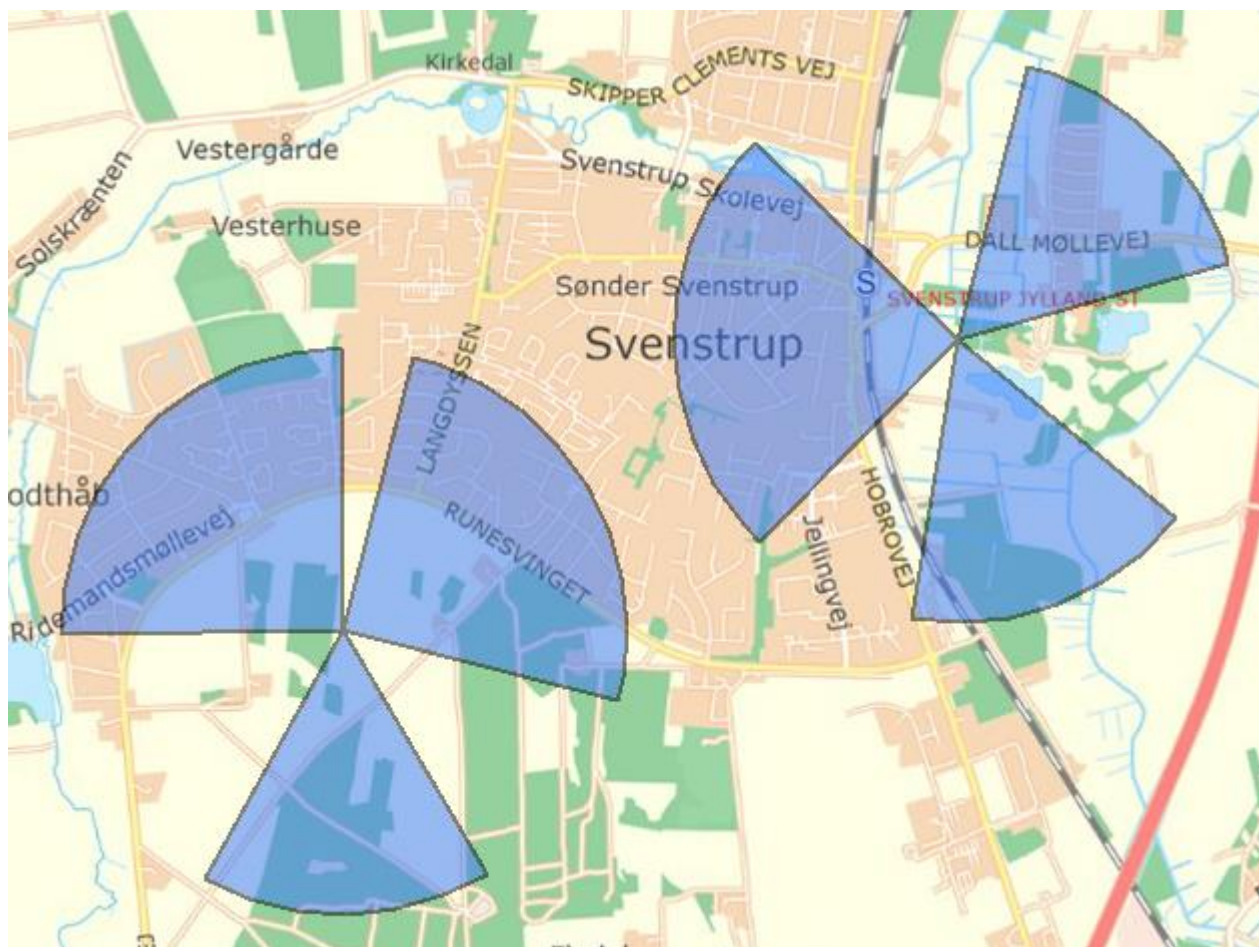precisely

DrawTools

For MapInfo Pro

October 2022

Installation

DrawTools is to be downloaded from the MapInfo Marketplace.

This distribution is automated so that you only have to click a button to get the add-in installed and loaded into MapInfo Pro.

If any updates to the tool gets published, you will see the small *Notification* symbol in the lower right corner of the MapInfo Pro window turn red. Double-click on the symbol to open the *Notification* window and from here access the updates from the MapInfo Marketplace where you easily can install the updates.

