

# Contents

- [1 Launching Scripts by Creating Listeners](#)
  - [1.1 Example of Script Launching by Listeners](#)
  - [1.2 Examples of Listeners](#)
- [2 Example of Script Launching Using the "Script Call" Macros](#)

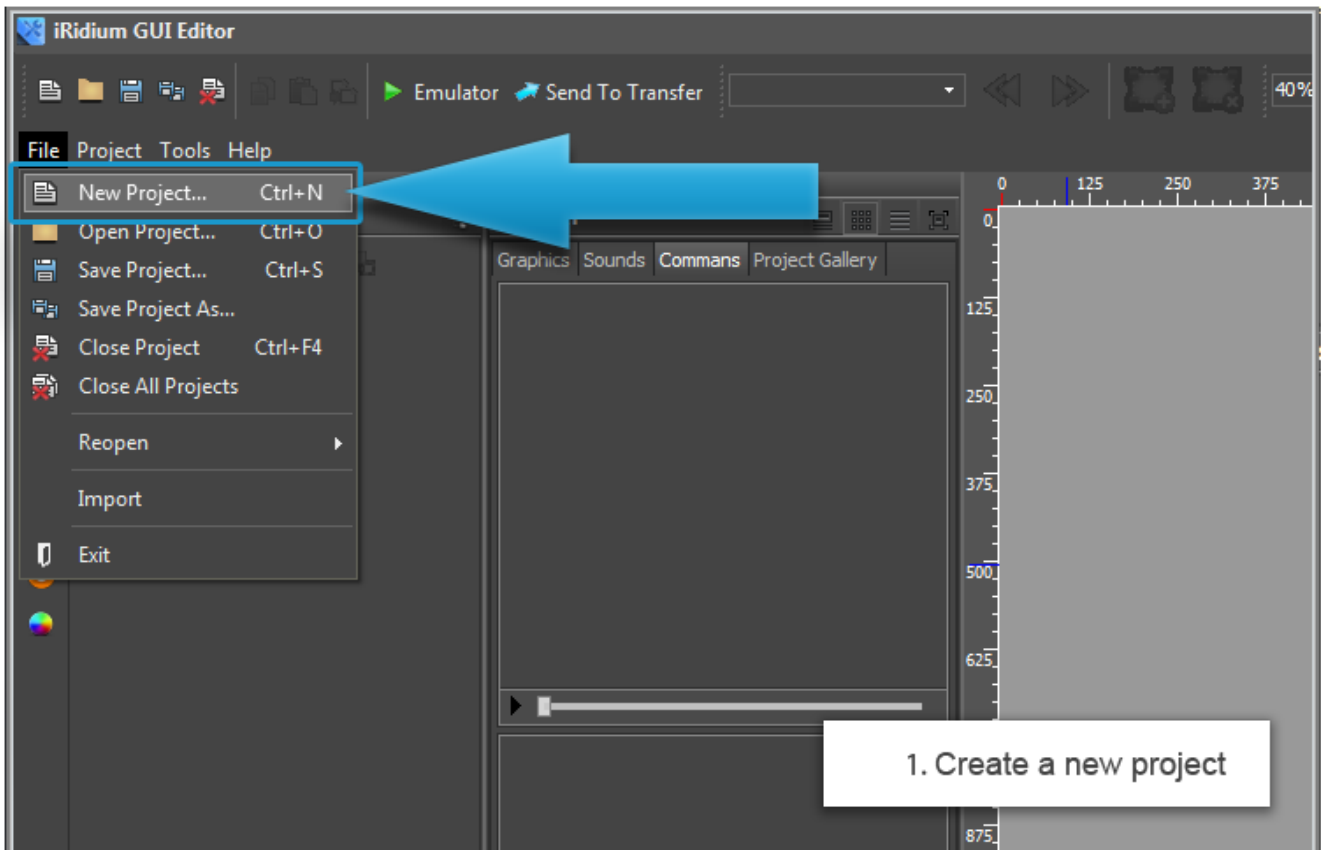
## Launching Scripts by Creating Listeners

