Contents

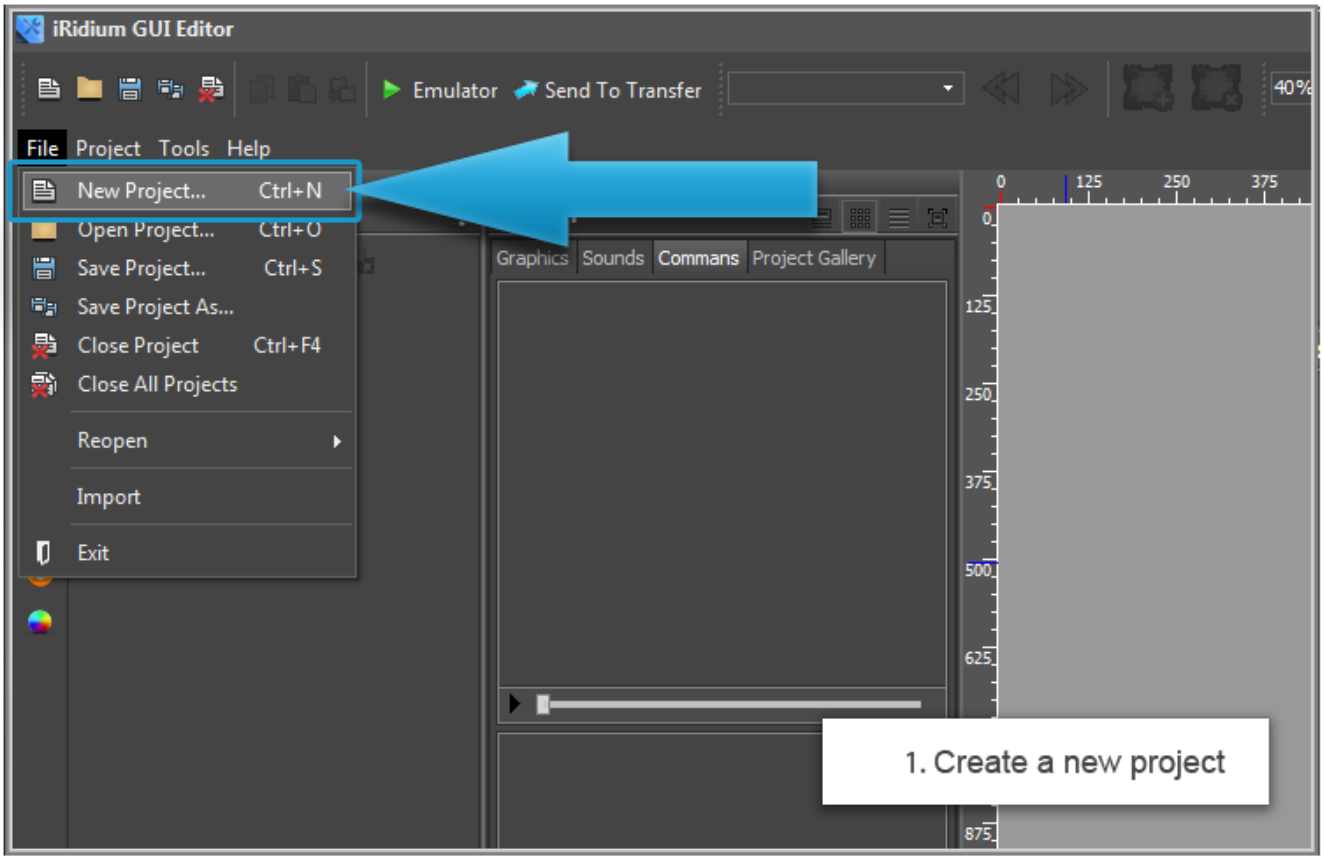
Launching Scripts by Creating Listeners

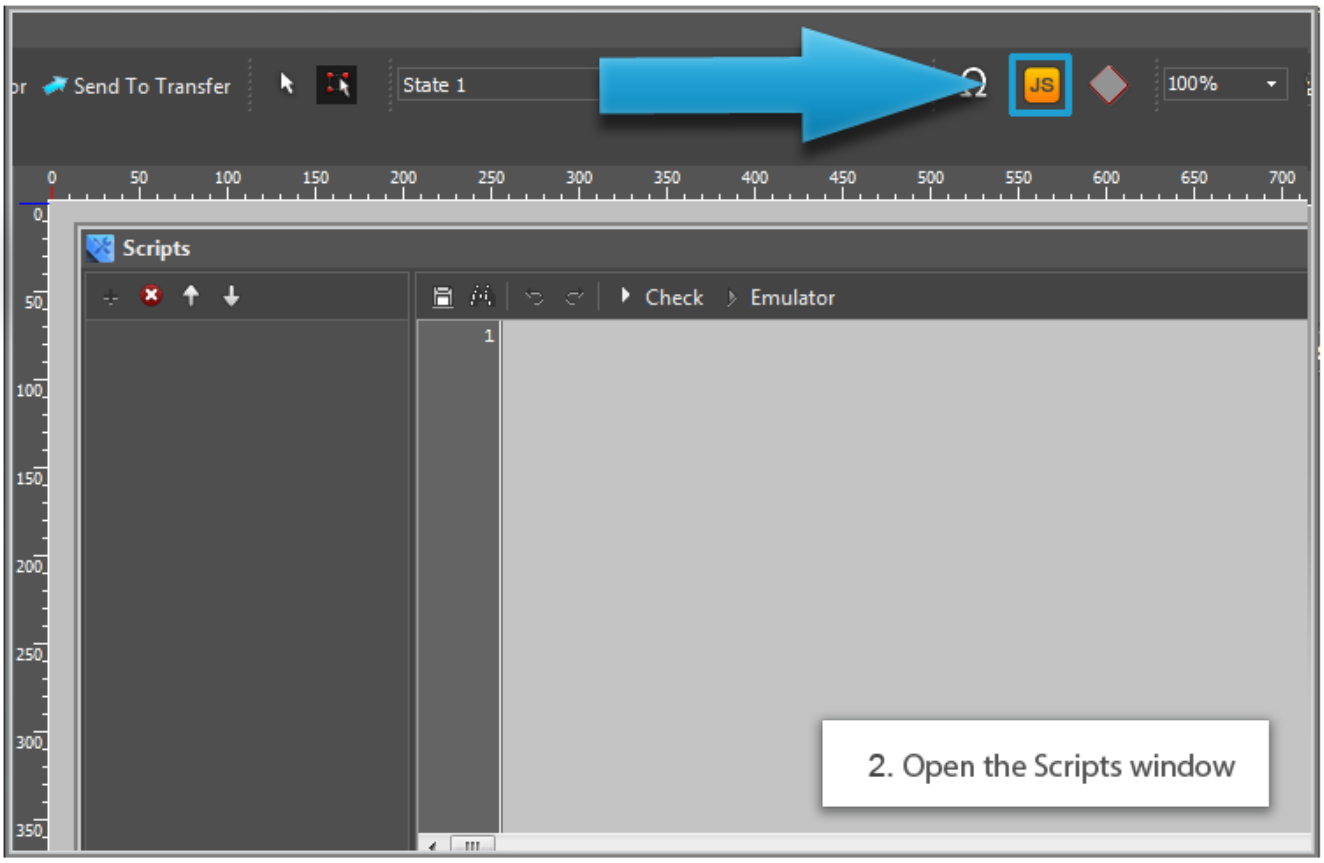
Listener – the function which is performed when the event assigned to it is activated.

