

Topological Editing

Combine Regions

Combine the selected region objects in the current layer into a composite object.



Usage Description

- Combining operations on two or more surface objects produces a new composite object. The combination of region objects in the region layer and the CAD layer is supported.
- The property information for the newly generated composite object is obtained through the callback function.
- When an object has an even number of overlapping faces, the area appears white after the combination, which is part of the result data.

Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.  
  
layer.setVisible(true);  
  
//True indicates that the layer is editable; false indicates that the layer is not editable.  
  
layer.setEditable(true);  
  
//True indicates that the layer is selectable; false indicates that the layer is not selectable.  
  
layer.setSelectable(true);
```

Step Two: Set the object action type to region combination

```
//Set the object action type to region combination  
  
mapControl.setAction(Action.COMPOSE_REGION);
```

Step three: Perform interactive operation on the device and submit

1. Trigger Combine Region action.
2. Click to select two or more region objects in turn.
3. Trigger Submit action.

```
mapControl.submit();
```

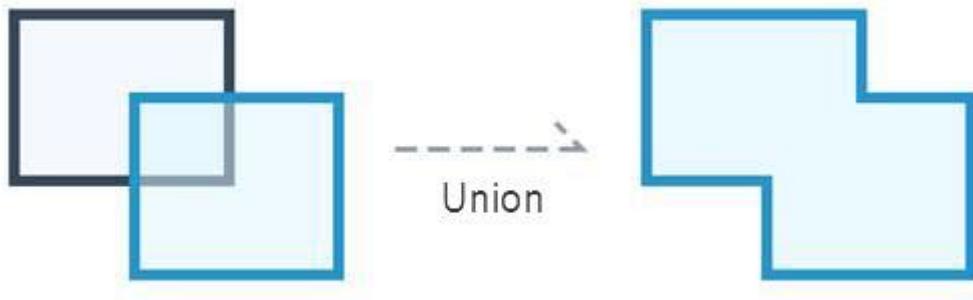
4. View combination results.

Notes:

- When there is contain relationship between the region objects, the results are also consistent with the results of the island and hollow polygon.
- After the combined operation in the region layer, the overlay part will display in white, which is part of the result.

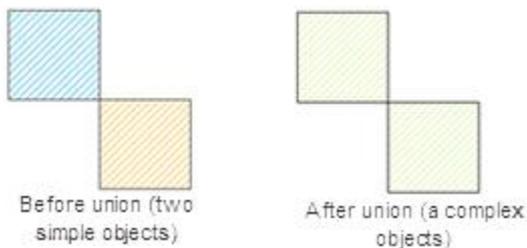
Merge Regions

In practical applications, we may need to merge the region objects. For example: When we want to merge 3 provinces into a single district on the map, we can select provinces three objects and combine them.

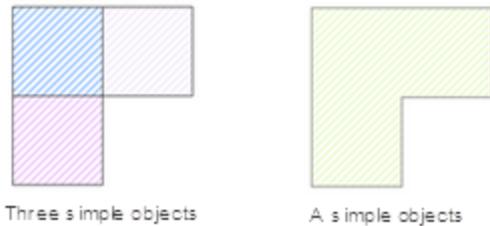


Usage Description

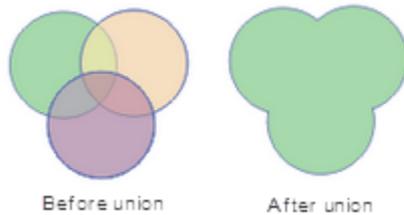
- The merge of region objects in the region layer and the CAD layer is supported.
- If the regions that participate in the operation intersect at points, the region objects will be merged into a compound region object (as shown below to produce a compound object with two child objects).



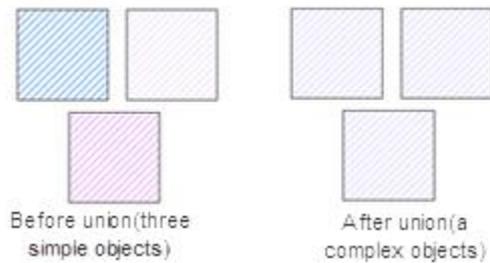
- If the regions that participate in the object's operations intersect at lines, the adjacent edges between the region objects will disappear, merging into a simple region object.



- If the regions that participate in the object's operations intersect at regions, they will be merged into a simple region object.



- If the regions that participate in the object's operations do not intersect, and are not adjacent to each other, the merge operation will generate a compound region object (the following figure demonstrates a compound object with three child objects).



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.
layer.setVisible(true);

//True indicates that the layer is editable; false indicates that the layer is not editable.
layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.
layer.setSelectable(true);
```

