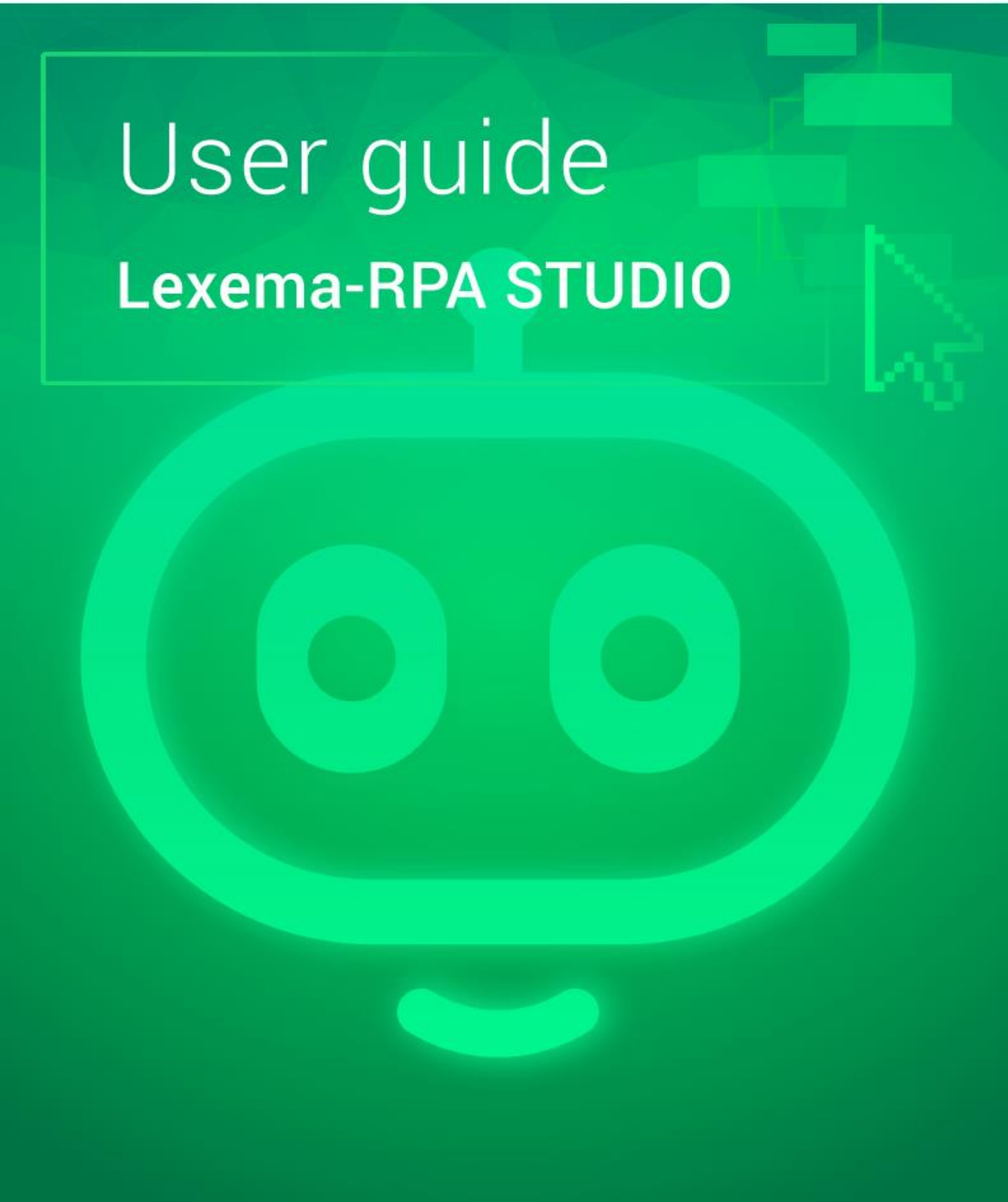
Lexema-RPA

Robotization

of business processes

User guide

Lexema-RPA STUDIO



User guide

Lexema-RPA Studio

CONTENTS

INTRODUCTION

The manual is made for people who want to learn how to make robots in Lexema-RPA Studio software. You will be introduced to all of the instruments and modules of the Studio. Instruments and modules are used to make robots.

Lexema-RPA Studio, hereinafter referred to as Studio, allows users to configure the sequence of interactions with different softwares. The interaction is configured with help of the programming modules, which are a part of the Studio. Every module is a separate number of features, necessary to perform a specific action, and mechanisms for its configuration.

Purpose of software

Lexema-RPA Studio is designed to develop software robots.

What is RPA?

Robotic process automation (or RPA) — is a form of business process automation technology (sometimes referred to as software robotics; not to be confused with robot software) based on metaphorical software robots (bots) or artificial intelligence (AI)/digital workers.

Simplified, ***RPAs*** — are software robots that emulate human actions performed by them on a computer.

What can robots do?

Robots can perform routine operations on a computer instead of humans:

- enter data from the keyboard, use a mouse, read information from a computer screen, receive and process email, interact with office software.
- integrate data from one system to another without the use of additional tools.

How long does it take to create robots?

Robots are made in a simple graphic interface, and the time spent on a robot depends on its complexity. You can make a simple robot within a few hours, for labor-intensive ones it can take a few days.

Areas of use of robots

Back up robots (integrators)

- Transferring information from one system to another;
- Data overflow.

HR-robots

- Search for vacancies by specific criteria.

Operator-robots

- Reading and entering data from scanned copies of documents.

Consolidator robots

- Consolidation of data into a report;
- Bulk sending or printing of documents.

Buying robots

- Monitoring marketplaces;
- Uploading tender information;
- Selection of suppliers based on specific conditions

Robot-salesman

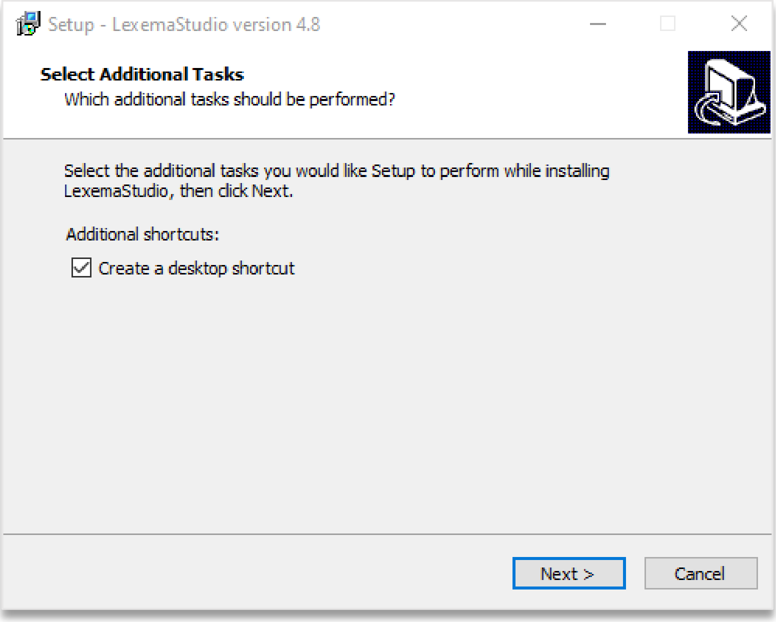
- Counterparty[1] Reliability Check;
- Formation of commercial offers

[1] Counterparty – the buyer or the seller in a financial contract.

Where to begin?

To get started, you need to download Lexema-RPA Studio from the following website *http://lexema.ru/lexemarpastudio/registratsiya/*. Before the download begins you will be offered to make an account. The signup is necessary to log into the Studio. After you sign up, the installer will begin downloading.

When the download is finished, open the installer. Press «Next» in the opened dialog box.



Setup - LexemaStudio version 4.8

Select Additional Tasks

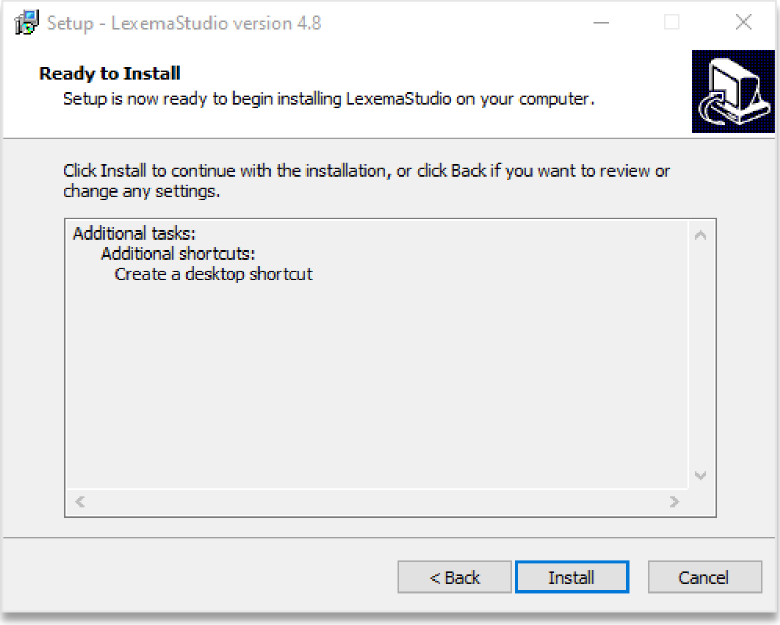
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing LexemaStudio, then click Next.

Additional shortcuts:

☑ Create a desktop shortcut

Then click «Install».



Setup - LexemaStudio version 4.8

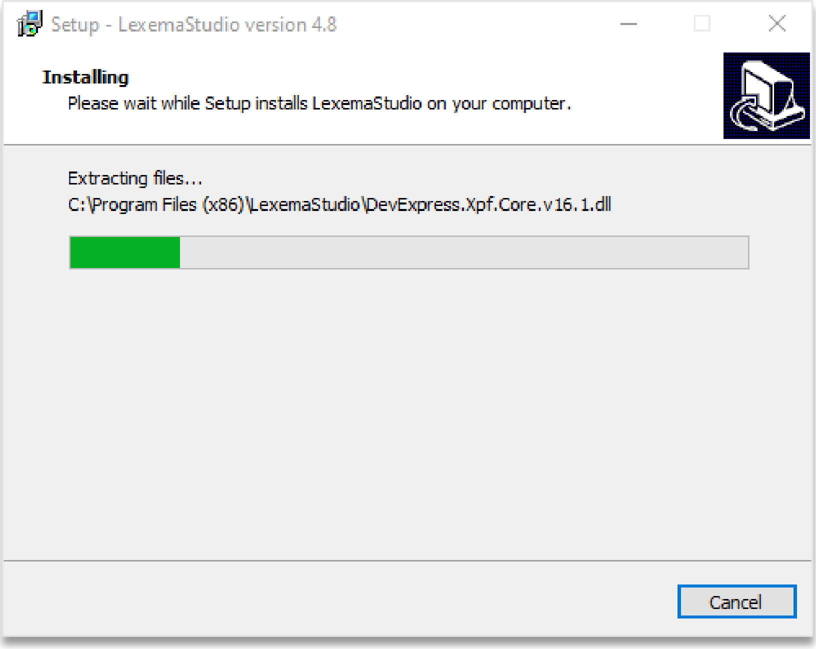
Ready to Install

Setup is now ready to begin installing LexemaStudio on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Additional tasks:
    Additional shortcuts:
      Create a desktop shortcut

After the installation process will begin.



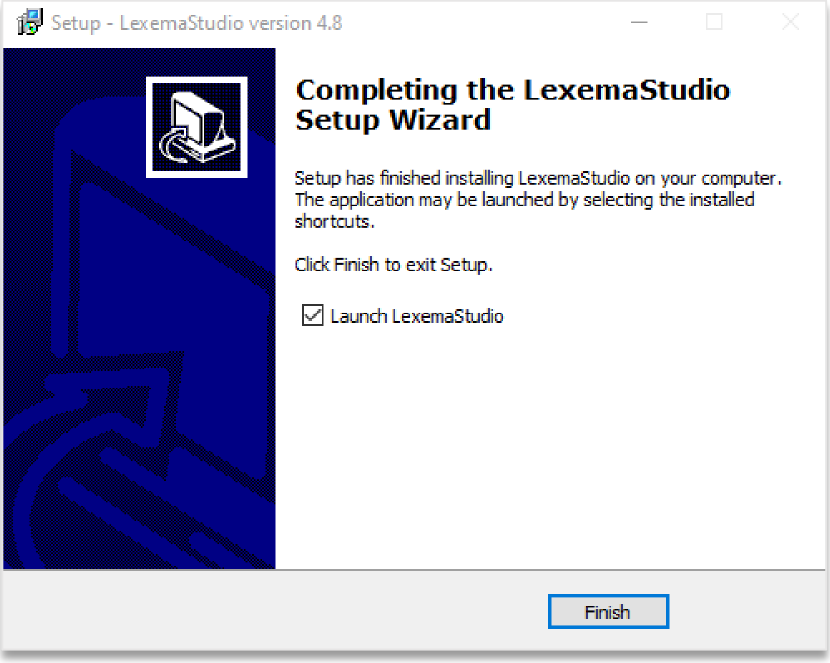
Setup - LexemaStudio version 4.8

Installing

Please wait while Setup installs LexemaStudio on your computer.

Extracting files...
C:\Program Files (x86)\LexemaStudio\DevExpress.Xpf.Core.v16.1.dll

After the studio is installed click «Finish».



Setup - LexemaStudio version 4.8

Completing the LexemaStudio Setup Wizard

Setup has finished installing LexemaStudio on your computer. The application may be launched by selecting the installed shortcuts.

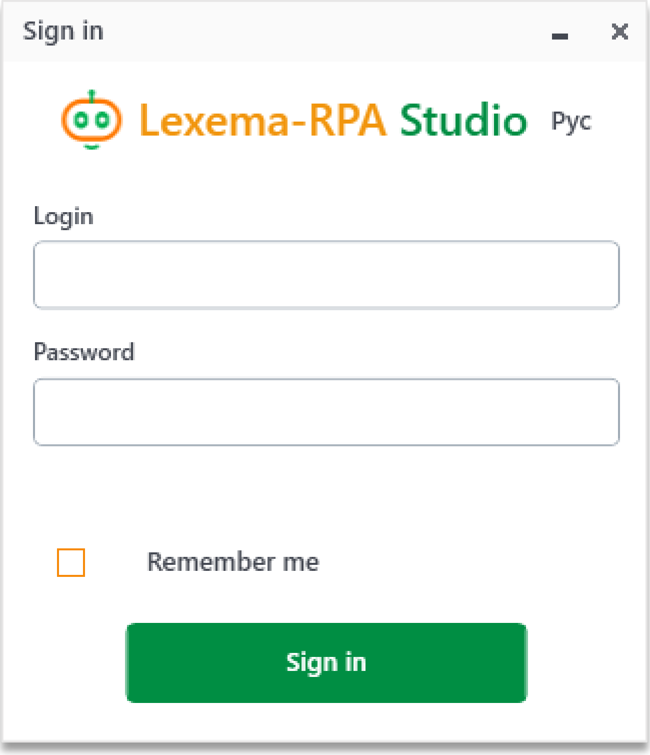
Click Finish to exit Setup.

☑ Launch LexemaStudio

The shortcut will appear on your desktop.

TIP. If you were unable to start or install the Studio, install or update *.NET Framework 4.6.1*. You can download *.NET Framework 4.6.1* from the Windows Download Centre: https://www.microsoft.com/ru-ru/download/details.aspx?id=53344.

After you launch the Studio an authorization window will appear, you will need to enter the login and password specified during the signup process. ***Authorization window appears in the latest versions of the software.***



«Remember me» checkbox will remember your username and password for subsequent logins from the current device.

«Login» button. By clicking it you will log into the Studio.

If you’re not running the latest version of the software, then before authorization you will be offered to download and install an update.

PART I. INTRODUCTION TO STUDIO

Chapter 1. Main window interface

User interface of Lexema-RPA Studio is characterized by the area for working with projects (workspace), located in the center of the main window. Right above the workspace is a bar with management buttons. To the left of the workspace there is a menu with modules for making robots, to the right of the workspace is a solution explorer, and right under the workspace is the debug output.

Main widow

Main window of the studio has multiple sections and main project control buttons (Figure. 1.1.1).

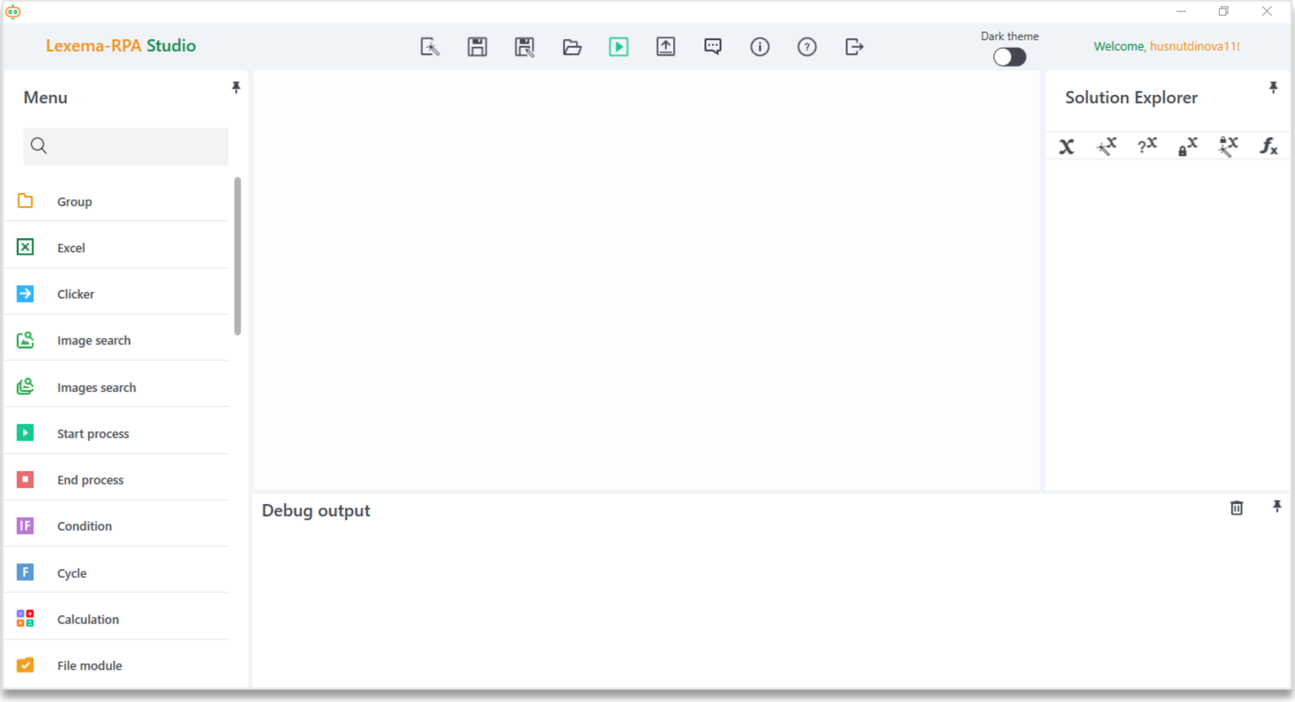


Figure. 1.1.1.Main window of the Studio

Main window interface provides user the ability to customize the width of sections up to minimizing them. You can change the width of sections with your mouse when hovering over the vertical dividers.

All panels (menu, solution explorer, and debug output) can be pinned or opened by hovering above them (Figure. 1.1.2).

To pin or unpin a panel you need to click the button with a drawing pin 📌 in the right corner of the panel.

The features of the main window of the Studio will be described below.

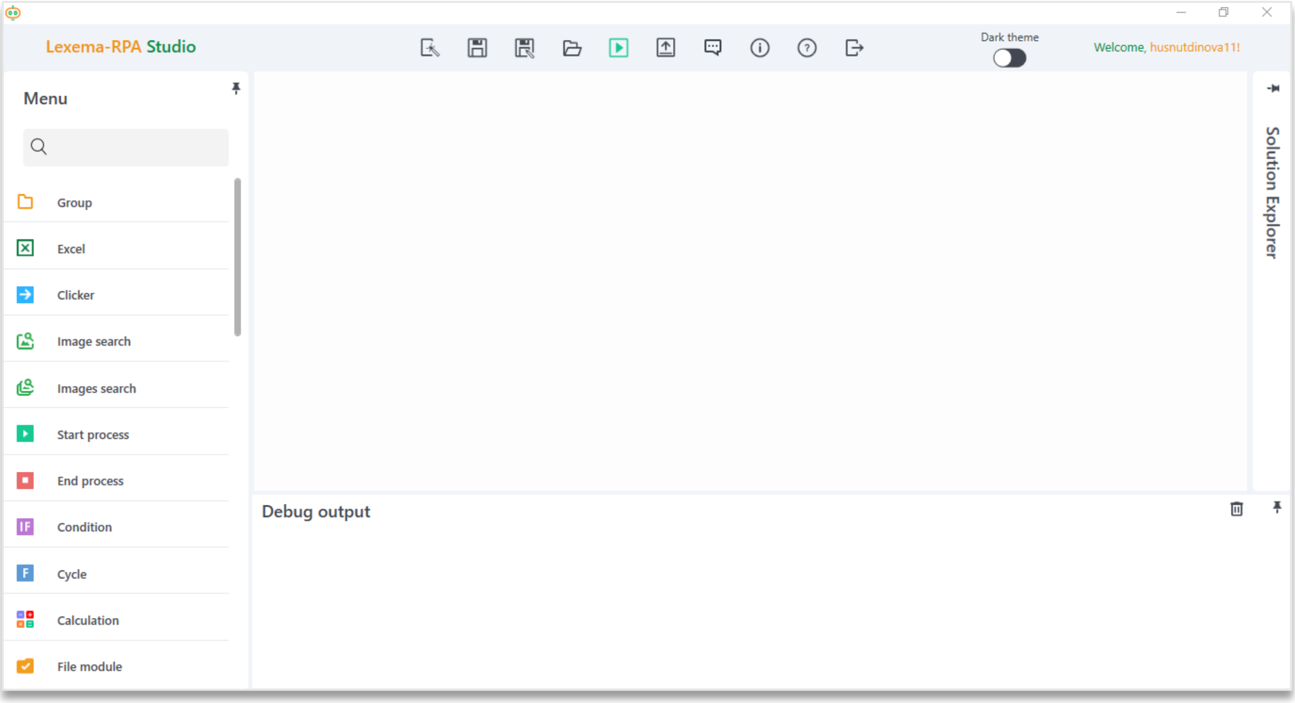


Figure 1.1.2. Hidden panel «Solution explorer»

Management buttons

Project management buttons are located at the top panel of the main window and are designed to perform basic project manipulations: create a new project, save project, open project, start of the robot, and exit the Studio.

Studio also supports hotkeys. For example, to create a project you can press Ctrl+N.

Supported hotkeys are described in more detail in the «hotkeys» section.

Create

Clicking on the “Create” button brings up the create new project window. (Figure. 1.1.3).

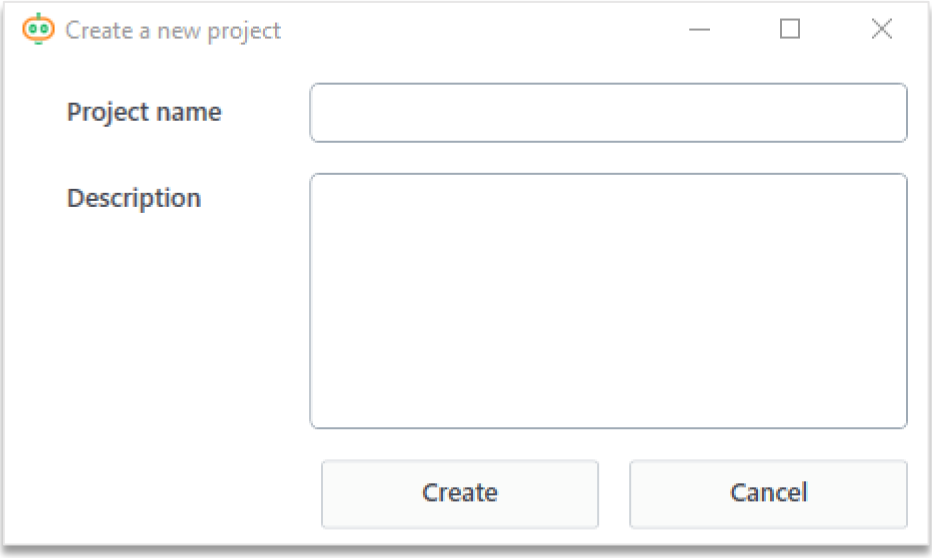


Figure. 1.1.3. Creating a new project

In the create new project window you need to enter the name of a project, and in addition, you can specify its description. Clicking the «Create» button will finish the process of creating a new project. New project will open in the main window of the Studio.

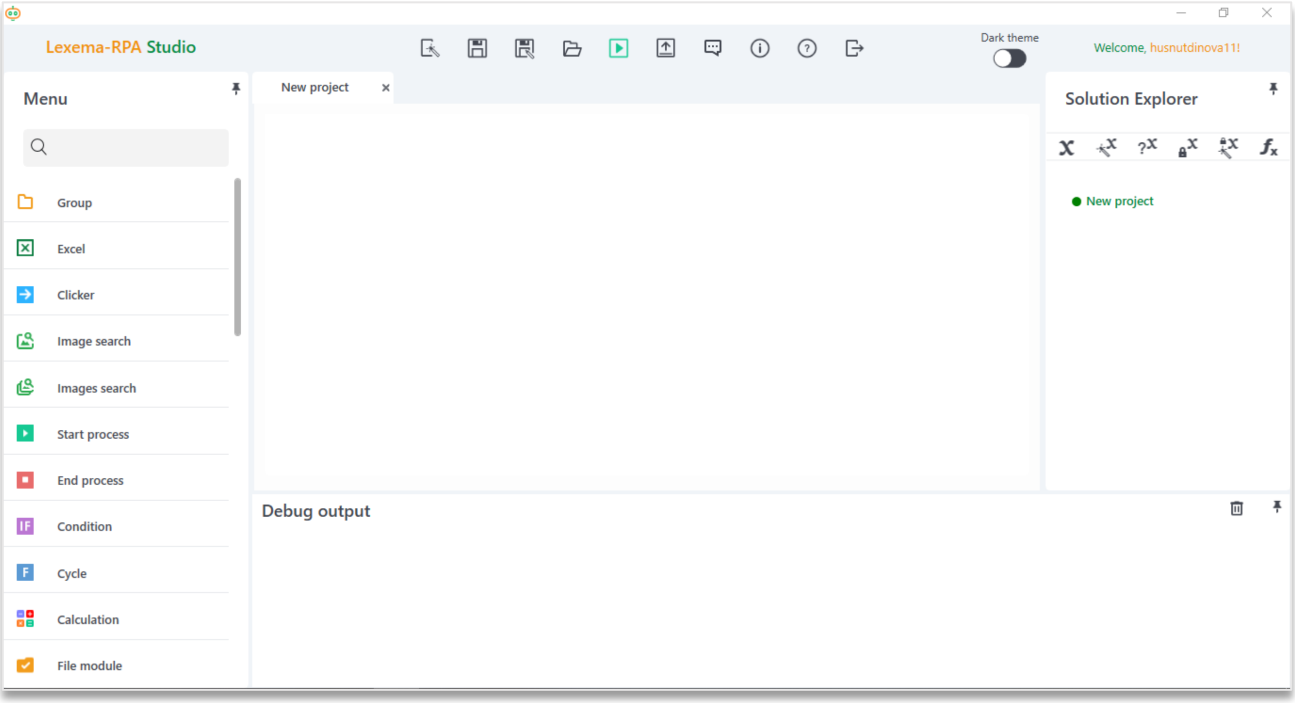


Figure. 1.1.4. New project window

Save project

«Save project» button opens the dialog window of saving a file.

For the new project in the dialog window you will need to choose a folder where you want the project to be saved into. Lexema-RPA Studio project files have «.rpa» format. After choosing the folder you need to press «Save» (Figure. 1.1.5). After that the file will be saved in the folder you selected. An existing project will be saved in its current location.

Save as

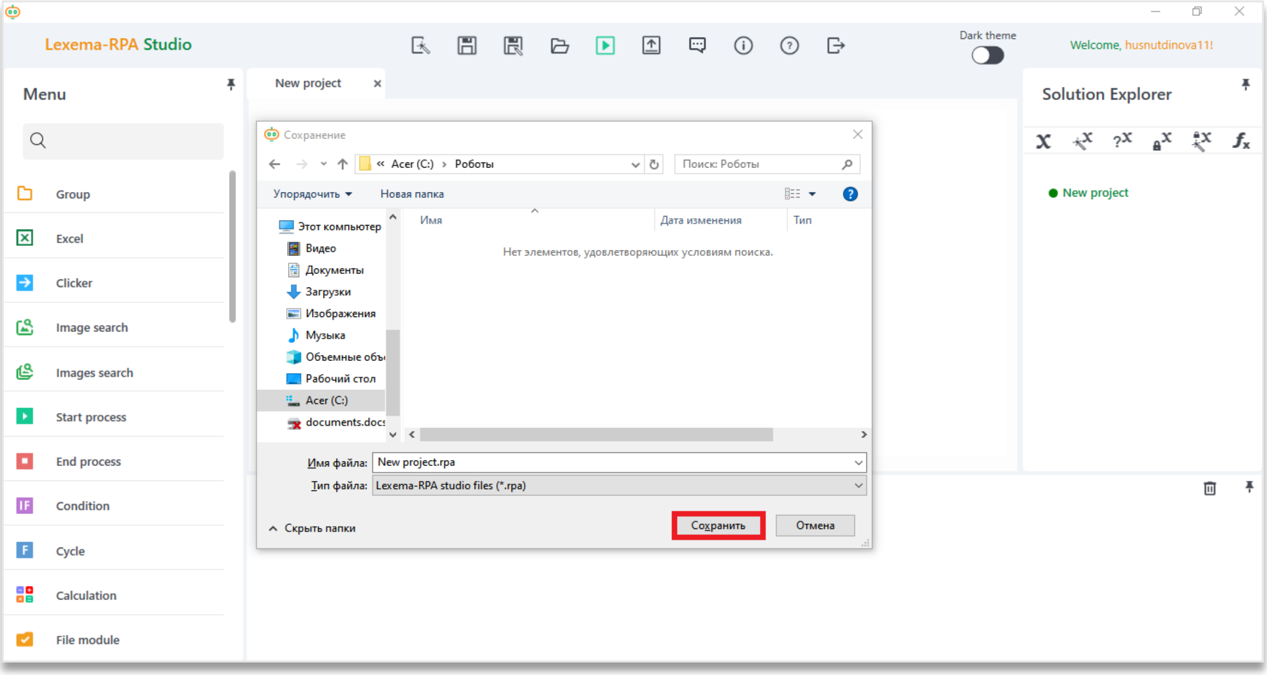
«Save as» button opens up a standard dialog window for saving a file similar to the window for saving a new project.

A dialog box allows you to change the name or location of the project file,

Open project

«Open project» button brings up a standard dialog window of selecting a file.

The chosen file will be opened in the main window of the Studio.



Имя файла: New project.rpa

Тип файла: Lexema-RPA studio files (*.rpa)

Нет элементов, удовлетворяющих условиям поиска.

Figure. 1.1.5. Dialog window of saving a project

Play

«Play» button starts all actions described in the project.

An attempt to launch will be done only if the project has at least one action. Actions are performed in the order described in the project structure..

In case of errors or exceptional situations, they will be displayed in the «Debug output» area (the «Debug output» area will be described further in the guide).

Publish

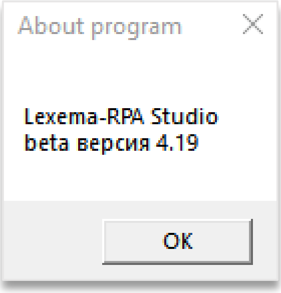
The «Publish» button brings up dialog window of publication of the current project into the robotic orchestrator[2].

Forum

The «Forum» button allows you to go to the website of the forum for technical support - http://talk.lexema.ru/

About

The «About» button brings up a dialog window with information about the current version of the Studio.



About program

Lexema-RPA Studio
beta версия 4.19

Figure. 1.1.6. Information about the current version of the Studio

Exit

The «Exit» button closes the Studio.

Menu

In the «Menu» section a list of available modules is displayed. Each module is responsible for performing a specific action or for interacting with a specific software.

List of available modules

The following modules are available in Lexema-RPA Studio: Excel, Clicker, Image Search (2 variants), Start/End process, Условие, Цикл, Вычисления, Работа с файлами, Код C#, Распознавание текста (обычное и ABBYY), SQL, Отправить/Считать письмо, HTTP GET/POST запросы, Break, JSON, Интерфейс, Закрыть Excel-файл, While.

All modules are shown in the available modules list. (Figure. 1.1.7).

[2] robotic orchestrator — service of scheduling robot launches.

Module search

Module search is done with the help of the text box located above the list of available modules. The list will be filtered as you type a search phrase. (Figure. 1.1.8).

Adding a module to the project

To add a module to the project you need to drag it to the workspace using your cursor. The new module can be positioned in the desired position in the overall sequence of actions. To do this, move the cursor to the module adjacent to the desired position. In this case, the neighboring module will be highlighted with red color, and the position of the new module will be displayed as a bright highlight above or below:above - if a new module will be added before the neighboring one, below - if after it (Figure. 1.1.9 – 1.1.11).

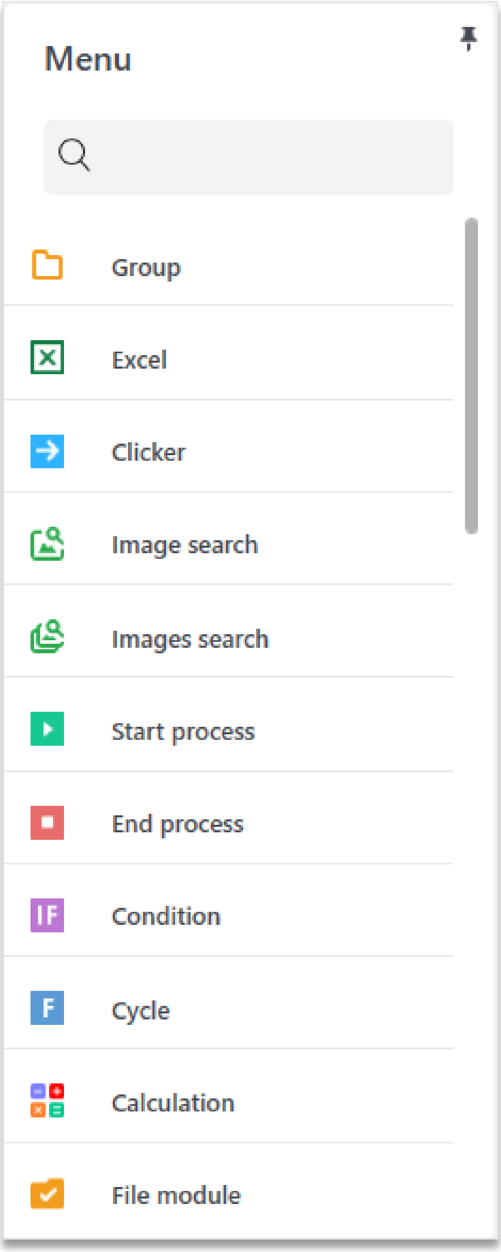


Figure. 1.1.7. Menu of available modules

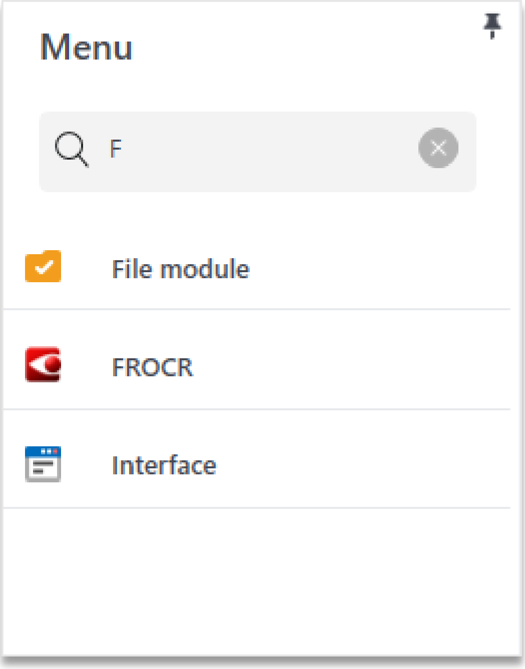


Figure. 1.1.8. Module search

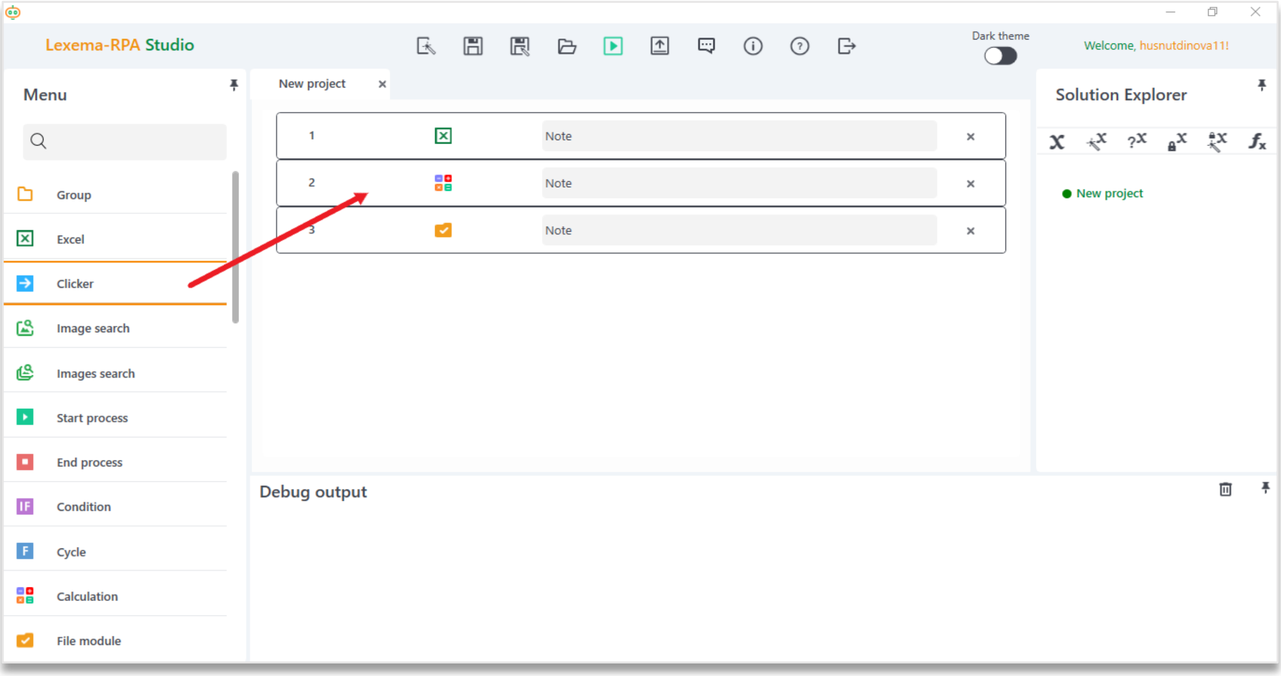


Figure. 1.1.9. Selecting and dragging a module to the desired area

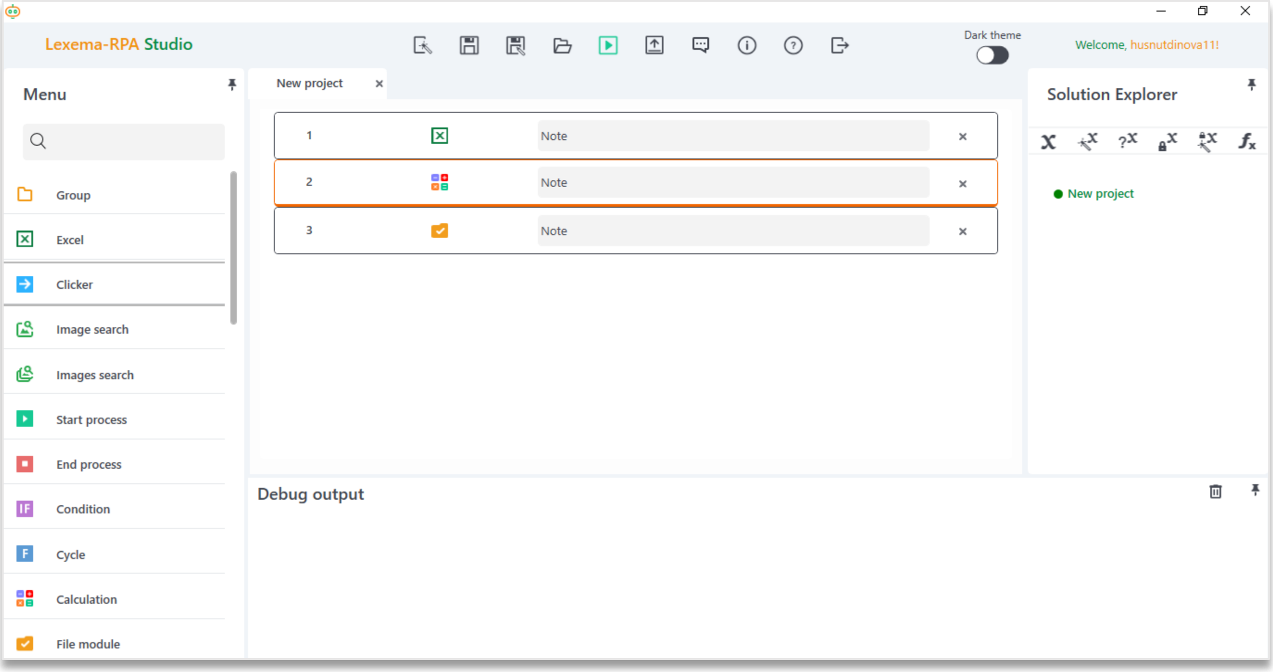


Figure. 1.1.10. Highlight of the second module while moving it

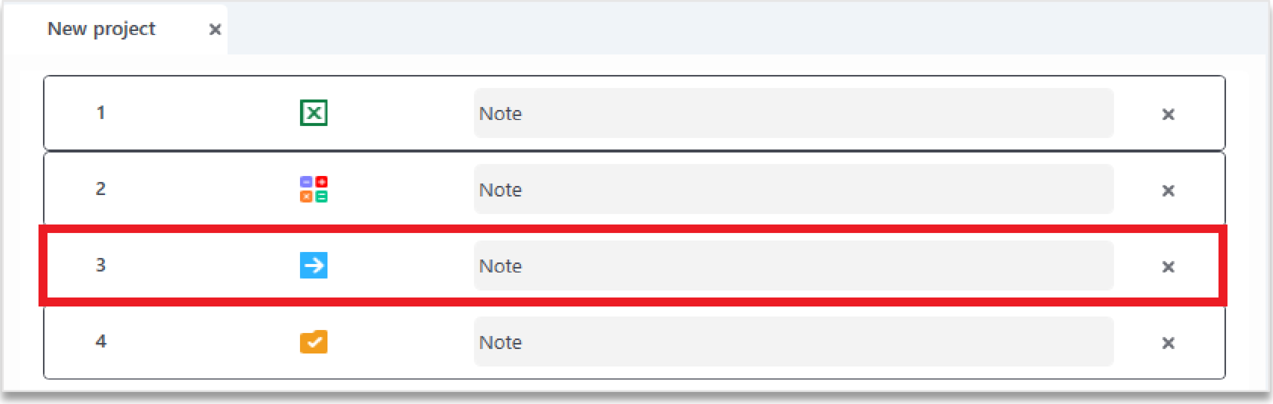
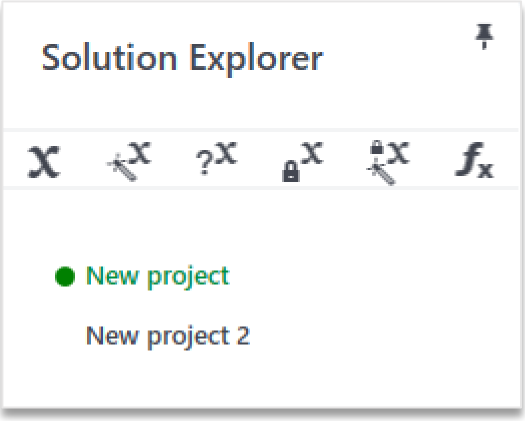


Figure. 1.1.11. The final location of the module

Solution explorer

«Solution explorer» module is located on the right side of the main window. At the top of the section is a control panel for variables and functions. Active project (whose window is now open) is highlighted with green color and has a green dot behind it.



Solution Explorer

● New project

New project 2

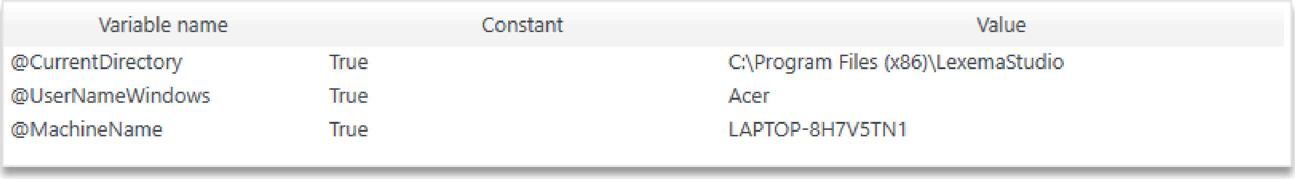
Figure. 1.1.12. Solution explorer

Control panel for variables and functions has the following buttons: «Variables», «Create a new variable», «Find out the value of a variable», «Functions». When you click on one of these buttons, those variables or functions that are related to the active project will be opened or created. A detailed description of the buttons will be given later.

Variable overview

In the Studio workspace, you can display information about the variables available in the active project.

The variable overview window is displayed in the «Variables» tab when you click on the «Variables» button in Solution Explorer.



Variable name	Constant	Value
@CurrentDirectory	True	C:\Program Files (x86)\LexemaStudio
@UserNameWindows	True	Acer
@MachineName	True	LAPTOP-8H7V5TN1

Figure. 1.1.13. Overview of project variables

Window contains a table with a list of all available variables in the project.

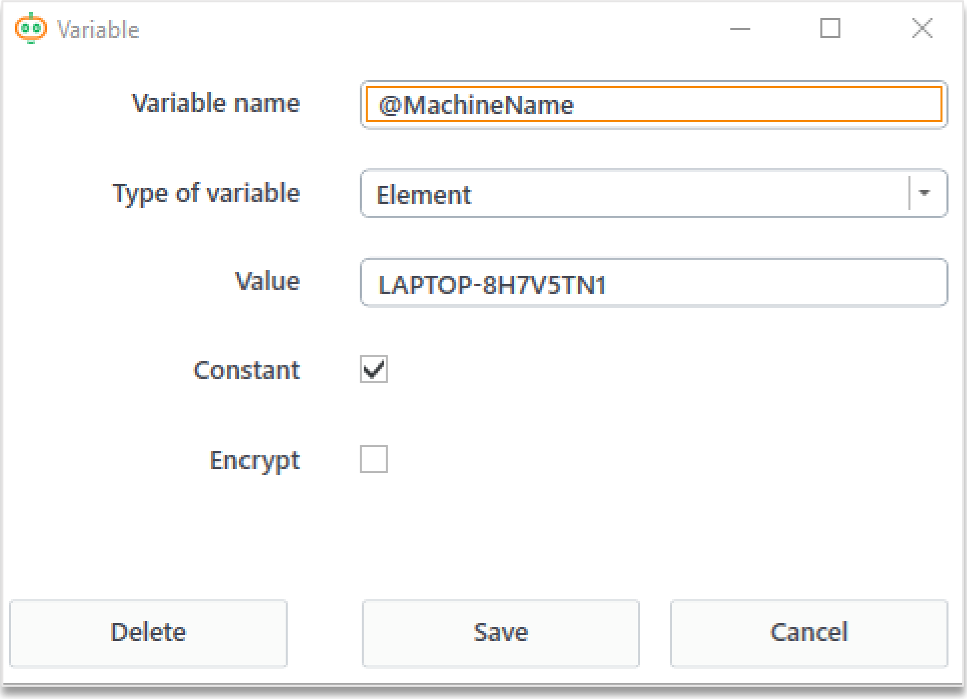
Table has 3 columns

- Name of a variable
- Constant - a flag showing whether it is possible to change the value of a given variable
- Variable value

By default, variables are created automatically when you specify them in a particular module. System variables are created automatically after creating a project.

- @CurrentDirectory – directory, where the studio is installed;
- @UserNameWindows – username in windows;
- @MachineName – computer name.

Double-clicking on a line with a variable opens a dialog box for editing variable properties. In a dialog box you can change properties of a variable or delete the variable from the project. A detailed description of working with variables will be discussed in Chapter 2.



Variable

Variable name: @MachineName

Type of variable: Element

Value: LAPTOP-8H7V5TN1

Constant ☑

Encrypt □

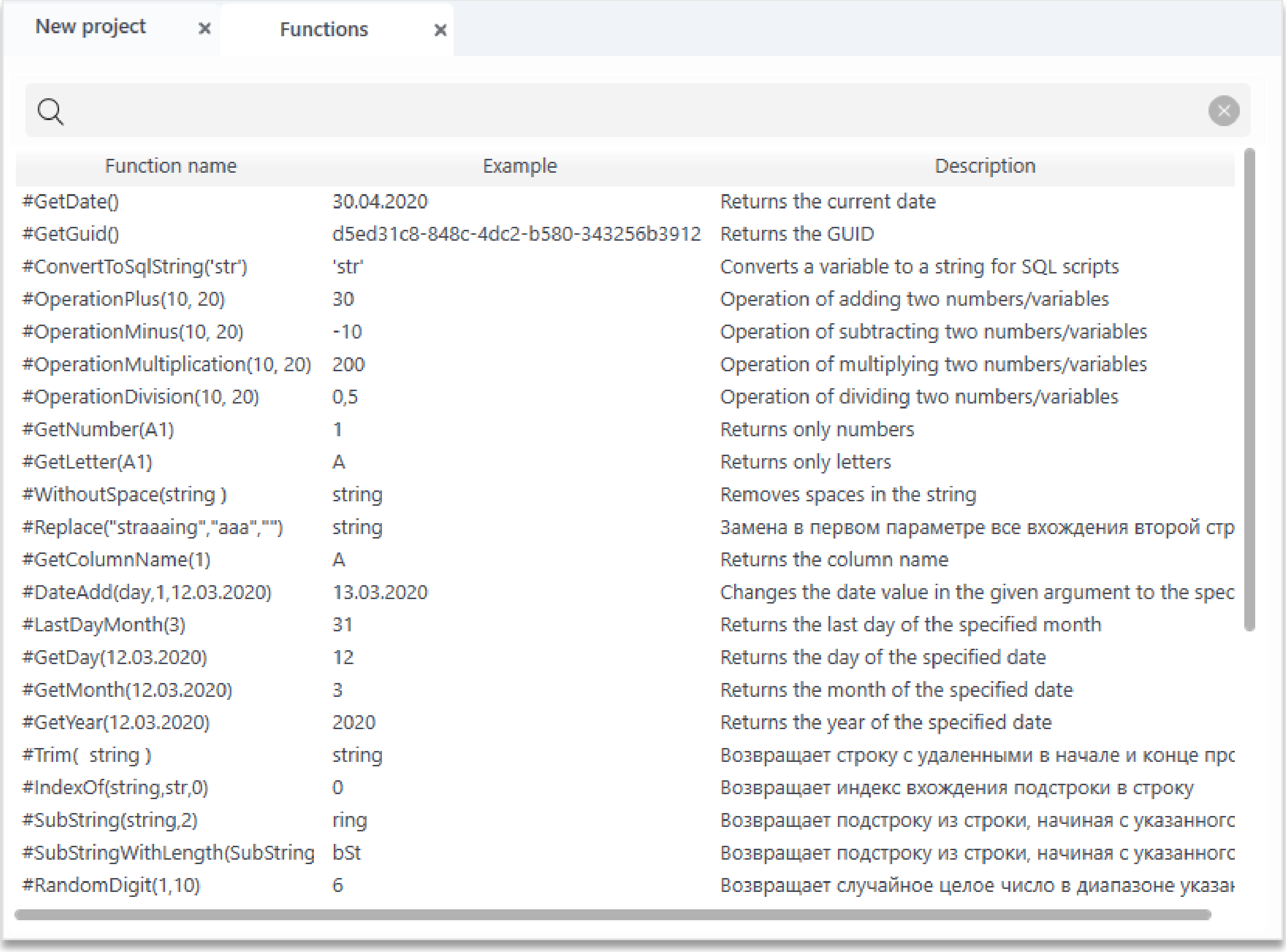
Figure. 1.1.14. Variable settings window

Functions overview

Functions available in a project are displayed in the Studio workspace in «Functions» tab after pressing the fx button in solution explorer.

The list of available functions is presented in a table with the following columns:

- Function name — the name of the function and an example of the arguments it takes;
- Usage example — an example of the result of performing the function of the first column;
- Description — general description of the function.

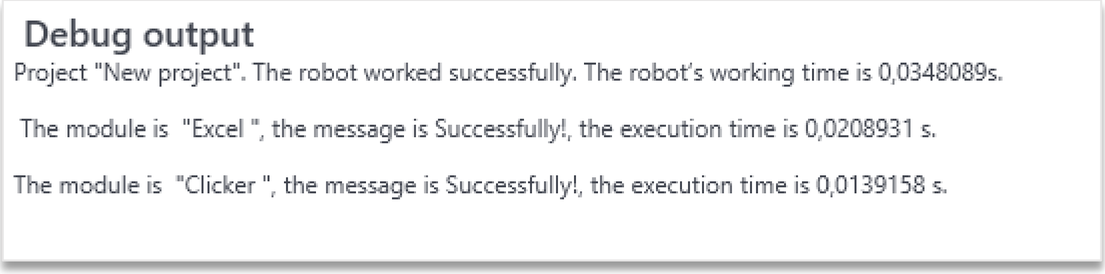


Function name	Example	Description
#GetDate()	30.04.2020	Returns the current date
#GetGuid()	d5ed31c8-848c-4dc2-b580-343256b3912	Returns the GUID
#ConvertToSqlString('str')	'str'	Converts a variable to a string for SQL scripts
#OperationPlus(10, 20)	30	Operation of adding two numbers/variables
#OperationMinus(10, 20)	-10	Operation of subtracting two numbers/variables
#OperationMultiplication(10, 20)	200	Operation of multiplying two numbers/variables
#OperationDivision(10, 20)	0,5	Operation of dividing two numbers/variables
#GetNumber(A1)	1	Returns only numbers
#GetLetter(A1)	A	Returns only letters
#WithoutSpace(string)	string	Removes spaces in the string
#Replace("straaaing","aaa","")	string	Замена в первом параметре все вхождения второй стр
#GetColumnName(1)	A	Returns the column name
#DateAdd(day,1,12.03.2020)	13.03.2020	Changes the date value in the given argument to the spec
#LastDayMonth(3)	31	Returns the last day of the specified month
#GetDay(12.03.2020)	12	Returns the day of the specified date
#GetMonth(12.03.2020)	3	Returns the month of the specified date
#GetYear(12.03.2020)	2020	Returns the year of the specified date
#Trim(string)	string	Возвращает строку с удаленными в начале и конце прс
#IndexOf(string,str,0)	0	Возвращает индекс вхождения подстроки в строку
#SubString(string,2)	ring	Возвращает подстроку из строки, начиная с указанногс
#SubStringWithLength(SubString	bSt	Возвращает подстроку из строки, начиная с указанногс
#RandomDigit(1,10)	6	Возвращает случайное целое число в диапазоне указаı

Figure. 1.1.15. Overview of available functions

Debug output

All messages that appear during the execution of the configured project actions after clicking the "Play" button appear in the «Debug output» section of the main window. Messages are displayed for all actions in order of their execution in the project.



Debug output

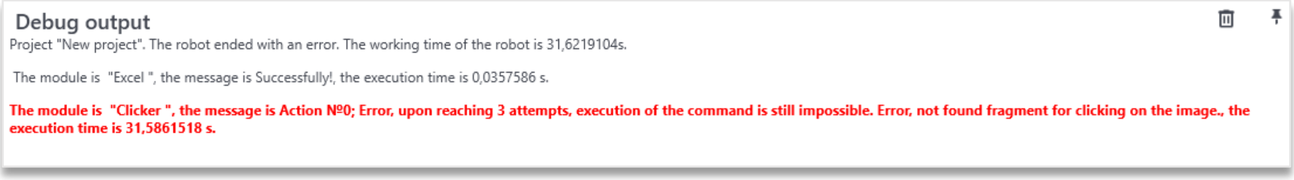
```
Project "New project". The robot worked successfully. The robot's working time is 0,0348089s.

 The module is  "Excel ", the message is Successfully!, the execution time is 0,0208931 s.

The module is  "Clicker ", the message is Successfully!, the execution time is 0,0139158 s.
```

Figure. 1.1.16. An example of successful output

If something goes wrong, the robot will stop working and information about the error will appear in «Debug output». The text of the first and last line of the debug output will be written in redx color.