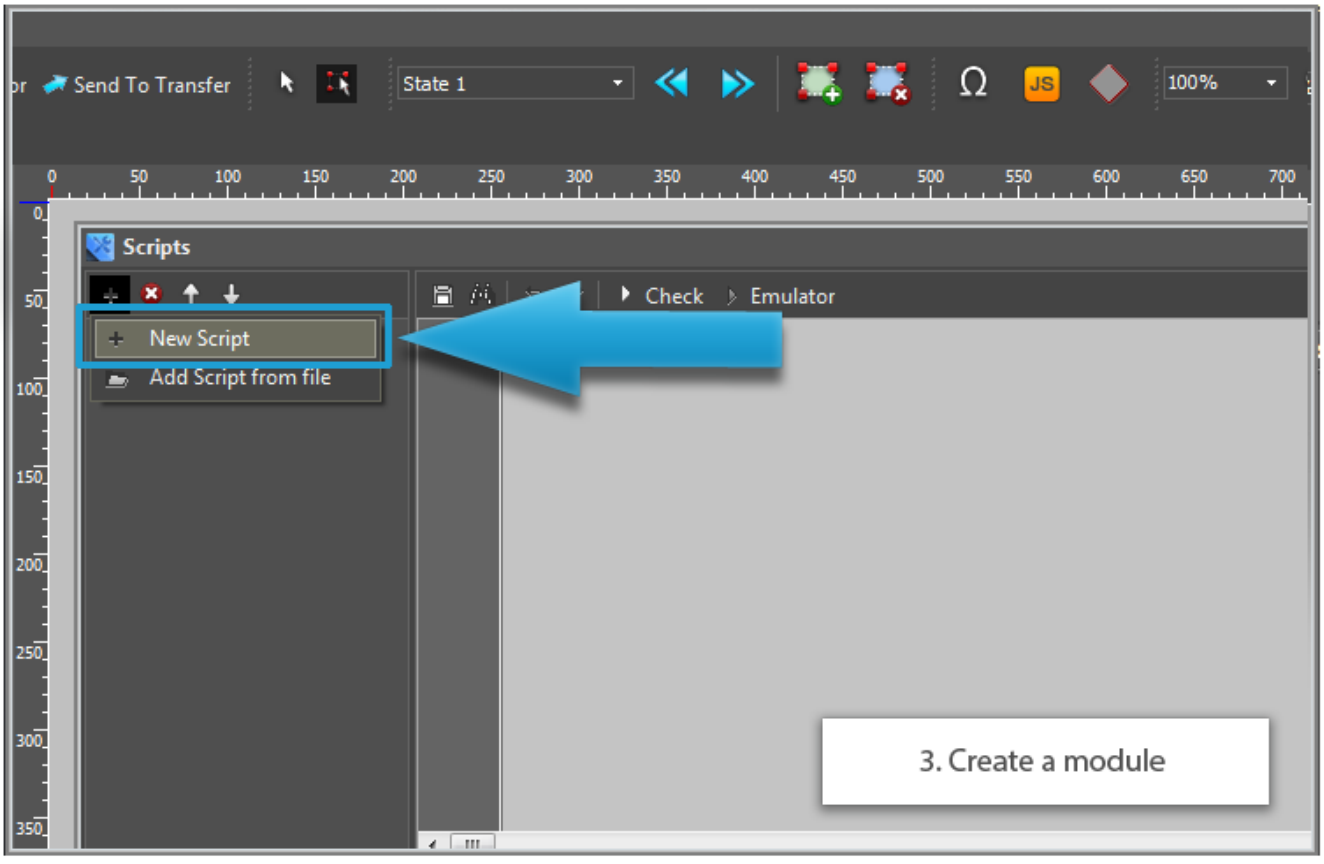
```
// Assign Listener for the event of the application launch
IR.AddListener(IR.EVENT_START, 0, function()
{
   // Body of the Listener
});
```

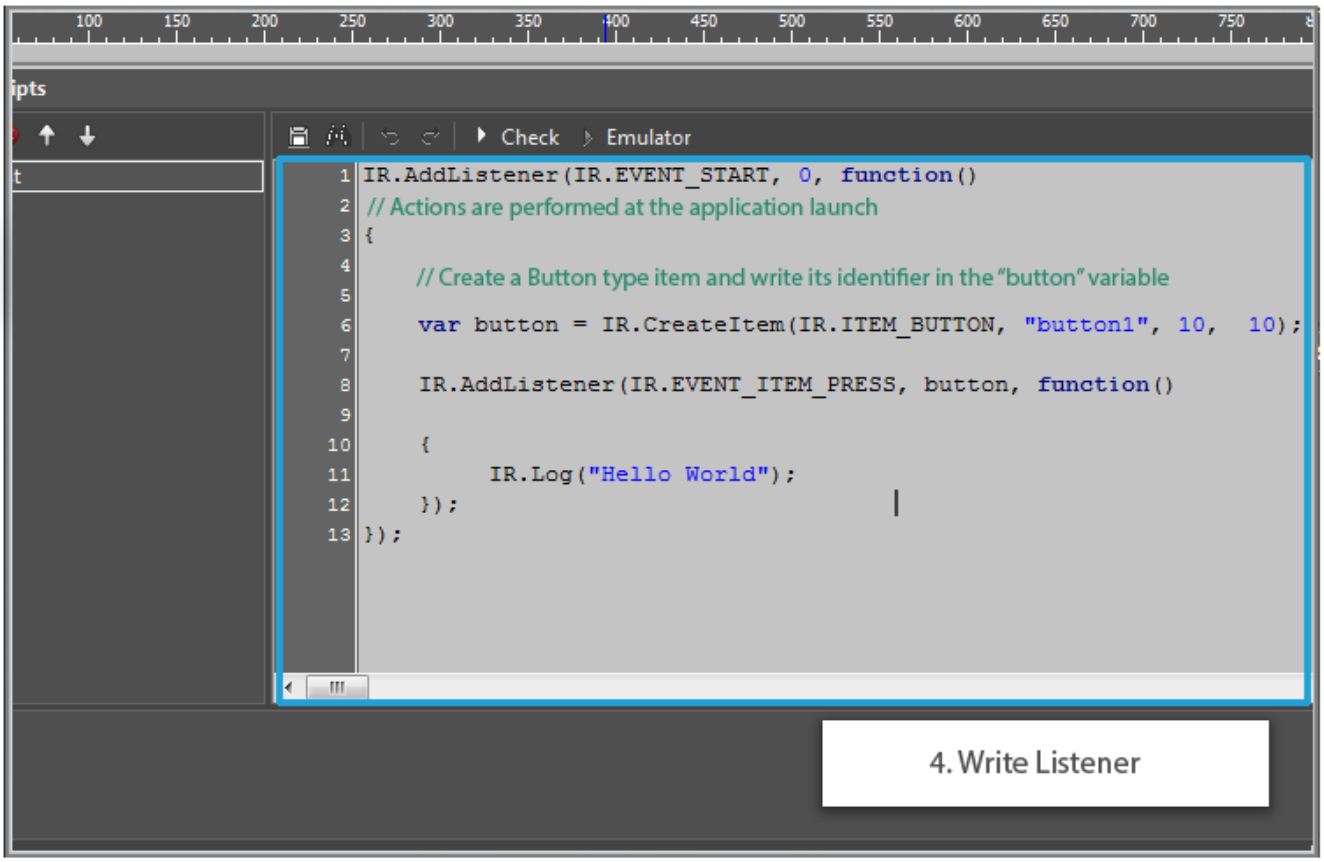
Example of Script Launching by Listeners

DOWNLOAD: Example of a project




Send To Transfer
State 1
JS
100%
Scripts
Check
Emulator
1
2. Open the Scripts window

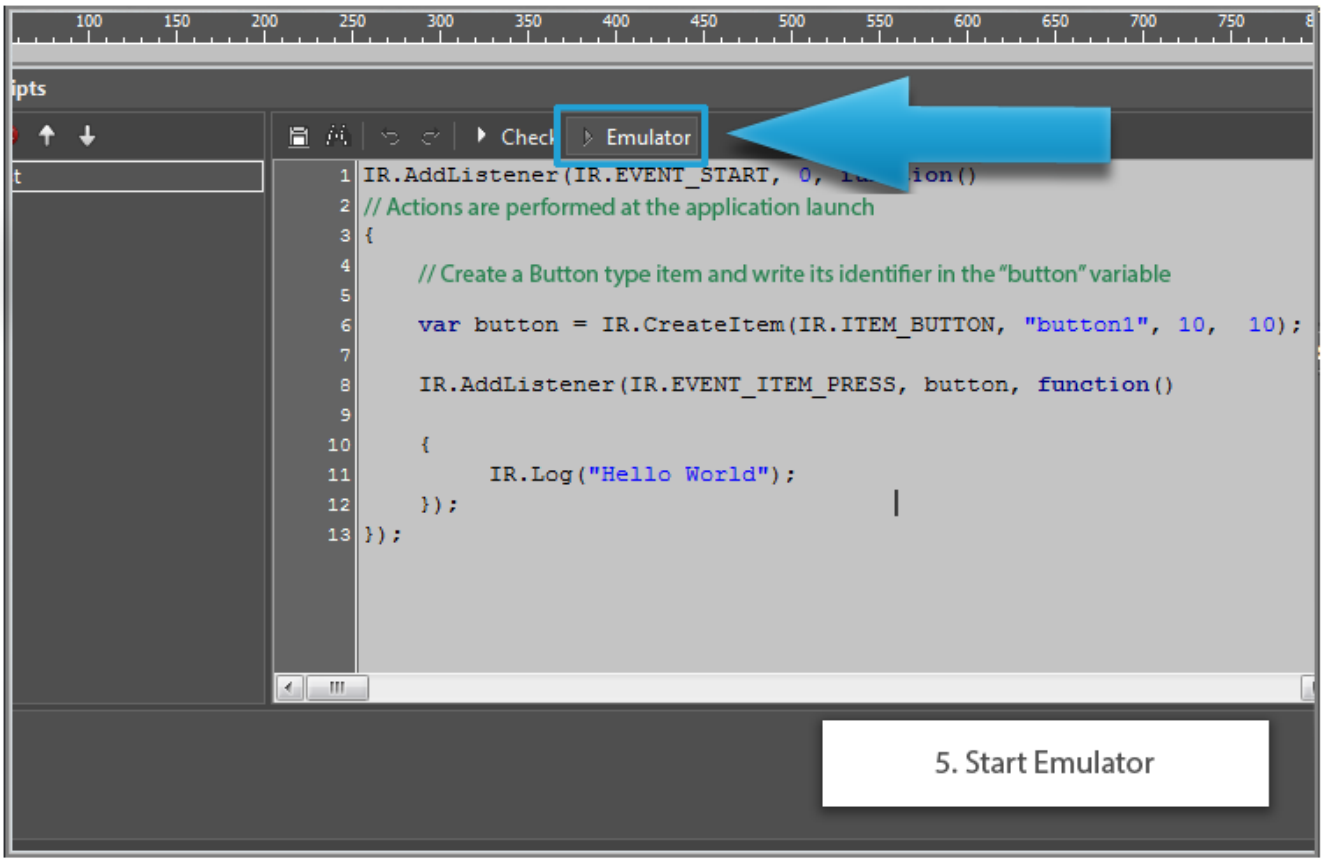

Send To Transfer
State 1
JS
100%
Scripts
Check
Emulator
New Script
Add Script from file
3. Create a module

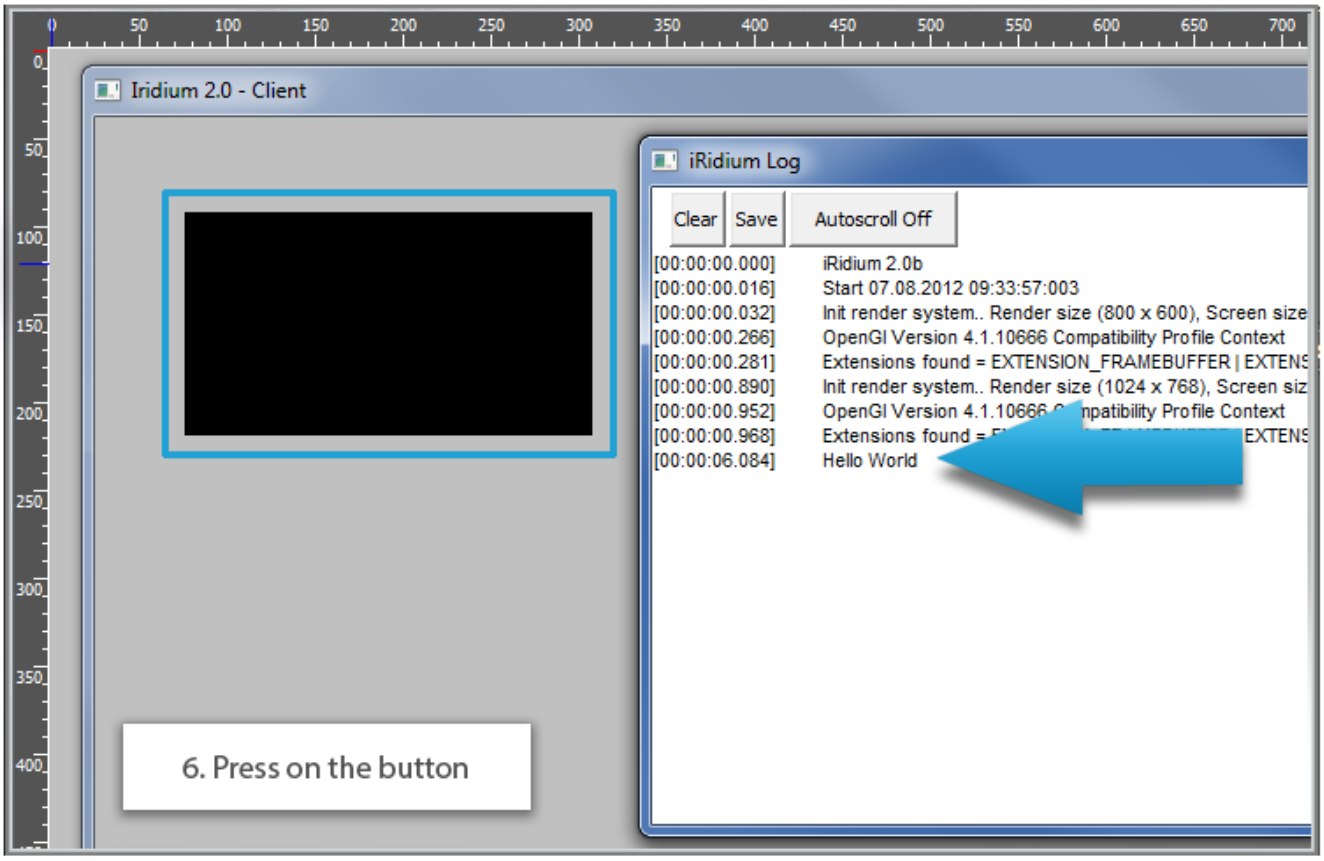


- Example of written Listener:

```
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");
    });
});
```


Emulator
1 IR.AddListener(IR.EVENT_START, 0,