Step Two: Set the object action type to region merge

```
//Set the object action type to region merge
mapControl.setAction(Action.UNION_REGION);
```

Step three: Perform interactive operation on the device and submit

1. Trigger Merge Region action.

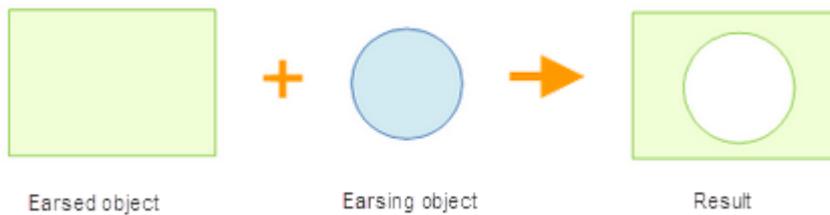
2. Select two region objects.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the merge results.

Erase Regions

The erase feature is used to delete the portion of the target object (erased object) that overlaps with the erase object.



Usage Description

- The erase feature is available only when a line object or region object is selected
- The erase features applies to region layers and CAD layers.
- Erasing and erased objects cannot be the same object.
- The erased object can be more than one region object, but the erasing object must be one region object.

Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.

layer.setVisible (true);

//True indicates that the layer is editable; false indicates that the layer is not editable.

layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.setSelectable(true);
```

Step Two: Set the object action type to region erase

```
//Set the object action type to region erase

mapControl.setAction(Action.ERASE_REGION);
```

Step three: Perform interactive operation on the device and submit

1. Trigger Erase Region action.
2. Draw a wiping polygon on the region object that needs to be erased;

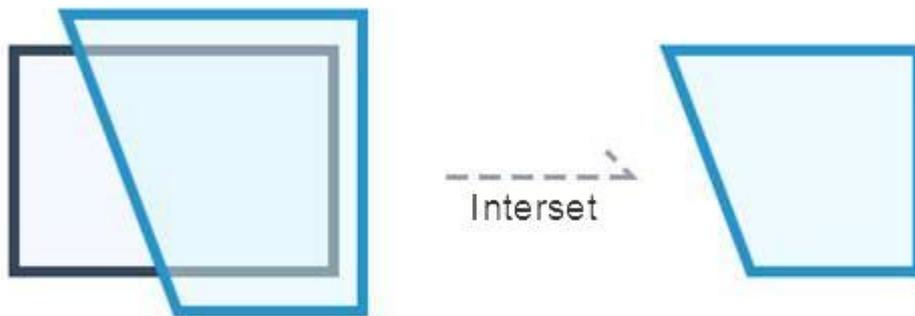
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the erase results.

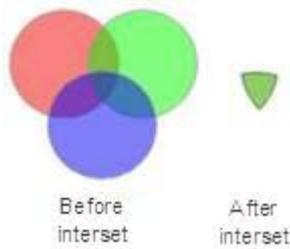
Intersect Regions

The intersection operation can get the common part of two or more objects and create a new object by manipulating the common areas of two or more region objects through the intersection operation. The common area of multiple objects is preserved and the remainder is deleted.

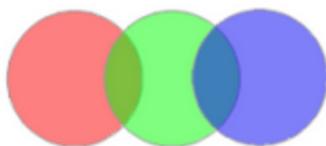


Usage Description

- If the intersection of all region objects that participate in the operation is not null, after intersection, a simple object shared by all region objects will be returned.



- If the intersection of all region objects that participate in the operation is null, after intersection, false will be returned, without any new objects returned.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.
```

```
layer.setVisible (true);

//True indicates that the layer is editable; false indicates that the layer is not editable.

layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.setSelectable(true);
```

Step Two: Set the object action type to region intersection

```
//Set the object action type to region intersection

mapControl.setAction(Action.INTERSECT_REGION);
```

Step three: Perform interactive operation on the device and submit

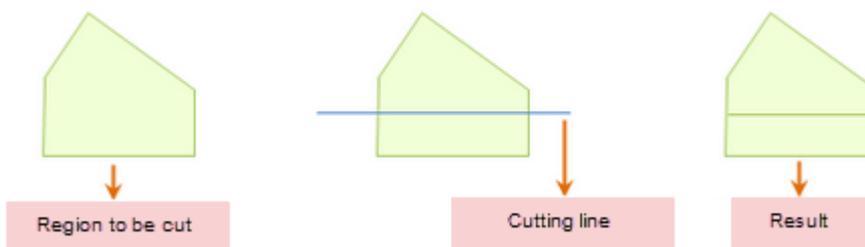
1. Trigger Intersect Region action.
2. Select two region objects.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the intersect results.