Debug output

```
Project "New project". The robot ended with an error. The working time of the robot is 31,6219104s.

 The module is  "Excel ", the message is Successfully!, the execution time is 0,0357586 s.

The module is  "Clicker ", the message is Action №0; Error, upon reaching 3 attempts, execution of the command is still impossible. Error, not found fragment for clicking on the image., the execution time is 31,5861518 s.
```

Figure. 1.1.17. Example of unsuccessful output.

Chapter 2. Basics

Now that you know the interface of the Studio, it’s time to learn how to work with the software. In this chapter you will learn about variables and functions, and little tricks (hotkeys), which you can use to speed up your work.

Variables

Variables are used to store data that should be available for all project activities. Most of the modules support reading and entering the data into the variables.

The following requirement is imposed on the name of the variables: ***the name of a variable must start with an @ symbol.***

You can access variables in the fields of the application module settings windows. When accessing a variable it is important to put in the full name of a variable as it was described above.

You can make 3 types of variables in a project:

- Element - a variable that has only one value. This might be a string or a number;
- List - a variable that consists of multiple elements;
- Table – a variable in the form of a table, whose cells contain elements

To create a variable click «create variable» in the solution explorer, after selecting the desired project (making it active). After doing that variable creation window will open, this window is similar to the variable properties editing window.

In the variable creation window enter the name of a variable according to the requirements of application (***the name of a variable must start with an @ symbol***) and pick the type of a variable from the list. Changing the type of a variable changes the available settings.

The following settings are available for variables of type "Element":

- field for entering a value
- sign «Constant» (for constants it is necessary to indicate a value);
- variable encryption.

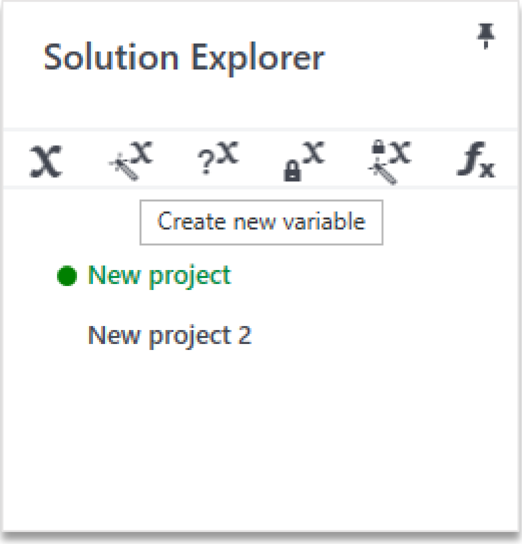


Figure. 1.2.1. «Create variable» button

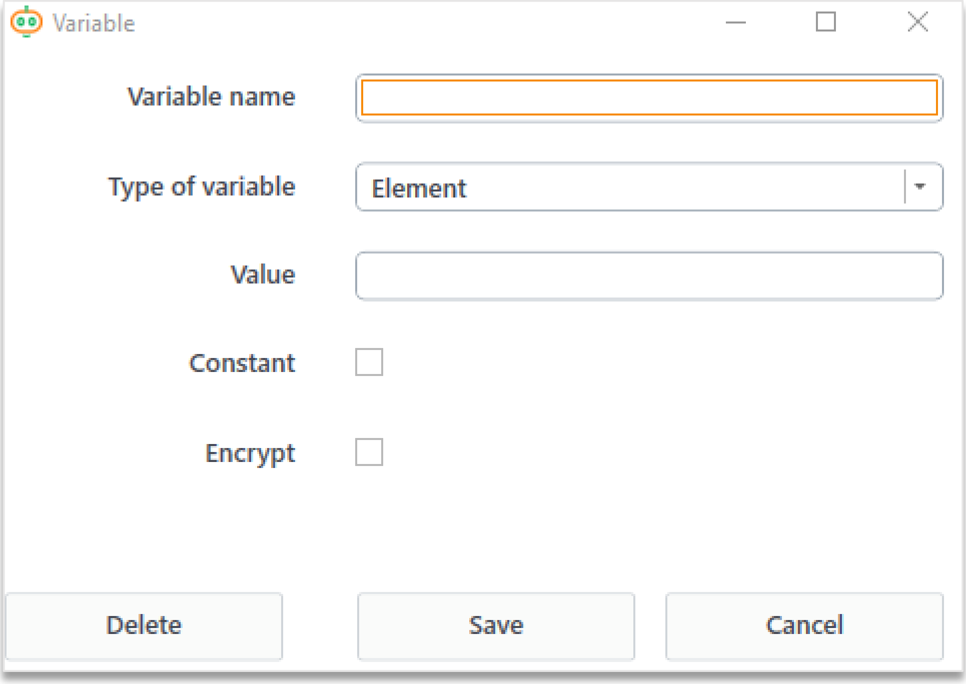
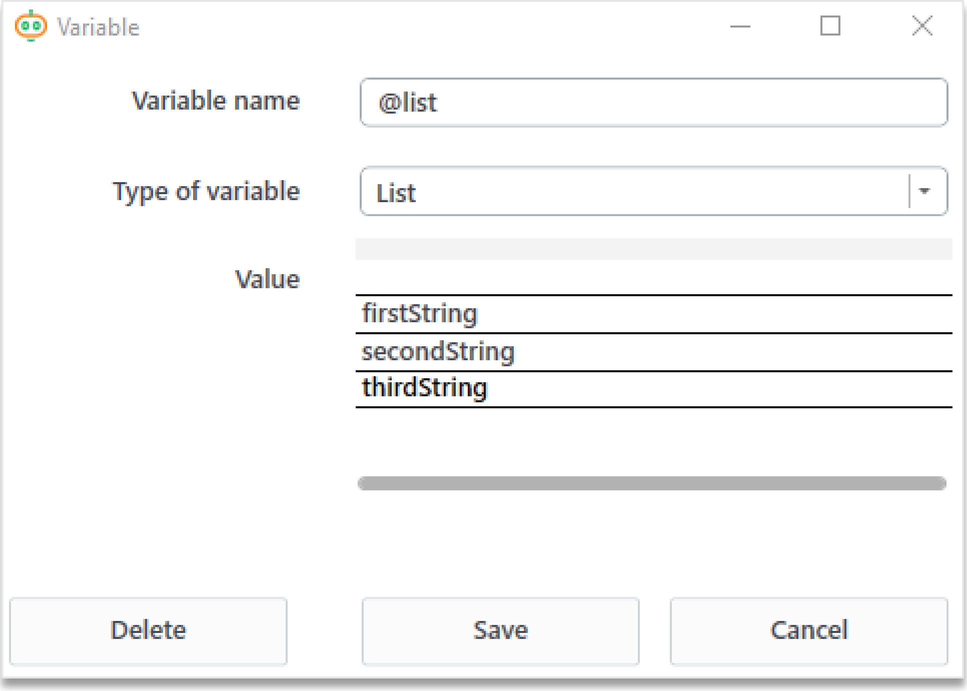


Figure. 1.2.2. Variable creation window

Variables using the type «List» store a set of variables. Each element of the list is referenced by its index, the index starts at 0. The index is indicated in square brackets right next to the name of the variable. For example, to refer to the element - @array[1]



Variable name: @list

Type of variable: List

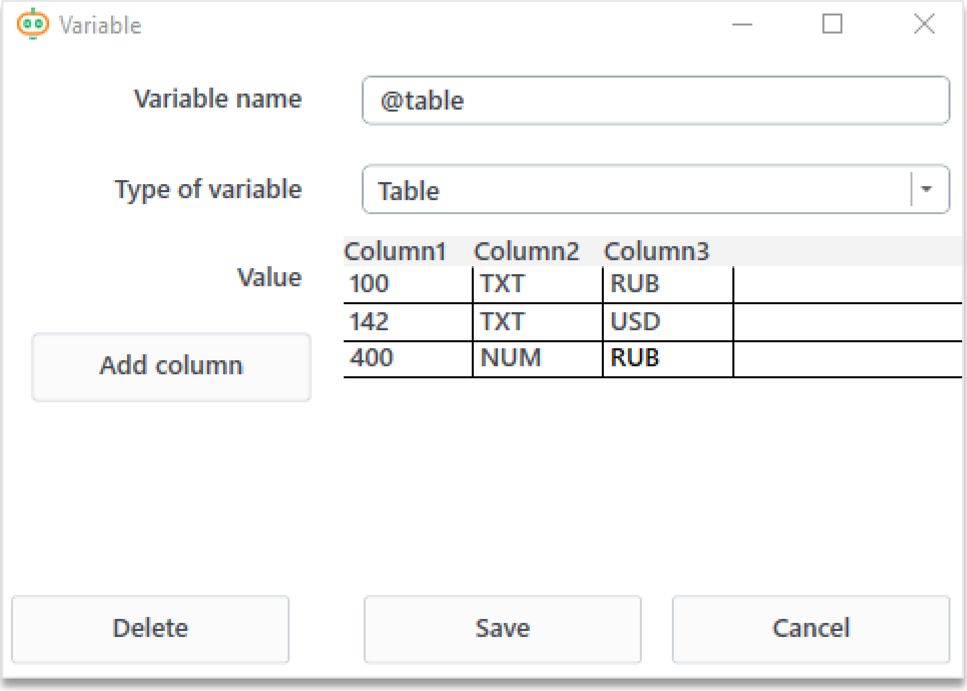
Value:

firstString
secondString
thirdString

Figure. 1.2.3. Creating a list-variable

To create list variables, a table becomes available in which you can fill in the required values. Values are entered in the cells. To add the next element press «Enter».

Variables of type «Table» are a table ((two-dimensional array), i.e. this is a list whose elements are lists. In other words, it is a set of the same type of data, having a common name and access to the elements of which is carried out by two indices. It is convenient to visualize a two-dimensional array in the form of a table in which n rows and m columns, and by a cell of the table standing in the i-th row and j-th column we mean some element of the array a [i-1, j-1] (since index starts from zero). This means that to refer to an element in the 3rd row and the 5th column, which is responsible for the table cell, two indices must be specified: @table [2, 4].



Variable

Variable name: @table

Type of variable: Table

Value

Column1	Column2	Column3
100	TXT	RUB
142	TXT	USD
400	NUM	RUB

FIgure. 1.2.4. Making a list-variable

To fill in the values of variables with the type «Table», a table becomes available, the set of columns of which can be expanded using the «Add Column» button. Adding rows is done by pressing «Enter». You can add value to every cell. ***Unfilled columns won't be used for creating a variable.***

By pressing the «Find out the value of a variable» button in the solution explorer a window opens, which helps you check the value of a variable of an active project. (Figure. 1.2.5 – 1.2.6). ***To check the value you need to enter the full name of a variable (including @).***

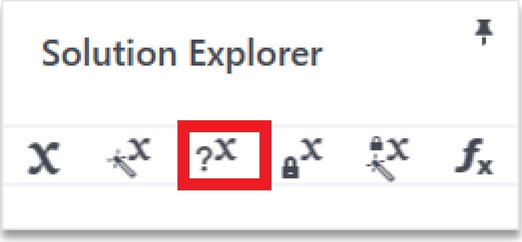


Figure. 1.2.5. «Find out the value of a variable» button

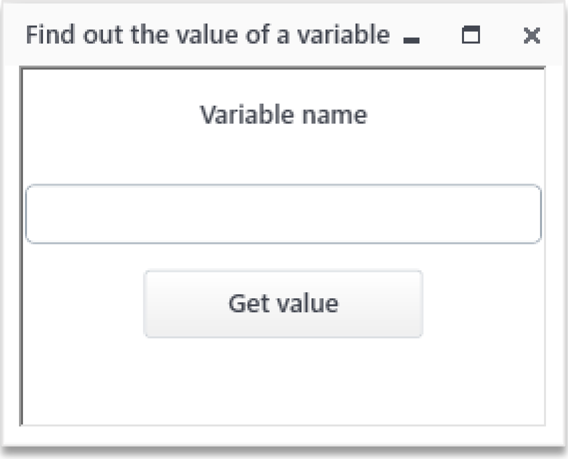


Figure. 1.2.6. Check the value of variable window

Functions

Pre-installed set of functions helps you with various calculations in the actions of the project.

To access a pre-installed function you need to put a # sign before its name, for example, #OperationPlus(1, 1). The arguments of the function on which an operation is performed should be separated by a comma (if a function is working with multiple arguments). You don't need to put a space between the name of a function and the brackets with its arguments.

Workspace

Workspace is the general section of the main window. Here you can configure the project structure and the actions performed in it.

Project structure

Actions are added with the available modules menu. After adding a module to the project structure it will show up as an element indicating the action being performed.

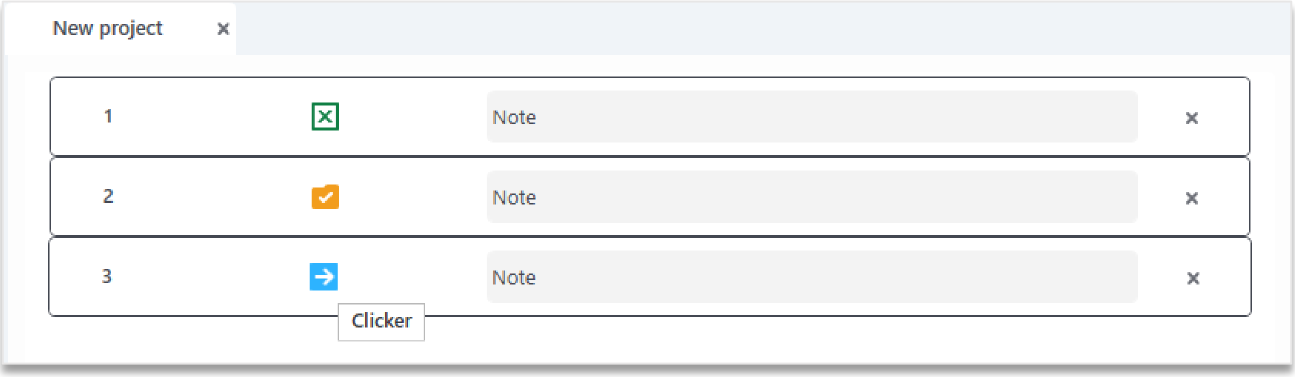


Figure. 1.2.7. Project structure

Action element has several components:

- serial number of an action;
- the name of the action;
- user configured note ;
- the button of deleting an action.

If you right click on an action in project structure you will see a context menu with following items:

- turn on/off;
- remove/set a breakpoint;
- copy;
- paste.

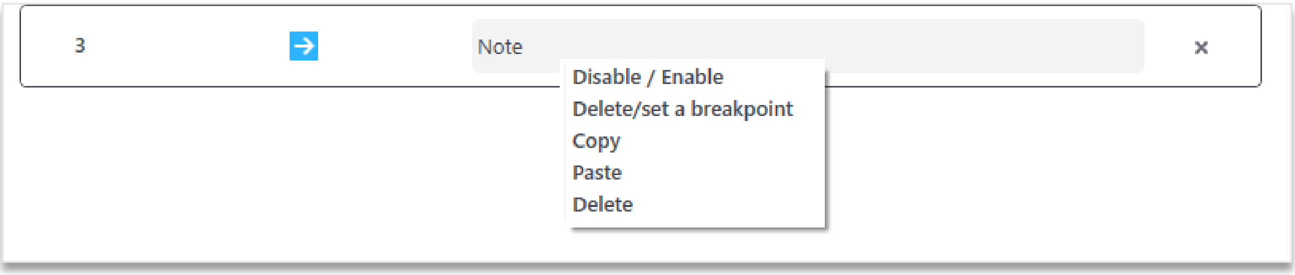


Figure. 1.2.8. Project structure context menu

Item «Turn on/off» allows you to turn on/off an action. A turned off action will be skipped when the project starts.

Item «Remove/set a breakpoint» removes or sets a breakpoint. Breakpoints simplify the process of setting up a sequence of actions.

Item «Copy» copies an action to the clipboard.

Item «Paste» inserts the copied action **after the selected action**. This item is also accessible by opening the context menu in Studio’s main window.

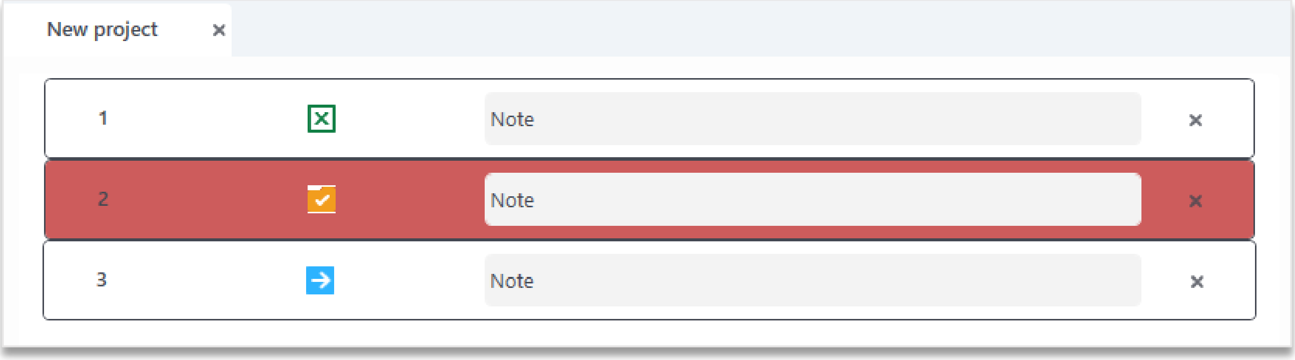
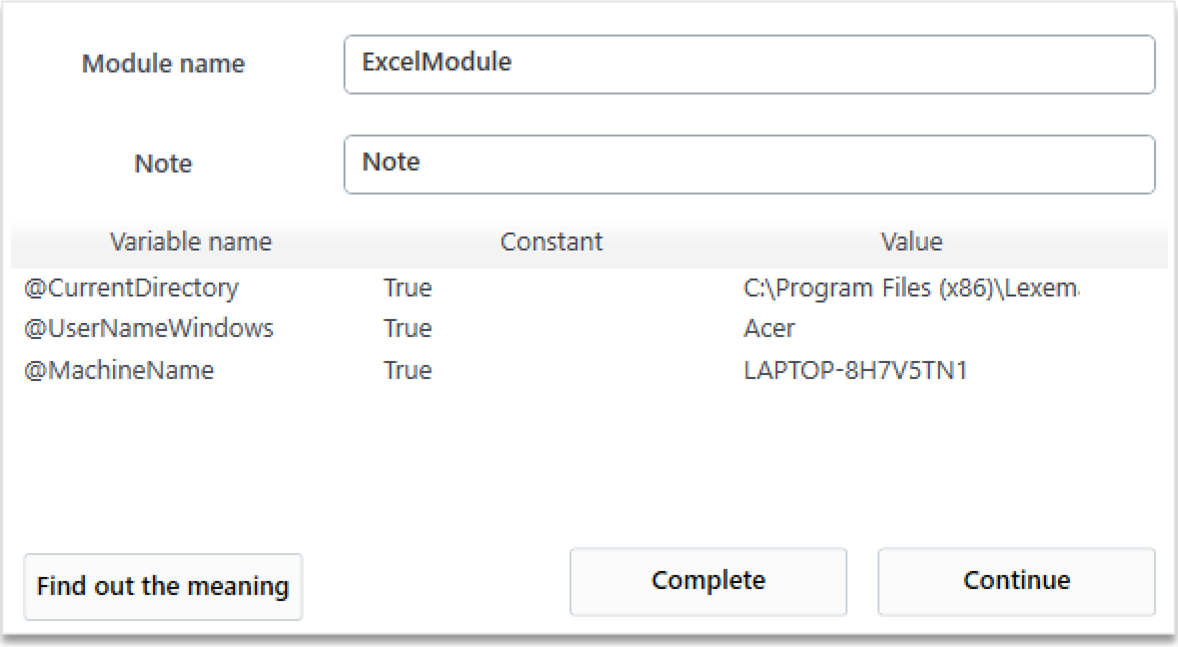


Figure. 1.2.9. Project with a turned off action and a breakpoint

Breakpoint

Breakpoint is a tool that helps debug while writing a robot. Such a tool allows you to stop the robot after executing the selected module, while a window appears indicating all the variables available at the moment of execution, to familiarize themselves with their contents.



Module name: ExcelModule

Note: Note

Variable name	Constant	Value
@CurrentDirectory	True	C:\Program Files (x86)\Lexem
@UserNameWindows	True	Acer
@MachineName	True	LAPTOP-8H7V5TN1

Figure. 1.2.10. Window with a list of variables available at the time of the robot execution

You can’t edit in that window. To determine the value of complex variables (переменные типов «Table» и «List» или объекты) you can use the «Find out the value» button.

After familiarizing yourself with the variables, you can continue executing the robot, or you can finish the process, in any case this window will be closed.

Configuration of added

Configuration window for added in project structure actions is opened by double-clicking with the left mouse button on a specified action. The set of properties available in this window depends on the selected action. Details on the available properties of each action will be described later in the guide.

Hotkeys

Studio supports following hotkeys:

Hotkey	Action
Ctrl+N	Create new project. Opens dialog window for creating a new project.
Ctrl+S	Save project. Saves project in the current directory.
Ctrl+Shift+S	Save project as. Similar to the «Save as» button, opens a dialog window for choosing a project's directory and/or name.
Ctrl+O	Open project. Opens a previously saved project using the system file selection dialog window.
Esc	Close the application.
F5	Run the robot in the current project.
F1	Get help at http://wikie.lexema.ru/.

PART 2. «EXCEL» MODULE.

Chapter 3. Module highlights

«Excel» module is the biggest module in the application. This module is made for processing Excel files in the background, not visible for the user. The module will sequentially run the list of commands given to it.

Module interface

The module window consists of several sections: «Add command», «General settings», «Command list» and a file preview window.

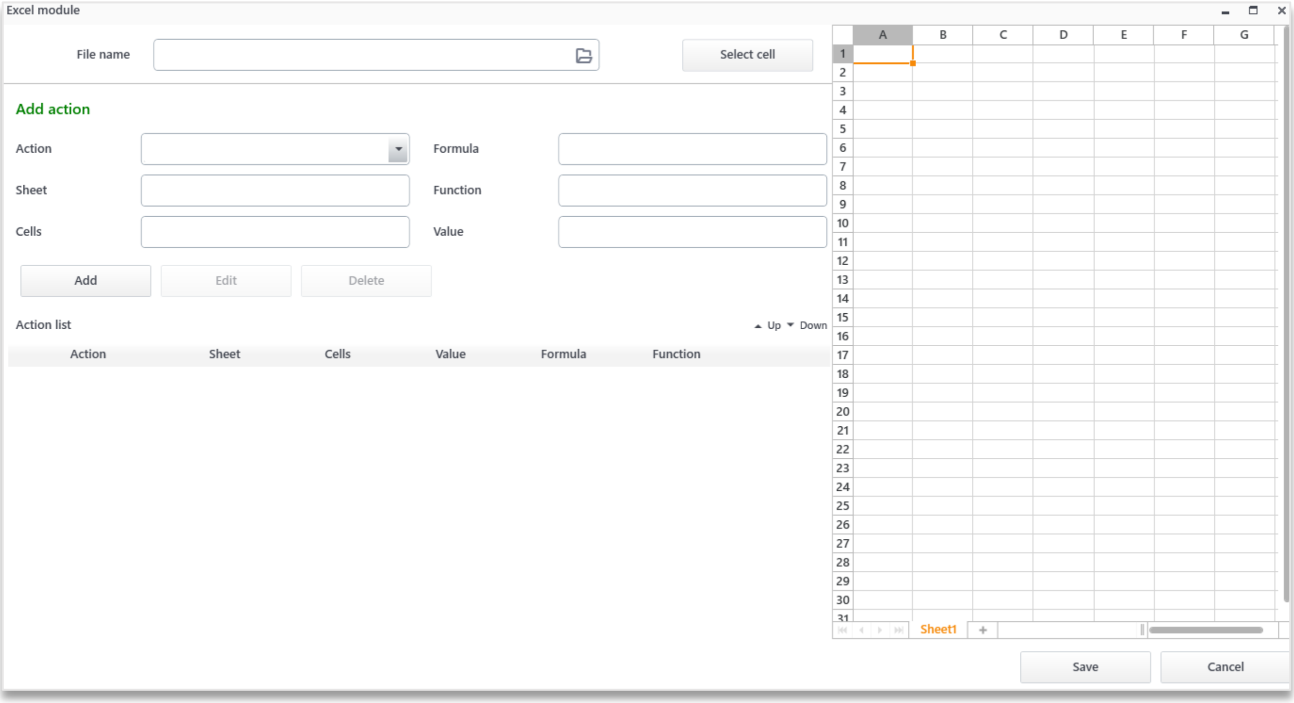


Figure. 2.3.1. Excel module window

The preview window allows you to view all sheets of the downloaded file using tabs with sheet names and scroll bar. It is also possible to enter your values into the cells, apply different formatting, but these changes will not be saved, that is, this feature is available only for «fitting».

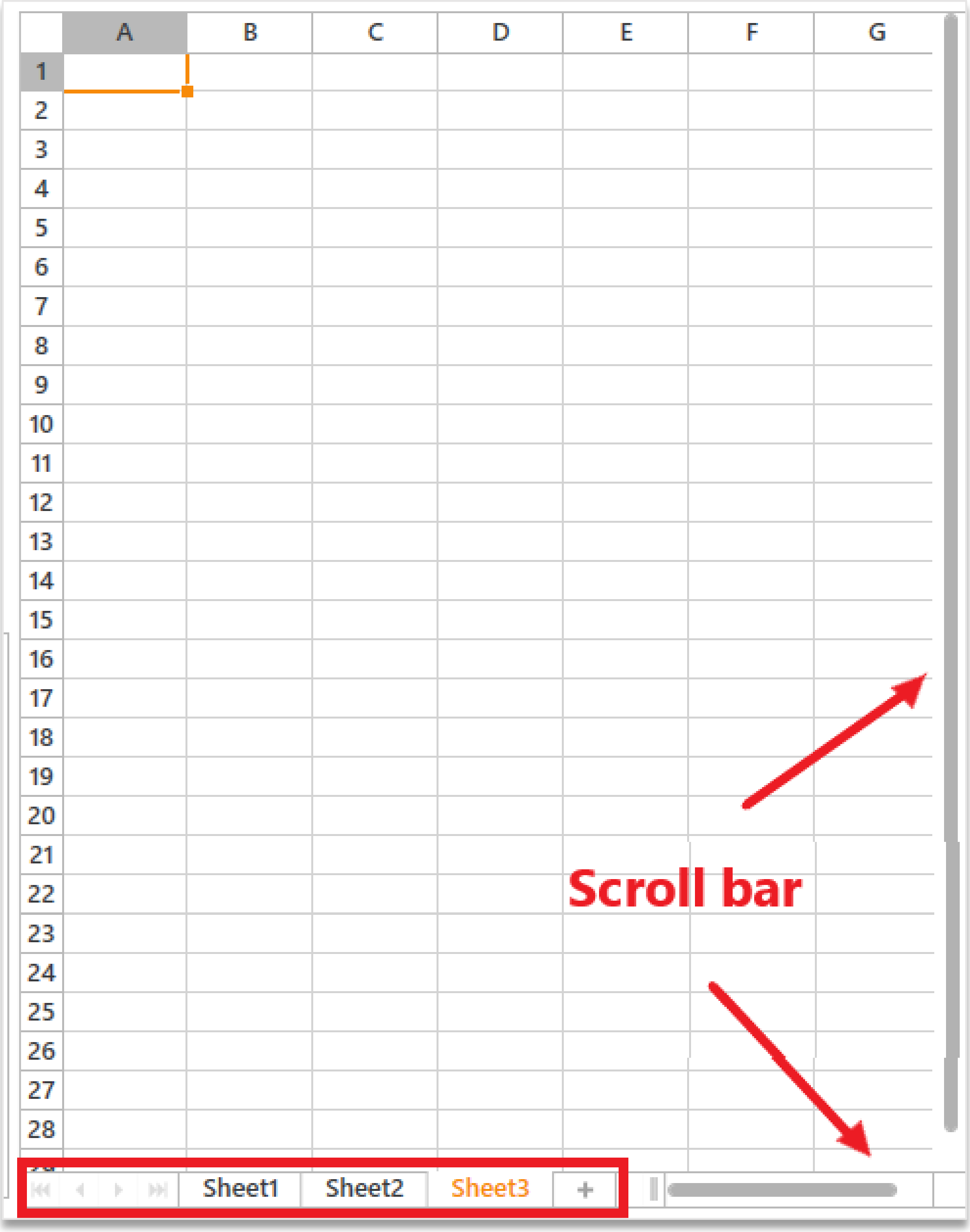


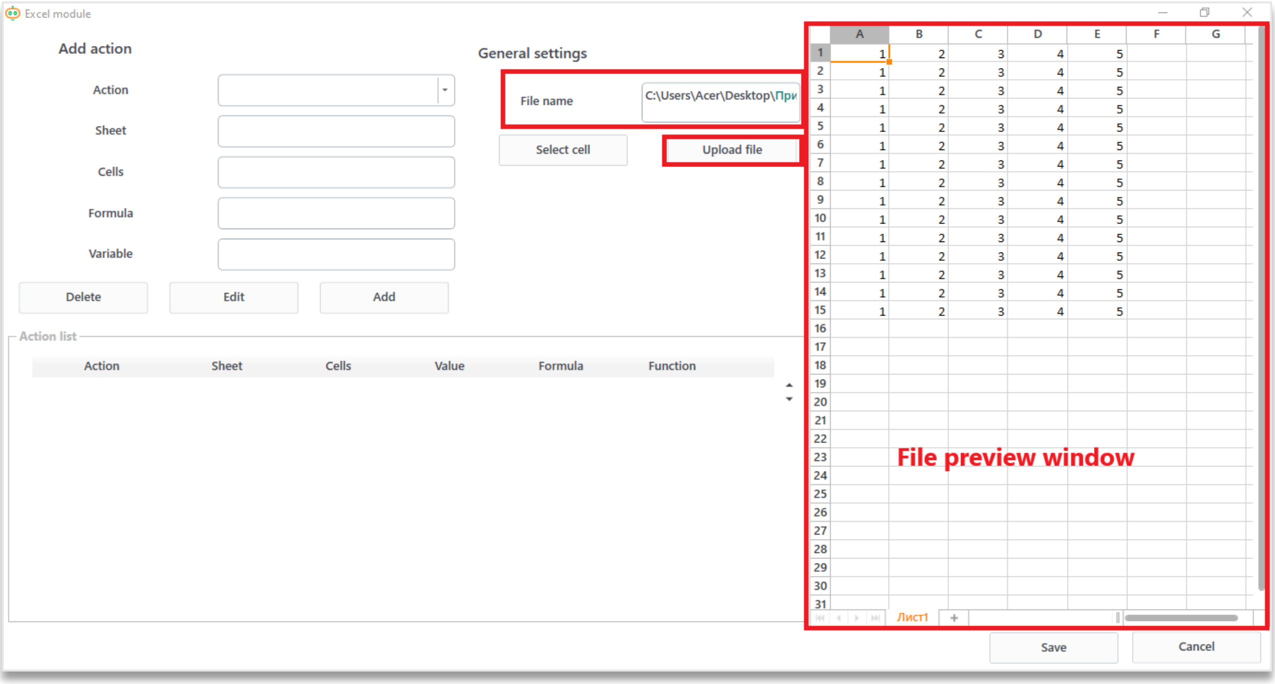
Figure. 2.3.2. Elements of the preview window

«General settings» section

Upload file

«General settings» section consists of the «File name» field and «Upload file» and «Select cell» buttons.

We recommend starting work with the Excel module by selecting a work file. After that, After that, as the file is selected by the “Download file” button, you need to wait a few seconds and the specified file will be displayed in the preview window. (Figure. 2.3.3).



File name: C:\Users\Acer\Desktop\При

Action	Sheet	Cells	Value	Formula	Function

	A	B	C	D	E	F	G
1	1	2	3	4	5		
2	1	2	3	4	5		
3	1	2	3	4	5		
4	1	2	3	4	5		
5	1	2	3	4	5		
6	1	2	3	4	5		
7	1	2	3	4	5		
8	1	2	3	4	5		
9	1	2	3	4	5		
10	1	2	3	4	5		
11	1	2	3	4	5		
12	1	2	3	4	5		
13	1	2	3	4	5		
14	1	2	3	4	5		
15	1	2	3	4	5		
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							

Figure. 2.3.3. General settings

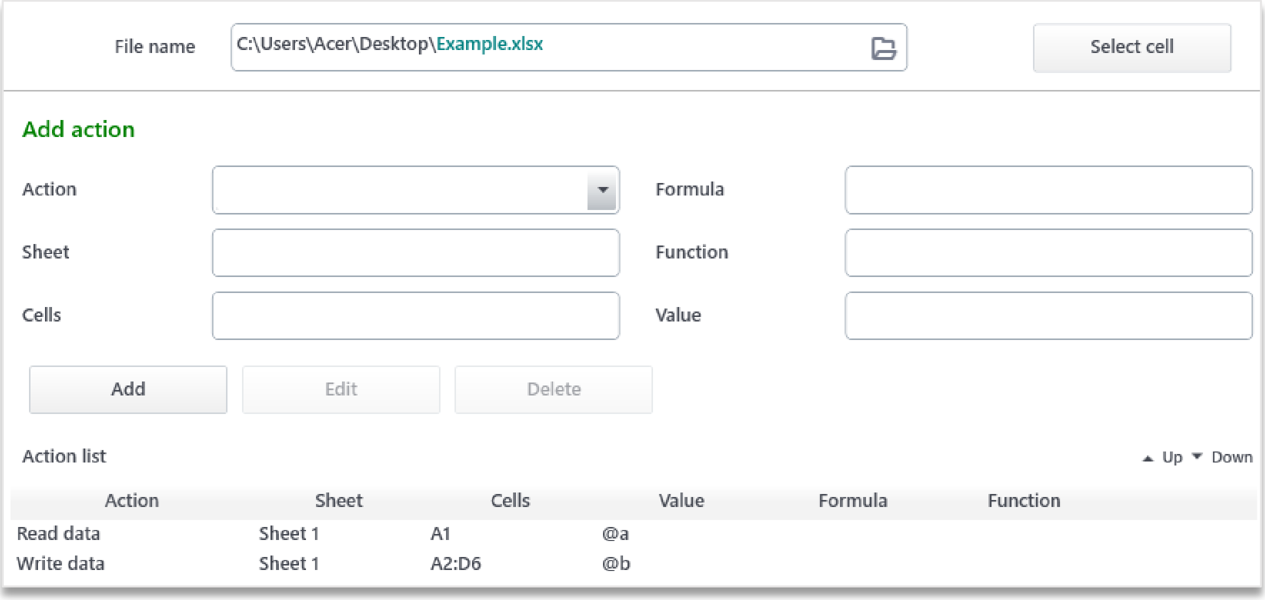
«Add command» and «Command list» section

«Add command» section is the main section of the module. It creates actions that mimic the work of the user, for example, such as reading a certain range of cells, then transfer it to another sheet, delete duplicates and build a pie chart. It this case 4 actions were described. These 4 actions must be sequentially selected and added to the «Command list» section - a table showing all the actions that will be performed by this module. The number of fields in this section is not always the same, it depends on a selected action in the first field.

«Add», «Edit» and «Delete» buttons are designed to work with the «Add command» and «Command list» sections.

Adding a command

To add a command click the «Add» button. It puts the created action into the “Command list” table.



File name C:\Users\Acer\Desktop\Example.xlsx

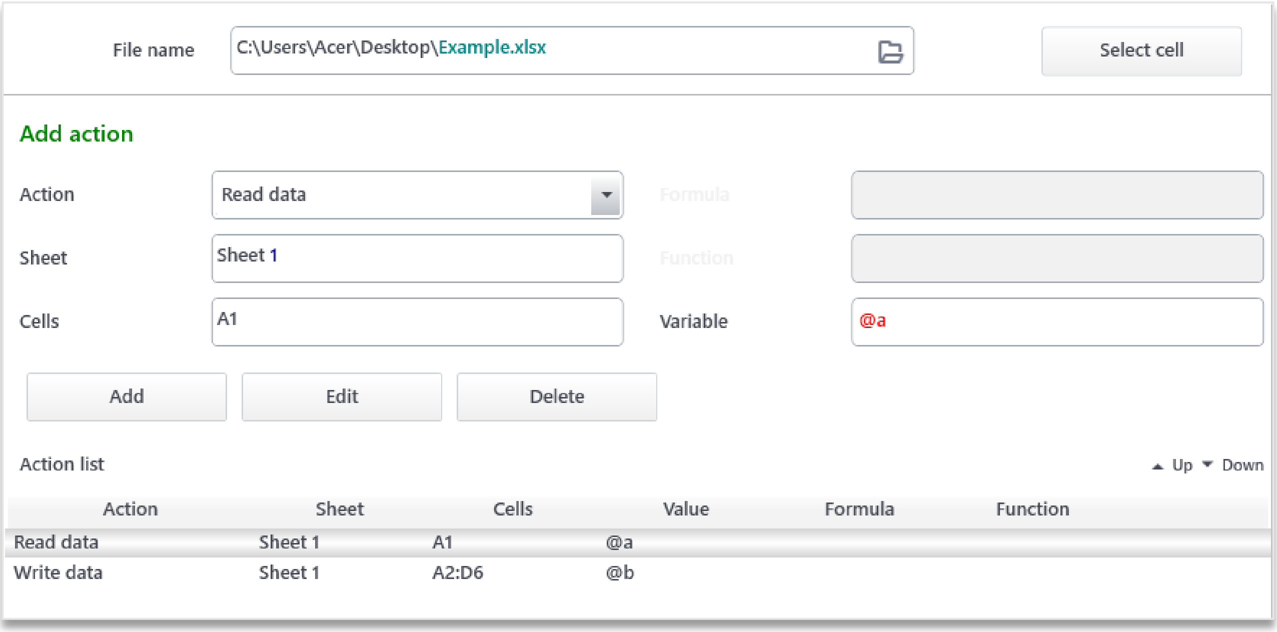
Action list

Action	Sheet	Cells	Value	Formula	Function
Read data	Sheet 1	A1	@a		
Write data	Sheet 1	A2:D6	@b		

Figure. 2.3.3. Adding a command

When choosing a row from the list of commands, information about the selected action will show up in the «Add command» section.

In addition to the specified functionality, the “Add” button is used to copy existing actions from the list of commands. To do that you need to select a row from the list of commands and click the «Add» button. Thus, if you need to perform an action several times, just copy it other than creating it again.



File name C:\Users\Acer\Desktop\Example.xlsx

Add action

Action Read data

Formula

Sheet Sheet 1

Function

Cells A1

Variable @a

Action list

Action	Sheet	Cells	Value	Formula	Function
Read data	Sheet 1	A1	@a		
Write data	Sheet 1	A2:D6	@b		

Figure. 2.3.4. Command display

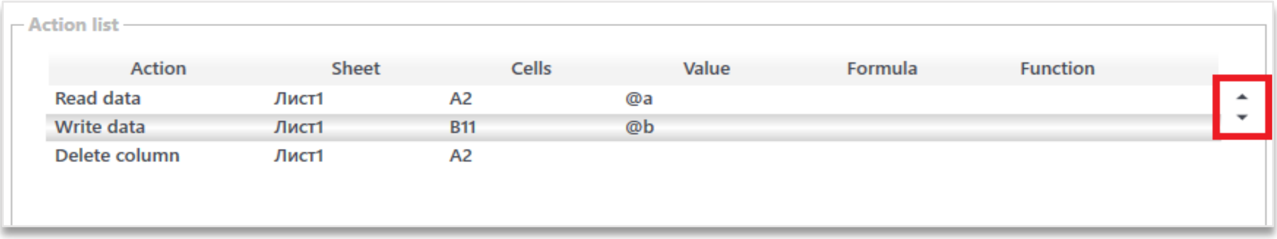
Editing a command

The «Edit» button is designed for editing commands. To change a command, select it from the list of commands, make all changes in the «Add command» section, then click «Edit» to save the changes.

Changing the order of actions

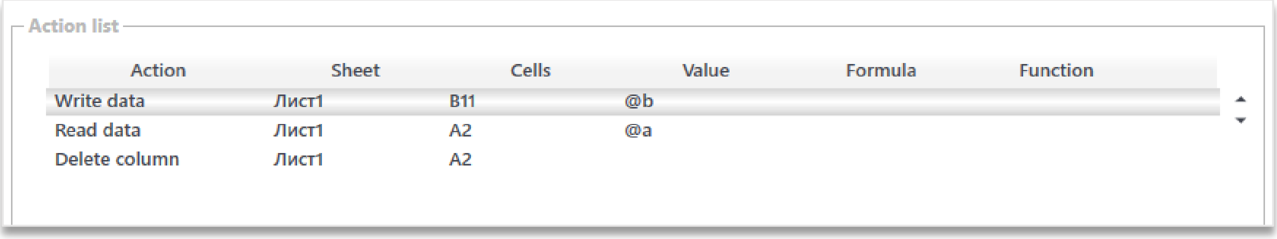
«Up» and «Down» buttons are designed to change the order of actions. Buttons are located to the right side of the list of commands.

Changing the order of actions is necessary when it is necessary to add an action that was not foreseen immediately and was not included in the list of commands. In this case you can add an action using the normal way, then just move it wherever you want.



Action list

Action	Sheet	Cells	Value	Formula	Function
Read data	Лист1	A2	@a		
Write data	Лист1	B11	@b		
Delete column	Лист1	A2			



Action list

Action	Sheet	Cells	Value	Formula	Function
Write data	Лист1	B11	@b		
Read data	Лист1	A2	@a		
Delete column	Лист1	A2			

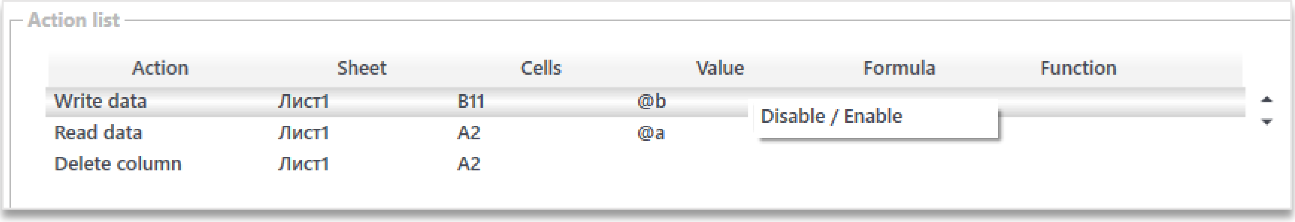
Figure. 2.3.4. Changing the order of actions

Deleting a command

The «Delete» button deletes the selected row from the list of commands.

Turning off a command

If you don’t wanna delete a command, because for example you wanted to see how the robot works without it, but then add it back to work, you can disable a command by right-clicking on the corresponding line and turn it off or, conversely, turn it on.



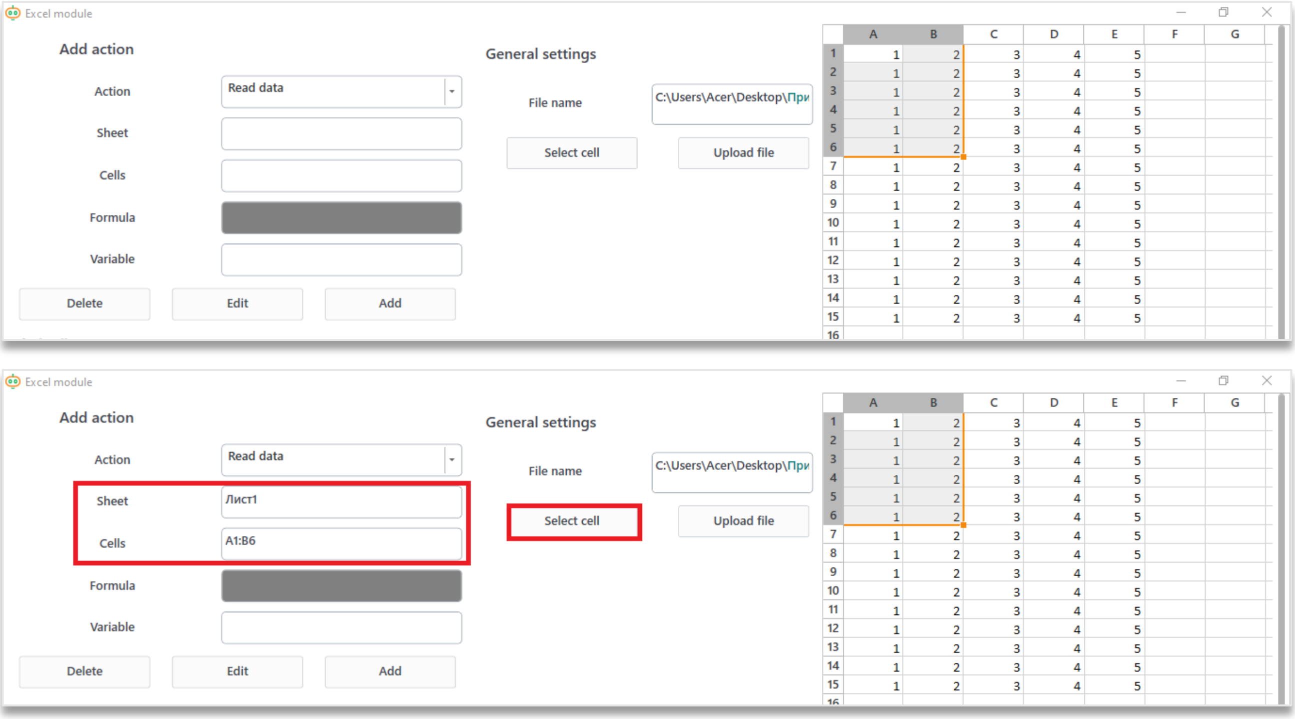
Action list

Action	Sheet	Cells	Value	Formula	Function
Write data	Лист1	B11	@b		
Read data	Лист1	A2	@a		
Delete column	Лист1	A2			

Figure 4.Command context menu

Selecting a cell

The button «Select cell» works for each action in its own way, but it has one meaning - it fills the «Cells» or «Range" fields (depending on which one is used in the selected action) with the name of the cell/range selected in the preview window and enters in the «Sheet» field the name of the sheet on which the selected cells are located.In some actions, only one cell is filled by clicking the button, although a range has been selected - this means that in this action the use of a range is impossible.



Excel module

Add action

Action: Read data

Sheet:

Cells:

Formula:

Variable:

General settings

File name: C:\Users\Acer\Desktop\При

	A	B	C	D	E	F	G
1	1	2	3	4	5		
2	1	2	3	4	5		
3	1	2	3	4	5		
4	1	2	3	4	5		
5	1	2	3	4	5		
6	1	2	3	4	5		
7	1	2	3	4	5		
8	1	2	3	4	5		
9	1	2	3	4	5		
10	1	2	3	4	5		
11	1	2	3	4	5		
12	1	2	3	4	5		
13	1	2	3	4	5		
14	1	2	3	4	5		
15	1	2	3	4	5		
16							

Excel module

Add action

Action: Read data

Sheet: Лист1

Cells: A1:B6

Formula:

Variable:

General settings

File name: C:\Users\Acer\Desktop\При

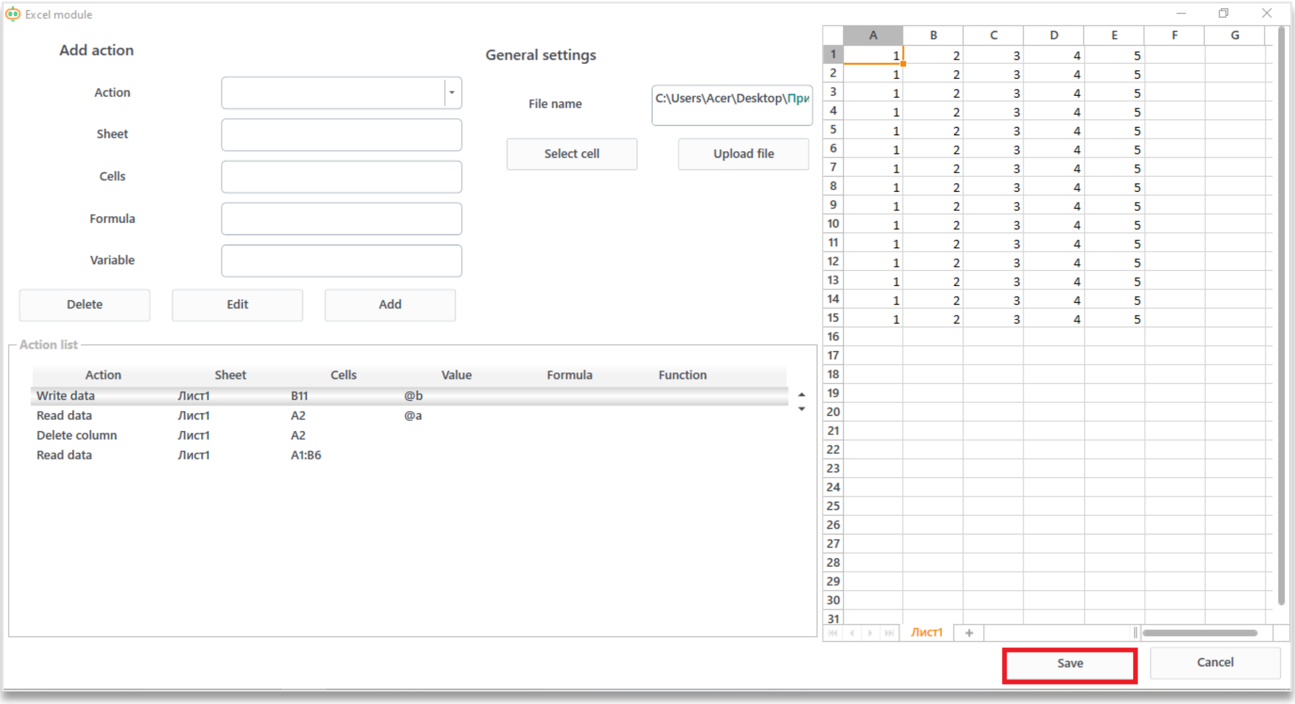
	A	B	C	D	E	F	G
1	1	2	3	4	5		
2	1	2	3	4	5		
3	1	2	3	4	5		
4	1	2	3	4	5		
5	1	2	3	4	5		
6	1	2	3	4	5		
7	1	2	3	4	5		
8	1	2	3	4	5		
9	1	2	3	4	5		
10	1	2	3	4	5		
11	1	2	3	4	5		
12	1	2	3	4	5		
13	1	2	3	4	5		
14	1	2	3	4	5		
15	1	2	3	4	5		
16							

Figure. 2.3.5. Selecting a cell

In some actions, the “Select Cell” button fills in the numbers of columns, rows. When working with files, it is recommended to use it to minimize the occurrence of syntax errors.

Saving a command

To save a command in the Excel module press the «Save» button in the bottom right corner.



File name: C:\Users\Acer\Desktop\При

Action	Sheet	Cells	Value	Formula	Function
Write data	Лист1	B11	@b		
Read data	Лист1	A2	@a		
Delete column	Лист1	A2			
Read data	Лист1	A1:B6			

	A	B	C	D	E	F	G
1	1	2	3	4	5		
2	1	2	3	4	5		
3	1	2	3	4	5		
4	1	2	3	4	5		
5	1	2	3	4	5		
6	1	2	3	4	5		
7	1	2	3	4	5		
8	1	2	3	4	5		
9	1	2	3	4	5		
10	1	2	3	4	5		
11	1	2	3	4	5		
12	1	2	3	4	5		
13	1	2	3	4	5		
14	1	2	3	4	5		
15	1	2	3	4	5		
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							

Figure. 2.3.6. Saving in Excel module

Chapter 4. Set of fields in the module

For each action in the section "Add a team" there is a set of fields necessary for it, which will be displayed when you select an action. By default fields «Sheet», «Cells», «Formula» and «Variable» are displayed. Almost all commands have the exact same set of fields, but some require slightly different information to complete an action.

List of possible fields with rules for filling them

1. Sheet - a field in which you must specify the name or number of the sheet (starting from 0), on which you must perform a specified action;
2. Cells - a field in which either the address of one cell (A1) or the range of cells (A1: C10) is expected. If the cell address is calculated dynamically and is unknown, then you can use the numbers of columns and rows of cells, that is, instead of «C1» you can write «3.1», where 3 is the column number, 1 is the row number. You can also specify a range - instead of «A3: D4» - «1.3: 4.4». The second format for writing cells can be used only in relation to the «Cell(s)" fields, but it is not possible in the «Formula» fields (see next paragraph);
3. Formula - a field for entering a formula in the standard Excel format. For example, «= IF (A2> B2: 1: 0)». For more convenient input of long formulas, the field can be expanded by clicking on the button with an arrow in the input field:

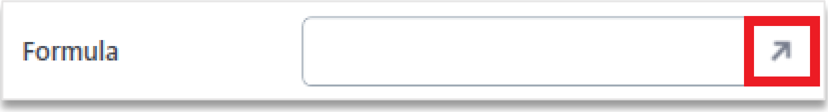


Figure. 2.4.1. Field «Formula»

4. Variable - field for entering a name of a variable, in which result of action is gonna be put in. Name starts with the «@» symbol;
5. Value - a field for entering text or variable name, in which lies the text used in the selected action;
6. Column – number (from 1) or the name of a column;
7. Row – row number (from 1). In some actions («Find data», «Find sheet») a text string is used to search in a file;
8. File directory – directory to the folder where a file will be stored. You can choose the file directory by pressing the button with the folder on it (Figure. 2.4.2);

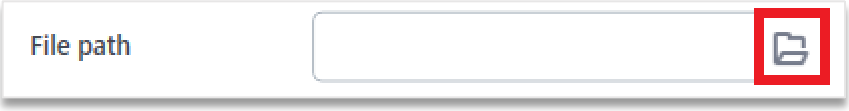


Figure. 2.4.2. «File directory» field

9. File name – the name of a file. You can specify the filename extension, otherwise the file will be saved with .xlsx filename extension;
10. Path of the copied file – is the path to the file with its extension, which will be used in this action in addition to the main file to copy data from it. You can choose the file using a button with a folder on it (Figure. 2.4.3):

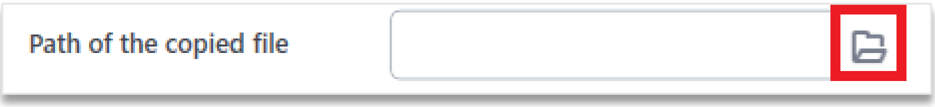


Figure. 2.4.3. «Path of the copied file» field

11. Sheet of the copied file – a name or a number of the sheet in a copied file;
12. Columns – similar to the «Column» field, only here they can be listed through «;» symbol;
13. Values – string values (or variables storing them) listed through «;»;
14. Row, column – number of a column and number of a row, separated by a comma;
15. A cell from the range of values – any cell from a table to which an action is applied (for example, if table A1: D10, then any cell from this range is B5);
16. Function – a drop-down list with the list of functions, which changes depending on an action selected; selection of one of the presented values is required;
17. Descending - the checkbox used in the «Sort» action, which is checked if it is necessary to sort in descending order and is not set otherwise;
18. Table range – similar to the «Cells field», but it has to be a range (meaning that «:» symbol must be present);
19. Insert sheet - name or number of the sheet (from 0);
20. Unique columns - used in the «Get unique column values» action, you must enter the names or numbers of columns (from 1), which must be unique;
21. For columns – similar to the «Column» field;
22. Each time a column changes – similar to the «Column» field;
23. Sheet, range - the field into which the sheet of the file should be entered (similar to the «Sheet» field), then a comma is placed and the range or cell is specified (similar to the «Cells» field);
24. Sheet, source range - similar to the «Sheet, range» field;
25. Sheet, range for insertion – similar to the «Sheet, range» field;

26. Fields in columns - used in the «Pivot table» action. You must enter the column name from the worksheet (not A, B, C, etc., but the column name of your table, for example, «Date of creation», without using quotation marks);
27. Fields in values – similar to the «Fields in columns»;
28. Fields in rows – similar to the «Fields in columns»;
29. Coped sheet – similar to the «Sheet» field;
30. Coped range/cell – similar to the «Cells» field;
31. Range/cell for insertion – similar to the «Cells» field;
32. File for insertion (not necessary) – the path to the file with its extension is an optional field;
33. Location (top left, bottom right.) – two cells separated by a comma that indicate the position - the first is the top left corner, the second is the bottom right;
34. Range of names – similar to the «Cells» field;
35. Range of arguments - similar to the «Cells» field;
36. Range of values - similar to the «Cells» field;
37. Insertion cell - similar to the «Cells» field, but singular;
38. Type of data – drop-down list, the selection of at least 1 value is necessary;
39. Operation – similar to «Type of data»;
40. Criterias – arguments for the selected operation can be numbers as well as rows, they are listened through «;»;
41. Filling color – color in ARGB format, separating each parameter with a comma, you can select a color through the color selection button in the field:

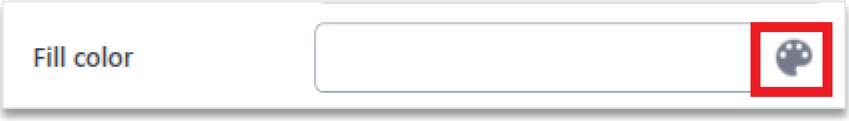


Figure. 2.4.4. «Filling color» field

42. Text color – similar to the «Filling color»;
43. Color – similar to the «Filling color»;
44. Comparison value – is a text or numeric value;
45. Cell/field – similar to the «Cells» field;;
46. Columns - a list of numbers (starting with 1) or the names of required columns separated by comma.
47. Width - a number, both integer and fractional (through «.» or «,»);

48. Lines - enumeration of numbers (from 1) lines separated by commas;

49. Height – similar to the «Width» field;

50. General position - a drop-down list, you must choose at least one value.

51. Alignment of contents – similar to the «General position» field;

52. Font – a drop-down list with all Microsoft Office fonts, you must choose one of them;

53. Text style – similar to the «General position» field;

54. Size – similar to the «Width» field;

55. Position – similar to the «General position» field;;

56. Style – similar to the «General position» field..

Variables can be used in all fields, but they must contain what the fields require, for example, you can write @column in the "Column" field, but then @column must contain the column number starting with one, or its name. Fields that are grayed out are not used in the selected action.

