

## [DOWNLOAD: Example of a project](#)

# Contents

- [1 Description](#)
- [2 Commands for Working with Items](#)
- [3 Creating Items](#)
- [4 Changing Items](#)
- [5 Deleting Items](#)
- [6 Counting Items on a Page](#)
- [7 Event Types](#)
- [8 Processing Events for Items](#)
- [9 Launching Macros Bound to an Item](#)
- [10 Working with States](#)
- [11 DOWNLOAD: Example of a project](#)

## Description

Using iRidiumScript you can create items and control item properties:

- Creating items
- Changing item properties

## Commands for Working with Items

There are the following commands for working with items:

- Creating items
- Changing items
- Deleting
- Counting
- Assigning events

## Creating Items

To create items use the command:

```
IR.CreateItem(Item_Type, Item_Name, Coordinate_X, Coordinate_Y, Item_Width, Item_Height);
```

- **Item\_Type** - indicate the type of the created item;
- **Item\_Name** - give the name to the item;
- **Coordinate\_X, Coordinate\_Y** - initial position of the item in the working area;
- **Item\_Width, Item\_Height** - width and height of the created item

```
IR.AddListener(IR.EVENT_START,0, function() //Event is activated at  
the application launch
```

```
{
IR.CreateItem(IR.ITEM_BUTTON,"Button",30,40,800,150);    //Creating an item
});
```

## Changing Items

To change items use the command:

```
IR.GetItem("Page_of_Item_Location").IRGetItem("Item_Name")
```

- **Page\_of\_Item\_Location** - the page with the item
- **Item\_Name** - the name of the item being changed

```
IR.AddListener(IR.EVENT_START,0, function()    //Event is activated at the
application launch
```

```
{
IR.CreateItem(IR.ITEM_BUTTON,"Button");    //Creating an item
```

```
IR.GetItem("Page 1").GetItem("Button").Width = 500;    // Changing the item
width
```

```
IR.GetItem("Page 1").GetItem("Button").Height = 100;    //Changing the item
height
```

```
IR.GetItem("Page 1").GetItem("Button").X = 0;    //Changing the item X-
coordinate
```

```
IR.GetItem("Page 1").GetItem("Button").Y = 100;    //Changing the item Y-
coordinate
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).Border = 5;    //Changing the
item border
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).Text = "This is new
items";    //Changing the text displayed on the item
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).Opacity = 200;    //Changing
the item opacity(0-255)
```

```
//Changing the image displayed on the item
```

```
//(To receive the name of the image use IntellHelp - press ctrl+space after
the command IR.GetItem("Page 1").GetItem("Button").GetState(0).Image =)
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).Image = "pict.jpg";
```

```
//Changing the icon displayed on the item
```

```
//(To receive the name of the icon use IntellHelp - press ctrl+space after
the command IR.GetItem("Page 1").GetItem("Button").GetState(0).Icon =)
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).Icon = "pict.jpg";
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).Font = "Tohoma";
```

```
//Changing the text font[text size]
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).TextEffect = 1;
//Adding font effects

});
```

## Deleting Items

- *under development*

## Counting Items on a Page

To learn the number of items on the page use the command:

### **IR.CurrentPage.ItemsCount**

The command gives the number of items on the selected page. The received number can be saved in the variable or displayed on an item:

```
IR.AddListener(IR.EVENT_START, 0, function() // Event is activated at
application launch
{
  // Receive the identifier of the item where the number of items is going to
be displayed
  var ShowCount = IR.GetItem("Page 1").GetItem("Item 107");

  // Receive the number of items and display
  ShowCount.Text = IR.CurrentPage.ItemsCount;
});
```

You can count the number of items on any other available page by substituting:

### **IR.CurrentPage** to **IR.GetItem("Page\_Name")**

- **Page\_Name** - the name of the page where the items will be counted on

```
IR.AddListener(IR.EVENT_START, 0, function() // Event is activated at
application launch
{
  // Receive the identifier of the item where the number of items is going to
be displayed
  var ShowCount = IR.GetItem("Page 1").GetItem("Item 107");

  // Receive the number of items and display
  ShowCount.Text = IR.GetItem("Page 2").ItemsCount;
});
```

# Event Types

To make the application interactive use the following events:

- EVENT\_ITEM\_PRESS - pressing

```
IR.AddListener(IR.EVENT_START, 0, function()           //Event is activated at
application launch
{
    var popup = IR.CreateItem(IR.ITEM_POPUP,"POPUP", 150, 10, 200, 200);
//Creating a popup
    var button = IR.CreateItem(IR.ITEM_BUTTON, "POPUP_BUTTON", 10, 10);
//Creating an item
    button.Text = "POPUP";

    IR.AddListener(IR.EVENT_ITEM_PRESS, button, function()           //Event
is activated when pressing on the button
    {
        IR.TogglePopup("POPUP");           //Showing the
popup
    });
});
```

- EVENT\_ITEM\_RELEASE - releasing

```
IR.AddListener(IR.EVENT_START, 0, function()           //Event is activated at
application launch
{
    var popup = IR.CreateItem(IR.ITEM_POPUP, "POPUP2", 150, 10, 200, 200);
//Creating a popup
    var button = IR.CreateItem(IR.ITEM_BUTTON, "POPUP_BUTTON2", 10, 10);
//Creating an item
    button.Text = "POPUP2";

    IR.AddListener(IR.EVENT_ITEM_RELEASE, button, function()           //Event is
activated when releasing the button
    {
        IR.TogglePopup("POPUP2");           //Showing the popup
    });
});
```

