

TABLE OF CONTENTS

1. First steps

1.1	Introduction.....	3
1.2	Creating a new project.....	4
1.3	Configuring a project.....	5
1.3.1	Managers.....	5
1.3.2	Device manager – defining devices, variables and alarms.....	5
1.3.3	Other managers – defining databases, trends and other objects.....	8
1.3.4	Project structure manager – defining project structure....	9
1.3.5	Designing project windows.....	10
1.3.6	Starting the project in runtime mode.....	11
1.4	Example	12
1.4.1	Creating a project.....	12
1.4.2	Adding a device and a variable.....	12
1.4.3	Connecting a device to a computer	13
1.4.4	Configuring components.....	13
1.4.5	Starting the project in runtime mode.....	14
1.4.6	Testing the created project.....	14

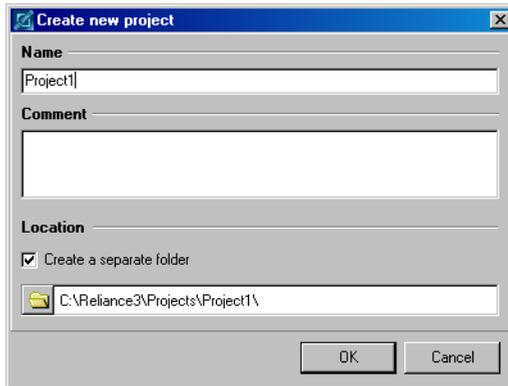
1. FIRST STEPS

1.1 INTRODUCTION

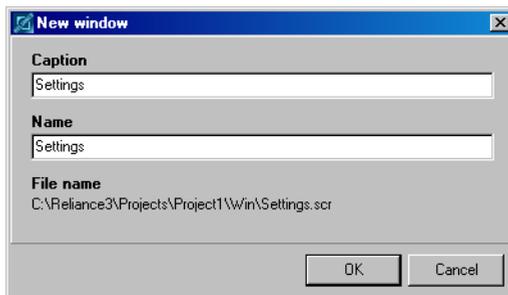
This quick start manual will give you a basic working knowledge of *Reliance* SCADA/HMI System. You will learn how to create a new visualization project, define a device, add variables and alarms, design a visualization window and run the project.

1.2 CREATING A NEW PROJECT

After starting the development environment of *Reliance Design*, select the **File** ▶ **New project...** command from the main menu. Then, the **Create new project** dialog box appears. Specify a name for the new project and select its location on the hard drive of your computer or leave the default path. Optionally, specify a comment.



After confirming the **Create new project** dialog box, the **New window** dialog box appears.



Specify a caption and name for the new visualization window and confirm the dialog box.

1.3 CONFIGURING A PROJECT

1.3.1 Managers

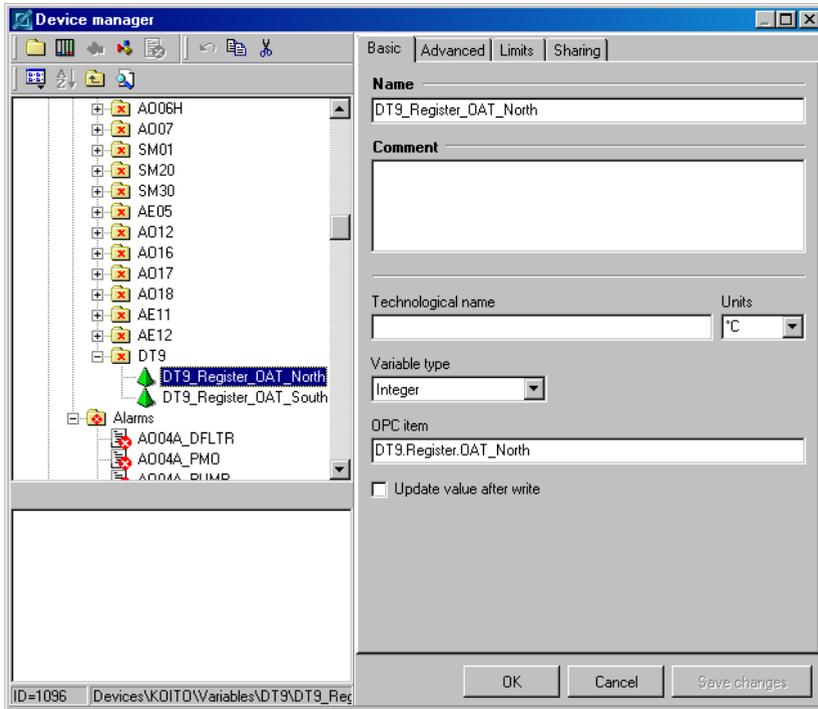
In the context of *Reliance*, managers are tools designed to define and configure objects that are the building blocks of every *Reliance* visualization project. These are computers, users, devices, databases, trends, reports, print reports, recipes, scripts, windows, components and others.

