n Is the point where the curve center resides. Note This can only be used with the PC n and PT n.

[Spiral Item]

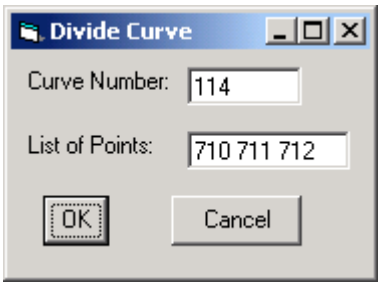
SLB Is the back spiral length.

SLA Is the ahead spiral length.

SL Is the spiral length for in and out spirals.

Dividing a Curve to Create New Points

A curve may be divided into equal length segments with the Divide Curve function. The required information is a curve number and a list of points. The curve will be divided into equal length segments. The number of segments is one more than the number of points in the list of points. There are two buttons at the bottom of the form. The OK button saves the points and closes the form. The Cancel button closes the form without saving any information.




```
DIVIDE CURVE c list of points
e.g. DIVIDE CURVE 114 710 711 712
```

Chapter 6

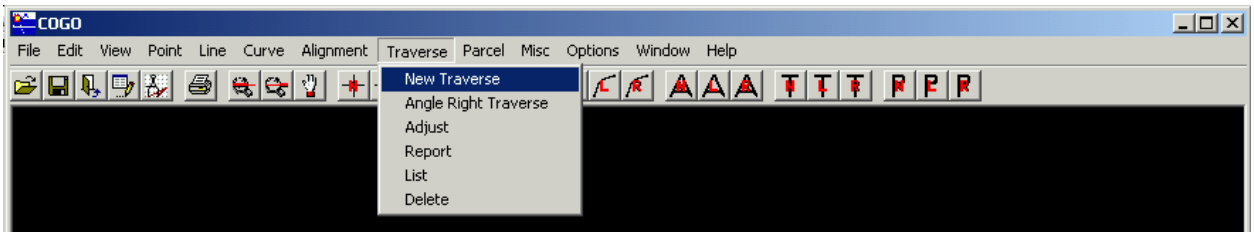
Creating and Adjusting Traverses

This chapter covers how to create traverses and how to adjust a traverse.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

This chapter covers the three methods of creating a traverse and the Adjust Traverse function. The three methods of creating a traverse are storing a traverse using existing points, creating a traverse using the distance / direction pairs to create the traverse and the points in the traverse, and the angle right survey method to create the traverse and the points in the traverse. Once the traverse is created it may be adjusted with the Adjust Traverse function.

The Traverse menu



The Traverse menu can create a new traverse, process the information from an angle right traverse, generate a traverse report, list the traverses in the model, and delete traverses.

Storing a Traverse

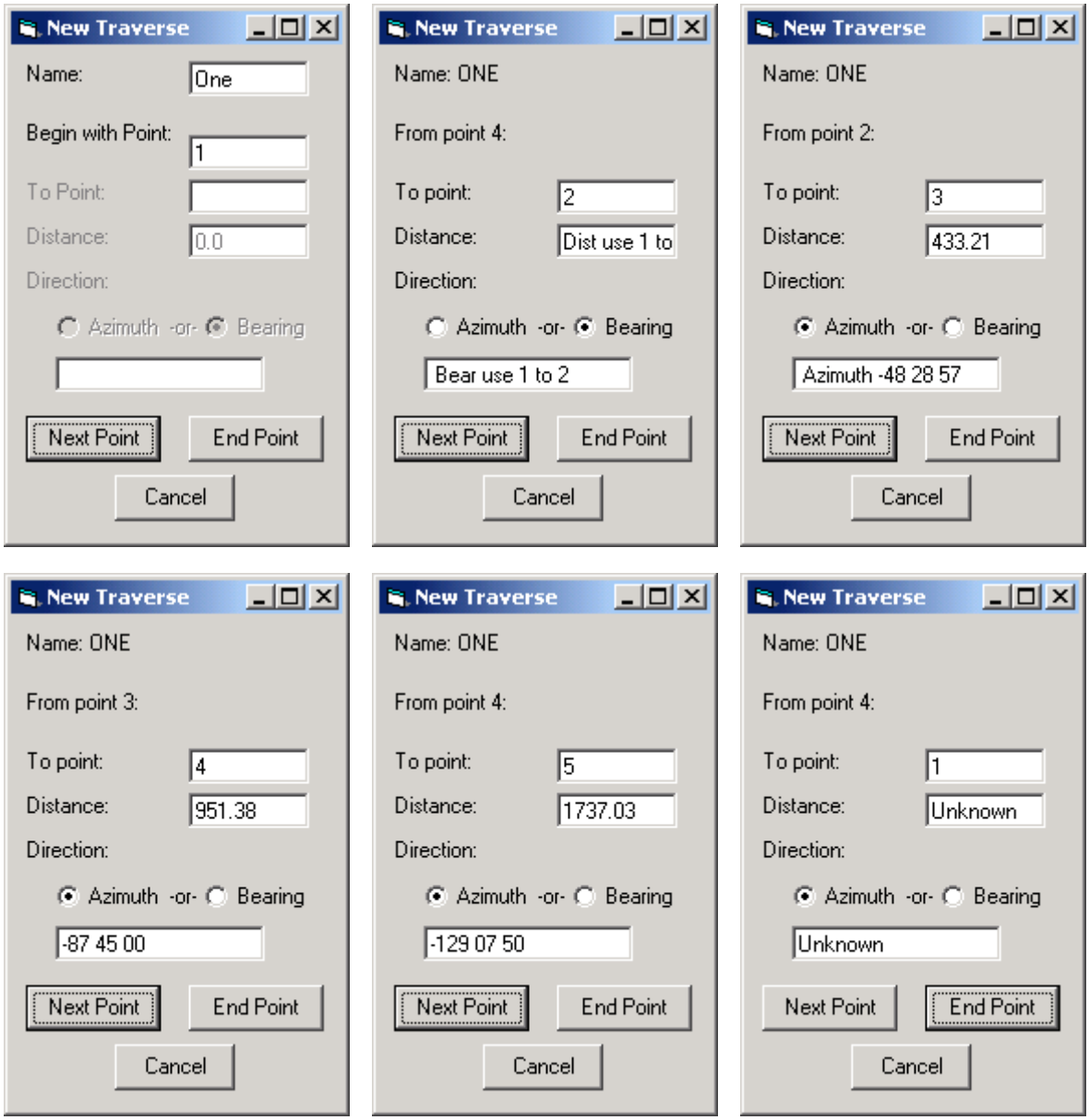
The Store Traverse function is only available in a COGO command.

The Store Traverse command relates two or more existing points. The required information for this command is a traverse name, and the points that define the courses in the traverse. The points may be in the form of a list or block of points. The Store Traverse command does not accept curve points as input.

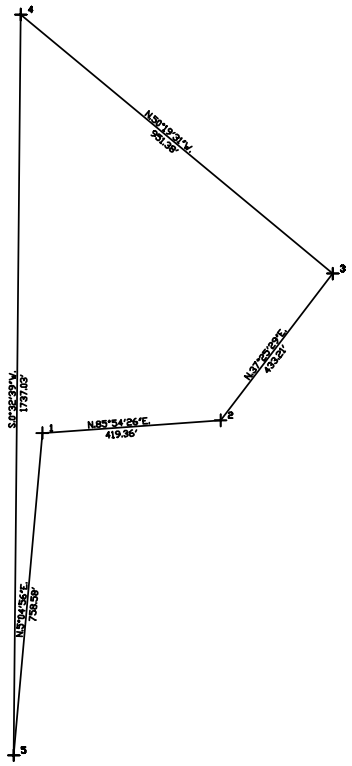
```
STORE TRAVERSE *name* (list of points/block of points)
STORE TRAVERSE *CONTROL* 1001 1003 1005 1007 to 1009 1001
```
(Traverse CONTROL is a closed traverse starting and ending at point 1001.)

Creating a Traverse Using Distance / Direction Pairs

The Traverse function in COGO defines the courses of a traverse with distance / direction pairs. The required information is the traverse name, and the distance / direction information for each course. One important option in the Traverse function is that the command word "Unknown" may be used for distance or direction information and COGO will calculate the missing information for you. Only two unknowns are allowed per traverse. The following is an example of the Traverse function.



In this case, the first course of the traverse is defined by the distance and direction between the points 1 and 2; therefore the Distance Use and Bearing Use options define the 1st course. The second course is 433.21 feet at an azimuth of –48 28 57 from point 2. The third course is 951.38 feet at an azimuth of –87 45 00 from point 3. The fourth course is 1737.03 feet at an azimuth of –129 07 50 from point 4. The last course's distance and bearing are unknown and close the traverse at point 1. The following figure is the results.

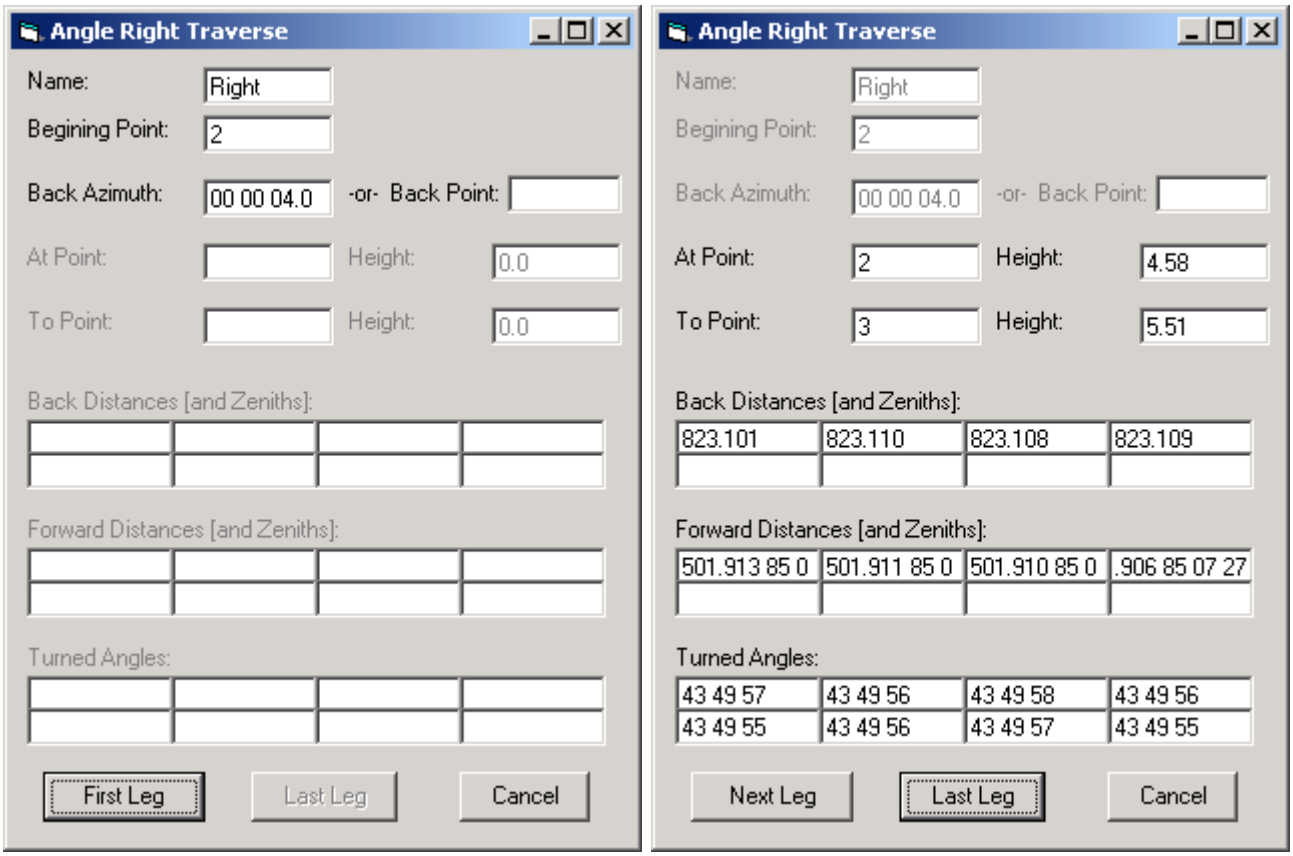


! Refer to the COGO Command Reference for all the options the Traverse function offers.

```
TRAVERSE *name* [REDEFINE] [PRINT]
BEGIN AT [POINT] n
      .
      .
      .
[FROM [POINT] m] TO [POINT] p distance direction
      .
      .
      .
END AT (POINT) k [PRINT]
```

Creating a Traverse Using the Angle Right Survey

The Angle Right Survey function is used to create a traverse from field data that has been collected using the angle right survey technique. A number of shots are taken during the data collection, so there may be a lot of information to enter into this function. The data is then statistically meaned to increase the accuracy of the data. The following is an example of an angle right survey from the CDOT Survey manual, pages 3-05-11 and 3-05-12.



In this case, the traverse begins at point 2. The back azimuth at point 2 is 00 00 04.0. The next point in the traverse is point three. The height of the instrument at point 2 is 4.58, and the height at point 3 is 5.51. There are four back and forward distances. The forward distances also have zenith angles (85 0…) that are typed immediately following the distances (501.9…). Due to the limited size of the data entry fields, only the first 3 numbers of the zenith angle can be seen. When the complete number cannot be displayed in the field the ⇦ (left arrow) and ⇨ (right arrow) on the keyboard are used to move through the number. In the 4th forward distance you can see the last part of the zenith angle. The turned angles from point 2 to point 3 are entered last.

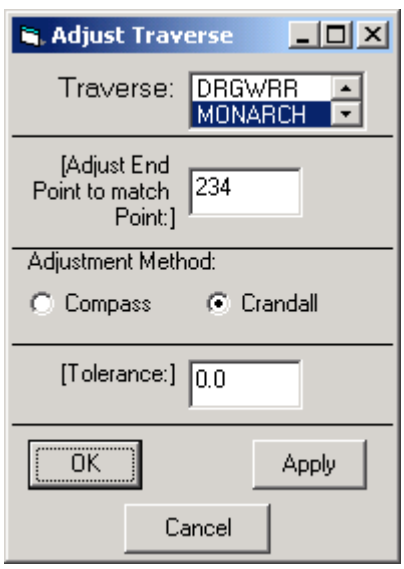
! Refer to the COGO Command Reference for all the options the Angle Right Traverse function offers.

```
TRAVERSE *name* [REDEFINE] [PRINT]
    BEGIN WITH [POINT] b  BA (azimuth/bearing/[PNT] a)
    .
    .
```

```
.
AT [PNT] m [height] TO [PNT] n [height] [*PID*]
  BACK distances(s) [zenith(s)]
  FORWARD distance(s) [zenith(s)]
  Angle angle(s)
.
.
.
END
```

Adjusting a Traverse

Occasionally traverses need to be corrected due to errors that creep in. COGO provides the ADJUST TRAVERSE function for this purpose. The information required for this function is the traverse name, the type of adjustment method to be used, and whether or not the user wishes to reset the tolerance used in the adjustment. The adjustment function can adjust a traverse that has up to 100 legs. If the error is greater than 0.0001, COGO will adjust the traverse by either using Crandall's Rule to adjust the distances, or the COMPASS method to adjust the latitude and departure. If you wish to force an adjustment no matter what the error is, set the tolerance to 0. Please refer to the COGO Command Reference for further information on the syntax of the Adjust Traverse command. The following is an example.



In this example traverse MONARCH will be adjusted using Crandall's Rule. The beginning point of the traverse will be adjusted to match point 234, and the adjustment will be forced since the tolerance is 0.0.

Chapter 7

Creating Alignments

This chapter covers storing, creating, and re-stationing alignments.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

This chapter covers the Store Alignment, Alignment, and Restation Alignment functions.

The Alignment Menu

The alignment menu creates a new alignment, generates alignment reports, lists the alignments in the model, and deletes alignments.

Storing an Alignment

The Store Alignment function is only available in a COGO command.

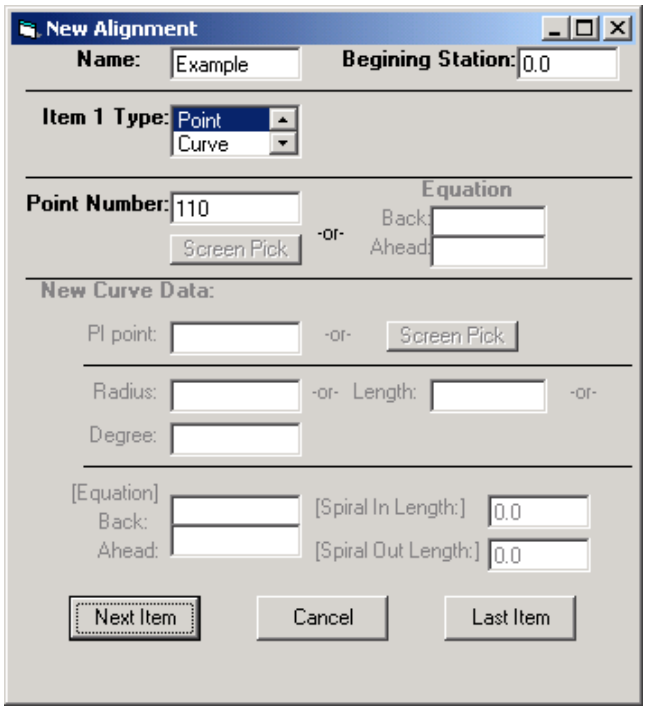
The Store Alignment command creates an alignment using points and curves. The required information for this command is an alignment name, and the points and curves that define the alignment. The points and curves may be in the form of a list or block of points/curves. The stationing of the alignment starts at 0 and cannot be set in the Store Alignment command.

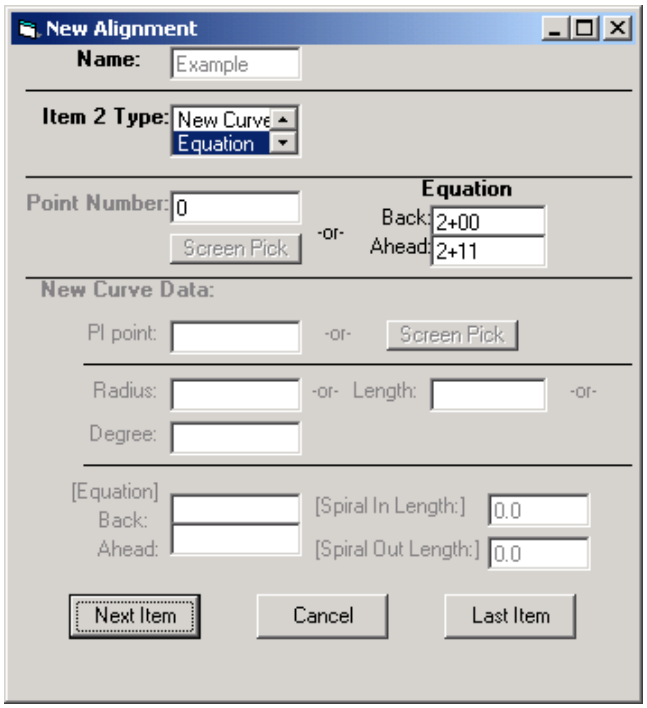
```
STORE ALIGNMENT *name* list
STORE ALIGNMENT *CENTER* POINT 100 CURVE 10 CURVE 11 POINT 1001
```

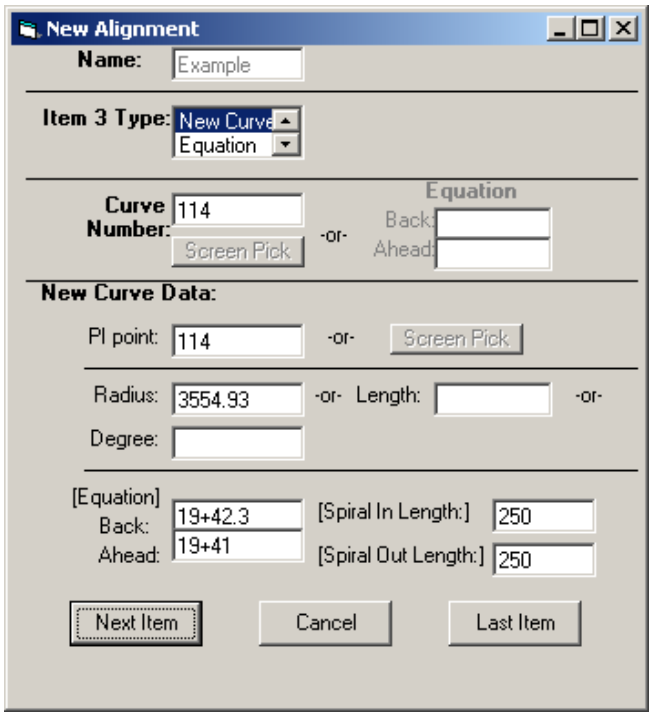
(Alignment Center starts with point 100, then has curve 10, curve 11, and ends at point 1001. The stationing starts at 0+00.0)

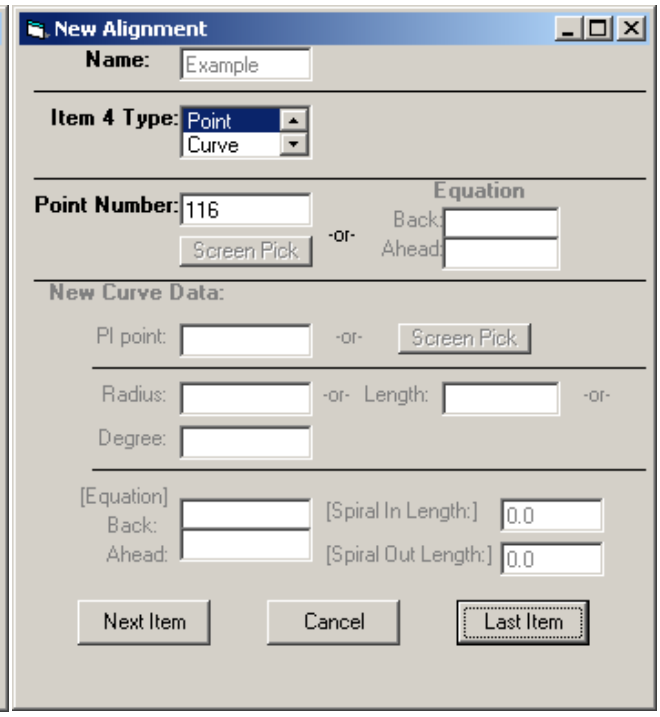
Creating an Alignment

The Alignment function defines the relationship between points, tangent lines, and curves. The resulting alignment is stationed and may have one equation per item in the alignment. The alignment function requires a name, beginning station, item type, and the item number. If the item type is a new curve, curve data is also required. If the item type is an equation, then station equation information is required. The item type determines what fields are active on the form. For example, the new curve data part of the form is active only when the item type is a new curve. The following is an example of a simple alignment.

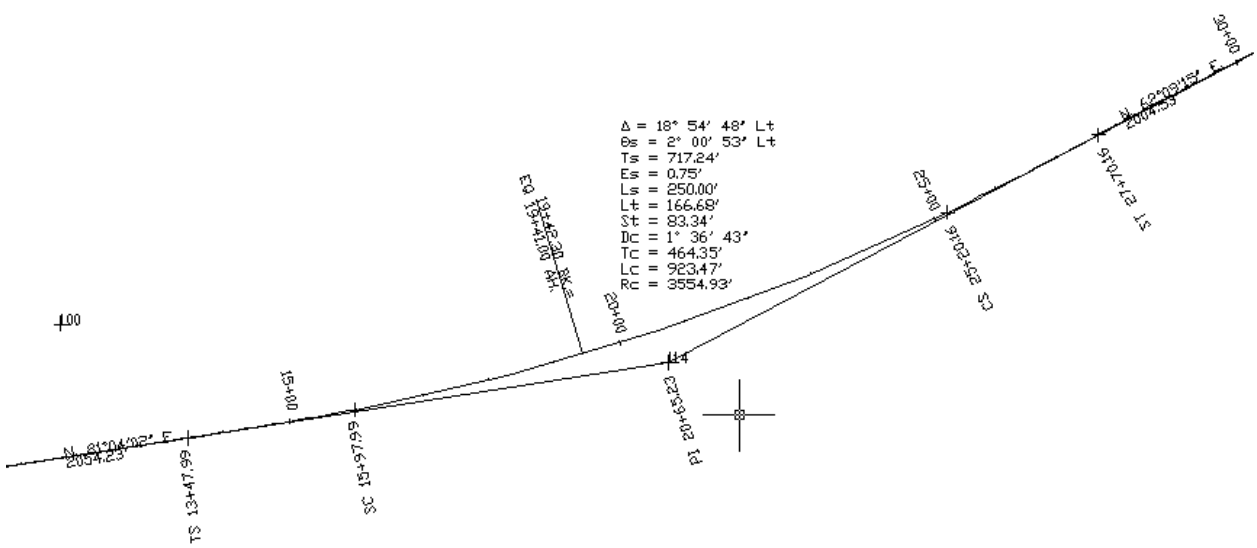








In this example, the alignment starts with point 110, has a tangent line equation, goes to complex curve 114 with an equation, then ends at point 116. The following figure is the results.

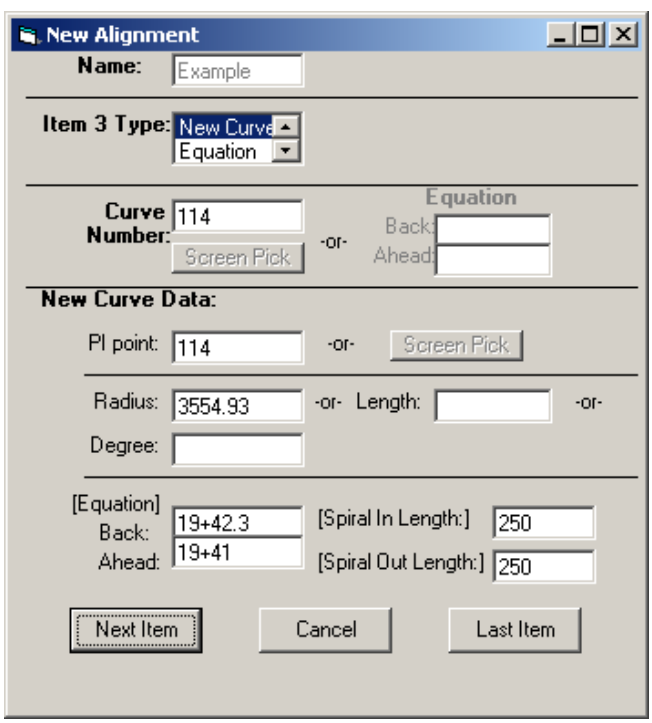


```
ALIGNMENT *name* [ROUND TO (MIN/SEC)] [REDEFINE] [PRINT]
BEGIN [AT] item 1 [STATION sta]
.
.
.
alignment item(s)
.
.
.
END AT item n [STATION sta] (PRINT)

e.g.  ALIGNMENT *Example* REDEFINE PRINT
      BEGIN AT 110 STA 0.0
      EQ 2+00 2+11
      CURVE 114 PI 114 RADIUS 3554.93 SL 250 EQ 19+42.3 19+41
      END AT 116
```

Creating Curves as Part of an Alignment

The easiest method to create curves is to create curves as part of an alignment because much of the information needed to create a curve is contained in the definition of the alignment. Instead of locating a back point on the tangent line or the PC/TS of the curve, this information is in the definition for the alignment.



In this example, new curve 114 is created as part of the alignment Example. Instead of having to create the curve points or enter more detailed information about the curve, the only required information is the PI point, the radius, and the spiral lengths.

Chapter

8

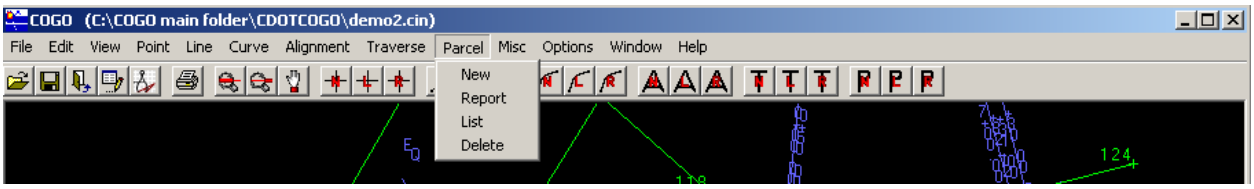
Creating Parcels

This chapter covers creating parcels.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

This chapter covers how to define parcels using the Parcel function. The main items in a parcel are courses and arcs. The parcel may be open or closed, and if the parcel is closed it has an area The Parcel function is used to describe a parcel or lot of land. Legal descriptions can be produced (see the chapter on Listing, Reporting, and Deleting Items).

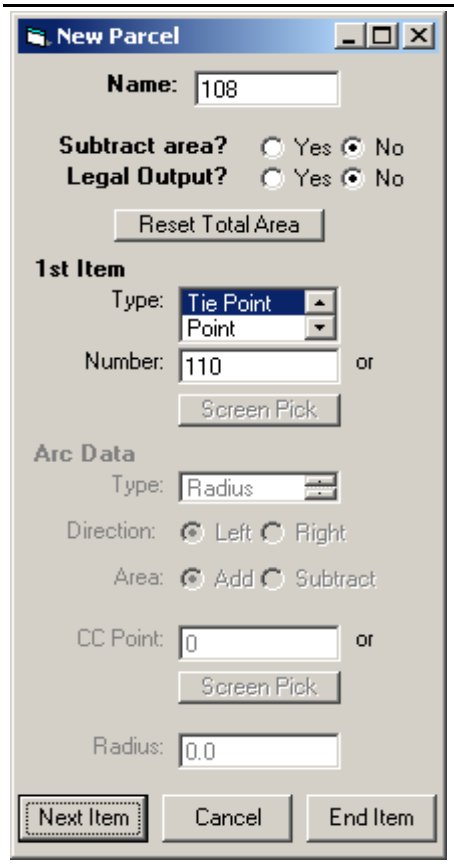
The Parcel Menu

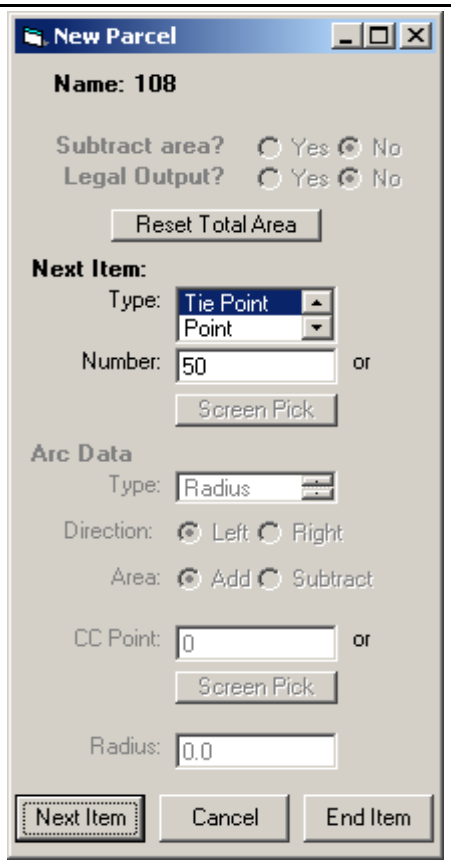


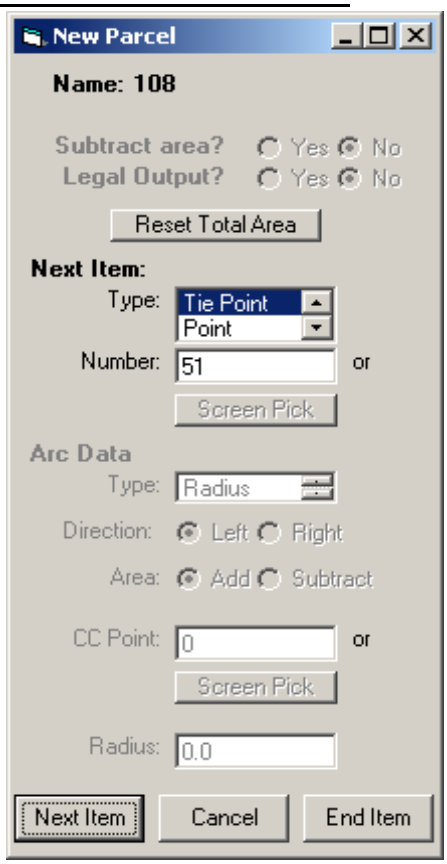
The parcel menu creates a new parcel, generates parcel reports, lists the parcels in the model, and deletes parcels.

