Trimble Civil Engineering and Construction

Understanding and Using Linestrings in Trimble Business Center

For version 5.10, May 2019

Contents

Chapter 1 | Understanding Linestings

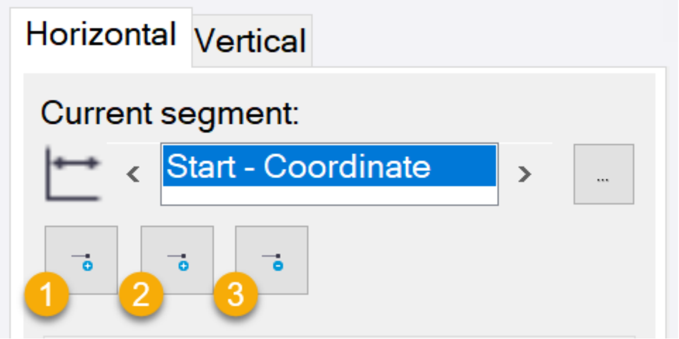
The best way to learn is by doing, so let's jump right in with a few common scenarios you can try. You will need a license of any type to enable the **Create Linestring** command.

Exercises to get you started

Using Undo with Linestrings

This is something to understand before you begin: If you undo a segment as you are creating a linestring, to resume creating segments, you will need to click the **Insert after last segment** icon in the **Edit Linestring** pane to add the next segment.

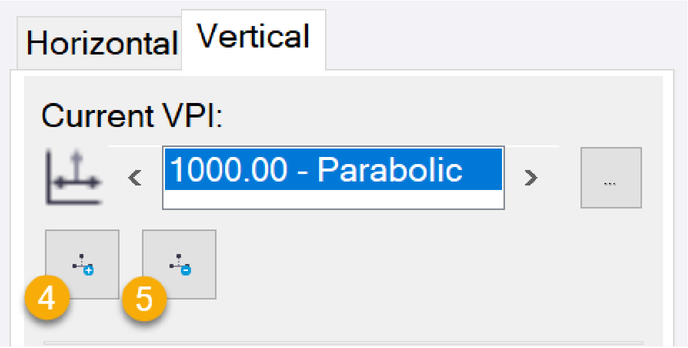
These other options are available on the **Horizontal** tab when you need to edit a linestring:



Current segment: Start - Coordinate

1. **Insert before current segment**
2. **Insert after last segment**
3. **Remove current segment**

Similarly, these options are available on the **Vertical** tab:

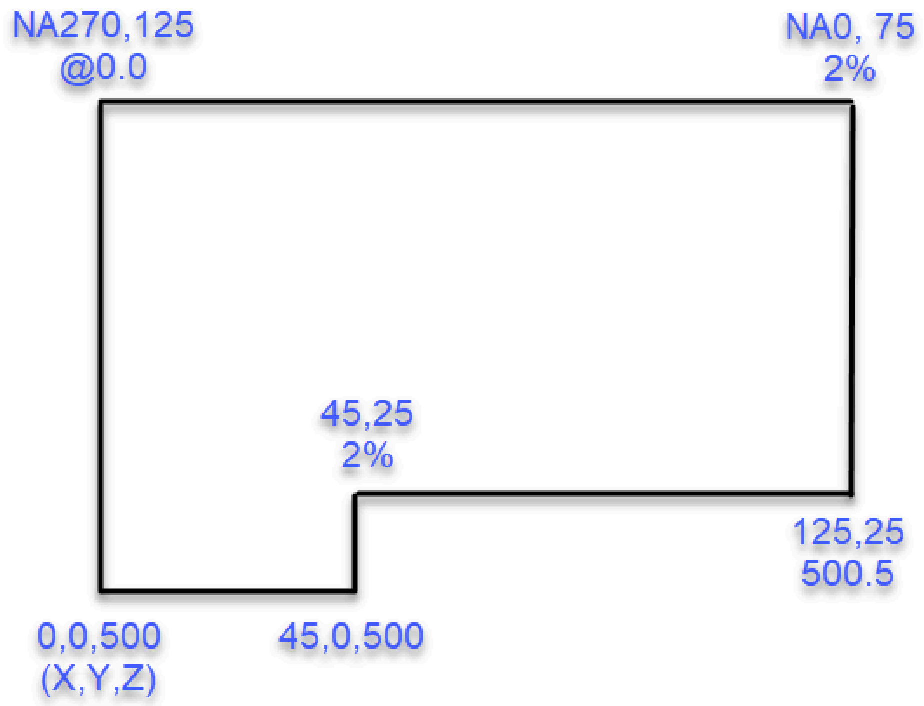


Current VPI: 1000.00 - Parabolic

4. **Add VPI**
5. **Remove current VPI**

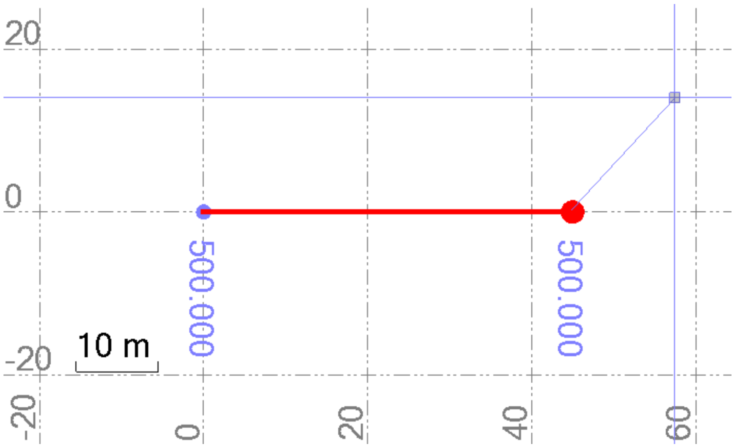
1.1 | Create a simple line with 3D nodes

Start by creating this simple site boundary using various methods to enter the X,Y,Z coordinates:

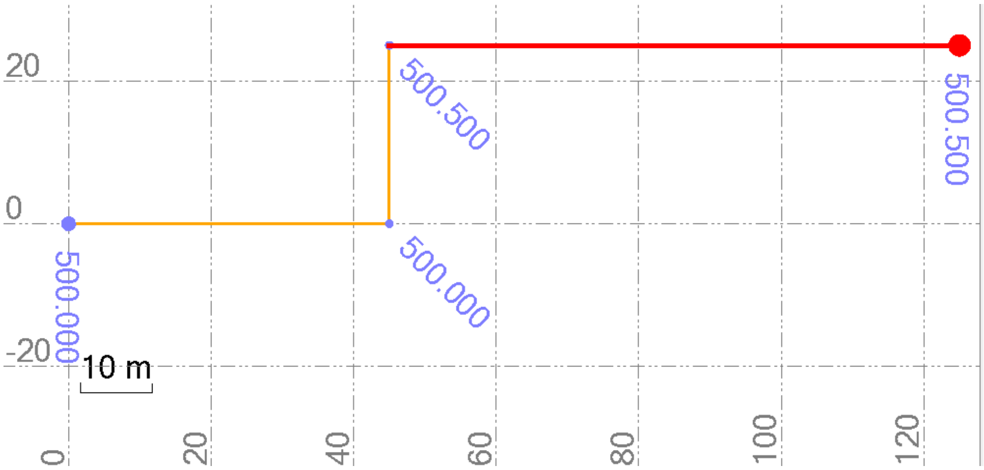


1. If desired, print the image above to make it easier to enter the values.
2. Open Trimble Business Center (TBC), and start a new default project.
3. Click the **New 3D View** icon on the **Quick Access Toolbar** so you can see your linestring in both 2D and 3D.
4. Right-click the **3D View** tab and select **New Vertical Tab Group** so you can see the **Plan View** and **3D View** side-by-side as you create.
5. On the ribbon, select **CAD > Create Linestring** (the command name may be abbreviated, but you can hover over each icon to see the full name).
6. Click **OK** to skip to the **Edit Linestring** pane (settings in the first pane are useful, but optional).
7. In the **Start Point** group's **Type** list, select/confirm **Coordinate**.
8. Click in the **Coordinate** field, and type *0,0* as the X,Y values to start at the origin of the model space.
9. Enter *500* for the elevation (Z) of the starting point in the **Elevation** field.
10. Click **Save** or press **[Enter]**. The first point of the linestring is saved and you are prompted to add the second point, designating how the segment between them is formed.

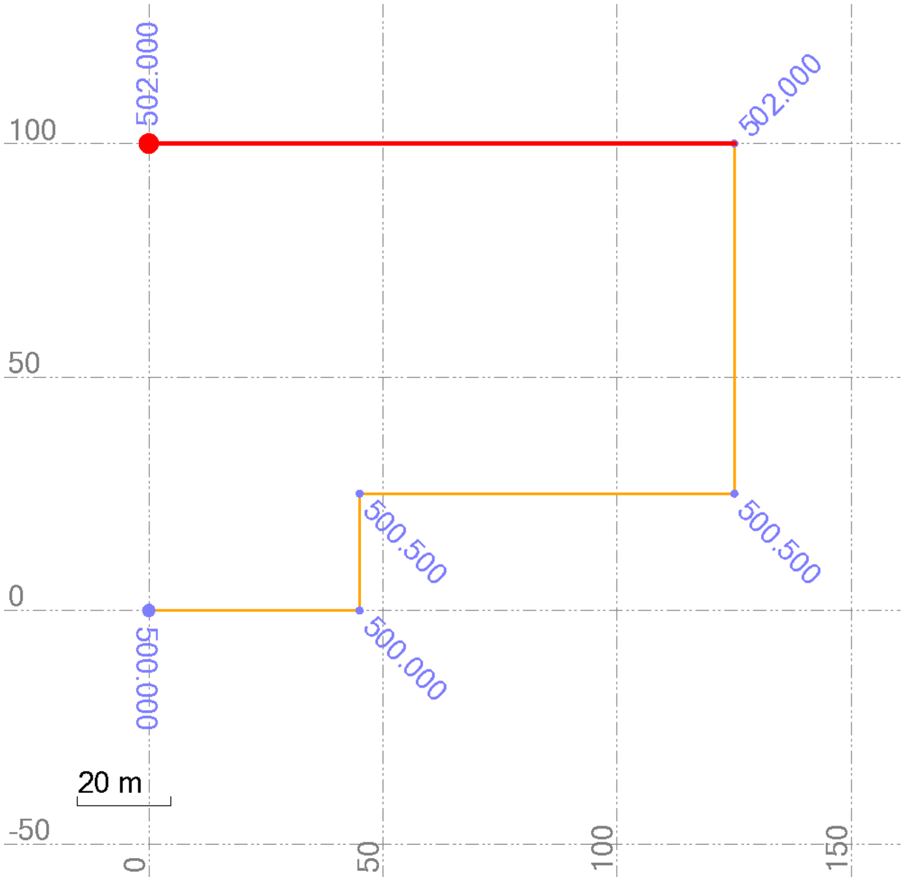
11. Using the values in the diagram above (work counter-clockwise), enter *45,0,500* in the **Coordinate** field, and press **Enter**. Note that the elevation was simply included as the 3rd coordinate in the X,Y,Z format, which avoids you having to type it separately in the **Elevation** field.
12. Double-click your mouse wheel to zoom extents (or simply roll the wheel to zoom in) until you see this in the **Plan View**:



13. Enter *45,25* in the **Coordinate** field, and 2% in the **Elevation** field, and press **Enter**. Note that you can enter coordinates, elevations, and other values in a variety of formats.
14. For the next coordinate, type *125,25*.
15. Tab to the **Elevation** field, and pick the last end point you entered on the linestring to use the same elevation.



