

# Configuring data endpoints

To use WCC Lite as a programmable logic controller, it needs to be configured in a particular way. The PLC functionality of the WCC Lite only allows for the use of data which has been configured in the Excel configuration spreadsheet. This has been done for security purposes and to preserve transmission medium only for data that is available. Only topics defined in the configuration can post or get data. If a certain data entry exists but it has not been linked to a PLC program, all calls from PLC runtime application to Redis database will be ignored. Therefore it is highly advised to prepare and upload the Excel configuration before using this signal in the PLC application.

Some parameters are mandatory for PLC usage. These parameters are shown in two tables below (one for Devices, one for Signals tab). Please note that other parameters can be used as well, but are not covered because they aren't specific to PLC functionality.

Table. Mandatory parameters for Devices tab

Parameter	Type	Description
name	string	User-friendly device name
device_alias	string	Device alias to used in configuration
enable	boolean	Enabling/disabling of a device
protocol	string	Selection of protocol (IEC 61499)

Table. Mandatory parameters for Signals tab

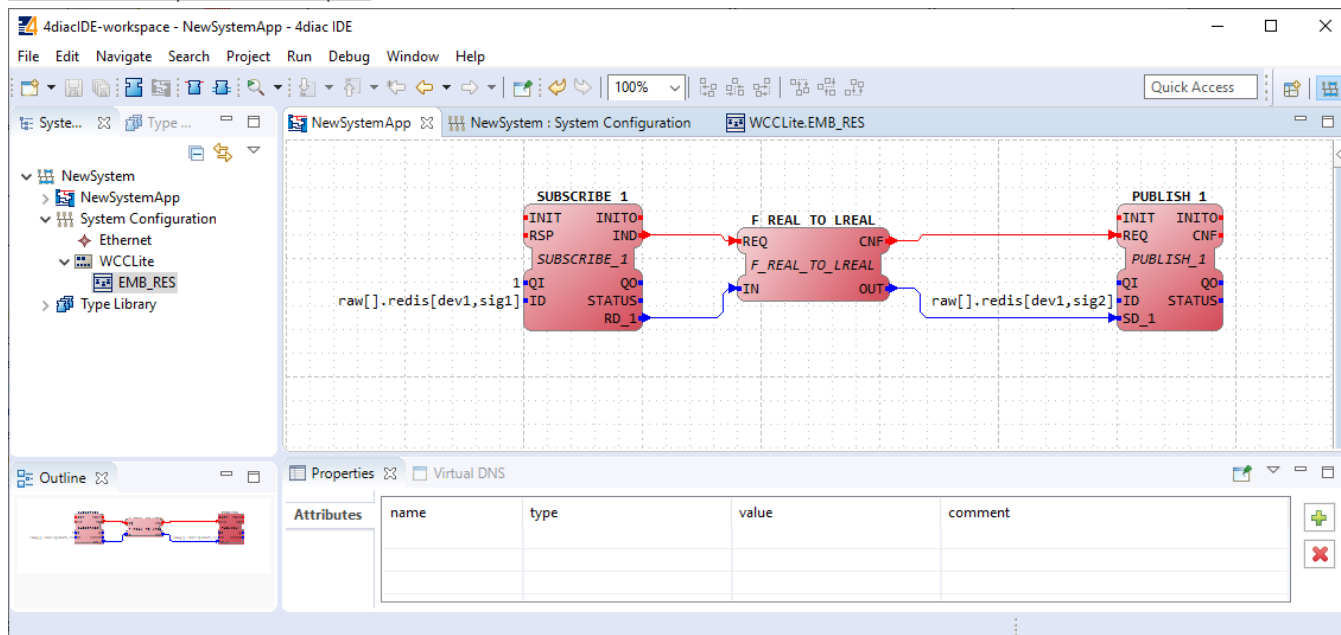
Parameter	Type	Description
signal_name	string	User-friendly signal name
device_alias	string	Device alias from a Devices tab
signal_alias	string	Unique signal name to be used
source_device_alias	string	device_alias of a source device
source_signal_alias	string	signal_alias of a source signal
enable	boolean	Enabling/disabling of an individual signal

If an upload consisting of configuration for IEC 61499 has been succesful, one should be able to access a configuration stored in /etc/iec61499.json file where protocol-specific parameters are shown in a JSON format. If the file is missing, make sure you have a correct firmware version installed and haven't made any typing errors.

Parameters mentioned earlier, namely device\_alias and signal\_alias, are the only parameters one needs to fill to bind Excel configuration to 4Diac framework. Two types of blocks are used for data transmission - PUBLISH blocks to write data to REDIS database and SUBSCRIBE blocks to acquire data from database as soon as it changes its value. Both of them have an ID connection. To connect a block to a datapoint, one should set this pin as `raw[[]].redis[device_alias,signal_alias]`, e.g. `raw[[]].redis[example_plc_device,example_plc_signal_alias]`.