Please note that DrawTools requires MapInfo Pro v17.0.3 or newer to work.

Node Features

From the *Nodes* dropdown in the *Edit* group on the *Spatial* tab, you can find 6 features added by DrawTools.
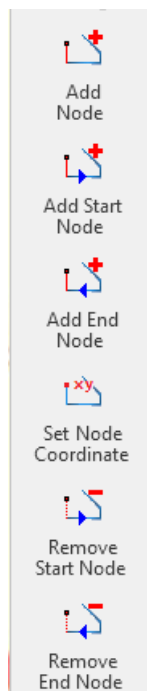
Add Node: Allows you to add a node to a selected polyline at the closest end to where you clicked.

Add Start Node: Allows you to add a node to a selected polyline at the start of the polyline.

Add End Node: Allows you to add a node to a selected polyline at the end of the polyline.

All these three tools work in a similar way. First, you need to select a polyline from the editable layer, then you click with the tool where you want the new start/end node to be. The tool will automatically add the new node to the polyline.

Set Node Coordinate: Allows you to change existing nodes of a polyline or polygon by entering coordinates for these. See more details below.

Remove Start Node: Allows you to delete the start node from the selected polyline.

Remove End Node: Allows you to delete the end node from the selected polyline.

Set Node Coordinate

To use this, you need to select a polyline or polygon from the editable layer.

Now select the *Set Node Coordinate* tool and click very close to the node you want to change. It's recommended to activate snap to find a matching node to where you click.

The tool will search for the nearest node in the selected object. If a node is found with the search distance, currently 10 meters, an error is shown.

If a node is found, the *Set Node Coordinate dialog* will be shown.

At the top of the dialog, you can see what segment and what node you currently are editing. In the map the current node is highlighted with a special symbol.

Enter the new coordinate for the node.

If you want to change more nodes, you can use the two arrows at the top right to move to the next or previous node. When you do, you can see that the changes to the polyline or polygon are rendered in the map with a new dotted line style. In this way, you can see the changes as you make them.

You can also delete the current node by clicking the *Delete* button on the dialog.

You can also use the *Zoom In* and *Zoom Out* button to control the zoom of the map window.
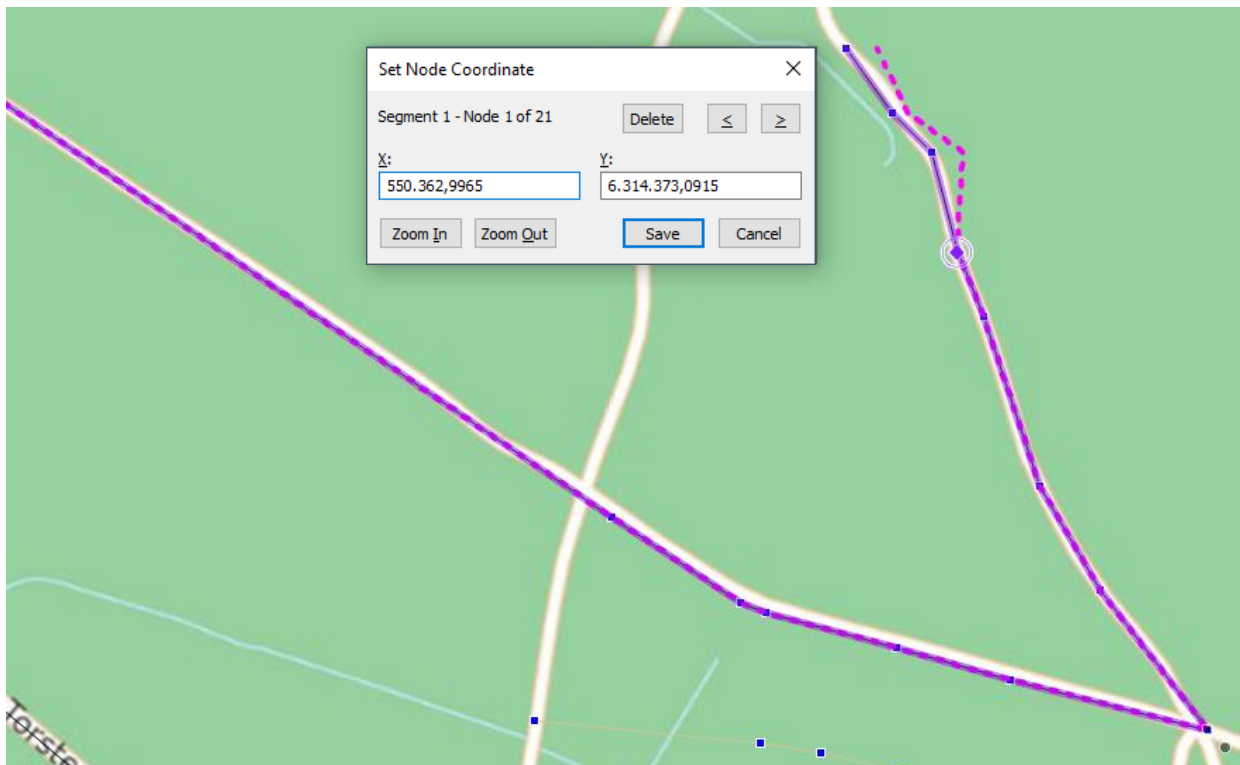
Hit *Save* to push your changes to the selected object.

