iRidium driver

is a program which allows your project to communicate with the equipment of your automation system.
Different kinds of equipment use their own protocols. That is why iRidium has a number of ready drivers for controlling [the most popular](#) automation systems. It also has the universal [AV & Custom Systems](#) driver on which basis it is possible to create any driver non-supported by iRidium by default. There are native (built-in, standard) and script drivers in iRidium. The script drivers are created on the basis of [AV & Custom Systems](#) and use [iRidium Script](#) for work.:
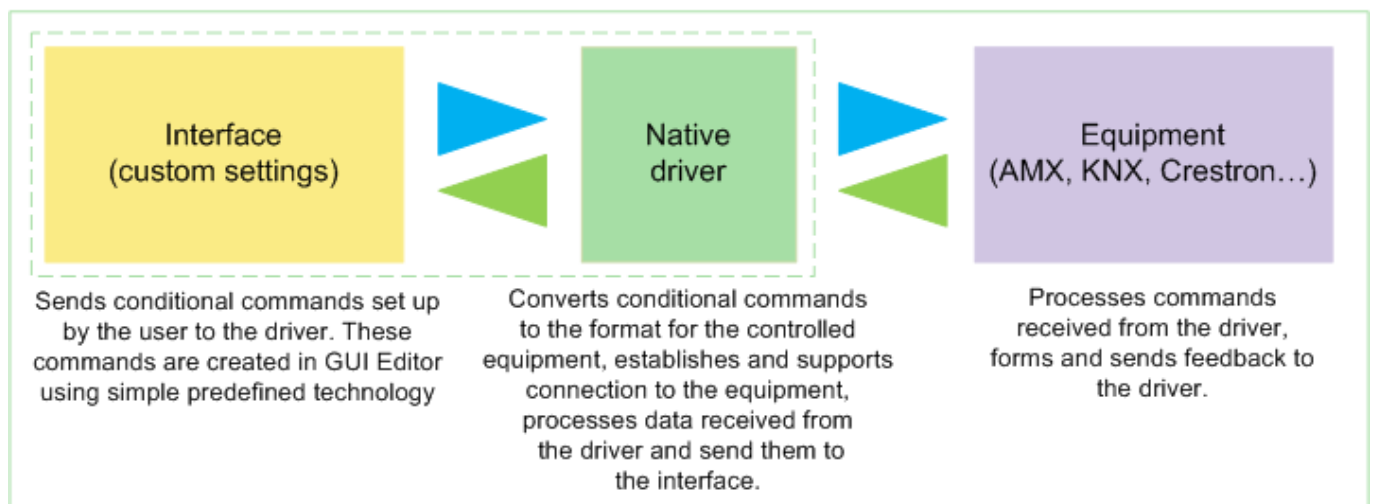
`C++` Native Driver

it is a driver written in C++ as the program of communication with a particular kind of equipment. Native drivers have deep integration with iRidium development environment and that enables their intuitive setting up and fast work on control devices.

Convenience of using a native driver is achieved due to the fact that it turns on all possible instructions for communication with the corresponding equipment, i.e. the syntax for messages is predefined. Native drivers have a set of instructions for sending commands and processing feedback. All you have to do to use them is to add the driver you need in your project and create commands and feedback channels (their settings are standard and exclude the possibility of sending wrong values).

Native drivers are fully described in GUI Editor. A native driver can be presented as a black box for which the user forms simple instructions using standard tools of GUI Editor. On the basis of the instructions the driver forms control communication with the controller, creates and supports the connection. The driver receives messages from the equipment which are output as simple numbers or strings in feedback channels after conversion..

Operation of iRidium Native Driver with Equipment (besides AV & Custom Systems):

Interface (custom settings)	Native driver	Equipment (AMX, KNX, Crestron…)
Sends conditional commands set up by the user to the driver. These commands are created in GUI Editor using simple predefined technology	Converts conditional commands to the format for the controlled equipment, establishes and supports connection to the equipment, processes data received from the driver and send them to the interface.	Processes commands received from the driver, forms and sends feedback to the driver.