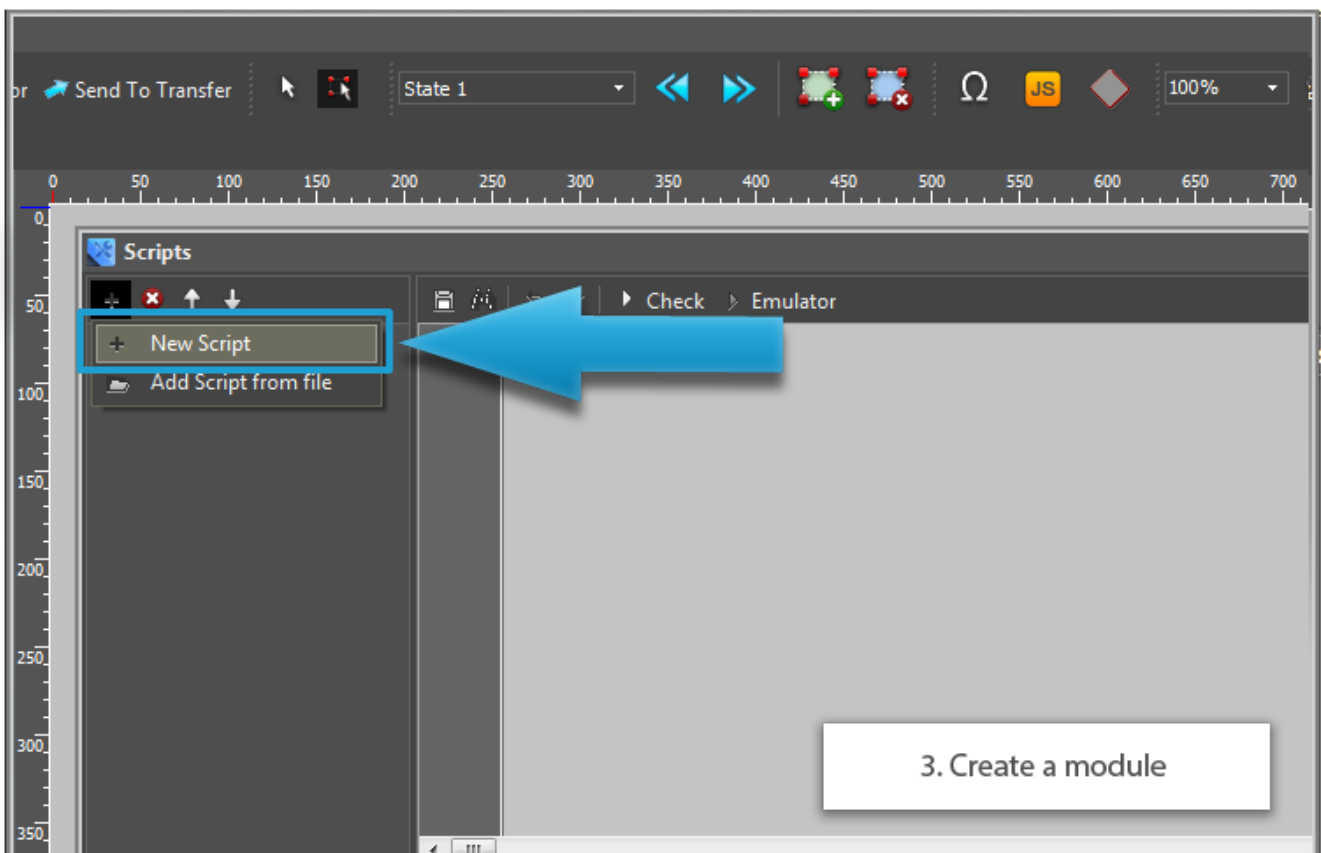
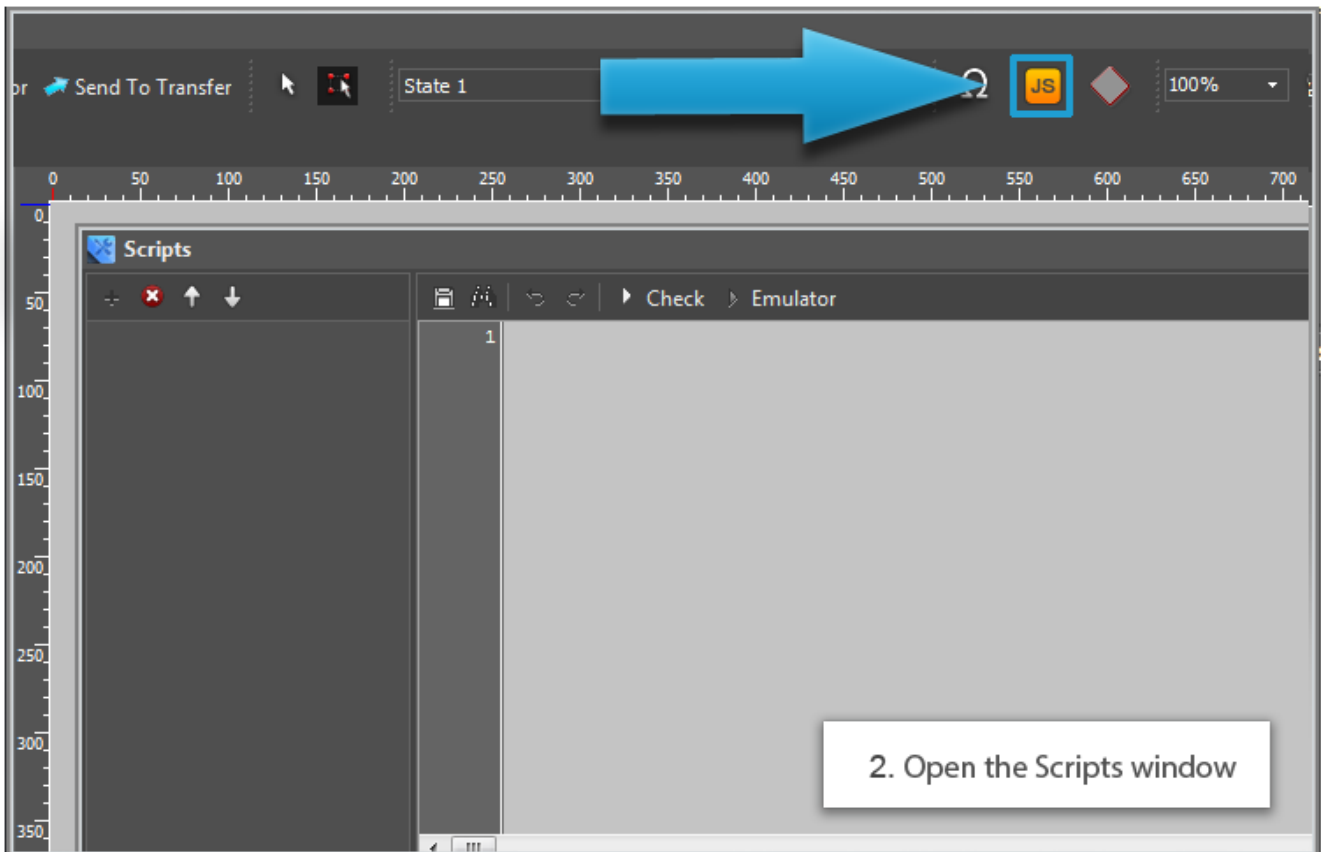
**Listener** - the function which is performed when the event assigned to it is activated.

