

[DOWNLOAD: Example of a project](#)

Contents

- [1 IR.CreateItem](#)
- [2 IR.GetItem](#)
- [3 IR.DeleteItem](#)
- [4 ItemsCount](#)
- [5 GetState](#)
- [6 IR.PagesCount](#)
- [7 IR.PopupsCount](#)
- [8 IR.CurrentPage](#)
- [9 IR.GetPage](#)
- [10 IR.GetPopup](#)
- [11 IR.ShowPopup](#)
- [12 IR.ShowPage](#)
- [13 IR.HidePage](#)
- [14 IR.HidePopup](#)
- [15 IR.HideAllPopups](#)
- [16 IR.EVENT_ITEM_PRESS](#)
- [17 IR.EVENT_ITEM_RELEASE](#)
- [18 IR.EVENT_ITEM_CHANGE](#)
- [19 IR.EVENT_MOUSE_DOWN](#)
- [20 IR.EVENT_MOUSE_UP](#)
- [21 IR.EVENT_MOUSE_MOVE](#)
- [22 IR.EVENT_TOUCH_DOWN](#)
- [23 IR.EVENT_TOUCH_UP](#)
- [24 IR.EVENT_TOUCH_MOVE](#)
- [25 IR.EVENT_ITEM_SHOW](#)
- [26 IR.EVENT_ITEM_HIDE](#)
- [27 DOWNLOAD: Example of a project](#)

IR.CreateItem

This function is used for creating [items](#)

Main Syntax: 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
```

```
});
```

An item is created using the main syntax on the current active page. The current active page is defined with the help of [IR.CurrentPage](#). If you need to create an item on the page which is closed now or on a popup you need to use the following syntax:

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

- **Page_or_Popup_Name** - the name of any selected project page or popup

```
IR.AddListener(IR.EVENT_START,0, function() // Event is activated at the  
application launch  
{  
IR.GetItem("Popup 1").CreateItem(IR.ITEM_BUTTON,"Button",30,40,800,150);  
});
```

Items created with the help of **IR.CreateItem** have a particular font cour.ttf, size 20. You cannot change it.

IR.GetItem

This function is used for editing items

Syntax: IR.GetItem("Page_of_Item_Location").GetItem("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
//IR.GetItem("Page 1").GetItem("Button").GetState(0).Image =)
//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";

// Changing the text font
// To change a font on the item you should input the file name of the font
// The font file should be included in the project.
IR.GetItem("Page 1").GetItem("Button").GetState(0).Font = "Tahoma.ttf";

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

IR.GetItem("Page 1").GetItem("Button").Visible = false;// Show/Hide item
(true/false)

IR.GetItem("Page 1").GetItem("Button").Enable = false;// Active/not Active
item (true/false)

// Example to read the property TriggerValue1 and TriggerValue2 of Trigger
Button:
IR.Log(IR.GetItem("Page 1").GetItem("Item 1").TriggerValue1);
IR.Log(IR.GetItem("Page 1").GetItem("Item 1").TriggerValue2);
});

```

IR.DeleteItem

- *under development*

ItemsCount

This function is used for counting items on the page.

Syntax: **IR.CurrentPage.ItemsCount**

It 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 it
  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 it
  ShowCount.Text = IR.GetItem("Page 2").ItemsCount;
});
```

GetState

This function is used for referring to the item state

Syntax:

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
```

```
});
```

IR.PagesCount

This function is used for counting pages in the project.

Syntax: **IR.PagesCount**.

The command gives the number which can be saved in the variable:

```
IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at application launch
{
    var count = IR.PagesCount; //Command for counting the number of pages in the project
});
```

IR.PopupsCount

This function is used for counting popups in the project.

Syntax: **IR.PopupsCount**.