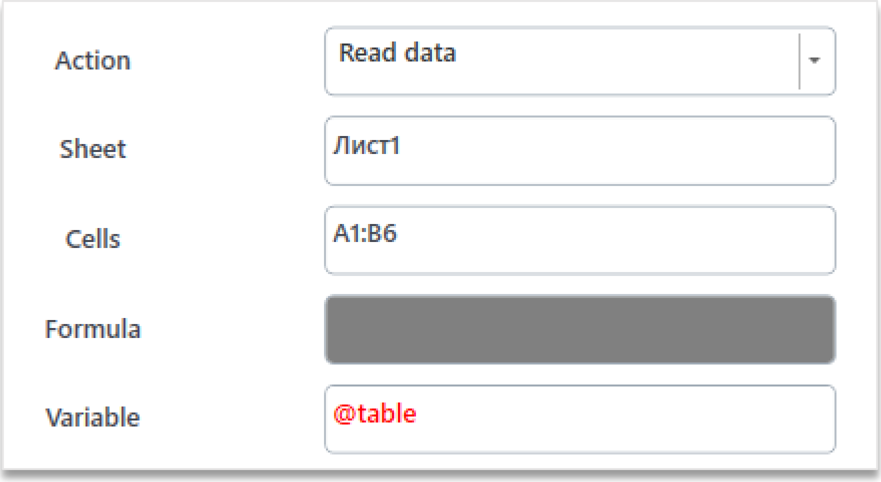
Chapter 5. Actions available in the module

Actions are divided into groups for more comfortable navigation through the drop-down list. If you know the name of an action, you can enter it into the «Action» field, and the software will try to give you the action you need.

«Work with data» group

Action «Read data»

This action allows the robot to read from one specific cell or from a range of cells. If only one cell is specified then the «Element» type variable will be made, but if a range of cells is specified then the «Table» type variable will be made.

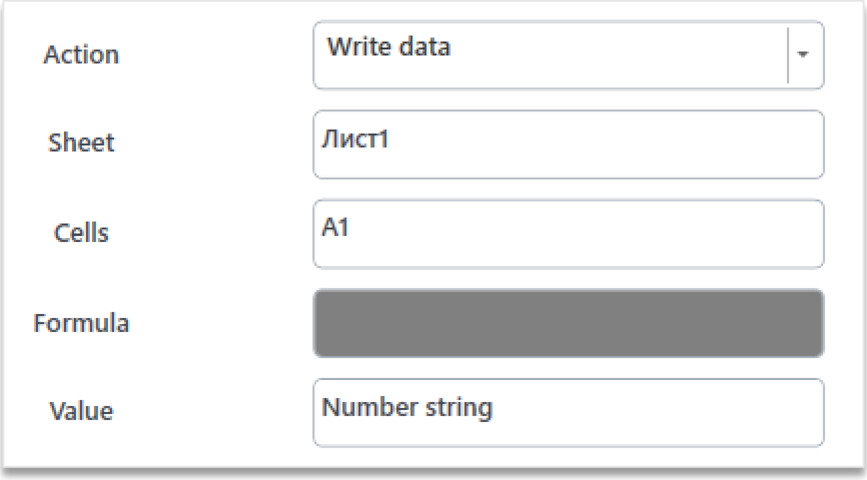


Action	Read data
Sheet	Лист1
Cells	A1:B6
Formula	
Variable	@table

Figure. 2.5.1. Read data

Action «Record data»

This action allows you to record the specified value in one or in a range of cells

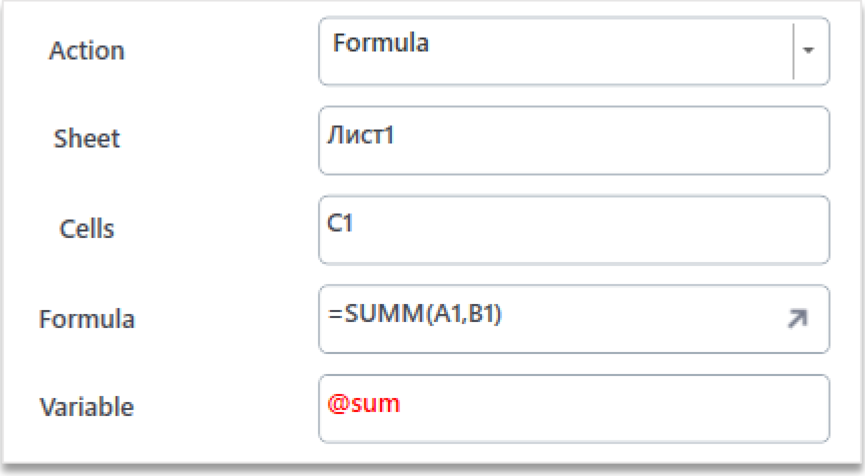


Action	Write data
Sheet	Лист1
Cells	A1
Formula	
Value	Number string

Figure. 2.5.2. Record data

Action «Formula»

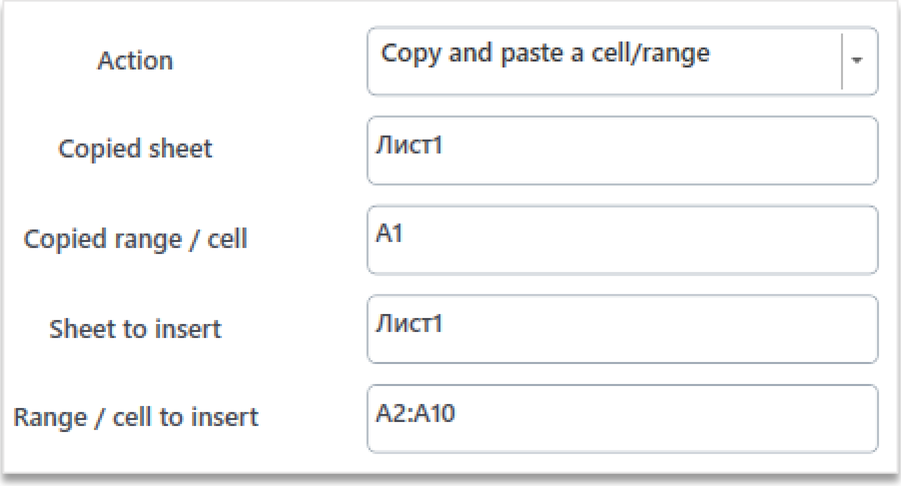
This action makes the robot write a formula in a specified cell (or a range of cells) and return the result to the specified variable. The «Variable» field can be left blank, if the result of a formula is not important.



Action	Formula
Sheet	Лист1
Cells	C1
Formula	=SUMM(A1,B1)
Variable	@sum

Figure. 2.5.3. Formula

TIP. Suppose you need to write a formula that applies to the current row and then stretch it to several others, as in Excel. Start by applying the «Formula» action for the first row. Then stretch it by using the «Copy and paste cell/range» action in relation to the adjusted formula. This is how the formula looks like:

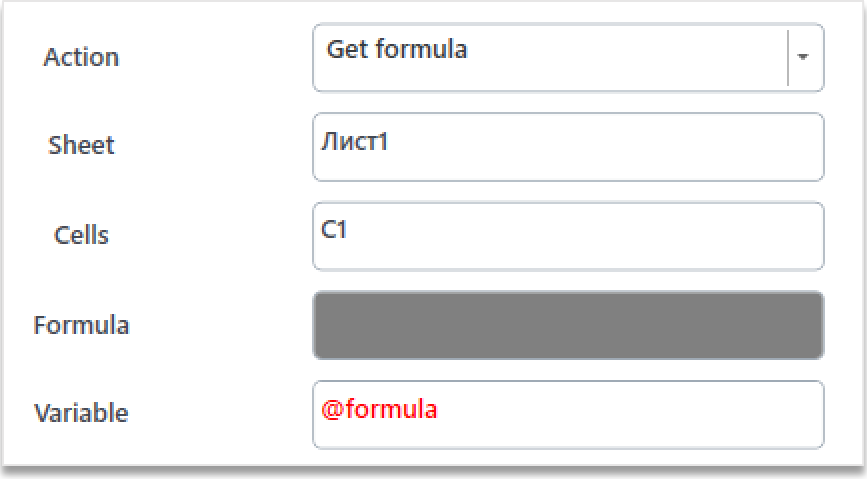


Action	Copy and paste a cell/range
Copied sheet	Лист1
Copied range / cell	A1
Sheet to insert	Лист1
Range / cell to insert	A2:A10

Figure. 2.5.4. Copy and paste cells

Action «Get formula»

The action returns the formula from the cell to the specified variable.

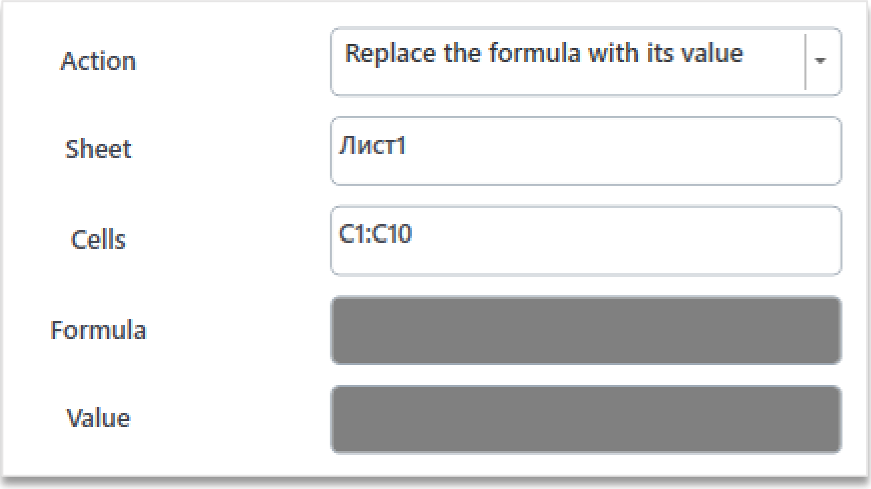


Action	Get formula
Sheet	Лист1
Cells	C1
Formula	
Variable	@formula

Figure. 2.5.5. Get formula

Action «Replace the formula with its value»

This action replaces the formula in the cell or range of cells with its result.



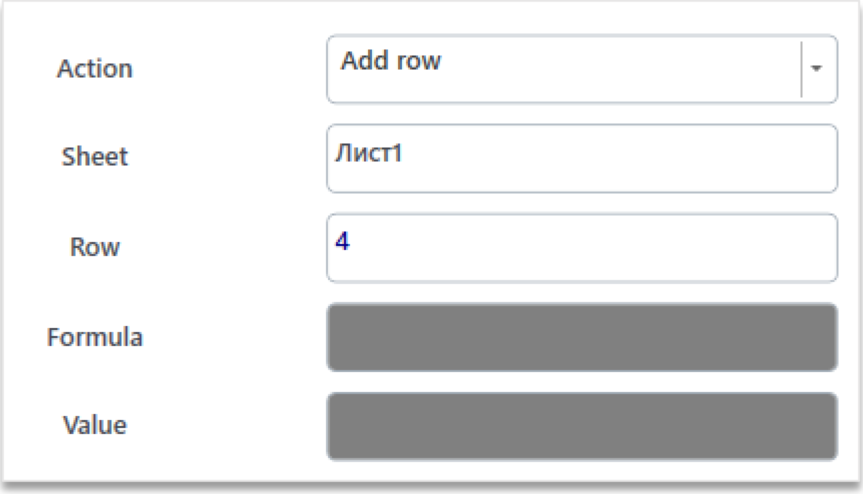
Action	Replace the formula with its value
Sheet	Лист1
Cells	C1:C10
Formula	
Value	

Figure. 2.5.5. Replacing the formula

«Work with rows» group

Action «Add row»

Action adds a row to the specified location.

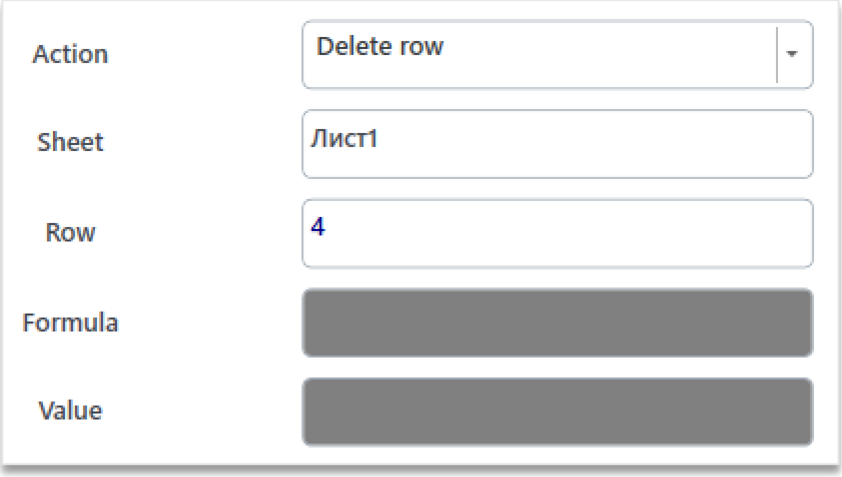


Action	Add row
Sheet	Лист1
Row	4
Formula	
Value	

Figure. 2.5.6. Adding a row

Action «Delete row»

Action deletes the specified row

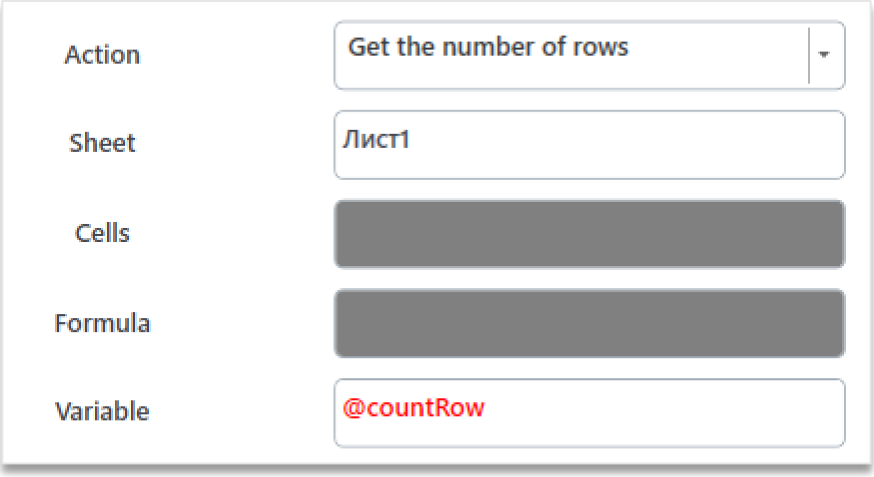


Action	Delete row
Sheet	Лист1
Row	4
Formula	
Value	

Figure. 2.5.7. Deleting a row

Action «Get row count»

The action returns the number of the last used row in the sheet to the specified variable.

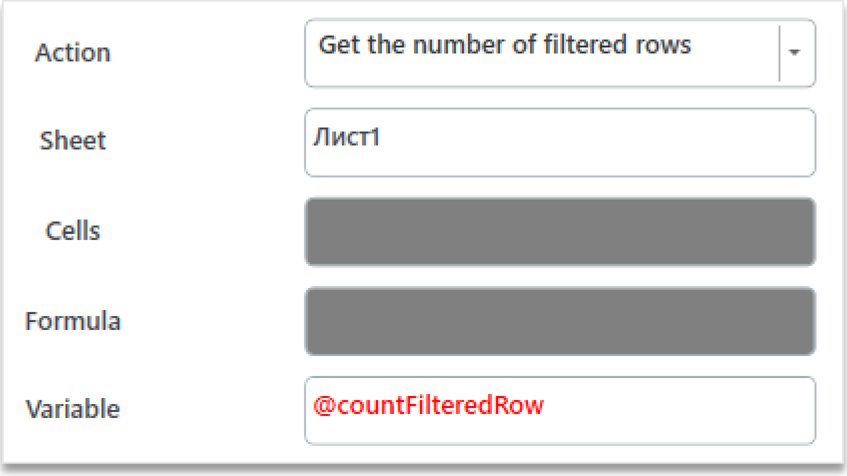


Action	Get the number of rows
Sheet	Лист1
Cells	
Formula	
Variable	@countRow

Figure. 2.5.8. Getting the row count

Action «Get the number of filtered rows»

Action returns the number of visible rows with the applied filter to the specified variable.



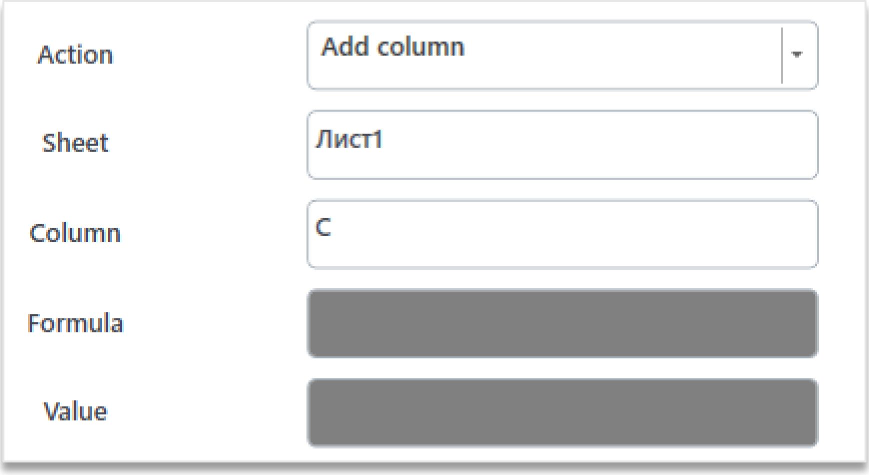
Action	Get the number of filtered rows
Sheet	Лист1
Cells	
Formula	
Variable	@countFilteredRow

Figure. 2.5.9. Getting the number of filtered rows

«Work with columns» group

Action «Add column»

Action adds a new column into the specified location.

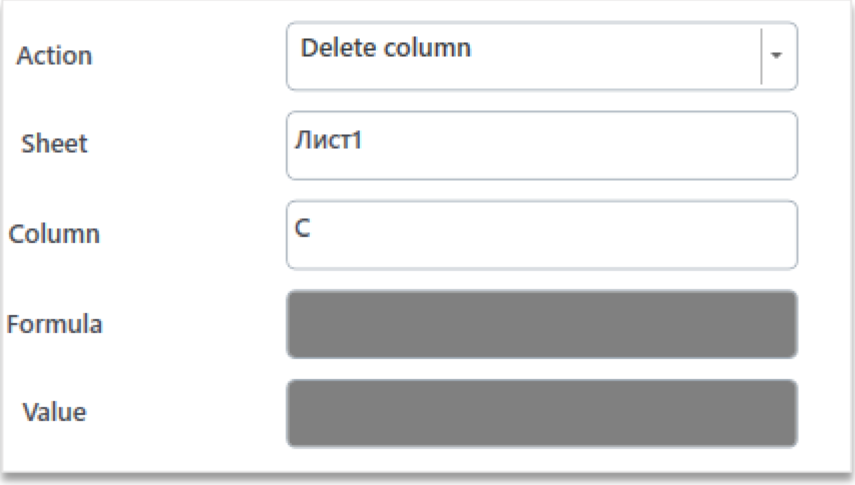


Action	Add column
Sheet	Лист1
Column	C
Formula	
Value	

Figure. 2.5.10. Adding a column

Action «Delete column»

Action deletes the specified column.

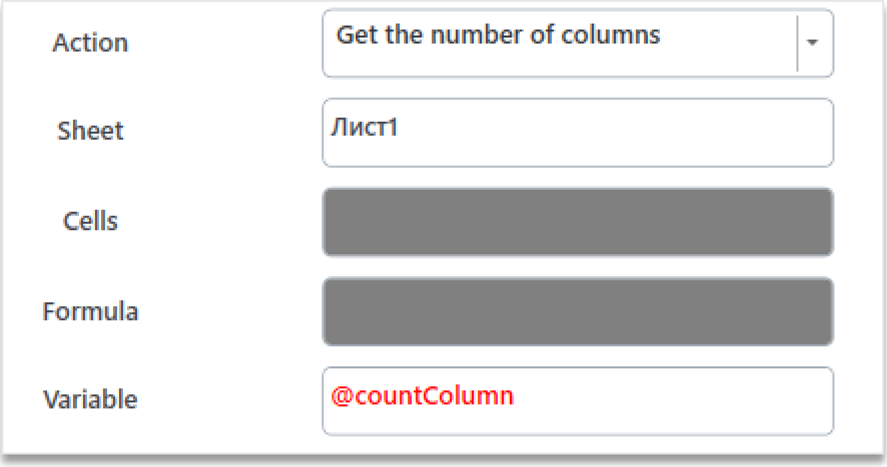


Action	Delete column
Sheet	Лист1
Column	C
Formula	
Value	

Figure. 2.5.11. Deleting a column

Action «Get the column count»

The action returns the number of the last used column in the sheet.



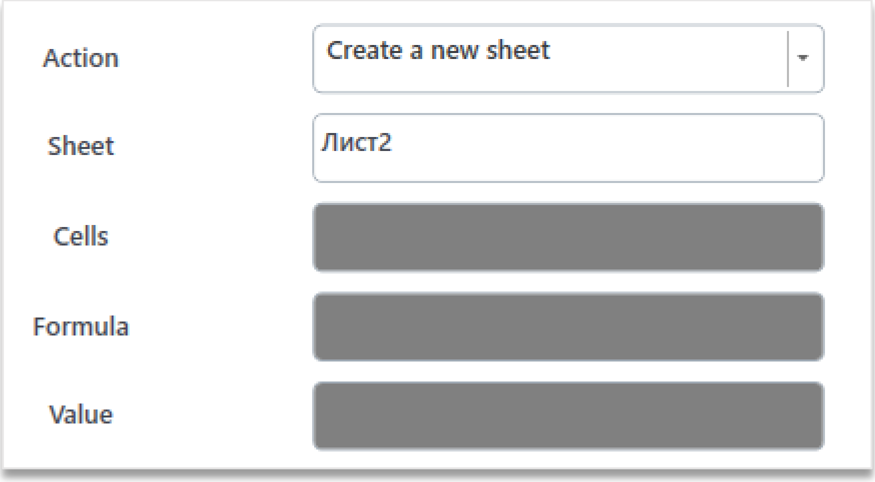
Action	Get the number of columns
Sheet	Лист1
Cells	
Formula	
Variable	@countColumn

Figure. 2.5.12. Getting the column count

«Work with sheets and files» group

Action «Create a new sheet»

The action creates a new sheet with a specified name..

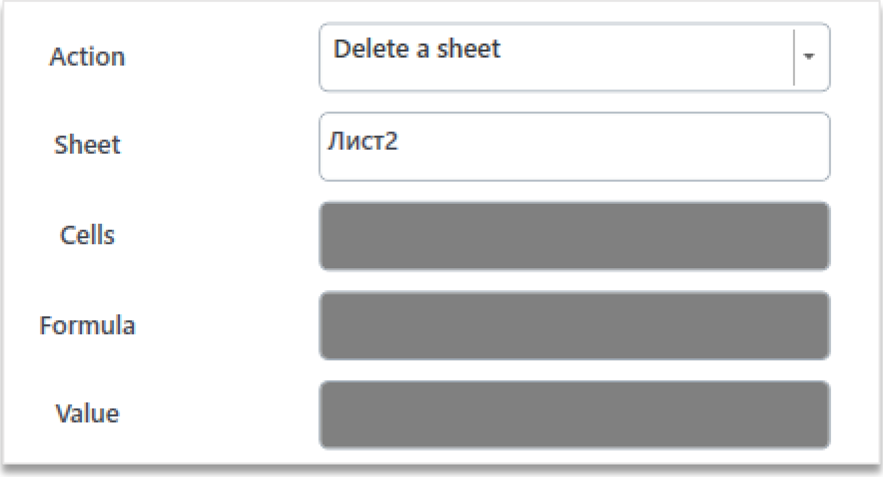


Action	Create a new sheet
Sheet	Лист2
Cells	
Formula	
Value	

Figure. 2.5.13. Creating a new sheet

Action «Delete sheet»

Action deletes a sheet with a specified name.

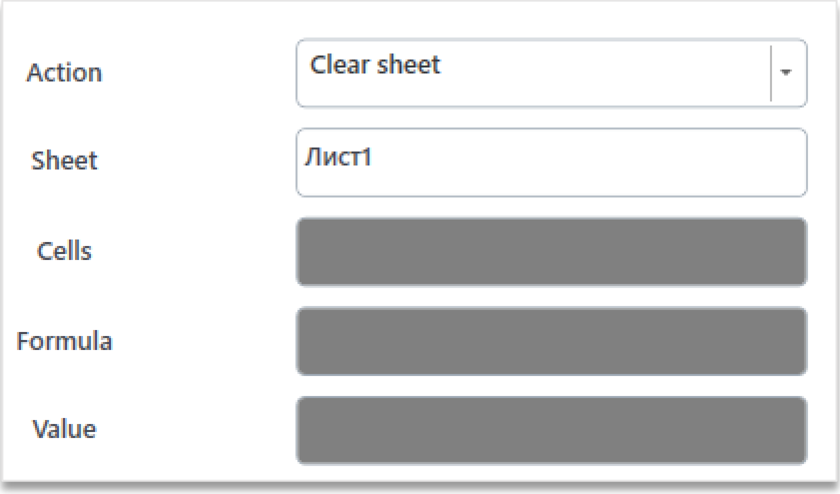


Action	Delete a sheet
Sheet	Лист2
Cells	
Formula	
Value	

Figure. 2.5.15. Deleting a sheet

Action «Clear sheet»

Action clears data from the cells. **Action doesn’t clear the formatting!**

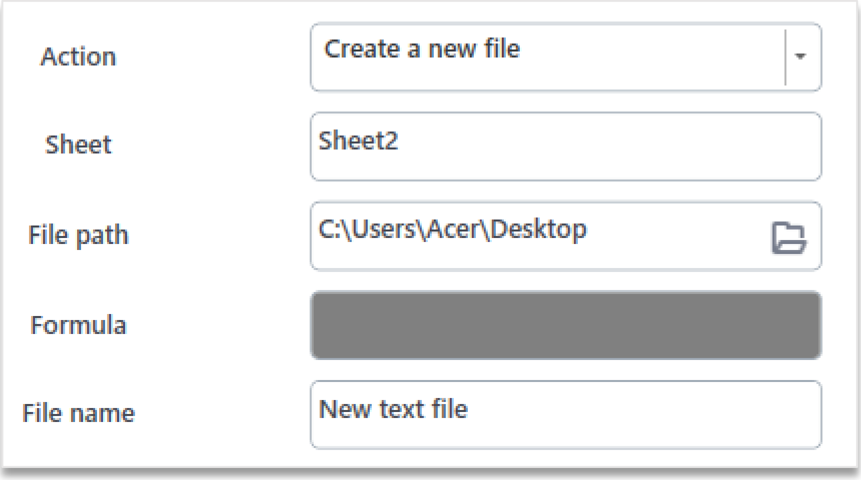


Action	Clear sheet
Sheet	Лист1
Cells	
Formula	
Value	

Figure. 2.5.16. Clearing the sheet

Action «Create a new file»

Action creates a new file in the specified folder with specified name and filename extension (if the filename extension is not specified, .xlsx file will be created.) If the «Sheet» field is filled, then the first sheet of the file will be named as specified in the «Sheet» field, if not it will leave it by default.

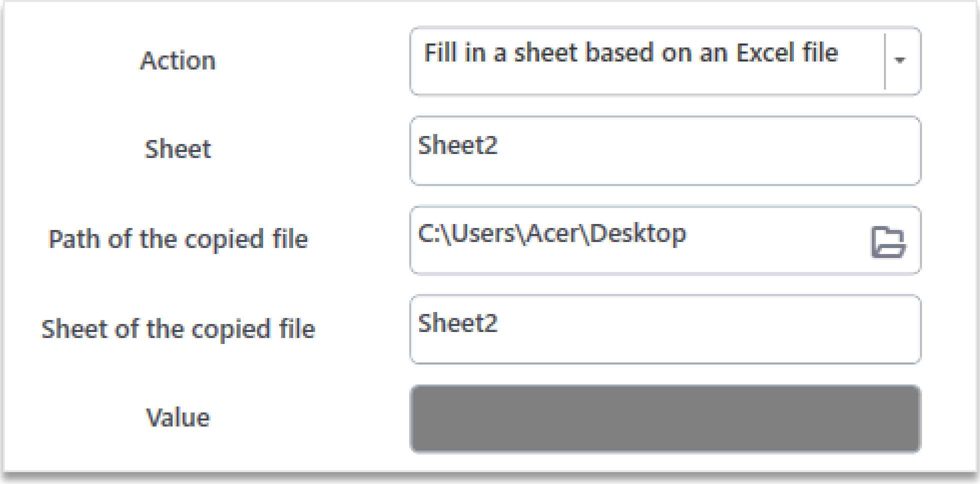


Action	Create a new file
Sheet	Sheet2
File path	C:\Users\Acer\Desktop
Formula	
File name	New text file

Figure. 2.5.17. Creating a new file

Action «Fill in a sheet, based on an Excel file»

This action fills the sheet of the current file by analogy with the sheet of some other file. In the «Sheet» field the sheet of the current file is specified.

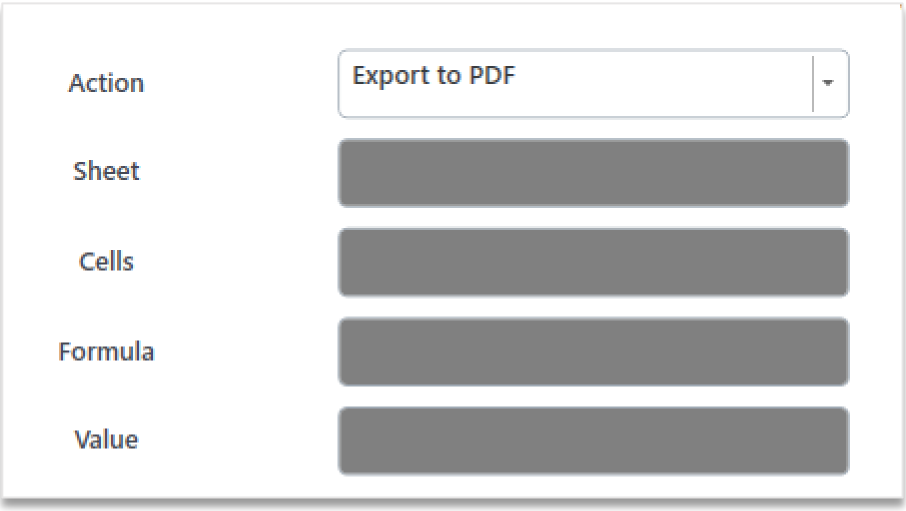


Action	Fill in a sheet based on an Excel file
Sheet	Sheet2
Path of the copied file	C:\Users\Acer\Desktop
Sheet of the copied file	Sheet2
Value	

Figure. 2.5.18. Filling in a sheet

Exporting to PDF

Action allows to save excel files in the PDF format using standard WIndows methods, meaning that the division of the sheet of the Excel file into sheets in the PDF document will be performed according to the standard settings specified in the Excel file (the default is A4 sheet size, with normal fields).



Action	Export to PDF
Sheet	
Cells	
Formula	
Value	

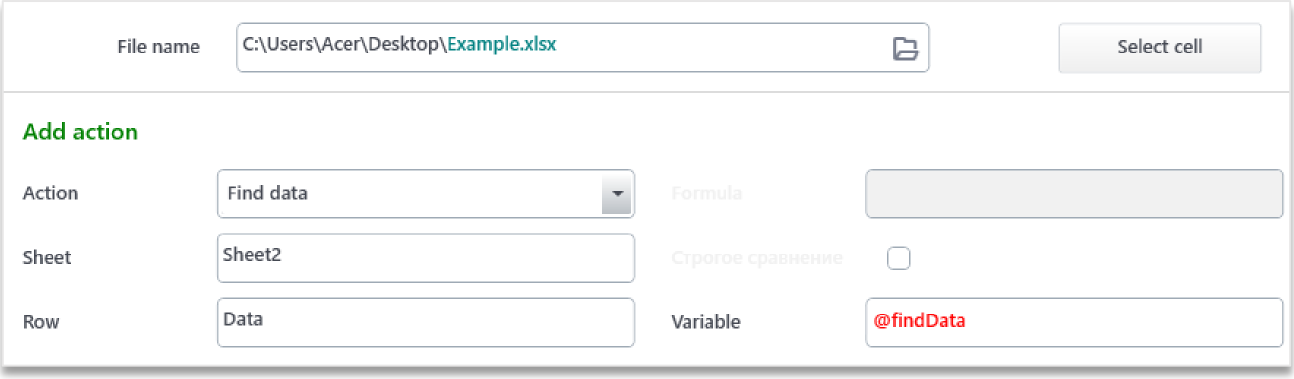
Figure. 2.5.19. Exporting to PD

The name of the file and its directory will coincide with the exported file.

«Search» group

Action «Find data»

Searches for the specified row in the specified sheet and returns the address of the first cell found.



File name	C:\Users\Acer\Desktop\Example.xlsx	Select cell

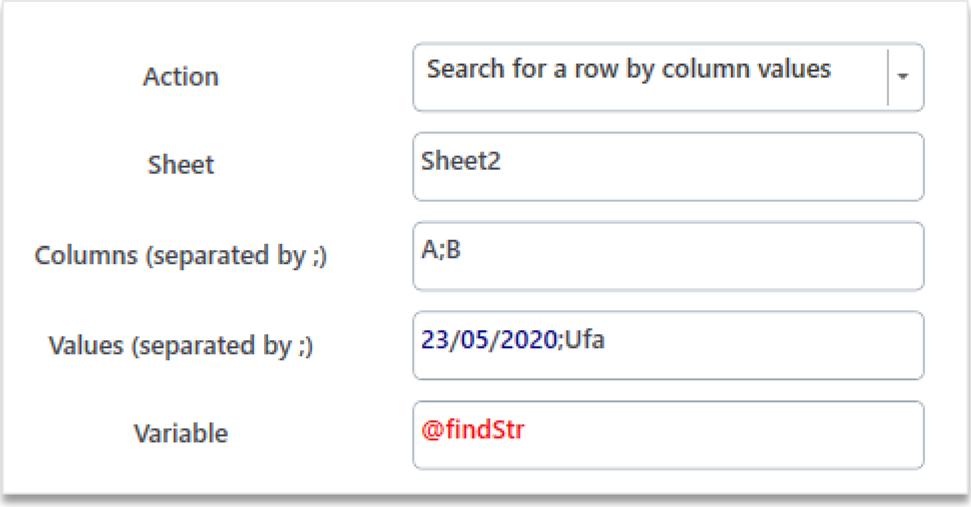
Add action

Action	Find data	Formula	
Sheet	Sheet2	Строгое сравнение	☐
Row	Data	Variable	@findData

Figure. 2.5.20. Find data

Action «Search for a row by column values»

This action searches and returns the number of the first row found, in the specified columns of which the specified values are written.



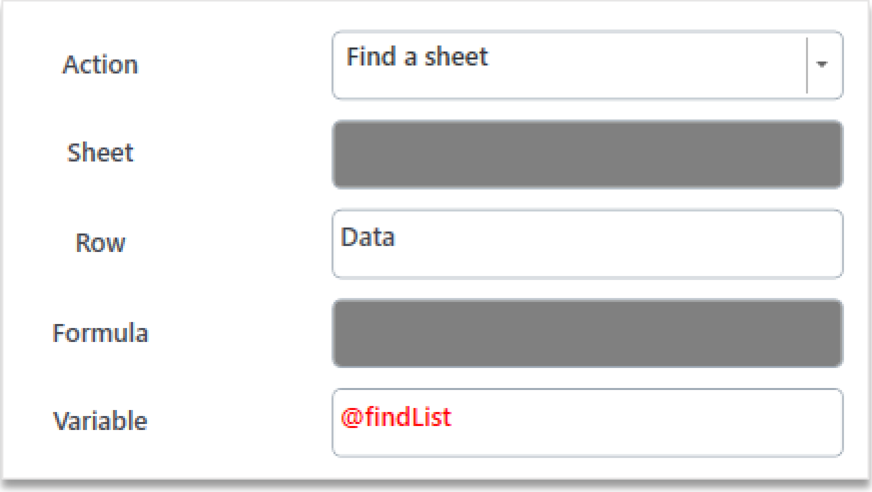
Action	Search for a row by column values
Sheet	Sheet2
Columns (separated by ;)	A;B
Values (separated by ;)	23/05/2020;Ufa
Variable	@findStr

Figure. 2.5.21.Row search

In this case, a row will be searched, in column A - 03/23/2020, and in column B - Ufa.

Action «Find sheet»

Searches for a sheet in which there is a cell with the value specified in the «Row» field and returns its name to the specified variable.



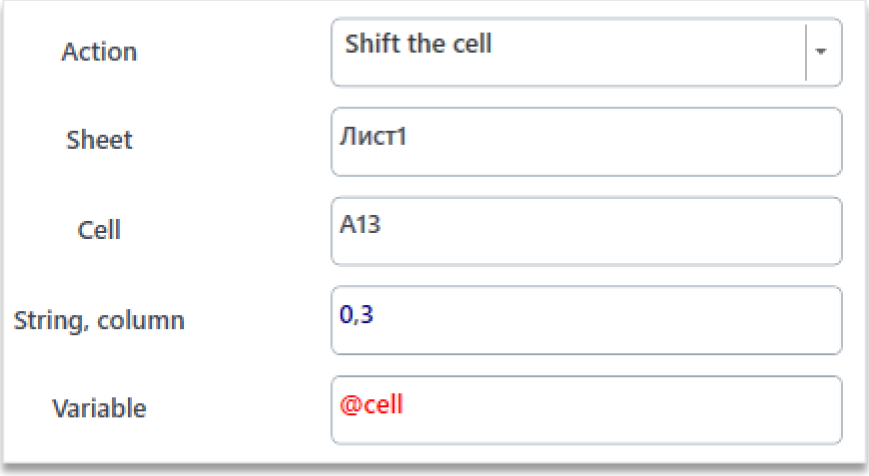
Action	Find a sheet
Sheet	
Row	Data
Formula	
Variable	@findList

Figure. 2.5.22. Find sheet

Group «Other functions»

Action «Shift cell»

Moves a cell on the specified number of rows and columns, and then returns the result to the variable (in this case, the result is D13).

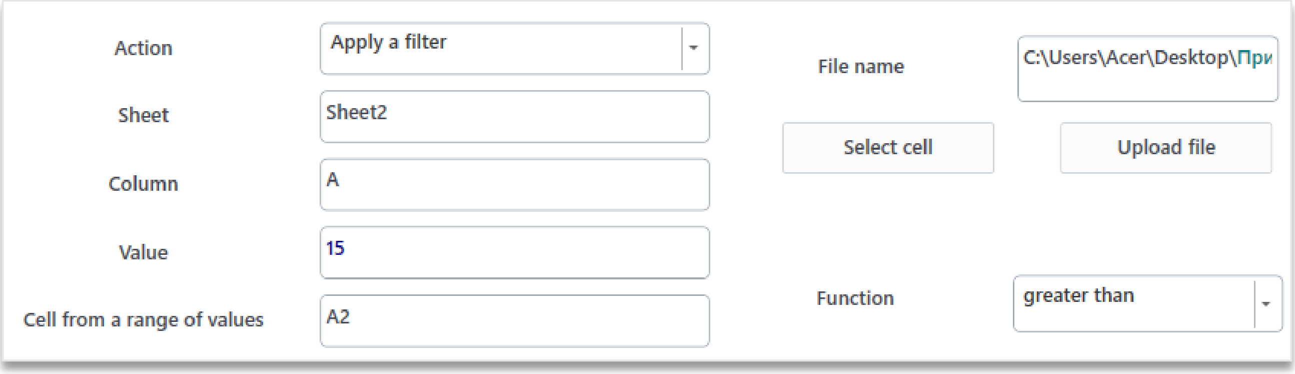


Action	Shift the cell
Sheet	Лист1
Cell	A13
String, column	0,3
Variable	@cell

Figure. 2.5.23. Move cell

Action «Apply filter»

This action adds a filter to the table specified on a sheet, the cell from which is specified in the «A cell from the range of values» field. The column which you want to apply a filter to is written in the «Column» field. You can only filter a single column. If you need to apply several filters to one table, then you need to add an action to each filter within the same Excel module. The function by which the comparison should take place is selected from the drop-down list in the «Function» field, the value with which the comparison will be performed during filtering is indicated in the «Value» field.

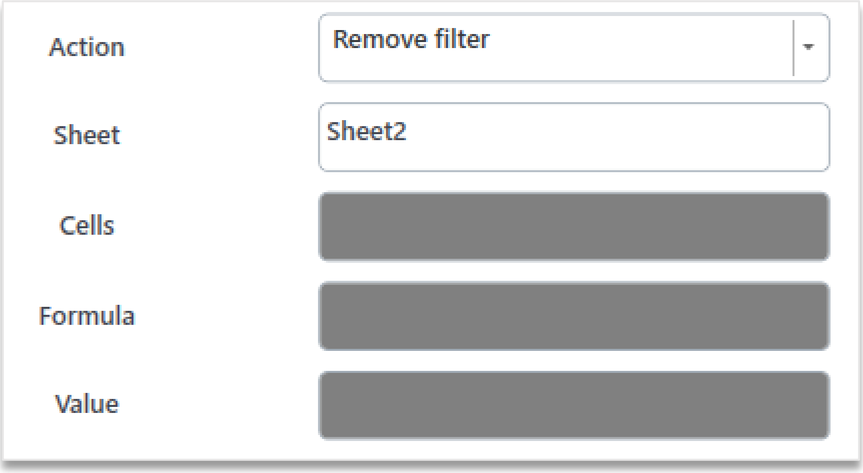


Action	Apply a filter
Sheet	Sheet2
Column	A
Value	15
Cell from a range of values	A2
File name	C:\Users\Acer\Desktop\При
Function	greater than

Figure. 2.5.24. Apply filter

Action «Delete filter»

The action deletes all filters from the sheet.

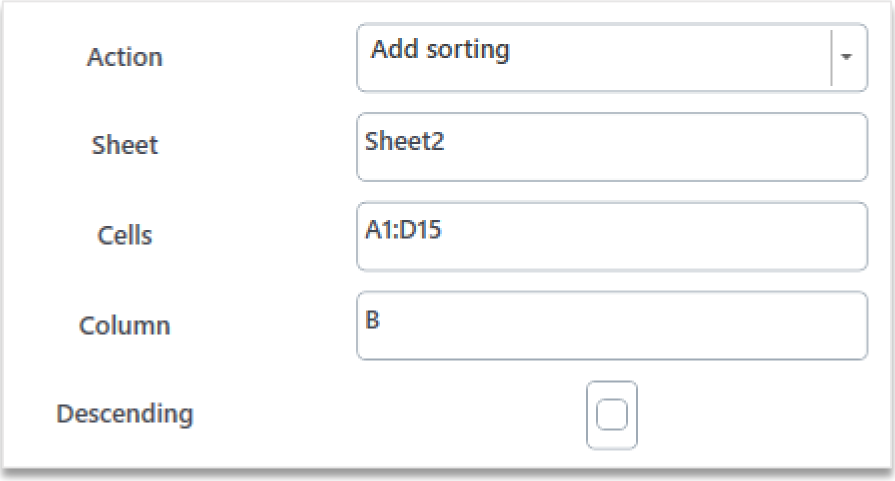


Action	Remove filter
Sheet	Sheet2
Cells	
Formula	
Value	

Figure. 2.5.25. Deleting the filter

Action «Add sorting»

The action adds sorting to the specified range by the selected column. Depending on the checkbox, it is determined whether the column is sorted in descending or ascending order. You can only sort a single column.

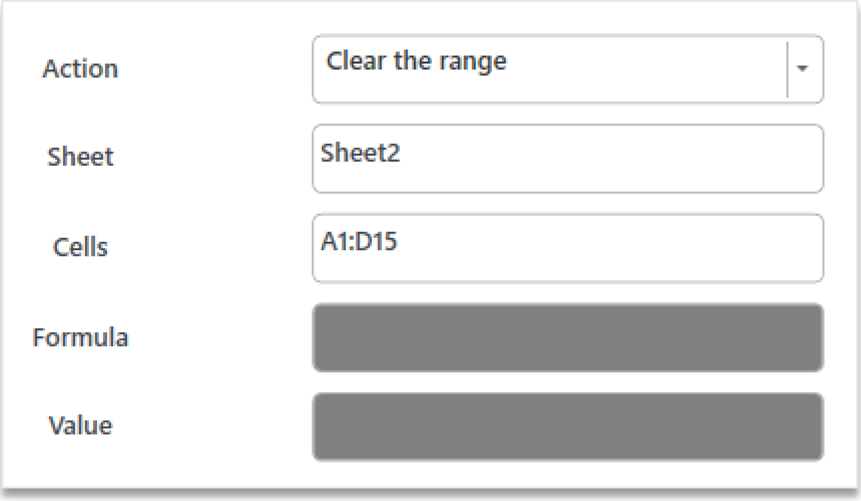


Action	Add sorting
Sheet	Sheet2
Cells	A1:D15
Column	B
Descending	□

Figure. 2.5.26. Adding sorting

Action «Clear the range»

The action deletes all cell values and formatting in the specified range.

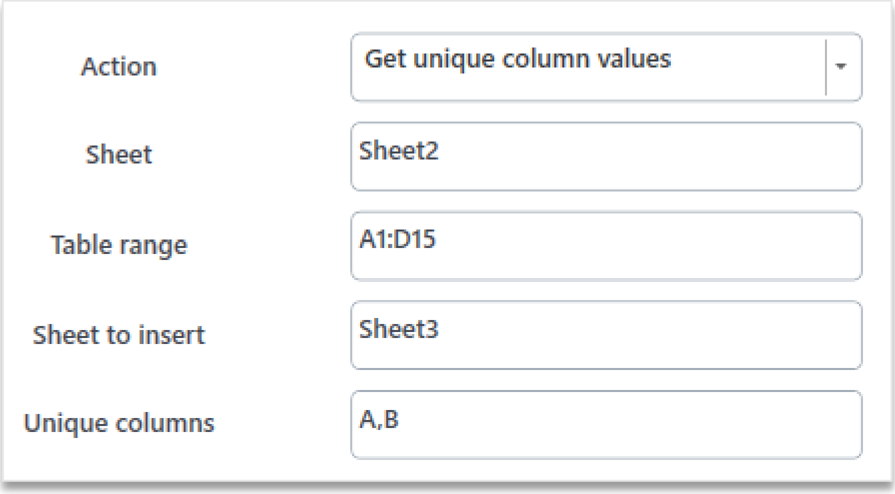


Action	Clear the range
Sheet	Sheet2
Cells	A1:D15
Formula	
Value	

Figure. 2.5.27. Clearing the range

Action «Get unique column values»

Similar to the «Delete duplicates» action in Excel, this action returns only the unique values of the specified columns from the range. The finished table can be inserted both into the same sheet and into another sheet, but in the same range. Depending on the presence of duplicates, the number of rows may be reduced.

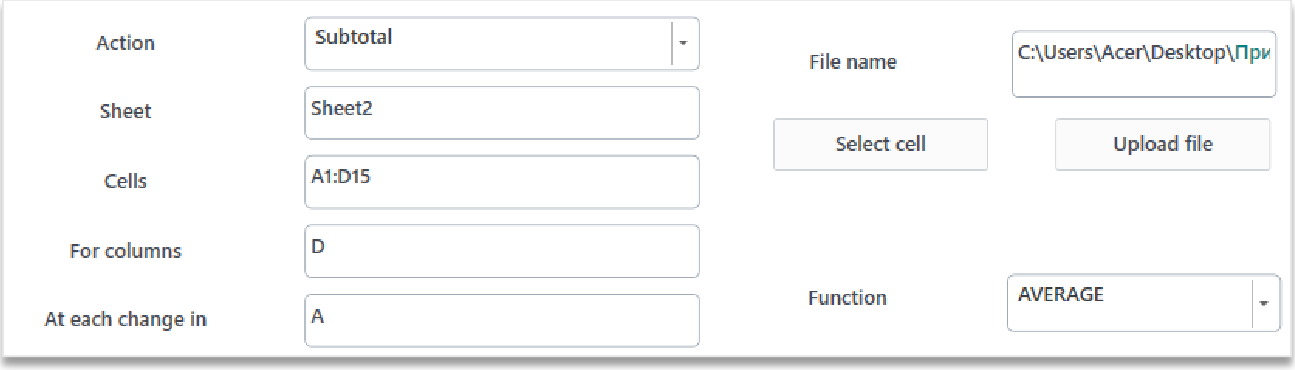


Action	Get unique column values
Sheet	Sheet2
Table range	A1:D15
Sheet to insert	Sheet3
Unique columns	A,B

Figure. 2.5.28. Getting unique values

Action «Subtotal»

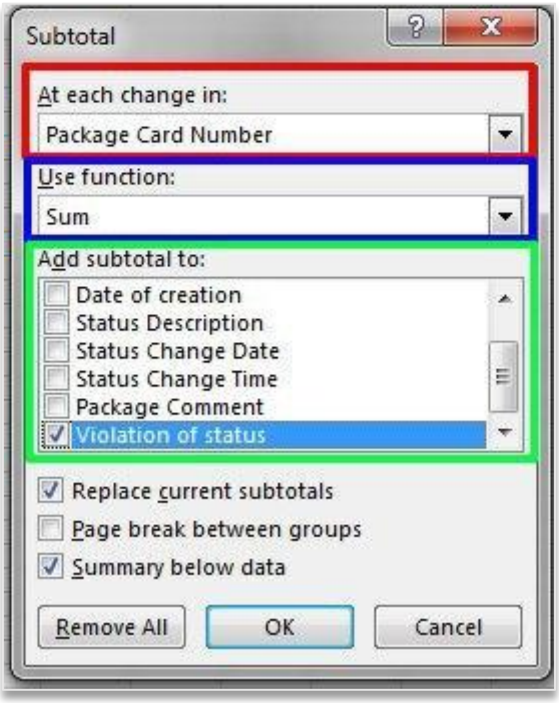
The action adds an Excel «Subtotals» operation to the specified sheet to the specified range.



Action	Subtotal	File name	C:\Users\Acer\Desktop\При
Sheet	Sheet2		
Cells	A1:D15		
For columns	D		
At each change in	A	Function	AVERAGE

Figure. 2.5.29. Subtotals

The «For columns» field matches the «Add totals by» field in Excel (Marked green on the Figure. 2.3.28), the «Each time a column changes» field matches the similar field in Excel, but must be filled in with the name of the Excel column (A, B , ... or 1, 2, ...) (highlighted in red in Figure. 2.3.28), the «Functions» field is filled with the operation that should be applied to all of the selected columns (highlighted in blue in Figure. 2.3.28).



Subtotal

At each change in: Package Card Number

Use function: Sum

Add subtotal to:

- □ Date of creation
- □ Status Description
- □ Status Change Date
- □ Status Change Time
- □ Package Comment
- ☑ Violation of status

☑ Replace current subtotals

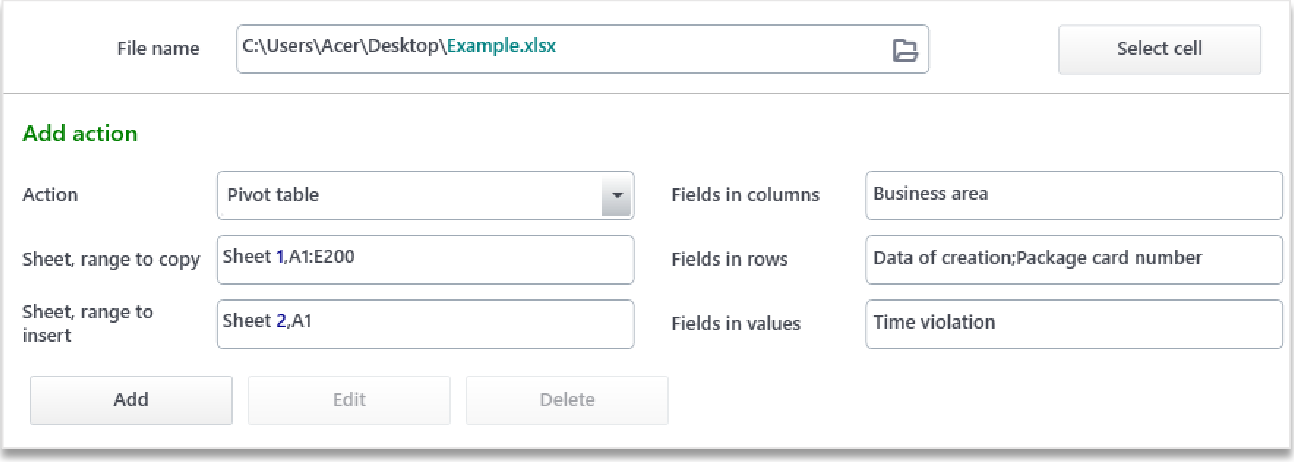
□ Page break between groups

☑ Summary below data

Figure. 2.5.30. Subtotals

Action «Pivot table»

Adds a summary table to the specified sheet and range (the «Sheet, range for insertion» field) based on the table data from the «Sheet, source range» field.



