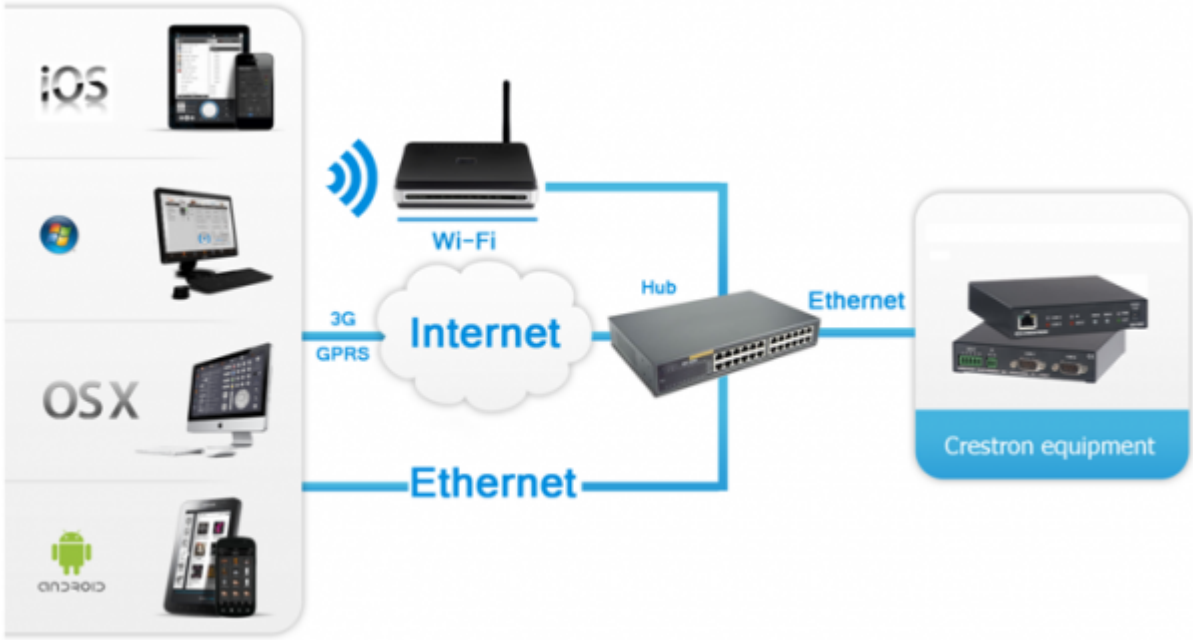
# Contents



# How It Works

The topology of communication between iRidium Apps and Crestron equipment:



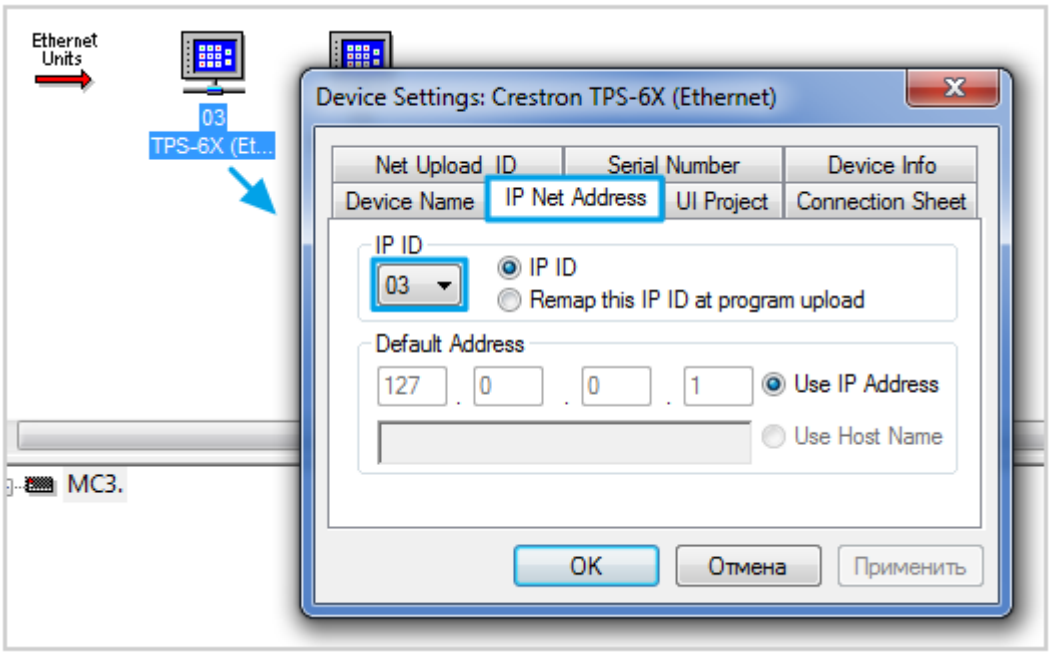

iRidium connects to equipment directly, without any additional hardware. iRidium communicates with Crestron joins. You can receive the list of joins from your *.smw (SIMPL™) project.

