

POLYLINE EDIT

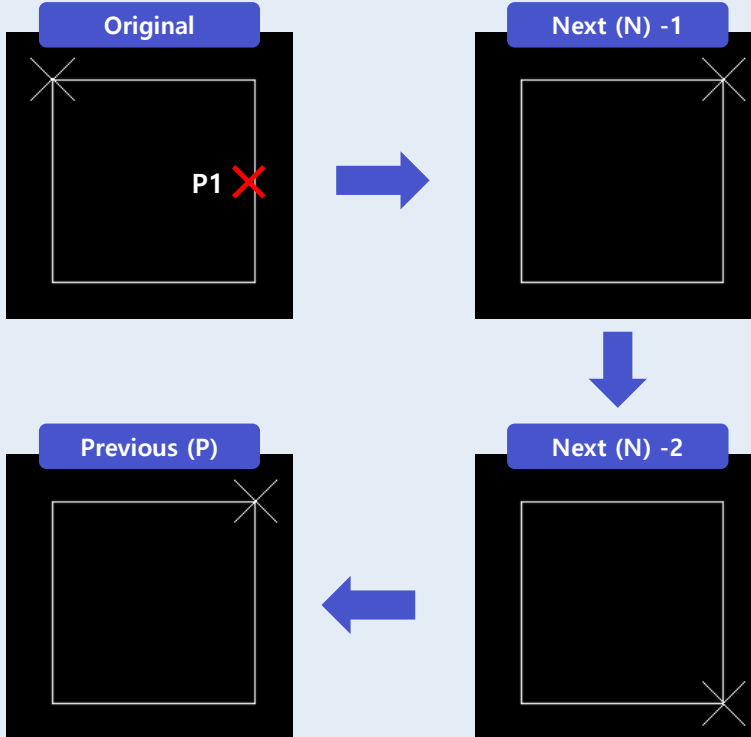
CADian 2020



Edit vertices – NEXT/PREVIOUS

command : pe(pedit)

Edit vertices – Next/Previous



Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline: ~ /Undo(U) /<eXit>:" e input and SPACE (Edit vertice)
4. "Edit vertices : ~ /eXit(X) /<Next>:" n input and SPACE (Next 2 times)
5. "Edit vertices : ~ /eXit(X) /<Next>:" p input and SPACE (Previous)
6. "Edit vertices : ~ /eXit(X) : ~ <Previous>:" x input and SPACE (Exit Edit vertices)
7. "Edit polyline : ~ <eXit(X)>:" x input and SPACE (Exit Edit polyline)

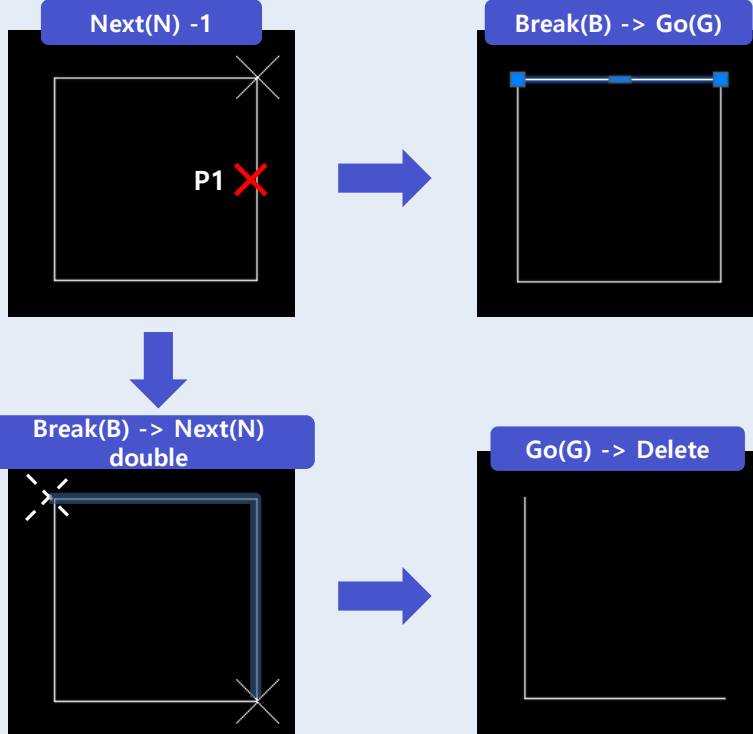
NOTE

- Step2, select multiple entities when selecting multiple(M)
- When you select a line or arc in step2, a polyline convert message appears at the command prompt
- After the last move, you have to go to Previous to go to the first vertex

Edit vertices – BREAK

command : pe(pedit)