Transform Features

From the *Transform* dropdown in the *Edit* group on the *Spatial* tab, you can find 4 features added by DrawTools.

Add to Region: Allows you to add area to an existing selected polygon. Make sure that the layer is editable. Draw the area/polygon that you want to add to the existing polygon. The drawn polygon will be merged/combined with the existing polygon.

Remove from Region: Allows you to add area to an existing selected polygon. Make sure that the layer is editable. Draw the area/polygon that you want to remove from the existing polygon. The drawn polygon will be used to erase from the existing polygon. This could be holes you want to cut or areas you want to remove.

Combine Selected into New: This allows you combine several selected objects and insert the new combined object into the editable layer. Note that the selected records will not be erased and that you can select from any layer, not just the editable layer.

Add Gaps: Allows you to create small polygons where you click in the map. These polygons will only be created if there is a gap in the polygons in the editable layer where you clicked. The gab will be created as a new polygon in the editable layer.

Insert Features

From the *Insert* dropdown in the *Create* group on the *Spatial* tab, you can find 2 features added by DrawTools.

Arrow: Allows you to draw an arrow in the editable layer based on the polygon you draw with the tool. See more below.

Isosceles Trapezoid: Allows you to draw an isosceles trapezoid in the editable layer using the line created with the tool. See more below.

Arrow

Make sure you have an editable layer in your map, then select the *Arrow* tool from the *Insert* dropdown on the *Spatial* tab.

This tool has a polyline draw mode which means you can draw a line with multiple nodes. Double-click to end the arrow and the *Arrow* dialog will pop up.

Now you can specify the size of your arrow.

Enter the width of the buffer around the polyline you have drawn and specify the resolution of the buffer.

For the two ends of the polyline, you can control how they should appear:

- Rounded
- Flat
- Arrow

If you choose *Arrow*, you also need to specify the with and length of the arrow. This is specified in percentage of the buffer width. 200 will make it twice the size of the buffer. These two values do not have to be the same.

The width must be larger than 100% and the length must be larger than 10%.
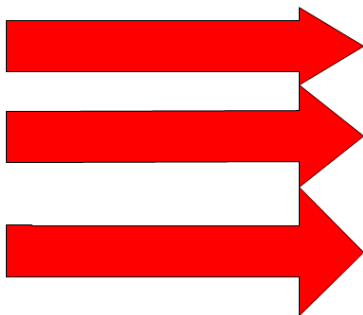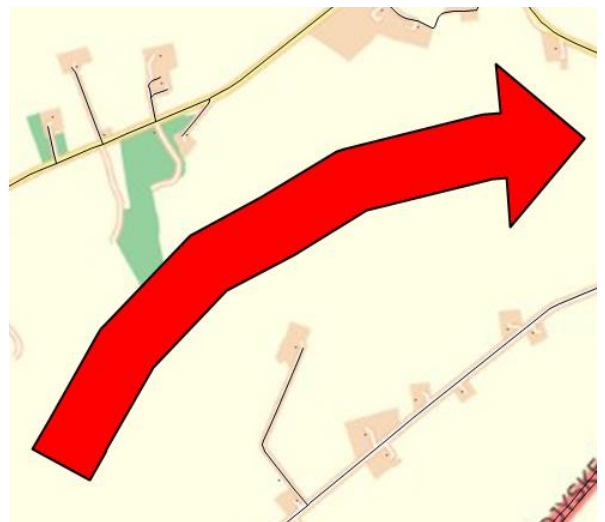
Here are 3 arrows with widths of 150, 200 and 250% for your reference. This might help you select the best width.