File name: C:\Users\Acer\Desktop\Example.xlsx

Add action

Action: Pivot table

Sheet, range to copy: Sheet 1,A1:E200

Sheet, range to insert: Sheet 2,A1

Fields in columns: Business area

Fields in rows: Data of creation;Package card number

Fields in values: Time violation

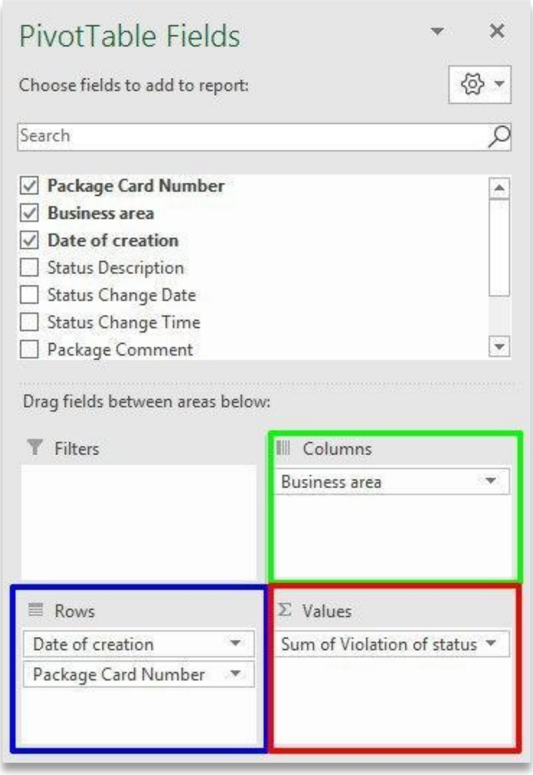
Figure. 2.5.31. Pivot table

In the «Fields in columns», «Fields in values» and «Fields in rows» fields the column **names of the source table** (and not the Excel table) are entered, as in Excel:

«Fields in columns» match the names of the columns of the table whose values should be located in the columns of the pivot table (highlighted in green in Figure. 2.5.32);

«Fields in values» cmatch the names of the columns of the table whose values should be located in the values of the pivot table (highlighted in red in Figure. 2.5.32);

«Fields in rows» match the names of the columns of the table whose values should be located in the rows of the pivot table (highlighted in blue in Figure. 2.5.32);



PivotTable Fields

Choose fields to add to report:

☑ Package Card Number
☑ Business area
☑ Date of creation
□ Status Description
□ Status Change Date
□ Status Change Time
□ Package Comment

Drag fields between areas below:

Filters:

Columns: Business area

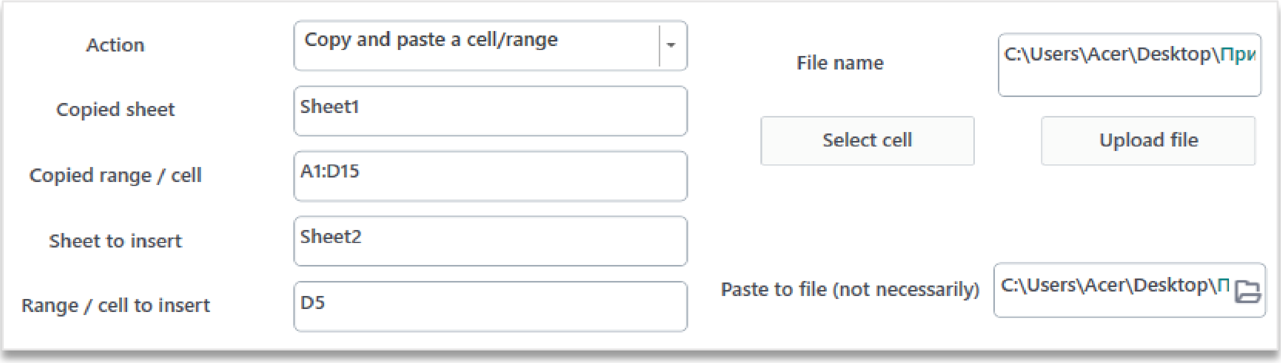
Rows: Date of creation; Package Card Number

Values: Sum of Violation of status

Figure. 2.5.32. Pivot table

Action «Copy and paste cell/range»

This action copies files and formatting from one file and inserts it to another. You can «stretch» formulas using this action (see. «Action «Formula»).



Action: Copy and paste a cell/range

Copied sheet: Sheet1

Copied range / cell: A1:D15

Sheet to insert: Sheet2

Range / cell to insert: D5

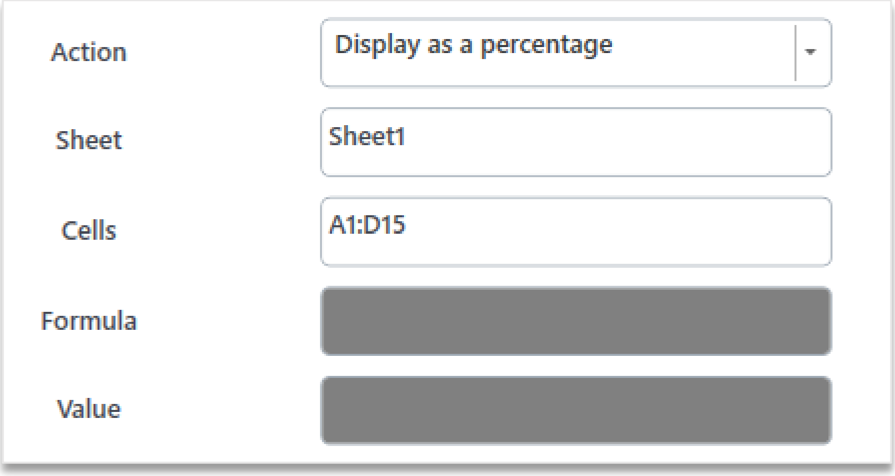
File name: C:\Users\Acer\Desktop\При

Paste to file (not necessarily): C:\Users\Acer\Desktop\П

Figure. 2.5.33. Copy/paste cell/range

Action «Show as percentage»

Action converts the selected range into percentage.



Action: Display as a percentage

Sheet: Sheet1

Cells: A1:D15

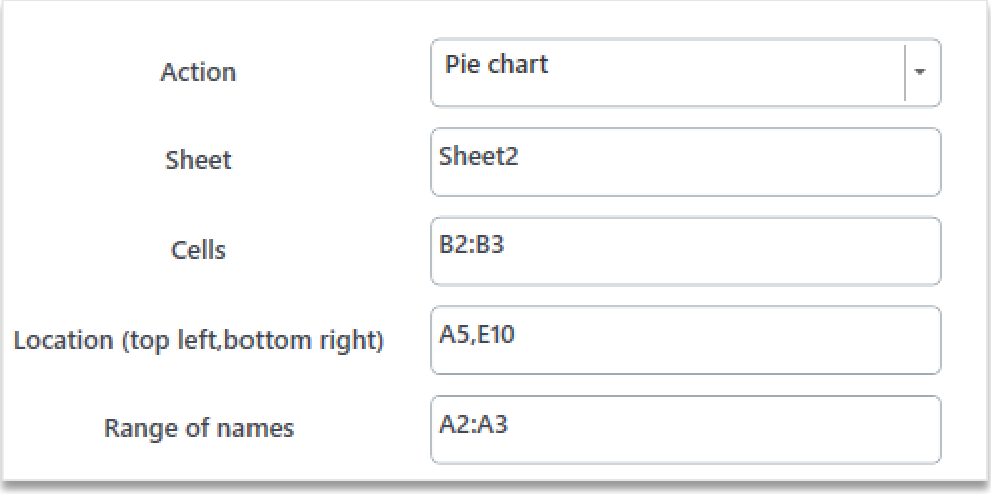
Formula:

Value:

Figure. 2.5.34. Show as percentage

Action «Pie chart»

Action creates a pie chart on a specified sheet. Values of the chart are taken from the range specified in the «Cells» field, a range with value names is indicated in the «range of names» field. In the «Location» field the borders of the location of the pie chart are indicated, especially the upper left corner and the lower right corner, separated by commas.

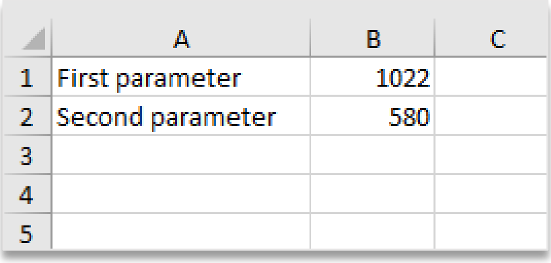


Action	Pie chart
Sheet	Sheet2
Cells	B2:B3
Location (top left,bottom right)	A5,E10
Range of names	A2:A3

Figure. 2.5.35. Creating a pie chart

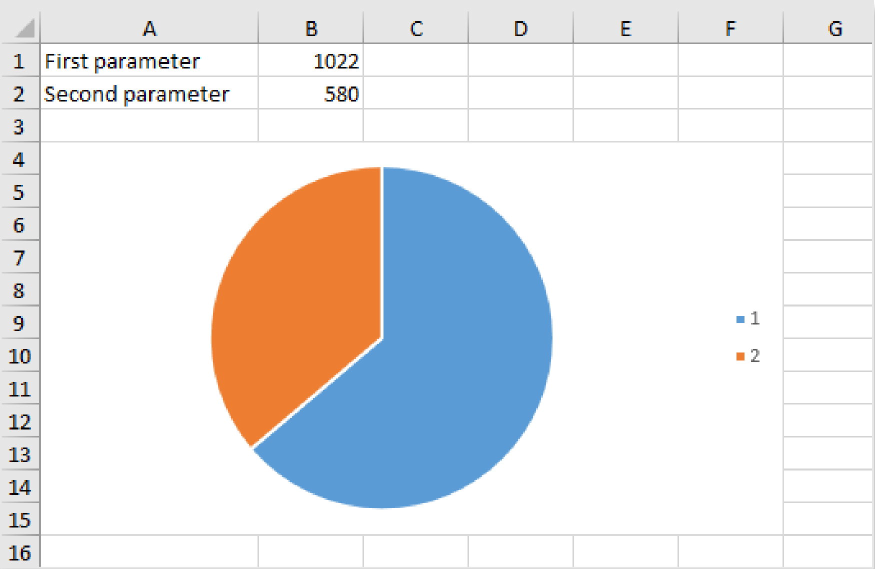
EXAMPLE.

The construction of a pie chart for the two parameters indicated in column A, according to the corresponding parameters indicated in column B. The completion of the action for this example is shown in Figure 2.5.35.



	A	B	C
1	First parameter	1022	
2	Second parameter	580	
3			
4			
5			

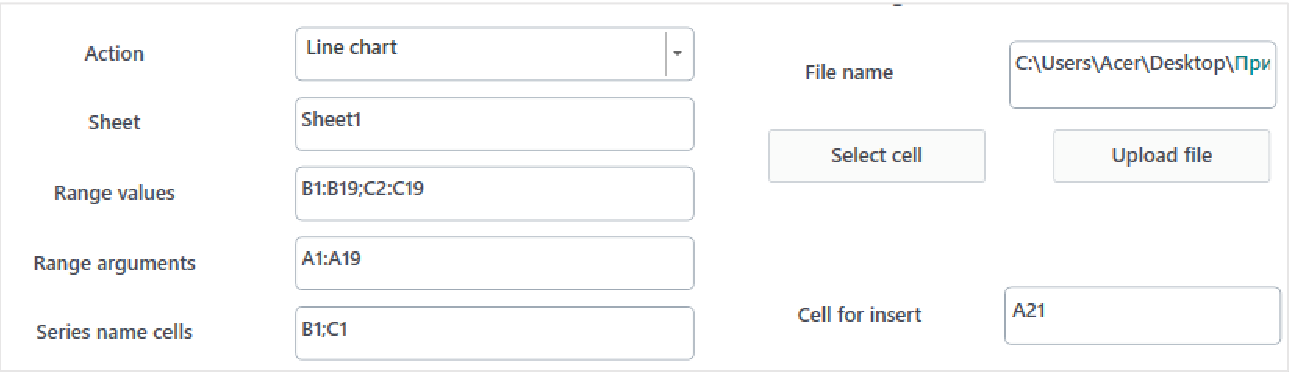
Result



	A	B	C	D	E	F	G
1	First parameter	1022					
2	Second parameter	580					
3							

Action «Graph»

The action creates a line graph with one or more rows.A row must be indicated in the «Range of values» field as a one-dimensional range (one column or one row). If there are more than one row, then following rows must be indicated through «;», as indicated in Figure. 2.5.35. There should be a single range of arguments, and must be indicated through «;». «Cells with row names» are filled with cell names through «;», their number should be equal to the number of rows, the first cell displays the name of the first row, etc. The cell in which the upper left corner of the graph will be located is recorded in the insertion cell.



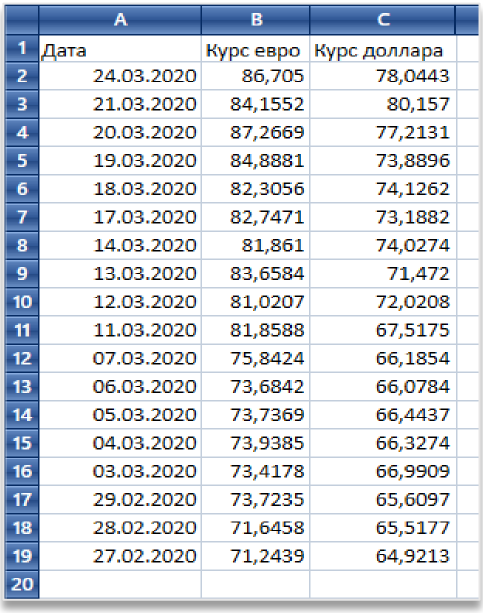
Action	Line chart
Sheet	Sheet1
Range values	B1:B19;C2:C19
Range arguments	A1:A19
Series name cells	B1;C1
File name	C:\Users\Acer\Desktop\При
Cell for insert	A21

FIgure. 2.5.36. Making a graph

EXAMPLE.

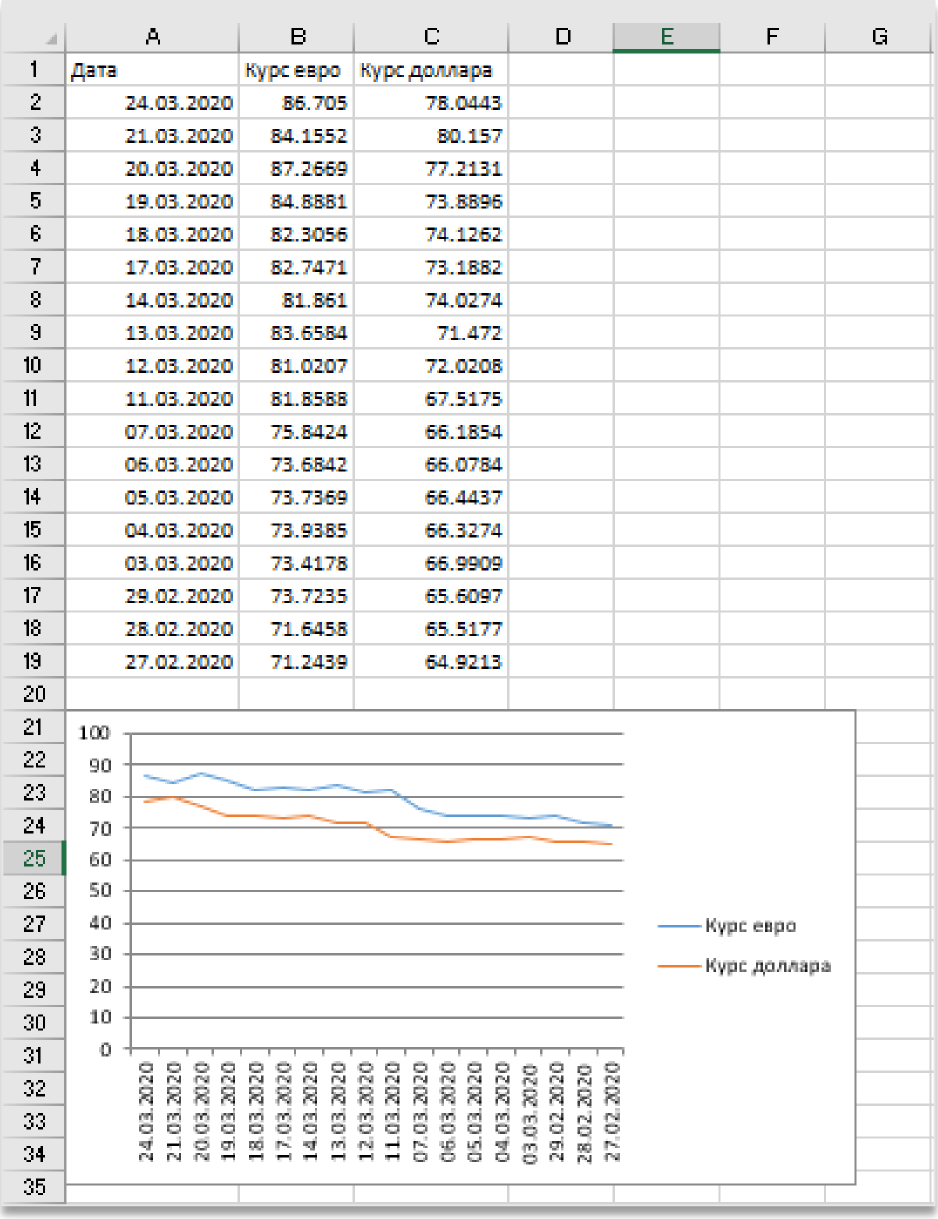
As an example, we’ll take euro to ruble, and dollar to ruble exchange rate for a certain period of time. The graph will consist of two rows, because there are two currencies. Dates will be our arguments - A, we’ll take 2 columns as values, first - B, euro to ruble exchange rate, and second - C dollar to ruble exchange rate. In Figure 2.5.35. you can see the completion of action.

Source data in the Excel file:



	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2669	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472
10	12.03.2020	81,0207	72,0208
11	11.03.2020	81,8588	67,5175
12	07.03.2020	75,8424	66,1854
13	06.03.2020	73,6842	66,0784
14	05.03.2020	73,7369	66,4437
15	04.03.2020	73,9385	66,3274
16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

The result:



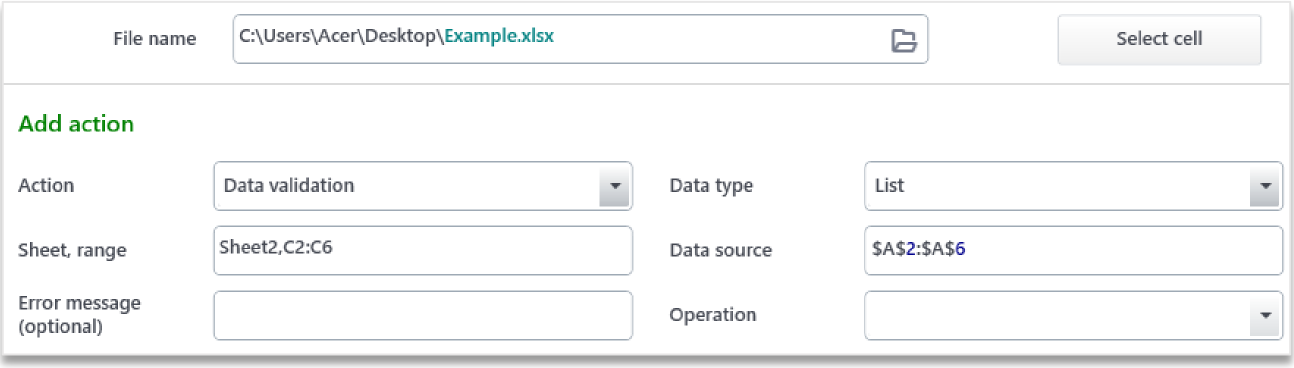
	A	B	C
1	Дата	Курс евро	Курс доллара
2	24.03.2020	86.705	78.0443
3	21.03.2020	84.1552	80.157
4	20.03.2020	87.2669	77.2131
5	19.03.2020	84.8881	73.8896
6	18.03.2020	82.3056	74.1262
7	17.03.2020	82.7471	73.1882
8	14.03.2020	81.861	74.0274
9	13.03.2020	83.6584	71.472
10	12.03.2020	81.0207	72.0208
11	11.03.2020	81.8588	67.5175
12	07.03.2020	75.8424	66.1854
13	06.03.2020	73.6842	66.0784
14	05.03.2020	73.7369	66.4437
15	04.03.2020	73.9385	66.3274
16	03.03.2020	73.4178	66.9909
17	29.02.2020	73.7235	65.6097
18	28.02.2020	71.6458	65.5177
19	27.02.2020	71.2439	64.9213

Action «Value check»

The action restricts the types of data that can be entered into selected cells by selecting the proposed rules. In the **«Sheet»** field, a sheet is entered, the cells of which you wanna restrict, in the **«Cells»** field, a range of cells is entered on which the values will be checked when entering. In the **«Data type»** field one of the offered data types is chosen – integer or real number, list, date, text length, time. **«Operation»** field consists of the list of available functions for creating a checking rule – equal, not equal, greater, less, greater or equal, less or equal, between and out. When using the «List» data type filling in the «Operation» field is not required, therefore you can’t edit it.

Last field changes depending on the data type – for the «List» data type - «Data source», for all others – «Criterias (through ;)». The **«Data source»** field is filled with one-dimensional range of values (range within a single column or single row), which will be specified in the list (if the list is static, then specify an absolute link). The **«Criteria (through ;)»** field is filled with values against which the verification rule will apply.

The «Error message» field is filled with the error text that will be displayed, if a value is entered in the cell that does not satisfy the rule set for it. This field is optional. If you leave it blank, then the following error message will appear: «The entered value is incorrect. The set of values that can be entered in a cell is limited».



File name	C:\Users\Acer\Desktop\Example.xlsx		Select cell

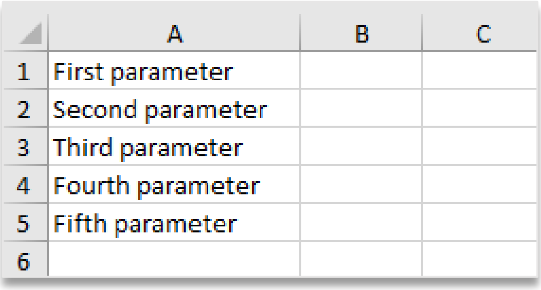
Add action

Action	Data validation	Data type	List
Sheet, range	Sheet2,C2:C6	Data source	$A$2:$A$6
Error message (optional)		Operation	

Figure. 2.5.36. Value check

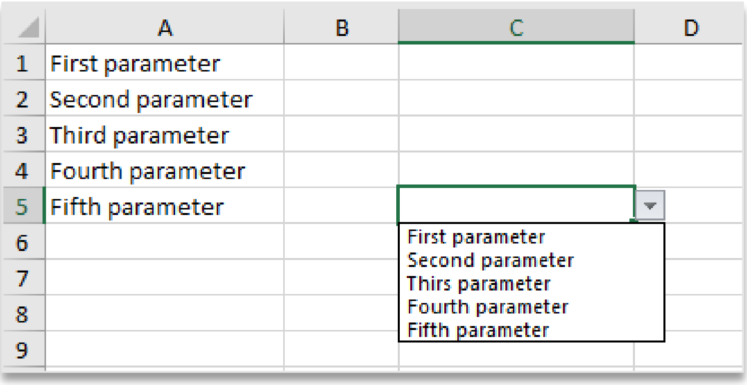
EXAMPLE.

To create a list that consists of values specified in any column, the action is filled, as shown on Figure. 2.5.36. If values A2:A6 are in cells:



	A	B	C
1	First parameter		
2	Second parameter		
3	Third parameter		
4	Fourth parameter		
5	Fifth parameter		
6			

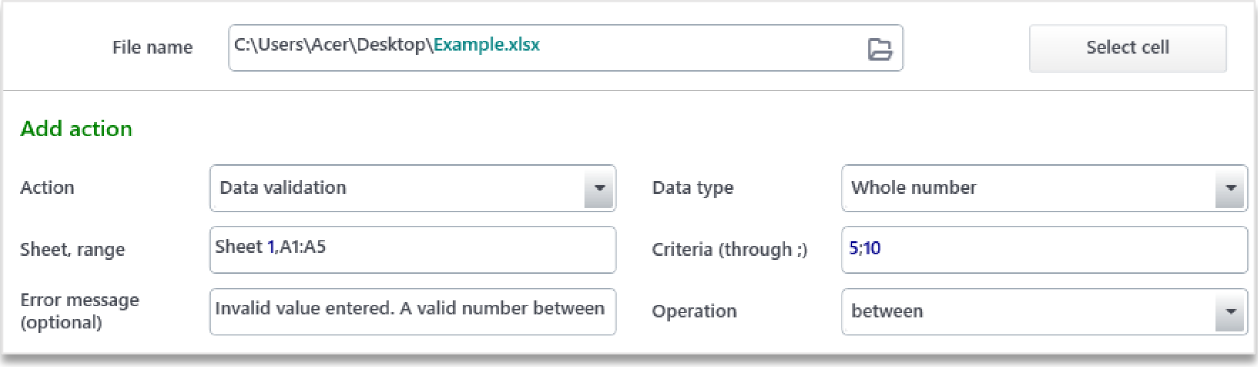
Then, after the robot runs, when you click on any cell from the range C2: C6, an arrow will appear that opens a drop-down list that will contain the selected values:



	A	B	C	D
1	First parameter			
2	Second parameter			
3	Third parameter			
4	Fourth parameter			
5	Fifth parameter			
6				
7				
8				
9				

First parameter
Second parameter
Thirs parameter
Fourth parameter
Fifth parameter

The second example is the creation of a rule that allows you to enter only real numbers between 5 and 10 in the range of cells A1: A5.Let an error be displayed when entering an incorrect value: «An invalid value has been entered. A valid number between 5 and 10 must be entered:»:

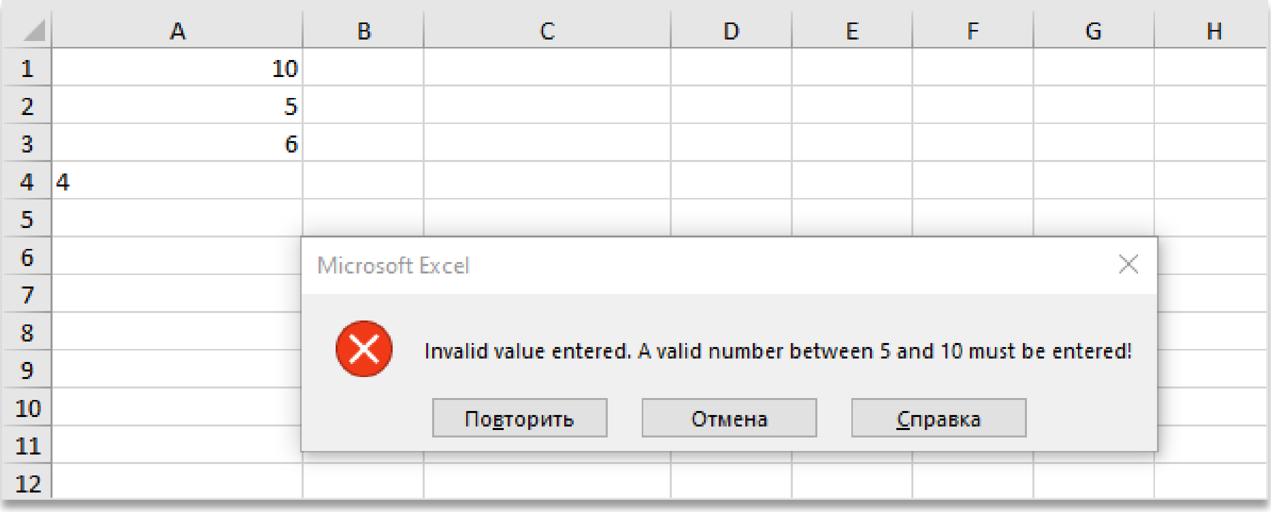


File name: C:\Users\Acer\Desktop\Example.xlsx

Add action

Action	Data validation	Data type	Whole number
Sheet, range	Sheet 1,A1:A5	Criteria (through ;)	5;10
Error message (optional)	Invalid value entered. A valid number between	Operation	between

After the robot runs, after trying to enter a number that is outside the range of 5 to 10, an error appears:



	A	B	C	D	E	F	G	H
1	10							
2	5							
3	6							
4	4							
5								
6								
7								
8								
9								
10								
11								
12								

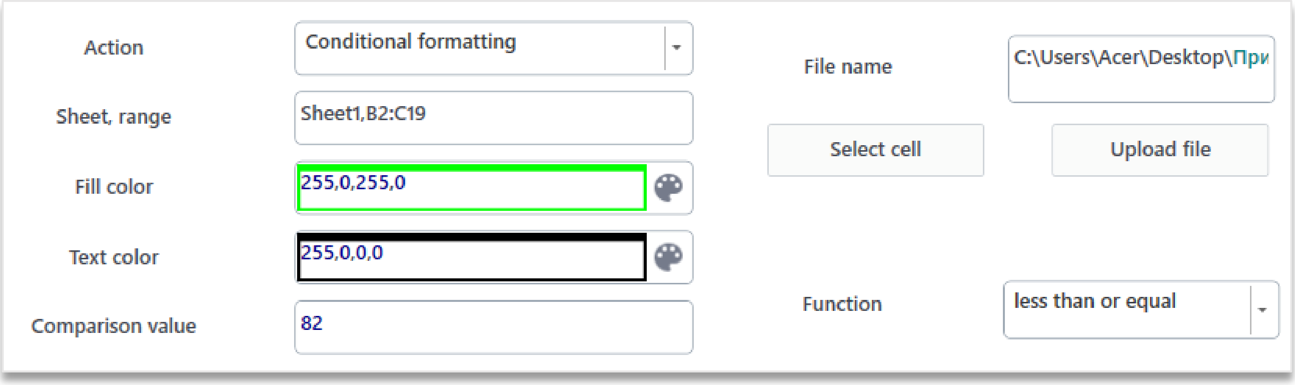
Microsoft Excel

Invalid value entered. A valid number between 5 and 10 must be entered!

«Formatting» group

Action «Conditional formatting»

The action formats the specified cells on the selected sheet, depending on whether they satisfy the specified condition. In the «Sheet, range» field, a sheet and range or a single cell from the sheet are entered, separated by commas, to which conditional formatting will be applied. The «Filling color» and «Text color» indicate the colors with which the cell will be filled and the text in it will be colored, if it satisfies the condition. The condition is made of the function selection and the indication of a value, for comparison. Available functions - equal, not equal, greater, less, greater or equal, less or equal, formula. If the «formula» function is chosen, then you need to enter a formula in Excel format (starting with an «=» symbol) in the «Comparison value» field.



Action: Conditional formatting

Sheet, range: Sheet1,B2:C19

Fill color: 255,0,255,0

Text color: 255,0,0,0

Comparison value: 82

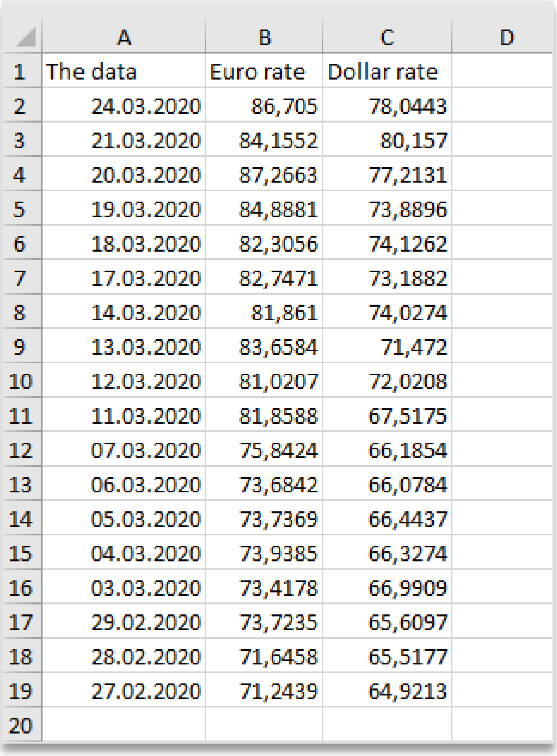
File name: C:\Users\Acer\Desktop\При

Function: less than or equal

Figure. 2.5.37. Conditional formatting

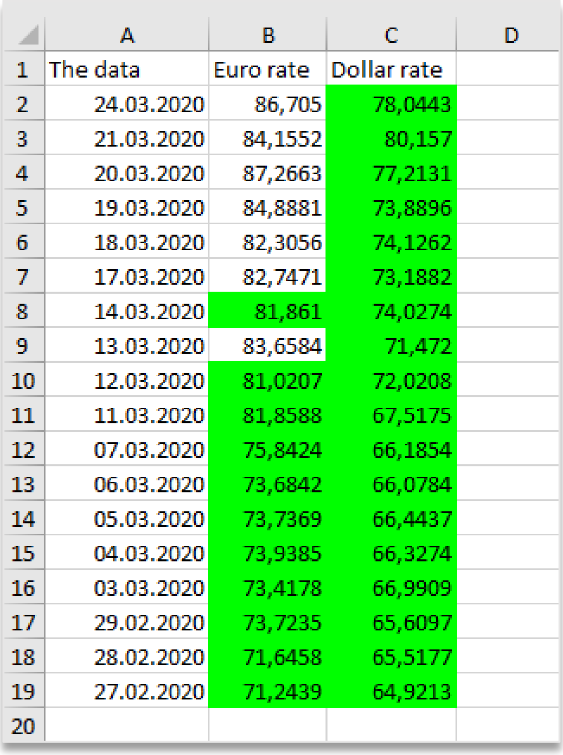
EXAMPLE.

Color all the cells in columns B and C green if the exchange rate is les than or equal to 82 . Action is completed as in Figure. 2.5.37. Before:



	A	B	C	D
1	The data	Euro rate	Dollar rate	
2	24.03.2020	86,705	78,0443	
3	21.03.2020	84,1552	80,157	
4	20.03.2020	87,2663	77,2131	
5	19.03.2020	84,8881	73,8896	
6	18.03.2020	82,3056	74,1262	
7	17.03.2020	82,7471	73,1882	
8	14.03.2020	81,861	74,0274	
9	13.03.2020	83,6584	71,472	
10	12.03.2020	81,0207	72,0208	
11	11.03.2020	81,8588	67,5175	
12	07.03.2020	75,8424	66,1854	
13	06.03.2020	73,6842	66,0784	
14	05.03.2020	73,7369	66,4437	
15	04.03.2020	73,9385	66,3274	
16	03.03.2020	73,4178	66,9909	
17	29.02.2020	73,7235	65,6097	
18	28.02.2020	71,6458	65,5177	
19	27.02.2020	71,2439	64,9213	
20				

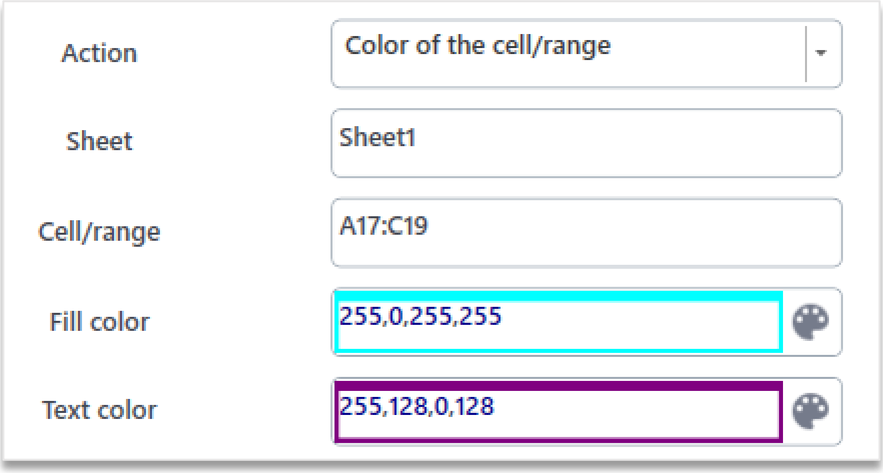
After:



	A	B	C	D
1	The data	Euro rate	Dollar rate	
2	24.03.2020	86,705	78,0443	
3	21.03.2020	84,1552	80,157	
4	20.03.2020	87,2663	77,2131	
5	19.03.2020	84,8881	73,8896	
6	18.03.2020	82,3056	74,1262	
7	17.03.2020	82,7471	73,1882	
8	14.03.2020	81,861	74,0274	
9	13.03.2020	83,6584	71,472	
10	12.03.2020	81,0207	72,0208	
11	11.03.2020	81,8588	67,5175	
12	07.03.2020	75,8424	66,1854	
13	06.03.2020	73,6842	66,0784	
14	05.03.2020	73,7369	66,4437	
15	04.03.2020	73,9385	66,3274	
16	03.03.2020	73,4178	66,9909	
17	29.02.2020	73,7235	65,6097	
18	28.02.2020	71,6458	65,5177	
19	27.02.2020	71,2439	64,9213	
20				

Action «Color of the cell/range»

The action changes the fill color of the text in the specified range/cell.



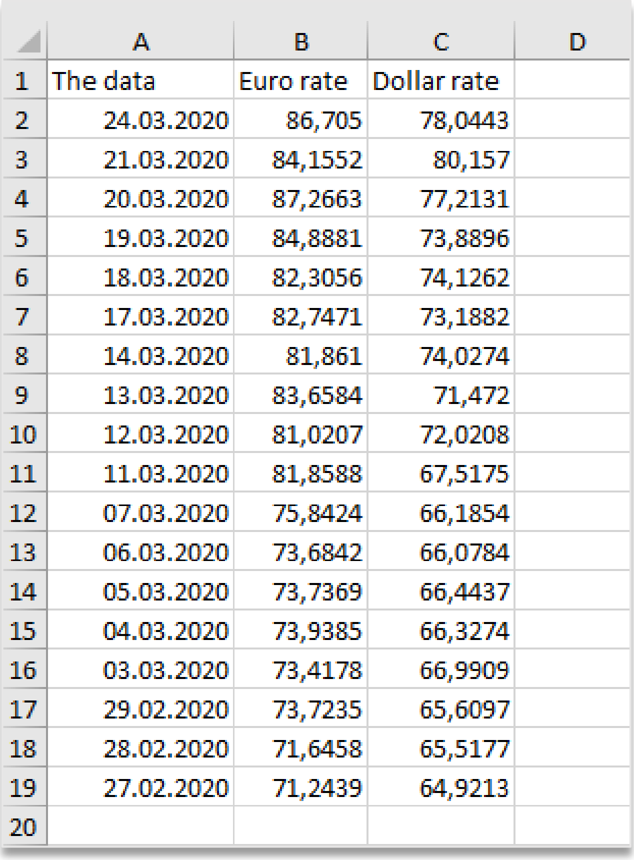
Action	Color of the cell/range
Sheet	Sheet1
Cell/range	A17:C19
Fill color	255,0,255,255
Text color	255,128,0,128

Figure. 2.5.38. Color of the cell/range

EXAMPLE.

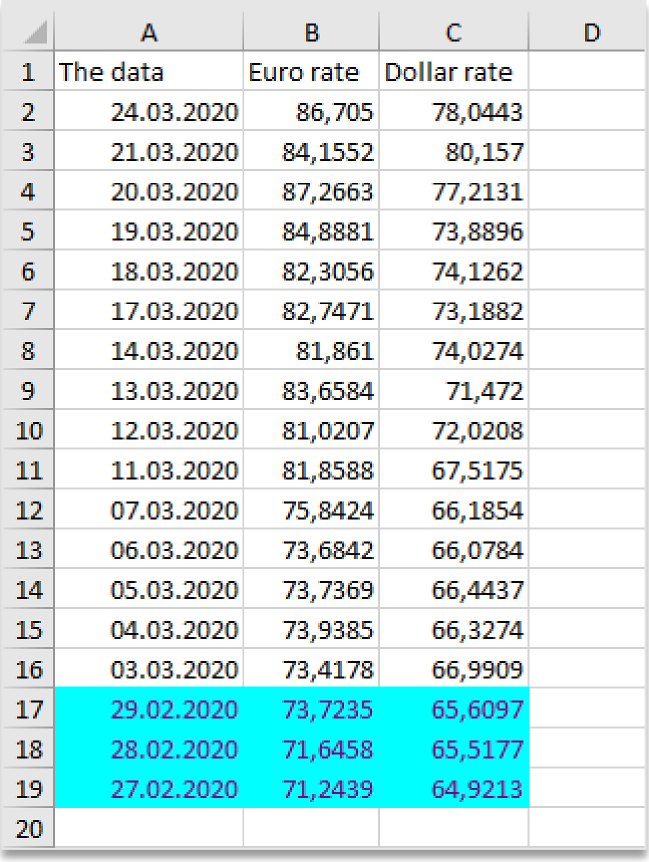
In the Figure. 2.5.38 it's demonstrated how to fill the A17:C19 range of cells with blue color, and the text in these cells with purple color.

Before:



	A	B	C	D
1	The data	Euro rate	Dollar rate	
2	24.03.2020	86,705	78,0443	
3	21.03.2020	84,1552	80,157	
4	20.03.2020	87,2663	77,2131	
5	19.03.2020	84,8881	73,8896	
6	18.03.2020	82,3056	74,1262	
7	17.03.2020	82,7471	73,1882	
8	14.03.2020	81,861	74,0274	
9	13.03.2020	83,6584	71,472	
10	12.03.2020	81,0207	72,0208	
11	11.03.2020	81,8588	67,5175	
12	07.03.2020	75,8424	66,1854	
13	06.03.2020	73,6842	66,0784	
14	05.03.2020	73,7369	66,4437	
15	04.03.2020	73,9385	66,3274	
16	03.03.2020	73,4178	66,9909	
17	29.02.2020	73,7235	65,6097	
18	28.02.2020	71,6458	65,5177	
19	27.02.2020	71,2439	64,9213	
20				

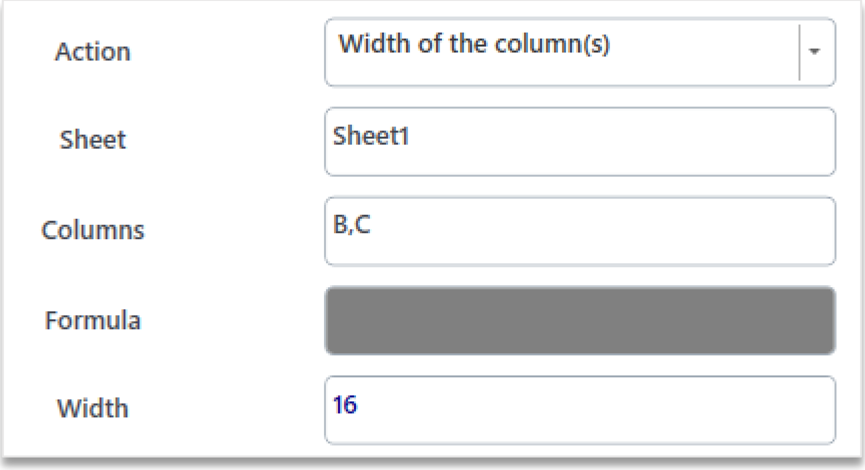
After:



	A	B	C	D
1	The data	Euro rate	Dollar rate	
2	24.03.2020	86,705	78,0443	
3	21.03.2020	84,1552	80,157	
4	20.03.2020	87,2663	77,2131	
5	19.03.2020	84,8881	73,8896	
6	18.03.2020	82,3056	74,1262	
7	17.03.2020	82,7471	73,1882	
8	14.03.2020	81,861	74,0274	
9	13.03.2020	83,6584	71,472	
10	12.03.2020	81,0207	72,0208	
11	11.03.2020	81,8588	67,5175	
12	07.03.2020	75,8424	66,1854	
13	06.03.2020	73,6842	66,0784	
14	05.03.2020	73,7369	66,4437	
15	04.03.2020	73,9385	66,3274	
16	03.03.2020	73,4178	66,9909	
17	29.02.2020	73,7235	65,6097	
18	28.02.2020	71,6458	65,5177	
19	27.02.2020	71,2439	64,9213	
20				

Action «Width of column(s)»

Action changes the width of the specified columns. Columns can be separated by a comma, you can put in their numbers instead of their names (starting from 1). Width is specified using the same unit of measure as in Excel.



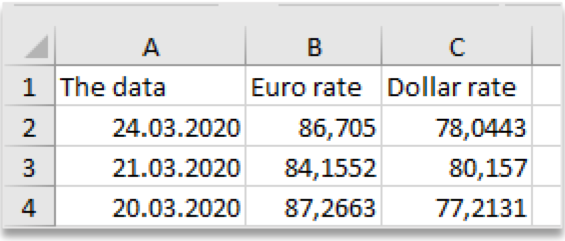
Action	Width of the column(s)
Sheet	Sheet1
Columns	B,C
Formula	
Width	16

Figure. 2.5.39. Width of column(s)

EXAMPLE.

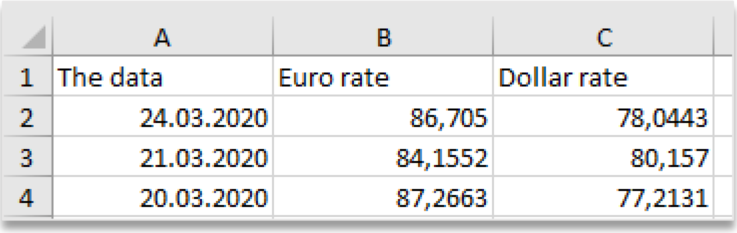
On the Figure. 2.5.39 it’s demonstrated how to fill in the action for setting the width of columns B and C in 16 rows. You can fill the «Columns» field with «2,3», meaning second and third tables.

Before:



	A	B	C
1	The data	Euro rate	Dollar rate
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2663	77,2131

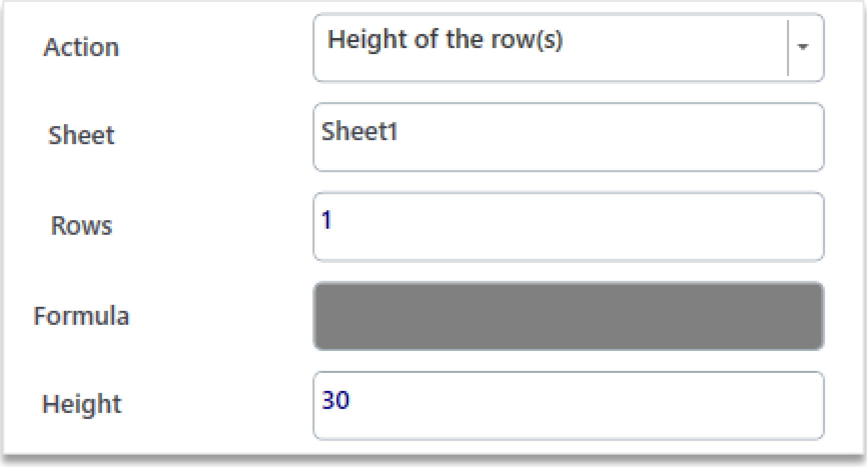
After:



	A	B	C
1	The data	Euro rate	Dollar rate
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2663	77,2131

Action «Height of row(s)»

Action changes the length of rows. Length is specified using the same unit of measure as in Excel.



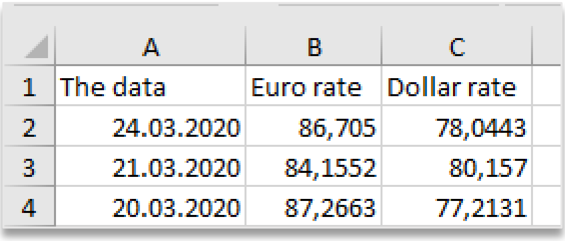
Action	Height of the row(s)
Sheet	Sheet1
Rows	1
Formula	
Height	30

Figure. 2.5.40. Length row(s)

EXAMPLE.

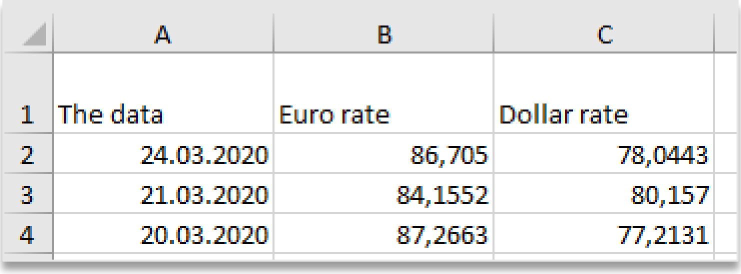
Let’s change the height of the first row, let's make it 30 points. To do that fill the action like in the Figure 2.5.40.

Before:



	A	B	C
1	The data	Euro rate	Dollar rate
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2663	77,2131

After:

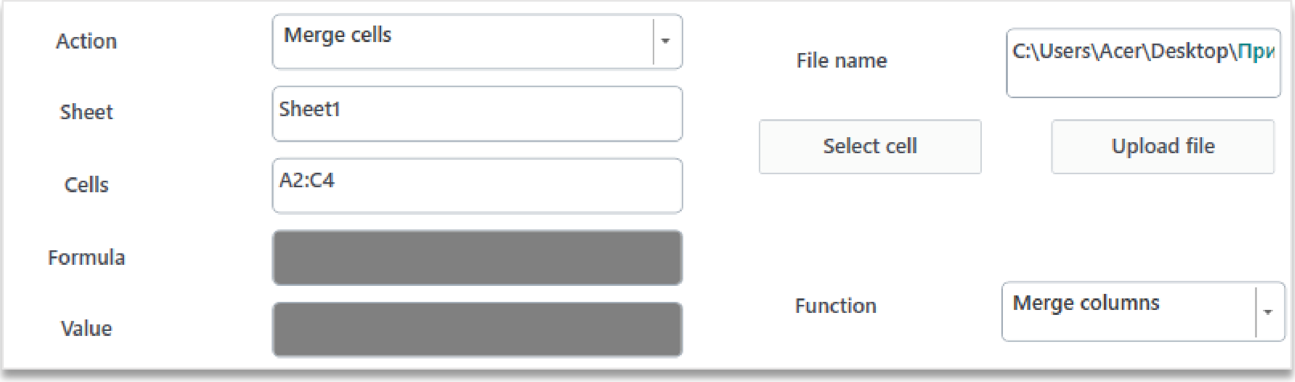


	A	B	C
1	The data	Euro rate	Dollar rate
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2663	77,2131

Action «Merging cells»

Action merges cells on the selected sheet. Module has 3 types of merging:

- simple merging – merges columns as well as range columns;
- row merging;
- column merging.



Action: Merge cells

Sheet: Sheet1

Cells: A2:C4

Formula:

Value:

File name: C:\Users\Acer\Desktop\При

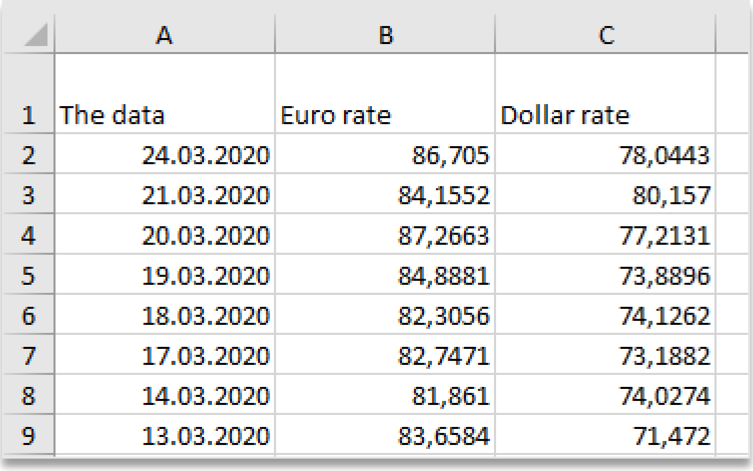
Function: Merge columns

Figure. 2.5.41. Merge cells

EXAMPLE.

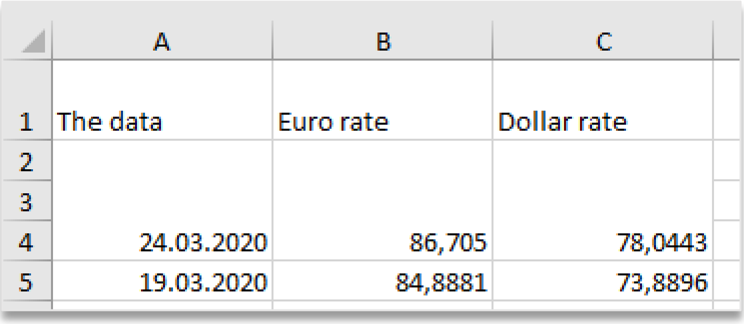
On Figure. 2.5.41 it’s demonstrated how to fill in the action to merge columns in a range of cells A2:C4.

Before:



	A	B	C
1	The data	Euro rate	Dollar rate
2	24.03.2020	86,705	78,0443
3	21.03.2020	84,1552	80,157
4	20.03.2020	87,2663	77,2131
5	19.03.2020	84,8881	73,8896
6	18.03.2020	82,3056	74,1262
7	17.03.2020	82,7471	73,1882
8	14.03.2020	81,861	74,0274
9	13.03.2020	83,6584	71,472

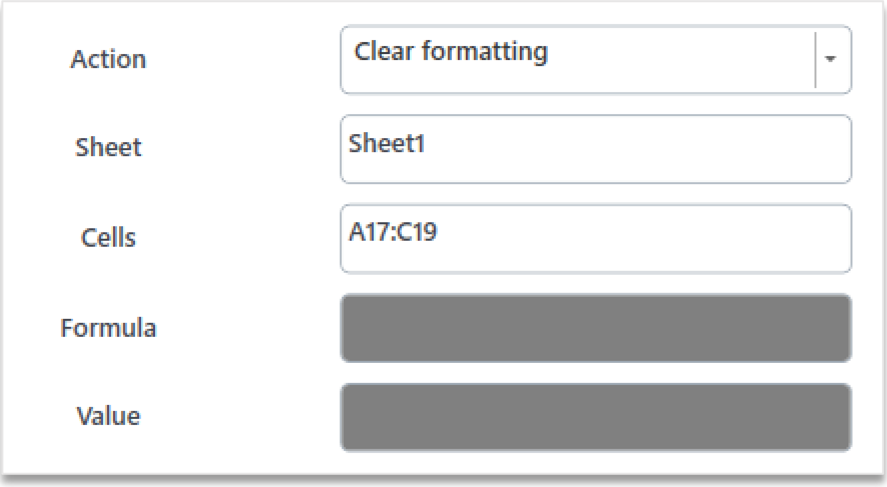
After:



	A	B	C
1	The data	Euro rate	Dollar rate
2			
3			
4	24.03.2020	86,705	78,0443
5	19.03.2020	84,8881	73,8896

Action «Clear formatting»

Action clears the formatting in the specified range of the selected sheet..



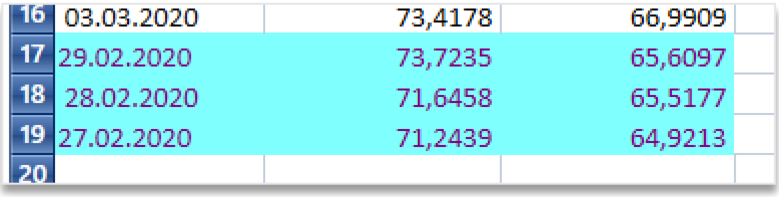
Action	Clear formatting
Sheet	Sheet1
Cells	A17:C19
Formula	
Value	

Figure. 2.5.42. Clear formatting

EXAMPLE.

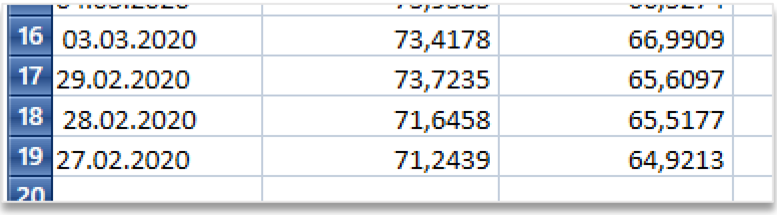
Let’s clear formatting in a file that is in the range A17: C19. To do that fill in the action as in Figure 2.5.42 and run the robot.

Before:



16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

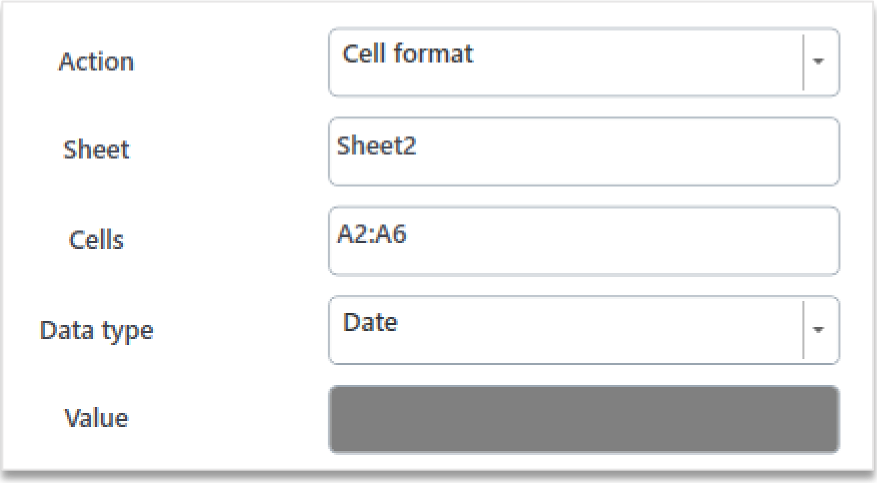
After:



16	03.03.2020	73,4178	66,9909
17	29.02.2020	73,7235	65,6097
18	28.02.2020	71,6458	65,5177
19	27.02.2020	71,2439	64,9213
20			

Action «Cell format»

Action changes the format of the specified cell. By default, all values entered in Excel by the studio have a string format, therefore, for the correct calculation of formulas or sorting, it is necessary to change the format of the data to your desired one.

