

# ROBOTASK

## **User's Guide**

2012

# Table of Contents

<b>Part I Quick Start Guide</b>	<b>2</b>
1 Features .....	2
2 Getting Started in the RoboTask Window .....	3
3 Create Tasks .....	5
4 Run Tasks .....	5
5 Edit Tasks .....	6
6 Customize RoboTask .....	6
7 Purchasing .....	6
8 License Agreement .....	6
9 Trademarks and Copyrights .....	7
<b>Part II Several practical recommendations</b>	<b>9</b>
1 Using Variables in Tasks .....	9
Using Variables .....	9
System variables .....	10
Custom variables .....	10
Local Variables .....	11
Create a Variable .....	12
Add Variables to a Task .....	13
Rules for using variables in expressions .....	13
2 Macro recorder .....	14
3 Running tasks simultaneously .....	15
4 Debugging of the task .....	16
5 Task synchronization .....	17
6 Using built-in Basic .....	17
7 Running RoboTask as an NT service .....	18
Installing and running .....	18
Using UNC names .....	18
Authentication parameters .....	19
Peculiarities of working in the system service mode .....	19
The visual differences between the work in ordinary application mode and in the system service mode.....	20
Peculiarities of work RoboTask Service in RDC mode.....	20
Peculiarities of work RoboTask Service under OC Vista/ Windows 7/ Windows 2008 Server.....	20
8 Errors handling .....	21
9 Control of RoboTask through network .....	22
Network Server .....	22
Network Client .....	23
Connect to.....	24
Define connections.....	24

<b>Part III Menus</b>	<b>26</b>
1 Task > New .....	26
2 Task > Edit .....	32
3 Task > Delete .....	32
4 Task > Run .....	32
5 Task > Stop .....	32
6 Task > Duplicate .....	32
7 Task > Create Shortcut on Desktop .....	33
8 Task > Exit .....	33
9 Options > View Log .....	33
10 Options > View Toolbar .....	33
11 Options > Settings .....	33
12 Options > Variables .....	38
13 Tools > Enable Automation .....	40
14 Tools > Enable All Tasks in List .....	41
15 Tools > Disable All Tasks in List .....	41
16 Help > Help Topics .....	41
17 Help > Send Bug Report .....	41
18 Help > About .....	41
<b>Part IV Actions</b>	<b>43</b>
1 General .....	43
Run Program/Open Document .....	43
Print Document .....	44
Send Keystrokes .....	45
Shift Keys State .....	49
Log Off .....	50
Pause .....	50
SQL Query .....	51
Wait for Process Finish .....	52
Wait for Process Start .....	53
Terminate Process .....	54
Comment .....	54
2 Dialogs .....	55
Show Message/Play Sound .....	55
Question Dialog .....	57
Open Dialog .....	59
Save Dialog .....	60
Browse for Folders .....	61
Browse for Computers .....	62
Input Box .....	63
3 Files and Folders .....	64
Copy/Move File .....	64
Rename File .....	65
Delete File .....	66

Write Text File .....	68
Search for Files .....	69
Change Folder .....	73
Create Folder .....	73
Rename Folder .....	74
Remove Folder .....	74
Synchronize Folders .....	75
Wait for file .....	76
SortText .....	77
Unique File Name .....	79
File Attributes .....	80
<b>4 Zip .....</b>	<b>81</b>
Pack Files .....	81
Extract Files .....	83
<b>5 Internet .....</b>	<b>84</b>
Check for Email .....	84
Send Email .....	85
Download Files .....	88
Ping Host .....	89
<b>6 FTP .....</b>	<b>90</b>
FTP Log On .....	90
FTP Log Off .....	91
FTP Download .....	91
FTP Upload .....	91
FTP Rename File .....	92
FTP Delete File .....	93
FTP Change Folder .....	93
FTP Create Folder .....	94
FTP Remove Folder .....	95
FTP PWD .....	95
FTP Transfer Type .....	96
FTP Command .....	96
FTP File Loop .....	97
FTP Synchronization .....	98
<b>7 SSL FTP .....</b>	<b>100</b>
SSL FTP Log On .....	100
SSL FTP Log Off .....	104
SSL FTP Download .....	104
SSL FTP Upload .....	105
SSL FTP Rename File .....	105
SSL FTP Delete File .....	106
SSL FTP Change folder .....	107
SSL FTP Create Folder .....	107
SSL FTP Remove Folder .....	108
SSL FTP PWD .....	108
SSL FTP Transfer Type .....	109
SSL FTP File Loop .....	109
SSL FTP Synchronization .....	111
<b>8 Dial-Up .....</b>	<b>113</b>
Dial Up/Hang Up .....	113
Check connection .....	114
<b>9 Variables .....</b>	<b>114</b>

Create Variable .....	114
Set Variable .....	115
Remove Variable .....	116
Increment Variable .....	117
Delimited variable .....	117
<b>10 Task Commands .....</b>	<b>118</b>
Enable Task .....	118
Disable Task .....	119
Start Task .....	119
Stop Task .....	121
Wait for Task .....	121
<b>11 Loops and Flows .....</b>	<b>122</b>
File Loop .....	122
Text Loop .....	125
Dataset Loop .....	126
Window loop .....	127
Simple Loop .....	129
While loop .....	129
End Loop .....	131
Break .....	131
Continue .....	131
If Then .....	131
Else .....	133
End If .....	133
<b>12 Window .....</b>	<b>133</b>
Window Command .....	133
Check for Window .....	134
Wait for Window .....	135
<b>13 For Windows NT .....</b>	<b>138</b>
Start Service .....	138
Stop Service .....	139
Lock Workstation .....	140
Run Program as User .....	140
<b>14 RoboTask Commands .....</b>	<b>141</b>
Minimize RoboTask .....	141
Restore RoboTask .....	141
Exit RoboTask .....	142
Enable Automation .....	142
Disable Automation .....	142
Log Message .....	142
<b>15 Clipboard .....</b>	<b>143</b>
Cut .....	143
Copy .....	143
Paste .....	144
Clear Clipboard .....	145
Get text .....	145
Put Text .....	145
<b>16 Mouse .....</b>	<b>146</b>
Move to object .....	146
Click .....	148
Move .....	149

<b>17 Registry</b>	<b>149</b>
Add key	149
Delete Key	150
Rename Key	151
Read Value	152
Write Value	153
Delete value	154
Rename value	155
Export	156
Import	157
<b>18 Network</b>	<b>158</b>
Send Command	158
Send Network Message	159
Network Plugin settings	160
<b>19 Email</b>	<b>161</b>
POP3 Open	161
POP3 Close	162
POP3 Mail Loop	162
POP3 Download Email	164
POP3 Delete Email	165
POP3 Messages Count	165
Parse Email	166
<b>20 Strings</b>	<b>170</b>
STR Replace	170
STR Trim	171
STR Delete	172
STR Insert	173
STR Extract	173
STR Pos	174
STR Between	175
<b>21 Basic</b>	<b>177</b>
Evaluate	177
Basic script	178
RoboTaskApp object	178
<b>22 CD/DVD Writer</b>	<b>183</b>
Write to disk	183
Erase Disk	185
Eject/Load CD Tray	186
<b>23 Web Documents</b>	<b>186</b>
Fill Web Form	186
Add or Edit Web Form Parameter dialog	190
Wait for Done	192
<b>24 Regular Expressions</b>	<b>195</b>
RegExp Loop	195
RegExp Replace	198
RegExp Match	201
RegExp COM object	203

## Part V Triggering Events

**207**

<b>1 Scheduler</b>	<b>207</b>
--------------------	------------

---

2 Hot Keys .....	208
3 Tray Icon .....	208
4 Cyclic .....	209
5 File Monitor .....	210
6 Dial-Up Connection Monitor .....	213
7 Expression monitor .....	214
8 Window Watcher .....	215
9 On Time Changed .....	218
10 On Battery Power is Low .....	218
11 On Spooler Changed .....	218
12 On Device Changed .....	218
13 On Palette Changed .....	219
14 On Display Changed .....	219
15 On Log Off .....	219
16 On Idle .....	219
17 Listener .....	220
<b>Part VI Appendix</b> .....	<b>223</b>
1 Full list of system variables .....	223
<b>Part VII Other software</b> .....	<b>226</b>
<b>Index</b> .....	<b>227</b>

# Part



## Quick Start Guide

# 1 Quick Start Guide

## 1.1 Features

RoboTask can easily automate any series of tasks on your computer system, whether you are a novice or advanced user. You don't even have to know how to program—use RoboTask to build your own automated solutions quickly and easily. Yet RoboTask is powerful.

For example, RoboTask can help you avoid losing important data when you create a backup task for placing files on a remote server.

The software can manage and trigger a wide variety of repetitive tasks to save your valuable time. With RoboTask, you don't need complicated batch files or scripts.

Here are many examples of what RoboTask can do for you:

- Runs applications and open documents.
- Manages automatic backups.
- Alerts you of any changes in your folders.
- Shuts down Windows and powers off your computer triggered by any event you choose.
- Processes files and folders (copies, moves, renames, deletes).
- Works with FTP and Web servers (downloads, uploads, etc).
- Sends letters and files by e-mail.
- Starts and stops Windows NT services.
- Sends network messages as WinPopups.
- Acts as an alarm clock.
- Much more!

RoboTask can be used for an unlimited number of tasks. When creating a task, you decide what the task needs to do and when. Every task has two parts:

1. Actions
2. Triggering events

### Actions

First, you must decide what actions you want to accomplish. You can select built-in actions in RoboTask to accomplish the following:

- Run applications.
- Open or print documents.
- Emulate your keystrokes.
- File and folder processing (copy, move, delete, rename, synchronize folders).
- Work with FTP servers (download, upload, rename, delete).
- Work with e-mail (check for e-mail, send e-mail).
- Work with ZIP files (pack files, extract files).
- Work with windows (focus, maximize, minimize, restore, close, hide, unhide).
- Flow control (if-then, file loop, text loop, dataset loop, and so forth).

### Triggering events

A triggering event can be one of the following:

- A specific time

- When something happens on your computer.

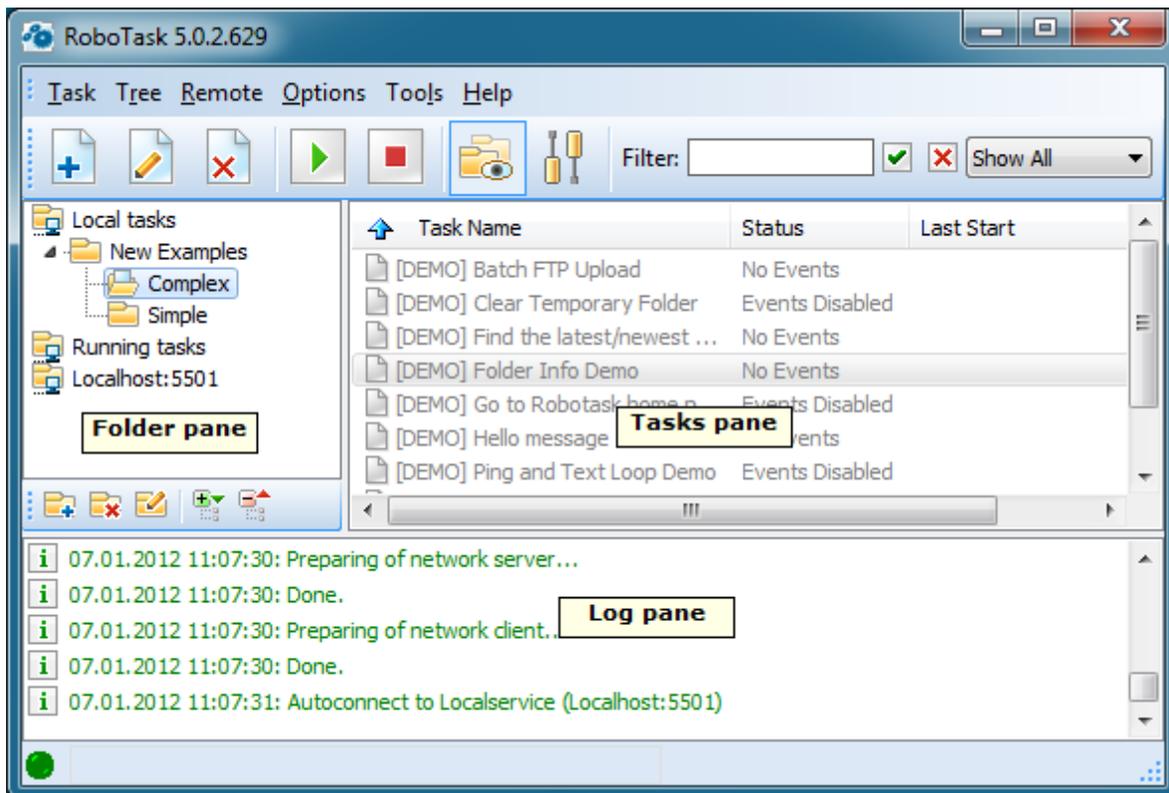
Practically anything that happens on your computer can trigger RoboTask to perform an action. An example would be when RoboTask automatically updates a log and uploads it to a website whenever you receive an email with a subject that is a particular string of words or contains a word such as "Important".

The following are examples of triggering events:

- Scheduler (at a specified time).
- Hot Keys (when a special key is pressed).
- Tray Icon (when you double-click it).
- File Monitor (when a particular file is created, destroyed, changed, renamed).
- Dial-up Connection Monitor (when a connection is made or broken).
- Window Watcher (when a window is created, activated, closed, and so forth).
- System Events (time, spooler, device, display changed, log on, log off, on idle).

## 1.2 Getting Started in the RoboTask Window

The following information introduces you to the RoboTask main window.



### Folder Pane

Displays a tree of the task folders. This is useful for organizing your tasks.

### Task pane

**Task Name Column**

Displays the name of the task.

**Status Column**

Displays whether the task is enabled or disabled. You must run the task manually if the task is disabled by double-clicking it. You can change the status of a task: select the task and right-click.

**Log Pane**

Displays the log messages when you open RoboTask or when a task runs. You can set options for the log when you add a task or edit one on the Advanced tab of the Add Task dialog box.

**New Task**

Allows you to add a new task. You can select actions and triggering events.

**Edit Task**

Allows you to modify a task. You can edit the settings for actions and triggering events.

**Delete Task**

Removes the task that you select in the main window.

**Run Task**

Starts the task that you select in the main window.

**Stop Task**

Stops the task that you select in the main window.

**Show/Hide Task Tree**

Displays or hides the task tree at the left of the main window.

**Settings**

Allows you to set preferences for RoboTask.

**New Folder**

Allows you to create a new folder and enter the name.

**Remove Folder**

Removes the task folder that you select.

**Rename Folder**

Allows you to rename the selected task folder.

**Expand All**

Hides the folders in the tree and displays only the top level folder.

**Expand All**

Displays all the folders in the tree.

**Shortcut Menus**

If you select a task in the **Task Name** column and right-click, the following commands appear on the shortcut menu:

- **Enabled** - Allows the task to start automatically when a triggering event in the task occurs.
- **New** - Creates a new task.
- **Edit** - Modifies the task that you select.
- **Delete** - Removes a task.
- **Run** - Starts a task.
- **Stop** - Ends a task.
- **Duplicate** - Copies a task. This is useful if you want to create a new task quickly and then modify it.
- **Create Shortcut** - Adds a shortcut to the task on your desktop. This is useful if you want to quickly start a task without opening RoboTask.

If you point to the Log pane and right-click, the following commands appear on the shortcut menu:

- **Clear Log** - Removes all messages from the Log pane.
- **Copy Selected to Clipboard** - Copies the lines that you have selected in the Log pane and places them on the Windows Clipboard.
- **Select All** - Selects all the lines in the Log pane so that you can quickly copy them.

## 1.3 Create Tasks

1. Click [Task > New](#)<sup>[26]</sup>.
2. On the **Actions** tab, click the **Add** button.
3. Select an action. Some of the actions require additional settings. A dialog box appears with specific settings that you can select.
4. After entering the settings, click the **Triggering Events** tab.
5. Click the **Add** button.
6. Select a triggering event. Some of the triggering events require additional settings. A dialog box appears with specific settings that you can select.
7. After entering the settings for the triggering event, click OK.
8. To run your task, click the **Run Selected Task** button in the main window.

### Tip

Some actions or triggering events do not have specific settings that you can enter.

## 1.4 Run Tasks

1. In the right pane of the RoboTask window, select a task.
2. Click **Task > Run** or press CTRL + ENTER on the keyboard
3. In the right pane of the RoboTask window, the status column displays the results of running the task.

### Tip

- You can run a task without opening RoboTask. Select a task in the RoboTask window and click **Task > Create Shortcut on Desktop**. This adds a shortcut to that task on your desktop. You can double-click the task to start it.

## 1.5 Edit Tasks

1. In the RoboTask window, select a task.
2. Click **Task > Edit**.
3. On the **Edit Task** dialog box, select the options that you want.

### Tip

- To remove a task from the list in the RoboTask window, select a task and click **Task > Delete**.

## 1.6 Customize RoboTask

1. Click **Options > Settings**.
2. On the **Preferences** dialog box, set the options you want.

## 1.7 Purchasing

Click **Help > Purchase**. Also you may purchase full version of RoboTask from [official web site](#).

## 1.8 License Agreement

### The RoboTask is distributed as try-before-you-buy. This means:

1. All copyrights to **RoboTask** are exclusively owned by the author - NeoWise Software.
2. Anyone may use this software during an evaluation period of 30 days. Following this evaluation period of 30 days or less, if you wish to continue to use **RoboTask**, you must purchase it.
3. Once purchased, the user is granted a non-exclusive license to use **RoboTask** on one computer (i.e. a single CPU), for any legal purpose, at a time:
  - **Personal license** entitles you to use RoboTask only on a single computer at home for personal purposes.
  - **Business license** entitles you to use RoboTask on server or workstation by employees of your company or organization at single location.
  - If you need to use RoboTask on a few computers (servers or workstations) you must purchase corresponding amount of licenses.
4. Registered user is granted free upgrades of **RoboTask** within one year period since purchasing of RoboTask. In order to prolong the period of free upgrades, user must

purchase new registration code with discount.

5. The licensed **RoboTask** software may not be rented or leased, but may be permanently transferred, if the person receiving it agrees to terms of this license. If the software is an update, the transfer must include the update and all previous versions.
6. The trial version of **RoboTask** may be freely distributed, provided the distribution package is not modified. No person or company may charge a fee for the distribution trial version of RoboTask without written permission from the copyright holder.
7. **RoboTask** IS DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.
8. You may not use, copy, emulate, clone, rent, lease, sell, modify, decompile, disassemble, otherwise reverse engineer, or transfer the licensed program, or any subset of the licensed program, except as provided for in this agreement. Any such unauthorized use shall result in immediate and automatic termination of this license and may result in criminal and/or civil prosecution. All rights not expressly granted here are reserved by NeoWise Software.
9. Installing and using **RoboTask** signifies acceptance of these terms and conditions of the license.
10. If you do not agree with the terms of this license you must remove **RoboTask** files from your storage devices and cease to use the product.

Thank you for using the original RoboTask.

NeoWise Software  
[www.robotask.com](http://www.robotask.com)

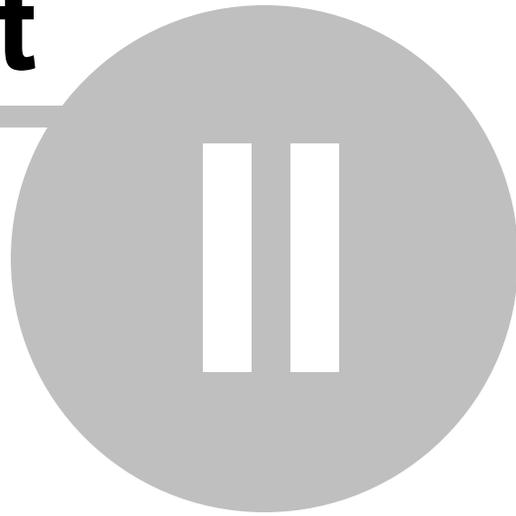
## 1.9 Trademarks and Copyrights

RoboTask is a trademark of NeoWise Software.

Microsoft, the Windows logo, Windows, and Windows NT are registered trademarks of Microsoft Corporation. All other brands and product names are trademarks or registered trademarks of their respective owners.

# Part

---



**Several practical recommendations**

## 2 Several practical recommendations

### 2.1 Using Variables in Tasks

#### 2.1.1 Using Variables

For building complex automation schemes, RoboTask provides two types of variables that you can use in actions and triggering events:

- System variables that are pre-defined.
- User variables that you can define yourself.

When you insert a variable name as a parameter or part of a parameter in an action or triggering event, the variable content is processed, instead of the variable name.

For example, if you want to show a message containing the day of the week, you should use the `{DayOfWeek}` variable.

When you run the task that contains the `{DayOfWeek}` variable, the variable is replaced by the name of current week day. If you insert this variable in a Show Message action that includes the message defined as "Today `{DayOfWeek}`!", when you run this task on Wednesday, the following message displays: "Today Wednesday!".

You can use as many variables as you need in an event or action. Any time that you define the same parameter many times, you should use a variable. This allows you to change the variable in one place and all your tasks that contain the variable update automatically.

#### **Variables that Contain Other Variables**

You can even use variables within other variables; this is useful if you want to specify a value of a variable for other variables. For example, you can define a variable with a name `{Yesterday}`. `{Yesterday}` contains another variable with a value `{MonthIncDays(-1)}/{IncDays(-1)}/{YearIncDays(-1)}`. When you use the `{Yesterday}` variable, it returns yesterday's date.

#### **Arrays**

To make your tasks even more powerful, you can use variables as arrays. This means you can specify more than one values separated with commas. To access the first value, use index 0: `{MyVar(0)}`. To access the second value, use index 1: `{MyVar(1)}`. And so on.

For example, if to add a variable with name `{MyArray}` and value *One, Two, Three*, you can use that variable as shown below:

- `{MyArray}` = One, Two, Three
- `{MyArray(0)}` = One
- `{MyArray(1)}` = Two
- `{MyArray(2)}` = Three

Another example:

- `{MyArray}` = "First item", "Second item", "Third item"

- {MyArray(0)} = First item
- {MyArray(1)} = Second item
- {MyArray(2)} = Third item

### 2.1.2 System variables

You can view [system variables](#)<sup>[223]</sup> in the RoboTask variables dialog box (*the Options | Variables menu*). The list is divided into categories. Each variable in the list has a brief description. Thus, it will not be difficult to study [the set of variables](#)<sup>[223]</sup> and we will not describe them here. There are more than 60 system variables available now. Actually they are functions that are executed at the moment of line generation.

Some variables require a parameter, for example:

**ExtractFileNameNoExt()**

You should specify the full name of the file as the parameter of this variable. For example:

**{ExtractFileNameNoExt(C:\Folder\File.ext)}**

It is possible to use embedded variables. Suppose we have a custom variable named *MyFileName* containing the name of a file. To extract the file name without its extension, type:

**{ExtractFileNameNoExt({MyFileName})}**

In this case, RoboTask will find {*MyFileName*} first and then use the found string as a parameter.

This rule applies to all system variables that require a parameter.

### 2.1.3 Custom variables

RoboTask allows you to save intermediate data in users variables. There is also a whole set of actions that cannot work without user variables. For instance, loops. While using variables, you should remember about some peculiarities.

#### **User variables are global for all tasks**

It means that one and the same variable is visible, can be used and can be modified in any task. On the one hand, it is very convenient: data can be passed from one task to another. But on the other hand, if you have two tasks running simultaneously, they use one and the same variable and it is done unintentionally, you may get a completely unexpected result.

#### **Tip**

*If you use variables that are not supposed to be used in other tasks, use special prefixes of this task in the names of these variables.*

*For example:*

*TSK1\_Counter, TSK1\_CurrentFile, etc.*

*Also you may simply declare necessary variables as local. See [Local variables](#)<sup>[11]</sup> topic*

### RoboTask saves user variables for future sessions

Variables you create in tasks are stored until you remove them using the «RoboTask variables» form (Options | Variables) or the «Remove Variable» action inside the task. When you exit RoboTask, variables save their values for future sessions.

As a rule, variables make sense only for the task they are created in and after using RoboTask for a while, you may get a rather long list of user variables that you do not need anymore.

#### **Tip**

*Insert actions removing unnecessary variables at the end of tasks.*

## 2.1.4 Local Variables

### What local variables are

Local variables of a task are custom variables that "live" only while the task is running.

Differences between local and global variables:

- Local variables "live" only while their task is running. Unlike local variables, global variables live starting from RoboTask startup (or since such a variable is created) and until RoboTask is exited or until such a variable is deleted.
- The values of local variables are not preserved till the next session of working with RoboTask.
- Local variables can be edited and accessed only within the running task. Global variables can be accessed and edited in any task.

They are similar to global variables in all other aspects.

### Advantages of using local variables:

We do not have to take care of deleting unnecessary variables when the task is complete because local variables are deleted from memory automatically when their task is complete.

When you use local variables, you do not have to worry that they will be changed by another task. And vice versa: the use of local variables guarantees that the values of variables will be preserved while the task is running.

If several tasks that can change one and the same variable are running simultaneously, the result may be unpredictable. Suppose we have two similar tasks that go through files in a loop, but they do it in different folders. Both tasks assign the name of the current file to the **CurrentFile** variable. If these tasks are running simultaneously, we may end up with quite a few problems. It is easy to avoid it if you declare the **CurrentFile** variable as local in both tasks.

Such a situation is not a rare one, especially if you have several hundreds of tasks. You can also create new tasks by copying an existing one and changing a few steps in it.

### How to declare local variables

You can declare local variables on the "[Local variables](#)" tab<sup>[30]</sup> of the task editor. There are two ways to declare them:

Enumerate the names of the variables;

Assign initial values to them.

If you enumerate them, just fill the list of names for local variables: one variables per line. For example, like this:

```
LineCount
FileToUpload
CurrentFile
```

etc.

To assign an initial value, specify this value after the "=" character. For example, like this

```
LineCount = 0
FileToUpload = c:\incoming\list.txt
CurrentFile = note.txt
```

etc.

You can assign initial values not to all variables. By default, local variables contain an empty value (an empty string).

By default, the list of local variables is empty. It means that all variables that are used in a task will be processed as global ones when you run the task.

### How to use local variables

There are no additional requirements regarding the use of local variables. The same tools are used to work with local variables as with global ones:

Actions from the Variables group

[Create variable](#)<sup>[114]</sup>

[Set Variable](#)<sup>[115]</sup>

[Remove variable](#)<sup>[116]</sup>

[Increment variable](#)<sup>[117]</sup>

[Delimit Variable](#)<sup>[117]</sup>

Inserting variables into strings in various actions.

For example:

```
Current folder is "{CurrentFolder}"
```

Using the methods of the [RoboTaskApp](#)<sup>[178]</sup> object: **ExpandText**, **SetUserVariable**, etc.

The rules according to which RoboTask processes local variables are very simple:

Once a variable is called, RoboTask first tries to find it among the local variables of the task.

Calling a variable means any call: assigning a new value or calculating the variables value when macro insertions with the help of variables are substituted.

If the local variable is declared, further operations (reading or assigning the value) will be applied to the local variable. Otherwise, RoboTask will process the variable as a global one.

Thus, if we declare a local variable that has the same name as a global variable, the local variable will completely override the global variable. The running task will not simply see the global variable.

It is better to declare all internal variables as local ones for most tasks. As a rule, these variables are used in a loop or to pass data from one step in the task to another. When you use local variables, you need not worry that the new task will mess up data from other tasks. And vice versa – that some other task running simultaneously changes the data of the current task. On the other hand, if you need to pass some data to another task, you should make sure that the necessary variables are absent in the list of local variables.

## 2.1.5 Create a Variable

1. To define a variable, click **Options > Variables**.

2. On the **User Variables** tab, click the **Add** button.
3. On the **User Variable** dialog box, enter the name of the variable in the **Variable Name** box.
4. Enter the values in the **Initial Values** box.

**Tip**

In the **Initial Value** edit box, you can also type the names of other variables to define a variable.

### 2.1.6 Add Variables to a Task

You can add variables to an action or triggering event contained in a task.

1. When you create a task, on the **Edit Task** dialog box, select an action or triggering event.
2. When you do this, some actions or triggering events display a dialog box in which you can enter specific settings for the action or event. On this dialog box, select a field and click the **{V}** button.
3. On the **RoboTask Variables** dialog box, select a variable in the list and click the **Select** button.
4. The variable appears in the field that you selected on the dialog box for the action or triggering event.

### 2.1.7 Rules for using variables in expressions

Using variables in actions permits to achieve more flexibility of an algorithm. Sometimes you just can't do without using variables. For example, in cycles.

If you use expressions with variables in parameters of actions, RoboTask at first expands all variables, and only after that uses the result for the action. In other words, RoboTask computes the variable value and substitutes the value for the variable in the expression.

For example, the expression

```
Temporary file is {UniqueFileName({TempDir}\*.tmp)}
```

will approximately have the following form after its computation:

```
Temporary file is C:\Users\User1\AppData\Local\Temp\DA0PUY08.tmp
```

The rules for computing values are simple:

- **The rule of visibility.** At first, RoboTask tries to find the variable in the list of local variables. Then, if it fails, it looks up among global ones. If the variable isn't found, substitution doesn't occur. System variables are always global. Inherently they are functions which return the result of some computations. This rule is described in the unit "Local Variables".
- **The depth of expanding variable.** If variables are used in an expression by means of braces **{}**, the variables are expanded to the whole depth. That is, firstly the value of variable is computed, and then the result is expanded if it contains variables. And so on, to the end. However, sometimes it is necessary to expand only the specified variable

without further recursive processing. To do this you have to specify variables with the character **#** after an opening brace: **{#MyVariable}**. More details in this unit below.

By default, RoboTask tries to expand all variables in the specified expression. Then it recursively reiterates the attempt to expand variables in resultant expression until all variables are expanded. However, sometimes it is necessary to protect a text from expanding variables. These can be some technical texts for automated processing. These texts aren't directly related to RoboTask. They can contain inserts coinciding with the variables of RoboTask. The use of braces is quite a widespread method of inserting some macros when texts are processed automatically.

For example, there is such a text file:

```
To automate mailing you can use macros to create a message:
insert of a customer address - {Email}
insert of a customer name - {User}
insert of a sender name - {SenderEmail}
insert of a current date into the message - {Date}
insert of current time into the message - {Time}
insert of a signature - {Signature}
```

You have to process this text in a cycle together with other files by means of RoboTask. For example, add HTML tags and save it as HTML file preserving all the file contents without changing. It is easy to see that if we try to write the file contents into a variable by means of a system variable **{#TextFile(file\_name)}**, expressions **{Date}** and **{Time}** will be replaced with current meanings of date and time.

To save a text to a variable without changing, you have to use the symbol **#** after the opening brace: **{#TextFile(file\_name)}**. In this case RoboTask doesn't try to expand all variables in the text recursively, but only returns the contents of the text file without changing. Naturally, you also have to use the variable containing the text with the symbol **#**, in order to prevent from expanding expressions in text. For example: **{#CurrentText}**.

## 2.2 Macro recorder

RoboTask has powerful tools of interactive automation, such as «Send Keystroke» operations and the mouse actions (Move, Move to object, Click). However, it is obviously not convenient to create an incremental task that implies mouse shortcuts. Sometimes several operations need to be performed as a single block.

For example:

- Activate the contextual menu;
- Select the menu item;
- Click on the needed element of the pop-up dialog.

In a step-by-step configuration, it is rather difficult to hit the required menu item. To simplify the creation of the interactive mouse shortcuts sequence you can use the automatic macro recording feature by clicking on «Record a macro» button in the task editor. Macro recorder allows automatic recording of the sequence of mouse pointer movements: namely, mouse button ups and downs, pauses between the shortcuts. When the recording is

over Macro Recorder transforms the recorded macro into actions and specifies them as the task steps.

To record the mouse shortcuts do the following:

- Open the [task editor](#)<sup>[32]</sup>
- Click «Record a macro». At that, all the RoboTask windows will be hidden and the window with start and stop recording buttons will be displayed.
- Click on start recording button .
- Then make the required mouse shortcuts (mouse pointer movements and pressing the buttons on the required window elements) and click on stop recording button .
- When the recording is over Macro Recorder will recover the task editor window and insert the recorded shortcuts into the editor.

As a rule, the recorded macro is ready to use. However, it is strongly recommended to edit the created steps manually. Macro Recorder records the actions properties exactly as they were made, and that is not always useful for the task creation.

For example, in such action as «[Move to Object](#)<sup>[146]</sup>» it is almost always needed to replace the full title bar with the specific title bar fragment. It should be done because the Windows application windows almost always change their title according to the window contents or the context. If the window properties are left unchanged, during the operation flow RoboTask may not find the window according to the properties. In this case, the task will be stopped with an error.

It is also useful to edit the pauses between the operations. As a rule, they can be significantly reduced to speed up the task execution. However, it is not recommended to remove the pauses at all, because you should wait a certain period of time until the application can respond to the mouse shortcut (e. g. load some data or open a new window).

## 2.3 Running tasks simultaneously

Each task runs in the background mode independently of other tasks. You can effectively use this feature while scheduling your tasks. Suppose you need to pack several independent folders into different archive files. You can create the following task:

```
Pack "Folder1" into "folder1.zip"  
Pack "Folder2" into "folder2.zip"  
Pack "Folder3" into "folder3.zip"
```

In this case, the task will pack Folder1, Folder2 and Folder3 one by one. That is, each next folder will be processed only after the previous one is packed.

The use of background processes would be more effective. In this case you need to create three different tasks and each of them will pack the corresponding folder. In this case, the main task will look like this:

```
Start task "Pack folder1" without waiting  
Start task "Pack folder2" without waiting  
Start task "Pack folder3" without waiting
```

Running such a task will be the same as running three tasks "Pack folder1", "Pack folder2" and "Pack folder3". Each of these tasks will run independently of the others and the total time of packing all three files will be a bit shorter than in case with successive processing.

## 2.4 Debugging of the task

In this chapter we will consider some recommendations (guidelines) concerning the debugging of the generated task. If the task consists of 3-5 sequential steps, there is no need for debugging as a rule. But if there are more than 30 steps with loops and branching in the task ([if .. then .. else](#)<sup>[13]</sup>), then there are surely hidden mistakes in it, that aren't visible ex facte.

### **By all means check the generated task by test example.**

For lack of experience, it is difficult to reproduce the entire algorithm in the mind, and to write it as a task without any mistakes. To err is human.

Testing is necessary to check workability and correctness of the task. It is possible to check the implementation of the task, for example, by files of the test folder or by files which are easily restored from a backup copy.

The task can be invoked only after checking its correct working. Otherwise you may break working files. Or the task will not work in time, and you will not get a necessary result.

### **Output of values of variables into the log file or into the screen.**

The task often works incorrectly because there is the wrong value in the variable: the empty value or the incorrect one.

In order to make sure the variables are filled in correctly it is possible to verify the values of variables before the implementation of the key actions. It can be done in the process of performing the task by several means:

- Output of values of required variables into the screen before the implementation of the necessary action. You can use the action "Show message" for it.
- Output of variable values or expressions into log of the task. Use the action "Log message".

As a text you can output one or several variables. For example:

```
{MyVar1}  
{MyVar2}
```

Also you can output the whole expression if you use the composite value consisting of some variables. For example:

```
"{CurrentUser}" <{CurrentEmail}>
```

### **Using of local variables.**

It is strongly recommended to use [local variables](#)<sup>[11]</sup> in the task. Their advantage is that they "live" only while the task is being implemented, and they are visible only in the context of the implemented task.

Imagine the situation:

You have created a task by copying an existing task. After copying you change the task a bit. All variables in it are global by default. By some reason you have forgotten to change the names of variables. What will happen if both these tasks are implementing at a time? The result must be unpredictable. Both tasks can change the same variables. The algorithm of every single task will be broken.

We can avoid it easily if we use local variables. Choose the tab "Local variables" in the editor of the task and enumerate the variables used in the task, one by one into a line (see the chapter "[Local variables](#)"<sup>[11]</sup>).

The global variables are necessary only in two cases:

- To transfer some data into another task.
- To save a variable value between sessions of RoboTask.

In other cases you'd better use local variables.

## 2.5 Task synchronization

Let us try to make the task described in the previous section a bit more complicated. Suppose you need to pack three folders in three different files and then write them to a CD. As you remember, we have started three different tasks to pack the folders. It means that we must wait until all three tasks are finished before we can write the archives to a CD.

The "Wait for Task" action is used for that. Let us modify our task in the following way:

```
Start task "Pack folder1" without waiting
Start task "Pack folder2" without waiting
Start task "Pack folder3" without waiting
Wait for Task "Pack folder1" for 10 min.
Wait for Task "Pack folder2" for 10 min.
Wait for Task "Pack folder3" for 10 min.
Write files folder1.zip, folder2.zip, folder3.zip to CD
```

Thus, we start three simultaneous processes in order to pack the necessary data and after all three processes are sure to be finished, we write the archive files to a CD.

## 2.6 Using built-in Basic

Sometimes we need to make some nonstandard calculation, analyze a string or modify it according to a certain algorithm. The Basic programming language will help you in that. The Basic plugin for RoboTask is not included in the main distribution package. You can download it from the [RoboTask site](#) and install additionally.

Basic is a simple and at the same time quite powerful programming tool oriented towards both beginners and professionals. You can use a BASIC script to access user and system variables in RoboTask, as well as other tasks. The BASIC plugin comes together with the full help on the BASIC language. The installation file also includes some sample uses. The description of the [Evaluate](#)<sup>[177]</sup> and [Basic script](#)<sup>[178]</sup> actions, as well as the mechanism of interaction with RoboTask is described below.

Almost anyone can learn how to write small procedures in the BASIC language. The use of BASIC in tasks extends available actions beyond the set of RoboTask actions. Experienced users can use Windows APIs and third-party COM objects in their procedures written in BASIC without any limitations.

## 2.7 Running RoboTask as an NT service

### 2.7.1 Installing and running

RoboTask can run in two modes:

- As a regular application
- As an NT service

An NT service is started together with the operating system and runs even if you are not logged on. If you correctly create your tasks and configure triggering events, it will be enough just to switch on the computer for RoboTask to start doing what it is supposed to do. This mode is important for those computers that are used as servers in your office.

The NT service mode is not available on the Windows 9x/Me platform. The NT service mode is not installed by default. In order to install the RoboTask service, you should click the following shortcut in the system menu: **Start | All Programs | RoboTask | NT Service | Install Service**.

The RoboTask group contains the following shortcuts for controlling the NT service mode:

- **Install Service** - installs the RoboTask service on the system;
- **Uninstall Service** - removes the RoboTask service from the system;
- **Reinstall Service** - reinstalls the RoboTask service with the default parameters;
- **Start Service** - starts the RoboTask service;
- **Stop Service** - stops the RoboTask service.

### 2.7.2 Using UNC names

If you access various network resources in your tasks, we recommend that you use UNC (UNC - Universal Naming Convention) names.

For example, instead of the path

```
z:\documents\report.doc
```

you can use

```
\\server1\incoming\documents\report.doc
```

A UNC name is formed in the following way:

```
\\<Server_Name>\<Resource>\<subfolders>\<file_name>
```

I.e. you should use the name of the server and the name of the resource instead of the disk name ("z:").

What benefits you get:

- A UNC name is always valid within your local area network and does not depend on how network drives are mapped on a particular computer.
- You can use UNC names to access network resources even if you have no connected (mapped) network devices. It is simply necessary in the NT service mode when there are no mapped drives for RoboTask in principle.

### 2.7.3 Authentication parameters

When RoboTask is running in the NT service mode, it does not see network resources by default. It happens because in this case RoboTask has not the rights of your session account, but the rights of the virtual "SYSTEM" account despite the fact that you can see the green icon in the tray and can manually control the program. To allow the program to log on to the network, you should specify authentication parameters. These settings make sense and are available only in the NT service mode. Open the Settings dialog box (Options | Settings) and switch to the "NT Service" tab.

**Log on Process as specified user.**

Select this checkbox if you want to allow RoboTask to log on to the local area network automatically when the RoboTask service is started. After that you should specify the authentication parameters.

**Domain Name.**

Specify the domain name where the user is registered. If the user is registered on the local computer, enter a dot (.) instead of the domain name.

**User Name.**

Specify the username.

**Password.**

Specify the password for the specified user.

**Try to Logon**

To test the authentication parameters and log RoboTask manually, click this button. RoboTask will try to log on using the specified parameters and you will see either "Log on successful" or "Log on failed".

### 2.7.4 Peculiarities of working in the system service mode

In this chapter we will consider the base differences between the two modes of RoboTask:

- The general application
- System Service

RoboTask is designed so that the application and the service could work at the same time, not preventing each other. When installing RoboTask Service interaction with the desktop is allowed by default. Sometimes this mode of the service tangles understanding.

**🔗 Related Topics**

[The visual differences between the work in ordinary application mode and in the system service mode](#)<sup>[20]</sup>

[Peculiarities of work RoboTask Service in RDC mode](#)<sup>[20]</sup>

[Peculiarities of work RoboTask Service under OC Vista/ Windows 7/ Windows 2008 Server](#)<sup>[20]</sup>

#### 2.7.4.1 The visual differences between the work in ordinary application mode and in the system service mode

- The RoboTask Service has a green icon in the system tray. The ordinary application has a blue one.
- The main window title of RoboTask terminates in the line "(NT Service)". For example: *RoboTask 3.5 (NT Service)*. The tool tip of the green icon also contains the substring "(NT Service)".
- RoboTask in application mode has another set of tasks than RoboTask Service. The tasks for the application mode are in the local folder of the user's profile (by default). The concrete name of the folder depends on the operating system and on the user name. The tasks for the service mode are in the subfolder <RoboTask\_Folder>\Tasks (by default "C:\Program Files\RoboTask\Tasks").

You can see and change the path to the task folder in the settings of RoboTask (the menu Options | Settings)

#### 2.7.4.2 Peculiarities of work RoboTask Service in RDC mode

In RDC (Remote Desktop Connection) mode you can't see the green icon. The system prohibits the interaction of service with desktop in RDC regardless of the settings of service. You can see the green icon and the main window of RoboTask only in the local session (i. e. from the local terminal).

But Windows 2003 Server gives an opportunity to intercept the local session for RDC. It is necessary for it to start up the client RDC with the parameter `/admin`.

The command line is:

```
%SystemRoot%\system32\mstsc.exe /admin
```

To know about the possible parameters you should implement the command:

```
%SystemRoot%\system32\mstsc.exe /?
```

#### 2.7.4.3 Peculiarities of work RoboTask Service under OC Vista/ Windows 7/ Windows 2008 Server

Operation systems Vista/ Windows 7/Windows 2008 Server prohibit interaction of services with the desktop within the user's session. After launching of RoboTask you can see the dialog box from system service "Interactive Service Detection" on the screen that the program can't display a message on your desktop. If you click the button "Show me the message", Windows opens the system session where you can see the main window of RoboTask service. Then, as usual, you can tune the work of RoboTask, edit the task list, etc.

You should run service "Interactive Service Detection" manually under the operation system Windows 2008. In addition the manual rerun of the service "Interactive Service Detection" activates the display of the service message on the screen immediately, not in a few minutes. In RDC mode under the OC Window 2008 you can get the access to the RoboTask the same way as from the local terminal. You should run service "Interactive Service Detection". But only users with administrative rights can get the access to the service.

If you don't want to get messages from service "Interactive Service Detection" any more, you can just stop this service.

You can also prohibit RoboTask service to interact with the desktop. Do the following:

- Open the properties of service "RoboTask service"
- Open the tab "Log on"
- Turn off the property "Allow service to interact with desktop"
- Rerun the service.

If you need to change something in the tuning of RoboTask, allow the interaction with the desktop for the RoboTask service.

## 2.8 Errors handling

If there is an error occurs in any step during a task, RoboTask records an error message to the task log and stops the task by default. You can change this behavior. To do it, you should select the "Advanced" tab in the task editor and set the "If Error Occurred" parameter to "Continue Execution". In this case, the task will not be stopped in case of an error.

The system variable **{IsError}** is used to analyze if there was an error in the previous step. This variable contains "True" if there was an error in the previous step and "False" if there was no error in the previous step.

Another method of the error handling is the starting up of another task when an error is raised. For this purpose you need to tune the task like this:

- Open task editor and go to the tab "[Advanced](#)"
- Choose the point "Run the Error Handling Task" of parameter "If Error Occurred"
- Choose the task in a drop-down list "Error Handling Task".

If an error was raised in any step when you perform the task, RoboTask starts up the handler-task and stops the main task. At the start of the error handler RoboTask passes the next parameters:

- **LastErrorTaskID** – the internal ID of the task, in which an error was raised. It is a formal identifier of the task like a number.
- **LastErrorTaskName** – the name of the task, in which an error was raised.
- **LastErrorTaskExternalName** – an External Name of the task, in which an error was raised.
- **LastErrorStep** – the number of the step (starting from 1), in which an error was raised.
- **LastErrorCode** – code of the error. There is the own kit of codes of errors for any type of actions (see description of actions). By default – 0.
- **LastErrorDescription** – the description of the error. It depends on a type of action. By default, it is the line "General error".

This set of parameters is enough to definitely identify the error and to undertake any actions. For example:

- Send an email message with the notice and description of error;
  - Make an record into text log file;
- and etc.

## 2.9 Control of RoboTask through network

RoboTask has the feature of the controlling over tasks through network. This opportunity is available to owners of the Business License. The operation of major components of Client-Server is also available in trial mode for a testing.

The control of remote RoboTask is necessary in the next cases:

- The control of a local RoboTask service. Beginning with Windows Vista, the system forbids interaction between a service and a desktop within a user session. Therefore the control of tasks is related with certain inconveniences (see the chapter Peculiarities of work RoboTask Service under OS Vista/ Windows 7/ Windows 2008 Server). The mode Client-Server permits to edit on-the-fly, to add, to delete, and to start service tasks, and also to view the tasks logs without calling at a system console.
- The control of RoboTask installed at the other network computers. RoboTask in client mode can connect to several instances of RoboTask simultaneously. It permits to view and to edit tasks at the other computers in the main window of RoboTask without using RDC (Remote Desktop Connection) or similar tools.

**Notice:** At startup of a task at a remote RoboTask the task is performed in the session in which the RoboTask server is working. It can be tracked only on a client the course of performing by the task log. If the task demands some interaction with the program (for example, dialog windows) it can't be done from a client.

### 2.9.1 Network Server

The server doesn't contain any visual interactive elements and serves only for authorization and handling of client requests. The server must be active if you want to control this copy of RoboTask remotely

All tunings of the server are in the dialog window of [RoboTask Settings](#)<sup>[33]</sup> (menu Options | Settings) on the tab "[Network Server](#)"<sup>[36]</sup>.

#### Start Server at RoboTask loading

It is on by default. Switch off this checkbox if you want to refuse from remote administration.

#### Port

This is the number of a port, on which the server accepts connections. By default, it is 5500. If you use more than one instance of RoboTask at your computer, apply different ports to avoid conflicts.

For example: 5500 is for an application, 5501 is for a Service.

#### Encrypt data

This is the mode of traffic encryption. By default, it is on. The mode of ciphering is specified by server. A client is tuned on the mode of encryption automatically. RoboTask provides the traffic encryption with dynamic keys using CryptoAPI system functions.

**Compress data**

This is the mode of traffic compression. By default, it is on. The mode of traffic compression can be useful when connection is slow (for example, DialUp or GPRS). The mode of traffic compression (packing) is specified by server. A client is tuned on the mode of compression automatically.

**User Rights**

This is the check mode of user's rights at connection. It is possible to choose between two modes of user checking:

- Use Internal list. When you press the button "Edit Internal List of Users" in this mode, you can define users, who have the right of connect to the server. This is just text in the form USERNAME=PASSWORD on every line.

For example:

```
admin=1234567890
```

At the checking of a user, the server inspects presence of the user in a list and correctness of a password.

- System Administrators. In this mode the server checks presence of administrator's rights on a local computer when a client connects.

If the user isn't checked, the server returns an error and closes the connection.

By default, the System Administrators mode is on.

**Start Server**

It starts the server.

**Stop Server**

It stops the server. At that rate, all opened connections are closed.

**View/Edit connections**

Here it is possible to view all opened connections and to close selected connections.

## 2.9.2 Network Client

Network Client is integrated into the main window of RoboTask. At connection to remote RoboTask, a new branch, fitted to the service tasks, is added on the tree of tasks. Subsequent work with network tasks is completely "transparent". That is, you can trigger, edit, and delete tasks, just as if you would work with local tasks.

RoboTask can open several connections at the same time: one connection on each server. All connections are visible at the tree of tasks.

### 2.9.2.1 Connect to

Choose the menu item Remote/Connect to for fast connecting to a server. For connecting it is necessary to specify following parameters:

- **Computer Name or IP address.** It can be also the global network name (for example: www.mydomain.com), if the server is in a global Internet network. One can use the name localhost or the address 127.0.0.1 for access to a local computer (for example, to connect to a local service).
- **Port.** For example, the port number which is specified on the server. By default, 5500.
- **Domain.** This parameter is necessary if only the mode of authorization System Administrators is specified on the server. Specify a domain name or "." (Point) for a local user on the server.
- **User Name**
- **Password**

### 2.9.2.2 Define connections

If you need to constantly control the group of servers, you can create predefined connections. At that rate, corresponding items of menu Remote appear and then you can connect to the server "in one click."

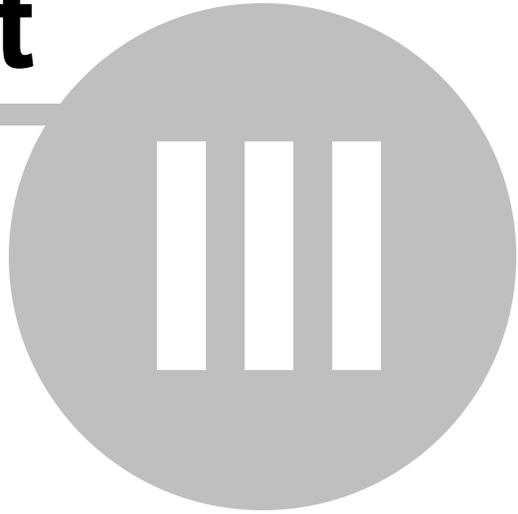
Choose the menu item **Remote | Define Connections** for creating/editing of predefined connections.

To create a define connection, you need to specify a name of connection for a menu item and then to specify parameters of connection.

If you not switch on the CheckBox "Save Password", RoboTask doesn't save a password for the next sessions. In this case, when trying to connect, a network client will request a password.

# Part

---



## Menus

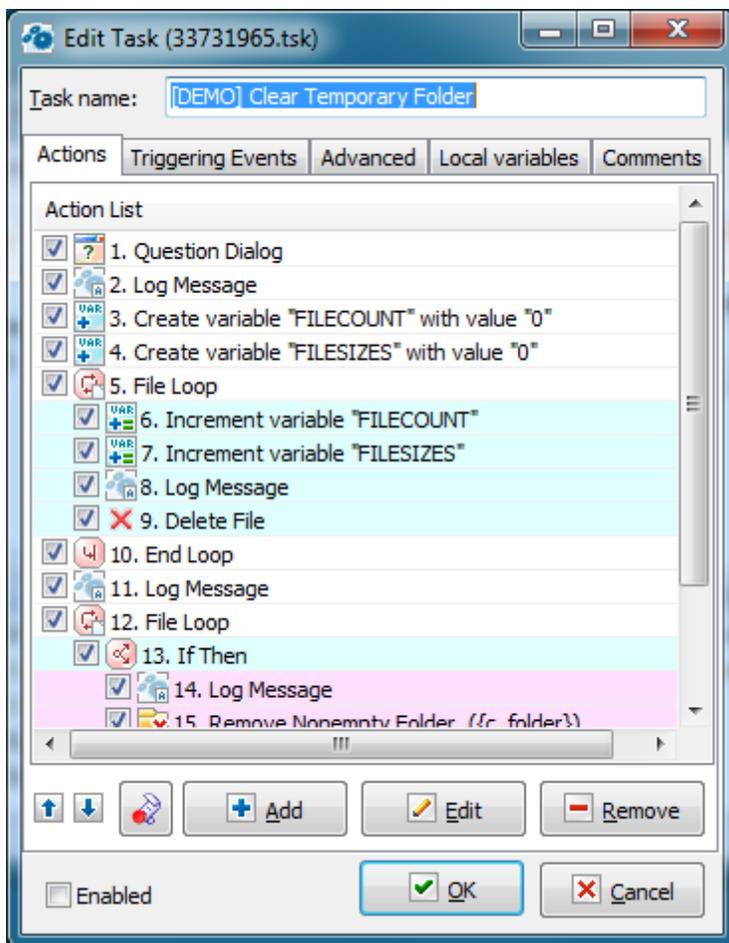
## 3 Menus

### 3.1 Task > New

Allows you to add a new task. After you select the command, the **Add Task** dialog box displays, allowing you to select the following tabs to create a task:

- [Actions tab](#)<sup>[26]</sup>
- [Triggering Events tab](#)<sup>[27]</sup>
- [Specific Settings tab](#)<sup>[28]</sup>
- [Local variables tab](#)<sup>[30]</sup>

#### Actions tab



#### Action List

Displays a list of the actions you have selected for the task you are creating or modifying. Click the **Add** button to add an action. Select an action in the **Action List** and click the **Edit** button to modify the action parameters.

When you start the current task, the actions run in the order in which they appear in this list. Click the & buttons to move an action up or down in the order listed.

**Record a macro**

In order to record mouse actions "as is" click this button. [Macro recorder](#)<sup>[14]</sup> catches mouse actions and saves them as a sequence of task actions.

**Add**

Place the cursor in a row of the **Action List** where you want to add an action. You can then click the **Add** button to display a list of actions that you can add to the task.

**Edit**

Select an action in the **Action List** and click the **Edit** button to modify the parameters, or settings, for the action.

**Remove**

Select an action and click the **Remove** button to remove the action from the task.

**Move Step Up**

Select an action in the **Action List** and click  to move the action up in the list. This ensures the action runs earlier in the process when the task runs.

**Move Step Down**

Select an action in the **Action List** and click  to move the action down in the list. This ensures the action runs later in the process when the task runs.

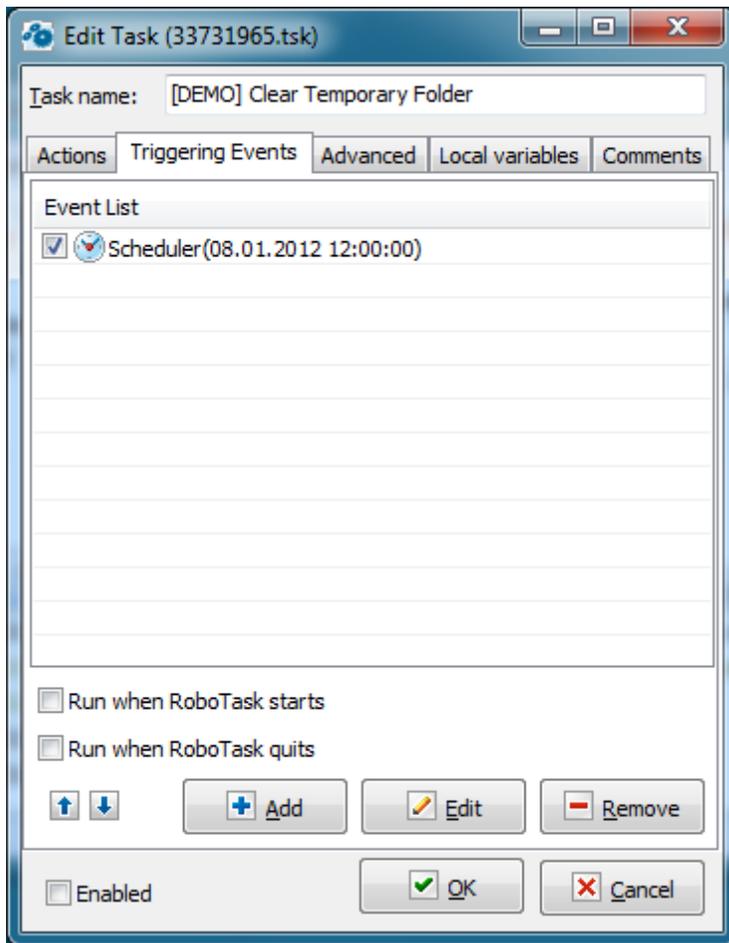
**Enabled**

If the check box is cleared, the task is disabled and you must run it manually by double-clicking the task in the RoboTask window.

**Tip**

- To change the **Enabled** setting, select a task in the RoboTask window and right-click.

**Triggering Events tab**



### Event List

Displays a list of the events you have selected for the task you are creating or modifying. The order in which events appear in the list does not matter.

### Run When RoboTask Starts

If this check box is selected, the task will run automatically when RoboTask starts.

### Run When RoboTask Quits

If this check box is selected, the task will run automatically when RoboTask quits.

### Add

Click the **Add** button to display a list of events that you can add to the task.

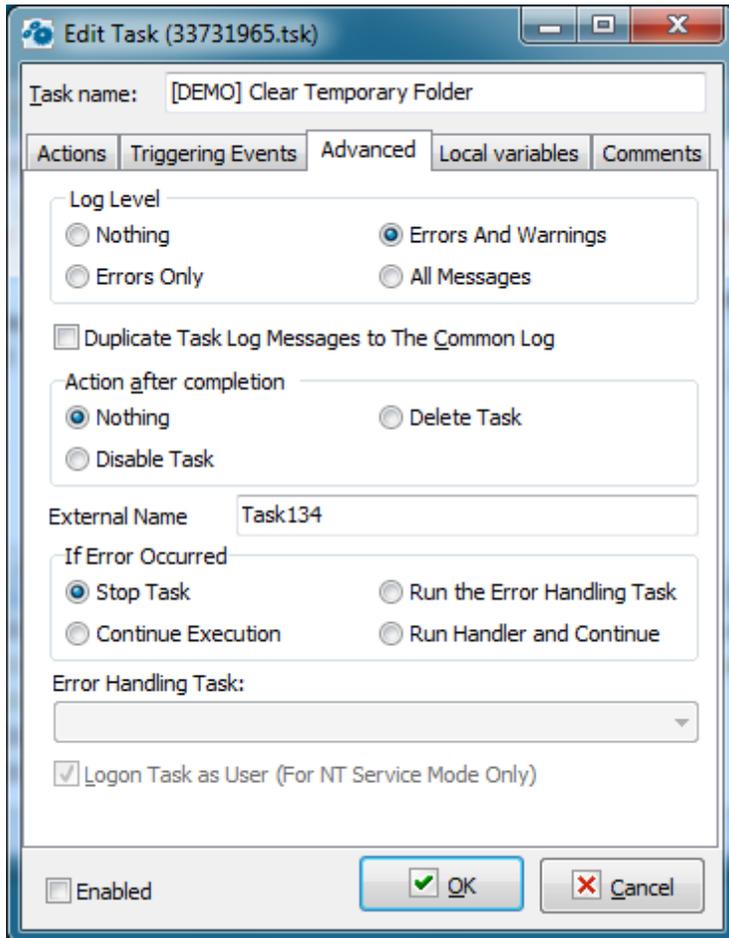
### Edit

Select an event in the **Event List** and click the **Edit** button to modify the parameters, or settings, for the event.

### Remove

Select an event and click the **Remove** button to remove the event from the task.

## Advanced tab



### Log Level

Select the status messages you want to record in the log file:

- **Nothing**
- **Errors Only**
- **Errors and Warnings**
- **All Messages**

### Action After Completion

Select an option to run a task:

- **Nothing** - Nothing happens when the task completes all actions.
- **Disable Task** - The task is disabled when the task completes all actions. This is useful if you want to run a task once and save the task to use again later. You can change this later.
- **Delete Task** - The task is removed from the computer permanently when the task completes all actions. This is useful if you want to run a task once and do not need it again. You cannot use the task again when you select this option and run the task.

