Reliance
Industrial SCADA/HMI system

Reliance 4

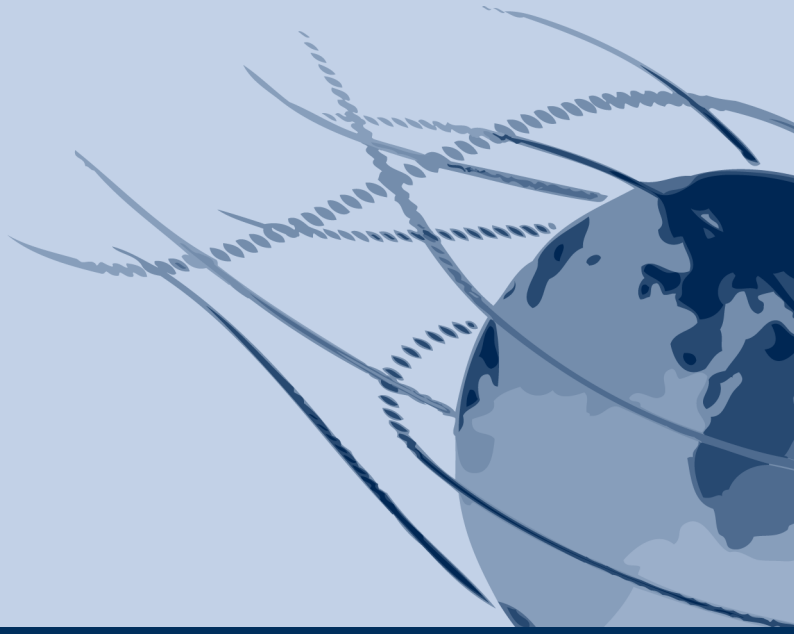
DATA EXCHANGE METHODS

GEOVAP

Reliance
Industrial SCADA/HMI system

Reliance 4

DATA EXCHANGE METHODS



GEOVAP

GEOVAP, spol. s r. o.
Cechovo nabrezi 1790
530 03 Pardubice
Czech Republic
+420 466 024 618
http://www.geovap.cz

Table of Contents

1 Introduction

The **Reliance** SCADA/HMI system can be used to exchange data with third party applications. **Reliance** supports many different standards, interfaces, and protocols, so, depending on transfer direction and third party application capabilities, the most suitable data exchange method can be selected. The goal of this document is to provide an outline of data transfer methods supported in the **Reliance** SCADA/HMI system, to describe advantages and disadvantages of each method, and to help the user (systems integrator) choose the most suitable method to solve the problem he/she wants to solve.

2 Supported Data Exchange Methods

2.1 Via Files

The data exchange method *Via Files* is based on a shared file to which data is logged by a first application and from which data is read by a second application. A file format has to be clearly specified and both applications must have access to this file (usually the CSV file format is used). Two files have to be used for bidirectional data transfer. Files can be accessed via scripts from the **Reliance** system. In *VBScript*, the `Scripting.FileSystemObject` object is used to perform read/write operations on files. The following examples demonstrating access to files are part of the **Reliance 4** system installation: `LogMessage`, `LogDataToFileAndDb`, `ExportToCSV`, `ImportFromCSV`.

2.2 Via SQL Database

Exchanging data *via an SQL database* is similar to the *Via Files* method, except that shared data storage is represented by an SQL database. A first application logs data to a data table and a second application reads this data table. A single data table can be used even for bidirectional data transfer. Both applications have to be able to connect to the SQL database and to access the specified data tables. The **Reliance** system allows for accessing an SQL database natively (historical data logging and alarm/event logging). User-defined access to an SQL database can be implemented via the VBScript `ADODB.Connection` object. The following examples demonstrating access to an SQL database are part of the **Reliance 4** system installation: `LogDataToSQLServer`, `SQLFromScript`, `ODBCFromScript`.

