

# OnNewLayerEx

## OnNewLayerEx picture event

### Declaration

Global handler:

```
ENTRY OnNewLayerEx(IN INT _refId, _newLayer, INT _x1, _y1, _x2, _y2, IN INT
_x3, _y3, BOOL _enable)
; actions
END OnNewLayerEx
```

### Parameters

_refId	Input parameter of the <i>Int</i> type. refID of object over which the mouse cursor is placed, or 0.
_newLayer	Input parameter of the <i>Int</i> type. Number of new layer.
_x1, _y1, _x2, _y2	Input-output parameter of the <i>Int</i> type. Visible resultant area after changing the layer.
_x3, _y3	Input parameter - position of mouse cursor in a picture.
_enable	Output parameter of the <i>BOOL</i> type. Permission to change a layer.

### Description

Picture event is generated before changing the layer in a picture. Unlike the picture event [OnNewLayer](#), which only informs about the change of layer, OnNewLayerEx enables to control actively the way of changing the visible layer.

Event handler can enable or disable changing the layer by the parameter *\_enable*. If changing the layer is disabled, the picture will change zoom but the layer will not be switched. The output parameters will not be used.

If changing the layer if enabled, the picture will change an active layer and set the visible area according to output parameters *[x1, y1][x2, y2]*. In the simplest situation, the event handler enables only switching the layers but not changing the values that define the visible area after switching.

### Example

```
ENTRY OnNewLayerEx(IN INT _refId, _layers, INT _x1, _y1, _x2, _y2, IN INT
_x3, _y3, BOOL _enable)
IF _layers=2 THEN
  _x1 := 100
  _y1 := 100
  _x2 := 500
  _y2 := 500
ENDIF
_enable := @TRUE

END OnNewLayerEx
```



#### Related pages:

[ENTRY - picture event handler](#)  
[Active picture events](#)