

# ROBOTASK

## User's Guide

2023

# Table of Contents

<b>Part I Quick Start Guide</b>	<b>1</b>
1 Features .....	2
2 Getting Started in the RoboTask Window .....	3
3 Create Tasks .....	5
4 Run Tasks .....	6
5 Edit Tasks .....	6
6 Customize RoboTask .....	6
7 Purchasing .....	6
8 License Agreement .....	7
9 Trademarks and Copyrights .....	8
<b>Part II Several practical recommendations</b>	<b>9</b>
1 Using Variables in Tasks .....	10
Usage of Variables .....	10
System variables .....	11
Custom variables .....	12
Local Variables .....	12
Create a Variable .....	14
Add Variables to a Task .....	14
Saving the result into a variable .....	15
Rules for using variables in expressions .....	15
Prefixes Used with Variables .....	16
How to use variables correctly .....	18
2 Macro recorder .....	18
Macro Recorder Settings .....	19
3 Running tasks simultaneously .....	20
4 Debugging of the task .....	21
5 Task synchronization .....	22
6 Using scripts .....	22
7 Drag and Drop .....	23
8 Running RoboTask as an NT service .....	23
Service mode vs User mode .....	23
Installing and running .....	26
Using UNC names .....	27
Authentication parameters .....	27
RoboTask Service manager .....	28
Peculiarities of working in the system service mode .....	35
The visual differences between the work in ordinary application mode and in the system service mode.....	36
Peculiarities of RoboTask Service working in RDC mode.....	36
Some peculiarities of the RoboTask service in Vista+ OS.....	36
How to access RoboTask Service.....	37
9 Errors handling .....	41

**10 Control of RoboTask through network ..... 42**

**Network Server ..... 43**

**Network Client ..... 44**

        Connect to..... 44

        Define connections..... 44

**Network Command utility ..... 45**

**11 Runtime module ..... 46**

**12 Useful Utilities in RoboTask Package ..... 47**

**Part III Menus ..... 49**

**1 Task ..... 50**

    Task - New ..... 50

    Task - Edit ..... 58

    Task - Delete ..... 58

    Task - Run ..... 58

    Task - Stop ..... 58

    Task - Find the task ..... 58

    Task - Local running task ..... 59

    Task - Export ..... 59

    Task - Import ..... 59

    Task - Copy ..... 59

    Task - Paste ..... 59

    Task - Duplicate ..... 60

    Task - Create Shortcut on Desktop ..... 60

    Task - Create Batch File ..... 60

    Task - Exit ..... 60

**2 Remote ..... 60**

    Connections manager ..... 61

    Remote - Connect to ..... 63

    Remote - Disconnect ..... 64

    Enable Remote Automation ..... 65

    Remote Variable & Settings ..... 65

    Remote Running Tasks ..... 67

    RoboTask Service manager ..... 67

    The list of predefined connections ..... 67

**3 Options ..... 67**

    Options - View Log ..... 67

    Options - View Toolbar ..... 67

    Options - Settings ..... 67

    Options - Variables ..... 71

    Options - Scale ..... 73

**4 Tools ..... 73**

    Tools - Enable Automation ..... 73

    Tools - Lock Task Editor ..... 73

    Tools - Enable All Tasks in List ..... 73

    Tools - Disable All Tasks in List ..... 73

    Tools - Deleted Tasks ..... 73

    Tools - Start selected tasks sequentially ..... 75

    Tools - Backup application tasks ..... 75

    Tools - Restore application tasks ..... 75

**5 Help ..... 76**

Help - Help Topics .....	76
Help - Send Bug Report .....	76
Help - Technical support .....	77
Help - Check for updates .....	77
Help - About .....	77

## Part IV Actions 78

<b>1 General .....</b>	<b>79</b>
Run Program/Open Document .....	79
Run Console Application .....	80
Print Document .....	83
Send Keystrokes .....	84
Shift Keys State .....	88
Log Off .....	89
Pause .....	90
SQL Query .....	91
Wait for Process Finish .....	93
Wait for Process Start .....	94
Terminate Process .....	95
Comment .....	96
<b>2 Dialogs and Notifications .....</b>	<b>96</b>
Show Message/Play Sound .....	97
Question Dialog .....	99
Open Dialog .....	100
Save Dialog .....	102
Browse for Folders .....	103
Browse for Computers .....	105
Input Box .....	106
Show Notification .....	106
Choice Dialog .....	109
Multichoice Dialog .....	111
Show Text .....	113
Custom Input Dialog .....	114
Log Dialog .....	117
<b>3 Files and Folders .....</b>	<b>117</b>
Copy/Move File .....	117
Rename File .....	119
Delete File .....	120
Write Text File .....	121
Search for Files .....	123
Set Current Folder .....	128
Create Folder .....	128
Rename Folder .....	129
Remove Folder .....	129
Synchronize Folders .....	130
Wait for file .....	134
Sort Text .....	135
Unique File Name .....	136
File Attributes .....	137
File Date and Time .....	138
Empty Recycle Bin .....	139
Get File Metadata .....	139
Read Text file .....	140

Get File Hash .....	141
<b>4 Zip .....</b>	<b>142</b>
Zip Pack Files .....	142
Zip Extract Files .....	144
<b>5 Internet .....</b>	<b>146</b>
Check for Email .....	146
Send Email .....	147
Download Files .....	151
Ping Host .....	152
Check HTTP address .....	153
Wake On LAN .....	155
REST Client/Invoke Web Service .....	156
<b>6 FTP .....</b>	<b>160</b>
FTP Log On .....	160
FTP Log Off .....	161
FTP Download .....	161
FTP Upload .....	161
FTP Rename File .....	162
FTP Delete File .....	163
FTP Change Folder .....	163
FTP Create Folder .....	164
FTP Remove Folder .....	164
FTP PWD .....	165
FTP Transfer Type .....	165
FTP Command .....	166
FTP File Loop .....	166
FTP Synchronization .....	170
<b>7 SSL FTP .....</b>	<b>172</b>
SSL FTP Log On .....	172
SSL FTP Log Off .....	176
SSL FTP Download .....	176
SSL FTP Upload .....	177
SSL FTP Rename File .....	177
SSL FTP Delete File .....	178
SSL FTP Change folder .....	179
SSL FTP Create Folder .....	179
SSL FTP Remove Folder .....	180
SSL FTP PWD .....	180
SSL FTP Transfer Type .....	180
SSL FTP File Loop .....	181
SSL FTP Synchronization .....	186
SSL FTP Check File .....	188
<b>8 SFTP (SSH FTP) .....</b>	<b>189</b>
SFTP Log On .....	189
SFTP Log Off .....	191
SFTP Download .....	191
SFTP Upload .....	192
SFTP Rename File of Folder .....	193
SFTP Delete File .....	194
SFTP Change Folder .....	195
SFTP Create Folder .....	195
SFTP Remove Folder .....	196

SFTP PWD (print working directory) .....	196
SFTP File Loop .....	197
SFTP Synchronization .....	201
SFTP File Permissions .....	204
SFTP Command .....	204
SFTP File Check .....	205
<b>9 Dial-Up .....</b>	<b>206</b>
Dial Up/Hang Up .....	206
Check connection .....	207
<b>10 Variables .....</b>	<b>208</b>
Create Variable .....	208
Set Variable .....	209
Remove Variable .....	210
Increment Variable .....	211
Split text .....	211
Time Difference .....	212
Count array elements .....	214
<b>11 Task Commands .....</b>	<b>214</b>
Enable Task .....	214
Disable Task .....	215
Start Task .....	215
Stop Task .....	217
Wait for Task .....	218
Task status .....	219
<b>12 Remote Task .....</b>	<b>220</b>
Remote RoboTask Connect .....	220
Remote RoboTask Disconnect .....	222
Enable/Disable Remote Task .....	222
Start Remote Task .....	223
Stop Remote Task .....	224
Wait for Remote Task .....	225
Remote Task Status .....	227
Active Connections Loop .....	228
<b>13 Loops and Flows .....</b>	<b>229</b>
File Loop .....	229
Text Loop .....	234
Dataset Loop .....	236
Window loop .....	239
Simple Loop .....	240
While loop .....	241
Process Loop .....	243
End Loop .....	244
Break .....	245
Continue .....	245
If Then .....	245
Else .....	247
End If .....	247
Elseif Then .....	248
Exit .....	251
GoTo .....	251
<b>14 Window .....</b>	<b>252</b>
Window Command .....	252

Check for Window .....	254
Wait for Window .....	256
Window Set Size and Position .....	259
Window Elements Loop .....	262
Window Get Size and Position .....	265
<b>15 System Functions .....</b>	<b>270</b>
Start Service .....	270
Stop Service .....	271
Service status .....	272
Lock Workstation .....	273
Run Program as User .....	273
Write to Event Log .....	275
<b>16 RoboTask Commands .....</b>	<b>276</b>
Minimize RoboTask .....	276
Restore RoboTask .....	276
Exit RoboTask .....	276
Enable Automation .....	276
Disable Automation .....	276
Log Message .....	276
Restart RoboTask .....	277
<b>17 Clipboard .....</b>	<b>278</b>
Cut to Clipboard .....	278
Copy to Clipboard .....	278
Paste from Clipboard .....	279
Clear Clipboard .....	279
Get Text from Clipboard .....	279
Put Text to Clipboard .....	280
Save Image from Clipboard .....	281
Put Image from File to Clipboard .....	281
<b>18 Mouse .....</b>	<b>282</b>
Move to an object .....	282
Click .....	284
Move .....	285
Move Mouse to Image .....	285
<b>19 Registry .....</b>	<b>288</b>
Add Registry Key .....	288
Delete Registry Key .....	289
Rename Registry Key .....	289
Read Registry Value .....	290
Write Registry Value .....	291
Delete Registry Value .....	292
Rename Registry Value .....	293
Export from Registry .....	294
Import to Registry .....	295
Registry loop .....	296
<b>20 Network .....</b>	<b>297</b>
Send Command .....	297
Send Network Message .....	298
Check TCP port .....	299
Network Plugin settings .....	300
<b>21 Email .....</b>	<b>301</b>
POP3 Open .....	301

POP3 Close .....	303
POP3 Mail Loop .....	303
POP3 Download Email .....	306
POP3 Delete Email .....	307
POP3 Messages Count .....	308
Parse Email .....	308
IMAP Open .....	312
IMAP Close .....	313
IMAP Select Mail Folder .....	313
IMAP Mail loop .....	314
IMAP Download Message .....	317
IMAP Get Message Flags .....	319
IMAP Set Message Flags .....	319
IMAP Delete Marked Messages .....	320
IMAP Copy Message .....	320
IMAP Folder Info .....	321
IMAP Folder List .....	322
<b>22 Strings .....</b>	<b>322</b>
STR Replace .....	323
STR Trim .....	324
STR Delete .....	325
STR Insert .....	325
STR Extract .....	326
STR Pos .....	327
STR Between .....	328
STR Length .....	330
STR format CSV line .....	331
TXT Number of Lines .....	332
TXT Extract Line .....	333
TXT Delete line .....	334
TXT Insert line .....	335
TXT Comma-Text Conversion .....	336
TXT UTF-8 Conversion .....	338
Operations on lists .....	339
<b>23 Embedded Chromium Browser .....</b>	<b>341</b>
Browser Fill Web Form .....	342
Add or Edit Web Form Parameter dialog.....	347
Browser Web Page Information .....	349
Browser Click Link .....	353
Browser Wait for Done .....	357
Browser Page Snapshot .....	360
Browser Window Size .....	362
Browser Run Javascript .....	363
<b>24 Regular Expressions .....</b>	<b>364</b>
RegExp Loop .....	365
RegExp Replace .....	367
RegExp Match .....	371
RegExp COM object .....	373
<b>25 Screenshot .....</b>	<b>376</b>
Window Snapshot .....	376
Window Pixel .....	381
Screenshot .....	383
Screen Pixel .....	384



Color Compliance .....	385
Wait for Pixels .....	386
Find Pixel on Image .....	392
<b>26 INI files .....</b>	<b>394</b>
INI read value .....	395
INI write value .....	397
INI delete value .....	398
INI delete section .....	399
INI loop .....	400
<b>27 XML Documents .....</b>	<b>401</b>
XML Create Node .....	402
XML Add Childnode .....	403
XML Loop .....	404
XML Save .....	407
XML Get Single Node .....	408
<b>28 Script tools .....</b>	<b>413</b>
Python Script .....	413
RoboTask object for Python.....	416
JavaScript/Node.js script .....	419
RoboTask object for node.js.....	422
PowerShell script .....	426
Script extesions .....	429
JS script .....	430
JS Evaluate .....	432
VB Script .....	434
VB Evaluate .....	437
<b>29 Excel .....</b>	<b>439</b>
Excel Open Document .....	439
Excel Get Cells .....	440
Excel Set Cells .....	441
Excel Cell Format .....	442
Excel Run Macro .....	444
Excel Save Document .....	445
Excel Save As .....	445
Excel Close .....	446
<b>30 Sound Volume .....</b>	<b>446</b>
Set Sound Volume .....	447
Get Sound Volume .....	447
Set Mute Status .....	447
Get Mute Status .....	448
Sound Volume Up .....	448
Sound Volume Down .....	449
<b>31 DataBase .....</b>	<b>449</b>
DB Open .....	449
DB Close .....	450
DB Exec SQL .....	450
DB Loop .....	453
DB Begin Transaction .....	457
DB Commit .....	457
DB Rollback .....	457
<b>32 CSV Data .....</b>	<b>457</b>
CSV Open Data .....	458

CSV Get Data Information .....	460
CSV Add Column .....	464
CSV Remove Column .....	465
CSV Sort by Column .....	466
CSV Move Column .....	468
CSV Read Cells .....	469
CSV Write Cells .....	470
CSV Merge Data .....	471
CSV Filter Data .....	474
CSV Data Loop .....	475
CSV Remove Duplicates .....	477
CSV Save Data .....	479
SQL Query to CSV File .....	481
<b>33 Graphics .....</b>	<b>486</b>
Open Image .....	486
Save Image .....	487
Resize Image .....	488
Crop Image .....	489
Rotate Image .....	490
Draw Text on Image .....	490
Draw Image on Image .....	492
Add Border to Image .....	493
Generate QR Code .....	494
Scan QR/Bar Code .....	495
<b>34 WebSockets .....</b>	<b>496</b>
WS Connection Manager .....	497
WebSocket Server Send Message .....	503
WebSocket Client Send Message .....	505
WebSocket Server HTTP Response .....	506
Start/Stop WebSocket Server .....	509
Connect/Disconnect WebSocket Client .....	509
<b>35 Console Application .....</b>	<b>510</b>
Open Console Session .....	511
Close Console Session .....	513
Write To Console Session .....	513
Read from Console Session .....	514
Wait for Text .....	514

## **Part V Task Triggers 516**

<b>1 Scheduler .....</b>	<b>517</b>
How to enable wake timer .....	518
<b>2 Hot Keys .....</b>	<b>520</b>
<b>3 Tray Icon .....</b>	<b>521</b>
<b>4 Cyclic .....</b>	<b>522</b>
<b>5 Process Watcher .....</b>	<b>523</b>
<b>6 File Monitor .....</b>	<b>524</b>
<b>7 Dial-Up Connection Monitor .....</b>	<b>530</b>
<b>8 Expression monitor .....</b>	<b>531</b>
<b>9 Window Watcher .....</b>	<b>531</b>

---

10 On Time Changed .....	534
11 On Battery Power is Low .....	534
12 On Spooler Changed .....	535
13 On Device Changed .....	535
14 On Palette Changed .....	536
15 On Display Changed .....	536
16 On Log Off .....	536
17 On Idle .....	536
18 On Resume .....	537
19 On High Memory Load .....	537
20 On High CPU Load .....	537
21 On Switching to Battery .....	538
22 On AC Power Restored .....	539
23 On Local/Remote Session Changes .....	540
24 Clipboard Monitor .....	542
25 Registry Monitor .....	543
26 Listener .....	545
27 Pixels Monitor .....	546
28 WebSocket Server On Message .....	550
29 WebSocket Client On Message .....	551
30 WebSocket Server On HTTP Request .....	553
<b>Part VI Appendix</b> .....	<b>555</b>
1 Full list of system variables .....	556
2 Command Line Parameters .....	561
3 RoboTaskApp object .....	562
<b>Index</b> .....	<b>570</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. Yes 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.

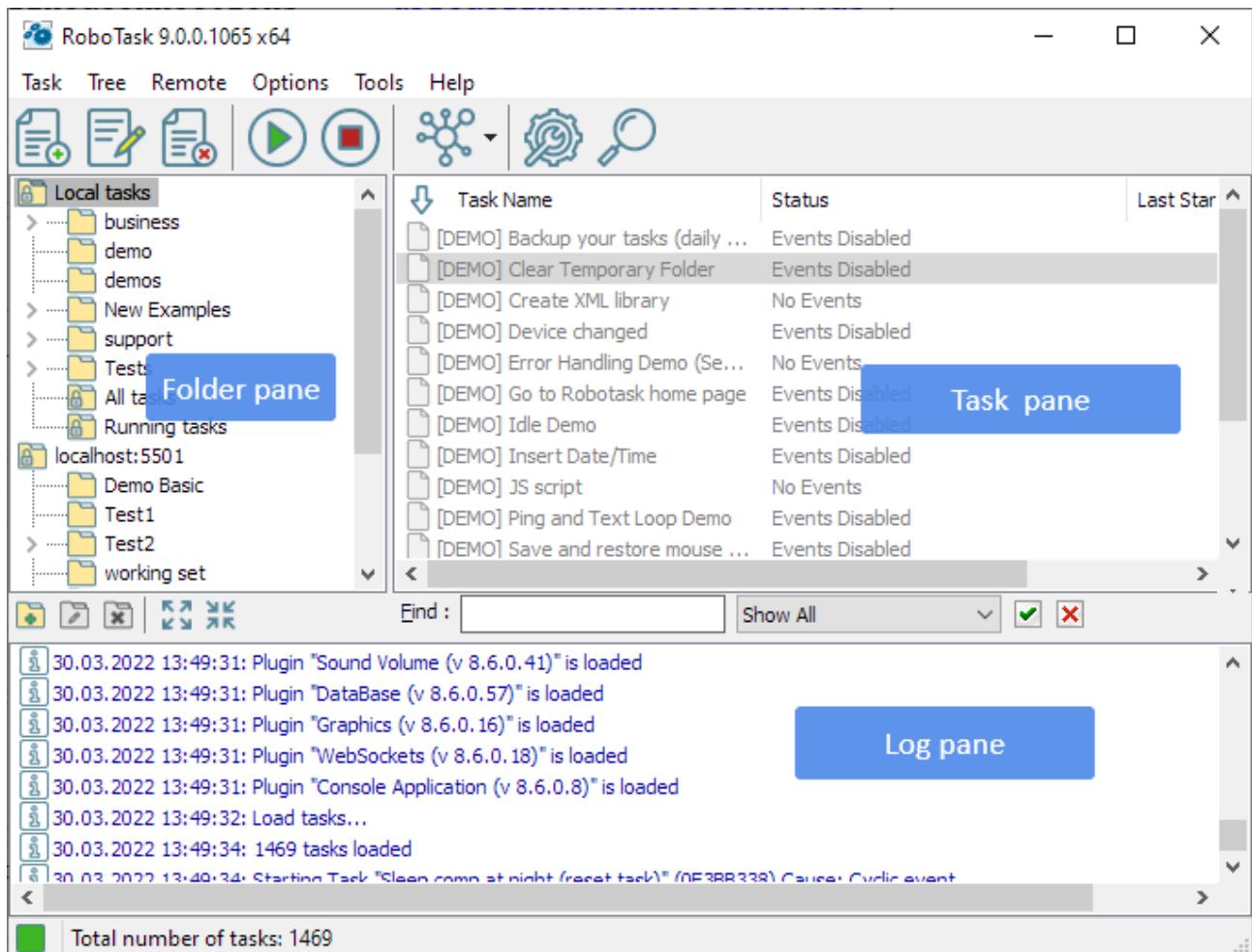
Practically anything that happens on your computer can trigger RoboTask to perform an action. An example would be when RoboTask automatically updates a log and uploads it to a website whenever you receive an email with a subject that is a particular string of words or contains a word such as "Important".

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.

**Connect button**

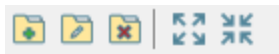
Connect to remote instance of RoboTask

**Settings**

Allows you to set preferences for RoboTask.

**Search task**

Find task by name

**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.

**Collapse 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.
- **Open current log in text editor** - You can open the current log in a text editor for analysis and searching of records. You will be prompted to choose a text editor. The default is the system NOTEPAD.EXE.
- **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** .
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** or press ALT+Enter .
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, 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.
  - The user can install and run *runtime module* to perform pre-made tasks on a separate computer without an additional license.
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 Usage of 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.

#### **The Usage of variable names in actions.**

Beginning with the version of *RoboTask 5.3* all the actions where variables are assigned allow to use other variables in the variable name. That can be useful, for example, when creating enumerated variables like these:

```
Line1  
Line2  
...  
Line10
```

In this case it is possible to specify a variable name like `Line{num}`, where *num* contains the ordinal iteration number of a cycle. Thereby, it is possible to create pseudo array.

*However, it is strongly recommended not to abuse that feature. Although such approach provides more flexibility, it is easy to get confused in variables names. Therefore we advise to specify the variables names explicitly, if it is possible to avoid dynamic names.*

#### **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>[556]</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>[556]</sup> and we will not describe them here. There are more than 80 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>12</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>[56]</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>[208]</sup>

[Set Variable](#)<sup>[209]</sup>

[Remove variable](#)<sup>[210]</sup>

[Increment variable](#)<sup>[211]</sup>

[Delimit Variable](#)<sup>[211]</sup>

Inserting variables into strings in various actions.

For example:

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



Using the methods of the [RoboTaskApp](#)<sup>562</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 variable's 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 Saving the result into a variable

During their operation almost all actions of the RoboTask demand that their results should be assigned into user variables. In this case it is necessary to specify the variable name.

A variable name is some string composed of Latin letters (ASCII symbols) and numbers without any spaces.

It is better to specify the variable name explicitly, that is without curly brackets.

But composing of a variable name using other variables is permitted. In this case variables should be separated with curly brackets like when composing any expression in the RoboTask.

What happens if you use curly brackets in a variable name?

- The RoboTask tries to compute the values of expressions in curly brackets (expand variables) and substitute the value into the expression.
- Then, the got expression is checked whether it is correspond to the rules of variable names.
- If the got name is correct, the RoboTask assigns the necessary value into a variable with this name.

### 2.1.8 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"](#)<sup>[12]</sup>.
- **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}**.

## 2.1.9 Prefixes Used with Variables

Prefixes can precede the name of a variable beginning with version 8.1.

There are 2 kinds of prefixes:

- Context prefix
- Protocol prefix

### Context prefix

A context prefix can have 2 values: **local** or **global**

A prefix precedes the name of a variable and ends with the "@" symbol. It is not case-sensitive.

For example:

```
local@MyVar
global@userinfo
```

When a context prefix is not specified, the general rules of visibility described in the chapter "[Rules for Using Variables in Expressions](#)" (see "The rule of visibility") apply.

### Protocol prefix

Only one protocol, JSON, is currently used. A protocol prefix ends with a colon ":"

For example:

```
json:userinfo.name.firstname
```

If a protocol prefix is not used, this means that we are using the value of the variable entirely as it is. We either extract it while reading or write the specified value.

A *json* prefix means that the *userinfo* variable should be processed as a JSON object and the value at the *name.firstname* destination should be extracted from it.

If you use this expression to write to a variable, RoboTask will accordingly write to the *userinfo* variable the value specified at the *name.firstname* destination as to a JSON object.

If you attempt to read just the value of the *userinfo* variable without prefixes, RoboTask will return the JSON object entirely in the form of a string.

For example:

```
{ "name" : { "firstname" : "John" , "surname" : "Dou" } , "address" : "New  
York, 5-th ave,125" }
```

Please note that the path to the value in the JSON object is case-sensitive. This is a JSON standard. In other words, the expressions

```
json:userinfo.name.firstname
```

and

```
json:userinfo.Name.Firstname
```

are different expressions that can contain different values.

In this way, the JSON format allows storing structured data and arrays of values.

We will skip the format description. For more information, go to the article:

<https://en.wikipedia.org/wiki/JSON>

### Use of Prefixes

Prefixes can be used in any sequence in front of the name of a variable. Both prefixes can be used at the same time.

For example, the expressions

```
local@json:userinfo.name.firstname
```

and

```
json:local@userinfo.name.firstname
```

have the same meaning.

If you specify a wrong prefix, RoboTask will treat this sequence of characters as a part of the variable name and you will most probably get an error message.

## 2.1.10 How to use variables correctly

This chapter is a short resume about variables.

### Usage of variables in expressions

As a rule, when any action is configuring, it is possible to use variables. To specify a variable in an expression (in a string) it is enough to put the variable name within braces "{" and "}".

### How it works.

When any action works the RoboTask tries to expand insets of variables in the expression. In other words, it replaces the variable name with the variable value in the expression.

The algorithm is easy:

At first, the program scans the source line for expressions which are put in braces;

If such expressions are found, the RoboTask tries to find the variable value using the variable name and parameters (if the parameters are specified);

If the variable exists, the RoboTask replaces the expression in braces with the found value;

If such variable doesn't exist, the RoboTask leaves the source expression without any changes.

### Assigning some value to a variable

Overwhelming majority of actions execute certain operations and demand to save some values to data transfer in other actions of the task. Specifying the variable name to save the value, the variable name has to be specified explicitly. A variable name is just a set of Latin symbols and figures without any blanks. Braces are not to be specified in this case.

But there is an exception to this rule. Beginning from the version 5.3 a variable name is possible to be formed with using of other variables. But you must understand why it is necessary.

What happens if you specify braces in the variable name (that is, do insets of other variables in the expression):

- The RoboTask computes the final expression which is the variable name. In other words, it replaces expressions in braces with variable values.
- After that, it checks if it is permissible to use the got name as a variable name. Otherwise, the RoboTask generates warning and skips the operation of assigning value.
- Only after checking, the RoboTask assigns the specified value to the variable with such name.

**Note:** *The usage of such compound names allows to get more flexibility when writing a task in some cases. But if this method is hard to understand, it is better to avoid it. It is necessary only to specify the variable name explicitly without any braces.*

## 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](#) 
- 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.

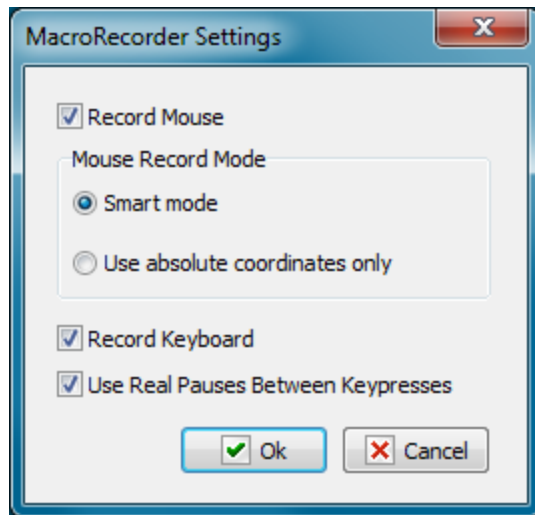
As a rule, the recorded macro is ready to use. However, it is strongly recommended to edit the created steps manually. Macro Recorder records the actions properties exactly as they were made, and that is not always useful for the task creation.

For example, in such action as «[Move to Object](#) » 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).

## 2.2.1 Macro Recorder Settings

You can open the settings dialog of macro recorder by pressing the button 



### Record Mouse

If you want to record actions of a mouse, switch on this option.

### Mouse record mode

The recording of the mouse actions can be performed in two modes:

- **Smart Mode** – in this mode a macro recorder fixes windows and control elements where mouse clicks take place. Also macro recorder records coordinates concerning the window and the control element.
- **Use absolute coordinates only** – a simplified mode of recording. In this mode recording only by absolute coordinates of a screen takes place. This mode can be useful in certain cases when elements of a window are absent.

### Use Real Pauses Between key presses

If you want to record pressings of keyboard keys, switch on this option.

In certain cases it is important to record pauses between pressings of keys. For example, if you use different combinations of keys to open a dialog window. If the pauses are not important, you can switch off this option. For example, when simply typing a text in entry field. In this case the standard 40 milliseconds pause is used (approx 25 pressings per second).

## 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>[245]</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>[12]</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>[12]</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 scripts

Sometimes we need to make some nonstandard calculation, analyze a string or modify it according to a certain algorithm. The script programming languages (VB script or Javascript) will help you in that. The [Script tools plugin](#)<sup>[413]</sup> for RoboTask is included in the main distribution package.

Script languages are simple and at the same time quite powerful programming tool oriented towards both beginners and professionals. You can use script to access user and system

variables in RoboTask, as well as other tasks. The installation file also includes some sample uses.

Almost anyone can learn how to write small procedures in script languages. The use scripts 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 VBscript or JavaScript without any limitations.

## 2.7 Drag and Drop

With the drag-and-drop mechanism, you can not only move tasks between task folders, but also easily export tasks to a file or email message as an attachment. You can also import tasks into RoboTask from a file or email message.

### **Export using drag-and-drop:**

Select one or more tasks from the task list, press the *SHIFT* key and drag the selected tasks to an open Windows Explorer window or email message editor.

### **Import using drag-and-drop:**

Simply drag the task from the Windows Explorer or email client message window to the RoboTask window. It is not necessary for the file to have a *TSK* extension. RoboTask will analyze the contents of the file itself. If the file does not contain a task, RoboTask will ignore the "dropped" file.

## 2.8 Running RoboTask as an NT service

### 2.8.1 Service mode vs User mode

RoboTask can work in two modes:

- As a common application
- As a system service.

Almost everybody knows what a common application is. Many people have an idea about what a system service is. But very often they know only that the service works even if a user has not logged in.

Yes, it is right. A user session is not necessary for a service. It is, undoubtedly, convenient. But it also puts a number of limitations. And it is necessary to remember about this. Let us consider all the pros and cons.

### **Fundamental differences between modes of the service and application for RoboTask.**

#### **Service and application have different sets of tasks.**

RoboTask application keeps its tasks in a user profile. Thereby, every user will have their own set of tasks by default. It is convenient, for example, in the mode of a terminal-server, when

several users work on one server in the mode RDC (Remote Desktop Connection). Every instance of RoboTask will work independently from others and not interfering with others.

RoboTask service keeps tasks in its folder in Program Files. To make RoboTask perform necessary actions in the service mode, it is not enough to simply install the service. *It is necessary to import or create required tasks for the service as well.*

Different sets of tasks allow the application and the service to work simultaneously and independently.

### **Service and application have different environment.**

The application inherits the environment of a user session as any other application which the user has run in their session.

The service uses the environment *LocalSystem*. Despite the fact that you can see both windows application and service on the screen (for example, in Windows 2000 and Windows XP), RoboTask will use different environments in these modes. More details below.

### **Application mode**

After the installing, RoboTask is ready to work in the application mode. By default, RoboTask runs automatically every time when a user opens a user session (log on to Windows).

#### Advantages

- Additional settings are not necessary; RoboTask is ready to work immediately after installing.
- Easy access to the main window and to the settings of RoboTask.
- The work of RoboTask in the user session allows to get access to the same resources which are available to the user (for example, network folders and disks, windows of other applications, etc.). This is inheritance of a user session environment.

#### Disadvantages

- In order that RoboTask may begin to work, it is necessary to log on.
- Logging off the session RoboTask also turns off as all other user applications. No tasks are performed after that.
- In operation system Vista and further ones, the system UAC (User Access Control) demotes the privileges of the started applications. It is done to increase security against malware and viruses.

Problems connected with the UAC manifest themselves in a possible error "Access Denied" during some operations. In case of starting certain applications with elevating of privileges (for example, by means of the utility RG.EXE from the package RoboTask) you have to confirm running of programs manually.

The problems of the UAC can be solved in two ways:

- Turning off the UAC. In this case all the applications in the session will work with user privileges. This mode is possible but not recommended because the security of the system against malware decreases

- Running of RoboTask in the mode "As Administrator". It is enough to turn on the checkbox "Run as Administrator" in the dialog Settings (menu Options -> Settings). But in this case you will have to confirm running of RoboTask manually when the UAC requests.

### **System service mode**

To install RoboTask service you can use the utility "[RoboTask Service Manager](#)"<sup>[28]</sup> from the package RoboTask. The utility allows to install, delete, start or stop the service. Also the utility allows to specify initial common settings of RoboTask for working in the service mode.

The usage of RoboTask as a system service has unquestionable advantages, but it causes some inconveniences as well.

#### Advantages

- RoboTask service begins the work when the system starts. There is no need to log on. It is enough just to turn on the computer.
- By default, all the tasks of RoboTask work with system privileges.

#### Disadvantages

- As a rule, the system service works in a hidden mode. In the system Windows Vista and further systems all interactive services work in a separate system session. A user cannot see the main window of RoboTask without additional manipulations. That complicates manual control of RoboTask.
- By default, services working with the privileges LocalSystem do not have access to network resources (i.e. the network folders on remote computers)
- It is necessary to avoid any interactive actions in service tasks: moving a mouse, pressing mouse keys, sending keystrokes or calling any dialogs.

The first two items cause some inconvenience, but these problems can be solved. See below how to do it.

The last rule is to be fulfilled for the following reasons:

- The system ignores moving the mouse and pressings the keys inside of a locked session. This is done for security of the system. For this reason, the simulation of mouse and keyboard actions will not work if the user session is locked. The user session is locked by default, when screensaver is activated .
- As a rule, a user does not have a direct access to windows of RoboTask in the system service mode. All the dialog windows require to press OK or Cancel. If the user does not do it, the task comes to a stop and waits for a user action endlessly.

### **How to access the service in the OS Windows Vista/7/2008/8/2012**

Access to the main window of service can be got by means of the system service "Interactive Service Detection". That is described in the chapter "[How to access RoboTask Service](#)"<sup>[37]</sup>. Then it is possible to change settings of RoboTask and service tasks.

The owners of Business License can get the access to the service through a local network. That is described in the chapter "[Control of RoboTask through network](#)"<sup>[42]</sup> and related chapters. The work of a network client of RoboTask is integrated into the main window of RoboTask and must not cause any difficulty.

Initial settings of the service (a number of a port, privileges of access, parameters of authentication, etc.) can be set with the utility "[RoboTask Service Manager](#)<sup>[28]</sup>".

### How to access the network resources in the service mode

By default, the RoboTask service works with privileges LocalSystem and does not have any access to network folders. This problem can be solved in two ways:

1. Start a service with privileges of any user. It can be done in system settings of the service. In this case the service inherits privileges of the stated user (including the privileges within the local network). But then interactivity of the service will be disabled.
2. Take notice that the user is to have the permission "Log on as a service". Input correct parameters of the user on the tab Logon Parameters in RoboTask Service Manager. In this case all started tasks by default inherit privileges of the user. The inheritance can be turned off in Task Editor if necessary. See the parameter "Log on Task as User" on the tab "Advanced".

## Summary

Both modes have their own advantages and disadvantages. You can use RoboTask as an application and as a service simultaneously. Thereby, you can fulfill the whole circle of tasks that can be solved by means of RoboTask.

How you will manage the process of automation on your computer or in a local network depends only on your needs.

## 2.8.2 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 use [RoboTask Service Manager](#)<sup>[28]</sup> utility.

RoboTask Service Manager allows to start, stop RoboTask service, and make the initial adjustment of the service.

### 2.8.3 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.

### 2.8.4 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.

Also [RoboTask Service Manager](#)<sup>[30]</sup> allows you to make this adjustment too.