Line Splits Region

Draw a temporary split line to split region objects.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.

layer.setVisible (true);

//True indicates that the layer is editable; false indicates that the layer is not editable.
```

```
layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.selectable(true);
```

Step Two: Set the object action type to line splits region

```
//Set the object action type to region combination

mapControl.setAction(Action.SPLIT_BY_LINE);
```

Step three: Perform interactive operation on the device and submit

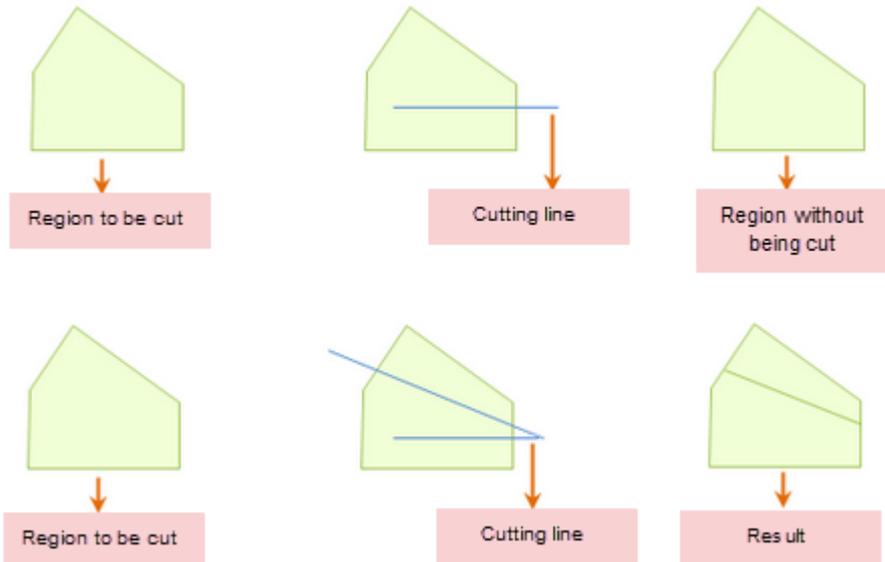
1. Trigger Split action.
2. Draw a temporary line over the region object.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View split results.

Notes:

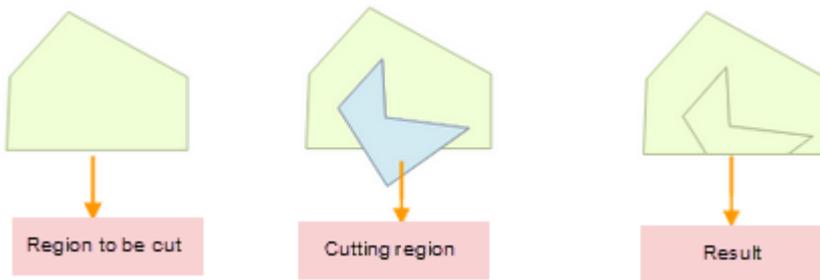
- Only if the temporary split line passes completely through the editable region geometry object, the geometric object can be split, as shown in the following figure.



- A temporary split line can only split one region object at a time.

Region Splits Region

Draw a temporary split region to split region objects.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.
layer.setVisible(true);

//True indicates that the layer is editable; false indicates that the layer is not editable.
layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.
layer.setSelectable(true);
```

Step Two: Set the object action type to region splits region

```
//Set the object action type to region splits region
mapControl.setAction(Action.SPLIT_BY_REGION);
```

Step three: Perform interactive operation on the device and submit

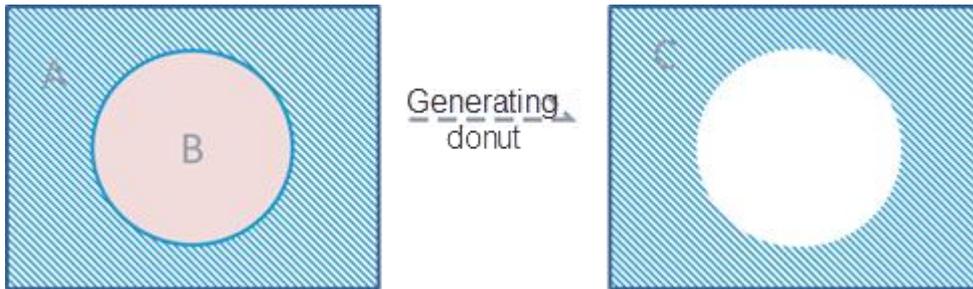
1. Trigger Split action.
2. Draw a temporary region over the region object.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View split results.