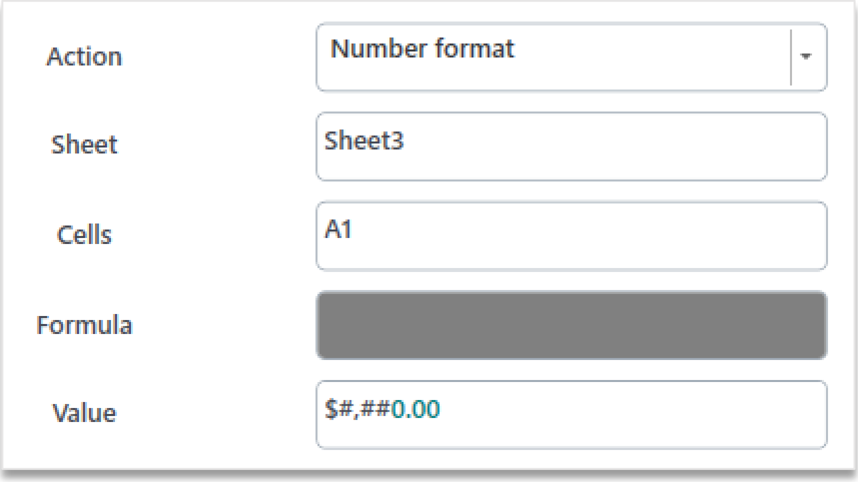


Action	Cell format
Sheet	Sheet2
Cells	A2:A6
Data type	Date
Value	

Figure. 2.5.43. Cell format

Action «Number format»

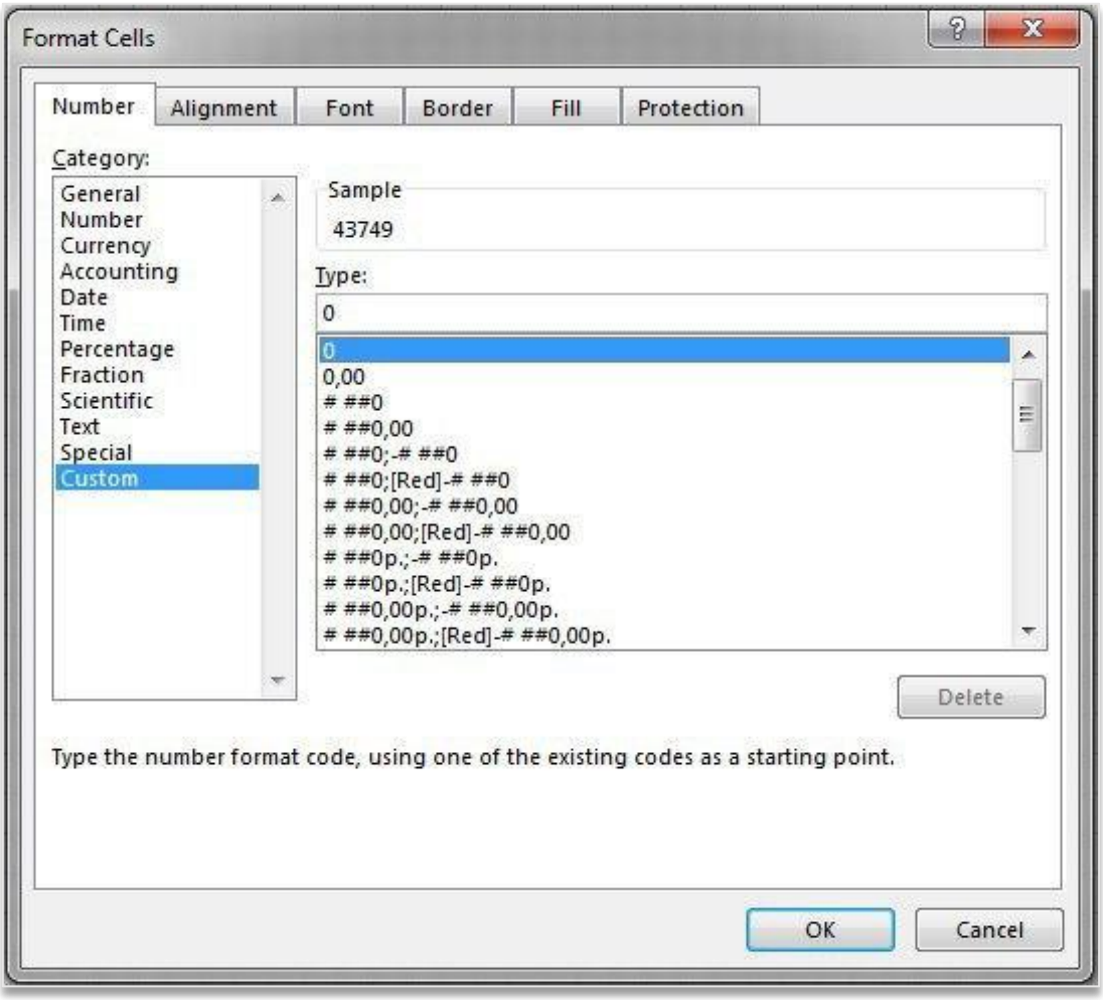
The action displays the number according to certain rules of the Excel number format



Action	Number format
Sheet	Sheet3
Cells	A1
Formula	
Value	$#,##0.00

Figure. 2.5.44. Number format

You can view all formats in Excel, just right click on a cell – «Cell format» – tab «Number» - (all formats):



Category:

General
Number
Currency
Accounting
Date
Time
Percentage
Fraction
Scientific
Text
Special
Custom

Sample
43749

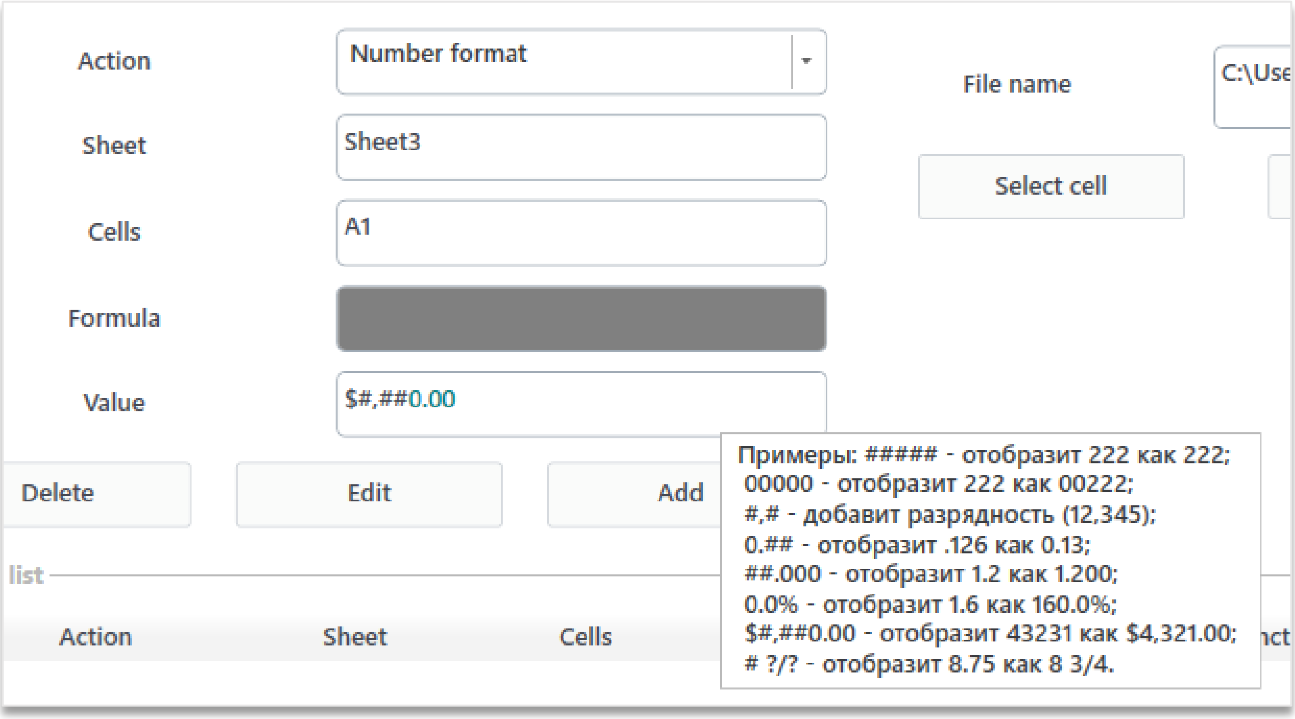
Type:
0

```
0
0,00
# ##0
# ##0,00
# ##0;-# ##0
# ##0;[Red]-# ##0
# ##0,00;-# ##0,00
# ##0,00;[Red]-# ##0,00
# ##0р.;-# ##0р.
# ##0р.;[Red]-# ##0р.
# ##0,00р.;-# ##0,00р.
# ##0,00р.;[Red]-# ##0,00р.
```

Type the number format code, using one of the existing codes as a starting point.

Figure. 2.5.45. Number format

The most commonly used formats are suggested if you hover your cursor over the field to enter a format.



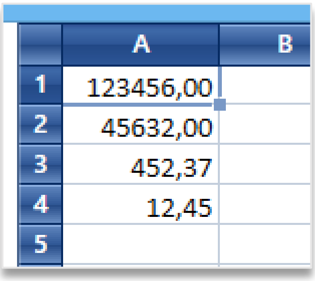
Action	Number format
Sheet	Sheet3
Cells	A1
Formula	
Value	$#,##0.00
File name	C:\Use

Примеры: ##### - отобразит 222 как 222;
00000 - отобразит 222 как 00222;
#,# - добавит разрядность (12,345);
0.## - отобразит .126 как 0.13;
##.000 - отобразит 1.2 как 1.200;
0.0% - отобразит 1.6 как 160.0%;
$#,##0.00 - отобразит 43231 как $4,321.00;
?/? - отобразит 8.75 как 8 3/4.

Figure. 2.5.46. Number format

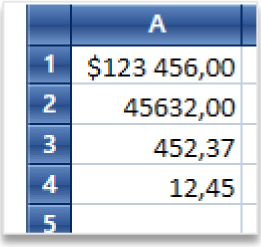
EXAMPLE.

Let’s format the number in the A1 cell to the dollar money format. To do this, fill out the action as shown in Figure 2.5.46.



	A	B
1	123456,00	
2	45632,00	
3	452,37	
4	12,45	
5		

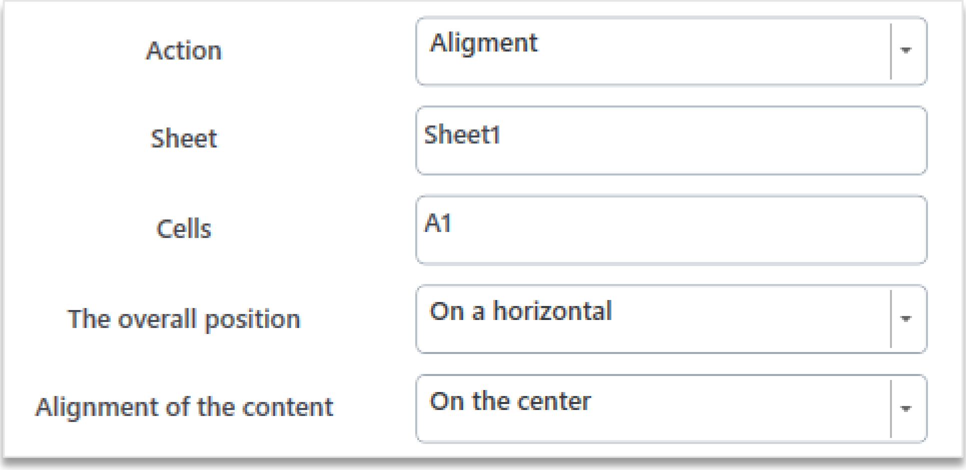
Result:



	A
1	$123 456,00
2	45632,00
3	452,37
4	12,45
5	

Action «Alignment»

The action applies the specified alignment in the specified cells. You can choose one of two things in the «General position» field - apply alignment vertically or horizontally. You can select the alignment in the «Align content» field. Three types of alignment are offered: by left/right edge, by the top/bottom edge and by the center

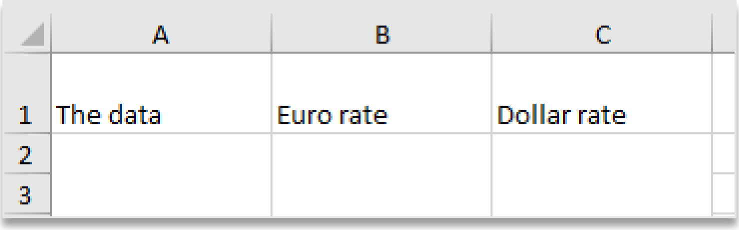


Action	Aligment
Sheet	Sheet1
Cells	A1
The overall position	On a horizontal
Alignment of the content	On the center

Figure. 2.5.47. Alignment

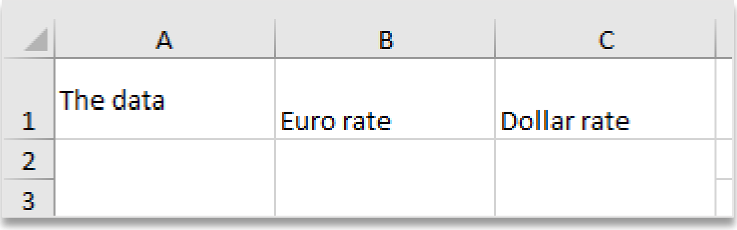
EXAMPLE.

In the filled out action in Figure 2.5.47, cell A1 is aligned by the center relative to the vertical. Original position of the text in the field:



	A	B	C
1	The data	Euro rate	Dollar rate
2			
3			

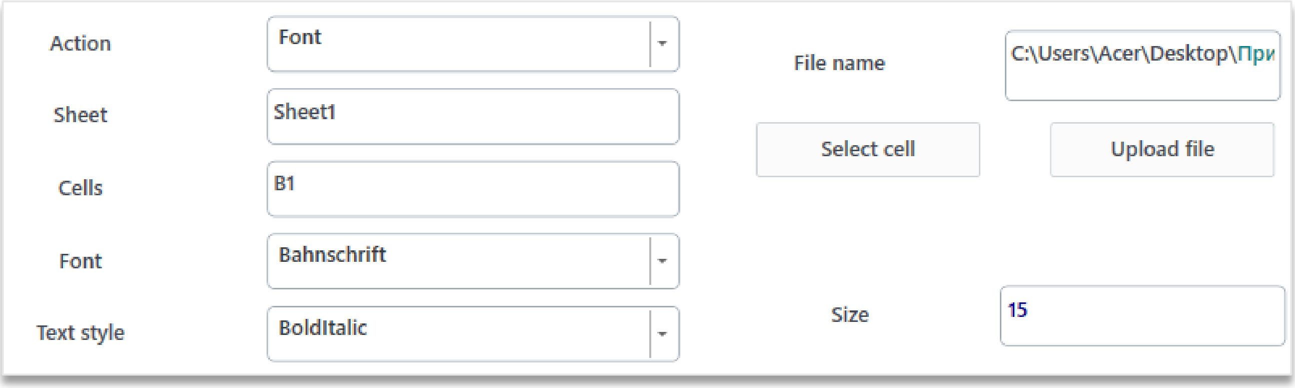
After running the robot:



	A	B	C
1	The data	Euro rate	Dollar rate
2			
3			

Action «Font»

Action changes the font, the style or the size of selected cells on the specified sheet. Module offers all default Microsoft Office fonts.

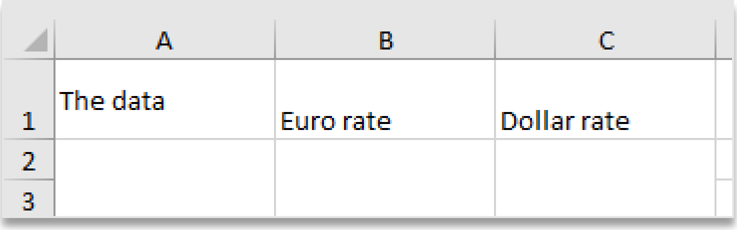


Action	Font
Sheet	Sheet1
Cells	B1
Font	Bahnschrift
Text style	BoldItalic
File name	C:\Users\Acer\Desktop\При
Size	15

Figure. 2.5.48. Changing the font

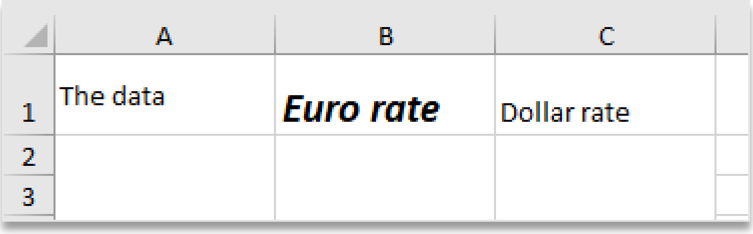
EXAMPLE.

Let’s change the font of the text in the B1 cell, fill out the action as shown in Figure. 2.5.48 – the font we’re gonna be using in this example is Batang, let’s set the text style – bold, italic, size 16. Original text formatting:



	A	B	C
1	The data	Euro rate	Dollar rate
2			
3			

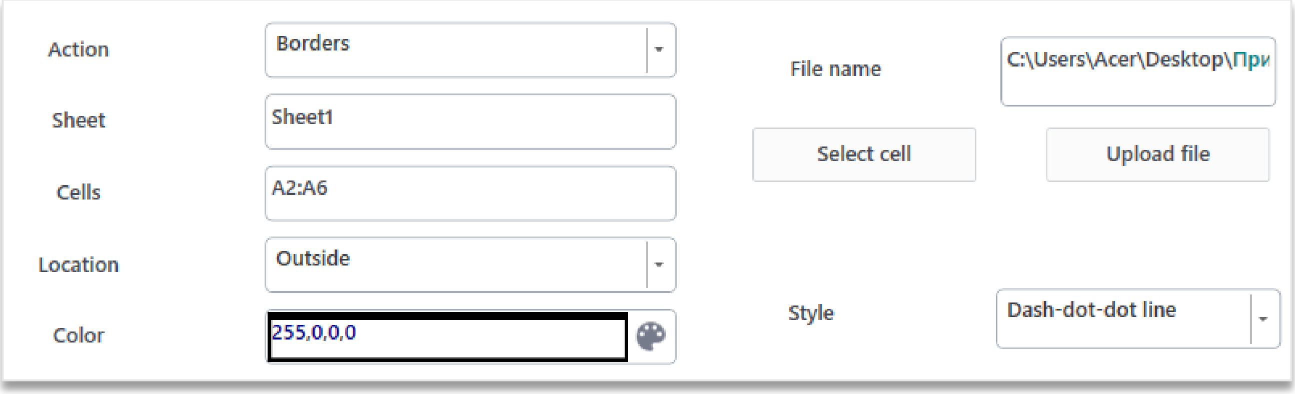
Result:



	A	B	C
1	The data	***Euro rate***	Dollar rate
2			
3			

Action «Borders»

Action adds or changes the cell borders of the selected cells on the specified sheet. This action involves configuring the borders - external, internal; only the right one, etc., the color of the borders and their style - thick, thin, dashed line, etc.



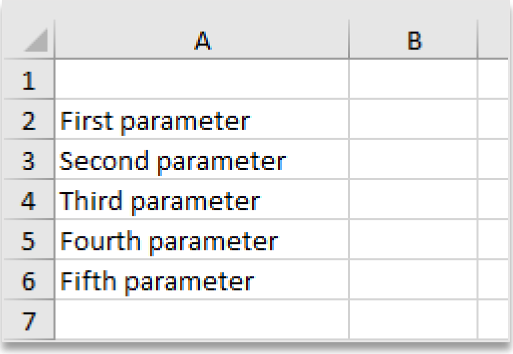
Action	Borders
Sheet	Sheet1
Cells	A2:A6
Location	Outside
Color	255,0,0,0
File name	C:\Users\Acer\Desktop\При
Style	Dash-dot-dot line

Figure. 2.5.49. Adding/changing borders

EXAMPLE.

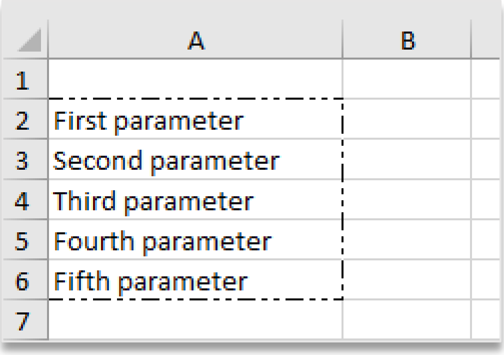
Add a dash-dotted outer border for a block of text in the range A2: A6. To do this you need to fill out the action like in the Figure. 2.5.49.

Original block of text:



	A	B
1		
2	First parameter	
3	Second parameter	
4	Third parameter	
5	Fourth parameter	
6	Fifth parameter	
7		

Result:



	A	B
1		
2	First parameter	
3	Second parameter	
4	Third parameter	
5	Fourth parameter	
6	Fifth parameter	
7		

PART III. «CLICKER» AND «IMAGE SEARCH» MODULES

Chapter 6. «Clicker» module

The «Clicker» is used to record a sequence of actions, performed by a user, those actions include mouse button clicks and typing on a keyboard. With the help of this module you can interact with any software, because this module mimics the actions of a user, meaning that if you can do this «Clicker» can too.

The use of clicker robots is recommended on monitors with the same resolution as the one a robot was written on.

Creating a macros

When you open the «Clicker» module the «Record macros actions» dialog window will open up. The dialog window consists of three elements – the «Start recording» button, the area in which the recorded actions will be displayed, and the «Save» button.

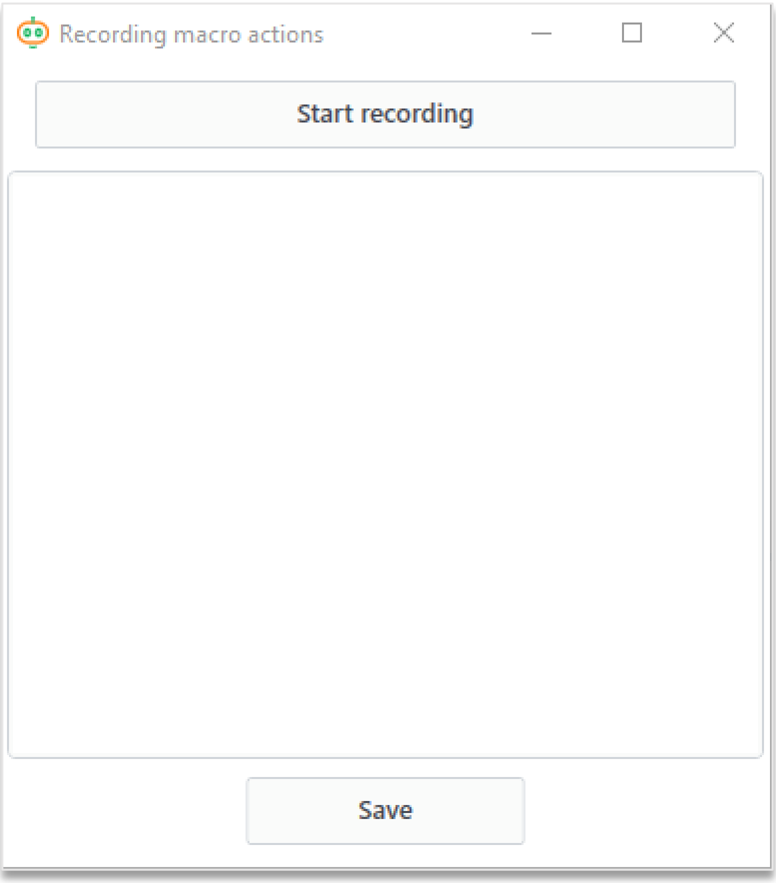
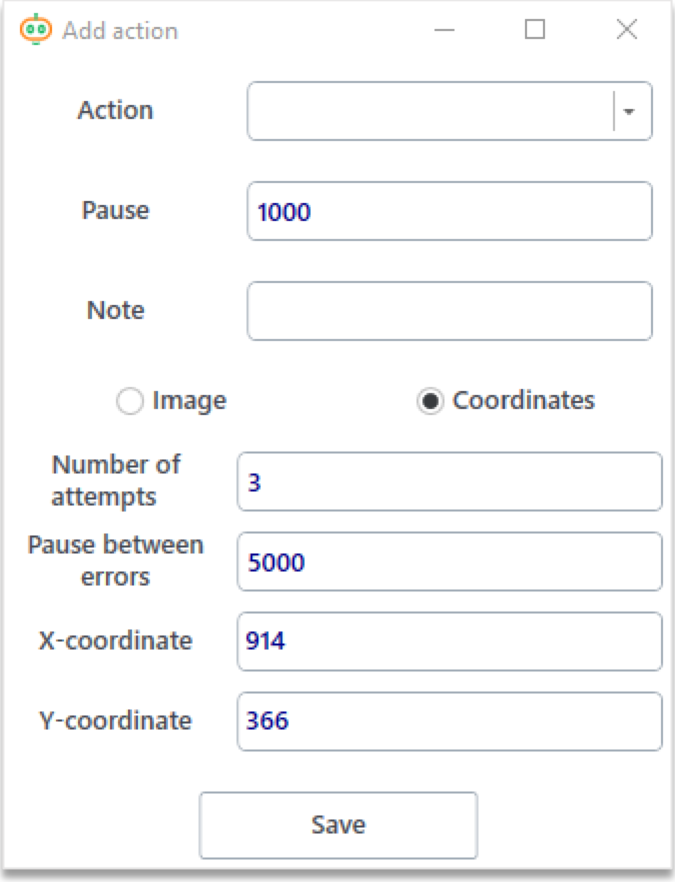


Figure. 3.6.1. Recording macros actions

By clicking the «Start recording» button, software will wait for you to press ***«TAB»***, after you press ***«TAB»*** the dialog window for making a new command (action) will open up.



Action

Pause 1000

Note

○ Image ● Coordinates

Number of attempts 3

Pause between errors 5000

X-coordinate 914

Y-coordinate 366

Figure. 3.6.2. Creating a command (action)

In this dialog window you can configure the command (detailed description of all actions is gonna be listed below). After you create a command and press the save button, a new action will be added to the macros.

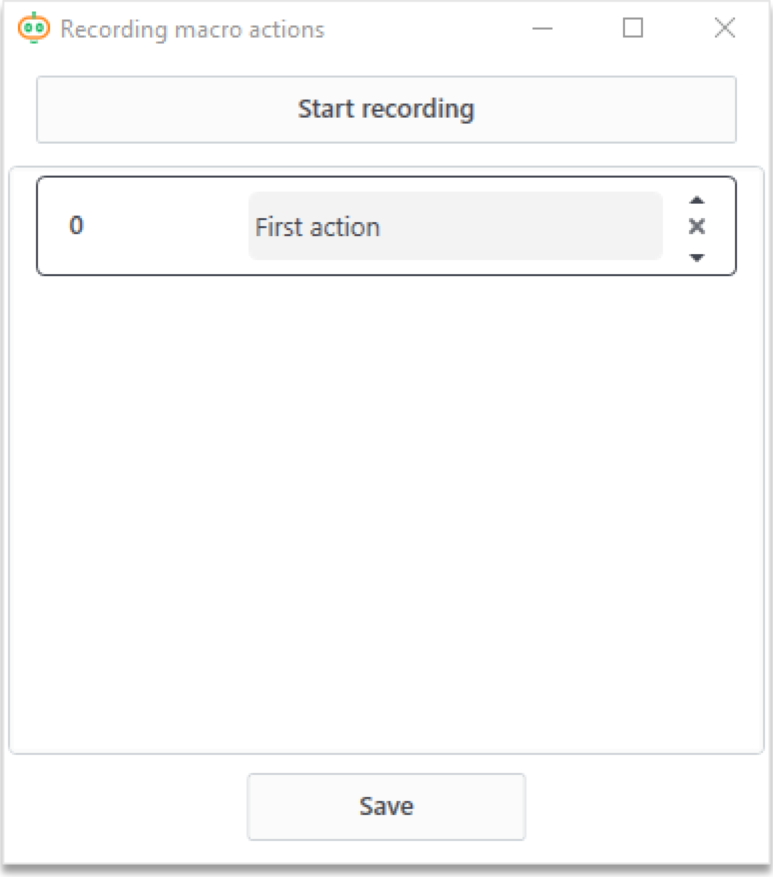


Figure. 3.6.3. Adding a command

You can change the position of an action in the list of commands using the arrow buttons located on the right side of the block (Figure 3.6.4).

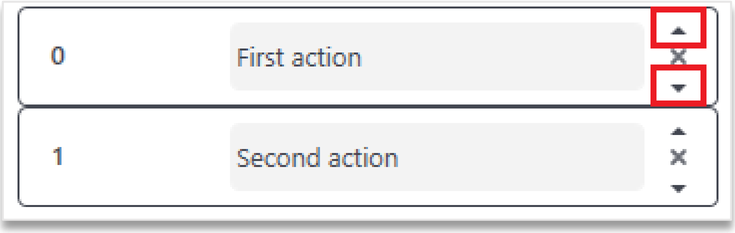


Figure. 3.6.4. Changing the order of commands

A button with a cross on it is used to remove an action from the list of commands.

To save the macros press the «Save» button.

Creating a command

Window for creating a command consists of static and non static fields. «Action», «Pause» and «Note» are static fields.

Static fields

Action

The «Action» field consists of a list of available actions for recording a macros – typing and clicking, most common hotkeys (copy, paste). All of the available actions in this module will be listed below.

Pause

The «Pause» field is designed for creating a pause after an executed command. The pause is specified in milliseconds.

For example, after clicking the browser icon robot needs to wait until the browser will launch – in this example, in the command for opening a browser (left click on its icon) you should enter 5000 (5 seconds) in the «Pause» field.

Note

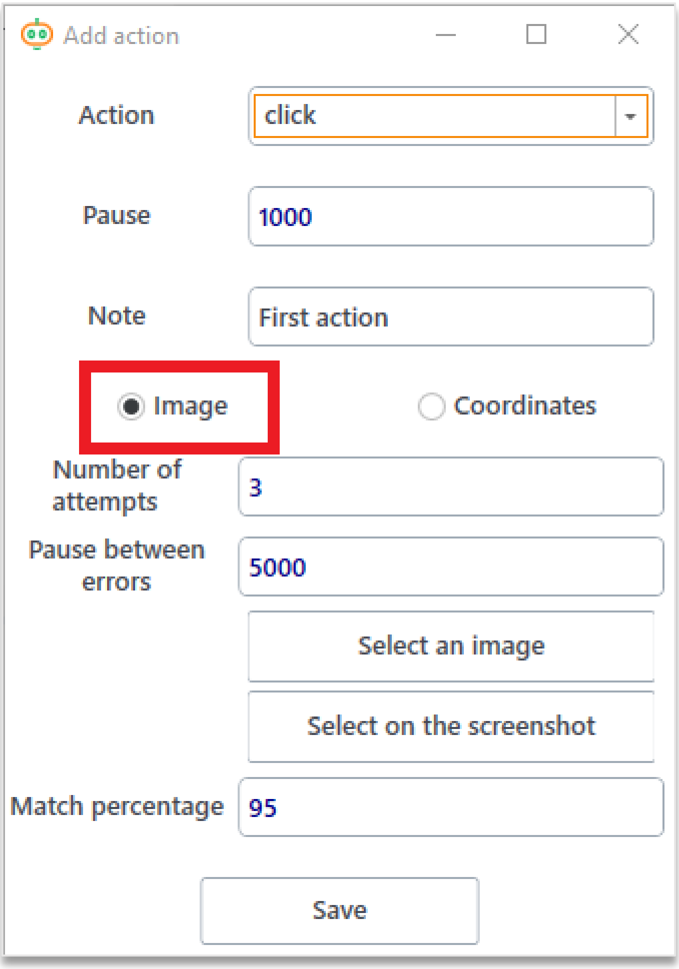
The «Note» field allows you to write comments so you don't get confused in the commands.

Fields that appear only with certain actions

Coordinates or image

When creating actions associated with a mouse click, you are asked to choose where to click - according to certain coordinates or by image. If a software you're working with has a static interface, meaning the button locations don't change, then you can use coordinates. Then the question comes up, "How to calculate coordinates?". It's really simple – when you press «TAB» you need to locate your cursor to the place you wanna click. The coordinates will be entered automatically to the dialog window.

But if a software doesn't have a static interface we highly recommend using the click on image feature. To do that just click the «Image» switch in the dialog window (Figure. 3.6.5.).



Add action

Action: click

Pause: 1000

Note: First action

● Image ○ Coordinates

Number of attempts: 3

Pause between errors: 5000

Match percentage: 95

Figure. 3.6.5. Creating a command

Depending on a switch selected the bottom part of the window will change – the «Select image», «Highlight image» buttons and «Match rate» input field will appear.

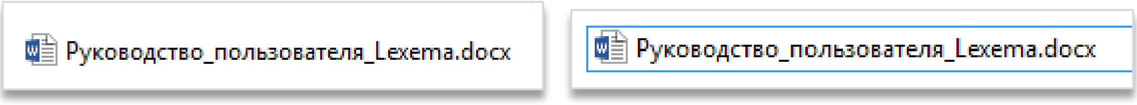
The image that you need to click on to run the action can be configured in two ways:

1) load an existing image by clicking the «Select image» button

2) Use the «HIghlight image» button to select a fragment of the window snapshot. After clicking the button Studio will take a screenshot and will offer you to select a fragment of a resulting image.

Keep in mind that if you use the “click on image” feature, that if there are multiple images on the screen the robot will apply the action to the first image found.

The «Match rate» field is necessary for the cases, where the selected image slightly differs from the reality (for example, if on the image you highlighted the button in some way. but in reality the button is unhighlighted). In these kinds of cases you need to lower the match rate to the point when the robot will be able to find the image. But if you make the match rate really low, the robot might find the thing you weren’t looking for.



Руководство_пользователя_Lexema.docx

Руководство_пользователя_Lexema.docx

Figure. 3.6.6. An example of slightly different images

Fields «Number of attempts» and «Pause between errors» are designed for when robot fails to find the image. You can specify how many times the robot should try to search for the desired image and how many seconds to wait between attempts. For example, after launching the browser, robot needs to press the «Plus» button to create a new tab. If browser launch time is gonna be longer than 5 seconds (this time is specified in the pause between actions field), then Studio will try to look for the «Plus», that's not there yet. In that case the Studio will wait the number of seconds you specified between attempts and try again. As soon as the image is found, the program will wait the number of seconds you specified in the «Pause» field of the command, and proceed to the next.

Scrolling

This field appears when the action «Scrolling» is applied. The amount of pixels you need to scroll down for is entered to the field. If you need to scroll up, the number of pixels should be positive, but if you need to scroll down, the number of pixels should be negative.

The use of variables in all command creation dialog windows is possible.

Description of all actions available in the module

Action «Click the mouse»

The main purpose of this module is pressing the left or right mouse button.

Action «Double-click the mouse»

Double-clicks on the specified coordinates or image, for example, to open applications from the desktop.

Action «Click the right mouse button»

Clicks the right mouse button, for example, to open a context menu.

Action «Click the scroll wheel»

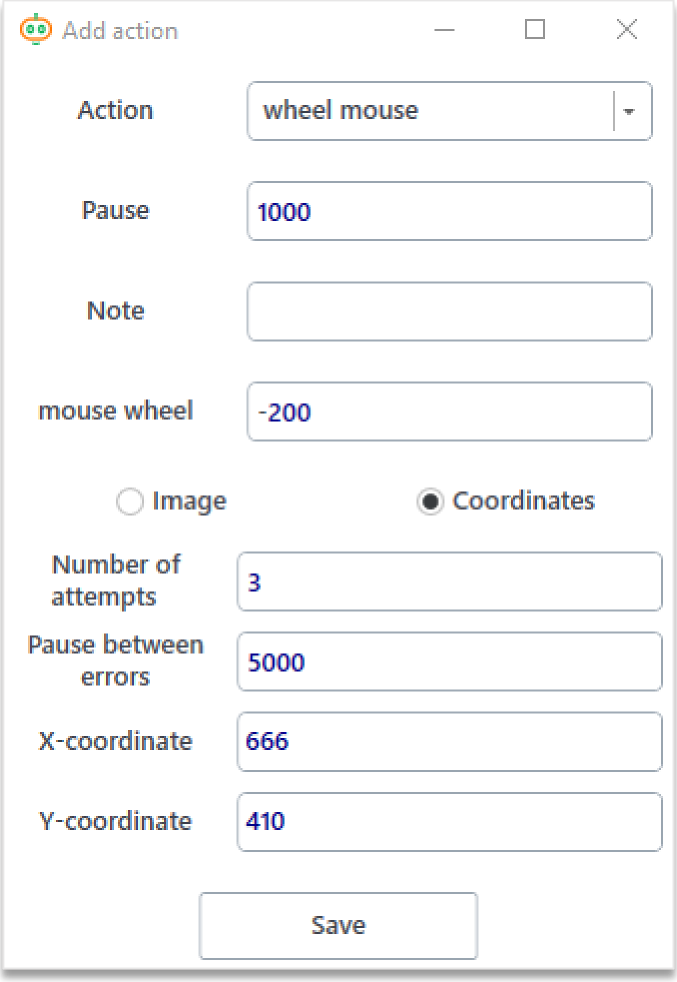
In some applications mouse wheel is used not only to scroll, but, for example to close the tabs. Action clicks the scroll wheel.

Action «Scrolling»

The action scrolls the page up or down. Since there might be multiple scrolling sliders on the screen you need to position your cursor to the one you need. To scroll down you need to put in a negative number of pixels in the «Scrolling» field (-150), to scroll up you need to put in a positive number (150).

EXAMPLE.

To scroll down a particular slider, you need to hover over it, press TAB, select the «Scrolling» action and fill in the «Scrolling» field:



Add action

Action	wheel mouse
Pause	1000
Note	
mouse wheel	-200

○ Image ● Coordinates

Number of attempts	3
Pause between errors	5000
X-coordinate	666
Y-coordinate	410

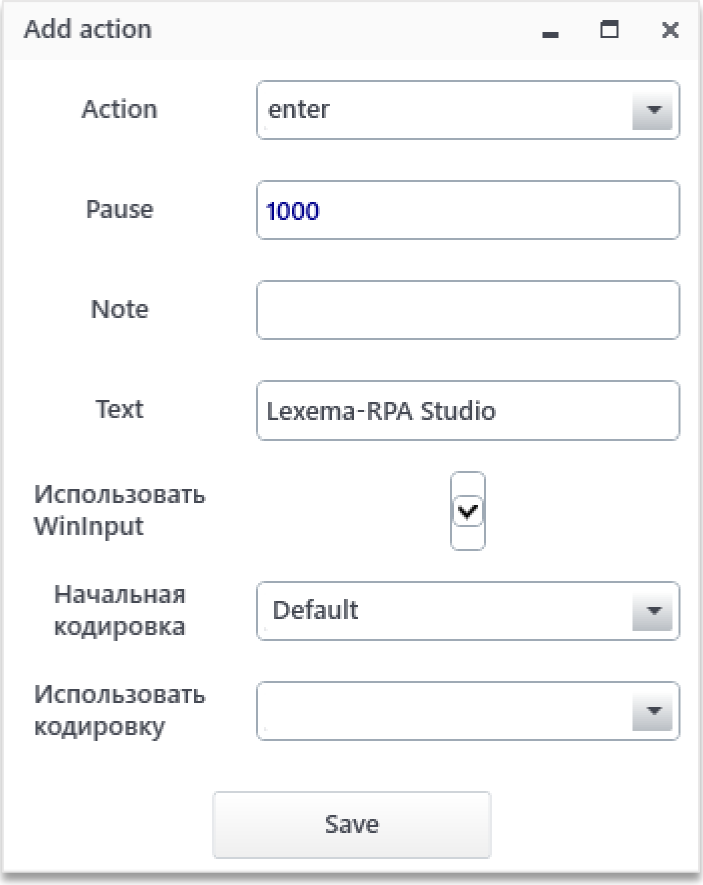
Action «Enter text»

This action enters text that is written in the «Text» field. The field where the text is entered should be selected (there should be focus).

Some applications require specific text input (in the specific encoding). In these cases you should pick one of the encoders from the “Use encoder” drop-down list or enter the name of the desired encoder yourself. The text can be in a variable and be copied from any program that uses the specified encoding. In this case, when pasting copied text into another program, you must first scan it in the encoding of the original program. The encoding of the source program is entered or selected in the «Initial coding» field. **Filling these fields is optional.**

The «Use WinInput» checkbox allows you to enable more correct and faster text input, but it may not be supported by some applications. **The use of this checkbox is optional.**

Example of filling in an action:



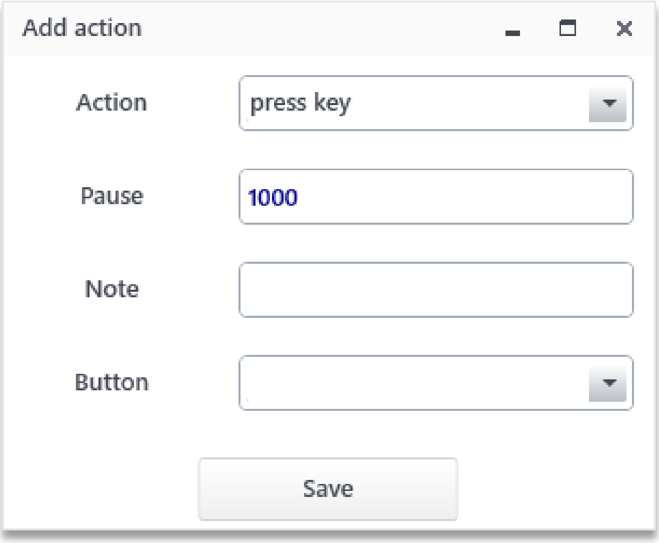
Add action

Action	enter
Pause	1000
Note	
Text	Lexema-RPA Studio
Использовать WinInput	☑
Начальная кодировка	Default
Использовать кодировку	

Figure. 3.6.7. Example of filling in an action:

Action «Press key»

Imitates the key presses of a keyboard. Choose the desired key in the «Key» drop-down list. All possible keys are available on the list, keys that might be present on some extended keyboards are present too (changing the volume, start/stop playback, etc.).



Add action

Action: press key

Pause: 1000

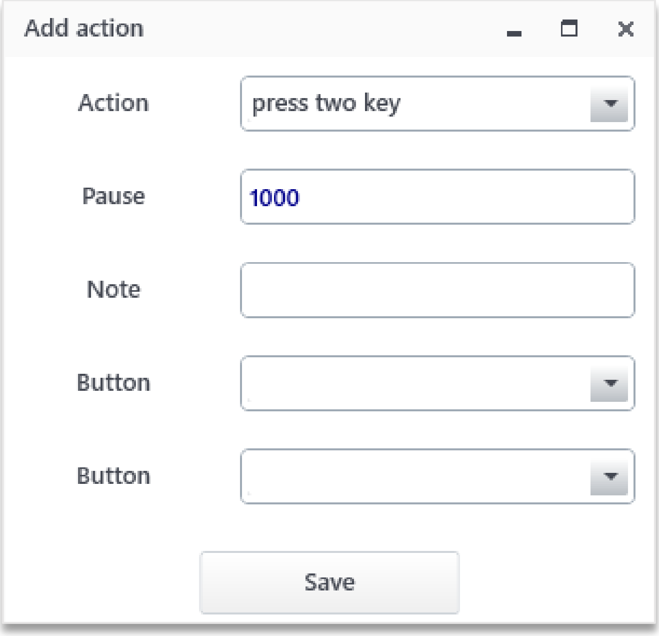
Note:

Button:

Figure. 3.6.8. Action «Press key»

Action «Press two keys»

Imitates double key presses, for example CTRL+A – selection of all text in the field. All required keys are selected in the «Кнопка» drop-down list. The list is the same as in the «Press key» action.



Add action

Action: press two key

Pause: 1000

Note:

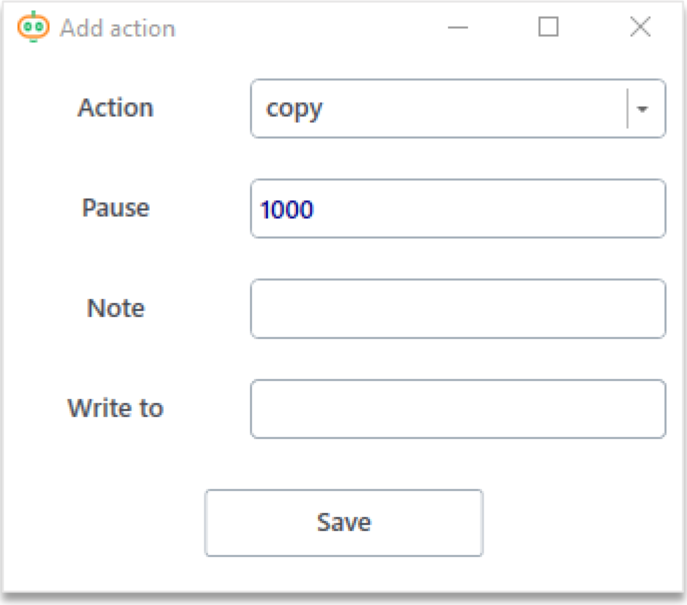
Button:

Button:

Figure. 3.6.9. Action «Press two keys»

Action «Copy»

Imitates the CTRL+C hotkey. It copies the selected value to the clipboard and inserts it, if necessary, to the variable specified in the «Write to» field. **This field is optional.** Variables always start with the «@» symbol.



Action: copy

Pause: 1000

Note:

Write to:

Figure. 3.6.10. Action «Copy»

Action «Paste»

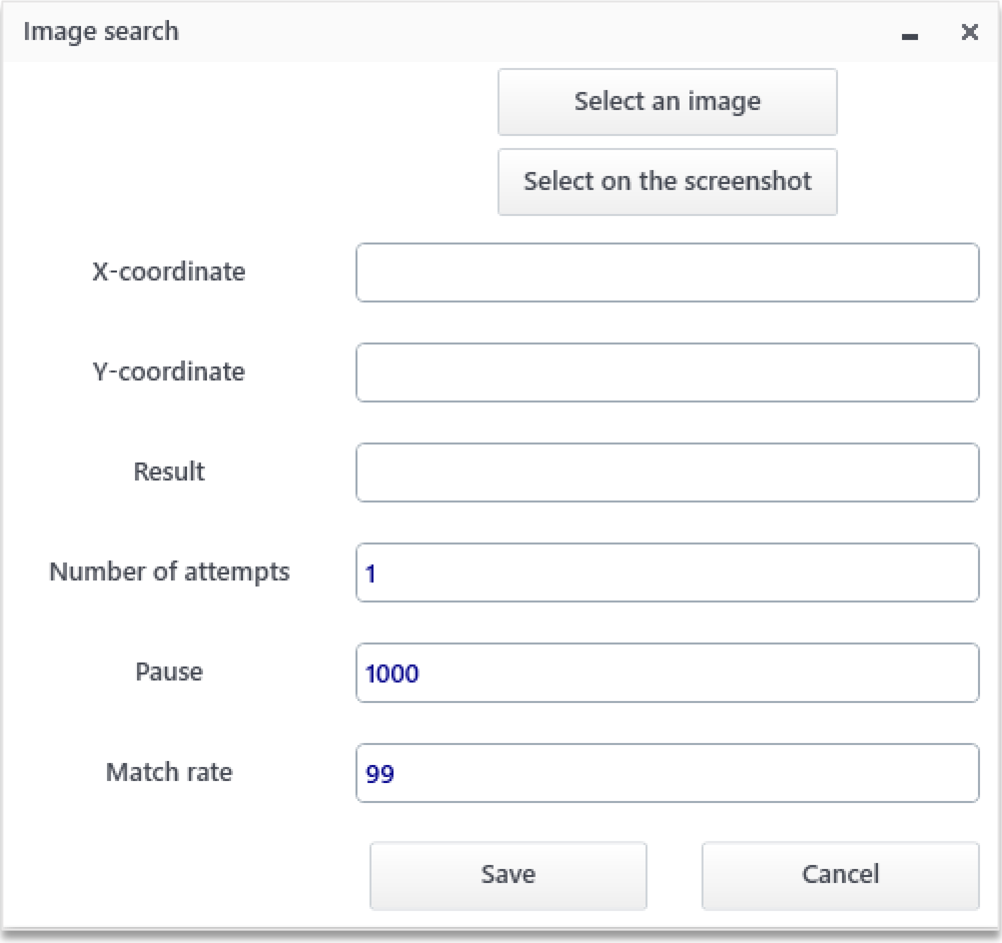
Imitates the CTRL+V hotkey. Inserts the value from the clipboard into the selected field.

Chapter 7. «Image search» module

The «Image search» module is usually used with the «Clicker» module. The module is designed to determine whether the selected image is present on the screen, and if it is, return the coordinates of its location.

Module interface

Window of this module looks like this:



X-coordinate

Y-coordinate

Result

Number of attempts 1

Pause 1000

Match rate 99

Figure. 3.7.1. WIndow of the module

The «Pick image» and «Select image» buttons allow you to select an image to search.

The «x coordinate» field. In the field, a variable is entered into which the x coordinate of the searched image will be written.

The «y coordinate» field. Similar to the «x coordinate» field.

The «Result» field. The name of the variable is gonna be entered into the field. The final result of the image search will be entered into the variable. The resulting value will be presented as a string: «true» - found, «false» - not found.

The «Number of attempts», «Pause» and «Match rate» fields are similar to the fields of the «Clicker» module.

The «Number of attempts» field. The field indicates the number of attempts made for image search.

The «Pause» field. The field is designed for creating a pause after an executed command. The pause is specified in milliseconds.

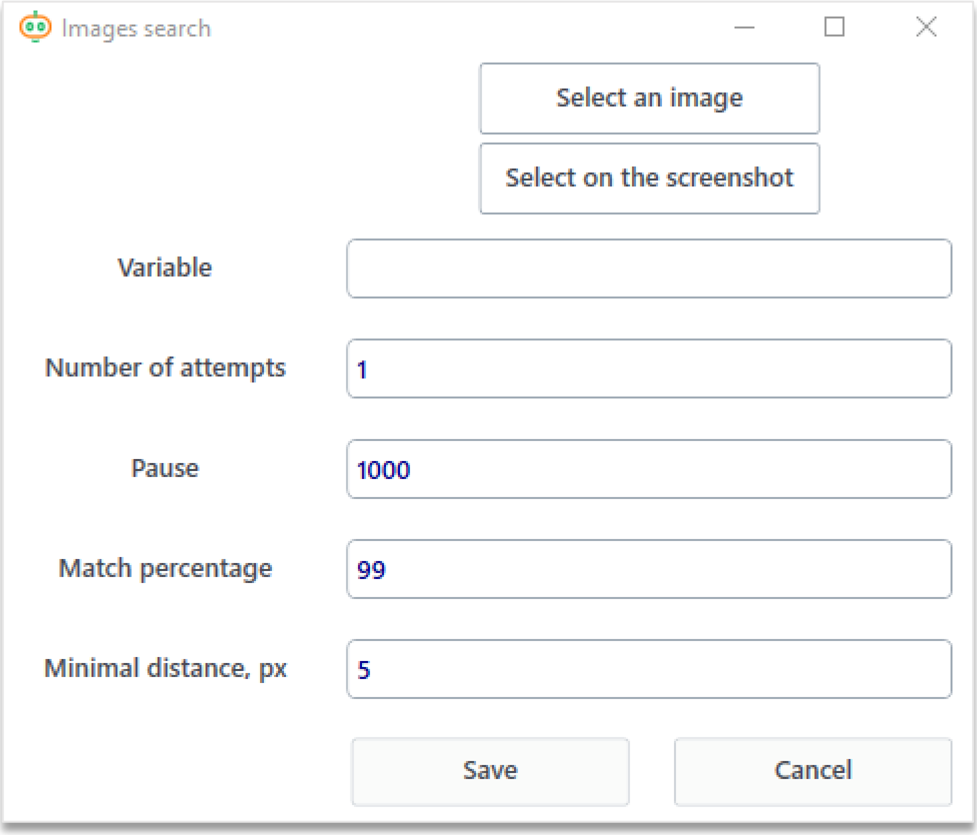
The «Match rate» field - percentage of how much the image to be searched matches with the image fragment on the current screen. A number is entered from 0 to 100, the more, the more similarities between the images should be.

Chapter 8. «Images search»

This module is similar to the «Image search» module, the only difference that this module returns a list of all found images, not just one.

Module interface

Interface of this module is also similar to the «Image search», but instead of «x coordinate», «y coordinate» and «result» fields there is only one field – «Variable», and also the «Minimal distance» field has appeared.



Variable

Number of attempts 1

Pause 1000

Match percentage 99

Minimal distance, px 5

Figure. 3.8.1. Module window

In the «Variable» field the variable name is entered, in which the search result will be indicated.

The «Number of attempts» field is filled in with a number of attempts made for searching an image.

The «Pause» field is designed for creating a pause after an attempt for image search. The pause is specified in milliseconds.

The «Match rate» field - percentage of how much the image to be searched matches with the image fragment on the current screen. A number is entered from 0 to 100, the more, similarities between the images should be.

The «Minimal distance» field is filled with the number of pixels indicating how many pixels should be minimum between two identical images. If some image is too close to what was found first, then it will not be included in the list of found images. **Please note** that if this distance is set very small or equal to 0, then one image can be found several times!

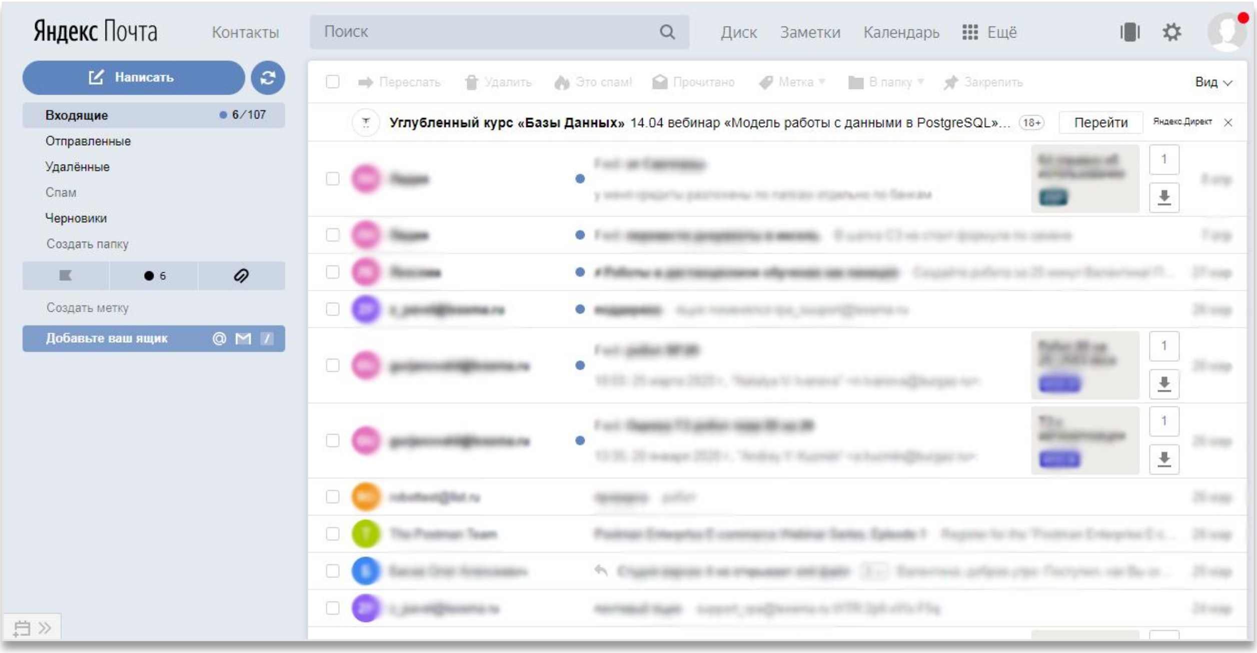
Output data

A list of objects will be written into the specified variable, the number of which will be equal to the number of images found on the screen. Every object matches the found image. To refer to a specific image of the list, it is necessary to indicate its number (numbering from 0) in square brackets after the variable name, for example @listImages[0].

Every object contains three fields: «x», «y» and «similarity». The «x» field contains the x coordinate of the selected image, the «y» field - y coordinate, and the «similarity» field - contains the percentage of similarity of the found image with the desired one.