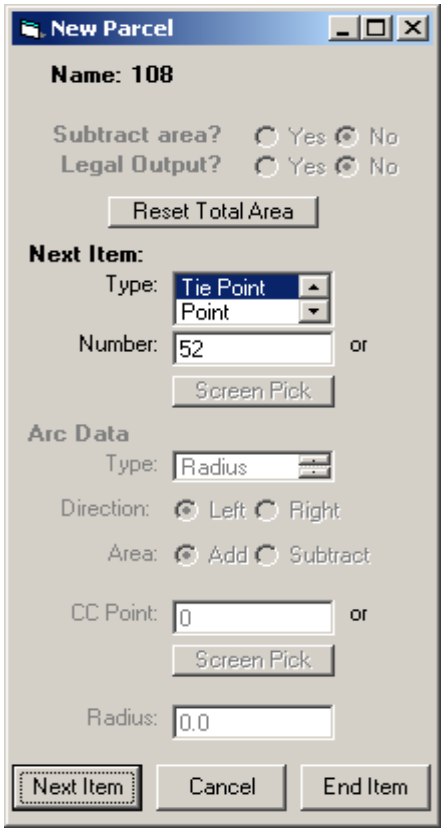
Creating a Parcel

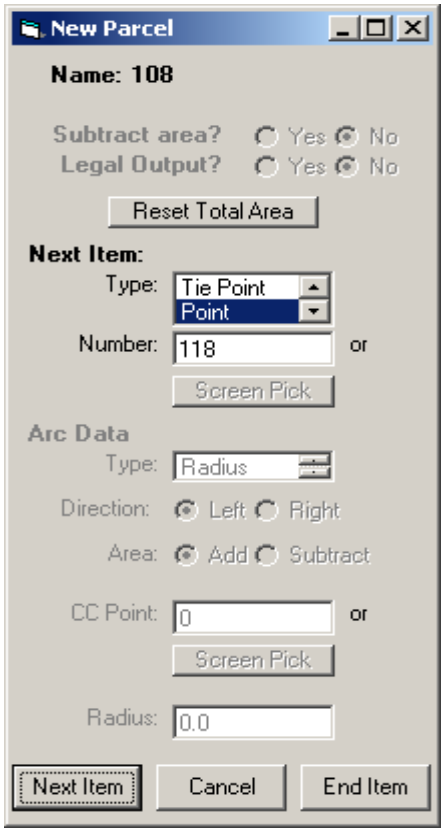
The Parcel function uses existing points to establish the relationships between the tie points leading to the parcel, the courses of the parcel, and the arcs that define a parcel of land. The parcel function requires a name, item type, and the item number. If the item type is an arc, arc data is also required. The item type determines what fields are active on the form. For example, the arc data part of the form is active only when the item type is an arc. The following is an example.

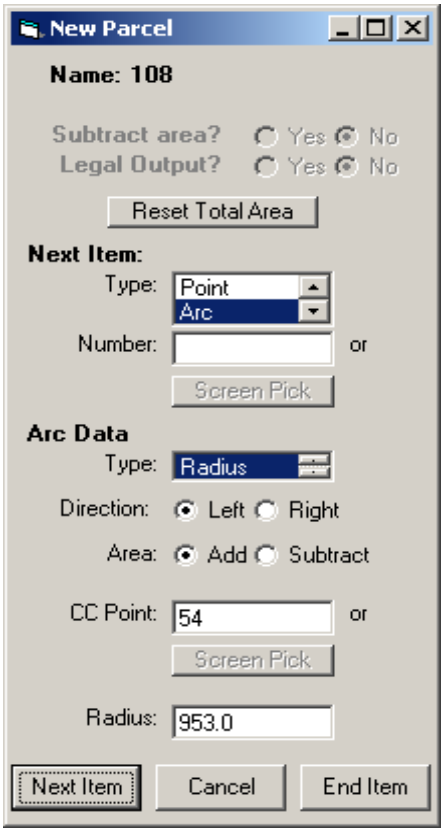

New Parcel
Name: 108
Subtract area? Yes No
Legal Output? Yes No
Reset Total Area
1st Item
Type: Tie Point
Point
Number: 110 or
Screen Pick
Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: 0 or
Screen Pick
Radius: 0.0
Next Item
Cancel
End Item

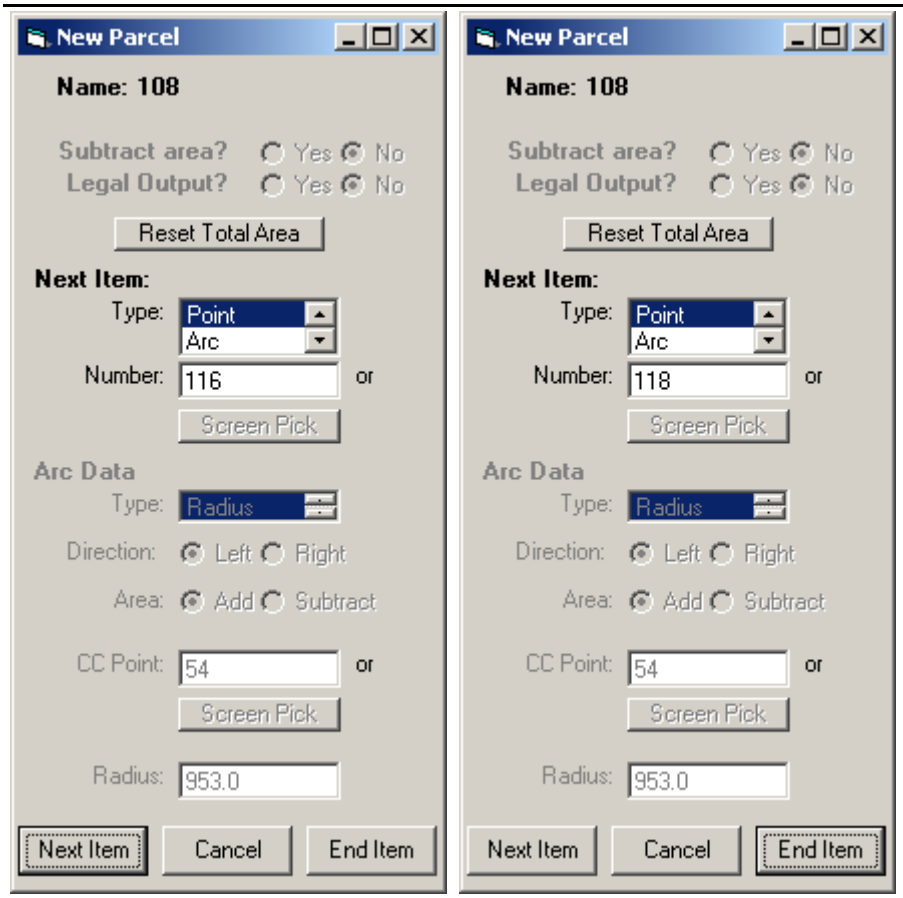

New Parcel
Name: 108
Subtract area? Yes No
Legal Output? Yes No
Reset Total Area
Next Item:
Type: Tie Point
Point
Number: 50 or
Screen Pick
Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: 0 or
Screen Pick
Radius: 0.0
Next Item
Cancel
End Item


New Parcel
Name: 108
Subtract area? Yes No
Legal Output? Yes No
Reset Total Area
Next Item:
Type: Tie Point
Point
Number: 51 or
Screen Pick
Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: 0 or
Screen Pick
Radius: 0.0
Next Item
Cancel
End Item


New Parcel
Name: 108
Subtract area? Yes No
Legal Output? Yes No
Reset Total Area
Next Item:
Type: Tie Point
Point
Number: 52 or
Screen Pick
Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: 0 or
Screen Pick
Radius: 0.0
Next Item
Cancel
End Item


New Parcel
Name: 108
Subtract area? Yes No
Legal Output? Yes No
Reset Total Area
Next Item:
Type: Tie Point
Point
Number: 118 or
Screen Pick
Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: 0 or
Screen Pick
Radius: 0.0
Next Item
Cancel
End Item


New Parcel
Name: 108
Subtract area? Yes No
Legal Output? Yes No
Reset Total Area
Next Item:
Type: Point
Arc
Number: or
Screen Pick
Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: 54 or
Screen Pick
Radius: 953.0
Next Item
Cancel
End Item



In this example, the parcel has tie that start with point 110 then goes to points 50, 51, and 52. The parcel begins at point 118 which is the start of an arc left with a radius of 953.0 and ends at point 116. From the end of the arc, the next course closes the parcel by ending at point 118.

! If one course in a parcel intersects with another the calculated area will be incorrect.

```
PARCEL *name* [SUBTRACT] [(PRINT/LEGAL)] [REDEFINE]
[TIE (POINT] m]
    .
    .
    .
BEGIN AT [POINT] n
    .
    .
    .
[POINT] q
    .
    .
    .
ARC   (LEFT/RIGHT) RADIUS r (PLUS/MINUS) CC [POINT] t
    .
    .
    .
ARC   [POINT] p (LEFT/RIGHT) (PLUS/MINUS)
    .
```

```
        .
        .
    END   AT [POINT] z

e.g.  PARCEL *Example*  REDEFINE
      TIE POINT 110
      TIE POINT 50
      TIE POINT 51
      TIE POINT 52
      BEGIN AT POINT 118
      ARC left RADIUS 953 PLUS CC 54
      POINT 116
      END AT POINT 52
```

Chapter 9

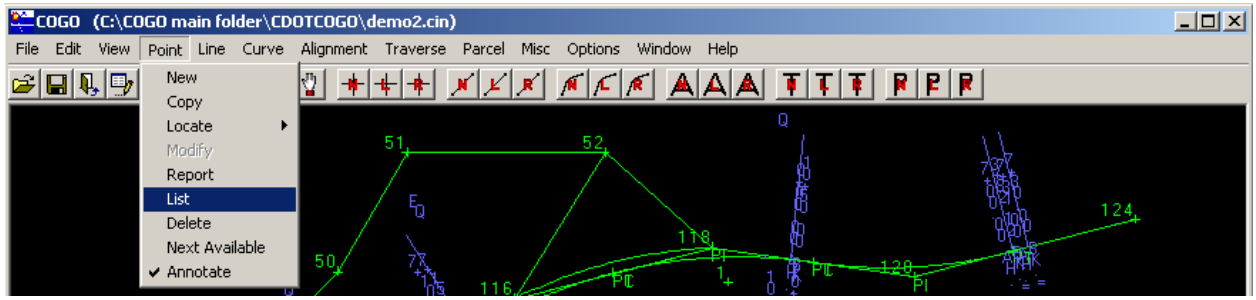
Listing, Reporting, and Deleting Items

This chapter covers how to determine what items exist in the geometry model, how to obtain reports about the items in the model, and how to remove items from the model.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

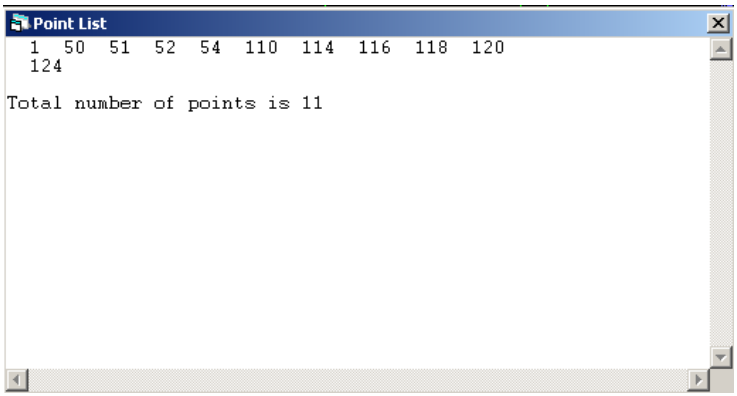
This chapter covers lists of items in the model, reports on the geometry of those items, and how to remove specified items from the model.

The Point, Line, Curve, Alignment, Traverse, and Parcel menus all have selections for listing items in the model, generating reports about those items, and deleting items from the model. The following is the point menu.



Listing Items in the Model

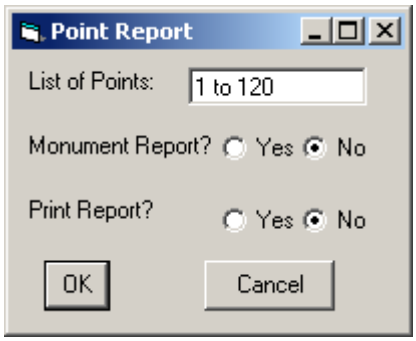
There is no required information for listing the items in a COGO model. Once you have clicked on the List menu item for the type of item desired, the list is displayed. The following is an example of a point list.



LIST item(s)

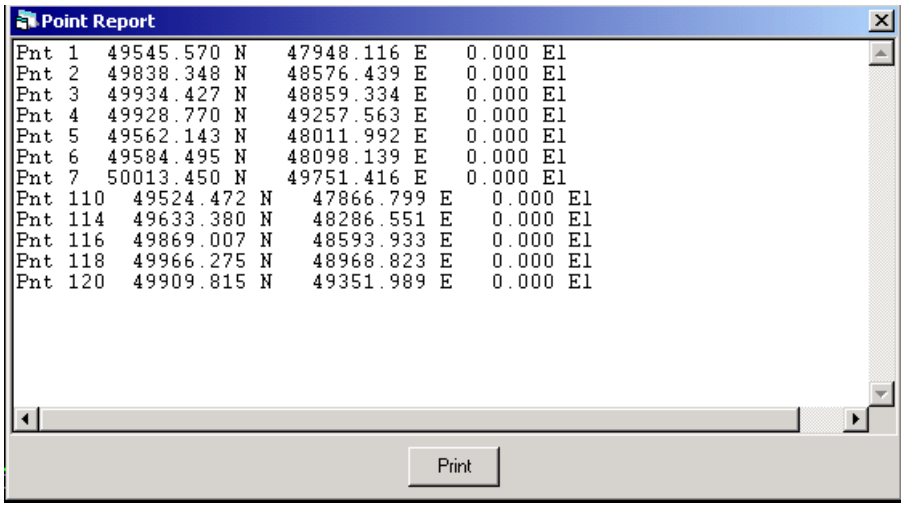
Item Reports

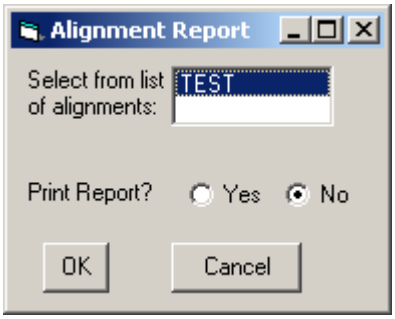
The reports are a description of the geometry of the item. The type of report generated depends on the type of item. The report function also allows you to save the report to a file or print out the report. The following are examples of reports.



In this example a report for points 1 to 120 was requested. The list of points may be individual point numbers or a range of points as in this example. Any undefined points in the range of points will not be reported.

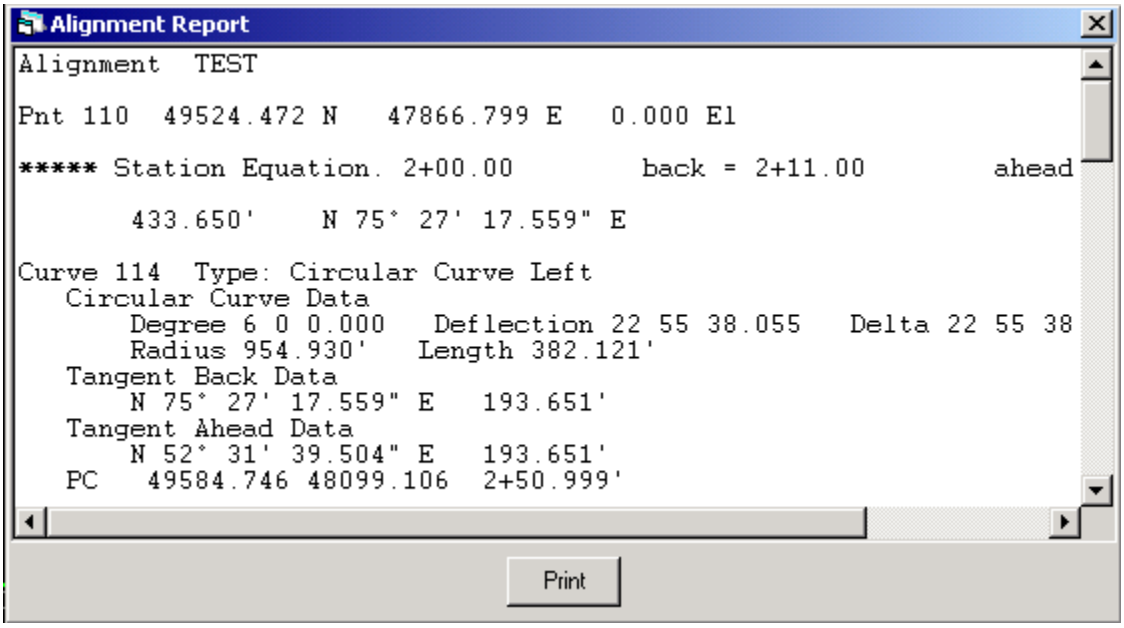
Two options are available. The monument report omits the "Pnt" in front of the point number, and formats the information differently for inclusion in monument listings. The print report option and the print button on the report will send the report to a specified printer.





In this example, an alignment report is desired. All of the alignments in the model are listed in the select list. In this case there is only one alignment in the model – TEST.

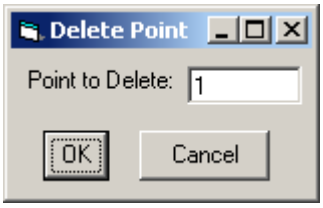
The option Print Report will send the report to a printer if Yes is selected. The print button on the report will print the report also.

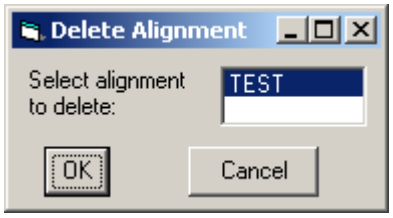


Print item(s)

Deleting Items

You remove items from the geometry model via the Delete menu entry. The item type determines what information is required to delete an item. In the case of points, lines, and curves the item number is required. In the case of alignments, parcels, and traverses the name of the item is selected from a list. The following are examples.





DELETE item(s)

Chapter 10

Other COGO Functions

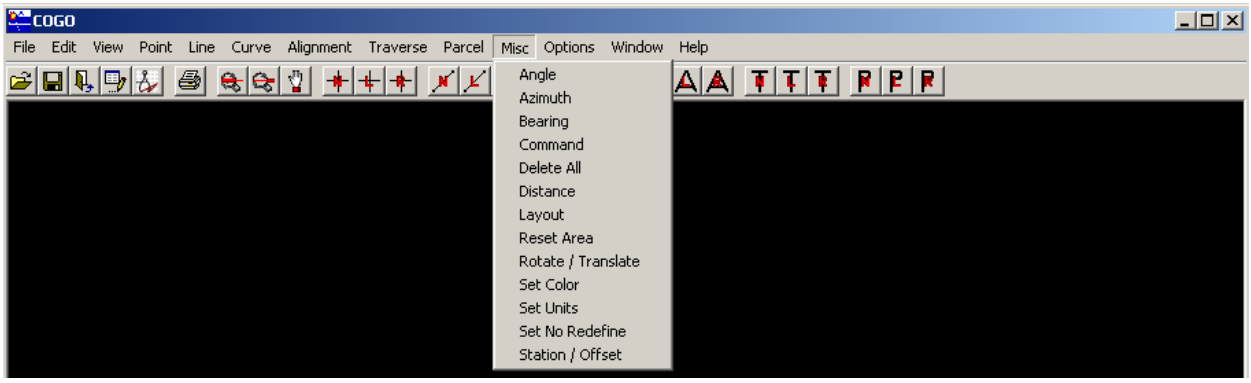
This Chapter covers other important COGO functions.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

THis chapter covers other functions in COGO.

The Misc Menu

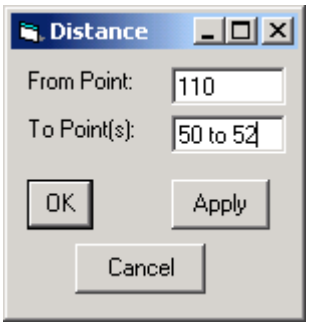
The Misc (miscellaneous) menu has a number of functions that report or modify the COGO items. The following is the Misc menu in Windows.



! The Misc menu is different in Windows and AutoCAD.

Angle, Azimuth, Bearing, and Distance

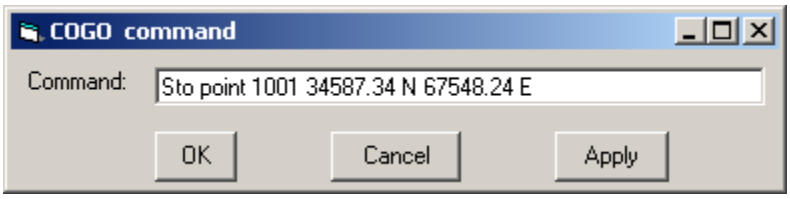
The Angle, Azimuth, Bearing, and Distance functions report information about the distance(s) and direction(s) between points. The following is an example of a distance function.



In this case, the distance and direction from point 110 to points 50, 51, and 52 will be reported. The To Points can be in the form of a single point number or a range of points.

Executing COGO Commands

The Command function provides the ability to enter a COGO command. The COGO commands have more functionality than some of the Windows forms, so there may be instances where executing a COGO command is desirable.

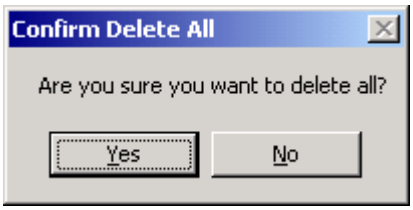


In this case, the Store Point command is used to store point 1001.

Deleting all COGO Items

! This function will delete all items in the COGO model.

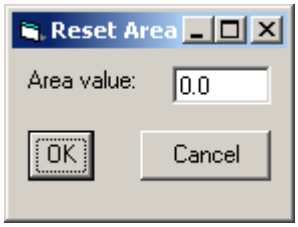
The Delete All COGO function removes all items. After execution of this function COGO will have no items in the geometry model. After clicking on the Misc | Delete All menu item, the following confirmation window appears.



If you click the Yes button, all items in the geometry model will be deleted. Clicking on the No button will cancel the Delete All function.

Resetting the Total Area Register

COGO has space set aside to keep track of areas called the Total Area Register. As Parcels are created their area is added to or subtracted from the Total Area Register. The Reset Area function sets the value of the Total Area Register.

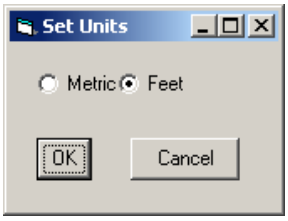


In this case, the Total Area Register will be set to 0.

Setting the Working Units

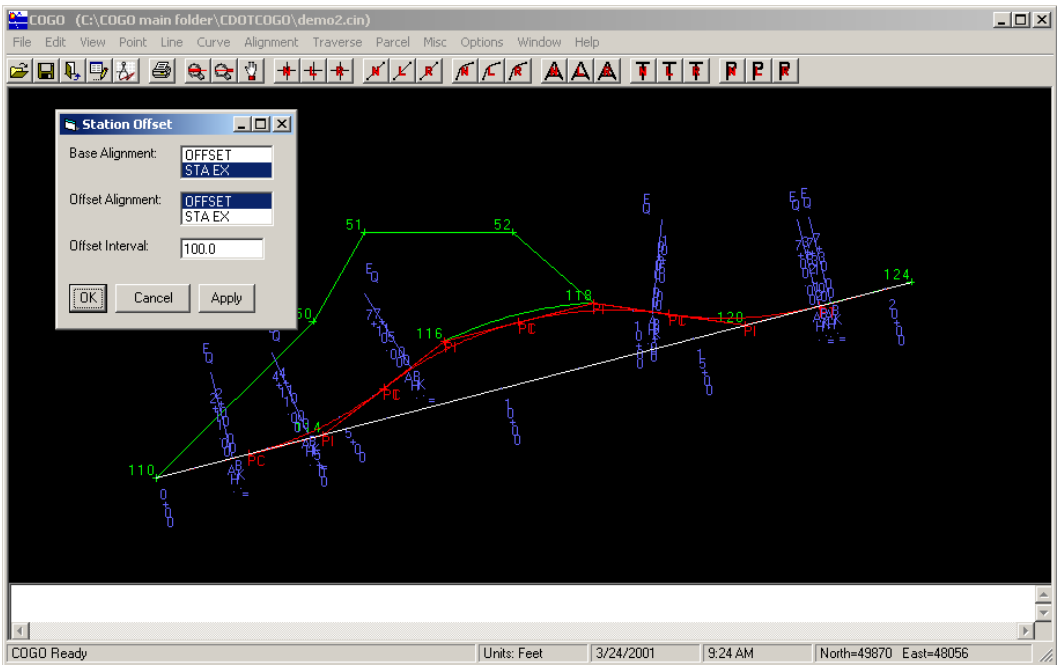
The Set Units function sets the working units to either feet or meters. The default units are set to feet. If you wish to work in feet you need not execute this function, but it is recommended.

! This should be the first function executed in each COGO run. If this function is executed more than once per COGO run and the units are changed, the results are unpredictable.



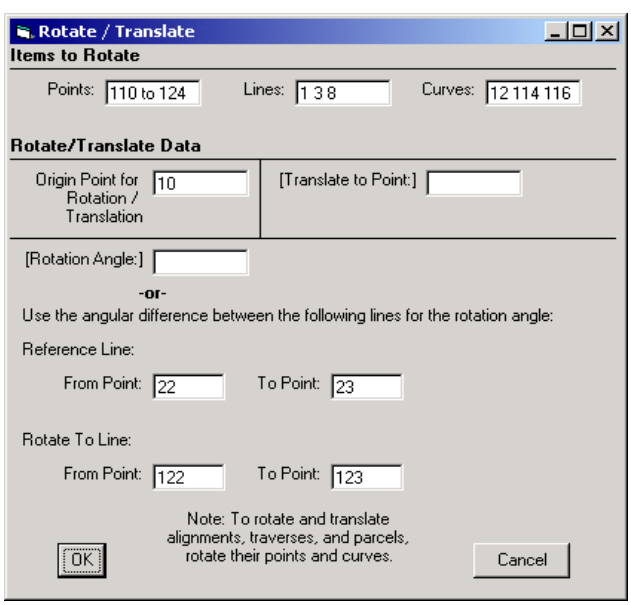
Station Offset Function

The Station Offset Alignment function produces a coordinate report by projecting points from the base alignment onto the offset alignment. The report produced also contains the distance and skew from one alignment to the other.



In this example, the base alignment is "STA EX" (shown in red) and the offset alignment is "OFFSET" (shown in white). The report will use 100' increments along alignment "STA EX" to produce the distance and skews from one alignment to the other.

Rotating and Translating COGO data in Windows and AutoCAD



This function rotates and, or translates individual COGO items. This function is available in Windows and AutoCAD. When this function is executed, the rotate / translate is documented in the COGO output (.COT) file.

In this case, the specified items will be rotated about point 10 the angular difference between the line segment 22 to 23 and the line segment 122 to 123 (clockwise 1 minute and 1.45 seconds). If a translate point was provided, the items specified will also be translated from point 10 to the translate point.

The rotation angle can be specified as an angle, or as the angular difference between two line segments.

The command for rotating / translating COGO items rotates/ translates the entire geometry model.

Chapter 11

Transferring Information to Other Engineering Applications

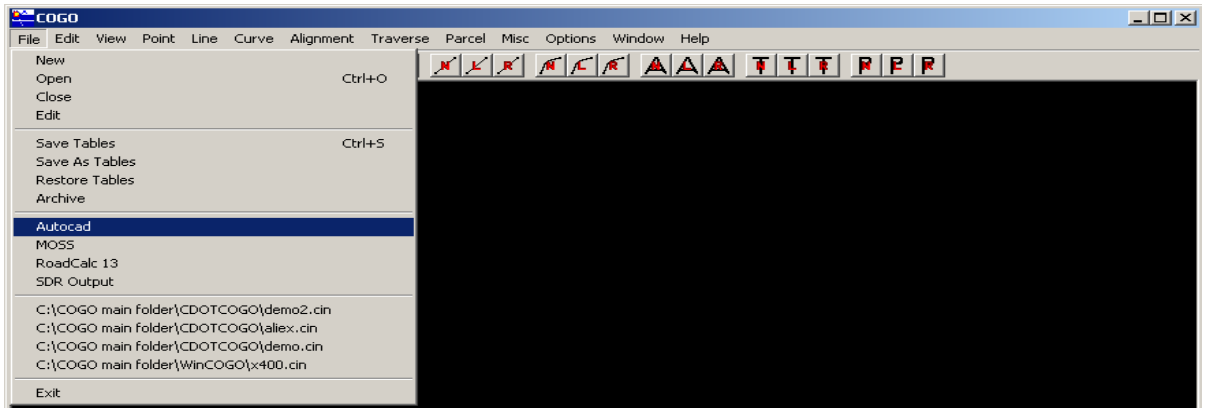
This Chapter covers the functions used to transfer information to other engineering applications and the SDR data collectors.

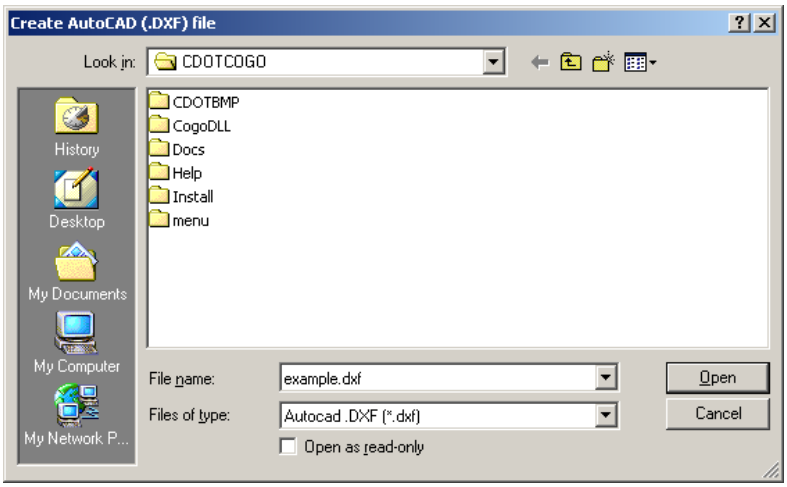
ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

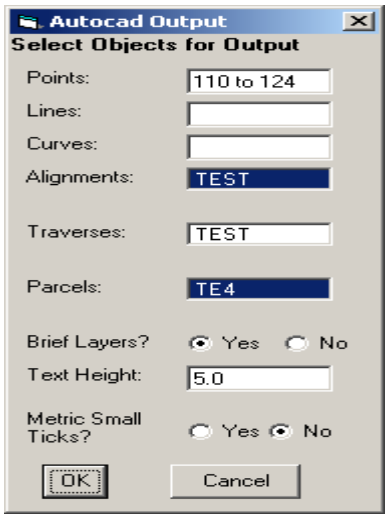
COGO can transfer information to MX (formerly MOSS), AutoCAD, SDR data collectors, PICS, and RoadCalc version 13. All of these functions are in the File menu, as shown below.