2.3 Via DDE

The Microsoft DDE standard (*Dynamic Data Exchange*) is designed to exchange data between applications (between a DDE client and a DDE server). The **Reliance** system supports the DDE technology and can act as both a server and a client.

2.3.1 Reliance as a DDE Server

If the **Reliance** system is configured as a *DDE server*, the values of tags defined in the project can be provided to *DDE clients*. For each tag that should be accessed by a client application, the so called **DDE Item** must be defined. The *DDE Item* is a value identifier used by the DDE protocol. A DDE client can be any other application supporting the DDE standard, e.g., MS Excel. The client application DDE connection syntax is in the following form:

`program_name| dde_topic! dde_item,`

where

`program_name`

is the runtime software filename without an extension,

`dde_topic`

is always "`DdeServer`" in the **Reliance** system.

Example:

A project contains a tag whose *DDEItem* is "`Tag1`". The project is run in the *Reliance Control* runtime software (`R_Ctl.exe`). The connection syntax will have the following form:

`R_Ctl| DdeServer! Tag1`

Note: In the *MS Excel* program, the syntax begins with the "=" character. The DDE-connected cell will thus contain: "`=R_Ctl| DdeServer! Tag1`" (without quotes).

2.3.2 Reliance as a DDE Client

If the **Reliance** system is configured as a *DDE client*, it can receive values from a *DDE server*. In the *Device Manager*, a new DDE device has to be defined. The name of the *DDE server* must be entered into the respective field (the filename of the server application without an extension). The *DDE Item* property has to be specified for tags defined within this device. **DDE Topic** (depends on the DDE server) has to be defined via the *Project Structure Manager* for the *Channel* object. In this configuration, data can be received, for example, from the

communication driver that is a DDE server or from other **Reliance** projects.

Both the *DDE server* and the *DDE client* must run on the same computer. DDE via a network (*NetDDE*) is not reliable and not supported by **Reliance**. The following examples demonstrating data exchange via DDE are part of the **Reliance 4** system installation: `DDEClient`,`DDEServer`.

2.4 Via COM/DCOM/ActiveX

Microsoft *Component Object Model* (COM) is an interface standard used for remote procedure calls and for data exchange between applications (between a *COM server* and a *COM client*). The standard fully replaces the older DDE standard. DCOM is a COM variant designed for communication among computers via a network. Using scripts, the **Reliance** system is able to connect to any COM server installed on the computer. If the COM server is built into an external application, it is possible to provide bidirectional data exchange between the application and **Reliance**. To use connection via COM, it is required to know a COM object identifier (`ProgId`) and the syntax of its procedures and functions.

It is recommended that both the *COM client* (**Reliance**) and the *COM server* should run on the same computer.

DCOM is rather unreliable and difficult to configure.

The following example demonstrating data exchange via COM is part of the **Reliance 4** system installation: `CommWithExternalProgram`.

2.5 Via Generic Driver

Generic Driver is a communication driver available with the **Reliance 4** SCADA/HMI system. It is designed for data exchange (through a serial port or Ethernet) between **Reliance** and devices for which a native driver or an OPC/DDE server is not available. In the future, *Generic Driver* should be used instead of *External Communicator*. Communication with a connected device is realized through special tags defined within the Generic device via the *Device Manager*. In comparison with *External Communicator*, *Generic Driver* is more deeply integrated into a visualization project, so it is easier to use, diagnostics is available, and it is likely to be further improved along with other native communication drivers. The following examples are part of the **Reliance 4** system installation: *Generic_SimpleText* and *Generic_Teco*.

2.6 Via COM (Internal COM Server)

The **Reliance** runtime software contains a built-in *COM server*, which enables external applications (*COM clients*) to run/end a project, to read/write data (current tag values), to activate a visualization window, etc. The built-in COM server interface is obsolete and available for backward compatibility only. For more information on this topic, please contact **GEOVAP**'s **Reliance** technical support (support@reliance.cz).