Edit vertices (Break)



Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~ /eXit(X) /<eXit>:" e input and SPACE (Edit vertice)
4. "Edit vertices : ~ eXit(X) <Next>:" n input and SPACE (Next)
5. "Edit vertices : ~ eXit(X) <Next>:" b input and SPACE (Break)
6. "Break : ~ Go(G) /eXit(X) <Next>:" g input and SPACE (Go)
7. "Edit vertices : ~ eXit(X) <Next>:" x input and SPACE (Exit edit vertices)
8. "Edit polyline : ~ <eXit(X)>:" x input and SPACE (Exit edit polyline)

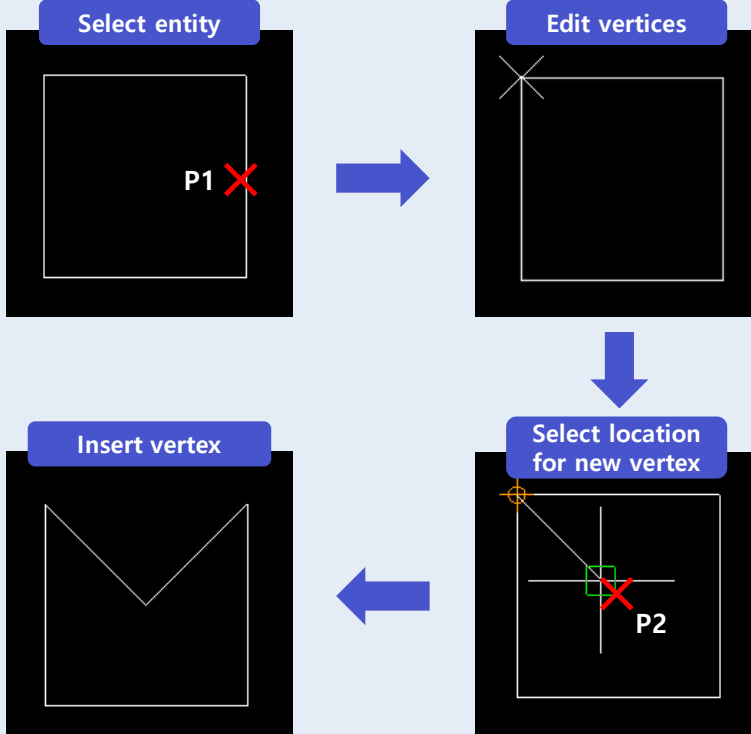
NOTE

- In step6, If you select Go, break is executed.

Edit vertices – INSERT

command : pe(pedit)

Edit vertices (Insert)



Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~/eXit(X)/<eXit>:" e input and SPACE (Edit vertice)
4. "Edit vertices :~eXit(X)<Next>:" i input and SPACE (Insert vertex)
5. "Location for new vertex: " P2 CLICK
6. "Edit vertices :~eXit(X)<Next>:" x input and SPACE (Exit edit vertices)
7. "Edit polyline : ~<eXit(X)>:" x input and SPACE(Exit edit polyline)

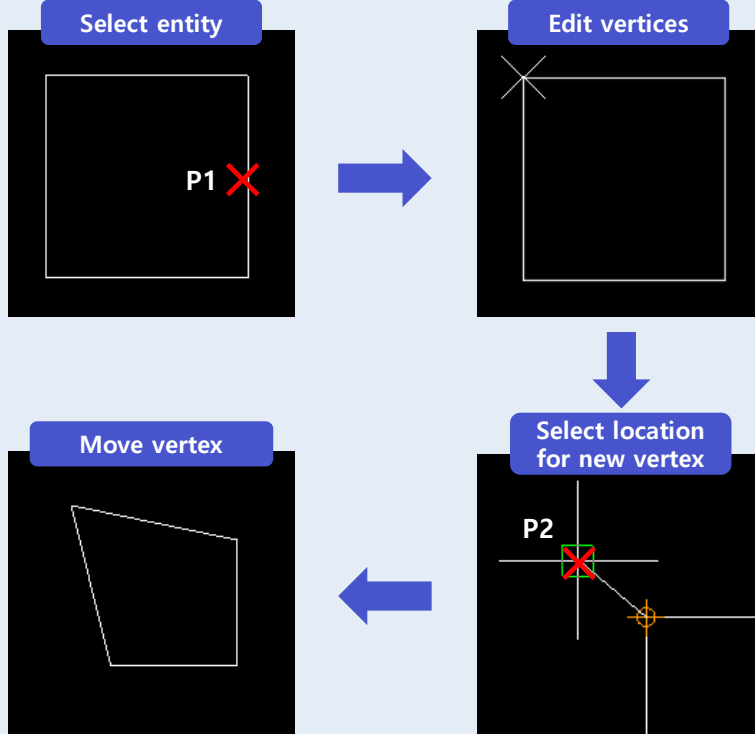
NOTE

-. If an insert is performed while the vertex is displayed, it is inserted at the next point