- EVENT\_ITEM\_SELECT - selecting
- EVENT\_ITEM\_CHANGE - changing
- EVENT\_MOUSE\_DOWN - pressing on the mouse button

```
IR.AddListener(IR.EVENT_MOUSE_DOWN, IR.GetItem("Page 1"), function()
//Event is activated when releasing the mouse button
    {
    {
IR.GetItem("Page 1").GetItem("Level").Value = IR.GetItem("Page
1").GetItem("Level").Value -4          //Changing an item property
(Decrementing the value by 4)
});
```

- EVENT\_MOUSE\_UP - releasing the mouse button

```
IR.AddListener(IR.EVENT_MOUSE_UP, IR.GetItem("Page 1"), function()
//Event is activated when pressing on the mouse button
    {
IR.GetItem("Page 1").GetItem("Level").Value = IR.GetItem("Page
1").GetItem("Level").Value +4          //Changing an item property
(Incrementing the value by 4)
});
```

- EVENT\_MOUSE\_MOVE - moving the mouse button

```
IR.AddListener(IR.EVENT_MOUSE_MOVE, IR.GetItem("Page 1"), function()
//Event is activated when moving the mouse
    {
IR.GetItem("Page 1").GetItem("Level").Value = IR.GetItem("Page
1").GetItem("Level").Value +4          //Changing an item property изменения
параметра итема (Incrementing the value by 4)
});
```

- EVENT\_TOUCH\_DOWN - finger pressing

```
IR.AddListener(IR.EVENT_TOUCH_DOWN, IR.GetItem("Page 1"), function()
//Event is activated when releasing the finger
    {
IR.GetItem("Page 1").GetItem("Level").Value = IR.GetItem("Page
1").GetItem("Level").Value -4          //Changing an item property
(Decrementing the value by 4)
});
```

- EVENT\_TOUCH\_UP - finger releasing

```
IR.AddListener(IR.EVENT_TOUCH_UP, IR.GetItem("Page 1"), function()
//Event is activated at finger pressing
    {
IR.GetItem("Page 1").GetItem("Level").Value = IR.GetItem("Page
1").GetItem("Level").Value +4          //Changing an item property (Incrementing
the value by 4)
});
```

- EVENT\_TOUCH\_MOVE - moving a finger

```

IR.AddListener(IR.EVENT_TOUCH_MOVE, IR.GetItem("Page 1"), function()
//Event is activated when moving the finger
{
IR.GetItem("Page 1").GetItem("Level").Value = IR.GetItem("Page
1").GetItem("Level").Value +4    //Changing an item property (Incrementing
the value by 4)
});

```

## Processing Events for Items

To process events use Listener:

**IR.AddListener(Event, Item, function(){Function\_Body})**

- **Event** - the event to be processed by Listener
- **Item** - the item identifier or the variable storing the item identifier
- **Function\_Body** - commands to be performed when activating the event

Example of processing events:

```

// Processing pressings
IR.AddListener(IR.EVENT_ITEM_PRESS, IR.GetItem("Color_Picker").GetItem("Item
20"), function()
{
    ColorPiker();
});

// Processing mouse movements
IR.AddListener(IR.EVENT_MOUSE_MOVE, IR.GetItem("Color_Picker").GetItem("Item
20"), function()
{
    ColorPiker()
});

// Processing finger movements
IR.AddListener(IR.EVENT_TOUCH_MOVE, IR.GetItem("Color_Picker").GetItem("Item
20"), function()
{
    ColorPiker()
});

```

## Launching Macros Bound to an Item

To refer to Macros use the command:

**IR.GetItem("Page\_Name").GetItem("Item\_Name").StartActions(Event\_Type)**

- **Page\_Name** - indicate to the command which page it should refer to
- **Item\_Name** - indicate to the command which item it should refer to
- **Event\_Type** - the event at which the Macros is activated (PRESS, RELEASE, SELECT, CHANGE,

MOUSE\_DOWN, MOUSE\_UP, MOUSE\_MOVE, TOUCH\_DOWN, TOUCH\_UP, TOUCH\_MOVE)

```
IR.AddListener(IR.EVENT_START,0,function() //Event is activated at the
application launch
{
IR.GetItem("Page 1").GetItem("Item 1").StartActions(IR.EVENT_ITEM_PRESS);
//Referring to the Macros
});
```

## Working with States

To refer to a state use the following command:

**IR.GetItem("Page\_Name").GetItem("Item\_Name").GetState(State\_Position).Properties;**

- **Page\_Name** - indicate to the command which page it should refer to
- **Item\_Name** - indicate to the command which item it should refer to
- **State\_Position** - indicate to the command which state it should refer to (counting from 0)
- **Properties** - indicate the property to be changed

```
IR.AddListener(IR.EVENT_START,0,function() //Event is activated at the
application launch
```

```
{
IR.CreateItem(IR.ITEM_BUTTON,"Button") //Creating an item
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(0).Opacity = 200; //Refer to
state 1 and change the opacity
```

```
IR.GetItem("Page 1").GetItem("Button").GetState(1).Opacity = 10; //Refer to
state 2 and change the opacity
```

```
IR.GetItem("Page 1").GetItem("Button").Feedback = 4 //indicate which way of
state switching to use
```

```
});
```

**[DOWNLOAD: Example of a project](#)**