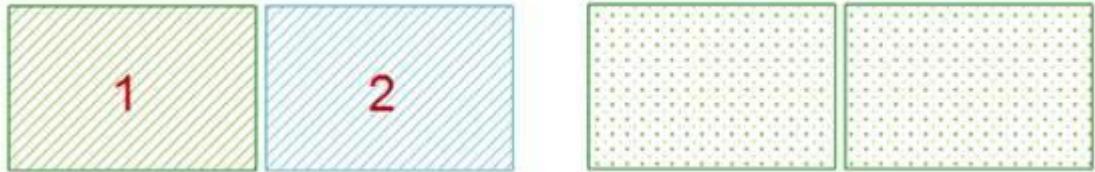
Generate Island and Hole

“COMPOSE_HOLLOW_REGION” Action allows you to select two or more region objects, to generate island and hole.

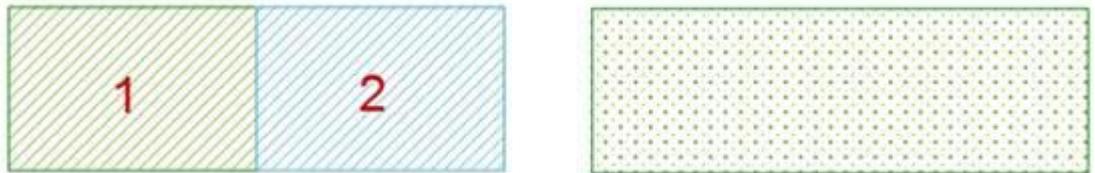


Usage Description

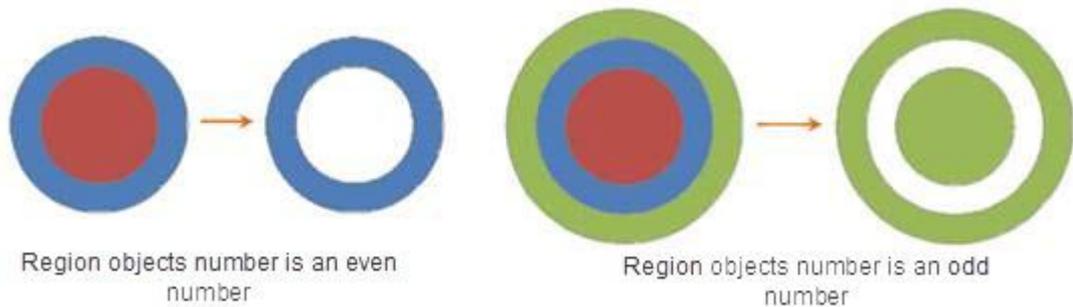
- The island hole polygon is applicable to the region layer or the CAD layer.
- If you select two or more region objects, and perform the island polygon operation, there will be several situations:
 - If the selected region objects are disjoint, a compound object will be generated.



- If the selected region objects intersect at points or lines, the region objects will be merged into a compound object.



- If the selected region objects intersect at a face but do not coincide with each other, the intersected parts will be maintained and a compound object will be returned if the number of region objects is odd. When the number of objects is even, the intersection will be deleted and a compound object will be returned.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.

layer.setVisible (true);

//True indicates that the layer is editable; false indicates that the layer is not editable.

layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.setSelectable(true);
```

Step Two: Set the object action type to generate island and hole