2 // Actions are performed at the application launch
3 {
4 // Create a Button type item and write its identifier in the "button" variable
5
6 var button = IR.CreateItem(IR.ITEM_BUTTON, "button1", 10, 10);
7
8 IR.AddListener(IR.EVENT_ITEM_PRESS, button, function()
9
10 {
11 IR.Log("Hello World");
12 });
13 });
5. Start Emulator


Iridium 2.0 - Client
iRidium Log
Clear
Save
Autoscroll Off
[00:00:00.000] iRidium 2.0b
[00:00:00.016] Start 07.08.2012 09:33:57:003
[00:00:00.032] Init render system.. Render size (800 x 600), Screen size
[00:00:00.266] OpenGl Version 4.1.10666 Compatibility Profile Context
[00:00:00.281] Extensions found = EXTENSION_FRAMEBUFFER | EXTENS
[00:00:00.890] Init render system.. Render size (1024 x 768), Screen siz
[00:00:00.952] OpenGl Version 4.1.10666
[00:00:00.968] Extensions found =
[00:00:06.084] Hello World
6. Press on the button

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");
});
```

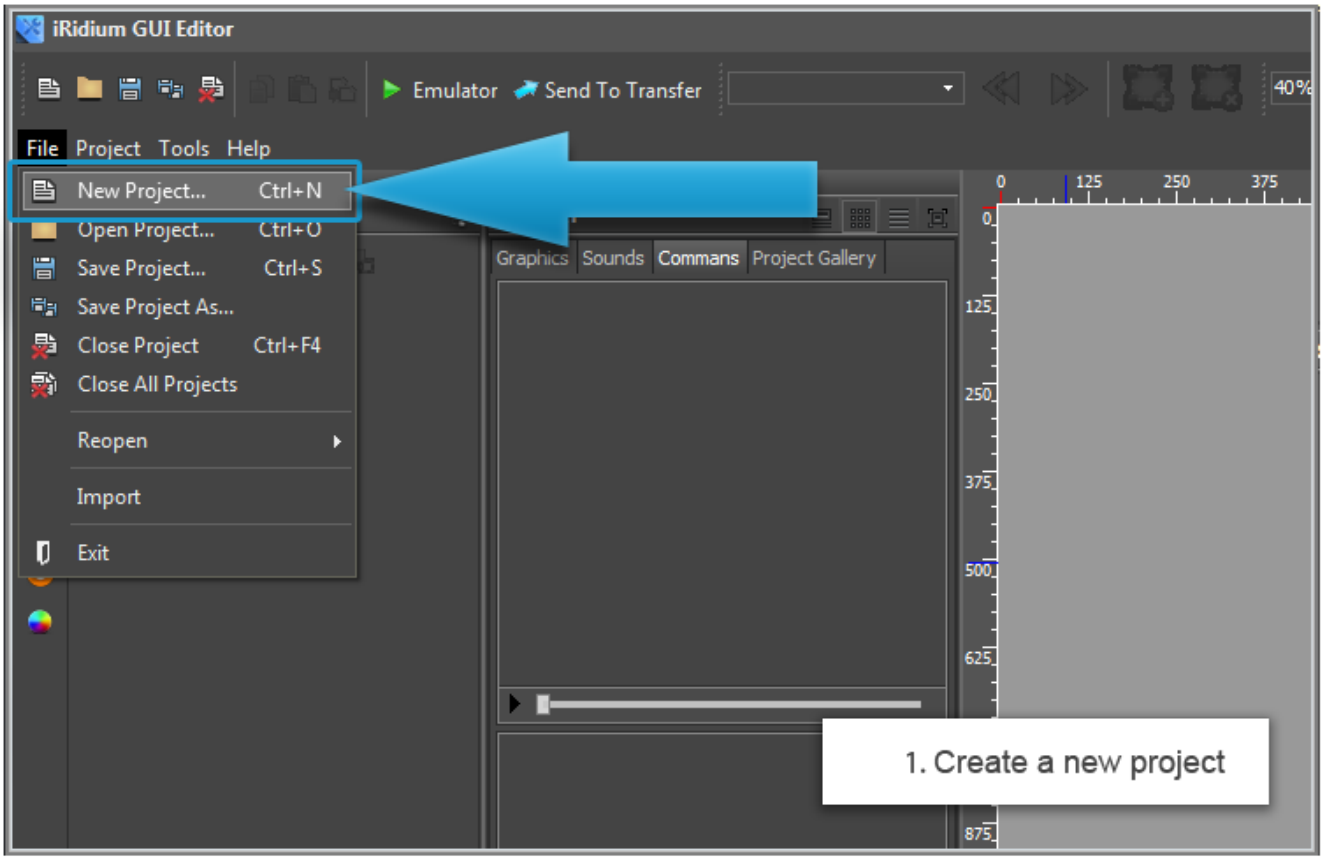
- **IR.EVENT_ITEM_CHANGE** – changing the Editbox item

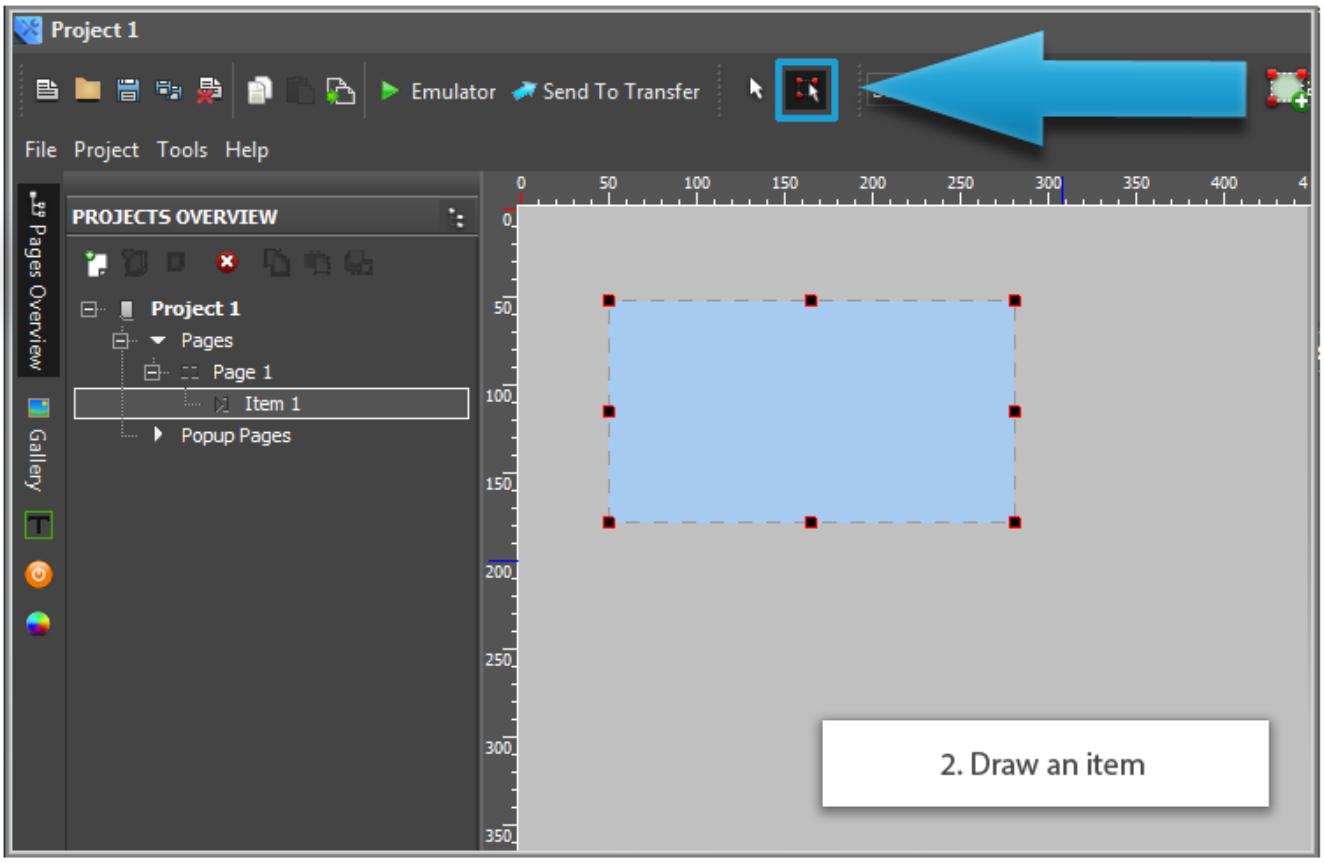
```
IR.AddListener(IR.EVENT_ITEM_CHANGE, IR.GetItem("Page 1").GetItem("Item
1"),function()
//Event is activated at change
{
 IR.Log(IR.GetItem("Page 1").GetItem("Item 1").Text);
//Outputting the changes in the log
});
```

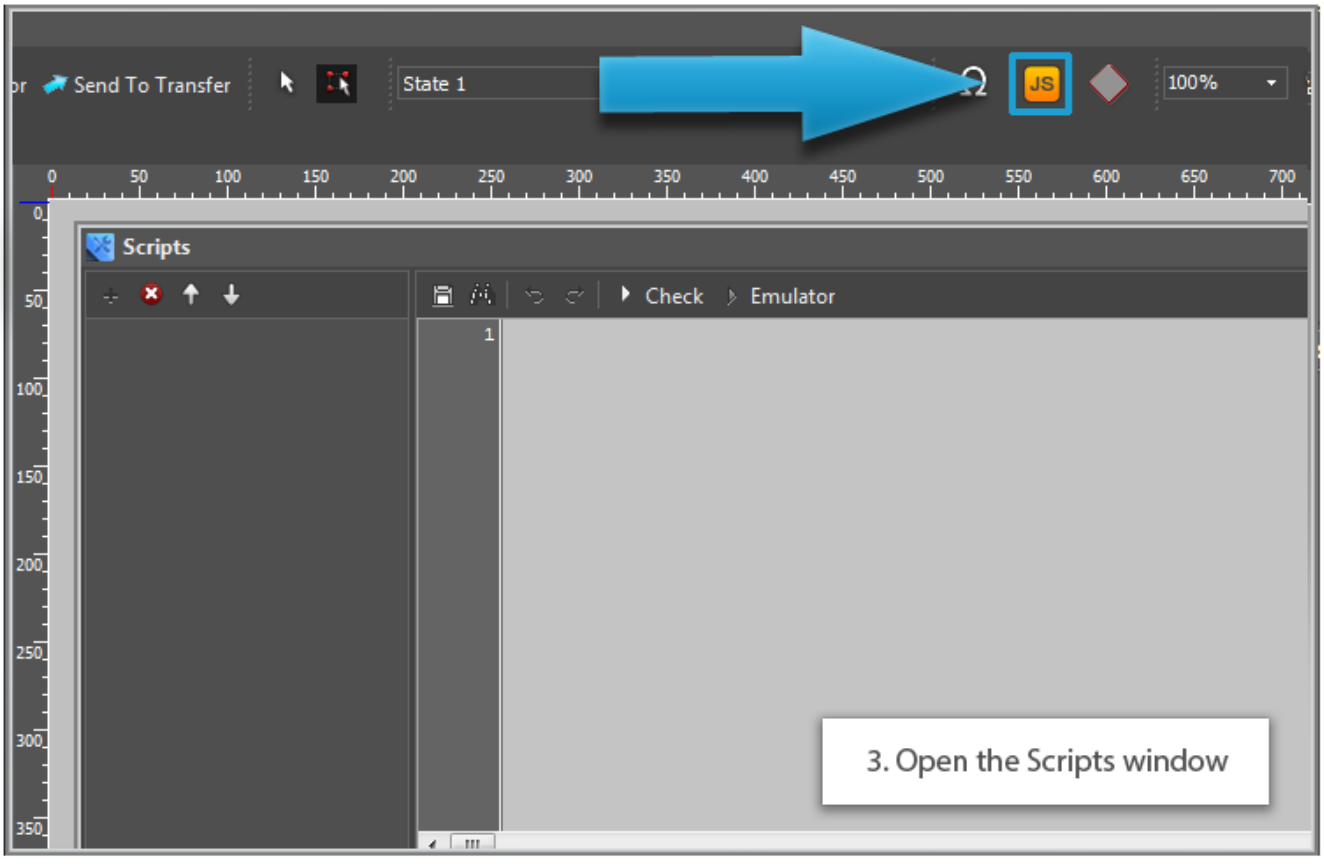
Example of Script Launching Using the "Script Call" Macros

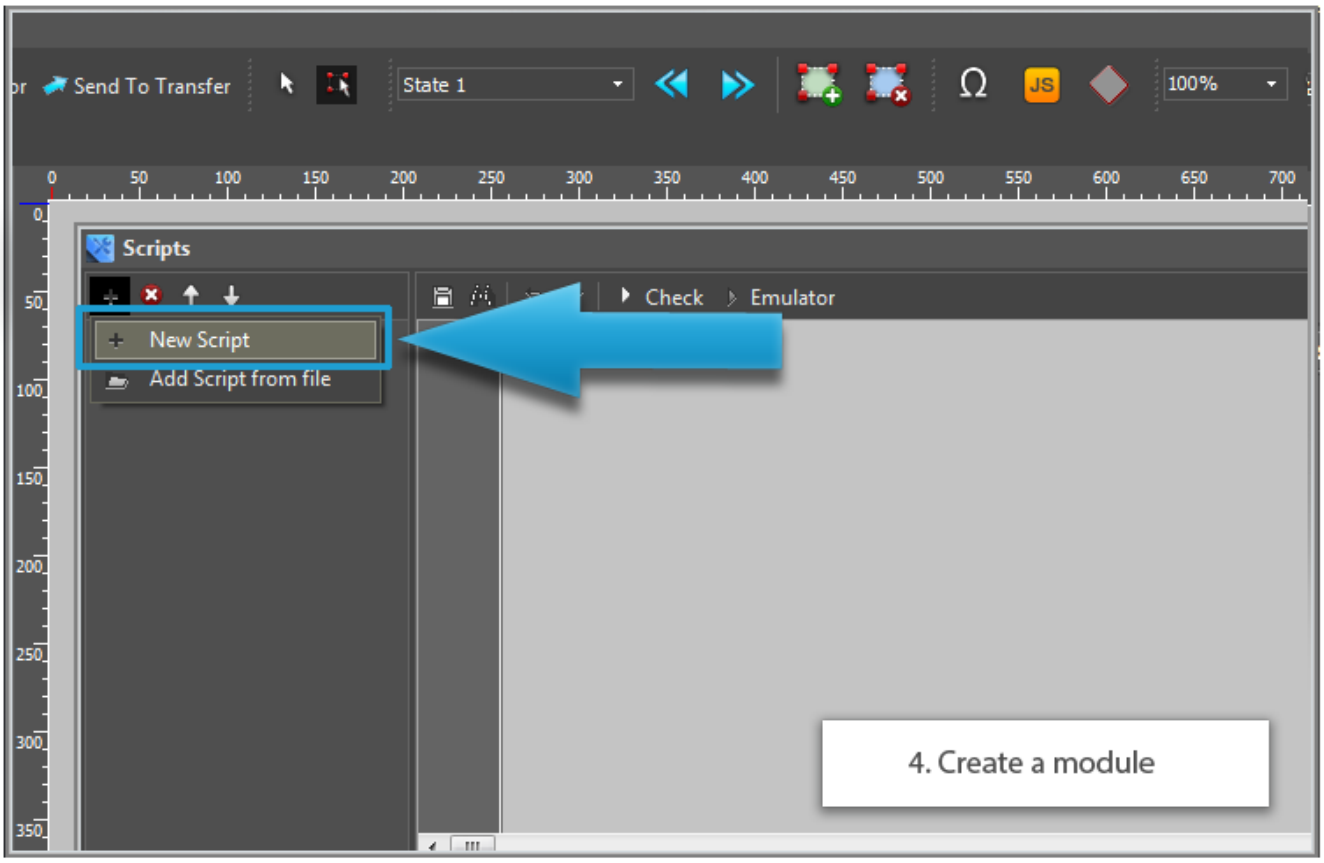
DOWNLOAD: Example of a project

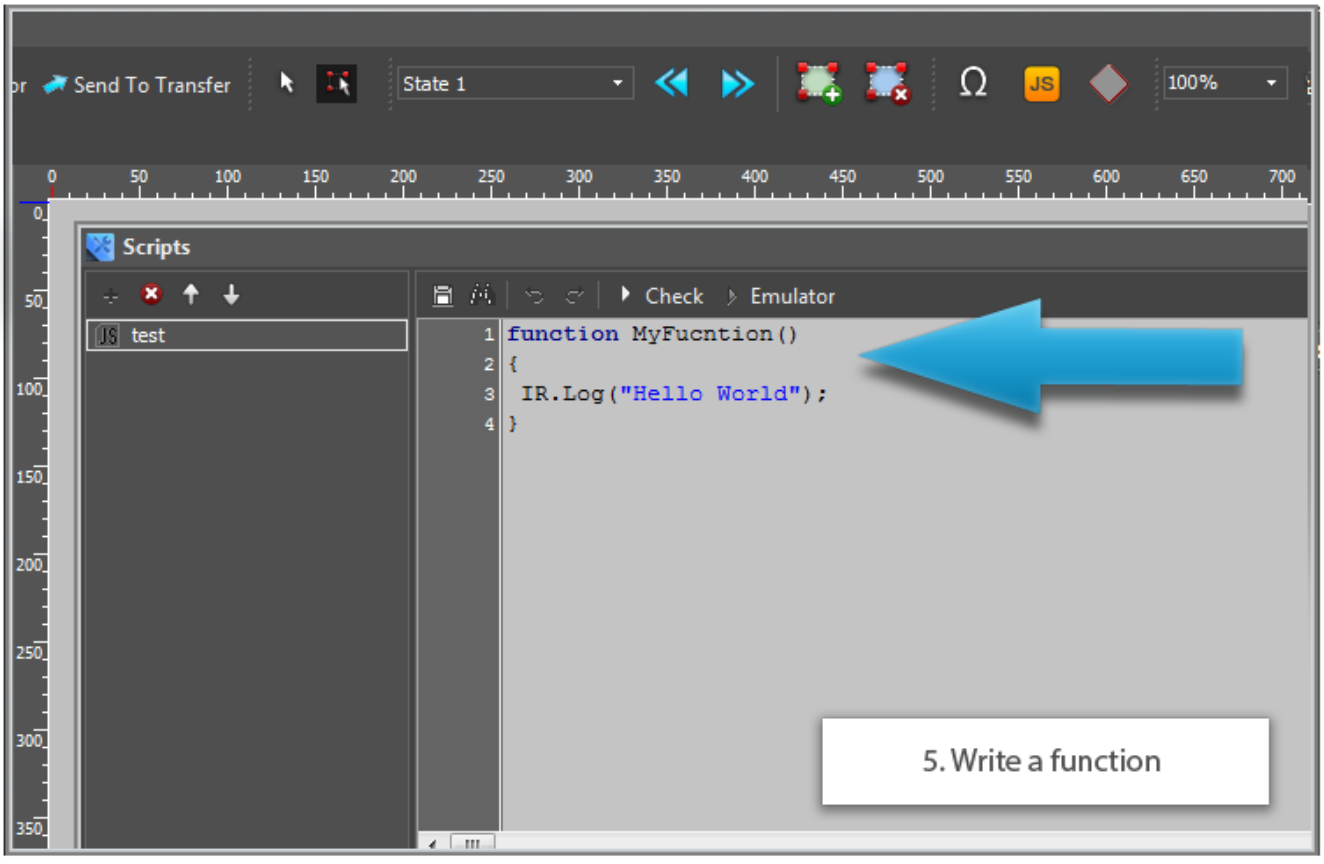