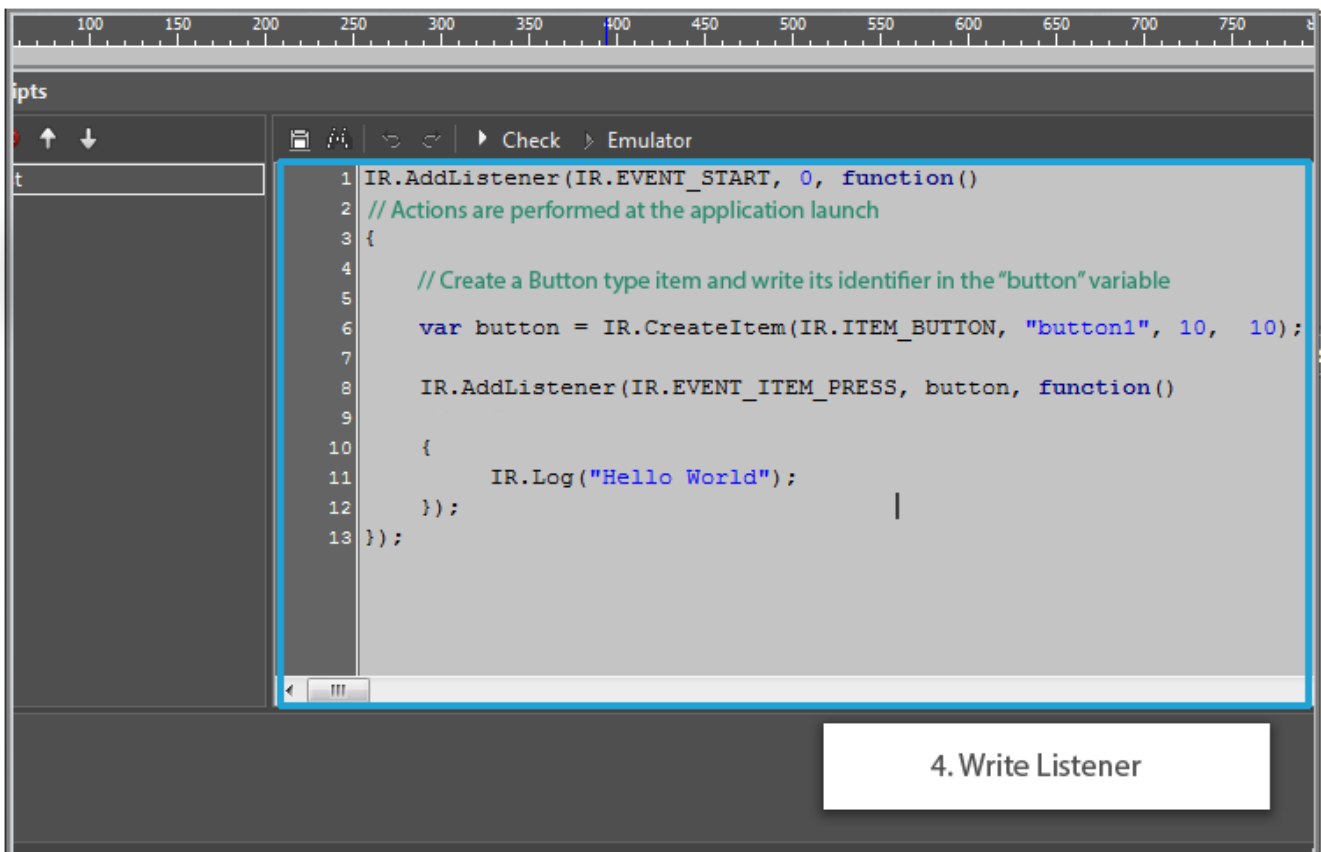
```
// Assign Listener for the event of the application launch
IR.AddListener(IR.EVENT_START, 0, function()
{
    // Body of the Listener
});
```

## Example of Script Launching by Listeners

[DOWNLOAD: Example of a project](#)