The command gives the number of popups. The received number can be saved, for example, in the variable:

```
IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at application launch
{
    var PopupsCount = IR.PopupsCount; //Command for counting popups in the project
});
```

IR.CurrentPage

This function is used for receiving access to the page opened at the current moment

Syntax: **IR.CurrentPage**;

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

```

application launch
{
    var current = IR.CurrentPage; //The command write the identifier of the
current page

// Reading and changing properties is available for IR.CurrentPage.

    // Example 1 – output the value Name in the log using direct reference to
the property.
    IR.Log(IR.CurrentPage.Name);

    // Example 2 – writing the link to the current page in the variable
// Property activation is made from the variable, this way should be used
// if you plan to use many properties of the current page.
// Using the link makes your work more efficient.
var Page = IR.CurrentPage;
IR.Log(Page.Name);

    // Example 3 – writing the property directly to the variable.
var Name = IR.CurrentPage.Name;
IR.Log(Name);

});

```

IR.GetPage

This function is used for referring to the page.

Syntax: **IR.GetPage(Page_Name_or_Identifier);**

```

IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at
application launch
{
    var MyPage = IR.GetPage("Page 1"); //Referring to the page by name
    var MyPage = IR.GetPage(0); //Referring to the page by identifier
});

```

IR.GetPopup

This function is used for referring to the popup.

Syntax: **IR.GetPopup(Popup_Name_or_Identifier);**

```

IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at
application launch
{
    var MyPopup = IR.GetPopup("Popup 1"); //Referring to the popup by name
    var MyPopup = IR.GetPopup(0); //Referring to the popup by identifier
});

```

IR.ShowPopup

This function is used for opening the popup.

Syntax: **IR.ShowPopup(Popup_Name_or_Identifier);**

```
IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at
application launch
{
    IR.ShowPopup("Popup 1"); //Opening a popup by its name
});
```

IR.ShowPage

This function is used for opening the page.

Syntax: **IR.ShowPage(Page_Name_or_Identifier);**

```
IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at
application launch
{
    IR.ShowPage("Page 1"); //Opening a page by its name
});
```

IR.HidePage

This function is used for closing the page.

Syntax: **IR.HidePage(Page_Name_or_Identifier);**

```
IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at
application launch
{
    IR.HidePage("Page 1"); //Closing a page by its name
});
```

IR.HidePopup

This function is used for closing the popup.

Syntax: **IR.HidePopup(Popup_Name_or_Identifier);**

```
IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at
application launch
{
    IR.HidePopup("Popup 1"); //Closing a popup by its name
});
```

IR.HideAllPopups

This function is used to close the popups on a specific page or all pages.

Syntax: **IR.HideAllPopups(Page_Name_or_Identifier);**

If **Page_Name_or_Identifier** is not specified, popups are hidden on all pages.

```
IR.AddListener(IR.EVENT_START, 0, function() //Event is activated at
application launch
{
    IR.HideAllPopups("Page 1"); //Closing a popups by Page 1
});
```

IR.EVENT_ITEM_PRESS

Event is activated when pressing on the item.

Syntax: **IR.AddListener(IR.EVENT_ITEM_PRESS,Referring_to_Item,function())**

```
IR.AddListener(IR.EVENT_ITEM_PRESS,IR.GetItem("Page 1").GetItem("Item
1"),function()
//Event is activated when pressing on Item 1 on Page 1
{
    IR.HidePopup("Popup 1"); //Closing a popup by its name
});
```

IR.EVENT_ITEM_RELEASE

Event is activated when releasing the item.

Syntax: **IR.AddListener(IR.EVENT_ITEM_RELEASE,Referring_to_Item,function())**

```
IR.AddListener(IR.EVENT_ITEM_RELEASE,IR.GetItem("Page 1").GetItem("Item
1"),function()
//Event is activated when releasing Item 1 on Page 1
{
    IR.HidePopup("Popup 1"); //Closing a popup by its name
});
```

IR.EVENT_ITEM_CHANGE

Event is activated when changing the EditText item

Syntax: **IR.AddListener(IR.EVENT_ITEM_CHANGE,Referring_to_Item,function(text)**

```
IR.AddListener(IR.EVENT_ITEM_CHANGE,IR.GetItem("Page 1").GetItem("Item
1"),function()
```



```
//Event is activated when changing the EditBox Item 1
//on Page 1
{
  //Outputting the text from the EditBox item in the console
  IR.Log(IR.GetItem("Page 1").GetItem("Item 1").text);
});
```