# Import of Joins from SIMPL Windows Projects

iRidium for Crestron enables you to create control interfaces for communication with Crestron controllers using Crestron SIMPL™ Windows® Software projects as sources of data about the joins used.

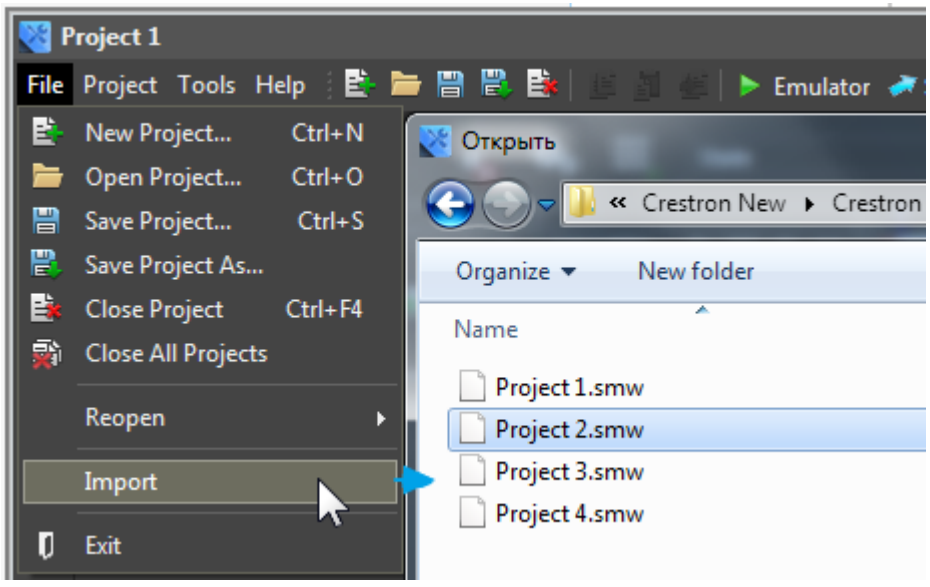
> *When creating a project in SIMPL Windows do not forget to add a panel in the project and*

*set its identifier IP ID – iRidium control devices will use this IP ID for connection to the Crestron controller:*

