MapCAD Tools executable via Exec() function

MapInfo Pro 2019 – MapCAD 2019 Build 34

CreateParallel

Purpose

Creates a parallel object given a source object and a distance; corresponds to choosing **Spatial > Edit > Buffer > Create Parallel**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","CreateParallel"
     ,SourceObject, Distance, Orientation, CloseObject, MakeRegion)
```

Return Value

Line, polyline or region object, depending on source object type and calling parameters. If the function failed the return object is a point(0,0).

Description

Use the **CreateParallel** function to create a new object parallel to an existing object given a distance. The parallel orientation always refers to the line direction of the source object. Parallels can be generated on one or both sides. Optionally, they can be closed or converted to regions. For regions, the orientation indicates the inside or outside parallel.

The MapCAD tool must be loaded. This function requires MapCAD and an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
SourceObject	Object value, consits a line, polyline or region object.
Distance	Float value, specifies the distance between the source object and the new parallel object. Distance unit is the current map distance unit.
Orientation	Integer value, specifies the figure according to the line direction of the source object. <table><tr><td>Line and Polyline</td><td>Region</td></tr><tr><td>1 = Left side 2 = Right side 3 = Both sides</td><td>1 = Inside 2 = Outside</td></tr></table>
CloseObject	Logical value, indicating whether the endpoints of the new object are connected to the endpoints of the source object. This option is only useful for line or polyline source objects.
MakeRegion	Logical value, indicating whether the new object will be converted to a region object. This option is only useful for line or polyline source objects and includes the option CloseObject=true.

Examples

Create a left side parallel object to a line or polyline object:

'OldObj is a line or polyline object

```
NewObj = Exec("MapCAD.mbx","CreateParallel", OldObj, 2.5, 1, "F", "F")
```

Create a both side parallel object to a line or polyline object, close endpoints, combine and convert to region:

'OldObj is a line or polyline object

```
NewObj = Exec("MapCAD.mbx","CreateParallel", OldObj, 2.5, 3, "T", "T")
```

Create a inside parallel object to a region object:

'OldObj is a region object

```
NewObj = Exec("MapCAD.mbx","CreateParallel", OldObj, 2.5, 1, "F", "F")
```

CalcAngle

Purpose

Calculate the angle between two lines; corresponds to choosing **Spatial > Edit > Calculate > Calculate Angle**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","CalcAngle", LineObj1, LineObj2)
```

Return Value

Float
If the function failed the return value is 999.

Description

Use the **CalcAngle** function to calculate the angle between two lines in degrees. The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
LineObj1	Object value, Line
LineObj2	Object value, Line

Example

Calculate the angle between two lines.

```
Dim t as float
t = Exec("MapCAD.mbx","CalcAngle",LineObj1, LineObj2)
note "Angle = " + t
```

CalcDirection

Purpose

Calculates the mathematically direction of a line object; corresponds to choosing **Spatial > Edit > Calculate > Calculate Direction**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

Exec("MapCAD.mbx","CalcDirection", *LineObj*)

Return Value

Float
If the function failed the return value is 999.

Description

Use the **CalcDirection** function to calculate the mathematically direction angle of a line object in degrees. This angle is defined by the slope of the line, so the direction of the line is taken into account. A directional angle is determined counterclockwise with respect to the x-axis.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
LineObj	Object value, Line

Example

Calculate the direction angle of a line.

```
Dim t as float
t = Exec("MapCAD.mbx","CalcDirection",LineObj)
note "Direction Angle = " + t
```

CartesianDirection / SphericalDirection

Purpose

Calculates the mathematically direction of a linear object; corresponds to choosing **Spatial > Edit > Calculate > Calculate Direction**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro also without Exec().

Syntax

```
Exec("MapCAD.mbx","CartesianDirection", LineObj)
Exec("MapCAD.mbx","SphericalDirection", LineObj)
```

Return Value

Float. If the function failed the return value is 999.

Description

Use the **CartesianDirection** and **SphericalDirection** functions to calculate the mathematically direction angle of a linear object in degrees. This angle is defined by the slope of the line, so the direction of the line is taken into account. A directional angle is determined counterclockwise with respect to the x-axis. Only line objects and polyline objects with two points are supported. Polylines with more than two nodes have no uniform direction.

The MapCAD tool must be loaded. These functions requires no active map window. Notice: The calculation of the angle is based on the session projection.

Parameters	Description
LineObj	Object value, Line or Polyline with two points

Example

```
Dim t as float
Set CoordSys …
t = Exec("MapCAD.mbx","CartesianDirection",LineObj)
note "Direction = " + t
```

ReverseLineDirection

Purpose

The line direction of all selected lines and polylines will be reversed; corresponds to choosing **Spatial > Edit > Modify > Create Parallel**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","ReverseLineDirection")
```

Return Value

Integer

Description

Use the **RevLineDirection** function to reverse the line direction of line and polyline objects. All objects of the current selection will be changed. Objects that do not have a suitable object type will be ignored. The return value of the function is the number of actually changed objects.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units.

Example

Reverse line direction of all selected lines or polylines.