1.3.2 Device manager – defining devices, variables and alarms

In the context of *Reliance*, a device is an object representing a physical device (e.g. PLC or other similar I/O device used for industrial automation and control) or a so-called virtual device. Within each device, you can define variables and alarms linked to the variables. The values of the variables are used to generate alarms (e.g. alerts), can be displayed via visual components, can be used to control the appearance and behavior of components, can be logged to databases, processed by scripts, etc. If a device represents a physical device, the runtime software obtains the values of the variables from the device's communication driver or another instance of the runtime software (e.g. through a network connection). Otherwise, the device's variables only exist in the memory of the computer on which the visualization project is running and are referred to as internal variables.

Every visualization project contains a predefined virtual device named *System*. This device is automatically accessible to each computer defined within a visualization project (i.e. you need not connect this device to a logical computer via the ***Project structure manager***). It is intended for defining private internal variables. The variables defined within the *System* device are not included in the total number of data points used in a visualization project.

To bring up the ***Device manager***, select the ▶ ***Managers ▶ Device manager...*** command from the main menu.



To add a new device, select the *New device* command from the toolbar or the local menu of the tree diagram. Properties of a device differ according to device type. A device always contains folders *Variables* and *Alarms*. Some types of devices can also contain a folder called *Communication zones*. Communication zones enable you to control and optimize reading data from an I/O device. Only variables covered with a communication zone will be updated by the communication driver. Communication zones do not affect writing data to an I/O device from a visualization.

To add a new variable, select the *New variable* command from the toolbar or the local menu of the tree diagram or press the *Insert* key on the keyboard. However, a new variable can only be added if the object selected in the tree diagram is the *Variables* folder or an object subordinated to this folder (e.g. a variable). Properties of a variable differ according to device type. However, some properties are common to variables from all device types, e.g. the name, units of measurement, critical and warning limits. Variables can also be imported to a device.

To add a new alarm, select the *New alarm* command from the toolbar or the local menu of the tree diagram or press the *Insert* key on the keyboard. However, a new alarm can only be added if the object selected in the tree diagram is the *Alarms* folder or an object subordinated to this folder (e.g. an alarm). Properties of an alarm determine the name, link to a variable, occurrence generating the alarm, alarm type, access rights required for acknowledgement of the alarm by the operator, other events that should occur after the alarm is generated, ended or acknowledged, etc.

To add a new communication zone, select the *New communication zone* command from the toolbar or the local menu of the tree diagram or press the *Insert* key on the keyboard. However, a new communication zone can only be added if the object selected in the tree diagram is the *Communication zones* folder or an object subordinated to this folder (e.g. a communication zone). Properties of a communication zone differ according to device type.

1.3.3 Other managers – defining databases, trends and other objects

In a visualization project, you can define other objects to enhance the functionality. Tools for defining and configuring these objects can be found in the ► *Managers* menu.

The **Database manager** allows you to define and configure databases. In the context of *Reliance*, a database is an object representing a physical database stored on a hard disk. By defining a database you enable the runtime software to access (read and/or write) historical data.

The **Trend manager** allows you to define and configure trends. In the context of *Reliance*, a trend is an object used for graphic presentation of the data stored in a database(s) (i.e. historical data). Trends can be displayed at runtime via the trend viewer.

The **Real-time trend manager** allows you to define and configure real-time trends. In the context of *Reliance*, a real-time trend is an object used for graphic presentation of a sequence of the most recent values (time samples) of the selected variables. The values (time samples) are only stored in the memory, not in a database. A real-time trend can be displayed at runtime by a *Real-time trend* component placed into a project window.

The **Report manager** allows you to define and configure reports. In the context of *Reliance*, a report is an object used for graphic presentation of the data stored in a database (i.e. historical data) in a tabular format. Reports can be displayed, printed and exported at runtime via the report viewer.