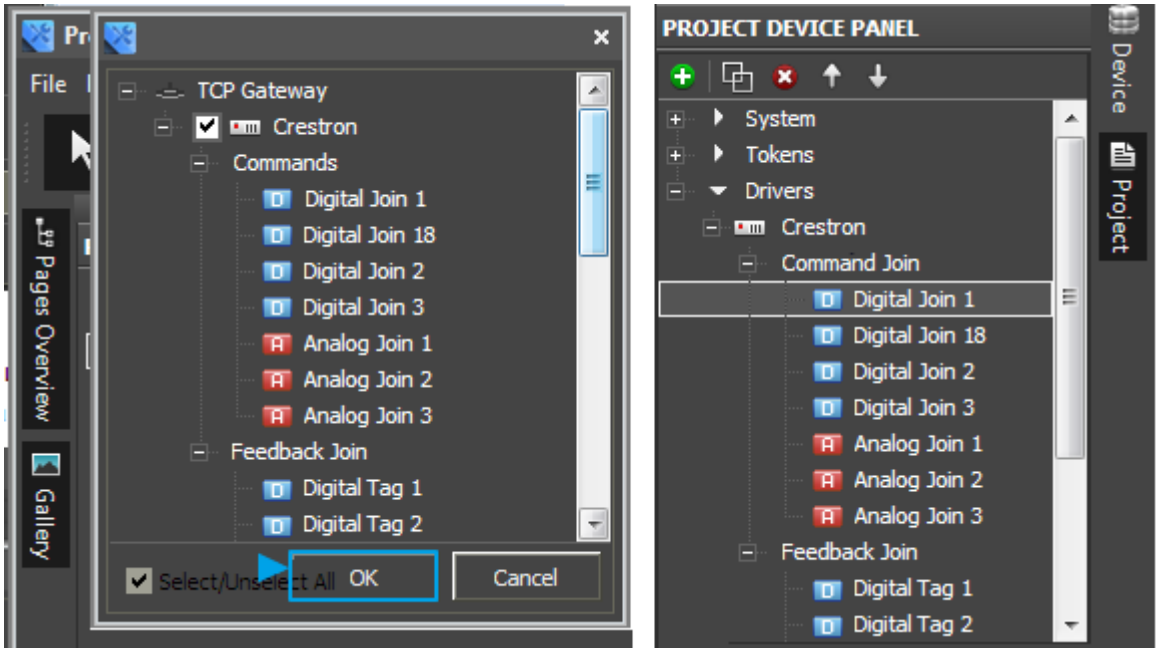



Save the SIMPL Windows (***.smw**) project so it is ready for import to iRidium GUI Editor. All joins are going to be imported from the project during the importing process.

To import joins from the *.smw file create a new project in GUI Editor and select “Import” in the “File” menu:

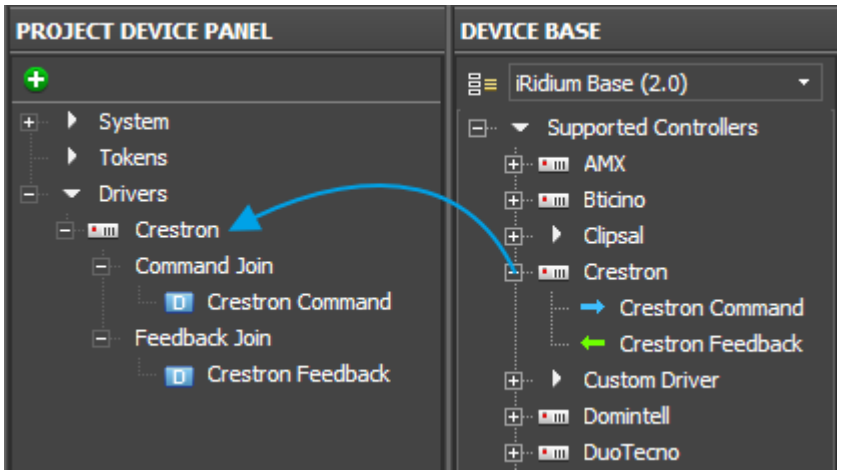



The import tool forms the controller template and the list of joins used by the controller. As a result you receive the tree with project devices and joins:




Indicate the properties of connection to the Crestron controller for the device. Commands to joins and channels of joins status should be bound to graphic items (Drag&Drop).

When there is no data for importing you can add a Crestron device in your project from iRidium Device Base manually (Drag&Drop):

