

# Entry - Picture Event Handler

## ENTRY - picture event handler

### Declaration

```
ENTRY EntryName [([IN] type1 paramName1[,paramName2, ...] [IN] type2  
paramName3]...)]  
; script actions  
END EntryName
```

```
ENTRYX EntryName [([IN] type1 paramName1[,paramName2, ...] [IN] type2  
paramName3]...)]  
; script actions  
END EntryName
```

### Parameters

EntryName	Picture event name. The name must be one of the defined <a href="#">picture events</a> .
type1	First parameter type.
paramName1	First parameter name.
type2	Second parameter type.
paramName2	Second parameter name. You can define up to 10 parameters <i>type</i> and <i>paramName</i> .
....	

#### Note:

Picture event parameters are given by the event type. Other combination than permitted is not allowed.

### Description

The action initiates the handler of particular picture event. Generally, picture event handler is a specialized procedure initiates with the keyword **ENTRY** (or **ENTRYX**).

Picture events generated by graphic objects can be handled in **two ways**:

- **special handler** of a picture event for a graphic object
- **global handler** of a picture event for all graphic objects

The first way (special handler) handles events generated by one graphic object. Such handler contains a special name consisting of the graphic object name (the [reference variable](#) connected to graphic object) and picture event name. This method uniquely identifies the handler name (for example: **BtnOK\_OnClick** . **\_BtnOK** is a Reference local variable connected to the graphic object and **OnClick** is an picture event type).

The second way (global handler) handles all picture events of given type (for example: **OnClick**) for all graphic objects, for which a special handler of given picture event.

If a picture event handler is defined by the keyword **ENTRYX**, requests for handling picture events will be reduced so, that last picture event, not handled yet, will be handled first. During generation of a new picture event, all the picture event of the same type waiting to handle will be ignored.

The keyword **ENTRYX** may not be used for the following picture events: [OnClose](#), [OnPopUpMenu\\_Result](#), [OnSubPictureClosed](#), [OnAXEvent](#), [OnItemValidate](#), [OnFetchDone](#), [OnUserInput](#).

### Example

Special picture event handler:

```
; event handler: Clicking the graphic object with
```

```
; reference variable _BtnOK assigned
```

```
ENTRY BtnOK_OnClick
    %HI_ClosePicture() ; closing picture
END BtnOK_OnClick
```

Global picture event handler:

```
; event handler: Clicking the graphic objects that
```

```
; has no special handler defined
```

```
ENTRY OnClick(IN INT _refId)
```

```
; _refId - the value of the reference local variable assigned to graphic
object
    IF _refId = _BtnOK THEN ; test, that determines the graphic object
                                ; the user clicked on
        %HI_ClosePicture() ; closing picture
    ENDIF
END OnClick
```

**Note:**

If there are defined both the handlers in the picture script, global handler will never be called for \_refId=\_BtnOK, because there is defined the special handler.



**Related pages:**

[Picture events](#)