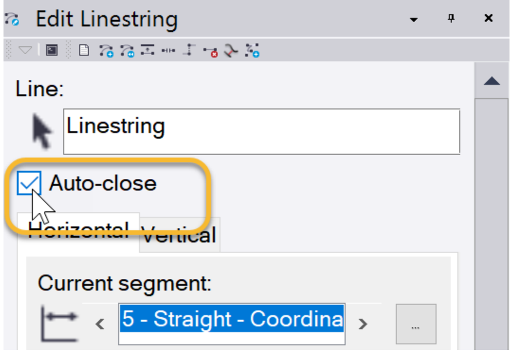
16. Press **Enter**. Enter the next coordinate using a bearing and distance.
17. Type *NA0,75* (for NA=north azimuth deflection of 0.0, and distance of 75 m).
18. For the elevation, enter *2%* again.
19. Enter *NA270,125* to add a segment 125 units long with a westward bearing.



20. For the **Elevation**, enter *@0* to specify a relative elevation change of 0.0. Adding @ before any value makes it a relative change from the last instance of that value.

    The last segment is the easiest to create.

21. Simply pick the starting point to close the boundary. Alternately, you could check the **Auto-close** box to close the shape.

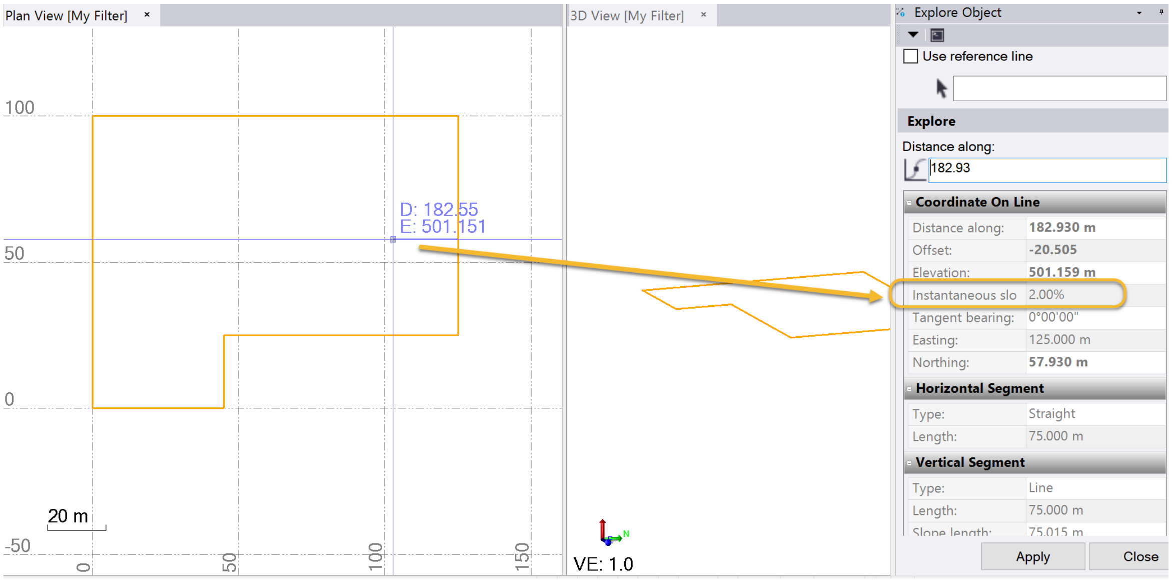


Linestring

☑ Auto-close

5 - Straight - Coordina

22. Then close the command.
23. To confirm the slope of the boundary, press F12 to refresh the Command pane.
24. Type *exploreobject* in the **Command Pane** and press **Enter**.
25. Pick the linestring and (using the 'rubber-band' line) click along any of the sides of the shape.
26. Check the **Instantaneous slope** value in the command pane.



Explore Object

□ Use reference line

Explore

Distance along: 182.93

Coordinate On Line

Distance along:	**182.930 m**
Offset:	**-20.505**
Elevation:	**501.159 m**
Instantaneous slo	2.00%
Tangent bearing:	0°00'00"
Easting:	125.000 m
Northing:	**57.930 m**

Horizontal Segment

Type:	Straight
Length:	75.000 m

Vertical Segment

Type:	Line
Length:	75.000 m
Slope length:	75.015 m

27. Close the command when you are done.

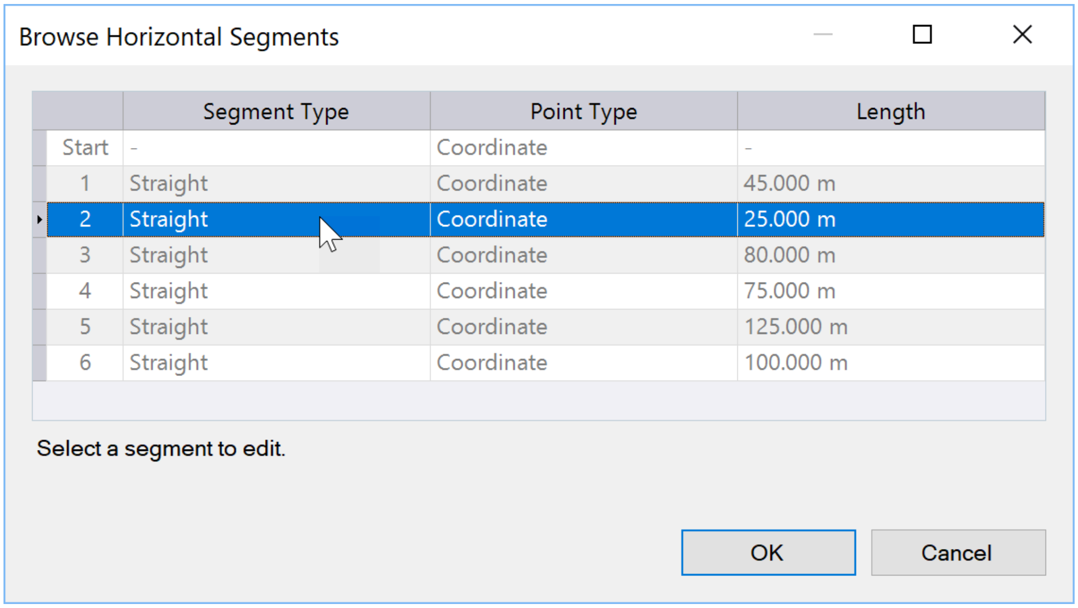
    Also see section 2.1.2 | Specific methods for creating a linestring.

1.2 | Create linestring segments using deflection angles

The benefit of creating a linestring using deflection angles is that they are relative to the last segment, so if you update a segment's bearing, the subsequent segments of the linestring are updated.

Use the **Deflection** segment type to create a straight segment that extends from the previous segment's end point to the new segment's end point based on the deflection angle and length you specify. If you select the **Specified angle** option, you can enter a custom angle or check the **Angle from backsight** box to specify that the angle be measured while backsighting the next point for the previous segment. Positive deflection angles are measured clockwise from the direction of the previous segment.

1. Select the linestring you just created, right-click, and select **Edit**.
2. Click the ... Browse icon to see the list of horizontal segments in the linestring.

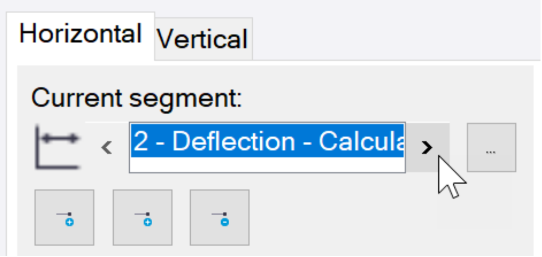


Browse Horizontal Segments

	Segment Type	Point Type	Length
Start	-	Coordinate	-
1	Straight	Coordinate	45.000 m
2	Straight	Coordinate	25.000 m
3	Straight	Coordinate	80.000 m
4	Straight	Coordinate	75.000 m
5	Straight	Coordinate	125.000 m
6	Straight	Coordinate	100.000 m

Select a segment to edit.

3. Select the third row (**2**) and click **OK** to edit it.
4. In the **Segment Type** list, select **Deflection**.
5. Click the **Perpendicular left** option for the **Deflection angle**. This specifies a 90° turn from the first segment to the segment.
6. Enter a **Length** of *25*, and press **Enter**. The geometry should not change.
7. To get to the next segment, click the right arrow in the **Current segment** list.

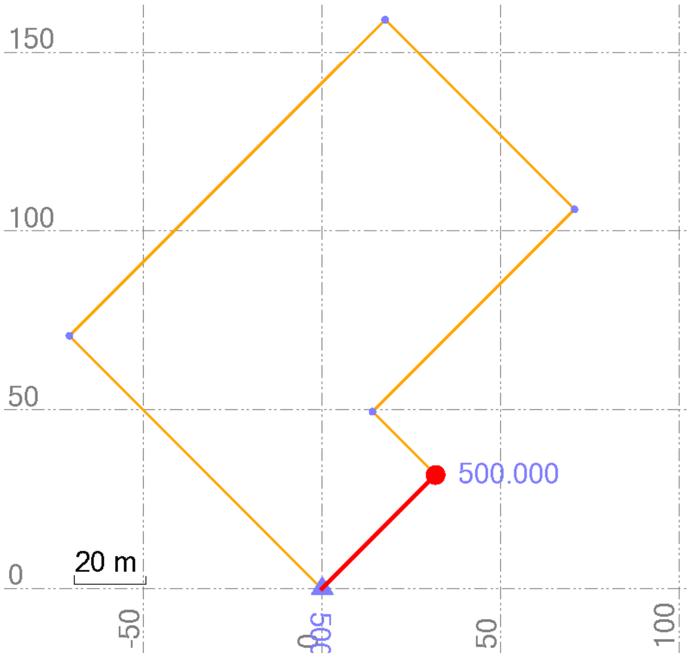


Current segment: 2 - Deflection - Calcula

8. Specify a **Perpendicular right** deflection of 80 m for this segment in the same way.
9. Edit the next three segments to use these deflections rather than absolute coordinates:
   a. 4 - **Perpendicular left, 75 m**
   b. 5 - **Perpendicular left, 125 m**
   c. 6 - **Perpendicular left, 100 m**

   The site boundary's shape has not changed, which is good, but you will now see the value of using deflections by changing the bearing of the first segment.

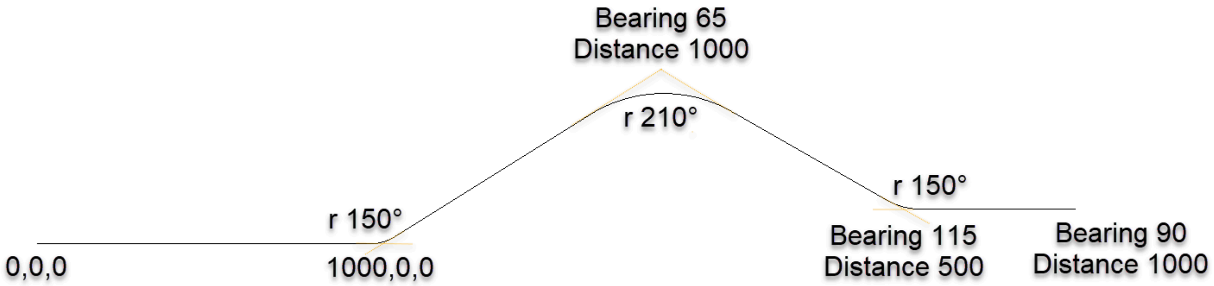
10. Click the left arrow next to the **Current segment** list to back up the the first segment, and enter *na45,45*.



The entire boundary is reoriented, but retains its shape and dimensions.