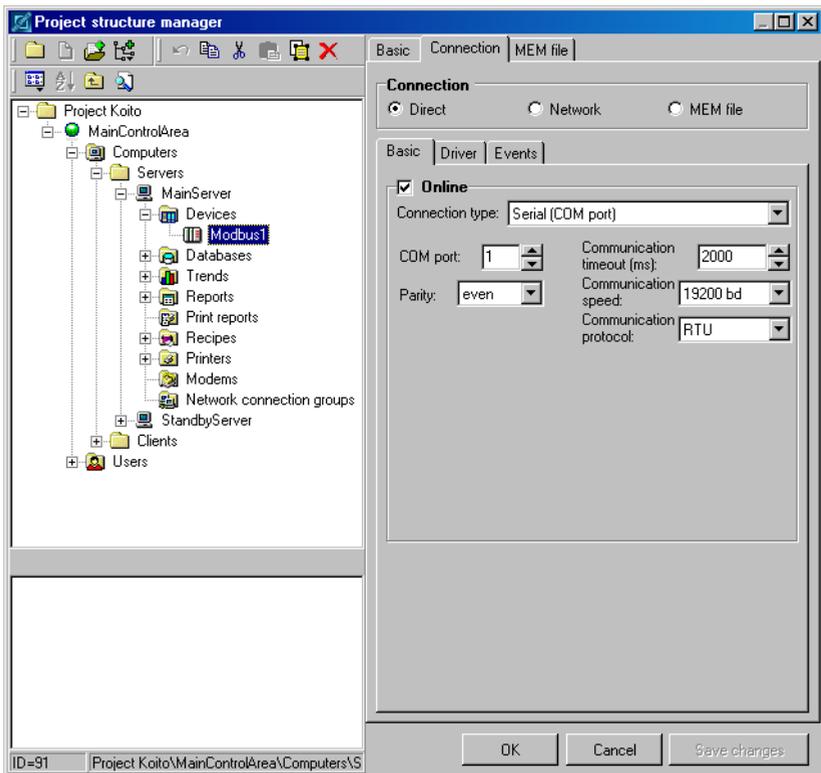
The **Print report manager** allows you to define and configure print reports. In the context of *Reliance*, a print report is an object used for graphic presentation of real-time values of the selected variables in a user-defined format. Print reports can be displayed, printed and exported at runtime via the print report viewer. When defining a new print report, it is necessary to prepare a template in text or HTML format and save it to a disk file. The template may also include graphic elements if using HTML format.

The **Recipe manager** allows you to define and configure recipes. In the context of *Reliance*, a recipe is an object representing a group of variables. It is used to store real-time values of the variables to a disk file. A stored recipe can later be loaded from the file and transferred to the respective devices.

1.3.4 Project structure manager – defining project structure

The *Project structure manager* is a tool designed for defining the structure of an entire visualization project. It enables you to define the structure consisting of control areas, computers, users, network connections, devices, databases and other objects, so that it corresponds to a real plant site.

A visualization project always comprises at least one control area. A control area comprises at least one computer and any number of users. A typical example of a control area is a control room. The objects defined through other managers, such as devices, databases, trends and reports, can be made accessible to a computer by connecting them to the computer, i.e. adding links to the objects to the computer's appropriate folder.



For example, if a computer is to read data from a device (e.g. a PLC), you have to connect the device (previously defined via the **Device manager**) to the computer. To do it, select the *Devices* folder of the computer in the tree diagram and choose the *Connect objects* command from the toolbar or the *Connect devices* command from the local menu of the tree diagram. Then configure the properties of the device's connection to the computer (connection type, communication options, etc.). The same applies to other previously defined objects. For example, if a computer is to log data to a database, you have to connect the database (previously defined via the **Database manager**) to the computer.

1.3.5 Designing project windows

A visualization project usually contains one or more project windows. A project window (also referred to as visualization window) is intended to become a custom graphical view of an industrial process. At runtime, a project window becomes part of the user interface of the runtime software, which is used by the operator to monitor (and control) the visualized process.

To view or edit the properties of a project window, choose the ► **Window properties...** command from the local menu of the window or double-click the window. This will bring up the **Window properties** dialog box.

When designing a window, add individual components from the component palette to the window. A component selected on the palette by clicking the left mouse button may be placed into a window either with its default size (by clicking on the window area), or with the size defined by dragging the mouse inside the window area (before releasing the mouse button).

☺ **HINT** To add multiple components of the same type, press the *Shift* key while selecting the component on the palette.

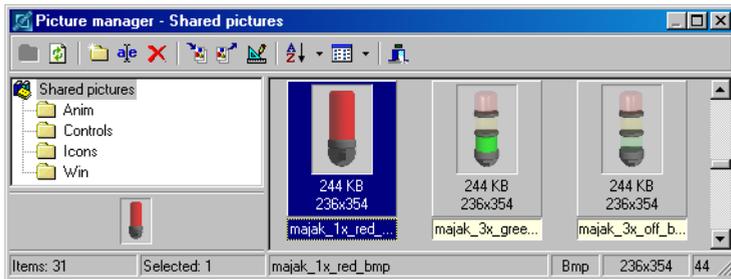
To change the position or size of components or component groups, edit functions may also be used – e.g. centering, alignment, changing the Z-order, modification of height and width, transformation, etc. The *Undo* and *Redo* commands allow you to undo and redo up to the last 100 edit actions.