### External Name

You can define an external name by hand. External Name is, per se, an alternative identifier of a task, which can help you to refer to a task ([see object RoboTaskApp<sub>178</sub>](#)). RoboTask controls the uniqueness of the External Name and, in case of name clash, resets it to the default value. The default value looks like TaskNNNN, where NNNN – is just an ordinal number.

### If Error Occurred

Here you can define the behavior of a task when an error is raised in any step. You can choose from 3 variants:

- **Stop Task** – the default action. When an error is raised, performance of a task is stopping.
- **Continue Execution** – when an error is raised, performance of a task is going on. In other words – it is ignoring of the error. We recommend to use this mode only if it is really necessary.
- **Run the Error Handling Task** – run the other task if an error is raised. The running task, per se, is an error handler. In this case, performance of the main task is stopping. See below the describing of the transferred parameters.

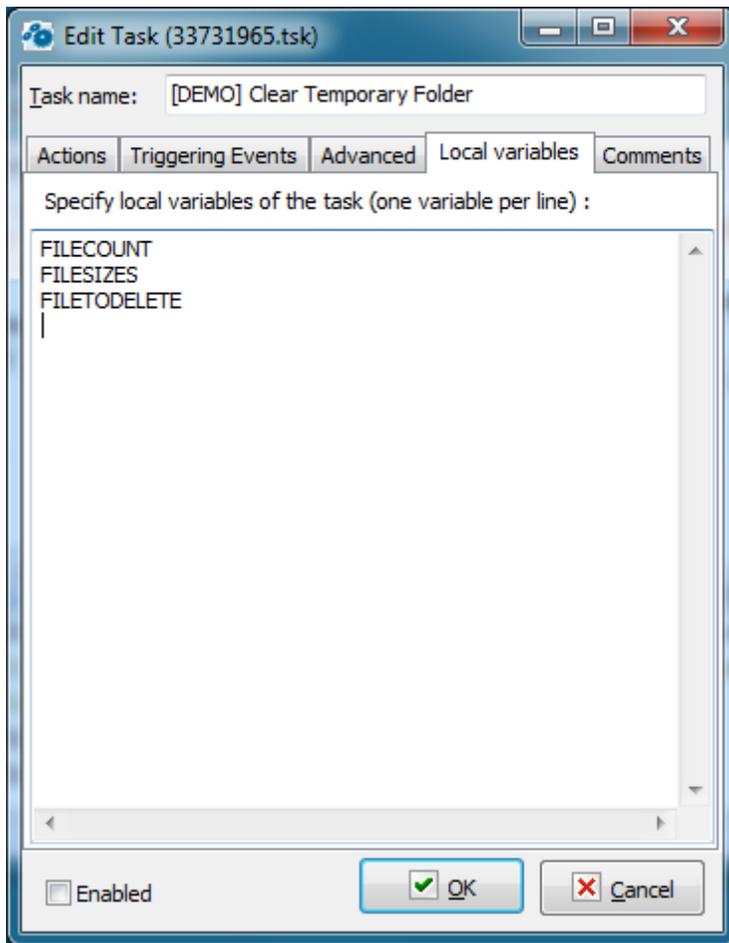
### Error Handling Task

Choose the task of error handler in a drop-down list. This task will be started up in case of appearance of error. When the task is being started up, the fixed kit of parameters is transferring to it:

- **LastErrorTaskID** – the internal ID of the task, in which an error was raised. It is a formal identifier of the task like a number.
- **LastErrorTaskName** – the name of the task, in which an error was raised.
- **LastErrorTaskExternalName** – an External Name of the task, in which an error was raised.
- **LastErrorStep** – the number of the step (starting from 1), in which an error was raised.
- **LastErrorCode** – code of the error. There is the own kit of codes of errors for any type of actions (see description of actions). By default – 0.
- **LastErrorDescription** – the description of the error. It depends on a type of action. By default, it is the line "General error".

LastErrorCode and LastErrorDescription depend on a type of action. Descriptions of error codes are put into the help of actions (help file)

### Local variables tab



You can declare local variables on the "Local variables" tab of the task editor. There are two ways to declare them:

Enumerate the names of the variables;

Assign initial values to them.

If you enumerate them, just fill the list of names for local variables: one variables per line. For example, like this:

```
LineCount
FileToUpload
CurrentFile
```

etc.

To assign an initial value, specify this value after the "=" character. For example, like this

```
LineCount = 0
FileToUpload = c:\incoming\list.txt
CurrentFile = note.txt
```

etc.

You can assign initial values not to all variables. By default, local variables contain an empty value (an empty string).

---

By default, the list of local variables is empty. It means that all variables that are used in a task will be processed as global ones when you run the task. See more information about [local variables](#)<sup>[11]</sup>.

## 3.2 Task > Edit

Allows you to edit a task. After you select the command, the Edit Task dialog box displays, allowing you to select the following items to edit the task:

- [Actions tab](#)<sup>[26]</sup>
- [Triggering Events tab](#)<sup>[27]</sup>
- [Advanced tab](#)<sup>[28]</sup>
- [Local variables tab](#)<sup>[30]</sup>

## 3.3 Task > Delete

Removes a task from the RoboTask and deletes its associated file.

## 3.4 Task > Run

Starts the task that you selected in the RoboTask window. This is useful for testing your task to ensure it runs correctly.

## 3.5 Task > Stop

Allows you to stop the task that you are running in the RoboTask window.

## 3.6 Task > Duplicate

Copies the task that you have selected in the RoboTask window. This is useful if you want to quickly create several tasks and then edit them.

### 3.7 Task > Create Shortcut on Desktop

Adds a shortcut to the task on your desktop. This is useful if you want to quickly start a task without opening RoboTask window. After you select one or more tasks in the RoboTask window, click **Task > Create Shortcut on Desktop**. The shortcut appears on your desktop with the name of the task.

For example, if you select a task named "Hello, World!" and click this command, a shortcut called "Hello, World!" appears on your desktop.

### 3.8 Task > Exit

Closes the RoboTask window and stops the software.

### 3.9 Options > View Log

Displays or hides the log window at the bottom of the RoboTask window. The log window shows the results of the task that you run and any errors that occur.

### 3.10 Options > View Toolbar

Displays or hides the buttons at the top of the RoboTask window. These buttons allow you to add, edit, run, and stop tasks.

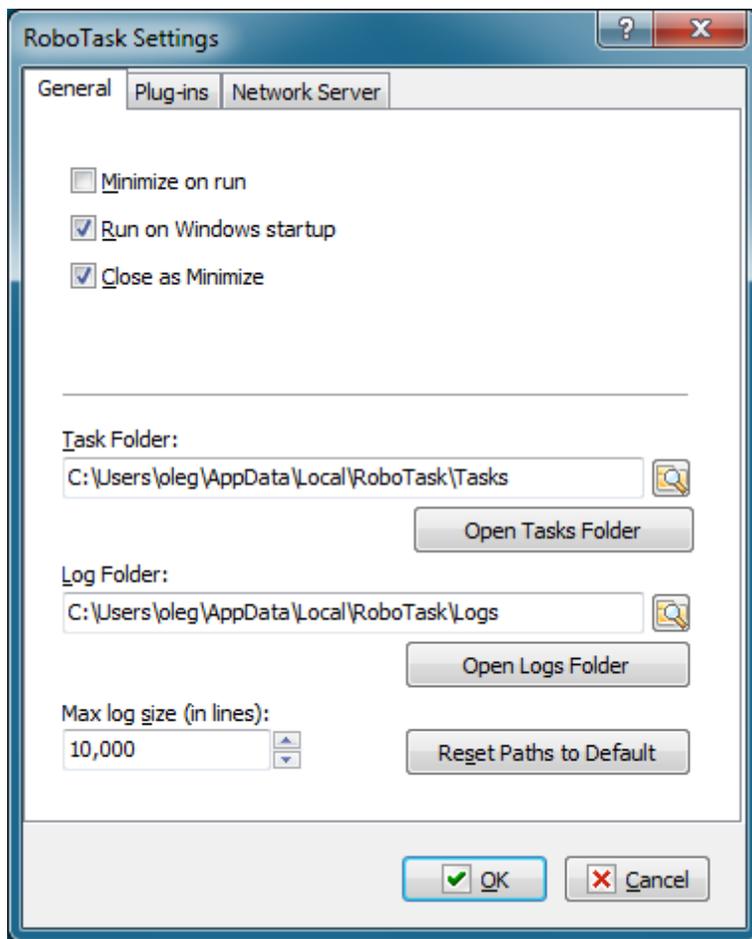


### 3.11 Options > Settings

Allows you to set preferences for RoboTask. You can set preferences for how RoboTask starts, plug-ins, or service mode.

- [General tab](#)<sup>[33]</sup>
- [Plug-ins tab](#)<sup>[35]</sup>
- [NT Service tab](#)<sup>[35]</sup>
- [Network Server Tab](#)<sup>[36]</sup>

#### General tab



### Minimize on Run

Click this check box so that RoboTask is minimized when you run the software.

### Run on Windows Startup

Runs RoboTask as soon as Windows starts up.

### Close as Minimize

If this option is selected, when you click the X button on the main window caption, RoboTask will minimize. This is useful for preventing of casual closing of RoboTask.

### Task Folder

Set the folder where you want to save tasks that you create. You can click the browse button to select a folder.

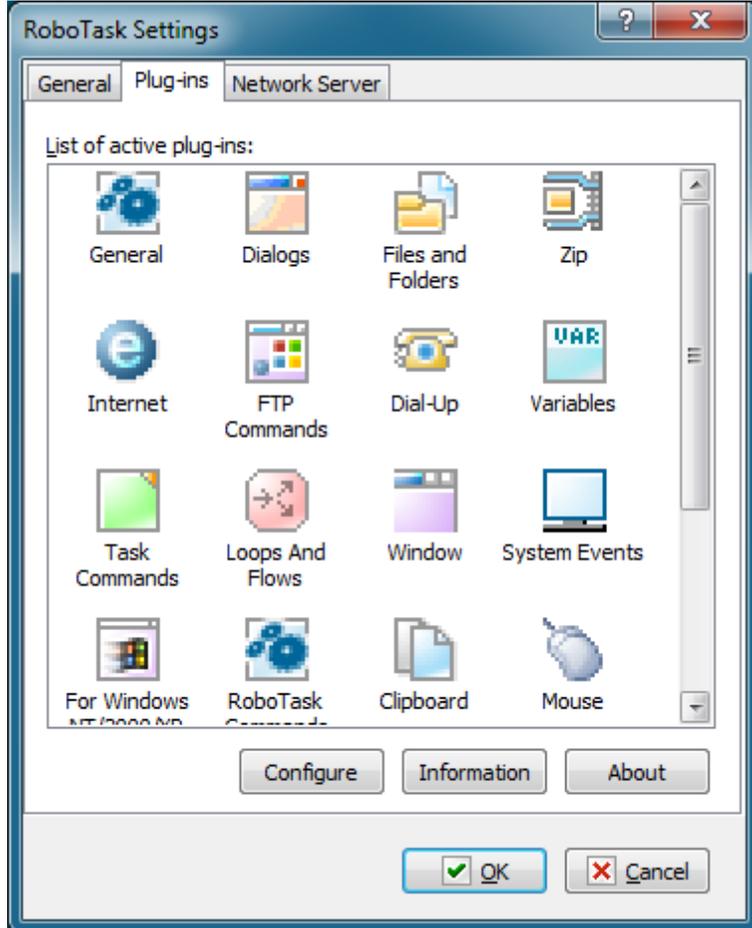
### Log Folder

Enter a folder where you want to save log files. You can click the browse button to select a folder.

### Max Log Size (in Lines)

Enter how many lines of data to record in a log file. For example, if you enter 300, the log file will contain no more than 300 lines of text.

## Plug-ins tab



### List of Active Plug-ins

Displays a list of available plug-ins for RoboTask. You can get more information when you click the **Information** or **About** buttons.

### Configure

Select a plug-in from the **List of Active Plug-ins** box and click **Configure** to enter settings.

### Information

Displays a list of components for the plug-in that you have selected in the **List of Active Plug-ins** box.

### About

Displays information about the company or authors of the plug-in that you have selected in the **List of Active Plug-ins** box.

## Service tab

On Windows NT, Windows 2000 and Windows XP you can run RoboTask as a Service. A Service is a background process that is loaded by the Service Control Manager of the Windows NT operating system. Services are usually loaded when Windows boots up, before anyone logs on.

Service mode is useful if you want to run a process in the background. However, services are background programs that run all the time and are activated automatically when Windows NT boots up. Services stop when Windows NT shuts down.



### When RoboTask is Running as NT Service

Allows you to specify settings when you want to run RoboTask as a Service.

### Logon Process as Specified User

Select this option if you want the RoboTask log on as Specified User and get its privileges.

### Domain Name

Enter the name of the domain or server whose account database contains the specified UserName account.

### User Name

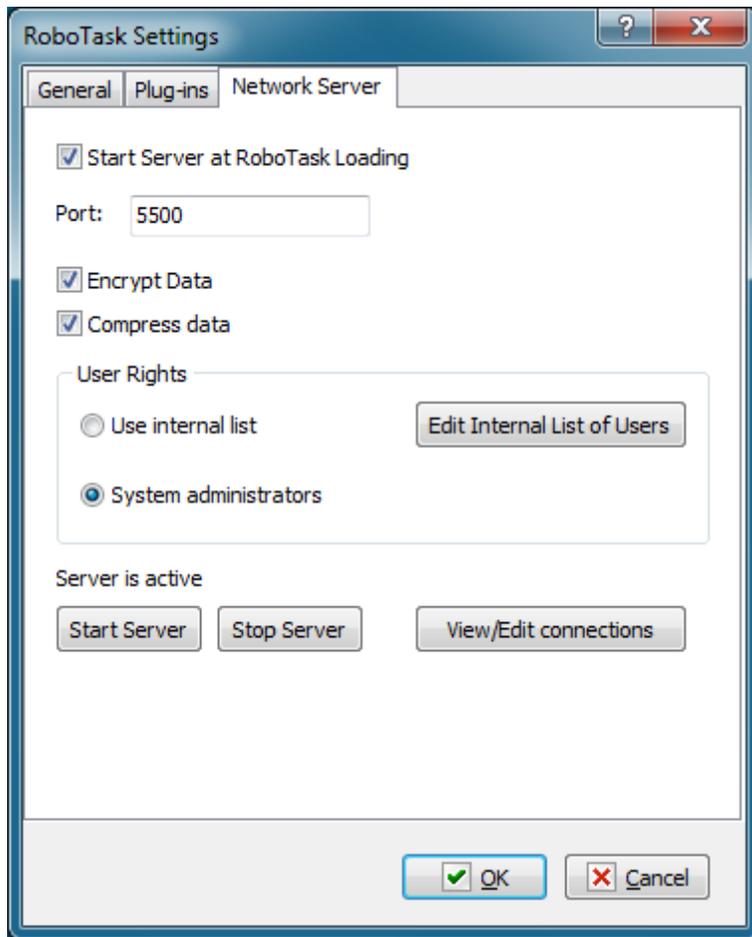
Enter the name of the user account to log on to.

### Password

Enter the password of the user account.

### Network Server tab

On this tab you can set up server parameters of RoboTask. Read more about client-server mode [here](#)<sup>[22]</sup>.



### Start Server at RoboTask loading

It is on by default. Switch off this checkbox if you want to refuse from remote administration.

### Port

This is the number of a port, on which the server accepts connections. By default, it is 5500. If you use more than one instance of RoboTask at your computer, apply different ports to avoid conflicts.

For example: 5500 is for an application, 5501 is for a Service.

### Encrypt data

This is the mode of traffic encryption. By default, it is on. The mode of ciphering is specified by server. A client is tuned on the mode of encryption automatically. RoboTask provides the traffic encryption with dynamic keys using CryptoAPI system functions.

### Compress data

This is the mode of traffic compression. By default, it is on. The mode of traffic compression can be useful when connection is slow (for example, DialUp or GPRS). The mode of traffic compression (packing) is specified by server. A client is tuned on the mode of compression automatically.

### User Rights

This is the check mode of user's rights at connection. It is possible to choose between two modes of user checking:

- Use Internal list. When you press the button "Edit Internal List of Users" in this mode, you can define users, who have the right of connect to the server. This is just text in the form USERNAME=PASSWORD on every line.

For example:

```
admin=1234567890
```

At the checking of a user, the server inspects presence of the user in a list and correctness of a password.

- System Administrators. In this mode the server checks presence of administrator's rights on a local computer when a client connects.

If the user isn't checked, the server returns an error and closes the connection.

By default, the System Administrators mode is on.

### Start Server

It starts the server.

### Stop Server

It stops the server. At that rate, all opened connections are closed.

### View/Edit connections

Here it is possible to view all opened connections and to close selected connections.

## 3.12 Options > Variables

Allows you to define variables for RoboTask. This command also displays a list of system variables available in RoboTask.

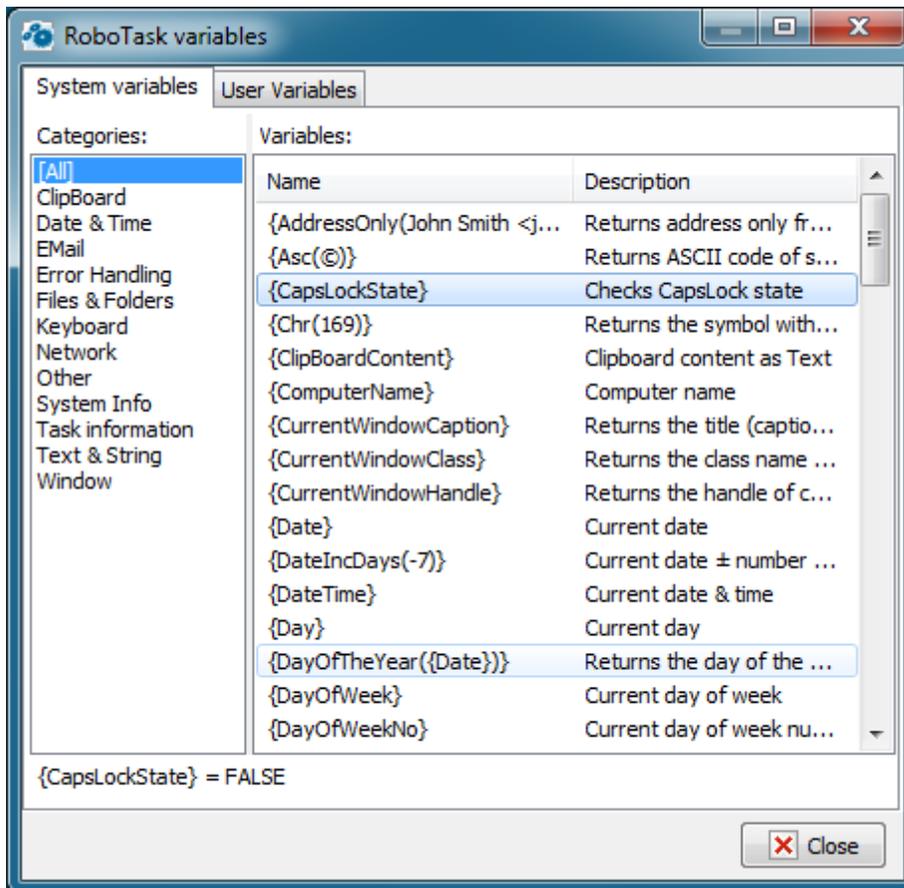
- [System Variables tab](#)<sup>[38]</sup>
- [User's Variables tab](#)<sup>[39]</sup>

Variables are strings that contain information such as drive, path, or file name. Variables can control the behavior of various processes. For example, the {TEMPDIR} system variable specifies the location where processes, such as tasks or programs, place temporary files.

You can not modify System variables, but can add, modify, or remove your own variables on the **User Variables** tab.

### System Variables tab

The **System Variables** tab displays a list of the system variables.



### Categories

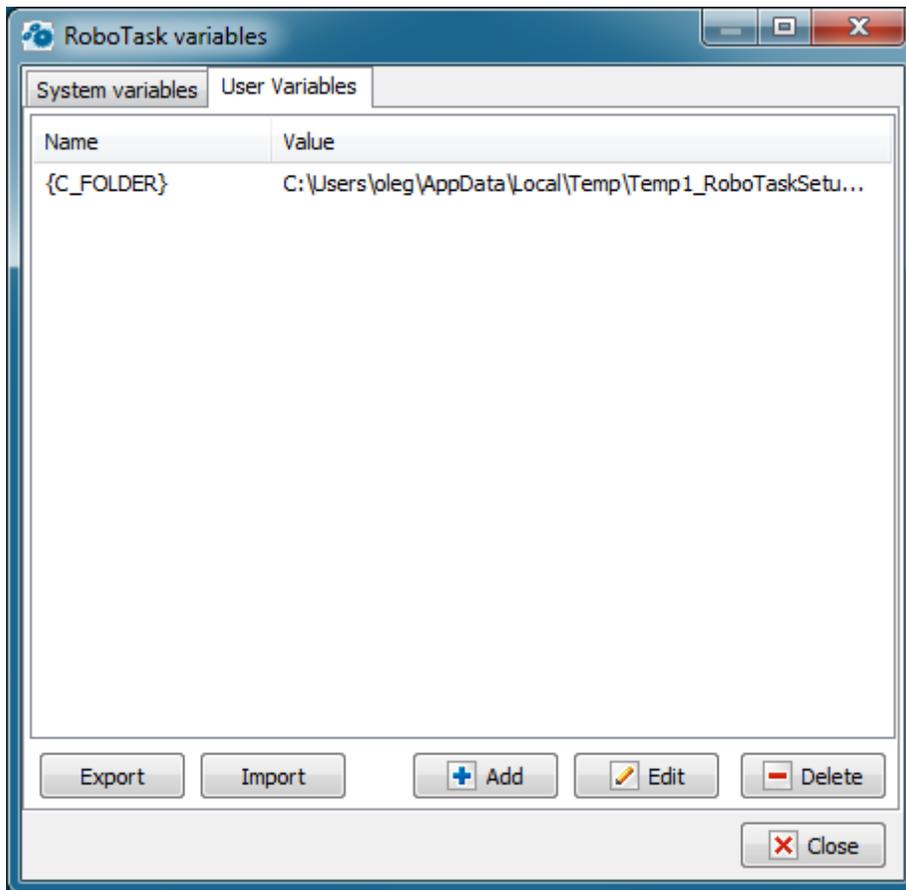
Displays a list of all the system variables and their values. You can click **[All]** to see a complete list or click one of the following categories:

- **Date & Time**
- **Files & Folders**
- **System Info**
- **Window**
- **etc...**

### Variables

Displays the system variable names and their descriptions.

### User's Variables tab

**Name**

Displays the name of the user variables that you create.

**Value**

Lists the value of the user variables that you set when you create or modify the variable.

**Add**

Allows you to create a new user variable to use in many tasks.

**Edit**

Allows you to edit the value of a variable.

**Remove**

Allows you to remove the selected variable.

### 3.13 Tools > Enable Automation

If this option is cleared, the tasks will not run automatically, i.e. all your tasks are disabled. You may run tasks only manually.

### 3.14 Tools > Enable All Tasks in List

Enables all tasks that are listed in the current task folder. To enable one task, double-click it and, in the appeared dialog window, select the **Enabled** check box and click OK.

### 3.15 Tools > Disable All Tasks in List

Disables all tasks that are listed in the current task folder. To disable one task, double-click it and, in the appeared dialog window, clear the **Enabled** check box and click OK.

### 3.16 Help > Help Topics

Displays this RoboTask Help.

### 3.17 Help > Send Bug Report

Displays a dialog window which allows to send a report about a bug you found to the developers of RoboTask. Enter information about a problem and then click the **Submit** button.

### 3.18 Help > About

Displays information about your version of RoboTask.

# Part

---



IV

## Actions

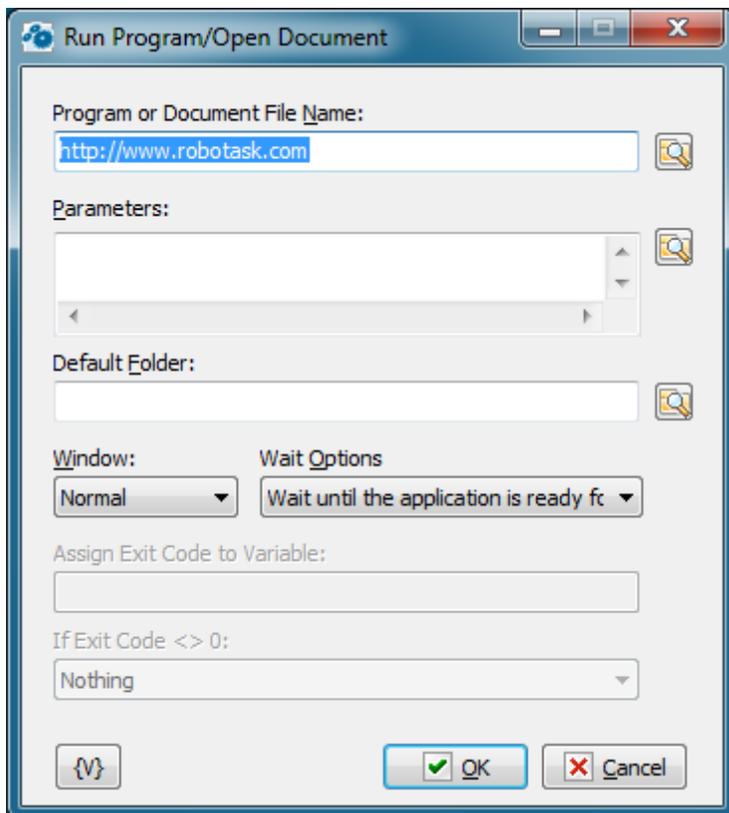
## 4 Actions

### 4.1 General

#### 4.1.1 Run Program/Open Document

Use this action to run the application or open a document that you specify. If a document is specified, the application associated with the document opens it.

**Run Program/Open Document** is one of the most commonly-used actions in RoboTask and one of the most powerful. The most common usage is to start an application or open a document and then send keystrokes or perform other interactive actions. For example, you can use this action to open a log file in Notepad immediately after you back up your files.



##### **Program or Document File Name**

Enter the file name of the target application or a document. If a document file is specified, the associated application is used to open it.

##### **Parameters**

Enter the parameters, if any. If the **Program or Document File Name** box specifies an executable file, you can specify the parameters to be passed to the application. The format of this parameter is determined by the application. If **Program or Document File Name** box specifies a document file, the information in the **Parameters** box is ignored.

**Default Folder**

Specifies the default folder to start the task in. If not specified, the folder of the application is used.

**Window**

Specifies how an application is displayed when it is opened.

**Wait Options**

There are three choices:

- **Don't wait** - the next action executes immediately after the program starts.
- **Wait until the application is ready for input** - the next action executes only after the program started and is ready for input. This option is useful when you want to perform any interactive actions like [Send Keystrokes](#)<sup>[45]</sup>.
- **Wait until the application exits** - the next action executes only after the program exits.

**Assign Exit Code to Variable**

You can assign exit code of running program to variable. This has a sense only if you set "Wait until the application exits" wait option.

**If Exit Code <> 0**

Usually exit code of command line utility equals to 0. Usually this means that executing of the program is OK. Non zero exit code may indicate that some problem has occurred at runtime.

You can choose one of three options:

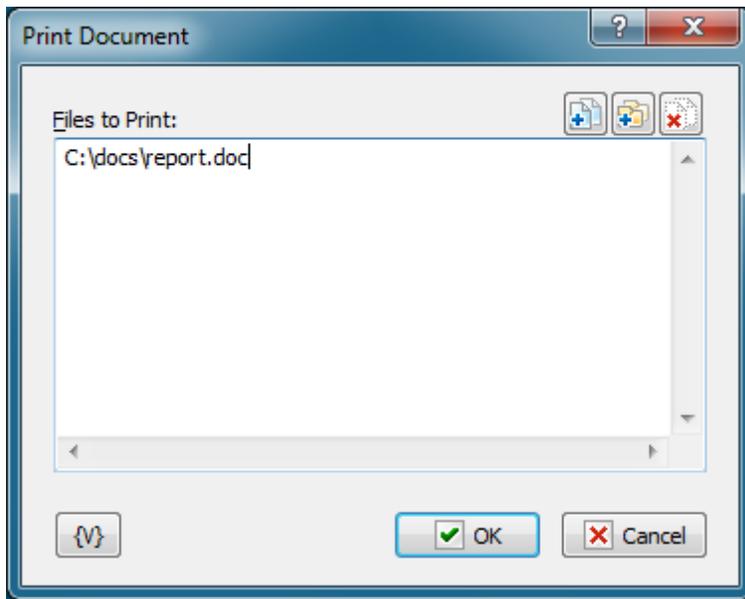
- **Nothing**
- **Stop the task** - if exit code is non-zero RoboTask stop the task
- **Raise the Error** - if exit code is non-zero RoboTask raises the error with code 1 and description "Exit code is not equal to 0" (also see [Error Handling](#)<sup>[21]</sup>)

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.1.2 Print Document

Use this action to print one or more document files, such as Microsoft Word .Doc files, to the default printer. For example, you can use this action to automatically print daily backup logs.



### Files to Print

Enter the files that you want to print. This is a required option. Wildcards are allowed. Click the Plus button to add files to the list. To clear the list, click the X button.

#### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

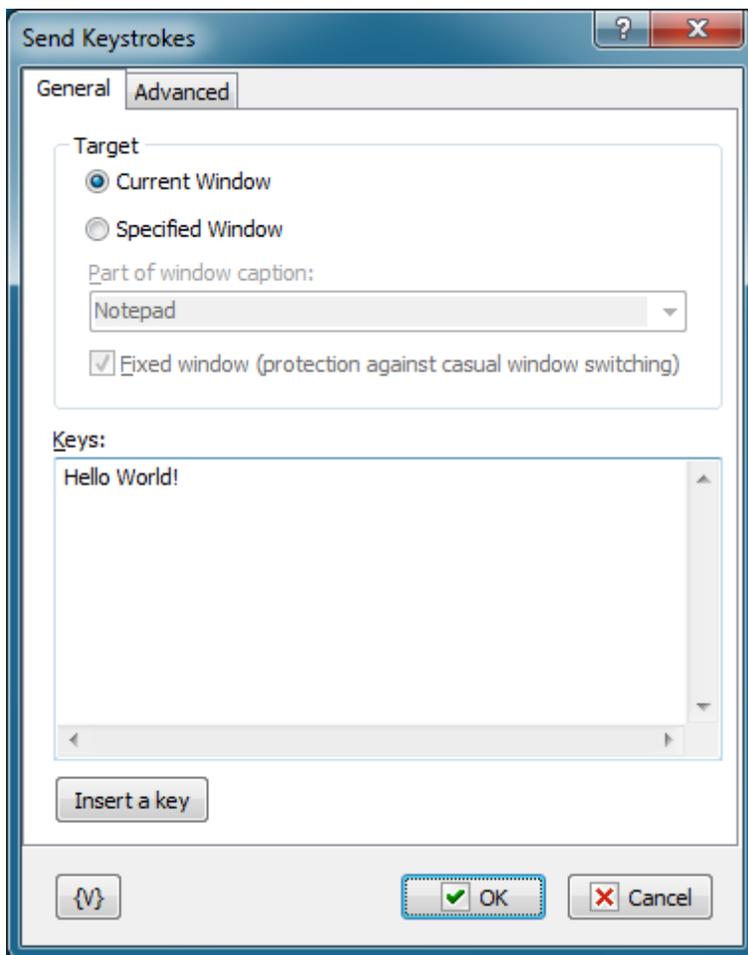
- **C:\Text Files\\*.txt** - all the files with the .txt extension in the C:\Text Files folder.
- **C:\Doc Files\mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **{bakext}** - file mask(s) from the {bakext} variable.

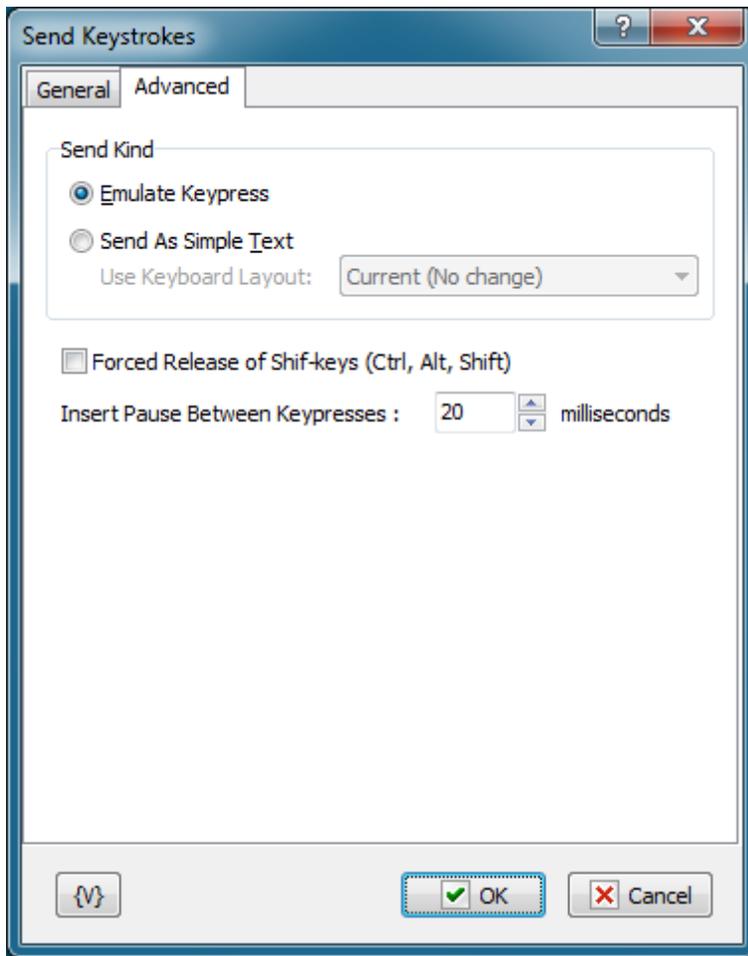
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.1.3 Send Keystrokes

Sends keystrokes to a window that you specify or the window that is currently in focus. After you have opened an application programmatically, you can use this action to access menu commands using shortcut, or hot, keys, such as ALT + F to open the **File** menu. **Send Keystrokes** is one of the most powerful actions for manipulating applications.





### Target

Specifies the window to send the keystrokes to. There are two choices:

- **Current Window** - the keystrokes are sent to the currently focused window.
- **Specified Window** - the specified window will be focused and the keystrokes are sent to it.

### Part of Window Caption

Enter the caption, or part of the caption, of the window to send keystrokes to. This option is available only when you the Specified Window option is selected.

### Fixed Window

Click this check box when you want to prevent a user from switching windows during the time that the action runs.

### Keystrokes

Specifies the keystrokes that the action sends to the window. For the alpha-numeric characters, type these keys on the keyboard. For the special keys, such as Esc, F11, Left Arrow Key, and Ctrl+Alt+A, click the **Insert a Key** button.

### Insert a Key

Click this button to insert special keys such as Esc, F11, Left Arrow Key, and Ctrl+Alt+A. You can also insert a pause between keystrokes to give time for program operations, such as opening a file.

### Send Kind

You can send keystroke in two ways: Emulate Keypresses or As Simple Text. The question is that different programs response on the simulation of keystroke differently.

- **Emulate Keypresses** –RoboTask emulates a hardware keystroke in this mode. It is possible to send keys combinations with Ctrl, Alt, Shift in this mode. Please note, that symbols ^, %, +, #, {, }, (, ) are used for recording keys combinations. These symbols are reserved for this mode. How to simulate sending of these keys is shown below.
- **Send As Simple Text** – simulation of a simple typesetting is produced in this mode. The text is sent as is, without using combinations. Pay attention that the sending of key presses is sensitive to the current Keyboard Locale.

### Forced Release of Shift-keys

Switch on this parameter, if you want to make forced release of Shift-keys. You can turn off this parameter if you want to send some keystrokes when some of Shift-keys are held down (see [Shift Keys State](#)<sup>[49]</sup>).

### Insert Pause Between Keypresses

Insert pauses between separate keys presses in milliseconds. This is important particularly for the "Emulate Keypresses" mode. Pauses are necessary to give the application some time for response to the press of keys or keys combinations. By default, 20 milliseconds are preset. As a rule, this is enough for most cases.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### Standard keys for GDI windows

When you work with emulated key presses, we recommend that you use standard key combinations for controlling dialog boxes. These combinations work in most dialog boxes in Windows applications and you will not have to use the mouse.

- Tab** – move to the next control element of the dialog box;
- Shift-Tab** – move to the previous control element of the dialog box;
- ESC** – the same as a click on the Cancel button in most cases;
- Enter** – the same as a click on the OK button in most cases;
- Ctrl-Tab** – move to the next tab if the dialog box contains tabs;
- Ctrl-Shift-Tab** - move to the previous tab;

### Key combinations

- Ctrl** - ^(<Letter>), e.g. Ctrl-S -> ^(S)
- Alt** - %(<Letter>), e.g. Alt-W -> %(W)
- Shift** - +(<Letter>), e.g. Shift-W -> +(W)
- Win** - #(<Letter>), e.g. Win-E -> #(E)

**How to simulate reserved symbols: ^ % + # { } ( )**

You can do this by using *Shift* combinations, like this

^	- Shift-6 , i.e. <b>+(6)</b>
%	- Shift-5 , i.e. <b>+(5)</b>
+	- Shift-= , i.e. <b>+(=)</b>
#	- Shift-3 , i.e. <b>+(3)</b>
{	- Shift-[ , i.e. <b>+([])</b>
}	- Shift-] , i.e. <b>+([])</b>
(	- Shift-9 , i.e. <b>+(9)</b>
)	- Shift-0 , i.e. <b>+(0)</b>

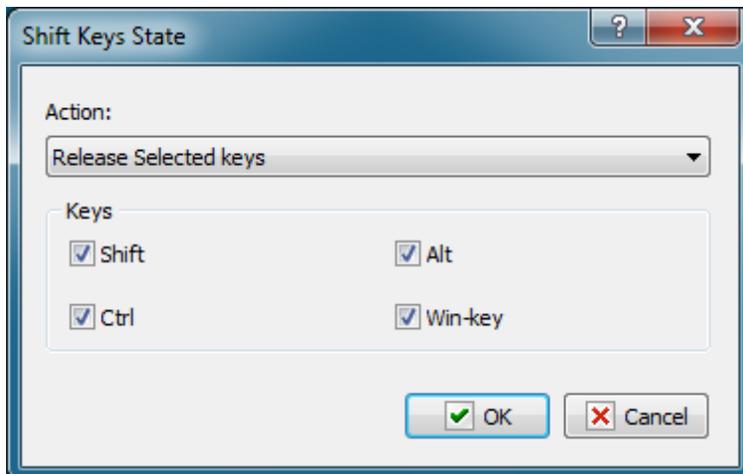
**4.1.4 Shift Keys State**

This action serves for imitation of retention of Shift-keys in pressed condition or in forced release. Shift-keys are used in various combinations: Shift, Ctrl, Alt, Win-key. Some programs use complex combinations.

For example:

- when Ctrl is pressed, push K and C sequentially
- pressing Left Button of a mouse when Ctrl and Alt are held down

The action "[Send Keystrokes](#)<sup>[45]</sup>" can't ensure imitation such combinations.

**Action**

Select a necessary action:

- **Release selected keys** – forced release of specified keys.
- **Hold down selected keys** – pressing of specified keys and holding them in pressed condition, until you see the action Release of Keys. Please note that all Shift-keys will be released, if you switch on the parameter "[Forced release of Shift-keys](#)<sup>[45]</sup>" when Send Keystrokes are in use.

**Keys**

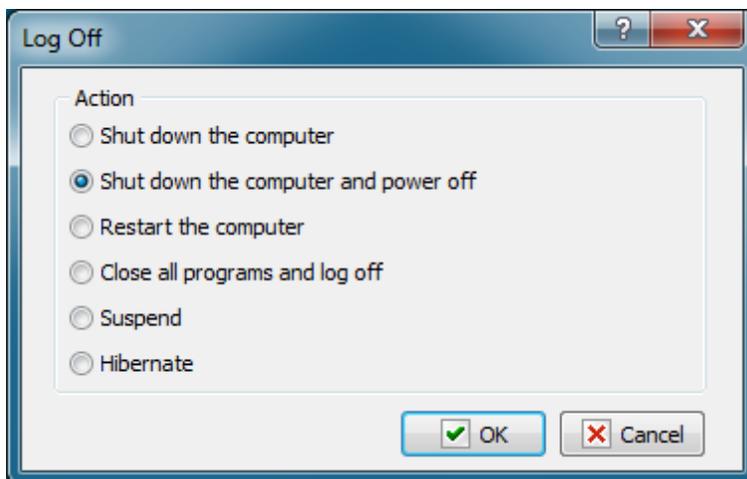
Mark necessary keys in any combination.

### 4.1.5 Log Off

Completes one of the following actions:

- Logs off the current user,
- Shuts down the computer,
- Restarts the system.

For example, you can use this action to automatically shut down a computer every night after your defrager program finishes cleaning up your hard drive.



#### Action

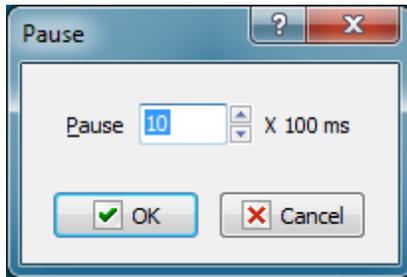
Select one of the following values:

- **Shut down the computer**
- **Shut down the computer and power off**
- **Restart the computer**
- **Log Off**

### 4.1.6 Pause

Waits for a specified number of milliseconds and then continues executing tasks. This action is useful when you want to perform extensive operations that require a delay between actions.

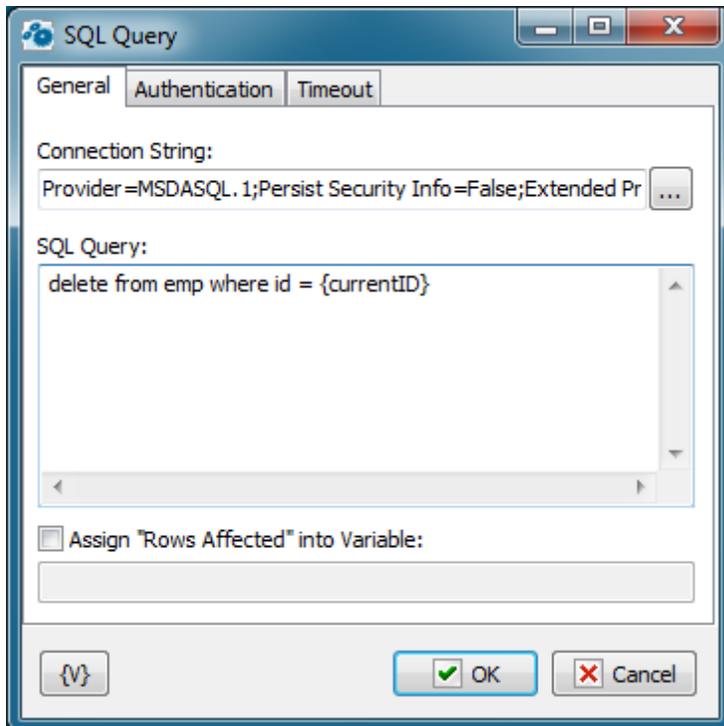
For example, you might want to create a pause between printing a document and closing it to prevent a printer error. This pause allows time for the document to spool.

**Pause**

Enter the number of milliseconds \* 100.

### 4.1.7 SQL Query

Executes an SQL query. This action is useful when you want to update a database. If you want to SELECT from a database, use the [Dataset Loop](#)<sup>126</sup> action instead.

**Connection String**

A string with attribute/value pairs for information required to log on to a database and point to a specific database. Click the browse button to open the Microsoft **Data Link Properties** dialog box. You can press F1 to access Help about this dialog box.

**SQL Query**

Enter the query string in SQL syntax.

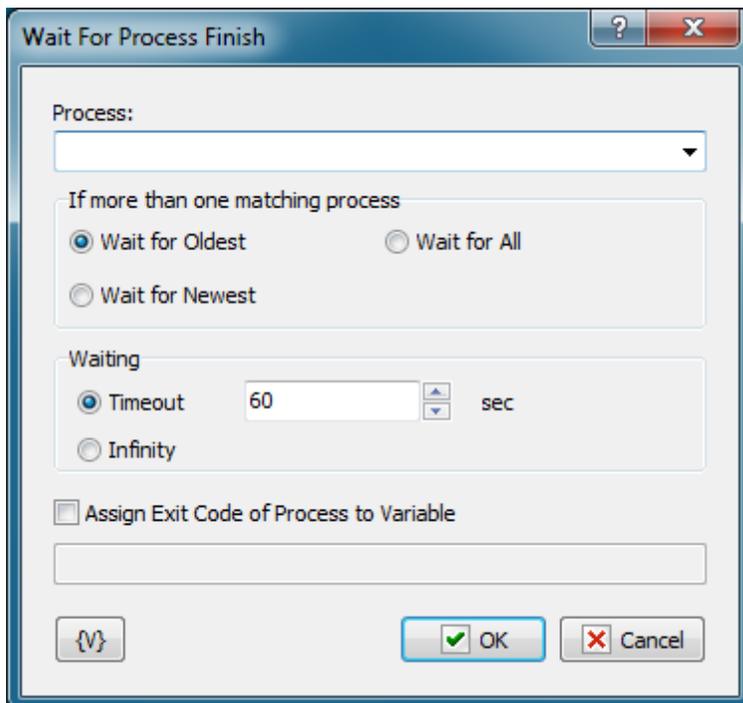
**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.1.8 Wait for Process Finish

Pauses the task and waits until the selected process (processes) is finished. After the monitored process is finished, the task is resumed. If the specified process is not in the memory at the moment when the action is due, the task is not paused.

This action is necessary to synchronize your task with an external application. For example, you may need to wait until the packing program finishes its work before you can copy the packed file to the Backup folder.



##### Process

You can either select the process from the list of processes that are currently running or type its name manually. You do not have to write the full path to the EXE file if it does not matter where this application was launched from.

##### If more than one matching process

If there are several running processes matching the specified one at the moment when the action is due, you can specify one of three modes

- **Wait for Oldest** - wait until the process launched first from the matching group is finished;
- **Wait for Newest** - wait until the process launched last from the matching group is finished;

- **Wait for All** - wait until all matching processes are finished.

### Waiting

This parameter defines the timeout in seconds to wait until the process is finished. You can also specify an infinite timeout.

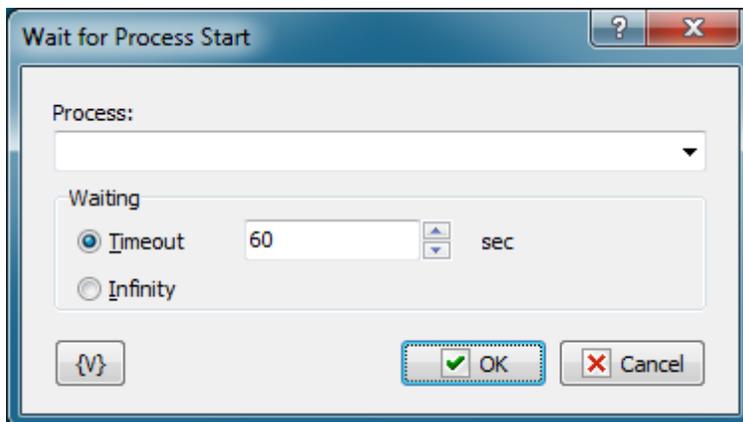
If the timeout is over, but the process is not finished, the action will generate the following error: Timeout is over.

### Assign Exit Code of Process to Variable

If you select waiting for one process (Wait for Oldest or Wait for Newest), you can assign its exit code to a user variable. A lot of console utilities (for example, packing software) can use this code to inform the application that initiated them about errors in their work.

## 4.1.9 Wait for Process Start

Pauses the task and waits until the selected process is started. Unlike action "[Wait for Process Finish](#) [52]", this one waits for the appearance of the specified process in memory. If that process is there already, there is no waiting, and control is passed to the next step of a task.



### Process

Specify or select from the list the name of the process that needs waiting for. You can use variables.

If you specify the name of EXE-file with a full path, RoboTask will look for the process in memory by its full name. If the path is not specified, RoboTask will look for the process only by the name of the EXE-file without a path.

### Waiting

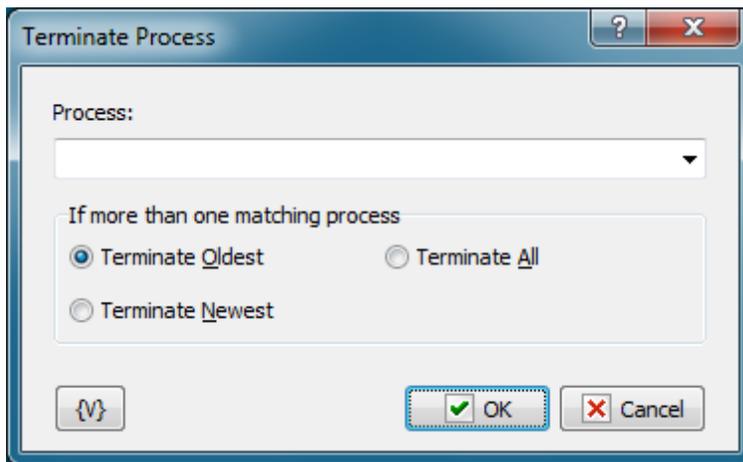
Specify the timeout, during which you need to wait for an appearance of the process. You can specify a timeout in seconds or select the value "Infinity". If the process does not appear in memory during specified time, the action generates an error type "Timeout error. Process didn't start"

### 4.1.10 Terminate Process

Forces the specified process to be terminated. This method is not a normal way to exit Windows applications. To exit some application normally, you will have to close the main window of this application. (see the operation Window command).

**Note:**

*When you force a process to be terminated, you may lose data (for example, a process is terminated when a file is being saved) or system resources occupied by the process may remain occupied. In this case, multiple process termination may result in very unpleasant consequences. Although quite often it is a safe procedure.*



**Process.**

Enter the name of the process that should be terminated. If the process already exists in memory, you can select it from the drop-down list. The name of the process coincides with the name of its EXE file. There is no need to specify the full path in the process name. If you do not specify the full path, RoboTask will search for the specified process without taking into account its path. Also, you can use variables when specifying the name of the process.

**If more than one matching process.**

Specify what should be done, if there are several matching processes in memory at the moment of the operation. For example, you started several instances of Notepad.exe with different text files.

There are three options to choose from:

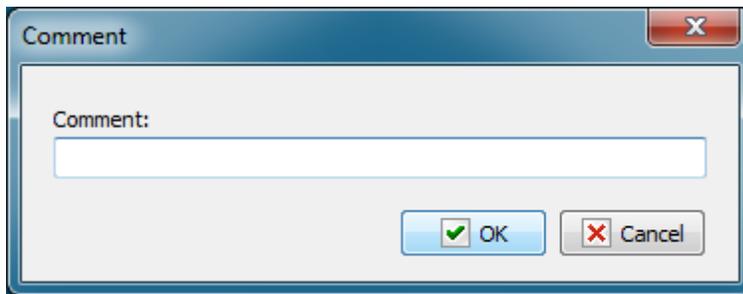
**Terminate Oldest** – terminate the process started first;

**Terminate Newest** – terminate the process started last;

**Terminate All** – terminate all matching processes.

### 4.1.11 Comment

This action does nothing and is used only as a visual comment for the task.



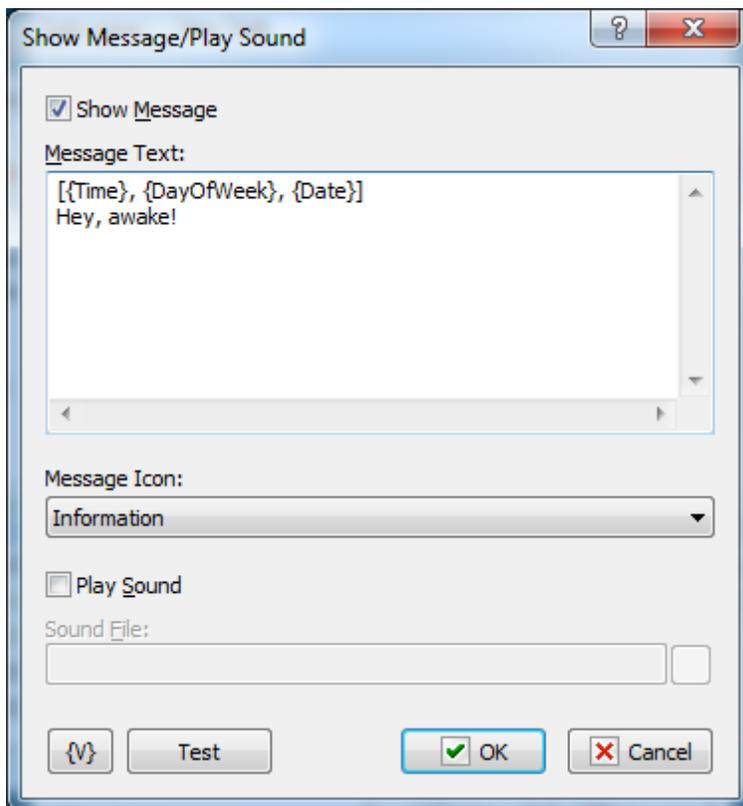
## 4.2 Dialogs

### 4.2.1 Show Message/Play Sound

Displays a message box that displays custom text or plays a sound file. For example, you can use this action to display a message and play a sound every time your backup finishes archiving the files.

#### Tip

- The message box displayed by this action has an **OK** button. You can display a message box with a **Yes** and **No** button with the [Question Dialog](#)<sup>[57]</sup> action.



#### Show Message

Click this check box to display the message box.

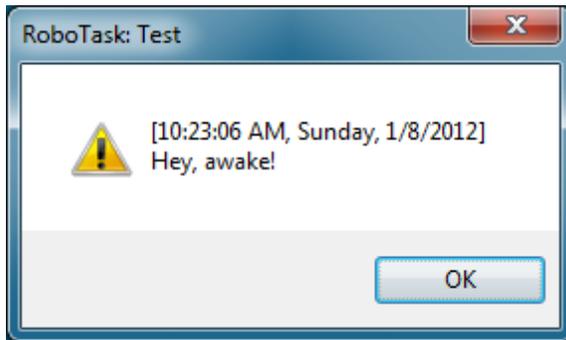
**Message Text**

Enter the text for the message box.

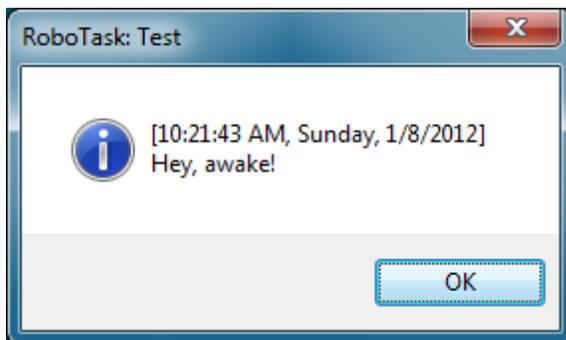
**Message Icon**

Select the style of the icon for the message box:

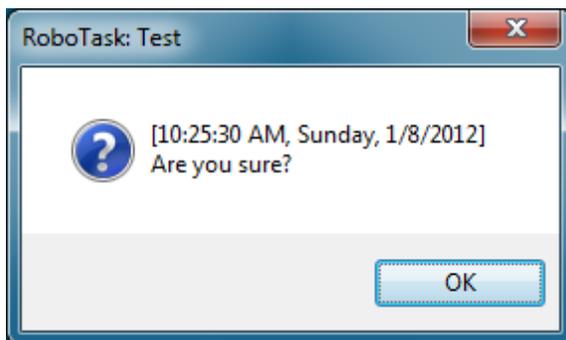
- **Exclamation**



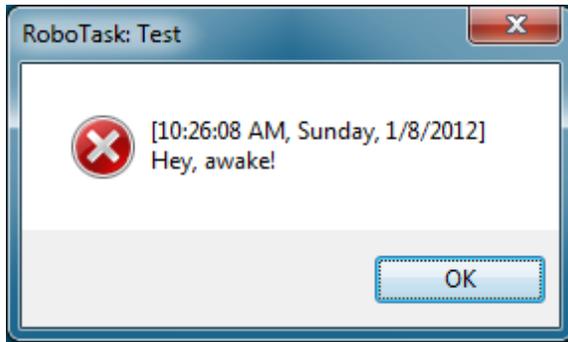
- **Information**



- **Question**



- **Error**

**Play Sound**

Click this check box to play a sound. When you click the check box, you can click the browse button to select a .wav file on the computer.

**Test**

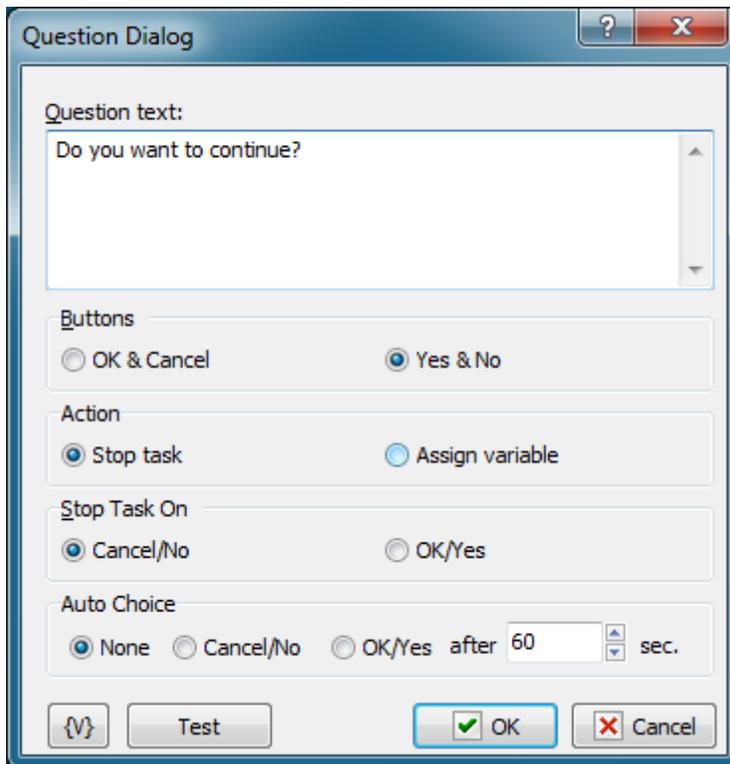
Click this button to ensure that the parameters for the current action work.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.2.2 Question Dialog

Displays a message box with specified text and two buttons: **OK&Cancel** or **Yes&No**. For example, after your system is backed up, you can use this action to display a message asking you if you want to open the log file.

**Question Text**

Enter the text for the message box.

**Buttons**

Click to set the style of buttons that appear on the message box.

- **OK & Cancel**
- **Yes & No**

**Action**

Select an option to specify an action when user clicks a button:

- **Stop task**
- **Assign variable**

**Stop Task On**

Select an option to specify when the task stops when the user clicks a button on the message box:

- **Cancel/No**
- **OK/Yes**

**Auto Choice**

Click an option to specify when the task continue or stops if a user does not select a button on the message box.