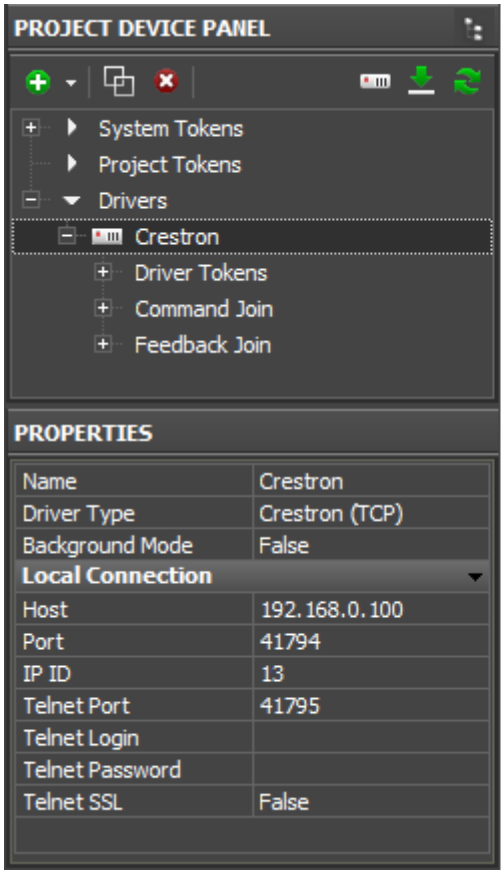


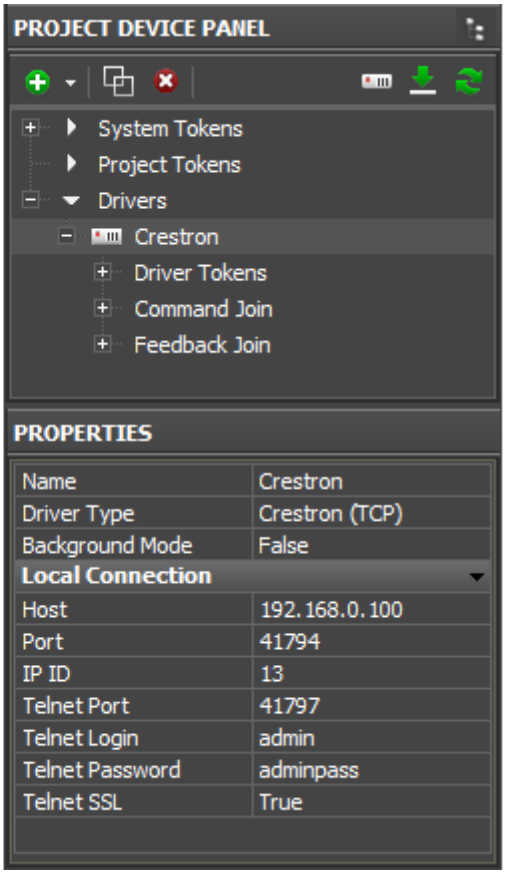

After the import or manual creation of the Crestron device you can proceed with setting up the connection to the controller and working with joins.

↑ Back

## Setting up Connection to Crestron

When connecting to the Crestron controller you can activate or deactivate SSL support. The SLL working mode affects the choice of the Telnet port for connection to the controller:




### SSL is deactivated:

**Host** – the controller IP-address

**Port: 41794** – the port for Crestron control

**IP ID** – the panel identifier in the project SIMPL (put is as a decimal number, 0D => 13)

**Telnet Port: 41795** – the Telnet port for deactivated SSL

**Telnet Login** – leave empty

**Telnet Password** - leave empty

**SSL: False** - SSL is deactivated

### SSL is activated:

**Host** - the controller IP-address

**Port: 41794** - the port for Crestron control

**IP ID** - the panel identifier in the project SIMPL (put is as a decimal number, 0D => 13)

**Telnet Port: 41797** - the Telnet port for activated SSL

**Telnet Login** – the SSL login

**Telnet Password** - the SSL password

**SSL: True** - SSL is activated

For working with the controller via the Internet indicate the public IP-address of the router the controller is connected to in the connection settings. Port Forwarding Service is set up for the router to enable remote control – referring to the local (private) address of the controlled controller from the Internet.

You can learn the external (public) IP-address of your router with the help of external resources, for example [1]

↑ Back

#### Switching Between the Local and Internet Connection

If the control panel should hold the connection with the system when the panel is out of the limits of the Wi-Fi network of the system, you need to set up the switch between the Internet and the local network.

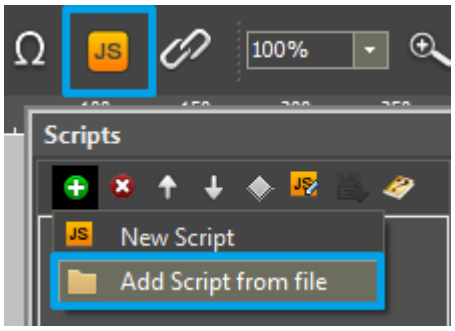
The remote mode suggests connection to the system via the Internet. At that the external IP-address or the domain name of the system to which you need to connect have to be used.

⚠ **In iRidium Wi-Fi/3G CANNOT be switched automatically.** For switching between the Internet and the local network you need buttons with special settings. See the settings below.

⚠ To control the system remotely you have to open the system for external access - to set up the Port Forwarding Service.

⚠ To secure equipment from unauthorized access we recommend using secure connection with the remote system (VPN).

**Setting up of the switch Wi-Fi/3G in iRidium projects:**




**1.** Open the script editor in iRidium GUI Editor.

**2.** Download and add into your project the template of the Wi-Fi/3G switch (Add Script from file): **download the template of the Wi-Fi/3G switch**

The Wi-Fi/3G switch is performed with the help of the script function SetParameters

*Setting up of parameters of the Wi-Fi/3G switch:*

```
function Internal_1() // Function name
{
    IR.GetDevice('Crestron').SetParameters({Host: '192.168.0.101', Port: '41794', NetID: '3', TelnetPort: '41795', TelnetLogin: '', TelnetPassword: '', TelnetSSL: 0}); // Driver Name + Parameters
}
```

```
function External_1()

{

    IR.GetDevice('Crestron').SetParameters({Host: '192.168.0.101', Port: '41794', NetID: '3',
    TelnetPort: '41797', TelnetLogin: 'admin', TelnetPassword: 'admin', TelnetSSL: 1});

}
```

*Indicate in the command settings:*

- Function name – the name of the switch function (command). Two functions cannot have the same name in a project.
- Driver Name – the name of the driver which parameters are changed
- Parameters – the set of the switch parameters which you need to apply to the driver

*Assign commands to buttons:*

1. Select the button which will be responsible for the Wi-Fi/3G switch.
   Open the properties of the button: Object Properties > Programming
2. Open Macros Editor of the button for the Press or Release events
3. Select the **Script Call** command and add it by double-clicking on it
4. Select the name of the function you want to activate in the drop-down list. Create the command.