Edit vertices – MOVE

command : pe(pedit)

Edit vertices (Move)



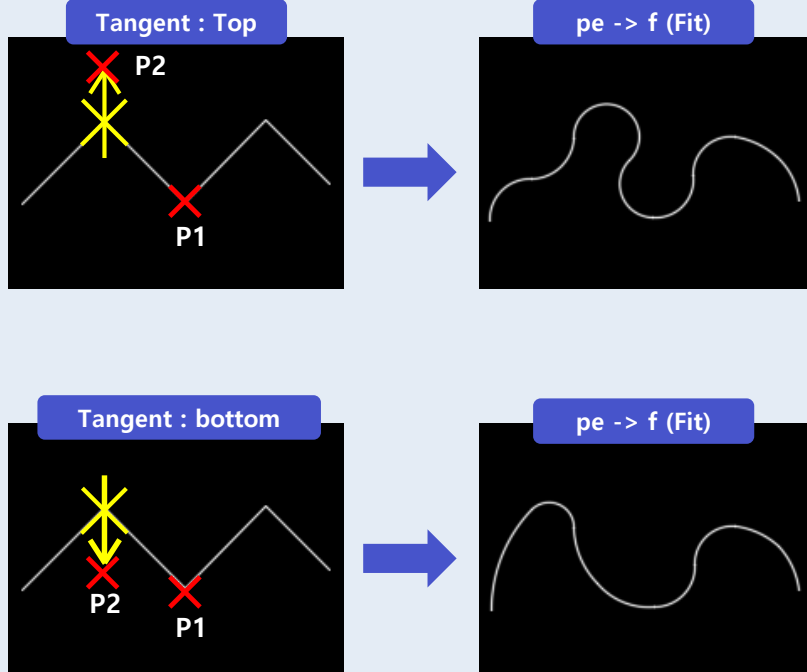
Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : "
P1 Select and SPACE
3. "Edit polyline : ~ /eXit(X) /<eXit>:"
e input and SPACE (Edit vertice)
4. "Edit vertices : ~ eXit(X) <Next>:"
m input and SPACE (Move vertex)
5. "New location for vertex : "
P2 Select and SPACE
6. "Edit vertices : ~ eXit(X) <Next>:"
x input and SPACE (Exit edit vertices)
7. "Edit polyline : ~ <eXit(X)>:"
x input and SPACE (Exit edit polyline)

Edit vertices – TANGENT

command : pe(pedit)

Edit vertices (Tangent)



Execution Process

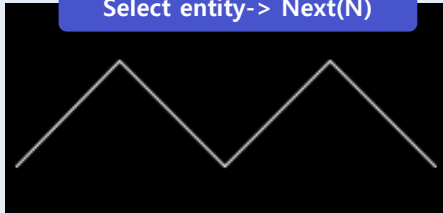
1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~ /eXit(X) /<eXit>:" e input and SPACE (Edit vertice)
4. "Edit vertices : ~ eXit(X) <Next>:" t input and SPACE (Tangent vertex)
5. "Specify direction of vertex tangent:" P2 Select and SPACE
6. "Edit vertices : ~ eXit(X) <Next>:" x input and SPACE (Exit edit vertices)
7. "Edit polyline : ~ <eXit(X)>:" f input and SPACE (Fit polyline)
8. "Edit polyline : ~ <eXit(X)>:" x input and SPACE (Exit edit polyline)

Edit vertices – STRAIGHTEN

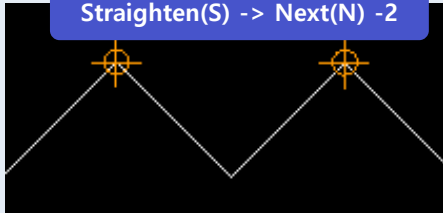
command : pe(pedit)

Edit vertices (Straighten)

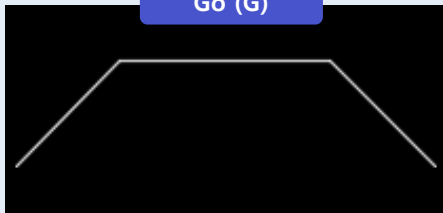
Select entity-> Next(N)



Straighten(S) -> Next(N) -2



Go (G)