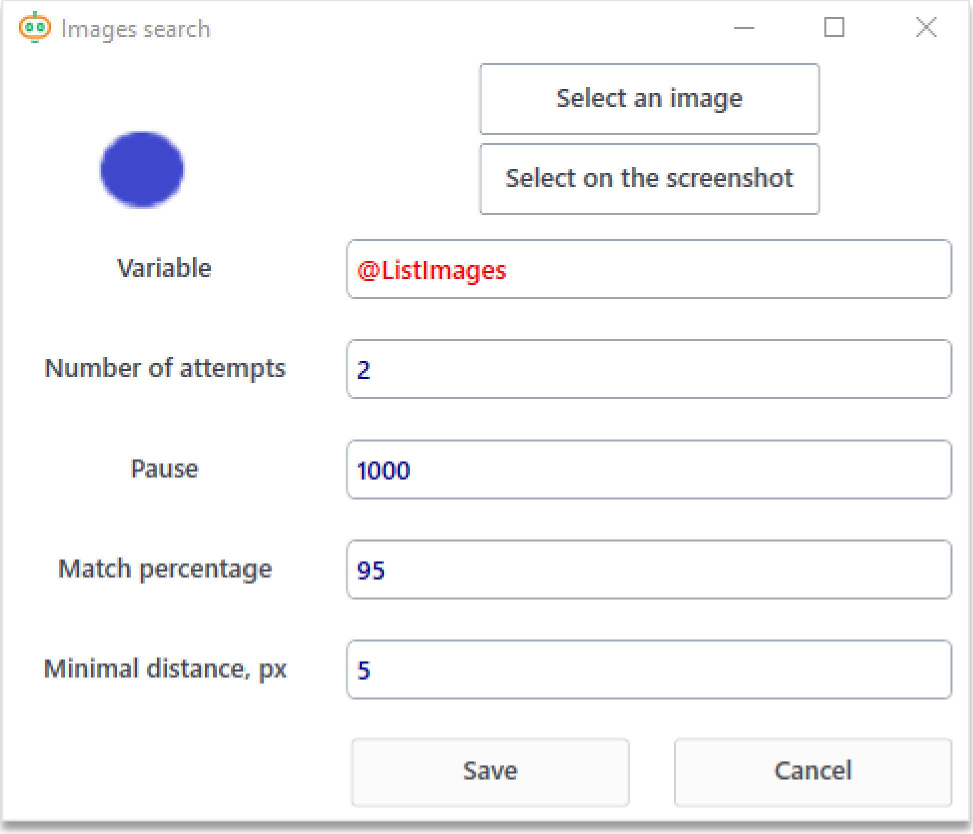
EXAMPLE.

Let's say we want to find multiple similar images using our robot, for example, unread emails from Yandex-mail, which are highlighted with a small blue circle.



Углубленный курс «Базы Данных» 14.04 вебинар «Модель работы с данными в PostgreSQL»... 18+ Перейти Яндекс.Директ

To do that let's create the «Search of multiple images» action and select the blue circle as an image we're gonna be searching:



Images search

Variable: @ListImages

Number of attempts: 2

Pause: 1000

Match percentage: 95

Minimal distance, px: 5

The number of images found is a number of unread emails will be @listImages[%].

For the robot to click every circle and read every message you need to add a cycle by the number of images and click the coordinates @listImages[@i].x and @listImages[@i].y, where @i – current step of the cycle.

PART IV. MODULE «WORK WITH FILES»

Chapter 9. Key aspects of the module

The «Work with files» module is designed for changing file directory, copying them, archiving them, making folders and transfer files to it, in general, to manipulate files and directories

Module interface

Module widow has three groups of elements.

First group of elements is a number of fields and buttons for creating a command. Second group of elements is a list of commands. And the third group of elements is a preview area for viewing text files.

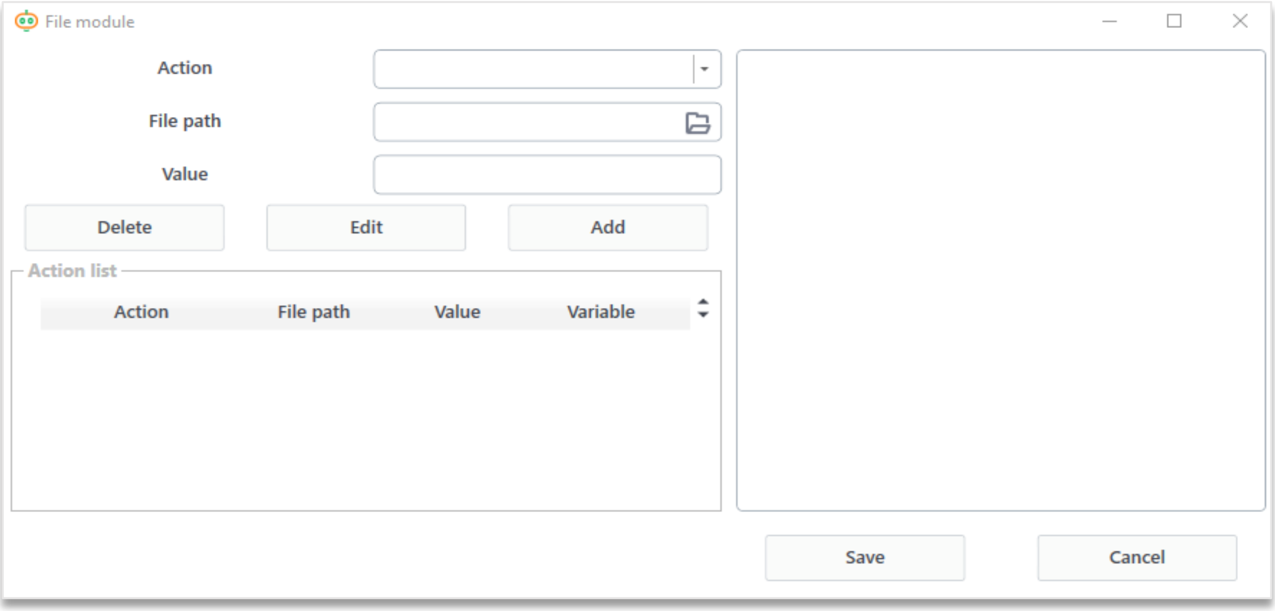


Figure. 4.9.1. Window module

The set of fields varies on a selected action, more on that in chapter 10.

Module functionality

Creating a command

To create a command you need to – select the desired action, fill out the remaining fields – click the «Add» button, and the command will be added to the «Command list».

Editing a command

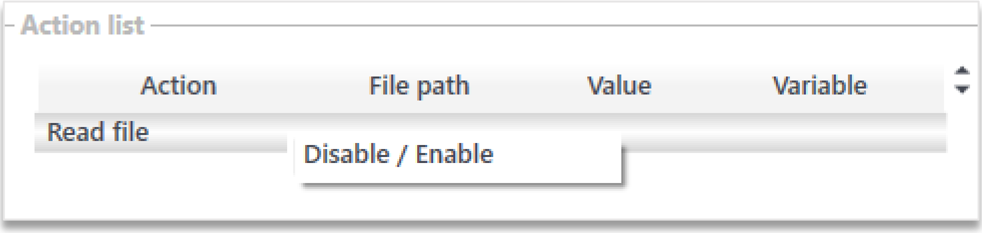
To edit a command you need to select it from the list of commands by left-clicking on it, after which all fields in the top left part of the «Work with files» window will be filled in accordance with the selected command. Change the fields you want to edit, making sure that the action you're editing is still selected in the command list.

Deleting a command

In order to delete a command select it in the command list and press the «Delete» button.

Turning on/off a command

You can turn the commands on and off. Turned off commands won't run. You can do that by right-clicking on the command and pressing «Turn on/off».



Action list

Action	File path	Value	Variable
Read file			

Figure. 4.9.2. Context menu of a command

Choosing file/folder

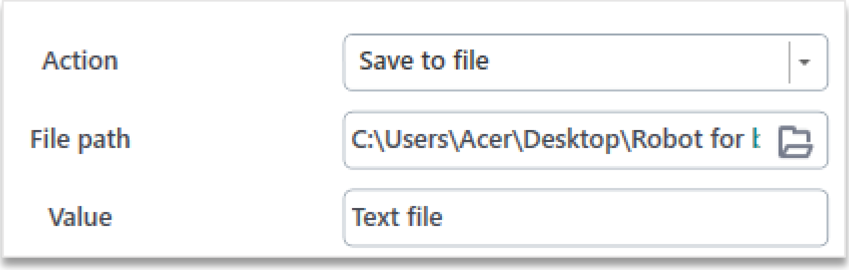
The 🗀 button in fill out fields opens up a standard dialog window for choosing a file or a folder, depending on a selected action.

Chapter 10. Description of actions available in the module

List and description of actions

Action «Save in file»

Action rewrites the existing file, by adding text entered in the «Value» field.

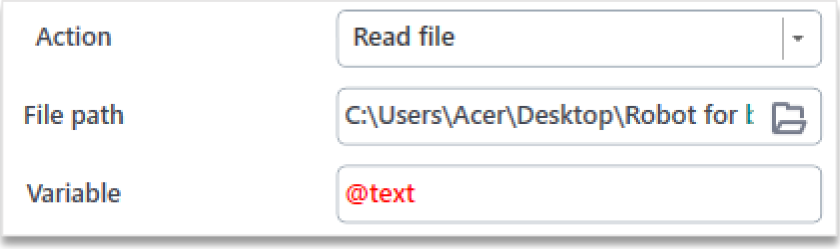


Action	Save to file
File path	C:\Users\Acer\Desktop\Robot for t
Value	Text file

Figure. 4.10.1. Save in file

Action «Read file»

Action reads all text from the existing file and enters it to the specified variable.

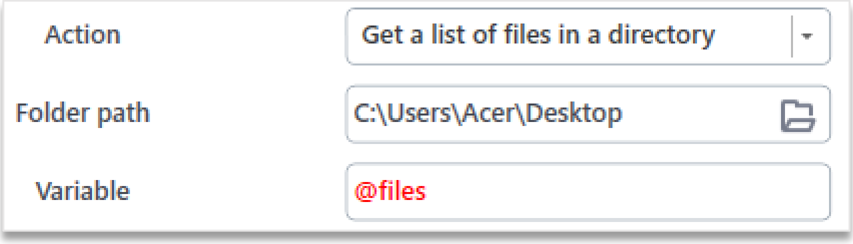


Action	Read file
File path	C:\Users\Acer\Desktop\Robot for t
Variable	@text

Figure. 4.10.2. Read file

Action «Get list of file in folder»

The action creates a list-variable that contains paths to each file in the folder.

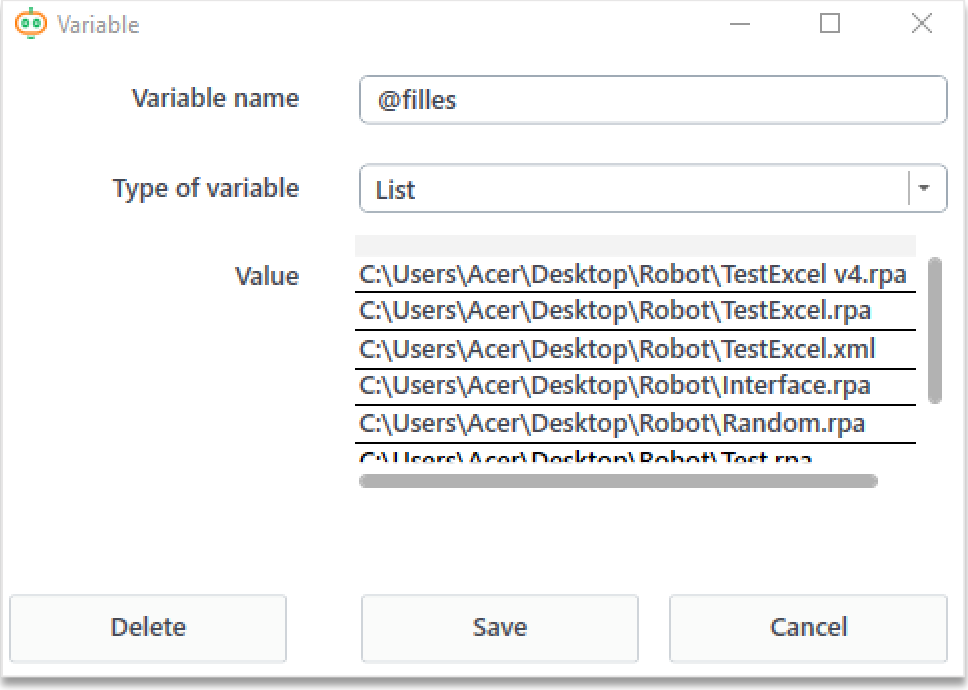


Action	Get a list of files in a directory
Folder path	C:\Users\Acer\Desktop
Variable	@files

FIgure. 4.10.3. Getting a list of files

EXAMPLE.

Let's say we have a folder called «Robot», in which 3 Excel files are located. Then, in the Excel module we can work with each of these files, for example, to work with the first one you need to enter @files[0] into the «File name» field . Variable after executing an action



Variable name: @filles

Type of variable: List

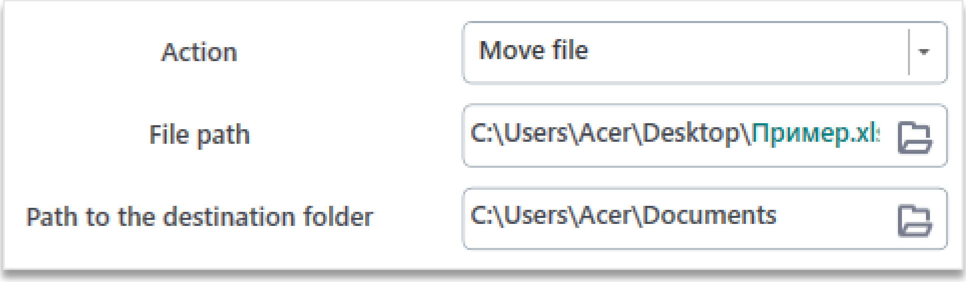
Value:

C:\Users\Acer\Desktop\Robot\TestExcel v4.rpa
C:\Users\Acer\Desktop\Robot\TestExcel.rpa
C:\Users\Acer\Desktop\Robot\TestExcel.xml
C:\Users\Acer\Desktop\Robot\Interface.rpa
C:\Users\Acer\Desktop\Robot\Random.rpa
C:\Users\Acer\Desktop\Robot\Test.rpa

Figure. 4.10.4. Example

Action «Move file»

Action moves the file into the specified folder.

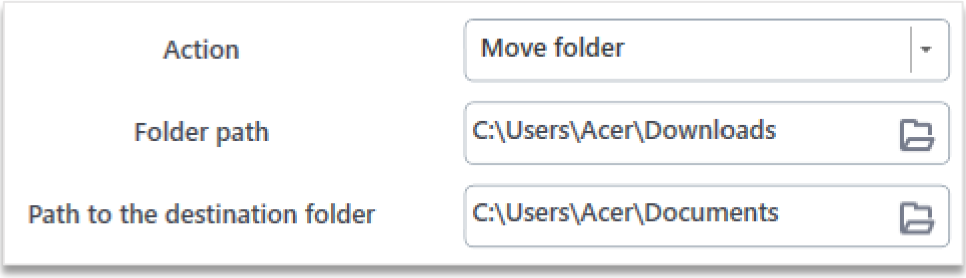


Action	Move file
File path	C:\Users\Acer\Desktop\Пример.xl:
Path to the destination folder	C:\Users\Acer\Documents

FIgure. 4.10.5. Moving a file

Action «Move folder»

Action moves the folder, specified in the second field into the directory specified in the third field.

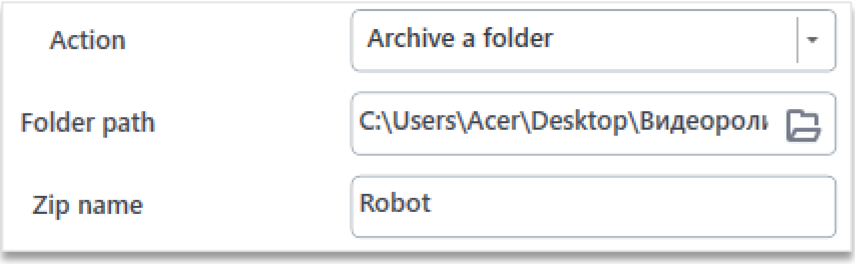


Action	Move folder
Folder path	C:\Users\Acer\Downloads
Path to the destination folder	C:\Users\Acer\Documents

Figure. 4.10.6. Moving a folder

Action «Archive folder»

Action creates a ZIP archive with the specified name, adding the contents of the selected folder to it.

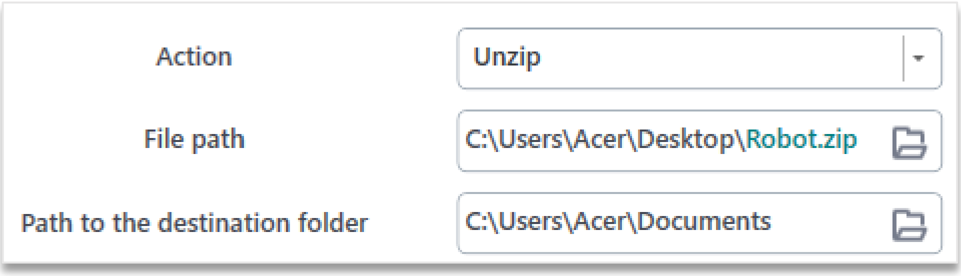


Action	Archive a folder
Folder path	C:\Users\Acer\Desktop\Видеороли
Zip name	Robot

Figure. 4.10.7. Archiving a folder

Action «Unzip archive»

Action unzips a ZIP archive into the specified folder.

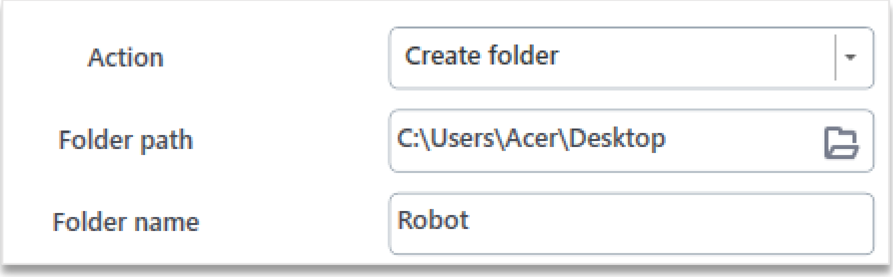


Action	Unzip
File path	C:\Users\Acer\Desktop\Robot.zip
Path to the destination folder	C:\Users\Acer\Documents

Figure. 4.10.8. Unzip archive

Action «Create folder»

Action creates a folder with a specified name in the selected directory.

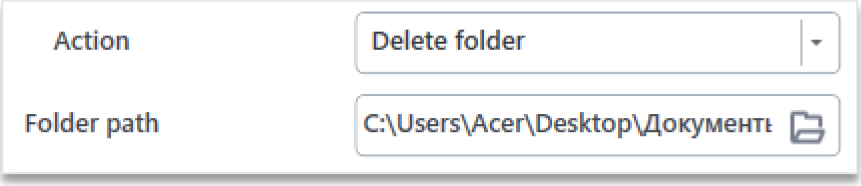


Action	Create folder
Folder path	C:\Users\Acer\Desktop
Folder name	Robot

Figure. 4.10.9. Creating a folder

Action «Delete folder»

Action deletes the selected folder.

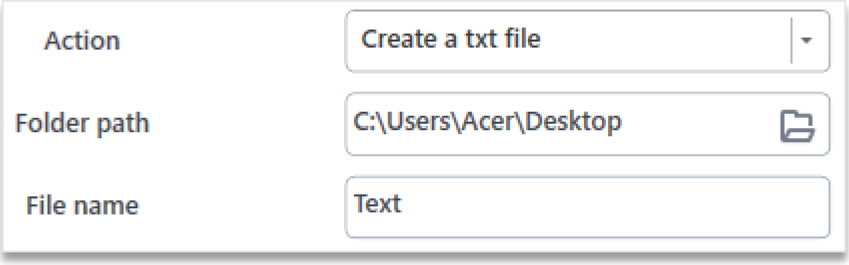


Action	Delete folder
Folder path	C:\Users\Acer\Desktop\Документы

Figure. 4.10.10. Deleting a folder

Action «Create .txt file»

Action creates a file with .txt extension.

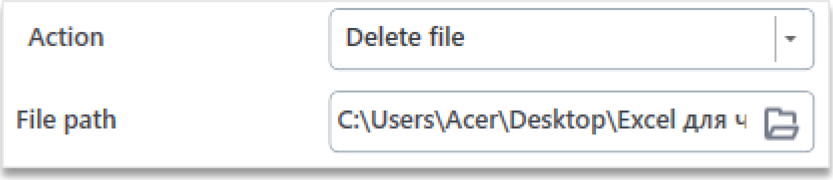


Action	Create a txt file
Folder path	C:\Users\Acer\Desktop
File name	Text

Figure. 4.10.11. Creating .txt file

Action «Delete file»

Action deletes the selected file.



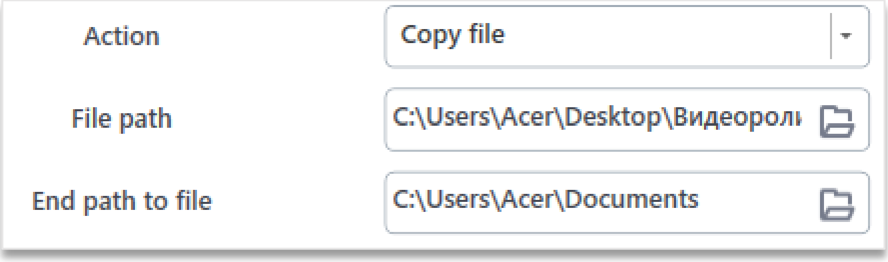
Action	Delete file
File path	C:\Users\Acer\Desktop\Excel для ч

Figure. 4.10.12. Deleting a file

Action «Copy file»

Action copies the selected file to the specified directory.

In the «Path to the destination file» field enter the path to the folder into which the file should be copied, and its name with the extension as if the file already exists. The new name may differ from the original.



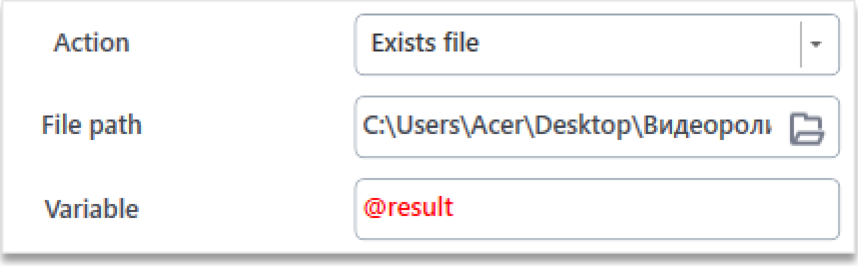
Action	Copy file
File path	C:\Users\Acer\Desktop\Видеороли
End path to file	C:\Users\Acer\Documents

Figure. 4.10.13. Copying a file

Action «Check file existence»

Action checks the existence of the specified file and projects the result to the variable.

«True» - if exists, «False» - if it doesn't.



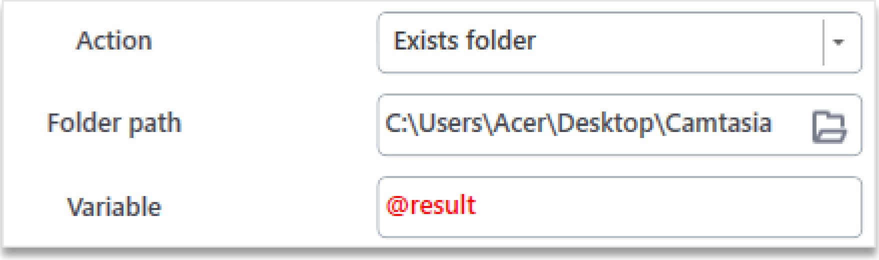
Action	Exists file
File path	C:\Users\Acer\Desktop\Видеороли
Variable	@result

Figure. 4.10.14. Checking the file existence

Action «Check folder existence»

Action checks the existence of the specified folder and projects the result to the variable.

«True» - if exists, «False» - if it doesn't.



Action	Exists folder
Folder path	C:\Users\Acer\Desktop\Camtasia
Variable	@result

Figure. 4.10.15. Checking folder existence

PART V. MODULES «CYCLE», «CONDITION», «WHILE», «BREAK» AND «RETURN»

Chapter 11. Module «Condition»

The modules described in this part do nothing by themselves, they should be used together with other modules. The described modules (except «Break»), in addition to the usual «Note» field, have a few more parameters that are required to fill out, but there is no separate settings window.

Module «Condition» module allows you to perform certain actions depending on the specified condition. The condition itself doesn't matter as long as it comes down to «True» or «False».

Module interface

Module consists of the fields «Note» and «Condition». Actions that must be performed when the value of the condition is «True», are added to the «THEN» section, if the value is «False» - to the «ELSE» section. Actions are added to the module by dragging them from the menu or another place in the workspace.

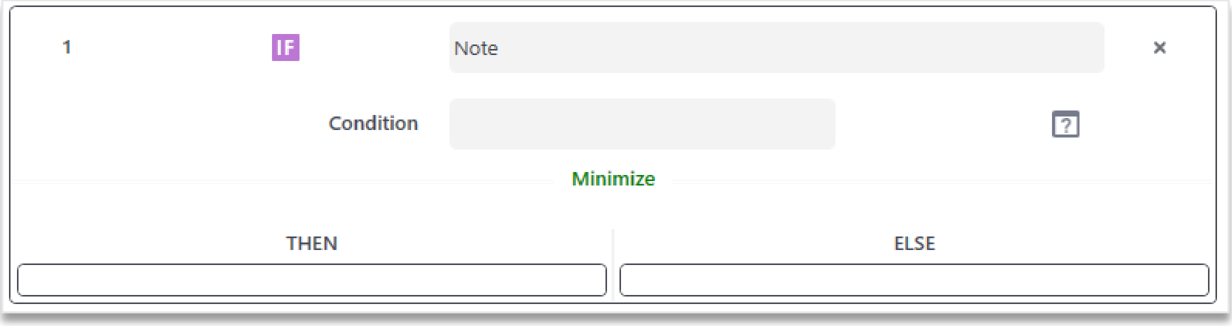


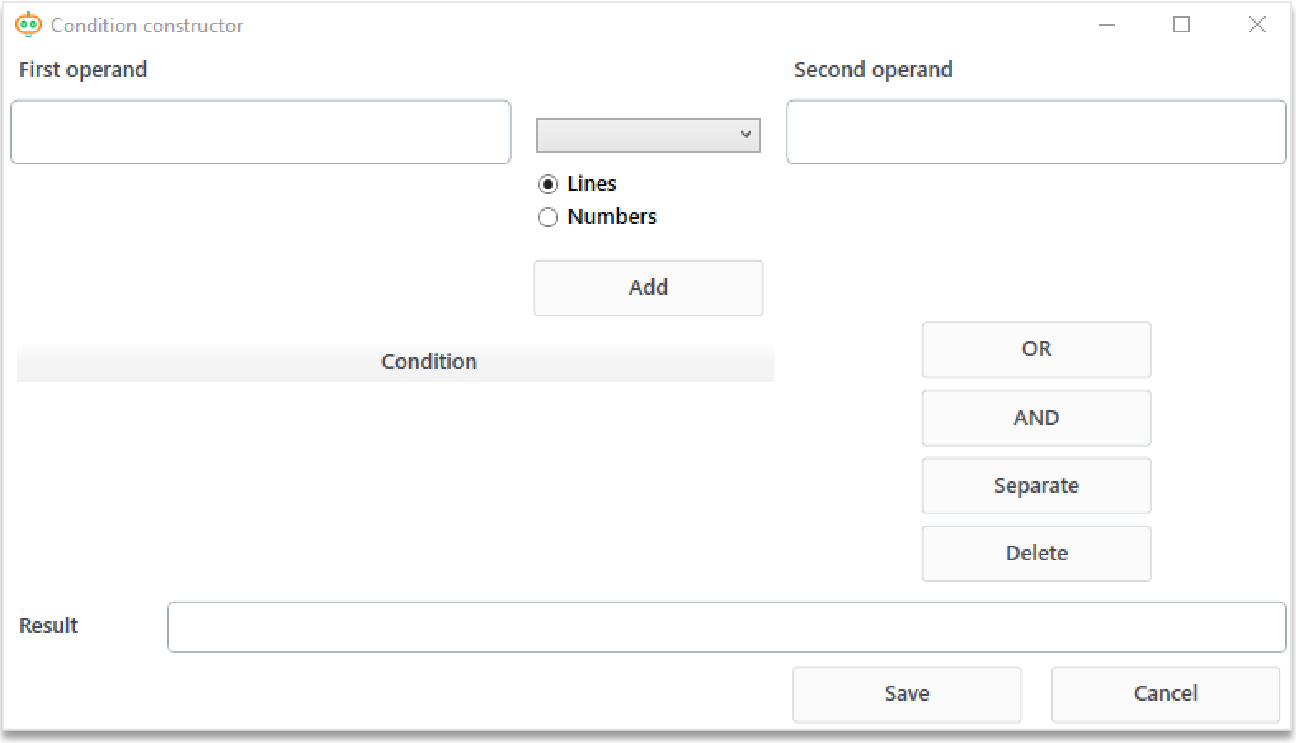
Figure. 5.11.1. The condition module

The «Note» field is designed for robot creators to leave notes/comments.

The «Condition» field is filled according to the rules of the C# programming language. For users unfamiliar with this language, help is offered in the form of a «Constructor of conditions», accessible by clicking the ? button to the right of the condition.

Constructor of conditions

Constructor of conditions allows creating conditions in accordance with the syntax of the C # language.



Condition constructor

First operand

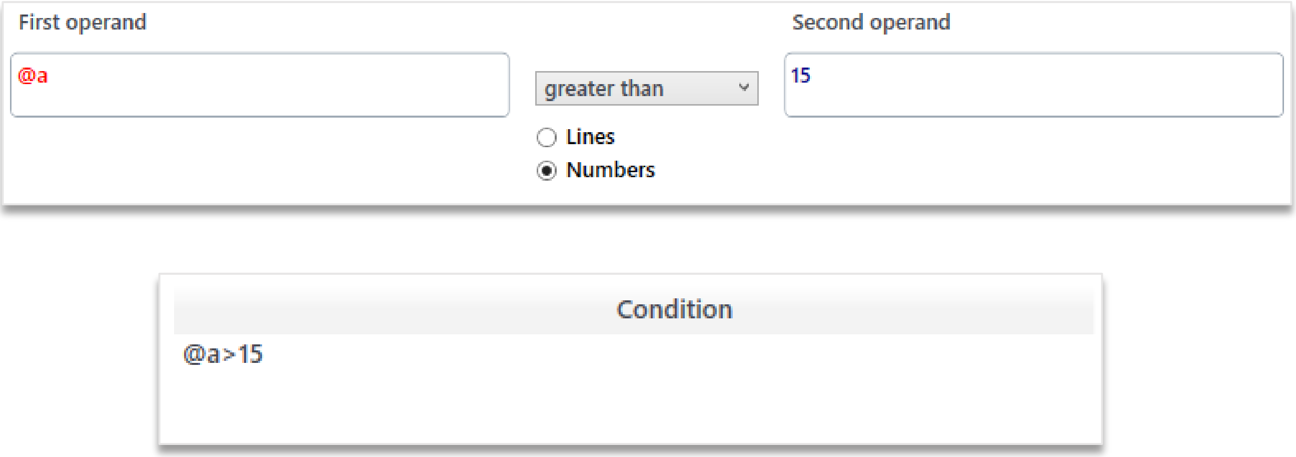
Second operand

● Lines
○ Numbers

Figure. 5.11.2. Constructor of conditions

Adding

In order to make a condition, it is necessary to fill in both operands, select the type of operands (rows or numbers), select the operator between them from the proposed one and click on the «Add» button:



First operand: @a

greater than

Second operand: 15

○ Lines
● Numbers

Condition

@a>15

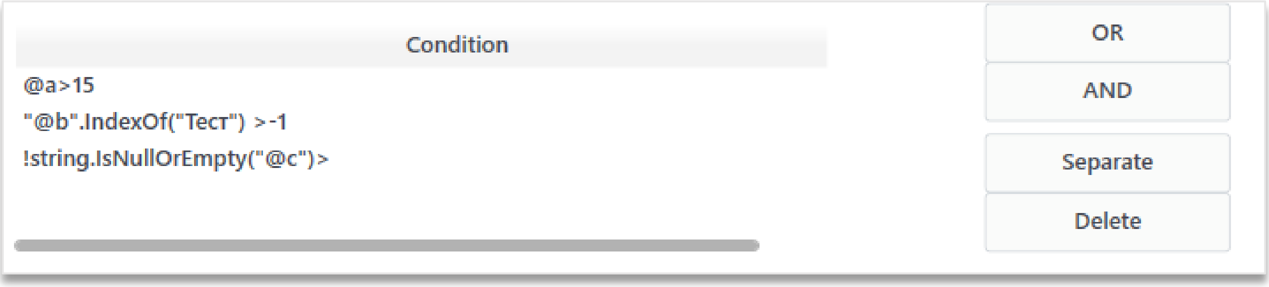
Figure. 5.11.3. Adding conditions

The list of operations for rows and for numbers is different, so first select the type of operands, and only then the operation.

Binding

You can bind multiple conditions with logical «AND» or logical «OR».

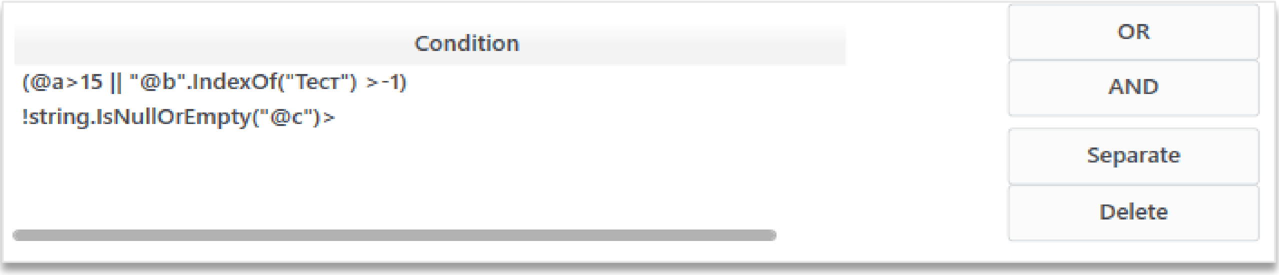
This is done as follows - holding the Shift key selects several conditions that must be connected by one operator, then the button with the corresponding logical operator is pressed:



Condition

```
@a>15
"@b".IndexOf("Тест") >-1
!string.IsNullOrEmpty("@c")>
```

Figure. 5.11.4. Binding conditions



Condition

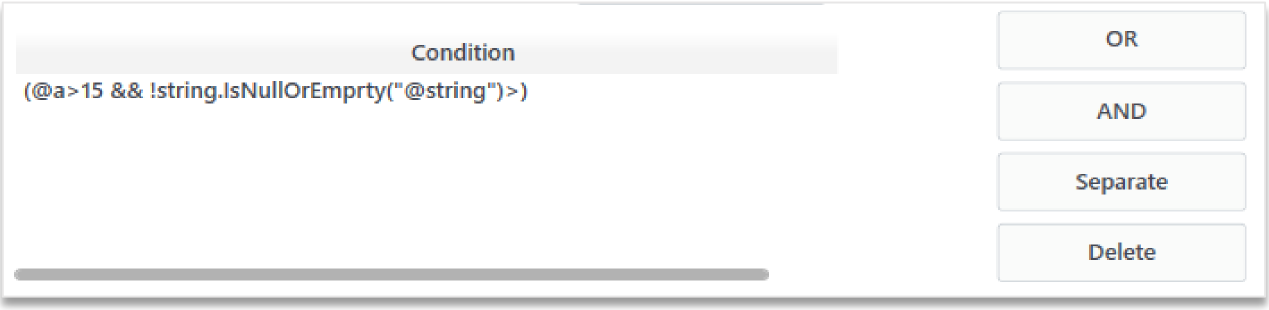
```
(@a>15 || "@b".IndexOf("Тест") >-1)
!string.IsNullOrEmpty("@c")>
```

Figure. 5.11.5. Result of binding action

The «||» symbol means logical OR, the «&&» symbol means logical AND.

Unbinding

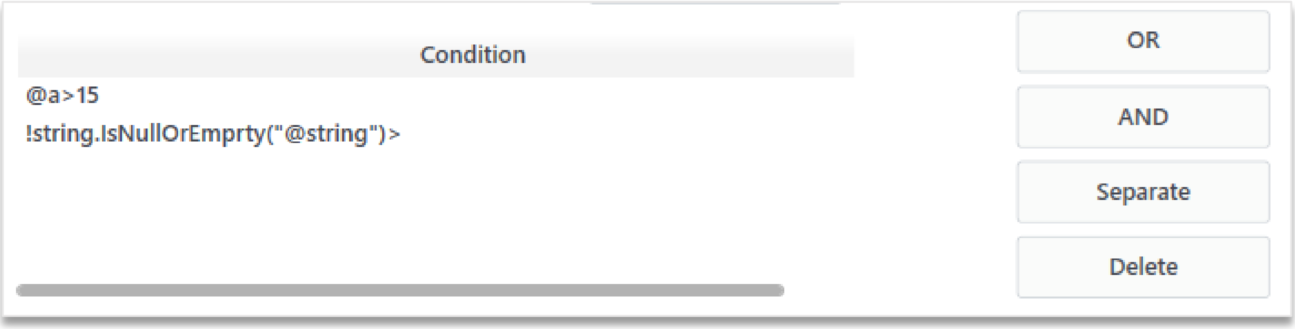
Conditions that consist of several simple conditions, meaning, those that use a logical AND or OR, can be disconnected by clicking on the «Unbind» button. Disconnection is done by logical operators.



Condition

```
(@a>15 && !string.IsNullOrEmprty("@string")>)
```

Figure. 5.11.6. Unbinding conditions

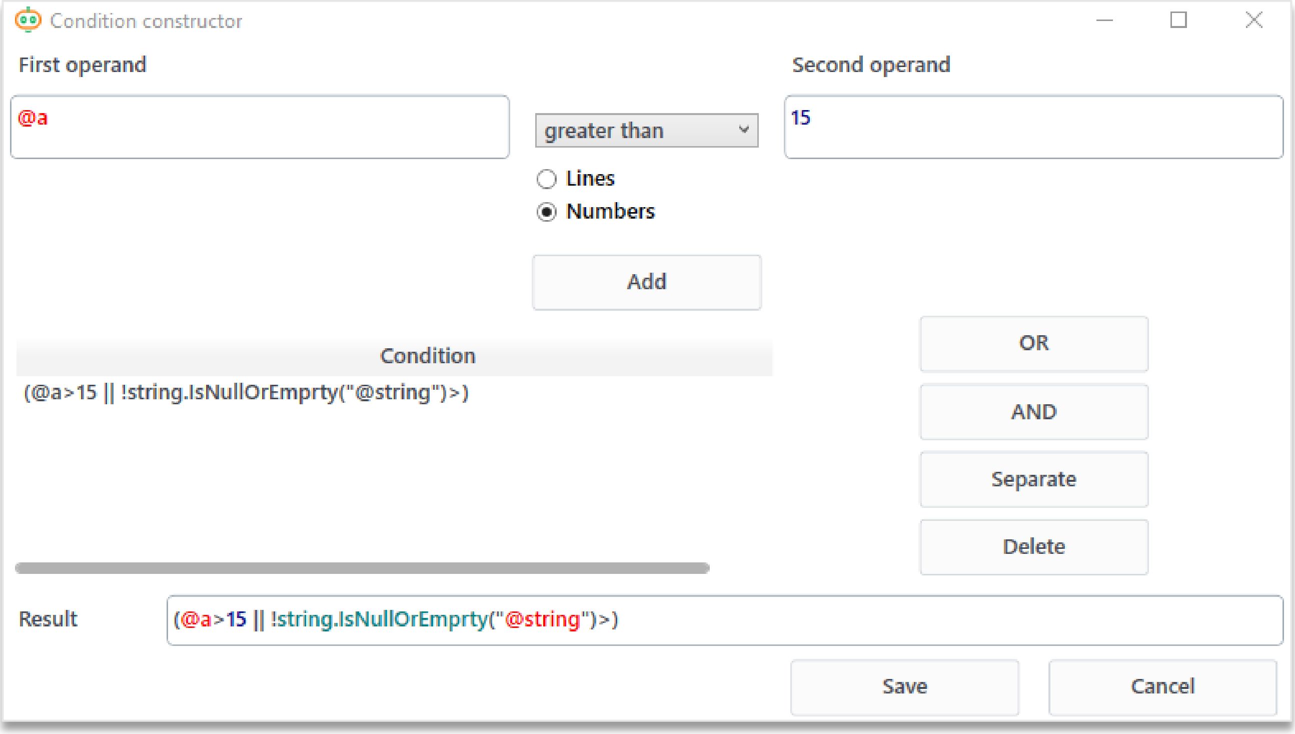


```
@a>15
!string.IsNullOrEmprty("@string")>
```

Figure. 5.11.7. Result of unbinding conditions

Saving

To save the condition, press the «Save» button. The final condition is inserted to the «Condition» field of the conditional module:



First operand: @a

greater than

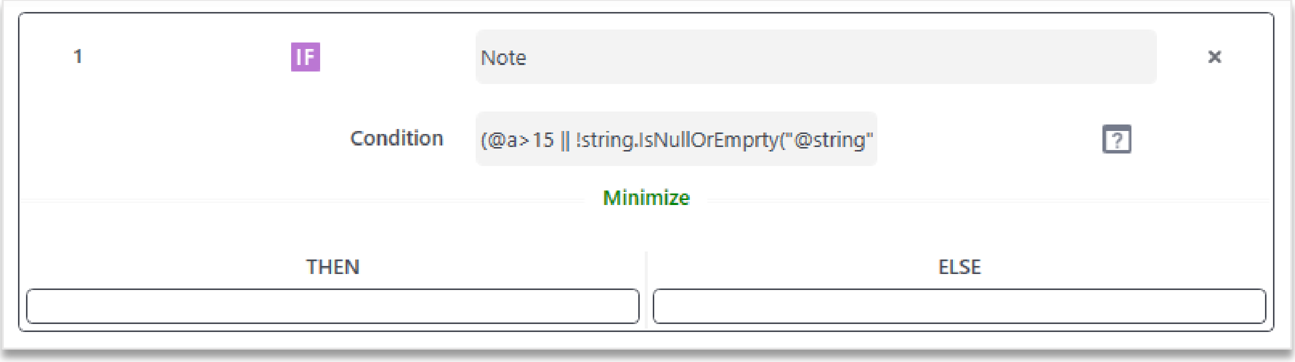
○ Lines
● Numbers

Second operand: 15

```
(@a>15 || !string.IsNullOrEmprty("@string")>)
```

Result: (@a>15 || !string.IsNullOrEmprty("@string")>)

FIgure. 5.11.8. Saving condition

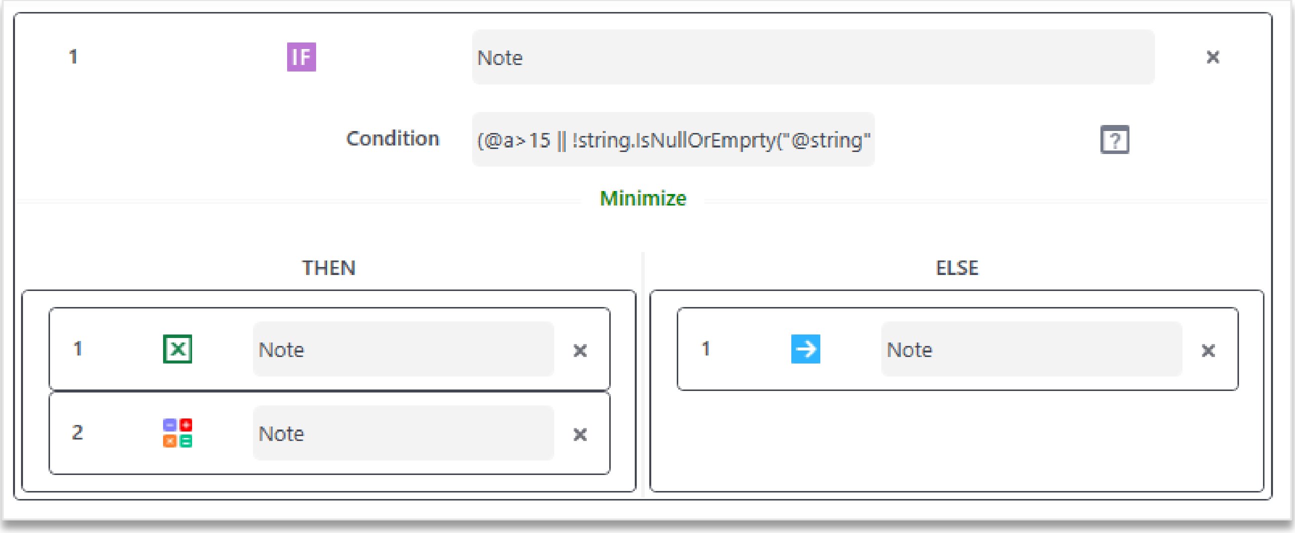


1 IF Note ×

Condition (@a>15 || !string.IsNullOrEmprty("@string"

Figure. 5.11.9. How the condition projects

Finished module will look like this:



1 IF Note ×

Condition (@a>15 || !string.IsNullOrEmprty("@string"

THEN:

1 Note ×

2 Note ×

ELSE:

1 Note ×

Figure. 5.11.10. How the condition projects

Minimize

To reduce occupied workspace you can click the «Minimize» button to minimize actions included in this module.

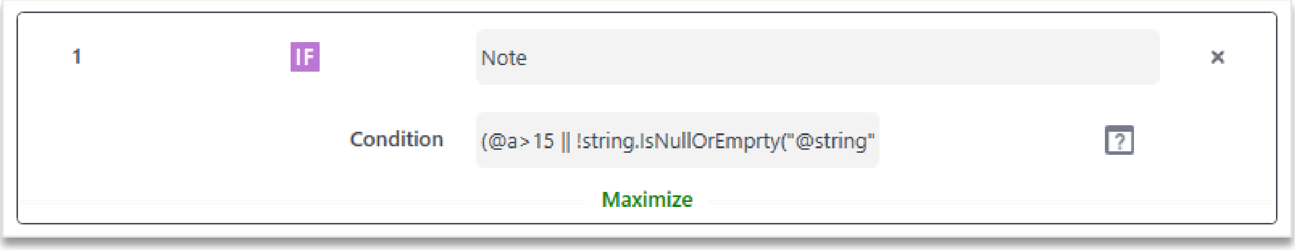


Figure. 5.11.11. Minimizing actions

You can also change the size of sections «THEN» and «ELSE» up to completely minimizing one of them, for example, if one of them is not being used.

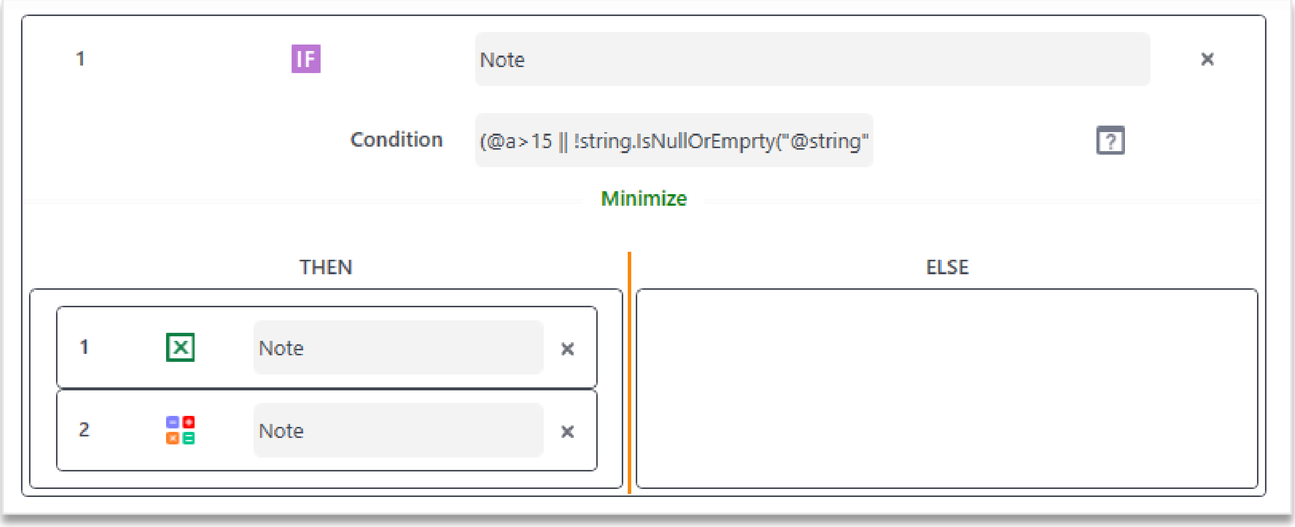
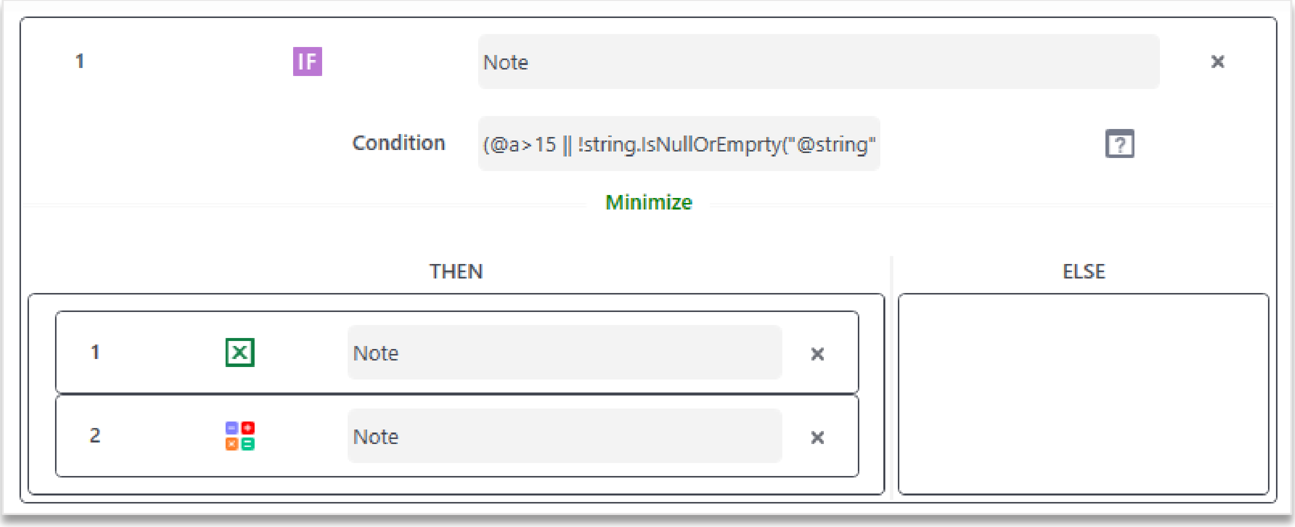


Figure 5.11.12 Changing the size of sections



1 IF Note ×

Condition (@a>15 || !string.IsNullOrEmprty("@string"

Figure. 5.11.13. Minimizing conditions

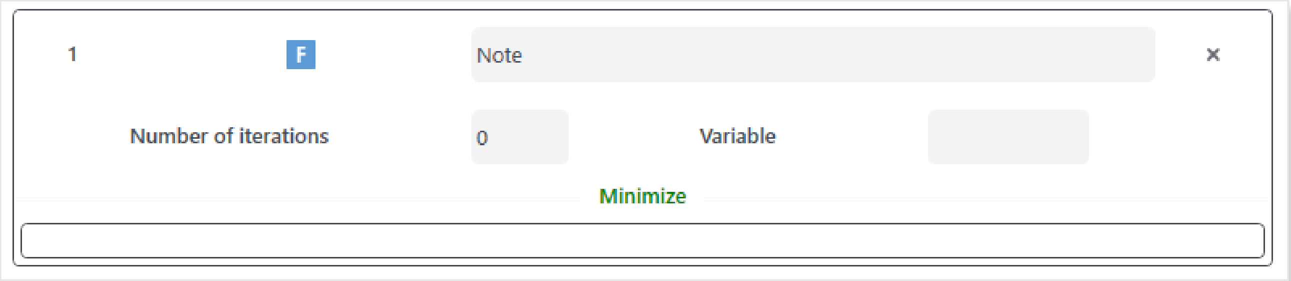
Chapter 12. «Cycle» module

The «Cycle» module is used to repeatedly perform a set of actions. Actions added to the module will be performed until the number of iterations specified in the settings has passed.

Module interface

Module consists of the fields «Note», «Number of iterations» and «Variable». The actions to be performed in the cycle are added to the block located at the bottom of the module. Actions are added to the module by dragging them from the menu or other part of the workspace.

The module looks like this:



1 F Note ×

Number of iterations: 0

Variable:

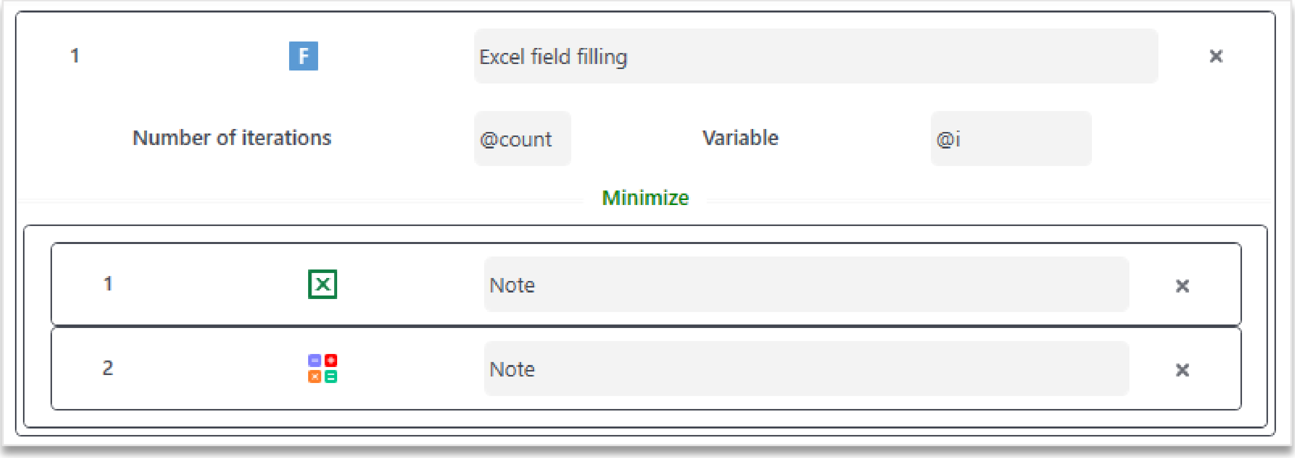
Figure. 5.12.1. Module window

The «Note» field, as in other modules. This field is used for adding a comment from a robot's creator.

The «Number of iterations» field is intended to enter a number that indicates how many times it is necessary to perform the actions contained in the module.

The «Variable» field - the variable in which the number will be stored - the current step of the cycle. The cycle starts with one.

An example of a completed cycle is shown in Figure 5.12.2.



1 F Excel field filling ×

Number of iterations: @count

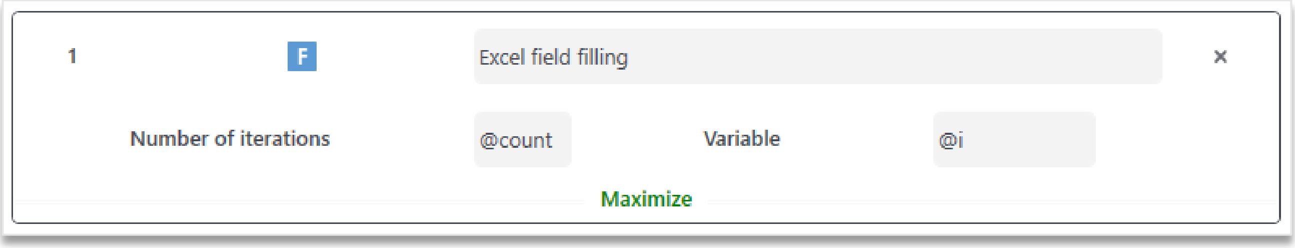
Variable: @i

1 Note ×

2 Note ×

FIgure. 5.12.2. Filling example

By clicking the «Minimize» button you can minimize the cycle actions to display the command more compactly



Excel field filling

@count

@i

Figure. 5.12.3. Minimizing the cycle

Chapter 13. «While» module

The «While» module (conditional cycle) performs the added actions until the specified condition returns «True».

Module interface

The following fields are located in the module window «Note» and «Condition». The block located at the bottom of the module should contain the actions that need to be performed in a cycle.

