system.device.addDevice

This function is used in **Python Scripting.**

Description

Adds a new device connection in Ignition. Accepts a dictionary of parameters to configure the connection. Acceptable parameters differ by device type: i.e., a Modbus/TCP connection requires a hostname and port, but a simulator doesn't require any parameters.

> When using this function, the arguments **MUST** be passed in as keyword arguments.

Client Permission Restrictions

Permission Type: Device Management

Client access to this scripting function is blocked to users that do not meet the role/zone requirements for the above permission type. This function is unaffected when run in the Gateway scope.

Syntax - Using deviceType

system.device. addDevice(deviceType, deviceName, deviceProps)

- Parameters

  String deviceType - The device driver type. Possible values are listed in the Device Types table below.

  String deviceName - The name that will be given to the the new device connection.

  PyDictionary deviceProps - A dictionary of device connection properties and values. Each deviceType has different properties, but most require at least a hostname. Keys in the dictionary are **case-insensitive**, spaces are omitted, and the names of the properties that appear when manually creating a device connection.

- Returns

  nothing

- Scope

  All

Device Types

> Note that this function may be called to add devices using 3rd party drivers: you simply need the driver type, which the module developer will be able to provide.

Driver Name	Device Type
Legacy Allen-Bradley CompactLogix	CompactLogix
Legacy Allen-Bradley ControlLogix	ControlLogix
Simulators Dairy Demo Simulator	DairyDemoSimulator
DNP3 Driver	Dnp3Driver
Allen-Bradley Logix Driver	LogixDriver
Allen-Bradley MicroLogix	MicroLogix
Modbus RTU	ModbusRtuOverTcp
Modbus TCP	ModbusTcp
Omron NJ Driver	com.inductiveautomation.omron.NjDriver
Allen-Bradley PLC5	PLC5

Driver Name	Device Type
Siemens S7-300	S7300
Siemens S7-400	S7400
Siemens S7-1200	S71200
Siemens S7-1500	S71500
Allen-Bradley SLC	SLC
Simulators SLC Simulator	SLCSimulator
Simulators Generic Simulator	Simulator
TCP Driver	TCPDriver
UDP Driver	UDPDriver

Device Properties

The `deviceProps` parameter is where you supply configuration values to the new connection. Value properties depend on which `deviceType` was specified. A listing of deviceProps keys can be found on the system.device.addDevice - deviceProps Listing page.

The keys in the `deviceProps` parameter are **case-insensitive**. Device properties not specified in the `deviceProps` paramet er will fallback to default values if not specified (where applicable: i.e., "hostname" typically does not have a default value).

Code Examples

Code Snippet

```
# Below is an example of creating a new Generic Simulator device connection.
# Note that we MUST pass a dictionary as the 3rd parameter, even if it's empty.

# Call the function
system.device.addDevice(deviceType = "Simulator", deviceName = "New_Generic_Simulator",
deviceProps = {} )
```

Code Snippet

```
# Add a device using the Allen-Bradley Logix Driver for firmware v21+ devices
deviceProps = {}
deviceProps["Hostname"] = "192.168.1.2"
system.device.addDevice(deviceName="Test1", deviceType="LogixDriver",
deviceProps=deviceProps)
```

Code Snippet

```
# Below is an example of creating a new S7-1500 device connection.

# Build a Dictionary of parameters
newProps = {
   "HostName" : "10.0.0.1",
   "Port" : 102 # <---If adding additional parameters, make sure to add a comma.
   }

# Call the function
system.device.addDevice(deviceType = "S71500", \
      deviceName = "My_S7_1500_Device",\
      deviceProps = newProps )
```