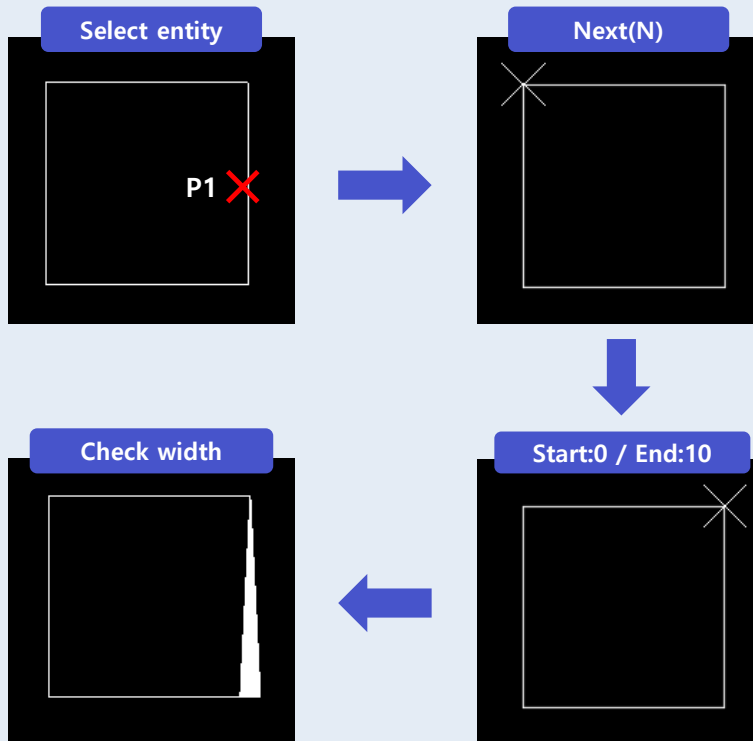
Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : "
P1 Select and SPACE
3. "Edit polyline : ~ /eXit(X) /<eXit>:"
e input and SPACE (Edit vertices)
4. "Edit vertices : ~ eXit(X) <Next>:"
n input and SPACE (Next)
5. "Edit vertices : ~ eXit(X) <Next>:"
s input and SPACE (Straighten)
6. "Straighten : ~ <Next>:"
n input and SPACE (Next 2 times)
7. "Straighten : ~ <Next>:"
g input and SPACE (Go)
8. "Edit vertices : ~ eXit(X) <Next>:"
x input and SPACE (Exit edit vertices)
9. "Edit polyline : ~ <eXit>:"
x input and SPACE (Exit edit polyline)

Edit vertices – WIDTH

command : pe(pedit)

Edit vertices (Width)



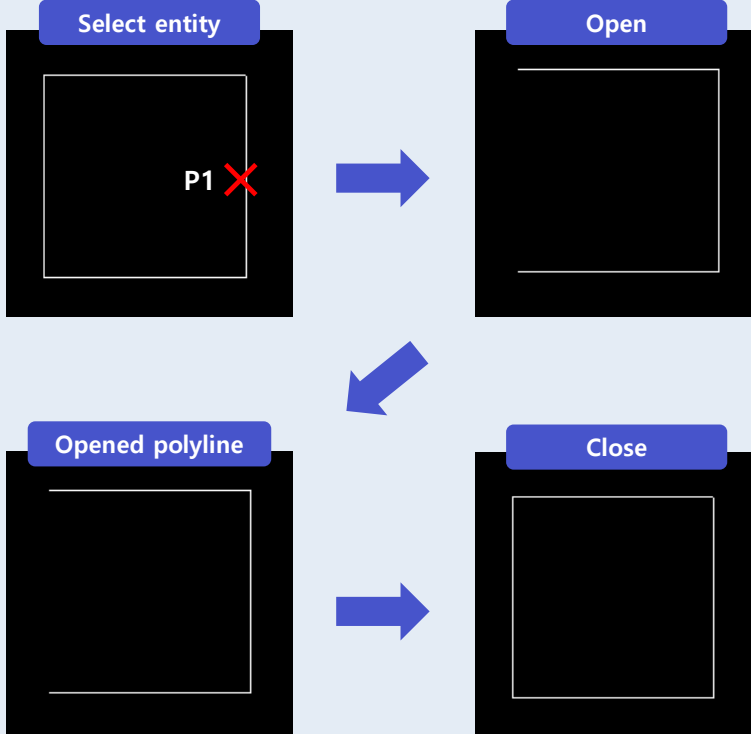
Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~ /eXit(X) /<eXit>:" e input and SPACE (Edit vertices)
4. "Edit vertices : ~ eXit(X) <Next>:" n input and SPACE (Next 2times)
5. "Edit vertices : ~ eXit(X) <Next>:" w input and SPACE (Width)
6. "Enter starting width <0>:" 0 input and SPACE (Starting width)
7. "Enter ending width <0>:" 10 input and SPACE (Ending width)
8. "Edit vertices : ~ eXit(X) <Next>:" x input and SPACE (Exit edit vertices)
9. "Edit polyline : ~ <eXit>:" x input and SPACE (Exit edit polyline)