- **None**
- **Cancel/No**
- **OK/Yes**

**After \_\_\_\_ Sec.**

Enter the number of seconds to wait before implementing the button that you selected in the

**Auto Choice** area.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

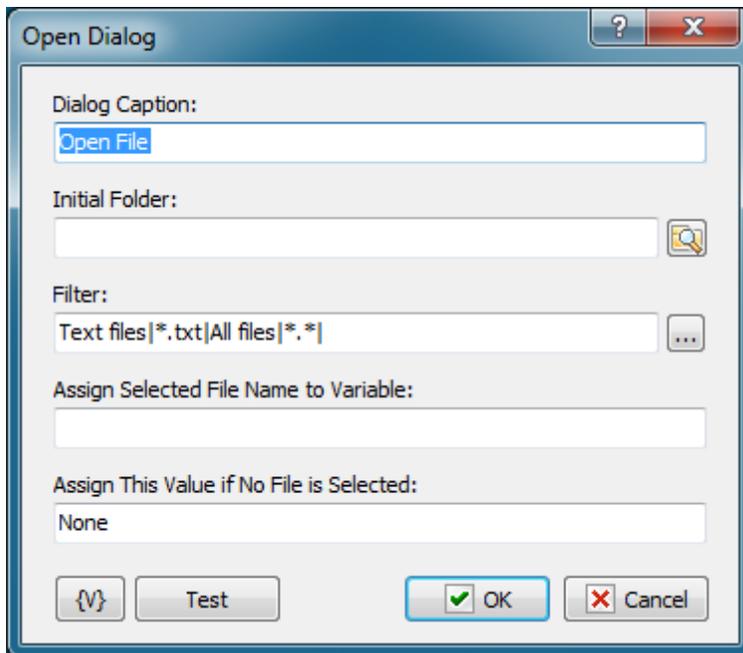
### Test

Click this button to ensure that the parameters for the current action work.

## 4.2.3 Open Dialog

Displays a dialog box that allows you to open file. You can set the title of the dialog box, the initial folder, and the type of files a user can open.

For example, you can use this action in a task in which you might want to select file for uploading to FTP server.



### Dialog Caption

Enter the caption of the dialog box. For example, you could enter a caption "Open Log Files" for the dialog box.

### Initial Folder

Set the folder that you want as the starting folder for the dialog box. For example, you could enter "My Documents" or "Daily Backups." Click the browse button to select a folder.

### Filter

Enter the criteria for including file types that you want the dialog box to open. These types

appear in options for opening files on the dialog box, such as **Files of Type**. Click the ... button to add types with the **Filter Editor** dialog box. You can click a field in this dialog box and add file type extensions and a name for the file type.

### Assign Selected File Name to Variable

Enter a variable name to store the selected file name to; for use in other actions of task.

### Assign This Value if No File Selected

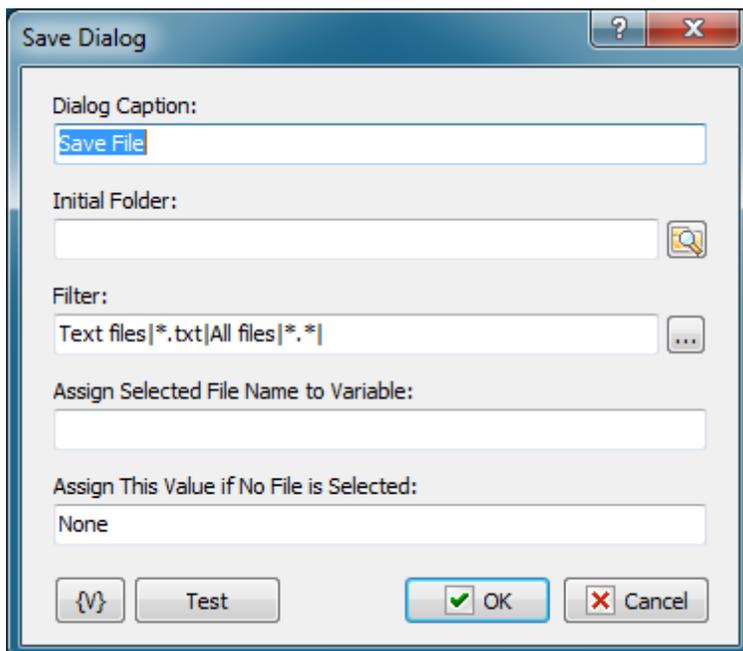
Assign a value to store if no file is selected, i.e. the **Cancel** button was clicked.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.2.4 Save Dialog

Opens a dialog box that allows users to save a file. You can specify the title of the dialog box, the start folder, and which types of files a user can save. For example, you might want to use this action that copies a file and you want to specify where to copy the file.



### Dialog Caption

Enter the caption of the dialog box. For example, you could enter a caption "Save Log Files" for the dialog box.

### Initial Folder

Set the folder that you want as the starting folder for the dialog box. For example, you could enter "My Documents" or "Daily Backups." Click the browse button to select a folder.

**Filter**

Enter the criteria for including file types that you want the dialog box to save. These types appear in options on the dialog box, such as **Files of Type**. Click the ... button to add types with the **Filter Editor** dialog box. You can click a field in this dialog box and add file type extensions and a name for the file type.

**Assign Selected File Name to Variable**

Enter a variable name to store the selected file name to; for use in other actions of task.

**Assign This Value if No File Selected**

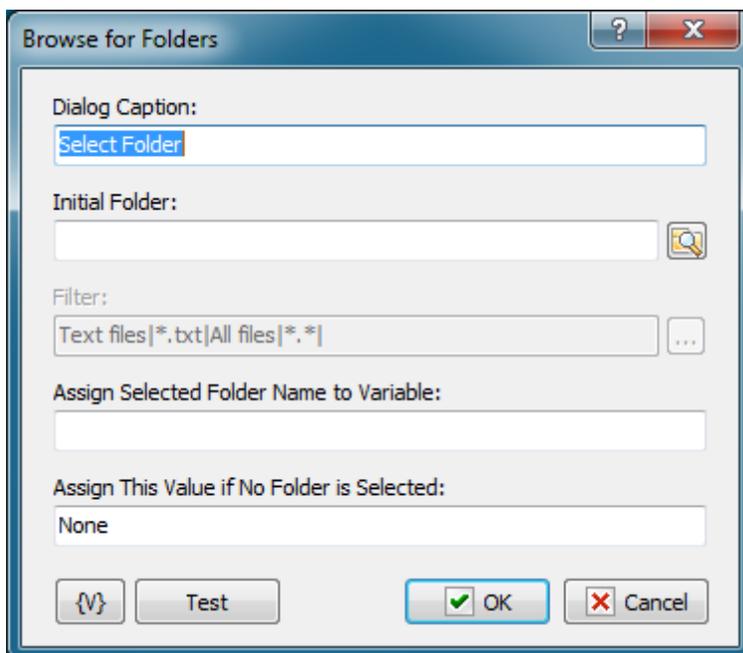
Assign a value to store if no file is selected, i.e. the **Cancel** button was clicked.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.2.5 Browse for Folders

Displays a dialog box that allows users to browse folders on a local or remote drives. You can set the title of the dialog box, the start folder, and what type of files a user can select. For example, if you create a task that downloads files from Internet, you can display this dialog box so you can select where to save the downloaded files.

**Dialog Caption**

Enter the caption of the dialog box. For example, you could enter a caption "Select Destination Folder" for the dialog box.

**Initial Folder**

Set the folder that you want as the starting folder for the dialog box. For example, you could enter "My Documents" or "Daily Backups." Click the browse button to select a folder.

**Filter**

This option is not available for this action.

**Assign Selected Folder Name to Variable**

Enter a value to store the selected file name in a variable expression for use in other actions or tasks.

**Assign This Value if No Folder Selected**

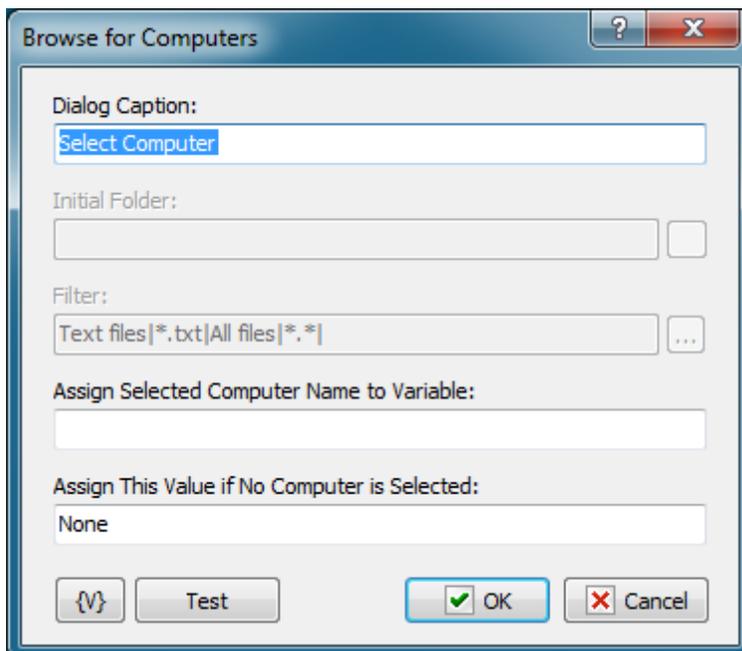
Assign a value to store if no folder is selected, i.e. the **Cancel** button was clicked.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.2.6 Browse for Computers

Displays a dialog box that allows a user to select computers on the local network. You can set the title of the dialog box and other settings. For example, in a task that sends a network message, you can select a computer to send a message to.

**Dialog Caption**

Enter the caption of the dialog box. For example, you could enter a caption "Browse Backup

Server" for the dialog box.

**Initial Folder**

This option is not available for the **Browse for Computers** action.

**Filter**

This option is not available for the **Browse for Computers** action.

**Assign Selected Computer Name to Variable**

Enter a variable name to store the selected computer name to; for use in other actions of task.

**Assign This Value if No Computer Selected**

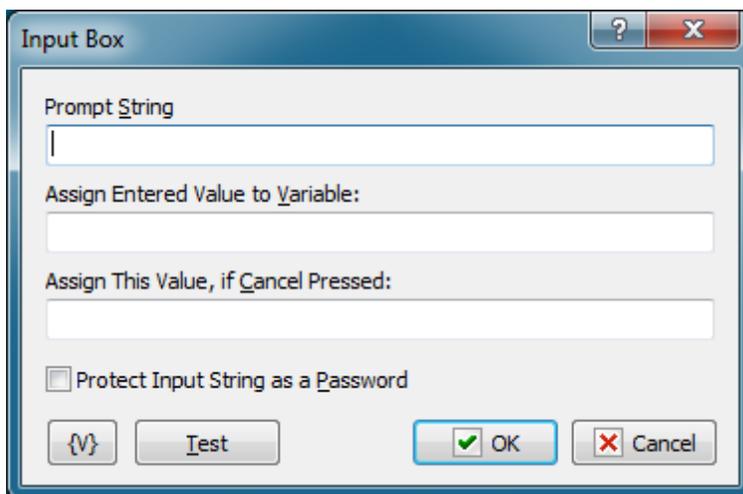
Assign a default value to store if no computer is selected.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.2.7 Input Box

Displays an input dialog box that lets the user enter a string and assigns the entered value to a specified variable.

**Prompt String**

Specify the text that prompts the user to enter input in the edit box.

**Assign Entered Value to Variable**

Enter a variable name to store the entered value to; for use in other actions of task.

**Assign this Value, if Cancel Pressed**

Assign a default value to store if Cancel is pressed.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

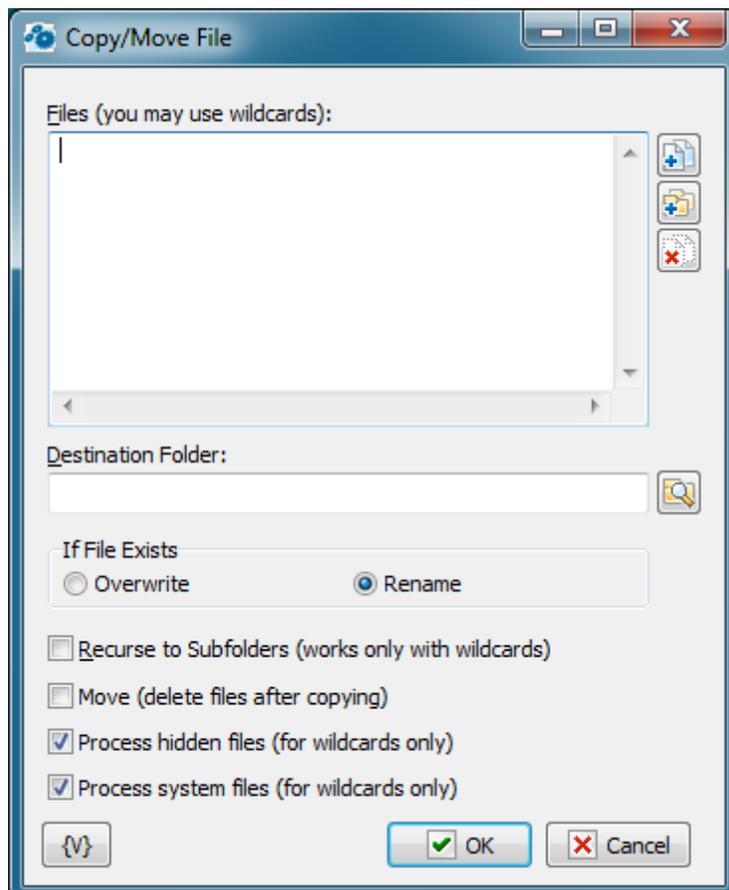
**Test**

Click this button to ensure that the parameters for the current action work.

## 4.3 Files and Folders

### 4.3.1 Copy/Move File

Copies or moves one or more files that you specify. You can set the source and the destination. Wildcards are allowed. For example, you can use this action to copy backup archives from your computer to another computer on the network.

**Files**

Enter the files you want to copy or move. Wildcards are allowed. Use the Plus buttons to add files to the list. To clear the list, click the X button.

**Tip**

You can also use wildcards in this option. Some examples of wildcards are the following:

- **C:\Text Files\\*.txt** - all the files with the .txt extension in the C:\Text Files folder.
- **C:\Doc Files\mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **{bakext}** - file mask(s) from the {bakext} variable.

**Destination Folder**

Enter the folder to which you want to move or copy the files. You can click the browse button to find folders.

**If File Exists**

Specify the action to take if a file already exists in the destination location.

- **Overwrite** - replaces the destination file with the file you are copying or moving.
- **Rename** - renames the source file when you copy or move it to the new location.

**Recurse to Subfolders**

Copies or moves all the files in the subfolders of the current directory.

**Move**

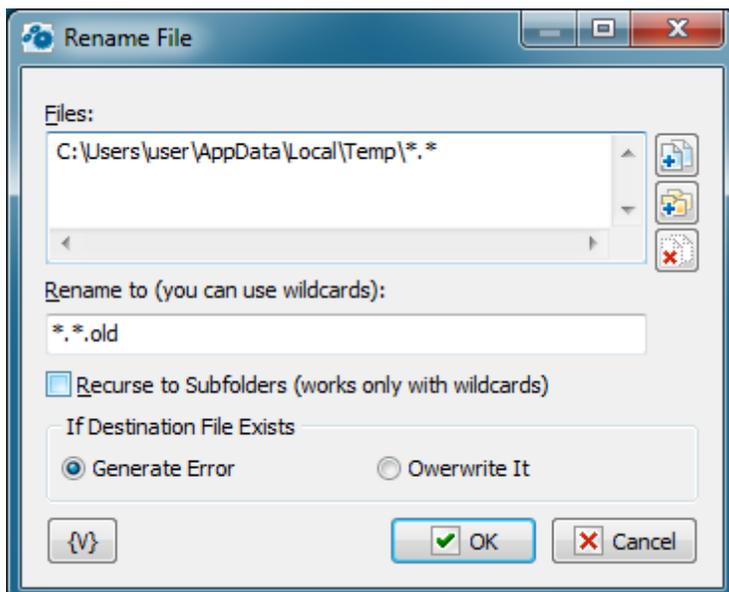
Deletes the original files after you copy or move them to a new location.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.3.2 Rename File

Renames one or more specified files. Wildcards are allowed. For example, you can copy the files to another folder and automatically rename them using this action.



**Files**

Enter the files you want to rename. Use the Plus buttons to add files or folders to the list. To clear the list, click the X button.

**Tip**

You can also use wildcards in this option. Some examples of wildcards are the following:

- **C:\Text Files\\*.txt** - all the files with the .txt extension in the C:\Text Files folder.
- **C:\Doc Files\mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **{bakext}** - file mask(s) from the {bakext} variable.

**Rename To**

Enter the name or partial name for the new filename. You can use a wildcard if you want to rename part of the filename.

**Recurse to Subfolders**

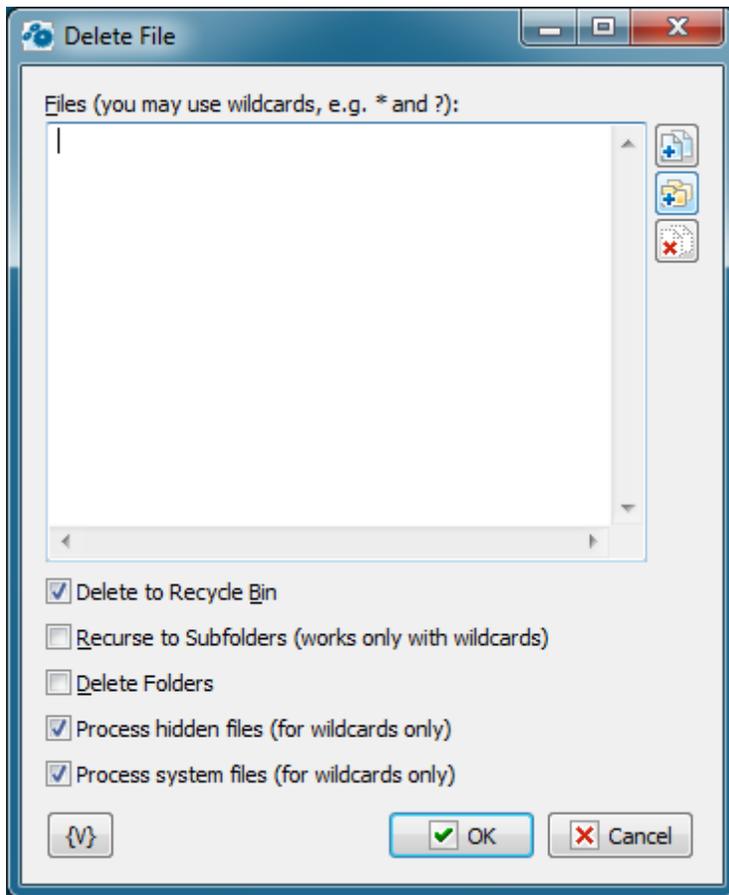
Renames the files in the subfolders of the current directory.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.3.3 Delete File

Deletes one or more files you specify. Wildcards are allowed. For example, you can use this action to automatically delete cookies from your Internet cache every day to protect your privacy and computer security.



### Files

Enter the files you want to delete. Use the Plus buttons to add files or folders to the list. To clear the list, click the X button.

### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

- **C:\Text Files\\*.txt** - all the files with the .txt extension in the C:\Text Files folder.
- **C:\Doc Files\mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **{bakext}** - file mask(s) from the {bakext} variable.

### Delete to Recycle Bin

Moves files to the Windows Recycle Bin. Use this flag to prevent possible data loss.

### Recurse to subfolders

Deletes the files in the subfolders of the current directory.

### Delete Folders

Removes the subfolders. This option works only with wildcards.

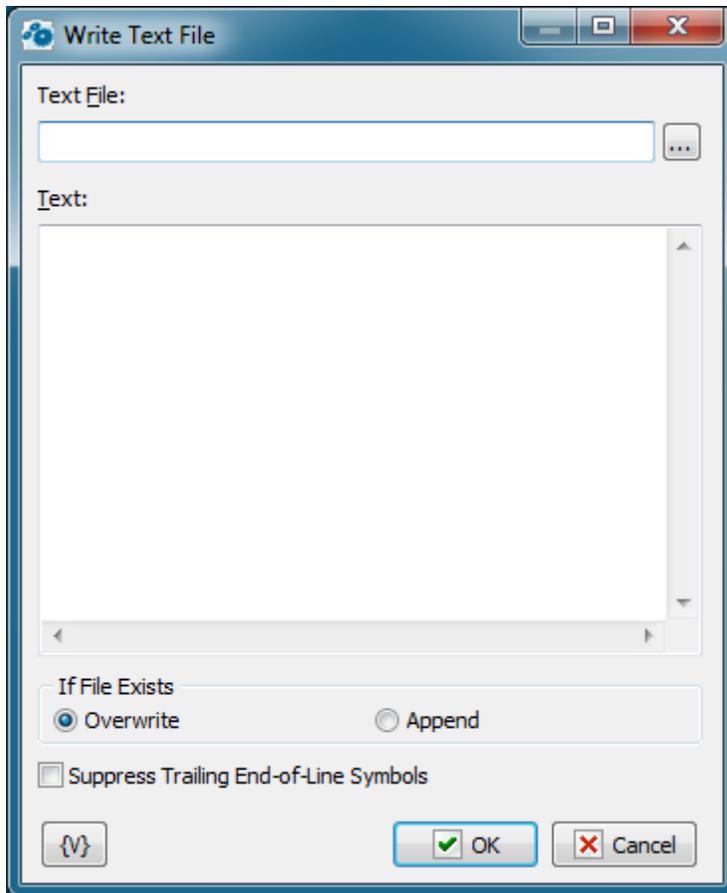
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then

click the {V} button.

#### 4.3.4 Write Text File

Creates a text file with the specified text. For example, you can write a text file listing which user is logged into a computer at a particular time.



##### **Text File**

Enter the name of the text file to save. You can click the browse button to select a file.

##### **Text**

Enter the text that you want saved in the file.

##### **If File Exists**

Specify the action to take if a file already exists in the destination location.

- **Overwrite** - replaces the destination file with the new file.
- **Append** - append specified text at the end of existing text file.

##### **Variable**

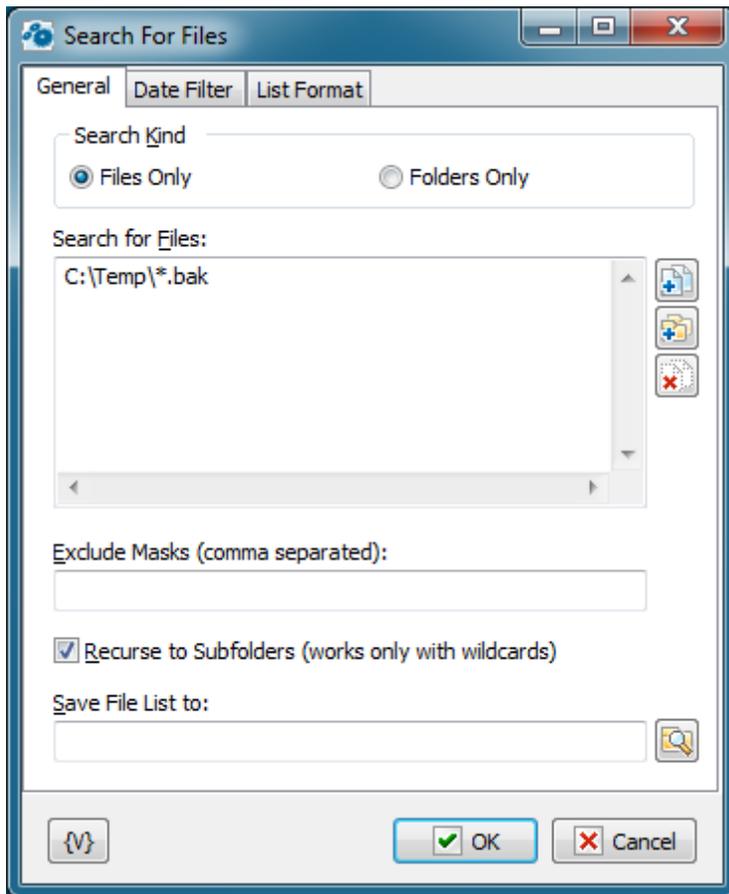
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then

click the {V} button.

### 4.3.5 Search for Files

Looks for files and creates a file list. This action is useful with pre-defined variables, such as % TextFile(file\_name)%. You can create a file list, such as a list of all .Png files, and then use this in a list of actions, such as copying, moving, or deleting files.

#### General tab



#### Search for Files

Enter the file path with file mask that you want to search for. Use the Plus buttons to add files or folders to the list. To clear the list, click the X button.

#### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

- **C:\Text Files\\*.txt** - all the files with the .txt extension in the C:\Text Files folder.
- **C:\Doc Files\mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **{bakext}** - file mask(s) from the {bakext} variable.

### Exclude Masks (Comma Separated)

Enter one or more file masks separated with commas that you do not want to search.

#### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

- **\*.txt** - all the files with the .txt extension.
- **mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **\*.tmp, \*.\$\$\$, \*.bak** - all the files with the .tmp, .\$\$\$ and .bak extensions.
- **{bakext}** - file mask(s) from the {bakext} variable.

### Recurse to Subfolders

Searches the files in the subfolders of the specified folder.

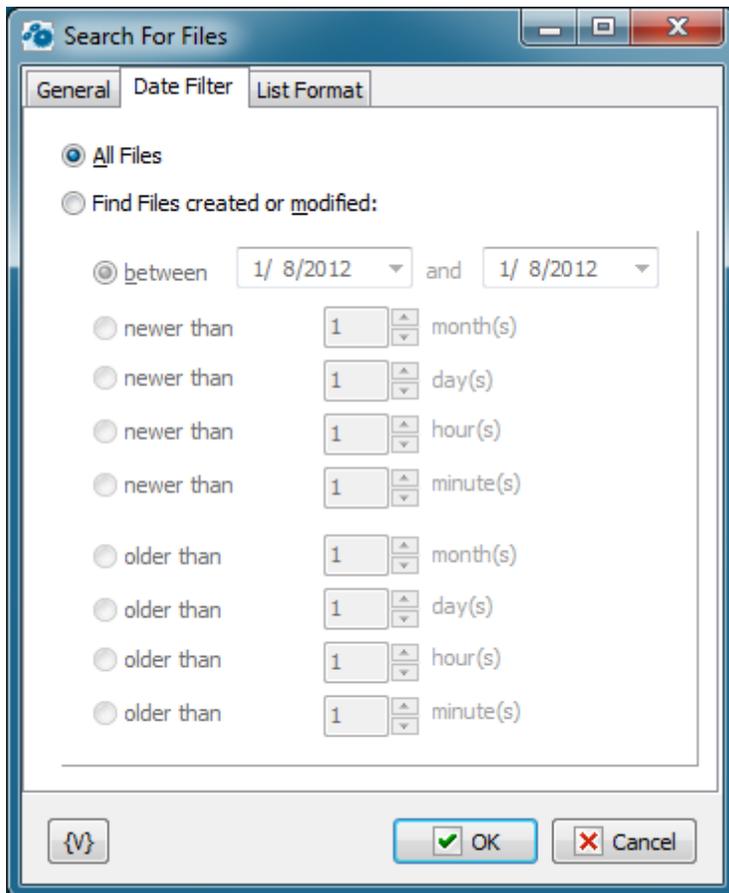
### Save File List To

Enter a path and name for a file that contains a list of the files that were searched. Click the **List Format** tab to set other options for this list.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### Date Filter tab



### All Files

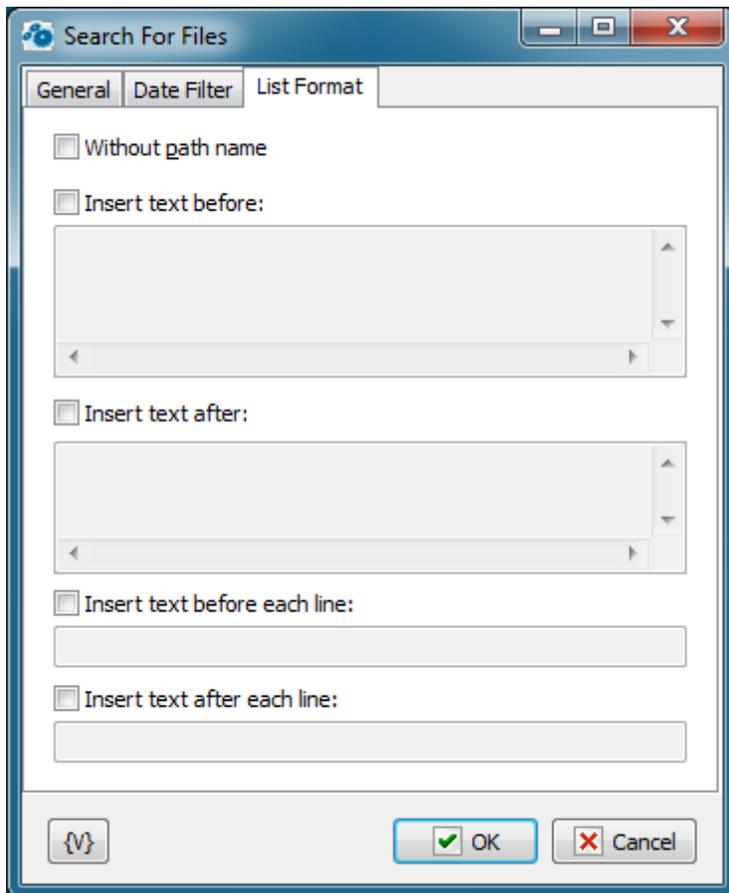
Select this option to search all files.

### Find Files Created or Modified

Click this option to search files according to one of the following criteria:

- **Between**
- **Newer Than \_\_\_ Months**
- **Newer Than \_\_\_ Days**
- **Older Than \_\_\_ Months**
- **Older Than \_\_\_ Days**

### List Format tab

**Without Path Name**

Lists files without the path names (i.e. file names only).

**Insert Text Before**

You can specify a text that will be inserted at the beginning of the list file.

**Insert Text After**

You can specify a text that will be inserted at the end of the list file.

**Insert Text Before Each Line**

You can specify a text that will be inserted at the beginning of each line.

**Insert Text After Each Line**

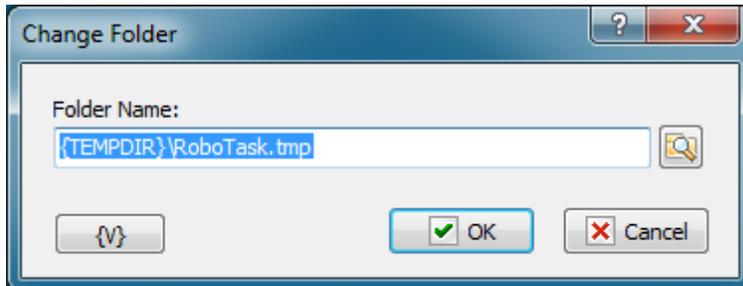
You can specify a text that will be inserted at the end of each line.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.3.6 Change Folder

Allows you to change the current folder. This is useful if you need to set the current folder during a series of actions or tasks. This action is useful with other file actions like copy, move.

**Folder Name**

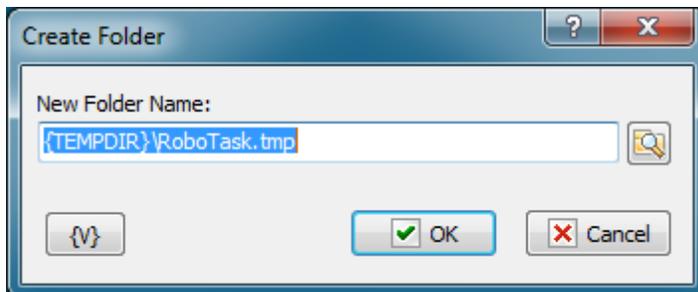
Enter the name of the destination folder. You can click the browse button to select a folder.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.3.7 Create Folder

Makes a new folder on a local or remote drive. For example, you could automatically create a new folder for your backups based on the current date.

**New Folder Name**

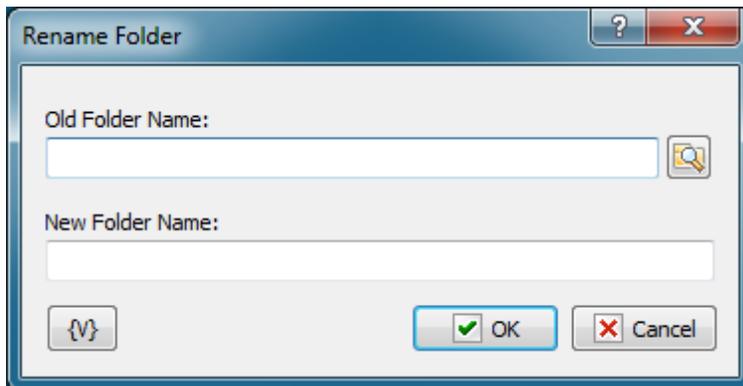
Enter the name for the new folder. You can click the browse button to select a folder.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.3.8 Rename Folder

Renames a existing folder. For example, if you have copied a folder with files, you can rename the original folder.



#### Old Folder Name

Enter the original name of the folder. You can click the browse button to select a folder.

#### New Folder Name

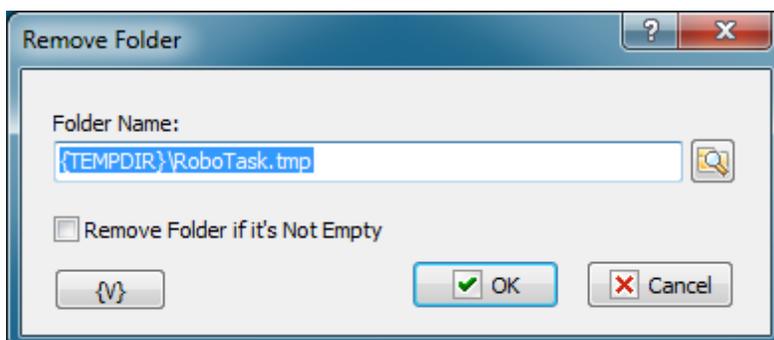
Enter the new name for the folder.

#### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.3.9 Remove Folder

Deletes a existing folder. For example, if you have archived your company financial reports every week for the past year, you can create a task that automatically deletes folders that are more than a year old.



#### Folder Name

Enter the name of the folder that will be removed.

### Remove Folder if It's Not Empty

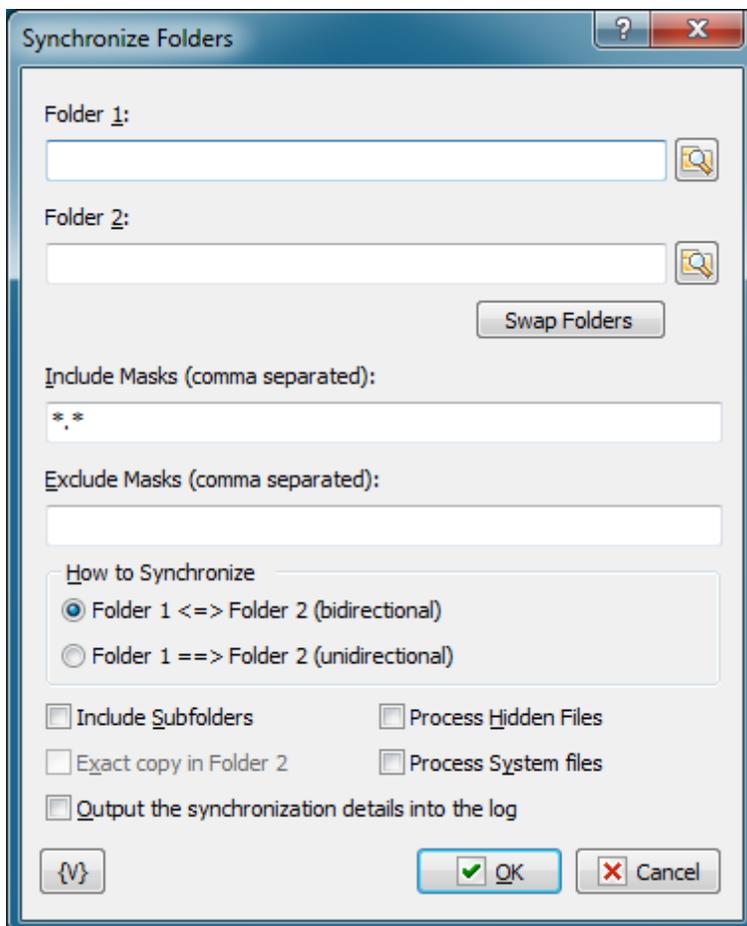
If this option is selected, the specified folder will be deleted even it is not empty (i.e. contains files and subfolders), otherwise the specified folder will be deleted only if it is not empty (i.e. does not contain files and subfolders).

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.3.10 Synchronize Folders

Synchronizes the two folders that you specify. The synchronization method may be bi-directional or uni-directional. For example, if you travel a lot, you might want to create a task that synchronizes data in a folder on your laptop computer and a folder on your desktop computer.



### Folder 1

Enter the first folder that you want to synchronize. You can click the browse button to select a

folder.

### Folder 2

Enter the second folder that you want to synchronize.

### Swap Folders

Click to place the folder listed in **Folder 2** in the **Folder 1** box and the folder in **Folder 1** in the **Folder 2** box.

### Include Masks (Comma Separated)

Enter one or more masks separated with commas to specify which files to synchronize, such as \*.exe, \*.cnt, \*.hlp. The default wildcard mask "\*" means that all folders and files are synchronized. You can specify one or more masks separated by commas.

#### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

- **\*.txt** - all the files with the .txt extension.
- **mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **\*.tmp, \*.\$\$\$, \*.bak** - all the files with the .tmp, .\$\$\$ and .bak extensions.
- **{bakext}** - file mask(s) from the {bakext} variable.

### Exclude Masks

Enter one or more masks separated with commas to specify which files to skip in the synchronization. You can specify one or more masks separated by commas; for example, \*.bak, \*.old, \*.\$\$\$.

### How to Synchronize

Select one of the following options:

- **Folder 1 <=> Folder 2 (Bi-directional)** - new or newer files that are available in Folder 1 are copied to Folder 2, and vice versa.
- **Folder 1 ==> Folder 2 (Uni-directional)** - only new or newer files that are available in the Folder 1 are copied to Folder 2; the contents of Folder 2 does not affect the contents of Folder 1.

### Include Subfolders

Click this check box to synchronize all files in the subfolders.

### Exact Copy in Folder 2

Available only with unidirectional synchronization when you select **Folder 1 ==> Folder 2**. This option ensures that files that are available in Folder 2, but not in Folder 1, are deleted.

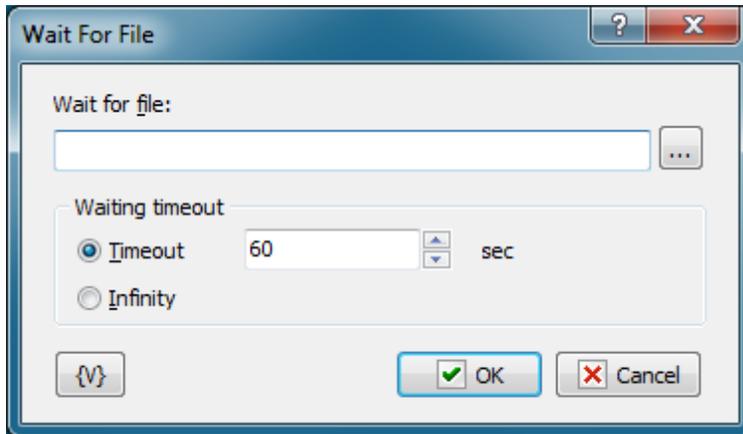
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.3.11 Wait for file

Pauses the task and waits until the selected file will be available. After the monitored file is

been available, the task is resumed. If the specified file is not opened in another process at the moment when the action is due, the task is not paused.



### **Wait for file**

You can either select the file name or type its name manually.

### **Waiting**

This parameter defines the timeout in seconds to wait until the process is finished. You can also specify an infinite timeout.

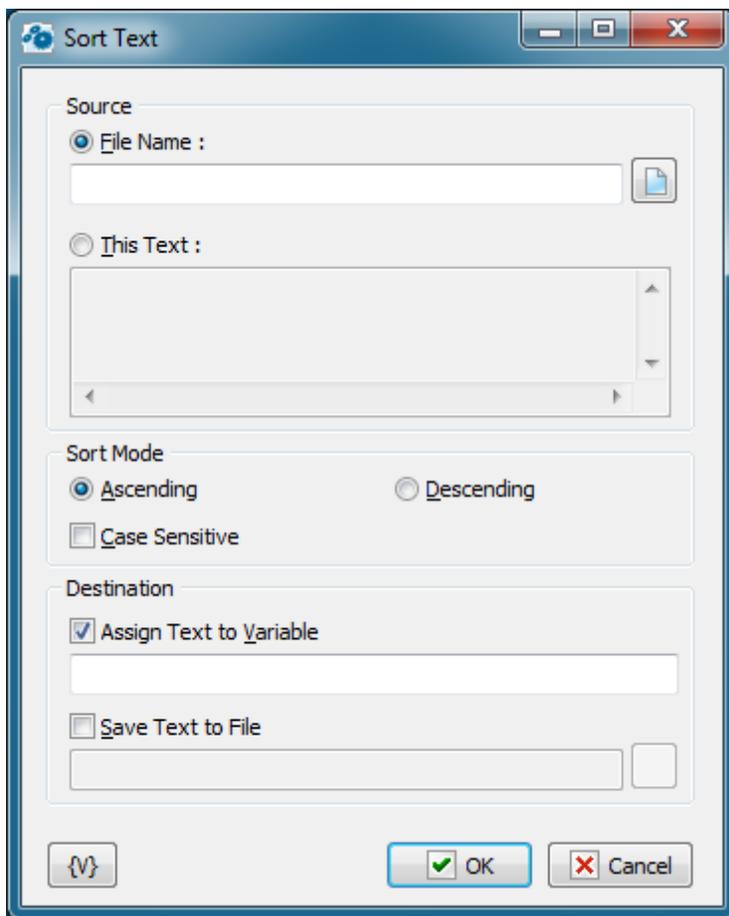
If the timeout is over, but the process is not finished, the action will generate the following error: Timeout is over.

### **Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## **4.3.12 SortText**

This action is used to sort strings in a text or in a text file in the ascending or descending order. It is possible to assign the sorted text to a variable for further processing (e.g. in "Text Loop") or save it to a file. You can specify the source text explicitly using variables or read it from an external file.



### Source

You can select the source for the source text in this group of parameters: specify an external file or specify the text explicitly.

- **File Name** – specify the full file name either manually or using the Open File dialog box by clicking the button to the right from the field. It is possible to use RoboTask variables.
- **This text** – if you want to specify the text explicitly, fill the text field manually. It is possible to use variables in the text. RoboTask will replace them with variable values during the action.

### Sort Mode

You can specify the string sorting mode in this group

Ascending or Descending

Case Sensitive

### Destination

Select at least one of the variants (or both at once) :

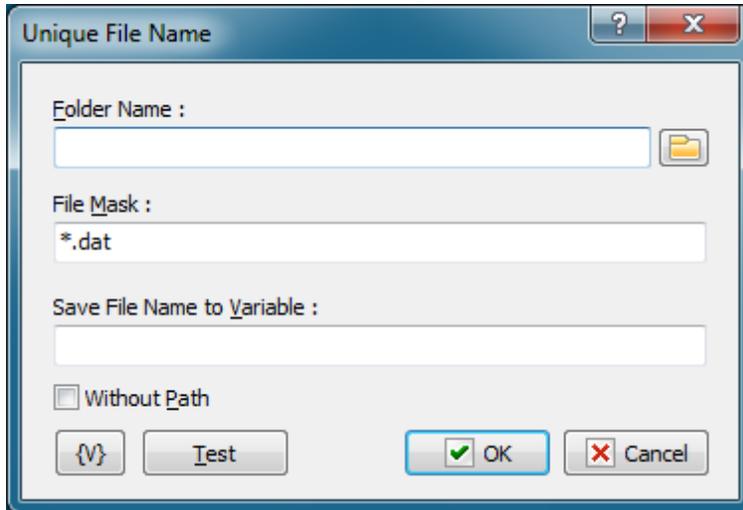
- **Assign text to Variable** – specify the name of the custom variable the sorted text will be assigned to. It is impossible to use variables.
- **Save text to File** – specify the full file name either manually or using the Open File dialog box. It is possible to use RoboTask variables.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.3.13 Unique File Name

The action generates unique file name in specified folder by the mask. Sometimes it's necessary to create the file with unique name where you need to store some temporary data. Next, the file name have to be saved into variable.



#### Folder Name

Specify the folder where you want to create the file. You can click the browse button to select a folder. The using of variables are allowed.

#### File Mask

It is the ordinary file mask with using characters "\*" and "?". Action replaces "\*" with 8 random characters and "?" with single random character.

You have to specify the mask of file name only. The path portion of the file name will be ignored. The using of variables are allowed.

For example:

```
tmp*.txt  
{Hours}{Minutes}-*.dat  
FileList-???.txt
```

#### Save File Name to Variable

Enter the variable name. The action saves new file name into this variable. The using of variables is unacceptable.

#### Without Path

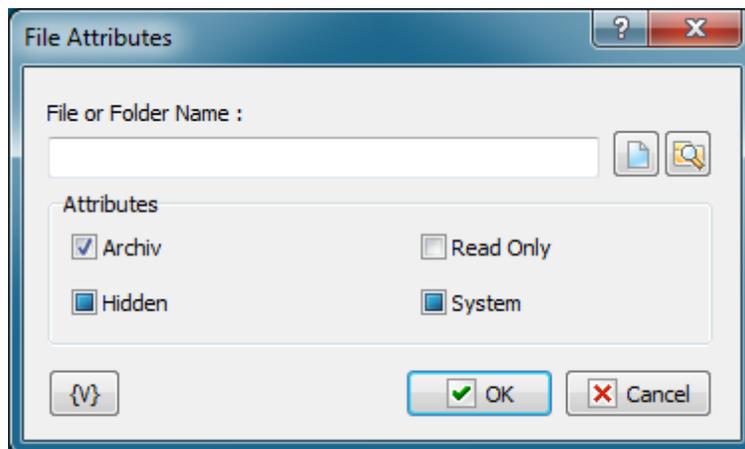
If you want to save the file name only without path, check this checkbox.

#### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.3.14 File Attributes

Sometimes you need to change file attributes in file operations. As a rule, the ReadOnly attribute inconveniences most of all. This action serves for changing of file attributes.



#### File or Folder name

Specify the name of a file or a folder. You can use buttons on the right of an edit field. A file mask is inadmissible; that is, the action can process only the specified file.

You can use variables.

#### Attributes

Specify the attributes, which you want to set or to remove. This action can control 4 attributes:

- Archive
- Hidden
- Read Only
- System

Please, notice that Checkboxes have three positions:

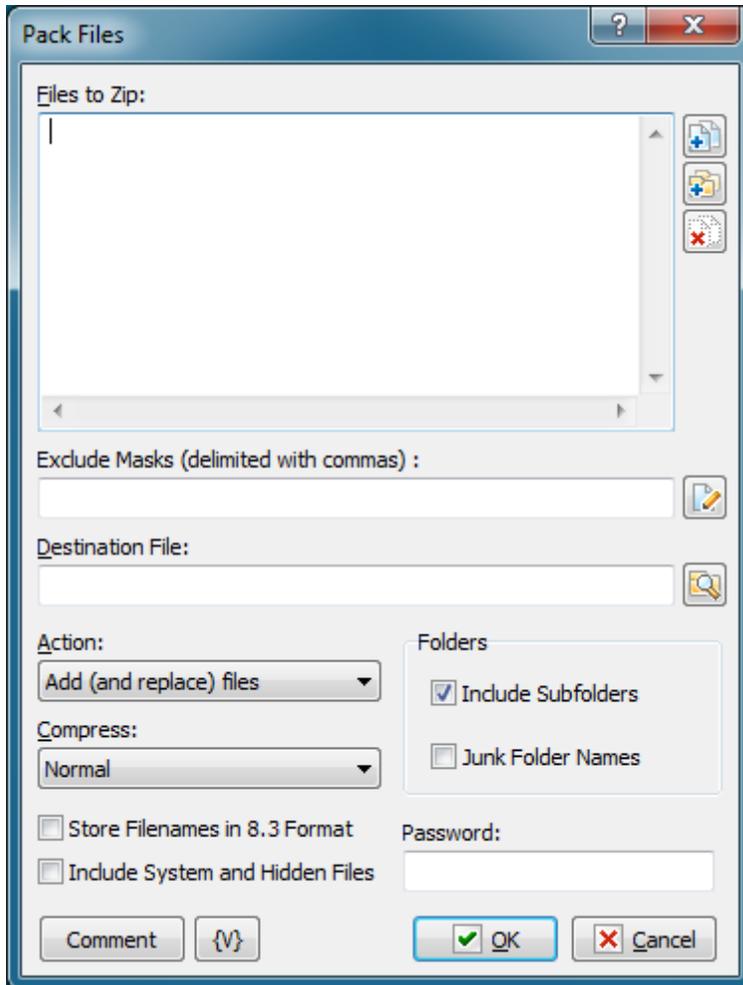
- **Switched on** – this means that you want to set an attribute
- **Switched off** – this means that you want to remove an attribute
- **Grayed, or intermediate state** – this means that you want to leave this attribute without changing.

By default, all the checkboxes are in Grayed state.

## 4.4 Zip

### 4.4.1 Pack Files

Creates and adds files to zip files. You can use this action to back up your files. For example, you can use this action to create a task that automatically zips weekly status reports files so that they can be copied to the central server for company summary reports every month.



#### Files to Zip

Enter the files you want to add to a zip file. Wildcards are allowed. Use the Plus buttons to add files or folders to the list. To clear the list, click the X button.

#### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

- **C:\Text Files\\*.txt** - all the files with the .txt extension in the C:\Text Files folder.
- **C:\Doc Files\mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **{bakext}** - file mask(s) from the {bakext} variable.

**Destination File**

Enter the filename for the resulting .zip archive.

**Action**

Select one of the following actions for creating or adding to the zip file:

- **Add and replace files** - add all files to the archive.
- **Freshen existing files** - add only changed files to the archive.
- **Move files** - add files to the archive and remove them from the local folder.
- **Update and add files** - add only changed or new files to the archive.

**Compress**

Select one of the items on the dropdown list to determine the level of compression, or size, of the archive file:

- **Maximum** - creates the smallest zip file possible, while requiring the most time to create the archive.
- **Normal** - creates a slightly compressed zip file taking the usual amount of time to create the zip file.
- **Fast** - produces slightly less compression than **Normal**.
- **Super Fast** - produces an archive with even less compression than **Fast** and **Normal**.
- **None** - produces an archive file that is not compressed. This level of compression is the fastest.

**Folders****Include Subfolders**

Click this check box to include the subfolders in the archive. Files are archived with the paths stored in the archive file.

**Junk Folder Names**

Does not include folder paths in the archive file. For example, if you add C:\documents\\*.doc, the .doc files are included into the root folder of the zip archive.

**Store Filenames in 8.3 Format**

Ensures that filenames are stored with the 8.3 name format, such as foo\_mon.txt, especially important for FAT file systems.

**Include System and Hidden Files**

Archives system and hidden files in the zip file.

**Password**

Enter a password that is required to unzip the archive, if you want to protect the file.

**Comment**

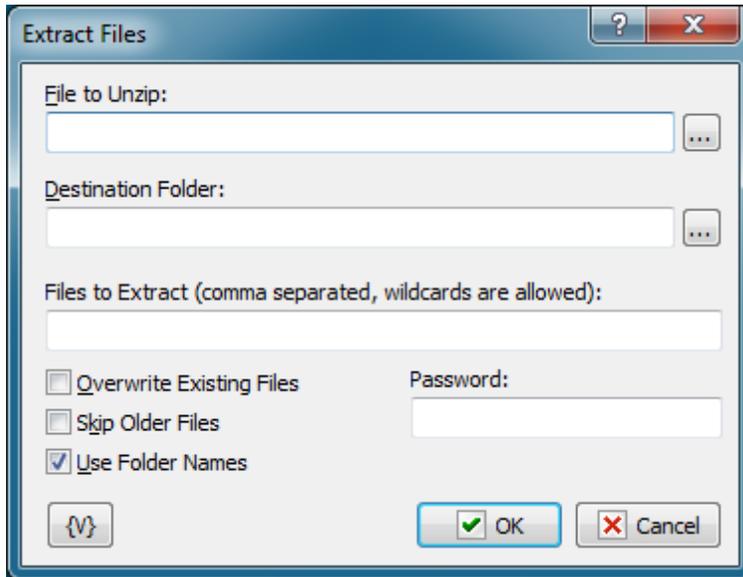
Enter a memo note for the zip file.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.4.2 Extract Files

Extracts files from an existing ZIP file to the folder that you specify. For example, you can create a task that automatically unzips files when you download .zip files to your hard drive.



### File to Unzip

Enter the name of the zip file to extract files from.

#### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

- **\*.txt** - all the files with the .txt extension.
- **mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **\*.tmp, \*.\$\$\$, \*.bak** - all the files with the .tmp, .\$\$\$ and .bak extensions.
- **{bakext}** - file mask(s) from the {bakext} variable.

### Destination Folder

Enter the destination folder where you want to extract files.

### Files to Extract

Enter the names of the files to unzip from the zip file, separated by commas. If left blank, all files are unzipped.

### Overwrite Existing Files

Click to allow the extracted files to overwrite existing files with the same name.

### Skip Older Files

Click to ensure the files in the zip archive are not extracted if they are older than the existing files on the local computer.

### Use Folder Names

Click this check box to extract files and the directory paths that are archived in the zip file. For

example, if a file is zipped with the path \temp\foo.txt, a \temp folder is created in the local directory and the foo.txt file is unzipped to that folder.

**Password**

Enter the password for the zip file, if the zip file is password-protected.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.5 Internet

### 4.5.1 Check for Email

Checks to see if email exists on a POP3 server that you specify. For example, you can create a task that checks e-mail in the morning and the evening automatically for you, thus minimizing distractions during your work day.

The screenshot shows a dialog box titled "Check For Email". It contains the following fields and options:

- POP3 server: [Empty text box]
- Port: [110] [Spin buttons]
- Use SSL: [ ] [checkbox]
- Username: [Empty text box]
- Password: [Empty text box]
- Action:  Stop task,  Assign variable
- Stop this Task if:  Mail is waiting,  Mail is NOT waiting
- Buttons: [OK] [Cancel]

**POP3 Server**

Enter the POP3 server for e-mail.

**Port**

Enter the port number. The default port number is 110.

**Use SSL**

Select this checkbox if you want to use SSL connection.

**Username**

Enter the user name for the e-mail account on the mail server.

**Password**

Enter the password for the e-mail account.

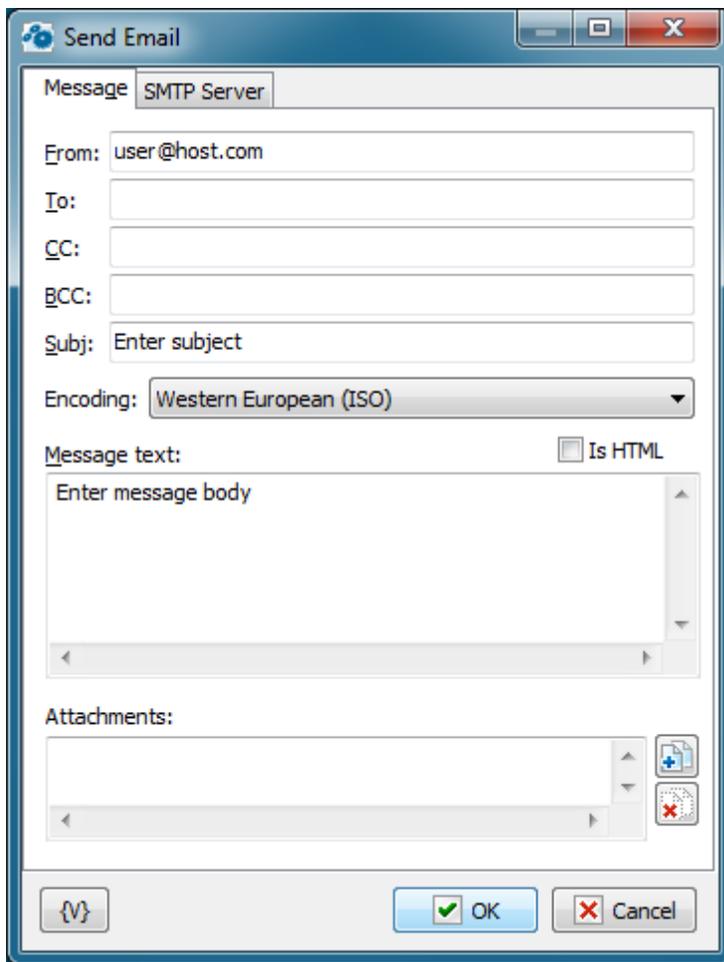
**Stop this Task if**

Enter whether you want the current task to do one of the following before e-mail is downloaded from the mail server:

- **Mail is Waiting** - stops the task if mail is on the mail server.
- **Mail is Not Waiting** - stops the task if mail is not on the mail server.

## 4.5.2 Send Email

Sends email through the specified SMTP server. Messages can have attachment files. For example, you can create a task that automatically e-mails monthly invoices to a specific customer outlining the sales charges for that month.

**Message tab**

The screenshot shows a Windows-style dialog box titled "Send Email" with a "Message" tab selected. The "SMTP Server" sub-tab is also visible. The dialog contains several input fields: "From:" with the text "user@host.com", "To:", "CC:", "BCC:", and "Subj:" with the placeholder "Enter subject". Below these is a dropdown menu for "Encoding:" set to "Western European (ISO)". A checkbox labeled "Is HTML" is present. A large text area for "Message text:" contains the placeholder "Enter message body". At the bottom, there is an "Attachments:" section with a list box and icons for adding and removing attachments. The dialog has standard "OK" and "Cancel" buttons at the bottom right, and a help icon at the bottom left.

**From**

---

Enter the e-mail address of the person sending the e-mail.

**To**

Enter the e-mail address to which you are sending the e-mail.

**CC**

Enter the address of any people to which you want to send a courtesy copy.

**Subj**

Enter the subject of the e-mail.

**Encoding**

The character set of the message. RoboTask doesn't convert specified text of the message, but you have to specify character set if the message contains non-latin characters.

**Message Text**

Enter the text that you want to send. If you enter .html tags, click the **Is HTML** check box.

**Tip**

- You can cut and paste .html code from Notepad into this window with the CTRL+V keys.

**Is HTML**

Click this check box if you have entered .html tags in the **Message Text** box.

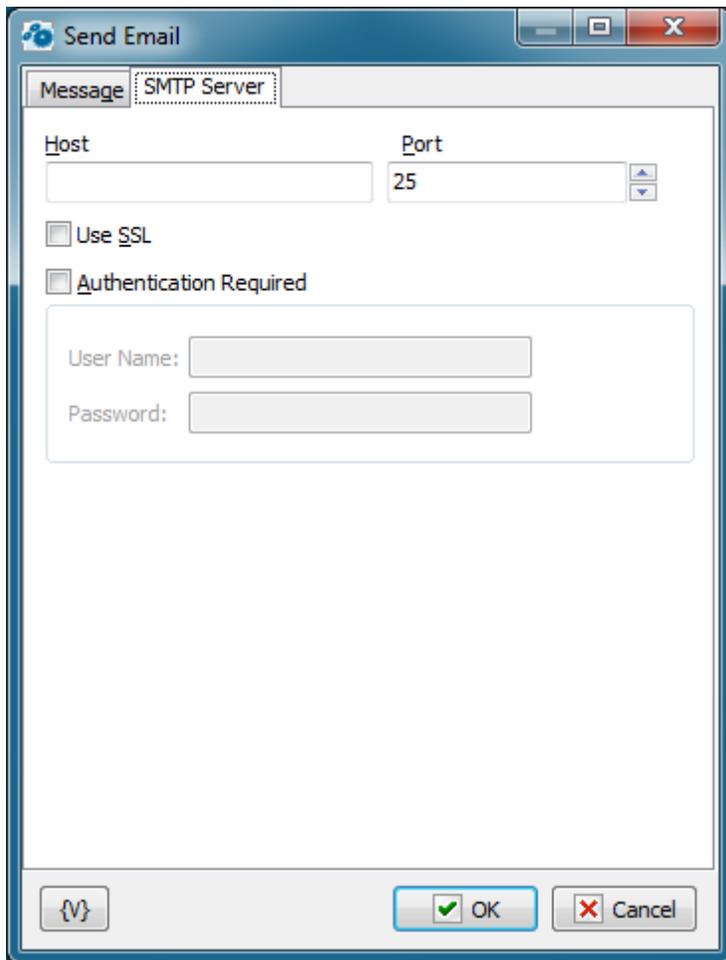
**Attachments**

Click the plus button to add attachments to the e-mail. You can add more than one attachment. Click the X button to remove attachments from the list.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

**SMTP Server tab**

**Host**

Enter the name of the host to which you want to send the e-mail using SMTP (simple mail transfer protocol).