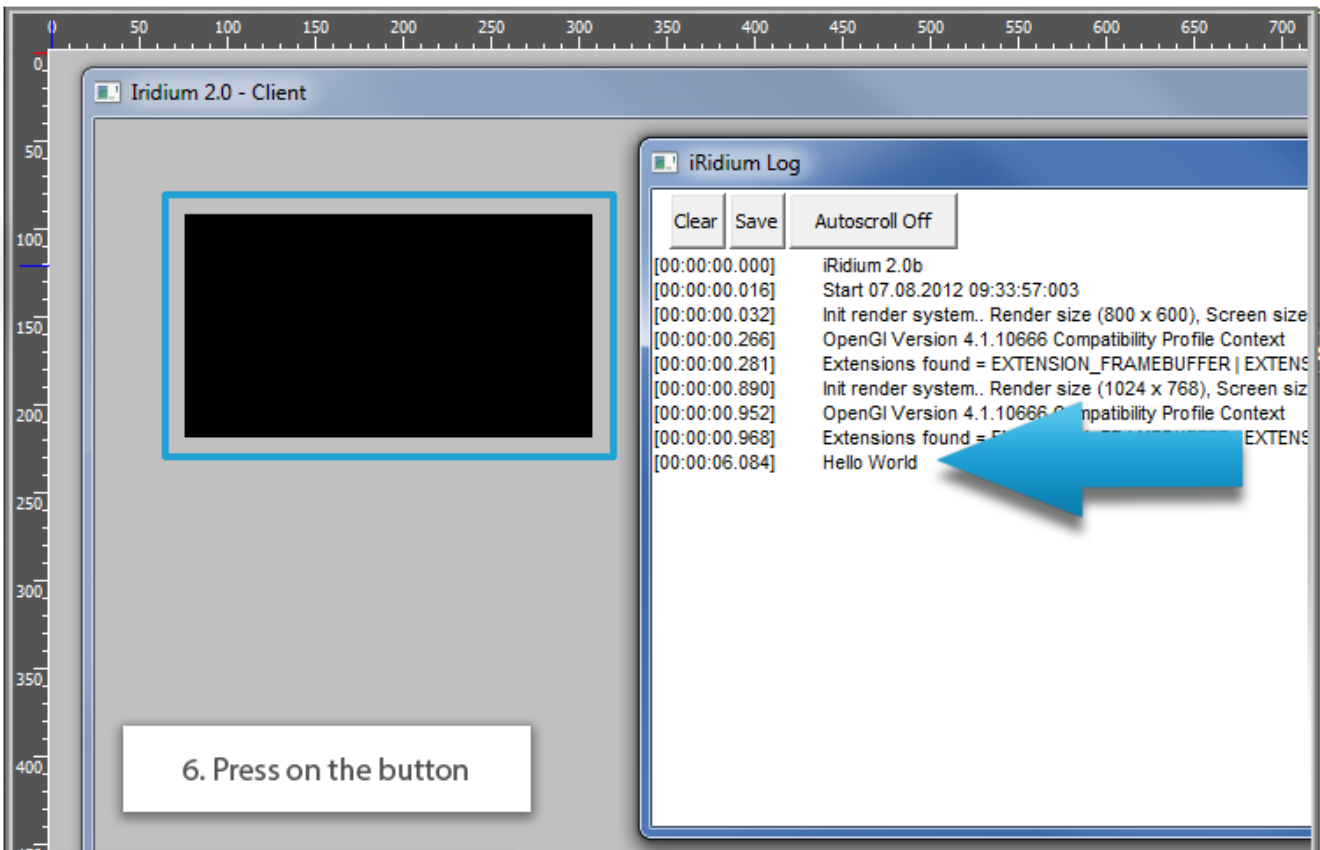
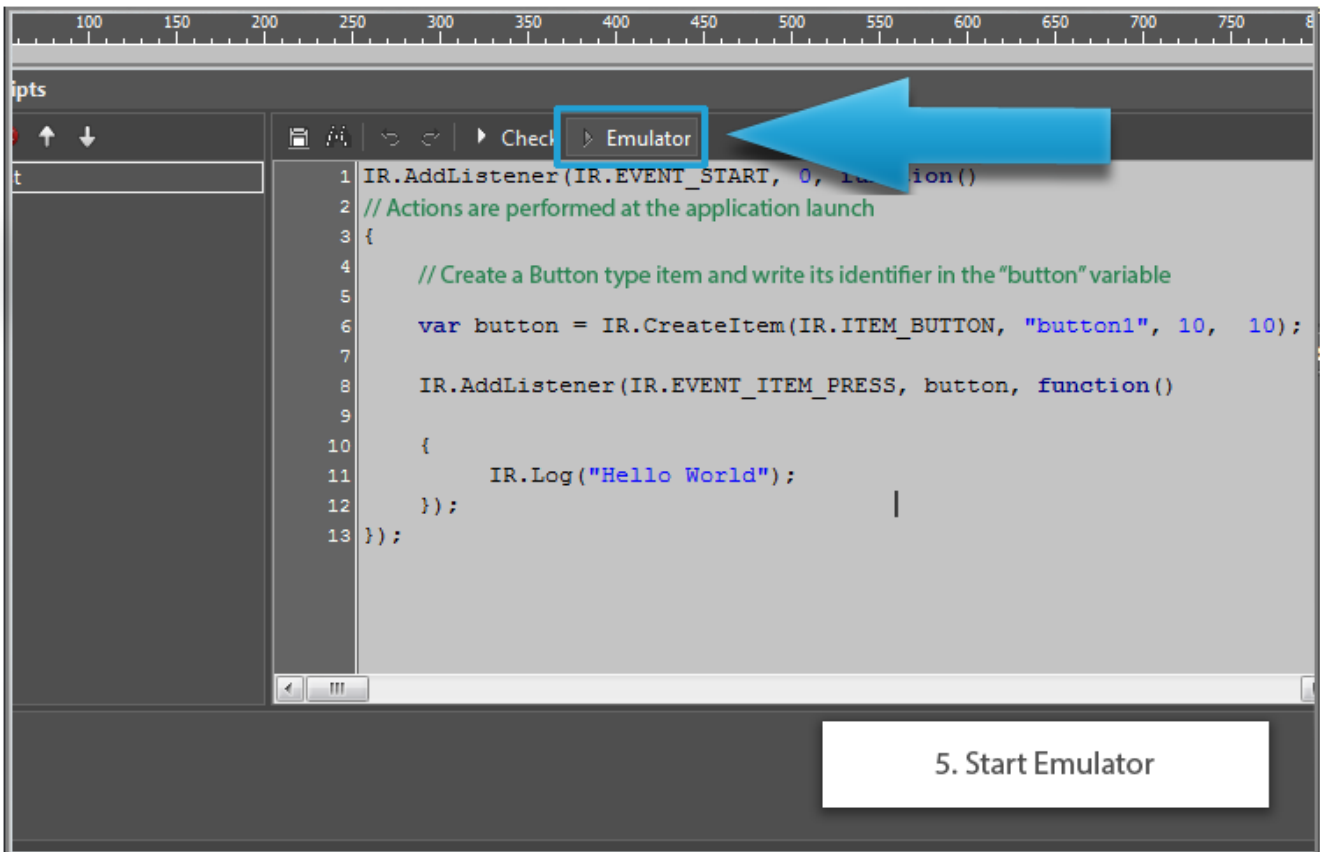




- Example of written Listener:

```
IR.AddListener(IR.EVENT_START, 0, function()
// Actions are performed at the application launch
{
    /* Create a Button type item and write its identifier in the "button"
variable */
    var button = IR.CreateItem(IR.ITEM_BUTTON, "button1", 10, 10);

    IR.AddListener(IR.EVENT_ITEM_PRESS, button, function()
// Actions are performed at the application launch
    {
        IR.Log("Hello World");
    });
});
```



## Examples of Listeners

- **IR.EVENT\_START** - start of Client work

```
IR.AddListener(IR.EVENT_START, 0, function() // Actions are performed at
start
{
    IR.Log("Start");
});
```

- **IR.EVENT\_WORK** - cycle performance of actions during the work of Client

```
var timer = 0;
var onTime = 800; // 800 ms = 0.8 s
IR.AddListener(IR.EVENT_WORK, 0, function(time) //Performing actions at work
{
    timer += time;
    if(timer > onTime)
    {
        timer = 0;
        IR.Log("Click");
    }
});
```

- **IR.EVENT\_EXIT** - Exiting Clientвыход из клиента

```
IR.AddListener(IR.EVENT_EXIT, 0, function() //Action is performed when
exiting
{
    IR.Log("Exit");
});
```

- **IR.EVENT\_ITEM\_PRESS** - pressing on the item (the item can be an object, popup or page)

```
IR.AddListener(IR.EVENT_START, 0, function()
{
    var popup = IR.CreateItem(IR.ITEM_POPUP, "popup1", 150, 10, 200, 200);
    var button = IR.CreateItem(IR.ITEM_BUTTON, "button1", 10, 10);
    button.Text = "popup1";

    IR.AddListener(IR.EVENT_ITEM_PRESS, button, function()
    {
        IR.TogglePopup("popup1");
    });
});
```