Edit polyline - OPEN / CLOSE

command : pe(pedit)

Open / Close

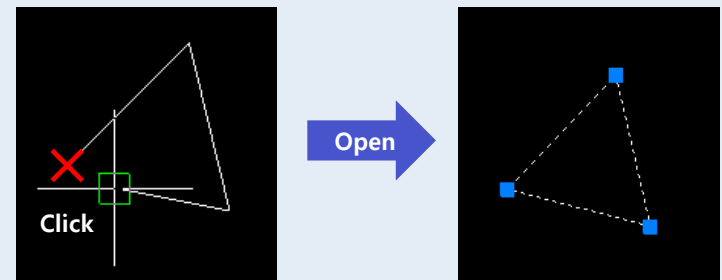


Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~ <eXit(X)>:"
 - o input and SPACE (Open)
4. "Edit polyline : ~ <eXit(X)>:"
 - c input and SPACE (Close)
5. "Edit polyline : ~ <eXit(X)>:"
 - x input and SPACE (Exit edit polyline)

NOTE

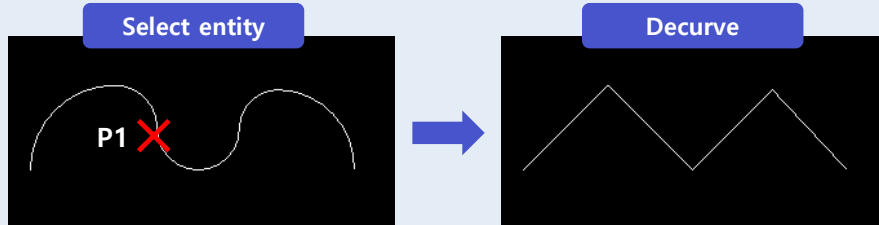
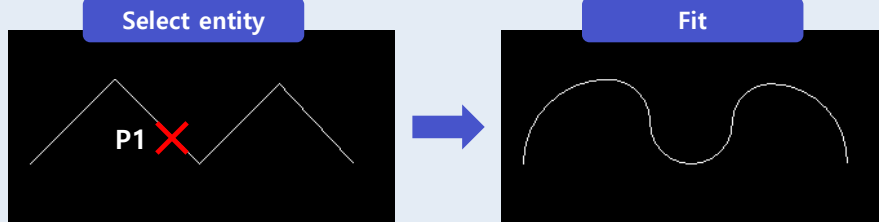
- . If you do not perform a close when creating a polyline, the Open function does not work



Edit polyline - FIT / DECURVE

command : pe(pedit)

Fit / Decurve



Execution Process (Fit)

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~ <eXit(X)>:" f input and SPACE (Fit)
4. "Edit polyline : ~ <eXit(X)>:" x input and SPACE (Exit edit polyline)

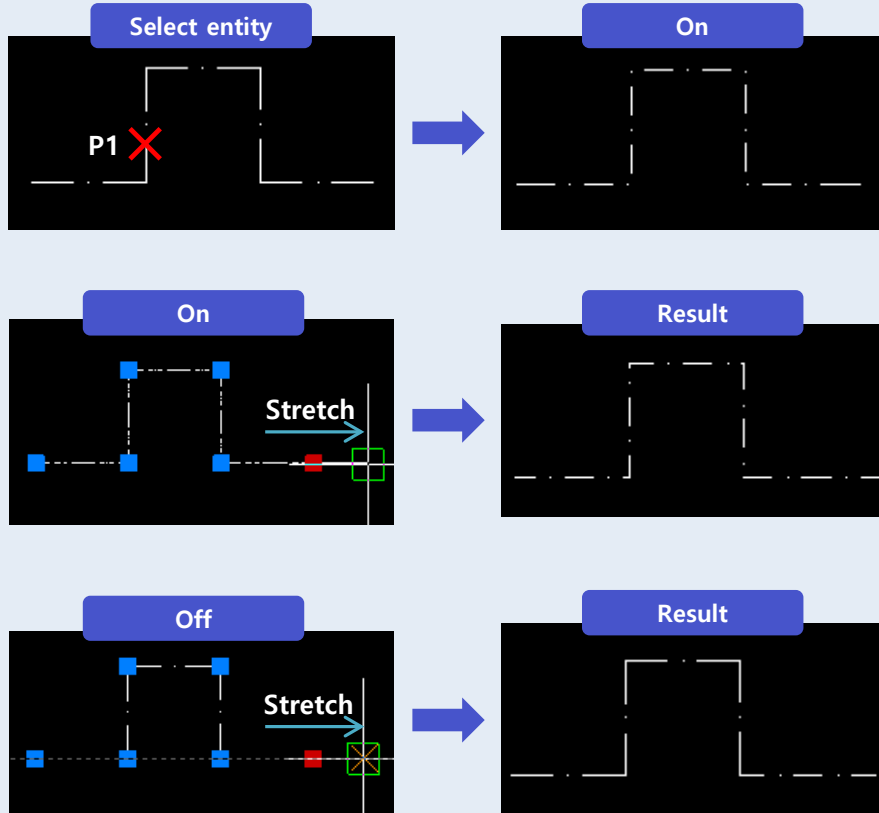
Execution Process (Decurve)