**Port**

Enter the port number to use.

**Use SSL**

Click this checkbox if you want to use SSL connection.

**Authentication Required**

Click this check box if the remote mail server requires user authentication and enter the user name and password to log into the remote server.

**Variable**

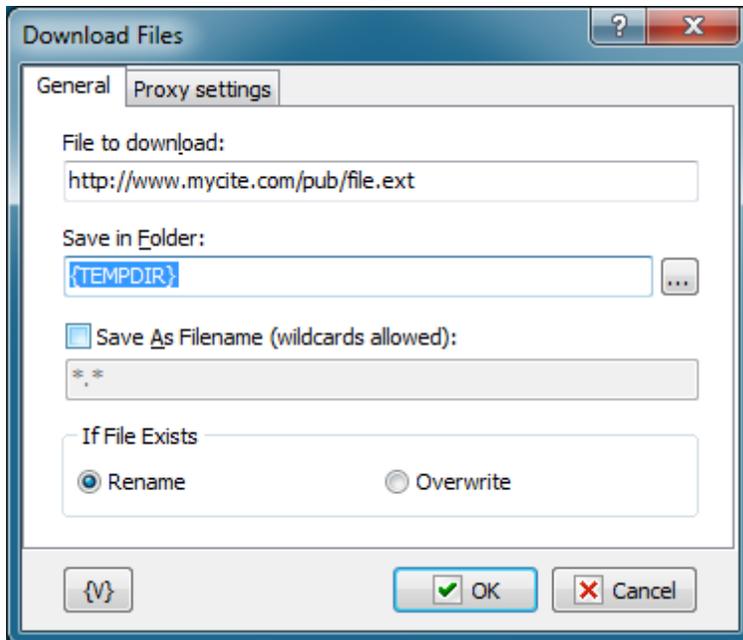
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.5.3 Download Files

Downloads one or more files through the Internet from specific Web or FTP sites. For example, if you are participating in a beta test program for new software, you can automatically download updates to the beta software on a daily basis so that you stay up-to-date.

#### Important

- Wildcards are not supported.



#### Files to Download

Enter the http, https or ftp URLs that you want to download.

#### Save in Folder

Enter the path of the folder in which you want to save the files.

#### If File Exists

Click an option to determine whether to overwrite existing files with the downloaded file or rename the downloaded file.

- **Rename**
- **Overwrite**

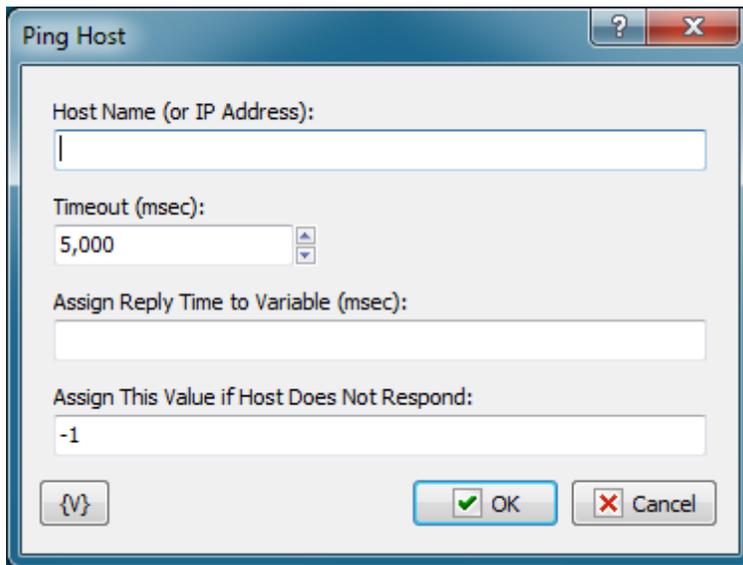
#### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.5.4 Ping Host

Allows you to send a ping message to a host. This action sends Internet Control Message Protocol (ICMP) Echo Request messages to verify IP-level connectivity to another TCP/IP computer. The action receives a receipt of corresponding Echo Reply messages, along with the round-trip times. **Ping Host** is based on the Ping command, the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.

For example, in a backup task, you can use this action to check to see if a host is up before you start a process to copy files to that host.



### Host Name

Enter the name or IP address of the host.

### Timeout (Msec)

Indicates the amount of time, in milliseconds, for the action to wait for the Echo Reply message that corresponds to a given Echo Request message to be received. If the Echo Reply message is not received within the time-out, the "Request timed out" error message is displayed.

### Assign Reply Time to Variable (Msec)

Enter a variable name for the reply time that you want to assign to.

### Assign This Value if Host Does Not Respond

Enter a value to assign if the host does not respond to the Ping message.

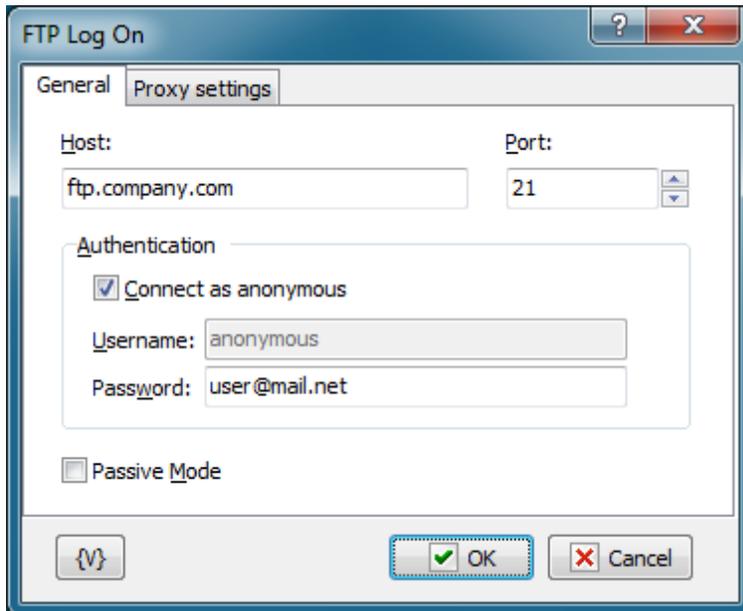
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.6 FTP

### 4.6.1 FTP Log On

Allows you to log on to an Ftp site, or remote server. For example, you can create a task to automatically Ftp updated daily reports from a co-worker's computer every day.

**Host**

Enter the name of the remote host that you want to log into.

**Port**

Enter the number of the port that you want to use. The default port is 21.

**Authentication****Connect as Anonymous**

Click this check box if the remote server allows anonymous logins.

**Username**

Enter the name of the log in account.

**Password**

Enter the password of the log in account.

**Variable**

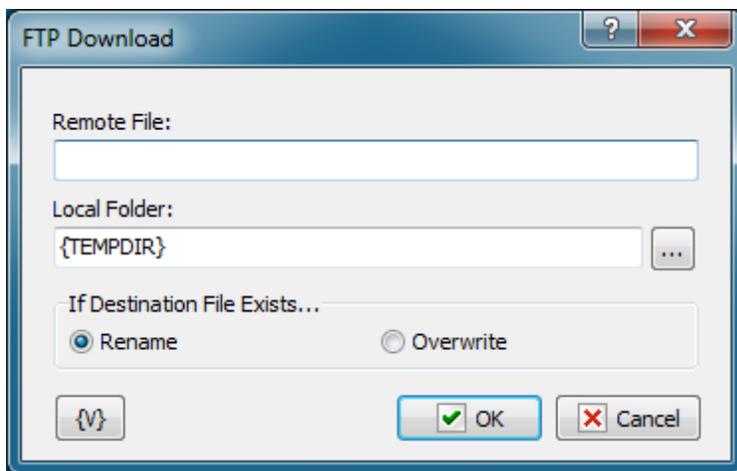
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.6.2 FTP Log Off

Allows you to log off an Ftp site, or remote server. No specific settings are associated with this action. For example, when you work from home, you can create a task that Ftps daily files that you have worked on to a remote server in your company's network to protect your data.

## 4.6.3 FTP Download

Copies remote files to the local computer using the current file transfer type. For example, if you are participating in a software beta program, you can download daily updates automatically to stay up-to-date.



### Tip

- Click the **{V}** button to insert a variable that holds the name of the local file or the remote folder.

### Remote File

Enter the name of the remote file that you want to copy to the local computer.

### Local Folder

Enter the path and name of the local directory to which you want to copy the remote file.

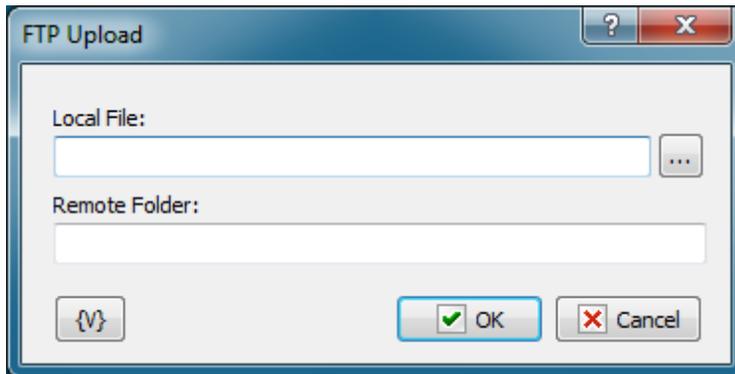
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the **{V}** button.

## 4.6.4 FTP Upload

Allows you to place files on an Ftp site. For example, if you must post daily updated financial reports to a web site, you can create a task to automatically upload the updated reports to the

company web site at the end of each day.

**Tip**

- Click the {V} button to insert a variable that holds the name of the local file or the remote folder.

**Local File**

Enter the name of the local file that you want to copy to the remote server.

**Remote Folder**

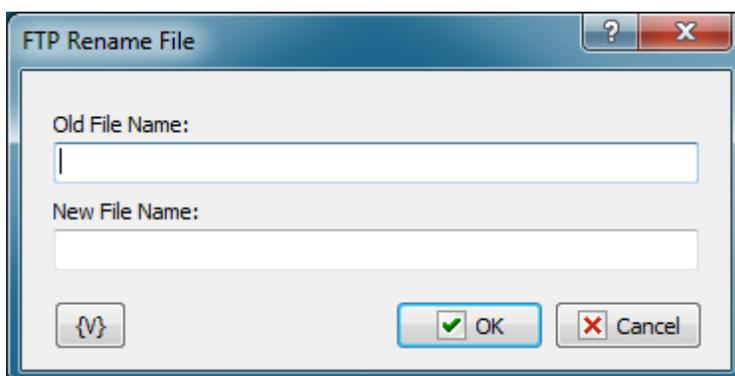
Enter the path and name of the directory on the remote server.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.6.5 FTP Rename File

Allows you to change the name of files on a remote server. For example, in an FTP session, you can use this action to rename files that you might accidentally overwrite on the remote server with updated files.

**Tip**

- Click the **{V}** button to insert a variable that holds the new file name or the old file name.

**Old File Name**

Enter the name of the file that you want to rename.

**New File Name**

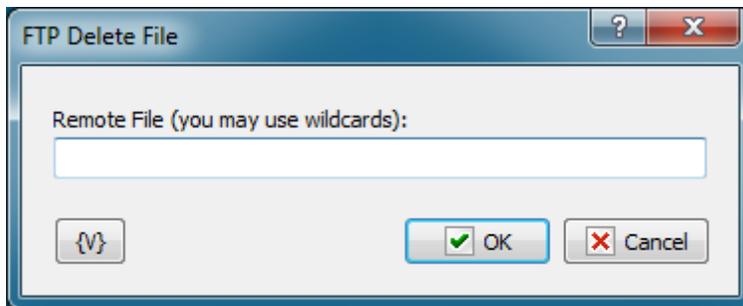
Enter the name that you want to give the file.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the **{V}** button.

#### 4.6.6 FTP Delete File

Allows you to remove files from a remote server. For example, you can use this action to remove old backup files on the remote server that are no longer needed.

**Remote File**

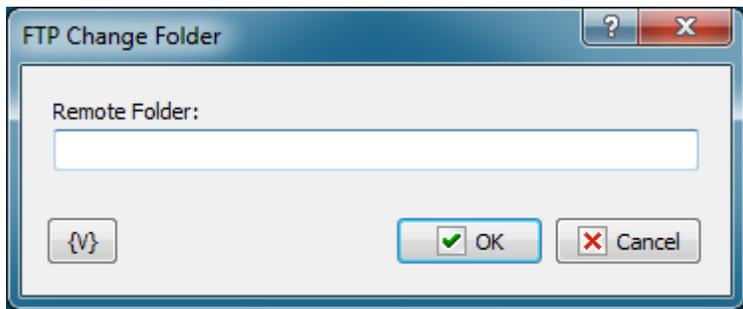
Enter the path and file name that you want to delete on the remote server. Click the **{V}** button to insert a variable that holds the path and name of the file.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the **{V}** button.

#### 4.6.7 FTP Change Folder

Allows you to switch the current folder on a remote server. For example, during an FTP task that backs up your files, you can use this action to copy files to different directories on the remote server.

**Remote Folder**

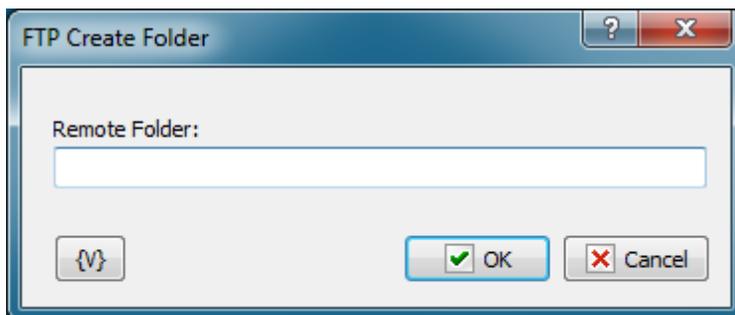
Enter the path or folder name on the remote server that you want to switch to. Click the **{V}** button to insert a variable that holds the name of the folder.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the **{V}** button.

#### 4.6.8 FTP Create Folder

Allows you to create a new folder on a remote server. For example, during an FTP task, you can use this action to create a new folder based on the current date for each set of daily backup files.

**Remote Folder**

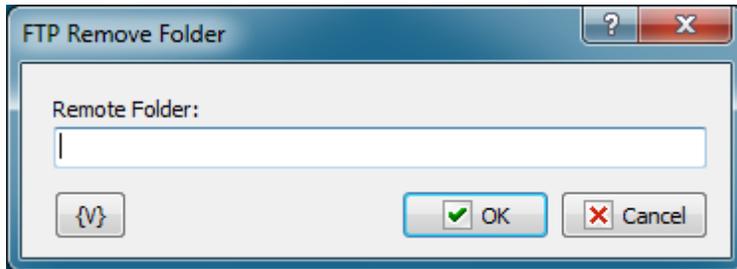
Enter the path or folder name on the remote server that you want to create. Click the **{V}** button to insert a variable that holds the name of the folder.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the **{V}** button.

### 4.6.9 FTP Remove Folder

Allows you to delete a folder on a remote server. For example, you can create a task to remove company reports that are more than six months old from the remote server.



#### Remote Folder

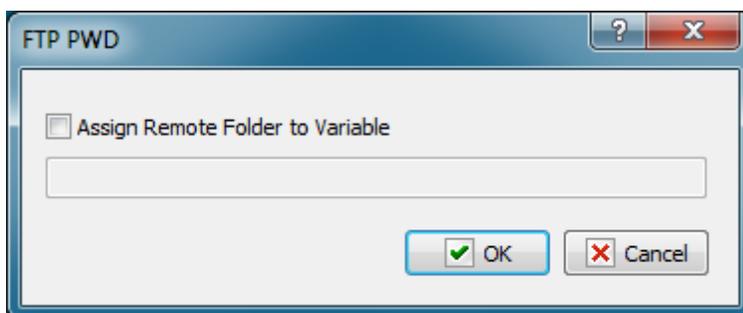
Enter the path or folder name on the remote server that you want to remove. Click the **{V}** button to insert a variable that holds the name of the folder.

#### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the **{V}** button.

### 4.6.10 FTP PWD

Allows you to print the current directory on the remote computer. This action saves the task log to the current directory on the FTP server. Also, you can assign the current directory to a custom variable for future use.

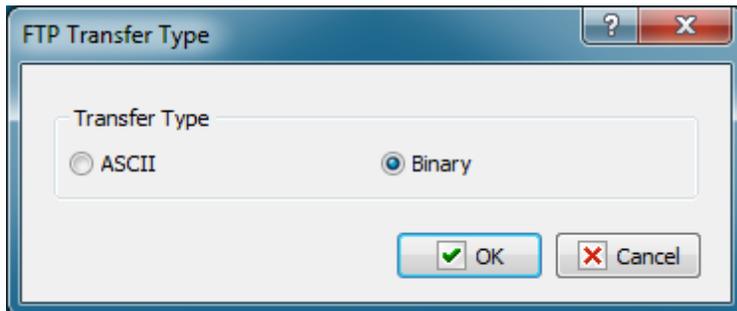


#### Assign Remote Folder to Variable

If you want to assign the name of the current FTP directory to a custom variable, select this checkbox and type the name of the variable in the field below.

### 4.6.11 FTP Transfer Type

Sets or displays the type of file you are transferring. For example, if you are transferring Microsoft Word reports to the company server every day, you must set the transfer type to binary. The default transfer type is **binary**.

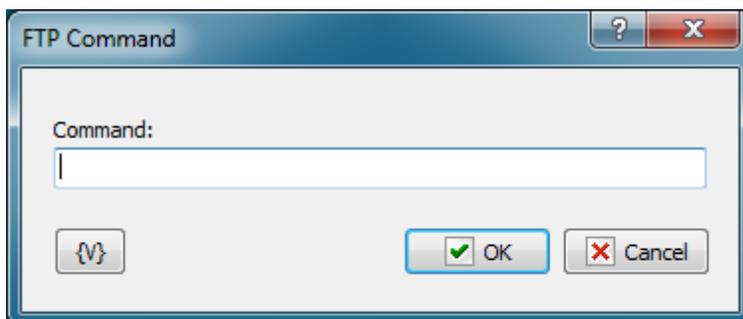


#### Transfer Type

Select whether you want to transfer the files as ASCII or binary during an FTP session. You should select **ASCII** when you want to transfer files that contain text strings, such as .html, .txt, or .xml. You should use **Binary** when transferring executable files, such as Office documents, .Exe files, and so forth. In binary mode, files are transferred in one-byte units.

### 4.6.12 FTP Command

Allows you to execute custom FTP commands.



#### Command

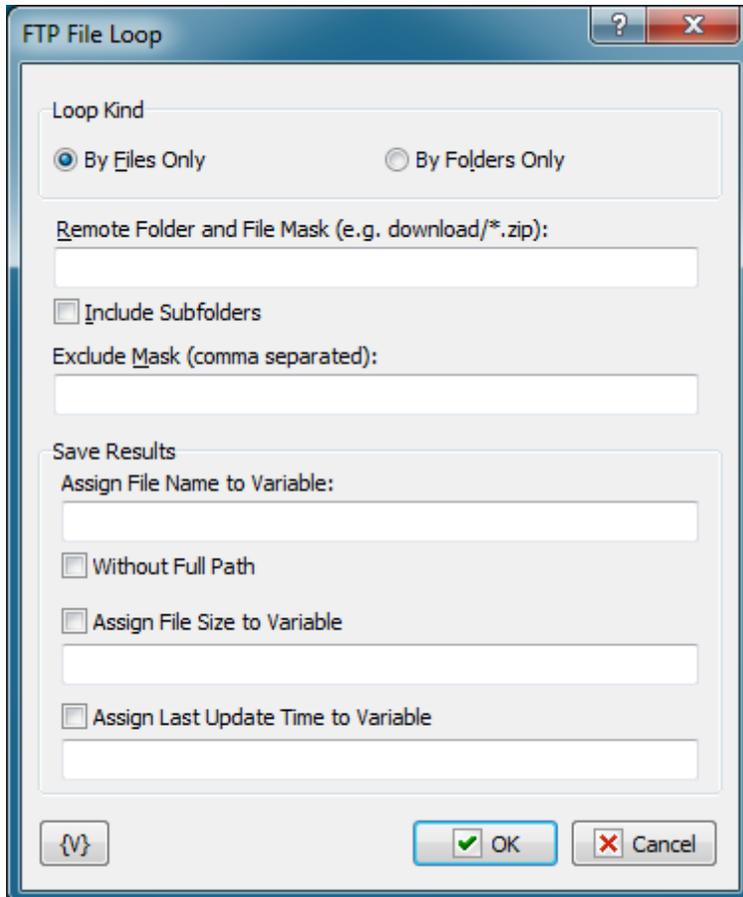
Enter the name of the command that you want to execute.

#### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.6.13 FTP File Loop

Executes a series of commands on files or folders on a remote server during an FTP session.



#### Loop Kind

Select the items you want to perform actions on: folders only, or files only.

#### Remote Folder

Enter the path or folder name on the remote server.

#### Save Results

Enter information about how you want to save the result of the file loop.

#### Assign File Name to Variable

Click this box and then click the **{V}** button to assign the file name to that variable. This allows you to use the file name string in other actions or tasks.

#### Assign File Size to Variable

Click this box and then click the **{V}** button to assign the file size to that variable. This allows you to use the file size data in other actions or tasks.

#### Assign Last Update Time to Variable

Click this box and then click the **{V}** button to assign the time the file was last updated to that variable. This allows you to use the last update time information in other tasks or actions.

**Variable**

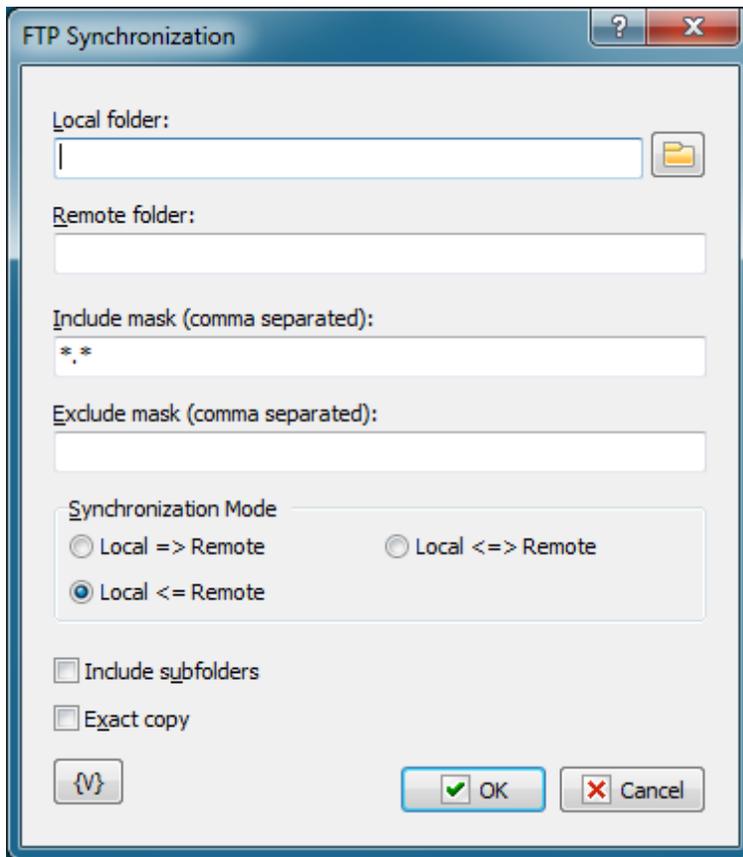
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

**4.6.14 FTP Synchronization**

Synchronizes files between a local folder and a remote folder on the FTP server. This high-level operation is very similar to "Synchronize folders", but there are several important differences:

- The time when a file is created on the FTP server is determined with the accuracy of one minute. So, if the difference in the versions of a file is several seconds, no file exchange will take place.
- Some FTP servers do not support time zones. In this case, the time when a file was created (last modified) corresponds to the local server time. Here is an example:  
You upload a file to the FTP server. Suppose your time zone is GMT+5 and the server is located in the GMT+8 time zone. After you upload the file, it will become 3 hours «younger». If your server supports time zones, it does not happen. The server makes all necessary corrections automatically.
- When you upload a file to the FTP server, the time when the file was last modified is not saved. The time when the file was created will correspond to the time when you finished uploading it to the server. So, the file you have just uploaded to the server becomes a bit «younger» than the initial file.

These differences present certain inconveniences while synchronizing folders with an FTP server, but still this operation will free you from creating a rather complicated synchronization task.



### Local Folder.

The local folder for synchronization. You can specify the path to the folder manually or you can use the folder selection button to the right from the input field. Also, you can use variables in the path to the folder.

### Remote Folder.

The folder on the FTP server for synchronization. You can use variables in the path to the folder.

### Include Mask.

The list of file masks that you need to synchronize. You should separate file masks with a comma. All files are specified by default ("\*.\*").

### Exclude Mask

The list of comma-separated file masks that should be excluded from synchronization. For example, you can specify \*.\* in the Include mask option and \*.txt in the Exclude mask option. It will mean that all files except \*.txt should be synchronized.

### Synchronization mode.

There are 3 options to choose from:

**Local => Remote** – newer files from the local folder will be copied to the FTP server

**Local <= Remote** – the reverse operation, i.e. copying new files from the server to the local folder.

**Local <=> Remote** – bidirectional synchronization. Downloading newer files from the FTP

server and uploading new files to the server.

**Include Subfolders.**

Select this checkbox if you want to include all subfolders in the specified folders to be included in the synchronization process.

**Exact copy.**

This option works only when the synchronization is one-directional. It does not make sense in case of bidirectional synchronization and so it is unavailable. If you want to get the exact copy of the initial folder, select this checkbox.

*Note that the "Exact Copy" mode first copies newer files from the source folder to the target folder and then **removes all files from the target folder that are absent in the source folder.***

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.7 SSL FTP

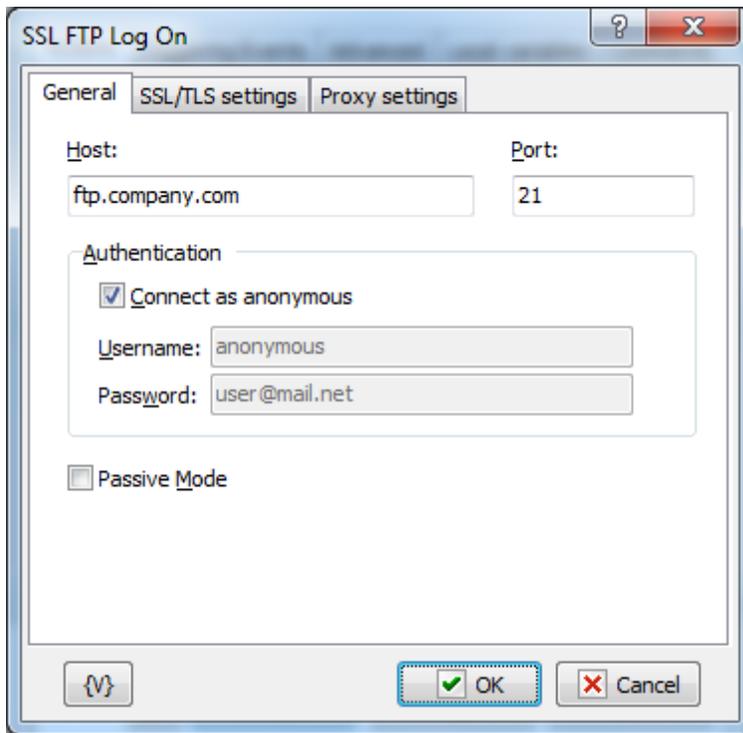
Encryption of traffic lets using of the FTP server in a public network safely (SSL - Secure Sockets Layer). At present most types of servers support the SSL FTP. Also there are freeware servers supporting SSL FTP (for example, FileZilla Server). Plugin SSL FTP allows using an ordinary FTP protocol with encryption. Functions of the plugin can be applied for the standard FTP protocol without encryption.

Thus the plugin SSL FTP can replace actions of FTP plugin.

### 4.7.1 SSL FTP Log On

Performs a connection to the FTP server and authentication of a user.

**General**

**Host**

Enter the name of the remote host that you want to log into.

**Port**

Enter the number of the port that you want to use. The default port is 21.

**Authentication:****Connect as Anonymous**

Click this check box if the remote server allows anonymous logins.

**Username**

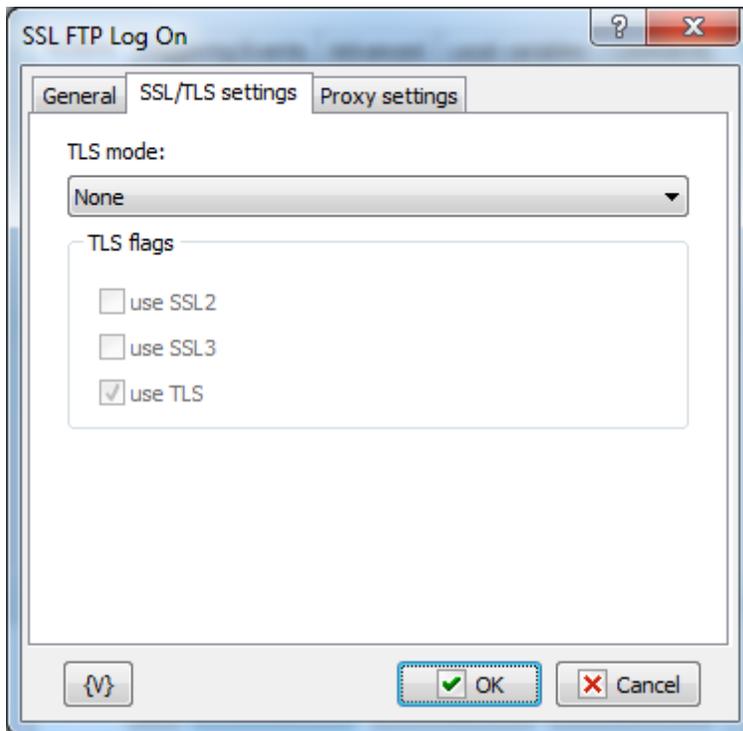
Enter the name of the log in account.

**Password**

Enter the password of the log in account.

**SSL/TLS Settings**

TLS – Transport Layer Security.



### TLS mode

Two separate methods were developed to invoke client security for use with FTP clients: Explicit or Implicit. The explicit method is a legacy compatible implementation where FTPS aware clients can invoke security with an FTPS aware server without breaking overall FTP functionality with non-FTPS aware clients. The implicit method requires that all clients of the FTPS server be aware that SSL is to be used on the session, and thus is incompatible with non-FTPS aware clients.

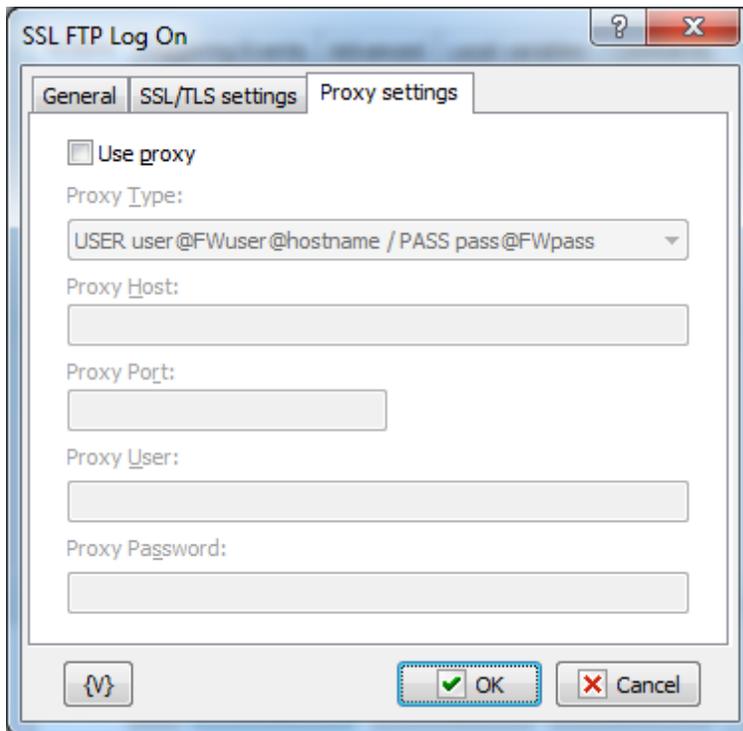
- **None** – plain FTP without encryption. SSL mode is OFF.
- **Implicit** – Negotiation is not allowed with implicit FTPS configurations. A client is immediately expected to challenge the FTPS server with a TLS/SSL ClientHello message. If such a message is not received by the FTPS server, the server should drop the connection.
- **Explicit** – In explicit mode (also known as FTPES), an FTPS client must "explicitly request" security from an FTPS server and then step-up to a mutually agreed encryption method. If a client does not request security, the FTPS server can either allow the client to continue insecure or refuse/limit the connection.
- **Auto** – it is auto-detection of a mode. Please note, that not all types of servers allow auto-detection of a mode.

### TLS Flags

Specifies modes of using SSL versions. It is usually enough only to turn on the flag Use TLS.

- Use SSL2
- Use SSL3
- Use TLS

### Proxy Settings



### Use proxy

If it is necessary to use PROXY, turn on this flag and fill in the parameters below. The parameters PROXY can be learned from the documentation for your PROXY or from your network administrator.

### Proxy Type

It is necessary to choose the type of PROXY.

- Send command USER user@hostname - Sends the command USER user@hostname to the firewall/proxy. This method is used by many programs like Wingate. The fields 'user name' and 'password' must be left empty.
- Send command SITE (with logon) - First send the user name and password to the proxy (if given), then the command 'SITE machine name'. The fields 'user name' and 'password' are usually needed for this type of proxy.
- Send command OPEN - First send the user name and password to the proxy (if given), then the command 'OPEN machine name'. The fields 'user name' and 'password' are sometimes not needed for this type of proxy.
- USER user@firewalluser@hostname / PASS pass@firewallpass - Use these two commands to logon.
- Transparent - First use the USER and PASS command with the firewall username and password, and then with the target host username and password.
- USER user@hostname firewalluser PASS pass ACCT firewallpass - Use these three commands to logon.

### Proxy Host

An IP address or a computer name in the network.

### Proxy Port

The number of a port.

### Proxy User

A username for the authorization in PROXY.

**ProxyPassword**

The password for the authorization in PROXY.

**Variable**

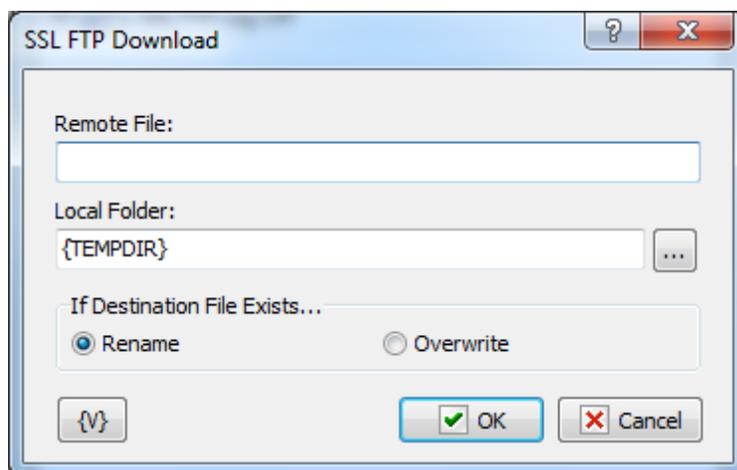
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.7.2 SSL FTP Log Off

Allows you to log off an Ftp site, or remote server. No specific settings are associated with this action. For example, when you work from home, you can create a task that Ftps daily files that you have worked on to a remote server in your company's network to protect your data.

### 4.7.3 SSL FTP Download

Copies remote files to the local computer using the current file transfer type. For example, if you are participating in a software beta program, you can download daily updates automatically to stay up-to-date.

**Tip**

- Click the {V} button to insert a variable that holds the name of the local file or the remote folder.

**Remote File**

Enter the name of the remote file that you want to copy to the local computer.

**Local Folder**

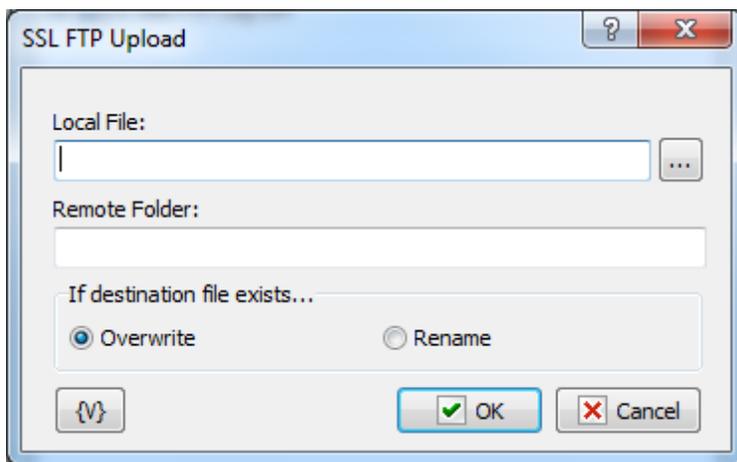
Enter the path and name of the local directory to which you want to copy the remote file.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.7.4 SSL FTP Upload

Allows you to place files on an Ftp site. For example, if you must post daily updated financial reports to a web site, you can create a task to automatically upload the updated reports to the company web site at the end of each day.

**Tip**

- Click the {V} button to insert a variable that holds the name of the local file or the remote folder.

**Local File**

Enter the name of the local file that you want to copy to the remote server.

**Remote Folder**

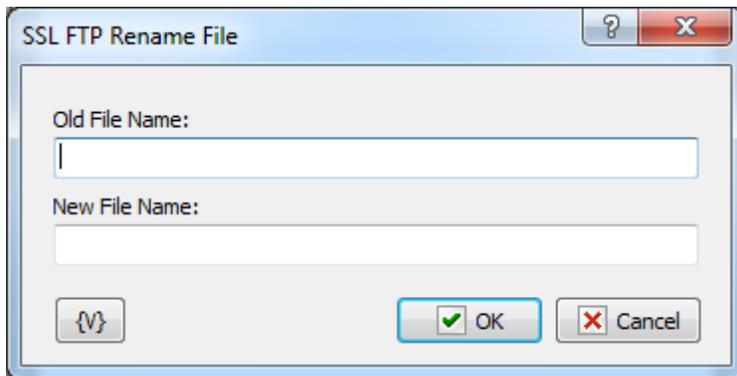
Enter the path and name of the directory on the remote server.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.7.5 SSL FTP Rename File

Allows you to change the name of files on a remote server. For example, in an FTP session, you can use this action to rename files that you might accidentally overwrite on the remote server with updated files.

**Tip**

- Click the **{V}** button to insert a variable that holds the new file name or the old file name.

**Old File Name**

Enter the name of the file that you want to rename.

**New File Name**

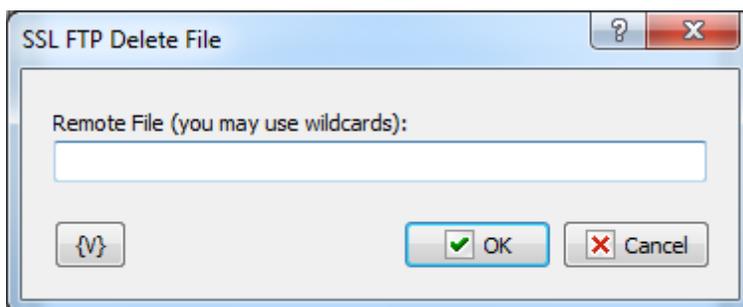
Enter the name that you want to give the file.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the **{V}** button.

#### 4.7.6 SSL FTP Delete File

Allows you to remove files from a remote server. For example, you can use this action to remove old backup files on the remote server that are no longer needed.

**Remote File**

Enter the path and file name that you want to delete on the remote server. Click the **{V}** button to insert a variable that holds the path and name of the file.

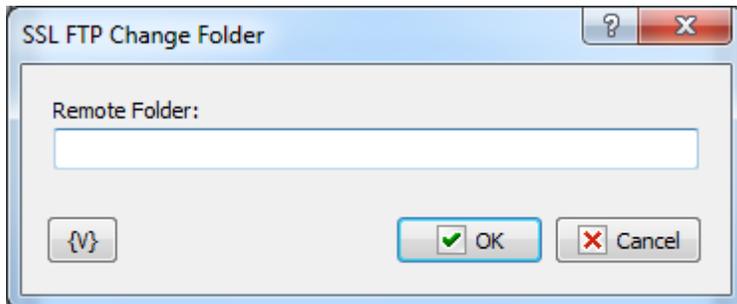
**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then

click the {V} button.

#### 4.7.7 SSL FTP Change folder

Allows you to switch the current folder on a remote server. For example, during an FTP task that backs up your files, you can use this action to copy files to different directories on the remote server.



##### **Remote Folder**

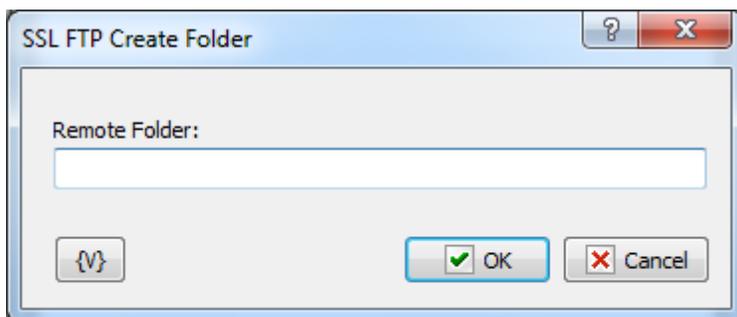
Enter the path or folder name on the remote server that you want to switch to. Click the {V} button to insert a variable that holds the name of the folder.

##### **Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.7.8 SSL FTP Create Folder

Allows you to create a new folder on a remote server. For example, during an FTP task, you can use this action to create a new folder based on the current date for each set of daily backup files.



##### **Remote Folder**

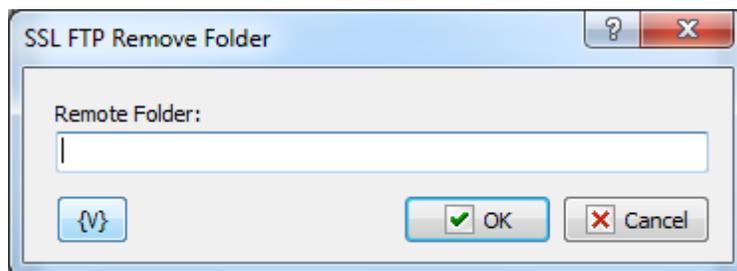
Enter the path or folder name on the remote server that you want to create. Click the {V} button to insert a variable that holds the name of the folder.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.7.9 SSL FTP Remove Folder

Allows you to delete a folder on a remote server. For example, you can create a task to remove company reports that are more than six months old from the remote server.

**Remote Folder**

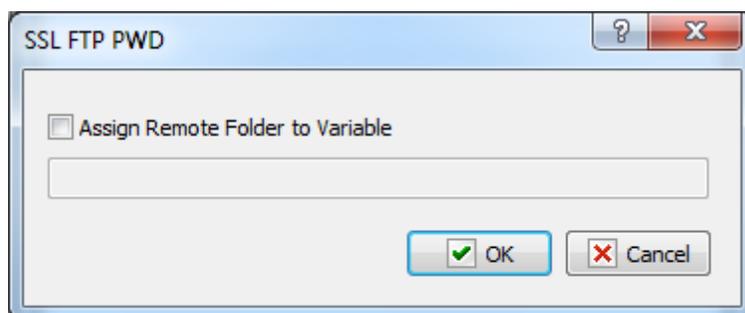
Enter the path or folder name on the remote server that you want to remove. Click the {V} button to insert a variable that holds the name of the folder.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.7.10 SSL FTP PWD

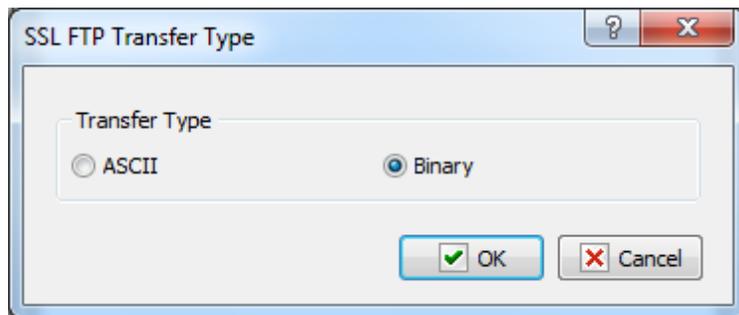
Allows you to print the current directory on the remote computer. This action saves the task log to the current directory on the FTP server. Also, you can assign the current directory to a custom variable for future use.

**Assign Remote Folder to Variable**

If you want to assign the name of the current FTP directory to a custom variable, select this checkbox and type the name of the variable in the field below.

#### 4.7.11 SSL FTP Transfer Type

Sets or displays the type of file you are transferring. For example, if you are transferring Microsoft Word reports to the company server every day, you must set the transfer type to binary. The default transfer type is **binary**.



##### **Transfer Type**

Select whether you want to transfer the files as ASCII or binary during an FTP session. You should select **ASCII** when you want to transfer files that contain text strings, such as .html, .txt, or .xml. You should use **Binary** when transferring executable files, such as Office documents, .Exe files, and so forth. In binary mode, files are transferred in one-byte units.

#### 4.7.12 SSL FTP File Loop

Executes a series of commands on files or folders on a remote server during an FTP session.

SSL FTP File Loop

Loop Kind

By Files Only       By Folders Only

Remote Folder and File Mask (e.g. download/\*.zip):

Include Subfolders

Exclude Mask (comma separated):

Save Results

Assign File Name to Variable:

Without Full Path

Assign File Size to Variable

Assign Last Update Time to Variable

{V}      OK      Cancel

**Loop Kind**

Select the items you want to perform actions on: folders only, or files only.

**Remote Folder**

Enter the path or folder name on the remote server.

**Save Results**

Enter information about how you want to save the result of the file loop.

**Assign File Name to Variable**

Click this box and then click the **{V}** button to assign the file name to that variable. This allows you to use the file name string in other actions or tasks.

**Assign File Size to Variable**

Click this box and then click the **{V}** button to assign the file size to that variable. This allows you to use the file size data in other actions or tasks.

**Assign Last Update Time to Variable**

Click this box and then click the **{V}** button to assign the time the file was last updated to that variable. This allows you to use the last update time information in other tasks or actions.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then

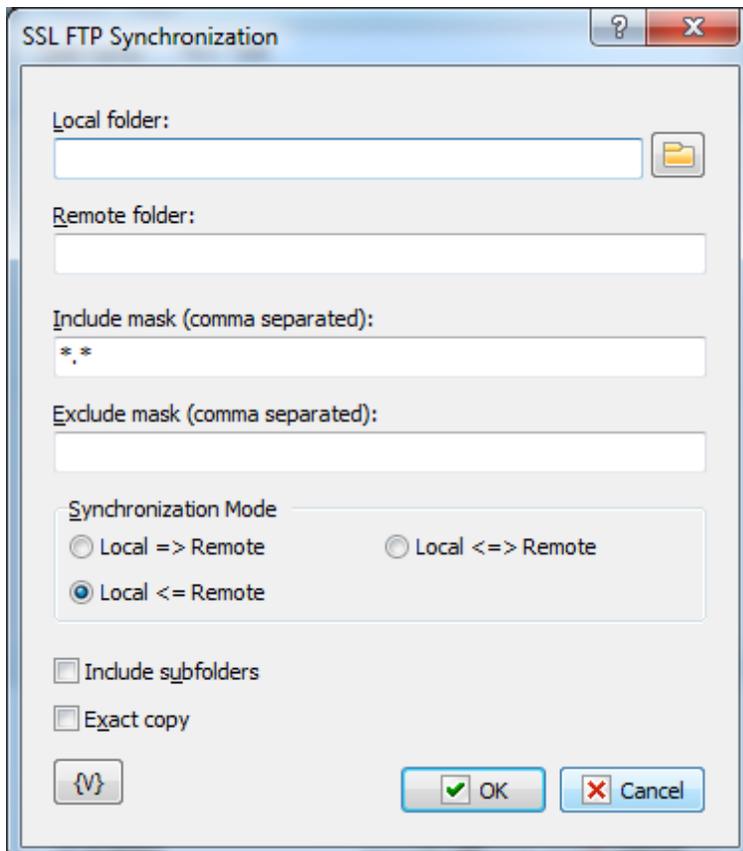
click the {V} button.

### 4.7.13 SSL FTP Synchronization

Synchronizes files between a local folder and a remote folder on the FTP server. This high-level operation is very similar to "Synchronize folders", but there are several important differences:

- The time when a file is created on the FTP server is determined with the accuracy of one minute. So, if the difference in the versions of a file is several seconds, no file exchange will take place.
- Some FTP servers do not support time zones. In this case, the time when a file was created (last modified) corresponds to the local server time. Here is an example:  
You upload a file to the FTP server. Suppose your time zone is GMT+5 and the server is located in the GMT+8 time zone. After you upload the file, it will become 3 hours «younger». If your server supports time zones, it does not happen. The server makes all necessary corrections automatically.
- When you upload a file to the FTP server, the time when the file was last modified is not saved. The time when the file was created will correspond to the time when you finished uploading it to the server. So, the file you have just uploaded to the server becomes a bit «younger» than the initial file.

These differences present certain inconveniences while synchronizing folders with an FTP server, but still this operation will free you from creating a rather complicated synchronization task.



**Local Folder.**

The local folder for synchronization. You can specify the path to the folder manually or you can use the folder selection button to the right from the input field. Also, you can use variables in the path to the folder.

**Remote Folder.**

The folder on the FTP server for synchronization. You can use variables in the path to the folder.

**Include Mask.**

The list of file masks that you need to synchronize. You should separate file masks with a comma. All files are specified by default ("\*. \*").

**Exclude Mask**

The list of comma-separated file masks that should be excluded from synchronization. For example, you can specify \*.\* in the Include mask option and \*.txt in the Exclude mask option. It will mean that all files except \*.txt should be synchronized.

**Synchronization mode.**

There are 3 options to choose from:

**Local => Remote** – newer files from the local folder will be copied to the FTP server

**Local <= Remote** – the reverse operation, i.e. copying new files from the server to the local folder.

**Local <=> Remote** – bidirectional synchronization. Downloading newer files from the FTP server and uploading new files to the server.

**Include Subfolders.**

Select this checkbox if you want to include all subfolders in the specified folders to be included in the synchronization process.

**Exact copy.**

This option works only when the synchronization is one-directional. It does not make sense in case of bidirectional synchronization and so it is unavailable. If you want to get the exact copy of the initial folder, select this checkbox.

*Note that the "Exact Copy" mode first copies newer files from the source folder to the target folder and then **removes all files from the target folder that are absent in the source folder.***

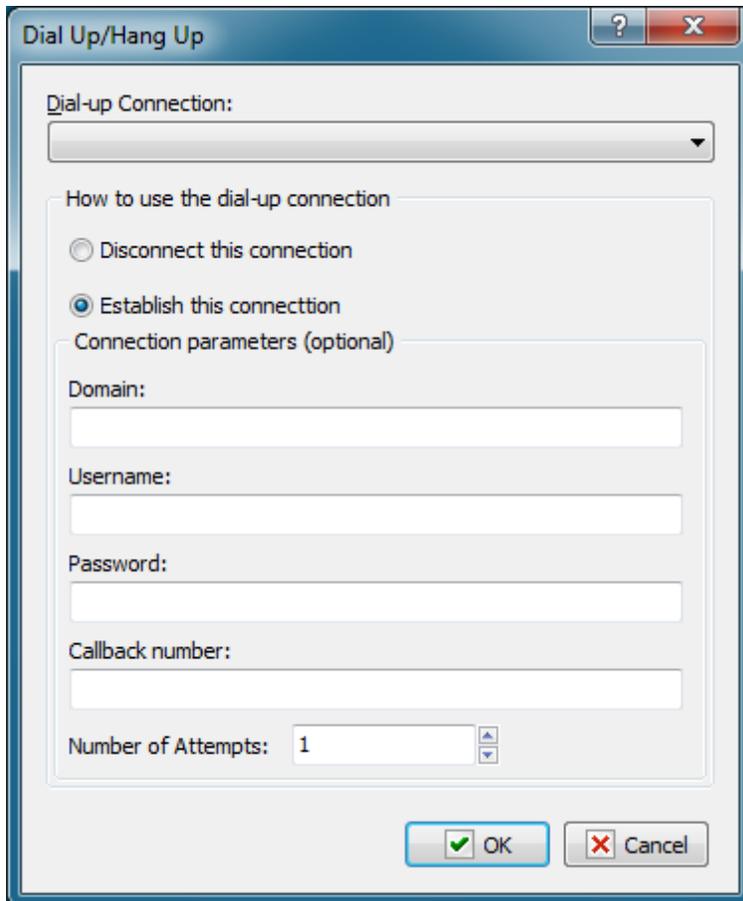
**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.8 Dial-Up

### 4.8.1 Dial Up/Hang Up

Establishes or disconnects a Windows RAS connection, also known as Dial Up Networking. Usually this feature is used to make a connection to the Internet and then complete a task and disconnect from the Internet. For example, you can create a task to automatically connect to the Internet every morning and check e-mail.



#### Dial-up Connection

Select the name of the dial-up connection that you want to connect or disconnect. You create this connection in the Windows **Control Panel**.

#### How to Use the Dial-up Connection

Select one of the following options to specify the action on the connection:

- **Disconnect this Connection** - breaks the connection to the Internet or remote network.
- **Establish this Connection** - connects the computer to the Internet or remote network.

#### Connection Parameters

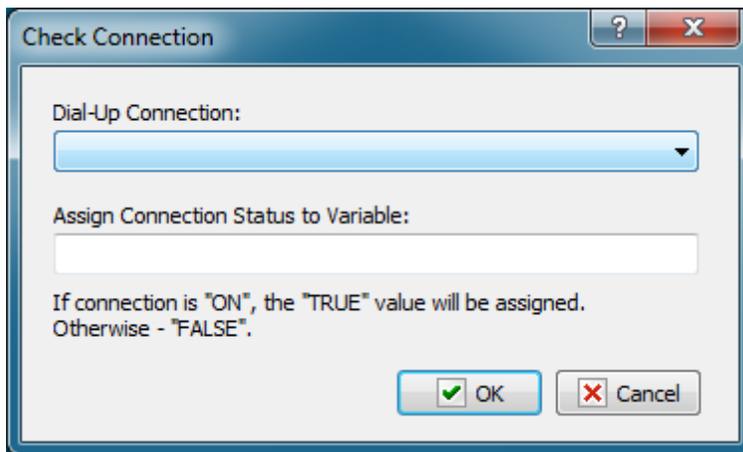
Enter the following optional parameters to establish a connection:

- **Domain**
- **Username**

- **Password**
- **Callback Number**
- **Number of Attempts**

## 4.8.2 Check connection

Checks the status of the selected dial-up connection and assigns the corresponding value to a user variable: TRUE if the connection is active and FALSE otherwise. This value can be later used in the "IF Then" action as a Boolean one.



### **Dial-Up connection**

Select the connection you need from the list.

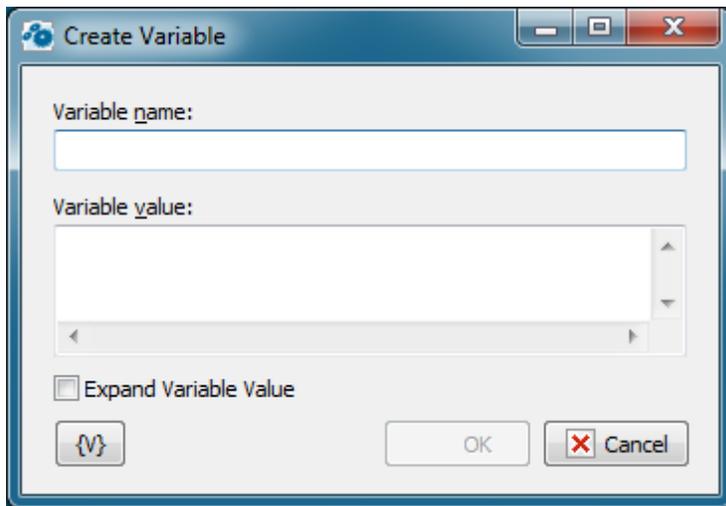
### **Assign Connection Status to Variable**

Specify the name of the variable the status of the connection should be assigned to.

## 4.9 Variables

### 4.9.1 Create Variable

Creates a new variable when you run a task. If you create a variable and then use it in several different tasks, you can change the variable once and update every task at the same time.



### Variable Name

Enter the name of the variable you want to set the value of.

### Variable Value

Enter the value that you want to set for the variable.

### Expand variable value.

If this parameter is on, RoboTask computes the value of an expression (that is, expands variables in the expression) before assigning a value to the specified variable.

Otherwise, the value is assigned without expanding variables. In this case the value is computed while further using the variable. Meanwhile, the variable value and the value of all variables are expanded recursively until the very end.

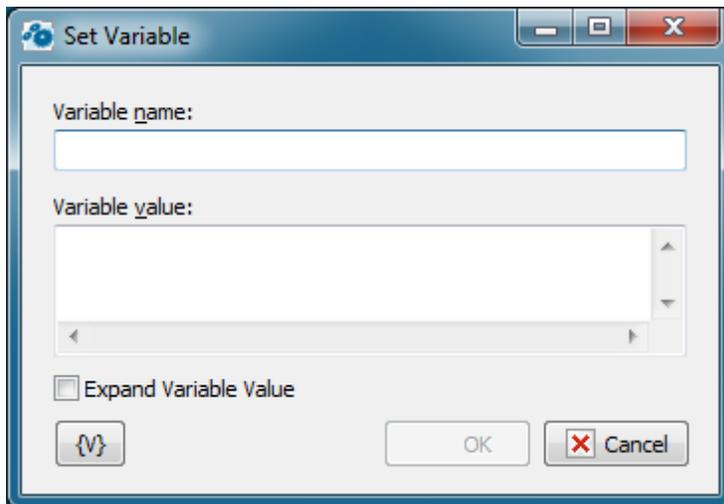
There can be necessity of using both modes in tasks:

- **Expand variable value is on** – when you have to fix the value in the moment of assigning it to the variable. For example: current time
- **Expand variable value is off** – when you have to compose a complex expression of many variables and then you have to use only one variable in further steps. You also don't need to turn on the expanding of variables if an assigned expression contains a static text and doesn't contain other variables.

By default, the parameter is off.

## 4.9.2 Set Variable

Defines the value of a variable when you run a task.

**Variable Name**

Enter the name of the variable you want to set the value of.

**Variable Value**

Enter the value that you want to set for the variable.

**Expand variable value.**

If this parameter is on, RoboTask computes the value of an expression (that is, expands variables in the expression) before assigning a value to the specified variable.

Otherwise, the value is assigned without expanding variables. In this case the value is computed while further using the variable. Meanwhile, the variable value and the value of all variables are expanded recursively until the very end.