AutoCAD

The data transfer to AutoCAD can be done in two ways. The first is to use the File menu, AutoCAD selection to generate a DXF format file that can be read into AutoCAD. The second is to run COGO in AutoCAD 2000 and answer that the COGO information does not exist in AutoCAD. The following is an example if the DXF file transfer.







Once you have specified the output file name, you then select what COGO items you want output to the DXF file, and whether you want brief layer names or long layer names, the text size of COGO annotation in AutoCAD, and if 20m stationing ticks are desired if the model is metric.

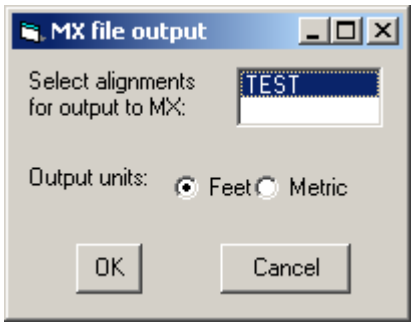
Layer Names in AutoCAD

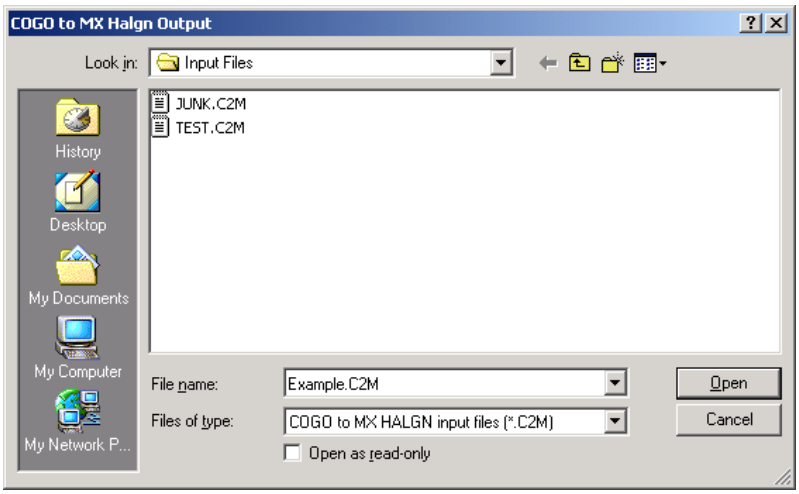
Three different layer names are possible in AutoCAD. There are two layer naming schemes in the DXF file transfer, and one layer name scheme if you are using COGO in AutoCAD. The schemes for layer naming are in the following table.

	Points	**Lines**	**Curves**	**Traverses**	**Alignments**	**Parcels**
DXF Brief	CPnt	Clin	Ccur	CTname & CTTXTname	CAname & CATXTname	CPname & CPTXTname
DXF Long	Cnumber	CLnumber	CCnumber	Same	Same	Same
Run in AutoCAD	CPnt & CPntTXT	CLine	CCurve	CTname & CTnameTXT	CAname & CAnameTXT	CPname & CPnameTXT

MX

The transfer of alignments to MX is done via the MX Halign file. The Halign file is then imported into MX as a horizontal alignment string. Refer to the COGO command reference for more information on the MX Halign file format. The following is an example.

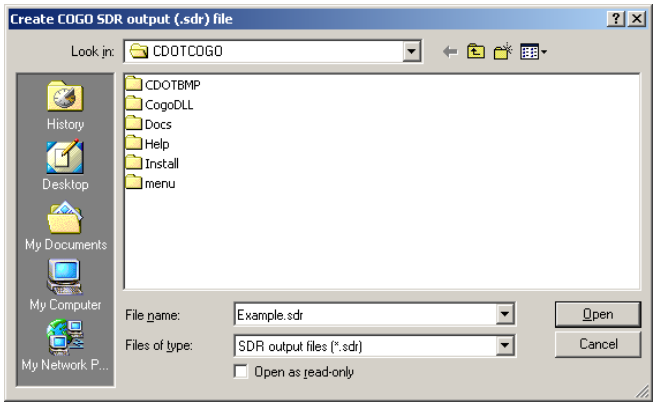




In this example, the alignment TEST is selected for output to the file Example.C2M in standard (feet) units.

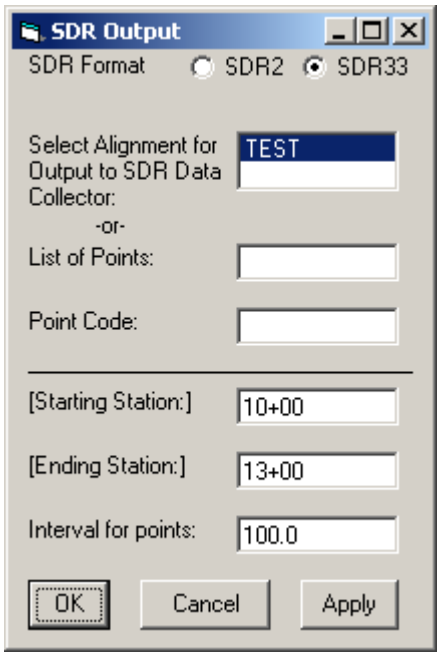
SDR Data Collectors and PICS

The SDR File Output function creates a coordinate file that can be downloaded to the Leitz SDR series 2, 20, and 30 data collectors and can be transferred to PICS.



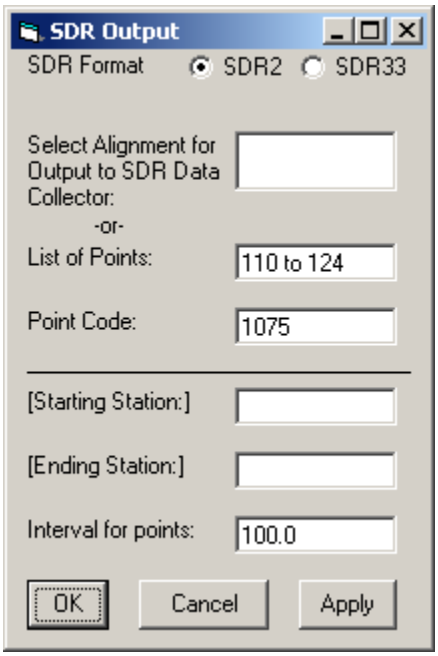
In this case COGO creates the SDR33 format file Example.sdr with the data from alignment TEST starting at station 10+00 to station 13+00 at 100' intervals.

The files created are fname.SDR and fname.COT. The SDR file contains the coordinates of the points along the specified alignment starting at the 'from' station (default = beginning station), ending at the 'to'



station (default = end station), at even intervals (default = 100'). The coordinates of the curve points are automatically output to the SDR data collector file. The COT file is a report of the contents of the SDR file.

SET SDR FILE OUTPUT *fname* ALI *aname* [FROM sta1] [TO sta2] [EVEN n]

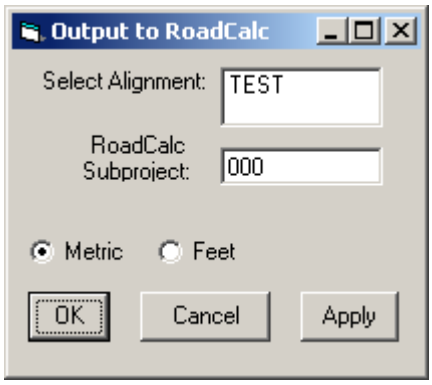


Entering only points, this example creates a SDR2 format file using the TMOSS point code of 1075. The list of points may be individual points, or in the form of a range of points (e.g. PNT 1 TO 10 PNT 15 PNT 20). The recommended TMOSS codes are 1086 and 1075 for the control points, and 1080 for property control points.

SET SDR FILE OUTPUT *name* *code* list of points

RoadCalc Version 13

The Output to RoadCalc function creates the subproject files in Roadcalc 13's internal format for alignments. It converts alignments from COGO internal format to Roadcalc internal format. All files necessary to transfer the survey or design alignment data to the Roadcalc SUBPROJECT nnn are created in the current directory. These files MUST reside in the directory specified for the RoadCalc subproject. The files created for the design alignment are named RCDAGnnn.DAT, RCDAGnnn.TCD, and if the alignment has station equations RCDEQnan.DAT.



ROADCALC FILE OUTPUT SUBPROJECT nnn ALIGNMENT *name* [DESIGN] [(METRIC/FEET)]

! COGO can create curves with overlaps. Roadcalc cannot handle curve overlaps. All curve overlaps in COGO must be resolved before transferring alignments to Roadcalc.

Chapter

12

COGO Input, Output, Log, Saved Tables, and Archive Files

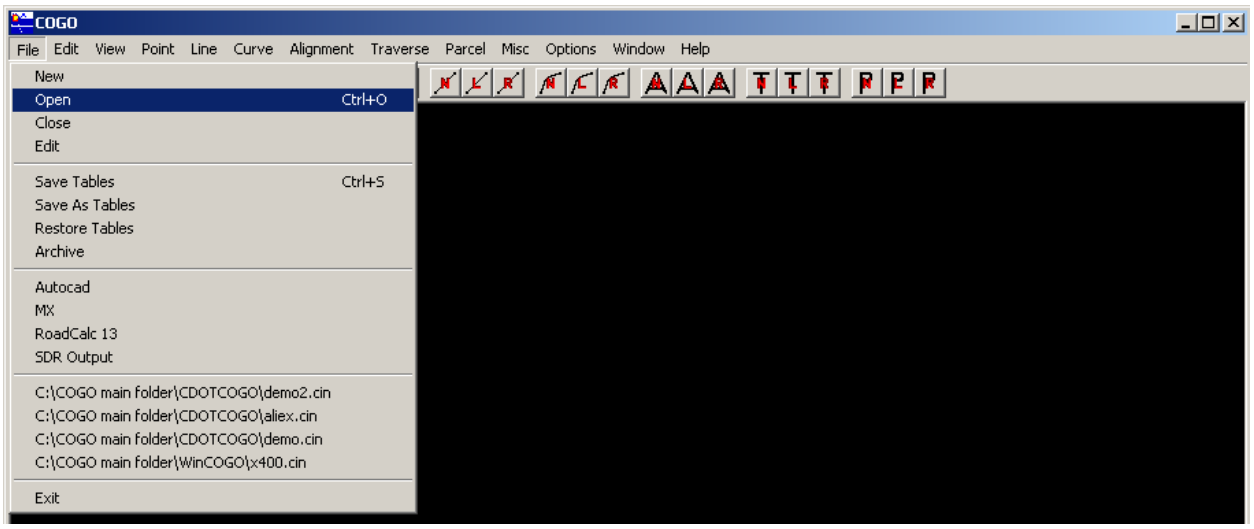
This chapter covers the COGO functions that use or create files. This chapter is essential reading.

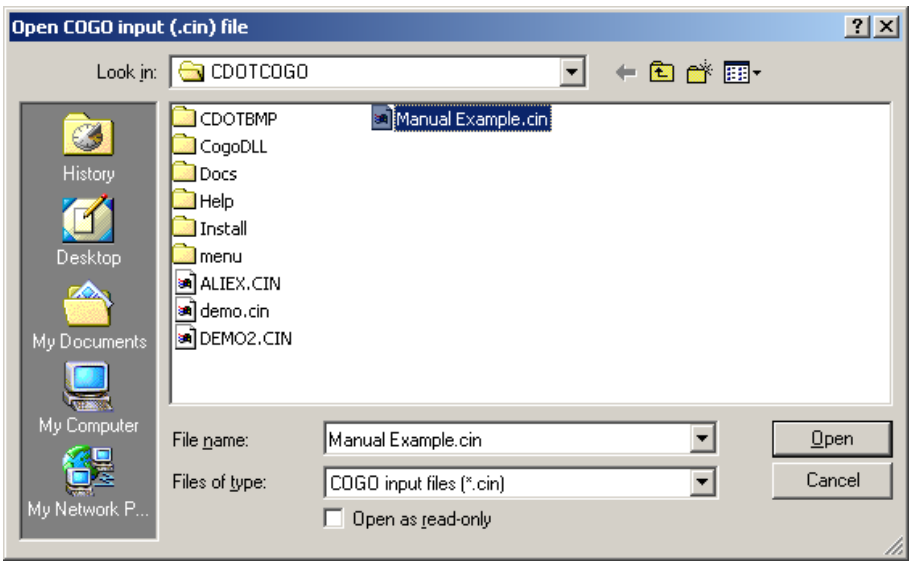
ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

This chapter covers the functions that input COGO commands, output COGO reports, logs functions executed during a COGO run, and archive the COGO model for future use.

COGO Input from a File

The primary method of creating a geometry model is the COGO input file (.CIN). The .CIN file is a collection of COGO commands that create the geometry model. The COGO commands may be entered into a .CIN file using a text editor such as PFE32 (Programmer's File Editor, 32 bit), then you can instruct COGO to read from that file. The functions that redirect input are the File | Input or the Set File Input command. COGO expects the file extension to be .CIN. The file name must conform to the file naming conventions for the operating system. In the Set File Input command, the file name is a string of characters and therefore must be surrounded by asterisks. The following are examples.





SET FILE INPUT *147REST*
(Instructs COGO to read input from file 147REST.CIN in the current directory.)

The COGO Log File

The COGO log file records all functions that you executed during a COGO run. COGO does this by recording most of the functions as COGO commands. The functions that are not recorded as commands are those functions that exist only in Windows or those that only exist in AutoCAD. The log file can be used in a number of ways. It can be used as a disaster recovery technique, or can be used to capture commands that may be used later as an input file. The log file may be edited and, if desired, added to existing input files.

COGO Output to a File

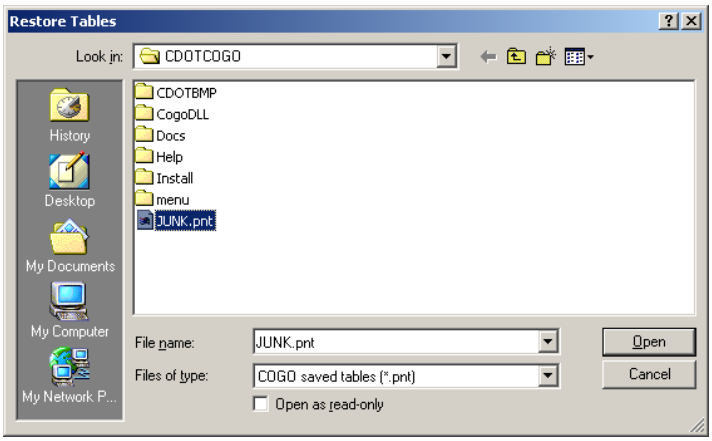
COGO automatically creates an output file named COGO.COT for each run. This file is over-written each time COGO is run. You may want to re-direct the COGO output to a file of your choosing. The Set File Output command is provided for this purpose. The Set File Output command redirects the output of the COGO.COT file to a file you specify. The file name must conform to the file naming conventions for the operating system. The file name is a string of characters and therefore must be surrounded by asterisks in the Set File Output command. The file name extension will be .COT. If desired, a "page break" can be introduced into the output file with the Eject command. The result of the Eject command is that the information following the Eject command will start on a new page. The following are examples.

SET FILE OUTPUT *89042A1*
(Redirects the COGO.LOG output to file 89042A1.COT in the default folder.)

SET FILE OUTPUT *C:\JUNK1*
(Redirects the output to file C:\JUNK1.COT.)

COGO Saved Tables

Saved Tables are files that contain the geometry model. The primary uses for saved tables are for saving the model for future use as either a recovery tool, or when a model becomes so large that re-running an input file is time consuming. The saved tables have file extensions of .PNT, .LIN, .CUR, .ALI, and .ATR. The two COGO functions that handle the saved tables are the Save and Restore functions in the File menu. The following are examples.



In the above examples, the geometry model is saved under the file name JUNK. The result is five files whose file names are JUNK (JUNK.PNT, JUNK.LIN, JUNK.CUR, JUNK.ALI, and JUNK.ATR). The model is Restored by specifying the JUNK.PNT as the file to restore. COGO then strips off the extension of the file and uses the file name to Restore all five files.

Archiving COGO Data for Future Use

The COGO ARCHIVE function gives the user the capability of creating a high accuracy input file which can be used years later to recreate the model for a particular project. The required information is an identification line that must immediately follow the command. The resulting file, named ARCHIVE.COT, has the COGO Store commands necessary to recreate the model. The northings, eastings, and distances in the archive file are accurate to 6 decimal places. The following is an example.

```
ARCHIVE
I-70, I-25 interchange.
```

(The file ARCHIVE.COT will be created with store commands that will recreate the COGO model. The identification line "I-70, I-25 interchange." will appear at the top of the file as a COGO comment line.)

! It is advisable to rename the ARCHIVE.COT file before executing the Archive function again.

Chapter

13

COGO in Windows Functions

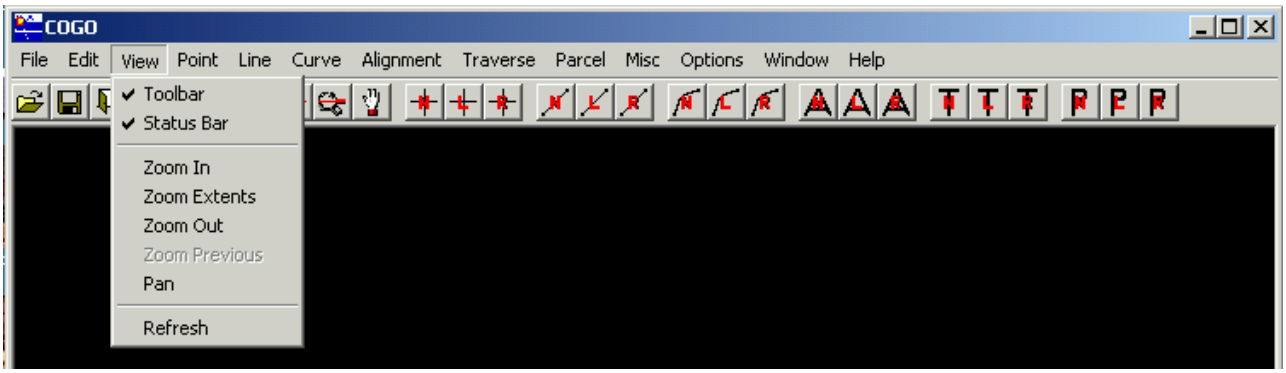
This chapter covers the COGO functions that exist only in Windows.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

COGO has some functions that exist only in Windows. This chapter covers those functions.

Most of the functions that exist only in Windows are graphical functions such as zooming and panning and are in the View menu.

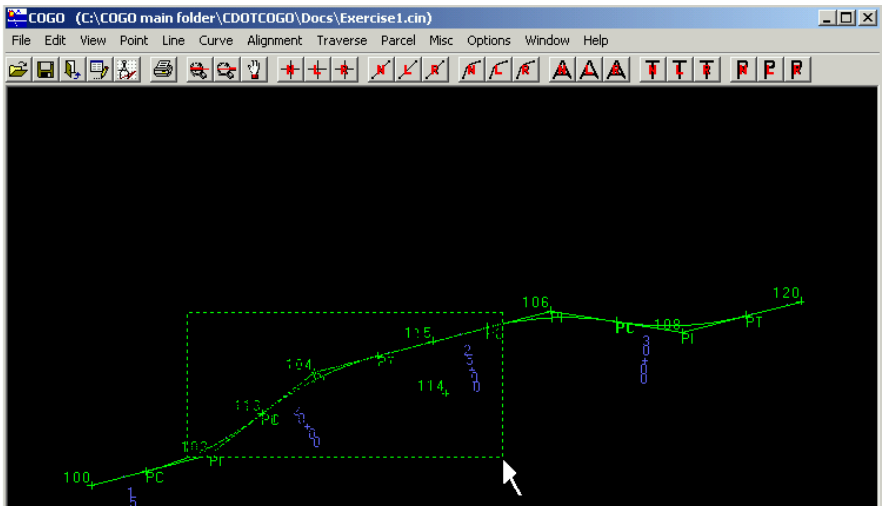
The View Menu



The View menu contains the functions that manipulate the display area in Windows.

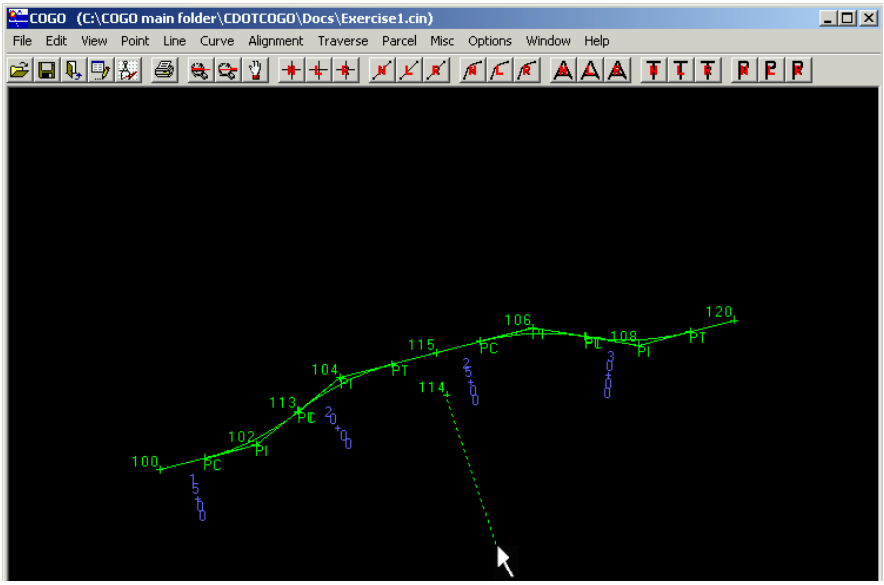
Zooming

There are four different zoom functions in the View menu. You can zoom in, zoom out, zoom to the extents of the COGO model, and return to the previous zoom. The only one of these that requires further input is the Zoom In function. To zoom in, click on the Zoom In menu entry, then click and drag from one corner of the zoom area to the other.



Panning

Panning allows you to re-center the display area. To pan, click on the Pan menu entry then click on the part of the drawing you want to move and drag the cursor to the area you want to move to.



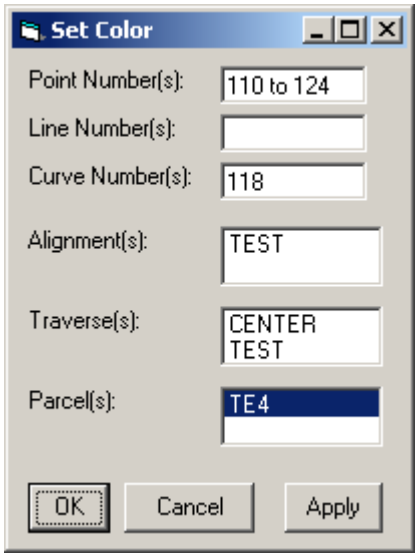
Refreshing the Display Area

Sometimes you need to redisplay or refresh the COGO display due to clutter or changes in the display. The Refresh function in the View menu re-generates the COGO display.

Setting Item Colors

The Set Color function changes the color of COGO items in Windows. The Set Color function is in the View menu.

Once the color of an item is changed, the color is active until the next Refresh of the graphic display in Windows. This function is a two-step process. You indicate which items you want to change the color of, and then indicate what color is to be used.



In this case points 110 to 124, curve 118, and parcel TE4 have been selected for the color change. After the OK button is clicked, the Color form is displayed. In this case, red is the chosen color. After the OK button is clicked, the display refreshes and the color of the items selected will be red.

Chapter 14

COGO in AutoCAD Functions

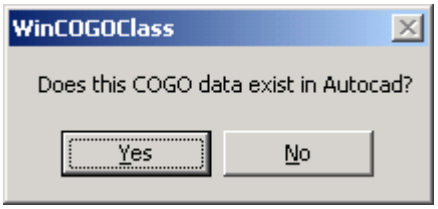
This chapter covers the COGO functions that exist only in AutoCAD.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

COGO has a number of functions that exist only in AutoCAD, such as selecting points from the graphic display area in AutoCAD. This chapter covers those functions.

Displaying COGO Data in AutoCAD

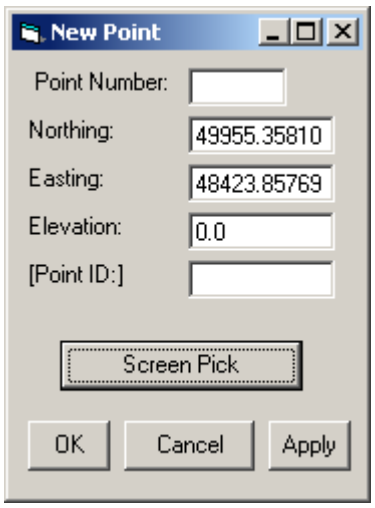
Reading in a COGO input file while executing COGO in AutoCAD causes COGO to create AutoCAD objects for all of the items on the geometry model. After that, only new geometry items are added to the AutoCAD drawing as you create new items in COGO. Since this may result in duplicate AutoCAD objects, COGO prompts you to determine if the COGO items already exist in AutoCAD when you read in a COGO input file. If you already have this COGO information in AutoCAD you should respond that the data does exist in AutoCAD.



It is your responsibility to properly maintain the COGO items in AutoCAD. Some tools are provided by COGO to assist you in maintaining COGO data in AutoCAD. See the section on graphically moving points for more information.

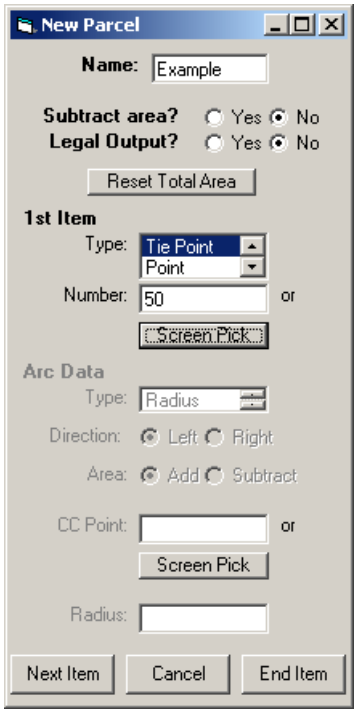
Graphically Selecting COGO Points

A number of the Windows forms have a "Screen Pick" button. This button is only active while COGO is executing in AutoCAD. The purpose of this button is to allow you to get information from the AutoCAD drawing for use in COGO. An example is the New Point form. If you choose to use the Screen Pick button, the coordinates for the point you pick in AutoCAD are captured and the Northing and Easting fields will be filled in for you. The following is an example.



All you need to do now is to fill in the Point Number, and optionally, the elevation and point ID, then click on the OK or Apply button.

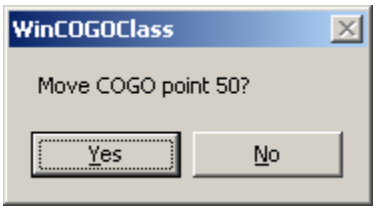
In other cases, the Screen Pick button captures the COGO point number rather than coordinate information. The following is an example.



In this case, the Screen Pick button for the item number was clicked. The New Parcel form was minimized, the user selected COGO point 50 in AutoCAD, and the point number was automatically entered in the Number field.

Graphically Moving COGO Points

When COGO is loaded in AutoCAD and you choose to move a COGO point in the drawing, COGO detects that you are moving a COGO point and, if you choose to, will update the geometry for that point to reflect the changes you made in the drawing. The following is an example.



In this case point 50 was moved in AutoCAD. COGO detected this change and asked if the point should be moved. If Yes is selected, the coordinates in the geometry model will be updated. If No is selected, the point will be moved back to its original location in AutoCAD. In this way, the data in the geometry model and the drawing match.

If only the text for point 50 was moved in AutoCAD, a warning message is displayed as follows.

Chapter 15

COGO Command Reference

This chapter covers all the COGO commands in detail.

ICON KEY	
	COGO Command
	Windows Information
A	AutoCAD Information
!	Important Note

COGO commands are used to build input files and archive files. Use this chapter as a reference to aid in understanding COGO commands and their syntax.

The commands are arranged in alphabetical order.

Adjust Traverse Command

COGO depends on surveying for some information. During the survey process it is possible that slight errors can creep in. Therefore, COGO contains the ability to adjust traverses to correct these errors. The ADJUST Command provides this capability. The command has two forms. Basically one handles closed traverses and one handles open traverses.

ADJUST TRAVERSE *name* [TO [POINT] n] (LS/COMPASS) [PRINT] [TOL t]

This command will compute and report the error of closure, and the difference between the northings and eastings of the start and end points for a stored traverse. If the error is greater than 0.0001, COGO will ADJUST the TRAVERSE by either using the Crandall Rule (LS) to adjust the distances, or the COMPASS method to adjust the latitude and departure. If used, the end point of the traverse is adjusted TO match POINT N. The PRINT option will report the results of the adjustment. The preset tolerance for adjustment is 1 foot to 10000. Ordinarily if the error is smaller than this, the traverse will not be adjusted. The TOL t option resets the adjustment tolerance. A tolerance of 0.0 will force the traverse to be adjusted.

! The traverse may not have more than 99 courses to be adjusted. The command assumes that the last point differs from

the first point by the error. After the adjustment the last point and the first point will have the same coordinates. The LS adjustment adjusts the distances and not angles.

Example:

```
ADJUST TRAV *16* LS PRINT
DISTANCE/BEARING ERROR IS 242.156/S32.25. 44.4 E (-204.394, 129.857) 1 FT.  IN 21.958 FT.
TRAVERSE 16 ADJUSTED

TRAVERSE 16
PNT      1    716134.744       2126198.996
           86.644       S 0. 19. 32.0 E
  PNT      50   716048.101       2126199.488
           64.701       N 88. 20. 21.0 E
  PNT      51   716049.976       2126264.163
         237.742       N 88. 20. 21.0 E
  PNT      52   716056.867       2126501.804
           92.592       S 55. 54. 51.0 E
  PNT      53   716020.561       2126586.981
         149.270       S 55. 8. 32.0 E
  PNT      54   715935.247       2126709.469
         193.874       S 69. 42. 43.0 E
  PNT      55   715868.023       2126891.315
         282.147       S 84. 33. 48.0 E
END TRAVERSE          16
```

Alignment Command

An alignment is made up of points, curves, and station equations. The ALIGNMENT command computes and stores the alignment and the items that makes up the alignment. The alignment is stationed and equations can be specified.

The alignment command is a multiple line command with the alignment name, the beginning of the alignment, each item in the alignment and the end of the alignment. Please note that blank lines between the begin and end statements will generate an error.

```
ALIGNMENT *name* [ROUND TO (MIN/SEC)] [REDEFINE] [PRINT]
BEGIN [AT] item 1 [STATION sta]
       .
       .
       .
alignment item(s)
       .
       .
       .
END AT item n [STATION sta] (PRINT)
```

The ALIGNMENT referred to by *NAME* (8 characters maximum), BEGINs at ITEM 1 with STATION sta followed by a list of ITEMS and ENDs AT ITEM n. If desired, you may force the ROUNDing of bearings TO the nearest MINute or SECond. This will cause points and PI's to be shifted slightly as necessary. The REDEFINE option instructs COGO to delete any alignment with the same name before creating the new alignment. The PRINT option will print the geometry of the alignment. The ALIGNMENT command accepts points, curves, and equations as items. The following are the accepted forms for specifying points, curves, and equations as part of the alignment.