An example with SUBSCRIBE and PUBLISH function blocks is shown below in image below.

Subscribe and publish examples



**Outputs of variable type ANY cannot be directly wired to inputs of the same type and therefore need to be explicitly typed using transitional function blocks**

**No more than 20 tags should be published over a period of 5 seconds, as this may overfill the queue. A "publish only on change" policy is advised.**

**Currently only PUBLISH\_1 and SUBSCRIBE\_1 function blocks are supported.**

If every step until now has been succesful, a user could now start debugging a PLC application.

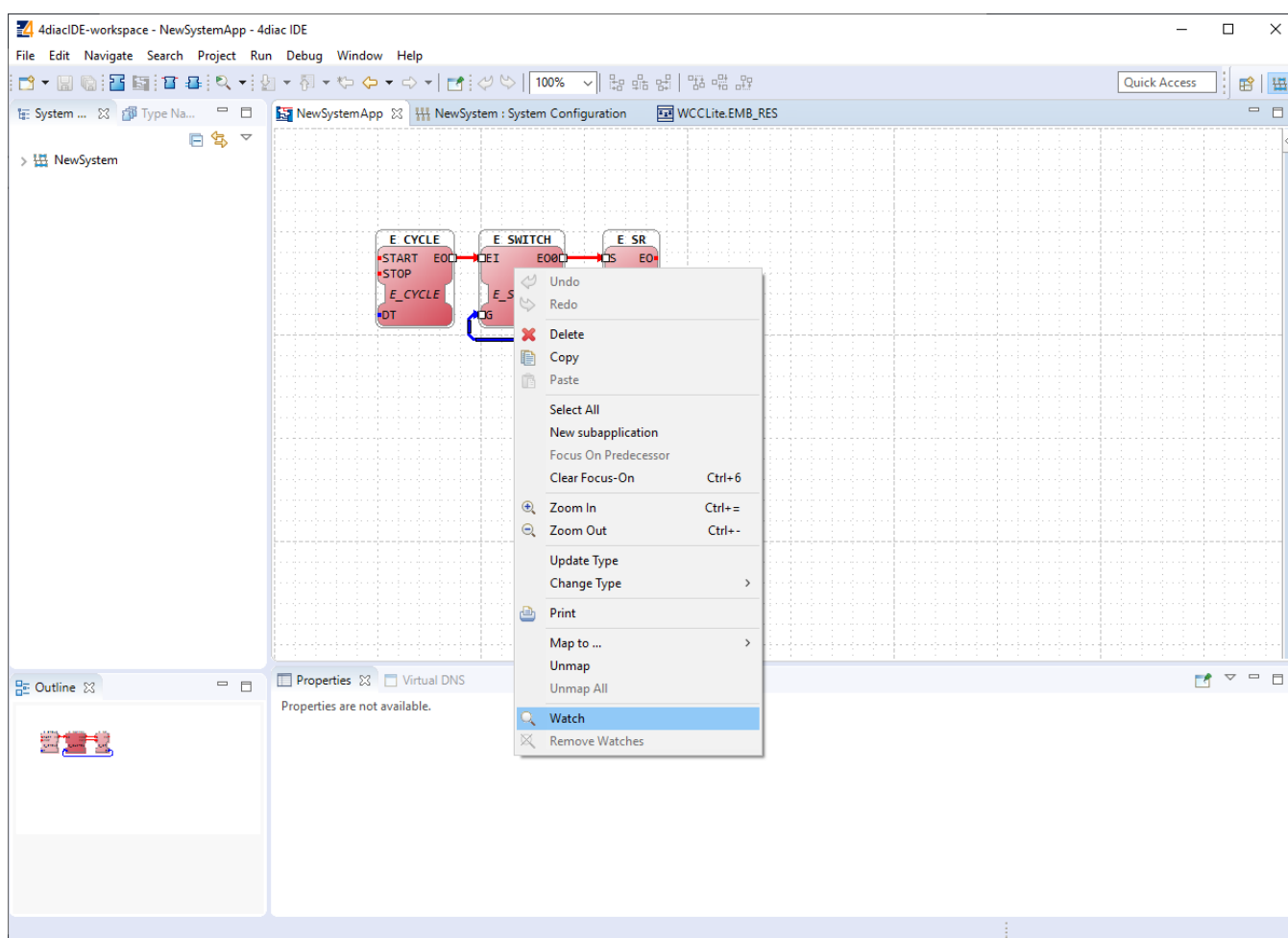
## Debugging an IEC 61499 application

After a project has been built and binded to an existing Excel configuration, a user would normally want to check if every part is working according to the prior requirements before compiling finished project and uploading it to production. Both 4Diac framework and WCC Lite offer tools for flexible debugging.