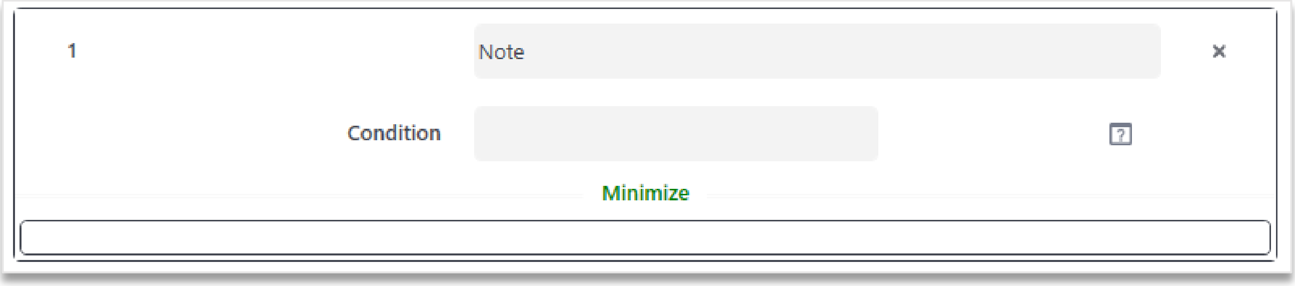
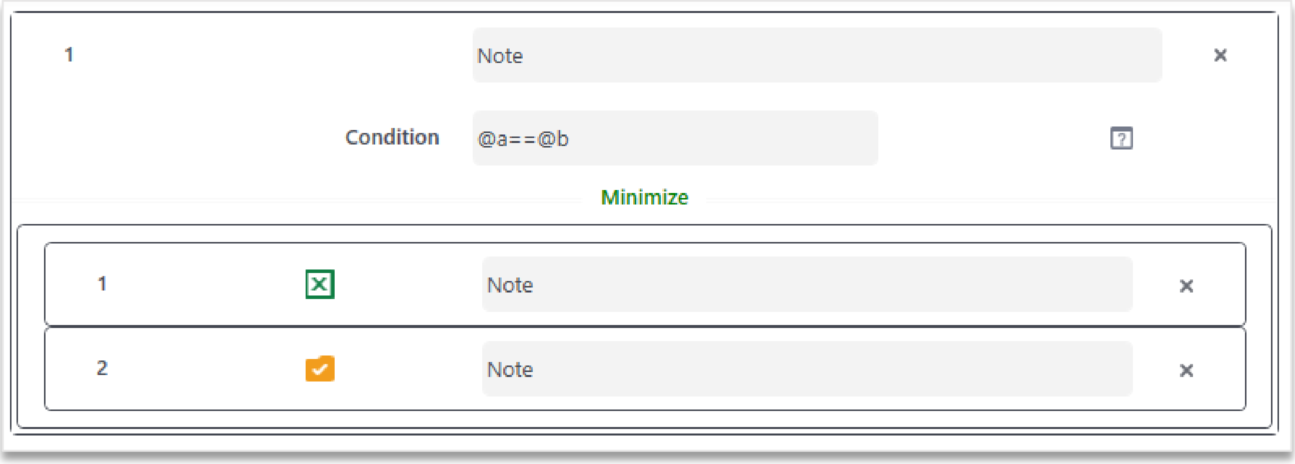


Figure. 5.13.1. Module window

The «Condition» field. A condition is entered into the field as in «Condition» module, which must be written in accordance with the C# syntax, but if you don't know this programming language you can press the «Constructor of conditions» button (see chapter 11).

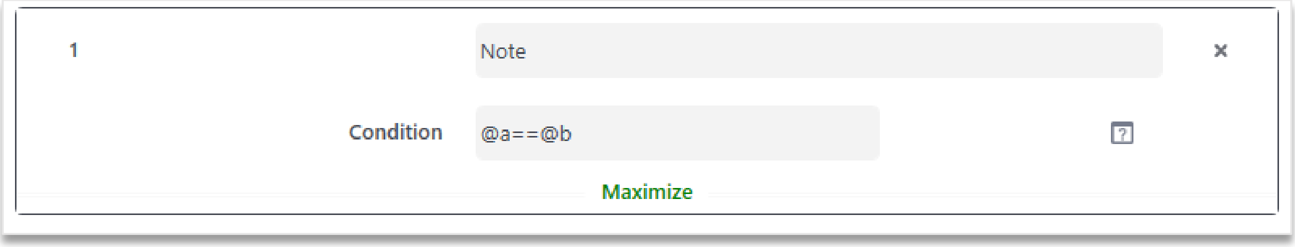
Adding actions to the module is done by dragging them from the menu or any other part of the workspace.



Condition: @a==@b

Figure. 5.13.2.Adding actions

By clicking the «Minimize» button, actions in the cycle minimize for the more compact view of the command



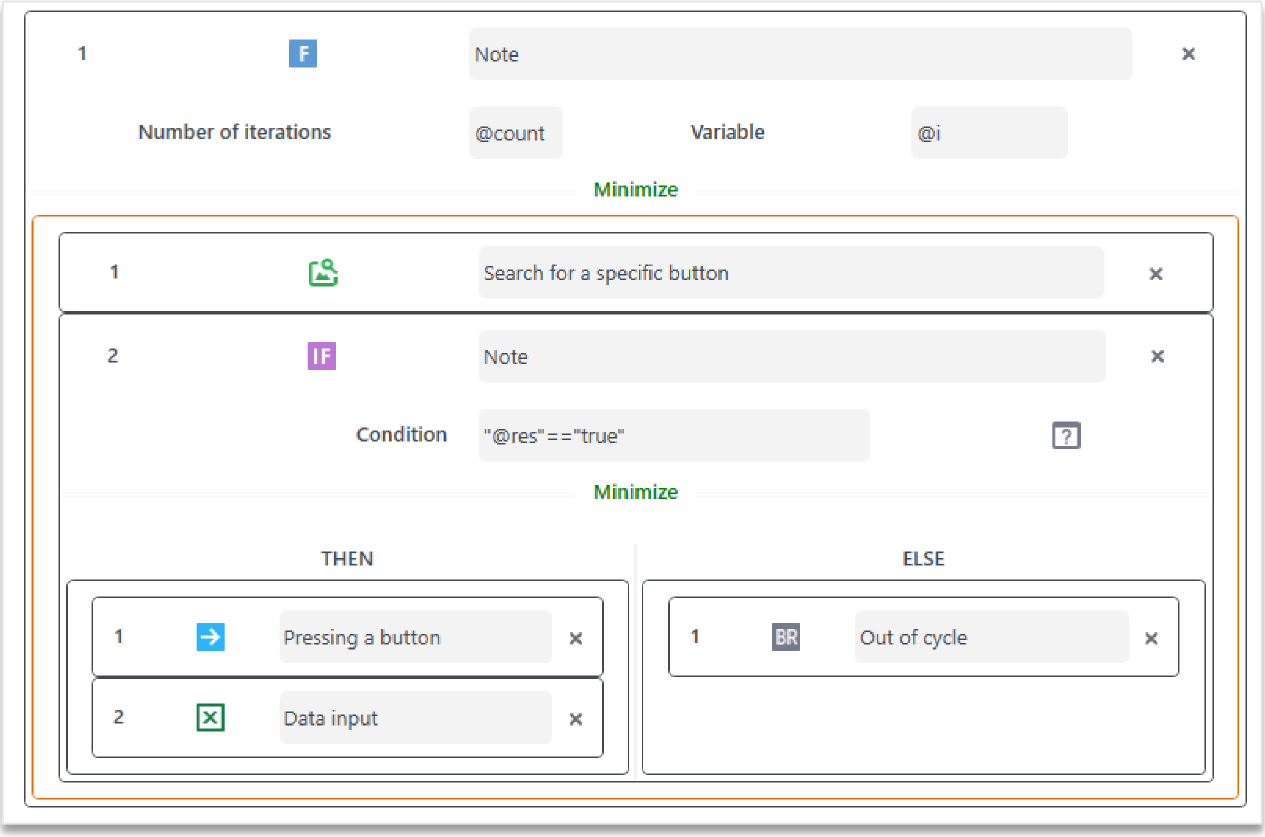
1 Note

Condition @a==@b

Figure. 5.13.3. Minimizing the cycle

Chapter 14. «Break» Module

For «Emergency» exit from a simple cycle, you can use the «Break» operation. For example, in a cycle, a specific image is searched, and if the image is found, the robot should click on it, read data from it and go to the next action, otherwise, exit the cycle and not run the next action. For this scenario you should create a cycle that contains «Condition» module, and in one of the sections there must be the module «Break»:



1 F Note

Number of iterations @count Variable @i

1 Search for a specific button

2 IF Note

Condition "@res"=="true"

THEN

1 Pressing a button

2 Data input

ELSE

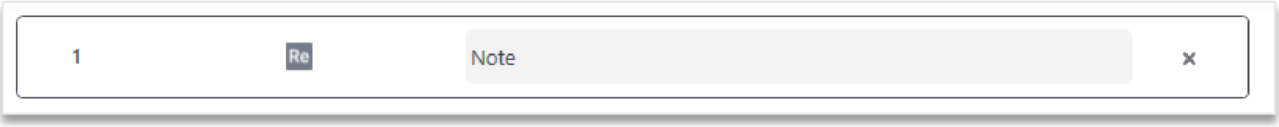
1 BR Out of cycle

Figure. 5.14.1. Module window

Module «Break» Doesn't need configuration but should only be used inside the «Cycle» module.

Chapter 15. «Return» module

There are situations when, under some condition, it is necessary to finish the work of the robot, for example, you need to process an email that should be sent, but it did not come. For such purposes there is a «Return» module.



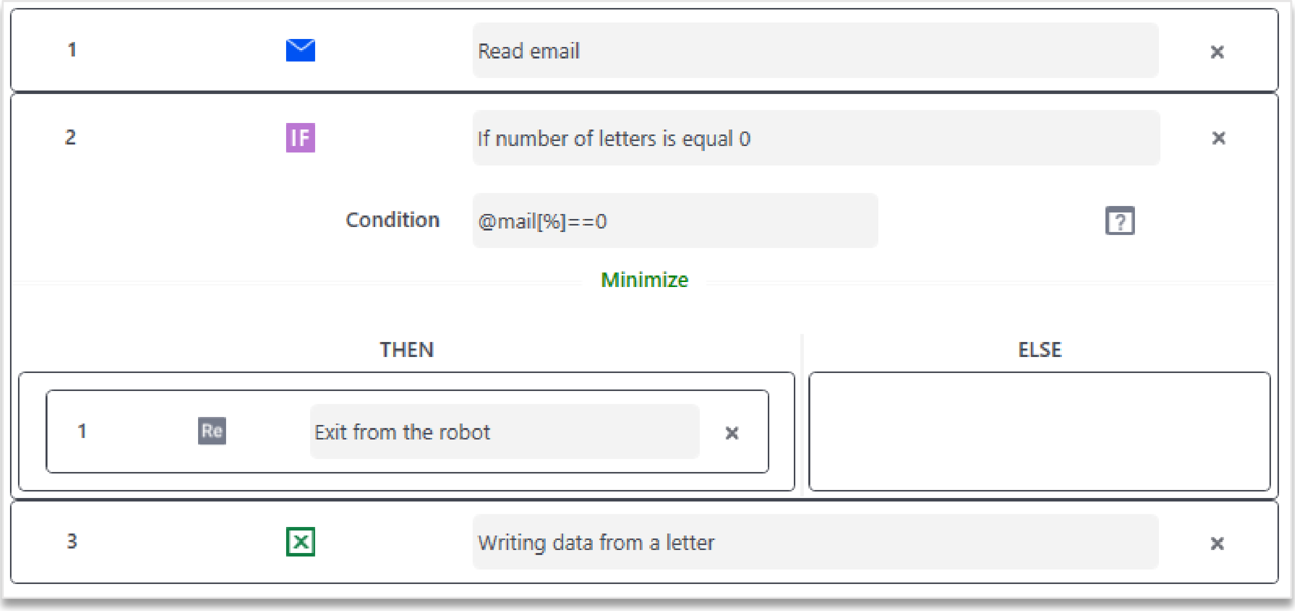
1 Re Note ×

Figure. 5.15.1. Appearance of the module

This module finishes the work of the whole robot and doesn't require configuration. It can be used anywhere.

EXAMPLE.

Below you can see an example of using this module: with the help of the «Read emails» module you can make the robot read all emails for a certain period of time.All subjects of the email and their contents must be entered into an Excel file. But what if you don't have the emails? To do this, let's check for emails, and if they're not there, the robot will stop working, but if they are, it will continue.



1 Read email ×

2 IF If number of letters is equal 0 ×

Condition @mail[%]==0

Minimize

THEN

1 Re Exit from the robot ×

ELSE

3 Writing data from a letter ×

Thanks to this module, you can write robots in the general workspace, and not in the branches of the «Condition» module. This way the robot looks more accurate and intuitive.

PART VI. MODULES «EMAIL AGENTS»

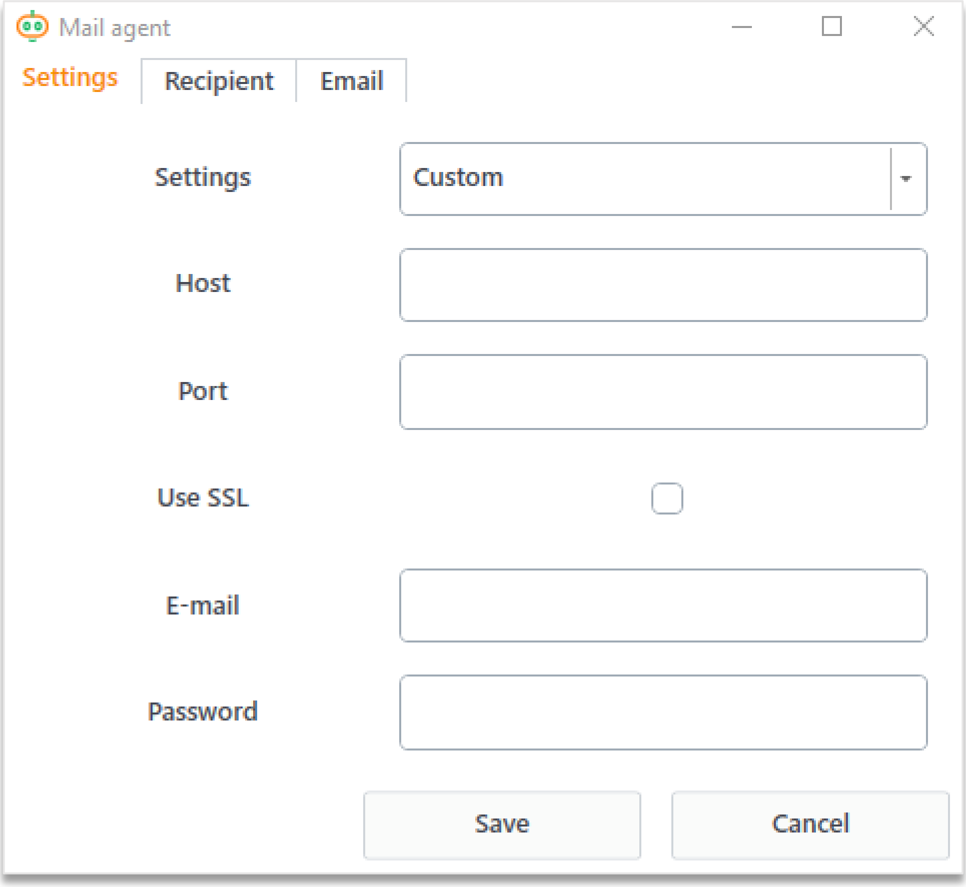
Chapter 16. Module «Send email»

Module «Send email» is designed for sending emails to one or multiple addresses.

Module interface

Module consists of multiple tabs: «Settings», «Receiver», «Email». Рассмотрим по порядку все вкладки. Начнем знакомство с рассмотрения вкладки «Настройки».

«Settings» tab



Settings: Custom

Host:

Port:

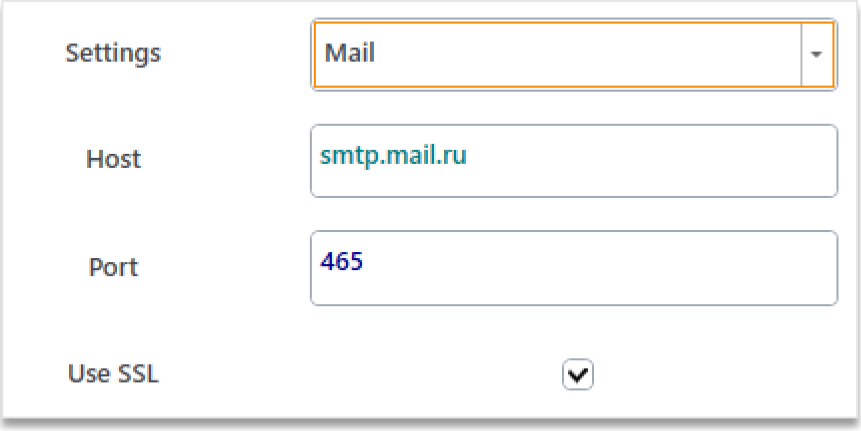
Use SSL: □

E-mail:

Password:

Figure. 6.16.1. Window module

The «Settings» field – is a drop-down list where you can choose between multiple mail agents. When choosing a mail agent the host and the port will be filled and the «Use SSL» checkbox will be checked according to its settings.The "Custom" item provides the ability to independently configure access settings for the mail service.



Settings: Mail

Host: smtp.mail.ru

Port: 465

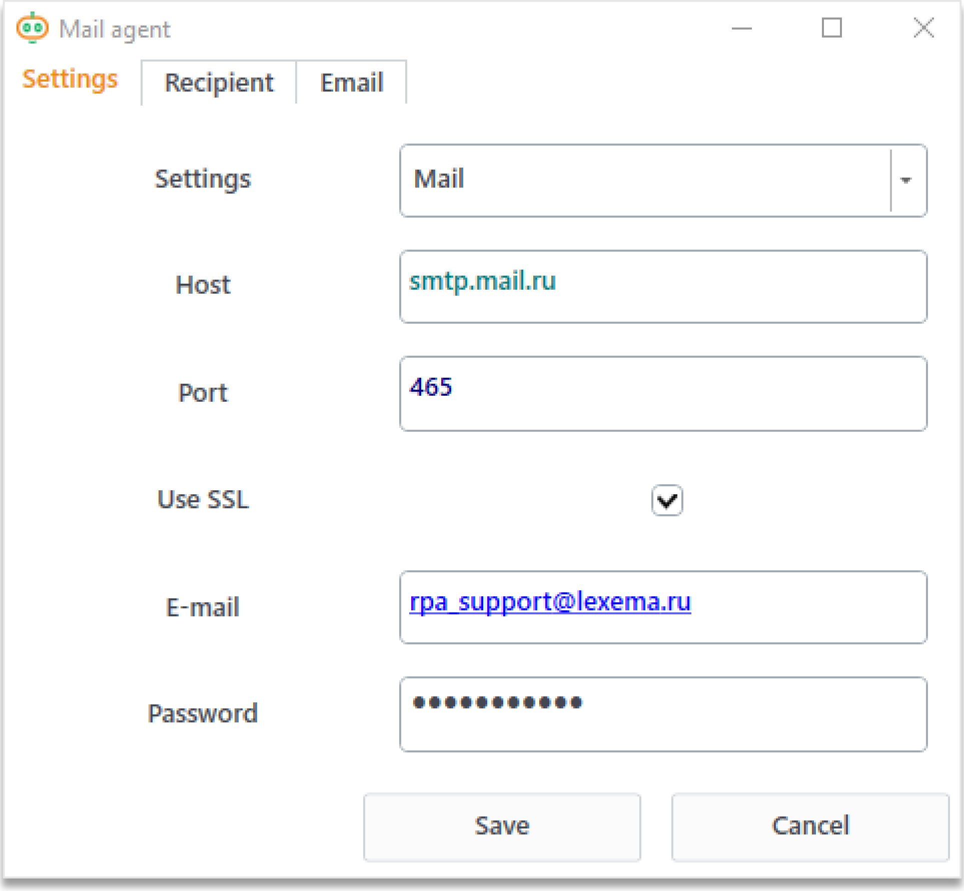
Use SSL: ☑

Figure. 6.16.2. Configuring a tab

Fields «Host», «Port» and «Use SSL» are required to be filled out only when configuring user settings.. They are a set of required fields for connecting to mail agents.

The «E-mail» fields are intended for specifying the email address, from which the messages are going to be sent name@mail.ru.

The «Password» field is intended for entering a password from the mail agent, specified in the «E-mail» field.



Settings: Mail

Host: smtp.mail.ru

Port: 465

Use SSL: ☑

E-mail: rpa_support@lexema.ru

Password: ●●●●●●●●●●●

Figure. 6.16.3. An example of the filled out «Settings» tab

«Recipient» tab

On this tab the information about the email recipients should be filled.

Module offers different methods of specifying recipients:

- manually entering a list of recipients (names and email addresses);
- from an element variable, in the form of one email;
- from a list variable, in the form of several emails.

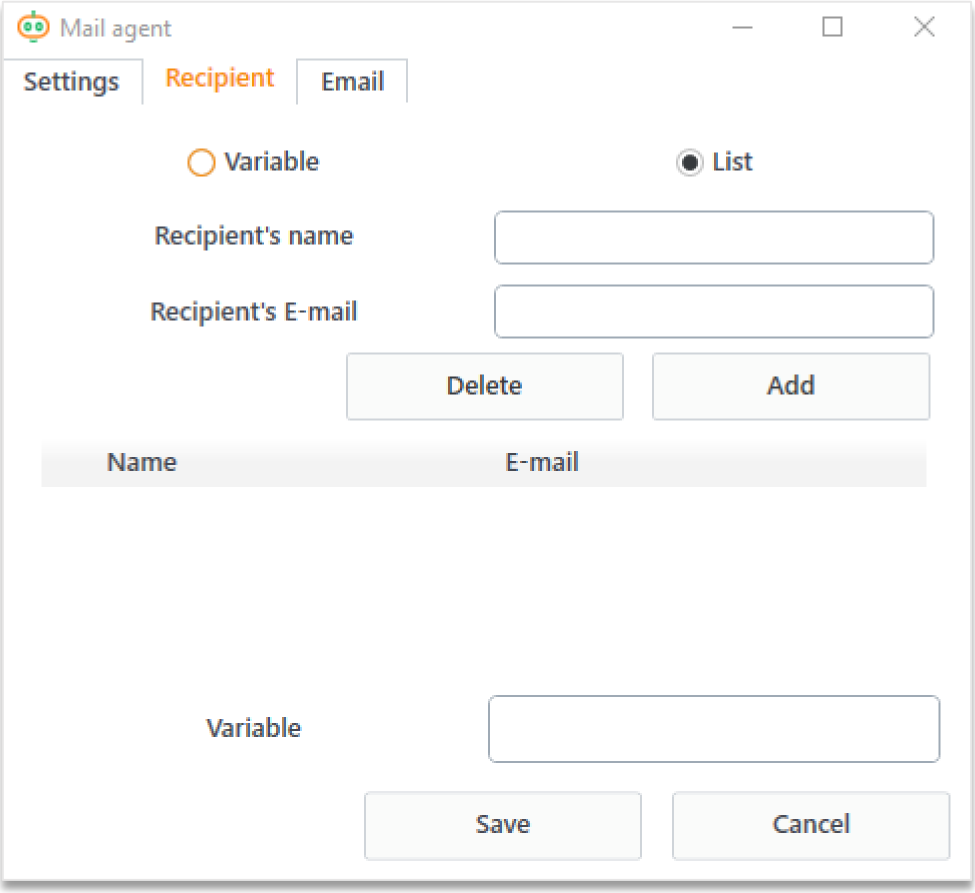
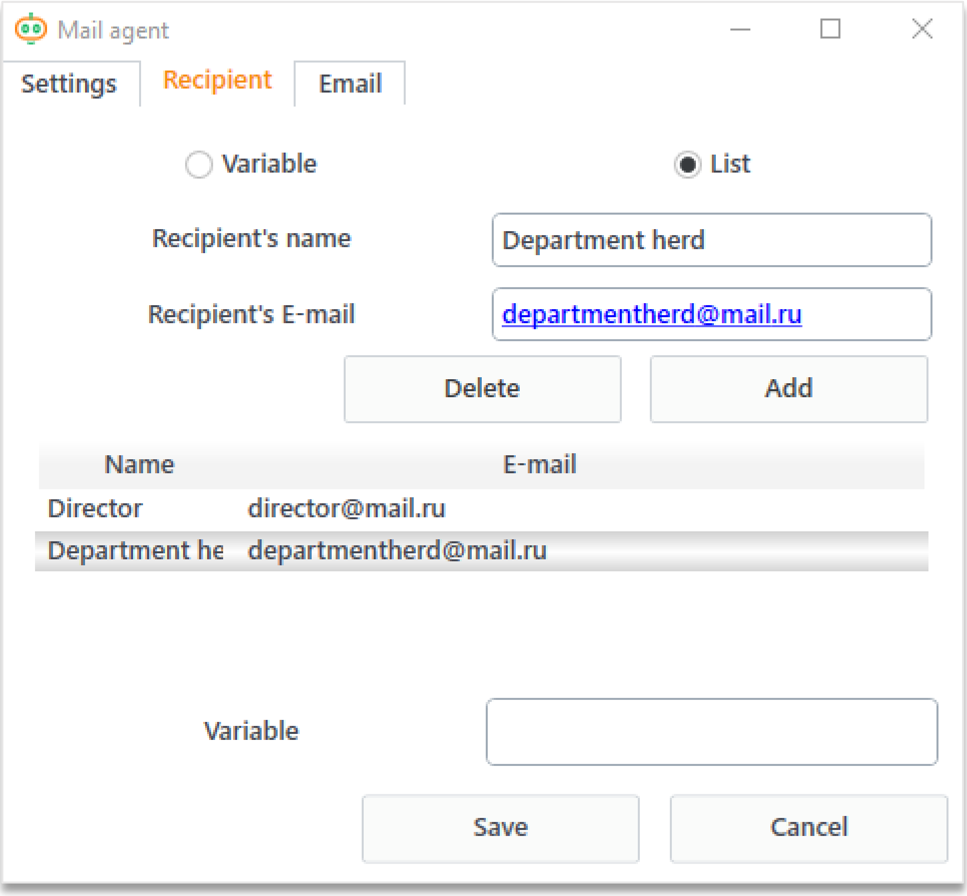


Figure. 6.16.4. «Recipient» tab

If the «List» switch is selected, then «Variable» field is available for input, otherwise «Variable» field is not available, but all other fields are.

To enter recipients using the «List» method, it is necessary to fill in the «Recipient's name» and «Recipient's E-mail» fields, then click on the button under the «Add» fields. After that, you will see the values you entered in the table in the center of the window. If you need to remove any recipient from the table, you must select a row with it and click on the «Delete» button.



○ Variable ● List

Recipient's name: Department herd

Recipient's E-mail: departmentherd@mail.ru

Name	E-mail
Director	director@mail.ru
Department he	departmentherd@mail.ru

Variable:

Figure. 6.16.5. Example of filling out a table of recipients

If the method of selecting recipients through a variable is used, then the fields «Recipient's name»,«Recipient's E-mail» and the table become unavailable. In the «Variable» field, you must enter the name of the variable starting with «@», from which the email’s of the recipients will be taken.

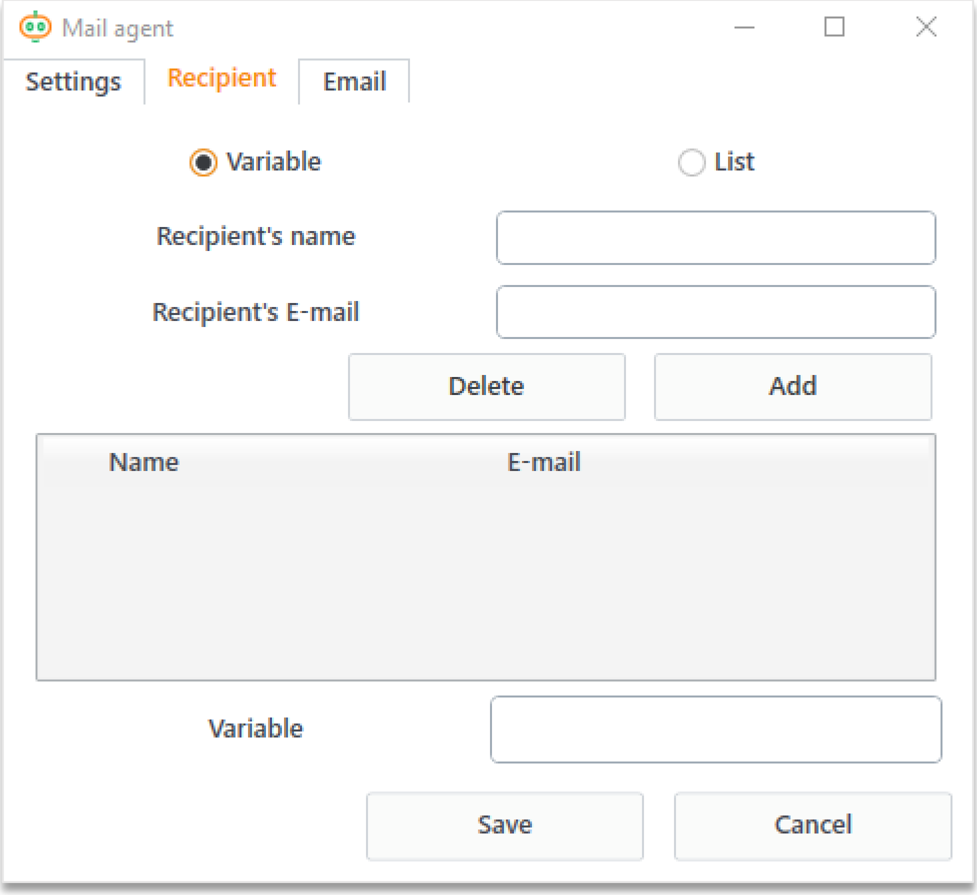
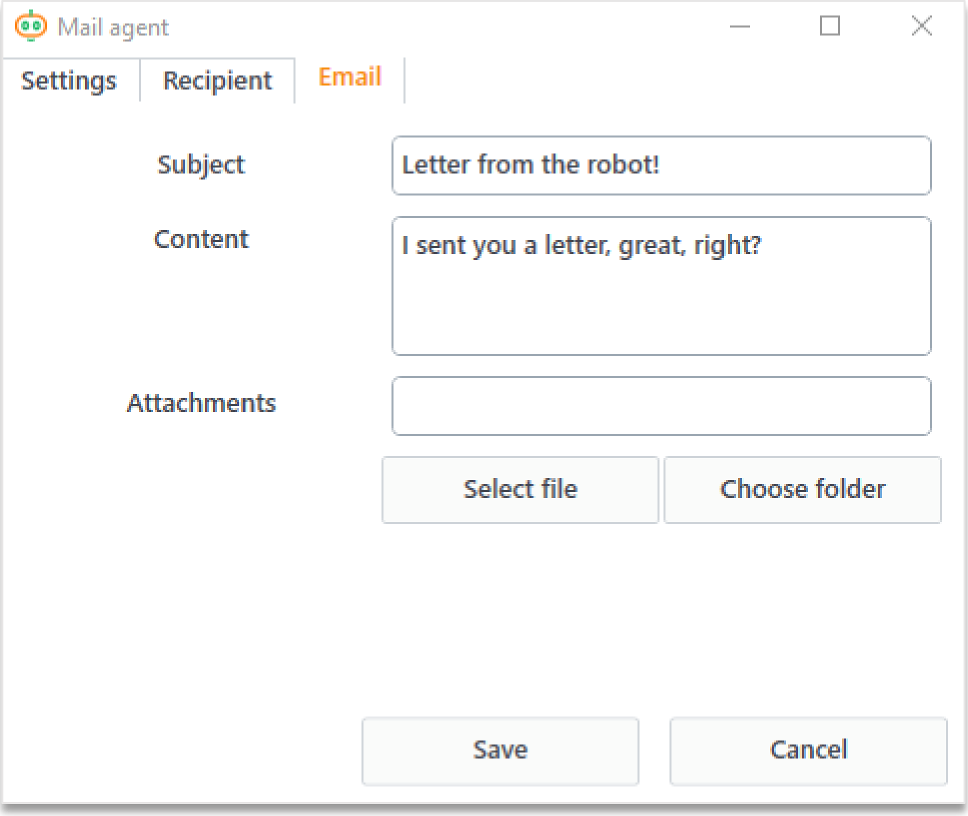


Figure. 6.16.6. «Recipient» tab

«Email» tab

On this tab the email itself is configured. All fields are required to fill out.



Subject: Letter from the robot!

Content: I sent you a letter, great, right?

Attachments:

Figure. 6.16.7. «Email» tab

To attach a file you need to click the «Select file» button and select the file in the opened window.

To attach multiple files you need to click the «Choose folder» and choose the folder you need in the opened window.

Any of these fields can be filled with a variable

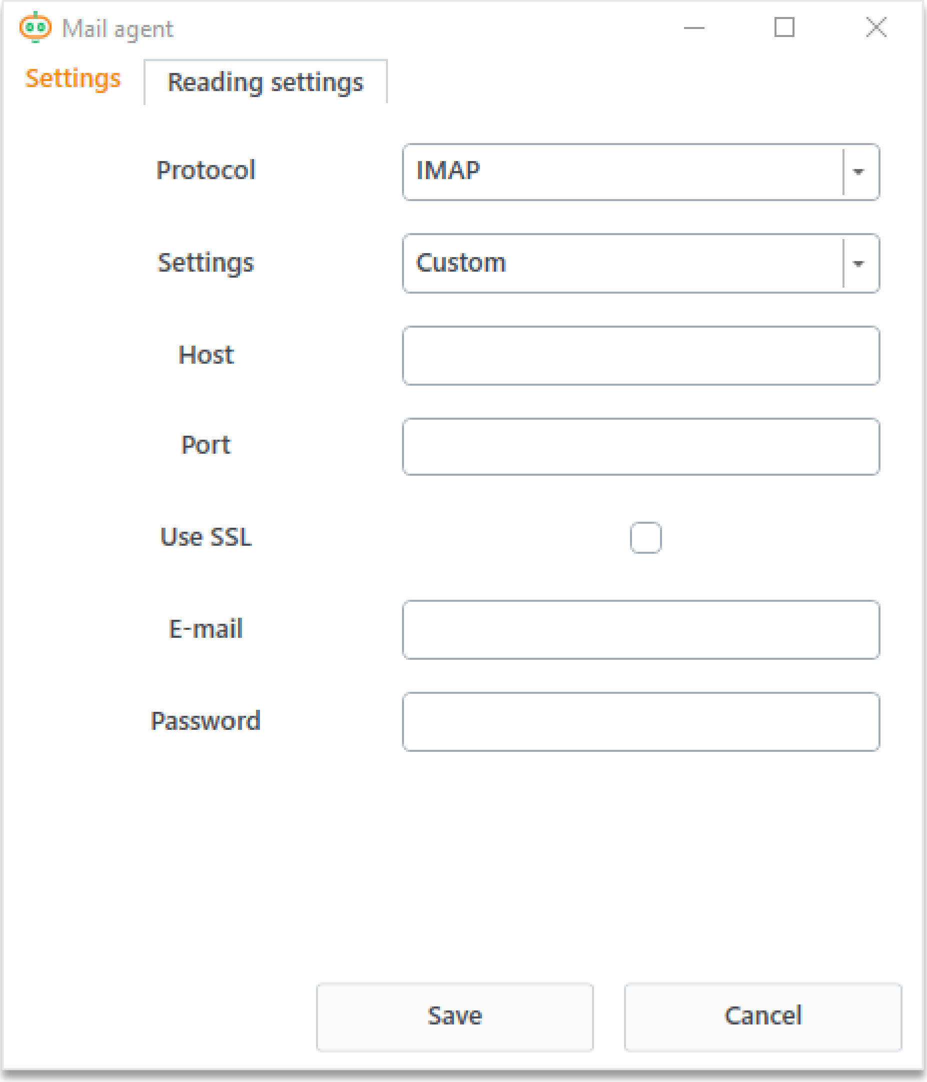
Chapter 17. «Read emails» module

The «Read emails» module is intended for getting a list of emails from a mail agent for further processing or just to download attachments.

Module interface

Module consists of two tabs: «Settings» and «Reading settings».

The «Settings» tab is similar to the tab with the same name in the «Send email» module, it has only one difference - this module provides the selection of sending protocols: IMAP or POP3. When you select the protocol the fields «Host», «Port» and «Use SSL» are filled automatically (except «User settings»).



Protocol: IMAP

Settings: Custom

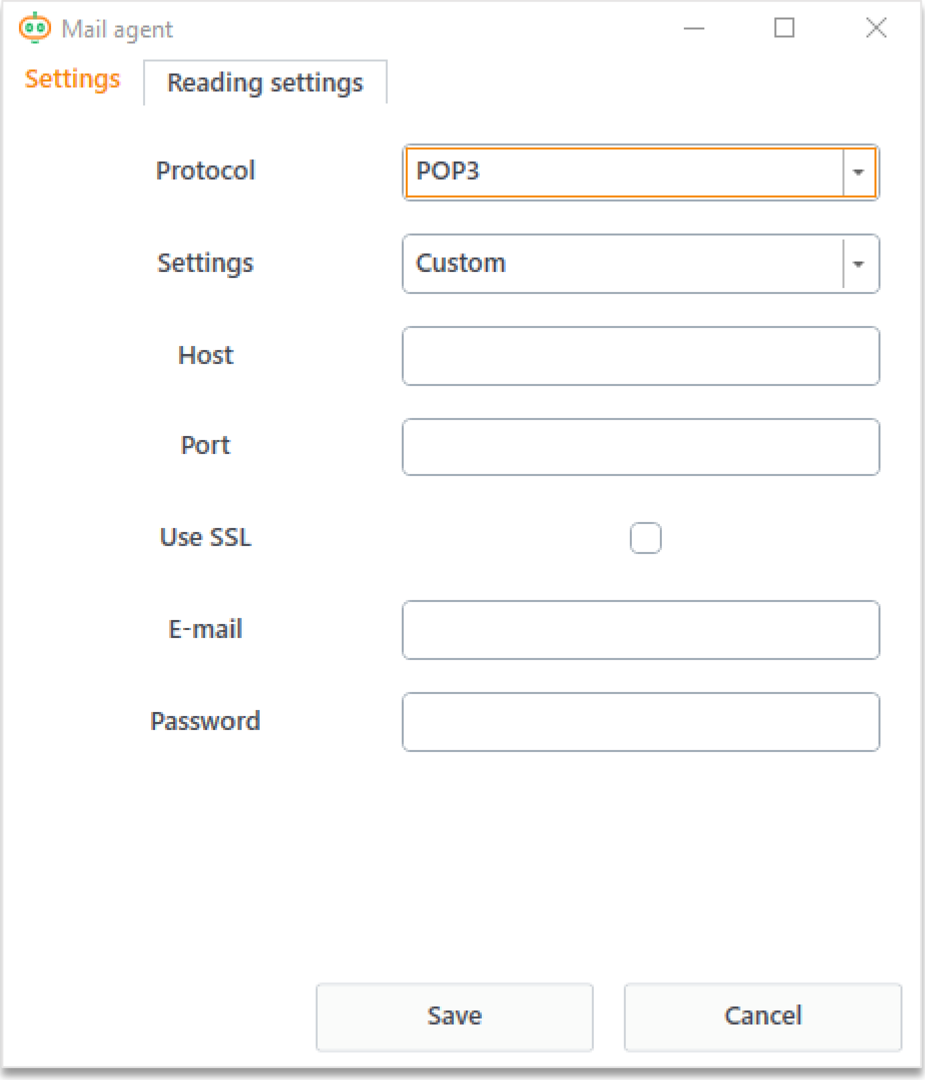
Host:

Port:

Use SSL: □

E-mail:

Password:



Protocol: POP3

Settings: Custom

Host:

Port:

Use SSL □

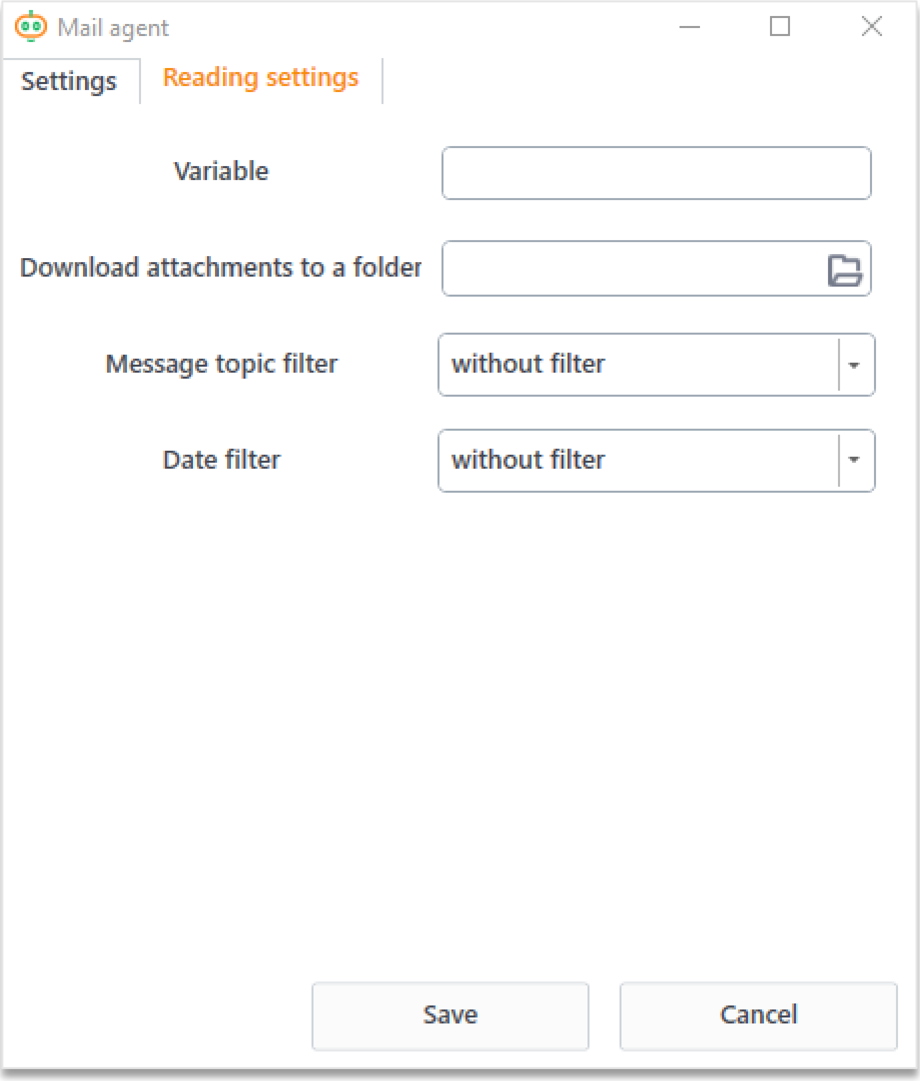
E-mail:

Password:

Figure. 6.17.1. Module window

Tab «Reading settings»

The «Reading settings» tab is a set of fields that is filled, depending on the type of emails the robot needs to read.



Variable

Download attachments to a folder

Message topic filter: without filter

Date filter: without filter

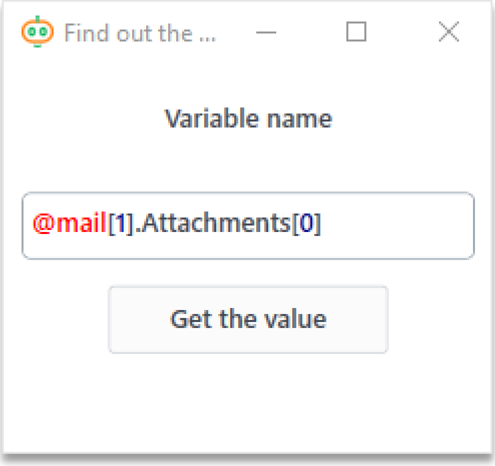
Figure 6.17.2.The «Reading settings» tab

Field «Variable». The field is intended for entering the name of a variable, in which the information about the read emails is gonna be put in. In the program, this field will be represented by a list of objects that have four fields:

1) Subject – the subject of the email;
2) Body – the body of an email, its main text;
3) Sender – the sender of a letter;
4) Attachments – list of paths to the downloaded attachments.

Examples of working with variables

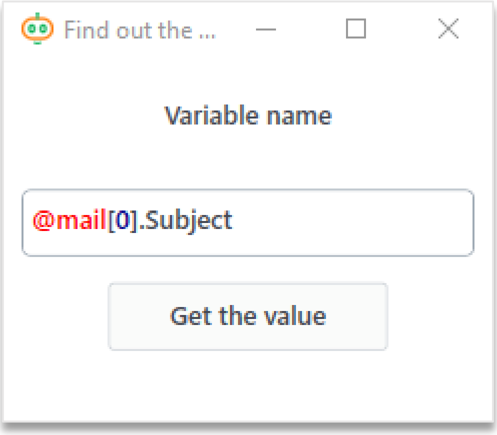
- @mail[1].Attachments[0] – getting the path to the first attachment of the second email.



@mail[1].Attachments[0]

Figure. 6.17.3. An example of getting the path to the first attachment of the second email.

- @mail[0].Subject – getting the subject of the first downloaded message :



@mail[0].Subject

Figure. 6.17.4. An example of getting the subject of the first downloaded message

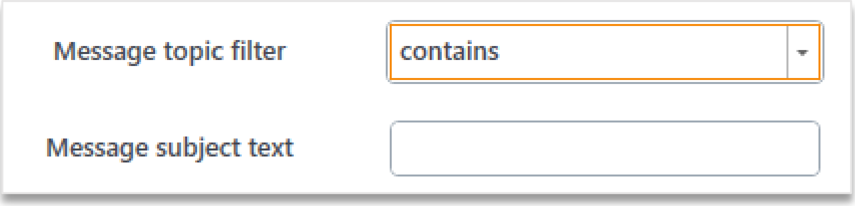
The «Download attachments into the folder» field is intended to enter the path to the folder into which all attachments of the read emails will be loaded. If you leave the field empty, the files will be downloaded to the temporary directory of the computer.

The «Only unread» checkbox allows scanning only unread emails. *Available only for IMAP protocol.*

The «Mark as read» checkbox allows setting a note about reading an email. *Available only for IMAP protocol.*

The «Read from folder» field is intended for a specified folder in the inbox from which the emails should be read. By default the robot reads emails from the «Inbox» folder. *Available only for IMAP protocol.*

The «Message topic filter» field is used to configure the filter by message topic. If you need to download emails with a topic that does or does not contain a specific line, then you should select the operation of the same name in the drop-down list and enter the desired line in the field that appears:



Message topic filter: contains

Message subject text:

Figure. 6.17.5. Message topic filter

The «Date filter» field is used to configure filtering of reading messages by date. When specifying a value in this field, the robot will read only those messages that satisfy the specified condition.

The following filtering methods are available:

- for a certain date – «On data»;
- starting with some data, including it – «With data»;
- all letter up to some date, including it – «To data»;
- «Between date».

After selecting a filtering method, one or two fields will appear under the «Filter by date» field to enter the desired values in the format DD.MM.YYYY.

PART VII. WORK WITH THE BROWSER

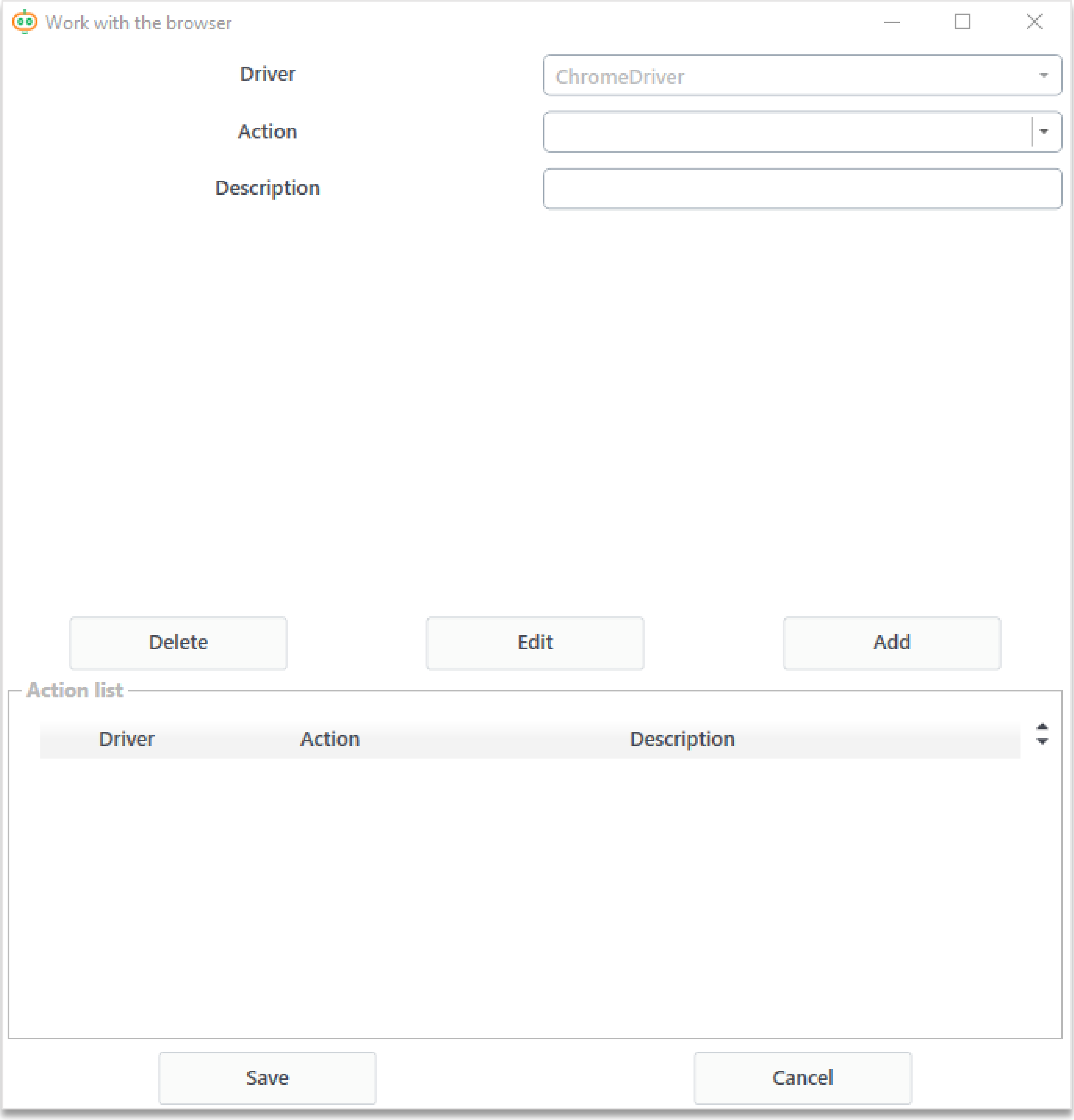
Chapter 18. Module highlights

The «Work with the browser» module is based on the Selenium WebDriver[3] and allows implementing basic elements with working with a browser page – open tab, пgo to the tab, find the element on the page by its key and process it.

Module interface

The module window is divided into the part of creating a command for working with a browser, a group of action control buttons and a table containing a list of created commands. Part of creating a command consists of several fields to fill out, and the number of these fields varies depending on the selected action in the «Action» field. The group of buttons consists of the following buttons – «Add», «Edit» and «Delete».

[3] Selenium WebDriver – is a software library for managing web bowsers.


Work with the browser

Driver: ChromeDriver

Figure 7.18.1. «Work with the browser» module window

Creating a command starts with selecting an action. After selecting it, the list of available fields in the window may change. You need to fill out the fields that are required for the action to work and click «Add». The created action is gonna be listed in the «Action list» table. To edit a command you need to select it in the list, edit the desired fields and then press the «Edit» button. But if you press the «Add» button, a new action is gonna be added but with the changes you made while editing. You can simply not change anything and click «Add», then the command is gonna be copied.

To delete actions you need to select it in the list and press the «Delete» button.

You can change the order of the commands by pressing the arrow keys that are located in the top right corner. It's really simple to do, first select the row tje position of which you wanna change, then click the arrow keys to move it.

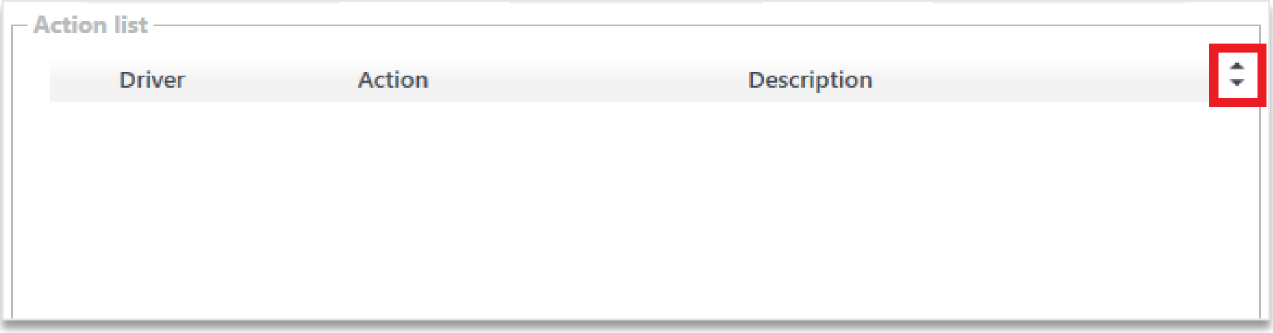


Figure. 7.18.2. Arrow keys for changing the order of commands

For all actions these three fields never change «Driver», «Action» and «Note».

The «Driver» field is currently unavailable for editing and the default driver for the Chrome browser is selected in it, but in the future it is planned to supplement this list with other browsers.

The «Action» field — is a drop-down list with all possible actions for this module. More detailed review of them is gonna be in the next chapter.

The «Note» field — a row with a note for the command.

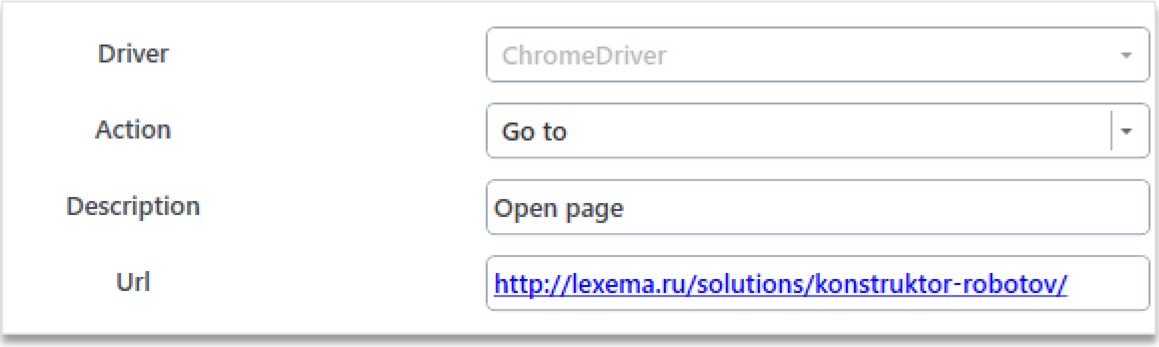
Chapter 19. Description of actions

Module offers 7 main actions – go to the page, reload the page, go back and forward, work with the page, work with tabs, and close the browser. Each action requires filling in some kind of fields

Please note that it is advisable to close the browser opened using this module, too, otherwise, if you close the browser yourself and then try to continue working with it using this module, an error will occur and you will have to restart the program.

Go to

Action goes to the specified page. The «URL» field is added.



Driver	ChromeDriver
Action	Go to
Description	Open page
Url	http://lexema.ru/solutions/konstruktor-robotov/

FIgure. 7.19.1. «Go to»

If any page is already open, then it will go from it to the specified one, meaning that the original page will be replaced with a new one. If the browser has not yet been launched, it will be opened with this page.

Reload the page

Action reloads the already opened page, before using the action make sure the page is opened.