This example describes the algorithm of script launching using **Script Call** and means of **GUI** Editor.




Project 1
Emulator
Send To Transfer
File
Project
Tools
Help
PROJECTS OVERVIEW
Project 1
Pages
Page 1
Item 1
Popup Pages
Pages Overview
Gallery
2. Draw an item

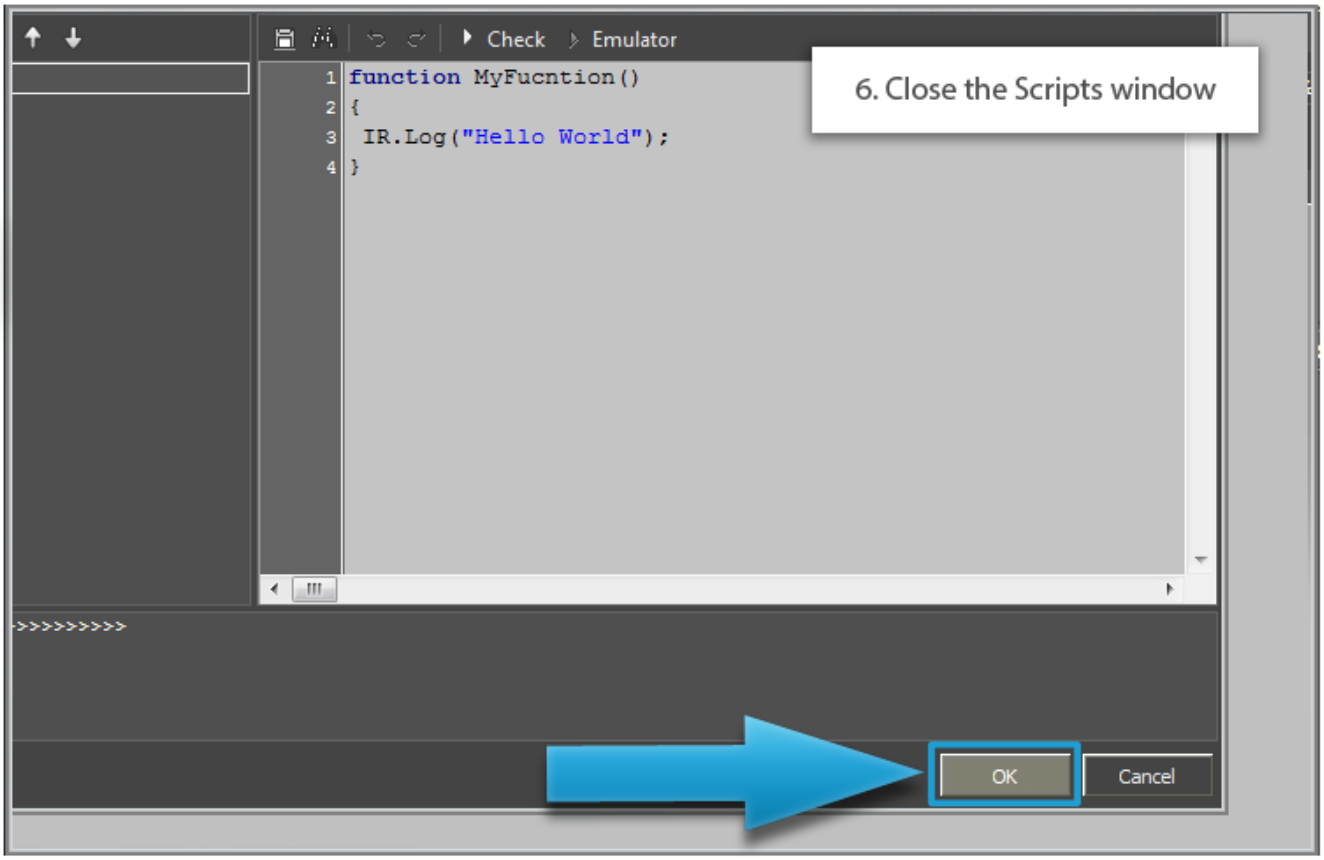

Send To Transfer
State 1
JS
100%
Scripts
Check
Emulator
3. Open the Scripts window

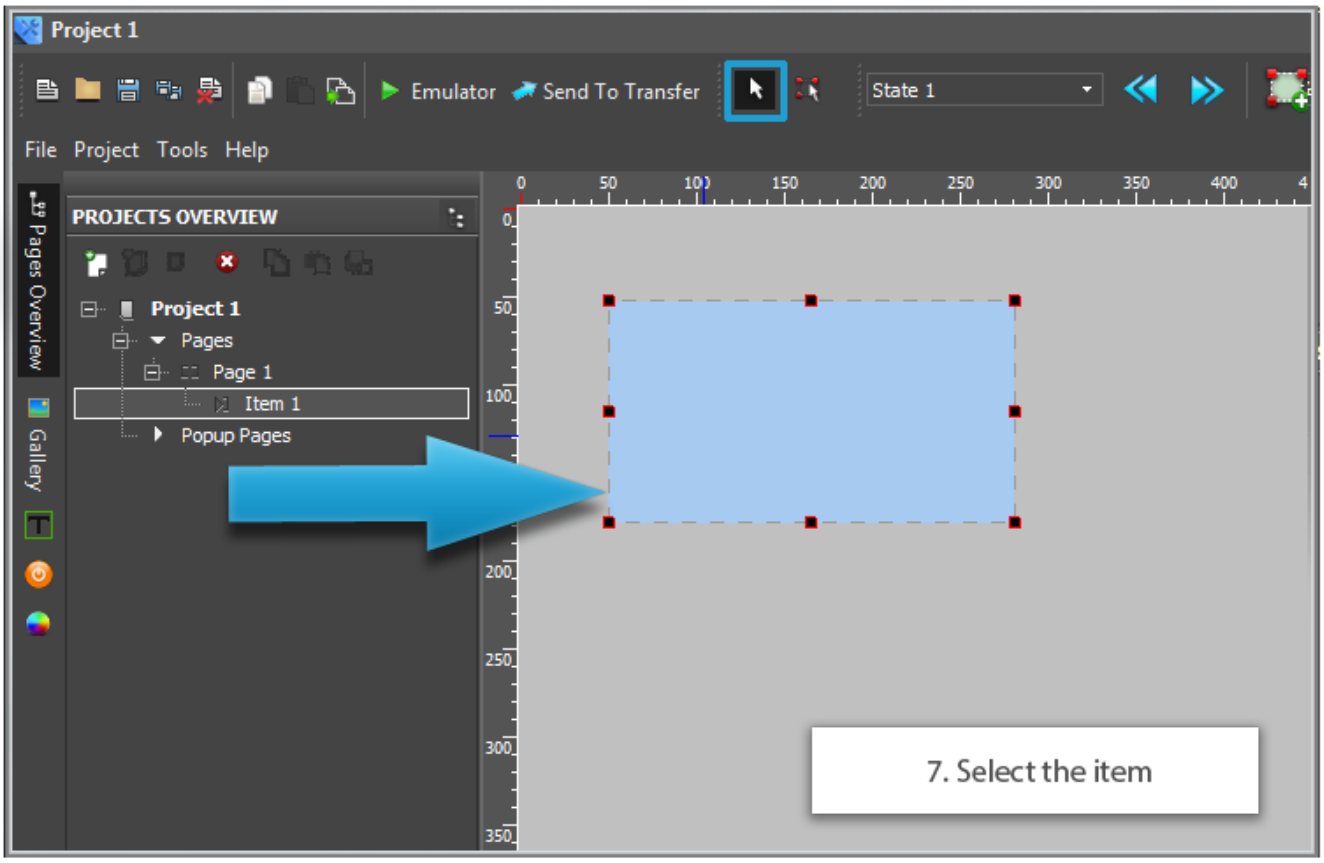


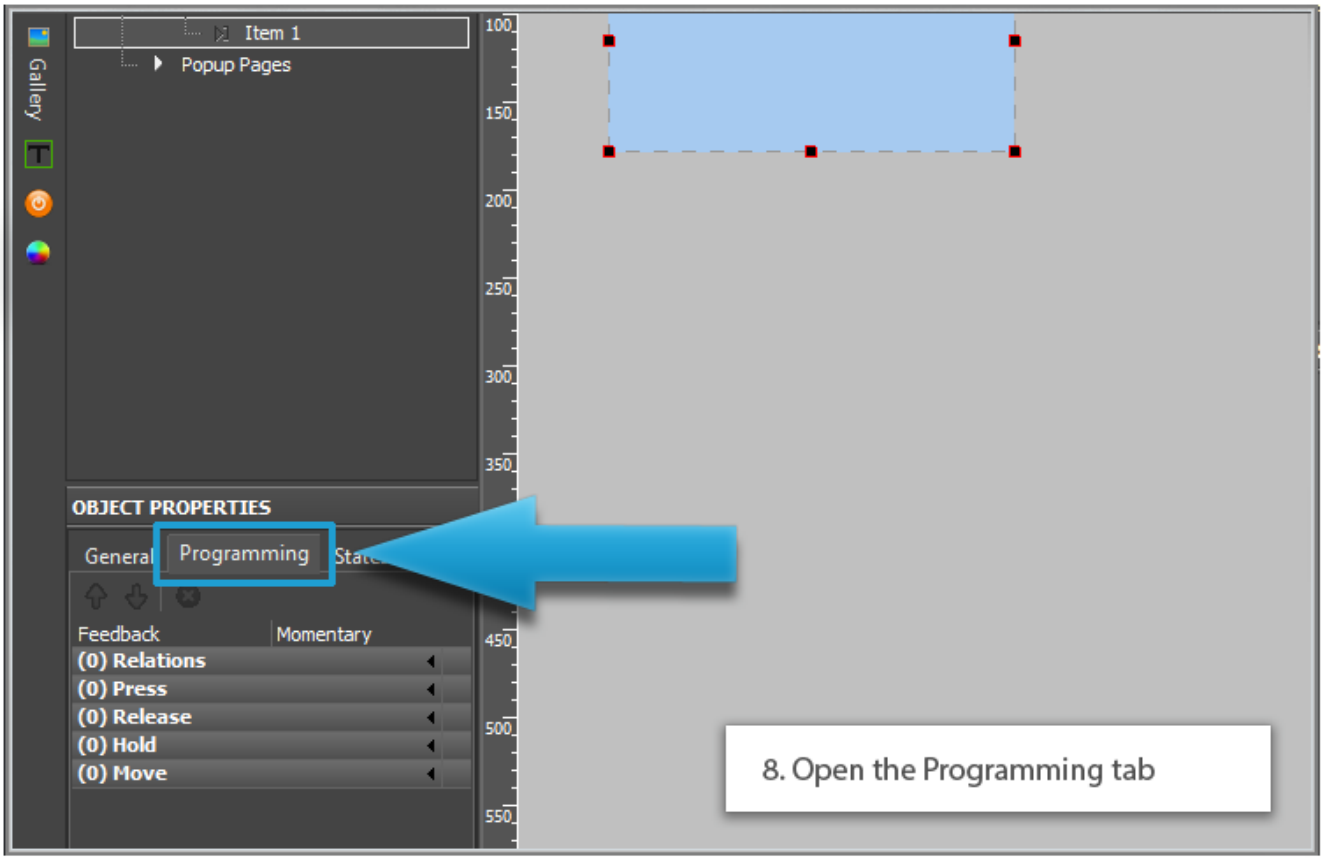


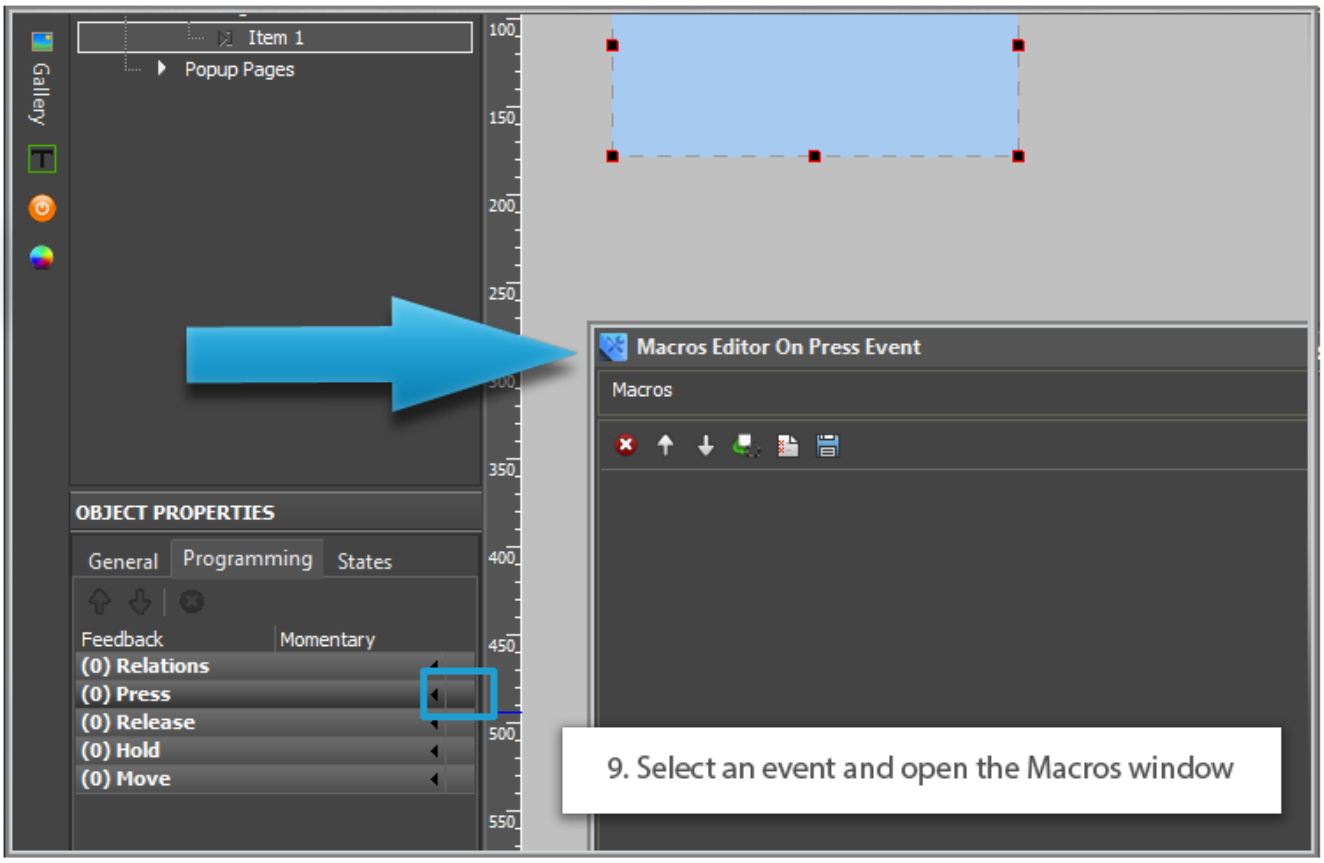
- Example of written function:

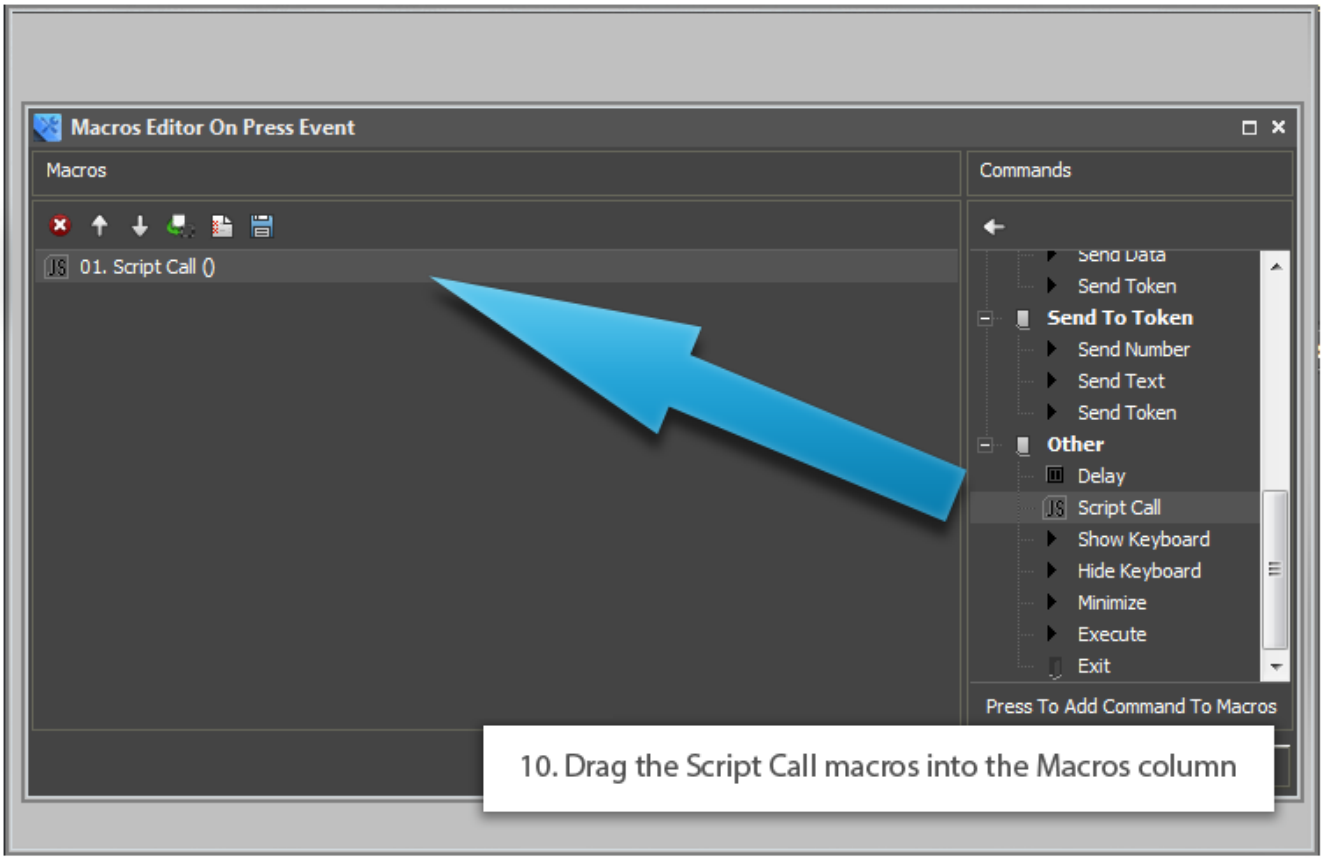
```
function MyFunction()
{
 IR.Log("Hello World");
}
```

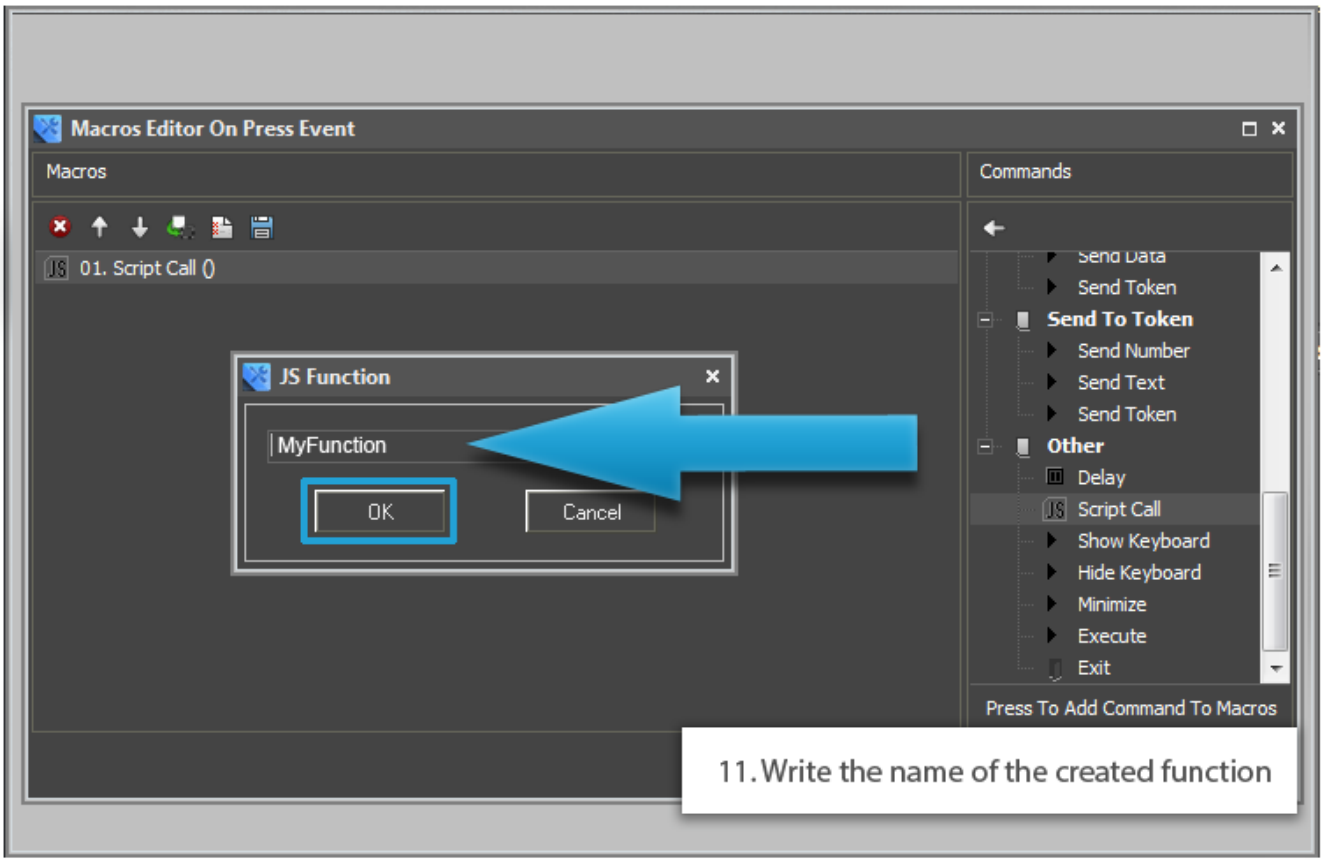


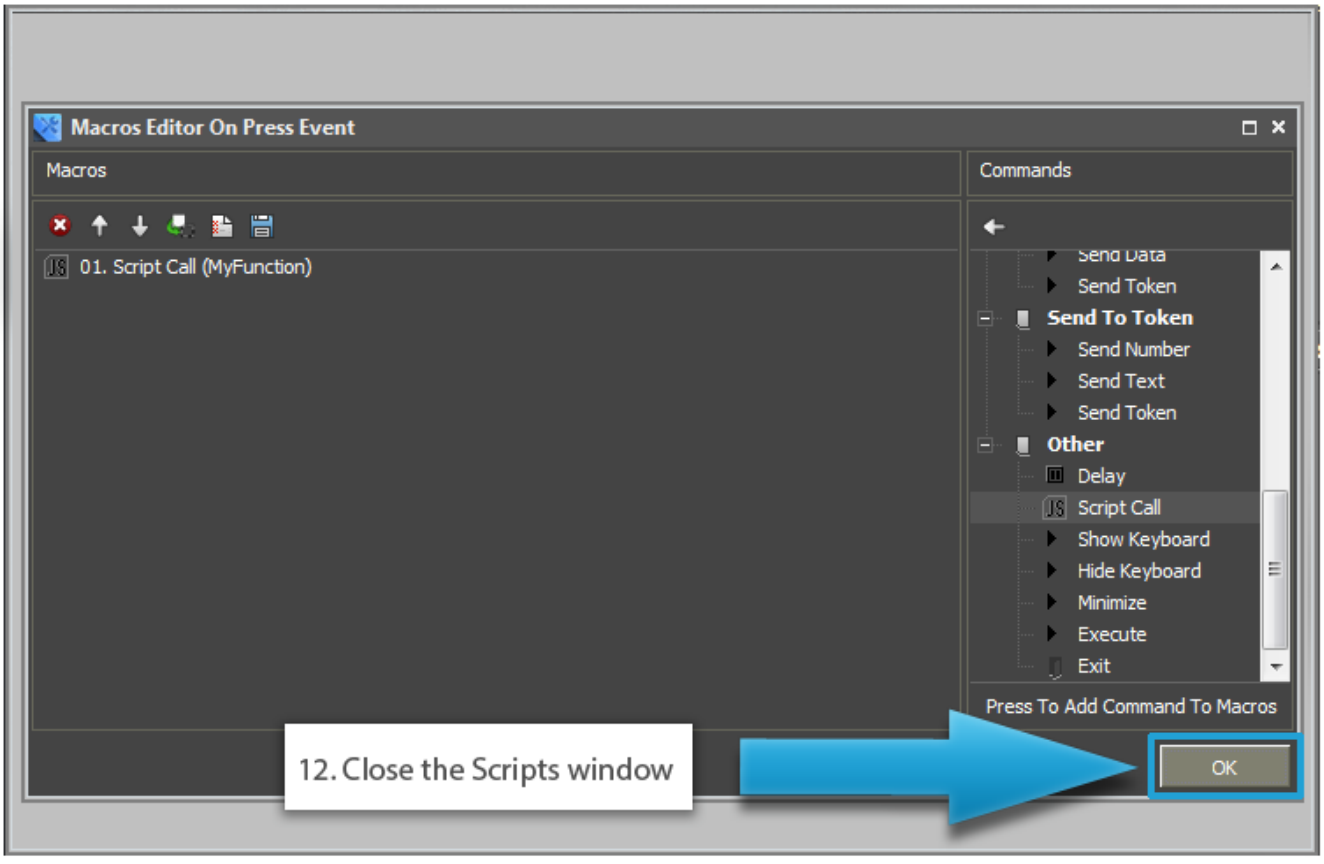