```
Dim i as integer
i = Exec("MapCAD.mbx","ReverseLineDirection")
note "Objects changed: " + i
```

SplitToLines

Purpose

Split all selected polyline and region objects into individual lines; corresponds to choosing **Spatial > Edit > Disaggregate > Disaggregate To Lines**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","SplitToLines")
```

Return Value

Integer

Description

Use the **SplitToLines** function to split all selected polyline and region objects into individual lines. All objects of the current selection will be changed. Objects that do not have a suitable object type will be ignored. The return value of the function is the number of actually changed source objects.

The MapCAD tool must be loaded. This function requires no active map window.

Example

Split all selected polyline and region objects into individual lines.

```
Dim i as integer
i = Exec("MapCAD.mbx","SplitToLines")
note "Objects changed: " + i
```

ConvertToLine

Purpose

Convert all selected polylines with two nodes to line objects; corresponds to choosing **Spatial > Edit > Covert Polyline To Line**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","ConvertPolylineToLine")
```

Return Value

Object

Description

Use the **ConvertPolylineToLine** function to convert all selected polylines with two nodes to line objects. All objects of the current selection will be changed. Objects that do not have a suitable object type or more than one nodes will be ignored. The return value of the function is the number of actually changed source objects.

The MapCAD tool must be loaded. This function requires no active map window.

Example

Convert all selected polylines with two nodes to line objects

```
Dim i as integer
i = Exec("MapCAD.mbx","ConvertPolylineToLine")
note "Objects changed: " + i
```

LineIntersectionPoint

Purpose

Create an intersection point of two lines; corresponds to choosing **Spatial > Edit > Modify > Line Intersection**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

`Exec("MapCAD.mbx","LineIntersectionPoint", LineObj1, LineObj2)`

Return Value

Point Object
If the lines do not intersect or the function failed, the return object is a point(0,0).

Description

Use the **LineIntersectionPoint** function to create the intersection point of two lines. The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
LineObj1	Object value, Line
LineObj2	Object value, Line

Example

Create an intersection point of two lines.

```
Dim NewObj as object
NewObj = Exec("MapCAD.mbx","LineIntersectionPoint",LineObj1, LineObj2)
```

CreateSpline

Purpose

Create a spline given a polyline or region; corresponds to choosing **Spatial > Edit > Modify > Spline**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx"," CreateSpline", Tolerance, SourceObj)
```

Return Value

Polyline or Region object, depending on the type of *th*e source object.
If no spline can be calculated or the function failed, the return object is a point(0,0).

Description

Use the **CreateSpline** function to create a polyline or region that is mapped from a calculated spline.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
Tolerance	Float value, specifies the maximum divergence for calculating the curvature. The smaller the value, the more precise the curve, but higher the number of nodes. Default value = 0.5.
SourceObj	Object value, Polyline or Region

Example

Create a polyline or region as a spline.

```
Dim NewObj as object
NewObj = Exec("MapCAD.mbx","CreateSpline",0.5, SourceObj)
```

CreateScaledObject

Purpose

Enlarg or reduce objects with a scaling factor; corresponds to choosing **Spatial > Edit > Modify > Scale**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","CreateScaledObject", SourceObj, ScaleFactor, Anchor)
```

Return Value

Depending on the type of the source object.
If no scaled object can be created or the function failed, the return object is a point(0,0).

Description

Use the **CreateScaledObject** function to scale (larger/smaller) a source object by a multiplication factor. The scaling is in reference to the anchor point.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
SourceObj	Object value, All types
ScaleFactor	Float value, specifies the multiplication factor. For example, a factor of 0.5 halves, a factor of 2 doubles.
Anchor	X , Y 1 = left,top 2 = center, top 3 = right, top 4 = left, center 5 = center, center 6 = right, center 7 = left, bottom 8 = center, bottom 9 = right, bottom

Example

Create a new scaled object.

```
Dim NewObj as object
NewObj = Exec("MapCAD.mbx","CreateScaledObject", SourceObj ,0.5,5)
```

MirrorHorizontal

Purpose

Mirror objects on its horizontal central axis; corresponds to choosing **Spatial > Edit > Transform > Mirror Horizontal**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

`Exec("MapCAD.mbx","MirrorHorizontal")`

Return Value

Logical
If an object can not be mirrored or the function failed, the return value=false.

Description

Use the **MirrorHorizontal** function to mirror all selected objects on their common horizontal center axis.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Example

Mirror selected objects horizontally.

```
Dim Result as logical
Result = Exec("MapCAD.mbx","MirrorHorizontal")
```

MirrorVertical

Purpose

Mirror objects on its vertical central axis; corresponds to choosing **Spatial > Edit > Transform > Mirror Horizontal**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","MirrorVertical")
```

Return Value

Logical
If an object can not be mirrored or the function failed, the return value=false.