*Set up access to the equipment from the Internet:*

In order to do that open the equipment ports for remote access.

Download the example of the Wi-Fi/3G switch (project) >>

↑ Back

# Principles of iRidium Operation with the Crestron Controller

When working with joins of the Crestron controller iRidium operates with the following definitions:

***Command Join (command)*** - path for sending data to a join. In the command settings it is

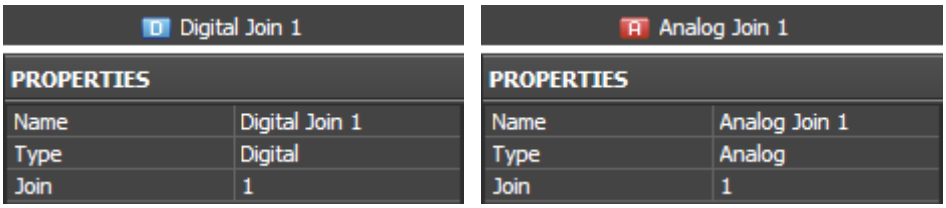
required to indicate the address of the join you want to send data to from iRidium.

***Feedback Join (status channel)*** - a path for receiving data from a join. In the channel settings it is required to indicate the address of the join you want to receive data from.

***Token (global variable)*** - a cell for storing data in iRidium which does not have direct relation to the Crestron controller. You can write any data from a Crestron channel or any graphic item in a token. The token stores the data and they will not be deleted at the App relaunch.

## Properties of Command Joins and Feedback Joins for Crestron Equipment

Command Joins in the iRidium GUI Editor device tree:



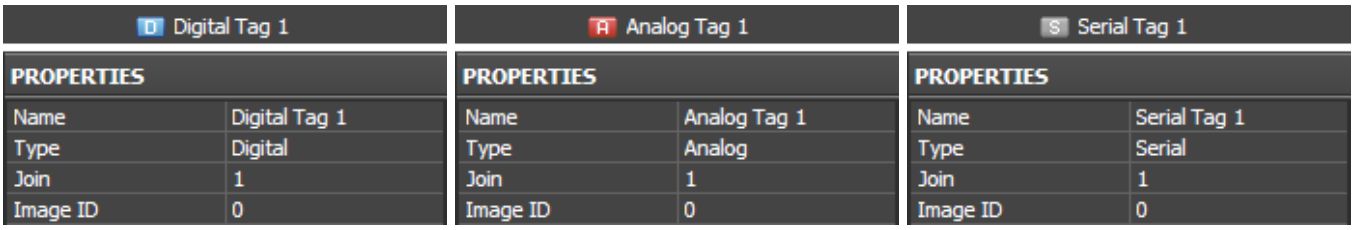

***Command Join Properties:***

***Name*** a join name, set at random

***Type*** a join data type (Digital, Analog)

***Join*** a join number (the address of the variable which value is sent)

Feedback Joins in the iRidium GUI Editor device tree:




***Feedback Join Properties:***

***Name*** a join name, set at random

***Type*** a join data type (Digital, Analog, Serial)

***Join*** a join number (the address of the variable which value is sent)

***Image ID*** an identifier of the dynamic image assigned to the join

## Crestron Join Types

There are the following types of Crestron joins (the *Type* property in the command or channel settings), defined by their control range and application:

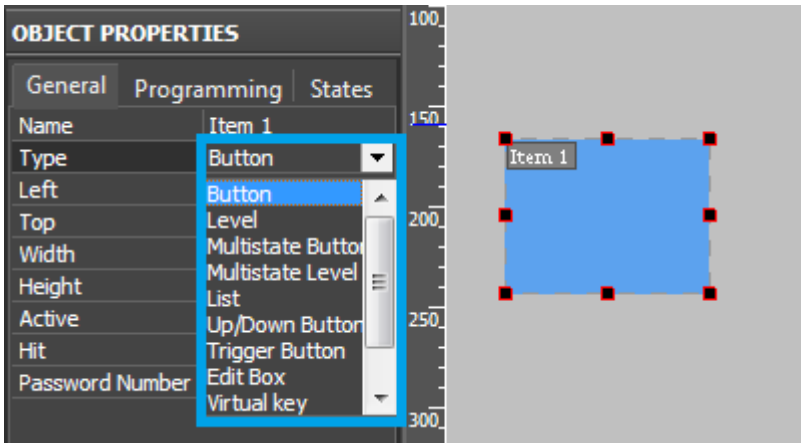
***Crestron joins types:***

***Digital*** a bit variable taking values 0 or 1. It can be used in commands and channels of join status.

| | |
|---|---|
| ***Analog*** | an information 16-bit unsigned variable taking values in the range from 0 to 65535. It is used in commands and channels of join status. |
| ***Serial*** | a variable storing data strings in the ASCII format. It can be used when creating status channels to receive data strings from the controller. |

## Types of Graphic Items for Controlling Joins

You can manipulate the Crestron controller which works with different data types (bit and information values, data strings) with the help of the following types of GUI graphic items:




| | |
|---|---|
| ***Button*** | Sending fixed values; displaying data received from the controller |
| ***Level*** | Sending values from the preset range by the slider; displaying the current value by the position of the slider |
| ***Trigger Button*** | Switching between two fixed values indicated at the item setting |
| ***Up/Down Button*** | Incrementing/decrementing the current value by the preset value in the preset range. The range and the step of increment/decrement are set at the item setting. |
| ***Multistate Button*** | Sending fixed values and receiving data accompanied by animation |
| ***Multistate Level*** | Sending and receiving values in the preset range where each value or group of values has its own image |
| ***Edit Box*** | Inputting a string of data to be sent to the bus |
| ***Joystick*** | Controlling RGB with the help of ColorPicker |

There are other types of graphic items which are not used for sending values to the controller directly but they serve as visualization objects. Types of graphic items are set in the *General* tab of the *Object Properties* window. Here you can indicate characteristic properties for different types of graphic items.

## Principles of Sending Commands to Crestron Joins

To create and use *Command Join* – a command for sending data to joins it is required to perform the following actions:

1. Create a command (a path for data sending) in the tree; indicate its name, data type and address of the controlled join.

2. Create a graphic item which is going to activate the command sending (properties of the graphic item are defined by the type of data used by the join and control tasks).

The choice of the item type depends on the required way of controlling the variable. The choice of the feedback type *(Feedback)* of the item affects the displaying of data received from the controller. The main *Feedback* types used (indicated in the *Programming* tab of the *Object Properties* window):

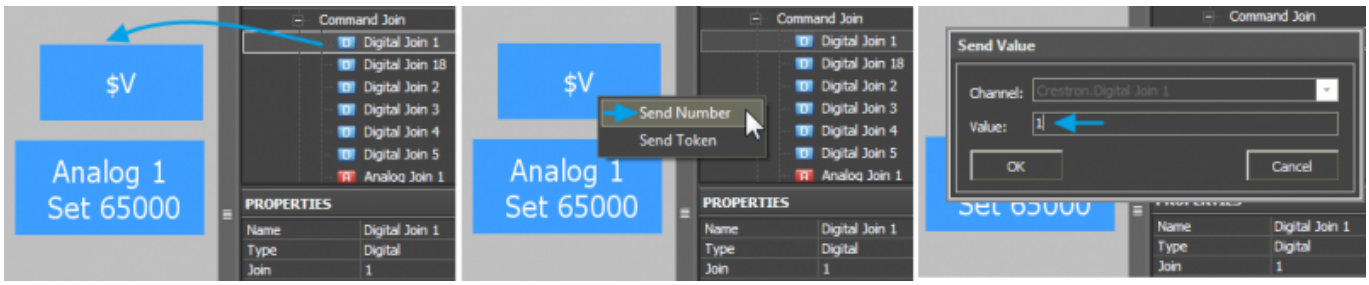
- **Momentary** – not to display data received from the channel but change the item state at pressing on it (returning to the initial state at releasing it)
- **Channel** – to display data received from the channel. The data can be input in the item text field, they can affect its state (switch its states depending on received values) or selected item properties.

The rest feedback types have specific purposes and are used less frequently.