- **IR.EVENT\_ITEM\_RELEASE** - releasing the item

```

IR.AddListener(IR.EVENT_START, 0, function()
{
    var popup = IR.CreateItem(IR.ITEM_POPUP, "popup1", 150, 10, 200, 200);
    var button = IR.CreateItem(IR.ITEM_BUTTON, "button1", 10, 10);
    button.Text = "popup1";

    IR.AddListener(IR.EVENT_ITEM_RELEASE, button, function()
    {
        IR.TogglePopup("popup1");
    });
});

```

- **IR.EVENT\_ITEM\_SELECT** - clicking on the list item

```

// Event is activated when clicking on the list item
IR.AddListener(IR.EVENT_ITEM_SELECT, IR.GetItem("Page 1").GetItem("Item 1"),
function(item, subItem)
{
    list.DeleteItem(item); // Delete the third list item
});

```

- **IR.EVENT\_GESTURE\_BEGIN** - beginning of a gesture

```

var button;

// It is activated at the application launch
IR.AddListener(IR.EVENT_START, 0, function()
{
    // Create a button for switching between pages
    var button = IR.CreateItem(IR.ITEM_BUTTON, "text", 10, 10);
    button.Text = "";

    // Gestures
    IR.AddRecognizer(IR.GESTURE_SWIPE_LEFT);
    IR.AddRecognizer(IR.GESTURE_SWIPE_RIGHT);
    IR.AddRecognizer(IR.GESTURE_SWIPE_UP);
    IR.AddRecognizer(IR.GESTURE_SWIPE_DOWN);
    IR.AddListener(IR.EVENT_GESTURE_BEGIN, IR.CurrentPage,
function(gesture)
    {
        switch(gesture)
        {
            case IR.GESTURE_SWIPE_LEFT:
                button.Text = "Left";
                break;
            case IR.GESTURE_SWIPE_RIGHT:
                button.Text = "Right";
                break;
            case IR.GESTURE_SWIPE_UP:
                button.Text = "Up";

```

```

                break;
            case IR.GESTURE_SWIPE_DOWN:
                button.Text = "Down";
                break;
        }
    });
});

```

- **IR.EVENT\_ONLINE** - connection to the device (driver)

```

IR.AddListener(IR.EVENT_ONLINE , 0, function()
{
    IR.Log("Device is online");
});

```

- **IR.EVENT\_OFFLINE** - disconnecting the device (driver)

```

IR.AddListener(IR.EVENT_OFFLINE , 0, function()
{
    IR.Log("Device is offline");
});

```

- **IR.EVENT\_RECEIVE\_DATA** - receiving data from the device (binary data)

```

IR.AddListener(IR.EVENT_RECEIVE_DATA , 0, function(text)
{
    IR.Log(text); //Outputting information received from the device in the
byte format
});

```

- **IR.EVENT\_RECEIVE\_TEXT** - receiving data from the device (the string format)

```

IR.AddListener(IR.EVENT_RECEIVE_TEXT , 0, function(text)
{
    IR.Log(text); // Outputting information received from the device in the
string format
});

```

- **IR.EVENT\_TAG\_CHANGE** - changing the tag value

```

IR.AddListener(IR.EVENT_TAG_CHANGE , 0, function(name,value)
{
    IR.Log("Name = "+name+" value = "+value);
//Outputting the name of the changed tag and its new value in the console
});

```

- **IR.EVENT\_KEYBOARD\_SHOW** - opening a keyboard

```

IR.AddListener(IR.EVENT_KEYBOARD_SHOW, 0, function()

```

```
{
    IR.Log("keyboard showed on screen");
});
```

- **IR.EVENT\_ORIENTATION** - changing device orientation

```
IR.AddListener(IR.EVENT_ORIENTATION, 0, function(orientation)
//Event is activated when changing the device orientation
{
    IR.Log(orientation);
//device orientation (0 - landscape, 1 - portrait)

});
```

- **IR.EVENT\_MOUSE\_DOWN** - moving the mouse down

```
IR.AddListener(IR.EVENT_MOUSE_DOWN, 0, function()
{
    IR.Log("Mouse Down");
});
```

- **IR.EVENT\_MOUSE\_UP** - moving the mouse up

```
IR.AddListener(IR.EVENT_MOUSE_UP, 0, function()
{
    IR.Log("Mouse Up");
});
```

- **IR.EVENT\_MOUSE\_MOVE** - moving the mouse in any direction

```
IR.AddListener(IR.EVENT_MOUSE_MOVE, 0, function()
{
    IR.Log("Mouse Move");
});
```

- **IR.EVENT\_TOUCH\_DOWN** - moving the finger down

```
IR.AddListener(IR.EVENT_TOUCH_DOWN, 0, function()
{
    IR.Log("touch down");
});
```

- **IR.EVENT\_TOUCH\_UP** - moving the finger up

```
IR.AddListener(IR.EVENT_TOUCH_UP, 0, function()
{
    IR.Log("touch up");
});
```

- **IR.EVENT\_TOUCH\_MOVE** - moving the finger in any direction



```
IR.AddListener(IR.EVENT_TOUCH_MOVE, 0, function()
{
    IR.Log("touch move");
});
```