IR.EVENT_MOUSE_DOWN

Event is activated when holding the mouse button down (for Windows only)

Syntax: **IR.AddListener(IR.EVENT_MOUSE_DOWN,Referring_to_Item,function())**

```
IR.AddListener(IR.EVENT_MOUSE_DOWN,IR.GetItem("Page 1").GetItem("Item
1"),function()
//Event is activated when holding the mouse button down on Item 1
//on Page 1
{
  IR.Log("The mouse is down");//Outputting the message in the console
});
```

IR.EVENT_MOUSE_UP

Event is activated when releasing the mouse button (for Windows only)

Syntax: **IR.AddListener(IR.EVENT_MOUSE_UP,Referring_to_Item,function())**

```
IR.AddListener(IR.EVENT_MOUSE_UP,IR.GetItem("Page 1").GetItem("Item
1"),function()
//Event is activated when releasing the mouse button on Item 1
//on Page 1
{
  IR.Log("The mouse is up");//Outputting the message in the console
});
```

IR.EVENT_MOUSE_MOVE

Event is activated when holding the mouse button down on the item and moving the mouse (for Windows only)

Syntax: **IR.AddListener(IR.EVENT_MOUSE_MOVE,Referring_to_Item,function())**

```
IR.AddListener(IR.EVENT_MOUSE_MOVE,IR.GetItem("Page 1").GetItem("Item
1"),function()
//Event is activated when holding the mouse button down on the item and
moving the mouse
{
  IR.Log("The mouse moving");//Outputting the message in the console
```

```
});
```

IR.EVENT_TOUCH_DOWN

Event is activated when holding the finger down on the item (for iOS and Android only)

Syntax: **IR.AddListener(IR.EVENT_TOUCH_DOWN,Referring_to_Item,function()**

```
IR.AddListener(IR.EVENT_TOUCH_DOWN,IR.GetItem("Page 1").GetItem("Item 1"),function()  
//Event is activated when holding the finger down on Item 1  
//on Page 1  
{  
  IR.Log("touch down");//Outputting the message in the console  
});
```

IR.EVENT_TOUCH_UP

Event is activated when releasing the finger from the item (for iOS and Android only)

Syntax: **IR.AddListener(IR.EVENT_TOUCH_UP,Referring_to_Item,function()**

```
IR.AddListener(IR.EVENT_TOUCH_UP,IR.GetItem("Page 1").GetItem("Item 1"),function()  
//Event is activated when releasing the finger from Item 1  
//on Page 1  
{  
  IR.Log("touch up");//Outputting the message in the console  
});
```

IR.EVENT_TOUCH_MOVE

Event is activated when moving the finger on the item (for iOS and Android only)

Syntax: **IR.AddListener(IR.EVENT_TOUCH_MOVE,Referring_to_Item,function()**

```
IR.AddListener(IR.EVENT_MOUSE_MOVE,IR.GetItem("Page 1").GetItem("Item 1"),function()  
//Event is activated when moving the finger on the item  
{  
  IR.Log("touch move");//Outputting the message in the console  
});
```

IR.EVENT_ITEM_SHOW

Event is activated when showing a page or popup.

Syntax: **IR.AddListener(IR.EVENT_ITEM_SHOW,Referring_to_Page_or_Popup,function()**

```
IR.AddListener(IR.EVENT_ITEM_SHOW,IR.GetItem("Popup 1"),function()  
// Event is activated when showing a page or popup.  
{  
    IR.Log("Popup 1 Show");//Outputting the message in the console  
});
```

IR.EVENT_ITEM_HIDE

Event is activated when hiding a page or popup.

Syntax: **IR.AddListener(IR.EVENT_ITEM_HIDE, Referring_to_Page_or_Popup,function()**

```
IR.AddListener(IR.EVENT_ITEM_HIDE,IR.GetItem("Popup 1"),function()  
// Event is activated when hiding a page or popup.  
{  
    IR.Log("Popup 1 Hide");//Outputting the message in the console  
});
```

[DOWNLOAD: Example of a project](#)