All actions in iRidiumScript are preformed in response to some event. There are the following events in iRidiumScript:

- **IR.EVENT_START** - starting Client's work

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

- **IR.EVENT_WORK** - actions are performed in cycles while the Client is running

```
var timer = 0;
var onTime = 800; // 800 ms = 0.8 s
IR.AddListener(IR.EVENT_WORK, 0, function(time)  //Actions are performed
while the Client's working
 {
        timer += time;
        if(timer > onTime)
        {
                timer = 0;
                IR.Log("Click");

        }
});
```

- **IR.EVENT_EXIT** - exiting the Client

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

- **IR.EVENT_ITEM_PRESS** - pressings on an item (object, popup,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); // Deleter the third list item
     });
```

- **IR.EVENT_GESTURE_BEGIN** - beginning of gesture

```
var button;

// Activated at start
IR.AddListener(IR.EVENT_START, 0, function()
{
        // Create a button for moving 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 a device (driver)

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

- **IR.EVENT_OFFLINE** - disconnection of a device (driver)

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

- **IR.EVENT_RECEIVE_DATA** - receiving data from a device (binary data)

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

- **IR.EVENT_RECEIVE_TEXT** - receiving data from a device (string format)

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

- **IR.EVENT_TAG_CHANGE** - changing a tag value

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

- **IR.EVENT_KEYBOARD_SHOW** - opening of 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 an EditBox item

```
IR.AddListener(IR.EVENT_ITEM_CHANGE, IR.GetItem("Page 1").GetItem("Item
1"),function()  //Action is performed at change
{
 IR.Log(IR.GetItem("Page 1").GetItem("Item 1").Text);  //The changes are
output in the log
});
```