Description

Use the **MirrorVertical** function to mirror all selected objects on their common vertical center axis.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Example

Mirror selected objects vertically.

```
Dim Result as logical
Result = Exec("MapCAD.mbx","MirrorVertical")
```

FitObjects

Purpose

Fits objects to its neighboring objects; corresponds to choosing **Spatial > Edit > Fix/Clean > Fit Objects**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","FitObjects",

Tolerance, DistanceUnit, ReferenceLayer, AddNodes)
```

Return Value

Logical
If an object can not be fitted or the function failed, the return value=false.

Description

Use the **FitObjects** function to fit all selected objects to its neighboring objects in a reference layer. Each selected object is considered, node by node, with neighboring nodes and edges from reference objects.

If close to the source node a reference node is found, the source node is moved to the reference node. If close to the source node a matching edge (without nodes) is found, the source node is aligned to the reference edge. In addition, a new node can be inserted into all found reference edges at the matching point.

The MapCAD tool must be loaded. This function requires no active map window.

Parameters	Description
Tolerance	Float value, circle radius around each node to search for matching reference object nodes and edges.
DistanceUnit	String value, distance unit for geographic operations (eg, "mi") according to the *Set distance units* statement.
ReferenceLayer	String value, layer name of the layer that contains the neighboring objects.
AddNodes	Logical value, If TRUE and the reference layer is writable, new nodes are inserted into all found reference edges at the match point.

Example

Fit all selected objects to its neighboring objects in layer *Lakes*. The maximum distance between the nodes or edges is 0.03 miles. New nodes are inserted at the matching points into all reference edges.

```
Dim Result as logical
Result = Exec("MapCAD.mbx","FitObjects",0.03,"mi","Lakes","T")
```

RotateText

Purpose

Rotate a text object; corresponds to choosing **Spatial > Edit > Transform > Rotate Text**. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","RotateText", OldObj, Angle, Method)
```

Return Value

Text Object
If the function failed the return object is a point(0,0).

Description

Use the **RotateText** function to rotate a text object by giving a rotation angle and a method.

Parameters	Description
OldObj	Object value, Text
Angle	Float value. A positive angle rotates counterclockwise, a negative angle rotates clockwise.
Method	Integer value. 1 = Additiv; The given rotation angle is added to the current rotation angle of the text object. 2 = Absolutely; The current rotation angle of the text object is replaced by the given rotation angle.

Example

Rotate a selected text object.

```
Dim NewObj as object
fetch first from selection 'must be a text object
NewObj = Exec("MapCAD.mbx","RotateText", selection.obj, 45, 1)
update selection set obj = NewObj where rowid=1
```

CircleFromPointsXY

Purpose

Creates a new circle object; corresponds to choosing
Spatial > Create > Insert > Circle From 3 Points. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","CircleFromPointsXY", X1, Y1, X2, Y2, X3, Y3)
```

Return Value

Circle Object
If the function failed the return object is a point(0,0).

Description

Use the **CircleFromPointsXY** function to create a circle object by giving three point coordinates.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
X1, Y1, X2, Y2, X3, Y3	Float values

Example 1

Creates a new circle object.

```
Dim NewObj as object

NewObj = Exec("MapCAD.mbx","CircleFromPointsXY",
434204.02,5566829.13,434219.39 ,5566833.997,434218.18,5566852.19)

Insert into [Table] (obj) values (NewObj)
```

Example 2

Updates objects with calculated circles from coordinate columns (X1 ... Y3).

```
Update [Table] set obj=Exec("MapCAD.mbx","CircleFromPointsXY", X1, Y1,
X2, Y2, X3, Y3)
```

ArcFromPointsXY

Purpose

Creates a new arc-polyline object; corresponds to choosing
Spatial > Create > Insert > Arc From 3 Points. This function requires the MapCAD Tool and can be executed via the Exec() command. You can issue this statement from the MapBasic Window in MapInfo Pro.

Syntax

```
Exec("MapCAD.mbx","ArcFromPointsXY", X1, Y1, X2, Y2, X3, Y3)
```

Return Value

Polyline Object (! not Arc)
If the function failed the return object is a point(0,0).

Description

Use the **ArcFromPointsXY** function to create an arc-polyline object by giving three point coordinates.

The MapCAD tool must be loaded. This function requires an active map window to determine the projection and units. To set the calculation accuracy, open the MapCAD settings dialog.

Parameters	Description
X1, Y1, X2, Y2, X3, Y3	Float values

Example 1

Creates a new arc-polyline object.

```
Dim NewObj as object

NewObj = Exec("MapCAD.mbx","ArcFromPointsXY",
434204.02,5566829.13,434219.39,5566833.997,434218.18,5566852.19)

Insert into [Table] (obj) values (NewObj)
```

Example 2

Updates objects with calculated arc-polylines from coordinate columns (X1 ... Y3).

```
Update [Table] set obj=Exec("MapCAD.mbx","ArcFromPointsXY", X1, Y1, X2,
Y2, X3, Y3)
```