3. Drag the command onto the graphic item (Drag&Drop).
When dragging the command you will see the window where you are required to select data which is going to be sent to the controller:

- **Send Number** – send numbers in the decimal format (1 –bit Digital or 16 –bit Analog)
- **Send Token** – send values taken by one of the graphic item properties or a global token to the bus (for example you can select sending the current slider position of the Level graphic item)

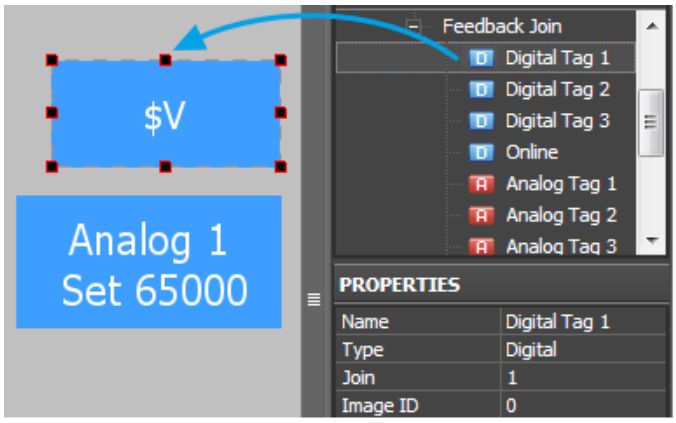
If the selected data type is *Send Number* indicate the value to be sent to the join when activating the graphic item.
If the selected data type is *Send Token* select the property which current value is to be sent to the bus when activating the item.

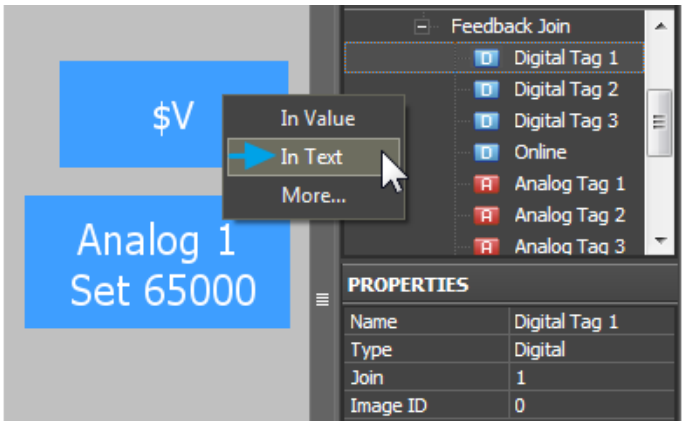


### Principles of Reading the Status of Crestron Joins

To create and use Feedback Join – a channel for receiving data from joins it is required to perform the following actions:

1. Create a status channel *(Feedback Join)* for receiving data from the join.
2. Drag the feedback channel onto the graphic item (Drag&Drop)
   - **Channel** – it is required to indicate this feedback type *(Feedback)* in the item properties if data of the channel bound to the item influence the behavior of the item (for example, if data from the bound channel has to be output in the text field; or when receiving "1" from the channel the item has to take the opposite (active) state).
3. Indicate if the value should be output in the item text field, affect its current value or other property:
   - **In Text** – the value received from the channel should be output as a number in the text field.
   - **In Value** – the value received from the channel should affect the item state (change the slider position of Level, switch Button on/off, initiate animation, etc.)
   - **More...** – select more complex way of communication between the channel and item properties (for example, change item coordinates when the join value is changed)