2.7 Via OPC

OLE for Process Control (OPC) is a standard designed for the exchange of data between devices and applications used in the process control field. The OPC standard is based on the COM interface (Component Object Model). An *OPC server* is software designed to communicate with a specific hardware device and passes data to *OPC clients* via a standard interface. The **Reliance** system supports the OPC technology and can act as both a server and a client.

2.7.1 Reliance as an OPC Server

The **Reliance** system can act as an *OPC server*, which allows passing data to *OPC clients*.

Reliance OPC Server is a Reliance 4 module that allows accessing tags defined within the visualization project via a standard OPC interface. It is, therefore, possible to pass data from the visualization project to information systems operating at a customer site. It can be, for example, a customer information system or another SCADA system acting as an *OPC client* and with which data exchange is required. *Reliance OPC Server* can also be used when it is necessary to interconnect two different **Reliance** visualization projects or projects running in different versions of **Reliance** (e.g., version 3 and 4).

Reliance's data server (i.e., *Reliance Server* or *Reliance Control Server*) can operate on a computer that is different from the one on which *Reliance OPC Server* is running. However, it is recommended that *Reliance OPC Server* and the *OPC client* should run on the same computer. Connection to a remote OPC server is possible but not advised (it is based on the DCOM protocol and therefore difficult to configure and not very reliable).

2.7.2 Reliance as an OPC Client

The **Reliance** system is an *OPC client*, which allows obtaining data from an *OPC server*. There are also OPC servers designed exclusively for sharing data between two *OPC clients*. Such OPC

servers do not communicate with any hardware device, the data is stored in memory instead, and provided to the clients (e.g., *MatrikonOPC Caching Server*). This OPC server can be an intermediary for transferring data between two different SCADA/HMI systems or different **Reliance** visualization projects.

It is recommended that both the *OPC server* and the *OPC client* should run on the same computer. Connection to a remote OPC server is possible but not advised (it is based on the DCOM protocol and therefore difficult to configure and not very reliable).

2.8 Via Web Services

Reliance's *data servers* (*Reliance Server* or *Control Server*) contain a built-in web server. The web server not only provides WWW pages, but it can also provide the so called **Web services**. A *Web service* is part of the data servers designed to exchange data with third party applications locally or via the Internet. For a detailed description of the Web service, please see the *Data Servers* document (chapter *Interface for third party applications*). The **Reliance 4** system installation contains the source code of the programs demonstrating connection to the interface. The examples are available for Object Pascal (Delphi), C#, VisualBasic.NET (Microsoft Visual Studio), Java, and Java (Android).

3 Recommendation

Due to a wide variety of requirements for data exchange between the **Reliance** SCADA/HMI system and external applications, it is not possible to generally recommend one of the above described methods. If **Reliance** is a data source (server), it is advised to use the web services, OPC, shared files, or SQL databases. If **Reliance** should be a client, the choice of the best method depends on the data exchange requirements and on the capabilities of an external application. The most frequently used methods are OPC, Generic Driver, shared files, or SQL databases.

4 Summary of Data Exchange Methods

	Runtime software	Reliance is a client	Reliance is a server	Network
Shared Files	all	yes	no	yes[2]
SQL Database	all	yes	no	yes
DDE	all	yes	yes	no
COM/DCOM/ActiveX	all	yes	no	yes[1]
Generic Driver	all	yes	no	yes
COM (internal)	all	no	yes	yes[1]
OPC	all	yes	yes	yes[1]
Web Services	Data servers	no	yes	yes

[1] Not recommended (problems with DCOM).
[2] Local network only.

	Platform independent	Deprecated (Obsolete)	Requires script programming	Requires application programming
Shared Files	no	no	yes	no
SQL Database	no	no	yes	no
DDE	no	no	no	no
COM/DCOM/ActiveX	no	no	yes	no[3]
Generic Driver	yes	no	yes	no
COM (internal)	no	yes	no	yes
OPC	no	no	no	no
Web Services	yes	no	no	yes

[3] When third party COM/DCOM is used.