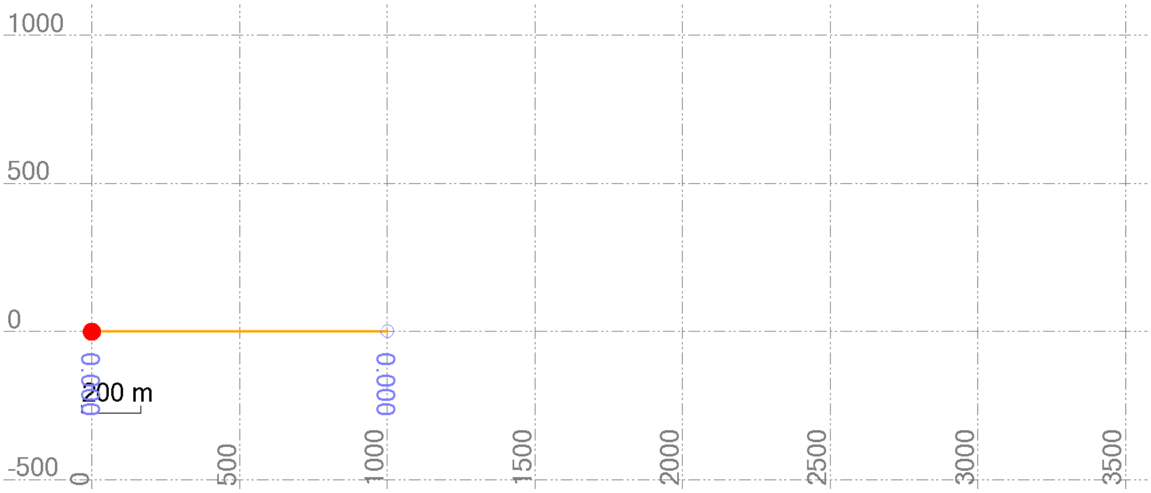
1.3 | Create a line with PI arcs using coordinates and points

Use the **PI Arc** segment type in linestrings to create arcs based on the point of intersection (PI) between two segments. For the first segment that specifies the PI, you will enter a radius and the coordinate (or point ID) of the PI. Based on the radius, this first segment will include a straight portion tangent to the arc and then the arc. If the radius is too large, the entire segment will be an arc. Then you will specify the next segment from the PI as usual.

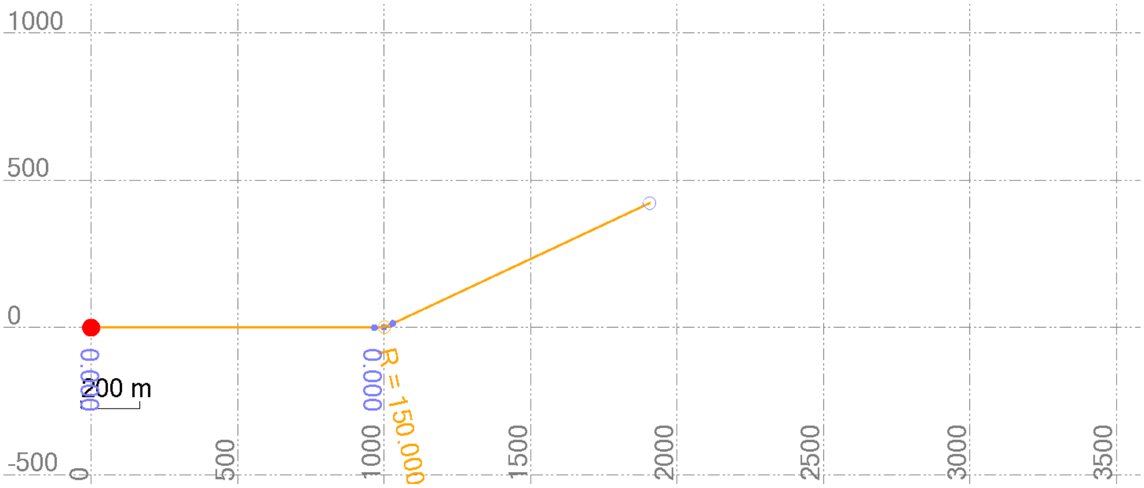


1. Print the image above to make it easier to enter the values.
2. Delete the linestring you created in the previous exercise, and close the **3D View** tab.
3. Reopen the **Create Linestring** command and click **OK** to skip the first pane.
4. Click in the **Coordinate** field, and enter the *0,0,0* as the starting point.
5. Skip the **Elevation** field and click **Save**. You specified the Z-level/elevation as 0 above.

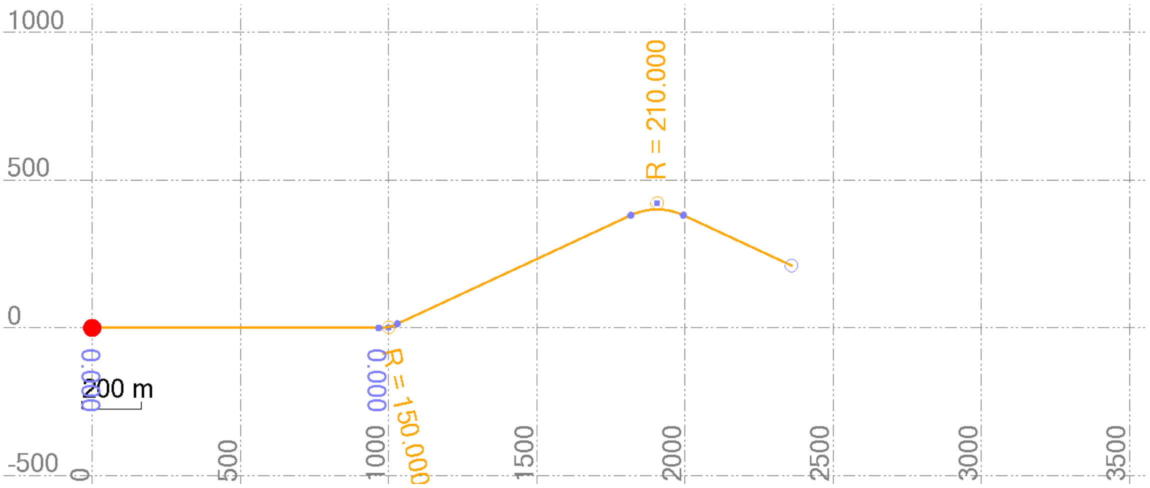
6. In the **Segment > Type** list, select **PI Arc**.
7. For the **Radius**, enter *150* as shown in the plan.
8. For the **PI Point > Coordinate**, enter *1000,0,?*. The ? indicates that the elevation is unspecified.
9. Press **Enter**.



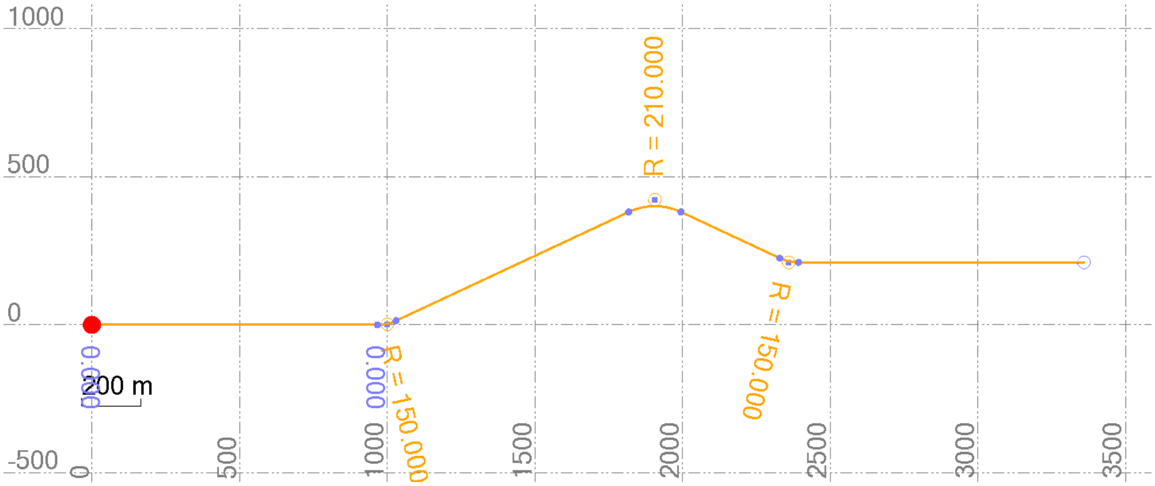
10. Leave the **Segment Type** as **PI Arc**, and change the **Radius** to *210* .
11. In the **PI Point > Coordinate** field, right-click and select **Bearing Distance** from the list of CoGo snaps.
12. Enter *65* in the **Bearing** field, *1000* in the **Distance** field, and press **Enter**.
13. Press **Enter** again to add the segment.



14. Leave the **Segment Type** as **PI Arc**, and change the **Radius** to *150.*
15. In the **PI Point > Coordinate** field, right-click and select **Bearing Distance** again.
16. Enter *115* in the **Bearing** field, *500* in the **Distance** field, and press **Enter**.
17. Press **Enter** again to add the segment.



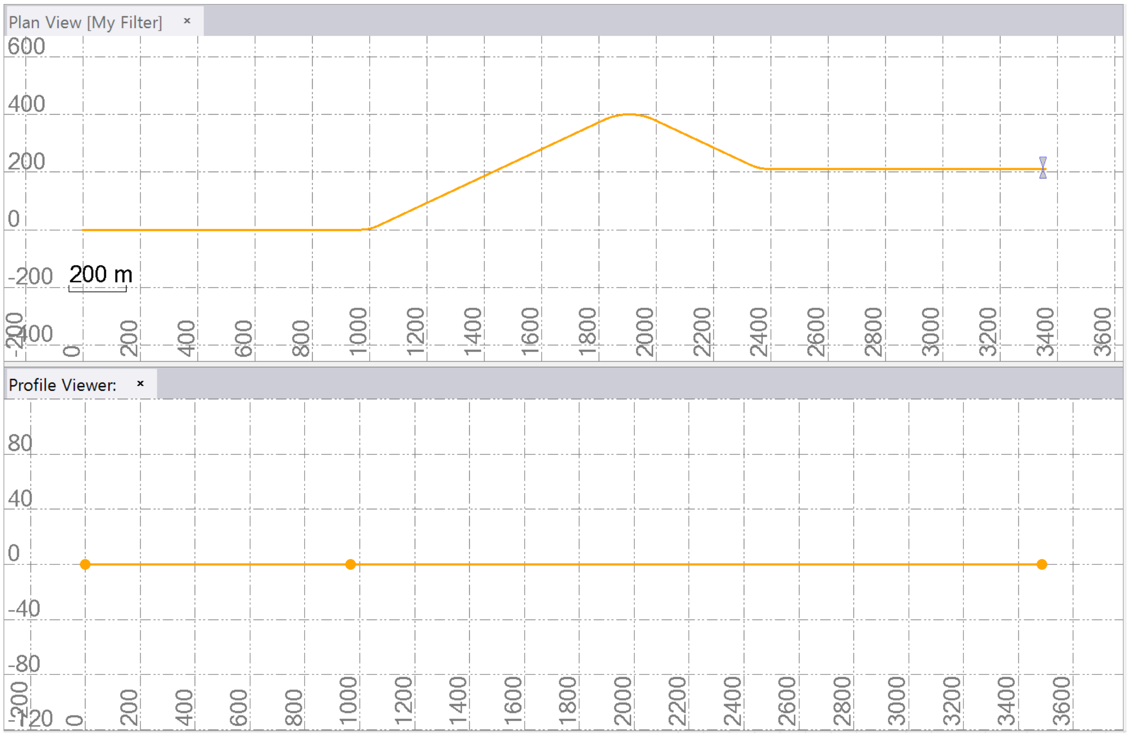
18. Select **Straight** as the **Segment type**.
19. In the **End Point > Coordinate** field, right-click and select **Bearing Distance** again.
20. Enter *90* in the **Bearing** field, *1000* in the **Distance** field, and press **Enter**.
21. Press **Enter** again to add the segment.
22. Leave the command open to add elevations in the next procedure.