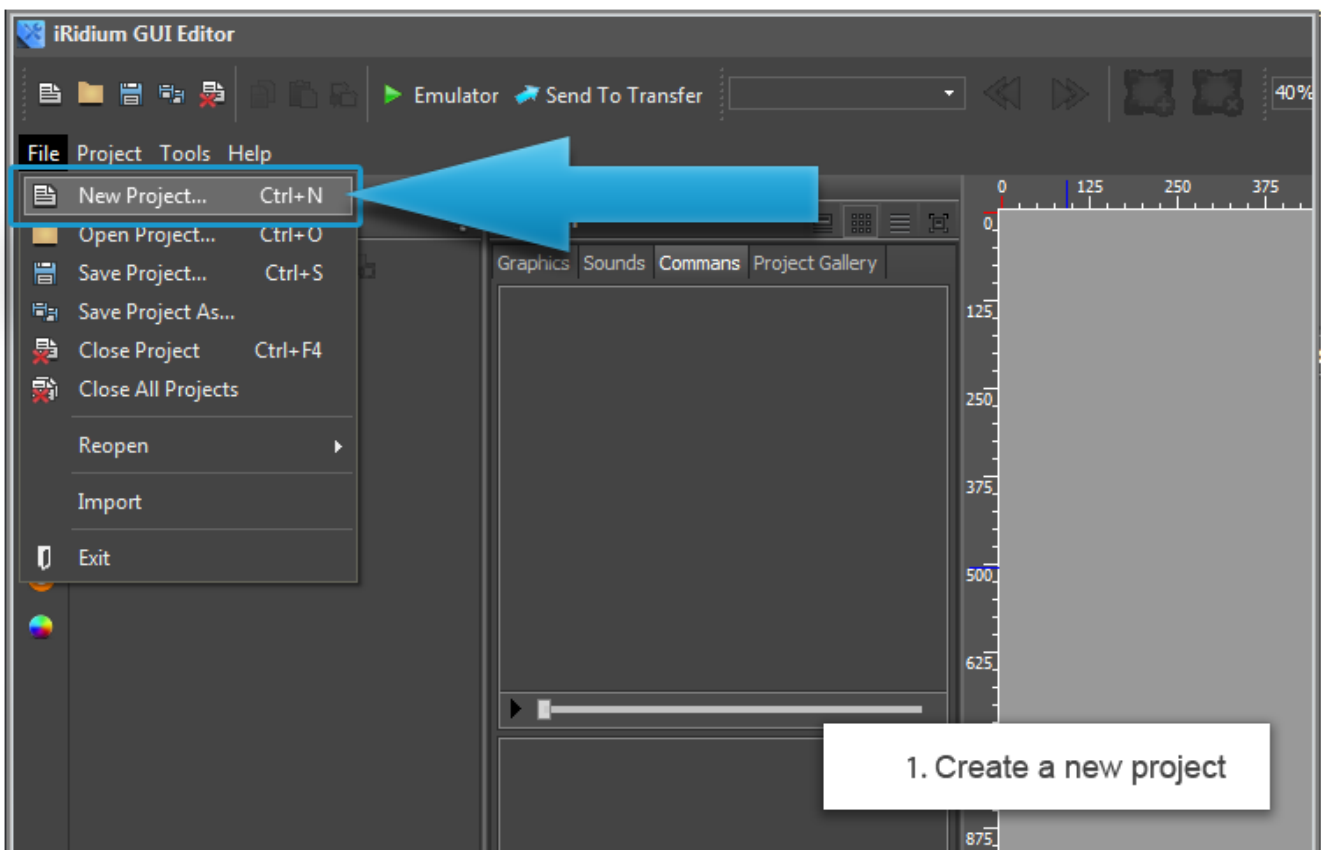
- **IR.EVENT\_ITEM\_CHANGE** - changing the Editbox item