The last step when designing a window is configuring the properties of the components, which affects their appearance and behavior. To configure the properties of a single component, use the component's property editor (double-click the component or choose the ► **Component properties...** command

from the component's local menu). The properties of individual components differ depending on their type.

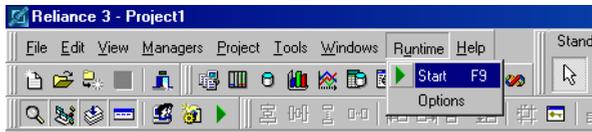
- ☺ **HINT** To configure the properties of multiple components at the same time, select the components and edit their properties through the **Component manager**.

Some components can display a picture or multiple pictures. If you want to use a picture in a visualization project, it must be first imported to a picture database of the project via the **Picture manager**. If the picture is to be only used in a single project window, it should be imported to the picture database of the window. Otherwise, it should be imported to the shared picture database. To use the **Picture manager** for managing the picture database of the window, choose the **Window picture manager...** command from the window's local menu. To use the **Picture manager** for managing the shared picture database, choose the **Managers ▶ Shared picture manager** command.



1.3.6 Starting the project in runtime mode

To start the open project from the development environment, select the **Runtime ▶ Start** command from the main menu.



1.4 EXAMPLE

1.4.1 Creating a project

1. Select the ▶ **File** ▶ **New project...** command from the main menu.
2. In the **Create new project** dialog box, specify a name and location for the new project or use the default settings.

Recommendation: Leave the *Create a separate folder* option active.

3. In the **New window** dialog box, specify a caption and name for the new visualization window or use the default settings.

1.4.2 Adding a device and a variable

1. Select the ▶ **Managers** ▶ **Device manager** command from the main menu.
2. Select the *New device* command from the toolbar and choose the *Virtual device* item from the popup menu to add a virtual device.
3. Optionally, change the name of the device on the **Basic** page.

Recommendation: Keep the default name “Virtual1”.

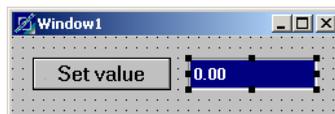
4. Select the *Variables* folder of the new device in the tree diagram.
5. Add a new variable by invoking the *New variable* command from the toolbar or pressing the *Insert* key on the keyboard.
6. Configure the variable’s properties on the **Basic** page as follows:
Name “Required_Temperature”, *Units* “°C”, *Variable type* “Integer”.
7. Configure the variable’s properties on the **Advanced** page as follows:
Character count = 5, *Dec. place count* = 1.
8. Close the **Device manager** by clicking the *OK* button.

1.4.3 Connecting a device to a computer

1. Select the **► Managers ► Project structure manager** command from the main menu.
2. Select the *Devices* folder of *Computer1* in the tree diagram.
3. Select the *Connect objects* command from the toolbar and choose the device (previously defined via the **Device manager**) from the list in the *Select device* dialog box.
4. Close the **Project structure manager** by clicking the *OK* button.

1.4.4 Configuring components

1. From the Component palette, add a new *Display* component (by default, the first from the left on the palette) to the project window.
2. Bring up the component's property editor (the **Display properties** dialog box) by double-clicking the component.
3. On the **Functions** page in the *Main variable* edit box, define a link to the variable *Virtual1.Required_Temperature*.
4. On the **Units** page, activate (check) the *Show* property.
5. Close the property editor by clicking the *OK* button.
6. From the Component palette, add a new *Button* component (by default, the second from the left on the palette) to the project window.
7. Bring up the component's property editor (the **Button properties** dialog box) by double-clicking the component.
8. On the **Functions** page in the *Main variable* edit box, define a link to the variable *Virtual1.Required_Temperature*.
9. On the **States/State 0** page in the *Text* edit box, change text to "Set value".
10. Close the property editor by clicking the *OK* button.
11. Widen the button so that the whole text is visible on it.



1.4.5 Starting the project in runtime mode

1. Select the ▶ *Runtime* ▶ *Start* command from the main menu.

1.4.6 Testing the created project

1. After starting the project in runtime mode, the *Display* component shows a value of 0 °C.
2. Click the *Button* component to bring up the *Enter value for variable* dialog box.
3. Enter a new value and close the dialog by clicking the *OK* button.
4. The new value should immediately be shown by the *Display* component.
5. Exit the runtime software by invoking the ▶ *Services* ▶ *Exit* command from the main menu.