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~ <eXit(X)>:" d input and SPACE (Decurve)
4. "Edit polyline : ~ <eXit(X)>:" x input and SPACE (Exit edit polyline)

Edit polyline – LINETYPE-MODE

command : pe(pedit)

Linetype-mode



Execution Process (Linetype-mode)

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : " P1 Select and SPACE
3. "Edit polyline : ~ <eXit(X)>:" | input and SPACE (Linetype-mode)
4. "~ along polyline: ON/OFF <OFF>:" on input and SPACE (On)
5. "Edit polyline : ~ <eXit(X)>:" x input and SPACE (Exit edit polyline)

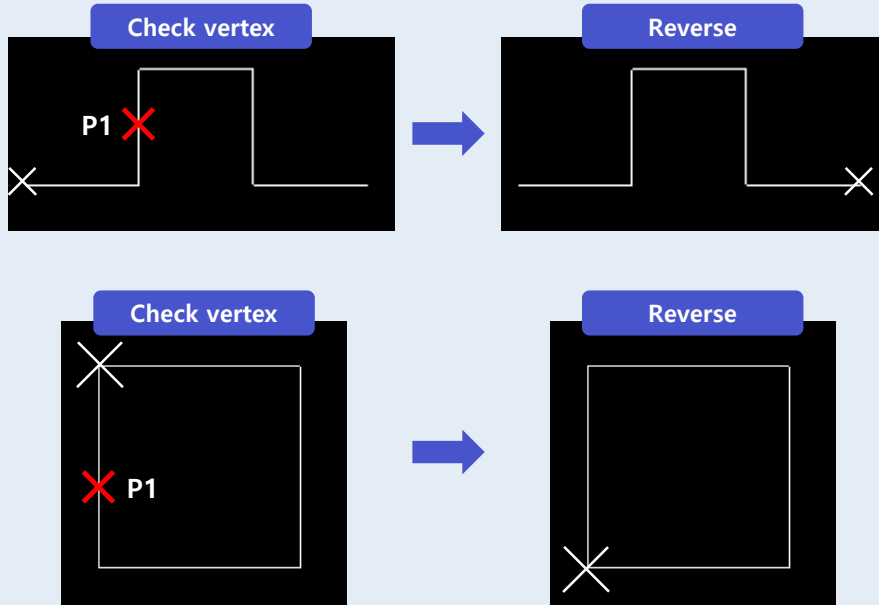
NOTE

- . If you turn on the linetype, confirm that it is applied when stretching
- . Default = Off

Edit polyline - REVERSE

command : pe(pedit)

Reverse of first vertex



Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : "
P1 Select and SPACE
3. "Edit polyline : ~ <eXit(X)>:"
r input and SPACE (Reverse)
4. "Edit vertices : ~ eXit(X) <Next>:"
e input and SPACE (Confirm vertex)
5. "Edit vertices : ~ eXit(X) <Next>:"
x input and SPACE (Exit edit vertices)
6. "Edit polyline : ~ <eXit(X)>:"
x input and SPACE (Exit edit polyline)