Project 1
Emulator
Send To Transfer
State 1
File
Project
Tools
Help
PROJECTS OVERVIEW
Project 1
Pages
Page 1
Item 1
Popup Pages
Pages Overview
Gallery
7. Select the item

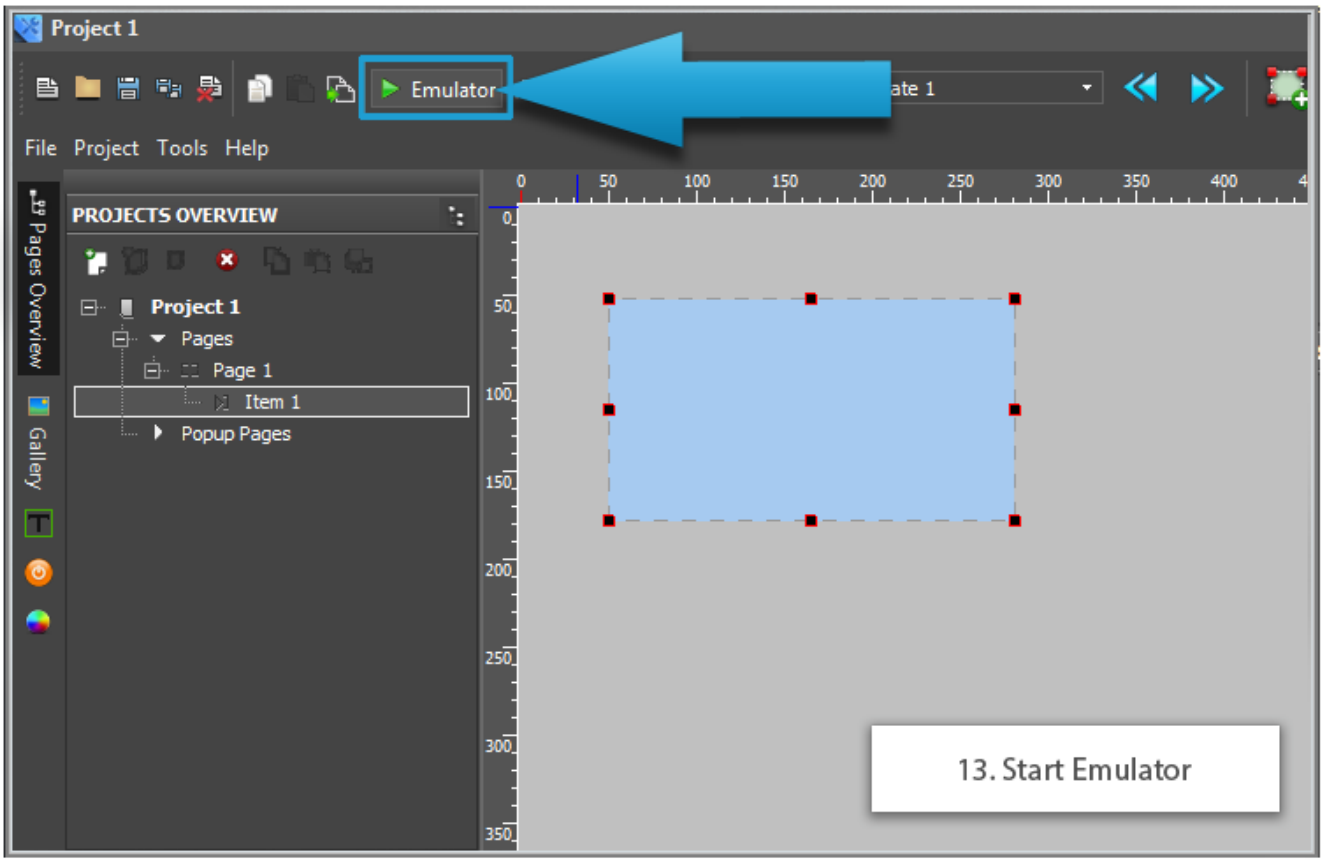

Item 1
Popup Pages
Gallery
OBJECT PROPERTIES
General
Programming
Feedback
Momentary
(0) Relations
(0) Press
(0) Release
(0) Hold
(0) Move
8. Open the Programming tab

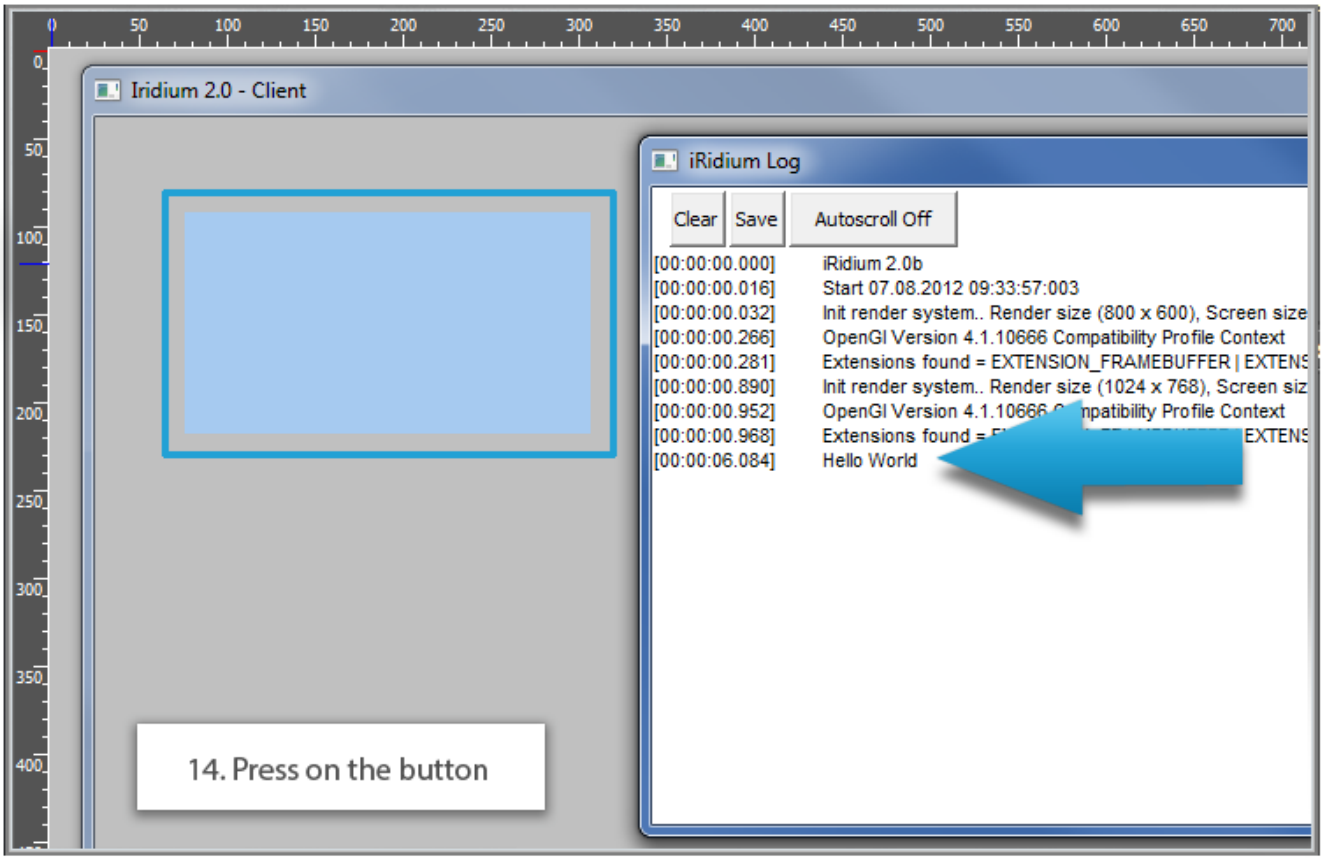

Gallery
Item 1
Popup Pages
OBJECT PROPERTIES
General
Programming
States
Feedback
Momentary
(0) Relations
(0) Press
(0) Release
(0) Hold
(0) Move
Macros Editor On Press Event
Macros
9. Select an event and open the Macros window


Macros Editor On Press Event
Macros
01. Script Call ()
Commands
Send Data
Send Token