```
//Set the object action type to generate island and hole

mapControl.setAction(Action.COMPOSE_HOLLOW_REGION);
```

Step three: Perform interactive operation on the device and submit

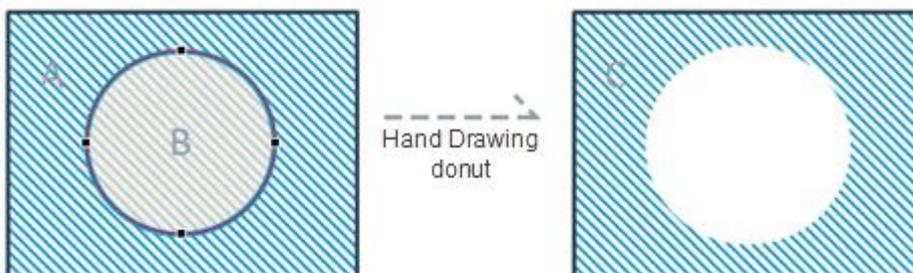
1. Trigger Generate Island and Hole action.
2. Select multiple regions that intersect but not coincident with the region.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the results.

Draw Islands

“DRAW_HOLLOW_REGION” Action draws the island and hole object by overlaying a region on the region object.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.
```

```
layer.setVisible (true);

//True indicates that the layer is editable; false indicates that the layer is not editable.

layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.setSelectable(true);
```

Step Two: Set the object action type to draw islands

```
//Set the object action type to draw islands

mapControl.setAction(Action.DRAW_HOLLOW_REGION);
```

Step Three: Interactively draw an island and hole object on the device and submit

1. Trigger Draw Island and Hole action.
2. Draw a region that intersects but not coincident with the region.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the results.

Fill Island and Hole

“FILL_HOLLOW_REGION” Action draws a temporary line across the island and hole object to fill the object into a simple region object.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.
```

```
layer.setVisible (true);

//True indicates that the layer is editable; false indicates that the layer is not editable.

layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.setSelectable(true);
```

Step Two: Set the object action type to fill island and hole

```
//Set the object action type to fill island and hole

mapControl.setAction(Action.FILL_HOLLOW_REGION);
```

Step Three: Interactively perform the fill island and hole object operation on the device and submit

1. Trigger Fill Island and Hole action.
2. Draw a temporary line across the island and hole object.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the results.

Island and Hollow Regions

The island-hole polygon is a kind of compound geometric object. In the editable status, process the overlap area of two or more objects with the relation of contain to finally form an island-hole polygon. If there is a lake in a region, an island-hole polygon can be obtained. The island hole polygon is applicable to the region layer or the CAD layer.

Generate Island and Hole: Select two or more region objects, to generate island and hole.

Draw Island and Hole: Draw the island and hole object by overlaying a region on the region object.

Fill Island and Hole: Draw a temporary line across the island and hole object to fill the object into a simple region object.

Supplement Island and Hole: Draw a temporary line across the island and hole object to supplement the lost part of the object with a drawn region object.

Supplement Island and Hole

“PATCH_HOLLOW_REGION” Action draws a temporary line across the island and hole object to supplement the lost part of the object with a drawn region object.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.
layer.setVisible (true);

//True indicates that the layer is editable; false indicates that the layer is not editable.
layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.
layer.setSelectable(true);
```

Step Two: Set the object action type to supplement island and hole

```
//Set the object action type to supplement island and hole
mapControl.setAction(Action.PATCH_HOLLOW_REGION);
```

Step Three: Interactively perform the supplement island and hole object operation on the device and submit

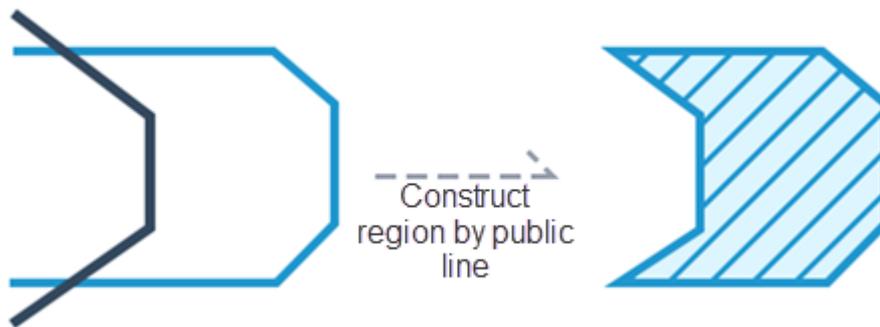
1. Trigger Supplement Island and Hole action.
2. Draw a temporary line across the island and hole object.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the results.