Isosceles Trapezoid

Make sure you have an editable layer in your map, then select the *Isosceles Trapezoid* tool from the *Insert* dropdown on the *Spatial* tab.

This tool has a line draw mode which means you can draw a simple line. Hold the left mouse button down, drag and release the mouse button and the *Isosceles Trapezoid* dialog will pop up.

The tool will use the start point of the line and the direction specified and populate these details into the dialog. The length of the line will also be populated into the dialog.

If you know the precise coordinates you can enter that.

You can also change the direction. Add 180 to or subtract 180 from the value to reverse the direction.

Specify the length if it is different from the length of the drawn line.

Specify the width of the trapezoid at the beginning and at the end.

You can use either a *Cartesian* or a *Spherical* calculation method. This depends on your data. For lat/long data, use *Spherical*. Otherwise use *Cartesian*.

Finally, you can choose to **Add Node at Center of parallel sides**. This makes it easier to find the center of the parallel sizes, for example of you want to add a center line to the trapezoid.

The result can look like this where the trapezoid shows an altitude zone (I think that's the term) at the end of a runway. For your information, I haven't used the actual size for such a zone but I think it gives you the idea

Functions

DrawTools also comes with several functions that can be used from the user interface in MapInfo Pro. These functions can be found in the *Functions* list in the *SQL Window,* and you can use them from the MapBasic window and from the Update dialog, too.

DTLineDirection

Calculates the direction of a line or polyline, East is zero, counter-clockwise. For polylines, the direction is calculated as the direction from the first to the last node.

```
Function DTDirection( oInput ) As Float
```
- *oInput*: A spatial object in the form of a line or polyline.

- Returns a float value as the direction of object in 360 degrees where East is 0 and the degrees grow counter-clockwise. Unsupported objects return -1,
- This function only works with projected coordinate systems

In the example below, we update an existing column (DIRECTION) with the direction of the (poly)lines of the table Highways.

```
Update Highways
  Set DIRECTION = DTLineDirection(OBJ)
```

I have mapped the result as a label expression on only longer lines to avoid the label clutter. Below you can see the values calculated for three lines that all seem to have a similar direction. One is however northbound (49°) where the other two (230° and 229°) are southbound.
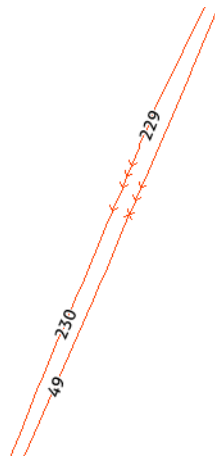
DTMath2NorthAngle

Converts a Math angle to a North/Compass angle. North angle has North as 0 and goes clockwise. A mathematical angle has 0 to the east and goes counter-clockwise. Bearings are often North-based angles and must be converted to mathematical angle for use in the functions in DrawTools.

```
Function DTNorth2MathAngle( fAngle ) As Float
```
- `fAngle`: The nortbased angle to be converted to a mathematical angle.
- Returns a float value in the form of the mathematical angle

In the example above we calculated the mathematical angle for some roads. If we need these values as north-based angles instead, we can convert these using the expression below

```
Update Highways
Set DIRECTION = DTMath2NorthAngle(DTLineDirection(OBJ))
```

DTNorth2MathAngle

Converts a North angle to a mathematical angle. North angle has North as 0 and goes clockwise. A mathematical angle has 0 to the east and goes counter-clockwise. Bearings are often North-based angles and must be converted to mathematical angle for use in the functions in DrawTools.

```
Function DTMath2NorthAngle( fAngle) As Float
```
- `fAngle`: The mathematical angle to be converted to a north-based angle.
- Returns a float value in the form of the north-based angle

If your angles are stored as north-based angles, you will have to convert these to math angles to be able to use them in the functions in DrawTools. The expressional below will do this conversion for you.