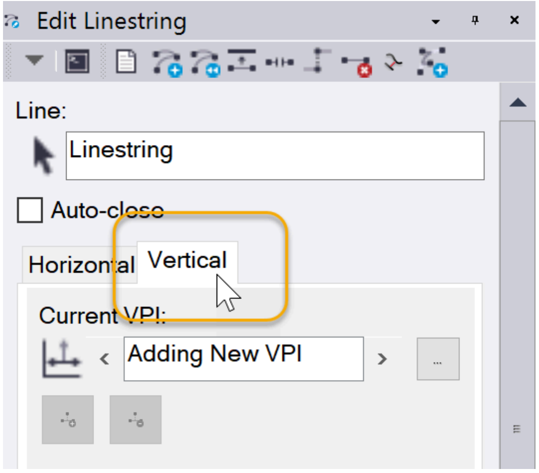
1.4 | Add elevations using Vertical Points of Intersection (VPIs)

If you do not want to simply add elevations to the end points of horizontal segments, you can add VPIs anywhere along a linestring. Elevating your linestring by adding VPIs enables you to preserve those elevations at their specified distances along the linestring even if the linestring's geometry changes. See **1.5.2 | Vertical linestring components** for more information.

1. Press **F12**, type *profileviewer* on the **Command** line, and press **Enter**. The **Profile Viewer** enables you to check the profile geometry of a single, selected linestring in a graphic view.



2. If you closed the **Edit Linestring** command pane from the previous steps, right-click the linestring you just created and select **Edit**.
3. Click the **Vertical** tab.


Edit Linestring

Line:

Linestring

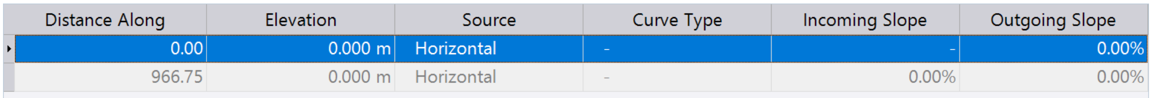
□ Auto-close

Horizontal | Vertical

Current VPI:

Adding New VPI

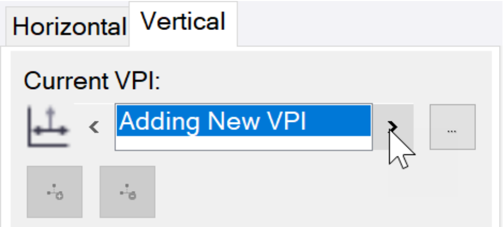
4. Click the … **Browse** icon to see the list of horizontal segments in the linestring. As you can see, there is no vertical information yet:



Distance Along	Elevation	Source	Curve Type	Incoming Slope	Outgoing Slope
0.00	0.000 m	Horizontal	-	-	0.00%
966.75	0.000 m	Horizontal	-	0.00%	0.00%

5. Click **OK**.

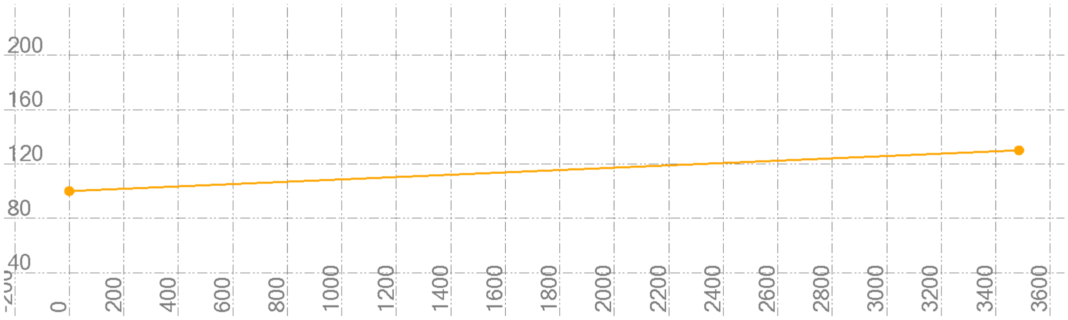
6. Click the **Vertical** tab again. Unlike on the **Horizontal** tab, there is no Current segment field, simply a Current VPI field; when you add VPIs, you can add them anywhere along the linestring.



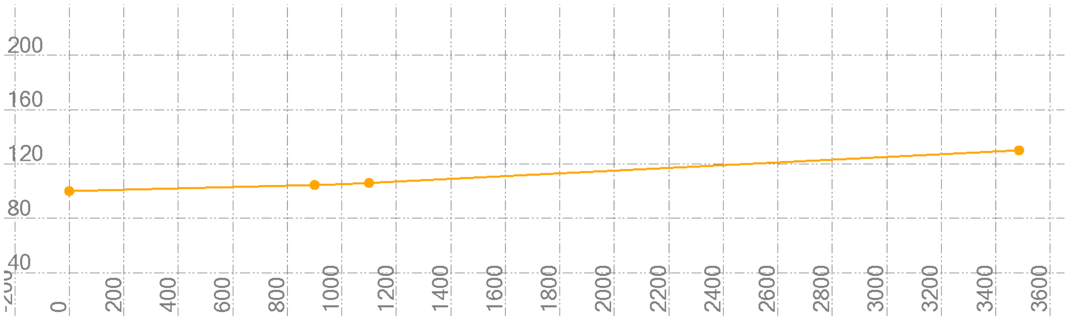
Current VPI: Adding New VPI

7. In the **Curve > Type** list, select **No curve**.
8. In the **Distance Along** field, enter *0* to elevate the linestring's starting point.
9. In the **Elevation** field, enter *100.* Now, if you add a segment before this point, this elevation will move to the new starting point.

   Add another **No curve** VPI with an elevation of 130 at the other end of the linestring (3485.29).

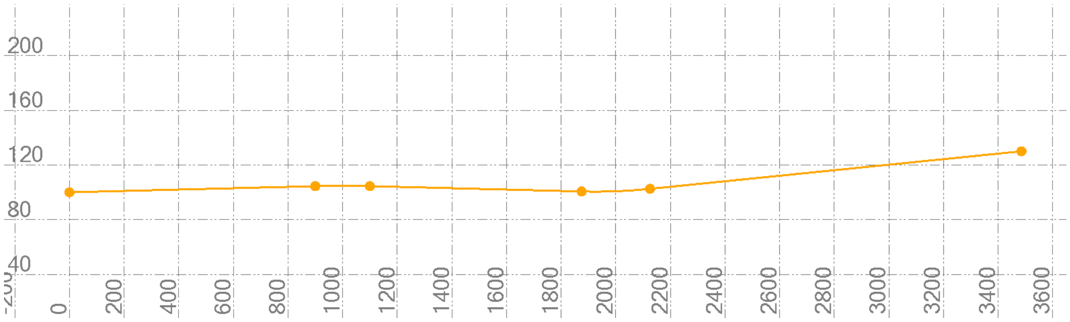


10. In the **Curve > Type** list, select **Parabolic**. Typically, the **Arc** VPI type is used in Europe and other parts of the world.
11. In the **Length** field, enter *200*.
12. In the **Distance Along** field, enter *1000* to place parabolic curve at the first bend in the linestring.
13. In the **Elevation** field, enter *105*. Click **Save** or press **Enter**.

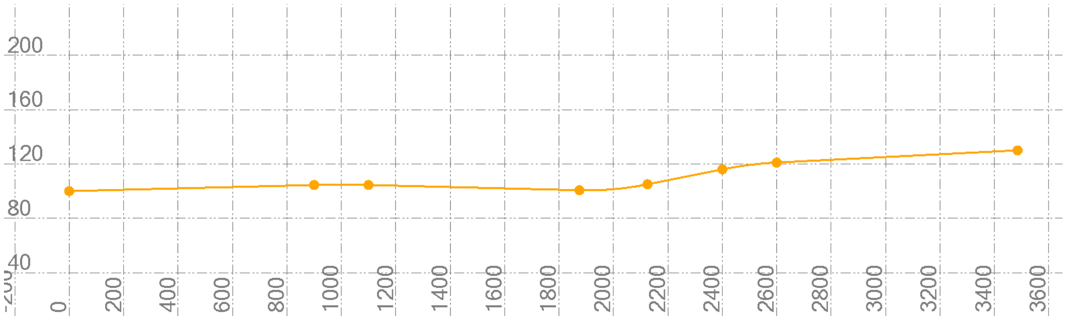


14. Leave the **Curve > Type** set to **Parabolic**.
15. In the **Length** field, enter *250*.

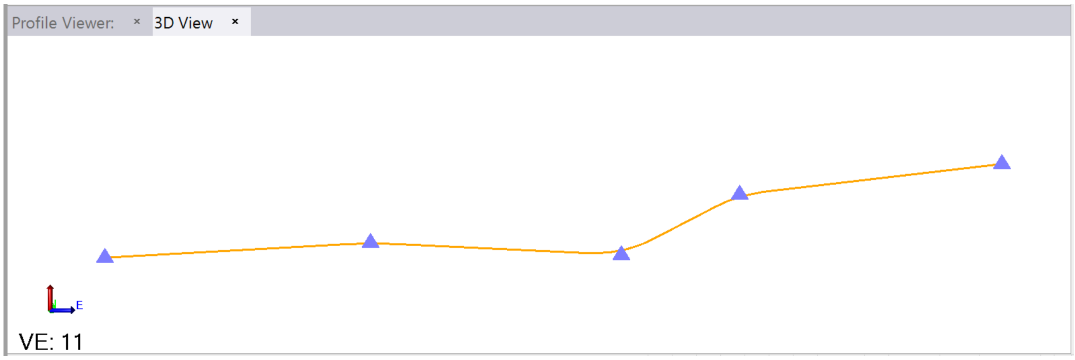
16. In the **Distance Along** field, enter *2000* to place parabolic curve at the second bend in the linestring.
17. In the **Elevation** field, enter *100*. Press **Enter**.



18. Leave the **Curve > Type** set to **Parabolic**.
19. In the **Length** field, enter *200*.
20. In the **Distance Along** field, enter *2500* to place parabolic curve at the second bend in the linestring.
21. In the **Elevation** field, enter *120*. Press **Enter**.



22. If desired, reopen a **3D View** to confirm the elevation changes. Press **Shift + Control** and roll the mouse wheel to increase the vertical exaggeration. Press **Shift** and roll the mouse wheel until you are viewing the profile (watch the compass triad).



23. Feel free to continue experimenting with different types of VPIs such as those with incoming, outgoing, and offset slopes, as well as those based on slope and line intersections.

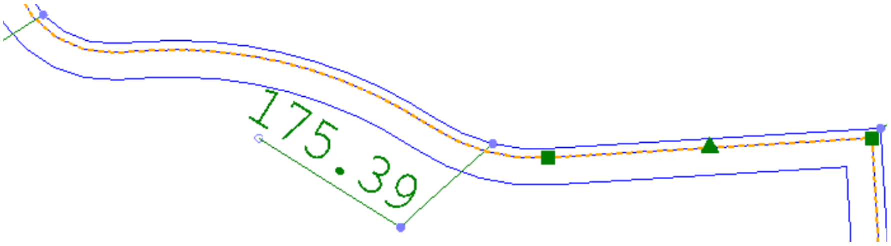
1.5 | Linestring fundamentals

What is a linestring?