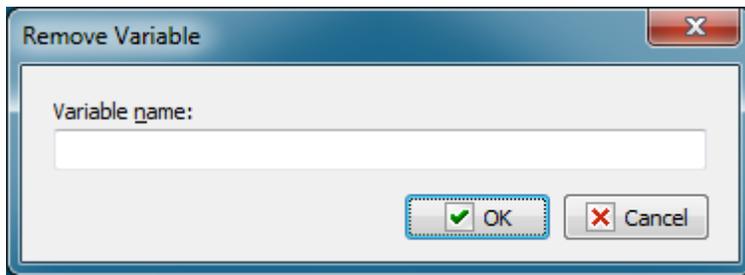
There can be necessity of using both modes in tasks:

- **Expand variable value is on** – when you have to fix the value in the moment of assigning it to the variable. For example: current time
- **Expand variable value is off** – when you have to compose a complex expression of many variables and then you have to use only one variable in further steps. You also don't need to turn on the expanding of variables if an assigned expression contains a static text and doesn't contain other variables.

By default, the parameter is off.

### 4.9.3 Remove Variable

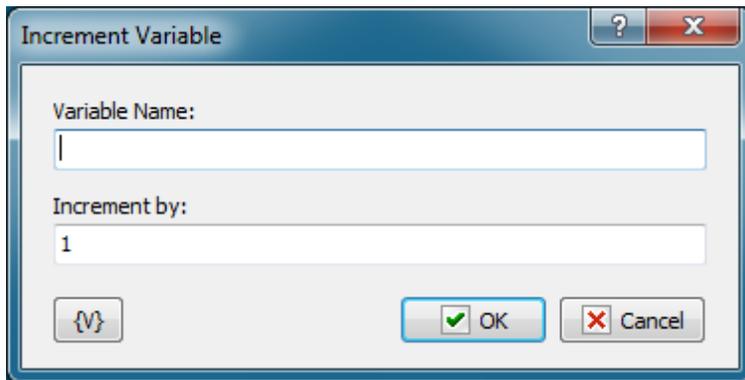
Deletes a variable when you run a task. This ensures that all variables are cleared from memory if a new task starts.

**Variable Name**

Enter the name of the variable you want to delete.

#### 4.9.4 Increment Variable

Increments or decrements a variable by a specified value.

**Variable Name**

Specify a variable name you want to increment or decrement

**Increment by**

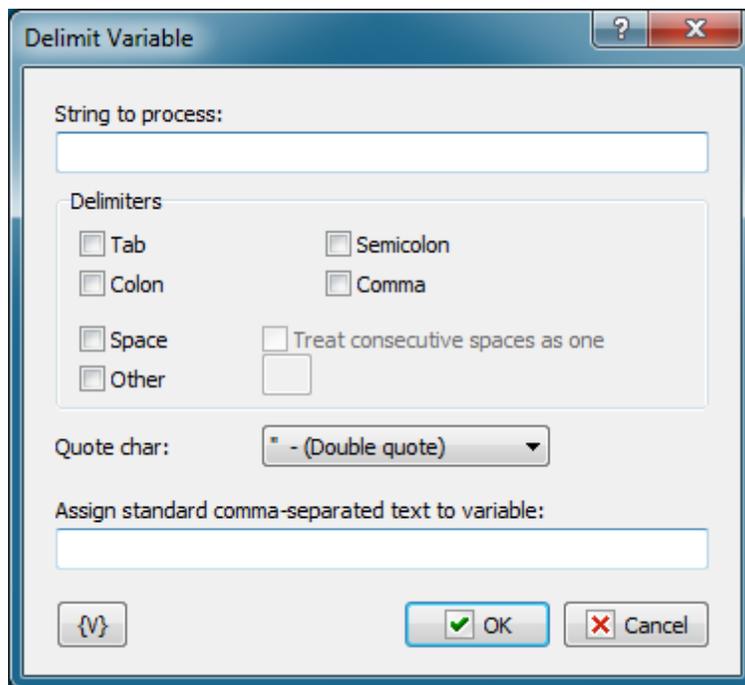
Enter a positive or negative value

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.9.5 Delimited variable

Turns a string with any delimiters into a standard comma-separated string. This standard comma-separated string can be used in user variables [as an array](#)<sup>[9]</sup>.

**String to process**

Specify the string to process. You can use variables.

**Delimiters**

Select which delimiters should be processed. You can select one or several delimiters.

**Quote Char**

Specify the character that should be interpreted as quotation marks.

**Assign standard comma-separated text to variable**

Specify the variable the processed string should be assigned to.

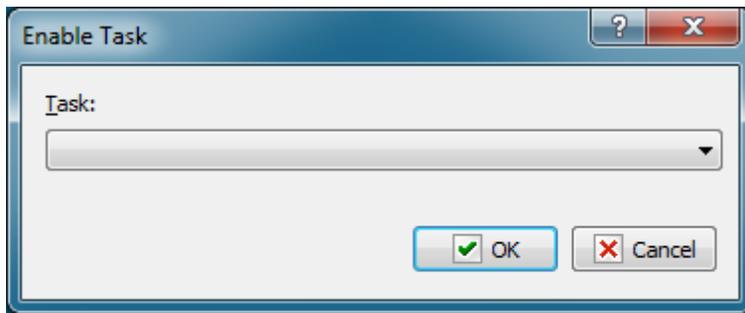
**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.10 Task Commands

### 4.10.1 Enable Task

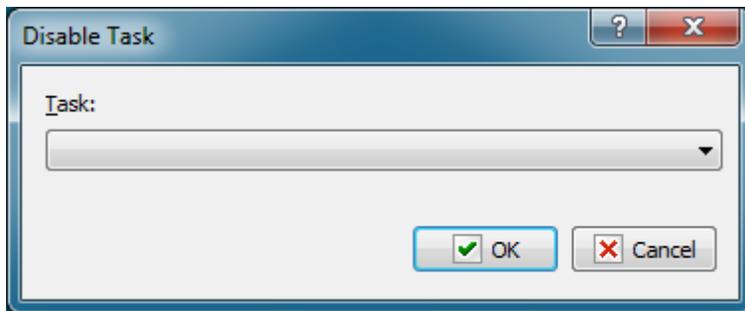
Enables a task. Disabled tasks cannot run automatically, i.e. their triggering events are disabled.

**Task**

Select a existing task.

### 4.10.2 Disable Task

Disables a task. Disabled tasks cannot run automatically, i.e. their triggering events are disabled.

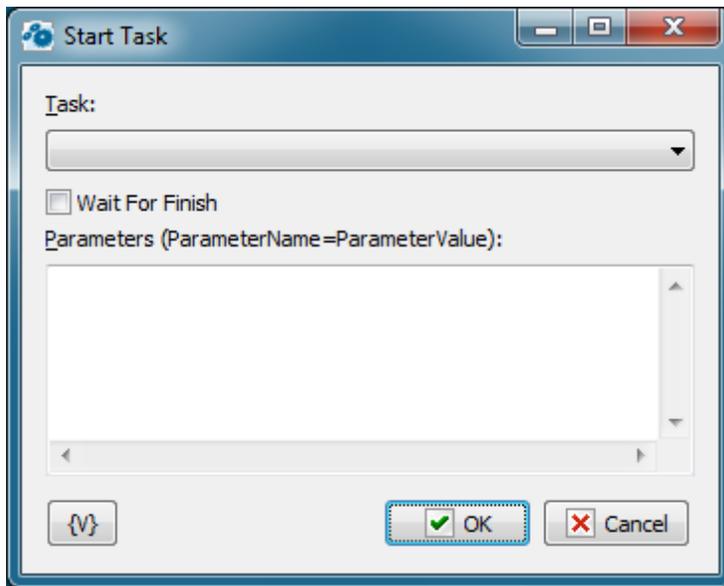
**Task**

Select a existing task.

### 4.10.3 Start Task

Runs a task. This action is useful if you want to run several tasks in a series. For example, you can create several different tasks in a series based on certain conditions that are detected in the following order:

1. A task that connects to the Internet and checks the e-mail server,
2. A task that logs off if no e-mail is present,
3. A task that displays a message box if you have received e-mail.



### Task

Select a existing task.

### Wait for Finish

Click this check box to ensure that the **Start Task** action starts a task and waits for completion of that task, then continues running. If the check box is clear, the action starts a task and continues running immediately.

### Parameters

You can pass some parameters into started task. By default, the list of parameters is empty. You need write down parameters like the text, each line should be like:

*ParameterName=ParameterValue*

- **ParameterName** – the name of parameter, any word consisting of letters and numbers;
- **ParameterValue** – value of parameter like any set of symbols.

You can refer to these parameters in the task as the local variables of the task. You don't need declare them in the task. Appropriate variables will appear automatically when the task is started.

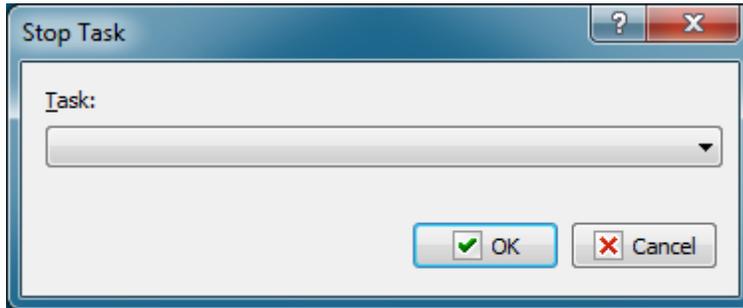
Pass of parameters of a task is used when you handle errors. In this case the fixed set of parameters is passing into the task:

- **LastErrorTaskID** – the internal ID of the task, in which an error was raised. It is a formal identifier of a task like a number.
- **LastErrorTaskName** – the name of the task, in which an error was raised.
- **LastErrorTaskExternalName** – the External Name of the task, in which an error was raised.
- **LastErrorStep** – the number of the step (starting from 1), in which an error was raised.
- **LastErrorCode** – code of the error. There is the particular set of error codes for any type of actions (see description of actions). By default – 0.
- **LastErrorDescription** – the description of the error. It depends on a type of action. By default, it is the line "General error".

The set of codes and descriptions will be increased in the over patching of present actions and development of the new one.

#### 4.10.4 Stop Task

Allows you to stop the selected task.

**Task**

Select the task you need from the list.

#### 4.10.5 Wait for Task

Waits until the selected task is finished. This action pauses the current task until the selected task is finished. If the selected task is not running at the moment when the action is due, the current task is not paused.

If the selected task is not finished during the specified timeout, the "Timeout is over" error is generated.

**Task**

Select a task from the list.

**Waiting**

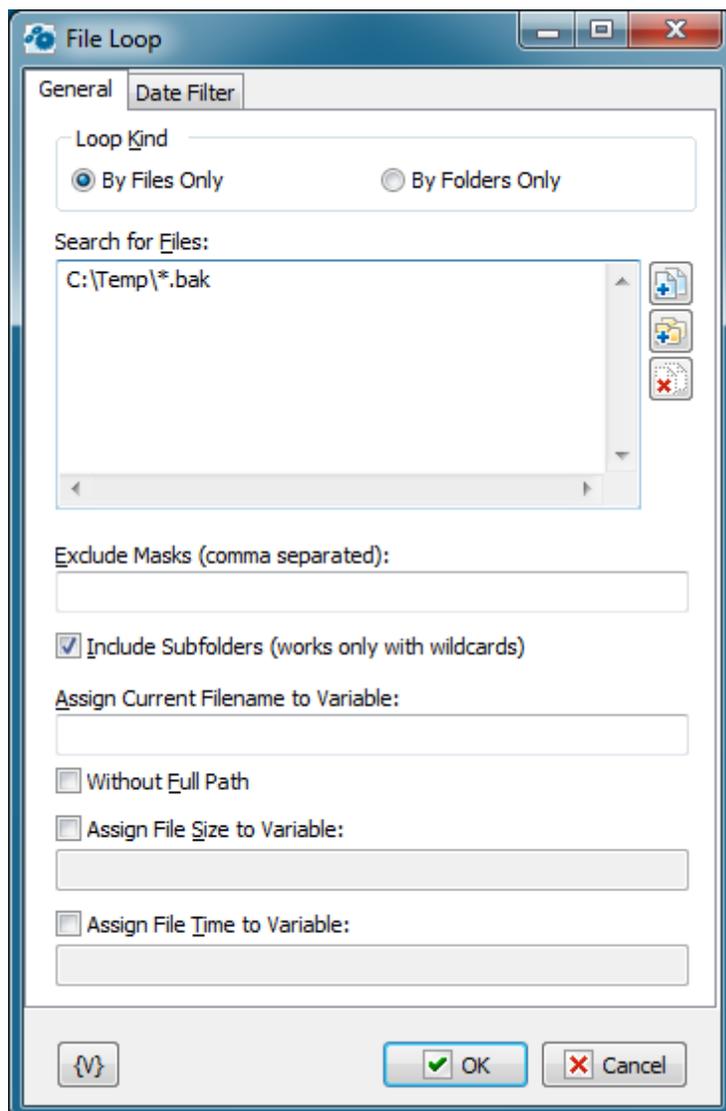
Specify the timeout in seconds to wait for the task to be finished. You can also specify an infinite timeout.

## 4.11 Loops and Flows

### 4.11.1 File Loop

Allows a task to iterate through the results of a file search. For example, you can create a task that searches all the employee status reports and identifies all status report files for the first of the month.

#### General tab



#### Search for Files

Enter the files that you want to include in the loop. You can click the browse buttons to the right of the list to add files, add folders, and remove items from the list.

**Tip**

You can also use wildcards in this option. Some examples of wildcards are the following:

- **C:\Text Files\\*.txt** - all the files with the .txt extension in the C:\Text Files folder.
- **C:\Doc Files\mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **{bakext}** - file mask(s) from the {bakext} variable.

**Exclude Masks (Comma Separated)**

Enter one or more file masks separated by commas for files that you do not want to include in the loop.

**Tip**

You can also use wildcards in this option. Some examples of wildcards are the following:

- **\*.txt** - all the files with the .txt extension.
- **mydoc?.doc** - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- **\*.tmp, \*.\$\$\$, \*.bak** - all the files with the .tmp, .\$\$\$ and .bak extensions.
- **{bakext}** - file mask(s) from the {bakext} variable.

**Recurse to Subfolders**

Click this check box to include subfolders in the loop.

**Assign Current Filename to Variable**

The current filename and path to the file in the loop is stored in the variable that you specify. You can use this information in other actions.

**Without Full Path**

Click this check box to save only the filename and not the path to the variable.

**Assign File Size to Variable**

The current file size in the loop is assigned to the variable that you specify. You can use this information in other actions.

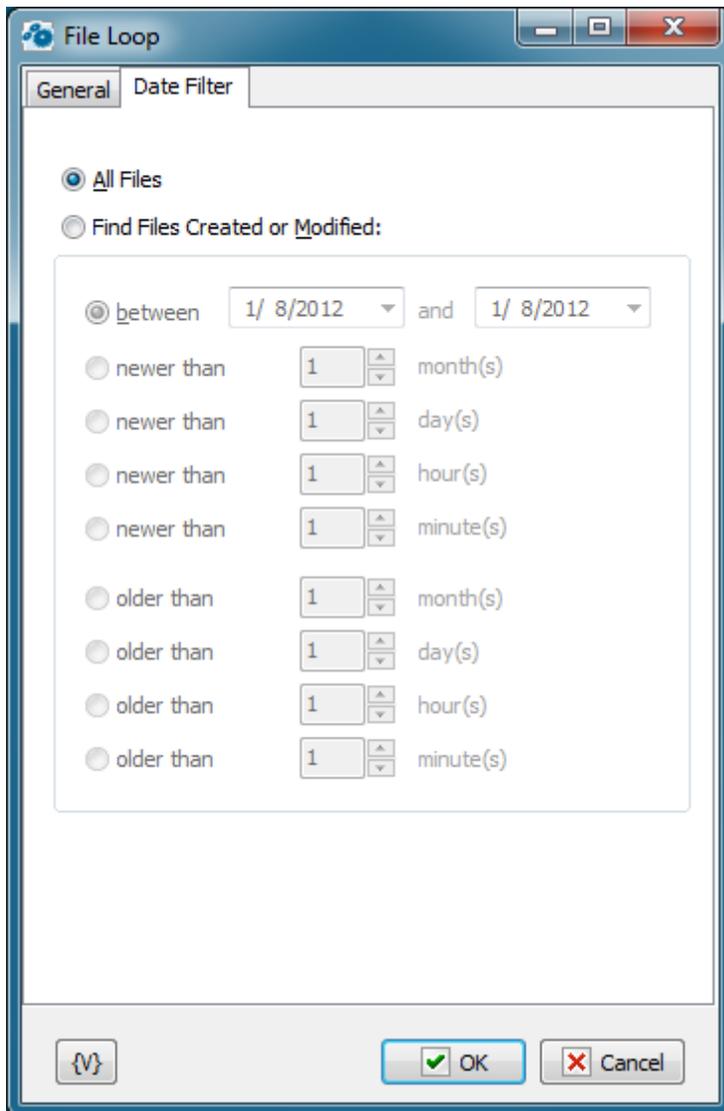
**Assign File Time to Variable**

The current time of the file in the loop is assigned to the variable that you specify. You can use this information in other actions.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

**Data Filter tab**



### All Files

Click this option to include all files in the file loop.

### Find Files Created or Modified

Enter dates and times to include files for certain time periods in the loop.

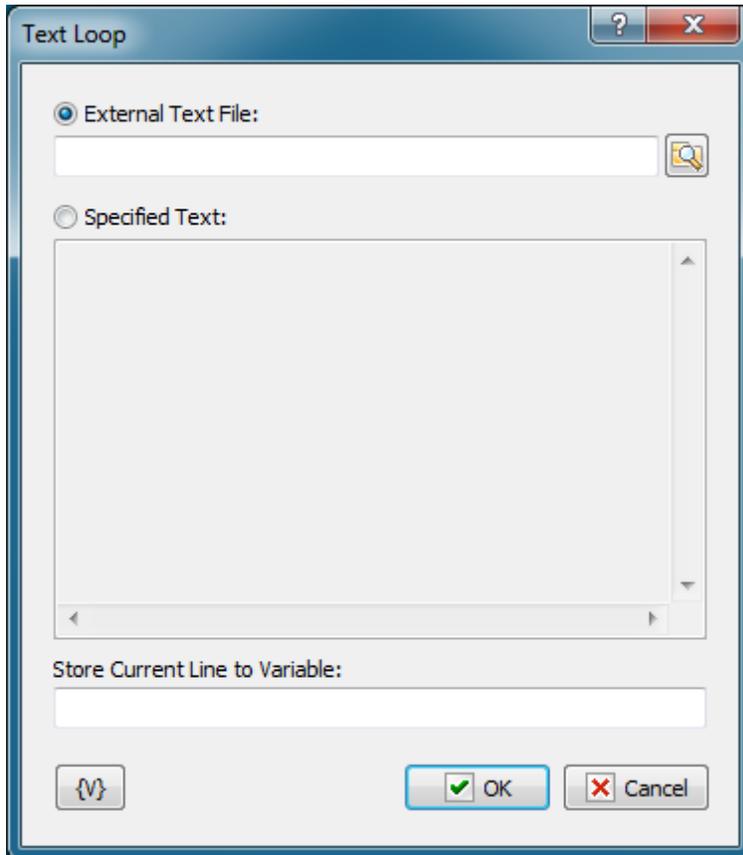
- **Between**
- **Newer Than \_\_\_ Months**
- **Newer Than \_\_\_ Days**
- **Older Than \_\_\_ Months**
- **Older Than \_\_\_ Days**

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.11.2 Text Loop

Allows a task to iterate through all lines of the text that you specify. You can enter text in the parameters of the action or you can refer to a text file.

**External Text File**

Enter the path and name of text file or click the browse button to select a text file.

**Specified Text**

Enter the text you want to loop through.

**Store Current Line to Variable**

Enter a variable name to store with the current line of text.

**Variable**

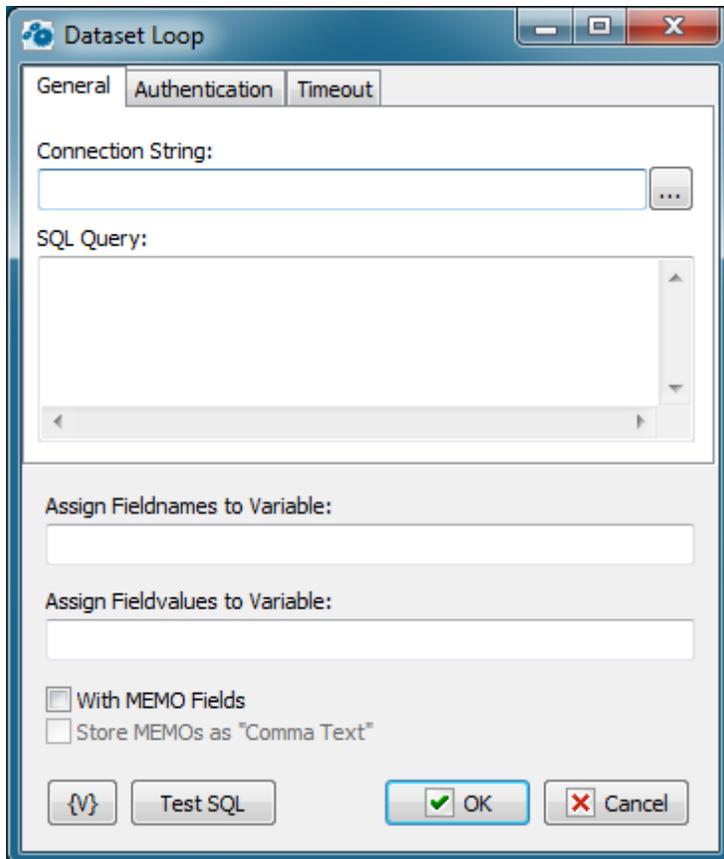
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.11.3 Dataset Loop

Allows a task to iterate through the values of the results of an SQL query. For example, you might create a task that connects to the company human resources database and queries for all records of employee sick leave reports every month and saves a report to a file.

#### Important

- You must be familiar with SQL syntax to use this action.



#### Connection String

A string with attribute/value pairs for information required to log on to a database and point to a specific database. Click the browse button to open the Microsoft **Data Link Properties** dialog box. You can press F1 to access Help about this dialog box.

#### SQL Query

Enter the query string in SQL syntax.

#### Assign Field Names to Variable

Stores the field names separated by commas in a specified variable.

#### Assign Field Values to Variable

Stores the field values separated by commas in a specified variable.

#### With Memo Fields

Specifies that the query should include memo fields.

**Store Memos as "Comma Text"**

Indicates that the memo fields should be stored as comma delimited text.

**Variable**

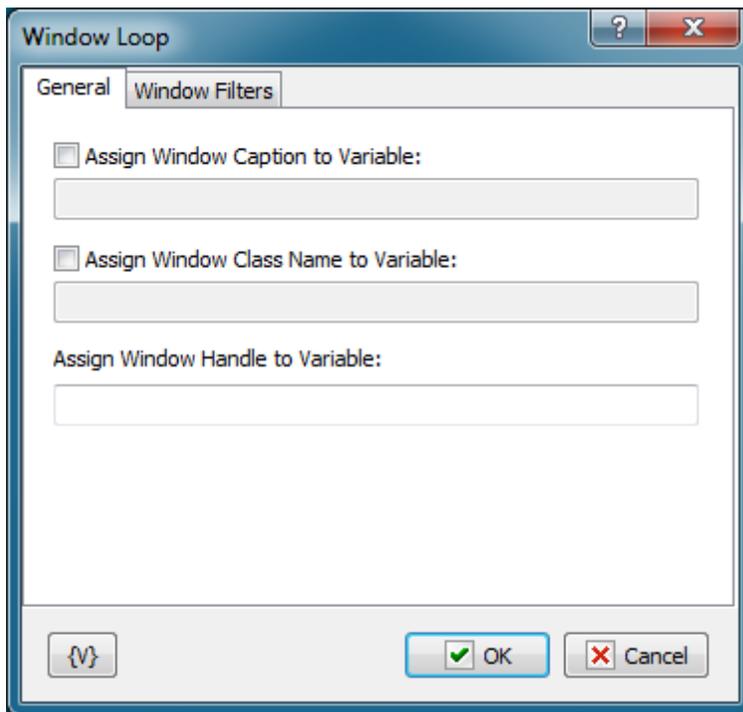
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

**Test SQL**

Allows you to test the SQL syntax of your query.

#### 4.11.4 Window loop

Enumerates windows existing in the system and allows you to assign their parameters to variables. You can use these values inside the loop.

**General****Assign Window Caption to variable**

You can specify the variable you want to assign the window caption to. If you clear this checkbox, the window caption will not be saved.

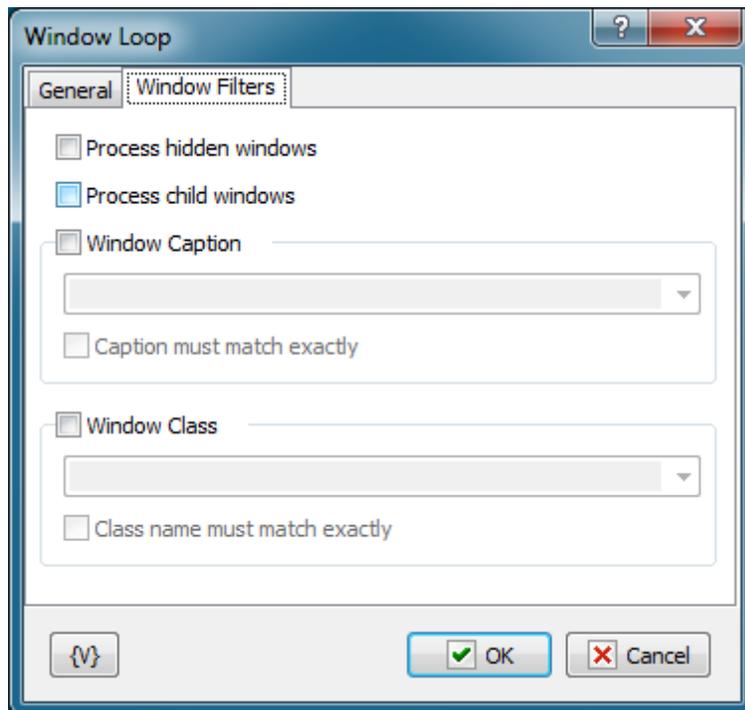
**Assign Window Class to variable**

You can specify the variable you want to assign the window class to. If you clear this checkbox, the window class will not be saved.

### Assign Window Handle to variable

Specify the variable you want to assign the window handle to. You cannot disable this option because the handle of a window is its unique identifier. You can use the value of this variable in the "Window Command" action.

## Window Filters



### Process Hidden Windows

If you want to enumerate hidden windows, you should select this checkbox. By default, Window loop enumerates only visible windows.

### Process Child Windows

Select this checkbox to enumerate child windows.

### Window caption

You can use this parameter to apply a filter to the captions of windows. Either select the caption you need from the list or type it manually.

### Window class

You can use this parameter to apply a filter to the classes of windows. Either select the class you need from the list or type it manually.

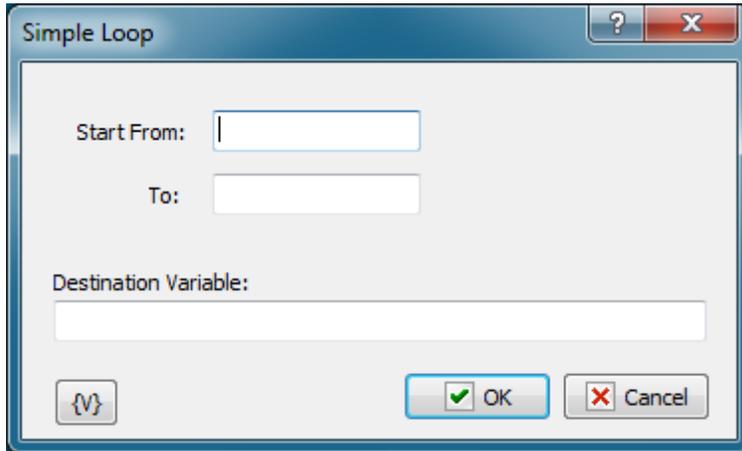
The "Window caption" and "Window class" lists depend on the "Process Hidden Windows" and "Process Child Windows" checkboxes.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.11.5 Simple Loop

Allows a task to iterate through a loop.

**Start From**

Enter the number at which to start the loop.

**To**

Enter the number at which to end the loop.

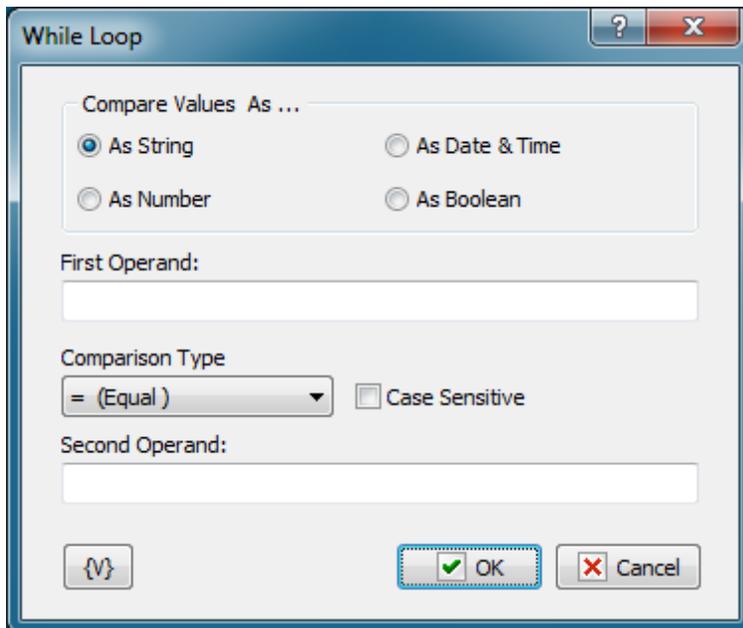
**Destination Variable**

Stores the current value of the loop. For example, if you enter 1 in the **Start From** box and 5 in the **To** box, then, in the first iteration, the destination variable contains 1. In the second iteration, the destination variable contains 2, and so forth.

### 4.11.6 While loop

The **While loop** is used if the number of iterations is unknown, but you know the condition that should be met for the loop to continue. The settings of this action are similar to the **If.. then** action and it has similar parameters.

**Attention:** it is possible to make an infinite loop with **While Loop** if you specify the condition incorrectly. So, be careful while creating a task if you do not intend to make an infinite loop, of course.



### Compare Values As

Select the type of data for the expressions that you want to compare.

- **As String** - compares two values with alphanumeric characters.
- **As Number** - compares two integer values.
- **As Date and Time** - compares two date/time values.
- **As Boolean** - checks for a true/false or yes/no condition. If you select this option, the **If Value Is** option appears, allowing you to select the condition of the Boolean.

### If Value Is

Appears if you select **As Boolean** in the **Compare Values As** area. This option allows you to set the value of the Boolean check. You can select false for certain actions to occur if a condition does not exist. You can select true for certain actions to occur if a condition does exist.

### First Operand

Enter the value of the first expression you want to compare.

### Comparison Type

Select the operator for the comparison from the following:

- = Equal to
- <> Not equal
- >= More than or equal to
- > More
- <= Less than or equal to
- < Less than

### Case Sensitive

Click to make the comparison dependent on whether upper case or lower case characters are used in the expressions.

### Second Operand

Enter the value of the second expression you want to compare.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

**4.11.7 End Loop**

Indicates the end of the loop block. Any loop, including a file loop, simple loop, and so forth, must contain the corresponding **End Loop** action.

**4.11.8 Break**

Stops the execution of the current loop so that the task can continue executing the actions after the **End Loop** action. The **Break** action is often used in conjunction with the **If Then** action.

**4.11.9 Continue**

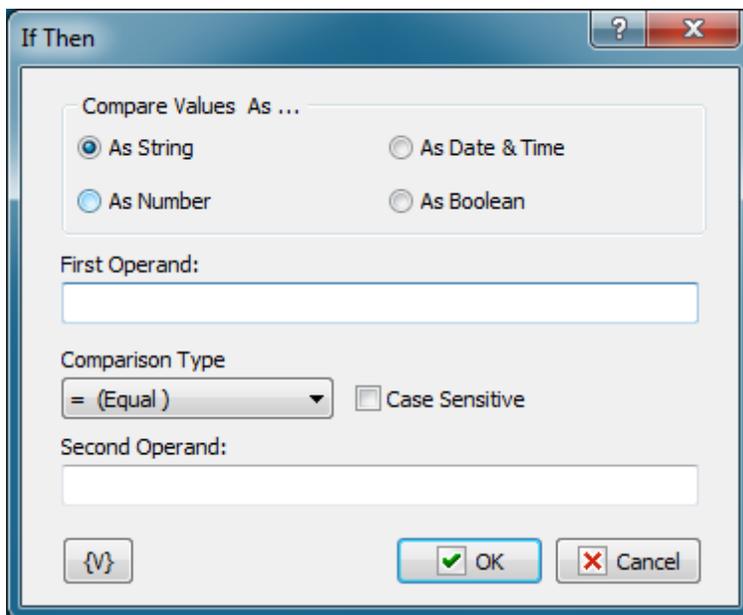
Skips all the actions between it and the **End Loop** action and begins a new iteration. No specific settings are associated with this action. The **Continue** action is used in conjunction with the **If Then** action.

**4.11.10 If Then**

Allows a task to make a comparison of several conditions and execute actions based on that comparison. You can use this action to run a specific statement or a block of statements, depending on the value of a condition. **If...Then...Else** statements can be nested to as many levels as you need.

**Important**

- The **If Then** action must be used in conjunction with the **End If** action.



### Compare Values As

Select the type of data for the expressions that you want to compare.

- **As String** - compares two values with alphanumeric characters.
- **As Number** - compares two integer values.
- **As Date and Time** - compares two date/time values.
- **As Boolean** - checks for a true/false or yes/no condition. If you select this option, the **If Value Is** option appears, allowing you to select the condition of the Boolean.

### If Value Is

Appears if you select **As Boolean** in the **Compare Values As** area. This option allows you to set the value of the Boolean check. You can select false for certain actions to occur if a condition does not exist. You can select true for certain actions to occur if a condition does exist.

### First Operand

Enter the value of the first expression you want to compare.

### Comparison Type

Select the operator for the comparison from the following:

- = Equal to
- <> Not equal
- >= More than or equal to
- > More
- <= Less than or equal to
- < Less than

### Case Sensitive

Click to make the comparison dependent on whether upper case or lower case characters are used in the expressions.

### Second Operand

Enter the value of the second expression you want to compare.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

**4.11.11 Else**

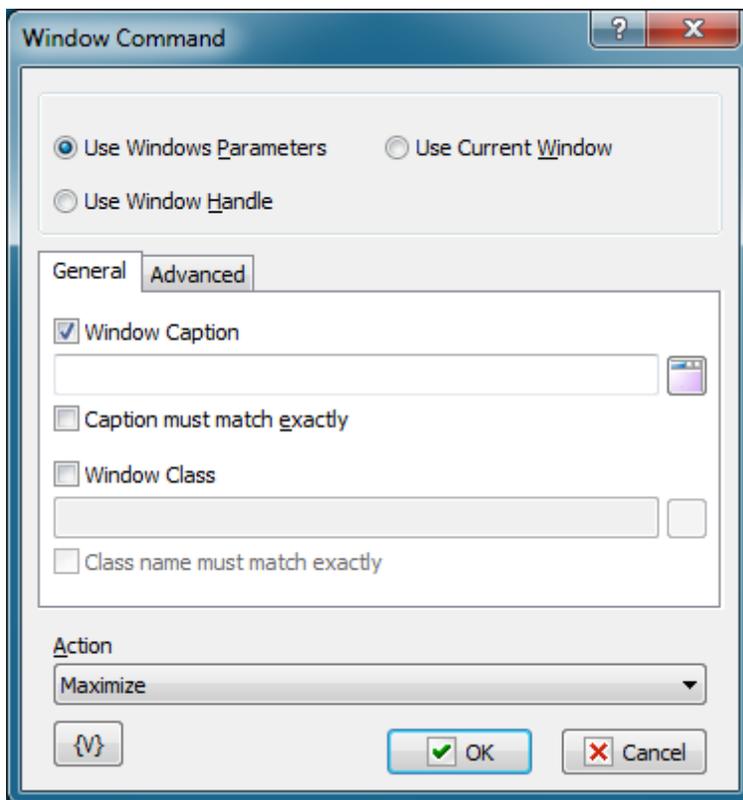
Allows a task to execute an action if certain conditions are not met. This action is used in conjunction with the **If Then** action. You can use this action to run a specific statement or a block of statements, depending on the value of a condition. If...Then...Else statements can be nested to as many levels as you need. No specific settings are associated with this action.

**4.11.12 End If**

Indicates the end of the **If Then** block.

**4.12 Window****4.12.1 Window Command**

Manipulates certain windows that you specify.



### Window Caption

Enter or select the caption of the window that you want to manipulate with the action you select in the **Action** list. You can enter part of a window caption.

### Caption Must Match Exactly

Select this check box if you want to manipulate only the window that exactly matches the window caption you selected. This check box is cleared by default so that the task can manipulate windows that contain only part of the window caption entered in the **Window Caption** box. For example, you can enter <notepad> for the window with a caption <Untitled - Notepad>.

### Action

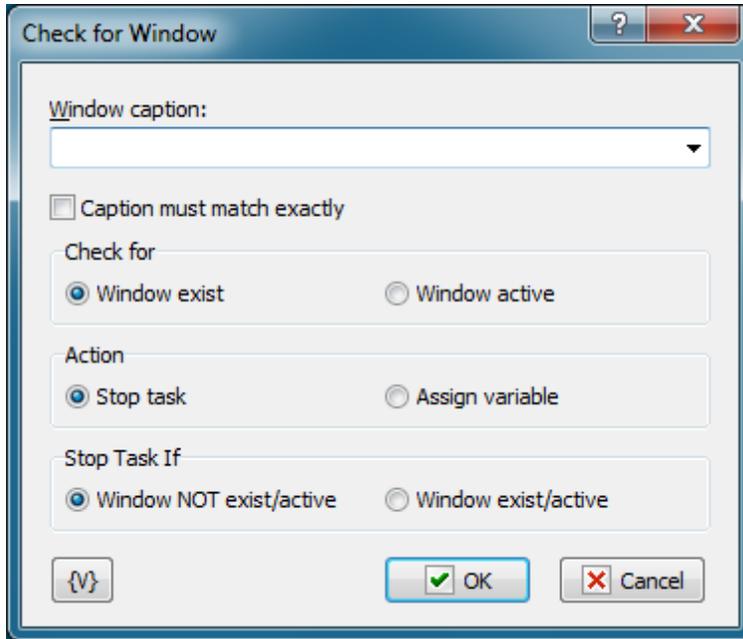
Select the action for manipulating the window.

- **Focus** - makes the window active.
- **Minimize** - reduces the window to an icon on the Windows taskbar.
- **Maximize** - restores the window to fill the entire screen.
- **Restore** - restores the window to its previous state.
- **Close** - closes the window.
- **Hide** - hides the window from view on the screen and Windows taskbar.
- **Unhide** - displays the hidden window.

## 4.12.2 Check for Window

Checks to see if a window exists or is active. For example, you might want to create a task

that starts your HotSync Manager for your Palm PDA every day and synchronizes the Palm device data with Outlook. You might want to check for the existence of the HotSync Manager window to ensure that it has started.



### Window Caption

Select the name of the window from the dropdown list. The contents of this list vary with the applications currently launched on your computer.

### Caption Must Match Exactly

Click the check box to check only windows that match the entire window caption. If this check box is clear, the action checks for windows that partially match the caption.

### Check For

Select one of the following options to check for existing windows or windows that are open.

- **Window Exist**
- **Window Active**

### Abort Task If

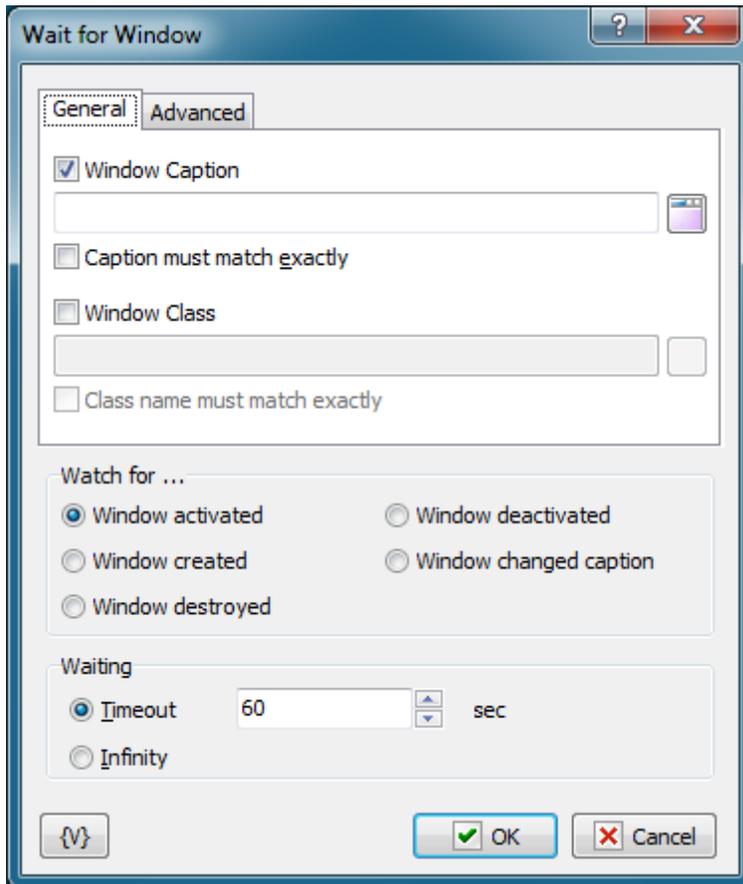
Stops task executing according with selected option:

- **Window Not Exists/Active** - stops the task if the window does not exist or is not active.
- **Window Exist/Active** - stops the task if the window exists or is active.

## 4.12.3 Wait for Window

This action is used to synchronize the task with opening or closing a certain window. The action pauses the task and waits for the necessary window operation for the specified timeout and after that resumes the task. If the action does not detect the necessary window operation

within the timeout, the "Timeout is over" error occurs.



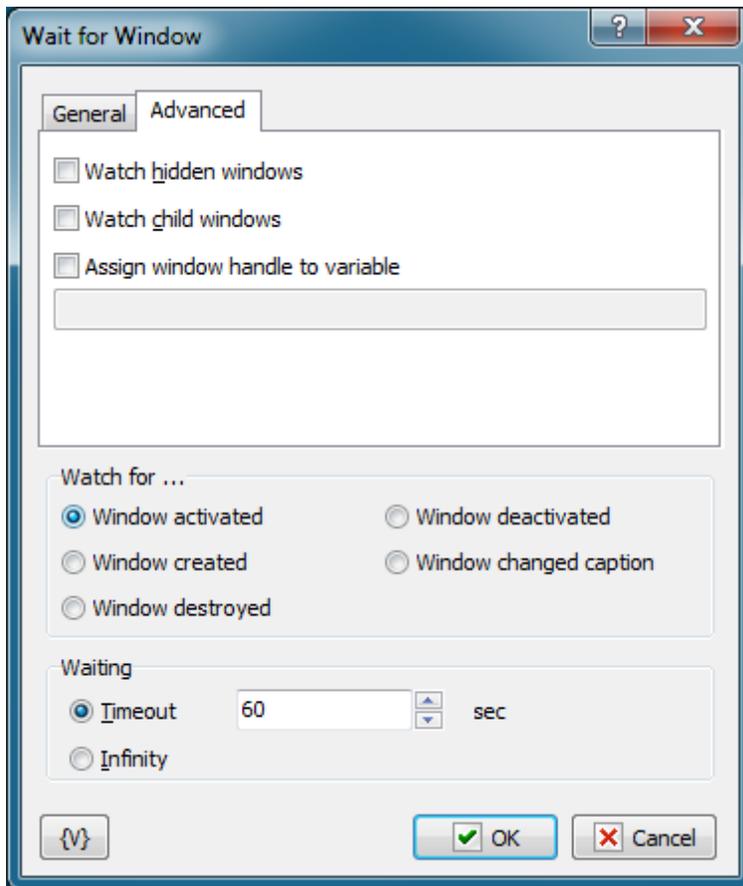
### General tab

**Window Caption** – Window caption. You can specify only the distinctive part of the caption if the window has no constant caption. You can select the window you need from the list if you click the button to the right from the input field.

**Caption must match exactly** – If you need the window caption to match the specified string exactly, select this checkbox.

**Window class** – Window class (or window type). You should either type the window class manually or select the window you need from the list by clicking the button to the left from the input field. You can type part of the window class as well.

**Class name must match exactly** – If you need the window class to match the specified string exactly, select this checkbox. The class of a window does not change all the time the window exists. So, as a rule, you have to enable this option.



### Advanced tab

**Watch hidden windows** – select this checkbox if you want the action to monitor hidden windows.

**Watch child windows** – select this checkbox if you want the action to monitor child windows. As a rule, these windows are dialog boxes and secondary windows opened by Windows applications.

**Assign window handle to variable** – if you want to save the HANDLE (or the system number) of a window to a variable, type the name of this variable in the field. Otherwise leave the field empty. The HANDLE of a window is a unique window identifier. And you can use it later in the Window command action or in Basic script while using it in the script of system API functions.

### Common settings

**Watch for** – select the window operation to wait for. There are 4 options to choose from:

- Window activated;
- Window created;
- Window destroyed;
- Window deactivated;
- Window changed caption.

**Waiting** – specify the timeout in seconds or select the Infinity option.

## 4.13 For Windows NT

### 4.13.1 Start Service

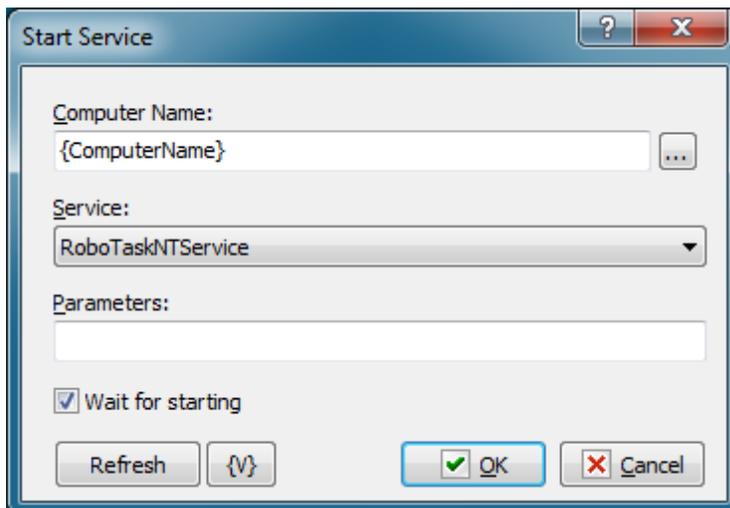
#### Important

- This action is available only on Windows NT 4, 2000, and XP.

Starts a service already installed on the local or remote NT server or workstation. A service is a special process that can be launched during operating system boot. The most important feature of a service is that you do not have to log on to the computer.

A service is started in the Local System account. You can start an existing service or a service that you created. Examples of services include FTP/HTTP servers, print spoolers, file sharing, and so forth.

Usually, Windows services do not have a user interface, but are managed through the Services applet of the Windows Control Panel, or a separate application or applet. Using the standard Services applet, you can start, stop, pause, or continue a previously installed service. The common way for a service to report a warning or error is recording an event into the system event log. You can review the log using the **Event Viewer** from **Administrative Tools** in Windows.



#### Computer Name

Enter the remote computer name. If you want to start a service on the local computer, use {ComputerName}. You can click the browse button to select a computer from the local area network.

#### Service

Select a service that you want to stop from the dropdown list. The items on this list depend on which services are present on the computer.

#### Parameters

Specify optional parameters for the service that you want to start.

#### Refresh

Click this button to refresh the **Service** list.

### Variable

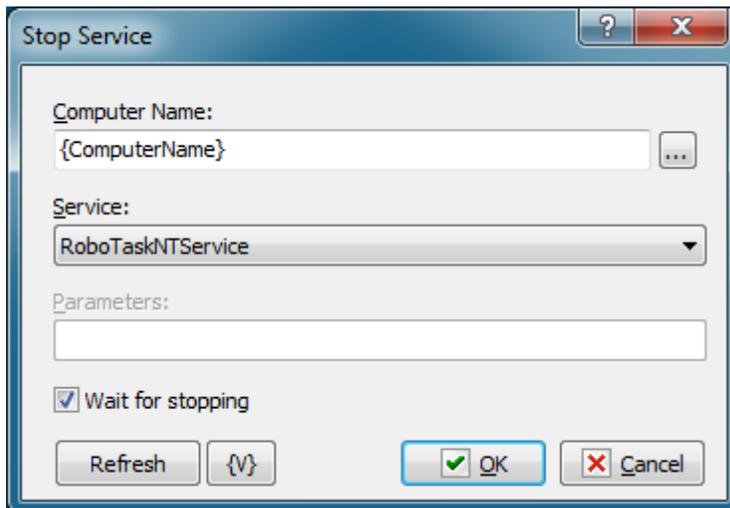
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.13.2 Stop Service

### Important

- This action is available only on Windows NT 4, 2000, and XP.

Stops a service already installed and started on the local or remote server or workstation. For more information, see [Start Service](#).<sup>[138]</sup>



### Computer Name

Enter the remote computer name. If you want to start a service on the local computer, use {ComputerName}. You can click the browse button to select a computer from the local area network.

### Service

Select a service that you want to stop from the dropdown list. The items on this list depend on which services are present on the local computer.

### Refresh

Click this button to refresh the **Service** list.

### Variable

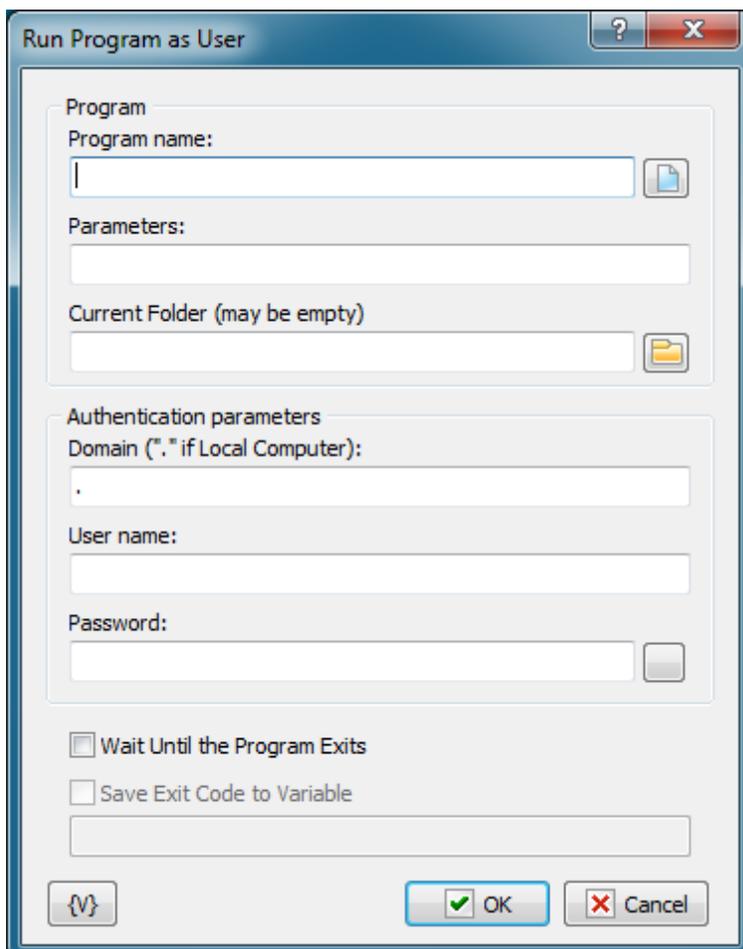
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.13.3 Lock Workstation

This action submits a request to lock the workstation's display. Locking a workstation protects it from unauthorized use.

### 4.13.4 Run Program as User

This action allows running some application with rights of another user. It is especially useful for the system service mode because the application (a spawned process) is implemented by default with the rights of SYSTEM account. Sometimes it can cause some problems. For example, the launched application needs the access into the network with certain rights, or it needs some user settings.



#### Program name

Enter the program name into this field. You can type the text manually or select a file via dialog. You can also specify the document name in this field. In this case the document is opened by the default program according to the system settings.

#### Parameters

Enter the parameters of the program.

**Current folder**

The name of the current folder when the program is started. You can leave this field empty. In this case the folder is specified automatically.

**Domain**

The domain name. If you want to use the username of a local computer, write the dot symbol (.).

**User Name**

The name of user.

**Password**

The password of user.

**Save Exit Code to Variable**

If you want to save the exit code of the program, turn on this flag and write the variable name below. It is possible to save the code of the program completion only if the idle mode of the program completion is on.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.14 RoboTask Commands

### 4.14.1 Minimize RoboTask

Collapses the RoboTask window and displays the RoboTask icon on the Windows Taskbar. No specific settings are associated with this action.

This action is useful if you do not want RoboTask to appear on the computer while the task runs.

### 4.14.2 Restore RoboTask

Activates RoboTask and displays the RoboTask window on the computer screen. No specific settings are associated with this action.

This action is useful for system administrators who want to troubleshoot a process. For example, if a "critical error" occurs, you can use the **Disable Automation** action, trap an error message with the message box, and then use the **Restore RoboTask** action.

### 4.14.3 Exit RoboTask

Closes RoboTask. No specific settings are associated with this action.

### 4.14.4 Enable Automation

Enables task triggering. If task triggering is disabled, the tasks will not run automatically, i.e. all your tasks are disabled. You may run tasks only manually.

### 4.14.5 Disable Automation

Disables task triggering. If task triggering is disabled, the tasks will not run automatically, i.e. all your tasks are disabled. You may run tasks only manually.

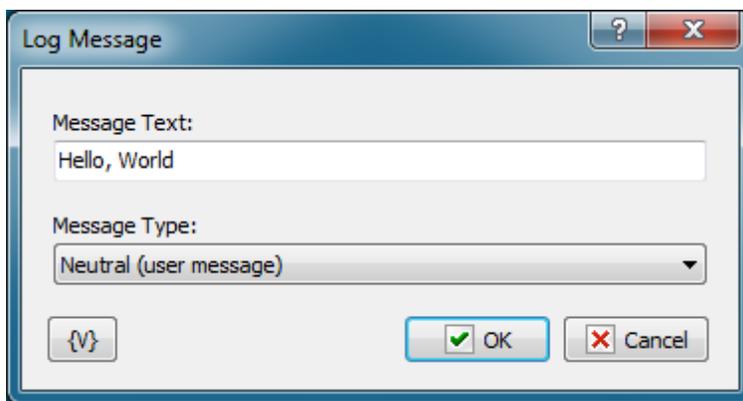
### 4.14.6 Log Message

Records a message to the task's log. This action is a powerful tool for recording the results of actions when they occur in a log text file. This action is useful for system administrators who want to troubleshoot a process.

The log filename is based on the internal unique ID number of the task. Each task has an internal unique ID number and its filename is NNNNNNNNN.tsk where NNNNNNNNN is its ID number. The log file name is NNNNNNNNN.log.

#### Important

- You can set options for logging on the [Advanced tab](#)<sup>[26]</sup> of the **Add Task** dialog box or the **Edit Task** dialog box.



#### Message Text

Enter the text that you want to appear in the message.

### Message Type

Select a type, such as **Information**, **Warning**, etc. The type determines the style of log message:

- **Information** - a blue text and "i" icon appears.
- **Warning** - an olive text and "!" icon.
- **Error** - a red text and "X" icon.
- **Neutral (user message)** - black text without icon. This kind of message can't be hidden even if you specify "Nothing" in "Log Level" parameter of the task.

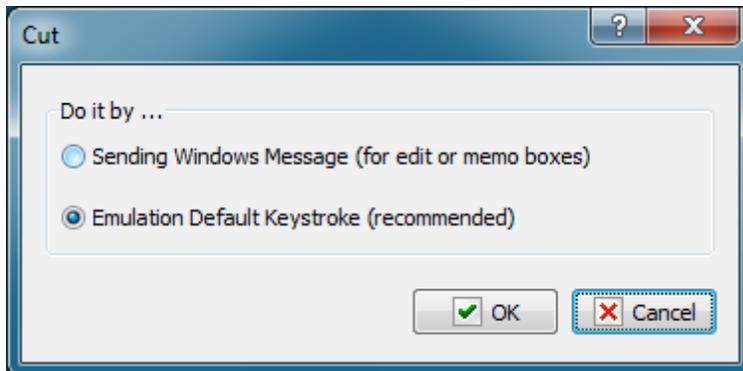
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.15 Clipboard

### 4.15.1 Cut

Cuts the selected text fragment (image) into the system clipboard from the active Windows application.



#### Do it by...

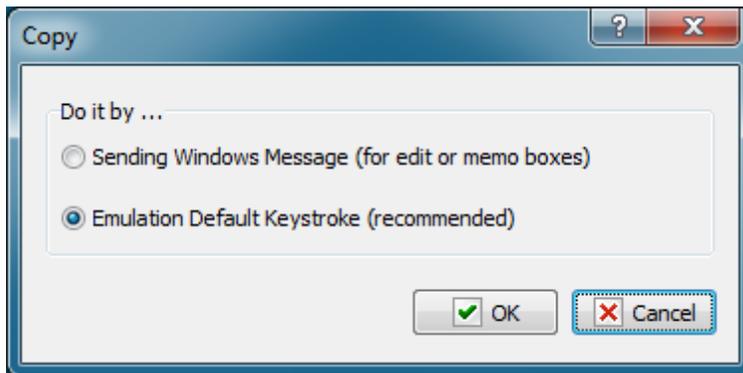
There are two ways to cut a fragment:

1. by sending a Windows message to the corresponding edit element
2. by emulating the key combination

It is recommended to use the second method because it is a universal mechanism for any application and any data type.

### 4.15.2 Copy

Copies the selected text fragment (image) into the system clipboard from the active Windows application.

**Do it by...**

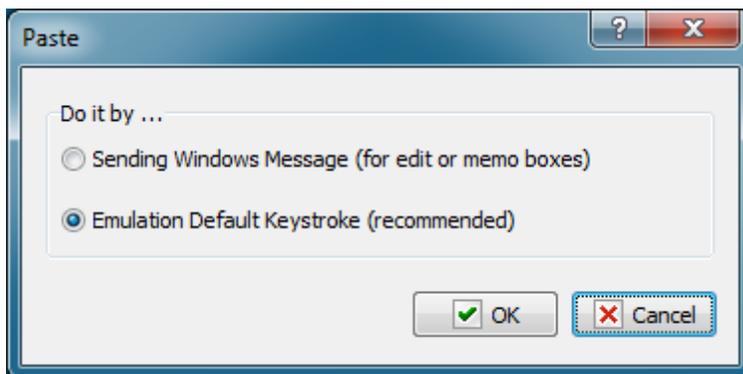
There are two ways to copy a fragment:

1. by sending a Windows message to the corresponding edit element
2. by emulating the key combination

It is recommended to use the second method because it is a universal mechanism for any application and any data type.

### 4.15.3 Paste

Pastes the copied fragment from the system clipboard to the current Windows application.

**Do it by...**

There are two ways to paste a fragment:

1. by sending a Windows message to the corresponding edit element
2. by emulating the key combination

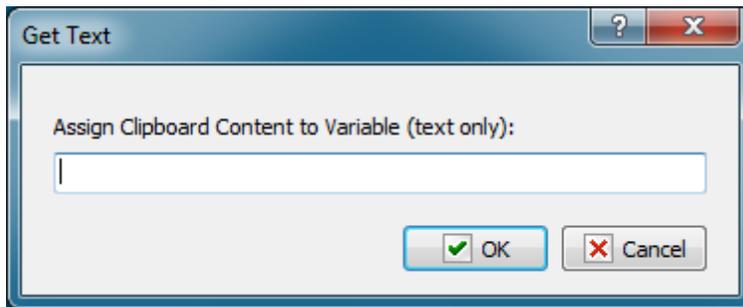
It is recommended to use the second method because it is a universal mechanism for any application and any data type.