DTCreateCircularSector
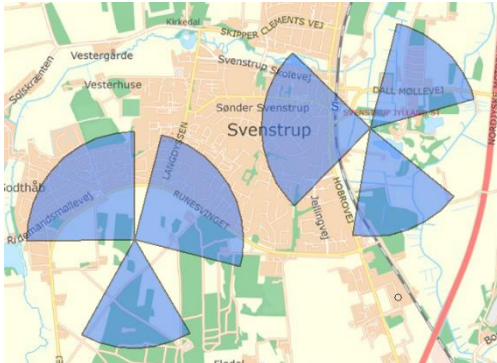
Create a circular sector using a point, direction, and sizes. If you are updating the spatial object of a table using this function, the parameters can either be fixed values or come from a column in the table.

The image here shows some circular sectors created with this function.

```
Function DTCreateCircularSector( oCenter, fDirection, fRadiusOuter, fAngleWidth, nResolution,
nCalculationMethod ) As Object
```

- *oCenter*: The center or starting point of the circular sector in the form of a spatial point.
- *fDirection*: The direction of the circular sector measured using a mathematical angle
- *fRadiusOuter*: The outer radius of the circular sector, in meters
- *fAngleWidth*: The width of the circular sector measured in angles
- *nResolution*: The resolution for the buffer used. Specifies the number of points to describe a full circle
- *nCalculationMethod*: Specifies if the calculations should use Cartesian (1) or Spherical (2) methods. For lat/long based coordinates, use Spherical (2), and for projected coordinates, use Cartesian (1).
- Returns a spatial object in the form of a circular sector using the specified values

In the example below, I am creating circular sector from a table with existing points.

```
Set CoordSys Table CellTowerPoints
Update CellTowerPoints
Set OBJ = DTCreateCircularSector(OBJ, DTNorth2MathAngle(BEARING), 900, SPREAD_ANGLE, 36, 2)
```

You can also use lat/long values from columns as input instead of the OBJ column.

```
Set CoordSys Table CellTowerPoints
Update CellTowerPoints
Set OBJ = DTCreateCircularSector(CreatePoint(LONG, LAT), DTNorth2MathAngle(BEARING), 900,
SPREAD_ANGLE, 36, 2)
```

DTCreateAnnulus
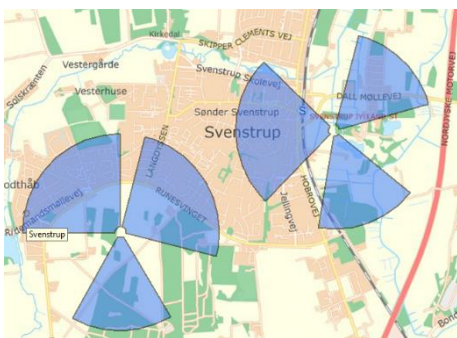
Creates an annulus using a point, direction, and sizes. An annulus is basically a circular sector where the center has been removed. If you are updating the spatial object of a table using this function, the parameters can either be fixed values or come from a column in the table.

The image here shows some circular sectors created with this function. Notice how the center has been removed compared to the circular sectors above.

```
Function DTCreateAnnulus( oCenter, fDirection, fRadiusInner, fRadiusOuter, fAngleWidth,
nResolution, nCalculationMethod ) As Object
```

- *oCenter*: The center or starting point of the annulus in the form of a spatial point.
- *fDirection*: The direction of the circular sector measured using a mathematical angle
- *fRadiusInner*: The inner radius of the annulus, in meters
- *fRadiusOuter*: The outer radius of the annulus, in meters
- *fAngleWidth*: The width of the annulus measured in angles
- *nResolution*: The resolution for the buffer used. Specifies the number of points to describe a full circle
- *nCalculationMethod*: Specifies if the calculations should use Cartesian (1) or Spherical (2) methods. For lat/long based coordinates, use Spherical (2), and for projected coordinates, use Cartesian (1).
- Returns a spatial object in the form of a annulus using the specified values

```
Set CoordSys Table CellTowerPoints
Update CellTowerPoints
Set OBJ = DTCreateAnnulus(OBJ, DTNorth2MathAngle(BEARING), 50, 900, SPREAD_ANGLE, 36, 2)
```