There is a special driver - **AV & Custom Systems (TCP, UDP, HTTP, RS232)** in the base of native drivers. Unlike all the rest drivers it doesn't have a part responsible for processing feedback as the driver was created for working with ANY equipment using one of the available transport protocols (TCP, UDP, HTTP, RS232) and not for a particular device (where feedback type is predefined).

The AV & Custom Systems driver enables forming of ANY command for ANY equipment, writing it as a string or sequence in GUI Editor and sending it to the selected device. The native AV & Custom Systems driver supports connection to the equipment and sends commands but it does not include a ready component for processing feedback.

Operation of iRidium Native Driver for AV & Custom Systems
(provides only one-way communication with any equipment):
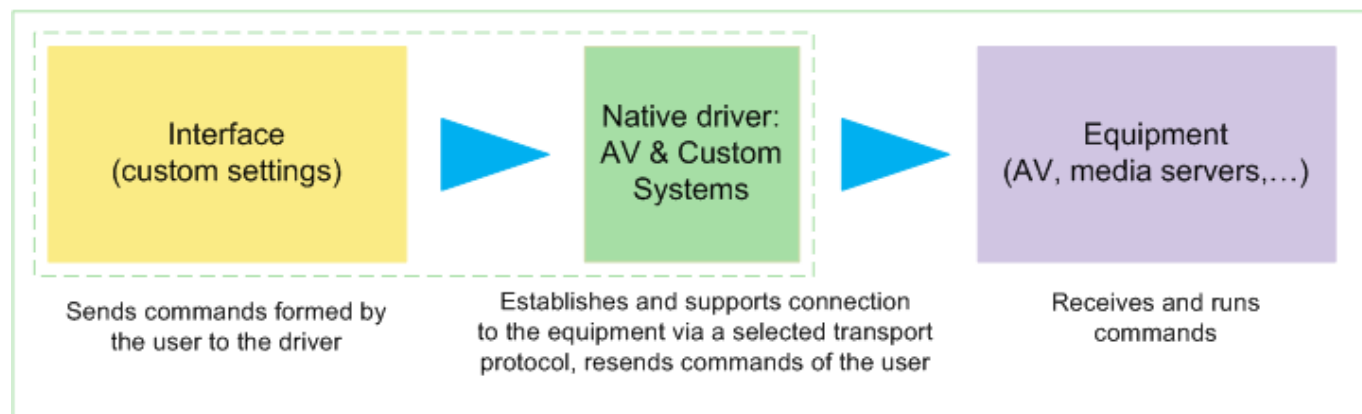
A disadvantage of using the AV & Custom Systems (TCP, UDP, HTTP, RS232) driver by itself is that you can create and send a command to the equipment but you cannot receive feedback from it as such data might vary for different equipment.