#### 4.15.4 Clear Clipboard

Clears the system clipboard. It has no parameters.

#### 4.15.5 Get text

Assigns the text from the system clipboard to a variable. If the clipboard contains a graphic object or an object of some other type that cannot be converted into text, an empty string will be assigned to the variable.

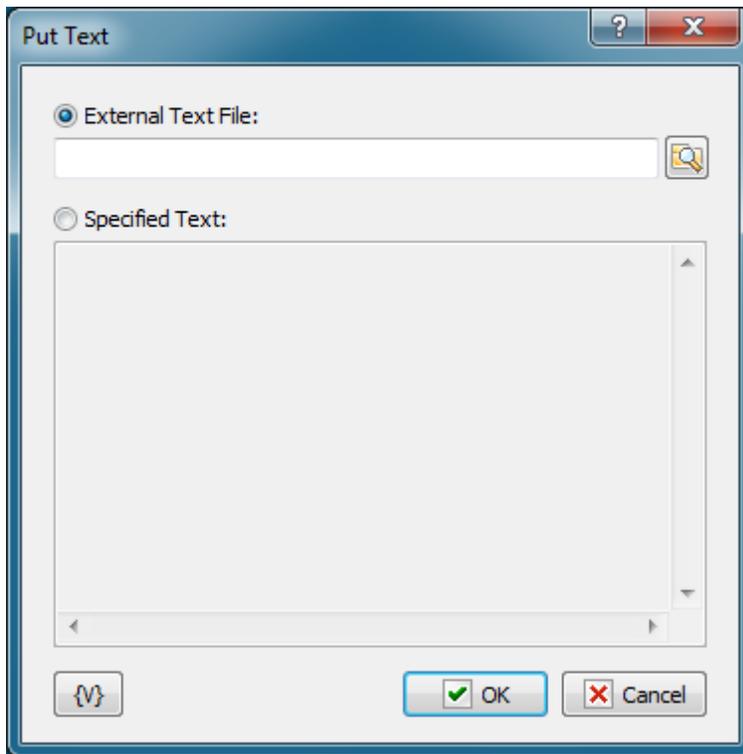


##### **Assign Clipboard Content to variable**

Specify the variable the text from the system clipboard should be assigned to.

#### 4.15.6 Put Text

Copies the specified text into the system clipboard. You can take text from a file or type it directly in the "Specified text" field. User and system RoboTask variables can be used.

**External Text File**

Specify the name of the file to take the text from.

**Specified Text**

Type the fragment that should be copied into the clipboard.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.16 Mouse

### 4.16.1 Move to object

Moves the mouse pointer to the specified object. You can select a control element in a window (button, menu, toolbar, etc.) or a window as an object.

To make editing this parameter more comfortable, you can drag the message **"Drag this text to desired control"** to the object you need. While you are moving the mouse pointer, elements recognized by RoboTask are highlighted with a frame. After you release the left mouse button, all the parameters are specified automatically depending on the position of the mouse pointer. After that you can edit the parameter manually if necessary.

#### **Window caption**

Specify the window caption completely or any part of it. If the caption does not matter, leave the field blank.

#### **Window caption must match exactly**

If you need the caption to match exactly the specified caption, select this checkbox. If you do not need the exact match, you can specify any part of the window caption.

#### **Window class**

Specify the window class completely. If the class does not matter, leave the field blank.

#### **Control Text**

Specify the text of the window control element completely or partially.

#### **Control Text must match exactly**

If you need the control text to match exactly the specified text, select this checkbox. If you do

not need the exact match, you can specify any part of the control element text.

### Control class

Specify the control class completely. If the class does not matter, leave the field blank.

### Control position

Specify the coordinates of the control element. The values are specified relatively to the window. The coordinates of the control are necessary in order to identify the control element because the window may contain several elements with the same Control Text and Control Class.

### Mouse Cursor Position

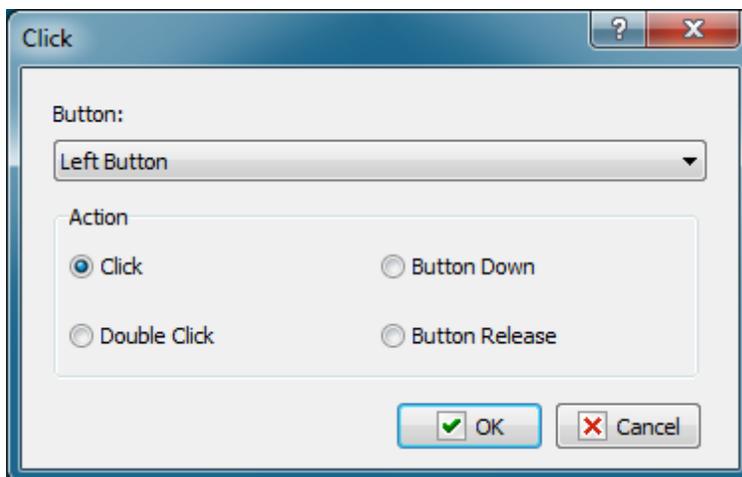
Specify the position of the mouse pointer relatively to the control element.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.16.2 Click

Emulates a mouse click in the current position.



### Button

Select the mouse button:

- Left Button
- Right Button
- Middle Button.

### Action

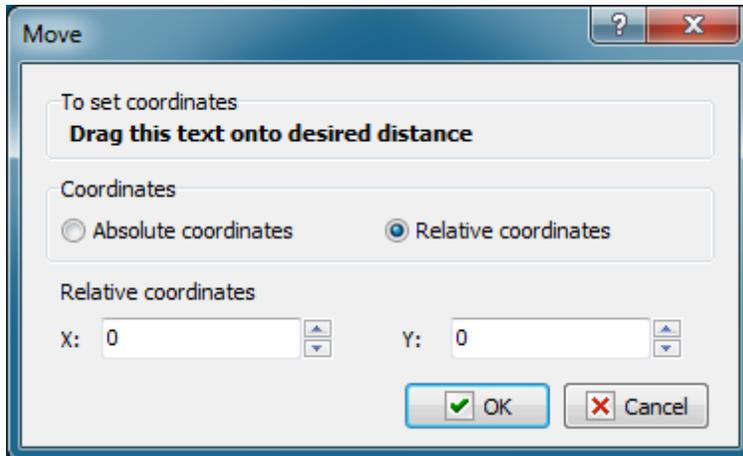
Select the action. There are 4 variants to choose from:

- Click
- Double Click
- Button Down

- Button release

### 4.16.3 Move

Moves the mouse pointer the specified distance away from the current position or to the specified point.



#### Coordinates mode

Coordinates: absolute or relative.

- Absolute coordinate - the absolute coordinates of a point on the screen will be specified in the next parameter.
- Relative coordinates - the coordinates of a point on the screen relative to the current position of the mouse pointer will be specified in the next parameter.

#### Absolute or Relative coordinates

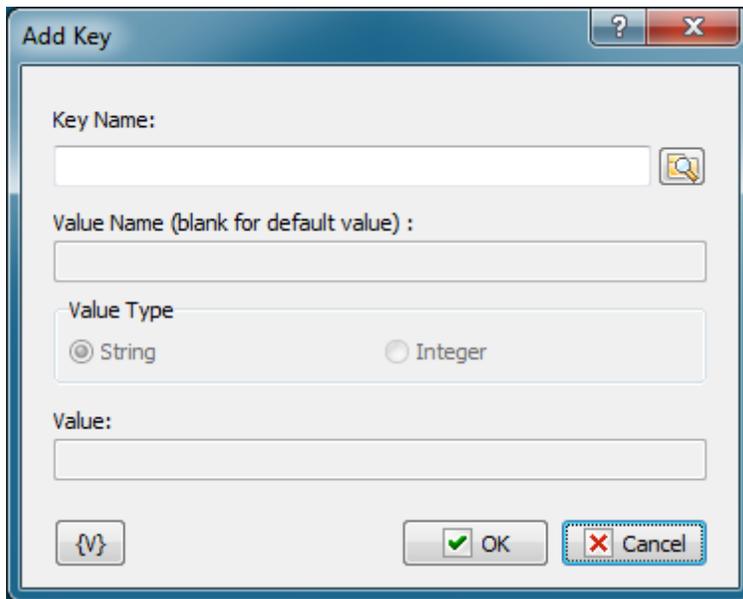
The coordinates of a point where the mouse pointer should be moved. The type of coordinates is determined by the previous parameter.

In order to set the coordinates interactively, you should drag the text in the upper part of the dialog box to the necessary point on the screen. The coordinate values will be specified automatically.

## 4.17 Registry

### 4.17.1 Add key

Adds a key (branch) to the registry.

**Key Name**

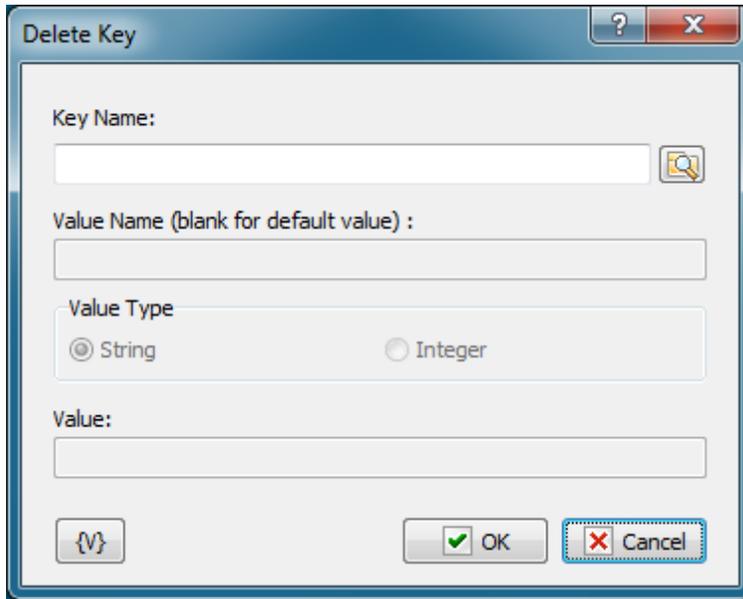
Name of the key to be added. You can use the dialog box to select the key interactively; to open it click the button to the right of the text box.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.17.2 Delete Key

Deletes the specified key (branch) from the registry. The specified key will be deleted together with all its subkeys and entries.



The image shows a Windows-style dialog box titled "Delete Key". It has a standard title bar with a question mark and a close button (X). The dialog contains the following fields and controls:

- Key Name:** A text input field with a magnifying glass icon to its right.
- Value Name (blank for default value):** A text input field.
- Value Type:** A group box containing two radio buttons: "String" (which is selected) and "Integer".
- Value:** A text input field.
- Buttons:** At the bottom, there is a button with a curly brace icon `{V}`, an "OK" button with a green checkmark, and a "Cancel" button with a red X.

**Key Name**

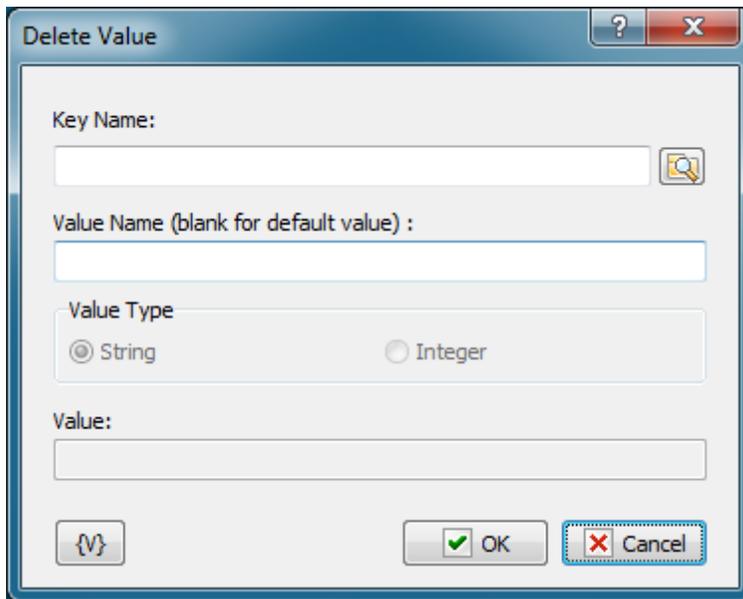
Name of the key to be deleted. You can use the dialog box to select the key interactively; to open it click the button to the right of the text box.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the `{V}` button.

### 4.17.3 Rename Key

Renames the specified key (branch) of the registry

**Key Name**

Name of the key to be renamed. You can use the dialog box to select the key interactively; to open it click the button to the right of the text box.

**New Key Name**

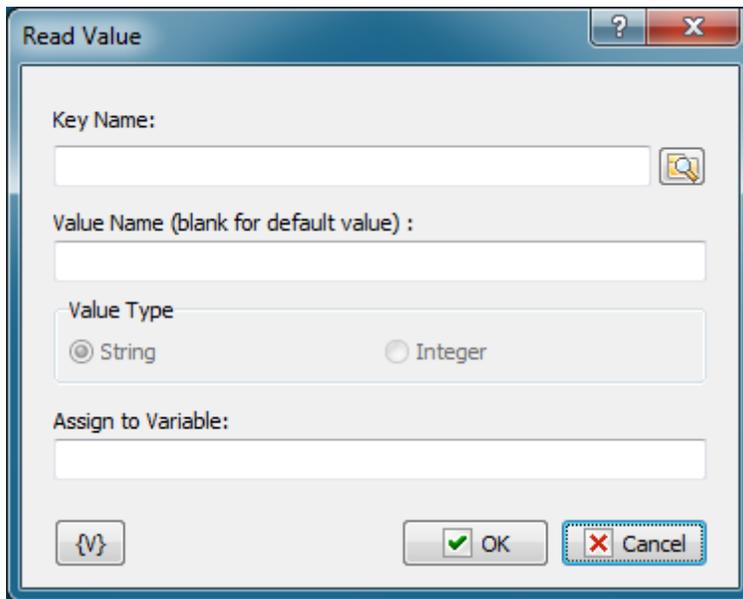
A new name that will be given to the specified key.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.17.4 Read Value

Read the specified value from the registry and assign it to the user variable. This action allows reading values of **String** and **Integer** types only.

**Key Name**

Full path to the value you want to read. You can use the dialog box to select the key and value interactively; to open it click the button to the right of the text box. When you click the OK button, the "Key Name" and "Value Name" fields will be filled, if you have selected the necessary value.

**Value Name**

Name of the value to be read. If you leave Value Name empty, RoboTask will read the default value of the key.

**Assign to Variable**

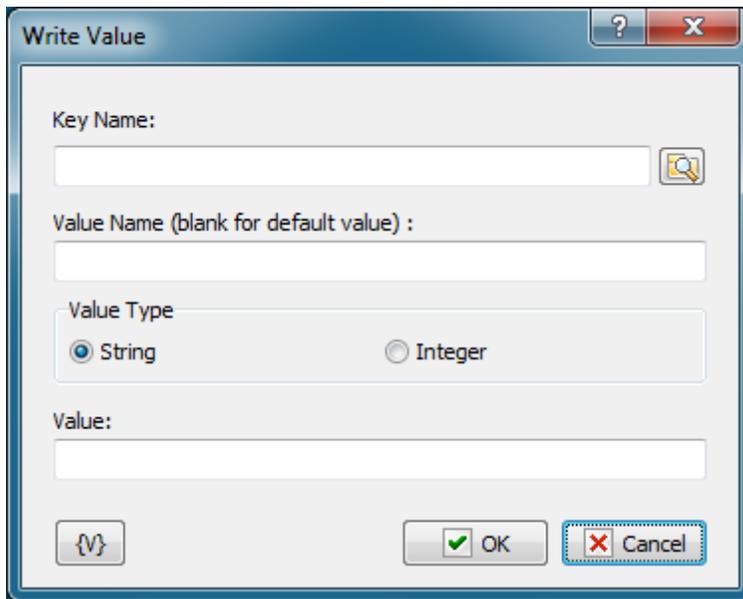
Name of the user variable that should be assigned the value we have read.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.17.5 Write Value

Write the value to the specified key in the registry. This action allows writing values of **String** and **Integer** types only.

**Key Name**

Full path to the value you'd like to write. You can use the dialog box to select the key and value interactively; to open it click the button to the right of the text box. When you click the OK button, the "Key Name" and "Value Name" fields will be filled, if you have selected the necessary value.

**Value Name**

Name of the value to be written.

**Value Type**

Select the type of value to be written: String or Integer.

**Value**

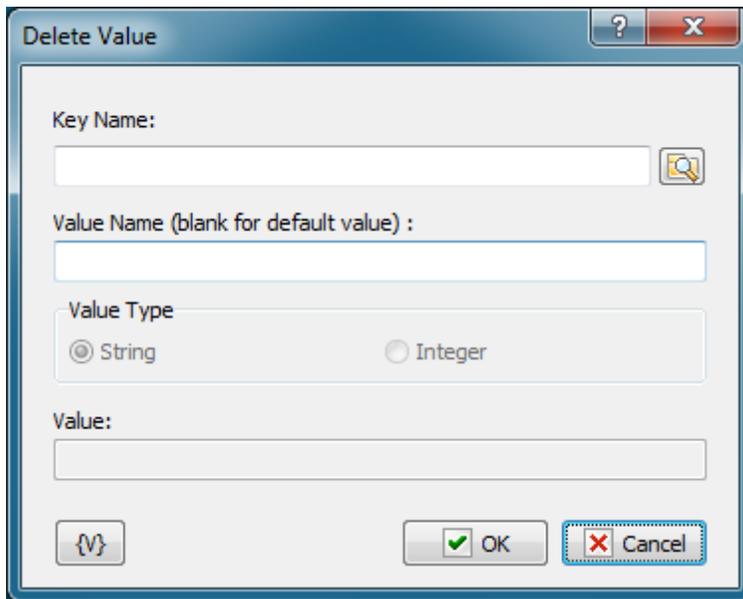
The value to be written.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.17.6 Delete value

Deletes the specified value from the registry.

**Key Name**

Full path to the value you'd like to delete. You can use the dialog box to select the key and value interactively; to open it click the button to the right of the text box. When you click the OK button, the "Key Name" and "Value Name" fields will be filled, if you have selected the necessary value.

**Value Name**

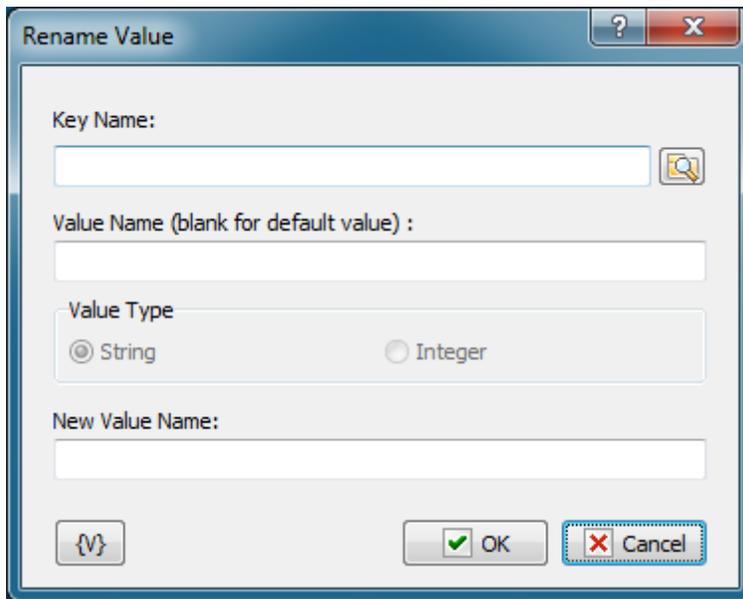
Name of the value to be deleted.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.17.7 Rename value

Renames the specified value in the registry.

**Key Name**

Full path to the value you'd like to rename. You can use the dialog box to select the key and value interactively; to open it click the button to the right of the text box. When you click the OK button, the "Key Name" and "Value Name" fields will be filled, if you have selected the necessary value.

**Value Name**

Name of the value to be renamed.

**New Value Name**

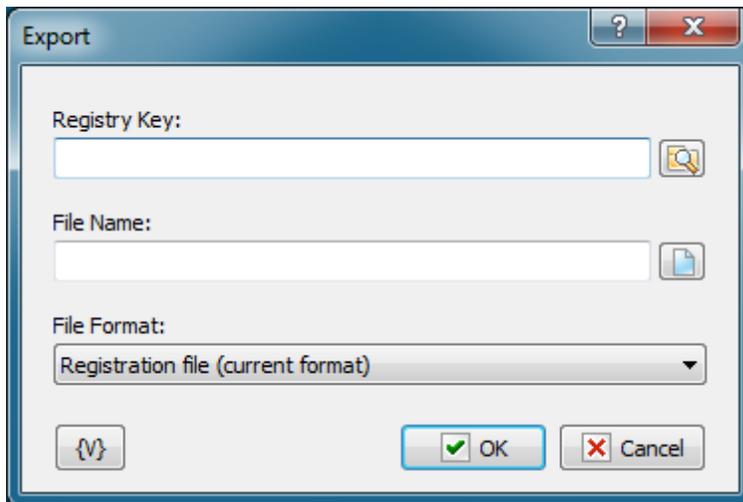
New name that will be assigned to the specified value.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.17.8 Export

Exports the specified registry branch to the specified file. The saved file has a format compatible with the **REGEDIT.EXE** system utility. This operation is useful when you make a backup copy of some information or move settings to another computer. You can easily import the export file using **REGEDIT** or the [Import](#)<sup>[157]</sup> action of RoboTask.



### Registry Key

The key (branch) of the registry that should be exported to a file. You can use the dialog box to select the key interactively; to open it click the button to the right of the text box.

### File Name

Name of the file where information should be exported.

### File Format

Choose the export file format.

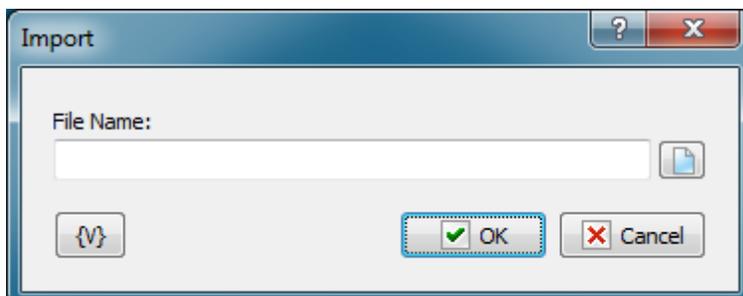
- Registration file (current format) - current default format for the installed operating system. For Windows 2000 and higher the format is Unicode text.
- Win9/NT4 Registration File - old format that provides compatibility with Windows 9x/NT4. This format has a structure of a regular text file.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.17.9 Import

Imports the export file to the registry. The export file should be prepared with the **REGEDIT.EXE** system utility or the [Export](#)<sup>[156]</sup> action.



**File Name.**

Name of the file to be imported.

**Variable**

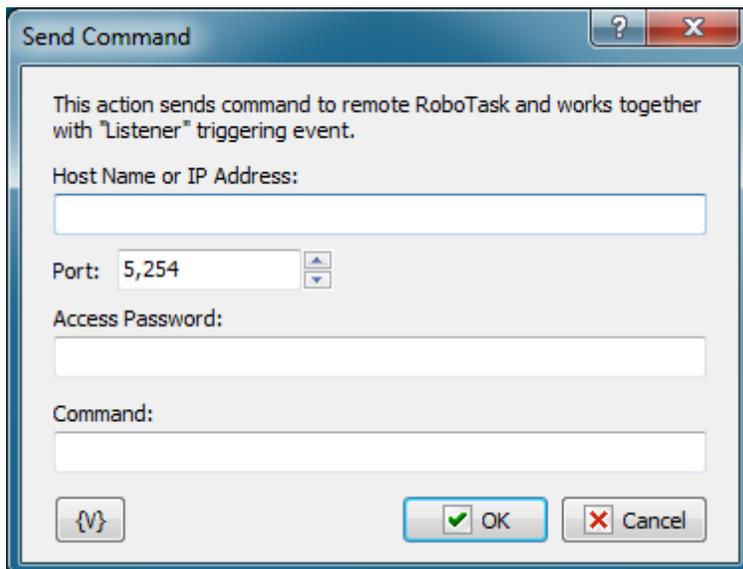
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.18 Network

### 4.18.1 Send Command

Allows you to send a string to a copy of RoboTask on another computer. This action works together with the ["Listener" triggering event](#)<sup>[220]</sup>. The Listener event receives the string and can start some task according to its settings. Thus, you can synchronize several copies of RoboTask working on the local area network of your office or via the Intranet/Internet.

*For example, it may be a command to pack IIS log files on the server and send the packed file by e-mail.*

**Host Name or IP Address**

Specify the Host Name or IP address of the computer where you need to send the command. The Host Name can be either similar to *www.mydomain.com* for the Internet or just the name of the necessary computer for the local area network.

**Port**

Specify the port the command should be sent to. By default, the "Listener" listens to port 5254. The port number can be changed to any other number (see Plugin settings).

**Access Password**

Specify the password for connecting to remote RoboTask. If the Listener on remote RoboTask is configured to work without a password, leave this field blank.

**Command**

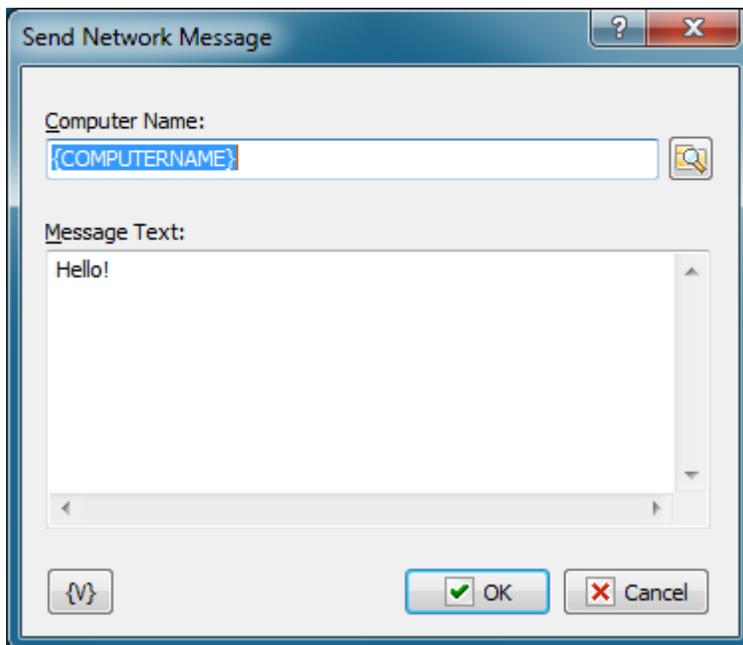
Specify the string (command) that should be sent to remote RoboTask. It can be any string. The "Listener" triggering event on the remote computer will analyze the string.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.18.2 Send Network Message

Sends a text message using a local network area (LAN) connection to the specified computer. For example, you can use this action to display a message every time you receive an e-mail from a special customer when you are logged into a different computer on your network and cannot check your e-mail.

**Computer Name**

Enter the name of the computer or domain name that the message should be sent to. When the "\*" symbol (without quotes) is specified, the message is broadcast to every computer in the primary domain of the system.

**Important**

*If the Windows Messenger service is not running on the target computer, the message will not be received.*

**Message Text**

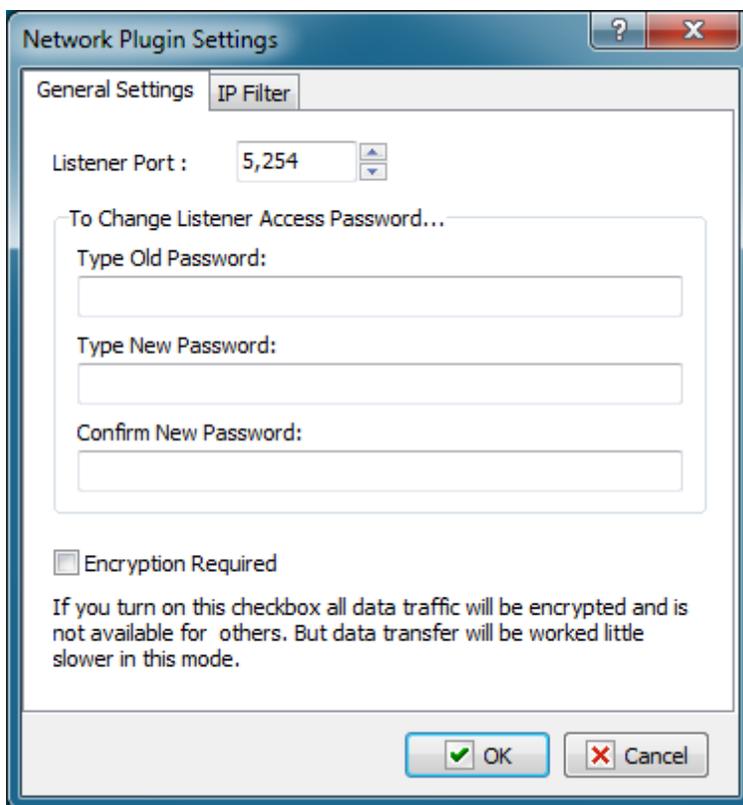
Specifies the text of the message to be sent.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.18.3 Network Plugin settings

The settings of the Network plugin are related to the "[Listener<sup>\[220\]</sup>](#)" triggering event and set the global parameters of the "[Listener<sup>\[220\]</sup>](#)".



### Listener port

Specify the port the "Listener" will listen to. The default is 5254, but it can be changed for any other available port.

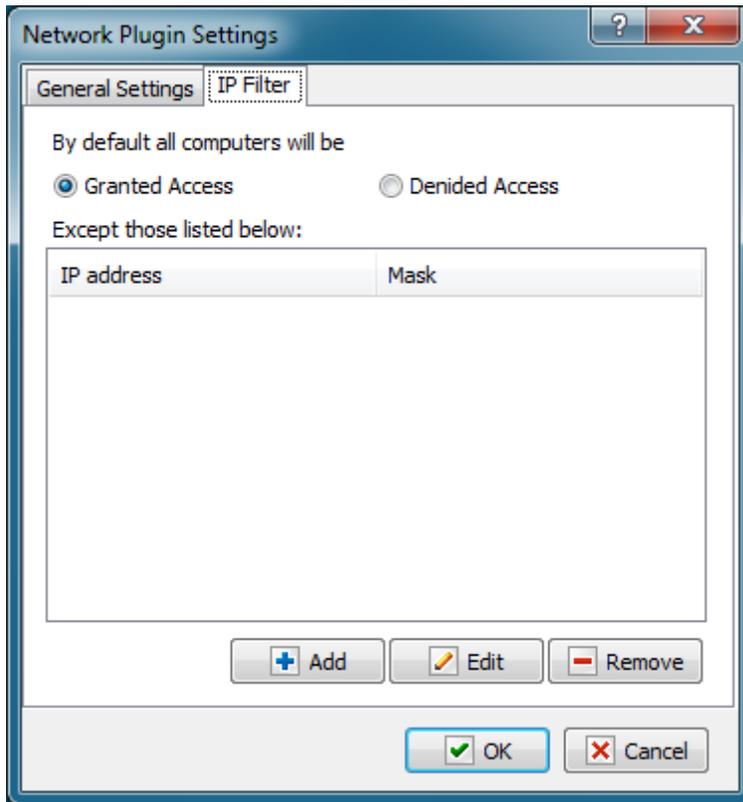
### To Change Listener Access Password...

To change the access password for the Listener, specify the old password and enter the new password twice. By default, the password is empty, i.e. the access is granted without a password.

### Encryption Required

Select this checkbox if you want to encrypt information when it is transferred between the computers. An 128-bit asymmetric encryption key is used for that. The key is generated

randomly for each session.



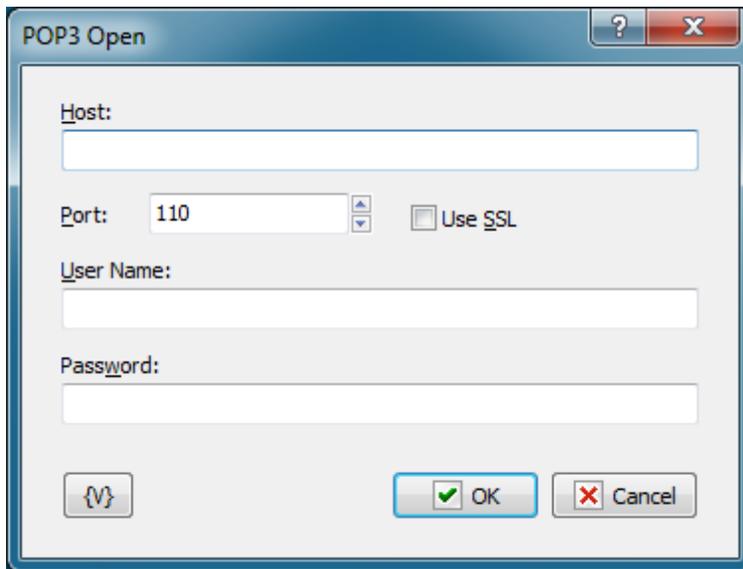
### IP Filter

You can use this tab to configure filtering requests by IP addresses. By default, the Listener is available for all IP addresses.

## 4.19 Email

### 4.19.1 POP3 Open

Opens a connection with the POP3 server. This action should be performed before you perform the "[POP3 Mail Loop](#)<sup>[162]</sup>", "[POP3 Download Email](#)<sup>[164]</sup>" and "[POP3 Delete Email](#)<sup>[165]</sup>" actions. There can be only one open connection with the POP3 server within a task. If you need to process several mail servers one by one, you should close the previous connection before opening the next one (see the "[POP3 close](#)<sup>[162]</sup>" action).

**Host**

IP-address or Host Name of the POP3 server.

**Port**

Port of the POP3 server. By default it's 110.

**Use SSL**

Select this checkbox if you want to use SSL connection.

**User Name**

Name of the mailbox user.

**Password**

User password.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

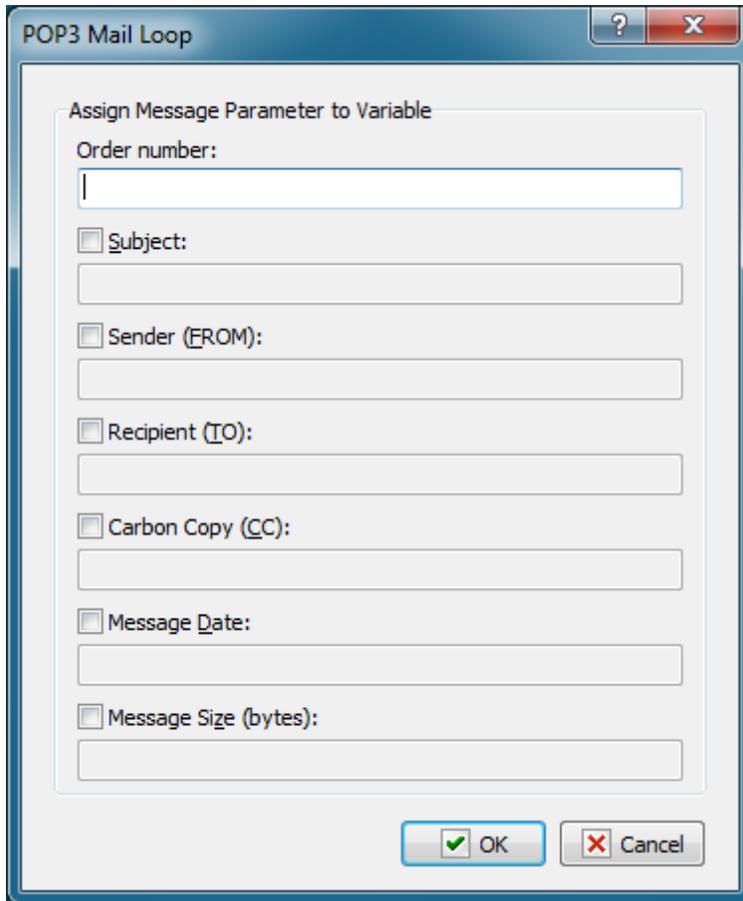
### 4.19.2 POP3 Close

Closes the open connection with the POP3 server. It doesn't require any additional parameters. When a task is finished, the opened server connection closes automatically.

### 4.19.3 POP3 Mail Loop

Allows processing messages on the POP3 server one by one in a loop. This action also allows parsing main parameters of a message without downloading the complete message from the

server. You can assign parameters of each message to a user variable and perform initial parse within a loop.



### Order Number

This loop parameter is compulsory as the current order number of a message serves as its identifier on the POP3 server. The numbering of messages starts with 1. The variable that contains the number of a message should be used in the loop in "[POP3 Download Email](#)<sup>[164]</sup>" and "[POP3 Delete Email](#)<sup>[165]</sup>" actions.

### Subject

Optional parameter. Allows saving the message subject to a variable and parsing it in the loop.

### Sender (FROM)

Optional parameter. Allows saving the sender address to a variable and using it within the loop.

### Receipt (TO)

Optional parameter. Allows saving the recipient address to a variable as a string and using it within the loop.

### Carbon Copy (CC)

Optional parameter. Allows saving the address of the recipient (or recipients) of a message copy to a variable as a string and use it within the loop.

### Message Size

Optional parameter. Allows saving the message size in bytes to a variable and using it within the loop.

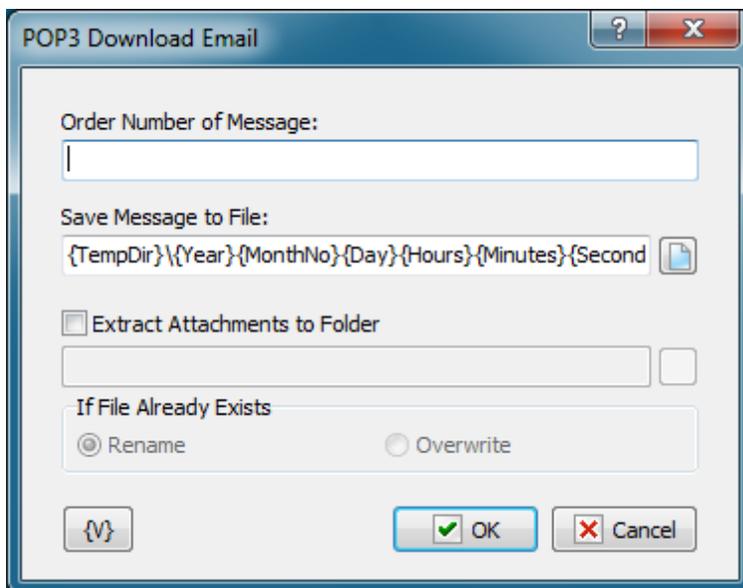
**Note:**

Addresses in the message can be written in form of:

- E-mail address only. For example:  
john@yahoo.com
- Recipient's name and address. For example:  
John Smith <john@yahoo.com>
- Several recipients separated with a semicolon. For example:  
John Smith <john@yahoo.com>; Melissa Carver <melissa@hotmail.com>

#### 4.19.4 POP3 Download Email

Receives a message from the POP3 server and saves it to a file.



**Order Number of Message**

Order number of the current message. It can be generated in the "POP3 mail loop" using a variable or you can set it manually. If the message number is invalid, an "Invalid number of message" error will occur when the action is carried out.

**Save Message to File**

Specify the name of file where the message should be saved. You can use variables to generate the file name. By default, the

`{TempDir}\{Year}{MonthNo}{Day}{Hours}{Minutes}{Seconds}{mSeconds}.eml`

string is specified. It provides a unique name and makes it more convenient to sort files by the time when the message was received.

### Extract Attachments to Folder

When you receive a message, you can extract all attachments to separate files. To do it, you should specify a folder where all attachments should be extracted.

### If File Already Exists

Specify an action that should be performed if the application file already exists.

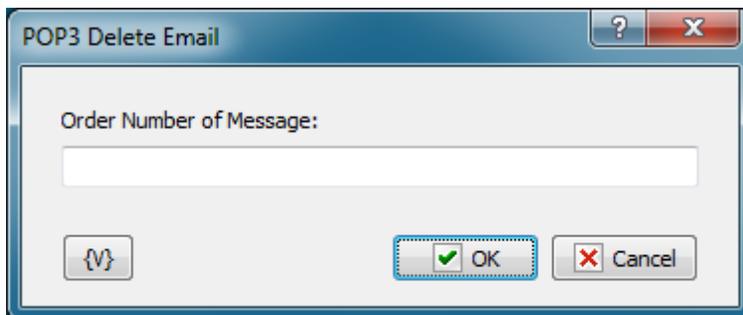
- Rename - in this case a suffix with an order number will be added to the file name;
- Overwrite - the file will be saved with the original name and the old file will be overwritten.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.19.5 POP3 Delete Email

Deletes the email message from the POP3 server.



### Order Number of Message

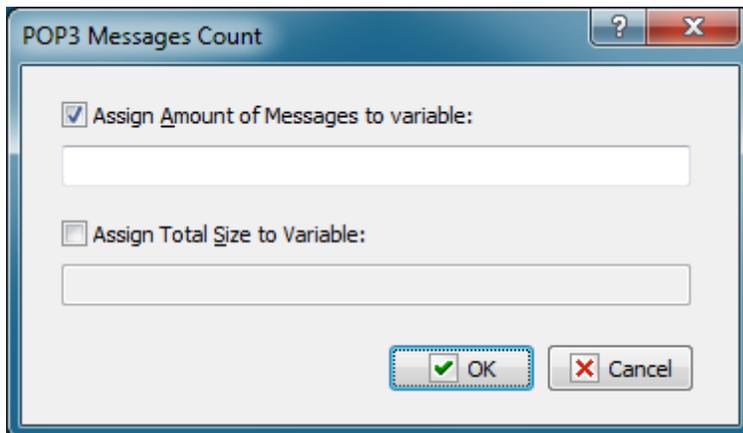
Order number of the message to be deleted. It can be generated in the "[POP3 mail loop](#)<sup>[162]</sup>" using a variable or you can set it manually. If the message number is invalid, a "No such message" error will occur when is action is carried out.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.19.6 POP3 Messages Count

It is used to count the number and total size of messages on the incoming mail server (POP3 server). To use this action, you should establish a connection with the POP3 server using the "[POP3 Open](#)" [action](#)<sup>[161]</sup>. You can save the number and total size of messages to custom variables to process them later in a task.



#### **Assign Amount of Messages to Variable**

Select this checkbox if you want to save the number of messages to a variable. You should enter the variable name into the field below.

#### **Assign Total Size to Variable**

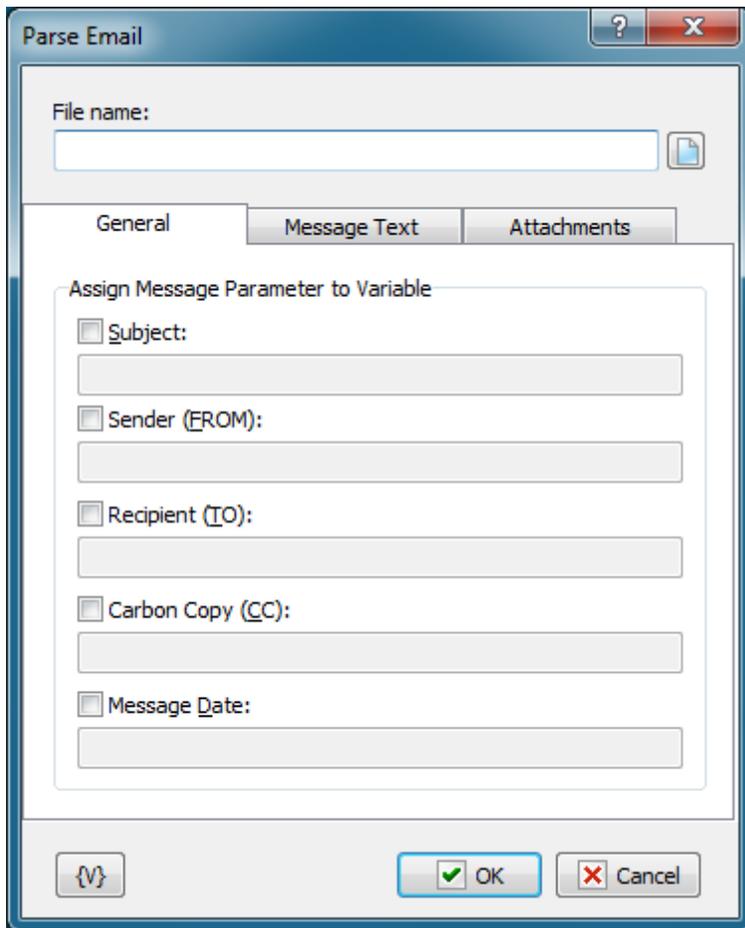
Select this checkbox if you want to save the total size of messages (in bytes) to a variable. You should enter the variable name into the field below.

While editing the step, you should specify at least one variable.

### **4.19.7 Parse Email**

Parses message in the file. It separates the message into its main parts:

- Message subject (the **SUBJECT** field);
- Sender (the **FROM** field);
- Recipient (the **TO** field);
- Recipient of a copy (the **CC** field);
- Message body (the text);
- Attachments.

**File Name**

Specify the filename of the message you want to parse. Such files usually have **EML** extension.

**General tab****Subject**

Specify the name of variable where the message subject should be saved.

**Sender (FROM)**

Specify the name of variable where the sender's address should be saved.

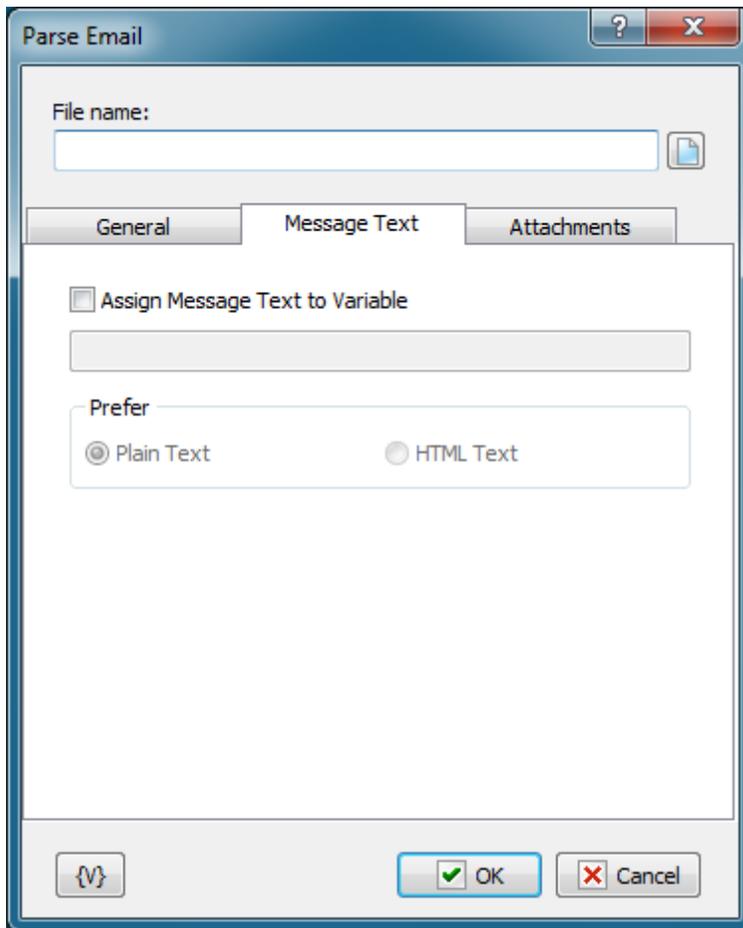
**Recipient (TO)**

Specify the name of variable where the recipient's address should be saved. This field can contain several addresses separated with a comma or a semicolon.

**Carbon Copy (CC)**

Specify the name of variable where the address of the recipient of a copy should be saved. This field can contain several addresses separated with a comma or a semicolon.

**Message Text tab**

**Assign Message Text to Variable**

Specify the variable where the message body should be saved.

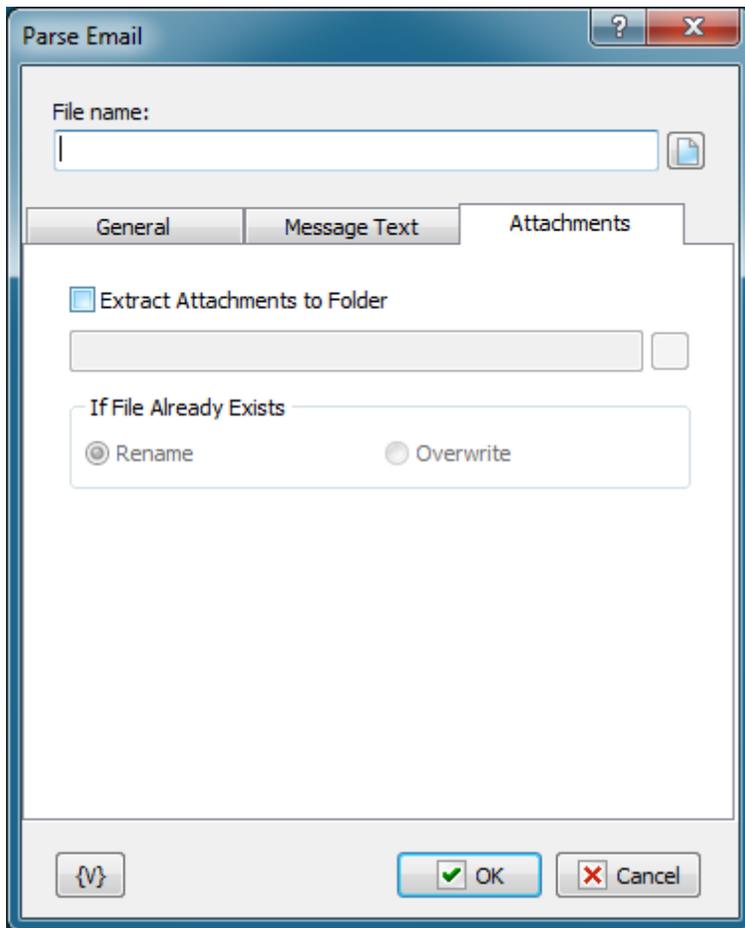
**Prefer**

Specify the format that should be preferred if the message body contains both message formats. The message body can contain the following formats:

- Plain Text
- HTML Text

If the message body contains one format only, this message text will be assigned to the variable no matter what value the Prefer parameter has.

**Attachments tab**



### Extract Attachments to Folder

Specify the folder where all attachments should be extracted.

### If File Already Exists

Specify an action that should be performed if the specified folder already contains a file with the application name.

- Rename - in this case a suffix with an order number will be added to the file name;
- Overwrite - the application will be saved with the original name and the existing file will be overwritten.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### Note:

Addresses in the message can be written in form of:

- E-mail address only. For example:  
john@yahoo.com
- Recipient's name and address. For example:  
John Smith <john@yahoo.com>

- Several recipients separated with a semicolon. For example:  
John Smith <john@yahoo.com>; Melissa Carver <melissa@hotmail.com>

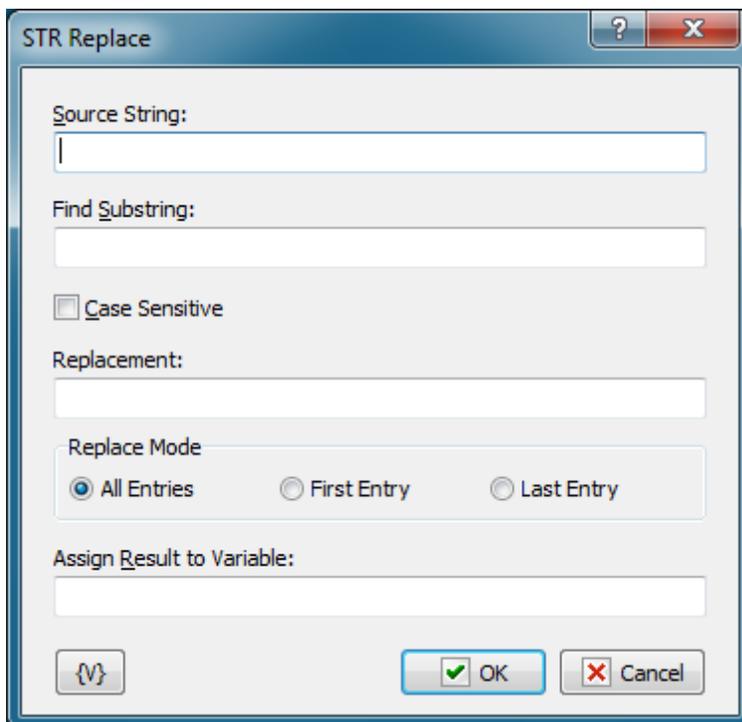
## 4.20 Strings

### 4.20.1 STR Replace

This action finds and replaces the specified string with the one you need. There are three replacement modes:

- All Entries;
- First entry;
- Last Entry.

The result is assigned to a user variable.



#### Source String

The source string to search and replace in. It is possible to use variables.

#### Find Substring

The substring that should be found and replaced. It is possible to use variables.

#### Replacement

The substring to replace the found substring with. It is possible to use variables.

#### Replace mode

Replacement mode. 3 options are possible:

- All Entries – all found substrings will be replaced;
- First Entry – only the first found substring will be replaced;
- Last Entry – only the last found substring will be replaced;

### Assign Result to Variable

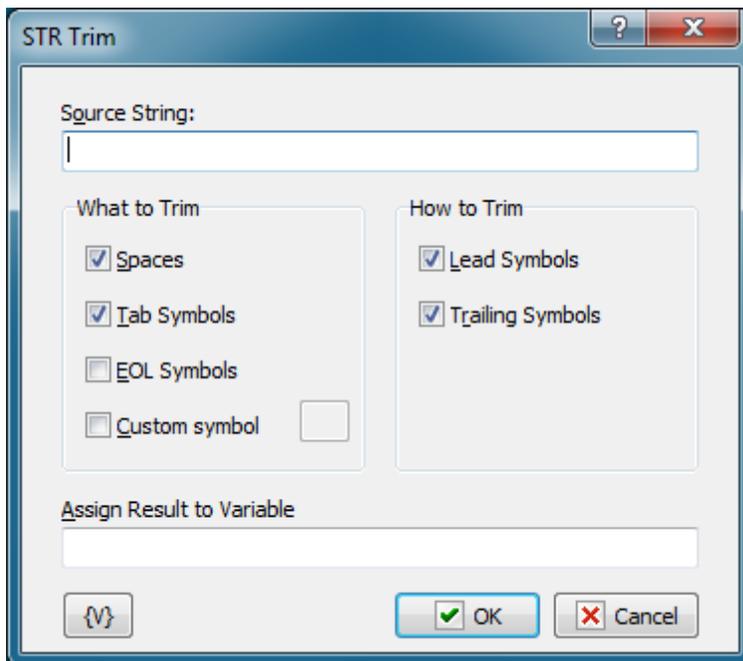
The name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.20.2 STR Trim

This action allows you to trim unnecessary characters at the beginning and at the end of the string.



### Source String

The source string to perform the action in. It is possible to use variables.

### What to Trim

The characters to trim.

- Spaces
- Tabs
- EOL symbols – end-of-line characters
- Custom symbol - you may specify any symbol

You can select the options independently of each other. It is enough to select at least one option.

### How to Trim

2 options are possible:

- Lead Symbols

- Trailing Symbols

You can select the options independently of each other. It is enough to select at least one option.

### Assign Result to Variable

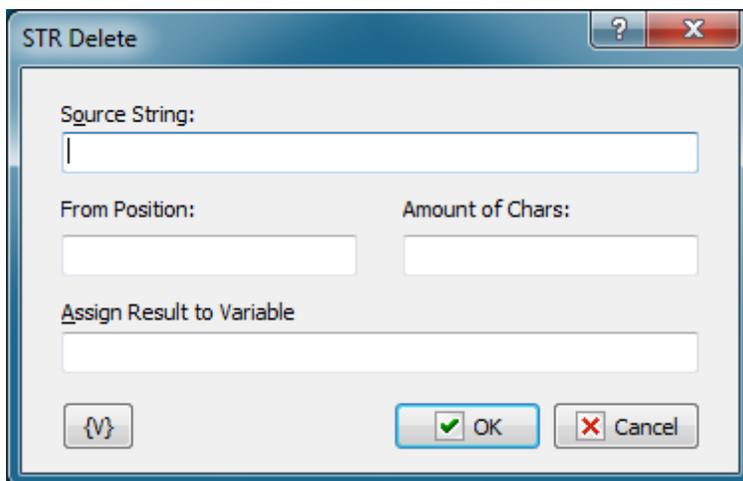
The name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.20.3 STR Delete

The action deletes the specified number of characters from the specified position in the source string.



### Source String

The source string to delete the characters in. It is possible to use variables.

### From Position

Specify the position to start deleting characters from. The value must be a positive integer. The positions of characters in a string start from 1. It is possible to use variables.

### Amount of Chars

Specify the number of characters that should be deleted starting from the specified position. The value must be a positive integer. It is possible to use variables.

### Assign Result to Variable

Specify the name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

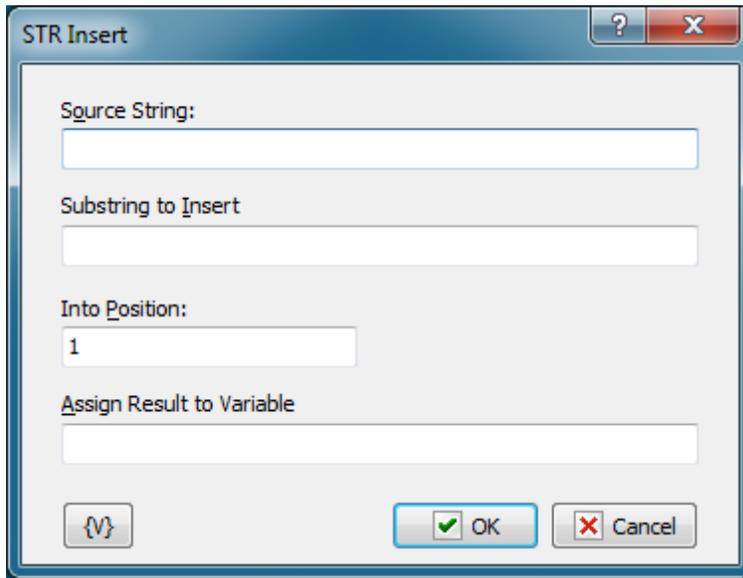
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then

click the {V} button.

#### 4.20.4 STR Insert

The action inserts a substring into the specified position in the source string.



##### **Source String**

The source string to insert the substring to. It is possible to use variables.

##### **Substring to Insert**

The substring to insert to the source string. It is possible to use variables.

##### **Into Position**

Insertion position. The positions of characters in a string start from 1. If you specify a position less than 1, the substring will be inserted at the beginning. If the position of the insertion is greater than the number of characters in the source string, the substring will be added to the end of the source string. It is possible to use variables.

##### **Assign Result to Variable**

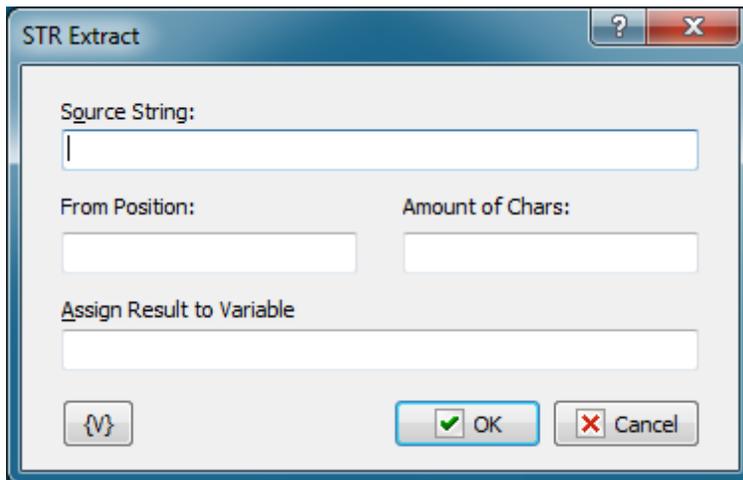
The name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

##### **Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.20.5 STR Extract

The action cuts the specified number of characters from the string starting from the specified position. The cut substring is assigned to a user variable.