**Examples of Communication with Joins of Crestron Equipment see here.**

↑ Back

# Emulation of Project Work

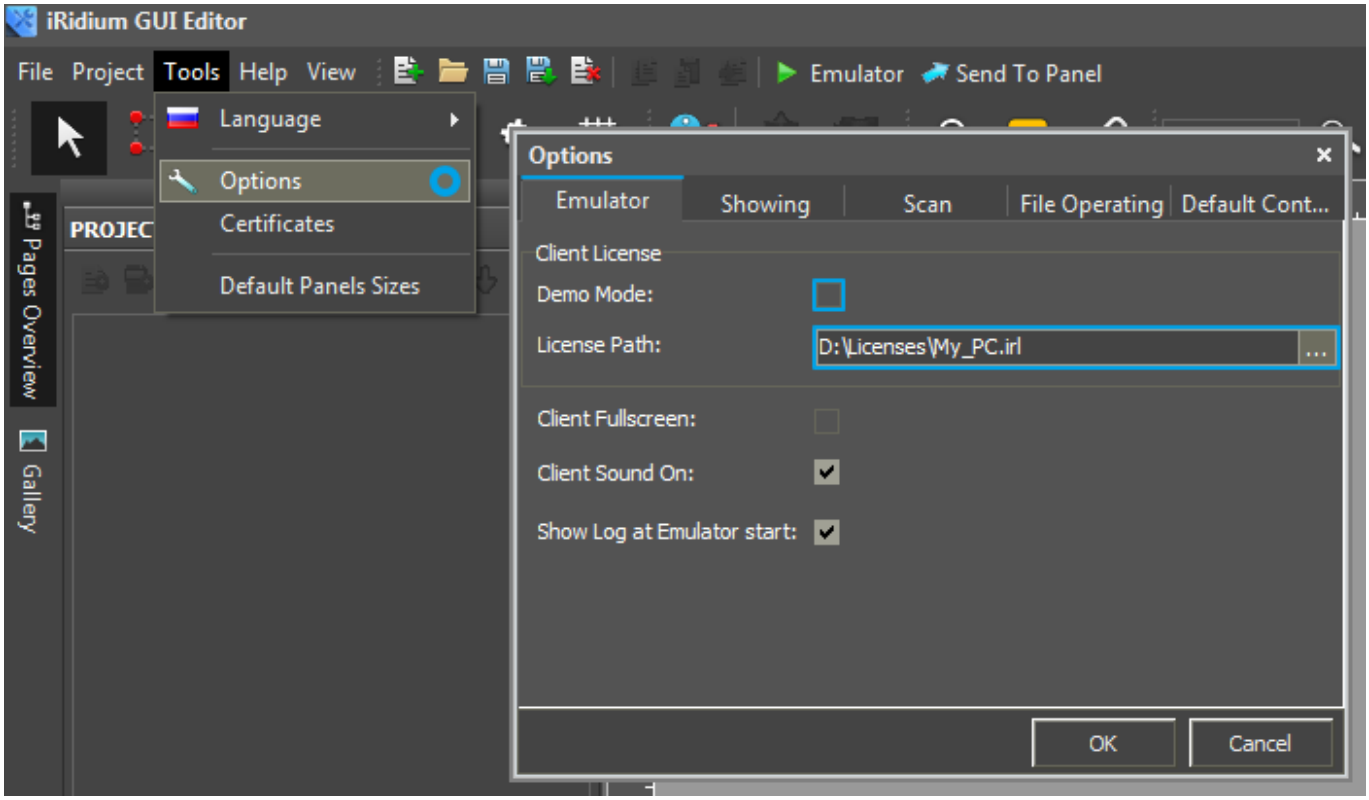
#### Emulator

- is an iRidium application for Windows which can be launched from GUI Editor for testing your projects. Emulator can work both with a license (with connection to the equipment) and without it (when only the project graphic part is functional).

**Operation modes for Emulator** (see GUI Editor > Tools > Options > Emulator):

> **Without the license** (Demo Mode: on) – no connection with the controlled equipment.
> **With the license** (License Path: [...]) – when the license file is selected, all Emulator functions work and there is connection with the controlled equipment.

**Indicate the path to the license file for Emulator** (GUI Editor > Tools > Options > Emulator):



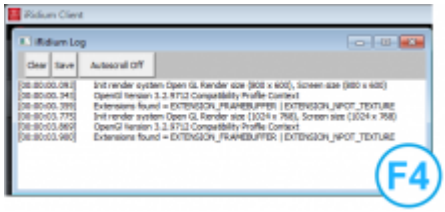

* *When **Demo Mode** is activated Gui Editor won't show the message about starting without the connection to the controlled equipment (without a license). Check if there is a license when setting up your project.*

**Emulator hot keys**

Click **F5** to start Emulator.
Click **F8** to open the Emulator settings (password: **2007**)
Click **F4** to open the Emulator log.

| | |
|---|---|
| ⚠ | For **fully functional work** of iRidium on your PC it is required to get an iRidium license and activate it for for your PC.<br>Activation of licenses for iPad/iPhone/Mac/Android based or other devices does NOT lead to the automatic licensing of the PC with the installed iRidium Environment. A license for your PC is required for iRidium client on your PC (including Emulator) to work in the fully functional mode. The license should be purchased separately or you can use free For testing purposes you can use free licenses). |

## Logging in Windows



**iRidium Log** is a window where information about iRidium work, error messages and iRidium Script logs (IR.Log) are output in the text format.

To open the iRidium log in Windows, click **F4**.

↑ Back

## Launching Projects on Control Panels

Uploading and launching of iRidium projects on control panels are performed with the help of the iRidium Transfer application installed on your PC. You can also upload your project on the panel from GUI Editor with the help of Transfer.

- Instructions for setting up properties of the iRidium project launch on control panels: **Properties for Launching iRidium Projects**
- Instructions for uploading iRidium projects on control panels: **iRidium Transfer**.
- Instructions for setting up iRidium projects on control panel: **iRidium App**

↑ Back