To receive feedback from the equipment there is a special iRidium script system which serves as an add-in for the native AV & Custom Systems driver. It allows you to receive, process and send to the interface data received from any equipment.
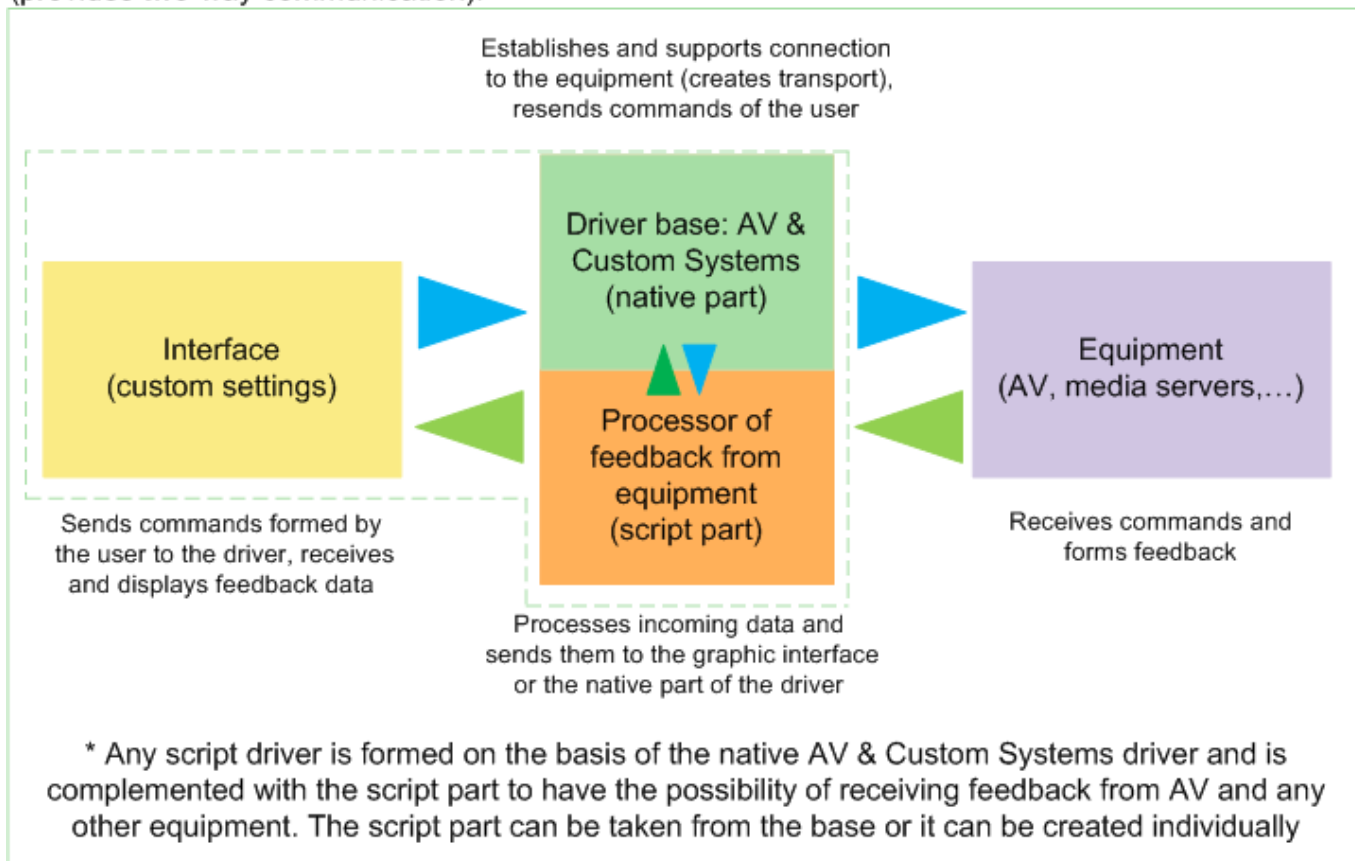
The complex of the native part on the basis of the AV & Custom Systems driver and the script part which processes data received from the equipment is called *Script driver*:

JS Script driver

it is a driver created on the basis of the native AV & Custom Systems driver with the help of iRidium DDK. Scripts enable receiving data and communication with ANY equipment. Driver scripts can be created by the user for operation with any control protocol.

Thus a *script driver* consists of the native partc (the native AV & Custom Systems driver) and the script part – a program written in iRidium Script which is responsible for receiving data from the equipment, their processing and sending to the interface. Also the script part can communicate with the native driver, form , process and send commands of the native driver:

Operation of iRidium Script Driver on the Basis of AV & Custom Systems with Equipment:
(provides two-way communication):

Establishes and supports connection
to the equipment (creates transport),
resends commands of the user

Driver base: AV &
Custom Systems
(native part)

Interface
(custom settings)

Equipment
(AV, media servers,…)

Processor of
feedback from
equipment
(script part)

Sends commands formed by
the user to the driver, receives
and displays feedback data

Receives commands and
forms feedback

Processes incoming data and
sends them to the graphic interface
or the native part of the driver

* Any script driver is formed on the basis of the native AV & Custom Systems driver and is complemented with the script part to have the possibility of receiving feedback from AV and any other equipment. The script part can be taken from the base or it can be created individually

Due to the *native part* a script driver forms the session of connection to the equipment, provides support of the connection and sends commands. Due to the *script part* the driver receives and processes data from the equipment and writes them in driver channels for outputting them on the items of the graphic interface.

Script drivers can be written by the user with the help of iRidium Driver Development Kit – instruction for writing script drivers (with examples). You can also use **the base of READY iRidium Script drivers** which is constantly enriched with new devices.