**Source String**

The source string to cut the substring from. It is possible to use variables.

**From Position**

Specify the position to start cutting characters from. The value must be a positive integer. The positions of characters in a string start from 1. If the position is less than 1, the substring will be cut from the beginning of the source string. It is possible to use variables.

**Amount of Chars**

Specify the number of characters that should be extracted starting from the specified position. The value must be a positive integer. It is possible to use variables.

**Assign Result to Variable**

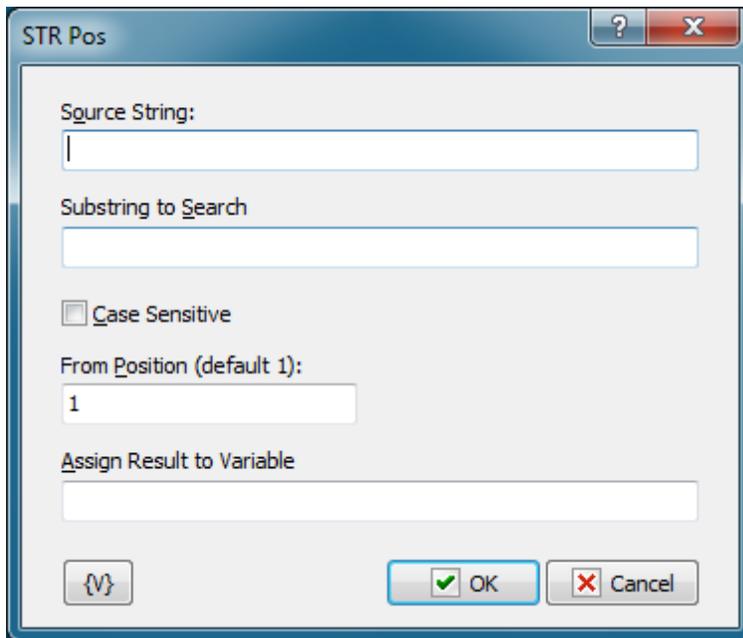
The name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.20.6 STR Pos

The action determines the position of the found substring in the source string. If the substring is found, the result will be positive. Otherwise, the result is 0.



#### Source String

The source string to find the substring in. It is possible to use variables.

#### Substring to Search

The substring to search for. It is possible to use variables.

#### From Position

The position to start searching from. Position 1 corresponds to the beginning of the string. It is possible to use variables.

#### Assign Result to Variable

The name of the variable the result of the action will be assigned to. The result is an integer. If the substring is found, the value is greater than 0. Otherwise, it is 0. It is impossible to use macros with variables.

#### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.20.7 STR Between

This action cuts a substring that is enclosed by two sequences of characters out of the source string.

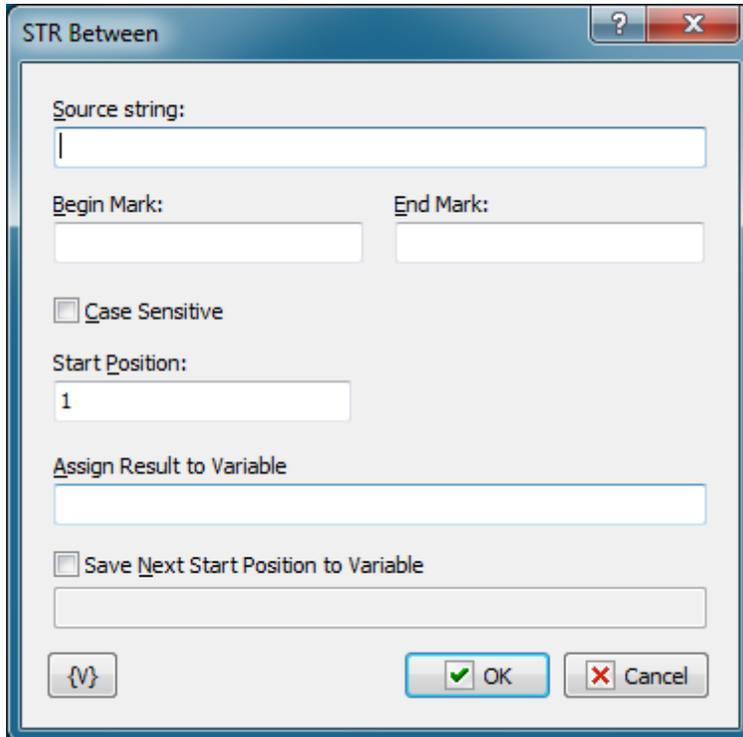
Suppose we have the following string:

```
<h3><b>Another news - at last week</b></h3>
```

We need to cut the meaningful string out, i.e. the string between the HTML tags. It is easy with the STR Between action. To do it, you should create the limiting sequences of characters:  
Start - **<h3><b>**

End - **</h3></b>**

The action cuts the string between the start and end marks out and assigns the result to a custom variable for further processing. If at least one mark is not found in the source string, the action will return an empty string.



#### **Source String**

Specify the source string to be processed. It is possible to use variables.

#### **Begin Mark**

The start mark of the substring. It is possible to use variables

#### **End Mark**

The end mark of the substring. It is possible to use variables

#### **Case Sensitive**

You can make the search for the substring case-sensitive. By default, the search for the substring is case-insensitive.

#### **Start Position**

The position of the character the search will be started from. Number 1 corresponds to the beginning of the string. If the number is greater than the length of the string, the action will return an empty result. It is possible to use variables.

#### **Assign Result to Variable**

Enter the name of the custom variable to assign the result to.

#### **Save Next Start Position to Variable**

If necessary, you can specify the name of the variable you can assign the start position for the next substring search to. If we reach the end of the string, the next start position will be equal to 0.

It is necessary, for example, in order to get all substrings meeting the specified criteria in the source string.

Suppose we need to get all substrings enclosed by parentheses in the following string:

```
Just (a few minutes) with RoboTask will give you (plenty of ideas) of how you can  
(increase) your productivity
```

You can easily do it using a loop. See the following example: "**[DEMO] STR Between - extract all entries**"

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

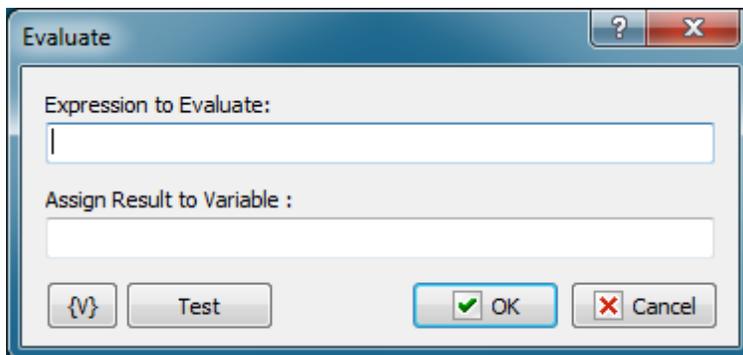
## 4.21 Basic

The **Basic** plugin is not included in the main installation package. You can download it additionally from <http://www.robotask.com/downloads/>.

### 4.21.1 Evaluate

Allows you to make calculations using the functions of mathematical operations and BASIC functions. The result is assigned to a user variable.

This action is convenient when you need to make some small calculations without writing a Basic script.



#### Expression to Evaluate

Mathematical expression. The result can be of any type. For more details on expression syntax see description of BASIC language.

For example:

```
{TotalSize}/{FileCount}  
Left(Right("{DateTime}",8),2)
```

### Assign Result to Variable

Specify the variable the result should be assigned to.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

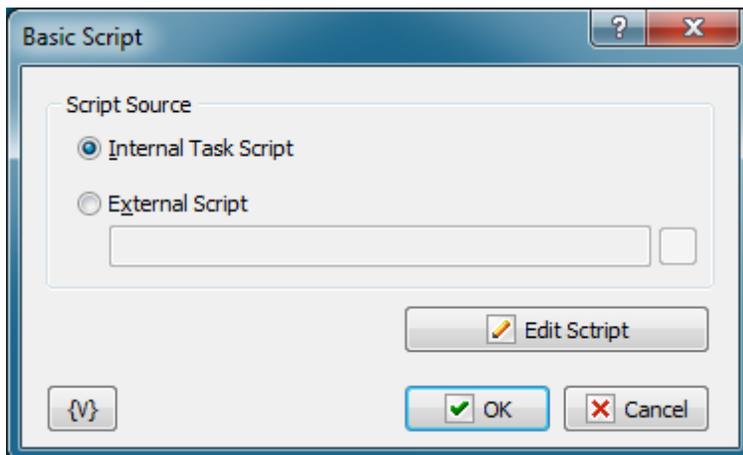
### Test

Click this button to ensure that the parameters for the current action work.

## 4.21.2 Basic script

It is a universal tool allowing you to execute scripts written in BASIC. Scripts can be virtually as complicated as you need. You can use such a script to access the RoboTaskApp COM object allowing you to access service functions of RoboTask, as well as other tasks. The description of the RoboTaskApp object is given below.

See additional information about the BASIC language in the BASIC help.



### Script source

You can select the script storage method

- **Internal** - the script is stored in the task file in this case. This storage method is good for small scripts consisting of one or several procedures.
- **External Script** - the script is stored in a separate file. This storage method is good for complex subroutines consisting of several modules (see the keyword "uses" or "include" in the Basic help).

## 4.21.3 RoboTaskApp object

There is no need to create the RoboTaskApp object when you are running or editing the script.

It is created before you run the script and is available as a global object inside the script.

**RoboTaskApp.AddUserVariable(By Val Name as String, By Val Value as String)**

Method.

Creates a user variable with the specified "*Name*" and assign the specified "*Value*" to it.

**RoboTaskApp.Automation**

Read-Write property, type - boolean

It is available for reading and writing. It contains the value of the RoboTask automation mode.

**RoboTaskApp.Close**

Method

Exits RoboTask.

**RoboTaskApp.DelUserVariable(By Val Name as String)**

Method.

Deletes the user variable with the specified "*Name*".

**RoboTaskApp.EventInfo(ByVal TaskNum as Long, ByVal EventNum as Long, EventID as String, Enabled as Boolean)**

Method.

Allows you to get information about the triggering event by the task number and the step number in the task

*TaskNum* - [in] task number

*EventNum* - [in] event number

*EventID* - [out] event type identifier (for example: E\_NET\_LISTENER, E\_WINDOW\_WATCHER, E\_GENERAL\_CYCLIC, etc.)

*Enabled* - [out] event status (enabled or disabled)

**RoboTaskApp.EventParameters(ByVal TaskNum as Long, ByVal EventNum as Long)**

Read-only property, type - String.

Returns the event parameters as text. Each text line is an expression of the following type:

ParameterName = ParamaterValue

The set of parameters depends on the event type (EventID)

*TaskNum* - [in] task number

*EventNum* - [in] the event number in the task

**RoboTask.EventsCount(ByVal TaskNum as Long)**

Read-only property, type - Long

Returns the amount of events of the task.

*TaskNum* - [in] task number

**RoboTaskApp.ExpandText(By Val Text as String)**

Read-only property, type - String.

Expands a string containing RoboTask variables and returns the value of the string type.

For example: the string "File c:\autoexec.bat has length {FileSize(c:\autoexec.bat)} bytes" will be turned into the string "File c:\autoexec.bat has length 153 bytes"

**RoboTaskApp.ExternalName(By Val ThreadID)**

Read-only property, type - String.

Returns the "External Name" of a task by its *ThreadID*. You can specify an External Name on the "Advanced" tab of the task editor. You can find out the ThreadID of the current task using the *RoboTaskApp.ThreadID* property. A sample use is given below.

```
Sub Main
    Dim extName As String
    extName = RoboTaskApp.ExternalName(RoboTaskApp.ThreadID)
    MsgBox("External name is "+ extName)
End Sub
```

### **RoboTaskApp.Minimize**

Method.

Minimizes RoboTask into the tray icon.

### **RoboTaskApp.Restore**

Method.

Restores the RoboTask window from the tray icon.

### **RoboTaskApp.ServiceMode**

Read-only property. It contains a Boolean value: "True" if RoboTask is running as an NT service; "False" if RoboTask is running as a regular Windows application.

### **RoboTaskApp.SetUserVariable(By Val Name as String, By Val Value as String)**

Method

Sets the value of the "Name" user variable to the specified "Value"

### **RoboTaskApp.StartTask(By Val ExternalName as String, By Val TaskParameters as String)**

Method

Starts the task with the specified "External Name".

*TaskParameters* - you can pass some parameters into started task. May be an empty string.

You need write down parameters like the text, each line should be like:

*ParameterName=ParameterValue*

- *ParameterName* - the name of parameter, any word consisting of letters and numbers;
- *ParameterValue* - value of parameter like any set of symbols.

### **RoboTaskApp.StepInfo(ByVal TaskNum as long, ByVal StepNum as long, ActionID as String, Enabled as Boolean, Name as String)**

Method.

Allows you to get information about the step using the task number and the step number in the task

*TaskNum* - [in] task number

*StepNum* - [in] step number

*ActionID* - [out] action type identifier (for example: A\_TASKS\_START, A\_TASK\_WAITCOMPETE, A\_GENERAL\_RUN\_PROG, etc.)

*Enabled* - [out] step status (enabled or disabled)

*Name* - [out] action name

### **RoboTaskApp.StepParameters(ByVal TaskNum as Long, ByVal StepNum as long)**

Read-only property, type - String

Returns the step parameters as text. Each text line is an expression of the following type:

*ParameterName = ParamaterValue*

The set of parameters depends on the action type (ActionID)

*TaskNum* - task number

*StepNum* - step number

**RoboTaskApp.StepsCount(ByVal TaskNum as long)**

Read-only property, type - Long

Returns the number of steps in the task by the task number

*TaskNum* - task number

**RoboTaskApp.StopTask(By Val ExternalName as String)**

Method.

Stops the task with the specified "*External Name*".

**RoboTaskApp.TaskCount**

Method.

The read-only attribute of the numerical type. Returns the number of tasks registered in RoboTask.

**RoboTaskApp.TaskEnabled(By Val ExternalName as String)**

A Boolean attribute available for reading and writing. Allows you to read or set the "Enabled" parameter of the task with the specified "*ExternalName*".

For example:

```
RoboTaskApp.TaskEnabled("Check_email") = False
```

**RoboTaskApp.TaskInfo(By Val NumTask as Long, ExtName as String, Name as String, ID as Long)**

Method.

Sets the properties of the task with the specified number ("*NumTask*").

*NumTask* - [in] task number

*ExtName* - [ou] External Name;

*Name* - [out] task name;

*ID* - [out] internal numerical identifier.

This method is used to enumerate all tasks and extract the parameter of each task. A sample use is given below.

```
Sub Main
    Dim cnt As Long
    Dim nam As String
    Dim extNam As String
    Dim id As Long
    Dim i As Long

    cnt = RoboTaskApp.TaskCount
    RoboTaskApp.WriteToLog(1,"Total "+Str(cnt)+" tasks",RoboTaskApp.ThreadID)

    For i = 0 To cnt - 1
        RoboTaskApp.TaskInfo(i,extNam,nam,id)
        RoboTaskApp.WriteToLog(3,extNam+"; "+nam+"; "+Str(id),RoboTaskApp.ThreadID)
    Next
    RoboTaskApp.WriteToLog(3,"Total " + Str(cnt) + " tasks",RoboTaskApp.ThreadID)
End Sub
```

**RoboTaskApp.TaskInfo2(ByVal NumTask as Long, FolderID as Long, State as TaskState)**

Method

An additional method for getting information about the task. It allows you to get the task folder identifier and the task status by the task number.

*TaskNum* - [in] task number

*FolderID* – [out] the identifier of the folder with the task.

*State* – [out] the task status at the moment when the method is called. It can take one of the following values:

- *tsManual* – 0, the task has no triggering events or its triggering events are disabled (see the *RoboTaskApp.TaskEnabled* property). It waits to be run manually or from another task.
- *tsIdle* – 1, the task has some enabled triggering events. It waits to be run automatically
- *tsDisabled* – 2, the task is disabled while it is being edited.
- *tsRunning* – 3, the task has been launched and it is running.
- *tsStopping* – 4, the task received the stop command, but it is yet running. After the task is completely stopped, its status becomes either *tsManual* or *tsIdle*.

### **RoboTaskApp.TaskNumByExternalName(ByVal ExternalName as String)**

Read-only property, type - Long

Returns the task number by *ExternalName*.

You can use the *TaskInfo* method to get *ExternalName* by the task number.

*ExternalName* – external task name. It is any string of characters. You can specify it while editing a task (see the "External Name" parameter on the "Advanced" tab of the task editor)

### **RoboTaskApp.ThreadID**

Read-only property, type - Long

Returns the thread ID of the task. This value is required in the *ExternalName* property and *WriteToLog* method.

### **RoboTaskApp.Version**

A read-only property of the string type. Contains the version number of RoboTask in it.

### **RoboTaskApp.WHandle**

A read-only property of the Long type. Contains the Handle of the main RoboTask window. This value is necessary for some Windows API functions to work properly (for example, *SendMessage*, *PostMessage*).

### **RoboTaskApp.WriteToLog(By Val MsgType as Integer, By Val MessageStr as String, By Val ThreadID as Long)**

Method

Records a message to the task log. To record a message to the system log of RoboTask, you should set the *TreadID* to 0.

*MsgType* – [in] message type: 0 – regular message, 1 – warning, 2 – error, 3 - user message;

*MessageStr* – [in] message string

*ThreadID* – [in] the Thread ID of the current task. It can be obtained using the *RoboTaskApp.ThreadID* property. If you set it to 0, the message will be recorded to the system log of RoboTask.

A sample use is given below.

```
Sub Main
    Dim thr As Long

    thr = RoboTaskApp.ThreadID
    RoboTaskApp.WriteToLog(3, "Hello World", thr)

End Sub
```

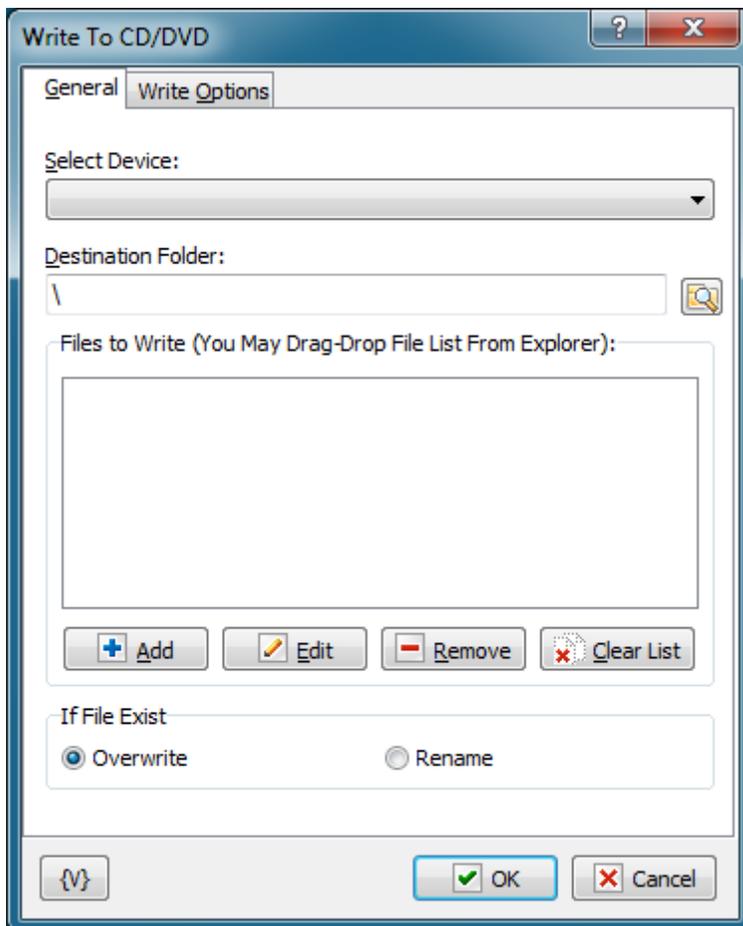
## 4.22 CD/DVD Writer

The **CD/DVD Writer** plugin is not included in the main installation package. You can download it additionally from <http://www.robotask.com/downloads/>.

### 4.22.1 Write to disk

Allows you to write files to a CD/DVD. This operation is very useful for automating backup processes.

#### General tab



#### Select Device

Use the list to select the device files will be written to.

#### Destination Folder

Specify the folder to write files to. You can use RoboTask variables. For example, it can be useful for backups if you specify the folder name based on the current date: `\{YEAR}\{MONTHNO}\{DAY}`

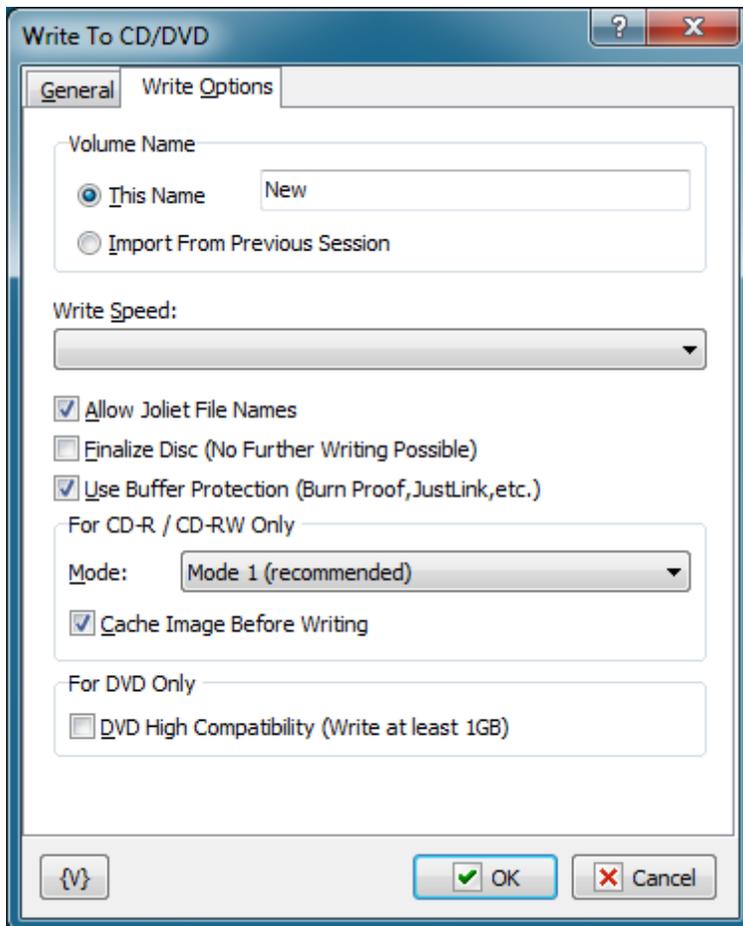
### Files to Write

Fill the list of files using the corresponding buttons. You can also fill it by dragging the necessary files and folders from a Windows Explorer window.

### If File Exists

Specify what to do if the files being written already exist on the disk.

## Write Options tab



### Volume Name

The name of the disk being burned. You can specify the name explicitly or inherit the name from the imported session.

### Write speed

Use the list to select the necessary record speed.

### Allow Joliet File Names

Select this checkbox if you want to enable long file names.

### Finalize Disk

Specify if it is necessary to finalize the disk. The disk will become read-only if it is finalized.

### Use Buffer Protection

Buffer protection. It is recommended to enable this option in order to ensure that the disk is burnt continuously in case of possible problems with reading the source files.

### For CD-R/CD-RW only

Specify the writing mode and whether it is necessary to cache information before writing it to the disk. These parameters make sense only when data is written to a CD-R/CD-RW and will be ignored when writing data to a DVD.

### For DVD only

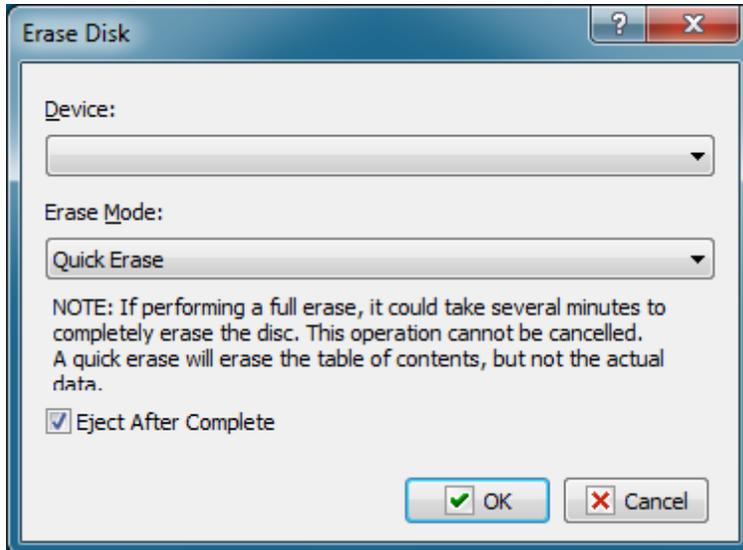
If it is known beforehand that the amount of the information is certainly more than 1 GB, you can enable the DVD High Compatibility option.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.22.2 Erase Disk

Allows you to erase the entire information from a CD/DVD RW.



### Device

Select a device.

### Erase Mode

Quick Erase and Full Erase modes.

If you use the Quick Erase mode, only the logical structure will be deleted without physically erasing the information from the disk. That is why the operation takes 1-1.5 minutes and the disk is considered empty and becomes available for writing.

In case with the Full Erase mode, the entire information is physically erased from the disk.

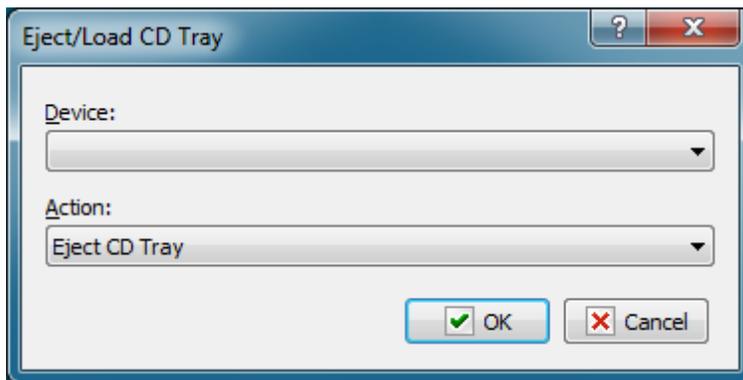
Depending on the drive and disk, such an operation may take 20 minutes or longer. This operation ensures that it will be impossible to restore any data from the disk.

### Eject After Complete

Select this checkbox if you want the drive to eject its tray after erasing is over.

## 4.22.3 Eject/Load CD Tray

Allows you to eject or load the tray of the CD drive.



### Device

Select a device.

### Action

Select whether to eject or load the tray of the CD drive.

## 4.23 Web Documents

Plugin for working with web pages and web forms.

The plugin contains two actions:

[Fill Web Form](#)<sup>[186]</sup>;

[Wait for done](#)<sup>[192]</sup>.

### 4.23.1 Fill Web Form

This action is destined for automation of filling web forms. Of course you can automate filling

forms by "[Send Keystrokes](#)<sup>[45]</sup>" and [mouse actions](#)<sup>[146]</sup>. But unfortunately, this is much more difficult and unreliable, because RoboTask tries to influence your external application (your web browser) and can't control the reaction of the program to this influence completely.

"Fill Web Form" works differently. It loads the web page with a form under its control, waits for a full loading of the page and can fill a form on the inside. If auto-submit is performed, the action will be waiting for the end of the resultant page loading. Thereby it is guaranteed that the task can pass to the next step only after the end of the page loading.

After filling the form you can keep on interactive working with the site (see the action "[Wait For Done](#)<sup>[192]</sup>") or close embedded Web browser.

The embedded web browser will be in of memory only during the work of a task. RoboTask will unload the object "embedded web browser" out of memory after the end of the task or in case an error appears.

## General options

The screenshot shows the "Fill Web Form" dialog box. It has a title bar with "Fill Web Form" and standard window controls. The dialog is divided into several sections:

- General options:** Includes tabs for "Autologon" and "Auto Download File".
- Web Address (Empty for Current Form):** A text input field with a "Load" button to its right.
- Form to Fill:** A dropdown menu.
- Manual Filling:** A text box containing the instruction "You can fill up the form manually in embegged web browser and press button 'Fill the List' -->" and a "Fill the List" button.
- Form Filling Parameters:** A table with columns "Field name", "N...", "Field Type", and "Value". To the right of the table are three buttons: a plus sign (+), a pencil (edit), and a red X (delete).
- Checkboxes:** "Autosubmit the Form" (checked) and "Keep Page for Next Actions" (unchecked).
- Bottom Bar:** Contains a code icon, "OK" and "Cancel" buttons.

### Web Address

Fill the web address of the page which needs loading. You can leave its field empty if you want to use the page which was loaded in the previous steps of the task. You can use variables.

On the right of the field there is the "Load" button. When you press it, the specified address will be opened in embedded web browser. After loading, there will be available the lists of forms, elements of a form and the lists of item values kind of "Select" and "Multiselect".

After the form loading into embedded web browser, all the visual elements of all forms at the

page will have pop-up hints, where the next information will be specified:

- The name of a form
- The name of the field
- The common ordinal number of the field
- The type of the edit field

### Form to Fill

The name of a form that you should fill. You can choose the name of a form from the drop-down list if you have loaded the page. You can also fill the name of the form manually. You can use variables.

There may be nameless forms on any pages. This is allowed by the HTML standard. In this case you can refer to the form according to an ordinal number in the list beginning from 0. In other words, the first form on the page will have the number 0, the second – 1, etc.

### Fill the form

After loading the web page in embedded web browser, you can fill the necessary form on the page manually, then press the button "Fill the List". After confirmation, an editor will fill the field list from the chosen form.

After auto filling, you can edit the list manually: to add/delete elements or to change the elements that are existing.

### Form Filling Parameters

A parameter list for a form. You can edit a list using buttons on the right. (See "[Add or Edit Web Form Parameter dialog](#)"<sup>[190]</sup>)

### Autosubmit the Form

Turn on this checkbox if you want to submit a form automatically after filling. By default this parameter is turned on. If auto-submit is performed, an action will be waiting for finishing loading of the resultant page. Otherwise, you need to submit manually. In this case the next step of the task must be the action "Wait for Done" and then performing necessary actions manually. This can be useful if you need to input the value CAPTCHA before pressing the button "Submit" or to perform any other interactive actions.

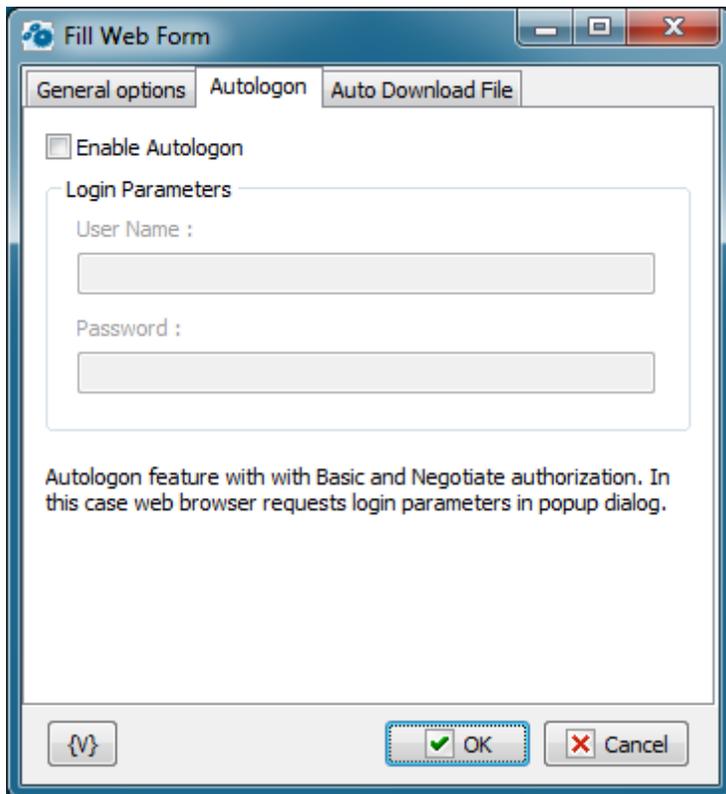
### Keep Page for Next Actions

Turn on this checkbox if you want to leave the resultant page for interactive work or for filling the next form. By default, this parameter is off. If this parameter is turned off, embedded web browser will be closed immediately after the end of the step.

### Autologon

You can specify parameters of automatic authorization on this tab if WEB browser demands authorization for access to resource. This way of automatic authorization works only with Basic and Digest authorization modes. In this case browser demands to input a user name and a password into the pop-up system window. If you specified parameters of automatic authorization, the action puts the parameters automatically when browser demands authorization.

If it is necessary to input the authorization parameters into the form on a page, then it is the so-called cookie authorization. In this case you should use the automatic filling of WEB form.



### Enable Autologon

It switches on the mode of automatic authorization using the specified User Name and Password. If the mode of automatic authorization is switched off, authorization occurs in a usual mode.

#### User name

A name of a user for authorization. Variables can be used.

#### Password

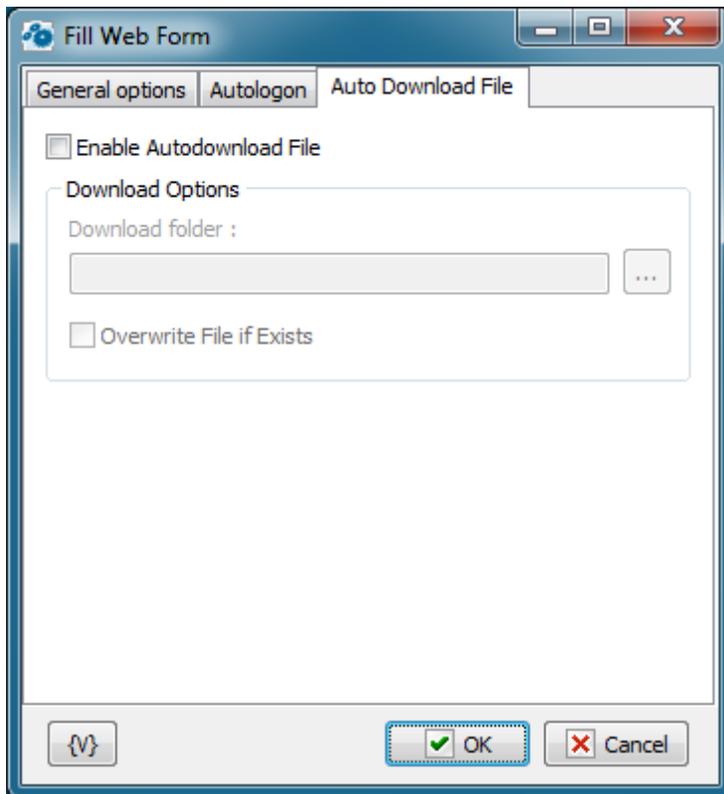
A password for authorization. Variables can be used.

### Auto Download file

On this tab you can tune parameters of automatic file downloading from a site.

For example, this mode is useful for automation of file downloading from the zone protected by a password. It occurs approximately in such a way:

- You specify URL of a file.
- When you try downloading a file, the browser demands a password or redirects the request to the Login form subject to the authorization mode on the site.
- When a password is requested, it is necessary to set up Autologon. When the request is redirected to a form, you should perform the automatic filling of the form.
- After you input parameters of authorization, the file will be downloaded into the specified folder automatically.



### Enable Auto-download File

It switches on the mode of automatic file downloading. If auto-download is off, downloading occurs in a usual mode requesting a folder and a file name.

### Download Folder

Specify the folder in which the file will be downloaded automatically. It will be downloaded with the file name by default.

### Overwrite File If Exists.

Switch on this checkbox if you want the action to overwrite a file, on condition there is a file with the same name in this folder. Otherwise the downloaded file will be renamed.

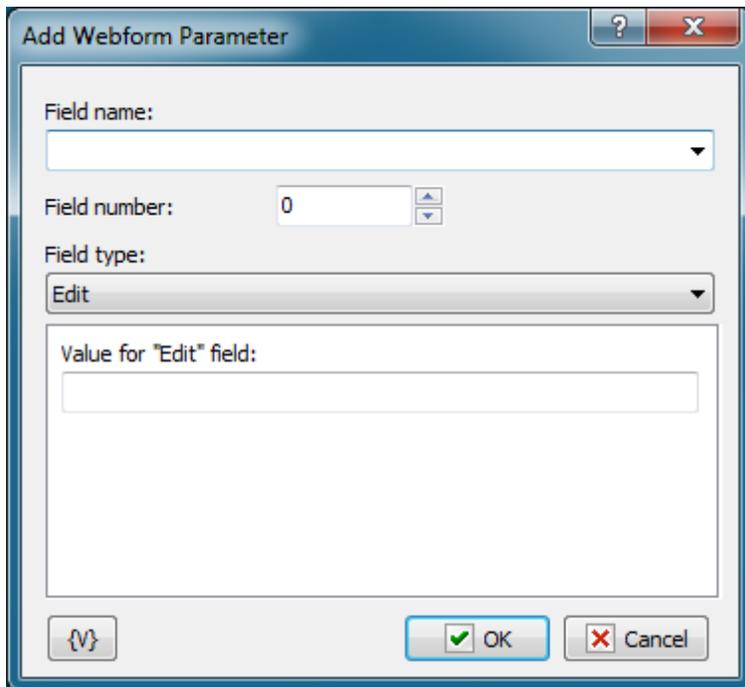
### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 4.23.1.1 Add or Edit Web Form Parameter dialog

Every parameter has four components for filling a form:

- Field name
- Field number
- Field Type
- Field value



### Field Name

The name of the field that needs filling with a new value. You can input the name manually or choose it from dropdown list. Pay attention that the dropdown list will be filled only if you have loaded the web-form into embedded web browser.

### Field Number

HTML standard allows of presence of several fields with the same name on a form. In this case, you need to specify the field number among the fields with the same name. Numeration begins from 0. If fields on the form have unique names the number of each field will be 0.

### Field Type

For right filling a field you need to specify its type. It is supported seven types in all;

- **Edit** – a simple single-line string field;
- **Memo** – multi-line text;
- **Select** – choice one option among the several from the proposed list;
- **Multiselect** – choice several options from the proposed list;
- **Check** – usually checkbox;
- **Radio** – radio button;
- **Hidden** – a hidden field of a form. This field is recommended to change only if you imagine clear, why you need to change values. As a rule, hidden fields are necessary for keeping various service information of a field.

Field Type will take the necessary value automatically when you choose the name of a field from a list (see the Field Name)

### Editing value of a field

View of a field value editor changes subject to its type

- **Edit** – input a field value. You can use variables.
- **Memo** – input .a text in the field. You can use variables
- **Select** – input a line (or part of a line), that need doing chosen. You can use variables. If the form is loaded, you can choose the necessary line from a dropdown list.
- **Multiselect** – input lines (or parts of lines), that need doing chosen. You can fill the list using available options pressing the button "Fill Up by Available Options", and then delete the needless.
- **Check** – choose one value form two – Unchecked and Checked
- **Radio** – choose one value form two – Unchecked and Checked
- **Hidden** – input a field value. You can use variables.

### Variable

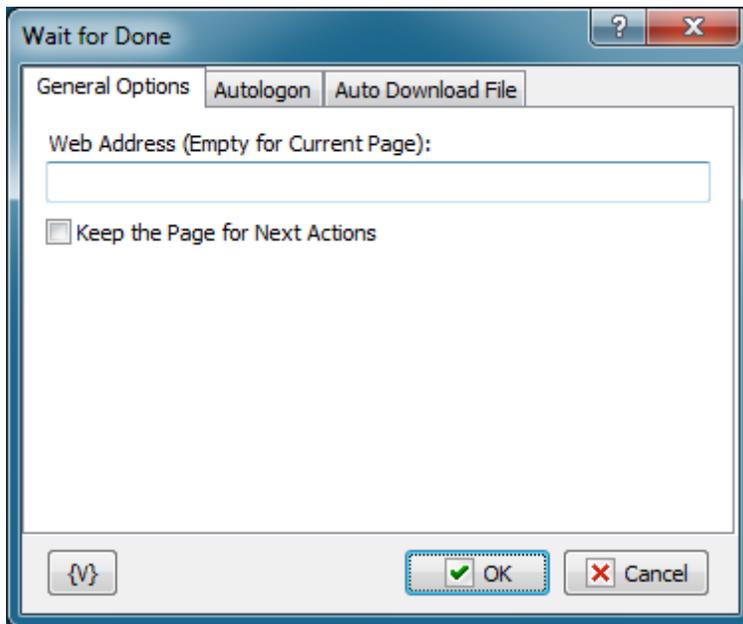
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.23.2 Wait for Done

This action is destined for free work with a web page. The action stops work of task until a user presses the button "Done" in the window Embedded Web Browser.

During of work of the action, buttons of navigation "Go Back" and "Go Forward" and button "Done" appear in the window Embedded Web Browser. The action is useful when the interactive interference of a user is needed. For example, to fill the value of CAPTCHA. Any interactive work with a web page is allowed: click on links, filling and submit of forms, etc. After finishing the interactive work it is necessary to press the button "Done".

### General options



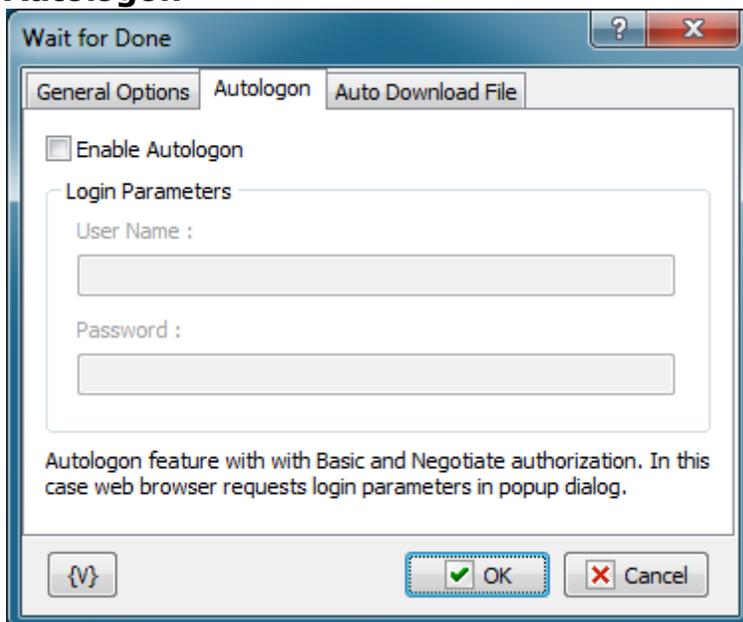
### Web Address

Fill the web address of the page that need loading. You can leave a field empty if you want to use the current page that has been loaded during the previous steps of the task. You can use variables.

### Keep Page for Next Actions

Turn this checkbox on if you want to leave the resultant page for interactive work or for filling the next form. By default, this parameter is turned off. If this parameter is turned off, Embedded Web Browser will be closed immediately after the finish of work of a step.

### Autologon



### Enable Autologon

It switches on the mode of automatic authorization using the specified User Name and Password. If the mode of automatic authorization is switched off, authorization occurs in a usual mode.

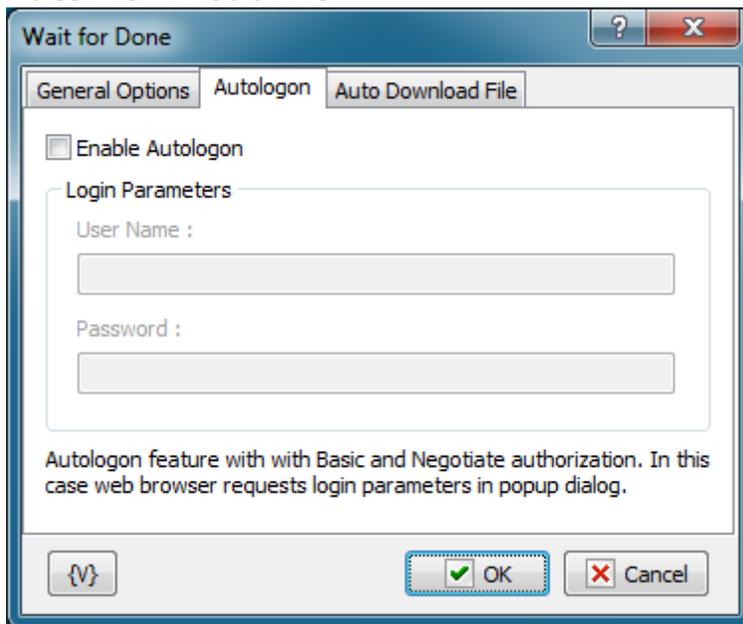
#### User name

A name of a user for authorization. Variables can be used.

#### Password

A password for authorization. Variables can be used.

### Auto Download file



### Enable Auto-download File

It switches on the mode of automatic file downloading. If auto-download is off, downloading occurs in a usual mode requesting a folder and a file name.

#### Download Folder

Specify the folder in which the file will be downloaded automatically. It will be downloaded with the file name by default.

#### Overwrite File If Exists.

Switch on this checkbox if you want the action to overwrite a file, on condition there is a file with the same name in this folder. Otherwise the downloaded file will be renamed.

### Variable

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.24 Regular Expressions

This plugin is for experts and connoisseurs of regular expressions. Regular expressions are very powerful mechanism for searching and replacing expressions in a text.

We give neither description nor syntax of regular expressions here because they would occupy a lot of space. Many books are devoted to regular expressions. Our description of them would virtually copy other sources.

If you read the book ***Mastering Regular Expressions*** by **Jeffrey E.F. Friedl** you will know everything about regular expressions (or almost everything). For quick study visit the site <http://www.regular-expressions.info/> . The site is completely devoted to regular expressions and different dialects. You can also find short manuals on syntax of regular expressions there. For example:

<http://www.regular-expressions.info/reference.html>

<http://www.regular-expressions.info/refadv.html>

<http://www.regular-expressions.info/refreplace.html>

You can find links to other editions on regular expressions here:

<http://www.regular-expressions.info/books.html>

In a few words, by means of regular expressions you can seek a phrase (an expression) in a text using an expression mask. You write a mask of an expression you seek and then the engine of regular expressions will find real phrases in a text which corresponds to the mask. Search of a substring (or of a symbol combination) in a text is the simplest example of using regular expressions.

### 4.24.1 RegExp Loop

Using RedExp Loop you can find all entries of a phrase you seek in a text and process them in a loop. A phrase mask is specified by a regular expression. The search result of every iteration is saved in a variable as a standard comma-separated string. The whole sought-for phrase is always a zero element. The following elements contain sub-expressions if they are present in a regular expression. Sub-expressions are a part of phrase which is put into parentheses. A sub-expression number is the ordinal number of an opening parenthesis if you read the expression from the left to the right.

For example, the regular expression for search of links addresses in HTML text is:

```
<a\s+href\s*=\s*"([\s])*">\s*(<?[^<>]*>?)\s*</a>
```

This regular express contains two sub-expressions:

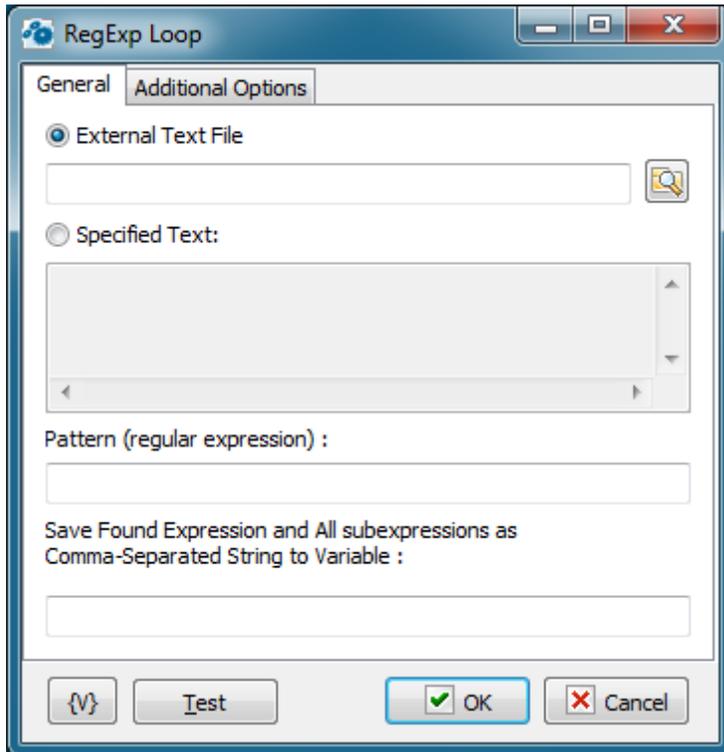
- The link address
- The link description

You can put the modifier which is not a sub-expression in the beginning of expression: (?i).

This combination means the search has to be done with no account taken to the register. The symbol ? in modifier always follows an opening parenthesis but goes before a closing one.

For example: (?im). Modifiers can be in any place of a regular expression. See below the description of search modes and corresponding modifiers.

## General



### External Text

Input the name of the text file in which the search will be implemented. Variables can be used.

### Specified Text

In this field specify the text in explicit form on which you want to search. You can use variables.

### Pattern (Regular Expression)

Input a regular expression (a phrase mask) for the search in a text. You can find description of regular expression syntax and manuals on the site <http://www.regular-expressions.info/>

### Save Found Expression to Variable

Input the name of a variable in which you want to save the search result. If the search-for phrase is found, the search result will be saved in the variable like a standard comma-separated string. To extract a separate element you need to refer to the variable as an array.

For example:

```
{MyVar(0)}
```

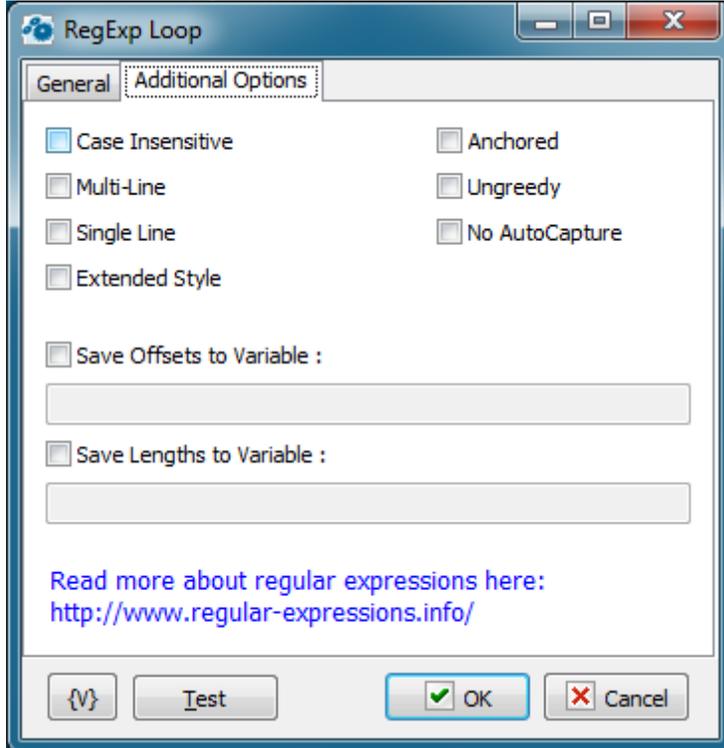
```
{MyVar(1)}
```

Etc.

The zero element of array is always the search-for phrase as a whole.

The next array elements are sub-expressions in the search-for phrase which are highlighted in the regular expression by parenthesis (certainly, if there are such). If you didn't specify any sub-expressions in the regular expression, the resultant array would consist of a single element, which is zero element.

## Additional options



### Case Insensitive

Tries to match the regex without paying attention to case. If set, 'Bye' will match 'Bye', 'bye', 'BYE' and even 'byE', 'bYe', etc. Otherwise, only 'Bye' will be matched. Equivalent to (?i) modifier. The modifier (?-i) switches off the mode case insensitive.

### MultiLine

The ^ (beginning of string) and \$ (ending of string) regex operators will also match right after and right before a newline in the Subject string. This effectively treats one string with multiple lines as multiple strings. Equivalent to Perl's (?m) modifier. The modifier (?-m) switches off the mode multiline.

### Single Line

Normally, dot (.) matches anything but a newline (\n). With Single Line mode, dot (.) will match anything, including newlines. This allows a multiline string to be regarded as a single entity. Equivalent to Perl's (?s) modifier. The modifier (?-s) switches off the mode single line.

Note that MultiLine and Single Line can be used together.

### Extended Style

Allow regex to contain extra whitespace, newlines and Perl-style comments, all of which will be filtered out. This is sometimes called "free-spacing mode".

### Anchored

Allows the regex to match only at the start of the subject or right after the previous match.

### Ungreedy

Repeat operators (+, \*, ?) are not greedy by default (i.e. they try to match the minimum number of characters instead of the maximum).

### No Autocapture

Is a non-capturing group, only named groups will be captured.

### Save Offsets to Variables

If you want to determine positions of found expressions from the beginning of a text switch on this flag and input a variable name, the positions will be saved as a comma-separated string. A zero element - this is a position of a whole expression. The next elements are positions of subexpressions. The beginning of a text corresponds to the position 1.

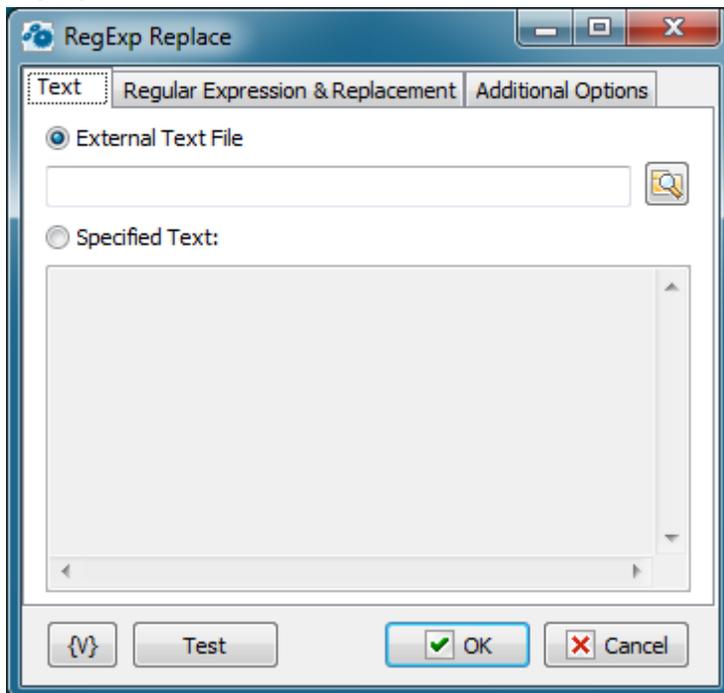
### Save Lengths to Variables

If you want to determine the lengths of found expressions (by symbols), switch on this flag and input the variable name. The lengths will be saved as a comma-separated string. A zero element is a length of a whole expression. The next elements are lengths of subexpressions.

## 4.24.2 RegExp Replace

This action performs search and replacement of regular expressions in a text. You can both replace the found expression entirely and combine a new one from the found subexpressions.

### Text



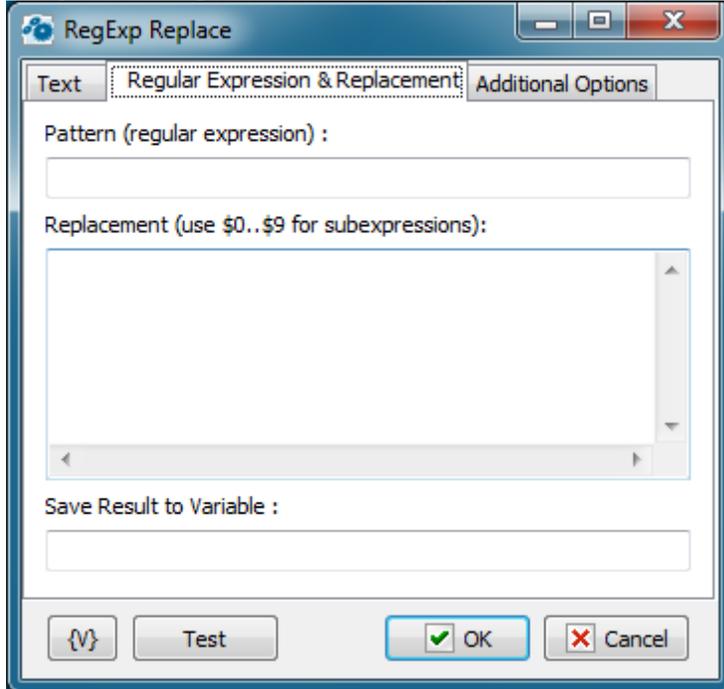
#### External Text

Input the name of the text file in which the search will be implemented. Variables can be used.

#### Specified Text

In this field specify the text in explicit form on which you want to search. You can use variables.

## Regular Expression & Replacement



### Pattern (Regular Expression)

Input a regular expression (a phrase mask) for the search in a text. You can find description of regular expression syntax and manuals on the site <http://www.regular-expressions.info/>

### Replacement

Input a text for replacement of the found expression. Variables and the found subexpressions can be used in the expression.

\$0 corresponds to the whole found expression.

\$1-\$9 are subexpressions from 1 to 9, one after another.

*For example:*

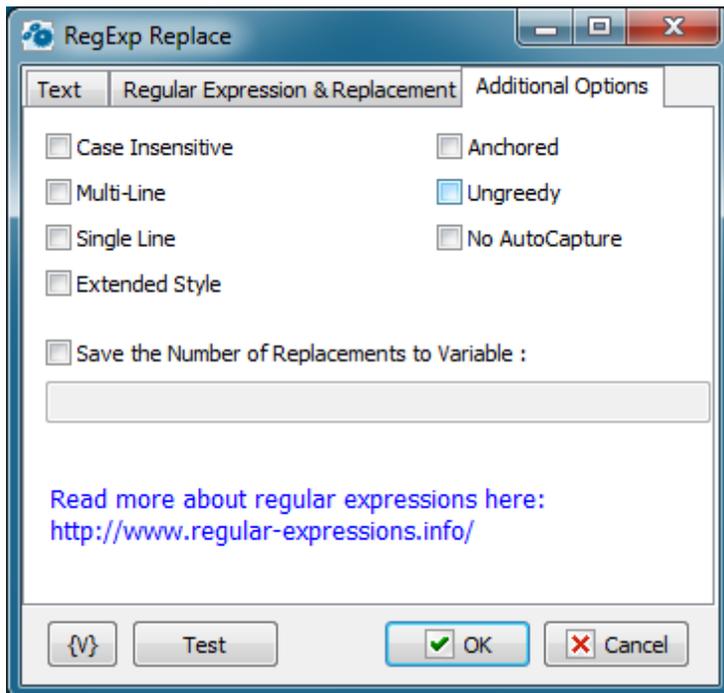
*If you want to put found expression into quotes write the following replacement:*

*"\$0"*

### Save Result to Variable

Input a variable name in which you want to put a text after performed replacements.

## Additional options



### Case Insensitive

Tries to match the regex without paying attention to case. If set, 'Bye' will match 'Bye', 'bye', 'BYE' and even 'byE', 'bYe', etc. Otherwise, only 'Bye' will be matched. Equivalent to (?i) modifier. The modifier (?-i) switches off the mode case insensitive.

### MultiLine

The ^ (beginning of string) and \$ (ending of string) regex operators will also match right after and right before a newline in the Subject string. This effectively treats one string with multiple lines as multiple strings. Equivalent to Perl's (?m) modifier. The modifier (?-m) switches off the mode multiline.