POINTS - A point may be defined in the following ways.

`[POINT] m [EQUATION aa.aa]`
Where: M is the number of a previously defined point. aa.aa is the ahead station for the point.

`[POINT] m NORTH n EAST *id* [EQUATION aa.aa]`
See the STORE POINT command for definitions.

CURVES - A curve may be added to the alignment in the following ways.

`CURVE c [HOLD]`
Where: C is the number of a previously defined curve.

HOLD prevents COGO from rounding the bearings of the elements of the curve. If the rounding option is specified, HOLD must be used.

`CURVE c, pi reference item, circular item, [spiral item], [Equation], [HOLD]`

Where: C is the curve number used for storing the curve.

PI REFERENCE ITEM (select one)

`PI n EQ s.s [HOLD]`
Where: N is a previously defined point.
S.S is a station equation.
HOLD prevents rounding adjustments.

`PI PNT as c [EQ s.s] [HOLD]`
Where: PNT AA C is a curve point (TS/PC, SC, CS, ST/PT, CC) for curve C. e.g., PI PNT PC 3.
S.S. is a station equation.
HOLD prevents rounding adjustments.

PI NORTH n EAST e EO s.sl [HOLD]
Where: N and E are the coordinates of the PI.
S.S is a station equation.
HOLD prevents rounding adjustments.

CIRCULAR ITEM (select one)

RADIUS r
Where: R is the radius of the curve.

DEGREE ddd mm ss.ss
Where: DDD MM SS.SS is the degree of curvature.

DELTA d
Where: D is the circular curve delta. Can only be used if the curve is non-spiraled.

LENGTH 1
Where: L is the length of the curve. Can be used in association with DELTA or delta will be computed from next PI or POT.

TANGENT t
Where: T is the length of the tangent from TS/PC to PI or PI to ST/PT. This element can be used in association with DELTA or delta will be computed from next PI or POT.

CC [AT] n
Where: N is the point number for the center of the curve. This point may be in the form of a point number of **a** defined point or previously defined curve point or north and east coordinates.

[SPIRAL ITEMS] (optional, select one or two of the following)

SLB dist [HOLD]
Where: DIST is the length of the back spiral.
See above for explanation of HOLD.

SLA dist [HOLD]
Where: DIST is the length of the ahead spiral.
See above for explanation of HOLD.

SL dist [HOLD]
Where: DIST is the spiral length of the back and ahead spiral.
See above for explanation of HOLD.

SAB ddd mm ss.ss [HOLD]
Where: DDD MM SS.SS is the back spiral angle.
See above for explanation of HOLD.

AA ddd mm ss.ss [HOLD]
Where: DDD MM SS.SS is the ahead spiral angle.
See above for explanation of HOLD.

SA ddd mm ss.ss [HOLD]
Where: DDD MM SS.SS is the spiral angle for both back and ahead spirals.
See above for explanation of HOLD.

[EQUATION] (optional)

EQ bb.bb aa.aa

Where: bb.bb is the back station on the curve or a curve point. Valid curve points are PC, TS, CS,SC, ST, PT.
aa.aa is the ahead station

EQUATION – An equation on a tangent line

EQ bb.bb aa.aa
Where: bb.bb is the back station
aa.aa is the ahead station.

! The station specified for the first item in the BEG AT line of the alignment command determines the stationing for the alignment. If the station is not provided in the BEG AT line, COGO will use 0+00 as the beginning station for the alignment.

Examples:

```
ALI *LEFT CL* REDEFINE PRINT
BEGIN AT 10 STA 470+00
CURVE 30 PI 30 DEG 5 SLA 350  EQ 485+12.51 487+10.00
CURVE 70 PI 70 RADIUS 2083.64 SLB 250 SLA 300
CURVE 100 PI 100 DEG I RADIUS 5730.00
CURVE 140 PI 140 SL 300 RADIUS 1146.00
CURVE 180 PI 180 RADIUS 1230.00 SLB 300
EQ 1127+22.72 1127+14.00
CURVE 210 PI 210 SL 300 RADIUS 1146.00
CURVE 240 PI 240 RADIUS 5814.00
CURVE 270 PI 270 RADIUS 1637.14 SL 200
CURVE 300 PI 300 RADIUS 5814.00
CURVE 330 PI 330 RADIUS 2949.00 SL 150
PNT 340
CURVE 360 PI 360 RADIUS 1432.50 SL 250 EQ PT 1562+92.2
CURVE 390 PI 390 RADIUS 11544.00
```

```
CURVE 420 PI 420 RADIUS 5814.00
CURVE 450 PI 450 SL 250 RADIUS 1432.5
CURVE 480 PI 480 SL 300 RADIUS 1146.00
END AT PNT 500

ALIGNMENT *DETOUR*
BEGIN AT 110 STA 311+15.99
CURVE 114 PI 114 RADIUS 954.930
CURVE 116 PI 116 RADIUS 954.930
CURVE 118 PI 118 RADIUS 954.930
CURVE 120 PI 120 RADIUS 954.930
END AT 124

ALIGNMENT *TEST* REDEFINE
      BEGIN AT 110 STA 0.0
      EQ 2+00 2+11
      CURVE 114 PI 114 RADIUS 954.930 EQ 4+10 4+11
      CURVE 116 PI 116 RADIUS 954.930 EQ 7+15 7+10
      CURVE 118 PI 118 RADIUS 954.930 EQ 10+50.0 0+00
      CURVE 120 PI 120 RADIUS 954.930 EQ PT 7+00.
      EQ 7+30.00 7+31.00
END AT 124
```

Angle Command

This command calculates the angle formed by three points.

ANGLE ([PLUS]/MINUS) AT [POINT] n FROM [POINT] m TO [POINT] p

This command determines the angle formed by three points. It calculates the angle FROM POINT M through the vertex AT POINT N, TO POINT P. COGO computes the angle in a clockwise direction (PLUS), unless MINUS is specified.

Examples:

ANGLE MINUS AT 10 FROM 3 TO 4

ANGLE AT 21 FROM 10 TO 15

Archive Command

The Archive Command creates an archive file. This file is an input file in the form of COGO store commands. This file is used for archiving a COGO model and transferring data from one computer to another. This command may also be used to convert a COGO model from feet to metric or visa versa.

```
ARCHIVE TABLES [(METRIC/FEET)]
identification line
```

The Archive Command is a two-line command that generates a file of COGO commands (ASCII file called ARCHIVE.COT) that will recreate the COGO model using STORE commands. The tables are unchanged by the command. The METRIC/FEET option allows for conversion from one unit of measurement to another. The 'identification line' **must** be on a separate line and immediately follow the ARCHIVE command. This identification line will be the first line of the file in the form of a comment.

Example:

```
ARCHIVE TABLES
Identification Line
```

Area Command

The area command calculates the area of a boundary defined by lines and arcs.

```
[SUBTRACT] AREA item list [PRINT]
```

The area command calculates an area. It SUBTRACTs from or adds to (default) the total area register the AREA bounded by an ITEM LIST of lines and curves. If the PRINT option is used, the points and the distance/angle from point to point will be reported along with the area. A maximum of 350 line/curve segments may be used to define the boundary. The first point in the list **must** be the last point in the list. The line segments are defined in COGO as [point] n [point] m. There are two ways to define a circular segment. The first is the point arc point syntax:

```
[POINT] a ARC [POINT] m (PLUS/MINUS) [POINT] p
```

In this syntax the ARC goes from POINT n to POINT p with the center at POINT m. The PLUS or MINUS indicate whether the resulting area is to be added or subtracted from the parcel area.

The second is the point arc radius syntax:

```
[POINT] n ARC (LEFT/RIGHT) RADIUS r (PLUS/MINUS) [POINT] p
```

In this syntax the ARC goes from POINT n to POINT p with the direction of curvature indicated by LEFT or RIGHT and RADIUS r. The PLUS or MINUS indicate whether the resulting area is to be added or subtracted from the parcel area.

! The calculated area is inaccurate if any side of the parcel should cross another side. If the Area command must be continued on more than one line, you should be careful dividing the description of the circular segment. The division of the input can occur only before the word ARC or after the word PLUS or

MINUS.

Example:

```
AREA 670 671 ARC RIGHT RADIUS 4358.0 MINUS 672 673 674 + 675 ARC LEFT
RADIUS 4558.0 PLUS 676 677 670 PRINT
```

Azimuth Command

The Azimuth Command calculates the azimuth and distance between two points, or between a point and a group of points.

AZIMUTH ([POINT] n [POINT] m / [POINT] n [POINT] m TO [POINT] p)

This command calculates the AZIMUTH and distance between points. The first type of azimuth command calculates the distance and azimuth between two points (POINT n and POINT m). The second type calculates the distance and azimuth between a point (POINT n) and a range of points (POINT m TO POINT p). Any undefined points in the range are ignored.

Examples:

```
AZIMUTH 15 PNT 100 TO 120
AZIMUTH 41 52
```

Bearing Command

The Bearing Command calculates the bearing and distance between two points, or between a point and a group of points.

BEARING [POINT] n [POINT] m / [POINT] n [POINT] m TO [POINT]

This command calculates the BEARING and distance between points. The first type of bearing command calculates the distance and bearing between two points (POINT n and POINT m). The second type calculates the distance and bearing between a point (POINT n) and a range of points (POINT m TO POINT p). Any undefined points in the range are ignored.

Examples:

```
BEARING 1 3
BEARING 15 30 TO 40
```

Deleting Points, Lines, Curves, Traverses, Alignments, Parcels and Text

This command is the primary command for deleting all COGO item types from the COGO model.

DELETE ([POINT]/LINE/CURVE/TRAVERSE/ALIGNMENT/PARCEL/TEXT) (list/ALL)

This command DELETEs the specified item(s) from the COGO model.

! POINTS is the only optional word when a list of numbers is used.

Examples:

```
DELETE POINT 1 TO 3
DELETE 1 5 1034
DELETE CURVE 10
DELETE ALI *Gl3GDET*
```

Deleting All Items in the COGO Model

The Delete All command deletes all the items in the COGO model.

DELETE ALL

This command clears all data from the COGO model.

Example:

```
DELETE ALL
```

Distance Command

The Distance Command calculates the distance and bearing between two points or a point and a range of points.

DISTANCE ([POINT] n [POINT] m / [POINT] n [POINT] a TO [POINT] p)

This command calculates the distance and bearing between points. The first type of distance command calculates the distance and bearing between two points (POINT n and POINT m). The second type

calculates the distance and bearing between a point (POINT n) and a range of points (POINT m TO POINT p). Any undefined points in the range are ignored.

Examples:

```
DISTANCE PNT 10 PNT 20
DISTANCE PNT 3 5 TO 15
```

Divide Line Command

The Divide Line command creates new points by dividing a line into specified lengths.

```
DIVIDE LINE 1 length [POINT] pl [DISTANCE] dl [POINT] p2 [DISTANCE] d2 etc.
```

This command creates new points by dividing up a line. It creates point (s) (P1 to Pn) in the point table by dividing line L of length LENGTH into the segments specified by the point/distance pairs. Up, to 18 point/distance pairs may be specified. The division of the line begins at the starting point for the line. The distance is calculated from the previous point specified, i.e., p2 will be d2' from p1 along line 1.

! It may be useful to execute the NO REDEFINE command before executing this command.

Example:

```
DIVIDE LINE 10 1000 PNT 11 135.1 PNT 12 200 PNT 13 551.3
```

Divide Line Segment Command

The Divide Line Segment command creates new points by dividing a line segment into equal length parts.

```
DIVIDE LINE SEGMENT [POINT] n TO [POINT] m list of points
```

This command creates new points (LIST OF POINTS) in the point table by dividing the specified line segment (POINT n TO POINT m) into X + 1 equal segments, where X is the number of points in the list. The LIST OF POINTS may be individual points; [point] P1 [point] P2 [point] P3.... or a range of points - [point] P1 TO [point] Pn.

! COGO locates POINT P1 from POINT N and then each point will be located from the previous point in the list. It may be useful to execute the NO REDEFINE command before executing this command.

Example:

```
DIVIDE LINE SEG PNT 10 TO PNT 20 PNT 30 TO 40
```

Divide Curve Command

The Divide Curve command creates new points by dividing a curve into equal length arcs.

```
DIVIDE CURVE c list of points
```

This command creates new points in the point table by dividing the CURVE C (TS/PC to ST/PT) into X + 1 equal segments where X is the number of points in the list. The curve may be simple or complex. The LIST OF POINTS may be individual points; [point] P1 [point] P2 [point] P3.... or a block of points - [point] P1 TO [point] Pn.

! It may be useful to execute the NO REDEFINE command before executing this command.

Examples:

```
DIVIDE CURVE 5 PNT 6 PNT 7 PNT 8 PNT 9 PNT 10
DIVIDE CURVE 10 PNT 20 TO PNT 31
```

Eject Command

The Eject command causes COGO to issue a form feed and print a new heading.

```
EJECT
```

This command creates a form feed and a page heading in the output.

Example:

```
EJECT
```

Finish Command

The Finish command terminates the current execution of COGO. This is an obsolete command.

FINISH

This command terminates COGO.

Example:

```
FINI
```

Layout Ties Command

The Layout Ties command calculates the distance and bearing from a point to the pertinent points on a particular COGO item.

LAYOUT TIES [point] n TO item

This command lays out COGO items for staking. It calculates the distance and bearings from POINT N to the specified ITEM. The TO item may be one of the following.

POINTS - points are referenced in the following ways

List of points

Where: The list is comprised of point numbers separated by blank spaces or commas.

POINT m TO POINT p

Where: M is the starting point number for the range of points and P is the last point number in the range. All undefined points in the range are ignored.

CURVES - curves may be referenced in the following ways

CURVE c

Where: C is the curve number of a previously defined curve. COGO reports the distance and bearing from point n to the TS/PC, SC, PI, CS and ST/PT of curve C.

TRAVERSES - traverses may be referenced in the following ways

TRAVERSE *name*

Where: Traverse NAME is **a** previously defined traverse. COGO reports the distances and bearings from point n to the points of the traverse.

ALIGNMENTS - alignments may be referenced in the following ways

ALIGNMENT *name*

Where: Alignment NAME is a previously defined alignment. COGO reports the distances and

bearings from 'point n' to each point of the alignment and the curve points of each curve on the alignment.

Examples:

```
LAYOUT TIES PNT 100 TO ALIGNMENT *SURV*
LAYOUT TIES POINT 10 TO POINT 100 TO 120
```

Layout Curve Command

The Layout Curve command is used for staking out curves. The distance and bearing from a point to the pertinent curve points is calculated and reported.

```
LAYOUT CURVE c [TRANSIT AT [POINT] m] [SIGHT on [POINT] p] [EVEN n]
```

This command allows you to layout a curve for staking. It calculates the deflection that will be turned from a SIGHT line (transit point to back sight point), the distance from the TRANSIT POINT M and bearing of the line defined by the transit point and the staking points for CURVE C. The staking points will be at each curve point (TS/PC, SC, PI, CS, ST/PT) and a point at each station increment specified by EVEN N. If even n is not specified the increment is assumed to be 100'.

Example:

```
LAYOUT CURVE 121 TRANSIT AT PNT 110 SIGHT ON 11 EVEN 150
```

Layout Alignment Command

The Layout Alignment command computes deflections, bearings and distances along an alignment.

```
LAYOUT ALIGNMENT *name* [EVEN n]
```

This command allows you to layout an alignment for staking. It calculates the deflection turned from the sight line (TS/PC to PI), the distance from the transit point (TS/PC) and the bearing of the line defined by the TS/PC and the staking point for every curve defined in ALIGNMENT *NAME*.

The option EVEN N is the increment along the curves you wish to stake. If you do not use this option, COGO assumes even 100.

Example:

```
LAYOUT ALIGNMENT *PROJ* EVEN 50
```

List Points, Lines, or Curves Command

This list command reports the defined points or lines or curve contained in the COGO model.

LIST (POINT/LINE/CURVE)

This command lists defined points, lines or curves. It lists by number all defined items of the specified type in the COGO tables. The list(s) will be in increasing numerical order.

Example:

```
LIST POINTS
```

List Alignments, Traverses, Parcels, or Text Command

This list command reports the defined alignments, traverses, parcels or text contained in the COGO model.

LIST (ALIGNMENT/TRAVERSE/PARCEL/TEXT)

This command lists the names of defined alignments, traverses, parcels and text. It lists all items of the specified type defined in the COGO tables, by name.

Example:

```
LIST ALIGNMENT
```

List Alignment, Traverse, or Parcel Contents Command

This list command list the items that make up the specified alignment, traverse or parcel contained in the COGO model.

LIST (ALIGNMENT/TRAVERSE/PARCEL) *name*

This command lists the items in the alignment, traverse, or parcel. It lists the items (points, curves, etc.) by name and number, which make up the specific traverse, alignment, or parcel.

Example:

```
LIST ALIGNMENT *SURVEY*
```

List All Items Command

This command lists all the defined items in the COGO model. The report produced is the equivalent of executing all of the other list commands combined.

```
LIST ALL
```

This command lists all the defined items in the COGO model. It lists all defined items in the COGO model by name and number.

Example:

```
LIST ALL
```

List Table Size Command

The List Table Size command is used to determine how many of the available slots for items have been used in the COGO model. This command is obsolete.

```
LIST TABLE SIZE
```

This command reports the number of defined items in the COGO model. It reports the number of defined points, lines, etc., as well as the maximum number permitted.

Example:

```
LIST TABLE SIZE
```

Locate Command Common Parameters and Options

The Locate Commands calculate and store new points based on previously defined points, lines, curves, traverses and alignments.

Most of the locate commands have parameters in common, and in some cases the parameters have multiple syntax. The following is a list of those parameters and their possible forms.

Locate Commands Common Parameters and Options

Parameter Name/ Form(s)	Description
ANGLE -	

ANGLE ±a The angle to be turned from the line defined by 'point m and point p'. If the angle is positive, it will be measured in a clockwise direction. If the angle is negative, it will be measured in a counter-clockwise direction.

ANGLE USE ([PLUS]/MINUS) AT [POINT] v FROM [POINT] s TO [POINT] t
This option gives the ability to define an angle using 3 points with 'point v' being the vertex of the angle. The word POINT is not optional for curve points.

DEFLECTION –

DEFLECTION ±d

The deflection to be turned from the line established by 'point m and point p'. The angle will be to the right if the deflection is positive and to the left if the deflection is negative.

DIRECTION –

AZIMUTH ddd mm ss.ss
The azimuth in degrees minutes and seconds. The word AZIMUTH is optional.

AZIMUTH USE [POINT] p TO [POINT] r
This option may be used instead of supplying an exact bearing. COGO calculates the azimuth of the line between the two points given. The word POINT is not optional for curve points.

AZIMUTH USE LINE k [REVERSED]
This option may be used instead of supplying an exact azimuth. COGO uses the azimuth of 'line k'. If specified the word REVERSED, COGO will reverse the direction.

[BEARING] (n/s) dd mm ss.ss (e/w)
A bearing in the form: [BEARING] (N/S) dd mm_ss.ss (E/W)

BEARING USE [POINT] p TO [POINT] r
This option may be used instead of supplying an exact bearing. COGO calculates the bearing of the line between the two points given. The word POINT is not optional for curve points.

BEARING USE LINE k [REVERSED]

This option may be used instead of supplying an exact bearing. COGO uses the bearing of 'line k'. If specified the word REVERSED, COGO will reverse the direction.

DISTANCE -

[DISTANCE] d

The distance the new point will be located from the defined point.

DISTANCE USE [POINT] p TO [POINT] r

This option may be used instead of supplying an exact measurement. COGO will calculate the distance between the two points specified. The word POINT is not optional for curve points.

NEAR -

NEAR [POINT] t

Ordinarily, the first solution found is returned. However, if this option is used, the whole item will be searched to find up to 7 solutions. Then the point closest to point t will be returned.

ITEM –

ITEM

An item can be a line segment, line, curve, alignment, or traverse. The proper command syntax for each type of item is listed below.

```
LINE SEGMENT [POINT] pl TO [POINT] p2
LINE 1
CURVE c
TRAVERSE *name*
ALIGNMENT *name*
```

OFFSET –

OFFSET ±d

The offset distance normal (90 degrees) to the line. The distance is always positive to the right and negative to the left. It is measured in the direction of the line created by the two points.

OFFSET USE ([PLUS]/MINUS) [POINT] p TO [POINT] r

COGO calculates the distance between the two points specified. It will then use this distance as the offset distance. The word POINT is not optional for each point that is a curve point. The plus or minus indicate whether the distance used will be positive or negative.

Locate a Point Given a Distance and Direction From Another Point

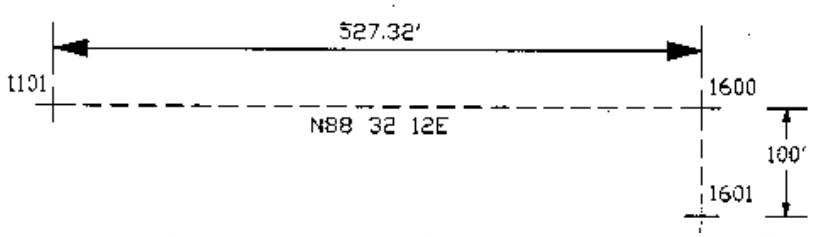
This locate command uses a distance and direction (bearing or azimuth) from a point to locate a new point.

```
LOCATE [POINT] n [*id*] FROM [POINT] m distance direction [offset]
```

This command locates a new point given a distance and direction from a point. It calculates and stores POINT N at the specified DISTANCE and DIRECTION FROM POINT M. If desired, the resulting point may be offset by means of the OFFSET option. Also, if desired, the new point may have an identification *ID* of up to twenty characters. See the section on common parameters for locate commands for a description of DISTANCE, DIRECTION, and OFFSET.

Examples:

```
LOCATE 1600 FROM 1101 DISTANCE 527.32 BEARING N88 32 12E
```



LOCATE 1601 FROM 1101 DIST 527.32 BEAR N88 32 12E OFFSET 100

Locating a Point Given a Distance and Deflection from the End of a Line Segment

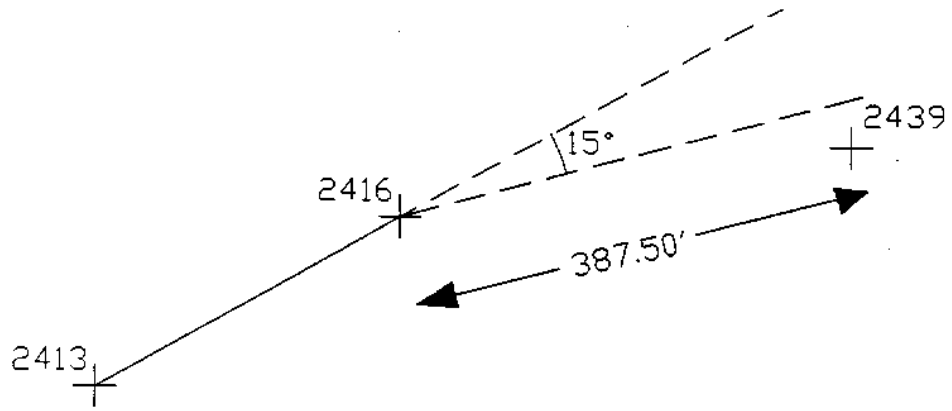
This Locate command uses a deflection from the end of a line segment to locate a new point.

```
LOCATE [POINT] n *id* FROM [POINT] m TO [POINT] p distance deflection [offset]
```

This command locates a point given a deflection from the end of a line segment. It locates POINT n using the line formed FROM POINT m TO POINT p as the line to turn the DEFLECTION from. The DISTANCE and optional OFFSET are applied along the line of deflection which starts at POINT P. Please refer to the section on common parameters for the locate commands for a description of the DISTANCE, DEFLECTION, and OFFSET parameters.

Examples:

```
LOCATE 3 FROM I TO 2 70.7106782 DEFL 90
```



LOCATE 2439 FROM 2413 TO 2416 387.5 DEFL 15 0 0 OFFSET 12.0

Locating a Point Using a Distance and Angle from the Beginning of a Line Segment

This Locate command uses an angle turned from the beginning of a line segment to locate a new point.

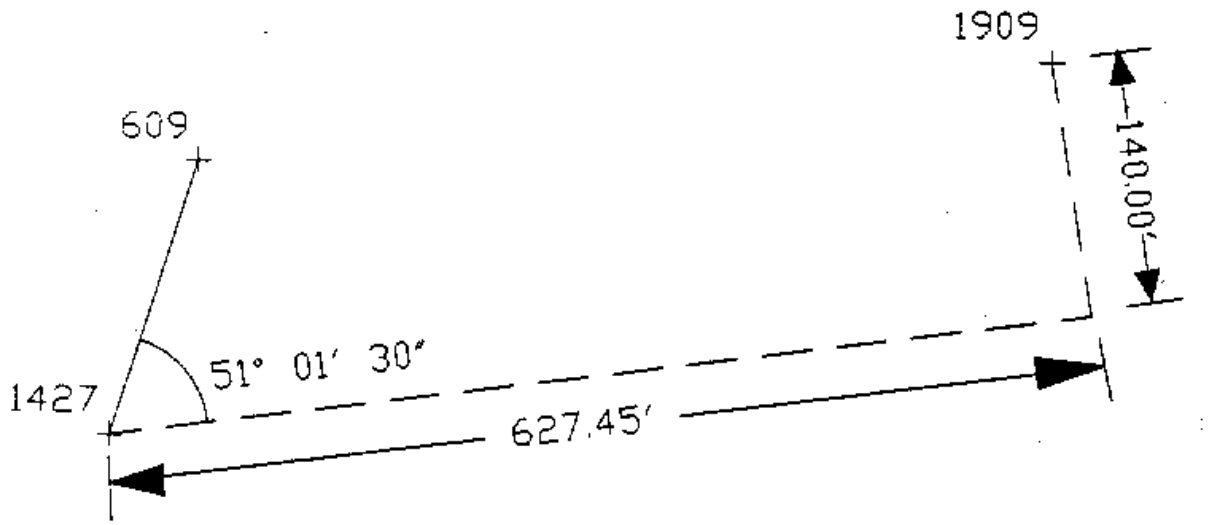
```
LOCATE [POINT] n *id* FROM [POINT] m distance angle FROM [POINT] p [offset]
```

This command locates a point given an angle turned from the start of a line segment. It locates POINT n using the line segment formed FROM POINT m to POINT p as the line to turn the ANGLE from. The DISTANCE and optional OFFSET are applied along the line segment of deflection which starts at POINT M. Please refer to the section on common parameters for the locate commands for a description of the DISTANCE, ANGLE and OFFSET parameters.

! The degrees, minutes and seconds for the angle **must** be included.

Examples:

LOCATE 3 FROM I 100 90 FROM 2



LOCATE 1909 FROM 1427 627.45 ANGLE 51 01 30 FROM 609 OFFSET -140.0

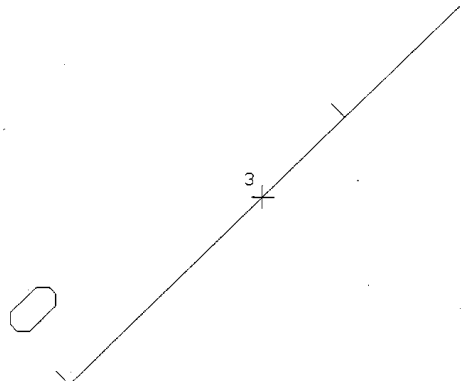
Locating a Point Using a Station on an Alignment

This Locate command uses a station value to locate a point on an alignment.

```
LOCATE [POINT] n *id* ON ALIGNMENT *name* [offset] AT [STA] station [NEAR
[POINT] p]
```

This command locates a point on an alignment at a specific station. It locates POINT n on the ALIGNMENT *name* AT the specified STATION. The point may be OFFSET from the alignment. Please refer to the section on common parameters for the locate commands for a description of the OFFSET parameter.

Example:



LOCATE 3 ON ALIGN *PROJ* AT STA 00+70.71

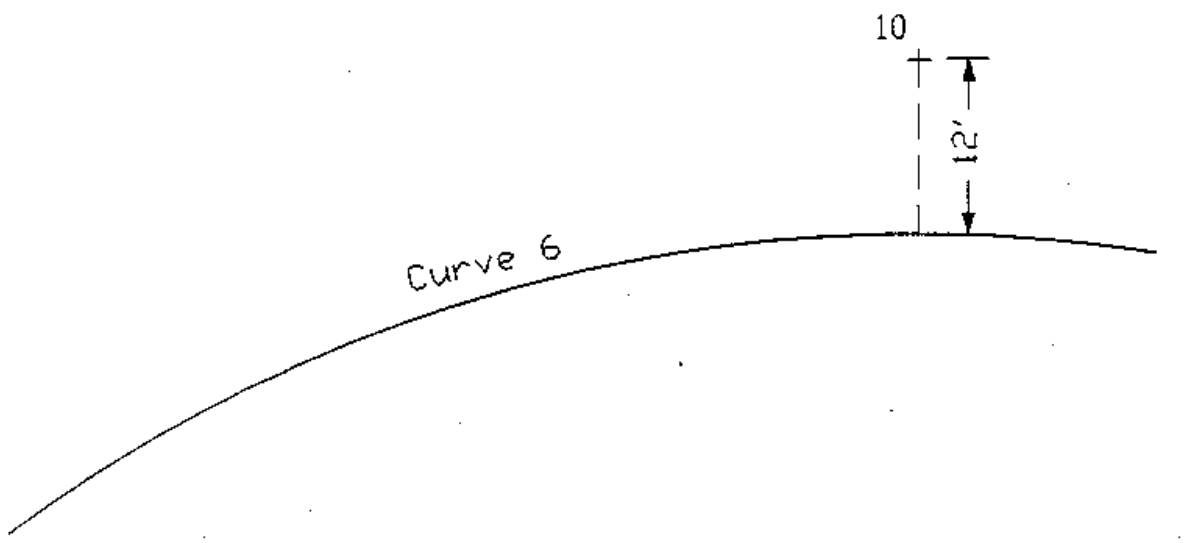
Locating a Point Which Is Offset from a COGO Item

This Locate command uses an offset distance from a specified COGO item to locate a new point.

```
LOCATE [POINT] a *id* OFFSET distance FROM [POINT] p ON item [near]
```

This command locates a point given an offset distance from an item. It locates POINT n OFFSET a DISTANCE FROM POINT p on the ITEM. Please refer to the section on common parameters for the locate commands for a description of the DISTANCE and NEAR parameters.