Construct Region with Common Edges

“**CREATE_POSITIONAL_REGION**” Action When you draw a region, it will trigger the Construct Region with Common Edges action. You can draw a region object at the area where there are common edges with the region to be drawn.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.  
  
layer.setVisible (true);  
  
//True indicates that the layer is editable; false indicates that the layer is not editable.  
  
layer.setEditable(true);  
  
//True indicates that the layer is selectable; false indicates that the layer is not selectable.  
  
layer.setSelectable(true);
```

Step Two: Set the object action type to construct region with common edges

```
//Set the object action type to construct region with common edges  
  
mapControl.setAction(Action.CREATE_POSITIONAL_REGION);
```

Step three: Perform interactive operation on the device and submit

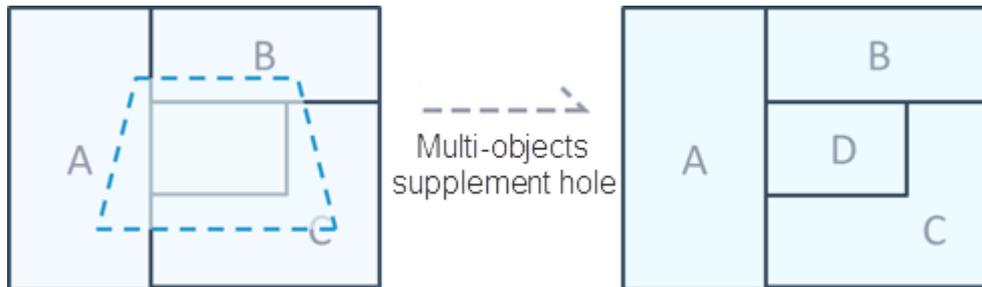
1. Trigger Construct Region with Common Edges action.
2. Draw a temporary region construction area at the original region object.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the results.

Fill Holes with Regions

“PATCH_POSOTIONAL_REGION” Action is used to supplement closed area formed by a multiple if region objects.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.

layer.setVisible(true);

//True indicates that the layer is editable; false indicates that the layer is not editable.

layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.setSelectable(true);
```

Step Two: Set the object action type to fill holes with regions

```
//Set the object action type to fill holes with regions

mapControl.setAction(Action.PATCH_POSOTIONAL_REGION);
```

Step three: Perform interactive operation on the device and submit

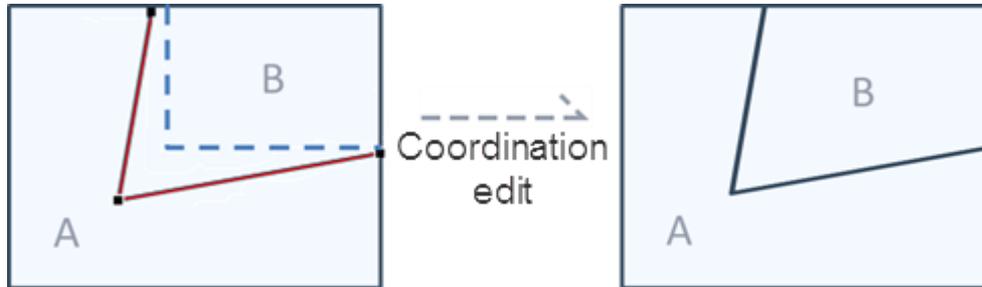
1. Trigger fill holes with regions action.
2. Draw a temporary region object containing a closed space surrounded by multiple objects at the position to be supplemented.
3. Trigger Submit action.

```
mapControl.submit();
```

4. View the results.

Coordinated Editing

“**MOVE_COMMON_NODE**” Action allows you to adjust two or more shapes with common edge objects by adjusting the nodes on the common edges of two or more region objects.



Implementation Steps

Step One: Operation layer settings

```
//True indicates that the layer is visible; false indicates that the layer is invisible.

layer.setVisible(true);

//True indicates that the layer is editable; false indicates that the layer is not editable.

layer.setEditable(true);

//True indicates that the layer is selectable; false indicates that the layer is not selectable.

layer.setSelectable(true);
```

Step Two: Set the object action type to single selection

```
//Set the object action type to single selection

mapControl.setAction(Action.SELECT);
```

Step Two: Set the object action type to coordinated editing

```
//Set the object action type to fill holes with regions

mapControl.setAction(Action.MOVE_COMMON_NODE);
```

Step three: Perform interactive operation on the device and submit

1. Trigger Single Selection action.
2. Click one of the two or more region objects for coordinated editing in an editable layer in a map window, lift the finger.
3. Select the nodes to be adjusted and draw the node to adjust the shape of the region.

4. Trigger Submit action.

```
mapControl.submit();
```

5. View the results.