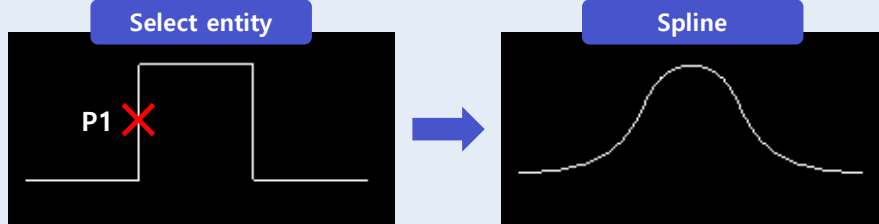
NOTE

- . Step 4 is executed to check the first vertex

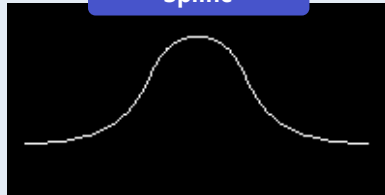
Edit polyline - SPLINE

command : pe(pedit)

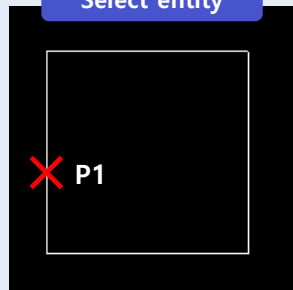
Spline



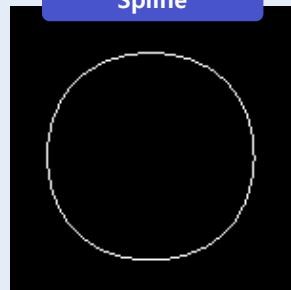
Spline



Select entity



Spline



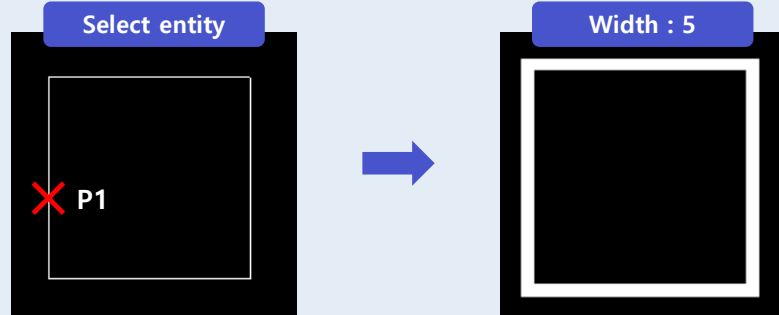
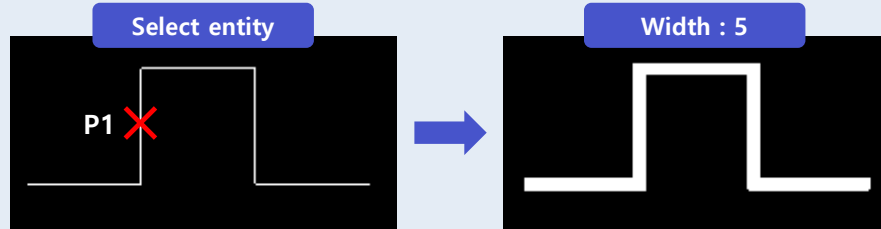
Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : "
P1 Select and SPACE
3. "Edit polyline : ~<eXit(X)>:"
s input and SPACE (Spline)
4. "Edit polyline : ~<eXit(X)>:"
x input and SPACE (Exit edit polyline)

Edit polyline - Width

command : pe(pedit)

Width



Execution Process

1. "command : "pe input and SPACE
2. "~ /Select polyline to edit [Multiple(M)] : "
P1 Select and SPACE
3. "Edit polyline : ~ <eXit(X)>:"
w input and SPACE (Width)
4. "Enter new width for all segments: "
5 input and SPACE
5. "Edit polyline : ~ <eXit(X)>:"
x input and SPACE (Exit edit polyline)

Edit polyline - Join

command : pe(pedit)

Training

Edit two polylines, apply joining and width to edited polylines

NOTE

- When joining, the end point of the entities must be connected

1 **pedit.dwg**

Select two polyline

2 **Move entity**

Copy to empty space

3 **TRIM execute**

TRIM

4 **Join and width of polyline**

1. "command : "pe input and SPACE
2. "~~/Select polyline to edit[Multiple(M)] :"
m input and SPACE
3. "Select entities: " Select entity to move
4. "Edit polyline : ~<eXit(X)>:"
j input and SPACE (Join)
5. "Enter fuzz distance or [Jointype] <0>"
0 input and SPACE
6. "Edit polyline : ~<eXit(X)>:"
w input and SPACE (Width)
7. "Enter new width for all segments: "
50 input and SPACE
8. "Edit polyline : ~<eXit(X)>:"
x input and SPACE (Exit edit polyline)