Example:



LOCATE POINT 10 OFFSET -12 FROM POINT CS 6 ON CURVE 6

Locating Points Which Are Offset from a Curve

This command locates points that are offset to the left and right of curve points. The number of points produced is dependent on the type of curve.

```
LOCATE [POINT] n TO m OFFSET (d / r RIGHT 1 LEFT / 1 LEFT r RIGHT) FROM
CURVE c
```

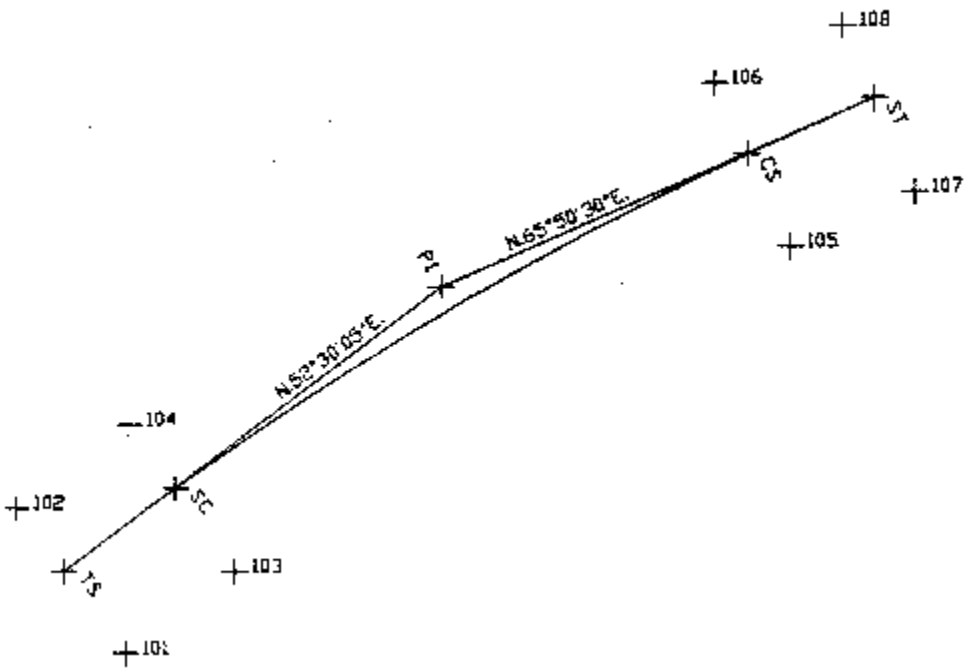
This command locates points offset from a curve at the curve points. It locates POINT n TO POINT m OFFSET FROM the TS/PC, SC, CS, ST/PT of a CURVE c. The points are offset from curve c equidistant (d), or at specific distance r to the RIGHT, and l to the LEFT. The block of points must be the required number, namely 4, 6, or 8 depending whether there are 0, 1, or 2 spirals. The points are located in the direction of the curve starting with the point to the right of the first curve point (PC/TS), then to the left of

the beginning curve point, then to the right of the next curve point, then to the left of the next curve point, etc.

! The OFFSET USE option may be used with this command.

Examples:

```
LOCATE POINT 10 TO 13 OFFSET 100 FROM CURVE 114
      PNT 10      49487.951   48124.220   PC +
      PNT 11      49681.541   48073.992   PC -
      PNT 12      49671.829   48501.080   PT +
      PNT 13      49830.558   48379.404   PT -
```



LOCATE 101 TO 108 OFFSET 85 LEFT 110 RIGHT FROM CURVE 2

Locating a Point by Projecting Another Point Onto a COGO Item

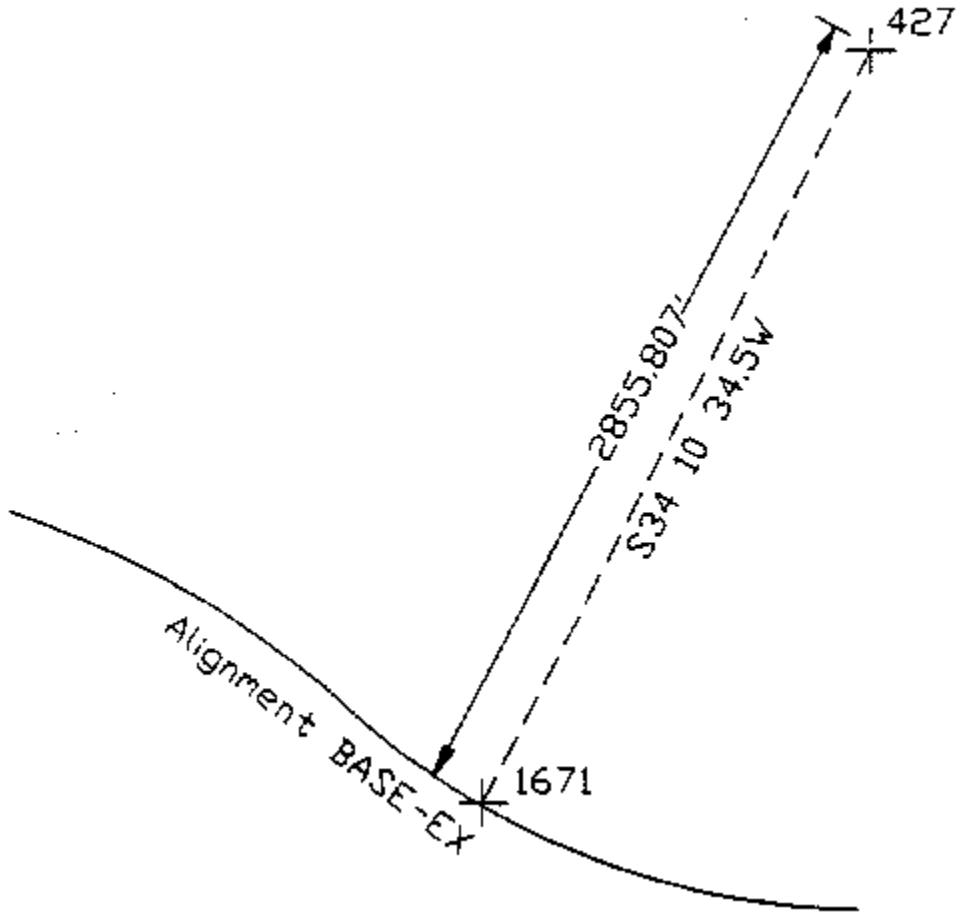
This Locate command projects a point onto a COGO item.

LOCATE [POINT] n *id* PROJECT [POINT] m ON item [offset] [near]

This command locates a point by projecting a point onto an item. It locates POINT n on the specified ITEM as a geometric projection of POINT m. This projection is perpendicular (normal) to the item. COGO reports the north and east coordinates, and the station of the new point. It also reports the distance and the bearing of the line of projection. Please refer to the section on common parameters for the locate commands for a description of the OFFSET and NEAR parameters.

Examples:

```
LOCATE 3 PROJECT POINT 4 ON ALIGN *TEST2*
```




```
LOCATE 1671 PROJECT 427 ON ALIGNMENT *BASE-EX*
```

Locating the Intersection of Two COGO Items

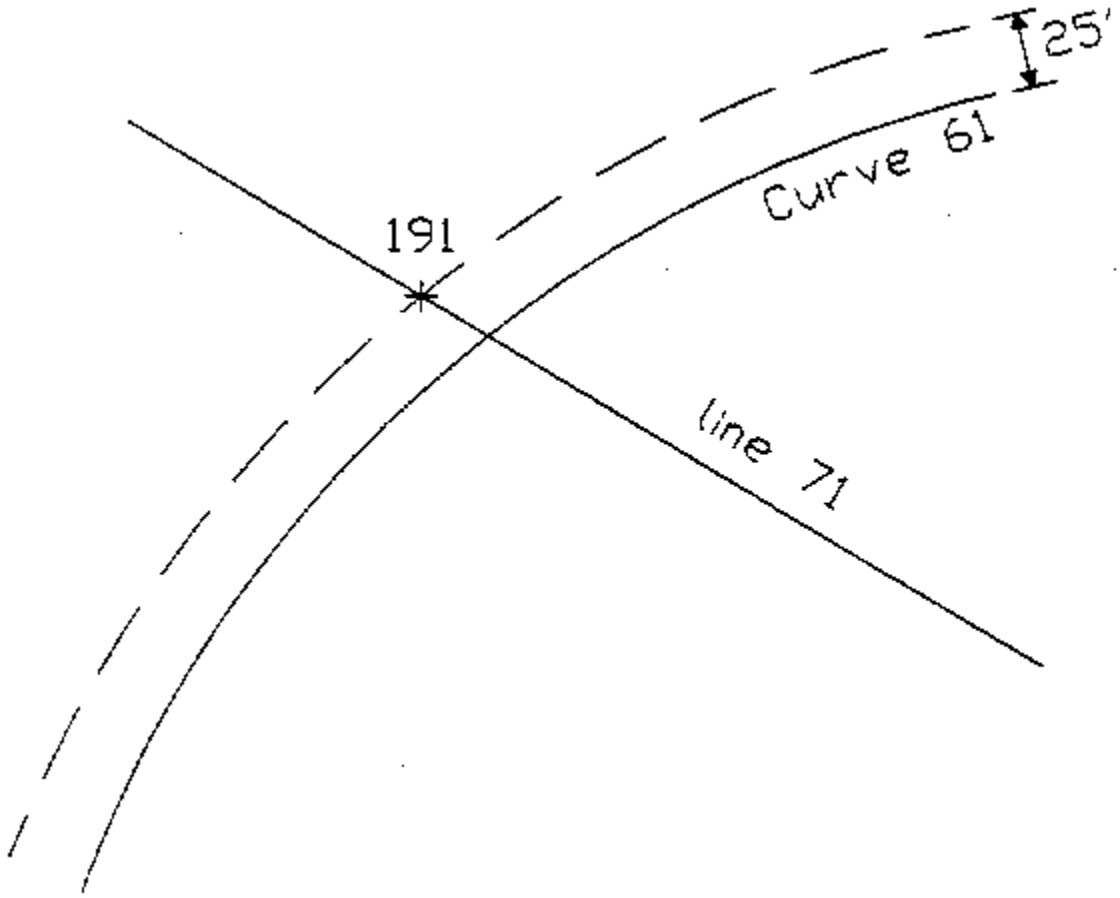
This Locate command intersects two COGO items.

```
LOCATE [POINT] n *id* INTERSECT item1 [offset] WITH item2 [offset] [near]
```

This command locates the intersection of two items. It locates POINT n as the intersection of ITEM1 and ITEM2. Please see the section on common parameters for the locate commands for a description of the ITEM, OFFSET and NEAR POINT parameters. 3D intersections are available for intersecting lines (see the SET 3D command).

Examples:

```
locate 14 intersect CURVE 6 with ali *test b*
locate 15 intersect ali *ramp b* with ali *test b*
```



LOCATE 191 INTERSECT LINE 71 WITH CURVE 61 OFFSET -25

Locating a Point Using Deflections From the Ends of a Line Segment

This Locate command turns deflections from the ends of a line segment and calculates where the deflections intersect.

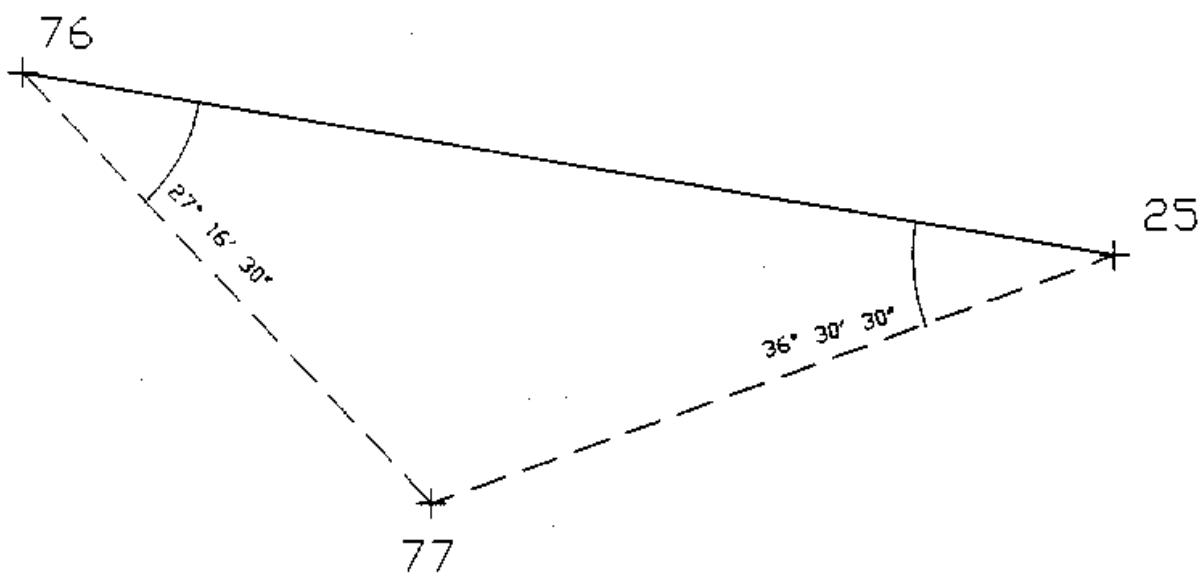
```
LOCATE [POINT] n [*id*] FORSECT [POINT] m (ANGLE1/BEARING1) AND [POINT] p
(ANGLE2/BEARING2)
```

This command locates a point when angles are turned from the ends of a line segment. It locates POINT n by intersecting ANGLEl/BEARING1 and ANGLE2/BEARING2 which are turned from the line segment

formed by POINT m and POINT p. Anglel/Bearing1 will be turned from the line segment at point m, Angle2/Bearing2 will be turned from point p. The first angle/bearing determines on which side of the line segment the calculated point will reside. Please refer to the section on common parameters for the locate commands for a description of ANGLES and BEARINGS.

Examples:

```
LOCATE 3 FORSECT I MINUS 45 00 00 AND 2 45 00 00
```



LOCATE 77 FORSECT POINT 25 ANGLE MINUS 36 30 0 AND 76 ANGLE 27 16 30

Locating a Point Using Two Distances from Two Different Points

This Locate command uses a distance from the first point and a distance from the second point to locate a new point.

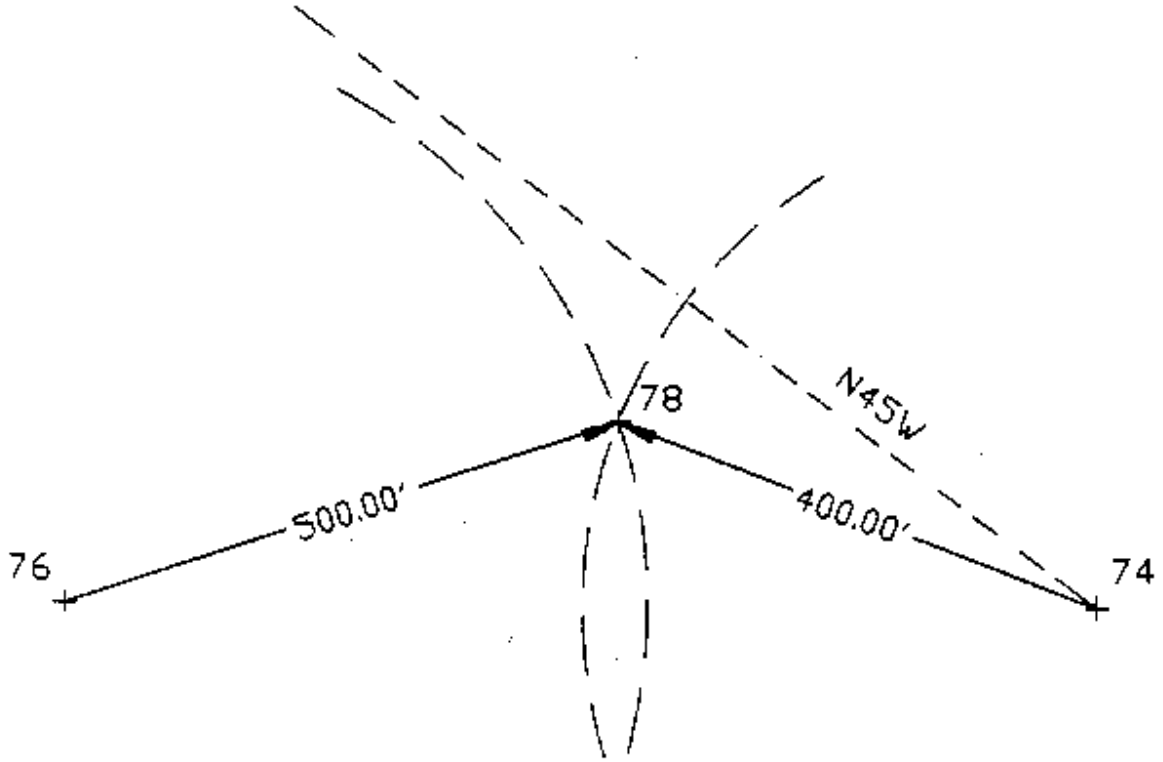
```
LOCATE [POINT] n [*id*] TIE [POINT] m DISTANCE dl TIE [POINT] p DISTANCE d2
[(APPROX direction / near)]
```

This command locates a point given distances from two points. It locates POINT n at the intersection of the two circles defined by the circle centers (POINT m, POINT p) and the radii (DISTANCE dl, DISTANCE d2). If the APPROX parameter is not used, the point chosen of the two possible solutions will be the one with the smaller azimuth for line segment formed by point m to point n. If the APPROX option is specified, the solution chosen will be the one such that the line segment point m to point n has an azimuth

closer to the given direction. Please refer to the section on common parameters for the locate commands for a description of DISTANCE, DIRECTION, and NEAR.

Examples:

LOCATE 21 TIE PNT 15 753.12 TIE PNT 17 123.47
LOCATE 4 TIE 1 50 TIE 2 50 NEAR 3



LOCATE 78 TIE 74 DIST 400 TIE 76 DIST 500 APPROX N45W

Locating a Point Using a Distance from a Point and an Angle Turned from a Line Segment

This Locate command uses a distance from a defined point and an angle turned from the starting point of a line segment to create a new point.

```
LOCATE (POINT) n *id* TIE [POINT] m distance TIE (POINT] p direction (NEAR
FAR / NEAR [POINT] t)
```

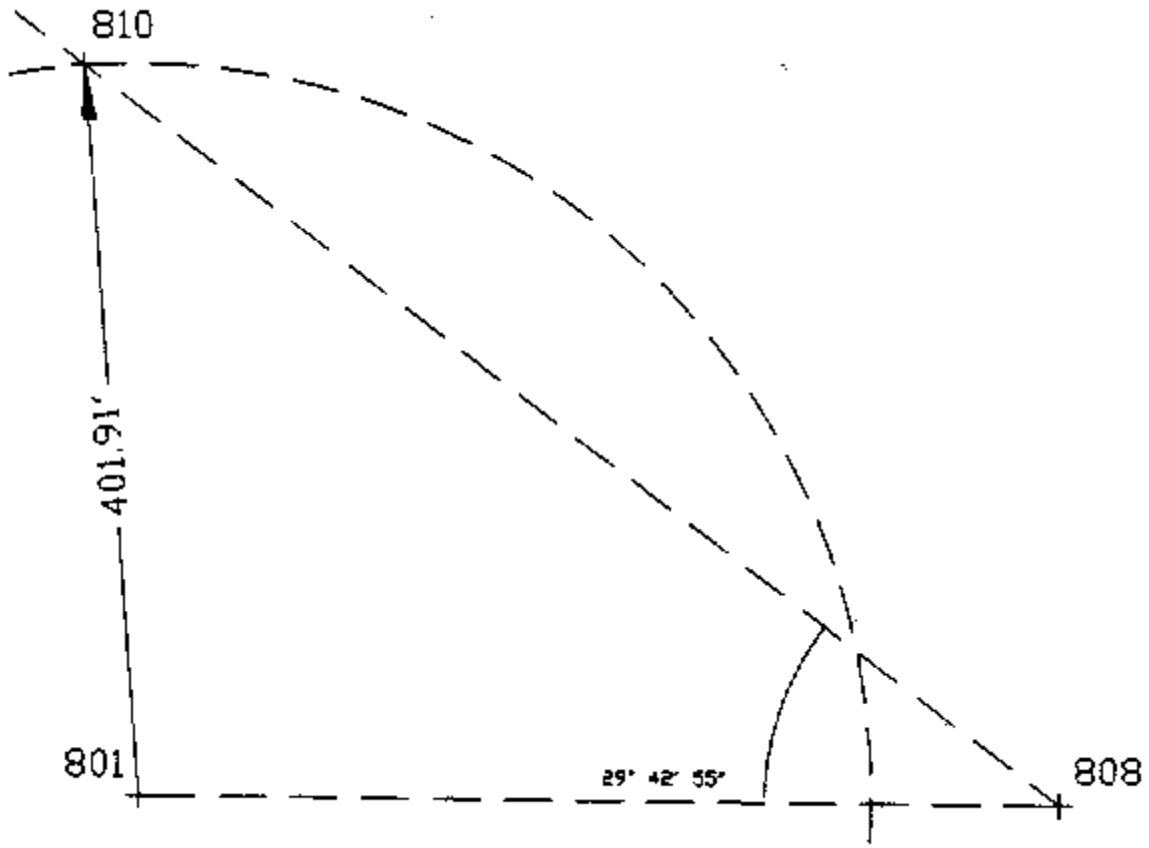
This command locates POINT n at the intersection of the circle with center POINT m**,** radius DISTANCE, and the line formed by POINT p and DIRECTION. The NEAR, FAR and options indicate that the

solution NEARest to or FARthest from point p is the result. Please refer to the section on common parameters for the locate commands for a description of DISTANCE, DIRECTION, and NEAR.

! The word 'ANGLE', 'BEARING', or 'AZIMUTH' **must** be used for the direction.

Examples:

LOCATE 3 TIE PNT 1 50 TIE 2 ANGLE 15 00 00 FAR



LOCATE 810 TIE PNT 801 DIST 401.91 TIE 808 ANGLE 29 42 55 FAR

Locating a Point on a Curve Where the Tangent to the Curve Is a Specified Direction

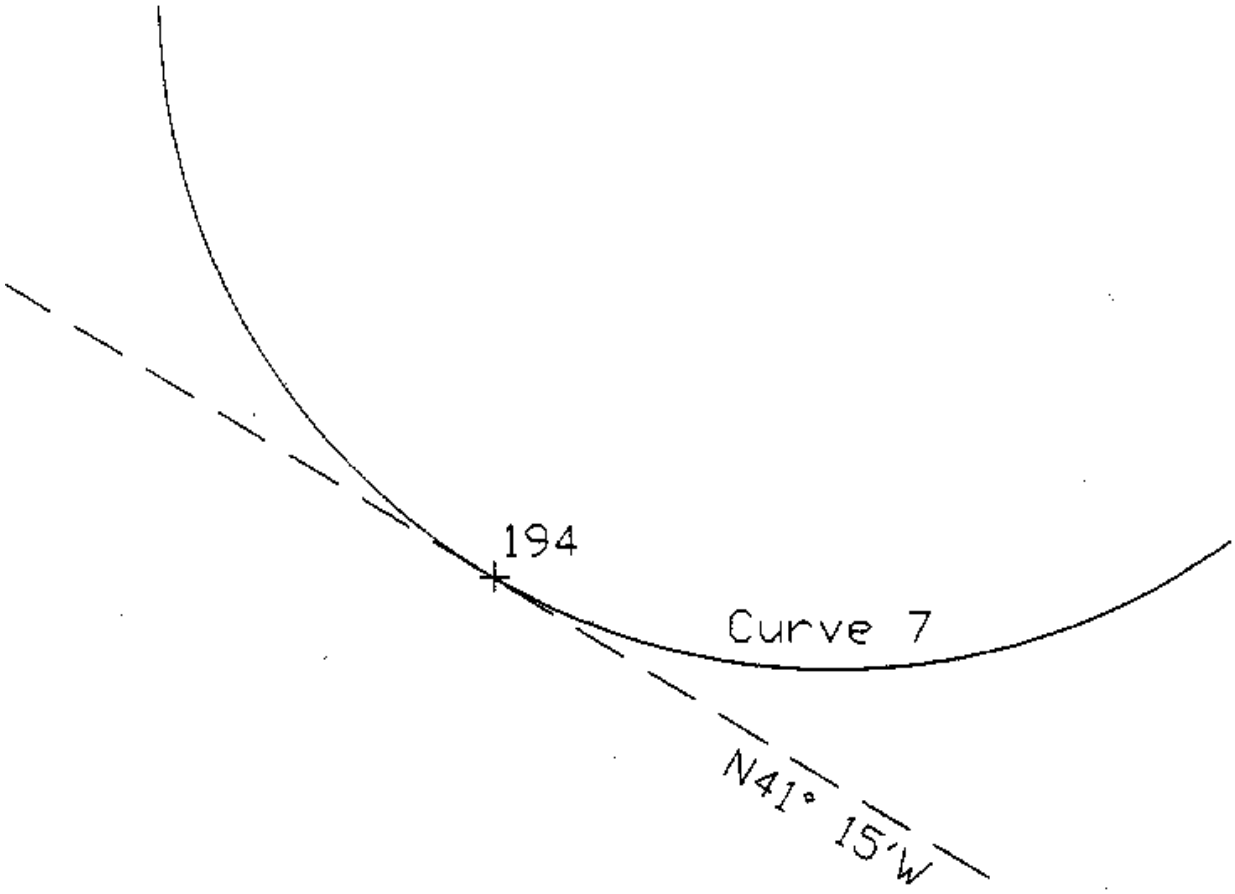
This command locates the point where a line of specific direction is tangent to a curve.

```
LOCATE [POINT] n [*id*] ON CURVE c TANGENT direction
```

This command locates the point of tangency to a curve given a direction. It locates POINT n ON CURVE c at the TANGENT with the specified DIRECTION. Please refer to the section on common parameters for the locate commands for a description of DIRECTION.

Examples:

```
LOCATE PNT 5 ON CURVE 1 TANGENT N90 00 00E
```



LOCATE 194 ON CURVE 7 TANGENT BEARING N41 15W

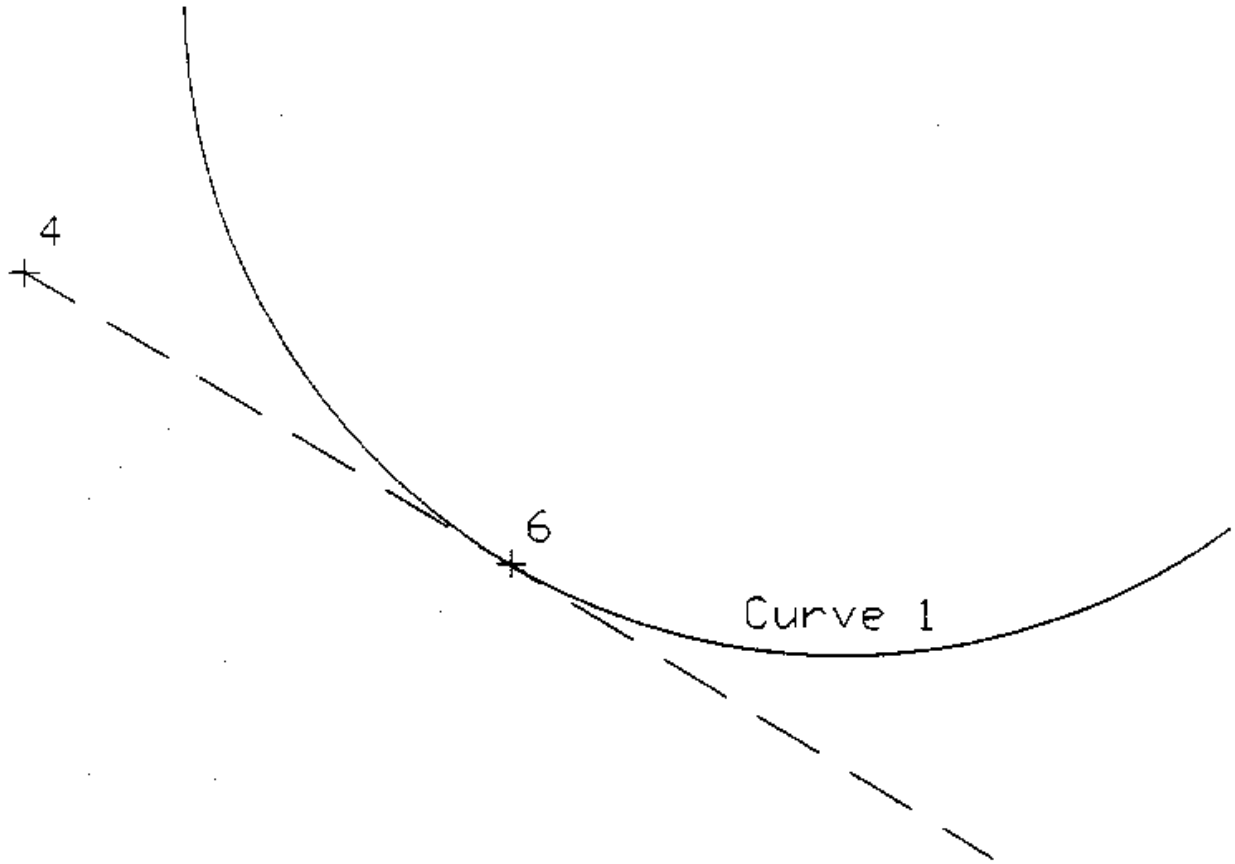
Locating a Point Where the Tangent to the Curve Passes Through Another Point

This Locate command calculates the point of tangency on a curve from a point near the curve.

LOCATE [POINT] n [*id*] ON CURVE c TANGENT THRU [POINT] m [near]

This command locates POINT n ON CURVE c at the TANGENT that passes THRU POINT m, optionally selecting the solution which is NEAR a specified point. Please refer to the section on common parameters for the locate commands for a description of NEAR.

Example:



LOCATE PNT 6 ON CURVE 1 TANGENT THRU PNT 4

Parcel Command

The parcel command stores information about a parcel in the alignment table. The information stored includes tie points, parcel points, arcs in the parcel, the parcel name, and the area if the parcel is closed. The parcel command will accept open and closed parcels.

```
PARCEL *name* [SUBTRACT] [(PRINT/LEGAL)] [REDEFINE]
 [TIE (POINT] m]
   .
   .
   .
BEGIN AT [POINT] n
   .
   .
   .
[POINT] q
   .
   .
   .
ARC   (LEFT/RIGHT) RADIUS r (PLUS/MINUS) CC [POINT] t
   .
   .
   .
ARC   [POINT] p (LEFT/RIGHT) (PLUS/MINUS)
   .
   .
   .
END   AT [POINT] z
```

The PARCEL named *NAME* is stored in the alignment table. If specified, the area of the parcel is SUBTRACTed from the total area register. Two parcel reports are available, either a working report (PRINT) or a LEGAL description. The working report of the parcel is output if the PRINT command word is used on the parcel command line. The legal description is obtained by using the word LEGAL. These report options are also available in the print command. The legal description will be output to a file. The file name will start with the parcel's name, and the extension will be ". COT", e. g., TE150A.COT. If necessary TIE POINTs may be included in the parcel description before the BEGIN POINT which starts the parcel. For closed parcels the begin point MUST be the same point number as the end point. If the begin point doesn't match the end point, the parcel is considered to be an open parcel (access). Each course in the parcel consists of a pair of points or an arc. The courses start at the point defined in the begin and stop at the point in the end. Course 1 is between the first and second points in the parcel definition, course 2 is between points 2 and 3, etc.

Defining Arcs in Parcels

There are two ways to define an arc. The first is the point-arc center-point, the second is the point-radius-direction-point.

<u>Point-Arc center-Point</u>

```
[POINT] n
ARC [POINT] m   (LEFT/RIGHT)   (PLUS/MINUS)
[POINT] p
```

In this syntax the ARC goes from POINT n to POINT p with the center at POINT m with the direction of curvature indicated by LEFT or RIGHT. The PLUS or MINUS indicate whether the resulting area is to be added or subtracted from the parcel area.

Point-Radius-Direction-Point

```
[POINT] n
ARC  (LEFT/RIGHT)   RADIUS r   (PLUS/MINUS)   CC POINT c
[POINT] p
```

In this syntax the ARC having RADIUS r goes from POINT n to POINT p with the direction of curvature indicated by LEFT or RIGHT. The PLUS or MINUS indicate whether the resulting area is to be added or subtracted from the parcel area. COGO will calculate the center (CC) point of the arc and store it in POINT c.

! The calculated area is inaccurate if any side of the parcel crosses another side.