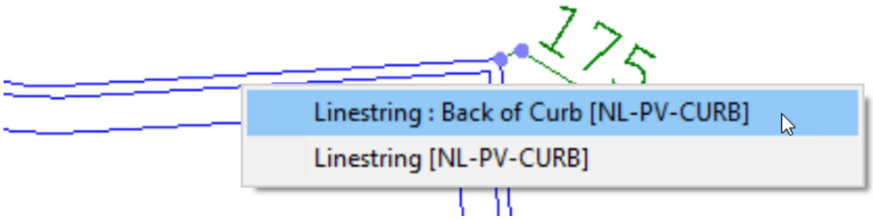
Early CAD system created very primitive 2d lines and arcs to recreate traditional drafting. As CAD systems evolved, new types of line objects were created to represent the complex combinations of 3d lines and curves. Unfortunately most systems are still limited to drafting needs and not the unique needs of construction modeling.
One advantage of creating a purpose built modeling system such as TBC, is that you can create your own objects that are smarter and address your needs. A linestring is a type of line designed to provide the specific needs of surface and construction models.

Linestrings are at the heart of a lot of TBC's power, so it's important to understand how they function and can be used to solve a variety of problems. Based on the previous exercises, you may know that a linestring:

- is a highly-versatile, multi-segmented type of line
- can have one or more straight or arc segments (horizontal connections with ends defined by nodes or point objects) and possibly 'vertical points of intersection' (VPIs) and vertical control points

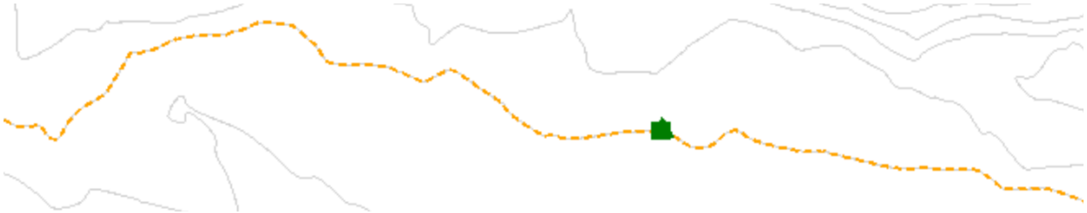


- can be either 2D or 3D
- can be uniquely named



- can be parametric if nodes (segment end points) are based on named points or station offsets from another line; if the points or line change, the linestring changes accordingly
- has separate controls for its horizontal segments and nodes versus vertical segments and points of intersection

- has basic CAD properties, such as a name, line style and weight, color, and layer
- has advanced properties, such as auto-close, extend vertical, and override segment length
- can be used to confine a surface texture image within an area
- can be set to act as a closed shape, even when its ends do not connect
- can be created by editing almost any other type of line (except an alignment)
- is created when you process line feature codes
- can be used for almost any function that requires a line



Some advantages of linestrings include:

- They provide a single way to create and edit a line object capable of representing numerous real-world objects and common site features.
- They minimize the number of line objects with which you must be familiar.
- They overcome the most common faults inherent in imported CAD line data.
- They support additional important properties than can be assigned to imported lines.
- They support more sophisticated geometric construction features than many imported lines can represent.
- They facilitate data preparation operations for job sites and corridors.
  See the YouTube video: [Data Prep Workflow - Trimble Business Center]
- They enable key field data preparation operations for Trimble Siteworks.
- They enable key survey operations.
  See the YouTube video: [Transform Survey Points]

Advanced Linestring Capabilities (Quick Reference Sheet)

The following is a summary of the advanced capabilities for entering linestring values. Linestrings can have horizontal nodes and vertical nodes to control the elevation. The horizontal nodes can be 3D or 2D. The vertical nodes can be elevations or have curves. Vertical control points are used when you need to add more vertical definition to the line beyond that defined by the existing horizontal control points or instead of adding more horizontal control points. They also add the ability for the user to define a vertical curve or symmetrical parabolic curve within the line.

Coordinate

- Enter a single value (Poin ID) to snap to a point object

  1234
- Enter a coordinate pair to specify XY

  500 700
- Enter a simple math expression (link to cogo expressions)

  500/2 300*3 when you tab off will display: 250 900
- Right click for snap modes (link to snap modes)
- Enter bearing and distance

  N45e 100

Elevation

- Select a text object to recall the value
- Pick the text

  Elevation = 200.123

  The program reads just the value.
- Enter a vertical angle/slope

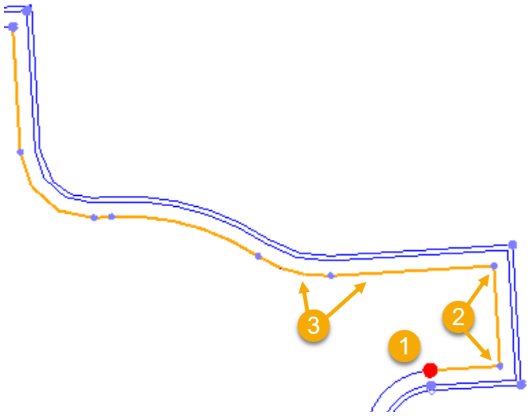
  2%

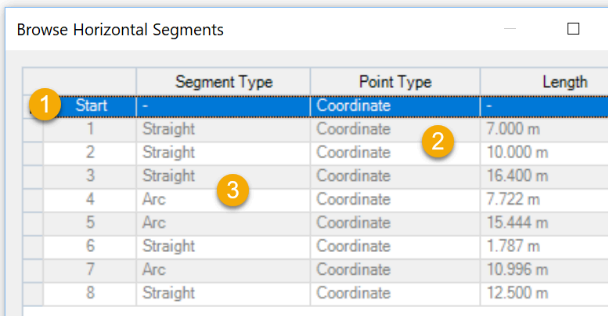
  The program calculates the elevation at 2% from the last elevation in the line.

1.5.1 | Horizontal linestring components

The geometry of each linestring is composed of both horizontal components and optional vertical components. The horizontal component is created by simply specifying:

1. a starting coordinate
2. subsequent nodes (coordinates or named points) that indicate a change in bearing or elevation relative to the last node
3. geometric connections (straight, curve, arc, deflection) in between each node





	Segment Type	Point Type	Length
Start	-	Coordinate	-
1	Straight	Coordinate	7.000 m
2	Straight	Coordinate	10.000 m
3	Straight	Coordinate	16.400 m
4	Arc	Coordinate	7.722 m
5	Arc	Coordinate	15.444 m
6	Straight	Coordinate	1.787 m
7	Arc	Coordinate	10.996 m
8	Straight	Coordinate	12.500 m

If the elevation and connection type between each subsequent node is consistent, you can simply specify 2D coordinates for the nodes in quick succession.

1.5.2 | Vertical linestring components

Vertical points of intersection (VPIs)

The vertical components (elevations) of a linestring can be established in one of three ways:

- By assigning elevations to the fixed start and end points of a linestring's segments (discussed above).
  - A VPI established in this manner cannot have a vertical curve. A VPI of this nature remains located at that particular horizontal segment end point, even if the horizontal alignment geometry is edited, thereby potentially placing that end point at a different distance along the linestring's length. The resulting VPI can be overridden by a VPI that might be assigned, as a part of the linestring definition, at the same distance along measurement at which the internally defined end point currently lies.
  - You can enter next node elevations relative to the last node’s elevation by entering a % or a slope ratio.
  - You can use a variety of snaps in each coordinate, elevation.
  - Override interpolated elevations entered on the horizontal tab by adding VPIs.
- By assigning elevations to dependent points that a linestring's segments use as start and end points. The elevation of the line at each point is defined by the point object's elevations. A VPI established in this manner cannot have a vertical curve. Vertical

control points that you add at the same distance along the linestring override VPIs established using either of the two ways described above.
- By adding vertical control points (discussed below) along the linestring.

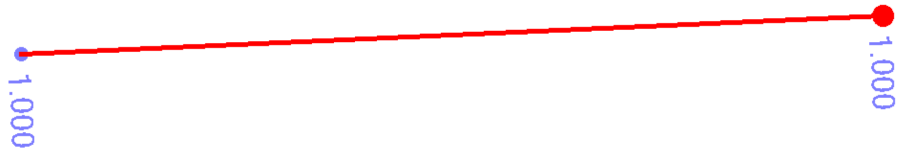
Vertical control points:

Each vertical control point is defined by:

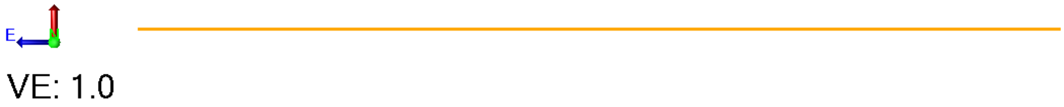
- A curve type (parabolic, arc, or no curve)
- A distance along the linestring
- An elevation

A VPI established in this manner has no influence on the line's horizontal alignment. Vertical control points that you add at the same distance along the linestring override VPIs established by fixed coordinates or named points. However, such a VPI will remain at that distance along the entire line as its horizontal geometry is edited, rather than remaining at that particular segment end point.

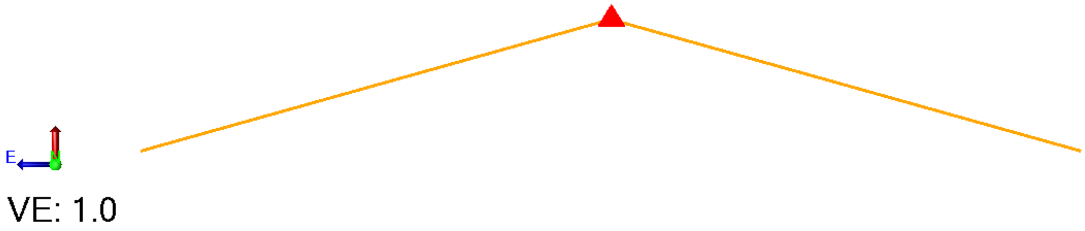
Here is a simple 100 m linestring segment with a constant elevation of 1.0 shown from above (**Plan View**):



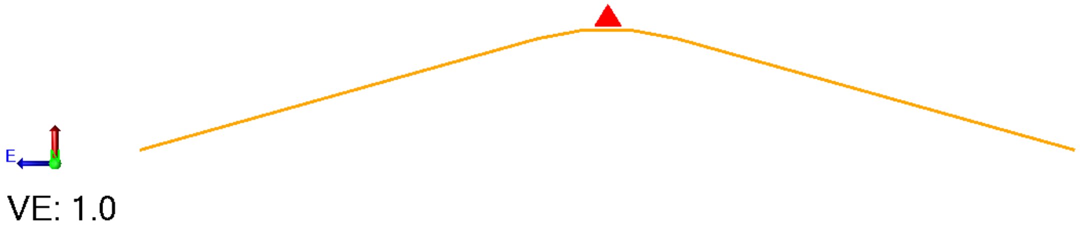
Here is the same segment shown from the side in the **3D View**:



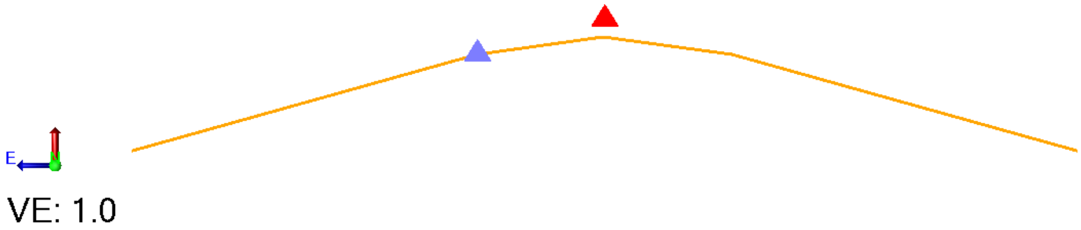
Here is a the same view after adding a simple **No curve** VPI with an elevation of 15 m halfway along:



Here is the same segment with a 15 m **Parabolic curve** VPI at the same location and elevation:



Here is the same segment with a 50 m **Arc** VPI at the same location and elevation:



You can also create curve and arc VPIs with specific incoming, outgoing, and offset slopes, as well as based on slope and line intersections.

