

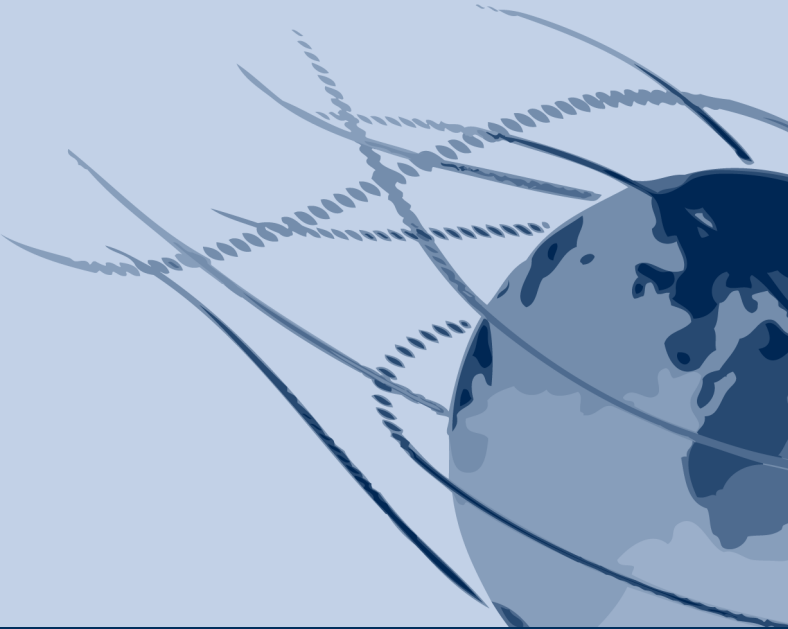
# Reliance 4

## DATA EXCHANGE METHODS



# Reliance 4

## DATA EXCHANGE METHODS



**© 2011 GEOVAP, spol. s r.o. All rights reserved.**

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

Products that are referred to in this document may be trademarks and/or registered trademarks of the respective owners.

While every precaution has been taken in the preparation of this document, GEOVAP, spol. s r.o. assumes no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall GEOVAP, spol. s r.o. be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

# Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>1</b>
<b>2</b>	<b>Supported Data Exchange Methods</b> .....	<b>2</b>
<b>2.1</b>	<b>Via Shared Files</b> .....	<b>2</b>
<b>2.2</b>	<b>Via SQL Database</b> .....	<b>2</b>
<b>2.3</b>	<b>Via DDE</b> .....	<b>3</b>
	<b>2.3.1 Reliance Configured as a DDE Server</b> .....	<b>3</b>
	<b>2.3.2 Reliance Configured as a DDE Client</b> .....	<b>3</b>
<b>2.4</b>	<b>Via COM/DCOM/ActiveX</b> .....	<b>4</b>
<b>2.5</b>	<b>Via Reliance External Communicator</b> .....	<b>4</b>
<b>2.6</b>	<b>Via COM (Internal COM Server)</b> .....	<b>4</b>
<b>2.7</b>	<b>Via OPC</b> .....	<b>5</b>
<b>2.8</b>	<b>Via Web Services</b> .....	<b>5</b>
<b>3</b>	<b>Recommendation</b> .....	<b>6</b>
<b>4</b>	<b>Summary</b> .....	<b>7</b>

# 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 that 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 (system integrator) to choose the most suitable method to solve the problem he/she wants to solve.

## 2 Supported Data Exchange Methods

Via files

Via SQL database

Via DDE

Via COM/DCOM/ActiveX

Via Reliance External Communicator

Via COM (internal COM server)

Via OPC

Via web services

### 2.1 Via Shared Files

The data exchange method *Via files* is based on a shared file to which data is logged by the first application and from which data is read by the second application. File format has to be clearly specified and both applications must have access to this file (usually 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 **Reliance 4** system installation contains the following examples demonstrating access to files: `LogMessage`, `LogDataToFileAndDb`, `ExportToCSV`, `ImportFromCSV`.

### 2.2 Via SQL Database

The data exchange method *Via SQL Database* is similar to the *Via file* method, except shared data storage is represented by an SQL database. The first application logs data to a data table and the 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 specified data tables. The **Reliance** system allows for accessing an SQL database natively (historical data and alarm/event logging). User defined access to an SQL database can be implemented via the VBScript **ADODB.Connection** object. The **Reliance 4** system installation contains the following examples demonstrating access to an SQL database: `LogDataToSQLServer`, `SQLFromScript`, `ODBCFromScript`.

