

Configuration of communications

Tip

To familiarize yourself with the basics of configuring communications, the following recordings from the webinar available on YouTube are also useful:

- [Communication: creating Modbus server and Modbus client](#)
- [Configuration of Modbus Client line](#)
- [Configuration of Modbus Client station](#)
- [Configuration of Modbus Client I/O tags](#)
- [Configuration of Modbus Server line](#)
- [Configuration of Modbus Server station](#)
- [Configuration of Modbus Server I/O tags](#)
- [Configuration of IEC-104 Client line](#)
- [Configuration of IEC-104 Client station](#)
- [Configuration of IEC-104 Client I/O tags](#)
- [Configuration of IEC-104 Server line](#)
- [Configuration of IEC-104 Server station](#)
- [Configuration of IEC-104 Server I/O tags](#)

We also recommend the blog:

- [SCADA architecture – DODM](#)

Architecture

Communication in D2000 is handled by the [D2000 KOM](#) process. The latter has as children communication lines (serial, TCP, UDP, OPC, file, etc.), these have children communication stations (on which the communication protocol is defined) and communication stations have children measured points (we call them *I/O tags*).

Note: this hierarchy is based on the serial communication model, where the communication line corresponds to the medium (or the serial port through which the line is accessed) and the stations correspond to the devices (PLC, RTU) connected to the medium.

Simple applications have a single KOM process which is the parent (grand-parent, grand-grand parent) of all communication objects. This process has an autostat enabled and is started by the D2000 Server when the D2000 system starts.

More complex applications may have dozens of communication processes (local and remote) licensed and running. Some communications require that the D2000 KOM process be run under a specific Windows user. An example is the OPC DA protocol (OPC Classic) or various file communications for accessing a remote server.

Advanced tip

Redundancy of the KOM process - instance processes.

The KOM process can also be started in instance (shadow) mode - then several KOM processes are connected to one D2000 Server as instances (no. 1, 2 ... upto 15) of the same KOM process (e.g. [1]_REMOTE.KOM , [2]_REMOTE.KOM - see start parameter [/W](#)). One instance is active (communicating), the others are passive.

Note: in the case of server protocols (e.g. [IEC 870-5-104 Server](#)), the KOM process can also be used in active-active mode (two/more KOM processes are active).

For more information, see the [Redundancy of Communication](#) chapter.

Communication objects are configured in the [D2000 Cnf](#) tool.

First, it is necessary to create a new [KOM process](#) or use an existing one (e.g. SELF.KOM).

Subsequently, it is necessary to create a [communication line](#) whose parent is the selected KOM process. It is important to select the correct line [category](#) (if it is necessary to change it later, this may require a restart of the KOM process). The support of communication lines for individual communication protocols is in the document [Communication Protocols](#).

In the next step, create a [communication station](#) whose parent is the configured line. The communication protocol and (for most communication lines) the station address are set at the station.

For some communication protocols, it is possible to set the protocol parameters in the line or station configuration.

The details of the configuration of lines and stations are in the documentation of individual [communication protocols](#).

Finally, the [I/O tags](#) are configured - the parent of the I/O tag is the communication station. The type must be defined for the I/O tag (input/output type digital/analog/integer/absolute time/relative time value). Filters, conversion to technical units, [process alarms](#), and other parameters can be [configured](#) at the I/O tag.

On the I/O tag, it is also possible to configure the [status text](#) (converting a number to a text constant for the display) and the [transformation palette](#) (which determines the standard format during display, unless it is changed by a [format mask](#)).

The KOM process performs the primary [processing of the values of the I/O tags](#). The details depend on the type of value (analog, integer, logic, four-state, text):

- Conversion to technical units (None / Linear conversion / Polynomial conversion / Pt 100 conversion)
- Value filtering (None / First order filter / Out of limits values are undefined / Value kept always in limits).
- Filtering based on the defined value absorption.
- Filtering based on the defined dead band.
- [Value limits](#) are checked.
- Value inversion (Boolean value)

For some protocols, it is possible to activate the reading of historical values (e.g. stored in electricity meters or data loggers) with the command [GETOLDVA](#). Examples of such protocols are [OPC](#), [DLMS/COSEM](#), [IEC 62056-21:2002 Serial](#), [Datalogger ESC8800](#), [Datalogger ESC8816](#), [ELGAS-2 \(CP II\)](#), or [IEC 870-5-104](#).

Tip

Communication tuning: on the same configuration tab as the line category is set, it is also possible to set [Communication tracing](#) - to the disk or to the screen (the process screen can be displayed with the [D2000 System Console](#) tool. Some protocols have further parameters that can be used to increase the logging level (e.g. [MODBUS Client](#) has [Full Debug](#) protocol parameter, and [IEC 870-5-104](#) protocol has [Debug Input](#) and [Debug Output](#) parameters).

Turning on additional tuning information for a specific I/O tag(s) is possible with the command [DI ON <mask>](#).

Advanced tips

- The KOM process is able to start and function even without a connection to the D2000 Server ([KOM Archive mode](#)). In this mode, it works with the last known configuration, it saves the values obtained from the communication to the disk and sends them to the D2000 Server (where they are archived) after the connection is restored. This mode is useful for remote KOM processes located on communication servers close to the technology - in case of loss of connectivity to the D2000 Server, historical data will not be lost.
- The [Generic User Protocol](#) is intended for the implementation of simple protocols in ESL script (see links to blogs in the protocol documentation).
- Custom protocols can be integrated into the KOM process as dynamic libraries (*dll* on Windows, *so* on Linux). The [D2000 KomAPI](#) interface is available for developing your own protocols.