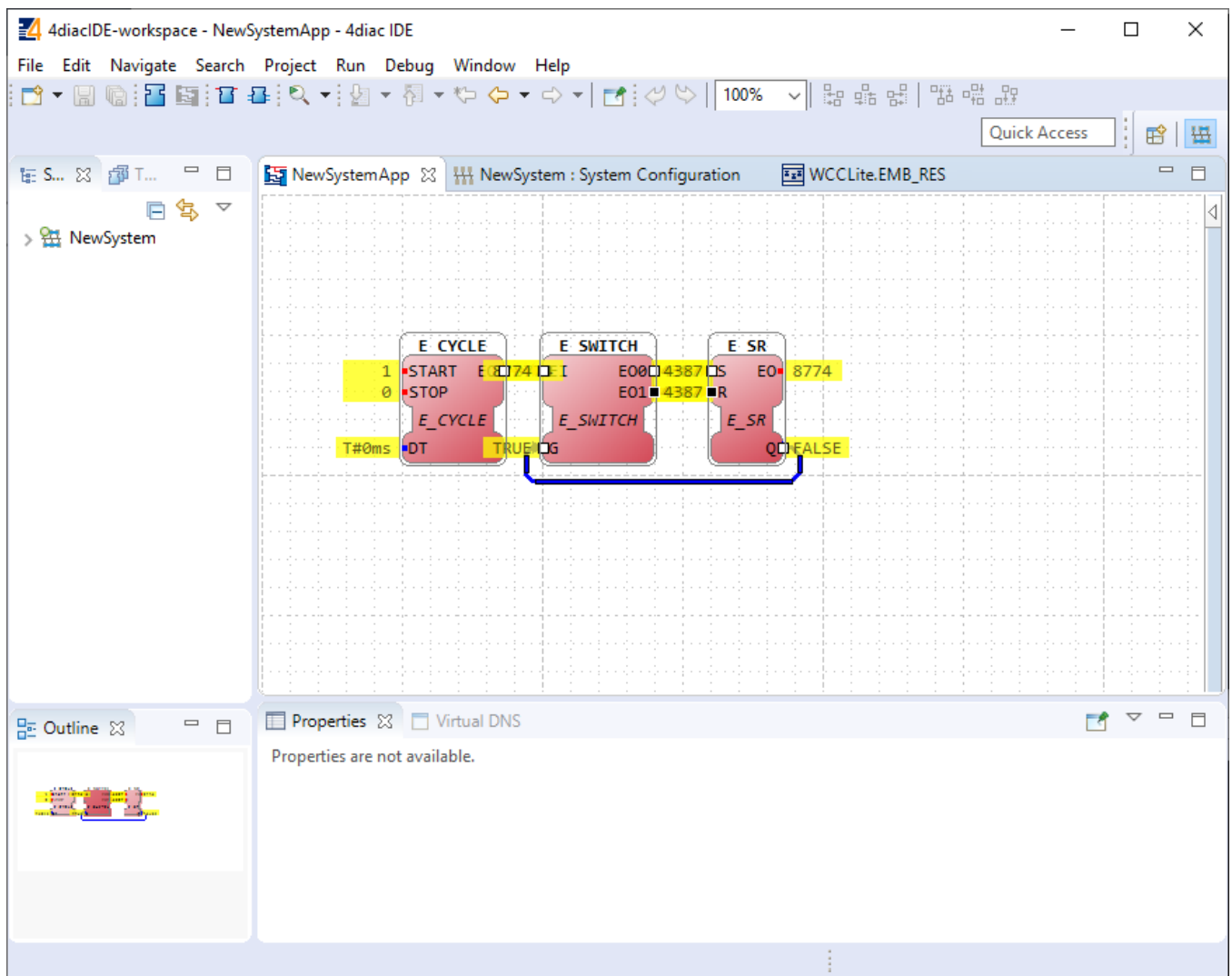
There is a possibility that 4Diac FORTE might not start as a process. It may happen if multiple faults occured and process has stopped. Process is also programmed to not start if no excel configuration file is found, therefore a user should make sure that Excel configuration is uploaded and ready for use.

Individual function blocks can be set to Watch mode: events can be triggered and values can be forced at inputs or outputs (look into images bellow). To monitor the function blocks, the application should be deployed and the IDE should be in Online mode (Debug -> Monitor System -> NewSystem).

Selecting watch mode:



Function blocks in watch mode:



Seeing information dynamically updated on 4Diac IDE might be very informative, however, some applications might require accessing WCC Lite via command-line interface. For example, in case of information not being updated one would want to assure that 4Diac FORTE in WCC Lite is not filtering data out but sending it to internal database (Redis). To run 4Diac FORTE debug from command-line interface, a user should write forte and press Enter. All possible choices are shown by adding -h flag. More flags are shown in a Table bellow. Make sure to stop any running process that could use the address that 4Diac framework is going to use.

Table. 4Diac FORTE command line debugging options:

- h - Display help information
- c <IP>:<port> - Set the listening IP and port for the incoming connections
- r - Show redis messages
- d <debug level> - Set debugging level
- f - Set the boot-file where to read from to load the applications