Go back

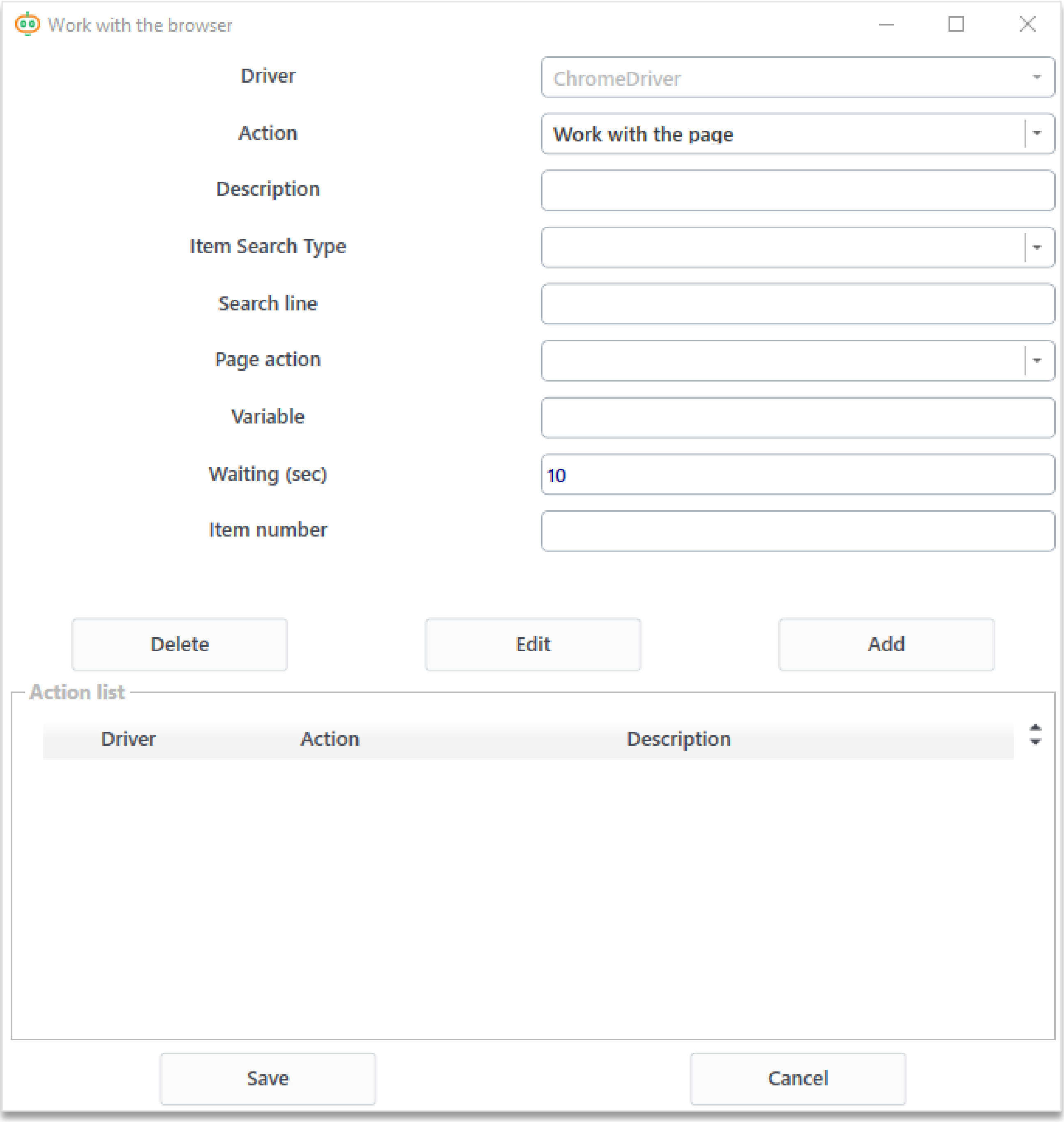
Goes to the page that was open before the current one.

Go forward

Goes to the page that was opened after the current one.

Working with a page

This action provides the ability to work with a specific item of the page. The following fields will appear when configuring the action – «Item search type», «Search line», «Page action», «Variable», «Waiting (sec)» и «Item number».



Work with the browser

Driver	ChromeDriver
Action	Work with the page
Description	
Item Search Type	
Search line	
Page action	
Variable	
Waiting (sec)	10
Item number	

Action list

Driver	Action	Description

Figure. 7.19.2. «Working with a page» window

The «Item search type» field is a drop-down list with a list of search methods for the element on the page, it is used in conjunction with the **«Search line»** field. Below will be listed all types of searches for an element with examples of filling the «Search line» field:

- GetElementsByName – search for elements by their name property.

```
<div id="element">
     <p>some content</p>
</div>
```

Figure. 7.19.3. «Search line» to find div – element.

- GetElementsById –search for elements by their id property.

```
<div name="element">
     <p>some content</p>
</div>
```

Figure. 7.19.4. «Search line»to find div – element.

- GetElementsByClass – search for elements by their class property.

```
<div class="myClass">
     <p>some content</p>
</div>
```

Figure. 7.19.5. «Search line» to find div – myClass element.

- GetElementsByCssSelector – search for elements by their css property

```
<div class="main">
    <p>text</p>
    <p> Another text </p>
</div>
```

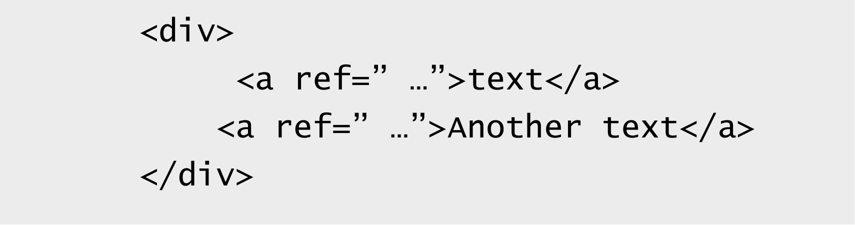
Figure. 7.19.6. «Search line» finding div – div.main element.

- GetElementByLinkText – search for link elements in text. Only items with a complete match are searched.

```
<div>
     <a ref=” …”>text</a>
    <a ref=” …”>Another text</a>
</div>
```

Figure. 7.19.7. «Search line» for finding the first element, text is in example.

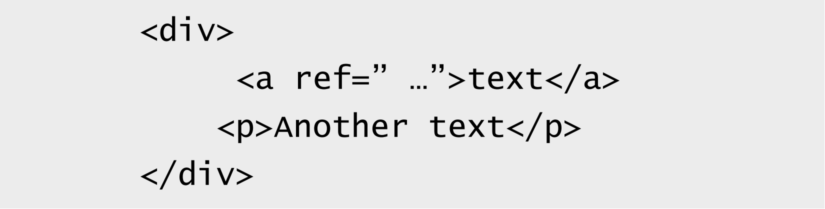
- GetElementsByPartialLinkText - search for link elements by their text. All items containing the given line are searched.



```
<div>
     <a ref=" …">text</a>
    <a ref=" …">Another text</a>
</div>
```

Figure. 7.19.8. «Search line» for finding two a – text elements.

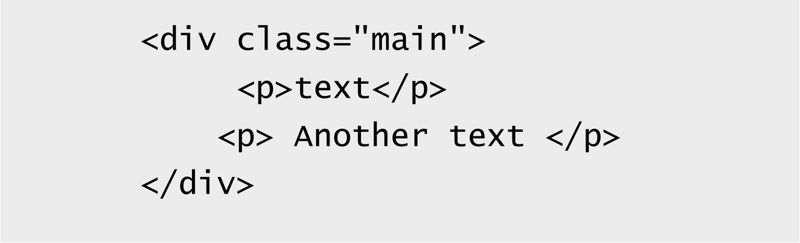
- GetElementsByTagName – search for elements by their tag[4].



```
<div>
     <a ref=" …">text</a>
    <p>Another text</p>
</div>
```

Figure. 7.19.9. «Search line» for finding the a – a element.

- GetElementsByXPath – search for elements by HTML – xpath query language.



```
<div class="main">
     <p>text</p>
    <p> Another text </p>
</div>
```

Figure. 7.19.10. «Search line» for finding the div - //div[@class='main'] element .

The «Page action» field offers a selection of an action from the drop-down list that must be performed on the found elements. The following actions are suggested:

- Record – enters specified text into the «Variable» field (for example, after clicking on a specific element to enter text);
- Get text - gets the value of the text properties of all found or specifically selected elements and returns the result as a list to the variable;
- Send - press ENTER - use the recorded data (for example, any text can be entered with actions «Record», and then «Send» it);
- Click – left-click on a specified element (filling out the «Item number» field is required);

[4] Tag is an element of a markup language.

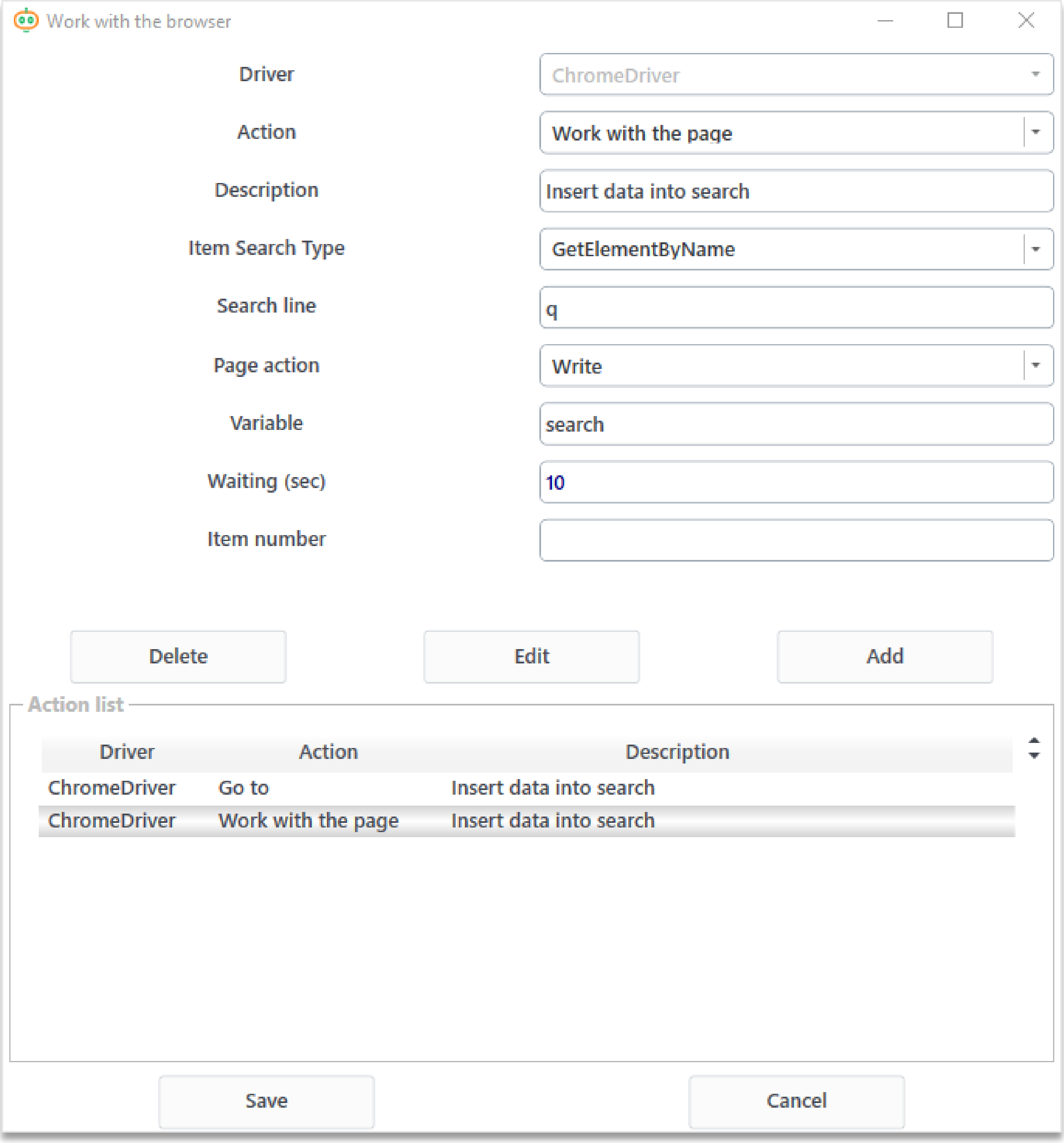
- Hover over a cursor – hovers over a specific element(filling out the «Item number» field is required);
- Get Property – when choosing an action the «Property name» field appears - returns the value of the specified in the «Property name» field property of all found elements or the specified as a list into the variable;
- Get CSS property – when choosing an action the «CSS property» field appears - returns the value of the specified in the «CSS property» field property of all found elements or the specified as a list into the variable;
- Get Attribute – when choosing an action the «Attribute property» field appears - returns the value of the specified in the «Attribute property» field property of all found elements or the specified as a list into the variable;

In the «Variable» field the name of a variable is entered starting with «@», in which the result of the executed command is gonna be put in, or otherwise from which the values for executing a command are gonna be taken from.

The «Waiting (sec)» field – the amount of seconds robot needs to wait after executing a command.

The «Item number» field – one index of the found item to which the action must be applied

An example of the filled out action:



Work with the browser

Driver	ChromeDriver
Action	Work with the page
Description	Insert data into search
Item Search Type	GetElementByName
Search line	q
Page action	Write
Variable	search
Waiting (sec)	10
Item number	

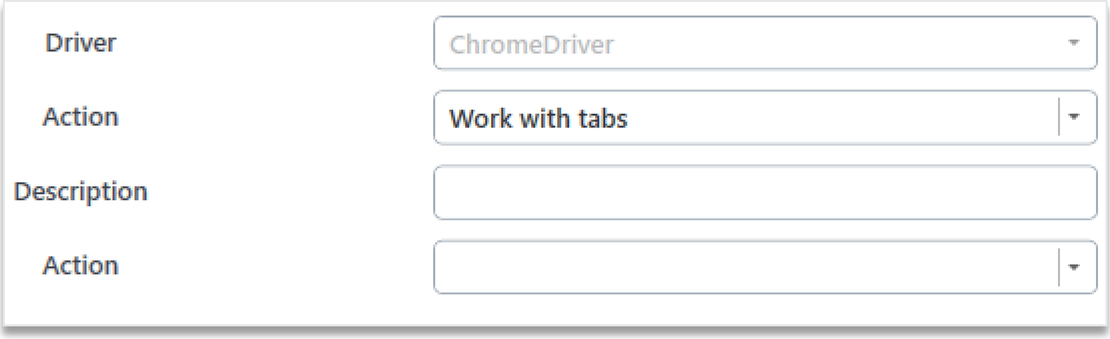
Driver	Action	Description
ChromeDriver	Go to	Insert data into search
ChromeDriver	Work with the page	Insert data into search

Figure. 7.19.11. An example

Here, the element with the name «q» is searched and the text «search» is put in it.

Work with tabs

This action allows manipulating with tabs in the browser window. For configuring at least one field is added – «Action», and other fields appear after you select the action.



Driver: ChromeDriver

Action: Work with tabs

Description:

Action:

Figure. 7.19.12. Fields when choosing an action «Work with tabs»

Available actions for working with tabs - add, close, go to the tab and get tab amount. More detailed description of them:

- Add tab - adds a new tab, the tab automatically becomes active;
- Close tab - closes the active tab;
- Go to the tab – when choosing this action the «Tab number» field will appear - goes to the tab the number of which is specified in the «Tab number» field (starting with 0);
- Get tab amount –when choosing this action the «Variable» field will appear - returns the amount of tabs into the specified variable.

Close browser

Closes the browser.

PART VIII. OTHER MODULES

Chapter 20. Module «Calculation»

The «Calculation» module is a connecting module or a module for pre- and post-processing data from other modules. This module is necessary for creating, calculating and changing values of variables.

Module interface

The module window is really simple and consists only of an entry field and the «Save» and «Cancel» buttons.

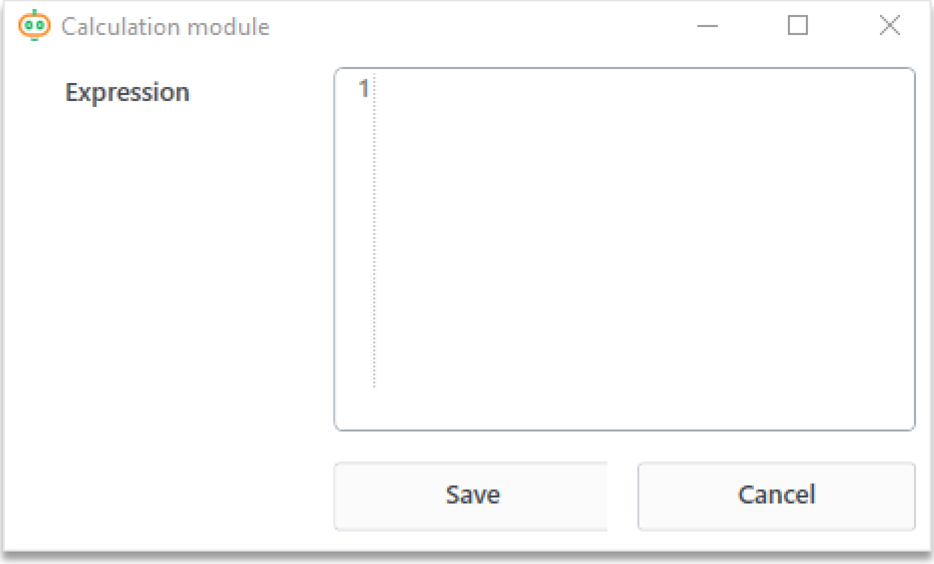


Figure. 8.20.1. Module window

The «Expression» field. The operations that need to be done with variables are entered into this field. This can be addition, subtraction, multiplication or function calculation.

Supported operations

- + (addition);
- - (subtraction);
- * (multiplication);
- / (devision);
- % (remainder of the division);
- \ (integer from division);
- ^ (exponentiation).

All lines should look like this «Variable = Operation;», operation can consist of a single number or a line, as well as several operands connected by the operators specified above, or it can be a function.

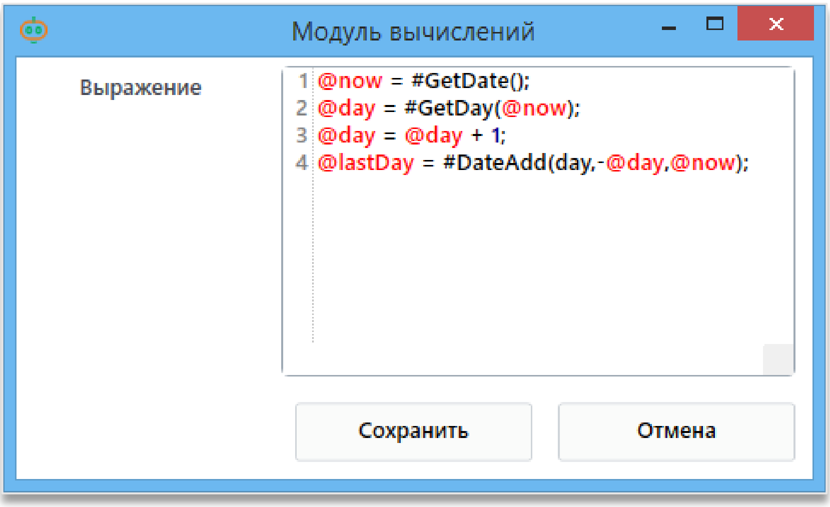
Each line should be separated by the «;» symbol.

Example of working with the module

Suppose you want to get the last day of the previous month.

You can do this by entering the following code:

```
       @now = #GetDate();
       @day = #GetDay(@now);
       @day = @day + 1;
       @lastDay  =  #DateAdd(day,
   -@day, @now);
```



```
1 @now = #GetDate();
2 @day = #GetDay(@now);
3 @day = @day + 1;
4 @lastDay = #DateAdd(day,-@day,@now);
```

Figure. 8.20.2. An example of filling in the module

What are we doing here? Here, we're getting today's date, for example, today is 25th of march, subtract one day more from today’s date than it actually is, meaning that, 26 days and end up in the last day of the previous month - February 29 (if the year is a leap year). In more details:

1st line – Using GetDate function which returns today's date and saves its result in a variable @now (@now = 25.03.2020);

2nd line - the use of the GetDay function, which takes one parameter - the date, and returns only the day from the date supplied to it. Let's set today's parameter as the parameter, which lies in the @now variable. We enter the result of the function to the variable @day (@day = 25);

3rd line – increasing the @day variable by one (@day = 26);

4th line – using the DateAdd function, which takes three parameters: the first parameter - «day», «month» or «year» - is the component of the date that we are working with, the second parameter is the amount that is added to the selected component of the date, and the third parameter is the date with which the manipulation is performeed. Subtract 26 days from today's date - use DateAdd with parameters:

1. day – subtracting date ;
2. –@day – we add the minus because the function adds by default
3. @now – today's date.

In result, we subtract 26 days from 25.03.2020 and end up on 29.02.2020.

Chater 21. Module «Process startup»

The «Process startup» module is intended to launch different applications. This can be the browser, MS Word, 1C and more.

Module interface

Module window consists of the following fields: «File path», «Parameters», «As administrator» checkbox, «Pause», «Variable with process name» and «Variable with process id», and also the «Save» and «Cancel» buttons.

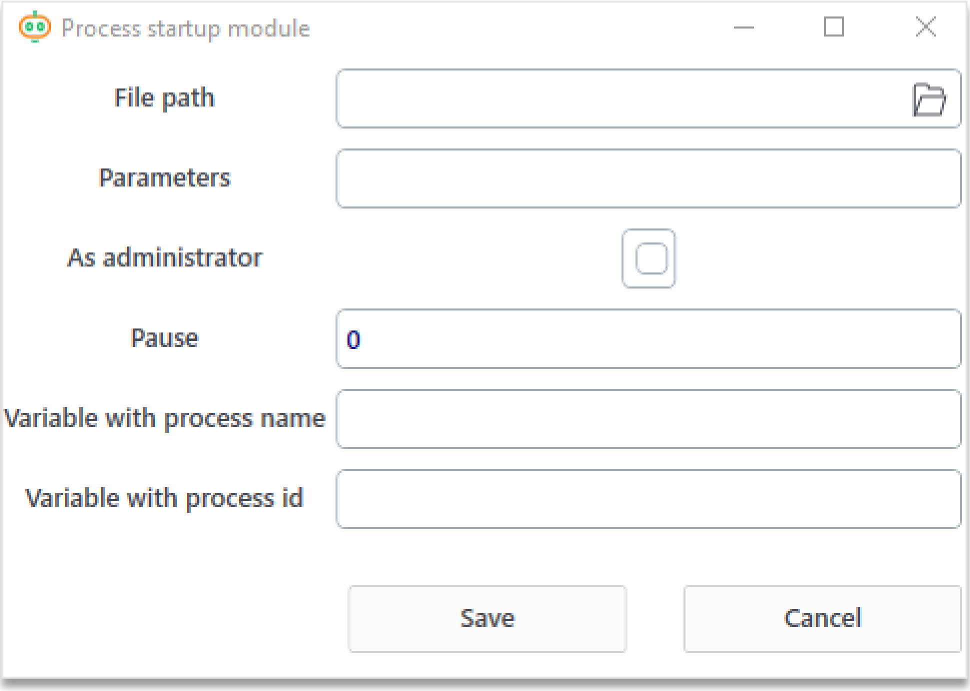
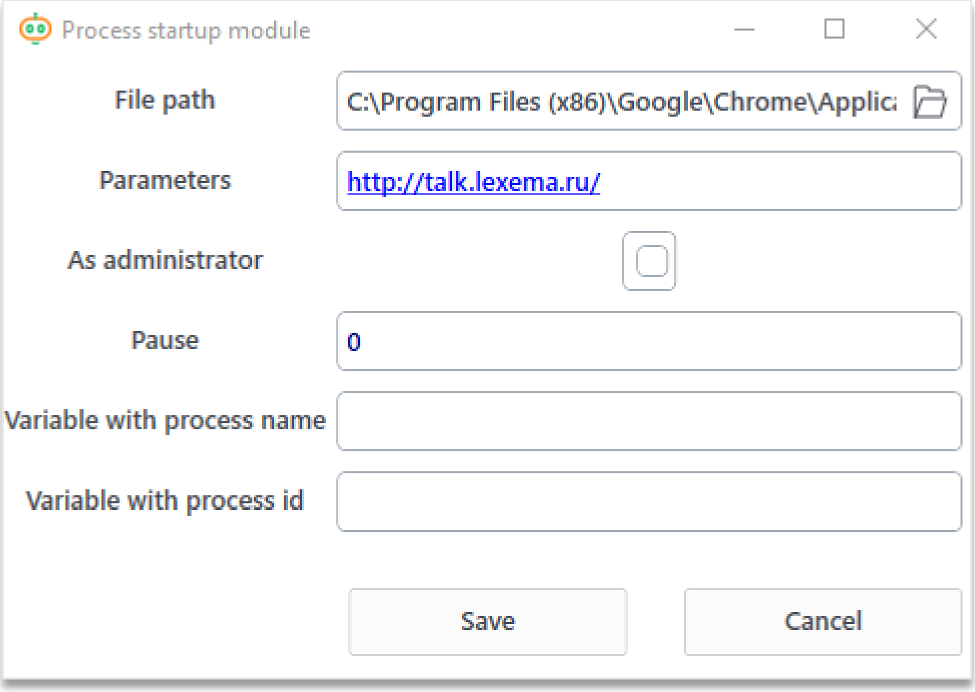


Figure. 8.21.1. Module window

The «File path» field. The name of the standard process in Windows is entered into this field (for example notepad), or the root directory of the application, for example, standard path to the Google Chrome browser - C:\Program Files (x86)\Google\Chrome\Application\chrome.exe.

The «Parameters» field is an optional field, arguments of the launched up are entered into it. For example, to open a new notepad window or a blank sheet, Excel arguments are not needed, but if you need to open a specific file, the name of the file is passed to the arguments. And if you need to open a browser the link to the website can be an argument.



Process startup module

File path: C:\Program Files (x86)\Google\Chrome\Applic

Parameters: http://talk.lexema.ru/

As administrator: □

Pause: 0

Variable with process name:

Variable with process id:

Figure. 8.21.2. An example of the filled out window

The «As administrator» checkbox launches the process with the right to change the system files. ***If you don't trust an application, don't check this checkbox.***

The «Pause» field sets the wait time after the start of the process.

The «Variable with process name/id». In this field the name and the id of the process you are launching are going to be put in. These parameters are needed to end (close) the specified processes. ***If you only need to start processes, you can leave these fields empty.***

Chapter 22. «Process completion» module

The «Process completion» module is intended to finish already started processes by their name and id.

Module interface.

The window consists of two fields - the name and the id of the process.

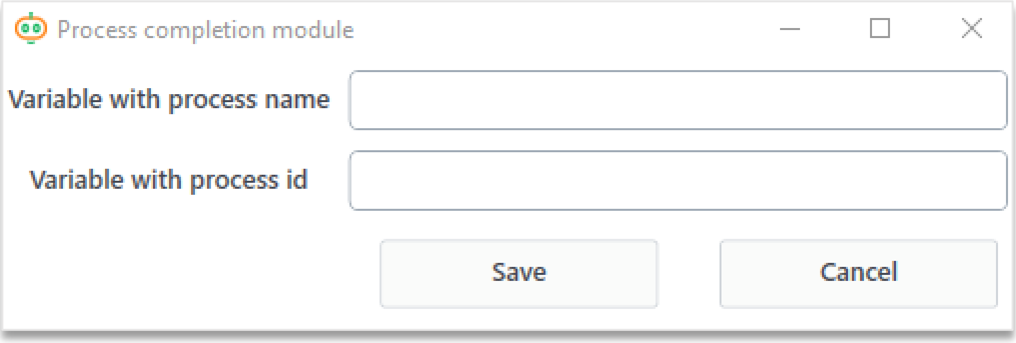


Figure. 8.22.1. Module window

If a process was launched through the «Process startup» module, then the variables specified in the fields with the same name in the process launch window are transferred to these fields. Otherwise if you only know the name of the process that you wanna complete, you can just enter it, (!) but in this case ***all processes with the similar name will be finished.*** For example, if you want to close an Excel file and just enter «Excel», then all opened excel files will be closed.

Chapter 23. «C# code» module

The «C# code» module is designed to use complex calculations that are not available in the Studio.

The module allows the use of Studio variables.

Module interface

This module consists of two fields: «Variable» and «Code».

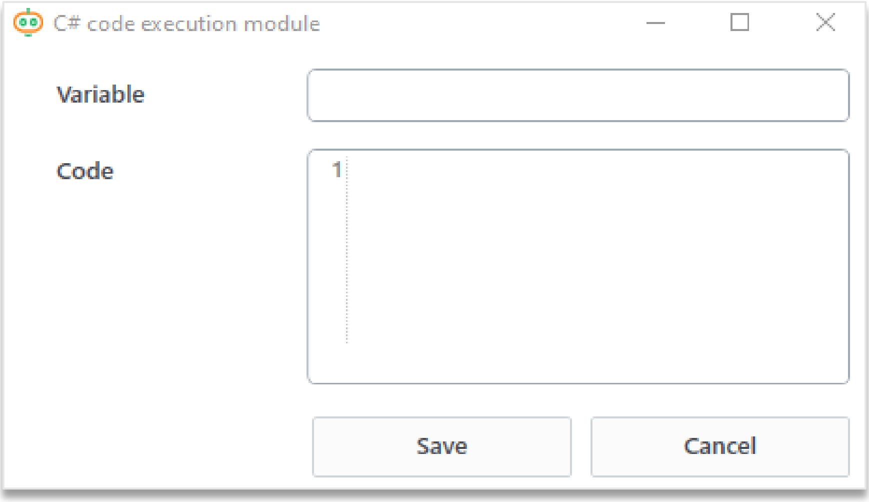
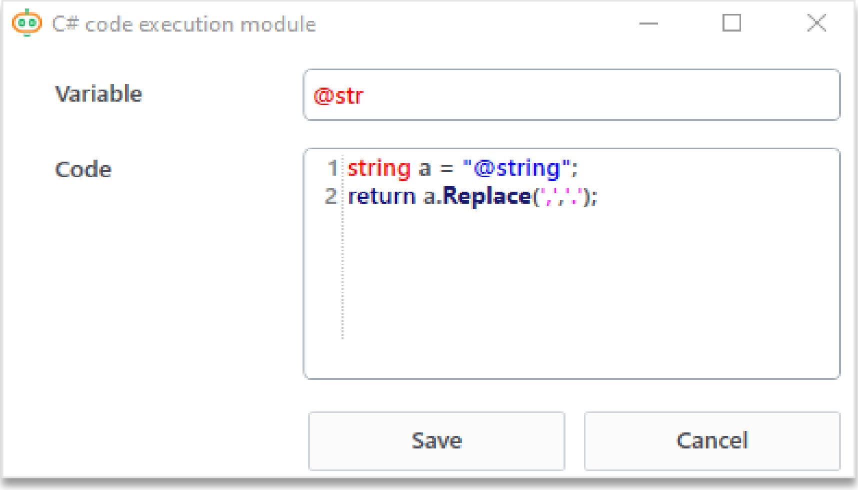


Figure. 8.23.1. Module window

The «Variable» field.The field indicates the variable in which the result will be entered in.

The «Code» field. It is necessary to enter part of the code in the field in accordance with the C # language syntax without describing the libraries used, as if it were a function inside an existing project, which has one main condition - **it must return something**, that is, have the word ***«return»***.



Variable: @str

```
string a = "@string";
return a.Replace(',','.');
```

Figure. 8.23.2. An exmaple

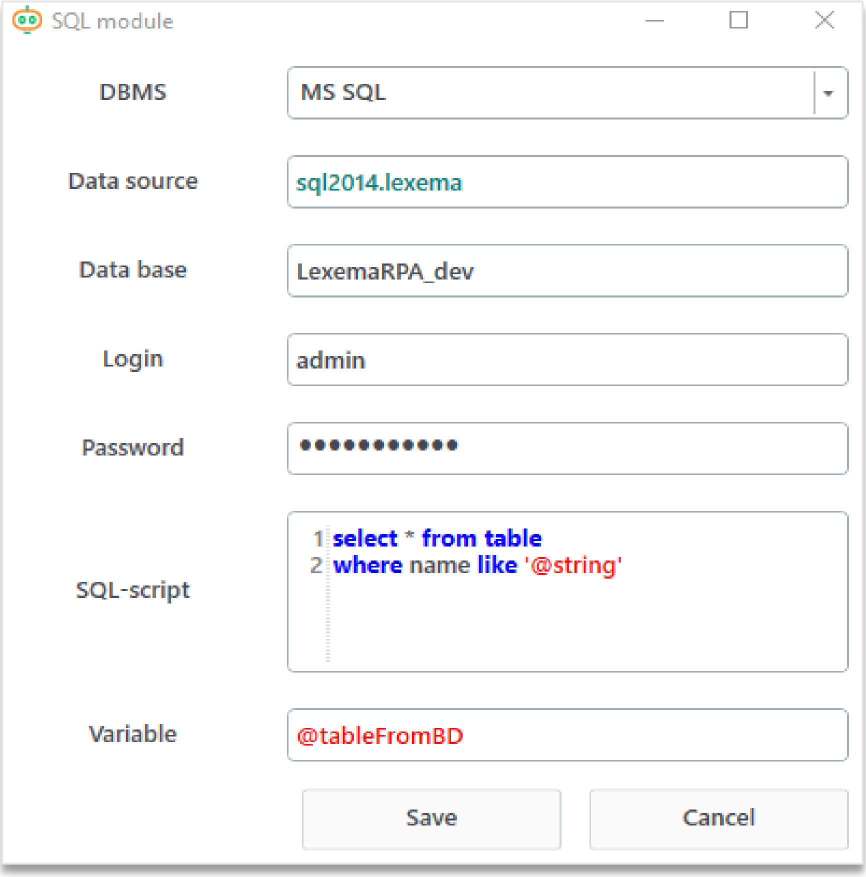
Chapter 24. «SQL» module

The «SQL» module provides work with databases. Supported DBMSs: MS SQL and PostgreSQL.

Module interface

This module provides work with databases. The following DBMSs are supported: MS SQL and PostgreSQL. To work with a database you need to fill out all the fields in the «SQL» module window.

The window consists of the following fields: «DBMS», «Data source», «Data base», «Login», «Password», «SQL-script» and «Variable».



DBMS: MS SQL

Data source: sql2014.lexema

Data base: LexemaRPA_dev

Login: admin

Password: ●●●●●●●●●●●

SQL-script:

```
select * from table
where name like '@string'
```

Variable: @tableFromBD

Figure. 8.24.1. Example of the filled out window

The fields «Data source», «Database», «Login» and «Password» provide a connection to the database.

The «SQL-script» field. The script in SQL language is entered into this field, entering variables of the Studio is supported (as in Figure. 7.20.1).

The «Variable» field. The name of the variable is entered in the field into which the response from the database will be entered in the form of a table. *Only one table can be returned in one SQL module, subsequent ones will be ignored.*

Chapter 25. «HTTP_GET» module

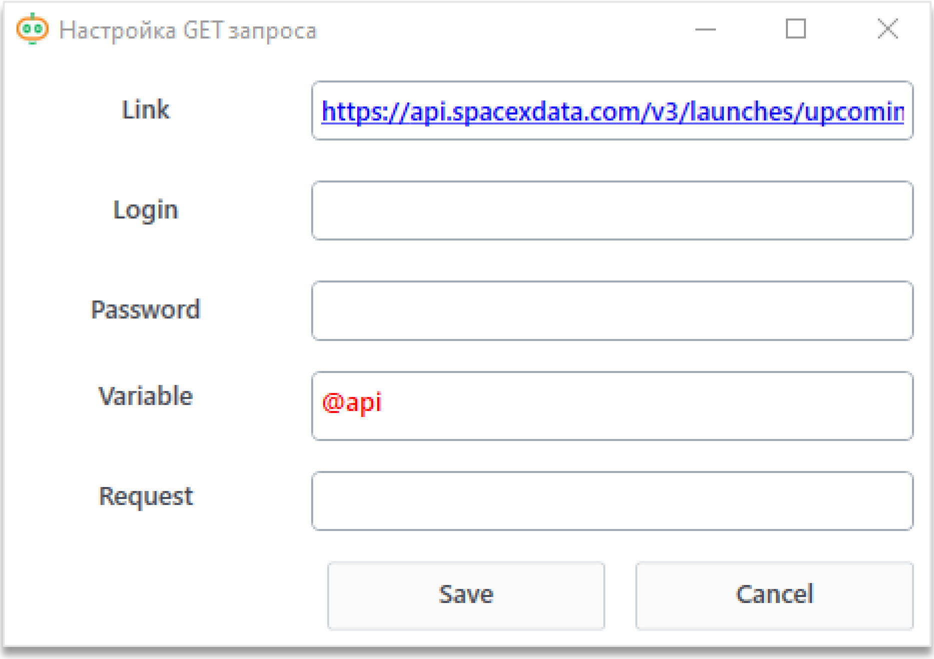
Http-requests - requests for transferring information between sites (or servers) and clients (for example, us).

The «HTTP_GET» module is designed to send Get-requests using the ***http***.[5] protocol.

To work with the received answers, you can use the «Parsing JSON» module (if, of course, the requested data format was the same).

Module interface

This module consists of the following fields: «Link», «Login», «Password» and «Variable».



Настройка GET запроса

Link: https://api.spacexdata.com/v3/launches/upcomin

Login:

Password:

Variable: @api

Request:

Figure. 8.25.1. Window of a GET request

The «Link» field . The required URL is entered into the field.

The «Variable» field. The name of the variable in which the response will be placed is entered in the field.

The «Login» and «Password» fields. *Are intended for entering the login and password, because some sites use the protection of their information and request authorization.*

[5] The **Hypertext Transfer Protocol** (**HTTP**) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen in a web browser.

Chapter 26. Module «HTTP_POST»

The «HTTP_POST» module is designed to send Post-requests using the ***http*** protocol.

Module interface

The form for sending a Post-request is similar to the form for sending a Get-request, with only one difference - there is a field for entering the request itself.

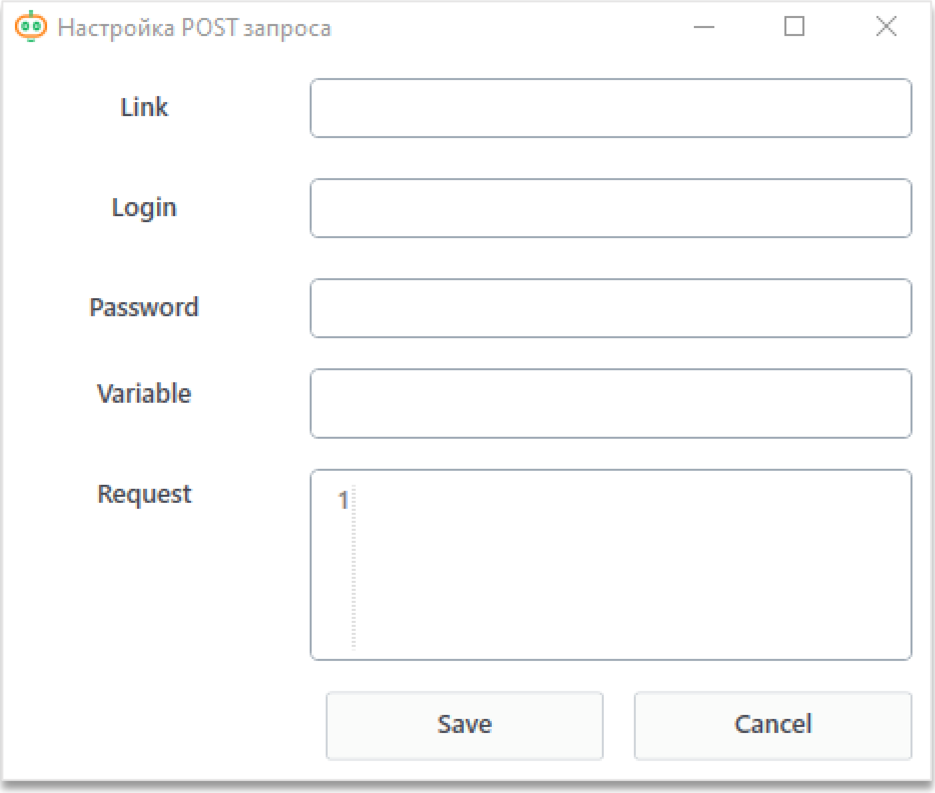


Figure. 8.26.1. POST-request window

The «Link» field. The desired URL address is entered into the field.

The «Variable» field. The name of the variable in which the response will be placed is entered in the field.

The «Login» and «Password» fields. *Are intended for entering the login and password, because some sites use the protection of their information and request authorization.*

The «Request» field. An HTTP message is entered in the field, according to the general rules of its structure.

Chapter 27. «JSON parsing» module

This module is designed to process data in the JSON format – parsing JSON-objects.

Module interface

In the module window three fields for entering values are located: «Variable» and «JSON», and the «Save» и «Cancel» buttons.

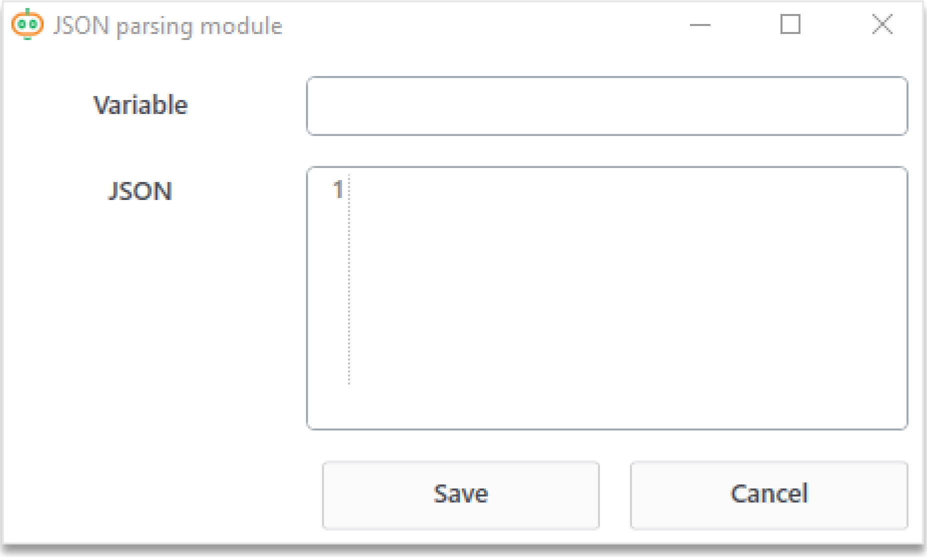


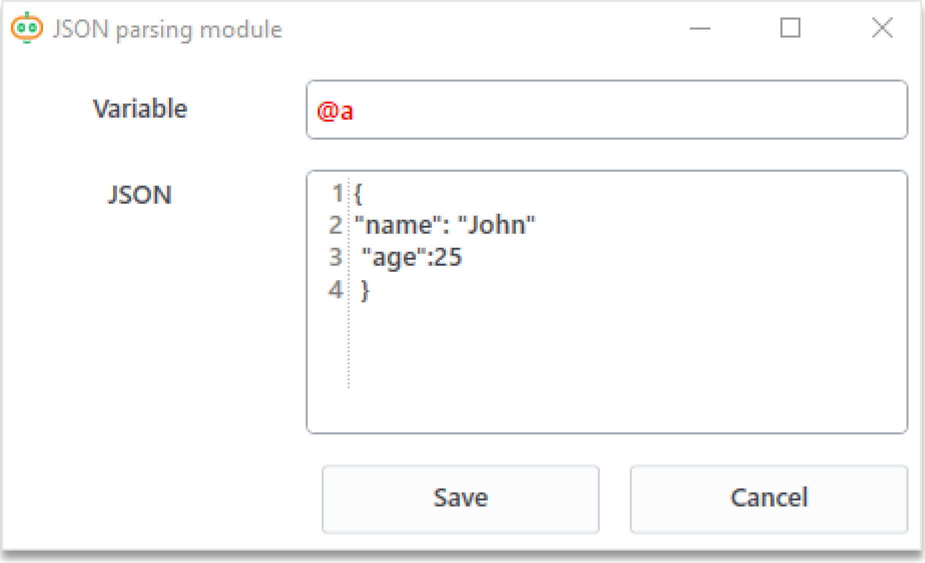
Figure. 8.27.1. Model window

The «Variable» field. The name of the variable is entered into the «Variable» field, which will be converted into an object variable with the fields available in the JSON object.

The «JSON» field. In this field you need to describe the object using the JSON format, in the form of «key»: «value», in square brackets. A JSON object can be described directly by text in the «JSON» field or can be created in the process of executing other modules. If the JSON object is formed in other modules, then you must enter the name of the variable in which the necessary object lies in.

EXAMPLE.

Let's say we have text in the JSON format. Let's enter it in the field of the JSON module and the result will be entered into the «@a» variable.

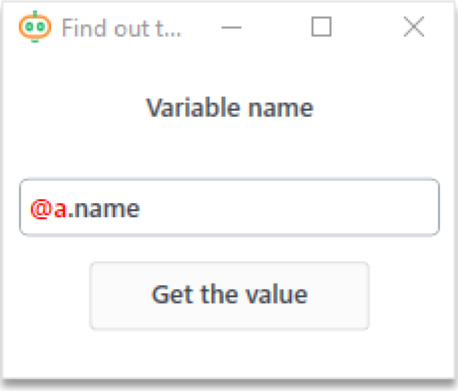


@a

```
{
"name": "John"
 "age":25
 }
```

Figure. 8.27.2.An example

As a result, the variable @a will become an object with the fields «name» and «age». You can access the fields of an object by entering the name of the field separated by a dot after the variable. You can check it by using the «Find out the value of variable», feature available by clicking the ⁎x buttons in the solution explorer.



@a.name

Figure. 8.27.3. Getting the «name» field of the «a» variable

In Figure. 8.27.3. we can see that a variable was created based on the text in the JSON format, with the fields specified in the variable and filled with values.

Chapter 28. «Pause» module

This module is designed for creating a pause between modules. All module configuration consists in filling in one field – «Pause» - the number of seconds you need to wait. The use of a variable is allowed.

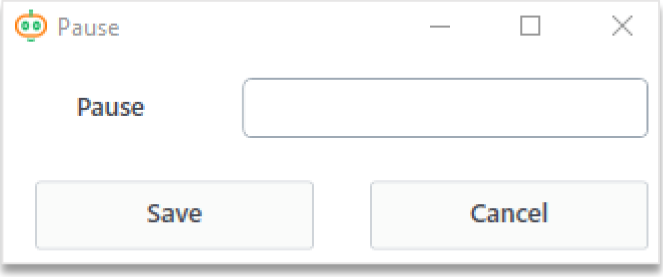


Figure. 8.28.1. Module window

PART IX. MODULES FOR WORKING WITH TEXT

Chapter 29. «Text processing»

The «Text processing» module offers several methods of processing the text/ Some tasks require knowledge of how much the text is different between the standard and the specified text. These are tasks related to computer linguistics and artificial intelligence.

Module interface

The module window consists of the part of compiling a command, command control buttons and a list of commands in the form of a table. The part of creating a command consists of 4 main fields: «Action» - drop-down list with methods for text processing, 2 fields «Line 1/2», are intended to enter two lines of text, or variables with text, and the last field is «Variable» - for the name of the variable in which the result of the module will be put.

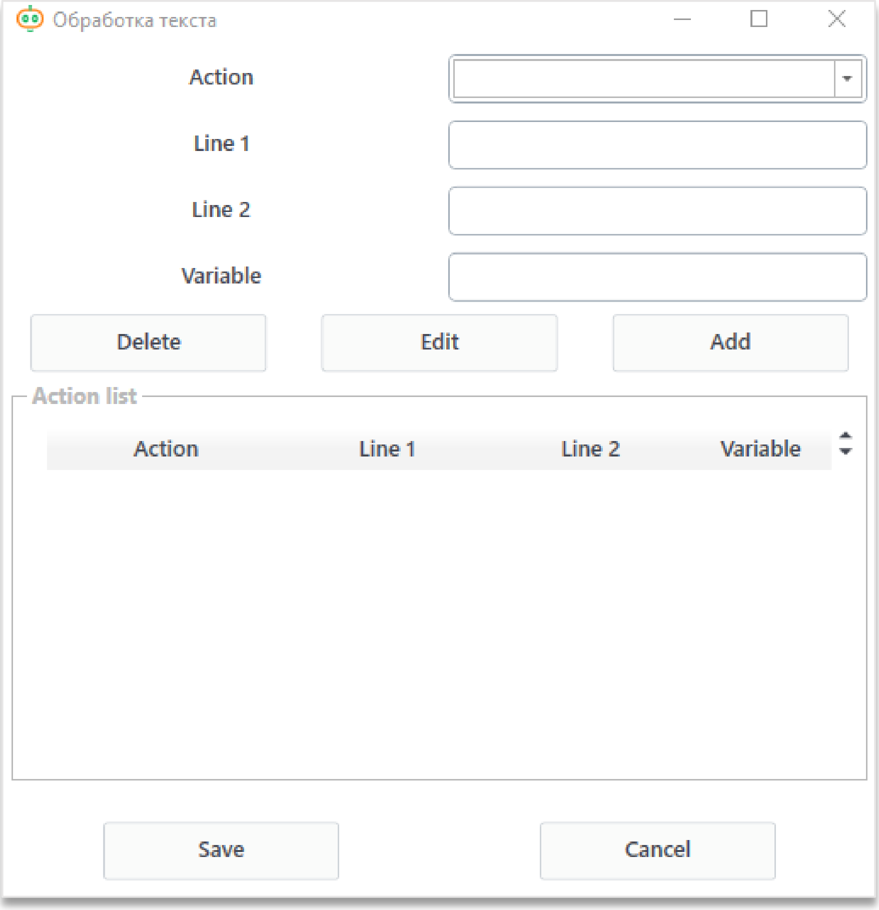


Figure. 9.29.1. Module window

By clicking the «Add» button the created command will be put into the «Command list» section. To edit a command you need to select it from the list, make all desired changes and then click the «Edit» button. To delete a command, select it from the list and click the «Delete» button. You can change the position of commands by clicking the arrow keys, as in Excel module (part 2).

Text processing methods

The following text processing methods are offered in this module:

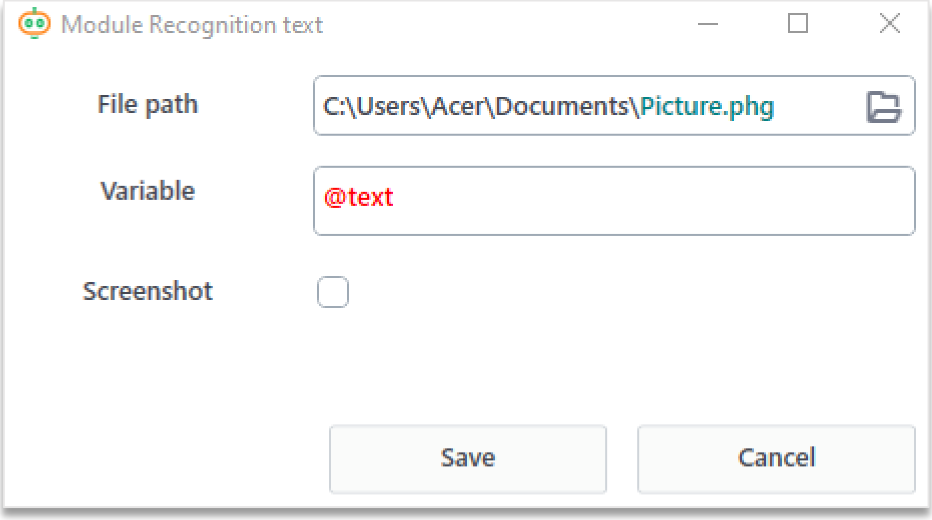
1. **Levenshtein distance** – calculates the difference between two lines. For example – «Lexema RPA» and «Lexema SR» differ by 3 characters – the words «Lexema» stay the same, and the rest of the characters are different, so the result that will be put into the variable is 3;
2. **3-граммы** – - a method based on working with n-grams, in our case n = 3 - the similarity of every 3 characters is estimated. The larger the number (up to 1), the greater the similarity of the lines. In the «Lexema RPA» and «Lexema SR» example the result is gonna be the 0,52 number.
3. **Jaro winkler similarity** is a measure of line similarity for measuring the distance between two sequences of characters. The smaller the Jaro-Winkler distance for two lines, the more similar these lines are. For the «Lexema RPA» and «Lexema SR» example the result is 0,5.

Chapter 30. «Text recognition» module

The «Recognition text» module is intended for reading text from a specified image and for representing the resulting text in XML format.

Module interface

The module consists of two fields – «File path» and «Variable», and the «Screenshot» checkbox.



File path: C:\Users\Acer\Documents\Picture.phg

Variable: @text

Screenshot □

Figure. 9.30.1.Module window

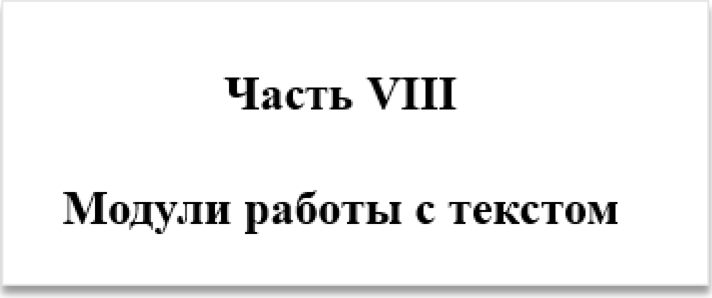
The «File path» field. The field is filled with the direction to the file you need to recognise. This can be a picture or a pdf document.

The «Variable» field. The field is filled with the name of the variable starting with «@» in which the recognition result will be placed.

The «Screenshot» checkbox. The checkbox is checked when it is necessary to recognize a screenshot of the screen at the time of operation of this module during the execution of the robot. In this case you don't need to specify the «File path».

EXAMPLE.

Let's say we have the following .jpeg image. We submit it to the text recognition module.



Часть VIII

Модули работы с текстом

Figure. 9.30.2. The submitted image

The result that is put into the variable:

```
<OCRPages><Page><number>1</number><word><value>Часть</value><x>160</x><y>63</y></word><word><value>VIII</value><x>221</x><y>63</y></word><word><value>Модули</value><x>89</x><y>123</y></word><word><value>работы</value><x>177</x><y>123</y></word><word><value>с</value><x>229</x><y>123</y></word><word><value>текстом</value><x>283</x><y>123</y></word></Page></OCRPages>
```

The result contains information about the pages, the coordinate and the meaning of each word in the image.

Chapter 31. «ABYY Fine reader OCP» module

The «ABYY fine reader OCP» works with PDF files. Unlike the previous module, it has higher accuracy and speed.

The recognition result is recorded in an object variable, in which you can refer to a specific row, table, word.

Module interface

This module consists of two fields – «Pth to pdf file» and «Variable».

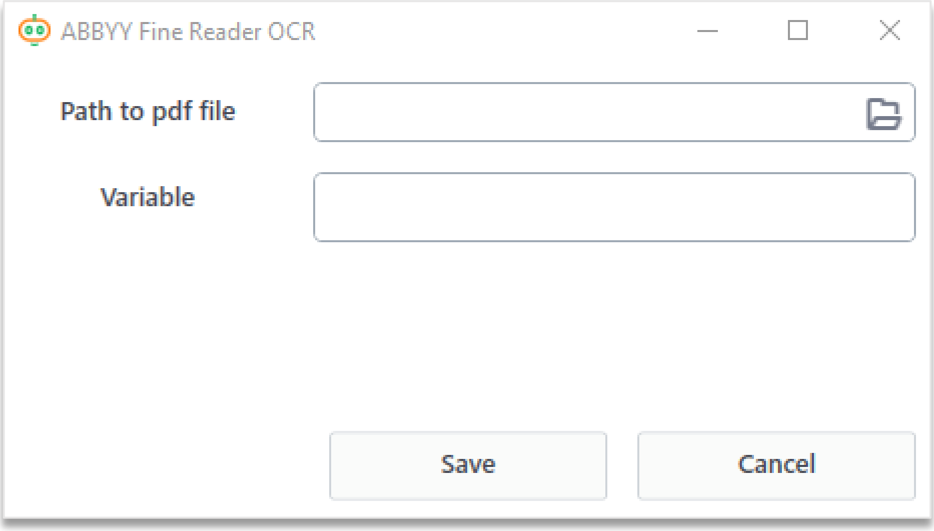


Figure. 9.31.1. Module window

The «Path to pdf file» field. The path to the picture/file you wanna recognise is specified here.

The «Variable» field. The name of a variable is put into this field.

A PDF file may consist of several pages, so the variable will be a list of pages. For example, @text [0] is the first recognized page of a file.

All recognized text is divided into two parts - the main text and the tabular part.

Main text

To access the text, you must refer to the **rows_word [0]** field (in this record, [0] means that the first text part is being accessed. In other versions of ABBYY, a larger text splitting is possible). Then it is indicated, as a separate field, the number of the line, and separated by a dot the word number in the line - **@text[0].rows_word[0].2.4**. Then, after selecting the right word, you can get its value, by referring to the value field - **@text[0].rows_word[0].2.4.value** – this line will return the value of the word, located on the first page of the recognised document, in the second line and is fourth on the left.

The tabular part

To access the field you need to refer to the **tables[0]** field ([0] is needed for the proper work of the robot, its semantic part is embedded in the ABBYY product). Then, the number of the table is indicated as a separate field, because there can be several tables on one sheet - **@text[0].tables[0].0**. After that, the number of the cell in the table is entered as a single number. If you want to find a cell and know its column and row number, you can get the index of the cell in the variable in the form of "row, column" by referring to the the index field - **@text [0].tables[0].0.5.index** (5 - fifth cell in an array of recognized cells). To get a values from a cell, you need to refer to its value, and then either collect the entire string written to the cell, or refer to a specific word by its serial number - **@text[0].tables[0].0.5.value.1.value** - the value of the second word from the fifth cell of the first table from the first page of the recognized text.

To get the amount of cells in the table you can use this construction - **@text[0].tables[0].0[%]**, specify the set of characters «[%]» after the field after which there is a field, the number of elements of which you need to find .

To get the amount of words in a cell you can use this construction - **@text[0].tables[0].0.5[%]**.