## 2.3 Via DDE

The Microsoft DDE standard (*Dynamic Data Exchange*) is designed to exchange data between applications (between DDE client and DDE server). The **Reliance** system supports DDE standard and can work both as a [Server](#) and as a [Client](#).

### 2.3.1 Reliance Configured 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, 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 DDE standard, e.g. MS Excel. 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:

Project contains a tag with the *DDEItem* defined as "Tag1". Project is run in the *Reliance Control* runtime software (`R_Ctl.exe`). Connection syntax will have the following form:

```
R_Ctl| DdeServer! Tag1
```

**Note:** In the *MS Excel* program, the syntax begins with the "=" character. DDE connected cell will contain: "=R\_Ctl| DdeServer! Tag1" (without quotes).

### 2.3.2 Reliance Configured 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). *DDE Item* has to be defined for tags defined in 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, the data can be received e.g. from communication driver, that is DDE server or from other **Reliance** projects.

*DDE server* and *DDE client* have to run on the same computer. DDE via network (*NetDDE*) is not reliable and is not supported by **Reliance**. The **Reliance 4** system installation contains the following examples demonstrating data exchange via DDE: `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 *COM Server* and *COM Client*). The standard fully replaces the older DDE standard. DCOM is a COM variant designed for communication among computers via network. The **Reliance** system is able to connect to any *COM Server* installed on the computer from scripts. If the COM server is built into an external application, it is possible to provide a bidirectional data exchange between the application and **Reliance**. To use connection via COM, it is required to know COM object identifier (`ProgId`) and its procedures and functions syntax.

It is advised to run *COM Client* (**Reliance**) and *COM Server* on the same computer, because DCOM is rather unreliable and difficult to configure.

The **Reliance 4** system installation contains the following examples demonstrating data exchange via COM interface: `DDEClient`, `DDEServer`.

## 2.5 Via Reliance External Communicator

*Reliance External Communicator* is a program designed for user-defined communication with a HW device via RS-232 port or via Ethernet (TCP/IP). It can also be used to exchange data with an external application communicating using the TCP/IP protocol. In this configuration the **Reliance system** is a *COM client*, **External Communicator** is a *COM server* and a *TCP client* at the same time and the external application is a *TCP server*.

**Note:** *Reliance External Communicator* is a COM server and therefore notes included in the [Via COM/DCOM/ActiveX](#) topic apply to it.

## 2.6 Via COM (Internal COM Server)

**Reliance** runtime software contains a built-in *COM server*, which enables external applications (*COM clients*) to run/end project, to read/write data (current tag values), to activate visualization window etc. Built-in COM server interface is obsolete and it is available for backward compatibility only. For more information on this topic, please contact the **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 HW device (via device specific interface) and with one or more *OPC clients* via standardized OPC interface. The **Reliance** system is an *OPC client* so it can connect to an *OPC server*.

There are also OPC servers designed exclusively for sharing data between two OPC clients. Such OPC servers don't communicate with any HW 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 to transfer data between two different SCADA/HMI systems or different Reliance visualization projects.

It is advised to run OPC server and OPC client on the same computer. Connection to a remote OPC sever 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

The **Reliance** *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 so called **Web services**. A *Web service* is a 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. Examples are available for Object Pascal (Delphi), C# and VisualBasic.NET (Microsoft Visual Studio) development environments.

### 3 Recommendation

It is not possible to generally recommend one of the described methods, because it depends on the data exchange requirements and/or on capabilities of an external application. If **Reliance** is a data source (server), it is advised to use the web services, shared files or DDE. If **Reliance** should be a client, the choice of the best method depends on the capabilities of an external application. Most frequently used methods are shared files and DDE.

## 4 Summary

	Runtime software	Reliance is a Client	Reliance is a Server	Network
Shared Files	all	yes	no	yes <sup>2</sup>
SQL Database	all	yes	no	yes
DDE	all	yes	yes	no
COM/DCOM/ActiveX	all	yes	no	yes <sup>1</sup>
External Communicator	all	yes	no	yes
COM (internal)	all	no	yes	yes <sup>1</sup>
OPC	all	yes	no	yes <sup>1</sup>
Web Services	Data servers	no	yes	yes

<sup>1</sup> Not recommended (problems with DCOM)

<sup>2</sup> 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 <sup>3</sup>
External Communicator	yes	no	yes	no
COM (internal)	no	yes	no	yes
OPC	no	no	no	no
Web Services	yes	no	no	yes

<sup>3</sup> When third party COM/DCOM is used