Example:

```
PARCEL *TE150A* LEGAL
      TIE PNT 650
TIE PNT 660
TIE PNT 665
BEGIN AT PNT 670
PNT 671
ARC RIGHT RADIUS 4358.0 MINUS CC PNT 700 PNT 672
PNT 673
ARC PNT 674 RIGHT PLUS
PNT 675
END AT POINT 670
```

```
Parcel TE150A

Beginning at the      ; Thence N. 39° 48' 06" E., a distance of 7811.75
feet; Thence N. 63° 27' 08" E., a distance of 4470.79 feet; Thence N. 21°
42' 17" E., a distance of 919.50 feet to the TRUE POINT OF BEGINNING;

1.    Thence N. 0° 03' 55" E., a distance of 52.67 feet;

2.    Thence along the arc of a curve to the right having a
radius of 4358.00 feet, a distance of 1052.69 feet (the chord of said arc
bears N. 52° 33' 06" W., a distance of 1050.13 feet);

3.    Thence S. 54° 35' 38" W., a distance of 287.76 feet;

4.    Thence S. 35° 24' 22" E., a distance of 200.00 feet;
```

```
5.    Thence N. 54° 35' 38" E., a distance of 123.74 feet;

6.    Thence along the arc of a curve to the left having a radius of
4558.00 feet, a distance of 794.42 feet (the chord of said arc bears S.
52° 44' 01" E., a distance of 793.41 feet);

7.    Thence S. 0° 03' 52" W., a distance of 10.69 feet;

8.    Thence N. 75° 14' 36" E., a distance of 227.55 feet, more or less,
to the TRUE POINT OF BEGINNING.

The above described      contains 5.342 acres/232703 square feet, more or
less.
```

Print Alignment Book Command

Once the final alignment has been settled on, it must be staked in the field. COGO can create horizontal alignment books for staking alignments in the field.

```
PRINT ALIGNMENT BOOK [FOR] ALIGNMENT *name* [EVEN n] [SPIRAL TENTH POINTS]
[CURVE c EVEN a]
IDENTIFICATION LINE
```

This command creates an ALIGNMENT BOOK file for ALIGNMENT *NAME*. The file name is name_BOOK.COT, where name is the alignment name. It reports points as 'POTS" and reports complete curve information. The EVEN N option changes the increment used to generate coordinates. If the increment is not specified the default is 100'. The SPIRAL TENTH POINTS option generates 10th points for spirals. The CURVE C EVEN M option specifies that for curve n the distance used for calculating coordinates should be changed to the distance m.

This command is a two-line command. The first line is the actual command that reports on the alignment at the increment requested. The second line is a 48 character line of identification that will be present at the top of each page of the horizontal alignment book.

! COGO will only allow an alignment to have a maximum of 15 curves for the horizontal alignment book.

Example:

```
PRINT ALIGNMENT BOOK ALIGNMENT *WB ML* EVEN 50 SPIRAL TENTH POINT +
CURVE 31 EVEN 10
Identification Line of 48 characters or less
```

Reporting the Geometry of Points, Lines, Curves, Alignments, Parcels, and Text

The Print commands report the geometry of COGO items in the COGO model.

PRINT ([POINT]/LINE/CURVE/TRAVERSE/ALIGNMENT/PARCEL/TEXT) (list/ALL)

This command is used for reports of an item's geometry. It reports the information stored in the COGO model for the specified item(s). The ALL option may be used with every item type. If the item type is points, lines, or curves and the ALL option is not used, a list of number(s) is required. If the item type is traverses, alignments, parcels or text and the ALL option is not used, a list of name(s) is required. If the PRINT PARCEL *name* is selected, there is another option available which is the option of printing a legal description of the parcel. The following is an example of this. PRINT PARCEL *102* LEGAL

Examples:

```
PRINT POINT 1
PNT 1             10151.107        9762.349


PRINT CURVE 6
CURVE    6    Type: Spiral-Circle-Spiral  Right
    Spiral In Data
       Length     125.000'    SA   3 38 45.0
       LT          83.351'    ST      41.683'
    Circular Curve Data
       Degree   5 50   .0     Deflection 151 15 52.7    Delta 143 58 22.7
       Radius     982.213'    Length        2468.108'   Es       2978.704'
    Spiral Out Data
       Length     125.000'    SA   3 38 45.0
       LT          83.351'    ST      41.683'
    Tangent Back Data (PC/TS to PI)
         S 10 18 23.4 E     Length        3899.526'
    Tangent Ahead Data (PI to PT/ST)
         N 39  2 30.7 W     Length        3899.526'
    TS   1       413408.491        633952.365      0+00.00'
    SC   1       413285.083        633972.112      1+25.00'
    PI   1       409571.885        634650.044     38+99.53'
    CC   1       413171.157        632996.529
    CS   1       412505.215        632274.543     25+93.11'
    ST   1       412600.592        632193.780     27+18.11'


PRINT ALI *TEST*
Alignment  TEST

Pnt 110  49524. N   47867. E   0. El      0+00.000'

***** Station Equation. 2+00.00        back = 2+11.00        ahead.

       433.65042829901'    N 75° 27' 17.559" E
```

```
Curve 114  Type: Circular Curve Left
   Circular Curve Data
       Degree 6 0 0.000   Deflection 22 55 38.055   Delta 22 55 38.055
       Radius 954.930'    Length 382.120'
   Tangent Back Data
       N 75° 27' 17.559" E   193.651332836867'
   Tangent Ahead Data
       N 52° 31' 39.504" E   193.651332836872'
   PC   49585. 48099.  2+50.999'
   PI   49633. 48287.  4+44.650'
***** Station equation. 4+10.00 back =  4+11.00 ahead.
   CC   50509. 47859.
   PT   49751. 48440.  6+34.120'

       387.303'    N 52° 31' 39.504" E

Curve 116  Type: Circular Curve Right
   Circular Curve Data
       Degree 6 0 0.000   Deflection 22 55 38.156   Delta 22 55 38.156
       Radius 954.930'    Length 382.121'
   Tangent Back Data
       N 52° 31' 39.504" E   193.651'
   Tangent Ahead Data
       N 75° 27' 17.660" E   193.651'
   PC   49751. 48440.  6+34.120'
   PI   49869. 48594.  8+27.772'
***** Station equation. 7+15.00 back =  7+10.00 ahead.
   CC   48993. 49021.
   PT   49918. 48781.  10+11.241'

       387.302'    N 75° 27' 17.660" E

*****Warning: Overlap between curve and curve.

Curve 118  Type: Circular Curve Right
   Circular Curve Data
       Degree 6 0 0.000   Deflection 22 55 38.574   Delta 22 55 38.574
       Radius 954.930'   Length 382.123'
   Tangent Back Data
       N 75° 27' 17.660" E   193.652'
   Tangent Ahead Data
       S 81° 37' 3.766" E   193.652'
   PC   49918. 48781.  10+11.243'
   PI   49966. 48969.  12+04.895'
***** Station equation. 10+50.00 back =  0+00.00 ahead.
   CC   48993. 49021.
   PT   49938. 49160.  3+43.366'

       387.303'    S 81° 37' 3.766" E

*****Warning: Overlap between curve and curve.

Curve 120  Type: Circular Curve Left
   Circular Curve Data
       Degree 6 0 0.000   Deflection 22 55 38.675   Delta 22 55 38.675
       Radius 954.930'   Length 382.123'
   Tangent Back Data
       S 81° 37' 3.766" E   193.652'
   Tangent Ahead Data
       N 75° 27' 17.559" E   193.652'
```

```
   PC   49938. 49160.  3+43.368'
   PI   49910. 49352.  5+37.021'
***** Station equation. 7+25.49 back =  7+00.00 ahead.
   CC   50883. 49300.
   PT   49958. 49539.  7+00.000'

       433.65042829901'   N 75° 27' 17.559" E

***** Station Equation. 7+30.00        back = 7+31.00        ahead.

Pnt 124  50019. N   49772. E   0. El                           9+40.998'
```

Print All Command

This Print command reports the geometry for all of the items in the COGO model.

PRINT ALL

This command outputs complete information on all points, lines, curves, traverses, alignments and text stored in the COGO tables.

Example:

```
PRINT ALL
```

Reset Area Command

The Reset Area command sets the total area register to a specific value.

RESET AREA [value]

This command RESETs this total AREA register to VALUE. If value is omitted, it is assumed to be zero.

Example:

```
RESET AREA 102.1
```

Restation Alignment Command

The Restation Alignment command is used to assign new station values to an alignment. This command can also introduce station equation while assigning new station values to the alignment.

RESTATION ALIGNMENT *name* [BEGIN AT STATION s.s] [equation list] [PRINT]

This command restations an Alignment. It RESTATIONS ALIGNMENT *name*. If specified, the stationing will BEGIN AT STATION s.s. If equations are needed they may be reintroduced by means of the LIST. A report of the restationed alignment is obtained by the PRINT option. Each equation in the equation list is entered in the following form.

```
EQ bb+bb.bb aa+aa.aa
```

Example:

```
RESTATION ALIGNMENT *PROJ* BEGIN AT 14+57.23 EQ 18+92.4 18+20.01
```

Restore Command

The Restore command retrieves a COGO model which was saved to disk with the SAVE command.

```
RESTORE [TABLES] (*name*/LOCAL)
```

This command RESTOREs saved COGO TABLES. The files restored are name.PNT, name.LIN, name.CUR, and name.ALI.

! Care must be taken when COGO models are saved and restored in the same run.

Examples:

```
RESTORE *12345Cl*
RESTORE LOCAL

RESTORE TABLES *name1*
SAVE TABLES *name2* no redefine
Identification line
```

In the above example, you restore a table and save a table in the same run, thus file 'name2' will replace file 'namel', if name1 and name2 are the same file name. COGO will in effect purge 'name1'. However, if the NO REDEFINE option is used, file 'name1' will be left alone and file 'name2' will be saved. Only file 'name2' will have the new design information.

Rotate Translate Command

The Rotate Translate command rotates and/or translates all items in the COGO model. The model can be rotated about a specified point, translated by a distance and direction (indicated by two points), or may be rotated and translated at the same time.

```
ROTATE TABLES [BY ddd mm ss.ss] [TRANSLATE TO POINT m] ORIGIN AT [POINT] n
```

This command ROTATEs all items in the COGO TABLES BY the angle specified with DD MM SEC about the ORIGIN AT POINT n, and if specified TRANSLATE the ORIGIN AT POINT n TO POINT m. If the BY DD MM SS.SS option is omitted the tables will not be rotated. If the TRANSLATE option is omitted, the tables will not be translated. One or both of the options MUST be used or an error will be issued. The ORIGIN AT POINT n is **not** optional. The angle may be positive (clockwise) or negative (counterclockwise). If the angle is negative the negative sign **must** be in front of the first non zero value in the angle. Point m and point n **must** be previously defined points in the COGO tables.

Examples:

```
ROTATE TABLES BY 45 00 00 TRANSLATE TO POINT 20 ORIGIN AT PNT 10
ROTATE TABLES BY 45 TRANSLATE TO POINT 20 ORIGIN AT PNT 10
ROTATE TABLES BY 00 -45 00 ORIGIN AT PNT 100
ROTATE TABLES TRANSLATE TO PNT 101 ORIGIN AT POINT I
```

Save Command

This command saves the COGO model in a compact form as a group of saved tables. The RESTORE command retrieves the saved tables.

```
SAVE [TABLES] (*name*/LOCAL)
identification line
```

This two-line command SAVES the COGO tables as files on disk. The names of the saved tables are name.PNT, name.LIN, name.CUR, name.ALI. If the 'local file' is used instead of the file name, COGO will name the saved tables LOCAL.PNT, LOCAL.LIN, LOCAL.CUR and LOCAL.ALI. The IDENTIFICATION LINE is used for identification of the saved tables. Please note that there can be no blank lines between the SAVE command line and the identification line.

Examples:

```
SAVE *89042C I* NO REDEFINE
This is an identification line

SAVE *12345C2* REDEFINE
Identification line...

SAVE *90134*
I-36 and I-25 junk
```

Set No Redefine Command

The SET NO REDEFINE command, when set to 'on' is designed to issue a warning when existing data in the COGO model is changed.

SET NO REDEFINE (ON/OFF)

The purpose of this command is to protect existing data. If the NO REDEFINE switch is SET 'ON' any attempt to redefine the values of a stored point will result in a warning. This switch may be used to protect against reusing a point number accidentally.

! If a point number is used more than once, the last coordinates used or calculated will be the coordinates stored in the COGO tables. This will occur if the point has been defined by storing it, or using it in a curve or in an alignment. It is wise for you to set the no redefine switch 'on' to warn about point redefinition.

Default Setting: SET NO REDEFINE OFF

Example:

```
SET NO REDEFINE ON
```

Set Date Command

The Set Date command sets the date which COGO uses for the current run. This command can be used to change the format of the date.

SET DATE *date*

This command sets the date COGO uses on reports to a desired date or format for the date. The date can be any I to 10 characters of text (letters/numbers) that will replace the date in subsequent headings. The date **must** be surrounded by asterisks.

Default Setting: yy/mm/dd

Where: yy = year, mm = month, dd = day

Example:

```
SET DATE *01/12/25*
```

Set File Commands

COGO is designed to use input files and output files. The SET FILE commands allow the COGO user to control the input and output via files, and provides interfaces to other engineering applications and devices.

Set Log File Command

The Set Log File command creates a log file of all commands that are entered interactively.

```
SET LOG FILE (ON/OFF)
```

This command logs interactive input to COGO. The command logs all input commands in the file "COGO.LOG" so the log file may be used as an input file for recovering lost work.

Default Setting: SET LOG FILE OFF

Example:

```
SET LOG FILE ON
```

Set File Input

The Set File Input command redirects input to COGO from the keyboard to an input file.

```
[SET FILE] INPUT *name*
```

This command redirects input to COGO from the keyboard to an input file. The file extension **must** be '.CIN', i.e., if the file name is 88047CG then the full file name would be 88047CG.CIN.

Example:

```
SET FILE INPUT *88037CG*
```

Set File Input TTY Command

This Set File Input TTY command redirects input from a file to the keyboard.

```
[SET FILE] INPUT *TTY*
```

This command is **a** special case of the SET FILE INPUT command. It informs COGO that the input should be read from the keyboard (*TTY*). If you do not want to display the commands input, the TTY option should be used.

! This command is generally used when you are done using an input file and wish to enter more commands from the keyboard.

Example:

```
SET FILE INPUT *TTY*
```

Set File Output Command

The Set File Output command redirects the COGO output from the screen to a file.

```
[SET FILE] OUTPUT *name* [(METRIC/FEET)]
```

This command allows you to redirect COGO output. The command redirects COGO output from the COGO.COT file to the file *NAME*. The file name (excluding extension) may be up to 36 characters in length (dependent on the type of computer). The file extension will be '.COT', i.e., if the file name is 88047CG then the full file name will be 88047CG.COT. The METRIC/FEET option will convert from feet to meters or meters to feet depending on the current units. Note that if the units are feet and feet output is specified, no conversion occurs (the same is true for metric/metric).

Examples:

```
SET FILE OUTPUT *88014RAMP*
SET FILE OUTPUT *86999*
```

Set MOSS File Output Command (COGO to MX Interface)

The SET MOSS FILE OUTPUT command is the interface between COGO and MX (formerly MOSS). The result of this command is a MX HALGN input file.

```
SET MOSS FILE OUTPUT *name* [(METRIC/FEET)] ( ALL /

                  ALIGNMENT
                  list of alignments
                  END /
                  TRAVERSE
                  list of traverses
                  END )
```

This command transfers geometry data to MX (formerly MOSS). The command converts alignments or traverses from COGO format to MX HALGN format and creates a file that can be input into MX. Conversion from feet to metric or metric to feet is done via the METRIC/FEET option. The command will read the COGO alignments/traverses and create a MX HALGN alignment composed of 'fixed" straight and 'fixed' curve elements, and provides two methods for transferring alignments/traverses to MOSS. The first will transfer all of the alignments/traverses in the COGO table to MX HALGN format (ALL). The second transfers a list of alignments/traverses. The list begins with the command word ALIGNMENT or TRAVERSE that immediately follows the file name. Each alignment/ traverse name in the list is then entered on a separate line and followed by the command word END.

! If the MOSS HALGN file specified already exists, the new alignment/traverse data will be added to the end of the existing file.

Due to the differences in COGO and MOSS it is possible to construct alignments in COGO that do not translate correctly to MOSS. It is your responsibility to be aware of possible conflicts between the two systems.

This command creates a file that is composed of five MOSS record types. The first record contains the MOSS command HALGN. The second record, minor option 300, defines the alignment control options. Minor options 301 and 305 and are used to define elements of the alignment. Minor option 999 terminates the alignment definition.

MOSS Record Definition:

HALGN - This is the first record in the file design the MOSS major option. You must supply an appropriate MOSS model name.

Minor Option 300 - This is the initial MOSS HALGN data record. The values assigned to the fields in this record are extracted from the COGO alignment or default values are used.

Field Descriptor | Description

LB The 4-character MOSS string beginning with the characters 'MC'. MOSS requires that the first two characters of an alignment be 'MC', the remaining three characters that will compose the string label will be the first 2 characters of COGO alignment name.

SC The starting station of the alignment. The default value will be the beginning station of the alignment.

CF The station of the first point of the first element of the alignment. 'Me default value will be the first station of the alignment.

CE Station interval of the alignment. The default value is 100 feet.

IM Defines the units to imperial.

Minor option 301 - This option contains the element data used to define the 'fixed' straight and curve elements of the MOSS alignment.

Field Descriptor | Description

X1 The east coordinate of the first point of the element being defined.

Y1 The north coordinate of the first point of the element being defined.

X2 The east coordinate of the second point of the element being defined.

Y2 The north coordinate of the second point of the element being defined.

Minor option 305 - This is a continuation record. This record is used to define the circular curve radius and the spiral transitions if applicable.

Field Descriptor | Description

RA The circular curve radius.

LI The length of the spiral transition into the circular curve.

TI The length of the spiral transition out of the circular curve.

Minor Option 999 - This is the termination record indicating the end of the data for the major option.

Examples:

(Input)
SET MOSS FILE OUTPUT *MOSS-FILE.INP* ALL

SET MOSS FILE OUTPUT *MOSS-FILE.INP* ALIGNMENT
WBL
CLM
END

(Output)
HALGN
300,LB=MCLM,CF=1155629.57,SC=1155629.57,CE= 100,IM
301,1,SX,X1=1539721.585000,Y1= 653507.444000,X2=1539721.616308,Y2=653507.441879
301,2,LX,X1=1539921.368427,Y1 = 653498.284248,X2=1540824.590341,Y2=653791.407792
305,RA=1527.887454,L1=200.000000,T1=200.000000
301,3,SX,X1=1540980.898211,Y1=653916.117941,X2=1541035.598000,Y2=653961.748000
999

Set Roadcalc File Output Command (COGO to Roadcalc Interface)

The Set Roadcalc File Output command creates the subproject files in Roadcalc 13's internal format for alignments. This command transfers survey and design alignments.

SET ROADCALC FILE OUTPUT SUBPROJECT nnn ALIGNMENT *name* [DESIGN] [(METRIC/FEET)]

This command transfers Alignments to Roadcalc. It converts alignments from COGO internal format to Roadcalc 13's internal format. All files necessary to transfer the survey or design alignment data to the Roadcalc SUBPROJECT nnn are created in the current directory. These files MUST reside in the directory specified for the subproject. The DESIGN option specifies that the design alignment files are desired rather than the survey alignment files, which is the default. A minimum of two files are created, a maximum of four may be created. The files created for the survey alignment are named RCSAGnnn.DAT, RCSAGnnn.TCD, and if the alignment has station equations, RCSEQnnn.DAT, and RCDEQnnn.DAT. The files created for the design alignment are named RCDAGnnn.DAT, RCDAGnnn.TCD, and if the alignment has station equations RCDEQnan.DAT.

! COGO can create curves with overlaps. Roadcalc cannot handle curve overlaps. All curve overlaps in COGO must be resolved before transferring alignments to Roadcalc.

Examples:

```
SET ROADCALC FILE OUTPUT SUBPROJECT 003 ALIGNMENT *SURV*
SET ROAD FILE OUTPUT SUE 010 ALI *ML WB* DESIGN
```

Set SDR File Output Command (Leitz data collector interface)

The Set SDR File Output Command creates a coordinate file that can be downloaded to the Leitz SDR series 2, 20, and 30 data collectors and can be transferred to PICS. There are two styles of the command to meet these purposes.

```
SET SDR FILE OUTPUT *fname* ALI *aname* [FROM sta1] [TO sta2] [EVEN n]
```

This command creates an SDR data collector file. It creates a file that may be uploaded to a Lietz SDR series data collector. The files created are fname.SDR and fname.COT. The SDR file contains the coordinates of the points along the specified alignment (*aname*) starting at the 'from' station (default = beginning station), ending at the 'to' station (default = end station), at even intervals (default = 100'). The coordinates of the curve points are automatically output to the SDR data collector file. The COT file is a report of the contents of the SDR file.

Examples:

```
SET SDR FILE OUTPUT *SURV* ALI *SURV* EVEN 25.0
SET SDR FILE OUTPUT *PROJ* ALI *PROJ* FROM 13+00 TO 14+00 EVEN 50
```

```
SET SDR FILE OUTPUT *name* *code* list of points
```

This SDR command style creates an SDR file with all of the points in the LIST OF POINTS using the CODE as the TMOSS code in the SDR file. The list of points may be individual points, or in the form of a range of points (e.g. PNT 1 TO 20 PNT 15 PNT 20). The recommended TMOSS codes are 1086 and 1075 for the control points, and 1080 for property control points.

Example:

```
SET SDR FILE OUTPUT *SDRfile* *1086* 105 to 111 50 54 79
```

Set Three Dimensions Command

This command primarily controls how the COGO LOCATE commands work. The default is two-dimensional point locations. If the three dimension switch is set, the locate commands are three-dimensional. At this time, the only command this affects is the LOCATE INTERSECT command for a 3d line/line intersect.

```
SET 3D (ON/OFF)
```

Examples:

```
Set 3D Off
Set 3D On
```

Set Units Command

The Set Units Command is used to set the units used in the COGO geometry model. **This must be the first command in an input file**, or the first operation in COGO. If this command is used at any other time, the results are unpredictable. The default units are set to feet.

```
SET UNITS (FEET/METRIC)
```

Examples:

```
SET UNITS METRIC
SET UNITS FEET
```

Station Alignment Command

The Station Alignment command produces a report of coordinates for an alignment at specific stations or at intervals on the alignment.

```
STATION ALIGNMENT *name* [EVEN d] [FROM  sta1] [TO sta2]
```

This command calculates alignment coordinates at regular intervals along an alignment. It calculates the north and east coordinates for each STATION on ALIGNMENT *****name***** requested. Coordinates will be generated for every EVEN increment of distance D on the alignment. If the EVEN option is not used, the default interval is 100 feet/meters depending on the units in use. If coordinates for only a part of the alignment are needed, the stations may start FROM STATION m and go TO STATION p.

Example:

```
STATION ALIGNMENT *PROJ* EVEN 50 FROM STA 10+18.23 TO STA 16+20
```

Station Offset Alignment Command

The Station Offset Alignment command produces coordinates on an offset alignment by projecting points in the base alignment onto the offset alignment. The report produced also contains the distance from one alignment to the other and the skew.

```
STATION OFFSET ALIGNMENT *namel* [OFFSET d] [STORE FROM m] FROM ALIGNMENT
*name2* [STORE FROM p] [EVEN n] [FROM STATION s] [TO STATION t]
```

This command calculates OFFSET ALIGNMENT coordinates, based on intervals along alignment name2. It uses the specified stations on ALIGNMENT name2 (the base alignment) and projects a point on ALIGNMENT namel (the offset alignment) at EVEN n intervals. If the points are to be stored in the point table the STORE FROM option should be used. The STORE FROM m will store the points for the offset alignment (namel), and the STORE FROM p will store the points for the base alignment (name2). Coordinates will be generated for every EVEN increment of distance N on the alignment. If coordinates for only a part of the alignment are needed, the stations may start FROM STATION s and go TO STATION t.

Examples:

```
STATION OFFSET ALI *ML WB* FROM ALI *ML EB* EVEN 100
STATION OFFSET ALI *PROJ I * FROM ALI *PROJ2* EVEN 50 FROM STA 144+10 +
TO STA 150+00
```

Store Alignment Command

The Store Alignment command creates an alignment in the COGO model.

```
STORE ALIGNMENT *name* list
```

This command defines and STORES ALIGNMENT *NAME* using a LIST of previously defined points and curves. The alignment name may be up to eight characters in length.

The list may be one or a combination of the following forms.

Point numbers

The point numbers of points previously defined. These points cannot be curve points. The word POINT is optional when listing the points.

Block of points

Point numbers designated by using the form 'n to m'.

Curve number

The curve or spiral number of previously defined curve or spiral. The word CURVE must be used in this option i.e., CURVE 6.

Block of curves

Curve or spiral numbers designated by using the form 'n to m'. The word CURVES must be used in this option. i.e., CURVES 6 TO 8.

! The store alignment command does not accept curve points as input.

Examples:

```
STORE ALIGNMENT *SH40SUR* 1 CURVE 2 CURVE 3 4
STORE ALI *PROJ* 10 TO 15 CURVE 3 20
```

Store Curve Command

The Store Curve command creates a curve in the COGO model.

```
STORE CURVE c, pi reference, ahead reference, circular item, [spiral item]
```

This command calculates and stores/redefines a curve. All of the elements of this command are order independent. There must be a PI reference, an ahead reference, a circular item, and optionally a spiral item. If you input more information than necessary COGO may ignore the extra information.

PI REFERENCE (select one of the following)

PB n PI m

PB n BA dir TTL dist

PB n BA dir TL dist

TS n BA dir

PC n BA dir

TS n PI m

PC n PI m

PC n

Where:

BA Is the back azimuth or bearing

PB Is the point on the back tangent. It may be a point number, curve point, or north east station.

PI Is the point of intersection of the tangents. It may be the same form as the PB without STA.

TS Is the tangent to spiral point. It may be the same form as the PB.

PC Is the tangent to circular curve point (point of curvature). It may be the same form as the PB. Note The PC n form may only be used in conjunction with the ahead ref item PT n and the circular ref. item CC n.

TTL Is the total tangent length, PB to PI.

TL Is the tangent length, PB to TS/PC.

! The PI station takes precedence over the PC station.

AHEAD REFERENCE (select one of the following)

PA n

AA dir

DEFL ±angle

DELTA ±angle

PT n

Where:

PA Is the point on the ahead tangent. It may be the same form as the PB. This point may be on either side of the end of the curve. Note PA can only be used if a PI is given.

AA Is the ahead azimuth or bearing.

DEFL Is the total deflection at the PI. + indicates right, - indicates left.

DELTA Is the circular curve delta. + indicates right, - indicates left.

PT Is the point of tangency. Note This option can only be used with a PC n and a CC n.

CIRCULAR ITEM (select one of the following)

RADIUS r

DEGREE deg min sec

LENGTH d

TANGENT d

CC AT n

CC n

Where:

RADIUS	Is the radius of the curve.
DEGREE	Is the degree of curvature.
LENGTH	Is the length of the curve. (only if delta or deflection given, non spiraled)
TANGENT	Is the tangent length (TS/PC to PI or PI to ST/PC). This may only be used with a DELTA ahead ref. (only if delta or deflection given, non spiraled)
CC AT n	Is the circular curve center. It can be the same form as the PB. (only if non-spiraled)
CC n	Is the point where the curve center resides. Note This can only be used with the PC n and PT n.

[SPIRAL ITEM]

SLB dist

SLA dist

SL dist

SAB angle

SAA angle

SA angle

Where:

SLB Is the back spiral length.

SLA Is the ahead spiral length.

SL Is the spiral length for in and out spirals.

SAB Is the back spiral angle.

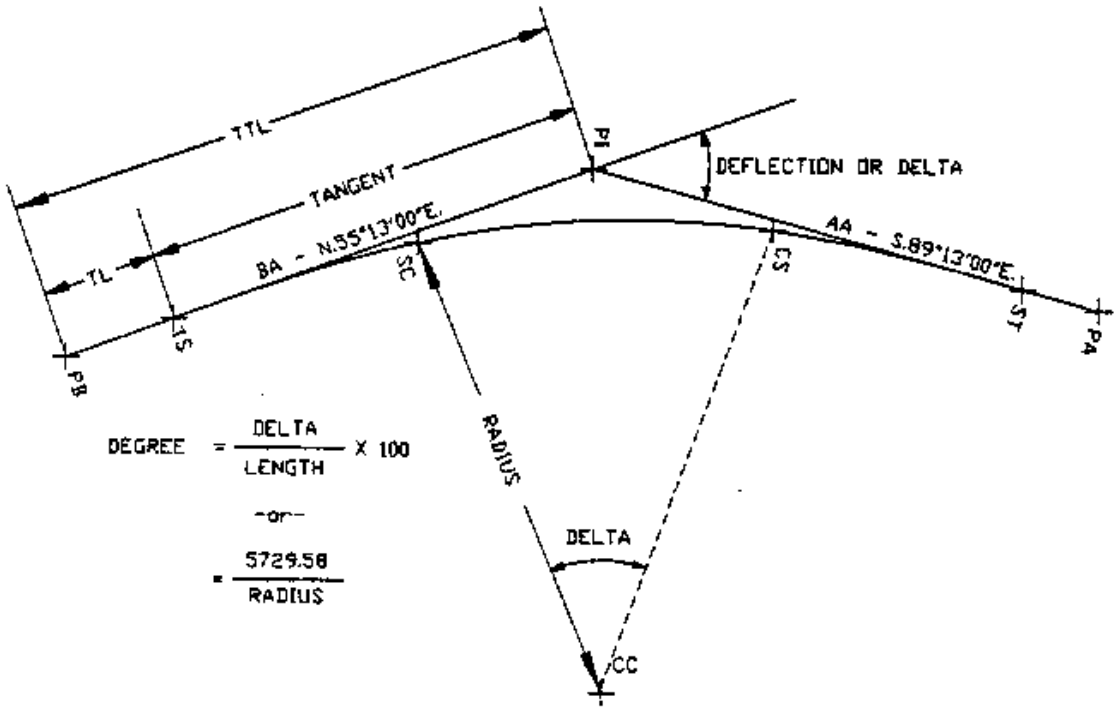
SAA Is the ahead spiral angle.

SA Is the spiral angle for both in and out spirals.

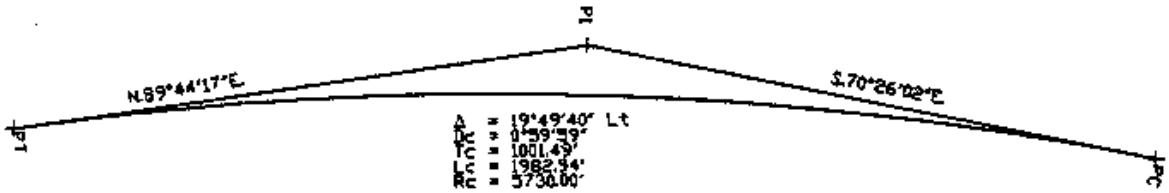
Please refer to next page for diagrams.