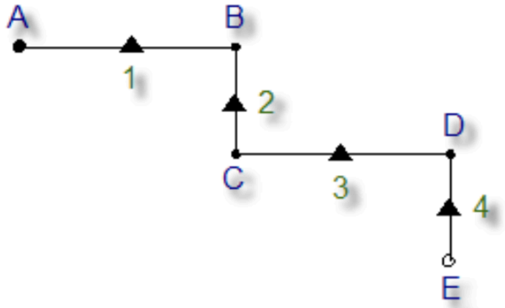
1.5.3 | Scenarios on Moving VPIs

Editing a linestring that has vertical points of intersection (VPIs) can occasionally affect the stations of those VPIs in unexpected ways. In general, if the 2D (x,y) coordinate of a VPI on the linestring is still on the linestring after the edit, the VPI is updated with the new station value, if any. In two instances, the VPIs are not updated with new station values:

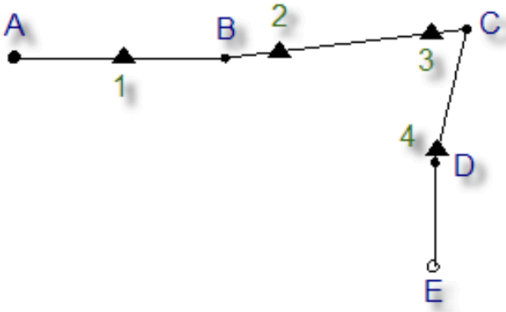
- The linestring is edited outside of the **Edit Linestring** command; e.g., if a named point that the linestring references is moved, the VPIs are not adjusted.
- The edit changes the order of the VPIs on the linestring; the VPIs are not adjusted. See the fourth scenario below.

Note: If you change the length of a linestring's first segment, any vertical points of intersection (VPIs) along the linestring retain their correct distance along the linestring. When making edits affecting the geometry of a linestring, adjustments are made to the distance along values associated with VPIs located on unaffected segments, thereby maintaining their former relative position along those segments.

Here is a linestring with horizontal segments denoted by lettered end points and VPIs denoted by numbers.

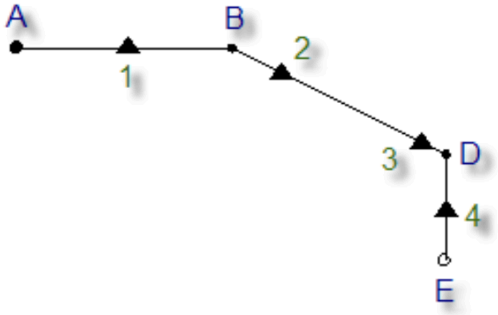


- If you have horizontal segments with VPIs on them, and you move the end point labeled C:
    - VPI 1 is adjusted (the station is actually unchanged)
    - VPIs 2 and 3 are not adjusted (the stations remain the same)
    - VPI 4 is adjusted (the station is changed)

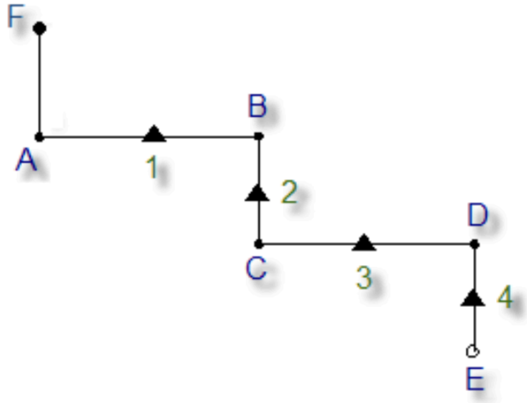


- If you have horizontal segments with VPIs on them, and you delete the B to C segment:
    - VPI 1 is adjusted (the station is actually unchanged)

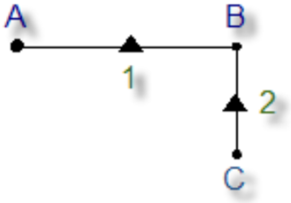
    - VPIs 2 and 3 are not adjusted (the stations remain the same)
    - VPI 4 is adjusted (the station is changed)



- If you have horizontal segments with VPIs on them, and you add the F to A segment:
    - VPIs 1, 2, 3, and 4 are adjusted (their stations are changed)

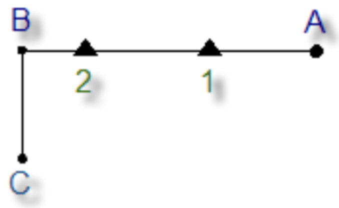


- If you have two horizontal segments with VPIs on them…



...and you move the A end point to a location that would change the order of the VPIs:

- VPIs 1 and 2 are not adjusted (the stations remain the same)



See the Retrieve video: Delete Vertex of a Linestring (from Getting Started > Editing Objects)

Chapter 2 | Creating, Editing, and Checking Linestrings

2.1 | Creating and editing linestrings

2.1.1 | General workflow for creating a linestring

The best way to understand this workflow is by working through the simple exercises at the start of this document.

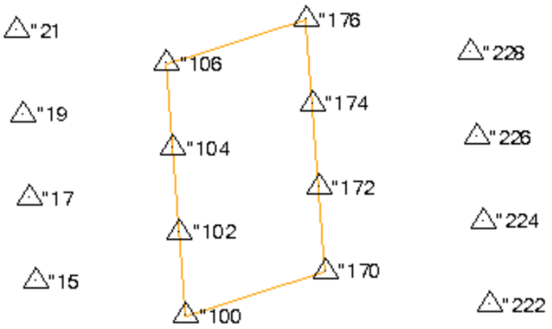
1. Enter the first node by specifying a 3D coordinate, selecting an existing point, or selecting a line from which to calculate points at an offset and elevation.
2. Enter the first segment, which can be straight (to a coordinate or point) or at a standard deflection, or curved (arc).
3. Continue entering new nodes with defined segment types in between.
4. If needed, add vertical control points (VPIs) anywhere along the linestring.

2.1.2 | Creating linestrings

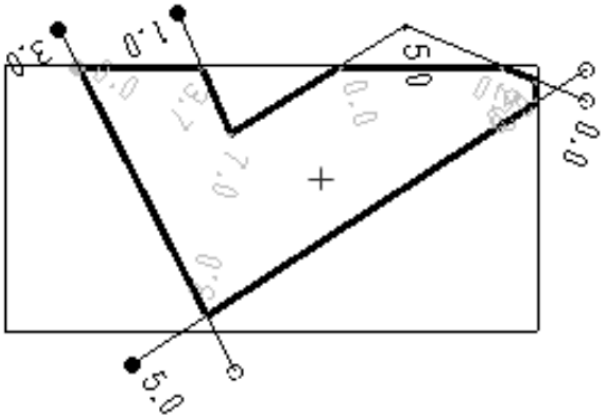
In addition to using the **Create Linestring** command, here are some of the other ways you can create a linestring:

Note: When you are in the program, you can press F1 to open the help. In the help, you can search for the command name and launch it from the related help topic.

- Use the **Edit** command to convert any type of CAD line (not including alignments) into a linestring. Right-click the line and select **Edit** to convert it into a linestring. Imported CAD lines are automatically converted into linestrings when you edit them.
- Use the **Create Linestring from Points** command to create a linestring through a series of points, either by specifying a range of point IDs or by selecting points graphically. The resulting linestring includes only horizontal segments; no VPIs are defined by this command. See the Retrieve video: 3.71 New Features > Create Linestring from Points (YouTube?)

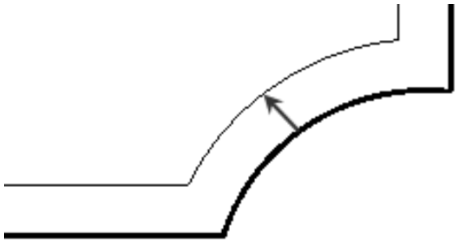


- Use any **Digitize**… command to 'digitize' (on-screen) a linestring, either by streaming points (adding points as you hold down the digitizer button and move the puck, and releasing the button at the end point). For simpler linestrings, you can simply pick individual points.
- Use the **Track Region Outline** command to create a 2D or 3D linestring representing the 2D outline of a region from a set of separate (potentially unconnected) lines and/or surface and site improvement edges. The region is derived from portions of the bounding lines/edges that planimetrically intersect (in **Plan View**) to form an enclosing region surrounding an interior 'seed' point. The region's area is expanded outward from the interior point in all directions until a selected line/edge is encountered. When an area is bounded using the Track Region outline command, the linestring that is computed now carries the "Closed" property.

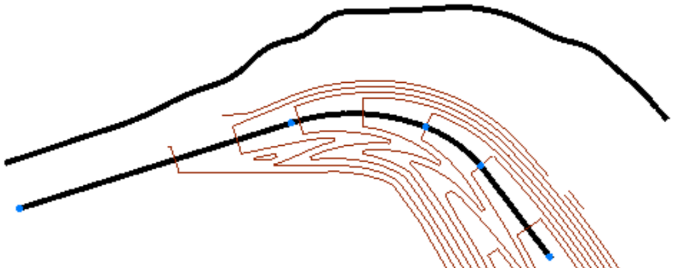


- Use the **Set a Line Elevation** command to apply a single, constant elevation to a 2D line to make it a 3D linestring. You can also specify a vertical offset from the elevation that you enter and a surface to which the linestring will be added.
- Use the **Offset Line** command to create a new linestring offset at a specified distance from an existing line. The line from which you are offsetting can be either 2D (such as a boundary, polyline, circle, or an alignment without a vertical alignment (VAL)) or 3D (such as a 3d linestring or an alignment with a VAL). In addition, the offset can be horizontal, vertical, or both. Line segments offset from spirals and smoothed segments are chorded. **Note:** You cannot set a line elevation on a line that already has elevations.

To modify the existing elevations, add vertical control points using the **Edit Linestring** command.



- Use the **Variable Offset Line** command to create a linestring offset at varied distances from an existing line. The line from which you are offsetting can be either 2D or 3D with an elevation or slope.
  See the Retrieve video: Create Linestring at a Variable Offset (from Getting Started > Tip of the Week)



- Use the **Create CAD Entities on Mobile Mapping Data** tool (in combination with TBC's photogrammetry measurement capabilities) to create linestings using mobile mapping data.
- Use the **Point Cloud Smart Picking** tool when creating linestrings to roughly pick a scan point near a curb, gutter, or roadmark edge line and have the software automatically compute the best face-of-curb, gutter flowline, or roadmark edge point and display that point in an editable cross-section view. You can continue to roughly select each point necessary to, for example, define linework that accurately represents a gutter and can be properly feature coded and attributed for finished CAD deliverables. If you are creating standalone CAD points, you can use the points to, for example, create a best-fit line or measure offsets.
- See the Retrieve video: Create Linestring (as part of the Getting Started > Draw Objects)
- See the Retrieve video: Create New Linestring (from New Features in version 3.7)
- See the Retrieve video: 3D Linestring (YouTube?)