5 **Result**

MOVE

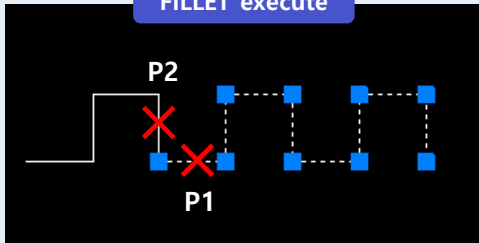
-261-

Case polyline join (Fillet)

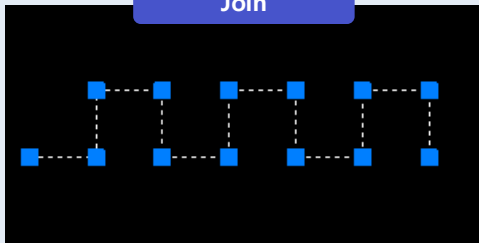
command : pe(pedit)

Polyline + Polyline

FILLET execute

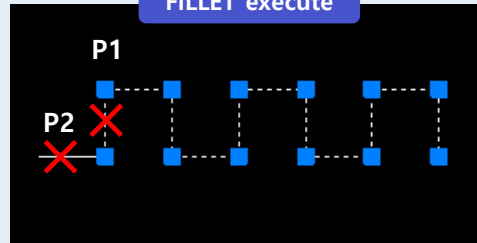


Join

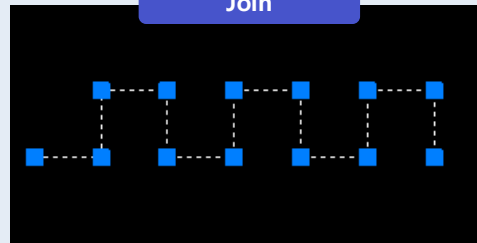


Polyline + Line

FILLET execute

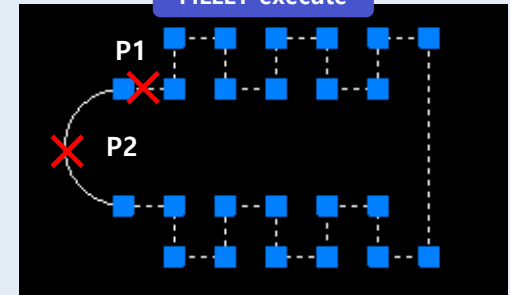


Join

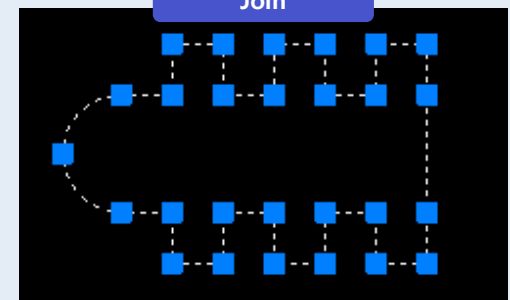


Polyline + Arc

FILLET execute



Join

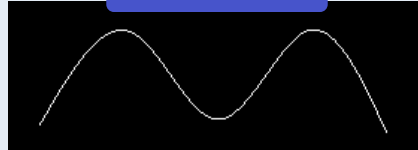


Spline → Polyline

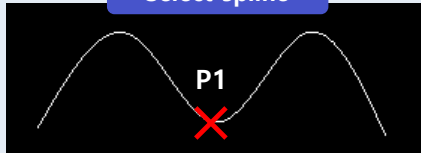
command : spe(splinedit)

Edit spline

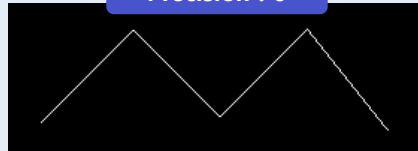
Precision : 10



Select spline

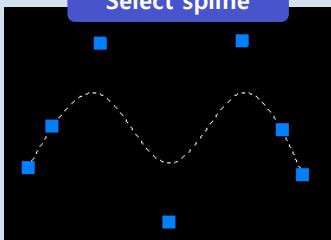


Precision : 0



Select (polyline, spline)

Select spline



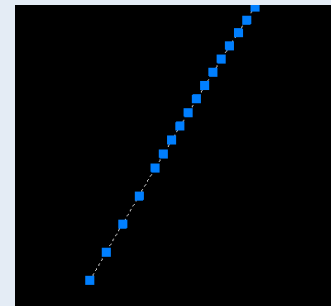
Select polyline



Execution Process

1. "command : "spe input and SPACE
2. "Select spline: " P1 Select and SPACE
3. "Enter an option ~<eXit(X)>: "
p Select and SPACE
4. "Specify a precision <0>:"
10 input and SPACE

NOTE



- When converted to polyline, the number of vertices increases with increasing precision