Send To Token
Send Number
Send Text
Send Token
Other
Delay
Script Call
Show Keyboard
Hide Keyboard
Minimize
Execute
Exit
Press To Add Command To Macros
10. Drag the Script Call macros into the Macros column


Macros Editor On Press Event
Macros
Commands
01. Script Call ()
JS Function
MyFunction
OK
Cancel
Send Data
Send Token
Send To Token
Send Number
Send Text
Send Token
Other
Delay
Script Call
Show Keyboard
Hide Keyboard
Minimize
Execute
Exit
Press To Add Command To Macros
11. Write the name of the created function


Macros Editor On Press Event
Macros
Commands
01. Script Call (MyFunction)
Send Data
Send Token
Send To Token
Send Number
Send Text
Send Token
Other
Delay
Script Call
Show Keyboard
Hide Keyboard
Minimize
Execute
Exit
Press To Add Command To Macros
12. Close the Scripts window
OK


Project 1
Emulator
File Project Tools Help
PROJECTS OVERVIEW
Project 1
Pages
Page 1
Item 1
Popup Pages
Pages Overview
Gallery
13. Start Emulator


Iridium 2.0 - Client
iRidium Log
Clear
Save
Autoscroll Off
[00:00:00.000] iRidium 2.0b
[00:00:00.016] Start 07.08.2012 09:33:57:003
[00:00:00.032] Init render system.. Render size (800 x 600), Screen size
[00:00:00.266] OpenGl Version 4.1.10666 Compatibility Profile Context
[00:00:00.281] Extensions found = EXTENSION_FRAMEBUFFER | EXTENS
[00:00:00.890] Init render system.. Render size (1024 x 768), Screen siz
[00:00:00.952] OpenGl Version 4.1.10666
[00:00:00.968] Extensions found =
[00:00:06.084] Hello World
14. Press on the button