#### **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".

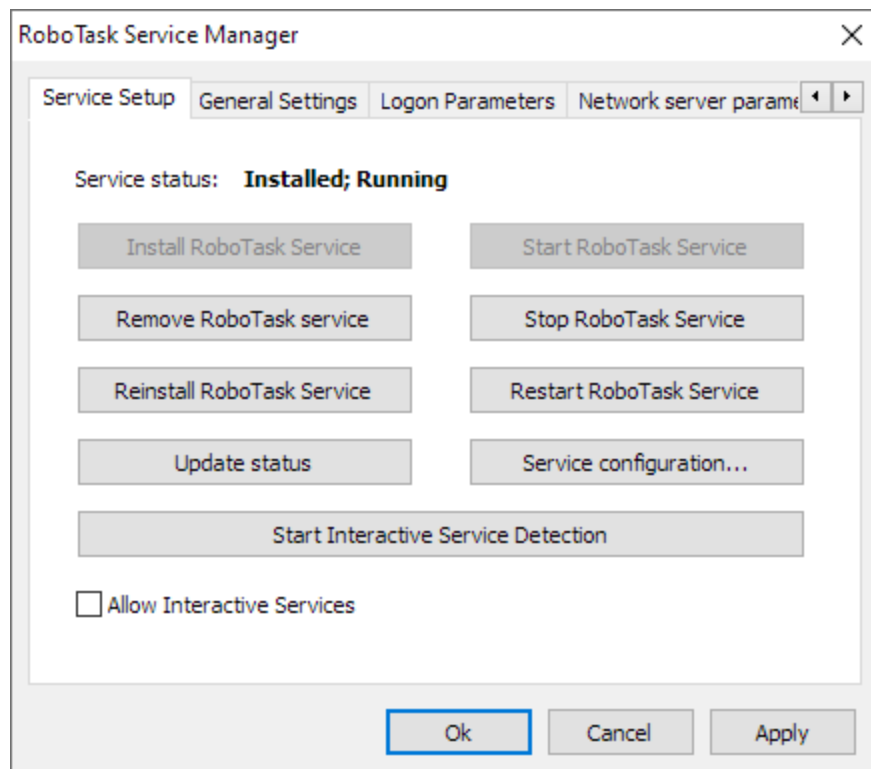
## 2.8.5 RoboTask Service manager

The RoboTask Service Manager is a separate utility within the installation package of RoboTask. The RoboTask Service Manager allows to install the RoboTask Service and perform the necessary initial settings of the service.

The functions of the service setting are separated according to their subjects onto different tabs.

### Service Setup

The functions of installing, starting and stopping of a service are located on this tab. It is also possible to change system settings of the service and to get access to the main window of the service here.



#### **Install RoboTask Service.**

Installs the RoboTask Service in your system if the service has not installed yet.

#### **Remove RoboTask Service.**

Removes the RoboTask Service from a local system.

**Reinstall RoboTask Service.**

Reinstalls the service with default system settings.

**Update Status.**

Updates displaying of service status in the window.

**Start RoboTask Service.**

Starts the RoboTask Service on a local computer.

**Stop RoboTask Service.**

Stops the RoboTask Service on a local computer.

**Restart RoboTask Service.**

This operation stops the RoboTask Service, pauses for a few seconds and then starts the service again.

**Service configuration**

You can perform system settings of the service here as well as in the system Service Manager (Control Panel => Administrative tools => Services). For ease of operation, the settings of the RoboTask Service are included into this utility.

**Start Interactive Service Detection**

This function is destined specially for operation systems Windows Vista and the later ones (Windows 7, Windows Server 2008, Windows 8). The function allows to start and stop Interactive Service Detection

The RoboTask service is an interactive service. This means that a user can get an access to the service window and do settings interactively. In the OS Vista (and in later OSs) the system hides interactive services by default. To get the access to the system window it is necessary to start the system service Interactive Service Detection. After that you can get an access to the session SYSTEM and then work with the window RoboTask Service interactively.

This option is not available on Windows 10 and later

**Allow Interactive Services.**

To get an access to the service window, interactive services must be allowed in the system. In the OS Windows 7/ Server 2008 R2 (and previous OS) interactive services are allowed by default. In systems beginning from Windows 8 interactive services are *forbidden* by default.

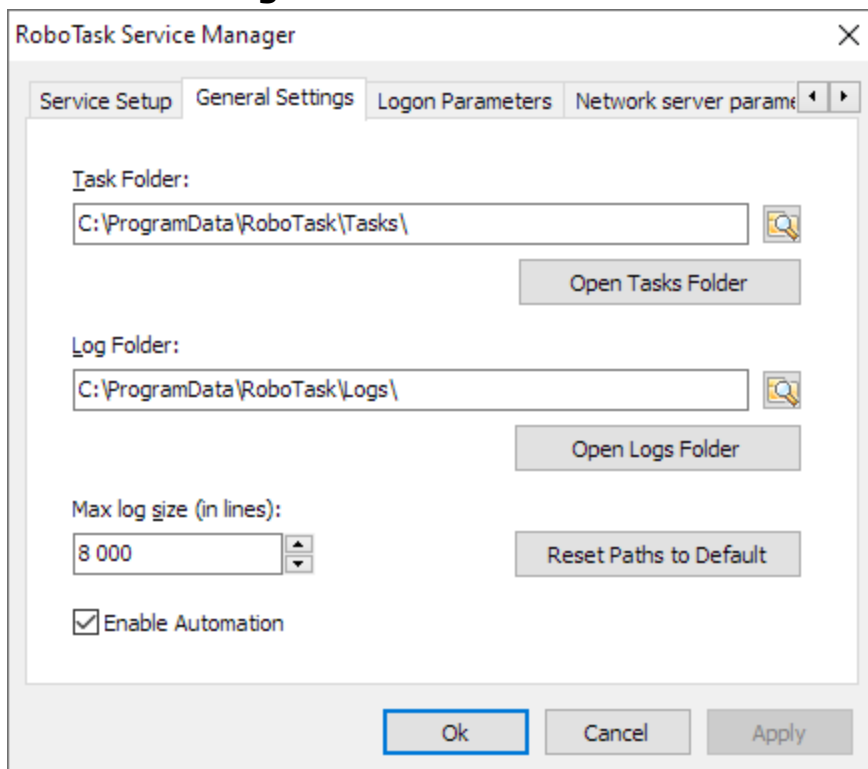
This option does not prevent RoboTask service from working. The service starts and works normally in any case. But if the option is off, you cannot get any access to the main window of the RoboTask service. If interactive services are not allowed, you will get an error when starting Interactive Service Detection.

To allow interactive services in the system it is enough just to switch on this checkbox.

This option is not available on Windows 10 and later



## General Settings



You can find general settings for the service on this tab:

- Location of tasks and logs.
- The maximum length of log-files (in lines).
- Switching on and off the automatic work of tasks.

You can also find these settings in the dialog [Settings on the General tab](#)<sup>68</sup>.

## Logon Parameters

The screenshot shows the 'RoboTask Service Manager' dialog box with the 'Logon Parameters' tab selected. The dialog has four tabs: 'Service Setup', 'General Settings', 'Logon Parameters', and 'Network server parameters'. The 'Logon Parameters' tab contains the following elements:

- A checkbox labeled 'Log on Process And Tasks As Specified User' which is currently unchecked.
- A text field labeled 'Domain Name (use dot "." for local computer):' containing a single dot '.'.
- A text field labeled 'User Name:' containing the text 'svcuser'.
- A password field labeled 'Password:' containing ten dots '.....'.
- A 'Try to Log on' button.
- At the bottom of the dialog are three buttons: 'Ok', 'Cancel', and 'Apply'.

### **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**

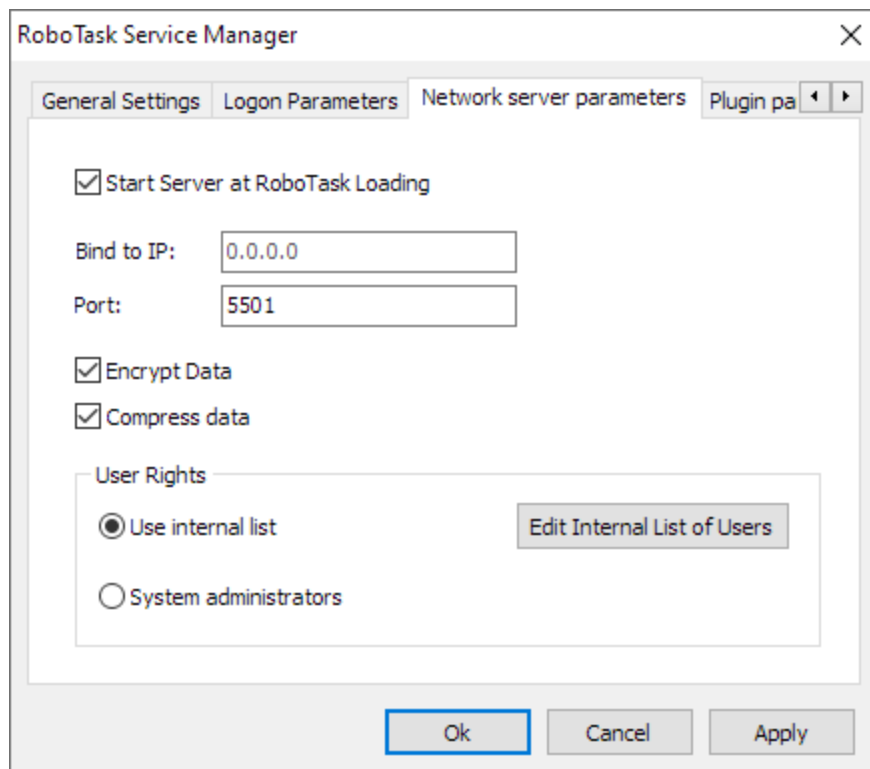
Enter the name of the user account to log on to.

### **Password**

Enter the password of the user account.

### **Network Server Parameters.**

On this tab you can set up server parameters of RoboTask. Read more about client-server mode [here](#)<sup>42</sup>.



### Start Server at RoboTask loading

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

### Bind to IP

By default this field is empty. This means that the network server is bound to all available addresses on the computer. The address 0.0.0.0 has the same effect. You can specify particular address. This is useful if the computer/server has several IP addresses.

Also you can specify local host address 127.0.0.1. In this case you can connect to server from local computer only.

### 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

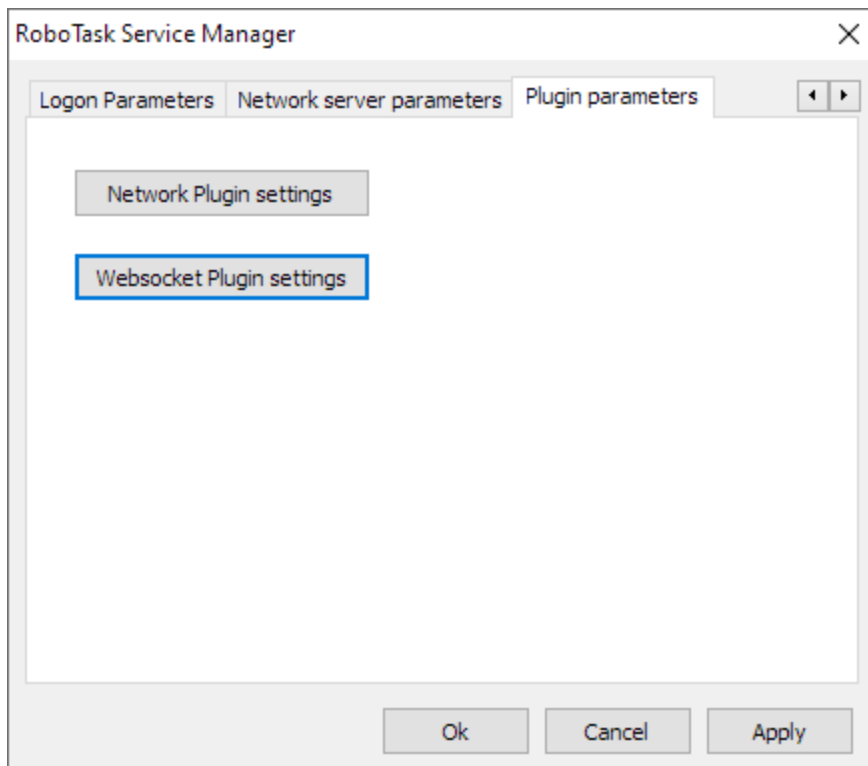
This is the check mode of user's rights at connection. It is possible to choose between two modes of user checking:

- 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.

### Plugin parameters



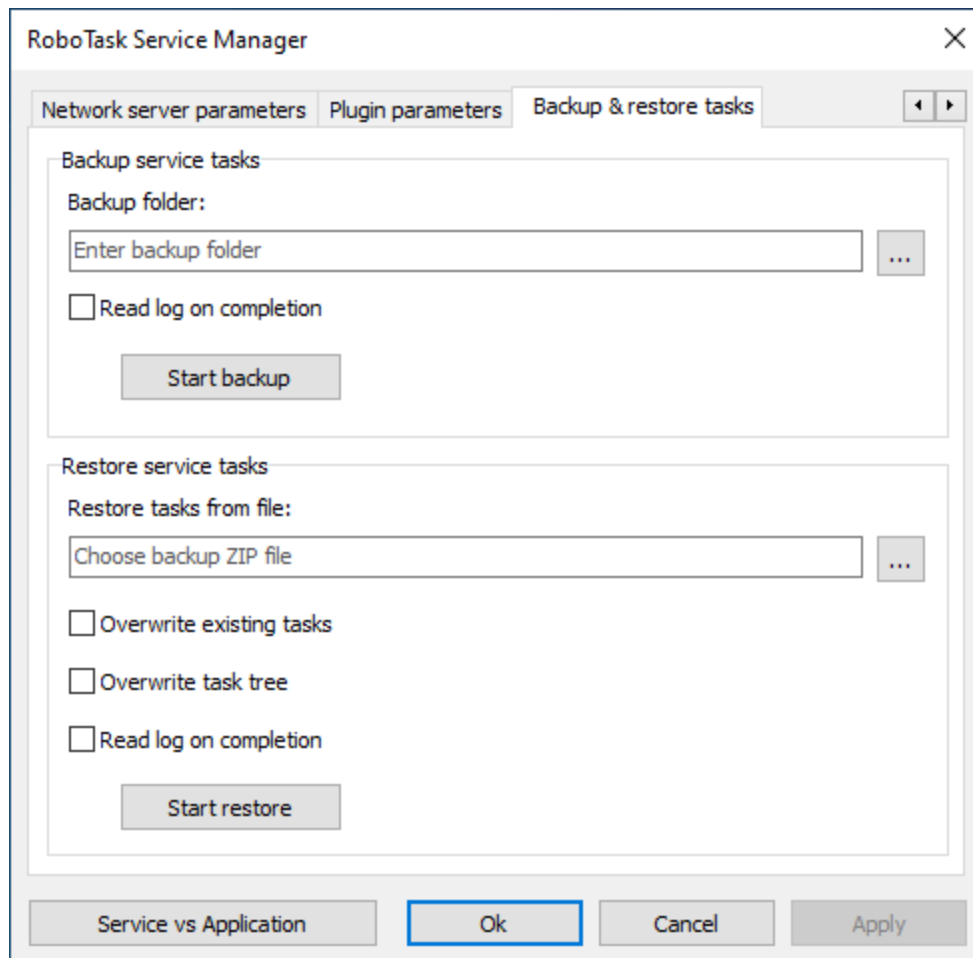
### Network Plugin settings

The form is the same as [the settings of the Network plugin](#)<sup>[300]</sup>.

### WebSocket Plugin Settings

The form is the same as [WS Connection Manager](#)<sup>[497]</sup>

## Backup & restore tasks



### Backup service tasks

Backing up tasks consists of packing the current set of tasks into a ZIP archive. If necessary, it is possible to restore tasks from a saved archive.

Backup is performed using a predefined task launched in the environment of runtime module.

### Backup folder

Select a folder to save the backup files. The default folder is in the user profile:

```
C:\Users\\RoboTaskBackup
```

If necessary, you can select another folder. The selected folder is saved for future work sessions.

The archive file name is generated automatically using the current date and time:

```
RoboTask- Service-Tasks-YYYYMMDD-hhmm.zip
```

where YYYYMMDD – date in "YYYYMMDD" format

### **Read the log when completion**

Check this box if you want to read the task execution log.

### **Restore service tasks**

Restoring tasks consists in unpacking tasks from a backup file into the tasks folder.

Restoration occurs with the help of a predefined task launched in the environment of runtime module.

When the task is restored, the *RoboTaskNTService* service is restarted.

### **Restore tasks from file**

Select backup zip file

### **Overwrite existing tasks**

If you want to overwrite existing tasks, then enable this checkbox.

### **Overwrite task tree**

If you want to overwrite an existing task tree, then enable this checkbox.

### **Read the log when completed**

Check this box if you want to read the task execution log.

## **2.8.6 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>36</sup>

[Peculiarities of RoboTask Service workin in RDC mode](#) <sup>36</sup>

[Peculiarities of RoboTask Service working under OC Vista/ Windows 7/ Windows 2008 Server](#) <sup>36</sup>

### 2.8.6.1 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*)

### 2.8.6.2 Peculiarities of RoboTask Service working 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 /?
```

### 2.8.6.3 Some peculiarities of the RoboTask service in Vista+ OS

Operation systems Vista/ Windows 7/8/Windows 2008/2012 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, or 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.

#### 2.8.6.4 How to access RoboTask Service

As we know Windows Vista+ operating systems prohibits the interaction with service applications by default.

Windows 10 doesn't allow interaction with services at all

How to access RoboTask service in this case?

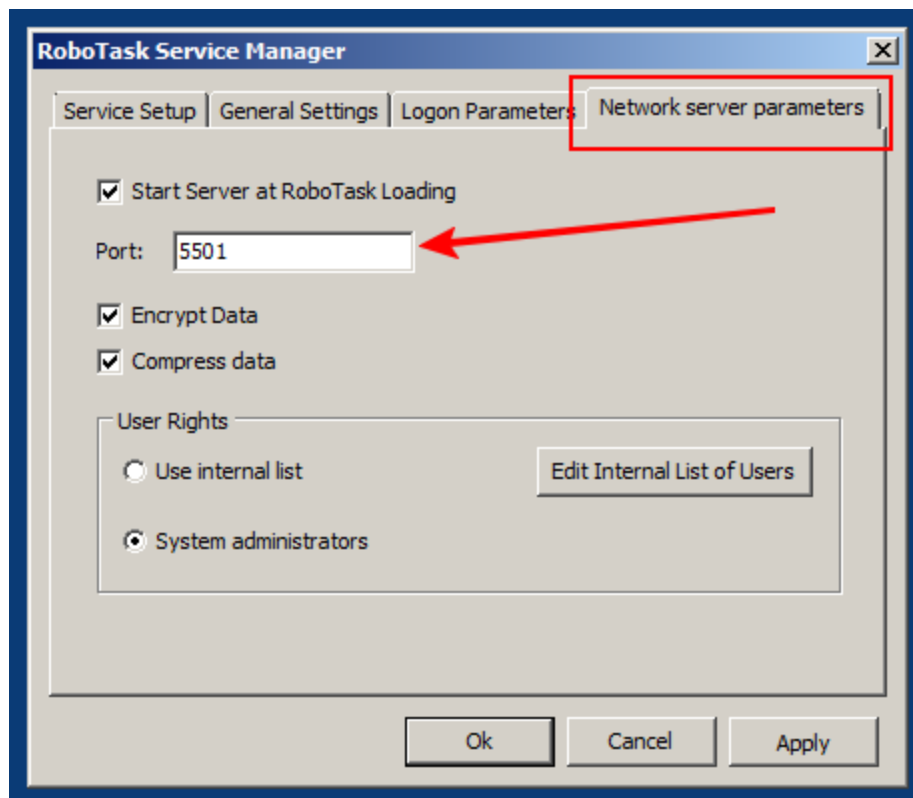
#### **You can access RoboTask service by using network client.**

This method is universal. It works on any version of Windows or Windows Server

Run RoboTask Service manager, and setup network server for RoboTask Service. Pay attention that port **MUST be different** from network server settings of application

I recommend you to use the next number - 5501

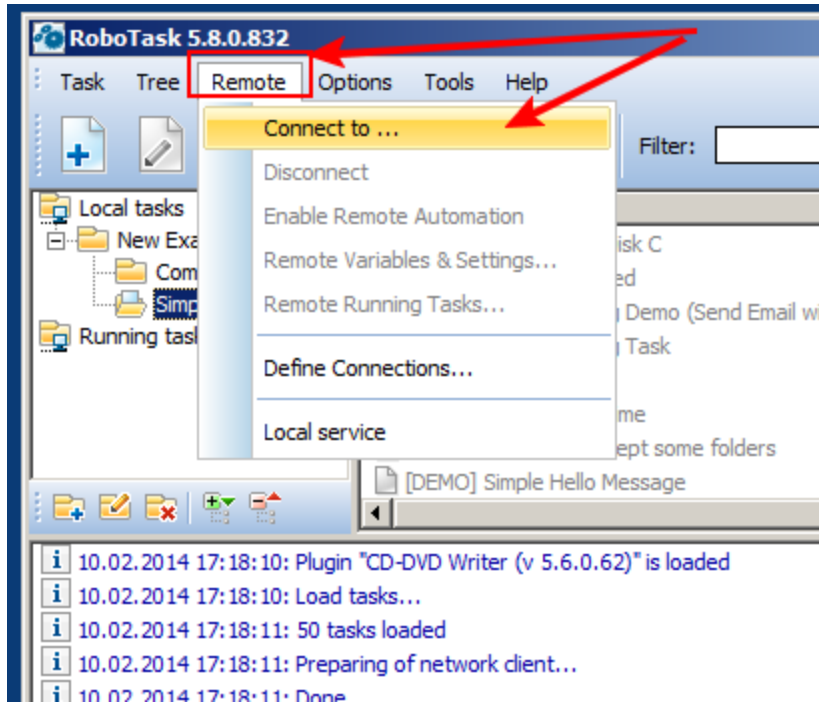
press "Apply" button to save settings and restart RoboTask service.



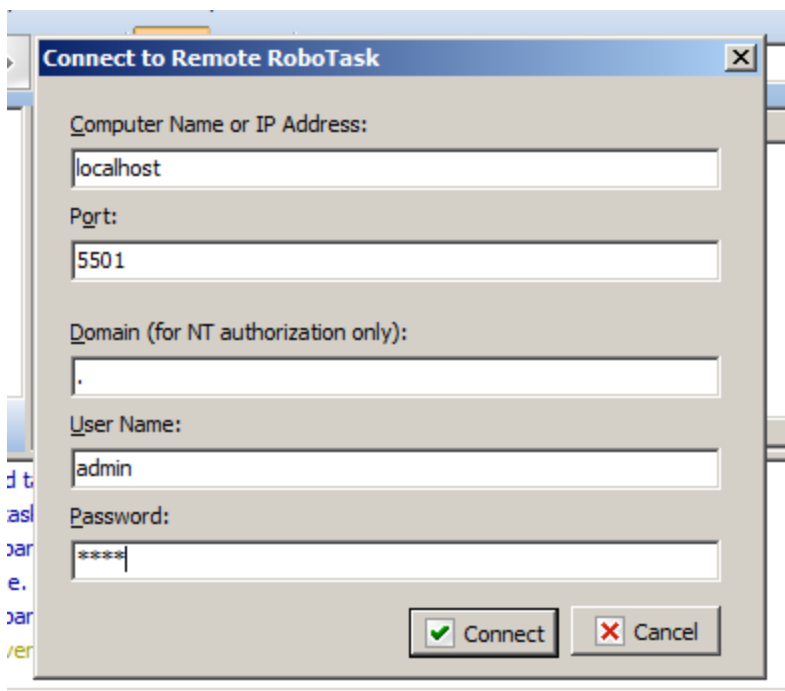


You can close close RoboTask service manager and open main window of RoboTask application.

Choose menu Remote->Connect to...

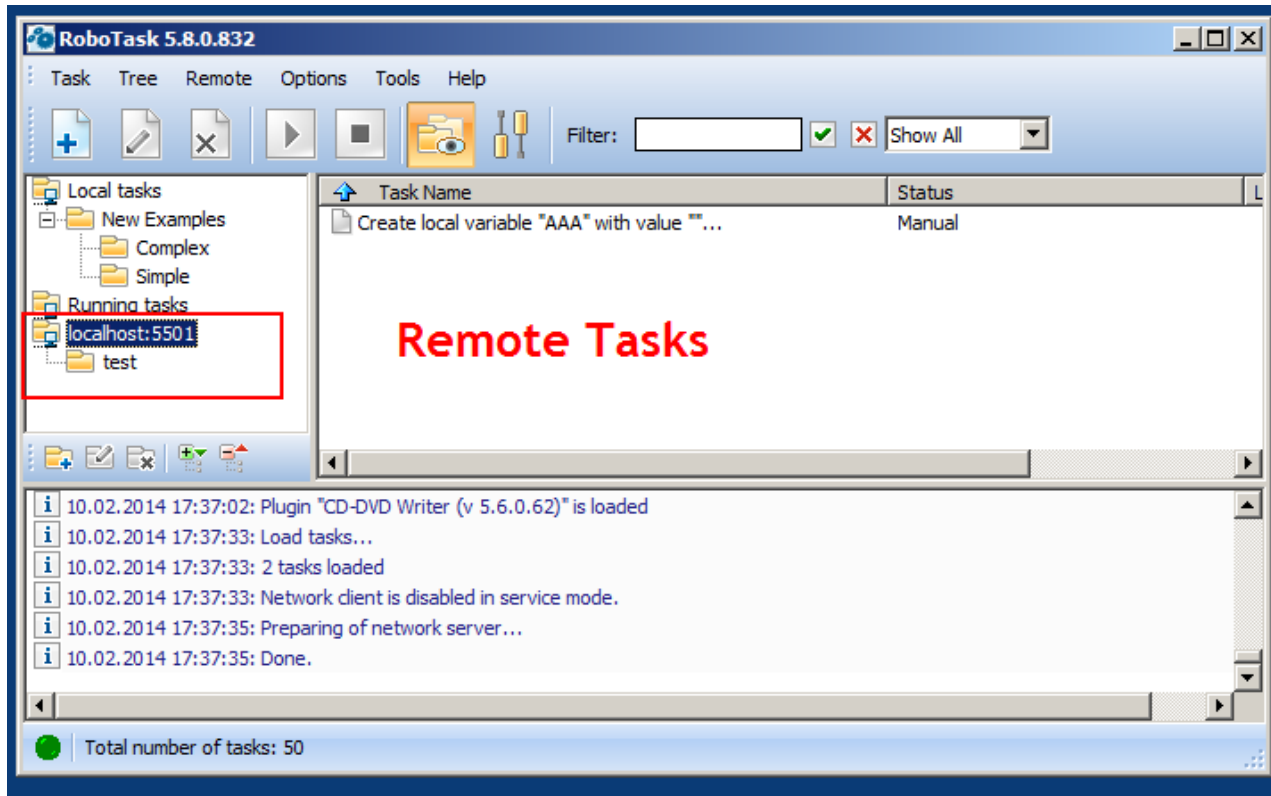


Input necessary parameters and press "Connect"



You can see new Root folder in the task tree: localhost:5501

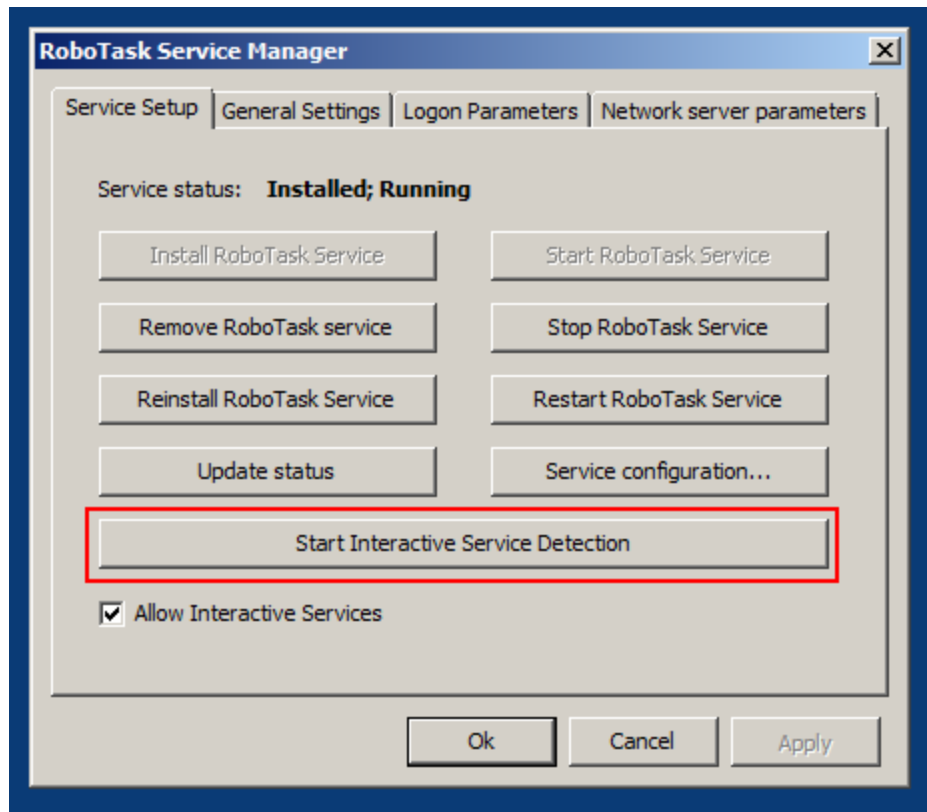
You can select it and edit remote tasks (also you can import, run interactively and remove tasks)



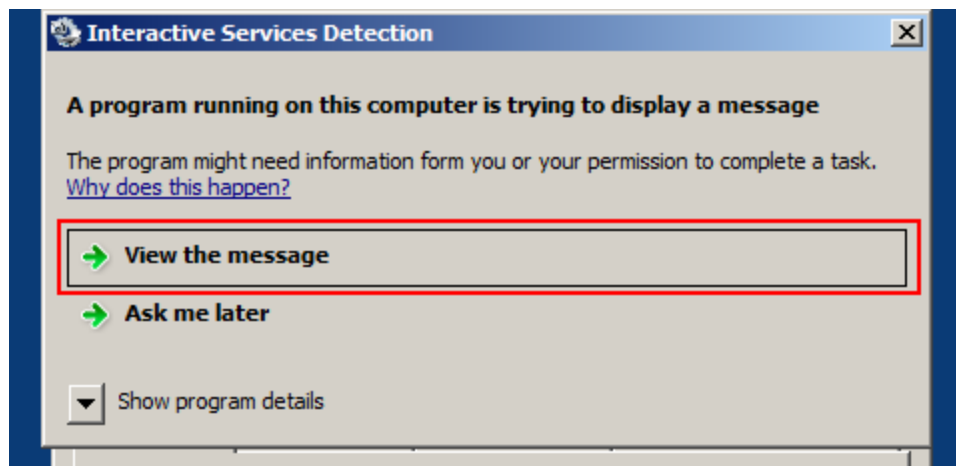
### By using "interactive service detection" service

(this method doesn't work under *Windows 10+* and *Windows server 2016+*. [Use network please instead](#)<sup>[37]</sup>)

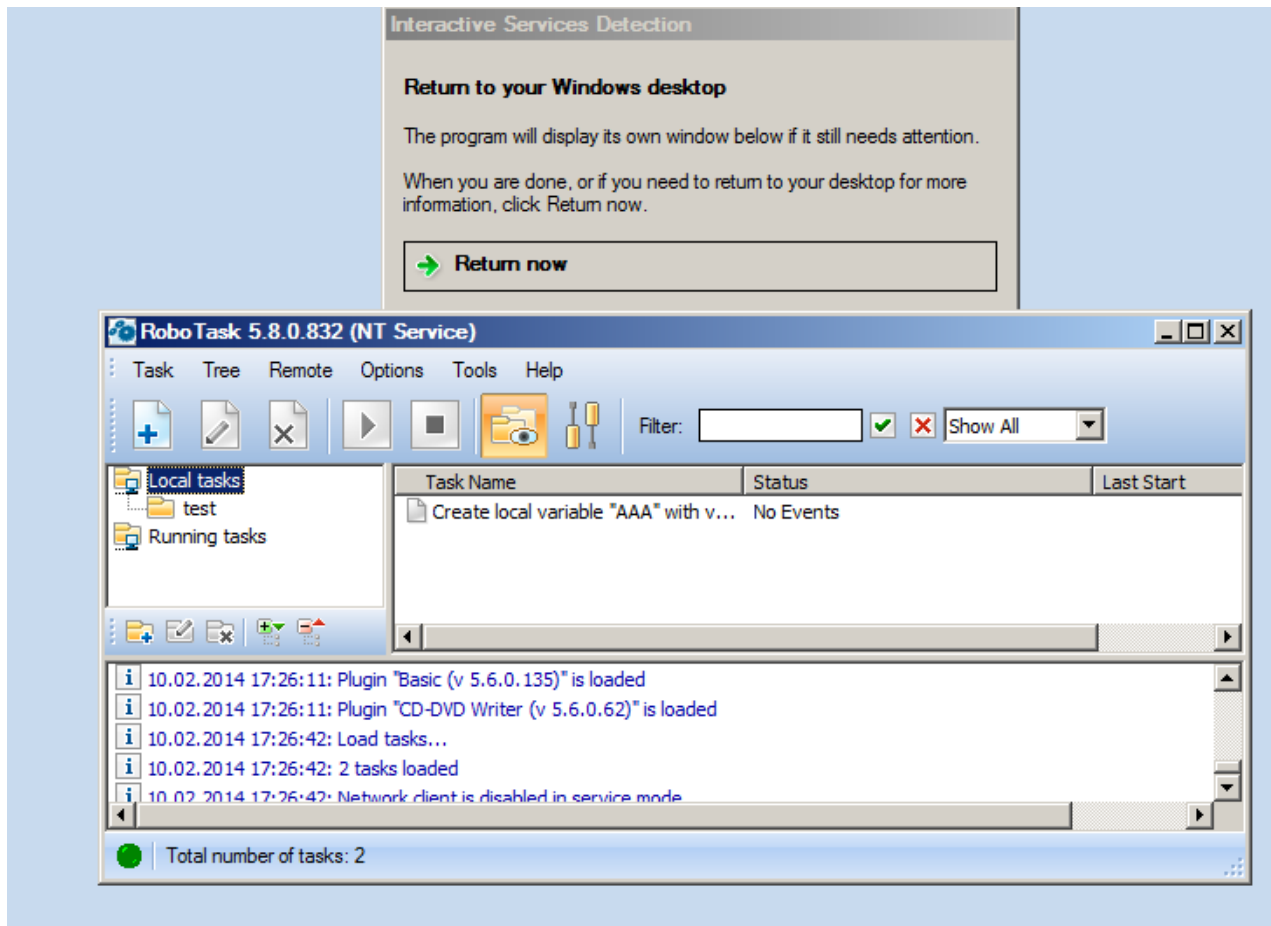
- start [RoboTask Service Manager](#)<sup>[28]</sup> and start "Interactive service detection". Pay attention that parameter "Allow Interactive Services" must be turned on. Otherwise turn it on and restart RoboTask service.



- You will see this message. Press "View the message"



- You can see RoboTask window in the system session. Configure tasks interactively (or import prepared tasks) and press Return Now button on system dialog.



## 2.9 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.

The variables *LastErrorCode* and *LastErrorDescription* can be used in the task in which the error occurred.

The conditions of the using are similar to the variable *IsError*: it is necessary to set the parameter *If Error Occurred* into the value *Continue Execution*, to not interrupt the performing of the task when an error appears.

## 2.10 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](#)<sup>[367]</sup>). 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 allows 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.

## 2.10.1 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 [Connection manager](#)<sup>[61]</sup> (menu Options | Settings) on the tab "[Network Server](#)<sup>[62]</sup>".

### Start Server at RoboTask loading

It is on by default. Switch off this checkbox if you want to refuse from remote [administration](#)<sup>[44]</sup>.

### 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.

## 2.10.2 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.

### 2.10.2.1 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**

### 2.10.2.2 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 | Connection manager](#)<sup>[61]</sup> 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.

### 2.10.3 Network Command utility

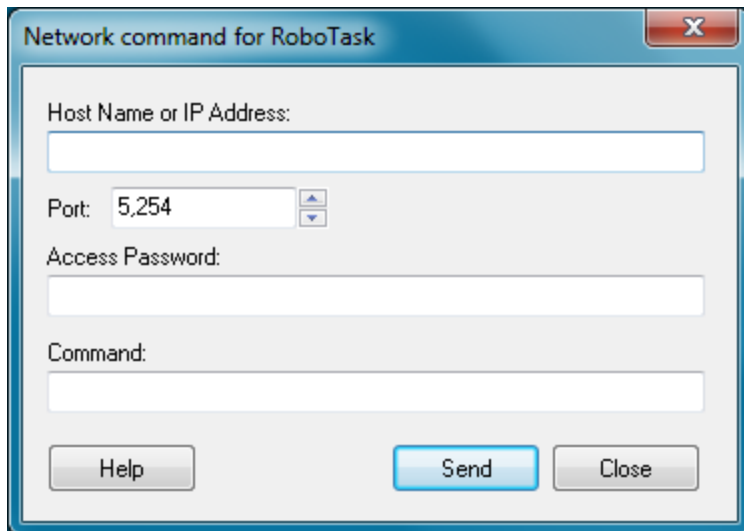
This is an independent utility mimicking the work of the action [Send Command](#)<sup>[297]</sup>. Now to run a task remotely with the trigger [Listener](#)<sup>[545]</sup>, you need only copy and use this utility on the other computer. The utility is autonomous (portable) and does not need any special installations.

The utility has two modes of work:

- Interactive
- From the command line

#### Interactive parameters

If run the utility without any parameters, the dialog window of parameters will load.



#### 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.

### Help

Shows description of command-line parameters

### Command line format

#### Usage:

```
NetWorkCommand.exe srv=ServerName cmd="any string" [psw=password]  
[port=PortNumber] [silent=yes|no]
```

#### Parameters:

**srv** - 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). By default port=5254.

**psw** - Access Password. Specify the password for connecting to remote listener. If the Listener on remote server is configured to work without a password, leave this field blank.

**cmd** - Specify the string (command) that should be sent to remote listener. It can be any string. The "Listener" triggering event on the remote computer will analyze the string. If the command contains spaces you should include the string between quotes. For example: cmd="Some random string".

**silent** - By default is Yes. Specify "No" if you want to see error messages or final dialog box. Also you may specify such values: Yes/No, Y/N, True/False, T/F, 1/0

## 2.11 Runtime module

The runtime module is designed to run a prepared task without the main RoboTask application and additional license. However, the runtime module uses the entire engine to run the task.

Command line:

```
RoboTaskRuntime.exe <Task_File_Name> [/stop]
```

<Task\_File\_Name> - full name of task file

/stop - (optional parameter) if you want to stop the application before exiting anyway to read the log

The application executes the task and exits during normal operation. When an error occurs, the application does not finish automatically and offers to read the task execution log.

It should be remembered that the runtime module guarantees the correct execution of autonomous tasks. That is, there are some restrictions:

- The runtime module does not have access to other RoboTask tasks. Accordingly, it cannot perform actions with other tasks: launch, enumerate, check the status, etc.
- The runtime module does not have access to RoboTask global variables.
- The runtime module does not process the task triggers. The task can only be started from the command line.

To install the engine only with runtime module on separate computer use the command line of setup program:

```
RobotaskSetup.exe /COMPONENTS=engine
```

By default setup application installs all modules and components.

## 2.12 Useful Utilities in RoboTask Package

### **CheckForUpdates.exe**

Checking for a new version on RoboTask's website. Notifies you when your subscription to updates expires according to your license.

### **InputRegKey.exe**

Provides activation of the license key on the computer.

### **NetWorkCommand.exe**

A standalone utility for sending a command to a [Listener trigger](#)<sup>545</sup> from a remote computer. So you can run the task remotely from any computer, as long as a network connection to the RoboTask instance is possible.

This utility is described in detail in the "[Network Command utility](#)<sup>45</sup>" chapter.

The utility is completely self-contained and does not require activation.

### **RTServiceMGR.exe**

---

The RoboTask Service Manager is a separate utility within the installation package of RoboTask. The RoboTask Service Manager allows to install the RoboTask Service and perform the necessary initial settings of the service. [Read more here](#)<sup>28</sup>

### **RG.exe**

Allows you to run the command line with administrator privileges. Please note that you will need manual confirmation to allow the operation, if UAC (User Access Control) is enabled in the system and the current user is not an administrator. UAC is enabled by default.

Usage:

```
RG.exe [/p] <command_line>
/p - RG waits when the command will be finished
```

### **RoboTaskRuntime.exe**

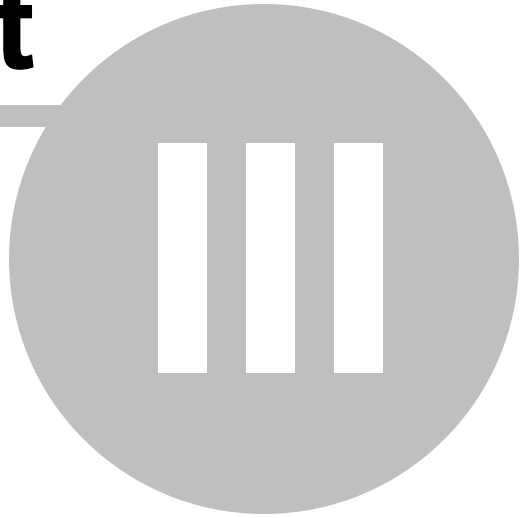
The runtime module is designed to run a prepared task without the main RoboTask application. However, the Runtime module uses the entire engine to run the task.

Usage:

```
RoboTaskRuntime.exe <Task_File_Name> [/stop]
<Task_File_Name> - full name of task file
/stop - (optional parameter) if you want to stop the application before
        exiting anyway to read the log
```

# Part

---



## Menus

## 3 Menus

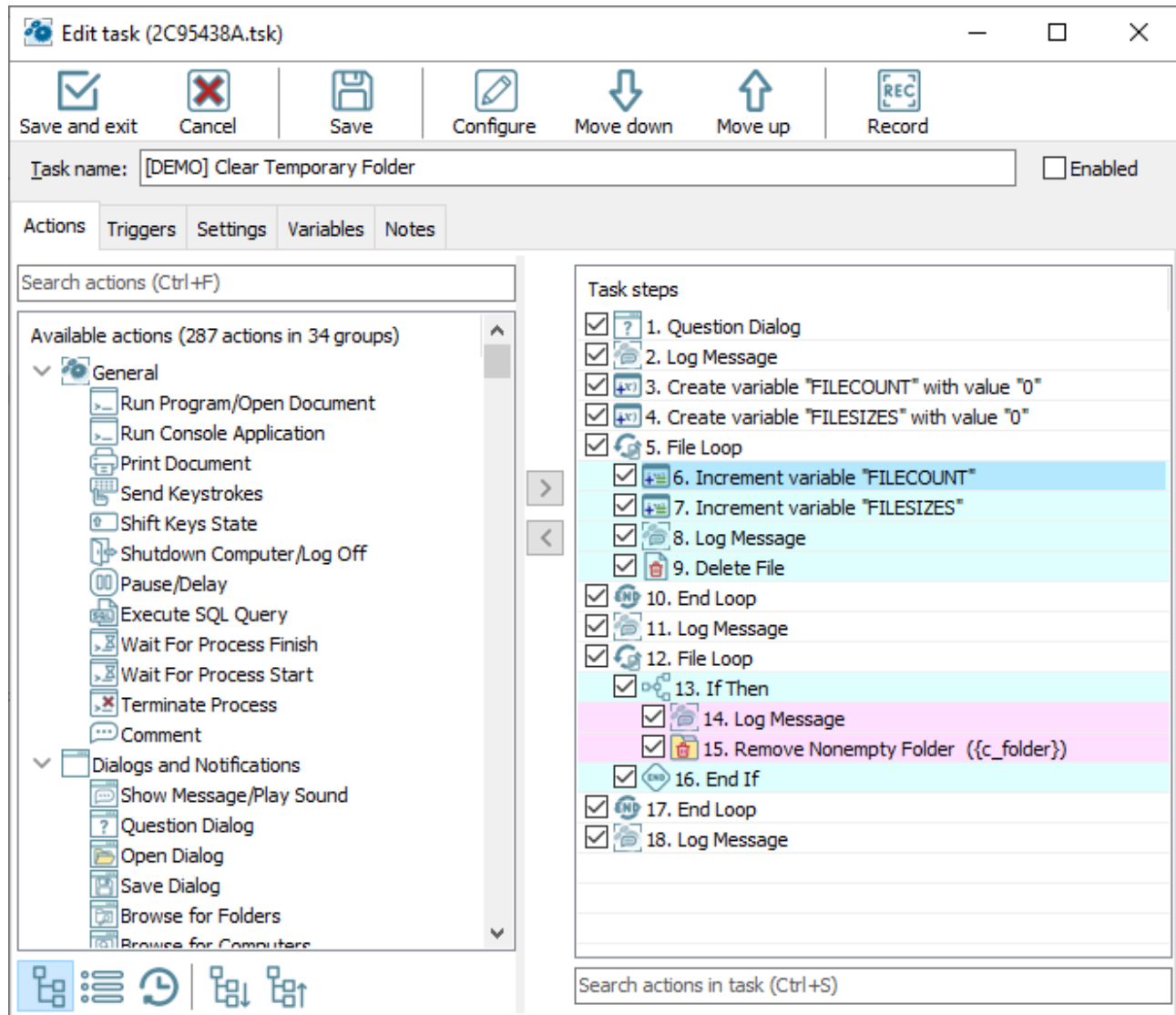
### 3.1 Task

#### 3.1.1 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](#) 51
- [Triggers tab](#) 53
- [Settings tab](#) 54
- [Variables tab](#) 56
- [Notes tab](#) 57

## Actions tab



### Save and exit

Save the task and close the editor

### Cancel

Close the task editor without saving.


### Save

Save the task. Then you can continue editing.


### 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.

### Record a macro

In order to record mouse actions "as is" click this button . [Macro recorder](#)<sup>18</sup> catches mouse actions and saves them as a sequence of task actions.

### Task name

Enter the name of task

### 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.


### Available actions


List of available actions that you can use in the task.

### Task steps



Displays a list of the actions you have selected for the task you are creating or modifying.

To add step select the action from **available actions** and click the **add** button . You can also drag the desired action to the list of steps.

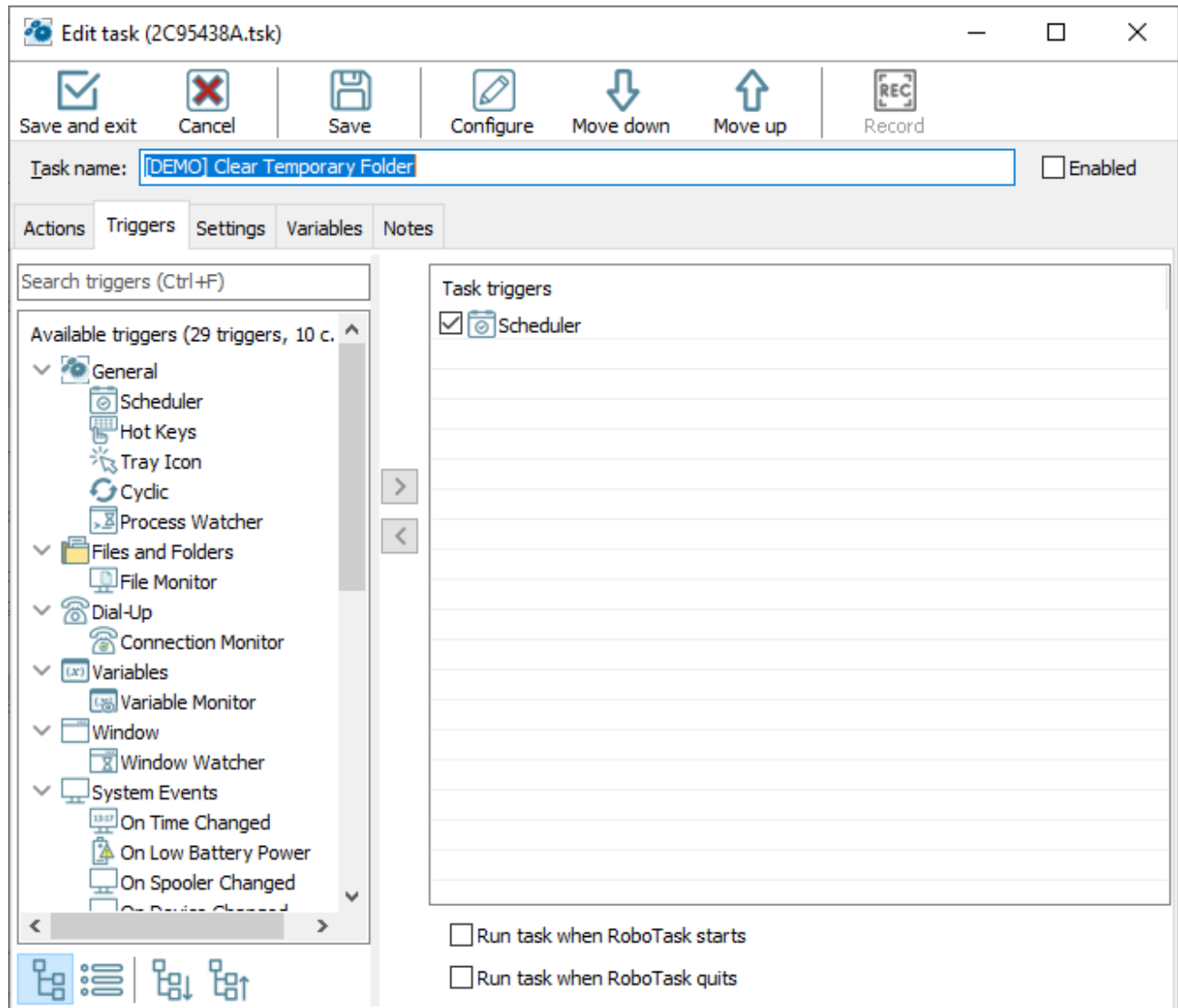
Select an action in the **task steps** and click the **configure** button  to modify the action parameters.

To remove step select the step from **task steps** and click the **remove** button . You can also drag the desired step back to action list.

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.

## Triggers tab



### Available triggers

List of available triggers that you can attach to the task.

### Task triggers

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.

To add trigger select trigger from available triggers and click add button .

Select trigger in the **task triggers** and click the **configure** button  to modify the trigger parameters.

To remove trigger select the step from **task triggers** and click the **remove** button .



### 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.

## Settings tab

The screenshot shows the 'Edit task' dialog box for a task named '[DEMO] Clear Temporary Folder'. The 'Settings' tab is selected. The dialog includes a toolbar with icons for 'Save and exit', 'Cancel', 'Save', 'Configure', 'Move down', 'Move up', and 'Record'. Below the toolbar, the task name is displayed in a text box, followed by an 'Enabled' checkbox. The main area contains several settings: 'Log level' is set to 'Errors And Warnings' with a dropdown arrow, and a 'Duplicate to Common Log' checkbox is present. Under 'If error occurred', there are radio buttons for 'Stop task execution' (selected) and 'Continue task execution', a checkbox for 'Run another task (error handler)' with an empty text box and a menu icon, and a checkbox for 'Disable task'. Below this is a checkbox for 'Disable task after execution'. The 'External Name' is 'Task210'. There are checkboxes for 'Do not stop the task when the application is shutting down' and 'Logon Task as User (For NT Service Mode Only)' (checked).

### Log Level

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

- **Nothing**
- **Errors Only**
- **Errors and Warnings**

- **All Messages**

**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 another task (error handler)**

Run the other task if an error is raised. The current task itself is an error handler. 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)

**Disable task after error**

Enable this option if you want to disable the task when an error occurs.

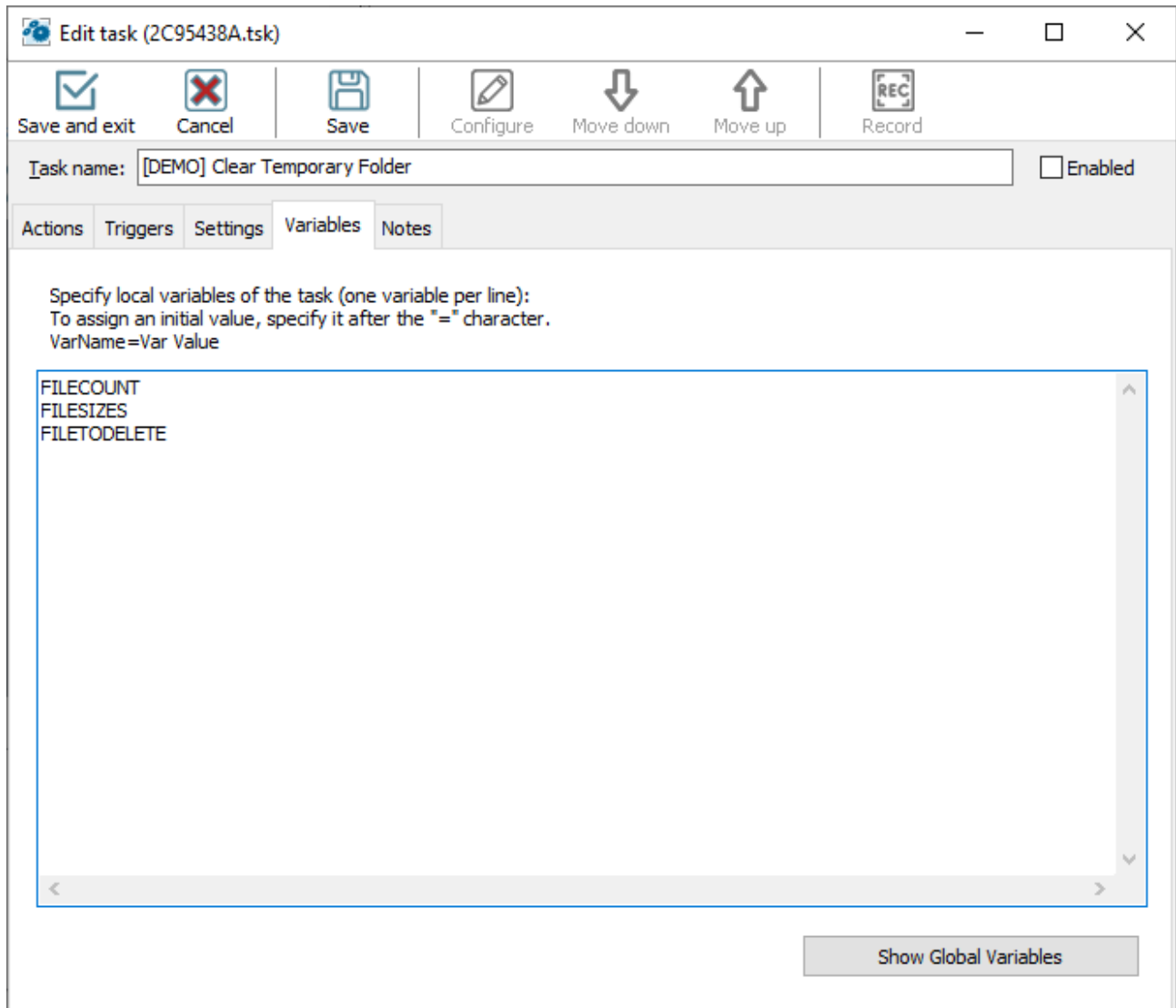
**Disable task after execution**

Enable this option if you want to disable the task after completion anyway in any case.

**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](#)<sup>[562]</sup>). 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.

## 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
```

and so on.