Examples:

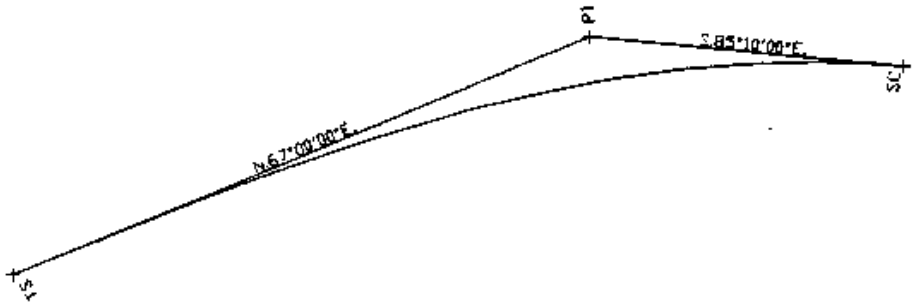
```
STORE CURVE I PC POINT PT 2 PI 1 PA 3 RADIUS 1015.5 SLB 250 SLA 300
STO CURVE 4 PC 1162 PT 1165 CC 1166 SL 200.0
STO CUR 5 PC 1163 PT 1164 CC 1167
STO CUR 10 PC 706 BA N82 15 01.3E DELTA 23 0 0 RADIUS 1254.32
STO CURVE 15 PB 14 PI 4131.8 5528.65 CC AT 1721.3 2474.4 DELTA 20 0 0
```



COMPLEX CURVE



SIMPLE CURVE



SPIRAL CURVE

Store Line Command (direction form)

The Store Line command creates a line in the COGO model.

```
STORE LINE n THRU (POINT) p direction [ZENITH zz zz zz.zz]
```

This command stores a line. It creates/redefines a line n using POINT p, a DIRECTION (bearing/azimuth), and an optional ZENITH zz zz zz.zz.

! Bearings are entered (n/s) deg min sec (e/w). If min/sec or sec are zero, they may be omitted. No blanks are required after the (n/s) or before (e/w).

Azimuths are entered 'deg min sec'. If min/sec or sec are zero, they may be omitted. An azimuth may be made negative by preceding the first non-zero value with a minus sign

Spaces are required between the degrees minutes and seconds in the bearings and azimuths.

Examples:

```
STORE LINE 10 THRU 10 N 45 00 OOE
STORE LINE 10 THRU 10 N 45 E
STORE LINE 12 THRU PNT SC 5 N 47 12 E
STORE LINE 20 THRU 25 BEARING USE POINT 35 TO 70
```

Store Line Command (point to point form)

This Store Line command creates a line segment in the COGO model.

```
STORE LINE n THRU [POINT] TO [POINT] k
```

This form of the STORE LINE command stores a line segment starting at POINT p ending at POINT k.

Example:

```
STORE LINE 1 1 THRU 15 TO 20
```

Store Point Command

The Store Point Command creates points in the COGO model.

```
STORE [POINT] p [NORTH] n [EAST] el [ELEV el] [*id*]
```

This command creates/redefines POINT p having northing NORTH n, easting EAST e, ELEVation el, and point *ID* in the point table. Any item omitted is undefined for POINT P. If you want to be informed of any changes in the point table the SET NO REDEFINE command may be used. The point id may be up to 20 characters long.

! If the optional words are not used, COGO assumes the order of the input to be NORTH, EAST, STATION, ELEVATION, POINT ID. If the identifiers are used, they are order independent. Once an item is omitted, the succeeding identifiers must be used.

Examples:

```
STORE PNT 10 56039.661 196794.108
STORE 20 N 55796.605 E 197236.294 ELEV 5282.519
STORE 10 10000 10000
```

Copy Point Command

The Copy Point command copies information from one point and stores it in another.

```
STORE [POINT] p POINT k [*id*]
```

This command copies a point. It copies a previously stored point or stored curve point. All information from the old point (point k) will be duplicated in the new point (point p). However, you can give the new point a different id by using the appropriate option.

Examples:

```
STORE 10 PNT 20 STA 132+00.0
STORE 20 point cc 88
```

Store Spiral Command

The Store Spiral Command stores a stand alone spiral in the COGO model.

```
STORE SPIRAL n back reference item, pi reference item, ahead reference item,
spiral reference item, degree of simple curve back, degree of simple curve
ahead
```

This command stores/redefines a stand alone spiral. All items must be specified, and are order independent.

Back reference item

PB n - where n is a point on the back tangent.

BA direction - where direction is the azimuth or bearing of the back tangent.

PI reference item

PI n - where n is the PI of the spiral.

Ahead reference item

PA n - where n is a point on the ahead tangent.

Spiral reference item

SL distance - where distance is the length of the spiral.

DELTA d - where d is the total spiral angle.

Deg of simple curve back

SDCB d - where d is the degree of curvature of the preceding simple curve.

Deg of simple curve ahead

SDCA d - where d is the degree of curvature of the simple curve ahead.

! If the simple curve ahead or back is a tangent enter zero as the degree of curve.

Examples:

```
store spiral I pb I pi 2 sl 180 pa 3 sdcb 0 0 0 sdca 2 45 0
store spiral 2 ba azimuth use 4 to 5 pi 5 sl 400 pa 6 sdcb 0 0 0 sdca 10
0 0
```

Store Text Command

The Store Text command stores comments in the COGO model, and if the text name is the same as an alignment or traverse associates it with the alignment or traverse.

```
STORE TEXT *name*
lines of text
END TEXT
```

This command is a multiple line command which stores textual information referencing it by *name*. Since this is a multiple line command, the continuation character is not required to continue input. The lines of text may consist of up to 25 lines of 80-column text. The text input starts after the command STORE TEXT *name* and ends at the line before the END TEXT command. The text name may be up to eight characters in length.

! If the text name is the same as an alignment/traverse, the text is automatically associated with the alignment/traverse of the same name.

Example:

```
STORE TEXT *TEXTI*
LINE 1
LINE 2
LINE 3
LINE 4 ABCDEFGHUKL
END TEXT
```

Store Traverse Command

The Store Traverse command creates a traverse in the COGO model.

```
STORE TRAVERSE *name* (list of points/block of points)
```

This command stores the TRAVERSE *NAME* using points that have been previously defined. A LIST OF POINTS consists of a series of point numbers separated by blanks or commas. A BLOCK OF POINTS is defined as [POINT] n TO [POINT] m. The traverse name may be up to eight characters in length.

! The store traverse command will not accept curve points as input.

Examples:

STORE TRAVERSE *TRAVER 1 * 1 2 3 4
STORE TRAVERSE *TRAV2* 1 TO 4

Traverse Command (distance/direction)

The Traverse command computes and stores the coordinates of a traverse based on distance/direction pairs or angle right traverse survey data. A traverse is a sequence of points. Traverses may be open (the first point is not equal to last point) or closed (the first point is equal to last point).

```
TRAVERSE *name* [REDEFINE] [PRINT]
BEGIN AT [POINT] n
        .
        .
        .
[FROM [POINT] m] TO [POINT] p distance direction
        .
        .
        .
END AT (POINT) k [PRINT]
```

Depending on the type of the TRAVERSE command, it will create a traverse from distances and bearings, or from an angle right traverse. The TRAVERSE Command is a multiple line command consisting of a Traverse NAME, which BEGINs AT POINT n, a definition of a line for each course and an ENDs at POINT k. This command will compute and store a traverse and its points. This command creates open or closed traverses and will solve for up to 2 unknown distances or bearings. The REDEFINE option instructs COG0 to delete the traverse *name* before it is created. If desired the results of the computations may be output via the PRINT option.

Each course in the traverse is defined as a line segment which starts at the end of the previous course and ends at the TO point. The command line which defines the course may include the FROM point, and requires the TO point and the distance and direction. The DISTANCE and DIRECTION may be one of the following forms.

Distance

[DISTANCE] d.d
Where: D. D is the distance from one point to the other point of a course of the traverse.

[DISTANCE] (UNKNOWN/U) [APPROX d.d]
This option informs COGO to calculate the distance from the starting point of the course to the ending point of the course based on the definitions of other courses. The APPROXIMATE option of this phrase informs COGO of the approximate length of the course.

Direction

AZIMUTH ddd mm ss.ss
Where: DDD MM SS.SS are the degrees minutes and seconds of the azimuth.

[BEARING] (N/S) dd mmm ss.ss (E/W)
Where: DD MM SS.SS are the degrees minutes and seconds of the bearing.

[DEFLECTION] ddd mmm ss.ss
Where: DDD MM SS.SS are the degrees minutes and seconds of the deflection.

(UNKNOWN/U) APPROX dir

Where : This option informs COGO that you wish COGO to calculate the direction of the course. The 'approximate' option of this phrase allows you to inform COGO of the approximate direction of the course. The direction is specified using the AZIMUTH, BEARING or DEFLECTION.

! A direction angle without an identifier is assumed to be a deflection. A deflection may not occur on a course immediately following a course with an unknown direction.

In some cases of two unknowns in a traverse, there are two possible pairs of solutions. The APPROX option allows you to choose the solution; however, this option should be used on only one of the two unknowns. It should be noted that this option may not allow COGO to choose the desired solution when the approximate direction and one of the solutions are on either side of the north axis.

COGO calculates the solution of one unknown in the traverse using two equations with one unknown. If the solutions differ by more than 1/2 %, a warning is printed. This indicates that the distance bearing descriptions are not consistent.

Examples:

```
TRAVERSE *EXAMPLEI* PRINT
BEGIN AT I 0 0
FROM 1 TO 2 5280 N 90 W
FROM 2 TO 3 440 -90 0 0
FROM 3 TO 4 100 -15 0 0
END AT 4

TRAV *EXAMPLE2* REDEFINE
BEGIN AT 101
TO 102 418.5 N33 40 30E
TO 103 508.8 12 30 0
TO 104 UNKNOWN -14 0 0
TO 105 600.1 UNKNOWN APPROX -5 30 0
END AT 105 PRINT

SOLUTIONS OF UNKNOWNS
COURSE        3     2093.398
            (ALT)  3291.098

            COURSE 4 N 28 28 33.8 E
            (ALT)  S 35 52 26.2 W


            CLOSURE
            DISTANCE     4203.522 S 32 56 48.2 W
```

Traverse Command (angle right)

This type of the Traverse command computes and stores the coordinates of angle right traverse survey data. A traverse is a sequence of points. Traverses may be open (the first point is not equal to last point) or closed (the first point is equal to last point).

```
TRAVERSE *name* [REDEFINE] [PRINT]
BEGIN WITH [POINT] b  BA (azimuth/bearing/[PNT] a)
.
.
.
AT [PNT] m [height] TO [PNT] n [height] [*PID*]
      BACK distances(s) [zenith(s)]
      FORWARD distance(s) [zenith(s)]
      Angle angle(s)
.
.
.
END
```

The traverse NAME can be REDEFINED and PRINTED, and can be OUTPUT to CMM file NAME.LSA. The traverse begins with known POINT B which is at a back AZIMUTH (BA) or BEARING from a known station. The format for the direction is

azimuth

-or-

BEAR bearing

If desired, POINT A may be given instead of the azimuth or bearing and COGO will use the azimuth from POINT B to POINT A for the direction value.

The legs of the traverse are defined with the instrument AT POINT M at HEIGHT, and the angle(s) turned clockwise from the azimuth of the previous leg to POINT N with the target at HEIGHT. If elevation calculations are to be performed, POINT M must have an elevation. The back distance(s) are required and entered one per line in the form

BACK distance [zenith]

The back distances will be meaned with the mean of the previous leg's forward distances. The forward distance(s) are required and entered one per line in the form

FORWARD distance [zenith]

If the zenith is not entered for the distances it is assumed to be 90 degrees (horizontal). The angles may be the raw data or the mean angles calculated from the raw data. The angles are entered one per line in the form

ANGLE angle

The distance(s), angle(s), and (optionally) zenith(s) are used to calculate the coordinates of point N. In addition, a PID (point ID), and ATTRibutes may be entered for POINT N. If the PID for POINT N is present the first 16 characters of the PID will be used as the station name for CMM, otherwise the point number for POINT N will be used as the station name.

The traverse ends with the command word END.

! The accuracy of the weighted least squares adjustment will be based on the number of angle/distance values entered for each leg in the traverse. The fewer the number of values entered, the less accurate the results.

Example:

The following is an example (from the 1992 edition of the Survey Manual pages 3-05-11 AND 3-05-12. Note: The initial back azimuth and distance are fictional).

```
TRAVERSE *RIGHT* REDEFINE PRINT OUTPUT *CMM*
BEGIN WITH 2  BA 00 00 04.0
    AT PNT 2 4.58 TO PNT 3 5.51 *Type 2 monument*
      BACK    823.101
      BACK    823.110
      BACK    823.108
      BACK    823.109
      FORWARD 501.913  85 07 34
      FORWARD 501.911  85 07 36
      FORWARD 501.910  85 07 34
      FORWARD 501.906  85 07 27
      ANGLE 43 49 57
      ANGLE 43 49 56
      ANGLE 43 49 58
      ANGLE 43 49 56
      ANGLE 43 49 55
      ANGLE 43 49 56
      ANGLE 43 49 57
      ANGLE 43 49 55
END AT PNT 3
```

Chapter 16

An Example of a COGO Project

This is a simple project with some points, curves, a traverse, an alignment, and a parcel. This is fictitious project.

Here's the information for the project.

A detour has to be quickly designed and implemented due to a bridge closure. The project manager has established where he wants the detour to start, some rough alignment information, and that a temporary easement is necessary near the center of the alignment.

Alignment Information:

Starting Point: 49524.472 N 47866.799 E

PI for 1st curve 433.65' at N 75° 27' 17.559" E from starting point

PI for 2nd curve 387.303' at N 52° 31' 39.504" E from 1st PI

PI for 3rd curve 387.303 at N 75° 27' 17.660" E from 2nd PI

PI for 4th curve 387.303' at S 81° 37' 3.766" E

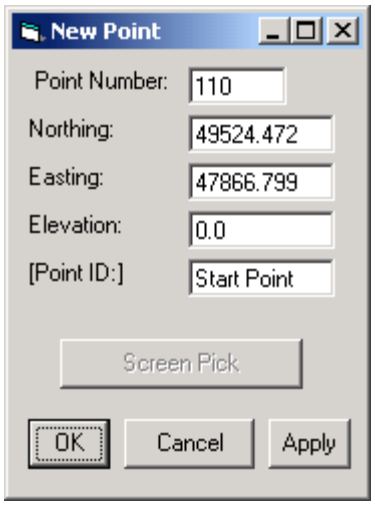
Ending point: 433.65' at N 75° 27' 17.559" E

All curves should have a degree of curvature of 6°

Temporary Easement Information:

Half-way through the detour a small parcel is needed for a temporary easement. The starting point for the project is the 1st tie point to the easement. The next tie point is 559.0169944' at N 45 E, then 259.5085 at N 30 E, then 375.0' at N 90 E. The parcel starts at the PI of the 3rd curve, follows the centerline offset 10' to the left of the centerline to the PI for the 2nd curve, and ends at the PI of the 3rd curve.

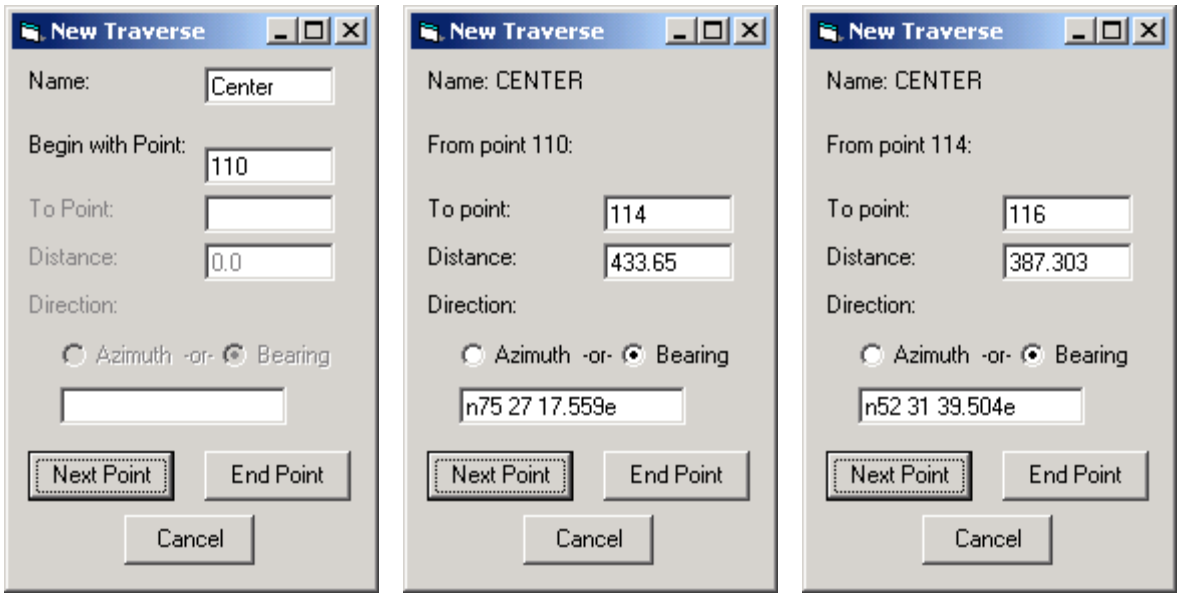
Starting the Project

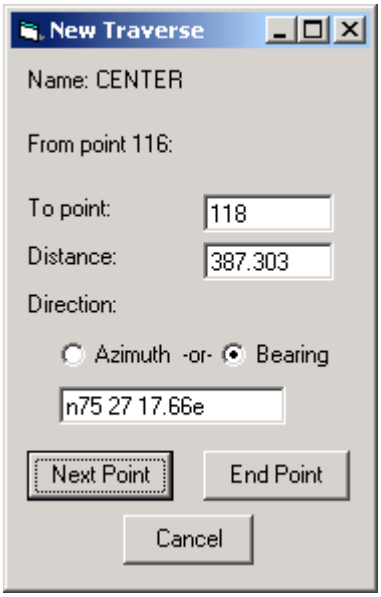


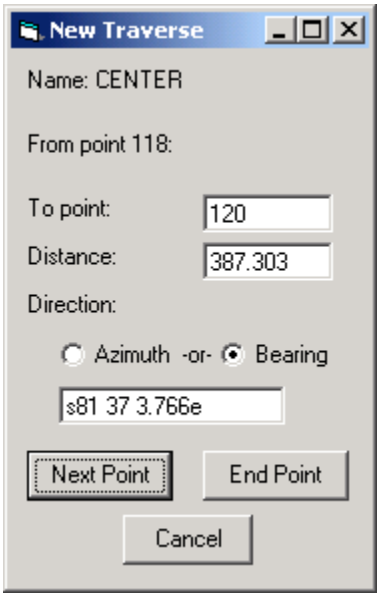
The first item needs to be the start of the alignment. The Store Point function is used to create the point.

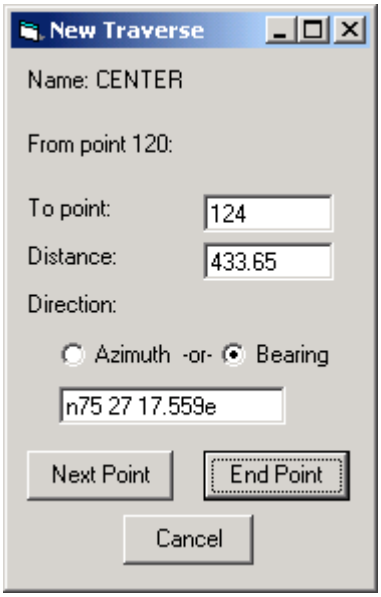
Creating the PIs for the Alignment

In this case there are two ways to create the PIs for the alignment. The first is to use the Locate Distance Direction function and the second is to use the Traverse function. Let's do the traverse.

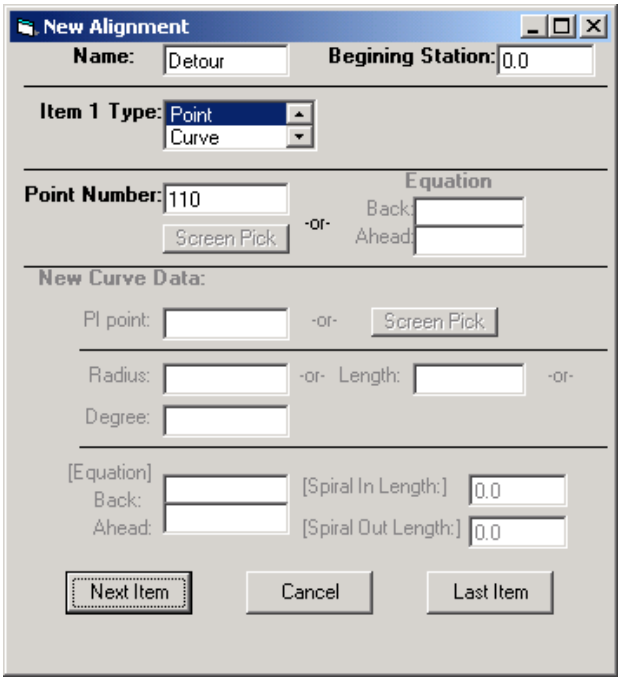


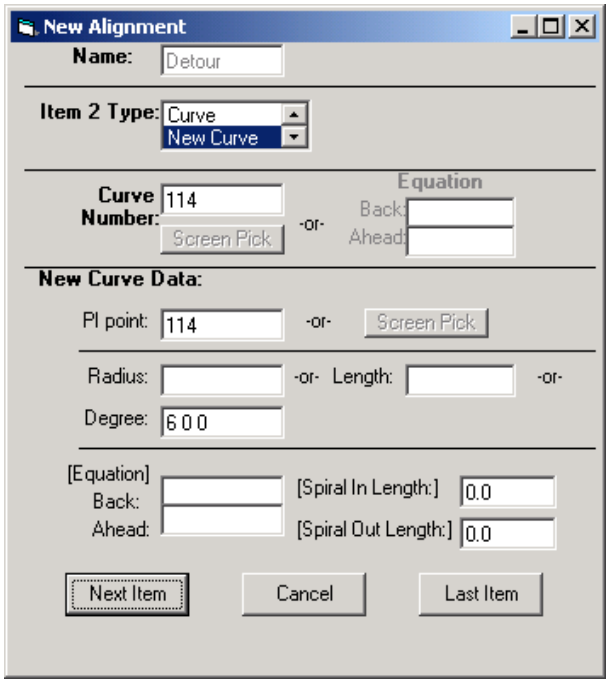


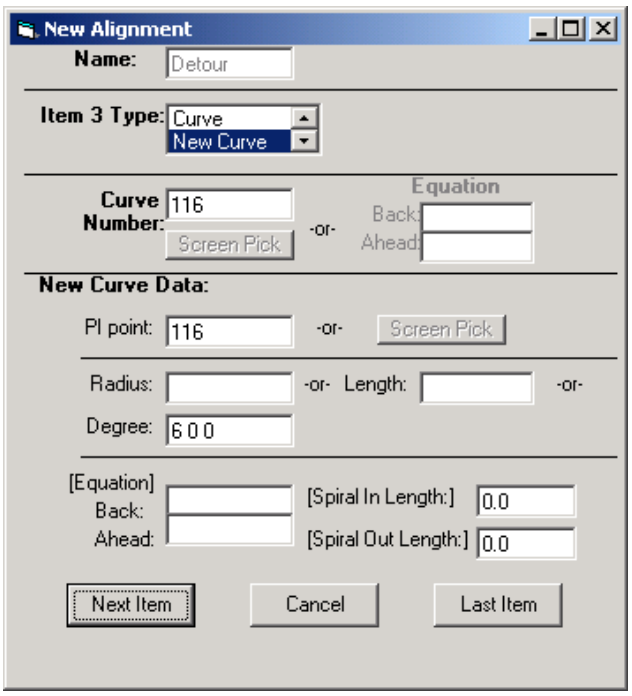


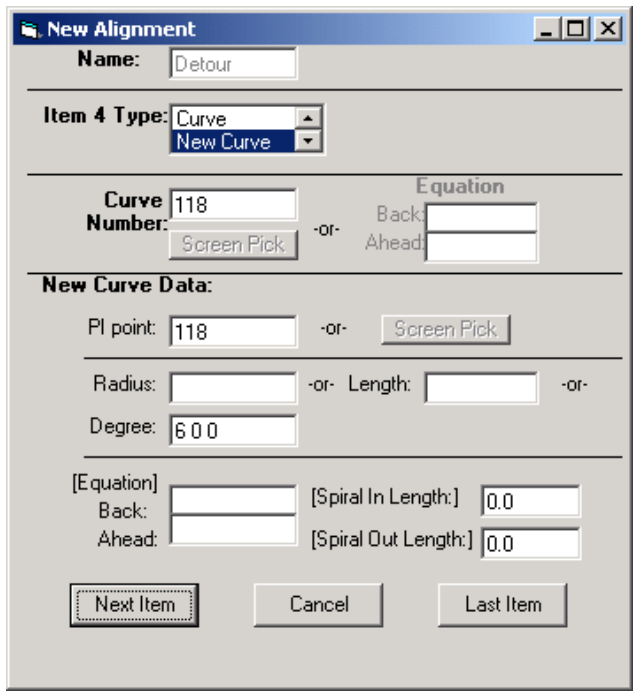


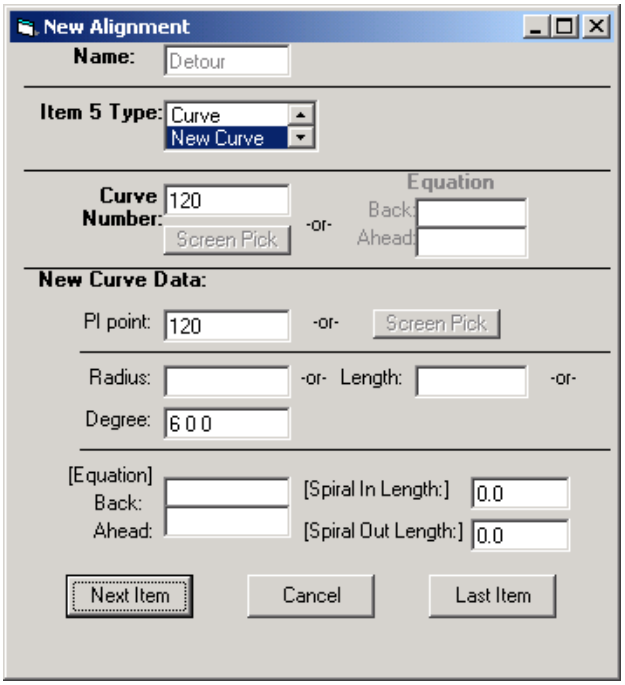
All the PIs are created, so here's the alignment.

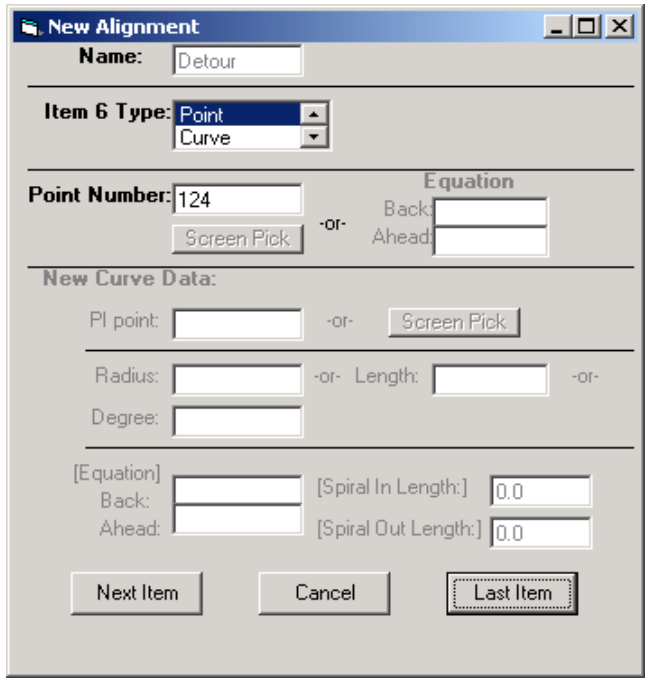






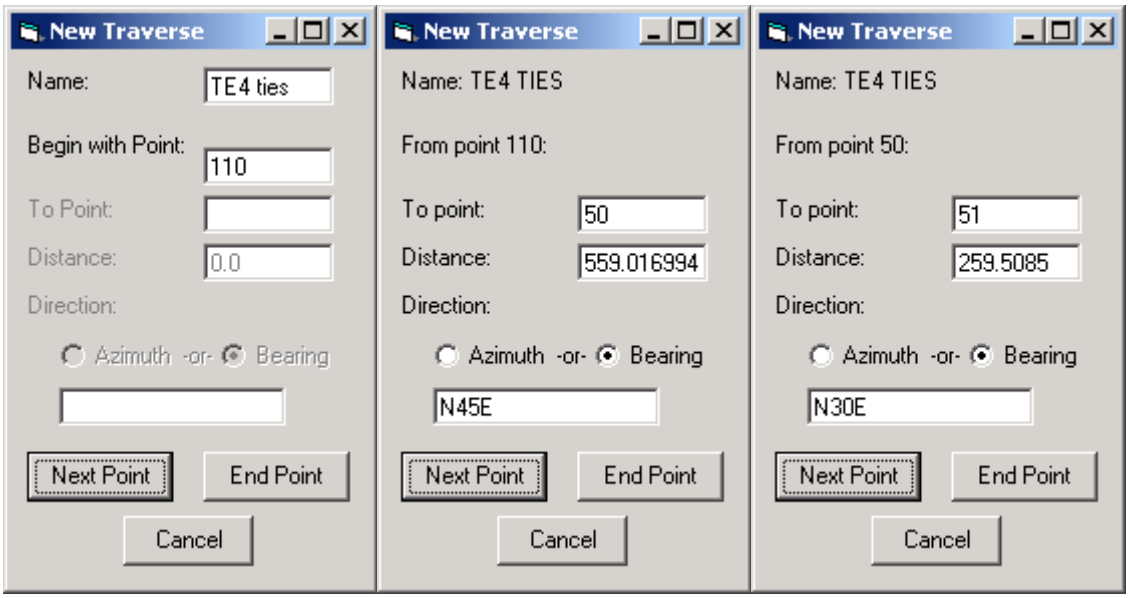


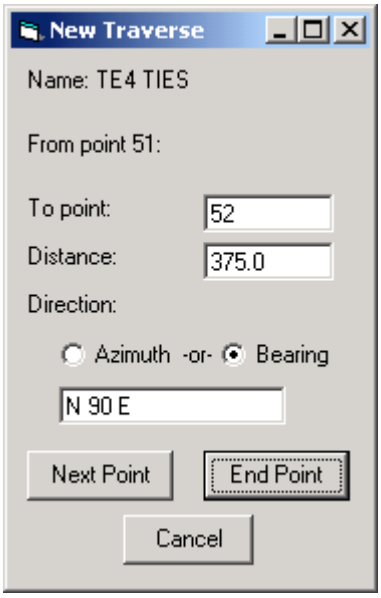




Now for the Parcel

Use the Traverse function to establish the tie points for the parcel.





With the tie points created, on to the parcel.