### Single Line

Normally, dot (.) matches anything but a newline (\n). With Single Line mode, dot (.) will match anything, including newlines. This allows a multiline string to be regarded as a single entity. Equivalent to Perl's (?s) modifier. The modifier (?-s) switches off the mode single line.

Note that MultiLine and Single Line can be used together.

### Extended Style

Allow regex to contain extra whitespace, newlines and Perl-style comments, all of which will be filtered out. This is sometimes called "free-spacing mode".

### Anchored

Allows the regex to match only at the start of the subject or right after the previous match.

### Ungreedy

Repeat operators (+, \*, ?) are not greedy by default (i.e. they try to match the minimum number of characters instead of the maximum).

### No Autocapture

Is a non-capturing group, only named groups will be captured.

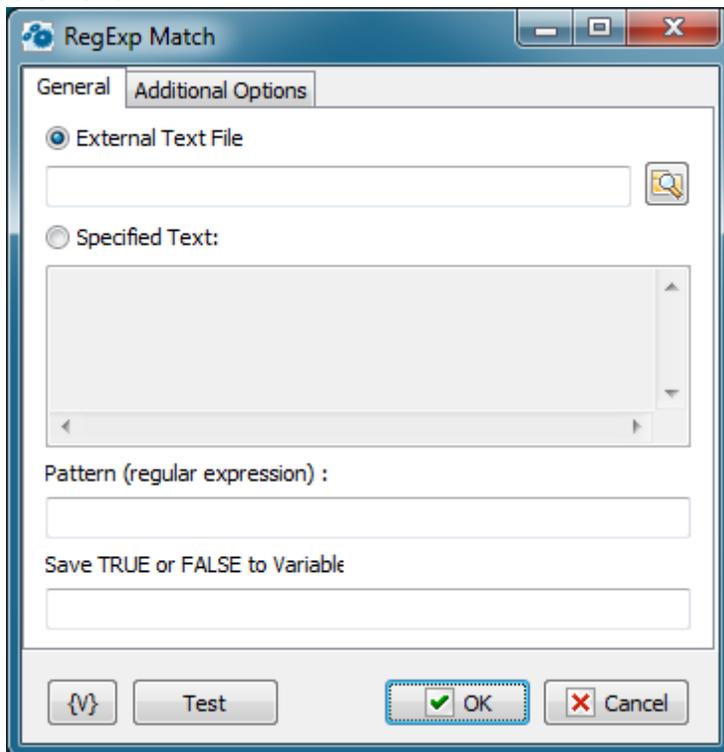
### Save the number of Replacements to Variable

If you need to save the number of performed replacements, switch on this checkbox and then input the variable name.

### 4.24.3 RegExp Match

This action is in fact a simplified version of RegExp Loop. By means of this action it is possible to determine if a string (or a text) corresponds to the specified regular expression and also to extract the first or the last entry.

#### General



#### External Text

Input the name of the text file in which the search will be implemented. Variables can be used.

#### Specified Text

In this field specify the text in explicit form on which you want to search. You can use variables.

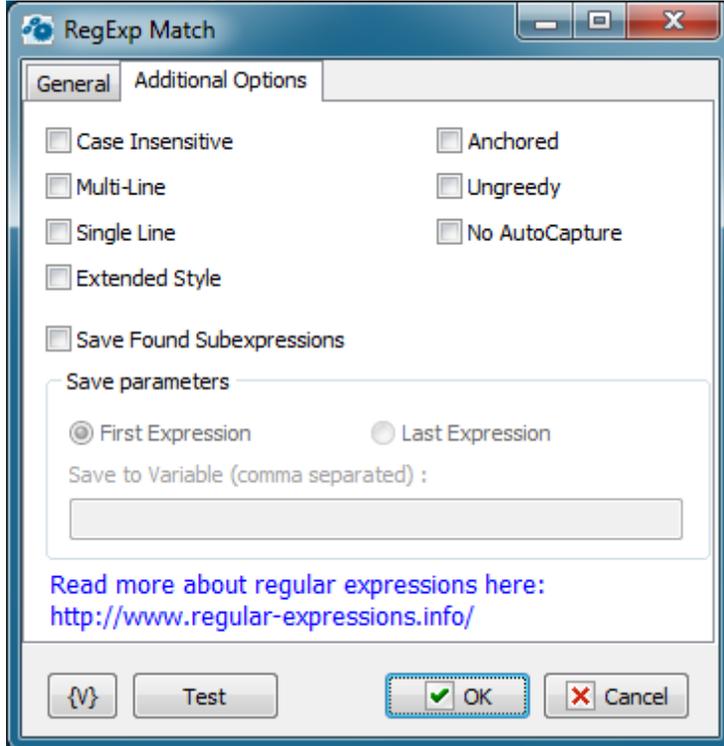
#### Pattern (Regular Expression)

Input a regular expression (a phrase mask) for the search in a text. You can find description of regular expression syntax and manuals on the site <http://www.regular-expressions.info/>

#### Save TRUE or FALSE to variable

Input the variable name in which you want to save the check result.

## Additional options



### Case Insensitive

Tries to match the regex without paying attention to case. If set, 'Bye' will match 'Bye', 'bye', 'BYE' and even 'byE', 'bYe', etc. Otherwise, only 'Bye' will be matched. Equivalent to (?i) modifier. The modifier (?-i) switches off the mode case insensitive.

### MultiLine

The ^ (beginning of string) and \$ (ending of string) regex operators will also match right after and right before a newline in the Subject string. This effectively treats one string with multiple lines as multiple strings. Equivalent to Perl's (?m) modifier. The modifier (?-m) switches off the mode multiline.

### Single Line

Normally, dot (.) matches anything but a newline (\n). With Single Line mode, dot (.) will match anything, including newlines. This allows a multiline string to be regarded as a single entity. Equivalent to Perl's (?s) modifier. The modifier (?-s) switches off the mode single line.

Note that MultiLine and Single Line can be used together.

### Extended Style

Allow regex to contain extra whitespace, newlines and Perl-style comments, all of which will be filtered out. This is sometimes called "free-spacing mode".

### Anchored

Allows the regex to match only at the start of the subject or right after the previous match.

### Ungreedy

Repeat operators (+, \*, ?) are not greedy by default (i.e. they try to match the minimum number of characters instead of the maximum).

### No Autocapture

Is a non-capturing group, only named groups will be captured.

### Save Found Subexpressions

If you need to save a found expression in a variable, switch on this checkbox and then fill in the saving parameters. You can save either the first entry or the last one. The found expression with its subexpressions will be saved as a comma-separated string. A zero element is the whole expression. The next elements are subexpressions in order.

#### 4.24.4 RegExp COM object

RegExp COM object is designed for using in Basic script.

Object name: RegExpCOM.RegexpClass

Object GUID: ['{0762F9AA-BD80-4BFC-B469-854866F114CC}']

#### Example of using

Below is an example of using RegExp COM object in Basic script

```
Sub Main

Dim obj As Object
Dim i As Long
Dim cnt As Long
Dim res As Boolean
Dim s As String

Set obj = CreateObject( "RegExpCOM.RegexpClass" )

obj.Subject = RoboTaskApp.ExpandText("{txt}")
obj.RegExp = RoboTaskApp.ExpandText("{regex}")

res = obj.Match
While res
    cnt = obj.SubExpressionCount
    s = ""
    For i = 0 To cnt
        s = s + obj.SubExpression( i ) + Chr(10)
    Next i
    MsgBox( s )
    res = obj.MatchAgain
Wend

End Sub
```

Also you can make permanent link to RegExpCOM Library in the script. In this case you should declare the object so:

```
Dim obj As RegExpClass
```

And create the object by using New directive

```
Set obj = New RegExpClass
```

#### Members of object

**Anchored**

Read-Write property, type – Boolean

Get or set anchored mode of object.

**Caseless**

Read-Write property, type – Boolean

Get or set case insensitive mode of object.

**Extended**

Read-Write property, type – Boolean

Get or set extended mode of object.

**Match**

Read-only property, type – Boolean

Find first match. Returns TRUE if expression was found.

**MatchAgain**

Read-only property, type – Boolean

Find next match. Returns TRUE if expression was found.

**Multiline**

Read-Write property, type – Boolean

Get or set multiline mode of object.

**NoAutoCapture**

Read-Write property, type – Boolean

Get or set NoAutoCapture mode

**RegExp**

Read-Write property, type – String

Get or set regular expression.

**Replace**

Read-only property, type - String

It replaced found expression by "replacement" and returns new expression.

This property can be used only after successful *Match* or *MatchAgain*.

**ReplaceAll**

Read-only property, type – Long

Replace all matches. Returns the number of replacements.

**Replacement**

Read-Write property, type – String

The replacement string. Used in *Replace* and *ReplaceAll*.

**SingleLine**

Read-Write property, type – Boolean

Get or set single line mode of object.

**SubExpression(ByVal idx as Long)**

Read-only property, type- String

Returns sub-expression by index.

This property can be used only after successful *Match* or *MatchAgain*.

**SubExpressionCount**

Read-only property, type- Long

Returns the number of sub-expressions.

This property can be used only after successful *Match* or *MatchAgain*.

**SubExpressionLength(ByVal idx as Long)**

Read-only property, type – Long

Returns the length of sub-expression by index.

This property can be used only after successful *Match* or *MatchAgain*.

**SubExpressionOffset(ByVal idx as Long)**

Read-only property, type – Long

Returns the offset (position) of sub-expression by index.

This property can be used only after successful *Match* or *MatchAgain*.

**Subject**

Read-Write property, type – String

Get or set the text to process.

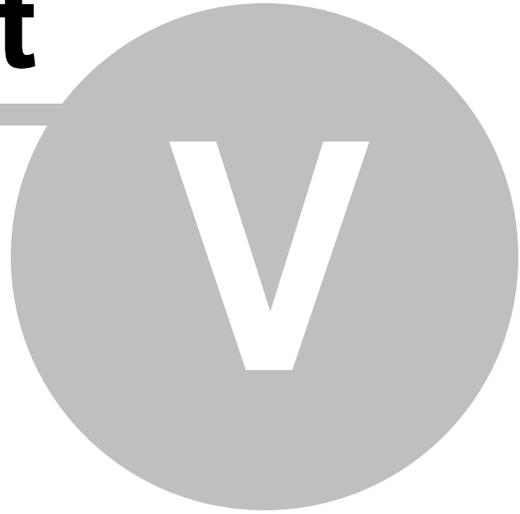
**Ungreedy**

Read-Write property, type – Boolean

Get or set un-greedy mode of object.

# Part

---

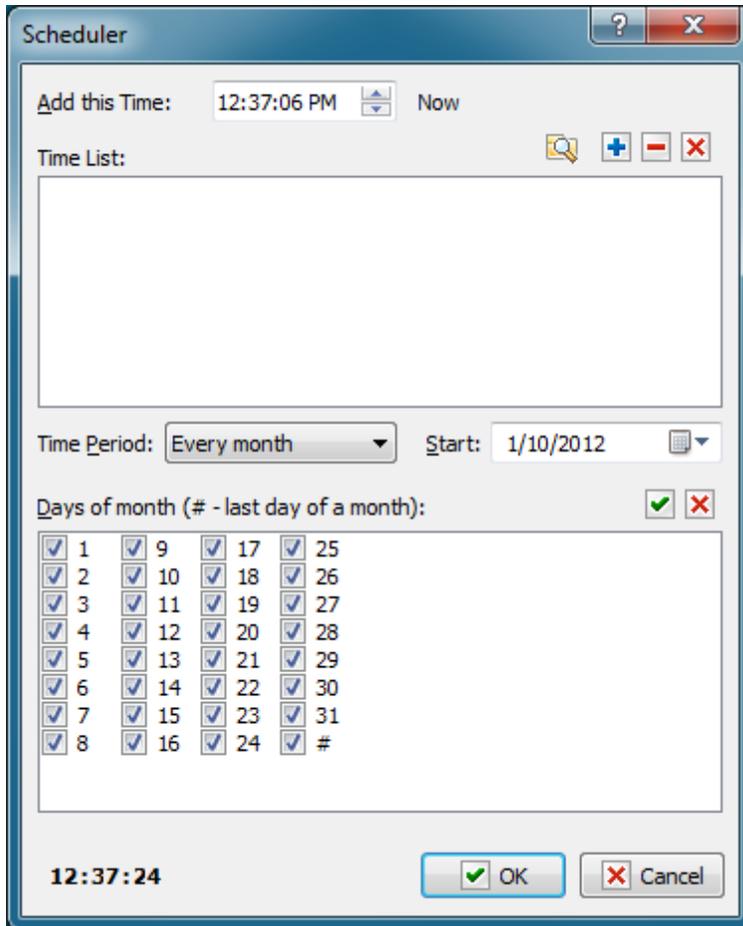


## Triggering Events

## 5 Triggering Events

### 5.1 Scheduler

Executes a task at a given time or interval. For example, you can use this triggering event to ensure your backup runs every day at midnight.

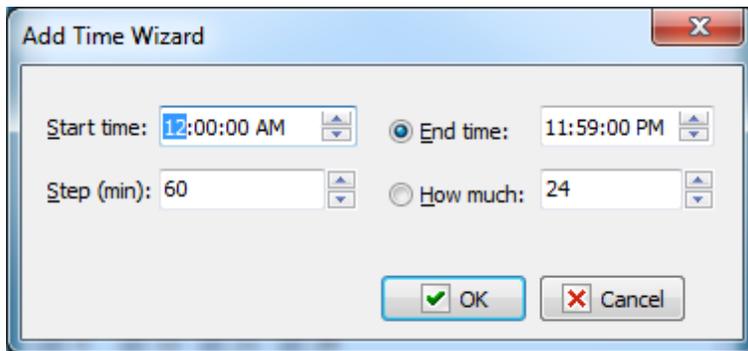


#### Add This Time

Enter a time to execute the action and then click the Plus button to add the time to the list.

#### Time List

Displays a list of times that you add with the Plus button. The Magnifier button displays the **Add Time Wizard**.



To remove a time, select a time in the **Time List** and click the Minus button. To clear the list, click the X button.

### Time Period

Enter whether you want the action to start every day, week, or month.

### Start

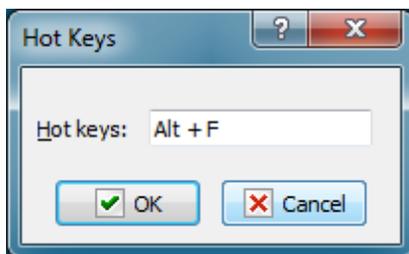
Select the date that you want the event starts to execute a task.

## 5.2 Hot Keys

Executes a task that should be started manually. You can set up a combination of hot keys, such as CTRL + ALT + F. For example, you could set up a combination of hot keys to start Internet Explorer from your keyboard every time you want to browse the Internet.

### Tip

- The **Tray Icon** triggering event also allows you start processes manually.



### Hot Keys

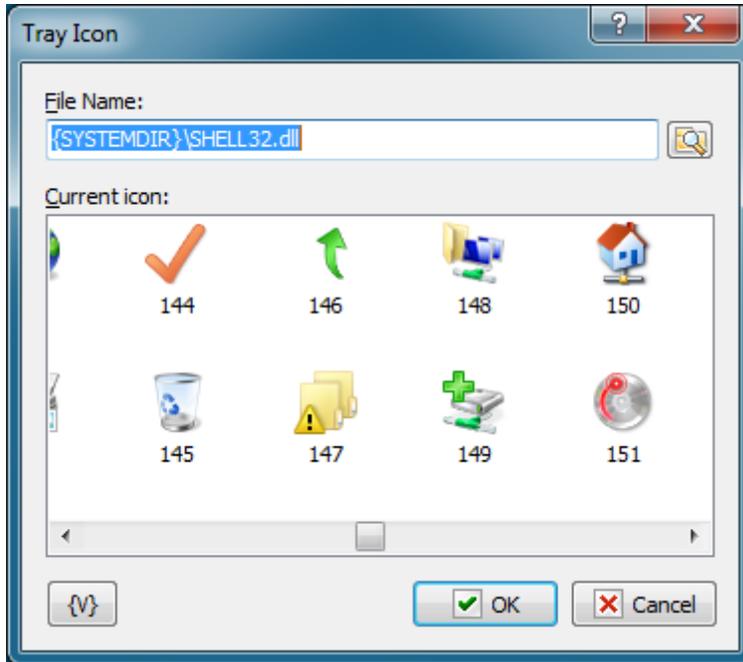
Place your cursor in the box and press the keys you want on the keyboard. For example, press the ALT key to see "Alt" appear in the box.

## 5.3 Tray Icon

Executes a task when you double-click a specific icon in the system tray of the Windows Taskbar. This is useful when you want a specific action to start, such as checking e-mail.

**Tip**

- The **Tray Icon** triggering event also allows you start processes manually.

**File Name**

Enter the name and path of an .Exe or .Dll file to select an icon from. You can click the browse button to select a file.

**Current Icon**

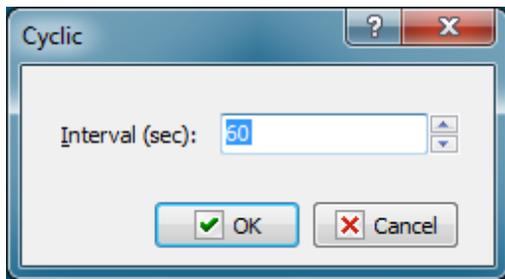
Displays icons available from the file that you select in the **File Name** box.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 5.4 Cyclic

Executes a task at the interval in seconds that you specify. This is useful if you want to repeat a series of actions at a specific interval. For example, you can ping your web site every two minutes or check for new e-mails every five minutes.

**Interval (Sec)**

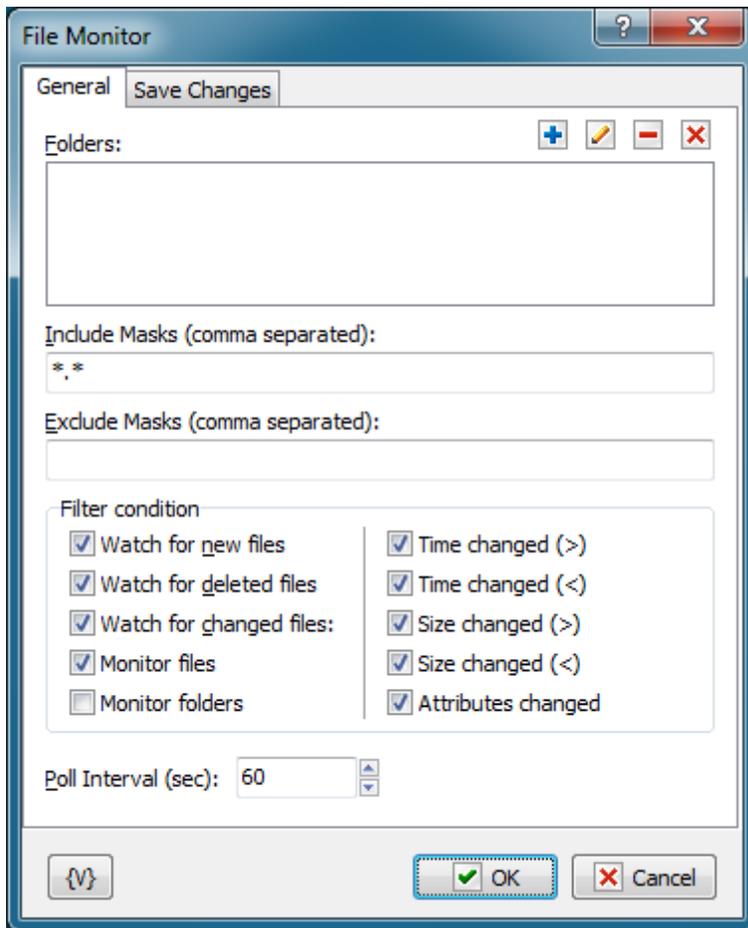
Enter the number of seconds that you want the task or actions to wait between loops.

## 5.5 File Monitor

Executes a task when a file is created, deleted, or changed.

- [General tab](#)<sup>[210]</sup>
- [Save Changes tab](#)<sup>[212]</sup>

**General tab**



### Folders

Select the folders that are monitored. Each folder can be monitored with subfolders. When you click the Plus button, the **Add Folder for Watch dialog box** appears.

### Include Masks (Comma Separated)

Enter one or more masks separated with commas. The wildcard mask "\*.\*)" means that all files are being monitored.

### Exclude Masks (Comma Separated)

Specify one or more masks for the kinds of files that you do not want to monitor. Separate each entry with a comma. Files that match these criteria do not appear on the result list.

#### Tip

You can also use wildcards in this option. Some examples of wildcards are the following:

- \*.txt - all the files with the .txt extension.
- mydoc?.doc - mydoc1.doc, mydoc2.doc, mydocA.doc, and so forth.
- \*.tmp, \*.\$\$\$, \*.bak - all the files with the .tmp, .\$\$\$ and .bak extensions.
- {bakext} - file mask(s) from the {bakext} variable.

### Filter Condition

Select one or more check boxes to specify which changes in files you want to monitor:

- **Watch for New Files**
- **Watch for Deleted Files**
- **Watch for Changed Files**
- **Time Changed**
- **Size Changed**
- **Attributes Changed**

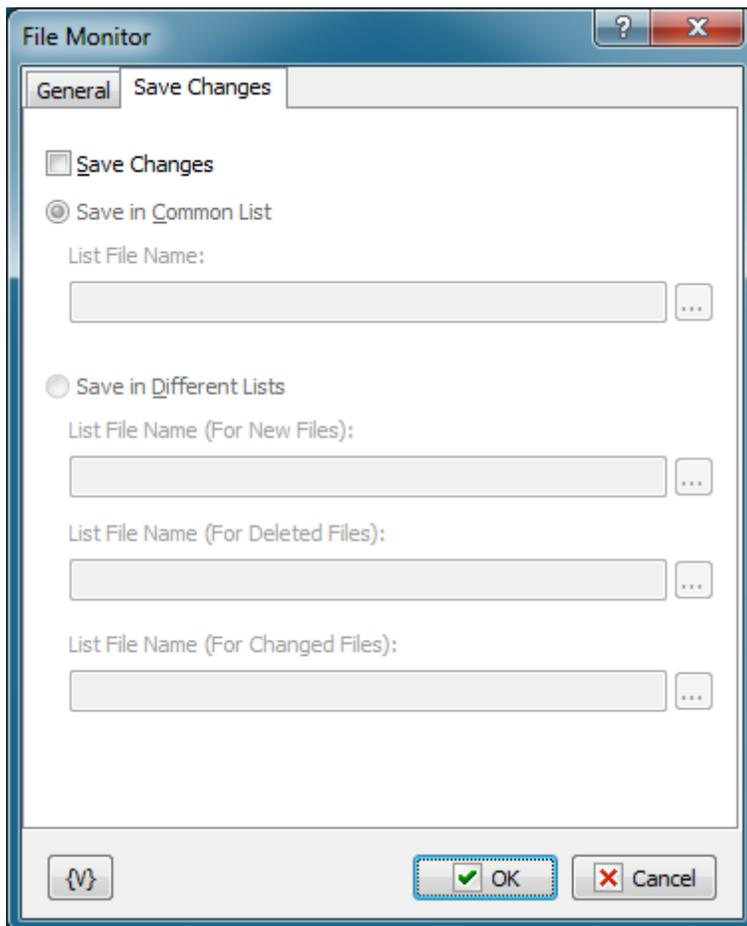
#### **Poll Interval (Sec)**

Enter the time in seconds that you want the event to wait between requests for data.

#### **Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### **Save Changes tab**



#### **Save Changes**

Click the check box to enable the task to save changes to files during the file monitoring session in a log.

**Save in Common List**

Enter the name of a log file that includes all changes during the file monitoring session. Click the browse button to select a file on the local system.

**List File Name**

Enter the file name of the log. Click the browse button to select a file on the local system.

**Save in Different Lists**

Click the check box to enable the task to save changes to file during the file monitoring session to several different logs based on the criteria you select.

**Tip**

- You can click a browse button next to each of the following options to select a file on the local system.

**List File Name (for New Files)**

Saves a log of the names of files created during the monitoring session.

**List File Name (For Deleted Files)**

Saves a log of the names of files deleted during the monitoring session.

**List File Name (For Changed Files)**

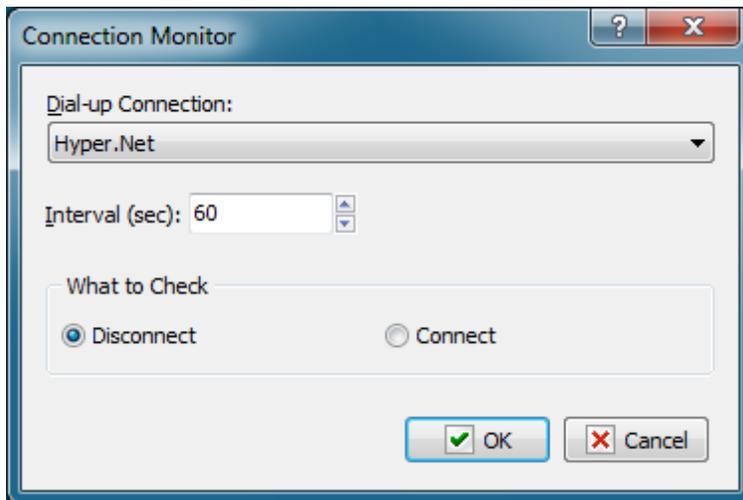
Saves a log of the names of the files modified during the monitoring session.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 5.6 Dial-Up Connection Monitor

Executes a task when the computer completes or terminates a specific dial-up connection. You can specify that RoboTask check for a connection at a specific interval; for example, you can specify that RoboTask check for a dial-up connection every 300 seconds, or every five minutes.



### Dial-up Connection

Select a connection in the dropdown list. The items on the list are the names of the Internet connections for dial-up modems that you create in Windows with **Network Connections** in the **Control Panel**.

### Interval (sec)

Enter the number of seconds that you want the task to wait between checks.

### What to Check

Click one of the following options to tell the task to check if the connection is active or disconnected:

- **Disconnect**
- **Connect**

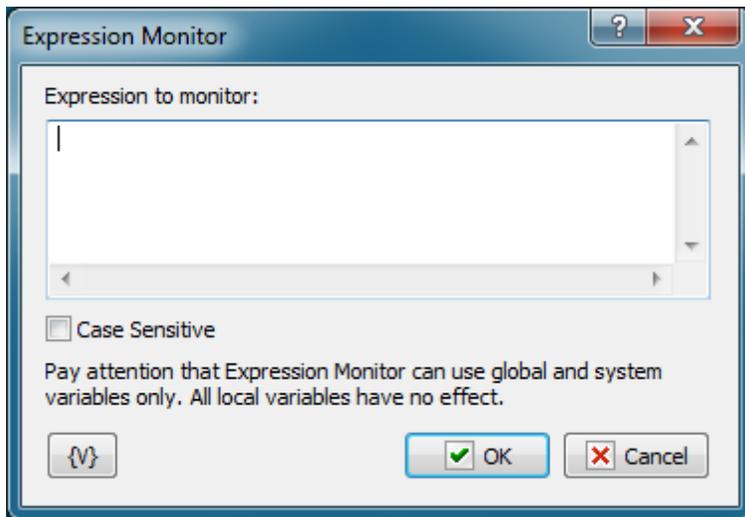
## 5.7 Expression monitor

This is a universal trigger which starts up a task in case of changing specified expression. You need to make up an expression using global variables, both systemic and user's. Expression Monitor will watch changing the value of expression after substitution of variable values. In other words, by means of the Expression Monitor, the changing of one or several variables can be controlled.

### Why is the trigger universal?

You can make up your own tasks which determine a certain value by necessary algorithm using all arsenal of RoboTask. These tasks can be started up, for example, by means of trigger [Cyclic](#)<sup>[209]</sup>. You save the received value into a global variable inside the task. Expression Monitor will start up the task in case of changing the variable.

Thereby, you can "invent" by yourself the parameter by which you need to start up a task. Expression Monitor checks the value of expression every second. So, the trigger will be activated in a second at most after changing the variable.



### **Expression to Monitor**

Input any expression using necessary variables. Notice that only global variables can be used.

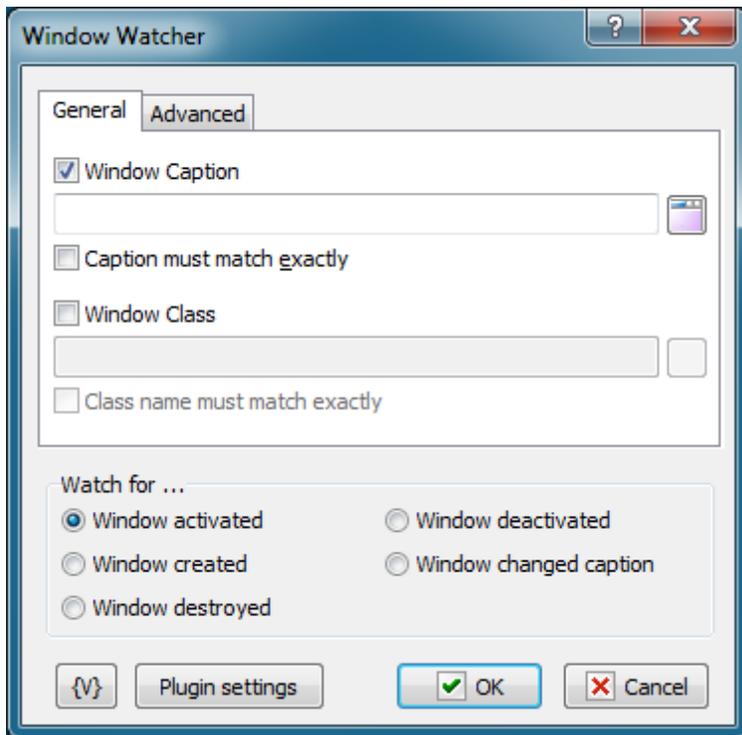
### **Case Sensitive**

Expression Monitor is insensitive to a register By default. If it is important to take into account the register, switch on this checkbox.

## **5.8 Window Watcher**

Runs the task when you perform some actions with windows. This task can respond to creation, destruction, activation and deactivation of a window with specified parameters. Such task can be useful if you want to introduce automatic corrective actions that respond to an error message.

### **General**



### Window Caption

Specify the window title that should be traced. You can specify a specific part of the title if it shouldn't match completely. For example:

- - Microsoft Internet Explorer
- Notepad

You can select a title from the list of windows present in the system by clicking the button to the right of the text box. When captions are compared, the case is not taken into account.

### Caption must match exactly

Select this checkbox if the caption should match exactly. In this case you should specify the full window title.

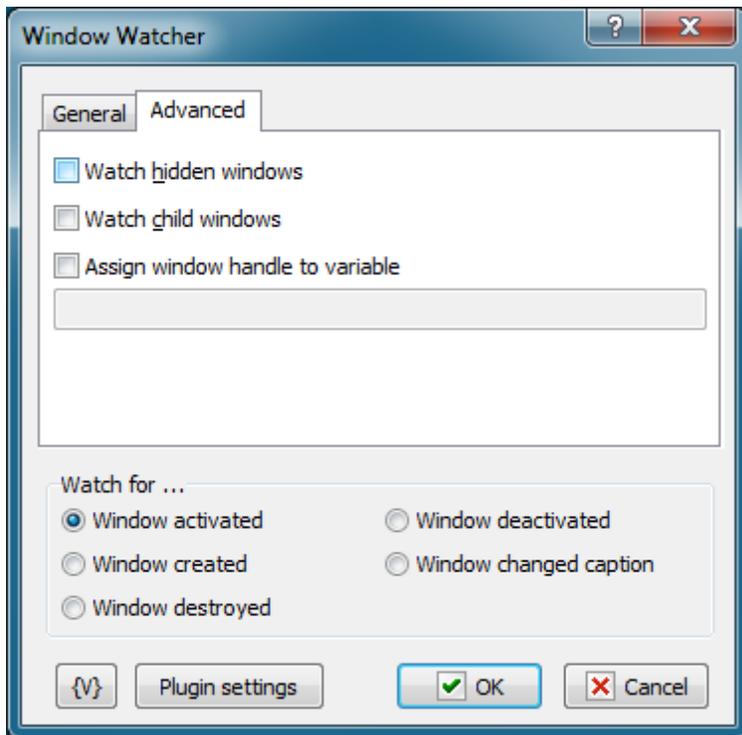
### Window class

Specify the window class that should be traced. You can select a window class from the list of classes present in the system by clicking the button on the right of the text box. When strings are compared, the case is not taken into account.

### Class must match exactly

Select this checkbox if the window class string should match exactly.

## Advanced

**Watch hidden windows**

Select this checkbox if you want the event to trace hidden windows.

**Watch child windows**

Select this checkbox if you want the event to trace child windows.

**Assign window handle to variable**

If you want to save Window Handle for further processing in the task, specify the name of the variable. Window Handle is a unique numeric identifier of a window in the system. It can be used in the [Window Command](#)<sup>[133]</sup> action.

**Watch For**

Click one of the following window states:

- **Window Activated** - the event is triggered when a new or existing window is activated. For example, a window activates when a user presses ALT+Tab on the keyboard.
- **Window Destroyed** - the event is triggered when the window is closed.
- **Window Created** - the event is triggered when a new window is created.
- **Window Deactivated** - the event is triggered when a new or existing window is deactivated. For example, a window deactivates when a user presses ALT+Tab on the keyboard.
- **Window changed caption** - the event is triggered when an existing window has changed own caption.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 5.9 On Time Changed

Executes a task when the computer system time changes. No specific settings are associated with this triggering event.

## 5.10 On Battery Power is Low

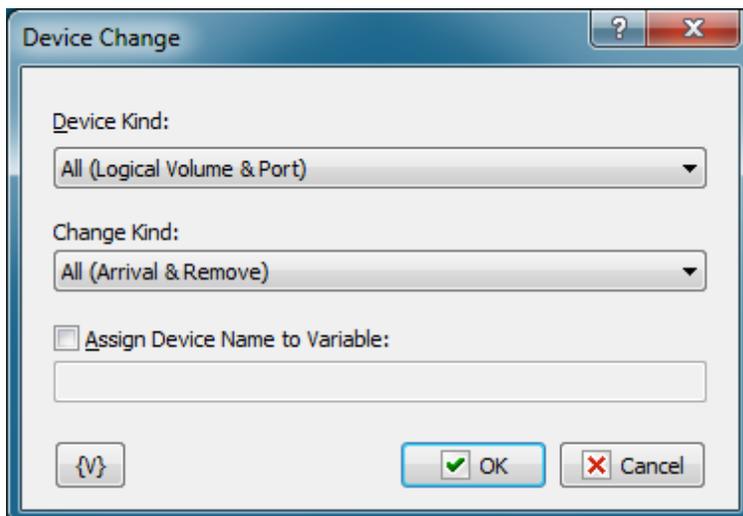
Executes a task when the computer battery is low on power. For example, you can use this triggering event to fire a message that displays on the screen, telling you to replace the computer battery on your laptop computer. No specific settings are associated with this triggering event.

## 5.11 On Spooler Changed

Executes a task when the printer spooler has changed. For example, you can use this triggering event to display a message if a particular document is sent to a printer. No specific settings are associated with this triggering event.

## 5.12 On Device Changed

Runs the task when a device (for example, CD or Flash Drive) is changed. The device name can be saved to a variable and used in the task.



**Device Kind**

Specify a type of device that triggers the task.

- All (Logical Volume & Port) - any type of device;
- Logical Volume - a logical volume mounted dynamically (CD, Flash Drive, etc.);
- Port - a COM or LPT port.

**Change Kind**

Specify a type of change that triggers the task.

- All (Arrival & Remove) - any type of event;
- Arrival - a device (drive) appears in the system. For example, you have inserted a new CD;
- Remove - a device is removed from the system. For example, you have pressed the eject button of the CD drive.

**Assign Device Name to Variable**

Specify a name of variable where you want to save the name of the device that has been changed.

## 5.13 On Palette Changed

Executes a task when the display palette, or colors on the computer screen, for Windows changes. This can happen when you switch between an MS-DOS application and a Windows-based application. No specific settings are associated with this triggering event.

## 5.14 On Display Changed

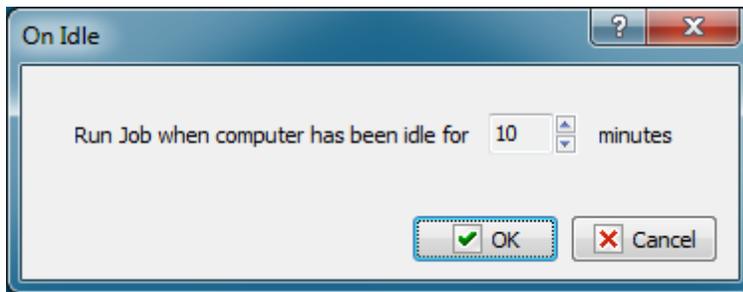
Executes a task when the display on the computer screen changes. No specific settings are associated with this triggering event.

## 5.15 On Log Off

Executes a task when a user logs off or the computer shuts down. No specific settings are associated with this triggering event.

## 5.16 On Idle

Executes a task when the computer has been idle for a specified period of time.

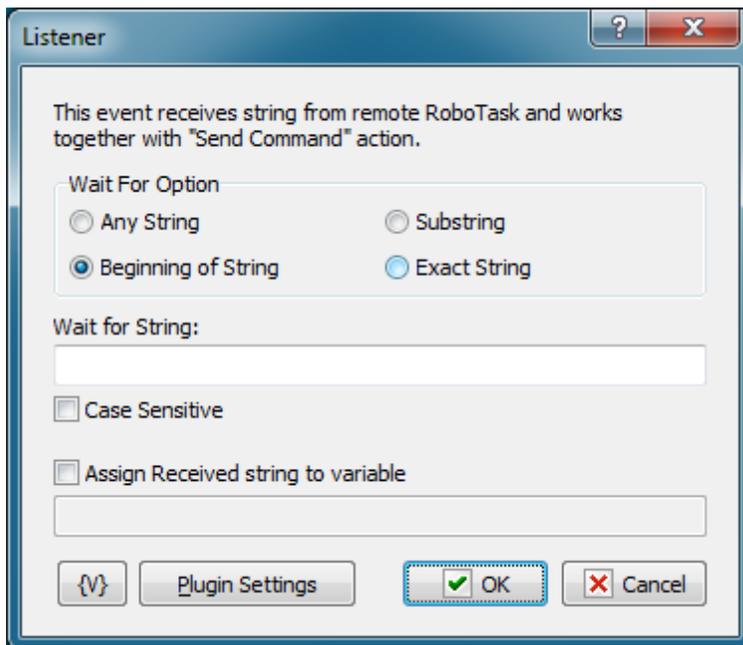


### Run Job When Computer Has Been Idle for \_\_\_ Minutes

Enter the time that you want in minutes.

## 5.17 Listener

Allows you to start a task when a command from another computer is received. Works together with the Send Command action. Performs the primary analysis of the received string and starts the corresponding task if the received string meets the specified conditions.



### Wait For Option

Select the mode of waiting for a string.

- Any string - any received string will start the task. For example, it is convenient if you exchange messages that should be just recorded to the log.
- Beginning of String - the received string should start with the key substring.
- Substring - the key substring can be in any position of the received string.
- Exact String - the received string must exactly match the key string.

**Wait for String**

Specify the key string that will start the task. This parameter does not make sense if the "Any String" mode is on.

**Case Sensitive**

Select this checkbox if strings should be compared with their letter case taken into account.

**Assign Received String to Variable**

Specify the name of the variable if the received string should be assigned to a variable. This value can be used in the task.

The simplest sample use is message exchange: the Listener receives a string, assigns it to a variable and starts the task. The task displays this variable as an informational message.

**Plugin settings**

You can set up [parameters of Network plugin](#)<sup>[160]</sup>.

**Variable**

Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

# Part

---



## Appendix

## 6 Appendix

### 6.1 Full list of system variables

#### Example of using

```
{AddressOnly(John Smith <j.smith@mydomain.com>)}
{Asc(©)}
{CapsLockState}
{Chr(169)}
{ClipboardContent}
{ComputerName}
{CurrentWindowCaption}
{CurrentWindowClass}
{CurrentWindowHandle}
{Date}
{DateIncDays(-7)}
{DateTime}
{Day}
{DayOfTheYear({Date})}

{DayOfWeek}
{DayOfWeekNo}
{DefScriptDir}
{DiskLoad(c:\)}
{EnvironmentVar(Path)}
{Eol}
{ExtractFileDrive(C:\Folder\File.ext)}
{ExtractFileExt(C:\Folder\File.ext)}
{ExtractFileName(C:\Folder\File.ext)}
{ExtractFileNameNoExt(C:\Folder\File.ext)}
{ExtractFilePath(C:\Folder\File.ext)}
{ExtractFilePathNoDrive(C:\Folder\File.ext)}
{FileAttributes(C:\AUTOEXEC.BAT)}
{FileExists({TempDir}\file.tmp)}
{FilesCount({TempDir})}
{FileSize(c:\autoexec.bat)}
{FileTime(c:\autoexec.bat)}
{FolderExists({TempDir})}
{FoldersCount({TempDir})}
{FolderSize({TempDir})}

{FreeDiskSpace(c:)}
{Hours}
{IncDays(-1)}
{IncMonths(-1)}
{IncYears(-1)}
{IPAddress({ComputerName})}
{IsError}
{IsFileFree(c:\autoexec.bat)}
{IsProcessExists(notepad.exe)}
{IsWindowActive(Notepad)}
```

#### Description

```
Returns address only from full address
Returns ASCII code of specified character.
Checks CapsLock state
Returns the symbol with specified ASCII code.
Clipboard content as Text
Computer name
Returns the title (caption) of current active window
Returns the class name of current active window
Returns the handle of current active window
Current date
Current date ± number of days
Current date & time
Current day
Returns the day of the year of specified or current date
Current day of week
Current day of week number
Default Basic script folder
Returns the disk load in percents
Get environment variable
End of line
Extracts file drive from full path
Extracts file extension from full path
Extracts file name from full path
Extracts file name without extension
Extracts file path from full path
Extracts file path from root folder
Returns attributes of file as string, like this: "AHR$"
Check file existence
Returns the amount of files in specified folder
File size in bytes
Returns the time of last update
Checks folder existence
Returns the amount of folders in specified folder
Returns the total size of files in specified folder (in bytes)
Free disk space in bytes
Current hours
Current day ± number of days
Current month ± number of months
Current year ± number of years
Returns IP address by computer name
Returns "True" when error occurred at previous step
Checks file for free access
Checks existence of process in memory
Check window existence
```

{IsWindowHidden(Notepad)}	Returns "True" if window invisible
{IsWindowMaximized(Notepad)}	Returns "True" if window maximized
{IsWindowMinimized(Notepad)}	Returns "True" if window minimized
{IsWindowVisible(Notepad)}	Returns "True" if window visible
{LastErrorCode}	Error code. Parameter of error handling task.
{LastErrorDescription}	Error description. Parameter of error handling task.
{LastErrorStep}	Error step number. Parameter of error handling task.
{LastErrorTaskExternalName}	External name of error task. Parameter of error handling task.
{LastErrorTaskID}	Error task ID. Parameter of error handling task.
{LastErrorTaskName}	Error task name. Parameter of error handling task.
{LowerCase(Sample String)}	Converts a string to lowercase.
{Minutes}	Current minutes
{MonthIncDays(-1)}	Current month $\pm$ number of days
{MonthName}	Current month name
{MonthNo}	Current month number
{mSeconds}	Current milliseconds
{NumLockState}	Checks NumLock state
{RandomValue(1000)}	Returns random integer value between 0 and parameter (default 1000)
{ScrollLockState}	Checks ScrollLock state
{Seconds}	Current seconds
{ShortDayOfWeek}	Current day of week short name
{ShortMonthName}	Current month short name
{ShortYear}	Two-digits current year
{ShortYearIncDays(-1)}	Two-digits current year $\pm$ number of days
{StringLength(Sample String)}	Returns the length of specified string.
{SystemDir}	Windows system directory
{TaskCategoryID}	Category ID of current task (works only in context of task)
{TaskExtName}	External name of current task (works only in context of task)
{TaskID}	ID of current task (works only in context of task)
{TaskName}	Name of current task (works only in context of task)
{TempDir}	Temporary directory
{TextFile(file_name)}	Contents of text file
{Time}	Current time
{TotalDiskSpace(c:\)}	Returns the total size of specified disk (in bytes)
{UniqueFileName({TempDir}*.tmp)}	Returns unique random file name by the mask
{UpperCase(Sample String)}	Converts a string to uppercase.
{UserName}	User name
{ValidFileName(c:\My<Invalid:File*Name>.txt)}	Returns valid file name
{VolumeLabel(c:\)}	Returns the label of specified volume
{WeekOfTheYear({Date})}	Returns the week of the year of specified or current date
{WinDir}	Windows directory
{WindowExists(Notepad)}	Check window existence
{Year}	Current year
{YearIncDays(-1)}	Current year $\pm$ number of days

# Part

---



VII

Other software

## 7 Other software

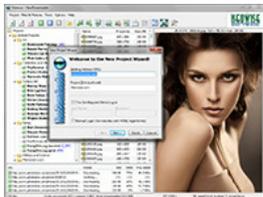


### Instant Gallery Maker

*for Windows 98/ME/2000/XP/2003/Vista/Windows 7*

Create professional-looking image galleries with a few mouse clicks. All you have to do is to click a few buttons and watch your new gallery come alive!

[Learn more...](#)



### NeoDownloader

*for Windows 98/ME/2000/XP/2003/Vista/Windows 7*

Download thousands of pictures, music and videos. Forget about pop-ups, banners and saving every file manually - just let this fully automatic tool do the work for you.

[Learn more...](#)

# Index

## - A -

### actions

- Add key 149
- Break 131
- Browse for Computers 62
- Browse for Folders 61
- Change Folder 73
- Check connection 114
- Check for E-mail 84
- Check for Window 134
- comment 54
- Continue 131
- Copy/Move File 64
- Create Folder 73
- Create Variable 114
- Dataset Loop 126
- Delete File 66
- Delete key 150
- Delete Value 154
- Delimited variable 117
- Dial Up/Hang Up 113
- Disable Automation 142
- Download Files 88
- Else 133
- Enable Automation 142
- Enable Task 118, 119
- End If 133
- End Loop 131
- Exit RoboTask 142
- Export 156
- Extract Files 83
- File Attributes 80
- File Loop 122
- FTP Change Folder 93
- FTP Command 96
- FTP Create Folder 94
- FTP Delete File 93
- FTP Download 91
- FTP File Loop 97
- FTP Log Off 91
- FTP Log On 90
- FTP PWD 95
- FTP Remove Folder 95
- FTP Rename File 92
- FTP Upload 91
- If Then 131
- Import 157
- Increment Variable 117
- Input box 63
- Lock Workstation 140
- Log Message 142
- Log Off 50
- Minimize RoboTask 141
- Open Dialog 59
- Pack Files 81
- Pause 50
- Ping Host 89
- Print Document 44
- query 51
- Question Dialog 57
- Read Value 152
- Remove Folder 74
- Remove Variable 116
- Rename File 65
- Rename Folder 74
- Rename key 151
- Rename Value 155
- Restore RoboTask 141
- Run as User 140
- Run Program/Open Document 43
- Save Dialog 60
- Search for Files 69
- Send command 158
- Send Email 85
- Send Keystrokes 45
- Send Network Message 159
- Set Variable 115
- Shif Keys State 49
- Show Message/Play Sound 55
- Simple Loop 129
- Sort Text 77
- Start Service 138
- Start Task 119
- Stop Service 139
- Stop Task 121
- Synchronize Folders 75
- Terminate Process 54
- Text Loop 125
- Transfer Type 96
- Unique File Name 79
- Wait for File 76

- actions
  - wait for process finish 52
  - wait for process startfinish 53
  - Wait for Task 121
  - Window Command 133
  - Window Loop 127
  - Write Text File 68
  - Write Value 153
- Add key 149
- administrative commands 141, 142
- alarm clock 207
- archiving
  - files 81
- Ascending 77
- assistance
  - getting 41
- automation
  - disabling 142
  - enabling 40, 142

## - B -

- basic 178
  - evaluate expression 177
  - RoboTaskApp object 178
  - script 178
- Basic script 203
- battery power
  - triggering event 218
- beginning
  - RoboTask 3
- Break 131
- breaking
  - loops 131
- browse dialog box
  - displaying 61, 62
- bugs
  - reporting 41
- buying
  - RoboTask 6

## - C -

- CD/DVD writer
  - eject 186
  - Erase disk 185
  - Write to disk 183

- changing 73
  - directories 73
  - folders 73
- Check connection 114
- Check email 165
- checking
  - e-mail 84
  - for windows 134
- Clear Clipboard 145
- Click the button 148
- Client-Server 22
- Clipboard
  - copy 143
  - Cut 143
  - get 145
  - paste 144
  - put 145
- Clipboard
  - clear 145
- closing
  - windows 133
- commands
  - using ftp 96
- comma-separated string 117
- comment 54
- computer display
  - triggering event 219
- computer idle
  - triggering event 219
- computers
  - locking 140
  - shutting down 50
- conditional statements
  - using 131, 133
- Connect to 24
- connecting
  - dial-up connections 113
- Continue 131
- Copy 143
- copying
  - files 64
  - tasks 32
- creating 12, 73
  - directories 73
  - expressions 12
  - file lists 69
  - folders 73
  - tasks 5, 26

- creating 12, 73
  - text files 68
  - variables 12, 114
  - Zip files 81
- Custom variables 11
- customizing
  - options 6
- Cut 143
- cyclic
  - triggering event 209

## - D -

- database 51
- datasets
  - looping 126
- Define connections 24
- delaying
  - tasks 50
- delete email 165
- Delete key 150
- Delete Value 154
- deleting 93
  - files 66, 93
  - ftp 93
  - tasks 32
- Descending 77
- device changed
  - triggering event 218
- Dial Up
  - Check connection 114
- dial-up connection monitor
  - triggering event 213
- dial-up connections
  - making 113
- directories
  - changing 73, 93
  - creating 73, 94
  - printing 95
  - removing 74, 95
  - renaming 74
  - synchronizing 75
  - transferring 96
- disabling
  - tasks 41, 119
- display
  - triggering event 219
- documents

- opening 43
- printing 44
- Downloading
  - email 164
  - files 88
  - ftp 91
- duplicating
  - tasks 32

## - E -

- editing
  - tasks 6, 32
- eject CD Tray 186
- Else statements
  - using 133
- e-mail
  - checking 84
  - sending 85
- enabling
  - automation 40
  - tasks 41
- End If statements
  - using 133
- End Loop 131
- Erase to disk 185
- error handling 21
- error processing 21
- evaluate expression 177
- events
  - Cyclic 209
  - Dial-Up Connection Monitor 213
  - Expression Monitor 214
  - File Monitor 210
  - Hot Keys 208
  - On Battery Power is Low 218
  - On Device Changed 218
  - On Display Changed 219
  - On Idle 219
  - On Log Off 219
  - On Palette Changed 219
  - On Spooler Changed 218
  - On Time Changed 218
  - Scheduler 207
  - Tray Icon 208
  - Window Watcher 215
- exiting
  - RoboTask 33, 142

Export Registry 156  
Expression Monitor  
  triggering event 214  
expressions 116  
  creating 12, 114  
  deleting 116  
  modifying 38  
  overview 9  
  removing 116  
  setting 115  
  starting 119

## - F -

File Attributes 80  
file lists  
  creating 69  
File Loop 122  
file loops  
  ftp 97  
file monitor  
  triggering event 210  
files 64, 81, 83  
  archiving 81  
  copying 64  
  creating 68  
  deleting 66, 93  
  downloading 88, 91  
  extracting 83  
  looping 122  
  moving 64  
  renaming 65, 92  
  searching 69  
  synchronizing 75  
  transferring 96  
  unzipping 83  
  uploading 91  
  zipping 81  
Fill  
  Web Form 186  
Fill web form 186  
focusing  
  windows 133  
folders  
  changing 73, 93  
  creating 73, 94  
  printing 95  
  removing 74, 95  
  renaming 74  
  synchronizing 75  
  transferring 96  
ftp  
  changing folders 93  
  creating folders 94  
  deleting 93  
  download 91  
  file loops 97  
  log off 91  
  log on 90  
  PWD 95  
  removing folders 95  
  renaming files 92  
  transfer 96  
  uploading 91  
  using commands 96  
FTP File Loop 97

## - G -

Get Text 145  
getting started 3

## - H -

Help  
  opening 41  
hiding  
  windows 133  
hosts  
  pinging 89  
hot keys  
  triggering event 208  
HTTP  
  downloading 88

## - I -

idle  
  triggering event 219  
If Then statements  
  using 131  
Import Registry 157  
Incoming email 161, 162, 164, 165, 166  
increment 117  
Input box 63

introduction 2  
 IsError 21

## - J -

jobs 119  
   adding variables 13  
   creating 5, 26  
   creating shortcuts 33  
   deleting 32  
   disabling 41, 119  
   duplicating 32  
   editing 6, 32  
   running 5, 32  
   starting 118, 142  
   stopping 32, 119, 142

## - K -

keyboard  
   hold down keys 49  
   release keys 49  
   strokes for 45  
 keystrokes  
   sending 45, 49

## - L -

LastErrorCode 21  
 LastErrorDescription 21  
 LastErrorStep 21  
 LastErrorTaskExternalName 21  
 LastErrorTaskID 21  
 Legal notice 6  
 listener 220  
 local variables 11, 16  
 locking  
   computers 140  
 log off  
   ftp 91  
   triggering event 219  
 log on  
   ftp 90  
 logging  
   messages 142  
 logging off  
   computers 50

logoff 50  
 logs  
   viewing 33  
 Loop 162  
 looping 126  
   datasets 126  
   files 122  
   queries 126  
   simple 129  
   text 125  
   Window 127  
 loops 129  
   breaking 131  
   continuing 131  
   ending 131  
   iterating 97

## - M -

Macro recorder 14  
 Mail loop 165  
 maximizing  
   windows 133  
 messages  
   displaying 55  
   logging 142  
   sending 159  
 Messages count 165  
 minimizing  
   RoboTask 141  
   windows 133  
 Mouse 14  
   click 148  
   move pointer 149  
   move to object 146  
 Move mouse pointer 149  
 Move pointer to object 146  
 Move to Object 146  
 moving  
   files 64

## - N -

Network  
   Client 23  
   Connect to 24  
   Control remote 22

## Network

- Defined connections 24
- listener 220
- plugin settings 160
- send command 158
- Server 22
- network messages
  - sending 159
- networking
  - dial-up connections 113

**- O -**

- open dialog box
  - displaying 59
- options
  - customizing 6
  - setting 33
- overview 2

**- P -**

- packing
  - Zip files 83
- palette
  - triggering event 219
- Parse 166
- Paste 144
- pinging
  - hosts 89
- Pop3 Server 161, 162, 164, 165, 166
- practical recommendations 17
  - custom variables 10
  - Debugging 16
  - task synchronization 17
  - tasks 15, 17
- printing
  - documents 44
  - folders 95
- process 52, 53
- programs
  - running 43
- purchasing
  - RoboTask 6
- Put Text 145

**- Q -**

- queries
  - looping 126
- query 51
- querying
  - loops 126
- question dialog box
  - displaying 57
- quitting
  - RoboTask 142

**- R -**

- RDC 20
- Read Value 152
- Record a macro 14
- RegExp 195
  - loop 195
  - match 201
  - replace 198
- RegExp Com object 203
- registering
  - RoboTask 6
- Registry
  - Add key 149
  - Delete key 150
  - Delete Value 154
  - Export 156
  - Import 157
  - Read Value 152
  - Rename key 151
  - Rename Value 155
  - Write Value 153
- Regular Expressions 195, 198, 201, 203
- remote servers
  - checking 89
- removing 74
  - directories 74
  - folders 74
  - variables 116
- Rename key 151
- Rename Value 155
- renaming 74, 92
  - directories 74
  - files 65, 92

- renaming 74, 92
  - folders 74
  - ftp 92
- reporting
  - bugs 41
- restoring
  - RoboTask 141
  - windows 133
- RoboTask
  - exiting 142
  - minimizing 141
  - restoring 141
  - stopping 33
- Run as User 140
- Run Script 178
- running
  - tasks 5, 32

**- S -**

- save dialog box
  - displaying 60
- schedulers
  - triggering event 207
- searching
  - files 69
- Send command 158
- Send Keystrokes 43
- sending
  - e-mail 85
  - keystrokes 45, 49
- services
  - starting 138
  - stopping 139
- setting
  - options 33
  - variables 115
- shortcuts
  - creating 33
- shut down
  - triggering event 219
- shutting down
  - computers 50
- Simple Loop 129
- simple loops
  - using 122, 129
- Sort Text 77
- sound

- playing 55
- spooler
  - triggering event 218
- SQL
  - loops 126
  - query 51
- starting
  - services 138
  - tasks 118, 119
- Stop Task 121
- stopping
  - RoboTask 33, 142
  - services 139
  - tasks 32
- String 175
  - between 175
  - delete 172
  - extract 173, 175
  - pos 174
  - replace 170
  - trim 171
- Strings 170, 171, 172, 173, 174, 175
- synchronize tasks 121
- Synchronize workstations 158, 220
- synchronizing 75
  - directories 75
  - files 75
  - folders 75
- system time
  - triggering event 218
- System variables 10, 11, 223
  - displaying 38

**- T -**

- task synchronization 17
- tasks 14, 17, 32, 118, 119
  - adding variables 13
  - copying 32
  - creating 5, 14, 26
  - creating shortcuts 33
  - delaying 50
  - deleting 32
  - disabling 41, 119
  - duplicating 32
  - editing 6, 14, 32
  - enabling 41, 118
  - running 5, 32

- tasks 14, 17, 32, 118, 119
  - starting 118, 119, 142
  - stopping 32, 119, 121, 142
  - waiting 121
- technical support 41
- Terminate Process 54
- text 68
  - looping 125
- Text file 77
- text files
  - creating 68
- toolbars
  - viewing 33
- Total size 165
- transferring
  - files 96
- tray icon
  - triggering event 208
- triggering events
  - Cyclic 209
  - Dial-Up Connection Monitor 213
  - Expression Monitor 214
  - File Monitor 210
  - Hot Keys 208
  - On Battery Power is Low 218
  - On Device Changed 218
  - On Display Changed 219
  - On Idle 219
  - On Log Off 219
  - On Palette Changed 219
  - On Spooler Changed 218
  - On Time Changed 218
  - Scheduler 207
  - Tray Icon 208
  - Window Watcher 215

## - U -

- unhiding
  - windows 133
- Unique File Name 79
- unzipping
  - files 83
- uploading
  - ftp 91
- user variables
  - displaying 38
- user-defined variables 13

- Using Basic 17

## - V -

- Variables 10, 11, 116
  - adding to tasks 13
  - creating 12, 114
  - deleting 116
  - Delimited variable 117
  - increment 117
  - list 223
  - modifying 38
  - overview 9
  - removing 116
  - setting 115
- viewing
  - logs 33
  - toolbars 33
- Vista 20

## - W -

- Wait for File 76
- wait for process 52, 53
- Wait for Task 121
- Wait for window 135
- Web
  - Document 186
  - Documents 192
  - Form 186
  - Forms 192
- While loop 129
- Window 135
- Window loop 127
- window watcher
  - triggering event 215
- windows 133
  - checking for 134
  - focusing 133
  - hiding 133
  - manipulating 133
  - maximizing 133
  - minimizing 133
  - restoring 133
  - unhiding 133
- Windows 2008 Server 20
- Windows 7 20

Windows NT/2000 services 138, 139  
WinPopup  
    sending 159  
Working as system service 19  
    differences 20  
    RDC 20  
    Vista 20  
Write to disk 183  
Write Value 153

## **- Z -**

Zip files  
    creating 81  
    extracting 83