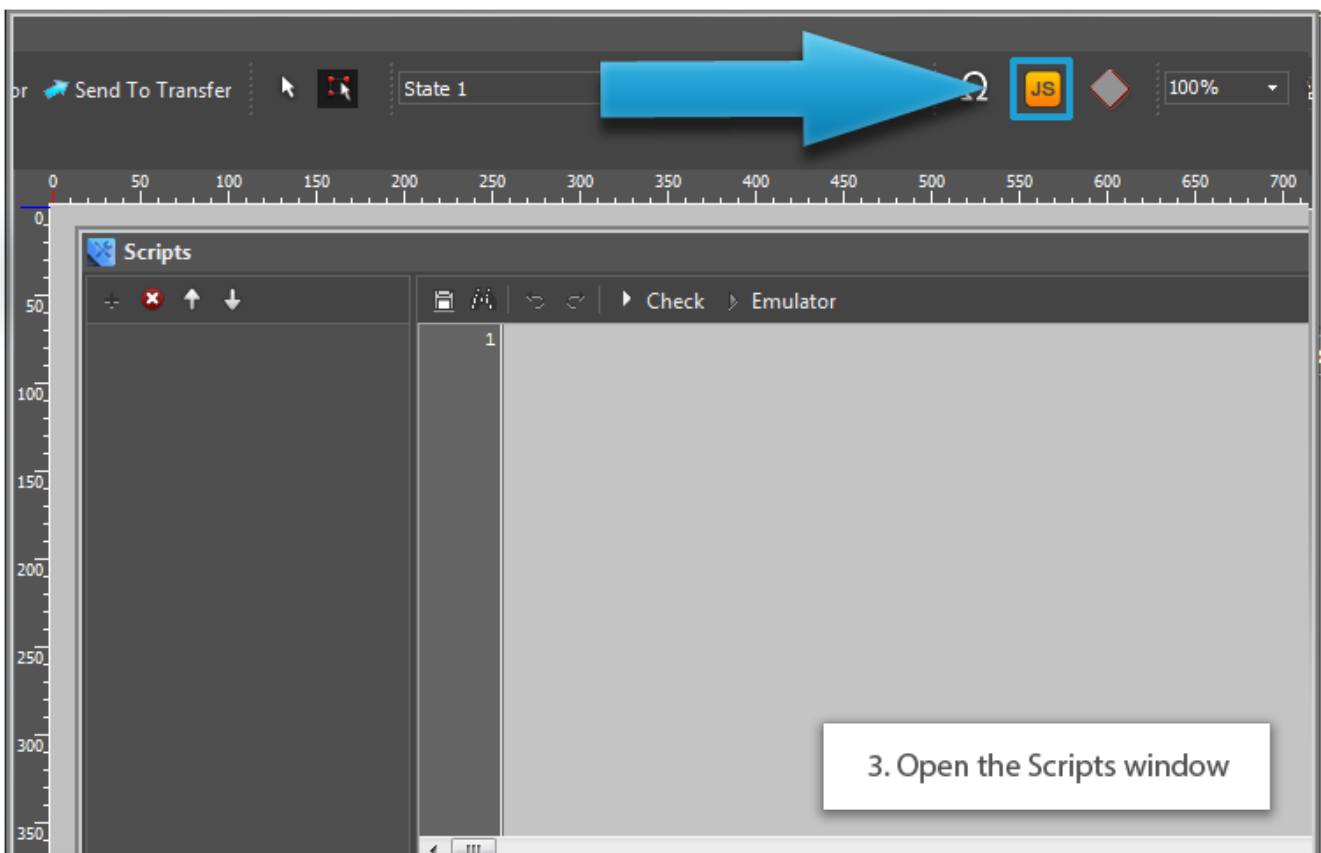
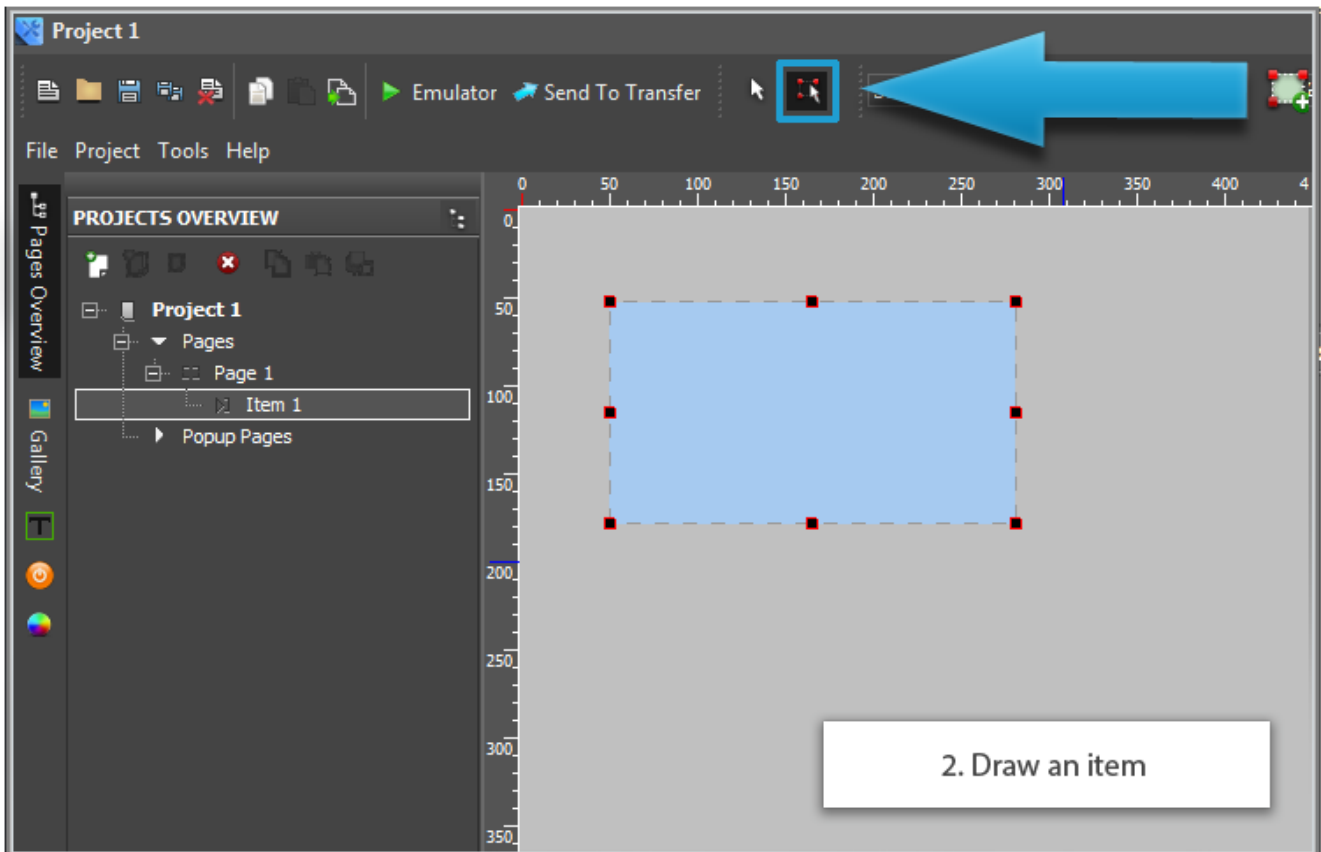
```
IR.AddListener(IR.EVENT_ITEM_CHANGE, IR.GetItem("Page 1").GetItem("Item
1"),function()
//Event is activated at change
{
    IR.Log(IR.GetItem("Page 1").GetItem("Item 1").Text);
//Outputting the changes in the log
});
```