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
```

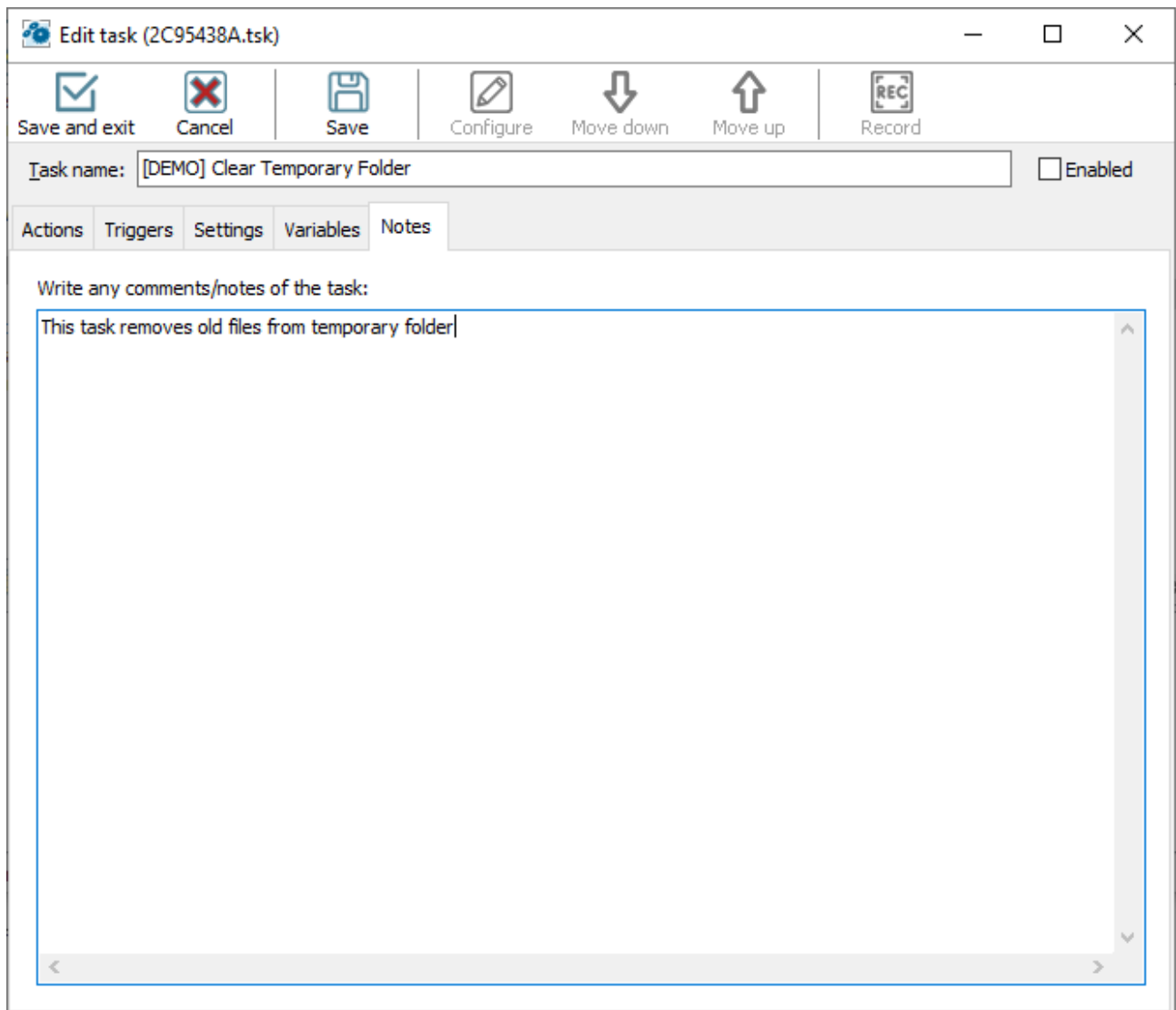
etc.

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>[12]</sup>.

## Notes tab



You can write any comment for this task.

### 3.1.2 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>51</sup>
- [Triggers tab](#) <sup>53</sup>
- [Settings tab](#) <sup>54</sup>
- [Variables tab](#) <sup>56</sup>
- [Notes tab](#) <sup>57</sup>

The task editor is described in the [previous chapter](#) <sup>50</sup>.

### 3.1.3 Task - Delete

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

### 3.1.4 Task - Run

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

### 3.1.5 Task - Stop

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

### 3.1.6 Task - Find the task

This is the service of a quick search of a task by the task name. You can see a dialog with a complete task list. Just begin typing a search string. The dialog will be filtering the task list when typing the string. Only the tasks containing the typed string in their name will be present in the list.

To go to the task, highlight the searched task and press the button OK. You also can jump to the task using the Double-Click on the searched task.

### 3.1.7 Task - Local running task

This is the service of a search of the task running at the moment. You will see only running at the moment tasks in the list of the dialog window.

To go to the task highlight the searched task and press the button OK or use the Double-Click on the searched task.

### 3.1.8 Task - Export

Export of a task to an external file. You can also export several tasks. To export tasks highlight the task (or several tasks) in the list and select this menu item. RoboTask will export the tasks into a text file.

Also you can export task by using [drag-and-drop mechanism](#)<sup>231</sup>

### 3.1.9 Task - Import

Import of a task from a text file to the repository of RoboTask. When you import you will be proposed to select the task file (or several files). The tasks will be imported to the active selected task folder.

Also you can import tasks by using [drag-and-drop mechanism](#)<sup>231</sup>

**Note:** Pay attention that the imported tasks are always disabled after importing. You have to enable it manually.

### 3.1.10 Task - Copy

You can copy a task by means of the method Copy-Paste using the system clipboard. Just highlight one or several tasks in a list and press Ctrl-C (or select this item of the main or context menu).

### 3.1.11 Task - Paste

If in the system clipboard is the copied task (or several tasks), you can paste them in an active task folder pressing Ctrl-V or selecting this item of the main or context menu.

---

The method of copying of tasks by means of the clipboard works with remote tasks too if the network connection with remote RoboTask is open. That is you can copy a local task and paste it in a remote session and vice versa.

**Note:** Pay attention that the pasted tasks are always disabled after the operation. You have to enable it manually.

### 3.1.12 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.

### 3.1.13 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.

### 3.1.14 Task - Create Batch File

Create a batch file to run a separate task from a command line. To create a batch file highlight the task in a list and select this menu item. After that the program will propose to select a folder and a file name.

### 3.1.15 Task - Exit

Closes the RoboTask window and stops the software.

## 3.2 Remote

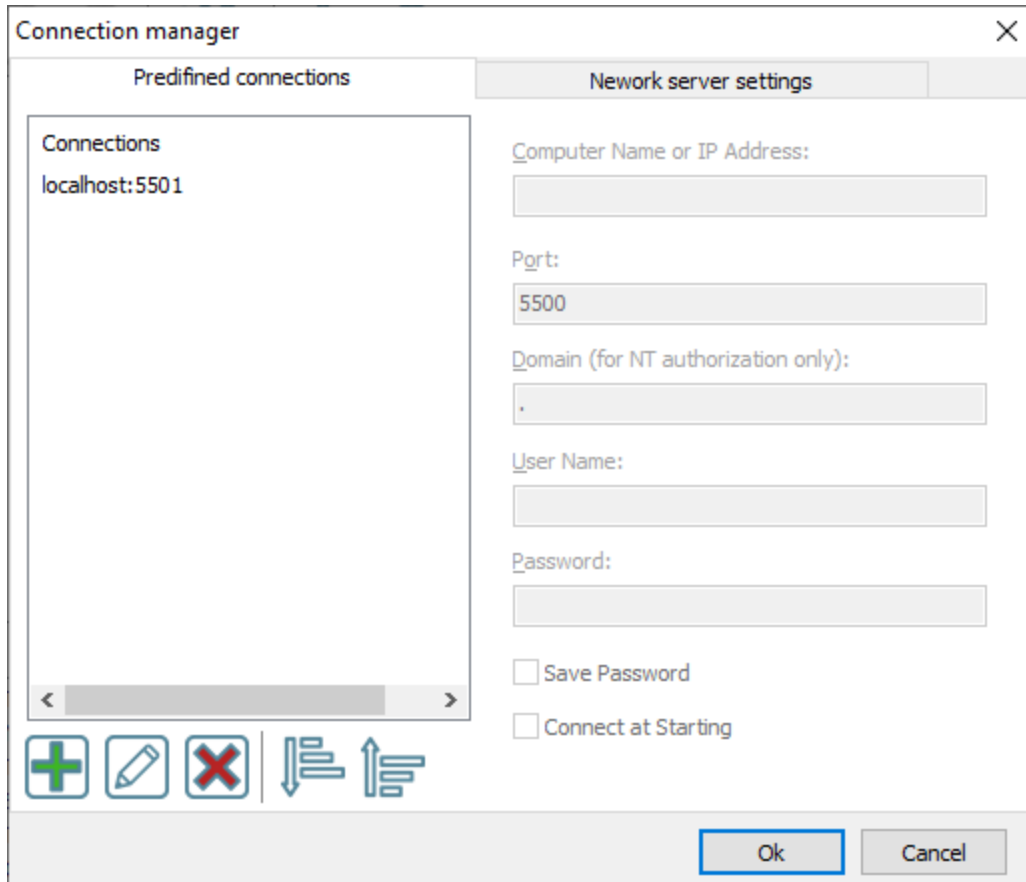
The actions for working with remote RoboTask are collected in this group.

### 3.2.1 Connections manager

Connection manager allows you define connection parameters for network client and parameters of internal network server.

#### Predefined connections

To connect to the necessary server of the RoboTask quickly, you can specify the connect parameters in advance.



Each predefined connection contains such parameters:

- **Computer name or IP address**
- **Port**
- **Domain (for NT authorization only)** - you can specify single dot (.) if you are using non-domain user (local computer user)
- **User name**
- **Password**

**Save Password** - switch on this checkbox if you want to save password

**Connect at startup** - switch on this checkbox if you want to activate this connection when RoboTask starts.



## Network server settings

On this tab you can set up server parameters of RoboTask. Read more about client-server mode [here](#)<sup>42</sup>.

The screenshot shows the 'Connection manager' dialog box with the 'Network server settings' tab selected. The dialog has a title bar with a close button (X). Below the title bar are two tabs: 'Predefined connections' and 'Network server settings'. The 'Network server settings' tab contains the following controls:

- Start Server at %s Loading
- Bind to IP:
- Port:
- Encrypt Data
- Compress data
- User Rights section:
  - Use internal list (with an 'Edit Internal List of Users' button)
  - System administrators
- Server is active section:
  - Start Server button
  - Stop Server button
  - View/Edit connections button
- Note: Note that you must allow incoming connections for RoboTask in your Windows Firewall settings in order to use the RoboTask network server.
- Ok and Cancel buttons at the bottom right.

### Start Server at RoboTask loading

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

### Bind to IP

By default this field is empty. This means that the network server is bound to all available addresses on the computer. The address 0.0.0.0 has the same effect. You can specify particular address. This is useful if the computer/server has several IP addresses.

Also you can specify local host address 127.0.0.1. In this case you can connect to server from local computer only.

### 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.

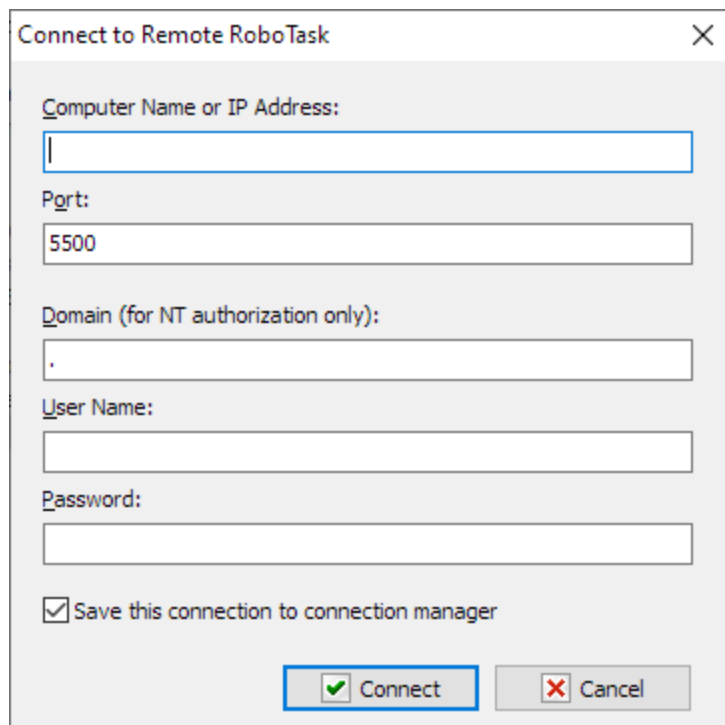
### View/Edit connections

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

Also see [Network Client](#)<sup>[44]</sup> chapter

## 3.2.2 Remote - Connect to

Connecting to remote RoboTask. It is necessary to fill in the fields in the connection parameters window



Connect to Remote RoboTask

Computer Name or IP Address:

Port:

5500

Domain (for NT authorization only):

.

User Name:

Password:

Save this connection to connection manager

### Computer Name or IP Address

The name of a computer in a local network or an IP address explicitly. The IP address should be specified according to the IPv4 standard.

### Port

The port for connection. On default, the port of the RoboTask server is 5500 for the RoboTask application and 5501 for the RoboTask system service.

### Domain

The name of a user domain for authorization. It is used if the network server of RoboTask uses a system authorization. It is necessary to specify the domain name of the server or a dot (.) if a local user of the server is specified.

### User Name Password

The parameters of connecting should correspond to the parameters of the RoboTask network server (see the [Server Settings](#)<sup>[62]</sup>)

## 3.2.3 Remote - Disconnect

Closes an active connection with the remote RoboTask.

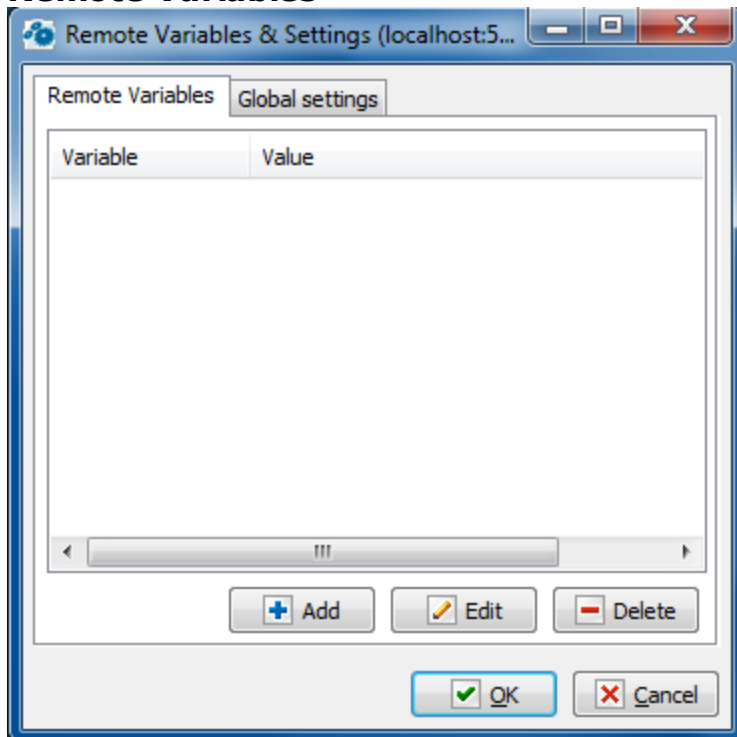
### 3.2.4 Enable Remote Automation

Switches on or off the automation mode for the remote RoboTask similarly to such function for local tasks ([Tools - Enable Automation](#)<sup>[73]</sup>).

### 3.2.5 Remote Variable & Settings

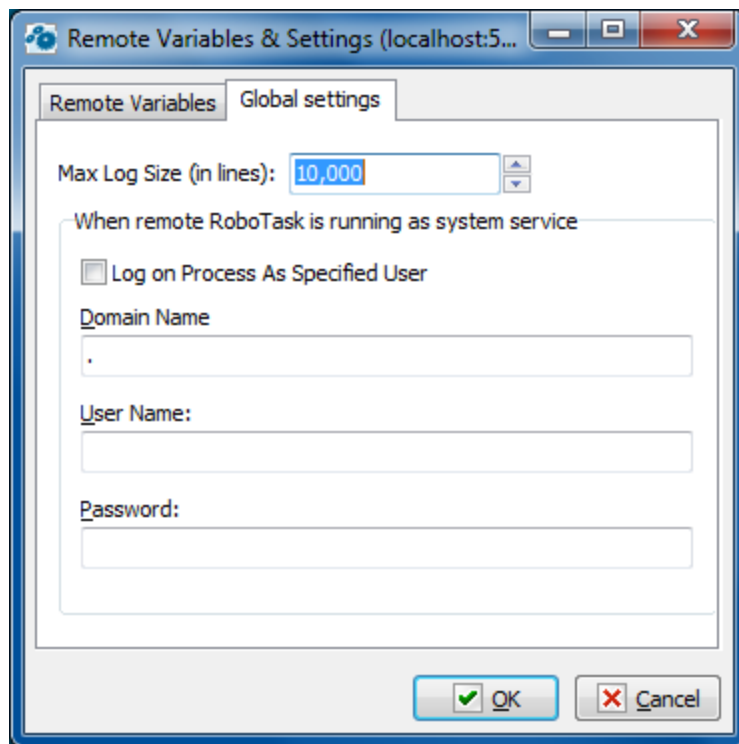
This dialog allows to see and edit the global variables and the global parameters of remote RoboTask.

#### Remote Variables



You can edit the list of the global variables of the remote RoboTask on the tab of global variables.

#### Global settings



### Max Log Size

The max size of log files in strings. If the size has been exceeded, the older records are deleted.

### When RoboTask is running as system service

These parameters are necessary to authorize a process with rights of specified user. These parameters make sense in the system service mode only. If the remote RoboTask is working in the mode of a common application, these parameters are inaccessible.

#### Log on Process as specified user.

Switch on this checkbox if you want to turn on the authorization mode. Otherwise, RoboTask will work with default rights (SYSTEM). This is quite enough in the most cases.

#### Domain Name

A domain name or a dot (.) if the local user of the remote computer is specified.

#### User Name

#### Password

### 3.2.6 Remote Running Tasks

Get the list of tasks which are active at the moment in the remote RoboTask. To go to a task, highlight the task in the list and press the button Select. Double-click on the highlighted task performs the jump to the task too.

### 3.2.7 RoboTask Service manager

Launch [RoboTask Service Manager](#)<sup>[28]</sup> to install and set initial parameters of RoboTask Service. You can [access service through network connection](#)<sup>[37]</sup> and edit or import necessary tasks for RoboTask Service.

Read more about service mode in the chapter [Service vs Application](#)<sup>[23]</sup>.

### 3.2.8 The list of predefined connections

You can see the list of predefined connections at the end of the menu Remote. Connection occurs in a single click. If you didn't specify any password for the connection, you will be proposed to type the password for the connection.

## 3.3 Options

### 3.3.1 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.

### 3.3.2 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.



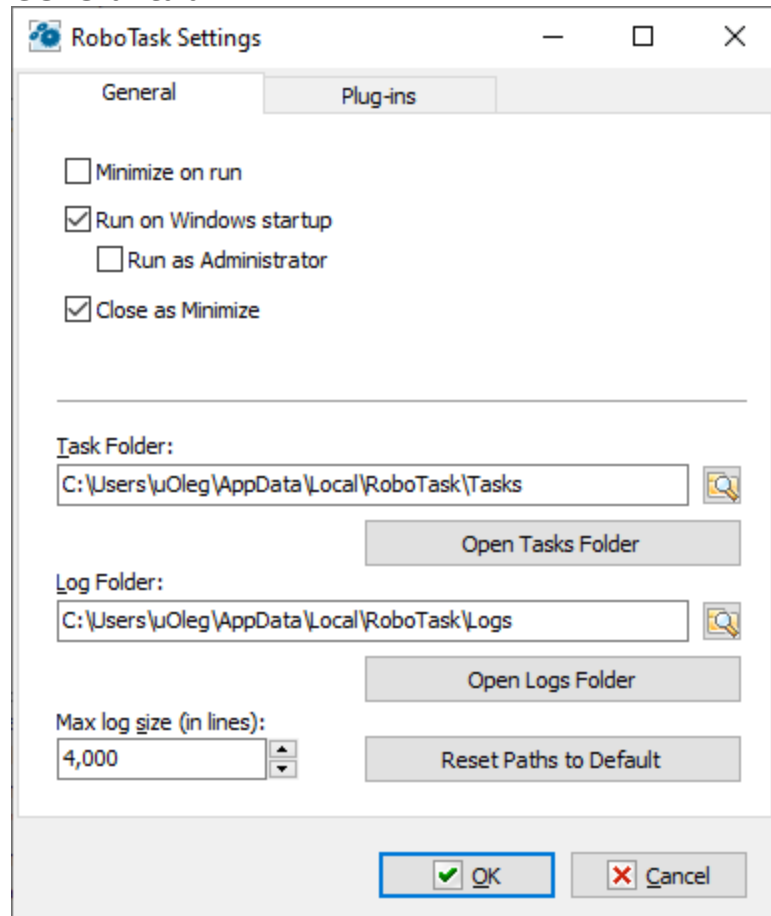
### 3.3.3 Options - Settings

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

- [General tab](#)<sup>[68]</sup>

- [Plug-ins tab](#) 

## 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.

### Run As Administrator

This parameter is available only when the RoboTask is used in application mode in the operating systems using the UAC (User Access Control). That is in Windows Vista and the following versions.

If a user has the administrator's rights on the given computer, the UAC, by default, demotes the rights of started applications in the session. Due to this, an error Access Denied is possible during some operations that demand entire rights. For example, when attempting to start or stop the system services, write down a value in a system register into the branch HKEY\_LOCAL\_MACHINE\SOFTWARE, etc.

Switch on this flag if you want the RoboTask to run with entire rights. In this case you have to confirm running the RoboTask manually when the UAC requests confirmation.

**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**

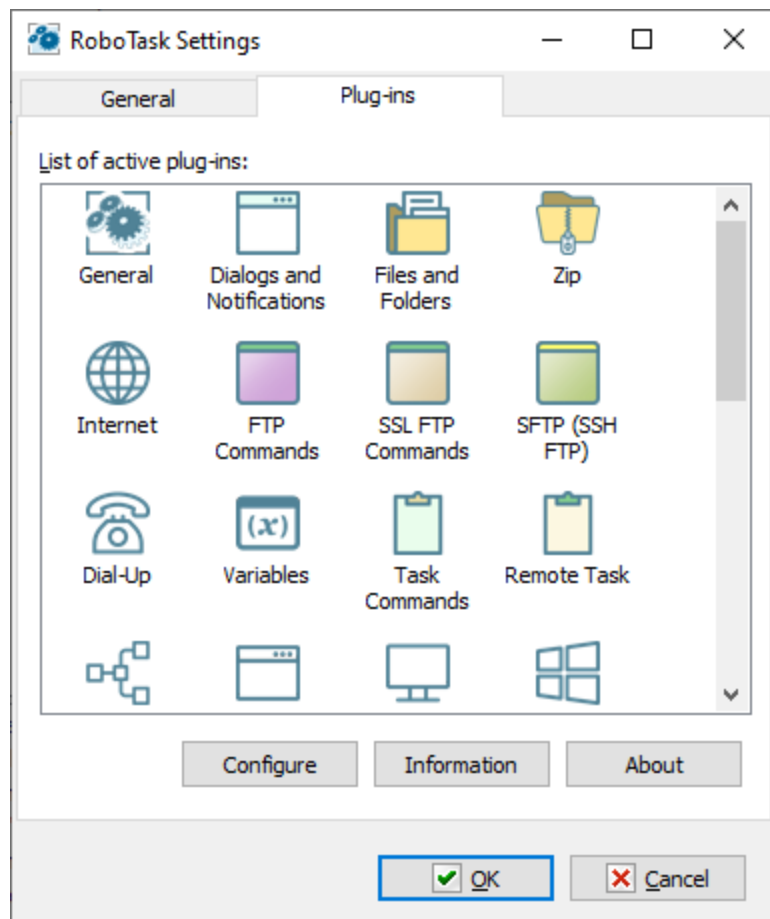
Enter a folder where you want to save log files. You can click the browse button to select a folder.

**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.

### 3.3.4 Options - Variables

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

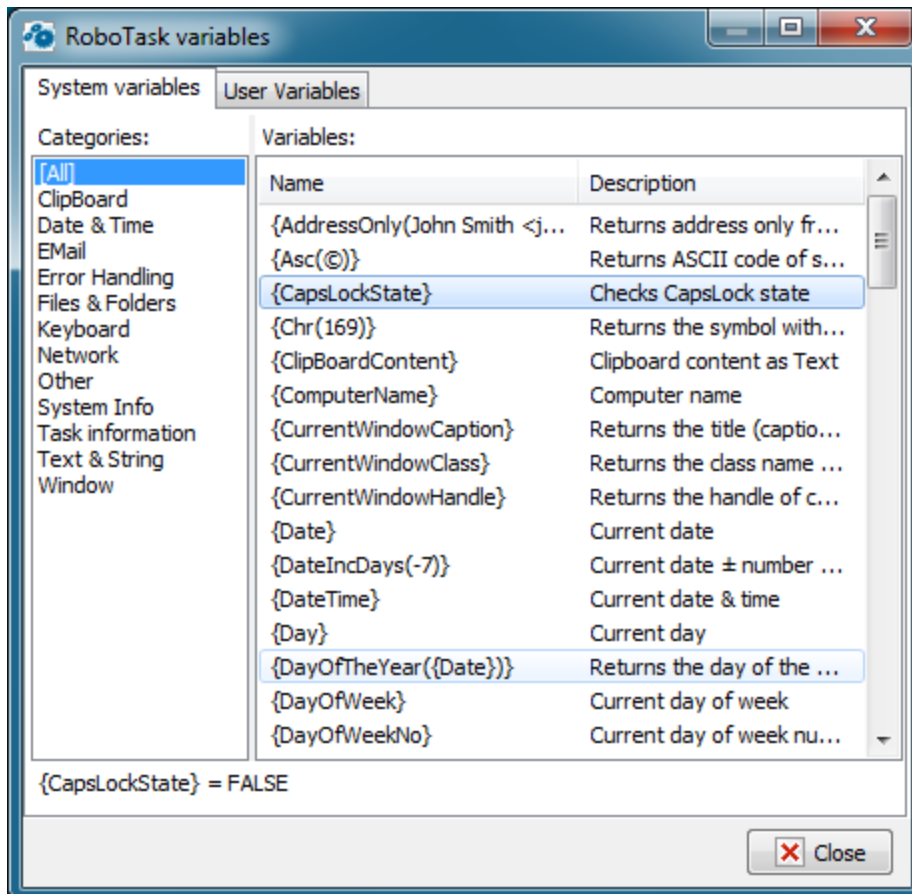
- [System Variables tab](#)<sup>[71]</sup>
- [User's Variables tab](#)<sup>[72]</sup>

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:

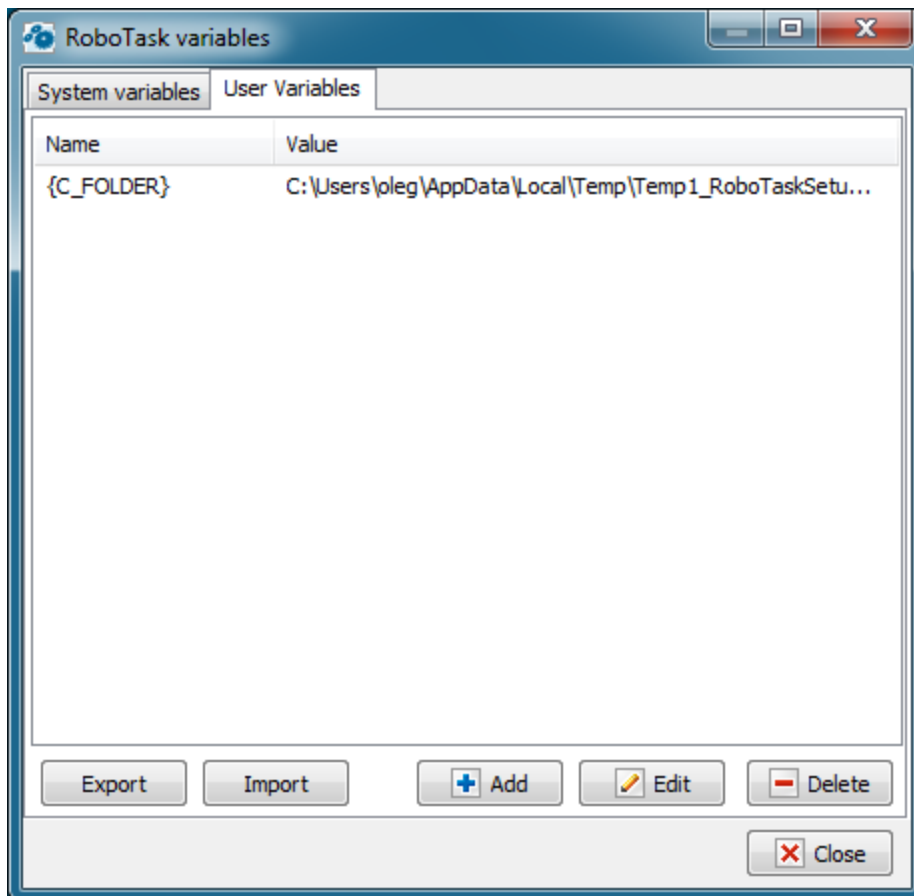
- **Date & Time**
- **Files & Folders**

- **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.

### 3.3.5 Options - Scale

This option permits to change the scale of user interface windows. It can be useful for people with poor eyesight and when using high resolution screens (150-200dpi and higher).

The scale could be changed from 100% (the original size) to 200%.

## 3.4 Tools

### 3.4.1 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.

### 3.4.2 Tools - Lock Task Editor

Locking of functions of task editing. The next functions are blocked in this case:

- Editing of tasks.
- Editing of a tasks tree.
- Editing of a global variables list.
- Import of new tasks.

Thereby, RoboTask goes to the Read-Only mode.

When you switch on the locking, you will be proposed to type the password of unblocking twice. Remember this password without fail.

### 3.4.3 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.

### 3.4.4 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.

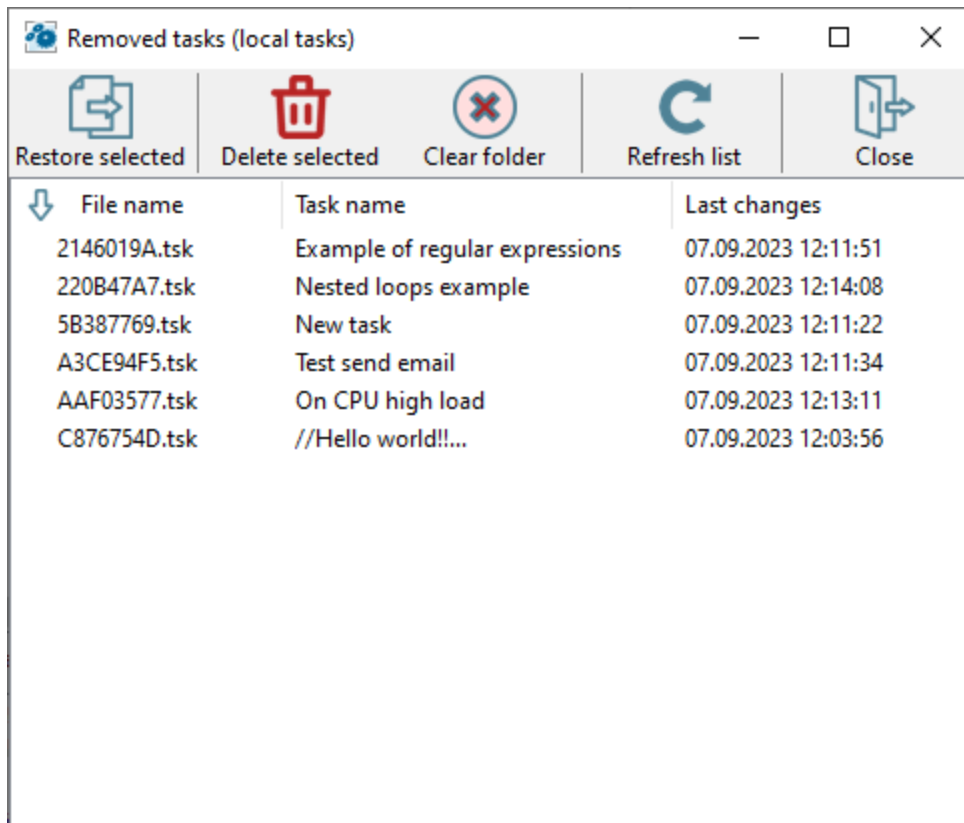
### 3.4.5 Tools - Deleted Tasks

When you delete a task, RoboTask doesn't completely remove it. The task is placed in a "trash bin" similar to the system Recycle Bin. You can view the list of deleted tasks and either

permanently delete them or restore them to the current task folder. This function works for both local tasks and network connections to another instance of RoboTask.

*Please note* that RoboTask does not save empty tasks. It does not make sense

The list consists of three columns: File name, Task name, Last changes. You can sort the list by any column.



Possible actions with the list:

- **Restore selected:** Select one or multiple tasks and click the "Restore selected" button or choose the corresponding option from the popup menu. Restoring selected tasks places them back into the current task folder. Task restoration uses the task import mechanism from an external file.
- **Delete selected:** Permanently delete files from the trash bin. Select one or multiple tasks and click the "Delete" button or choose the corresponding option from the popup menu.
- **Clear folder:** Empty the trash bin, which means deleting all files.
- **Refresh list:** Tasks may be deleted from another computer over the network. This function allows you to refresh the list of deleted tasks.
- **Filter list:** You can filter the list of files based on a set of characters. To invoke the filter, press the CTRL+F combination or select the corresponding option from the popup menu.

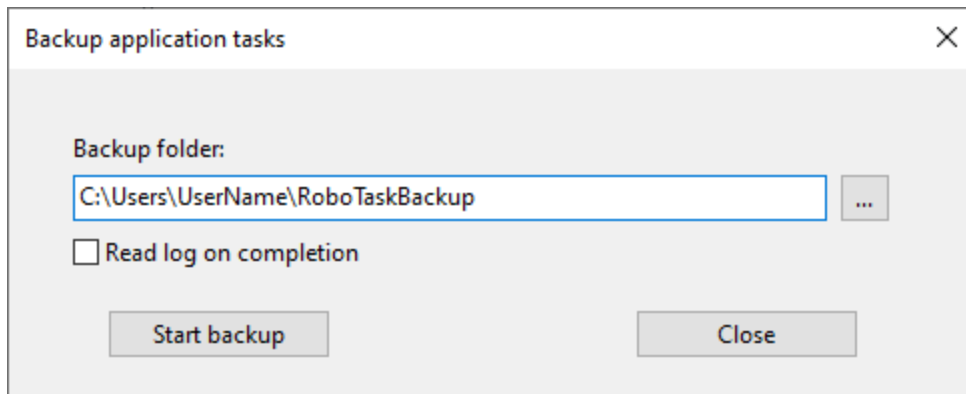
### 3.4.6 Tools - Start selected tasks sequentially

Starting of highlighted tasks sequentially. The tasks will be started according the list. The next task will be started only after finishing the previous one. This mode is sometimes useful for the manual start of tasks.

### 3.4.7 Tools - Backup application tasks

Backing up tasks consists of packing the current set of tasks into a ZIP archive. If necessary, it is possible to restore tasks from a saved archive.

Backup is performed using a predefined task launched in the environment of runtime module.



#### Backup folder

Select a folder to save the backup files. The default folder is in the user profile:

`C:\Users\\RoboTaskBackup`

If necessary, you can select another folder. The selected folder is saved for future work sessions.

The archive file name is generated automatically using the current date and time:

`RoboTask-App-Tasks-YYYYMMDD-hhmm.zip`

where YYYYMMDD – date in "YYYYMMDD" format

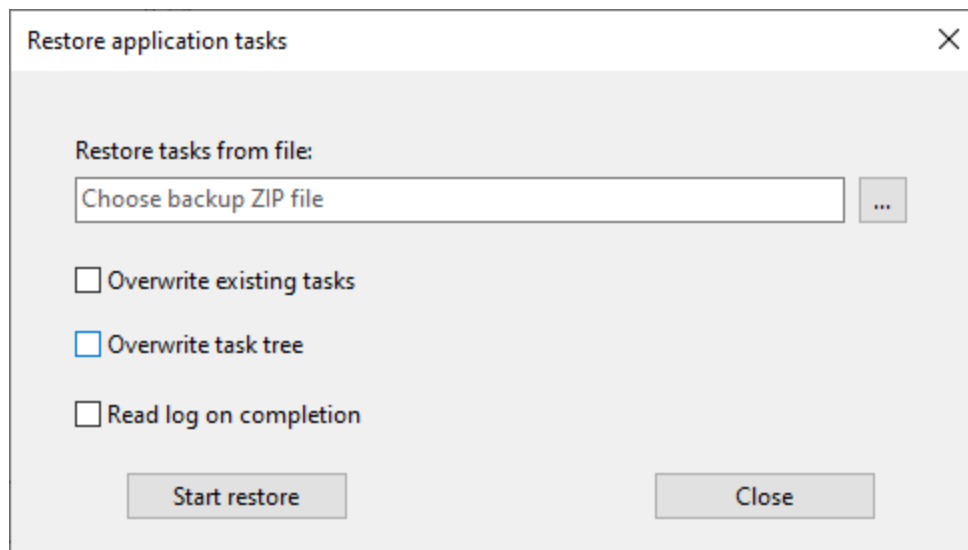
#### Read log on completion

Check this box if you want to read the task execution log.

### 3.4.8 Tools - Restore application tasks

Restoring tasks consists in unpacking tasks from a backup file into the tasks folder.

Restoration occurs with the help of a predefined task launched in the environment of runtime module.

**Restore tasks from file**

Select backup ZIP file

**Overwrite existing tasks**

If you want to overwrite existing tasks, then enable this checkbox.

**Overwrite existing task tree**

If you want to overwrite an existing task tree, then enable this checkbox.

**Read log on completion**

Check this box if you want to read the task execution log.

## 3.5 Help

### 3.5.1 Help - Help Topics

Displays this RoboTask Help.

### 3.5.2 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.

### **3.5.3 Help - Technical support**

This menu item redirect you to the page of a support site of the product.

There you can get links to additional information about the product, ask a question or go to our forum.

### **3.5.4 Help - Check for updates**

Get the information about updates of RoboTask for your version. On default, checking for updates occurs one time a week. You can specify another periodicity or switch on the manual check mode.

### **3.5.5 Help - About**

Displays information about your version of RoboTask.



# Part

---



IV

## Actions

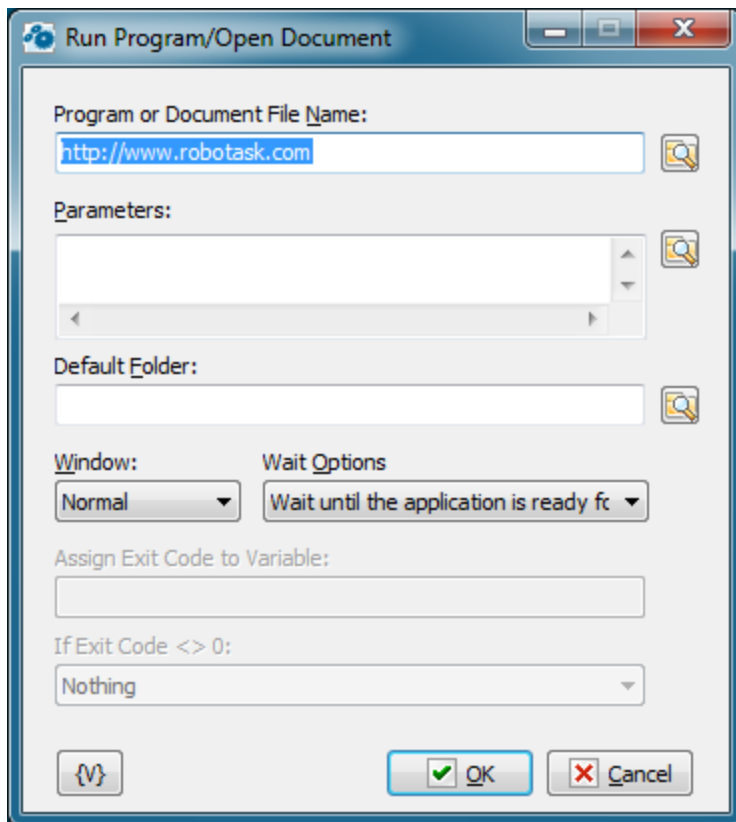
## 4 Actions

### 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>[84]</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>[41]</sup>)

### Variable

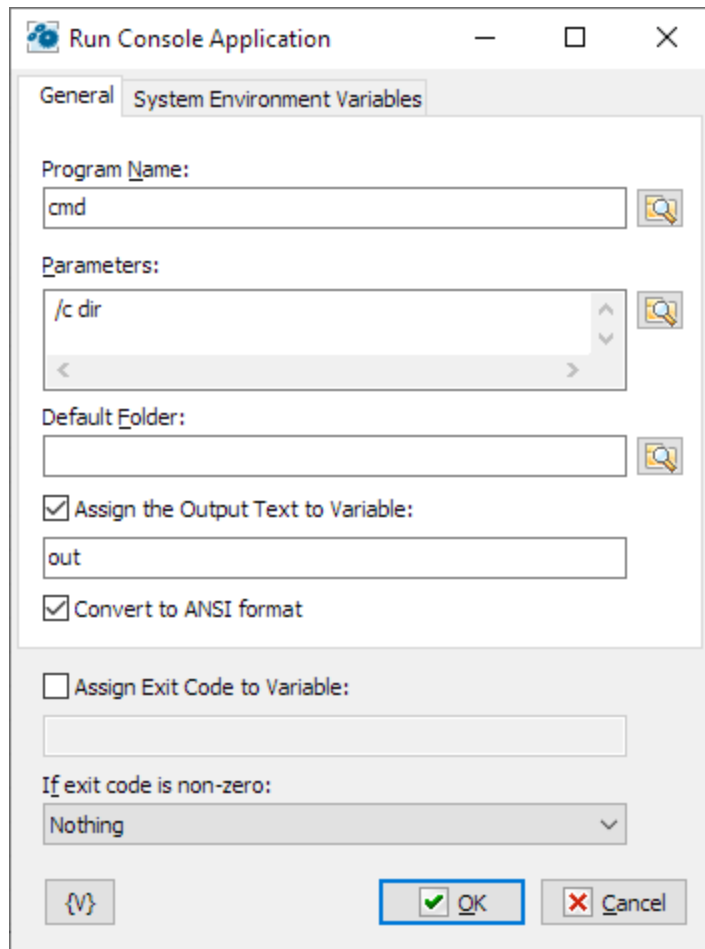
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.1.2 Run Console Application

This action is intended for console command utilities. Sometimes it is important not only perform such utility with certain parameters but analyze output data, that is, the text displayed on the screen by the utility.

The action allows to save an output stream of the utility into a variable. Then in the task you can save this information into a file or analyze the output text.

### General



### Program Name

Specify the file name of an/the utility. Unlike the action Run Program/Open Document Parameters, it is prohibited specifying of a document file in this field. Only the name of the executed document must be specified here.

You can specify the parameters of a command line here. For convenience, it is possible to type each parameter on a new line. When executing the action, the RoboTask forms the common parameter line from multi-line text.

### Default Folder

Specify the folder name, which will be current when the utility starts.

### Assign the Output Text into Variable

If you need to save the output text, switch on this checkbox and specify a variable name. Then you can use this variable for the next processing.

### Convert to ANSI format

Some utilities output a text with non-latin symbols in OEM format (or in the format DOS). In this case switch on this checkbox for automatically transformation of the string in ANSI coding.

### Assign Exit Code into Variable

If you need to save the exit code of the utility into a variable, switch on this checkbox and input the variable name. The exit code is a number. As a rule, it is 0 when it works regularly or an error code when some trouble appears. It depend on the started utility.

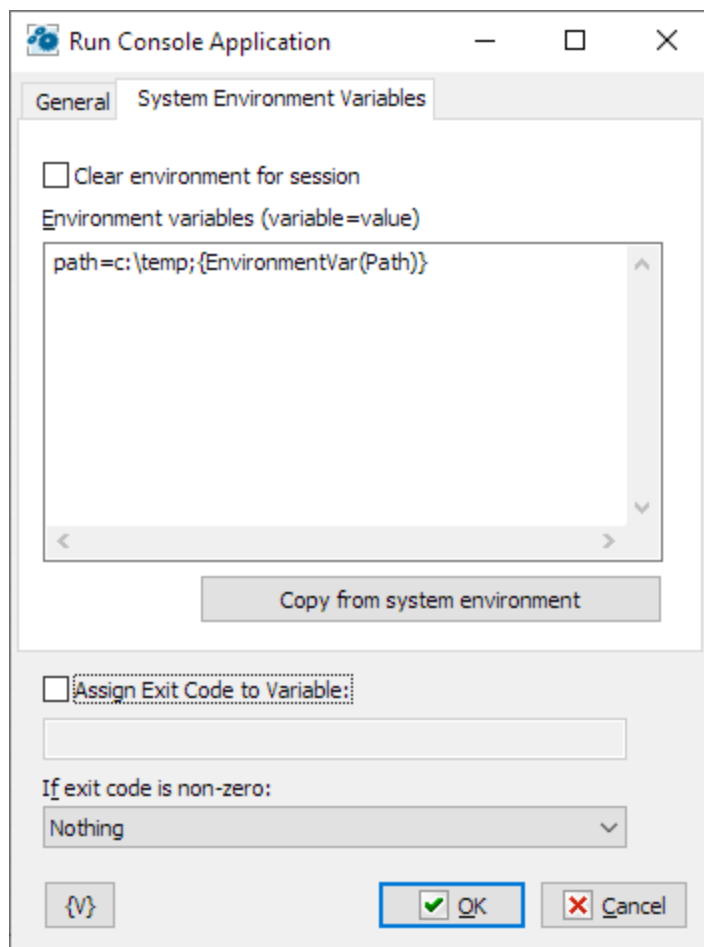
### If Exit Code $\neq$ 0

The action when the exit cod is not 0. Select one of the 3 alternatives:

- Nothing
- Stop the task
- Raise the Error.

On default it is Nothing

## System Environment Variables



### Clear environment for session

Check this box if you want to replace the environment completely. The action completely replaces the environment variables with the values of the Environment Variables parameter.

### Environment variables

Specify the environment variables that you want to replace or add. The variables are shown as text, and each line looks like the following:

```
variable_name=value
```

### Copy from system environment

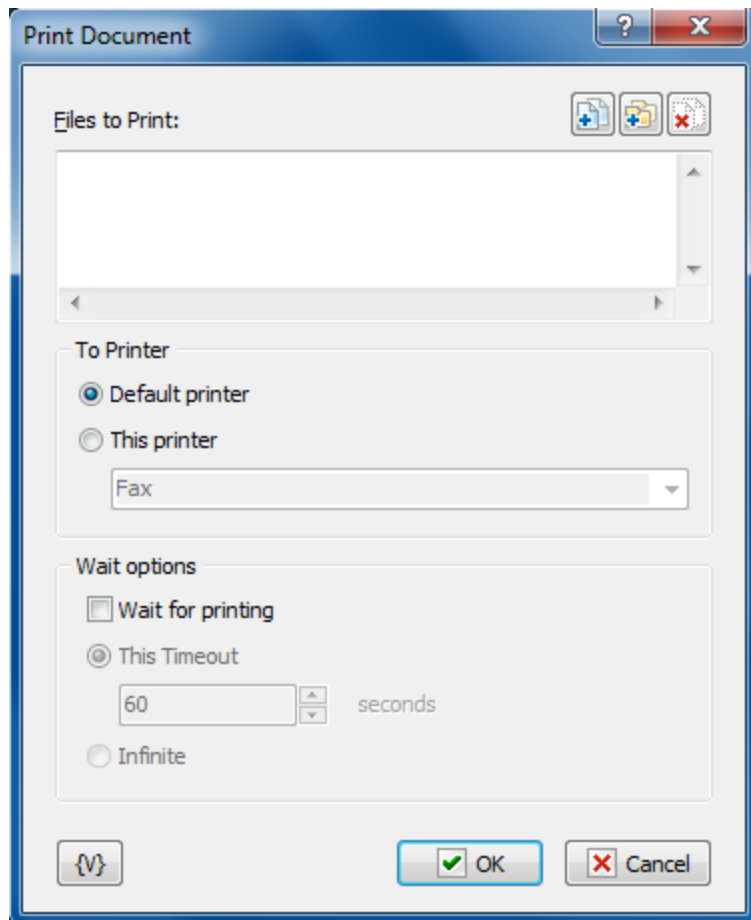
This button allows you to fill the list with the current values of the environment variables.

## 4.1.3 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.

To print a document, the Print Document action uses a default application for the specified kind of documents. For example, for a Word document (DOC file) it is usually the MS Word from the MS Office package, for a PDF file it is the Acrobat Reader, etc.

The action can direct a document to any printer registered in the system. Of course, the default application is supposed to support printing on the specified printer. As a rule, all the basic applications support this function.



## 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.

## To printer

- **Default printer** – direct a document to a default printer;
- **This printer** – direct a document to the specified printer. When you select a printer you can pick out the one from a dropdown list or write the name of the printer in the entry field. When writing the name of a printer, variables can be used. For example, {PrinterName}.

## Wait options

This parameter defines the timeout in seconds to wait until the document is being printed. You can also specify an infinite timeout.

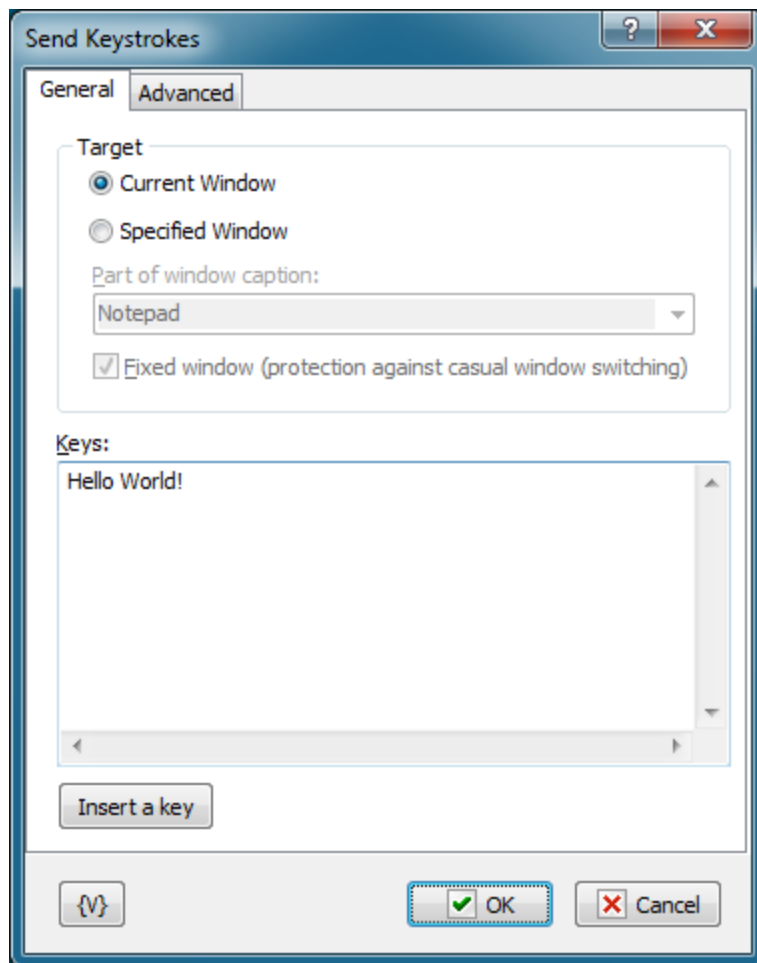
If the timeout is over, the action stops the printing application.

## Variable

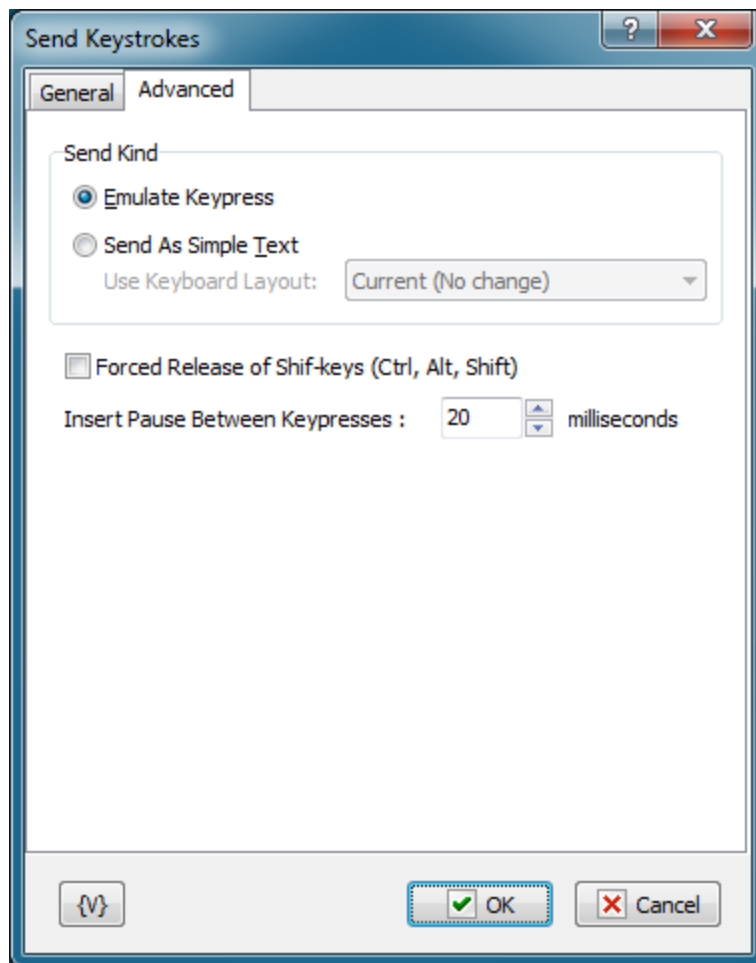
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.1.4 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>[88]</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

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

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

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

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

Also it's possible to use these symbols within braces

<b>^</b>	<b>{^}</b>
<b>%</b>	<b>{%}</b>
<b>+</b>	<b>{+}</b>
<b>#</b>	<b>{#}</b>
<b>{</b>	<b>{{}}</b>
<b>}</b>	<b>{}}</b>
<b>(</b>	<b>{(}</b>
<b>)</b>	<b>{)}</b>

For example to print the string: **^ % + # { } ( )**

use the combination: **{^} {%} {+} {#} {{} {}} {(} {)}**

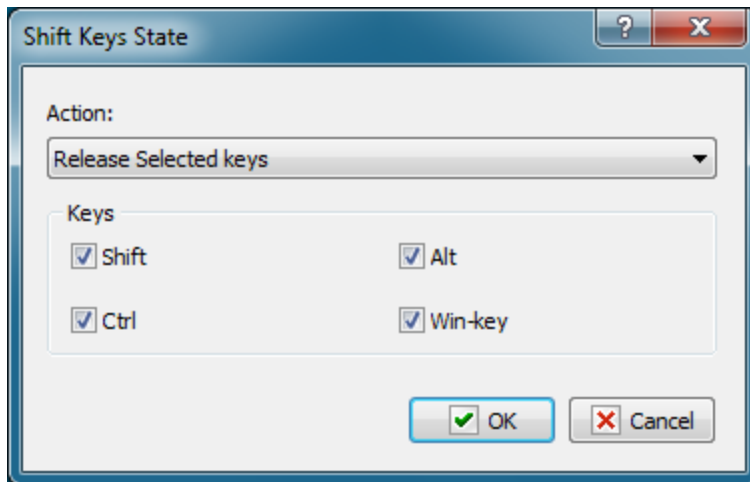
### 4.1.5 Shift Keys State

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>84</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>[84]</sup>" when Send Keystrokes are in use.

**Keys**

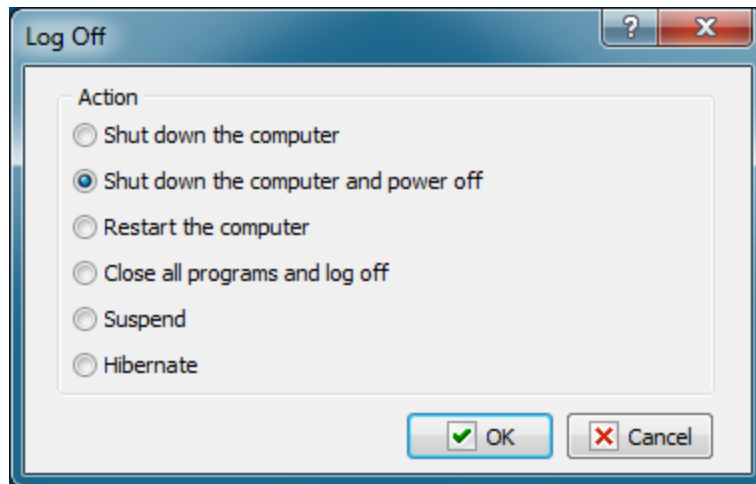
Mark necessary keys in any combination.

#### 4.1.6 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 defragger program finishes cleaning up your hard drive.



### Action

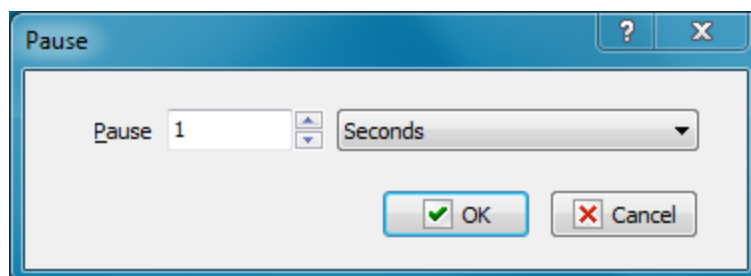
Select one of the following values:

- **Shut down the computer**
- **Shut down the computer and power off**
- **Restart the computer**
- **Log Off**
- **Suspend** - sleep mode
- **Hibernate** - hibernate computer if it's possible.

## 4.1.7 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 pool.



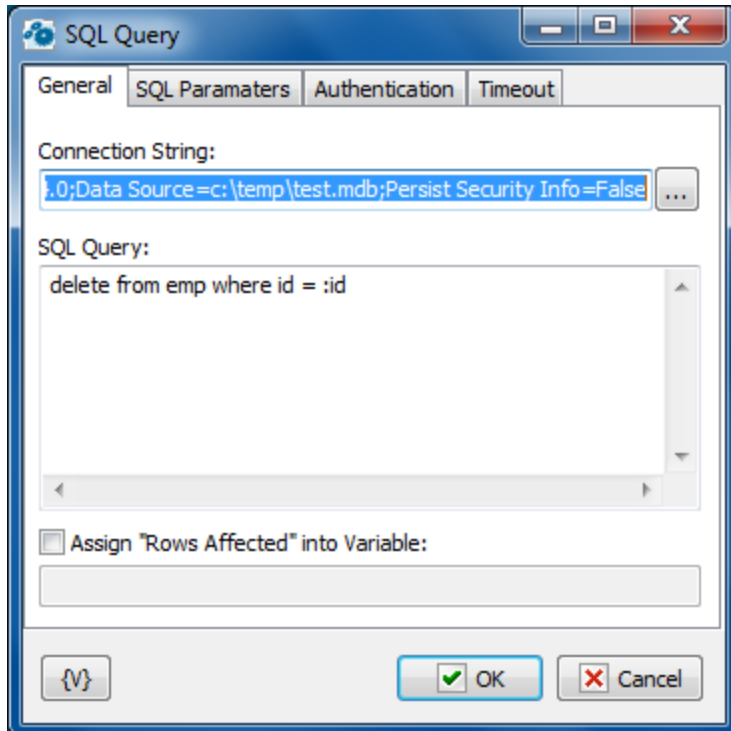
### Pause

Enter the pause. You can specify the dimension as:

- Milliseconds
- Seconds
- Minutes

## 4.1.8 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](#) 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.

### SQL parameters

Sometimes using of parameters of a query is more convenient instead of generation of the query by means of variables. For example, when the variable value contains quotes or another special symbols. If you use variables (or macro substitution) when forming SQL inquiry, availability such symbols can "break" the syntax of the query and you will get a runtime error.

Parameters SQL are designated in the inquiry by means of colon (:) before the parameter name.

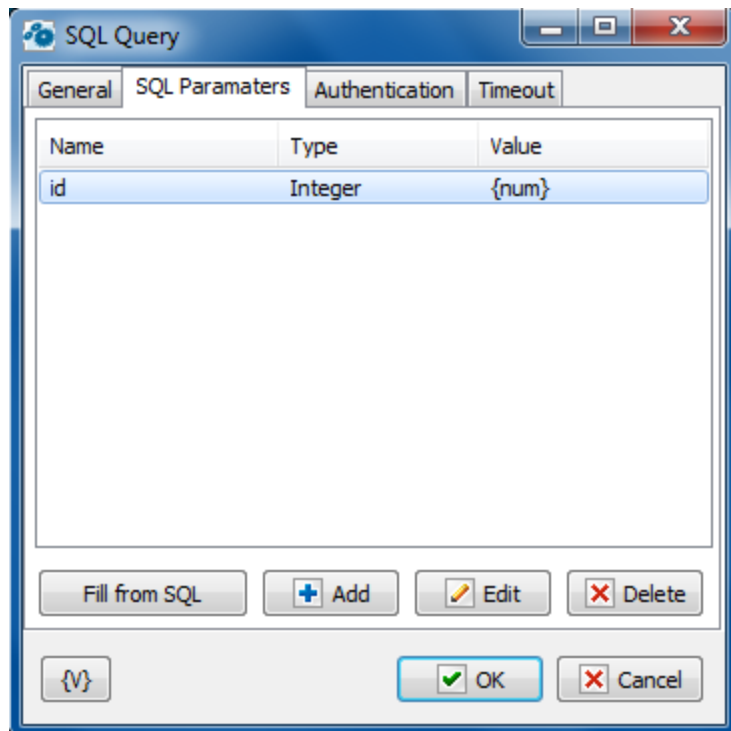
For example:

```
Select * from employee where hiredate > :dt and salary >= :amount
```

*:dt* and *:amount* are parameters of the query.

Parameters can be several types:

- **String** – any string
- **Integer** – an integer number
- **Float** – a real number or the number with a floating dot.
- **Boolean** – a logical value True or False.
- **DateTime** – a date and time. The value of such parameter must be defined in current system format of date and time (see the system regional settings). Notice that using of date-time can have its own characteristics for different databases. For example, it is better specify such parameters for the MySQL as a string parameter in the format YYYY-MM-DD or YYYY-MM-DD hh:nn:ss.



### Fill from SQL

Fill the list of parameters from the current query. If a parameter is already in the list, this parameter will be without change. The function fills only the missing parameters in the list.

### Add

Add a parameter manually. You will be proposed to type or select from the list the parameter name, select the parameter type and input an expression as the parameter value.

Using of variables in a parameter name and in its value is allowed. The RoboTask will substitute variable values instead of variables before performing the SQL.

### Edit

Edit the properties of the selected parameter.

### Delete

Delete the selected parameter from the list.

If you specify the parameter which is absent in the query, the action ignores this parameter when performing and writes the proper warning in the log 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.

## 4.1.9 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.

Wait For Process Finish

Process:  
notepad.exe

If more than one matching process

Wait for Oldest  Wait for All  Wait for Newest

Waiting

Timeout 10 sec  
Raise TIMEOUT error (default option)  
res

Infinity

Assign Exit Code of Process to Variable

{V} OK Cancel



## 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, you can specify an action:

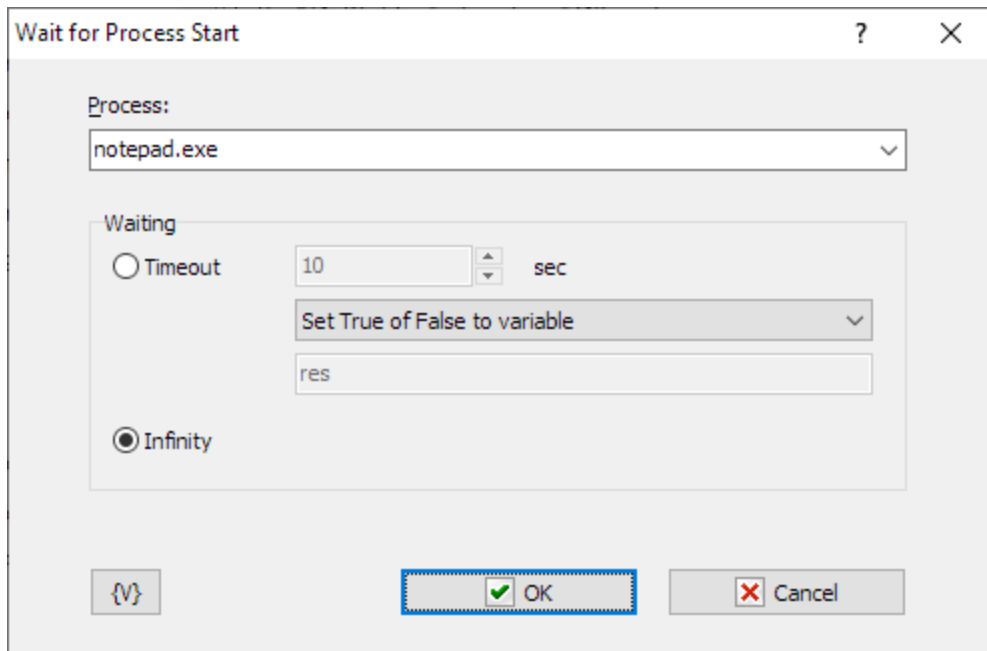
- **Raise Timeout error** - this is default option
- **Set True or False to variable** - in this case the task will continue work without error.

### 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.

## 4.1.10 Wait for Process Start

Pauses the task and waits until the selected process is started. Unlike action "[Wait for Process Finish](#)", 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 you can specify an action:

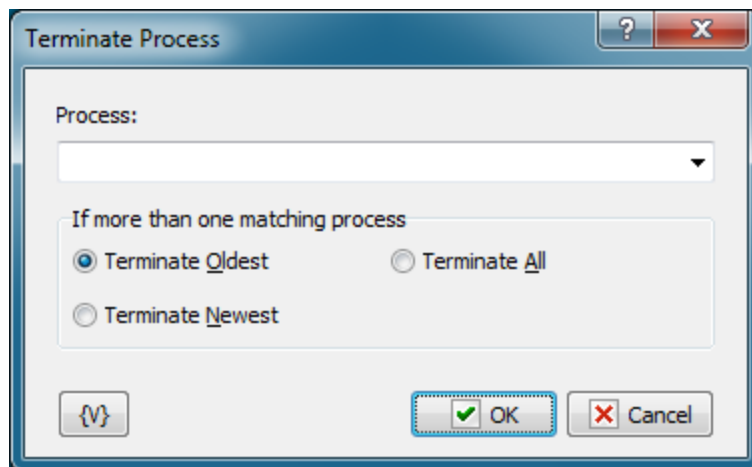
- **Raise Timeout error** - this is default option
- **Set True or False to variable** - in this case the task will continue work without error.

## 4.1.11 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:

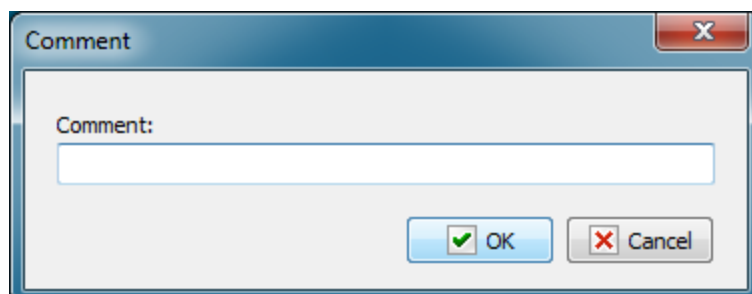
**Terminate Oldest** – terminate the process started first;

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

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

## 4.1.12 Comment

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



## 4.2 Dialogs and Notifications

## 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](#) action.

The screenshot shows a dialog box titled "Show Message/Play Sound". It has a title bar with a question mark and a close button. The dialog contains the following elements:

- Show Message
- Message Text:  
[{Time}, {DayOfWeek}, {Date}]  
Hey, awake!
- Message Icon:  
Information
- Play Sound
- Sound File:  
[Empty text field]
- Buttons: {V}, Test, OK, Cancel

### Show Message

Click this check box to display the message box.

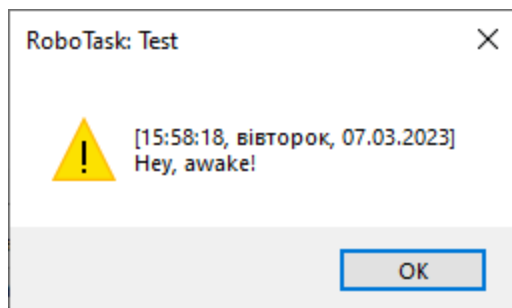
### Message Text

Enter the text for the message box.

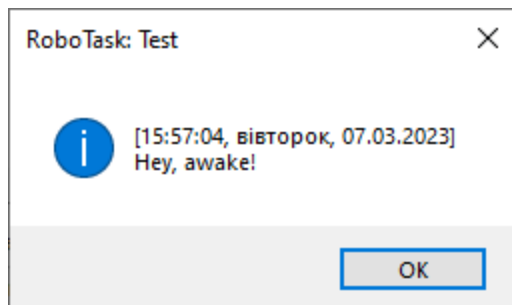
### Message Icon

Select the style of the icon for the message box:

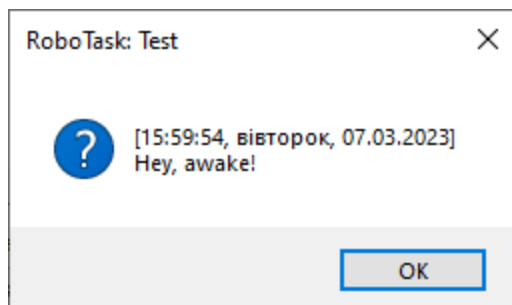
- **Exclamation**



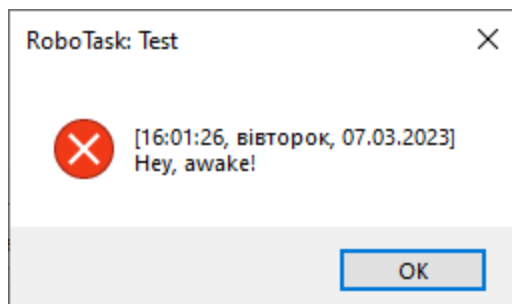
- **Information**



- **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.

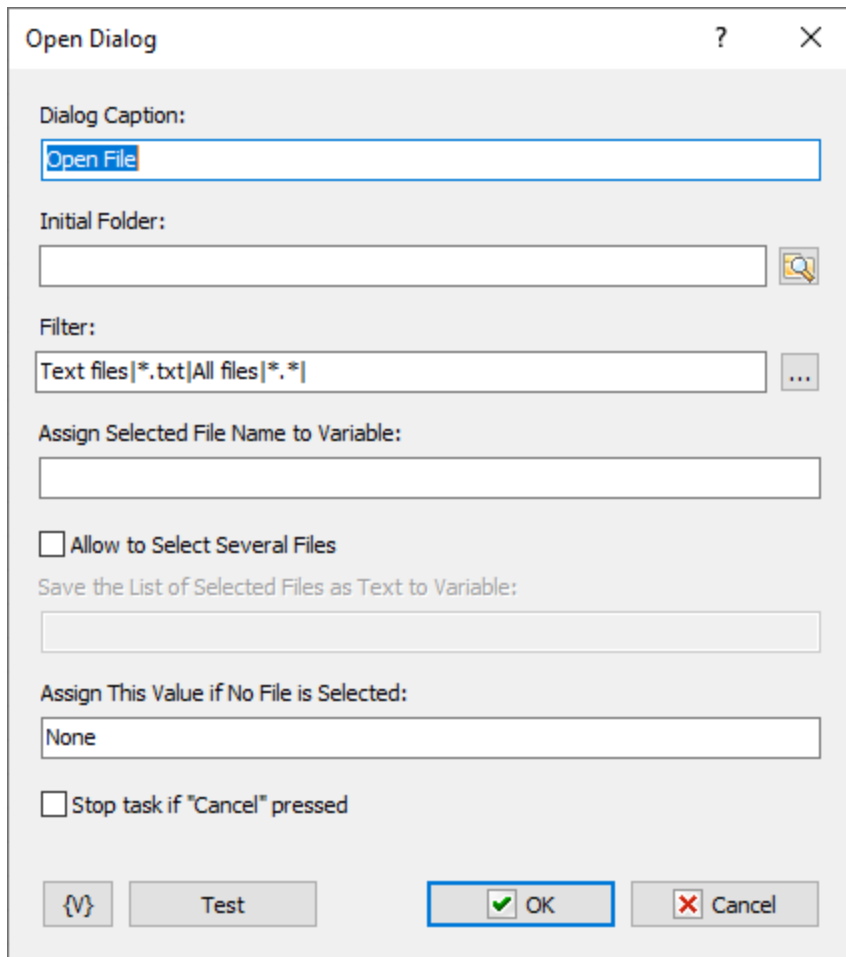
**Test**

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.

### Allow to select several files

Switch on checkbox if you want to select several files.

### Save the list of selected files as Text to Variable



Specify the name of variable where the action saves the result. If you select several files then the result will be a text: one file name in each line.

### Assign This Value if No File Selected

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

### Allow to select several files

Switch on checkbox if you want to select several files.

### Save the list of selected files as Text to Variable

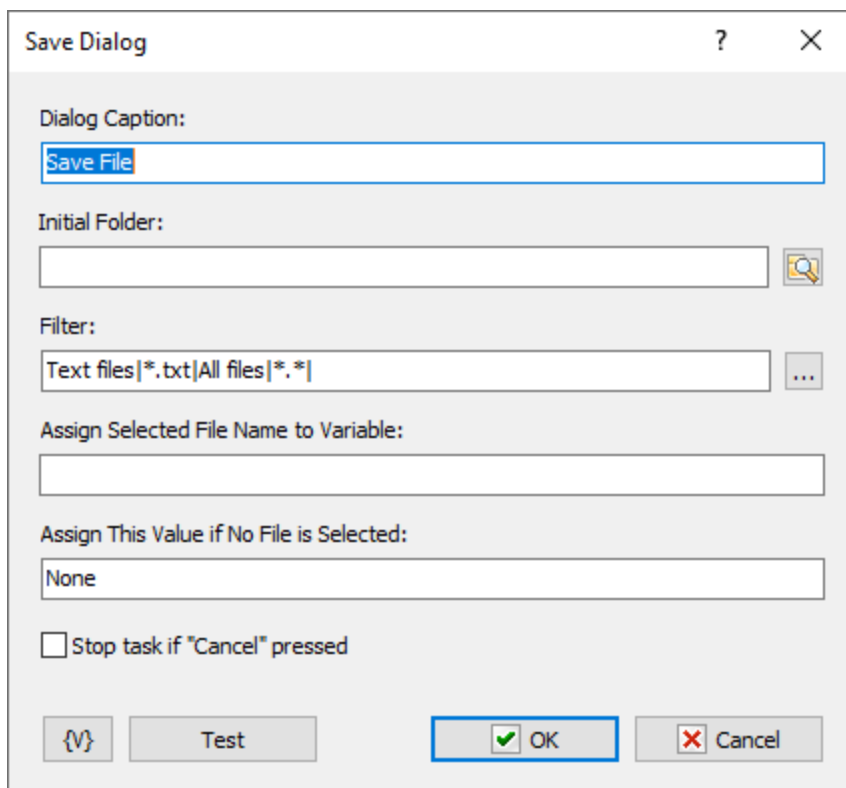
Specify the name of variable where the action saves the result. If you select several files then the result will be a text: one file name in each line.

### 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.

## 4.2.4 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.



The screenshot shows a 'Save Dialog' settings window with the following fields and controls:

- Dialog Caption:** A text box containing 'Save File'.
- Initial Folder:** An empty text box with a search icon to its right.
- Filter:** A text box containing 'Text files|\*.txt|All files|\*.\*' and a dropdown arrow to its right.
- Assign Selected File Name to Variable:** An empty text box.
- Assign This Value if No File is Selected:** A text box containing 'None'.
- Stop task if "Cancel" pressed
- Buttons at the bottom: {V}, Test, OK (highlighted with a blue border), and Cancel.

**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**

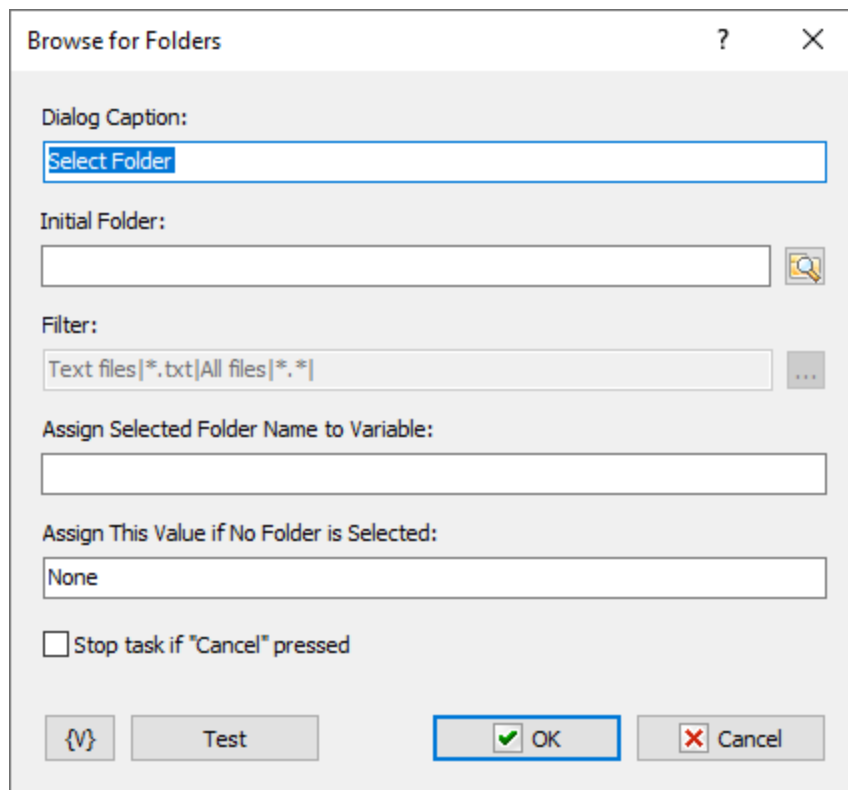
Assign a value to store if no file is selected, i.e. the **Cancel** button was clicked.

**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

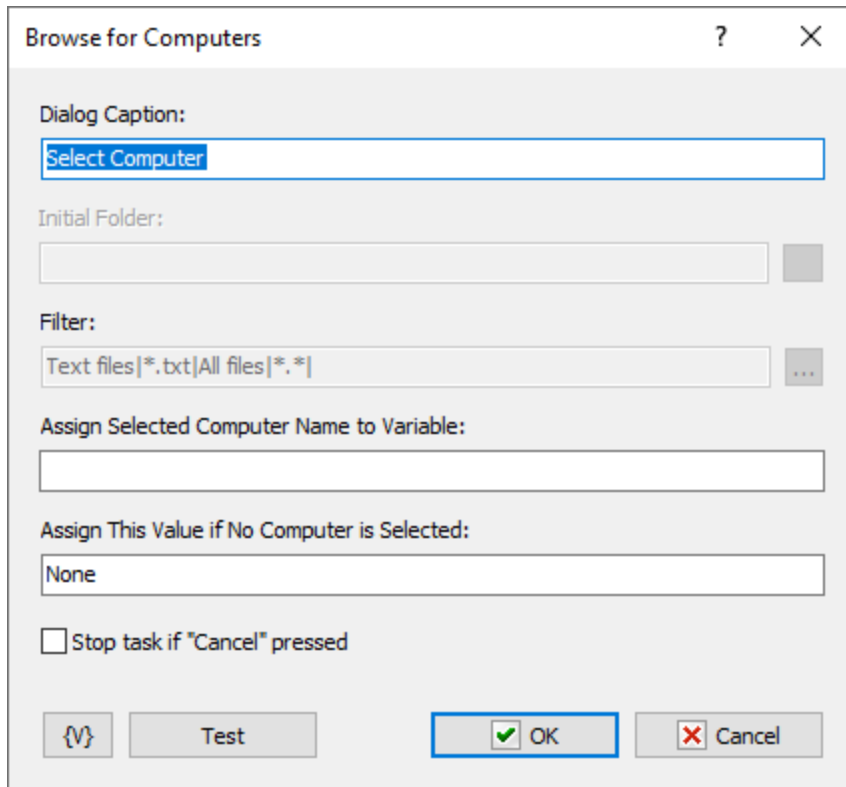
Assign a value to store if no folder is selected, i.e. the **Cancel** button was clicked.

### 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

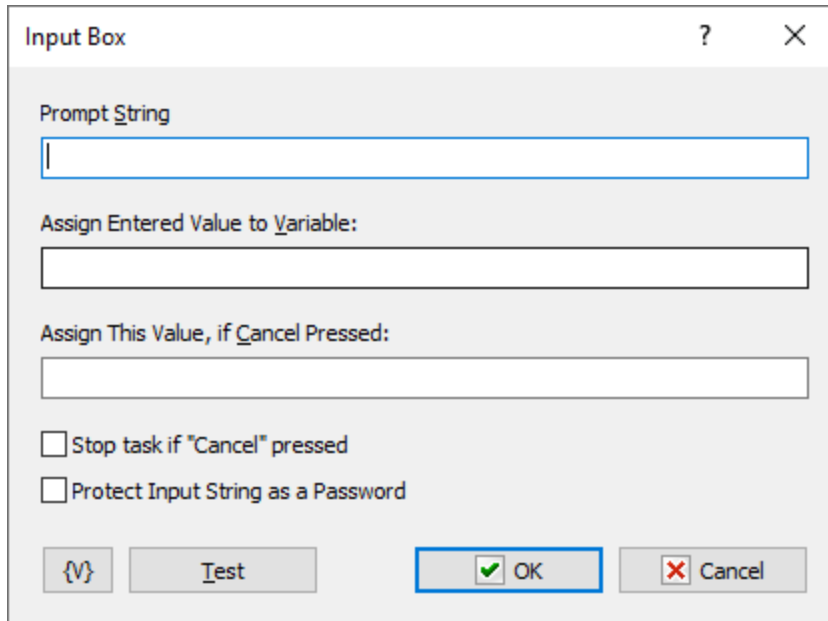
Assign a default value to store if no computer is selected.

### 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.

## 4.2.8 Show Notification

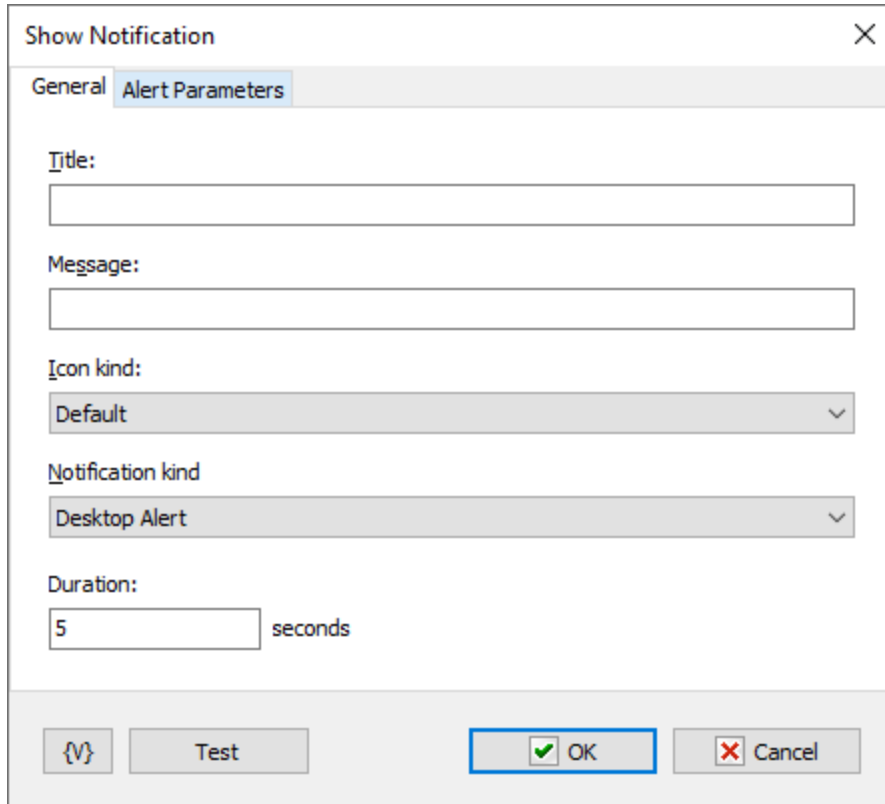
This action is for displaying different popup messages on a screen. These messages do not demand any interactive action (e.g. pressing the key OK). The message closes automatically in several seconds after its displayed. The message can be displayed in two ways:

- As a tray icon balloon

- As popup windows in the corner of the screen

You can press the button Test in the editor of action parameters and then watch how your message looks with the specified parameters.

## General



The screenshot shows a dialog box titled "Show Notification" with a close button (X) in the top right corner. The dialog has two tabs: "General" and "Alert Parameters", with "Alert Parameters" currently selected. The "Alert Parameters" tab contains the following fields:

- Title:** A text input field.
- Message:** A text input field.
- Icon kind:** A dropdown menu with "Default" selected.
- Notification kind:** A dropdown menu with "Desktop Alert" selected.
- Duration:** A text input field containing "5" followed by the label "seconds".

At the bottom of the dialog, there are four buttons: a button with curly braces "{V}", a "Test" button, an "OK" button with a green checkmark icon, and a "Cancel" button with a red X icon. The "OK" button is highlighted with a blue border.

### Title

The title of a message. The usage of variables is allowed.

### Message

The body of a message. The usage of variables is allowed.

### Icon kind

Select a message icon. Available variants are:

- Information
- Exclamation
- Error
- Default

### Alert Kind

Select the kind of a message:

- **Desktop Alert** – a message as a popup window in a screen corner. If the RoboTask sends several messages in succession, all of them will be displayed one above another.
- **Tray Icon Balloon** – the display of the message as a standard system balloon. If the RoboTask sends several messages running, then only the last message will be displayed on the screen. Every next message will overlay the previous one.

### Duration

The duration of a message display. At the end of this time, the message disappears from the screen automatically. It is possible to change the duration of the message from 2 to 20 seconds. By default, it is 5 seconds.

### Alert Parameters

Additional parameters, available for the message display mode “Desktop Alert” only.

The screenshot shows the 'Show Notification' dialog box with the 'Alert Parameters' tab selected. The 'Size' section contains two checked checkboxes: 'Default Width' and 'Default Height'. The 'Width' field is set to 330 and the 'Height' field is set to 76. The 'Position:' dropdown is set to 'Bottom-Right corner of screen'. The 'Color scheme:' dropdown is set to 'Default colors'. At the bottom, there are buttons for '{V}', 'Test', 'OK' (highlighted with a blue border), and 'Cancel'.

### Width

The width of the message window. On default, it is 330 pixels. This is enough for the majority of short messages. When specifying the width of the window the usage of variables is allowed.

### Height

The height of the message window. On default, it is 76 pixels. This is enough for the majority of short messages. When specifying the height of the window the usage of variables is allowed.

### Position

Select one of the four screen corners where the message will appear. On default, it is the right lower screen corner, i.e. where the system tray is.

### Variable

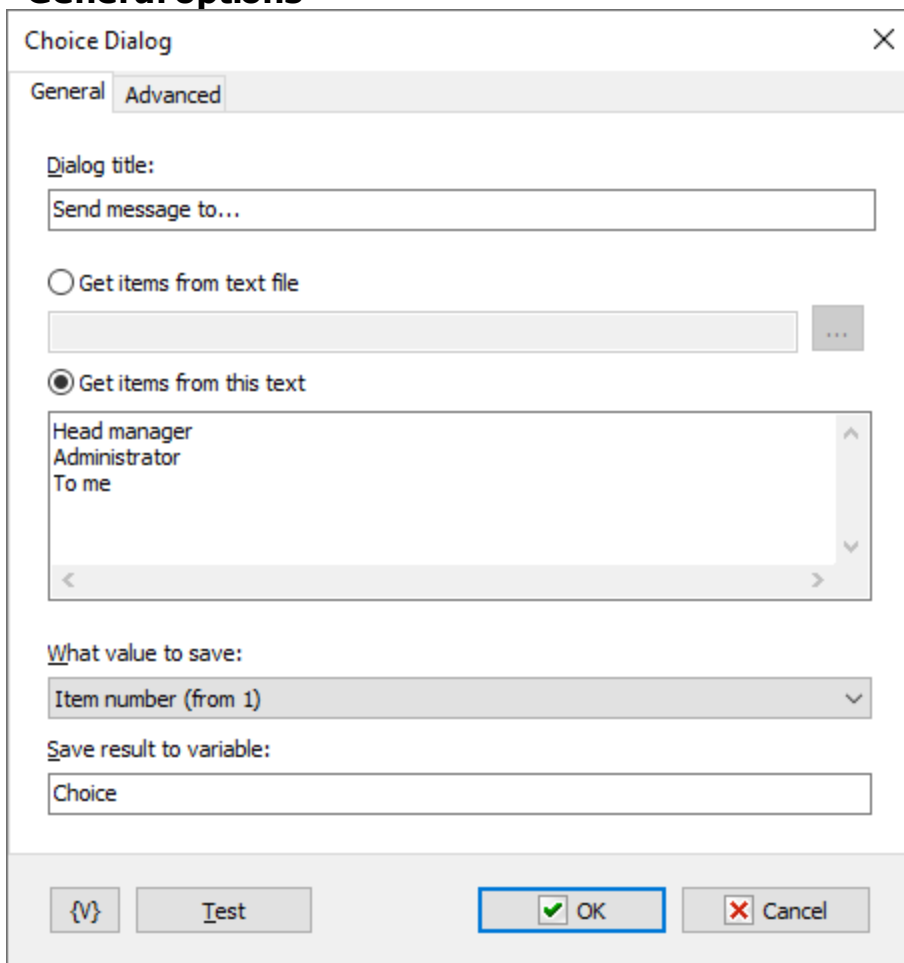
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.2.9 Choice Dialog

The Choose Dialog is necessary to select one value of several alternatives. The action saves the number of chosen item (beginning from 1) into a variable. The action can save the text of item too. Then it is possible to do actions according to a choice.

If a user rejects a choice (that is, the button *Cancel* is pressed), the value of the choice is 0.

### General options



The screenshot shows a dialog box titled "Choice Dialog" with a close button (X) in the top right corner. The dialog has two tabs: "General" (selected) and "Advanced".

**General tab options:**

- Dialog title:** A text box containing "Send message to..."
- Get items from text file:** An unselected radio button next to a text box and a browse button (...).
- Get items from this text:** A selected radio button next to a text area containing "Head manager", "Administrator", and "To me".
- What value to save:** A dropdown menu showing "Item number (from 1)".
- Save result to variable:** A text box containing "Choice".

**Buttons:** At the bottom, there are four buttons: "{V}" (variable), "Test", "OK" (highlighted with a blue border), and "Cancel".

### Dialog Title



---

Specify a title for a choice menu.

**Get Items From Text File**

You can fill in the edit box with file name or choose file. You can use variables in the file name

**Get Items From This Text**

Fill in the text in the input field. Variables are allowed.

**What value to save**

- Item number (from 1)
- The text of item. If Cancel pressed then 0

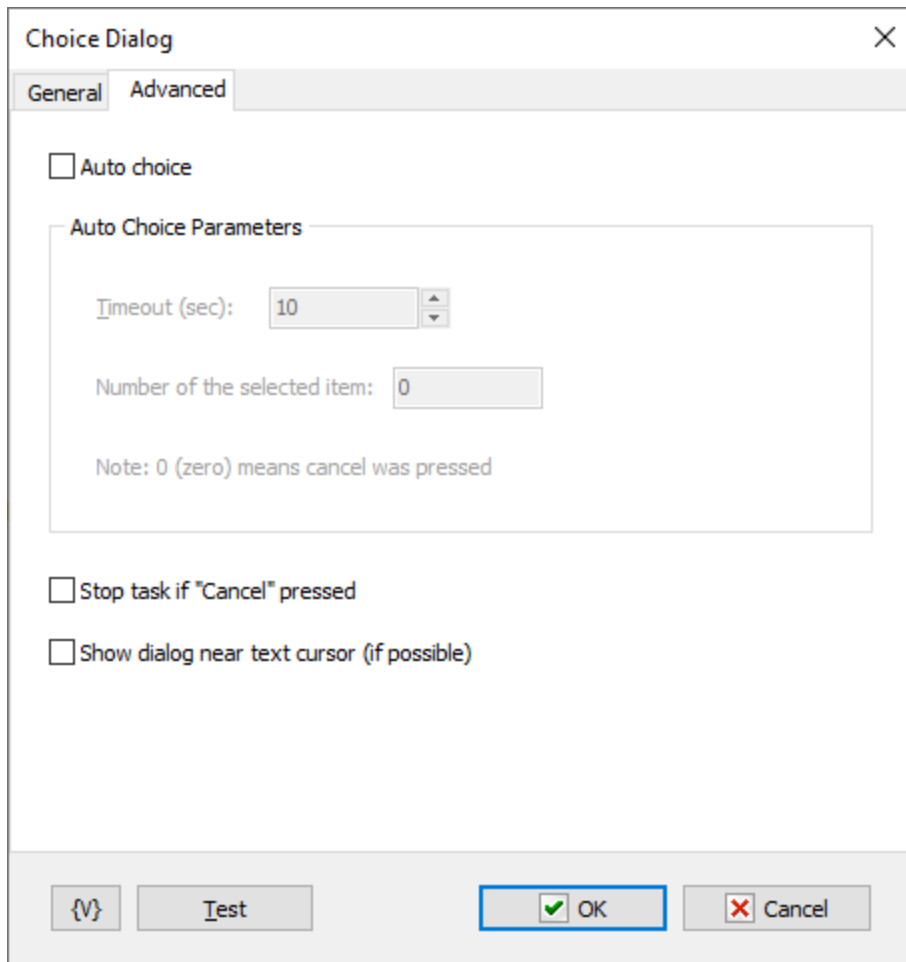
If the user refused the choice (Cancel pressed), then 0 is stored in the variable in any case.

**Save Result to Variable**

Specify the name of a variable into which is necessary to save the choice result. If one of items is chosen, the number of the item will be saved beginning from 1.

If the button Cancel is pressed, then 0 will be saved.

**Advanced options**



### Auto choice

If the user does not perform any action within the specified timeout, then the action will perform the selection automatically and store the specified value in a variable.

### Stop the task if "Cancel" pressed

Check this switch if you want to stop task immediately if "Cancel" pressed.

### Show dialog near text cursor

If you have a text editor or input field open, it can be useful to display a selection dialog near the active text cursor. If possible, the action will show the dialog near the cursor position.

## 4.2.10 Multichoice Dialog

The Multi Choice Dialog allows to switch on or off each option of the proposed set.

For example, a task collects certain information and allows to pass the information to several workers:

- Send the Message to Mike;
- Send the Message to administrator;

- Copy Message to Boss.

The Multi Choice Dialog allows to select alternatives in any combination. The RoboTask saves the values of a choice into separate variables for each option (True or False)

Multi-Choice Dialog

Dialog title:  
Send file to ...

Caption	Variable
Manager	opt1
Administrator	opt2
Copy to me	opt3

+ Add    Edit    Delete    ↑    ↓

Save True (Ok) or False (Cancel) into variable:  
res

Stop task if "Cancel" pressed

Show dialog near text cursor (if possible)

{V}    Test    OK    Cancel

### Dialog Title

Specify the title for a choice menu.

### Items

The list of menu items for a choice. Each item is related to a variable into which the value of a choice of the item is saved. True or False.

### Save True (OK) or False (Cancel) into Variable:

Specify the variable into which it is necessary to save the work result of dialog. If a choice is made, the value True is saved. If not (that is, the button Cancel is pressed), the value False is saved.

### Stop the task if "Cancel" pressed

Check this checkbox if you want to stop task immediately if "Cancel" pressed.

### Show dialog near text cursor

If you have a text editor or input field open, it can be useful to display a selection dialog near the active text cursor. If possible, the action will show the dialog near the cursor position.

## 4.2.11 Show Text

This action can show long text. if you need to show a large text then the action [Show Message/Play Sound](#)<sup>[97]</sup> is not suitable for this. Use the action *Show Text*

The screenshot shows a dialog box titled "Show Text Dialog". It has a close button in the top right corner. The dialog contains the following elements:

- A "Title:" label followed by an empty text input field.
- Two radio buttons: "Text from file" (unselected) and "Use this text" (selected).
- Under "Text from file": a text input field and a file selection button (...).
- Under "Use this text": a large text area with scrollbars.
- At the bottom: a checkbox for "Auto-close dialog after:" followed by a spinner box set to "10" and the word "seconds".
- Four buttons at the bottom: a variable placeholder "{V}", a "Test" button, an "OK" button with a green checkmark, and a "Cancel" button with a red X.

### Title

Input dialog title. The action put this into the caption of dialog window

### Text from file

Choose this option and enter the filename. Also you can choose file by open file dialog. You can compose a file name using variables.

### Use this text

Choose this option if you want specify text in the action parameters. Input the text in the text editor below. Of course you can use variables to compose the text.

## Auto Close Dialog

You can specify the timeout for automatic closing of the window or choose *Manually* option.

### 4.2.12 Custom Input Dialog

This action allows you to create an input dialog for multiple values, not just one string (as in the Input box action, for example). The action allows you to create a simple dialog for entering multiple values at once.

The result of the dialog execution is a JSON structure. For example:

```
{
  "attachment": "C:\\Incoming\\Report20230308.csv",
  "subject": "New incoming file",
  "body": "Hello\r\n\r\nLook at the attached file\r\n\r\n---\r\n best regards\r\n\r\n UserName"
}
```

Each element of the structure looks like *"Element\_name": Element\_value*.

You can read the value of a separate element using the JSON prefix. For example:

```
{json:DialogResult.attachment}
{json:DialogResult.subject}
```

Note that the JSON structure is case-sensitive. The element name must be specified exactly as it appears in the structure, including case.

More information about prefixes can be found in the chapter "Prefixes used with variables".

Custom input dialog supports fields of different types:

**Text edit** - a regular editing field for entering a string

**Multiline text edit** - a simple text editor

**Checkbox** - a switch that has 2 values: ON and OFF

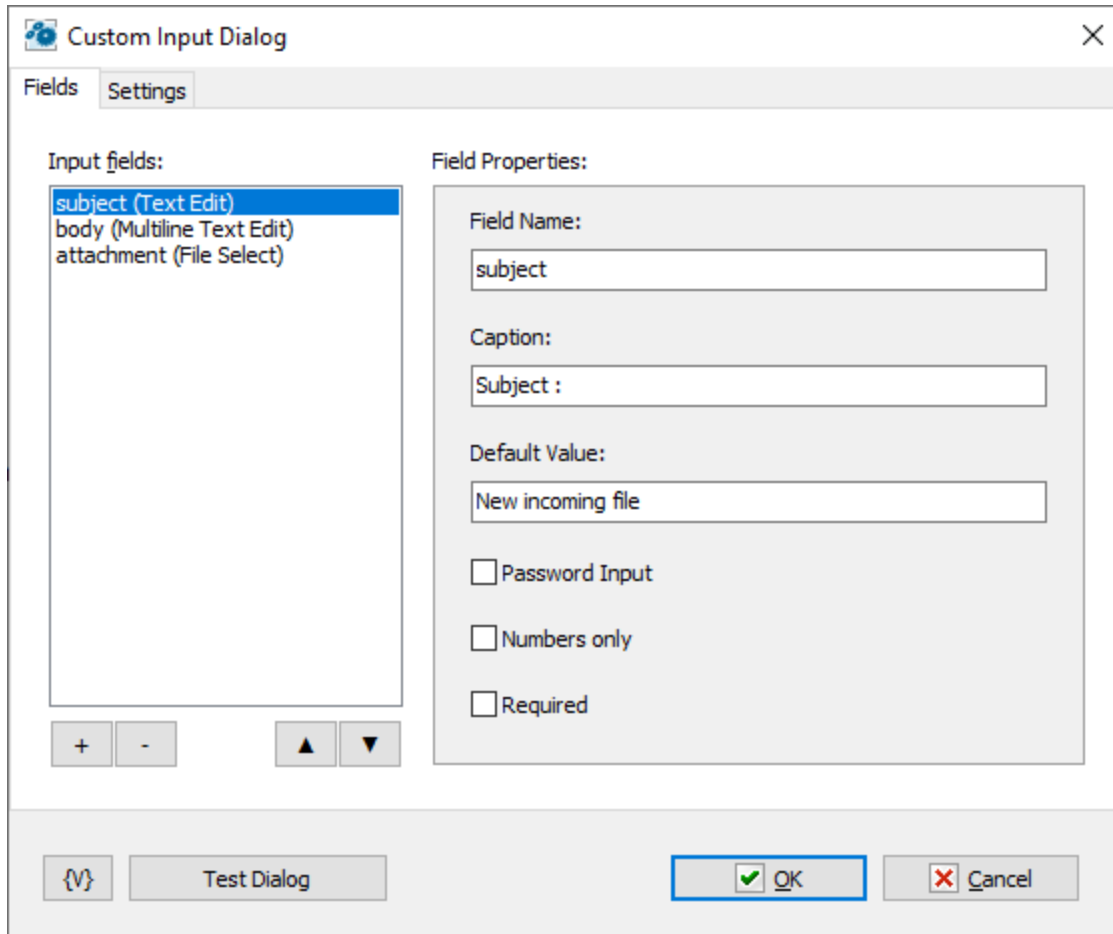
**Combobox** - an editing field with a drop-down list of predefined values

**File select** - a field for entering a file name or selecting a file using a selection dialog

**Folder select** - a field for entering a folder name or selecting a folder using a selection dialog

Each field type has a mandatory parameter - *Field name*. This parameter cannot be empty and must be unique within the dialog.

If the user cancels the selection (by clicking the *Cancel* button), the action can either interrupt task execution or save the specified value to a variable.



### Input Fields

A list of input fields. The input fields in the dialog will be in the order specified in the list. The sequence can be changed using the move item buttons.

To add a field to the dialog, click the "+" button and select the field type. To delete a selected item, select the item in the list and click the "-" button.

### Field Properties

Each input field type has its own set of parameters. All field types have common parameters:

- *Field Name* - a required parameter that serves as the identifier for the field in the dialog;
- *Caption* - a comment for the input field.

### Test Dialog

You can see what the dialog looks like when performing a task.

### Window Caption

Enter the caption for the dialog window.

### Save Result to Variable

Enter the name of the variable to which the dialog execution result should be saved.

### If Cancel Pressed

If the user declines to input, there are two possible actions:

- Abort task execution.
- Save the user's value to a variable as the execution result. An empty value is allowed.

### Appearance

Parameters for creating the appearance of the dialog.

### Window Size

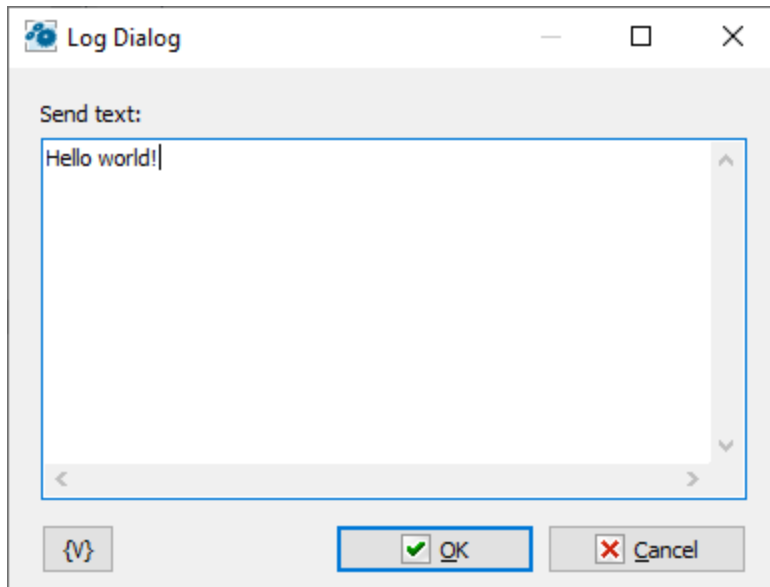
Enter the size of the screen window. Use the Test Dialog button to find the most appropriate option.

### 4.2.13 Log Dialog

This action is used to display user messages in a separate pop-up window. A separate window is opened for each task. This is useful for visualizing the progress of a task, for instance, during task debugging. The text size is limited to 10 000 lines.

This action does not stop the execution of the task, unlike the "[Show Message/Play Sound](#)<sup>[97]</sup>" or "[Show Text](#)<sup>[113]</sup>" actions. The next invocation of this action in the task simply appends text to the existing content.

The visible text in the window can be saved to a file or to the system Clipboard. If you close the window, the text is not saved.



#### Send text

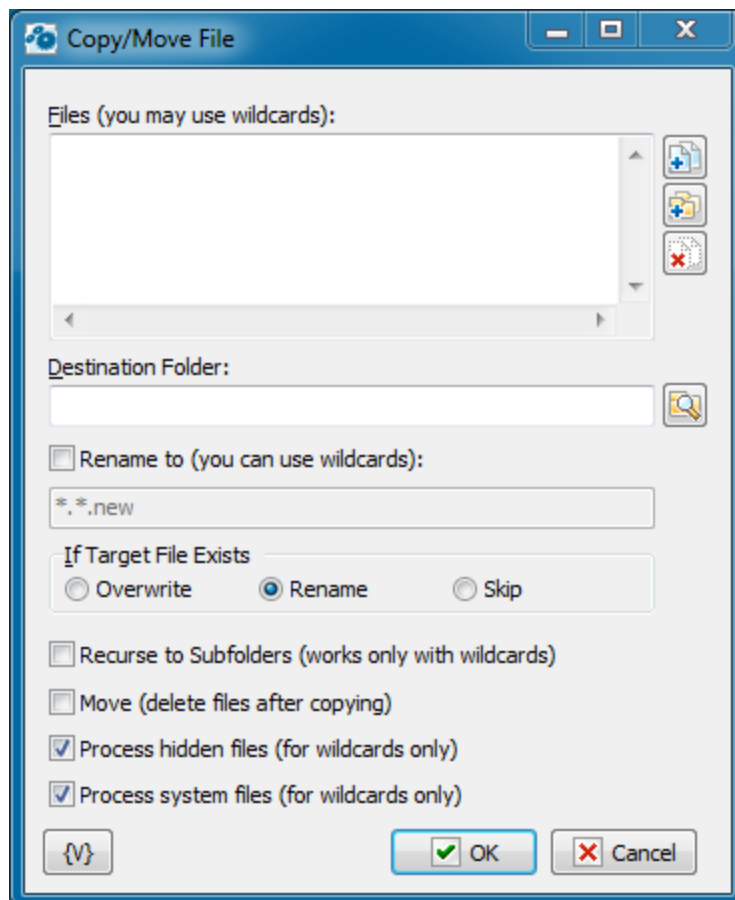
Enter the text that needs to be sent for display. You can use variables.

## 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.

### Rename to (you can use wildcards)

If you want to change the original name of file you should turn on this checkbox and specify new file name or the mask of new name in the field below. This option allows you to copy the file to new filename in the same folder.

### 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.
- **Skip** - the action does nothing and skip the file.

### Recurse to Subfolders

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

### Move

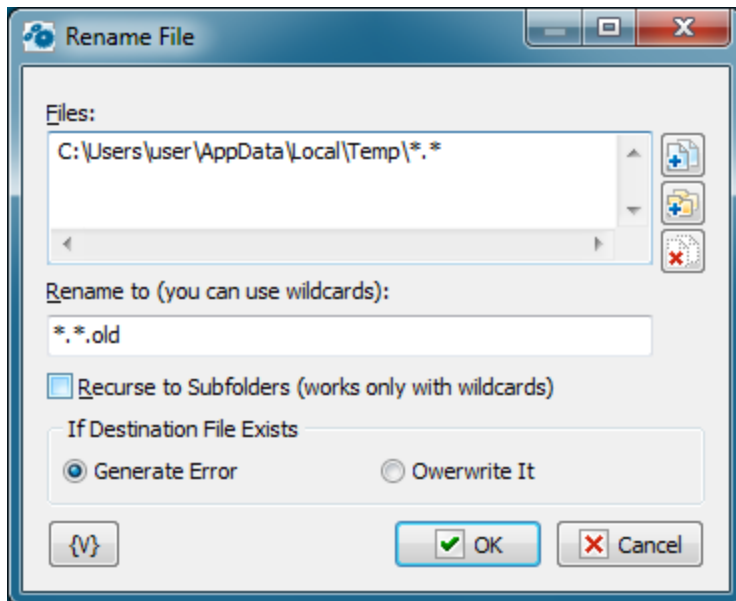
Deletes the original files after you copy or move them to a new location.

### 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**

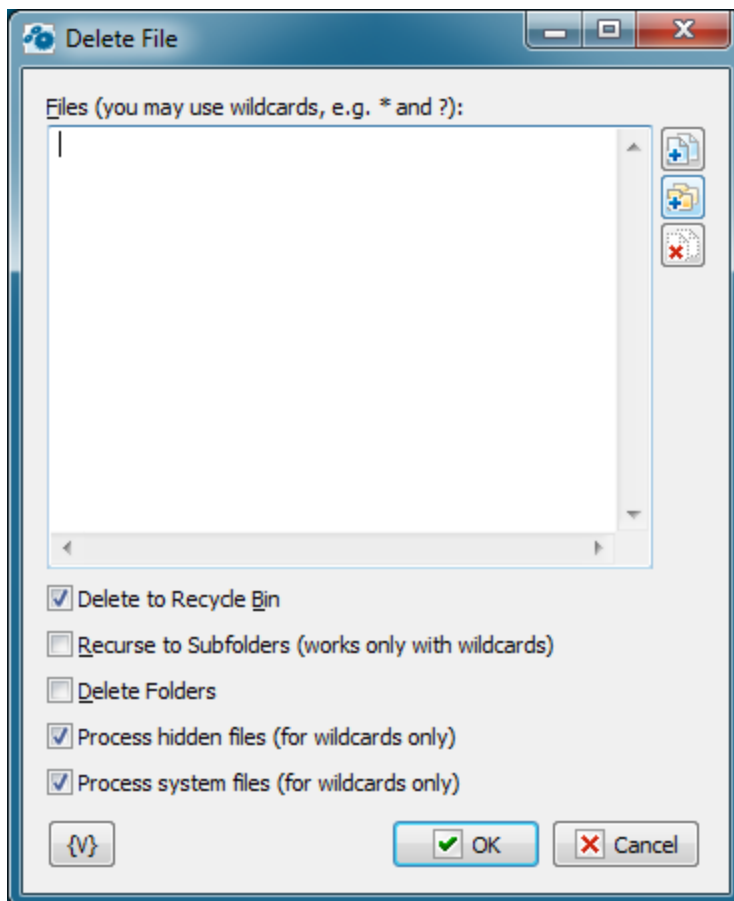
Renames the files in the subfolders of the current directory.

**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**

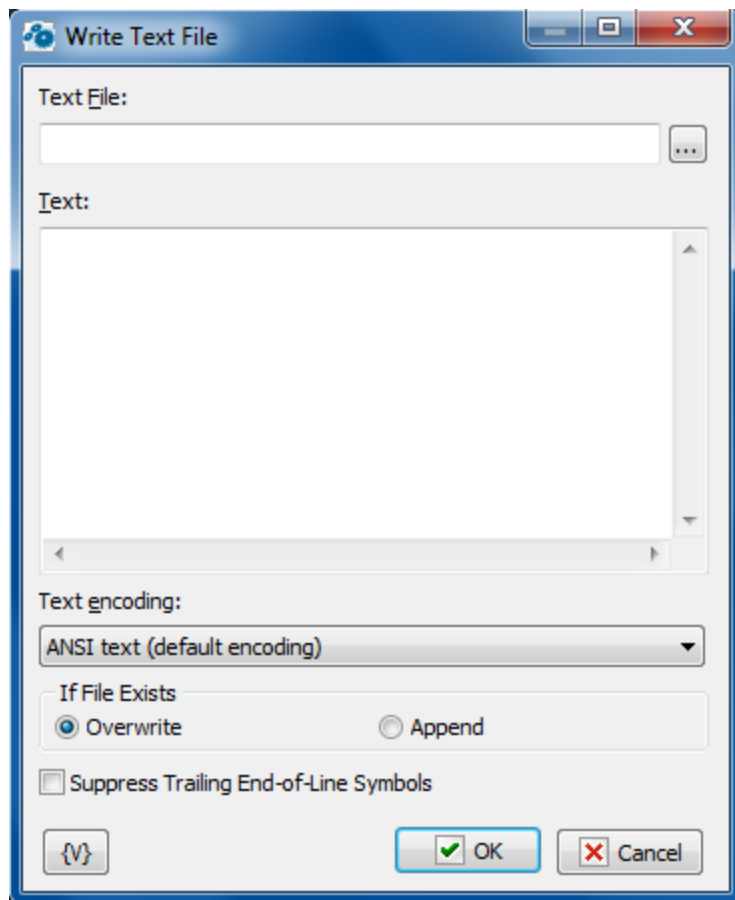
Removes the subfolders. This option works only with wildcards.

**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.

**Text Encoding**

The action can provide the following coding:

- ANSI text
- UTF-8 text
- Unicode text

**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.

**Suppress Trailing End-Of-Line symbols**

If you want to remove trailing EOL symbols switch on this checkbox.

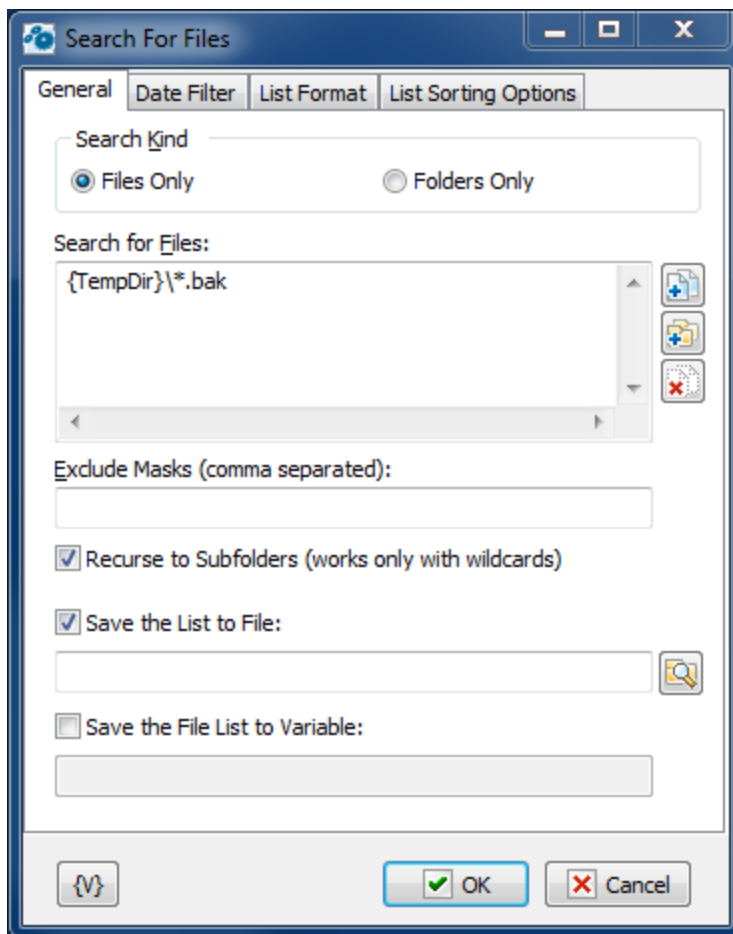
**Variable**

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.

### 4.3.5 Search for Files

Looks for files and creates a file list as multi-line text. Each file name on new line of text. 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.

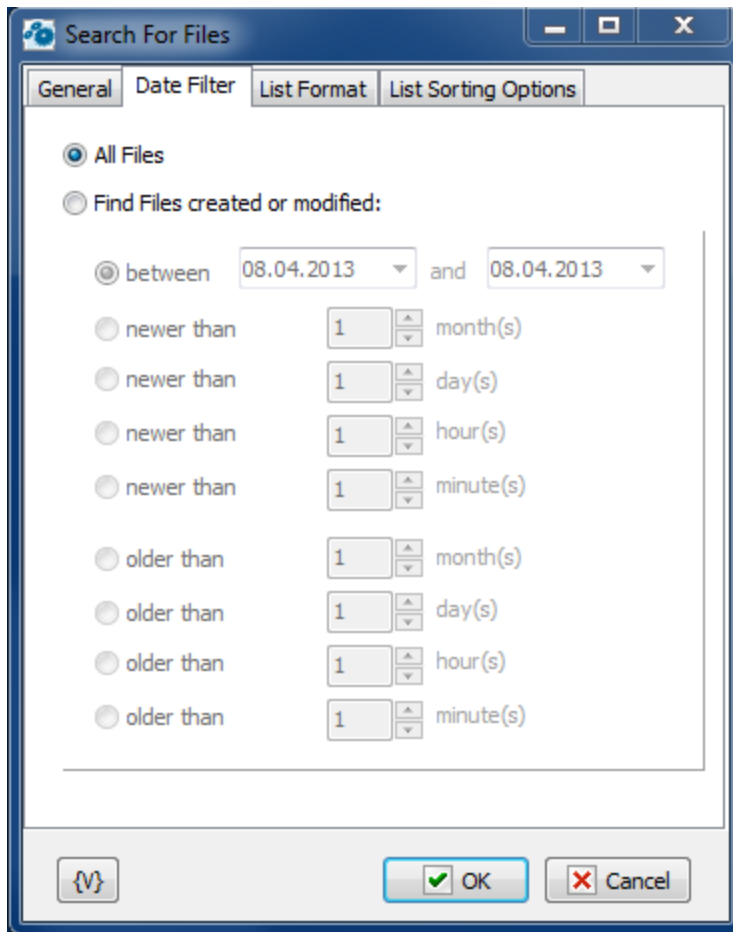
### Save File List To

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.

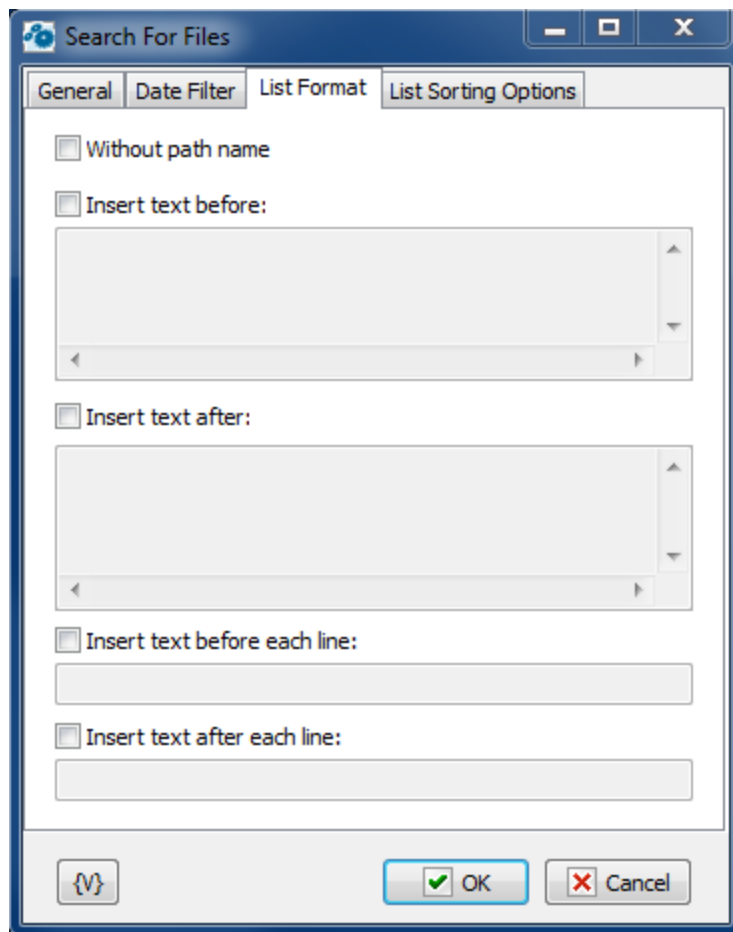
### Find Files Created or Modified

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.

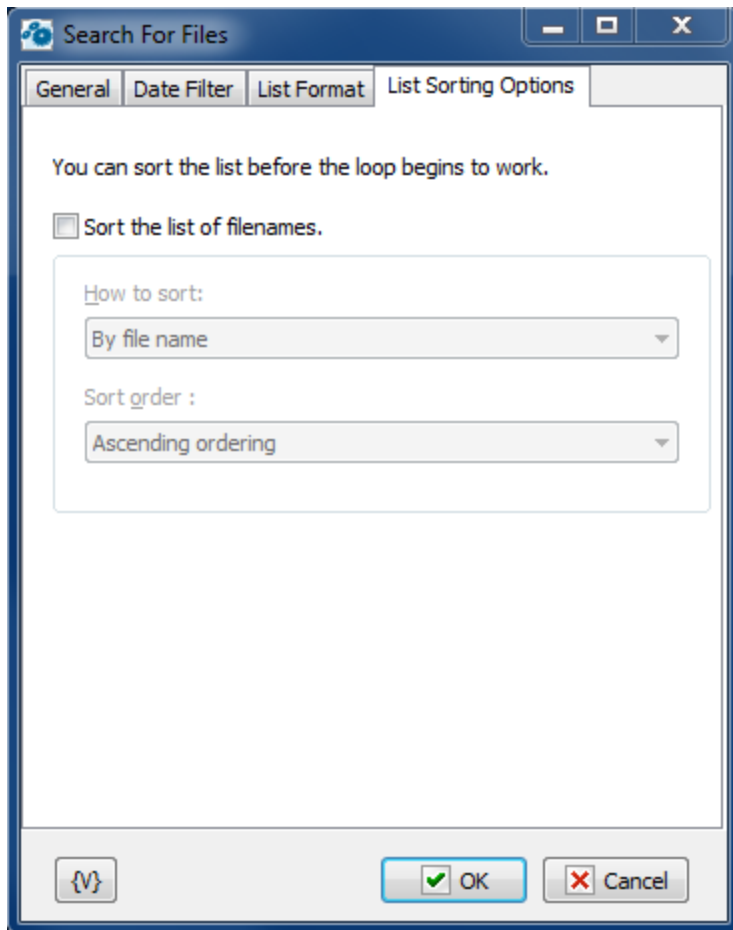
**Insert Text Before Each Line**

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.

**List Sorting options**



Parameters of internal sorting of the file list

### **Sort the list of filenames**

Turn on the checkbox if you want to sort the list

### **How to sort**

There are four sorting modes:

- By file name
- By file time
- By file size
- By depth of nested folders

### **Sort order**

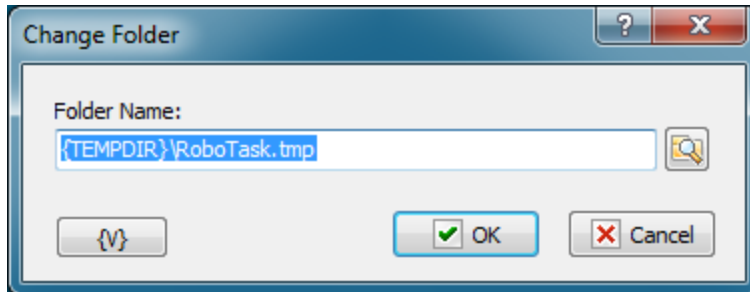
There are two modes: ascending or descending ordering

### **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 Set Current 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

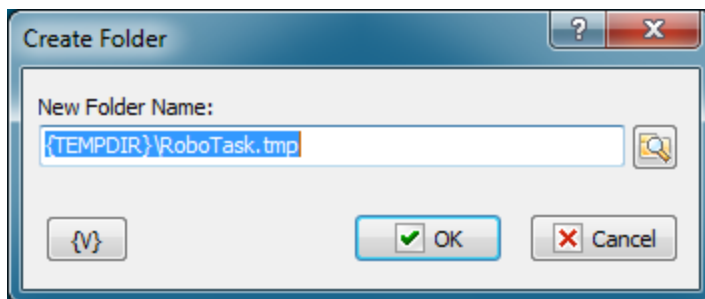
Enter the name of the destination folder. You can click the browse button to select a folder.

#### 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

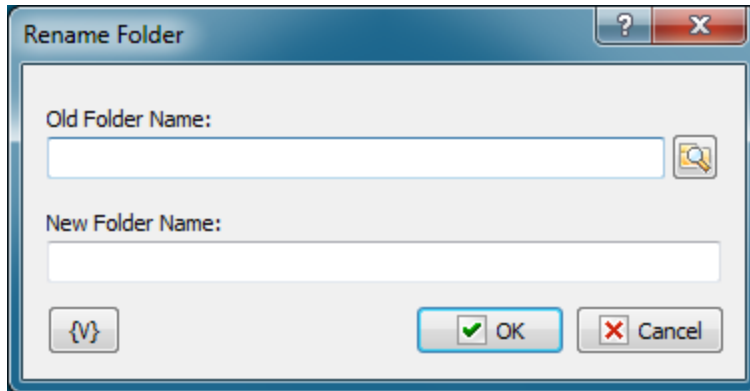
Enter the name for the new folder. You can click the browse button to select a folder.

#### 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**

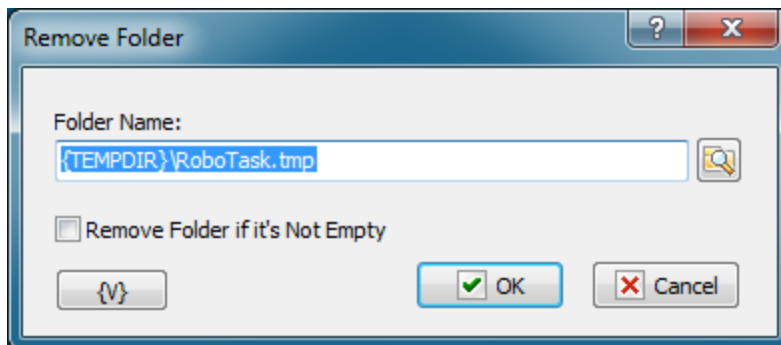
Enter the new name for the folder.

**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**

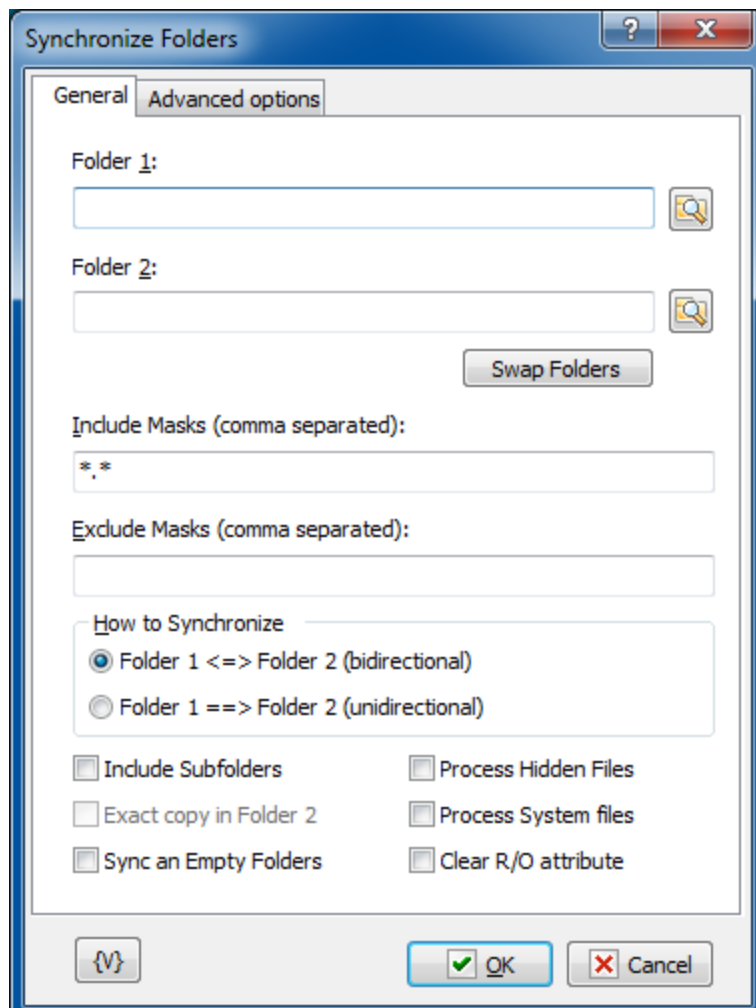
If this option is selected, the specified folder will be deleted even it is not empty (i.e. contains files and subfolders), otherwise the specified folder will be deleted only if it is not empty (i.e. does not contain files and subfolders).

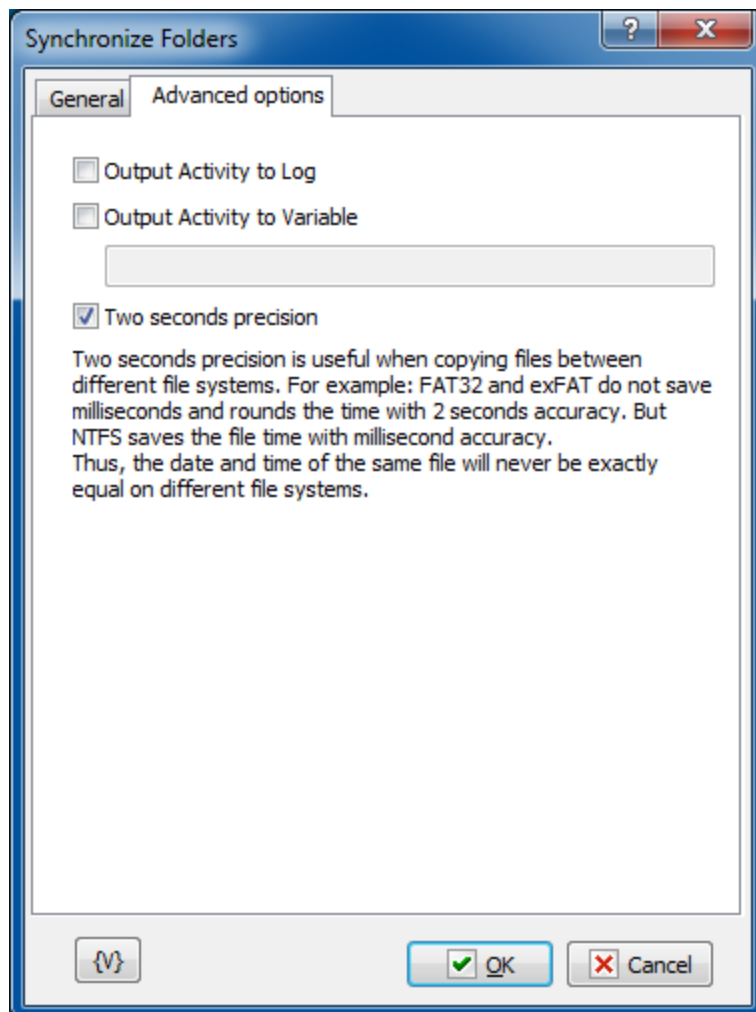
**Variable**

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.





## General tab

### 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 1, 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.

### Sync an Empty Folders

If this checkbox is switched on, the action copies the structure of empty folders. Otherwise, only files will be synchronized.

## Advanced options

### Output Activity to Log

If you want to write the log of synchronization into the task log, switch on this checkbox. The action will put the information about copied and removed files into the task log.

### Output Activity to variable

If you want to write the log of synchronization into a variable, switch on this checkbox and put the variable name in the field. The action will put the information about copied and removed files into the log.

### Two seconds precision

Two seconds precision is useful when copying files between different file systems. For example: *FAT32* and *exFAT* do not save milliseconds and rounds the file time with 2 seconds accuracy. But *NTFS* saves the file time with millisecond accuracy.

Thus, the date and time of the same file will never be exactly equal on different file systems.

### 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

Wait for file:

Waiting Mode

Wait for accessibility of the file  Wait for removal of the file

Wait for appearance of the file

Waiting timeout

Timeout 60 sec

Raise TIMEOUT error (default option)

Infinity

{V} OK Cancel

#### Wait for file

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

#### Waiting mode

There are two modes of waiting

- Wait for Accessibility of the file
- Wait for Appearance of the file
- Wait for removal of the file

#### Waiting timeout

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 you can specify an action:

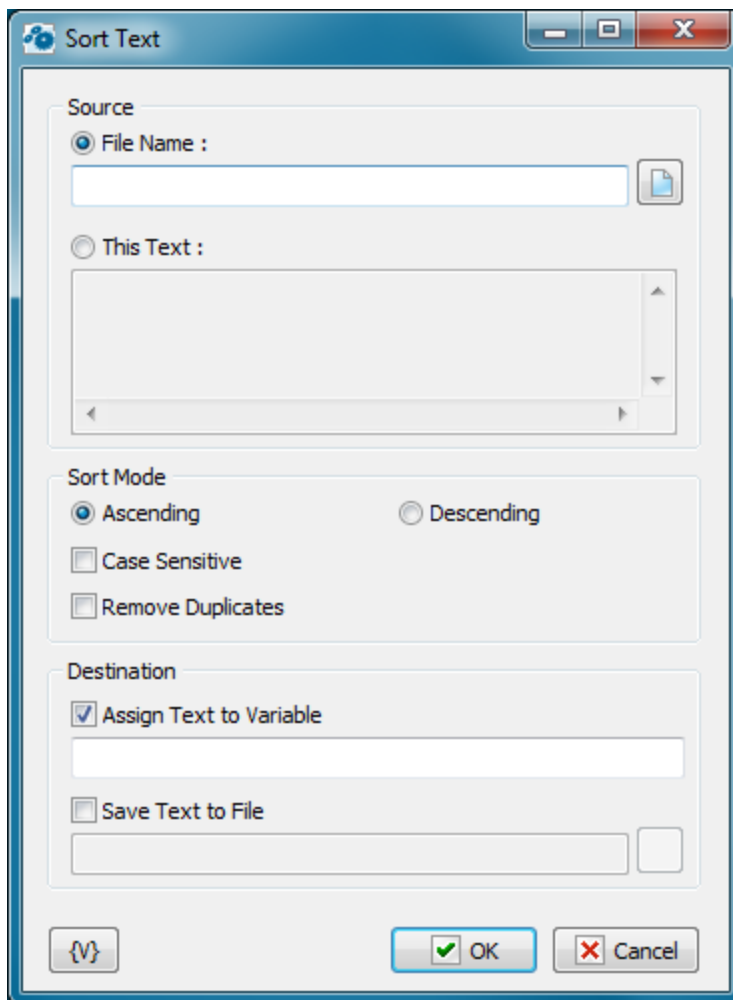
- **Raise Timeout error** - this is default option
- **Set True or False to variable** - in this case the task will continue work without error.

### 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 Sort Text

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
- Remove duplicates - this allows to keep only unique lines in the text.

### 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.

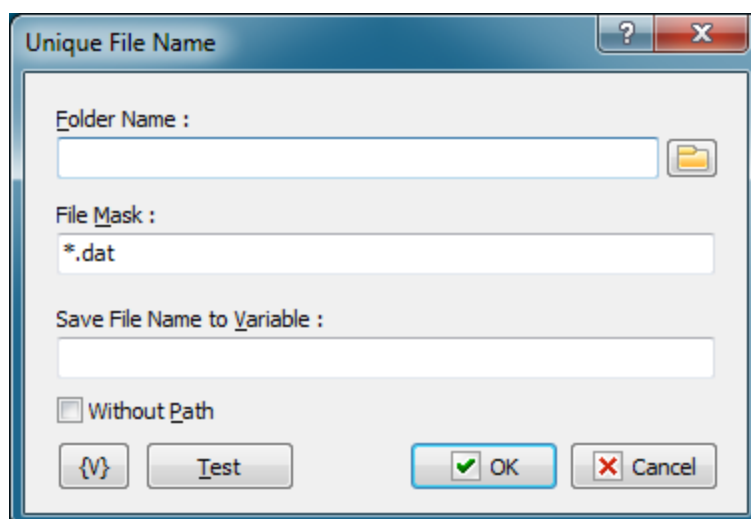
**Save text to File** – specify the full file name either manually or using the Open File dialog box. It is possible to use RoboTask variables.

### 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

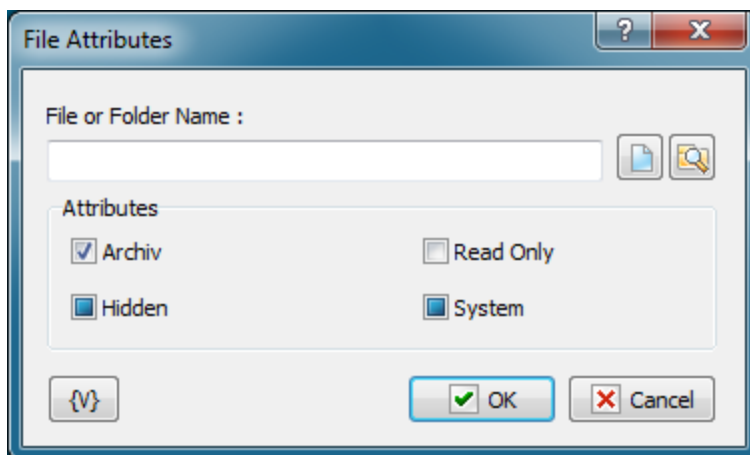
If you want to save the file name only without path, check this checkbox.

### 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

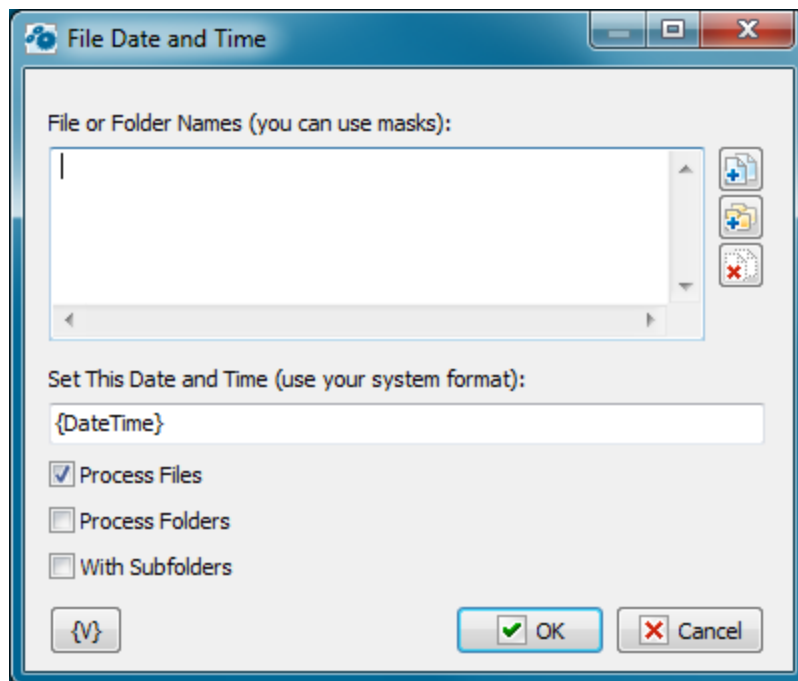
Please, notice that Checkboxes have three positions:

- **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.

### 4.3.15 File Date and Time

This action is for changing date and time of files or folders. Date and time are specified as a line. The format has to correspond to the regional settings of your computer. To check the date format it is enough to find out what the form which the variable **{DateTime}** returns. If the date and time are specified incorrectly, you will get run-time error.



### File and Folder Names

Enumerate file names, one name per line. The usage of a file mask is allowed (for example, c:\incoming\\*.txt). It is also allowed to use variables when the file name is formed.

### Set This Date and Time

Specify the date and the time which are necessary to assign to the files. The date and the time are specified as string. The format has to correspond to the regional settings of your computer. The usage of variables is allowed.