2.1.3 | Specific methods for creating a linestring

- See the Retrieve video: Draw Linestring by Bearing Distance (from Getting Started > Draw Objects)
- See the Retrieve video: Create Linestring By Deflection

2.1.4 |Converting CAD lines into linestrings

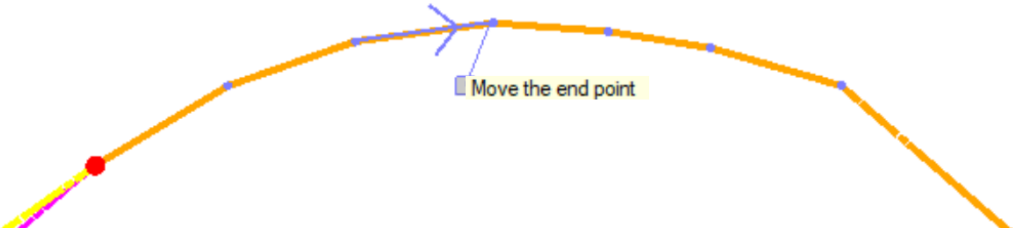
Frequently, you will want to convert one or more (or thousands) of CAD lines into linestrings so they have more properties and are more powerful. Sometimes a linestring is needed simply because only linestrings can have the **Sharp and texture boundary** setting that is required to constrain texture maps to a certain surface area (or to apply null textures to where no triangles were provided in the source model, but there were no boundaries provided to constrain the holes). You can use the following commands to convert CAD lines to linestrings:

- Use the **Convert to Linestring** macro command to convert many CAD lines (breaklines, polylines, arcs, rectangles, etc.) and horizontal alignments into linestrings without having to edit each one.

> ***Note***: Macro commands require a **Survey Advanced** or higher license. For more information, contact your dealer or visit the **TBC Macros and Extensions** page.

2.1.5 | Editing linestrings

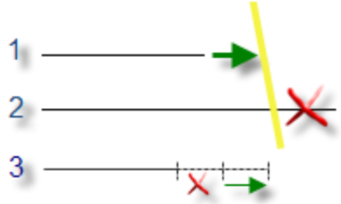
> ***Note***: Editing a linestring that has vertical points of intersection (VPIs) can occasionally affect the stations of those VPIs in unexpected ways. In general, if the 2D (X,Y) coordinate of a VPI on the linestring is still on the linestring after the edit, the VPI is updated with the new station value, if any. In two instances, the VPIs are not updated with new station values. See 1.5.3 | Scenarios on Moving VPIs in this document.



See the Retrieve video: Edit Linestring (as part of the Data Prep workflow)
See the Retrieve video: Editing Objects (from Getting Started)

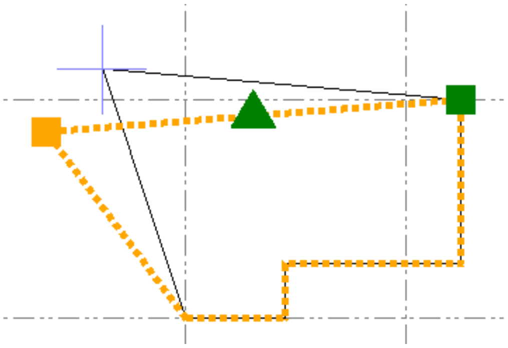
- Use the **Append Tracked Line** command to track the geometry of an existing line to create segments in an existing linestring.
  See the YouTube video: Append Tracked Line - YouTube
- Use the **Move Objects** command to update or move any objects whose locations depend on the object being moved while you are dragging the parent object. For example, if the end of a linestring is based on a point, the linestring will change if the point is moved.

- Use the **Break Line** command to split a linestring into two separate linestrings at the point you specify. The linestrings remain coincidental at the break point, but must be edited separately thereafter.
- Use the **Join Lines** command to join two or more contiguous lines into individual linestrings using manual and automated methods. When lines are joined, they are converted into linestrings. The resulting linestring's direction and properties are determined by the first (base) line that you choose.
  See the Retrieve video: [Connect Linestrings without Joining them]()
- Use the **Trim/Extend** command to:

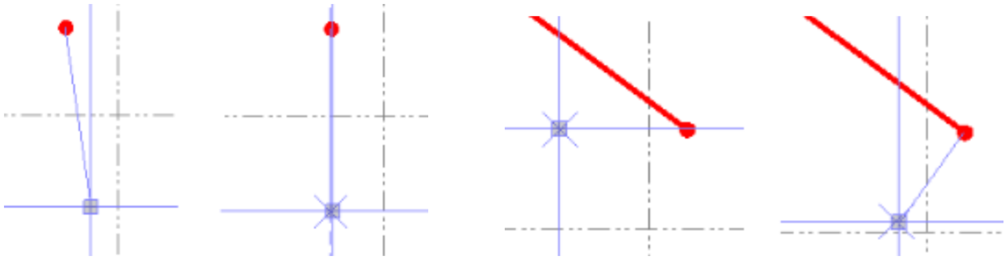
  

  - lengthen or
  - shorten selected lines to their planimetric intersection with a specified bounding line (in yellow)
  - shorten or lengthen selected lines by a specified distance

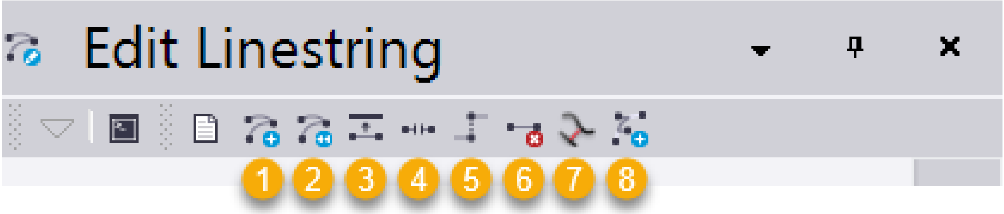
- Use the **Divide Lines** command to divide a linestring into multiple segments defined by a specified number of segments or segment length. You can then opt to create CAD points or orthogonal lines ending with CAD points at each division.
  See the Retrieve video: [Divide Linestring]() (from BC-HCE 4.0 New Features)
- Use **grips** in the **Plan View** to move and modify linestrings. When using a grip to move a linestring segment or end point, hold the **Shift** key while dragging the grip to lock into either a vertical or horizontal movement. Press the **Control** key to move the entire linestring. See Move and Modify CAD Objects Using Grips in the help.

- Use the **Running snap modes** > **Ortho tracking** option when you are creating a new segment for a linestring. You have the option of creating the new segment orthogonal (perpendicular) to the axis or orthogonal to the previous line segment, depending on the location of the cursor.

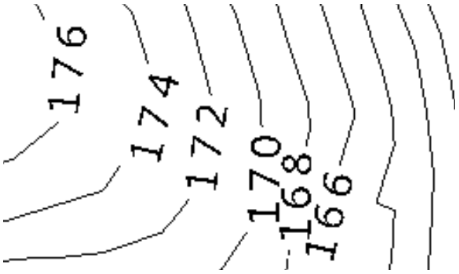


- To make edits to any of the individual linestrings in the COGO collection, select the segment in a graphic view (or select the entire linestring in the **Project Explorer**), right-click, and select **Edit** to display the **COGO** tab in the **Create COGO** command pane.
- Within the **Edit Linestring** command, you can also access these commands/functions:



   1. **Create Linestring**
   2. **Reverse Linestring Direction** - Switch the order of the linestring's segments and the start and end points.
      See the Retrieve Video: Reverse Linestring Direction
   3. **Set Line Elevation**
   4. **Break Line**
      a. See the YouTube video: Break and Edit Linestrings
   5. **Join Lines**
   6. **Delete Line Segment**
   7. **Toggle Segments Curved/Straight** - Change all straight segments in the linestring to smooth curve segments, and vice versa. If there are straight or arc segments in the linestring already, this has no effect.
   8. **Append Tracked Line**

- There are several ways to label different types of linestrings
   - Use the **Label Contours by Crossing** command to label selected contour lines (or other lines) with their elevations where they are crossed by a line drawn between a From and To point.



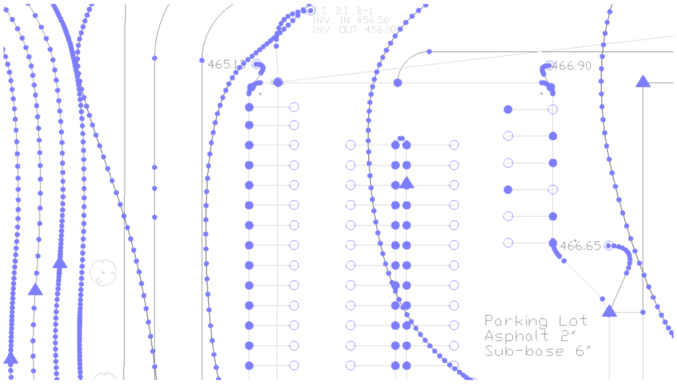
- See the Community question: Easy way to label a linestring with cut/fill from a surface
- See the Community question: Label linestrings by attribute
- See the YouTube video: Labeling Points, Lines and Polygons - Trimble Business Center

2.2 | Processing line feature codes into linestrings

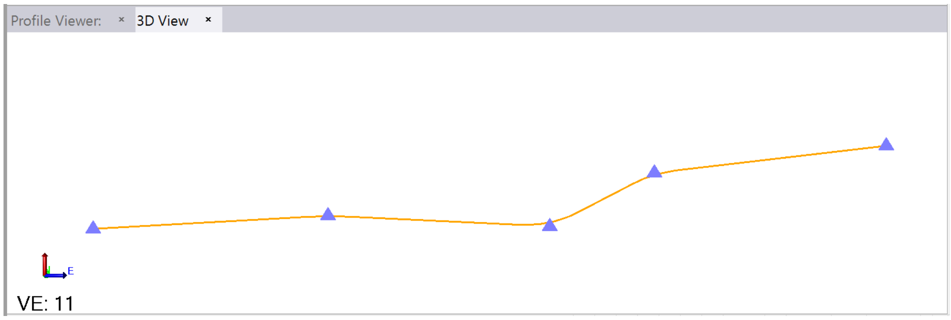
- A line feature is used to identify a feature with a linear shape, such as a fence or curb. When a line feature is processed in TBC, it is converted to a linestring that can be easily edited using the Edit Linestring command. If offset lines are specified for the line feature, additional linestrings are created. The line style used for the linestring is determined by the feature definition used to define the line feature. For more information, see View and Edit Line Feature Properties.
- Feature code processing settings allow you to create new linestrings when attributes change. If the linestring was derived from an imported line feature during feature code processing, feature property fields are displayed in the Properties pane, including the name of the feature and any attributes assigned to it.
- If the linestring was not derived from an imported line feature (it was created in TBC), feature property fields are displayed in the Properties pane only if you have imported feature definitions into your project (see the Import Feature Definition (.fxl) Files help). When the feature property fields are displayed, you can assign a feature to the linestring and specify attribute values as necessary.
- When a feature is assigned to a linestring, most of the other properties in the Properties pane inherit the values that are defined for the feature (for example, the line style). You can make changes to these properties as necessary. You can also add or change any value displayed in the feature property fields, including the feature name and feature attribute values. However, if the linestring was derived from an imported line feature and you reprocess feature codes in your project after making changes in the Properties pane, your changes will be lost.

2.3 | Displaying and exploring linestrings

- Use the **Toggle Line Marking** command to display or hide markers and labels for horizontal and vertical values along linestrings in 2D views to make viewing, understanding, and editing them easier. Markers are symbols that distinguish between horizontal segment end points, arc mid points, vertical control points , and the overall line's start and end points.