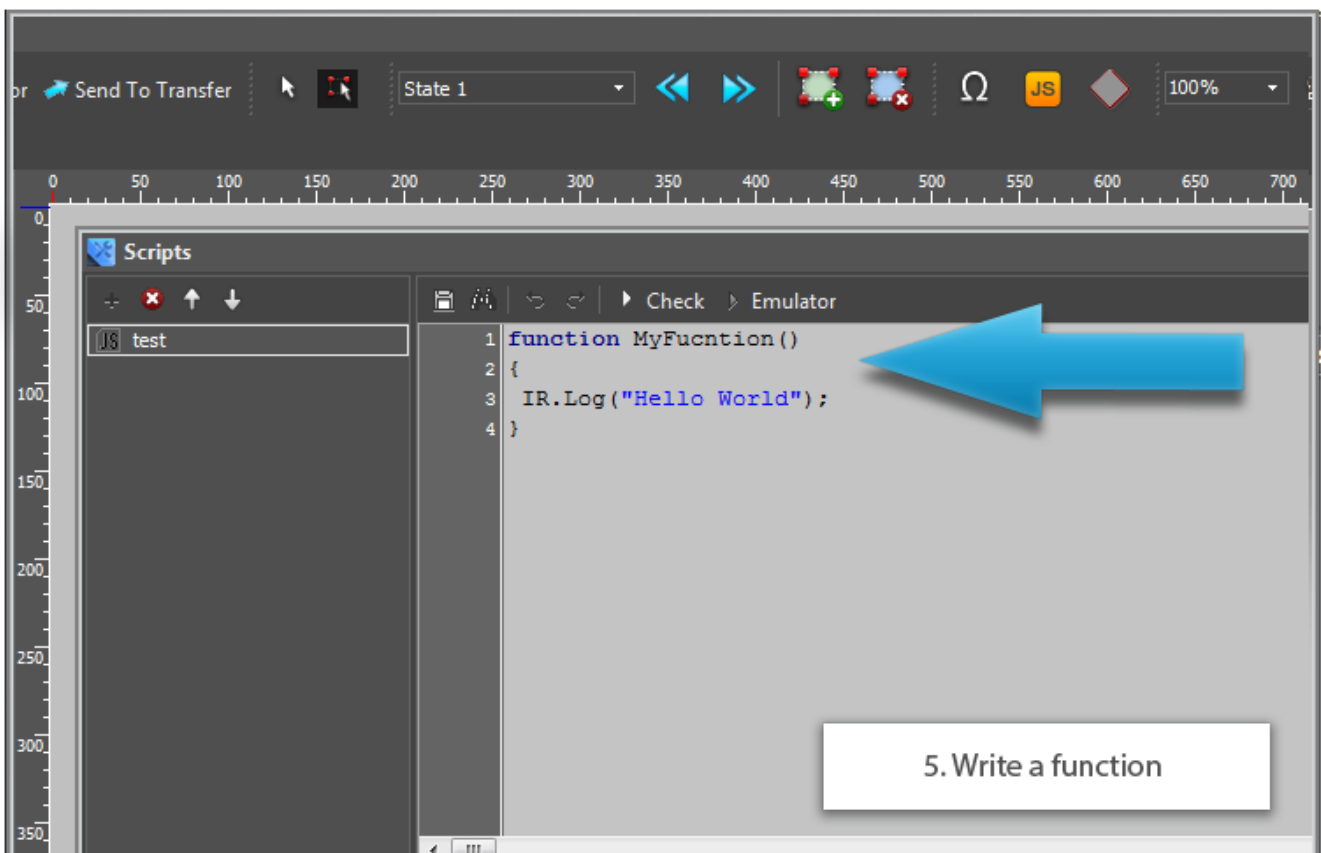
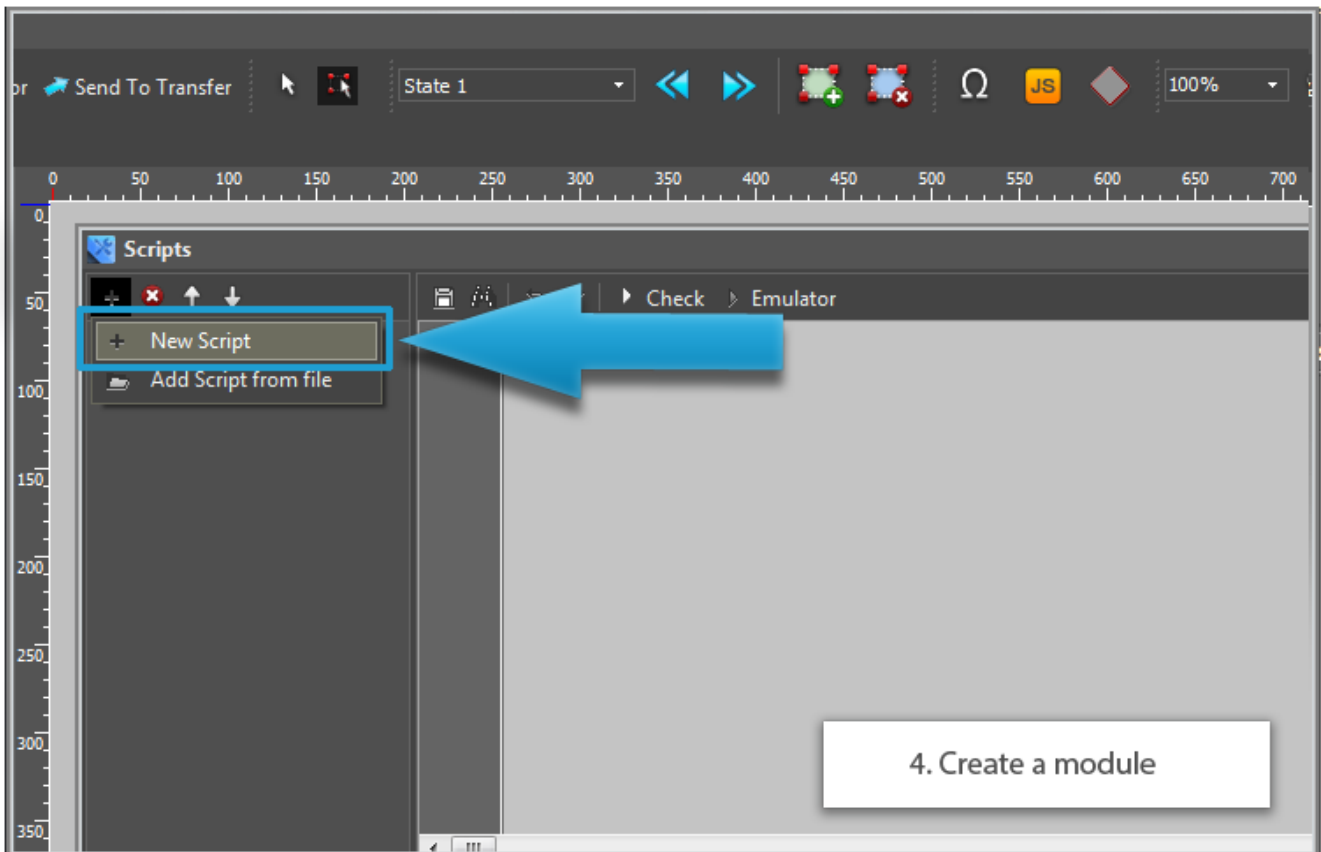
## Example of Script Launching Using the "Script Call" Macros

[DOWNLOAD: Example of a project](#)

This example describes the algorithm of script launching using **Script Call** and means of **GUI Editor**.







- Example of written function:

```
function MyFunction()  
{  
  IR.Log("Hello World");  
}
```