New Parcel

Name: TE4

Subtract area? Yes No
Legal Output? Yes No

Reset Total Area

1st Item
Type: Tie Point / Point
Number: 110 or
Screen Pick

Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: or
Screen Pick
Radius:

Next Item | Cancel | End Item

New Parcel

Name: TE4

Subtract area? Yes No
Legal Output? Yes No

Reset Total Area

Next Item:
Type: Tie Point / Point
Number: 50 or
Screen Pick

Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: or
Screen Pick
Radius:

Next Item | Cancel | End Item

New Parcel

Name: TE4

Subtract area? Yes No
Legal Output? Yes No

Reset Total Area

Next Item:
Type: Tie Point / Point
Number: 51 or
Screen Pick

Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: or
Screen Pick
Radius:

Next Item | Cancel | End Item

New Parcel

Name: TE4

Subtract area? Yes No
Legal Output? Yes No

Reset Total Area

Next Item:
Type: Tie Point / Point
Number: 52 or
Screen Pick

Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: or
Screen Pick
Radius:

Next Item | Cancel | End Item

New Parcel

Name: TE4

Subtract area? Yes No
Legal Output? Yes No

Reset Total Area

Next Item:
Type: Tie Point / Point
Number: 118 or
Screen Pick

Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: or
Screen Pick
Radius:

Next Item | Cancel | End Item

New Parcel

Name: TE4

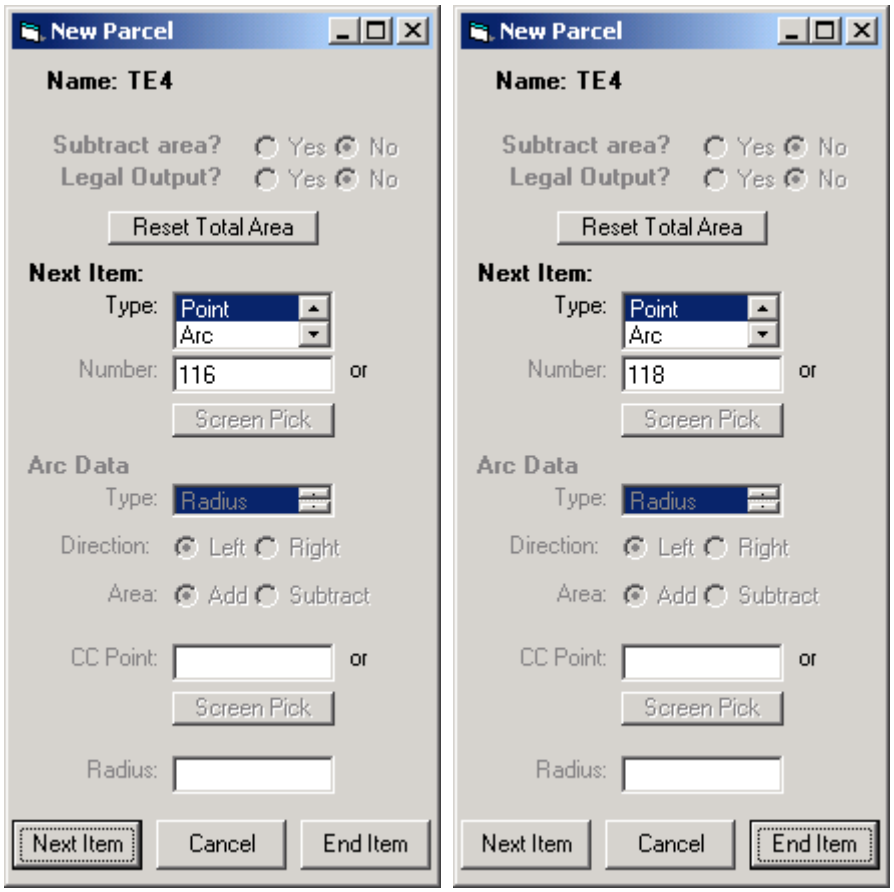
Subtract area? Yes No
Legal Output? Yes No

Reset Total Area

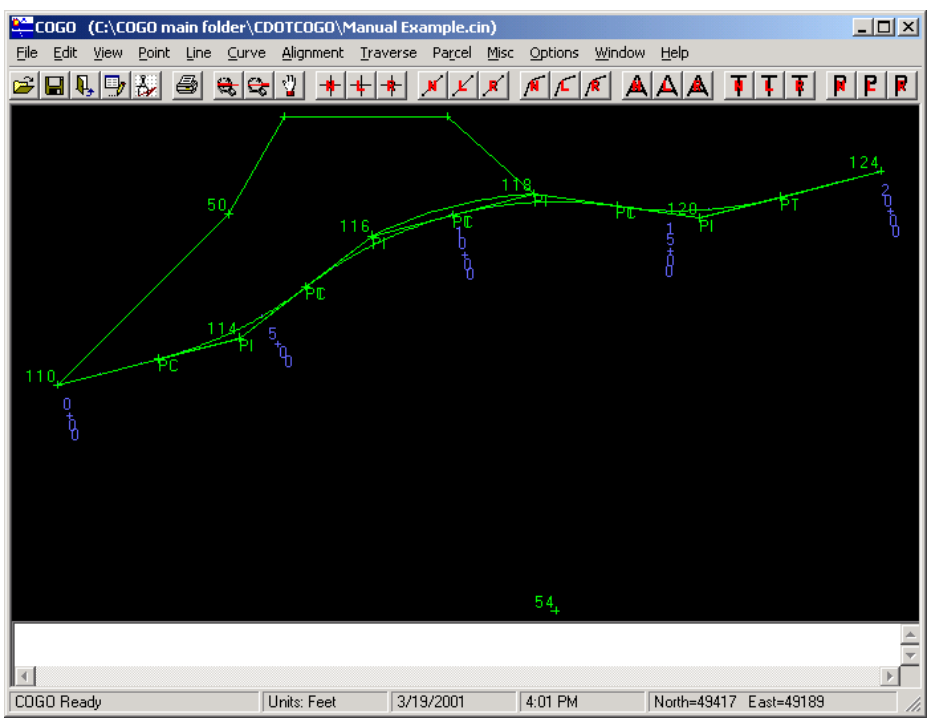
Next Item:
Type: Point / Arc
Number: or
Screen Pick

Arc Data
Type: Radius
Direction: Left Right
Area: Add Subtract
CC Point: 54 or
Screen Pick
Radius: 964.93

Next Item | Cancel | End Item



This is the result.



Now that the geometry model is built, what do you do next? The best thing to do is make sure you have saved all your work in the form of an input file. The input file can be adjusted as necessary to finish the project, and re-run as many times as necessary. Can you imagine having to fill out the forms again every time a change comes through? To save all of this work in the form of an input file, you need to find the file COGO.LOG. Typically, this will reside in the C:\Program Files\CDOTCOGO folder. This file contains all the functions you performed in the form of COGO commands, even all the mistakes. Copy this file to your work folder and rename it to a .CIN file. You now have your input file for later use. Here's the COGO.LOG for this sample project (comments are added for clarity) and the COGO.COT (output file).

COGO.LOG

```
* Sample project for the COGO manual

* Store the starting point for the project
STORE POINT 110 49524.472 47866.799

* Create the PIs for the detour alignment
Trav *Center* redefine print
beg at 110
to 114 433.65 N75 27 17.559E
to 116 387.303 N52 31 39.504E
to 118 387.303 N75 27 17.66E
to 120 387.303 S81 37 3.766E
to 124 433.65 N75 27 17.559E
end at 124

* Create the detour alignment
ALIGNMENT *Detour* REDEFINE PRINT
BEGIN AT 110 STA 0.0
CURVE 114 PI 114 DEGREE 6 0 0.0
CURVE 116 PI 116 DEGREE 6 0 0.0
CURVE 118 PI 118 DEGREE 6 0 0.0
CURVE 120 PI 120 DEGREE 6 0 0.0
END AT 124

* Create the ties for the temporary easement
TRAVERSE *TE4 Ties* REDEFINE
BEGIN AT POINT 110
TO 50 559.0169944 N45E
TO 51 259.5085 N30E
TO 52 375 N90E
END AT 52

* Create the temporary easement
PARCEL *TE4*  REDEFINE
TIE POINT 110
TIE POINT 50
TIE POINT 51
TIE POINT 52
```

```
BEGIN AT POINT 118
ARC left RADIUS 964.93 PLUS CC 54
POINT 116
END AT POINT 118
```

COGO.COT

```
 * Sample project for the COGO manual

 * Store the starting point for the project
 STORE POINT 110 49524.472 47866.799

 * Create the PIs for the detour alignment

 Trav *Center* redefine print
 beg at 110
 to 114 433.65 N75 27 17.559E
 to 116 387.303 N52 31 39.504E
 to 118 387.303 N75 27 17.66E
 to 120 387.303 S81 37 3.766E
 to 124 433.65 N75 27 17.559E
 end at 124

     Closure
     Distance       1968.015     S 75 27 17.7 W

 TRAVERSE    CENTER

 PNT   110    49524.472 N     47866.799 E

            433.650'      N 75 27 17.6 E

 PNT   114    49633.380 N     48286.551 E

            387.303'      N 52 31 39.5 E

 PNT   116    49869.007 N     48593.932 E

            387.303'      N 75 27 17.7 E

 PNT   118    49966.275 N     48968.822 E

            387.303'      S 81 37  3.8 E

 PNT   120    49909.815 N     49351.988 E

            433.650'      N 75 27 17.6 E

 PNT   124    50018.723 N     49771.740 E
 END TRAVERSE    CENTER
```

```
* Create the detour alignment

ALIGNMENT *Detour* REDEFINE PRINT
BEGIN AT 110 STA 0.0
CURVE 114 PI 114 DEGREE 6 0 0.0
CURVE 116 PI 116 DEGREE 6 0 0.0
CURVE 118 PI 118 DEGREE 6 0 0.0
CURVE 120 PI 120 DEGREE 6 0 0.0
END AT 124
  Stationing summary for alignment   DETOUR
     0 equations

ALIGNMENT   DETOUR

PNT   110    49524.472 N     47866.799 E           0+00.00'

           433.650'      N 75 27 17.6 E

CURVE  114    Type: Circular Curve        Left
   Circular Curve Data
      Degree   6  0  0.0     Deflection  22 55 38.1    Delta  22 55 38.1
      Radius     954.930'    Length         382.121'
   Tangent Back Data (PC/TS to PI)
          N 75 27 17.6 E     Length         193.651'
   Tangent Ahead Data (PI to PT/ST)
          N 52 31 39.5 E     Length         193.651'
   PC 114        49584.746 N         48099.106 E       2+40.00'
   PI 114        49633.380 N         48286.551 E       4+33.65'
   CC 114        50509.070 N         47859.283 E
   PT 114        49751.193 N         48440.241 E       6+22.12'

           387.303'      N 52 31 39.5 E

CURVE  116    Type: Circular Curve        Right
   Circular Curve Data
      Degree   6  0  0.0     Deflection  22 55 38.2    Delta  22 55 38.2
      Radius     954.930'    Length         382.121'
   Tangent Back Data (PC/TS to PI)
          N 52 31 39.5 E     Length         193.652'
   Tangent Ahead Data (PI to PT/ST)
          N 75 27 17.7 E     Length         193.652'
   PC 116        49751.193 N         48440.241 E       6+22.12'
   PI 116        49869.007 N         48593.932 E       8+15.77'
   CC 116        48993.316 N         49021.200 E
   PT 116        49917.641 N         48781.377 E      10+04.24'

           387.303'      N 75 27 17.7 E

***** Warning: Overlap between curve and curve.


CURVE  118    Type: Circular Curve        Right
```

```
   Circular Curve Data
      Degree   6  0  0.0     Deflection  22 55 38.6    Delta  22 55 38.6
      Radius     954.930'    Length         382.123'
   Tangent Back Data (PC/TS to PI)
         N 75 27 17.7 E     Length         193.653'
   Tangent Ahead Data (PI to PT/ST)
         S 81 37  3.8 E     Length         193.653'
   PC 118        49917.641 N         48781.376 E      10+04.24'
   PI 118        49966.275 N         48968.822 E      11+97.89'
   CC 118        48993.316 N         49021.199 E
   PT 118        49938.045 N         49160.406 E      13+86.36'

           387.303'      S 81 37  3.8 E

***** Warning: Overlap between curve and curve.


CURVE  120    Type: Circular Curve        Left
   Circular Curve Data
      Degree   6  0  0.0     Deflection  22 55 38.7    Delta  22 55 38.7
      Radius     954.930'    Length         382.123'
   Tangent Back Data (PC/TS to PI)
         S 81 37  3.8 E     Length         193.653'
   Tangent Ahead Data (PI to PT/ST)
         N 75 27 17.6 E     Length         193.653'
   PC 120        49938.045 N         49160.404 E      13+86.37'
   PI 120        49909.815 N         49351.988 E      15+80.02'
   CC 120        50882.774 N         49299.611 E
   PT 120        49958.449 N         49539.434 E      17+68.49'

           433.650'      N 75 27 17.6 E

PNT   124    50018.723 N     49771.740 E          20+08.49'
END ALIGNMENT   DETOUR



TRAVERSE *TE4 Ties* REDEFINE PRINT
BEGIN AT POINT 110
TO 50 559.0169944 N45E
TO 51 259.5085 N30E
TO 52 375 N90E
END AT 52

    Closure
    Distance       1092.933     S 55 26 15.1 W

TRAVERSE    TE4 TIES

PNT   110    49524.472 N     47866.799 E

           559.017'      N 45  0  0.0 E
```

```
PNT    50    49919.757 N     48262.084 E

           259.509'      N 30  0  0.0 E

PNT    51    50144.498 N     48391.838 E

           375.000'      N 90  0  0.0 E

PNT    52    50144.498 N     48766.838 E
END TRAVERSE    TE4 TIES



PARCEL *TE4*  REDEFINE PRINT
TIE POINT 110
TIE POINT 50
TIE POINT 51
TIE POINT 52
BEGIN AT POINT 118
ARC left RADIUS 964.93 PLUS CC 54
POINT 116
END AT POINT 118

-------------------
| PARCEL: TE4     |
-------------------

Begins at point   110         49524.472 N       47866.799 E

Thence N. 45^ 00' 00" E., a distance of 170.389 meters (559.02

To tie point       50         49919.757 N       48262.084 E

Thence N. 30^ 00' 00" E., a distance of 79.098 meters (259.51 feet);

To tie point       51         50144.498 N       48391.838 E

Thence N. 90^ 00' 00" E., a distance of 114.300 meters (375.00

To tie point       52         50144.498 N       48766.838 E

Thence S. 48^ 34' 34" E., a distance of 82.105 meters (269.37 feet);

To point     118         49966.275 N       48968.822 E

1. Thence along the arc of a curve to the left having a radius of
 294.111 meters (964.93 feet), a distance of 118.857 meters
 (389.95 feet), (the chord of said arc bears  S.  75^ 27' 18" W.,
 a distance of 118.050 meters (387.30 feet) );

 The area of the arc being        5079 SQFT, (      0.117 Acres)
 Has been Added to the total area.
```

To point 116 49869.007 N 48593.932 E

2. Thence N. 75^ 27' 18" E., a distance of 118.050 meters (387.30 feet), more or less, to the TRUE POINT OF BEGINNING.

To point 118 49966.275 N 48968.822 E

The above described parcel contains 0.047 hectares/471.9 square meters (0.117 acres/5079 square feet), more or less.

Total area 471.89 SQM (0.04719 Hectares)
5079.33 SQF (0.12 Acres)

```
----------------------
| END PARCEL TE4     |
----------------------
```

Appendix A – COGO command word abbreviations

Command	Abbreviation(s)
ALDJUST	ADJ
ALIGNMENT	ALI, ALIGN
ANGLE	ANG
ANNOTATE	ANNO
APPROXIMATE	APPROX
ATTRIBUTE	ATTR
AZIMUTH	AZI
BEARING	BEA,BEAR
BEGIN	BEG
BOUNDARY	BOU
CURVE	CUR
COMPASS	COMP
DEFLECTION	DEFL
DEGREE	DEG
DELETE	DEL
DESIGN	DES
DISTANCE	DIS
DIVIDE	DIV
EAST	E
ELEVATION	ELEV
EQUATION	EQ
FINISH	FIN
INTERSECT	INTER
LAYOUT	LAY
LENGTH	LENG
LINE	LIN
LOCATE	LOC
MESSAGE	MESS
MINUTE	MIN
NORTH	N
NUMBER	NUM
ORIGIN	ORI
PARCEL	PAR

POINT	PNT
PROJECT	PROJ
REDEFINE	REDEF
RESTATION	RESTA
REVERSED	REV
ROADCALC	ROAD
ROTATE	ROT
SCALE	SCA
SECOND	SEC
SEGMENT	SEG
SOUTH	S
STATION	STA
STORE	STO
SUBPROJECT	SUB
SUBTRACT	SUB
TABLES	TABL
TANGENT	TANG
TEMPLATE	TEMP
TERRAIN	TERR
TOLERANCE	TOL
TRAVERSE	TRAV
UNKNOWN	U,UNKN
WEST	W

Appendix B - COGO Error Messages

ERROR NO.	DESCRIPTION OF THE ERROR
101	W POINT UNDEFINED
102	LINE UNDEFINED
103	W LINE BEING REDEFINED
104	W CURVE BEING REDEFINED
105	CURVE UNDEFINED
106	W COGO OBJECT UNDEFINED OR OF WRONG TYPE
107	COGO OBJECT ALREADY DEFINED
108	W ALIGNITRAV/TEXT UNDEFINED
109	INPUT CHARACTER STRING BEGINS WITH ILLEGAL SYMBOL
110	INPUT NAME NOT DELIMITED CORRECTLY
111	INPUT NAME LONGER THAN 8 CHARACTERS
112	INPUT WORD UNKNOWN (PERHAPS MISSPELLED)
113	INPUT NUMBER ERROR
114	IMPROPER CONTINUATION OF INPUT
115	POINT NUMBER MISSING
116	NO DEFINITIONS ON STORE POINT
117	POINT BEING REDEFINED
118	FIRST WORD OF COMMAND NOT LEGITIMATE
119	SET METRIC INCORRECT
120	COMMAND SYNTAX ERROR
121	POINT NUMBER OUT OF RANGE
122	MORE THAN 500 ALIGNMENTS
124	N/S DELIMITER NOT CORRECT
125	NO DEGREE IN ATTEMPTED BEARING SPECIFICATION
126	E/W DELIMITER NOT CORRECT
127	LINE NUMBER MISSING OR OUT OF RANGE
131	CURVE NUMBER MISSING
132	UNRECOGNIZABLE CURVE DATA ITEM
134	CURVE BACK REFERENCE ITEM NOT CORRECT
135	CURVE AHEAD REFERENCE ITEM NOT CORRECT
136	CURVE CIRCULAR REFERENCE ITEM NOT CORRECT
137	CURVE SPIRAL ITEM NOT CORRECT
138	UNDEFINED POINT IN CURVE REFERENCE ITEM
139	CIRCULAR ITEM INCORRECT OR SPIRALS TOO LONG FOR DEFL
140	AHEAD DATA ITEM INCORRECT OR INCOMPLETE
142	PI UNDEFINED OR INSUFFICIENTLY DEFINED
143	CURVE NUMBER OUT OF RANGE
146	POINT NUMBER MISSING OR USE OF TO INCORRECT
147	NAME MISSING OR NOT CORRECT
149	CURVE POINTS NOT TS,SC,P1,CC,CS,ST,PC,PT
150	NO TS/ST ON NON-SPIRALED CURVES
151	NO PC/PT ON SPIRALED CURVES
155	IMPROPER USE OF TO IN ALIGNMENT LIST

156 ALIGNMENTS NESTED MORE THAN 7 DEEP
157 MORE THAN 40 INPUT ITEMS ON CONTINUATION LINE
160 NO BEGIN
161 UNIDENTIFIABLE ITEM - PROBABLY MISSING END
162 POINT DATA NOT CORRECT
163 EQUATION NOT CORRECT
167 MORE THAN 25 LINES OF TEXT - PERHAPS MISSING END
168 TEXT FILE BAD ON RETRIEVAL
178 ZERO LENGTH CURVE SPECIFICATION NOT CORRECT
180 ITEM IN DELETE LIST NOT CORRECT
183 MISSING POINT NUMBER OR DATA ON BEGIN OR END
184 POINTS OUT OF SEQUENCE IN TRAVERSE
186 COURSES WRONG - PROBABLY MISSING END
187 DEFLECTION GIVEN AFTER UNKNOWN
189 MORE THAN TWO UNKNOWNS
190 DISTANCE NOT CORRECT
192 AZ/BEARING/DEFLECTION NOT CORRECT
193 TRAVERSE DATA BAD ON RETRIEVAL
194 BEGIN OR END POINT UNDEFINED WITH UNKNOWNS
196 UNDEFINED POINT IS DISTANCE OR BEARING
201 DISTANCE MISSING OR INCORRECTLY GIVEN
202 OFFSET MISSING OR INCORRECTLY GIVEN
203 ANGLE MISSING OR INCORRECTLY GIVEN
204 NO SOLUTION TO ARC/ARC TIE
205 NO SOLUTION TO ARC/LINE TIE
206 APPROXIMATE ANGLE DOES NOT INDICATE SOLUTION
207 APPROXIMATION INCORRECTLY GIVEN
208 NEAR POINT GIVEN INCORRECTLY
209 NAME OR NUMBER MISSING OR INCORRECTLY GIVEN
210 FROM POINT NOT ON LINE
211 DISTANCE FROM POINT GOES OFF CURVE
212 FROM POINT NOT ON TRAVERSE
213 FROM POINT NOT ON ALIGNMENT
215 NO TANGENT SOLUTION
216 POINT TO PROJECT MISSING OR INCORRECTLY GIVEN
218 NO PROJECTION
219 FIRST OR LAST ITEM AN ARC
222 POINT DISTANCE PAIR INCORRECT
225 POINT IN USE OPTION INCORRECTLY GIVEN
226 LINE SEGMENT NOT CORRECT
228 NO INTERSECTION IN OFFSETS - BAD DATA
233 NO SOLUTION
234 W MORE THAN 7 SOLUTIONS; FIRST 7 KEPT
236 CC DOES NOT PROJECT
238 FILE NAME BAD
239 FILE ACTION NOT CORRECT
240 ERROR IN LIST
241 BAD CURVE POINT
242 MISSING POINT BETWEEN ARCS
243 W UNEQUAL RADII
244 END NOT BEGINNING POINT
245 W TOO MANY SOLUTIONS FROM ITERATION
246 LINE NUMBER NOT FOUND IN INPUT FILE
249 BAD STATION VALUE

250 BAD ALIGNMENT
252 INCORRECT BLOCK OF POINTS SPECIFIED
253 CC POINT BAD
254 STATION NOT FOUND
258 STATION COMMAND STORE OPTION ERROR
261 LAYOUT POINT ERROR
266 UNABLE TO OPEN SAVED TABLE
269 IS NOT GREATER/EQUAL TO M
273 TOO MANY EQUATIONS, LIMIT = 10
274 EQ ITEM NOT ON ALIGNMENT
275 SPIRAL LENGTH IGNORED - NOT CONSISTENT
276 TOO MANY COURSES IN TRAVERSE TO BE ADJUSTED
277 KEYWORD FOR USE MISSING
280 EVEN SPECIFICATION BAD
300 END OF FILE ENCOUNTERED ON READ
302 TEMP. FILE ERROR
304 DIAMETER LESS THAN CHORD LENGTH
305 TOO MANY ITEMS IN DISPLAY LIST

Appendix C - COGO Command Quick Reference

Command	Description [Examples]
ADJUST	Adjusts a traverse. [ADJUST TRAV *PROJ* COMPASS PRINT TOL 0.0]
ALIGNMENT	Creates a COGO alignment. [ALI *PROJ* REDEFINE PRINT BEG AT PNT 110 PNT 115 EQ 21+05.53 21+08.52 CUR 3 EQ PT 23+83.22 END AT PNT 120]
ANGLE	Calculates the angle between 3 points. [ANGLE MINUS AT 20 FROM 101 TO 107]
AREA	Computes the area of a closed figure. [AREA 100 102 ARC 103 PLUS 109 ARC LEFT RADIUS 1908 PLUS 117]
ARCHIVE	Archives the COGO model as an input file. [ARCHIVE TABLE]
AZIMUTH	Calculates the azimuth and distance between points. [AZIMUTH PNT I PNT 2]
BEARING	Calculates the bearing and distance between points. [BEARING PNT 15 PNT 20]
DELETE	Removes a COGO item from the COGO model. [DELETE PNT 1 TO 20]
DISTANCE	Calculates the distance and bearing between points. [DIST 410 420]

DIVIDE

Divides a line, line segment or curve into smaller segments.
[DIVIDE LINE 20 100 PNT 2 120.3 PNT 3 245.73 PNT 4 100.2]
[DIVIDE LIN SEG PNT 25 TO PNT 30 PNT 26 PNT 27 PNT 28]
[DIVIDE CUR 3 PNT 100 PNT 101 PNT 102]

EJECT

Causes the printer to form feed and prints **a** heading.
[EJECT]

FINISH

Terminates a COGO run normally.
[FINI]

LAYOUT

Produces a staking layout of **a** COGO item.
[LAYOUT TIES PNT 101 TO ALI -PROJ*]
[LAYOUT CUR 5 TRANSIT AT POINT 101 EVEN 25.01 [LAYOUT ALI *SURV* EVEN 25]

LIST

Lists COGO items
[LIST PNT ALL]
[LIST ALL]
[LIST ALI *PROJ*]

LOCATE

Calculates the coordinates of a point based on given information.

GIVEN: A distance and direction from a point
[LOCATE PNT 10 FROM PNT 20 1842.67 N15E]

GIVEN: A distance and deflection from the end of a line segment.
[LOCATE 15 FROM 32 TO 33 512.36 DEFL 15 32 10.4]

GIVEN: A distance and angle from the beginning of a line segment.
[LOCATE 22 FROM 20 378.24 187 02 13.3 FROM 21]

GIVEN: A station on an alignment.
[LOCATE PNT 201 ON ALI *RAMPE* AT STA 144+15.2]

GIVEN: An offset distance from a point on a COGO item.
[LOCATE PNT 32 OFFSET FROM PNT 101 ON ALI *PROJ*]

GIVEN: A curve and offsets to the left and right of the curve.
[LOCATE PNT 10 TO 13 OFFSET 30 LEFT 32 RIGHT FROM CURVE 102]

GIVEN: A point and **a** COGO item that the point is projected onto.
[LOCATE PNT 35 PROJECT PNT 101 ON ALI *MAINWB*]

GIVEN: Two COGO items that intersect.
[LOCATE PNT 82 INTERSECT LIN SEG 3 TO 4 WITH ALI *SURV*]

GIVEN: Two points and deflections from the line formed by the two points.
[LOCATE PNT 65 FORSECT PNT 82 ANGLE MINUS 34 20 01.2 AND PNT 201 ANGLE 22 00 01.3]

GIVEN: Two distances from different points.
[LOCATE PNT 101 TIE 10 DIST 741.0 TIE 11 DIST 241.3 APPROX N30W]

GIVEN: A distance from one point, a second point and an angle turned from the line between the first and second point.
[LOCATE 901 TIE PNT I DIST 456.2 TIE PNT 2 ANGLE 30 00 01.01

GIVEN: A direction and a curve. (Locates the point where a line of direction is tangent to the curve.)
[LOCATE PNT 18 ON CUR 15 TANGENT BEAR N33 49 01.4W]

GIVEN: A point near a curve and the curve. (Locates the point where a line from the point to the curve is tangent.)
[LOCATE PNT 24 ON CUR 120 TANGENT THRU PNT 871

PARCEL

Creates a COGO parcel.
[PARCEL *TE10OA- SUBACCOUNT 09000 PRINT TIE POINT 10
BEGIN AT 101
ARC RIGHT RADIUS 824.5 CC 102
PNT 103
ARC 100 LEFT PLUS
PNT 104
END AT 105]

RESET AREA

Sets the COGO area register to a value
[RESET AREA 0.0]

RESTATION

Restations a COGO alignment.
[RESTATION ALI *PROJ* BEG AT STA 100+00.0 PNT 20 EQ 132+10.4]

RESTORE

Restores a saved COGO model.
[RESTORE TABLE *87042Cl*]

ROTATE

Rotates the COGO model about a point.
[ROTATE TABLES BY 00 *05* 15.2 ORIGIN AT 1 10]

SAVE

Saves a COGO model.
[SAVE TABLE *87042Cl*]

SET

Sets input/output files, date, log file, MOSS output, Roadcalc output, SDR data collector output and protection.
[SET FILE INPUT *COGOIN*]
[SET FILE OUTPUT *COGOOUT*]
[SET DATE *91/02/26*]
[SET LOG FILE ON]

[SET MOSS FILE OUTPUT ALI PROI END]
[SET ROADCALC FILE OUTPUT SUB 010 ALI *WB ML*]
[SET SDR FILE OUTPUT *SDRFILE* ALI *SURV*]
[SET NO REDEFINE]

STORE

Stores a COGO item in the COGO model.
[STORE PNT I 10000 10000 114+01.0 5280.1 *Control point*]
[STORE LIN 3 THRU PNT 15 N71 57 82.3E]
[STORE CUR 5 PC 51 PT 53 CC 52]
[STORE SPIRAL 2 PB 10 PI 11 SL 190.0 PA 14 SDCB 0 0 0 SDCA 2 45 0.01
[STORE ALI *PROJ* 101 102 CUR 15 103 CUR 17 110]
[STORE TRAV *SURV* 100 101 102 103 104 105]
[STORE TEXT *PROJ*
LINE 1
LINE 2
LINE 3
END TEXT]

STATION

Calculate the coordinates on or offset from an alignment.
[STATION ALI *PROJ* EVEN 25.01
[STATION OFFSET ALI *RAMPA* FROM *CENTER* EVEN 25.01

TRANSLATE

Translates all coordinates in the COGO model by an indicated distance and direction. [ROTATE TABLES
TRANSLATE TO PNT 10 ORIGIN AT 11

TRAVERSE

Creates a COGO traverse item.
TRAVERSE *SURVI* REDEFINE PRINT
BEGIN AT 101
TO 102 490.12 N15 83 10.2E
TO 103 1024.71 22 18 9.2
TO 105 912.01 UNKNOWN APPROX 15 20 8
END AT 101]

Index

Alignment
- command, 82
- creating, 52
- defined, 7
- equation, 52, 83, 85
- layout command, 93
- offset, 64
- station, 52
- storing, 51

Angle
- command, 86

Archive, 74, 86

AutoCAD, 66, 67, 81
- DXF, 66
- functions, 78
- layer names, 67
- macros, 17
- Screen Pick, 79

Azimuth
- command, 88
- definition, 21
- use, 22

Bearing
- command, 88
- definition, 22
- use, 23

Colorado DOT web site, 12

Curve
- broken back, compound, 4
- complex, 6
- compound, 4
- creating in an alignment, 54
- divide command, 91
- dividing, 45
- layout command, 93
- menu, 42
- reversing, 4
- simple, 3
- storing, 43

Curve point, 7, 30, 84

Distance
- command, 89
- use, 21

Eject
- command, 91

Entering
- characters, 20
- directions, 21
- distances, 21
- numbers, 19
- stations, 24

Equation, 83, 85, 116

Executing COGO
- in AutoCAD, 17
- in Windows, 16

File
- archive, 74
- input, 71
- log, 72
- output, 73
- Saved Tables, 73

Installation Instructions, 10

Line
- **best fit, 41**
- **definition, 2**
- **divide, 37**
- **divide command, 90**
- **dividing, 40**
- **menu, 38**
- **segment, 2, 39**
- definition, 2
- temporary, 3
- temporary, 39
- **storing, 39**

Misc
- **menu, 62**

MX, 66
- **Halign, 68**

Parcel
- **command, 109**
- **creating, 55**
- **defined, 9**
- **menu, 55**

PICS, 66

Point
- **copying, 26**
- **definition, 2**
- **deleting, 61**
- **listing, 59**
- **locate**
- 2 distances/2 points, 34
- angles from a line segment, 34
- at tangent to curve, 36
- curve tangency from a point, 36
- distance from point/point on a line, 35
- distance/direction, 27
- intersection, 33
- Line segment/direction, 27
- offset, 30
- offset from curve, 30
- projecting, 31
- station, 29
- **report, 60**
- **storing, 26**

Point of Tangency, 3, 36

Radians, 3

Ray, 2

RoadCalc, 66, 70, 123

Rotate/translate, 65

SDR, 68

Set
- **color, 77**
- **date, 118**
- **units, 64**

Total Area Register, 64

Transit, 93

Transition spiral, 5, 6

Traverse
- **adjusting, 50**
- **angle right, 49**
- **defined, 8**
- **distance/direction pairs, 47**
- **Least squares adjustment, 50, 81**
- **menu, 46**
- **store, 46**

Zenith angle
- **definition, 23**