- Use the **Profile Viewer** command to check the geometry of a linestring. The viewer displays a vertical, graphic view of a single, selected linestring. You can also show one or more surface profiles where they coincide with the linesting's profile. Each linestring opens in its own viewer.



  See the Retrieve Video: View Linestring Profile (from v3.30 New Features)
- Use the **Explore Object** command to check linestrings by viewing geometric values calculated at specific locations based on the position of the cursor on the object in a graphic view. Certain values are displayed dynamically at the cursor as you move it along the object. Additional values are displayed statically in the command pane when you click in the view to specify a fixed location. Linestrings use the concept of 'distance along', which is analogous to stationing for alignments.
  See the Community question/answer: Display slope of a linestring...
- In drafting and plotting workflows, linestrings will display in paper space, such as a **Sheet View**. Create and display in the Plan View and Sheet View a table from selected line segments (linestring and polyline segments, rectangle and polygon segments, and arcs and circles) that automatically enumerates their attributes (for example, length, azimuth, and slope) based on the selected label table style.
- Use the **Create Text** command to identify a linestring as a **Smart text** target. Smart text can be used to derive one or more selected properties from the linestring.

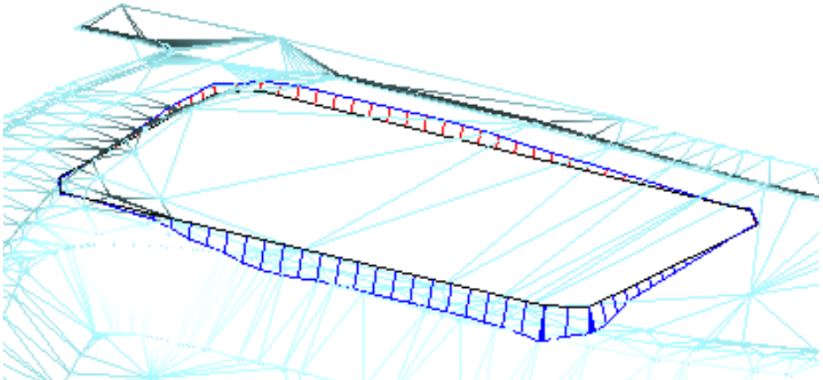
2.4 | Reporting on linestrings

- Use the **Area, Length, Count Report** to a generate a report showing the total number of linestrings in a selection, the length of each open line in the selection, the perimeter length of each closed line in the selection, the area of each closed polyline in the selection, and the total of those areas.
  See the YouTube video: Area Length Count Report
- Use the hidden* **Geometry Report** command to generate a Microsoft Excel spreadsheet of the geometry of one or more linestrings in your project. You can choose to include HAL and/or VAL control points, as well as delta station locations.
  *Type geometryreport in the Command Pane.

Chapter 3 | Using Linestrings

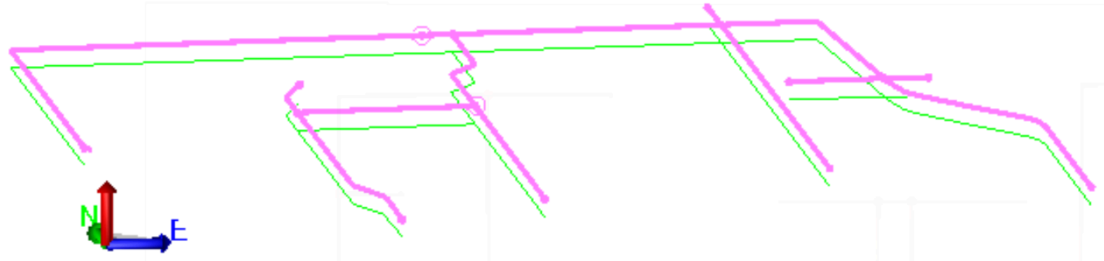
3.1 | Using linestrings in other commands

- **Superelevation Diagram** - Linestrings can be added to a superelevation diagram to add reference geometry.
- **Create Surface Tie** - Use a linestring as the reference line from which to project side slope connections (ties) up or down a surface at specific cut and fill cross-slopes. You can add more than one surface tie to a linestring; specify one as the start tie and one as the end tie in the Properties pane.



- **Create Breakline** - Often, you will create a breakline by adding linestring segments between points where surface triangles converge (at vertices), but you can also add simple breakline segments freely between any two locations on or near the surface. The addition, deletion, or any edit to a 3D line serving as a surface breakline will cause the entire surface to be reformed. Unlike more complex linestrings, simple breaklines do not use vertical control points; you can enter elevations for each new vertex or derive elevations from existing vertices.
- **Create Surface Edge Breakline** - When you create a breakline on the outer edge of a surface to prevent trimming or recalculated triangles on the current surface's edge, the breakline can subsequently be edited as a linestring.
- **Create Surface Intersection** - Create a point on a surface at a given bearing and vertical angle from a specific position (reference point). Specify multiple bearings and vertical angles to create a series of surface intersection points connected by a linestring. This is useful when you need to calculate a daylight line, such as from the top of a pad down to an existing ground surface.
- **Project Cleanup** - Select the Join lines with small gaps, overlaps, and misalignments option to merge two (and only two) lines that share an end point into one linestring. The lines must be on the same layer to be joined.

- **Create Alignment** - Create a HAL using the geometry of a line that you import or create. The geometry of the linestring is copied and appears as segments in the Alignment Editor. The first segment copied from the line (based on the line's direction) is appended to and placed after the last segment of the existing alignment.
  See the Retrieve video: Create Linestring
  See the Community question/answer: Re: Road Alignment Questions
- **Append Alignment** - Use the horizontal and/or vertical geometry of a linestring to extend an alignment. The geometry of the linestring is copied and appears as additional segments in the Alignment Editor. The first segment copied from the line (based on the line's direction) is appended to and placed after the last segment of the existing alignment.
- **Create Horizontal Alignment Label** - When you edit an individual alignment label tick mark, it is converted into a linestring that you edit in the Edit Linestring command.
- **Convert to Haul Road** - Linestrings can be converted into haul roads for site mass haul analyses. If the direction is wrong you can reverse the linestring's direction.
- **Create Utility Line** - Pressure systems are typically modeled with multiple line segments with nodes representing valves, bends, and junctions at intersections. You should first create one or more linestrings at the proper location and then use the Create Utility Line command to create a pipe at the same location. When using this command, the elevation of the linestring is used as the centerline of the utility line.



- **Media Files** (in Project Explorer) - When one or more media files are assigned to a linestring in your project (either in the field or in the office), a media folder is automatically created and attached to the linestring to contain the files. The media folder is created with the same name as the linestring. You can view any of the media files contained in the folder, add more files to the folder, and remove files from the folder as necessary.
- **Corridor Earthwork Report** - The report lists when a corridor model is using linestrings with vertical parabolic curves.
- **Drape Objects on Surface** - Vertically project linestrings onto a surface to create new 3D linestrings lying on the surface. For draped lines, the geometry is dependent on the location and shape of the originating linestring and the surface's topography.
- **Elevate Lines by Text, Point, and Intersection** - Elevate linestrings using a variety of objects that either intersect or lay nearby them. The elevations of text labels and their leader lines, points, and intersecting lines can be used to create and elevate vertical points of intersection (VPIs) along the lines. To easily identify which lines have been

elevated, set your zero-elevation data to an alternate color in the View Filter Manager and open a 3D View alongside the Plan View before using this command.

- **Create Corridor/Edit Corridor** - When you are creating or editing a corridor, you can add (or remove) linestrings as reference lines in the corridor (e.g., as an offset, to calculate directions, or to tie other template instructions to).
  Each of the reference lines in the corridor automatically creates a reference node in one or more cross-section templates used to define the corridor. You can base the location of a new node on a reference line node so that as the reference line offset or slope changes from one station to the next along the corridor, the offset or slope for the instruction node changes accordingly.
- **Create Contours/Create Quick Contours** - To smooth contours (which are linestrings), first explode them using the Explode command. Then use the Toggle Segments Straight/Curved option in the Edit Linestring command.

3.2 | Importing and exporting linestrings

3.2.1 | Exchanging linestrings with Trimble field data systems and other software

The linestring is a native TBC object type, so you will generally only export them, except when importing TBC Project Link files (.vcl). You can send linestrings to these other products:

- **SCS900**
- **GCS900** (SiteVision/AccuGrade/CAT Grade Control)
- **PCS900** (AccuGrade for Paving)
- **Trimble Connected Community**
- **VisionLink, Trimble Access**
- **SketchUp**

See their help systems for how linestrings are handled.

3.2.2 | Exchanging linestrings with other 3rd-party software

- **AutoCAD Civil 3D -** The Feature Line is most equivalent object to a TBC linestring. Before the Feature Line was introduced by Autodesk, users typically used a 3D Polyline, which is a simple CAD object. Some users may still to use the 3D Polyline.
- **Bentley ProjectWise** - When you import a native 3D Linear Element in an i-model (.icm.dgn) file from a Bentley civil software, the resulting Linear Entity 3D becomes a linestring in TBC.

Chapter 4 | Other

4.1 | Troubleshooting linestrings

- **Symptom:** You try to edit an arc, but you receive an error message saying "This object cannot be edited as a linestring because its UCS is rotated."
  - **Possible cause:** The object, which was imported from a CAD product that allows User-defined Coordinate Systems (UCSs) is referencing the UCS.
  - **Solution:** Remove the UCS from the object by opening the source file in a CAD program (such as the one the arc was created in); reset the coordinate system to the World Coordinate System (WCS), save the file, and re-import it into this program.
- **Symptom:** Offsetting a linestring results in inaccurate geometry.
  - **Possible cause:** When offsetting linestrings that contain vertical curves, the number of nodes generated in the offset line is insufficient to accurately model the offset line in 3D.
  - **Solution:** A new setting has been added in Project Settings > Computations > Surface > Breakline Approximation > Vertical tolerance. A setting of 0.01 is recommended.

4.2 | Known Issues and Bugs

- Imported linestrings with user-defined coordinate systems (UCSs):
  The user imported 3D CAD linework for a large highway project created from a Bentley InRoads DTM. After converting the lines to linestrings using the Elevate Lines command, there are some lines that came in with a UCS, making them uneditable (cannot break or trim them).
  See the Community question/answer: Editing Lines with UCS

4.3 | Selected linestring properties

With one or more linestrings selected, use the **Properties** pane to see and edit these unique and important properties that affect how the linestrings are formed and or displayed.

- **Line style scale** - Specifies the relative size of dashes, gaps, and symbols in a non-solid linestring. The higher the line style scale number, the larger the scale. If, for example, you have a dashed line with 12 dashes and you change the line style scale from 1.0 to 2.0, the line will then have 6 dashes. If your graphic view is showing lines at a small scale, some line styles display as solid to improve performance. You may have to zoom in to see a line's style.

- **Surface sharpness** - Defines how smoothly surface colors graduate at the line.
    - Select **Soft** to specify that shading for the line gradually changes, giving the visual appearance of a soft edge.
    - Select **Sharp** to specify that shading for the line abruptly changes, giving the visual appearance of a sharp edge.
    - Select **Sharp and texture boundary** to specify that the line be a drapeline as well.
- **Extend vertical** - Specifies that an interpolated elevation be used for the end segment in cases where elevations are specified for the preceding segments, but not the end segment.
- **Auto-close** - Forces the last line segment to automatically connect with the start point of the first line segment, creating a closed linestring.

4.4 | Keyboard shortcut

If you frequently create linestrings, use the **Command Shortcuts** command to set up a keyboard shortcut to launch the Create Linestring command; using the shortcut **Control + L** is recommended.