### Process Files

Switch on this checkbox if you want to process the files corresponding to the list.

### Process Folders

Switch on this checkbox if you want to process the folders corresponding to the list.

At least one of two options "Process Files" or "Process Folders" has to be on. It is possible to switch on both options at the same time.

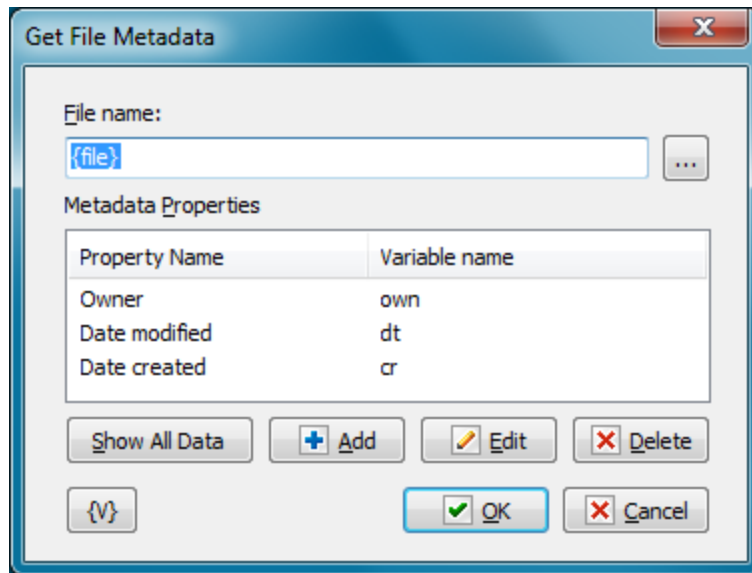
## 4.3.16 Empty Recycle Bin

This action empties a recycle bin of a user on all hard disks. The action does not required any additional settings.

## 4.3.17 Get File Metadata

This action is designed to extract the system properties of any file. In Windows Explorer, you can see these properties if you call the context menu of the file and select *Properties -> Details tab*

**Note** that the composition of these properties (metadata) depends on the file type. And also the metadata names depend on the system language and the version of Windows™. That is, in different systems, the same property can have different names



### File Name

Specify the file name. You can use variables. You can also select a file interactively

### Metadata Properties

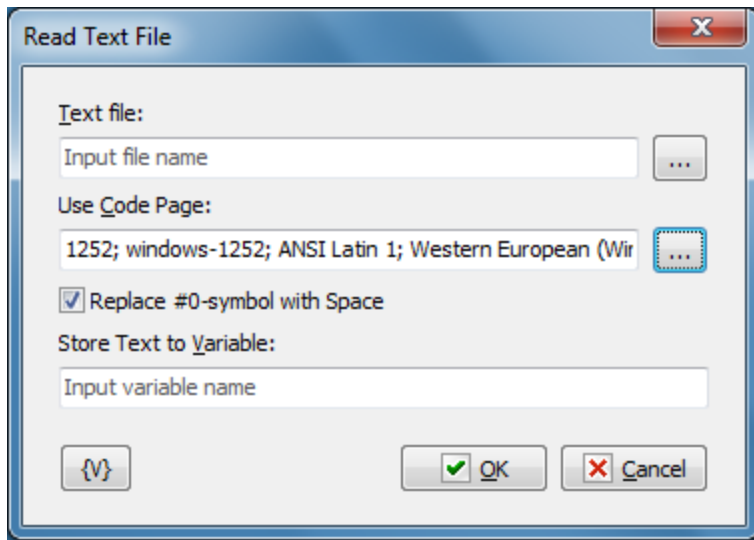
The list of necessary properties. The value of each metadata property will be stored in a separate variable for the next processing.

### Show All Metadata

You can see all non-empty metadata properties if you specify file name explicitly.

## 4.3.18 Read Text file

This action is necessary to read a text file with plain text, but with a code page different from the system one. If the code page is different from the system page, then additional text conversion is required for correctness. In addition, the text can contain null-characters, which "breaks" the text.

**Text file**

Specify the name of text file to read.

**Use Code page**

Choose the codepage of the text. By default this parameters equals to system code page.

**Replace #0-symbol with Space**

Switch on this checkbox if you want to replace all #0 symbols in the text with spaces

**Store Text to Variable**

Enter the name of the variable in which you want to save the text.

### 4.3.19 Get File Hash

A hash value is a unique value that corresponds to the content of a file. A hash can be considered as a file identifier regardless of its name. There are many algorithms for calculating the hash of a file. This action allows you to use the most popular algorithms for calculating the hash.



Get File Hash

File name:  
Enter or select file name

Hash algorithm:  
MD5

Save result to variable:  
Enter variable name

{V} OK Cancel

**File name**

Enter or select file name.

**Hash algorithm**

Choose the algorithm. The are:

- CRC32
- MD5
- SHA1
- SHA256
- SHA512

**Save result to variable**

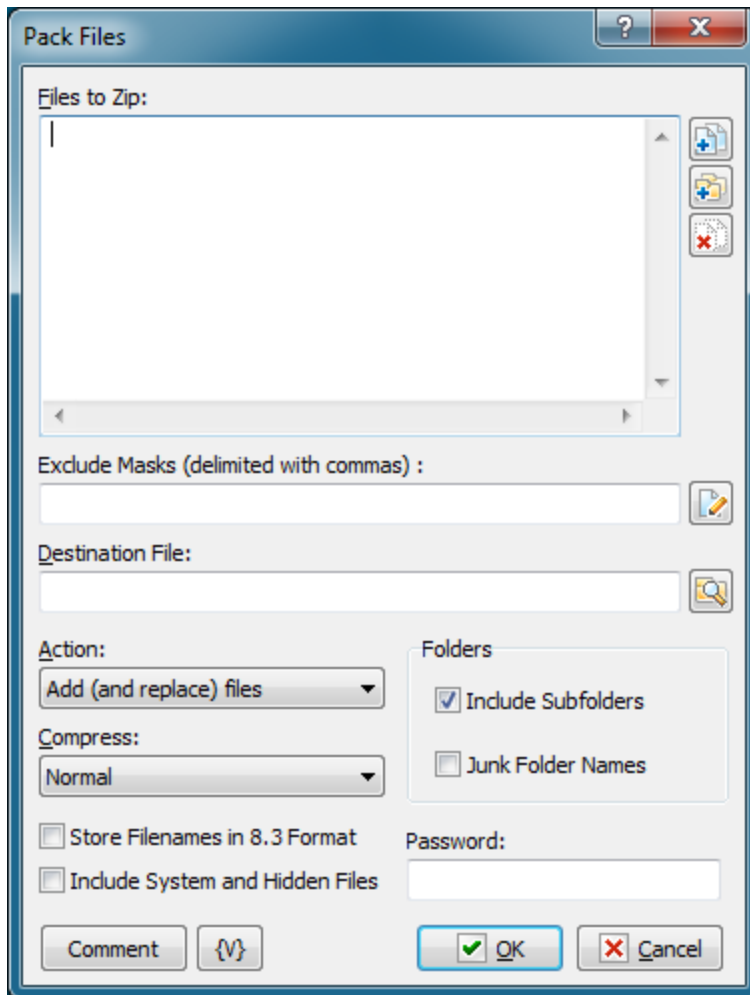
Enter the name of the variable in which you want to store the value. Hash value is an alphanumeric string

For example: `892B0DF594808BF7F6B2C585F8C00775`

## 4.4 Zip

### 4.4.1 Zip 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

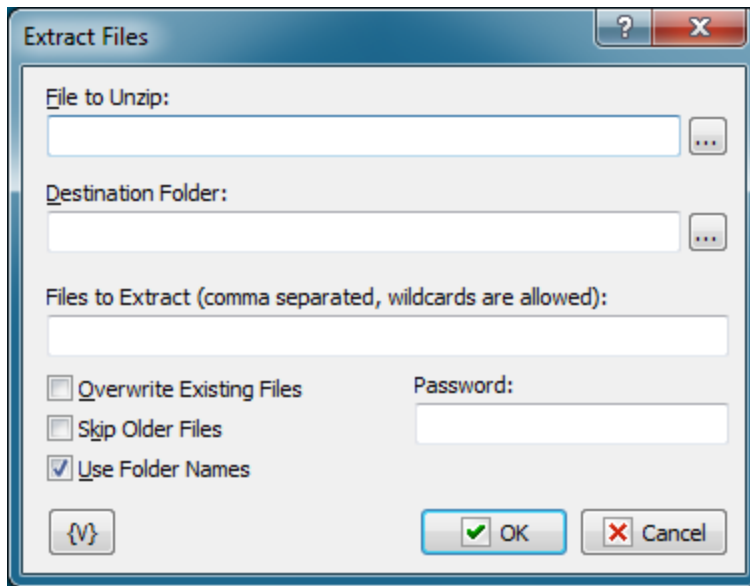
Enter a memo note for the zip file.

#### 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 Zip 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.

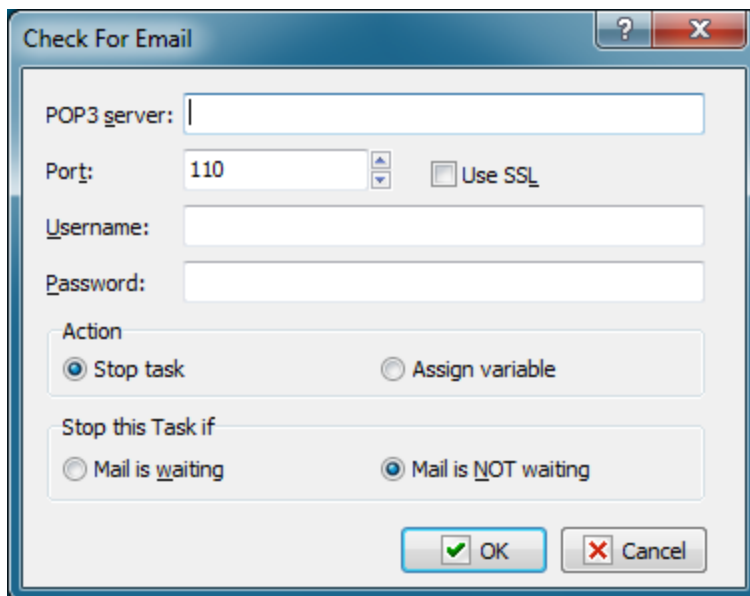
## Variable

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.



#### 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

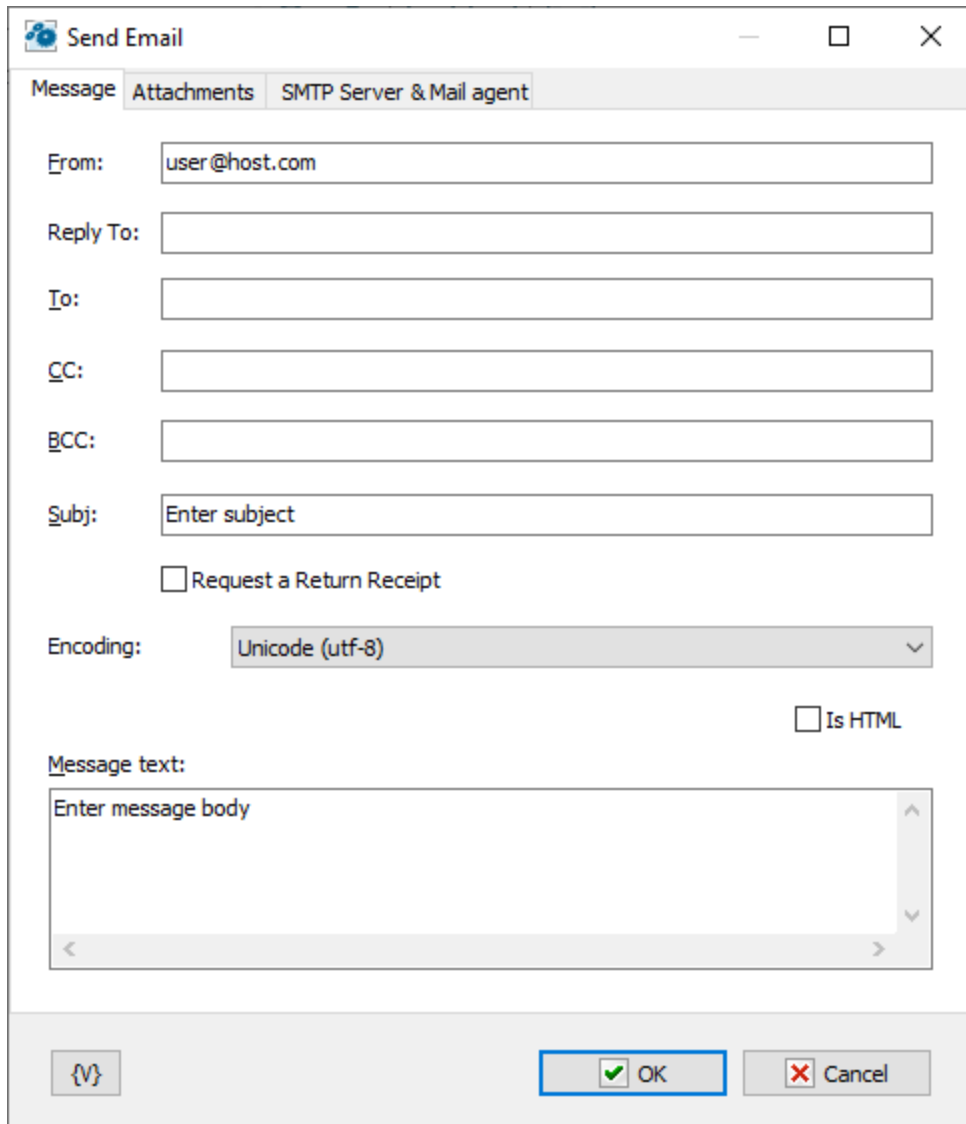
Enter whether you want the current task to do one of the following before e-mail is downloaded from the mail server:

- **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 dialog box titled "Send Email" with three tabs: "Message", "Attachments", and "SMTP Server & Mail agent". The "Message" tab is active. It contains the following fields and options:

- From:** user@host.com
- Reply To:** (empty)
- To:** (empty)
- CC:** (empty)
- BCC:** (empty)
- Subj:** Enter subject
- Request a Return Receipt
- Encoding:** Unicode (utf-8) (dropdown menu)
- Is HTML
- Message text:** Enter message body (text area)

At the bottom, there is a button with curly braces {}, and two buttons: "OK" (with a green checkmark) and "Cancel" (with a red X).

#### From

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

#### Reply to

---

If the reply-to address is different from the sender's address, you can provide an alternate email address for replies in this field. You can leave this field blank.

**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.

**Request a Return Receipt**

If you want to get a confirmation of a reading, switch on this checkbox. Pay attention, it does not guarantee getting a receipt even if an addressee has opened and read the letter. It depend on settings of a mail client of the addressee.

**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.

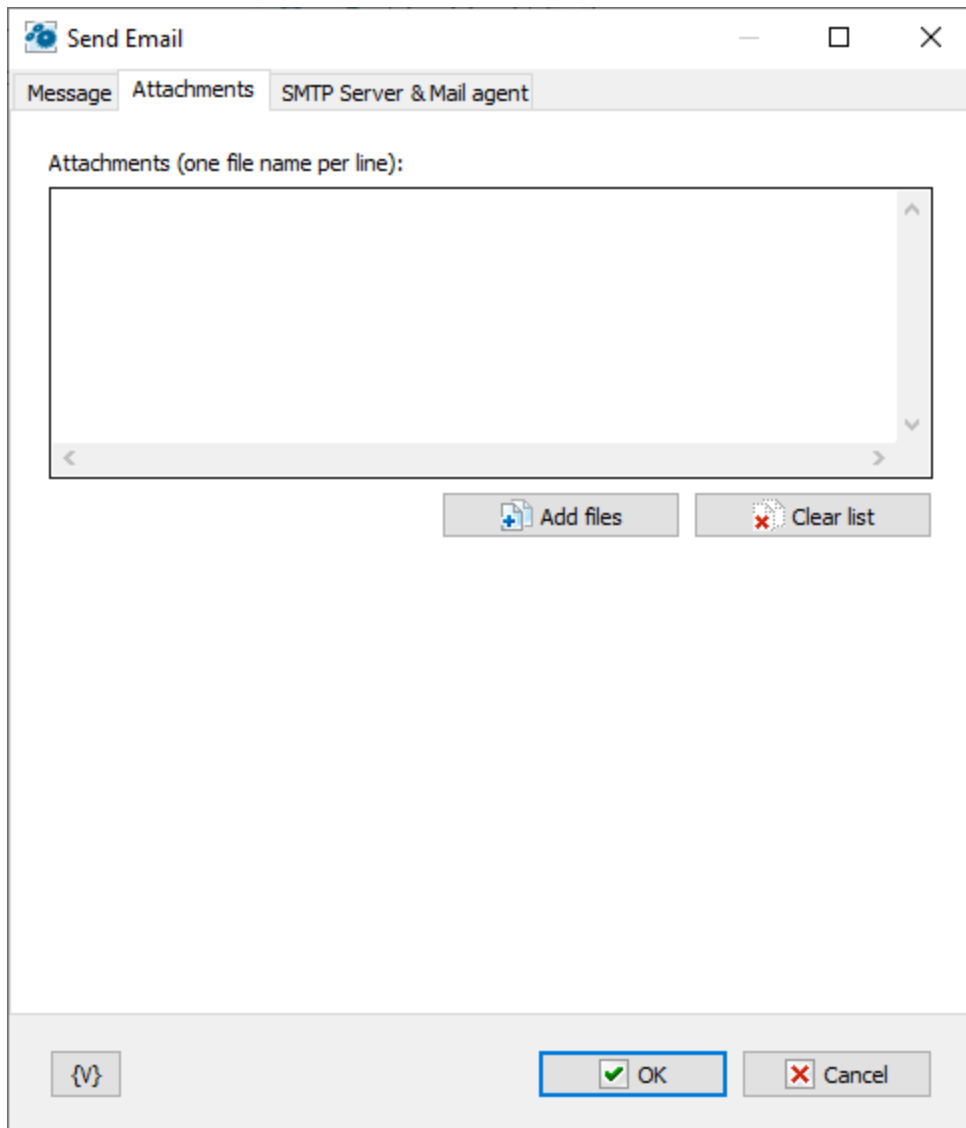
**Tip**

- 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.

**Attachment tab**



### Attachments

Attachments list is the list of file names, one file name per line.

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 & mail agent tab



The screenshot shows a dialog box titled "Send Email" with three tabs: "Message", "Attachments", and "SMTP Server & Mail agent". The "SMTP Server & Mail agent" tab is active. It contains the following fields and controls:

- Host:** A text input field.
- Port:** A text input field containing the value "25".
- SSL/TLS mode:** A group box containing three radio buttons: "None" (selected), "SSL", and "TLS". Below the "TLS" radio button is a dropdown menu showing "TLS 1.2".
- Authentication Required:** A checkbox that is currently unchecked.
- User Name:** A text input field.
- Password:** A text input field with a "Show Password" button to its right.
- Mail Agent:** A text input field containing the value "RoboTask".
- Message Priority:** A dropdown menu showing "Normal".
- 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.

**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.

**SSL/TLS mode**

You can set SSL/TLS mode of the connection. There are three options:

- None
- SSL
- TLS

**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

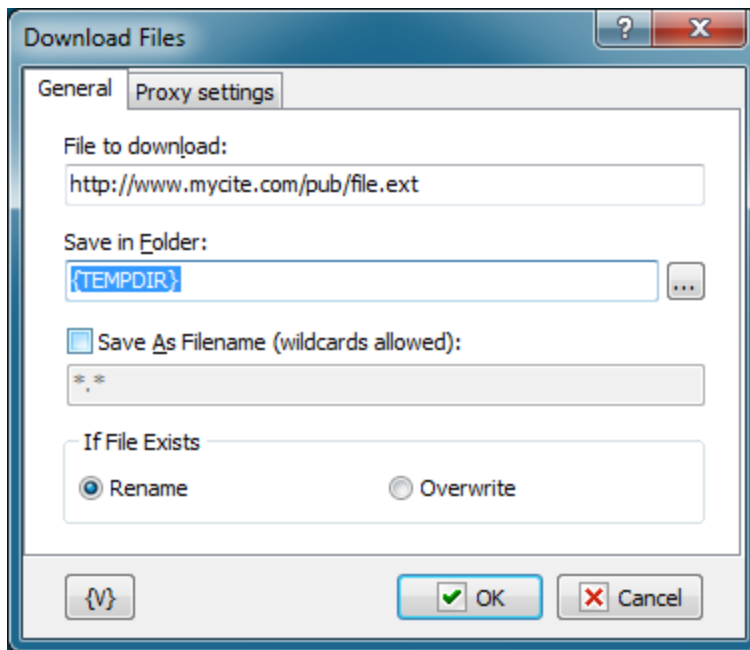
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 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

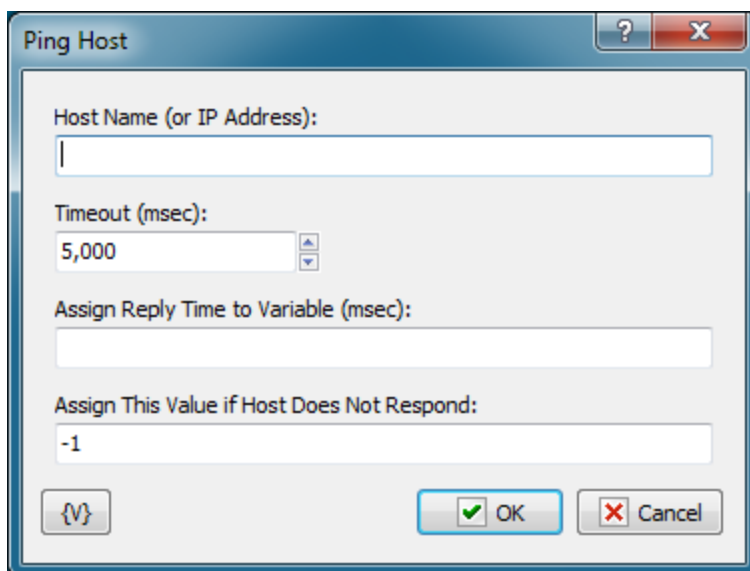
Allows you to add a variable to the action or triggering event that you selected. You must

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**

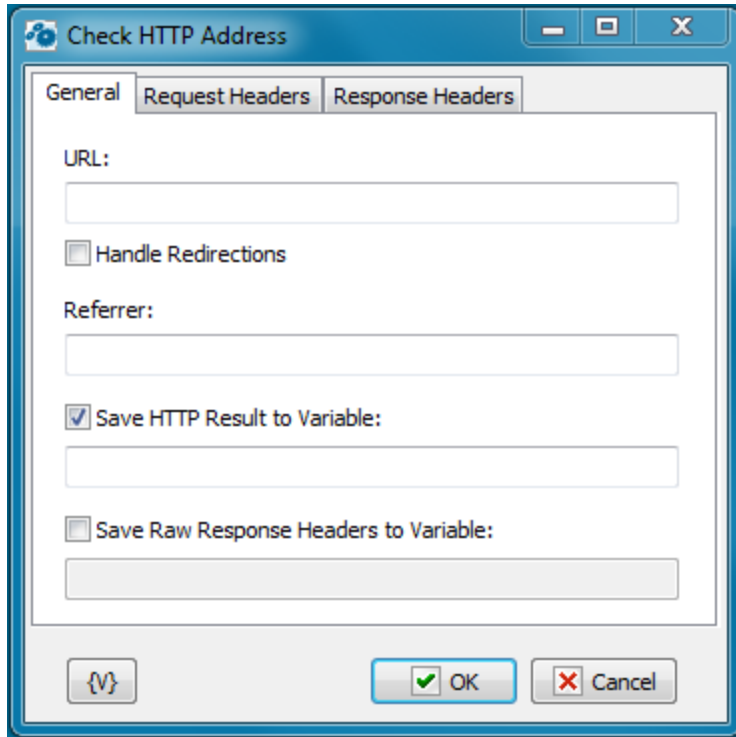
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then

click the {V} button.

#### 4.5.5 Check HTTP address

This action is for checking the Web URL and receiving the captions of the response to analyze. The action works according to the **HTTP** and **HTTPS** protocols.

##### General tab



##### URL

Specify the HTTP address of the checked click link. For example: *http://www.robotask.com/*  
You can specify addresses with parameters: *http://www.robotask.com/forum/viewtopic.php?f=4&t=1441*

It is also allowed the usage of the Secure HTTP (HTTPS) addresses. For example:  
*https://www.google.com/search?q=robotask*

The address must begin from the **http://** or **https://**

##### Handle redirections

Switch on this option if you want the action to process the redirection of the query automatically. But you should remember that you will get the response from the final URL.

##### Referrer

Specify a referrer for a query. Referrer is the URL from which this HTTP query was performed. Many sites use this parameter of query to form a response.

### Save HTTP Result into Variable

If you need to save the HTTP result into a variable, switch on this checkbox and input the variable name. HTTP result is a number. For example:

- 200 – OK
- 403 Forbidden
- 404 – page not found
- etc.

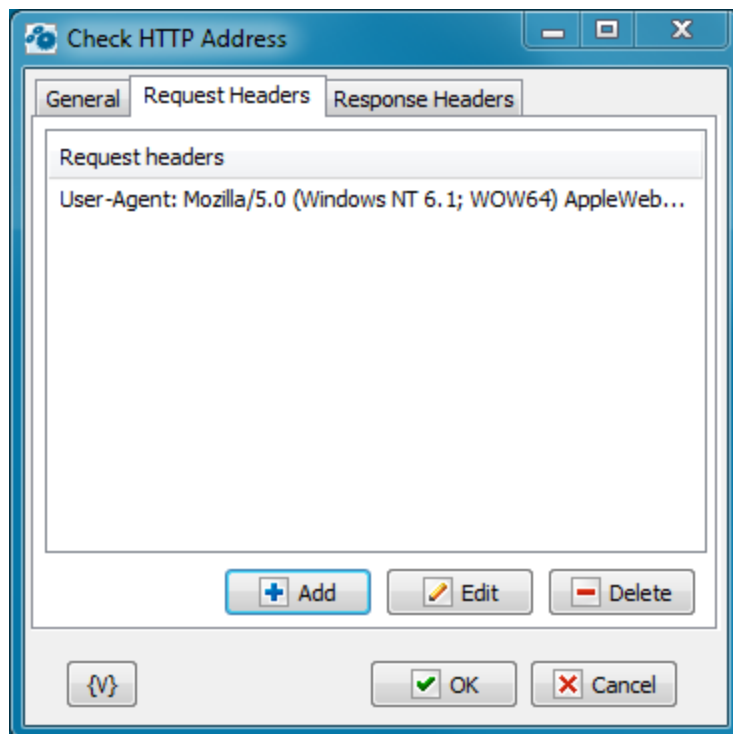
You can read more about HTTP results here:

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

### Save Raw Response Headers into Variable

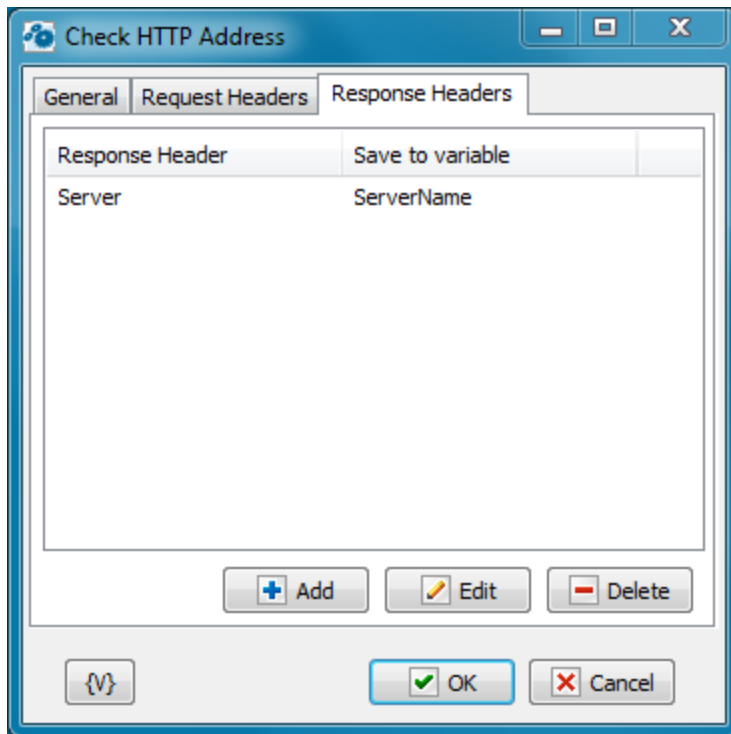
To save all captions of the response into a variable, switch on this checkbox and input the variable name.

## Request headers



You can define the user captions of the request on this tab. You can add captions on a list, edit or remove the selected caption.

## Response headers



If you need to save values of certain captions into variables, you should use this tab.

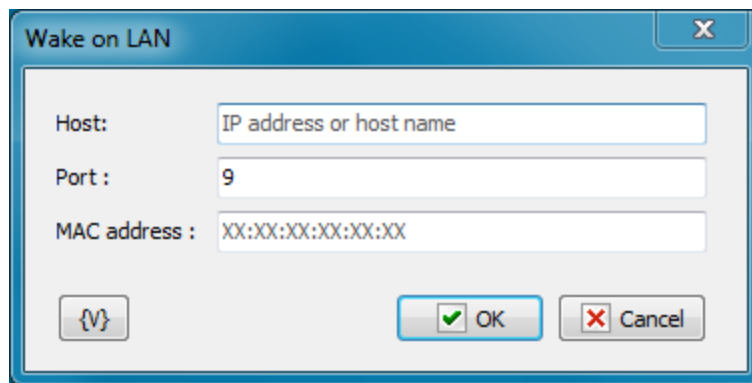
To add a caption on a list, it is necessary to specify the caption of the response and the name of the variable into which you need to save the value of the caption. If the specified caption does not exist, the action saves empty value.

#### 4.5.6 Wake On LAN

The action permits to "wake up" another computer by means of sending the special signal through a net. You can read about this technology here in detail: <https://en.wikipedia.org/wiki/Wake-on-LAN>. To have the computer awoken, this function should be permitted in the setting BIOS of the computer. As a rule, this function is permitted on default.

The function Wake-on-LAN works stably for stationary (desktop) computers with wire connection to network. The Wake-On-LAN permit to "wake up" a computer out of any status: sleep, hibernate and even switched off. The necessary condition is that the computer should be connected to a power supply (an outlet) and have physical connection to a network.

The possibility of awakening for laptop depends on the manufacturer and the model of computer and perhaps it depend on the observance certain condition. For example, in our tests it is necessary that the lid of the laptop was opened and the computer was connected to its power module.



### Host

The name of the computer or an IP address. You can specify both an exact address (for example, 192.168.1.36) and a broadcast address (for example, 192.168.1.255).

In the mode of one segment of a local network it is permitted even not to specify any exact address of a sub-network and to do a broadcast request to all computers of the segment (for example, 255.255.255.255).

### Port

The port of recipient. On default, it is 9. You can specify any port if you use the local network without firewall. If a target computer is in a segment of the network closed with firewall, you should tune a port forwarding to the signal can get the target.

### MAC address

This is a hardware address of a network interface (a physical address). To retrieve it, it is necessary to specify the command in the command line:

```
ipconfig /all
```

You can see a MAC address through the control panel:

*Control Panel -> Network and Sharing Center*

Then, you should click the link *Local Area Connection* (or another connection) and open the window "Details".

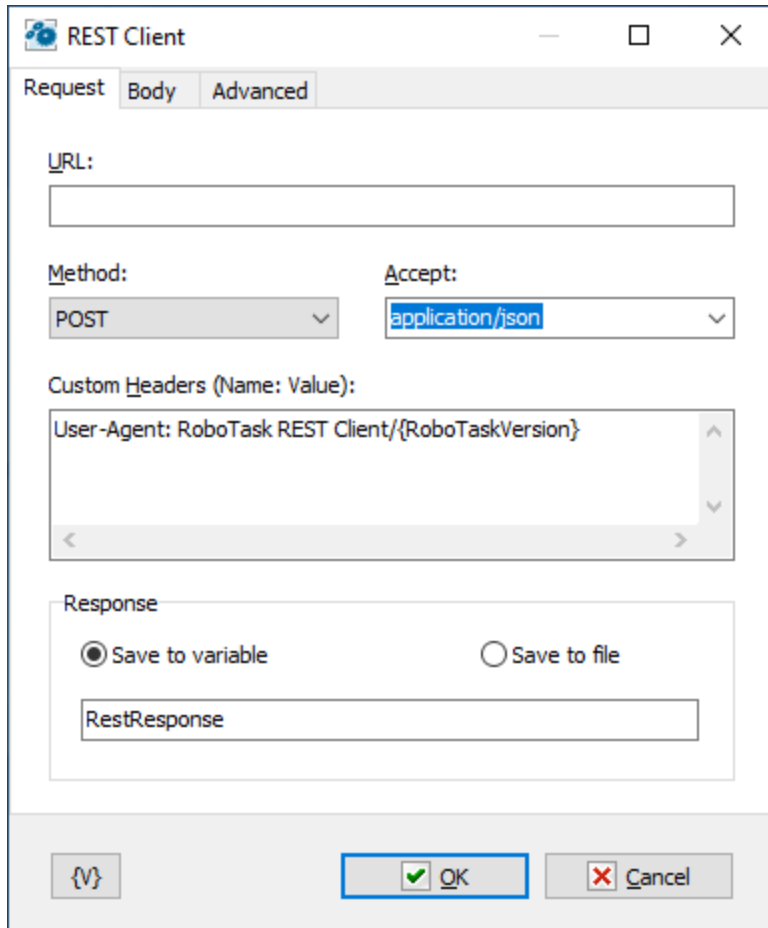
MAC address is important parameter. The function Wake-On-LAN works only if the specified address coincides with a hardware one. Otherwise, the target computer just ignores the command. Therefore, in the local network it is possibly only to send a broadcast request, because the computer reacts only when MAC address coincides. The rest of computers of the segment just ignore the request.

## 4.5.7 REST Client/Invoke Web Service

This action makes HTTP requests and receives responses from the Web service.

### Request

Request parameters

**URL**

WEB address of the service. The address must start with the protocol: *HTTP://* or *HTTPS://*

**Method**

Request method. Select the desired option from the list.

Available methods are:

- GET
- POST
- PUT
- DELETE
- PATCH
- HEAD
- CONNECT
- OPTIONS
- TRACE

**Accept**



The value of the request header *"accept"*. You can choose from the list or write your own.

### Custom headers

If you need to specify additional headers, you can specify them in this field

Header format:

```
<header_name>: <header_value>
```

### Response

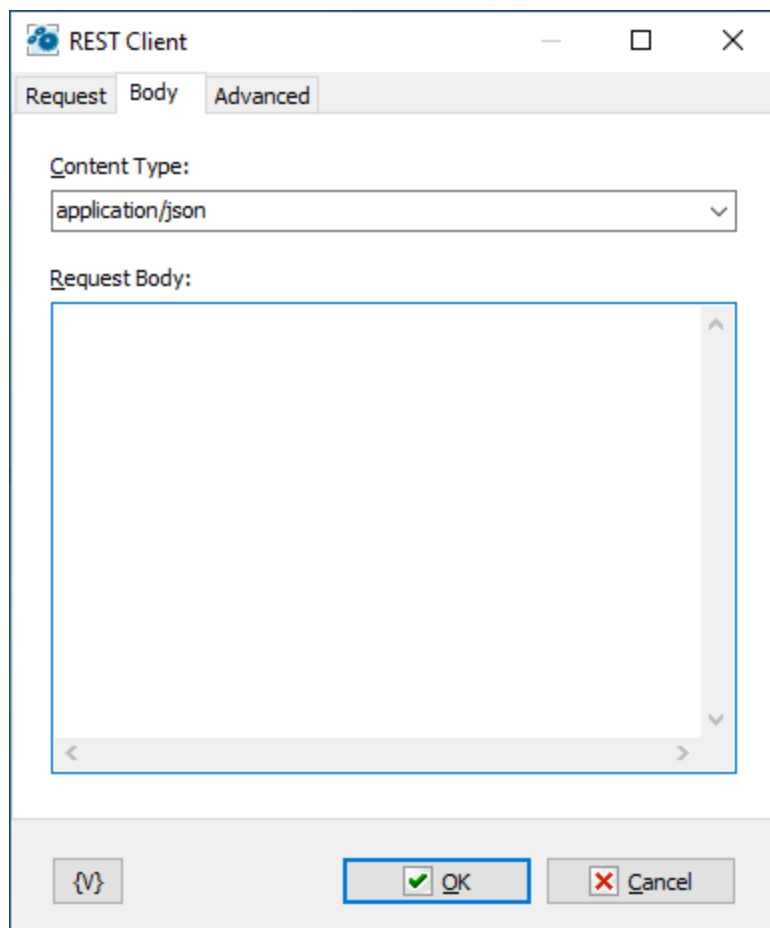
The response value can be written to a variable or to a file. Select the desired value and enter the variable name or file name respectively.

The value of a variable or file name, may be empty. In this case, the value will not be stored.

### Body

Filling in these parameters is possible if the request method is *POST*, *PUT* or *PATCH*.

The remaining methods do not require a request body.



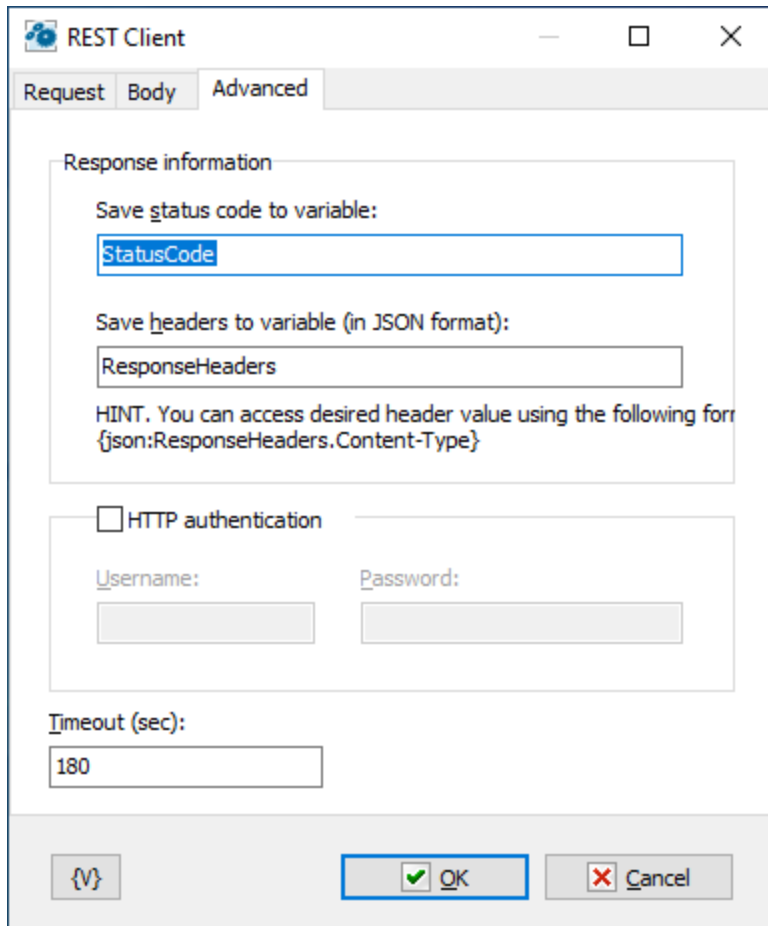
### Content type

Select a product from the list or enter your own option.

## Body

There is no single format or rule for how to form a request body. This should be specified in the web service documentation.

## Advanced



The screenshot shows the 'Advanced' tab of the REST Client application. It features three main sections: 'Response information', 'HTTP authentication', and 'Timeout (sec)'. The 'Response information' section includes a text box for 'Save status code to variable:' containing 'StatusCode', and another for 'Save headers to variable (in JSON format):' containing 'ResponseHeaders'. A hint below these fields reads: 'HINT. You can access desired header value using the following for {json:ResponseHeaders.Content-Type}'. The 'HTTP authentication' section has an unchecked checkbox and two text boxes for 'Username:' and 'Password:'. The 'Timeout (sec):' section has a text box containing '180'. At the bottom, there are three buttons: a '{}' button, a checked 'OK' button, and an 'X' 'Cancel' button.

### Save status code to variable

Enter a variable name. This field can be empty.

### Save headers to variable

Enter a variable name. This field can be empty.

### HTTP authentication

If the web service requires authorization, then enable this checkbox and enter a username and password.

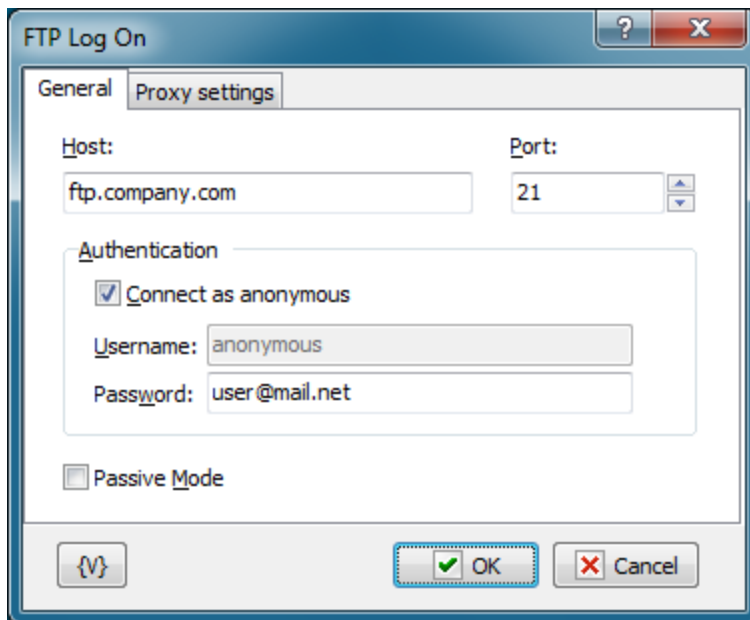
**Timeout**

Enter the response timeout. The default value is 180 seconds.

## 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**

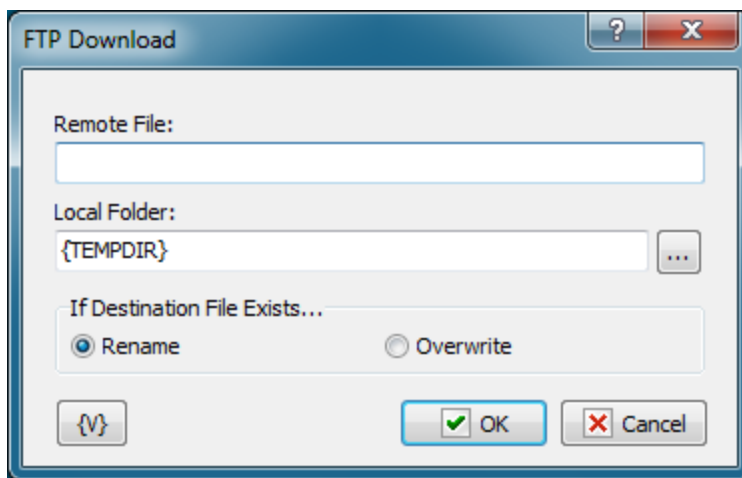
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 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

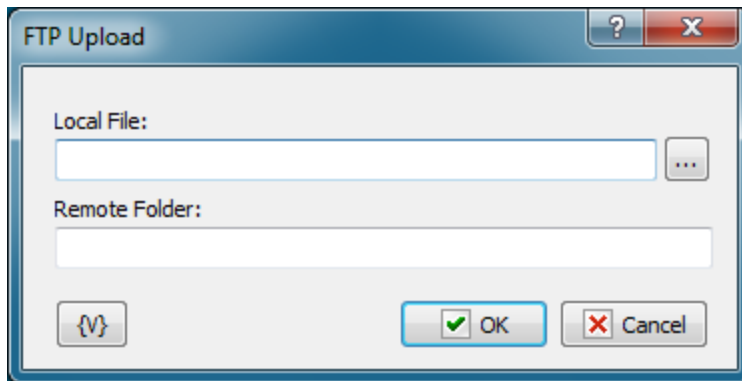
Enter the path and name of the local directory to which you want to copy the remote file.

#### 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**

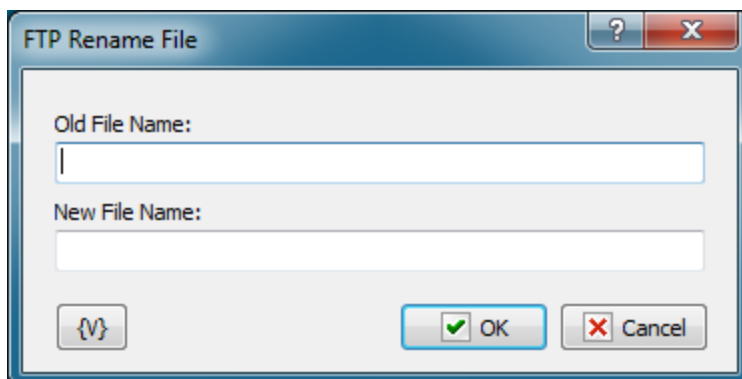
Enter the path and name of the directory on the remote server.

**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**

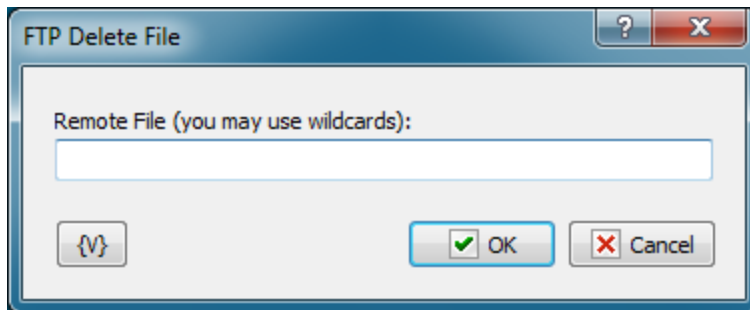
Enter the name that you want to give the file.

**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**

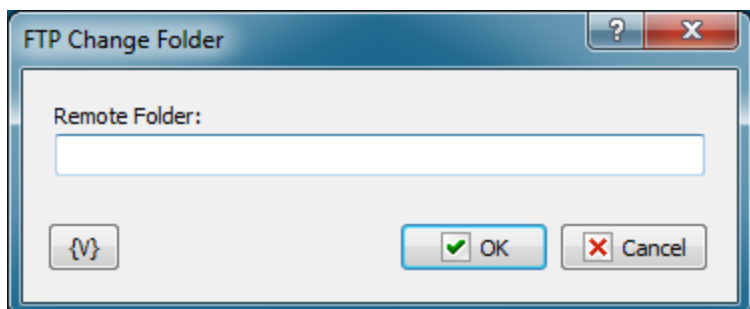
Enter the path and file name that you want to delete on the remote server. Click the {V} button to insert a variable that holds the path and name of the file.

**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**

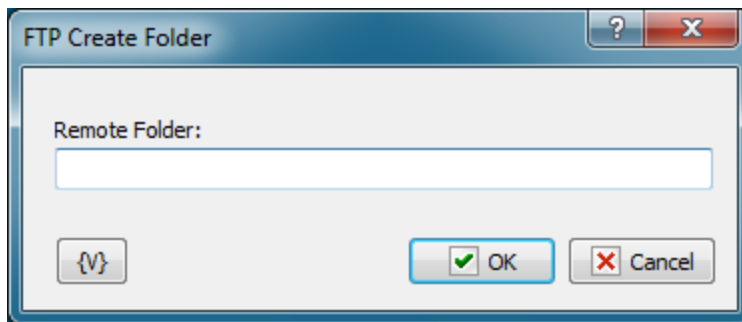
Enter the path or folder name on the remote server that you want to switch to. Click the **{V}** button to insert a variable that holds the name of the folder.

**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**

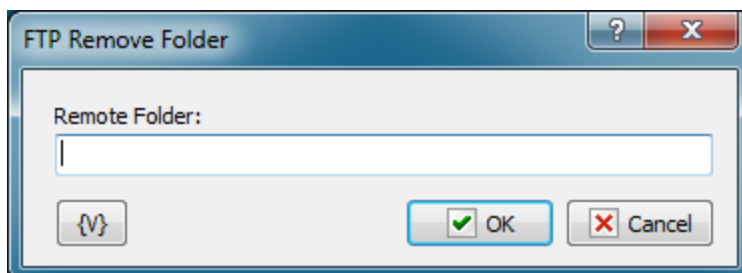
Enter the path or folder name on the remote server that you want to create. Click the **{V}** button to insert a variable that holds the name of the folder.

**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.

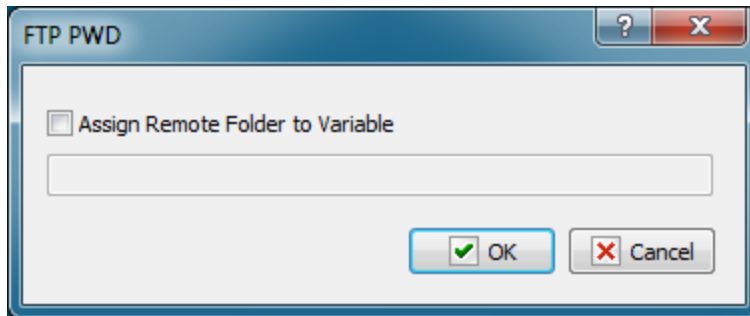
### Variable

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.

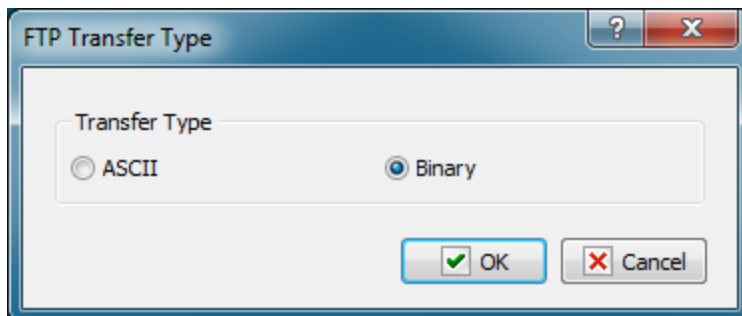


### Assign Remote Folder to Variable

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**.



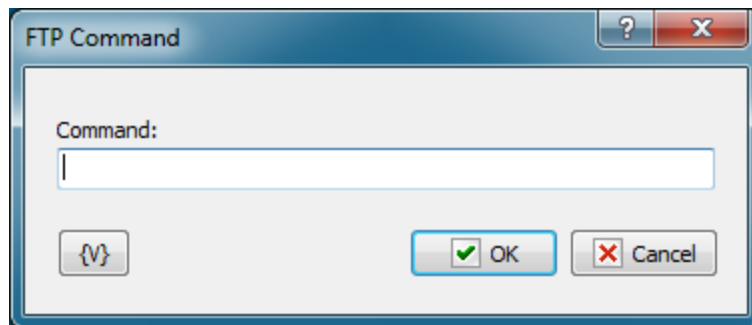


### Transfer Type

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.

#### General tab

FTP File Loop

General | Date Filter | Parameters of Sorting

Loop Kind

By Files Only  By Folders Only

Remote Folder and File Mask on each line (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 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**

Enter the name of the variable into which it is necessary to save the file name

**Assign File Size to Variable**

Switch on this checkbox and enter the name of the variable into which it is necessary to save the size of the file. This allows to use the file size data in other actions or tasks.

### Assign Last Update Time to Variable

Switch on this checkbox and enter the name of the variable into which it is necessary to save the date and the time of the file. This allows to use the last update time information in other tasks or actions.

### Date Filter tab

The screenshot shows the 'FTP File Loop' dialog box with the 'Date Filter' tab selected. The dialog has three tabs: 'General', 'Date Filter', and 'Parameters of Sorting'. Under 'Date Filter', there are two radio button options: 'All Files' (selected) and 'Find Files Created or Modified:'. The 'Find Files Created or Modified:' section contains a 'between' option with two date input fields, both containing '26.09.2022'. Below this are eight radio button options for time-based filters: 'newer than' and 'older than' for 'month(s)', 'day(s)', 'hour(s)', and 'minute(s)'. Each option has a numeric input field set to '1' and a dropdown arrow. At the bottom of the dialog, there is a '{V}' button, an 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

#### All Files

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

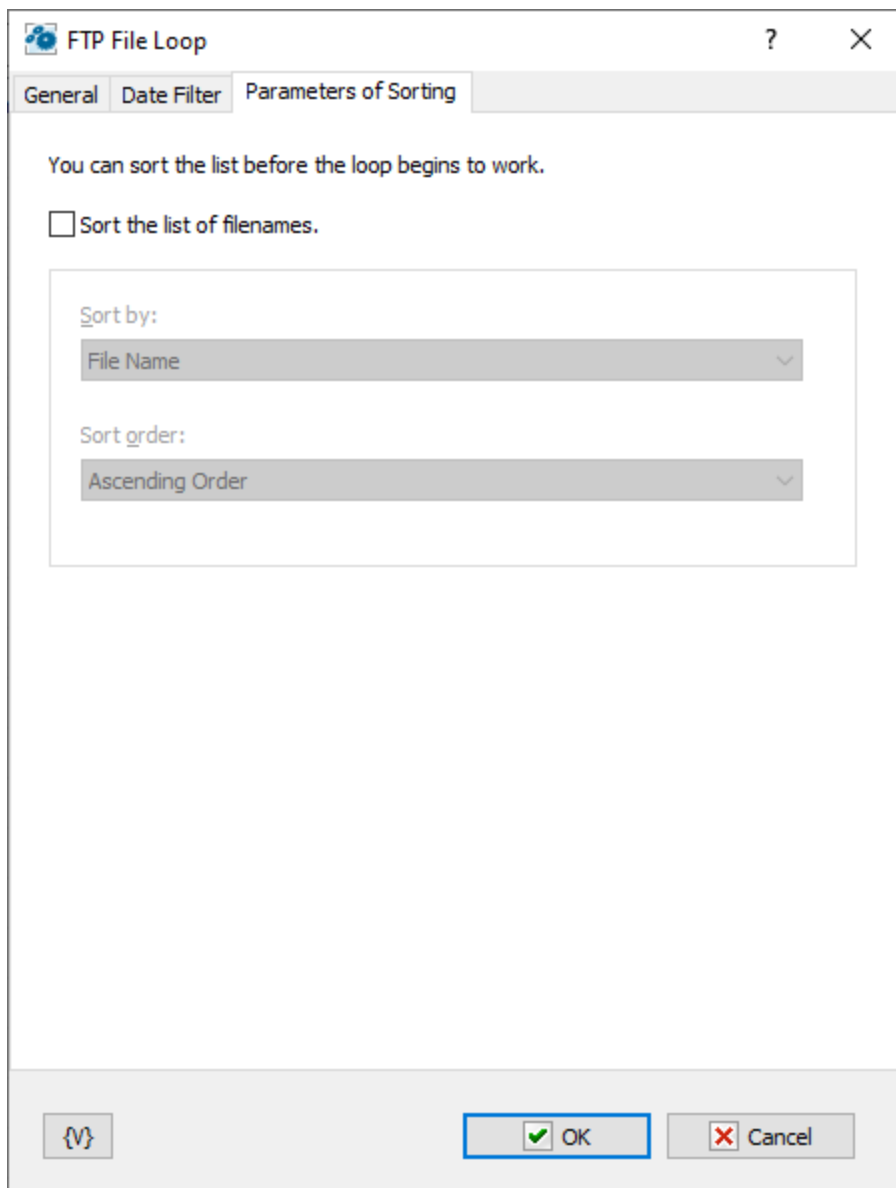
#### Find Files Created or Modified

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

- **Between** – You can specify two dates explicitly selecting them from a calendar. Or you can specify the expression using variables. This expression should contain the date in the format set on your computer.
- **Newer Than \_\_\_ Months**
- **Newer Than \_\_\_ Days**
- **Newer Than \_\_\_ Minutes**
- **Older Than \_\_\_ Months**
- **Older Than \_\_\_ Days**
- **Older Than \_\_\_ Minutes**

## Sorting parameters

Parameters of internal sorting of the file list



### Sort the list of filenames

Turn on the checkbox if you want to sort the list

### How to sort

There are four sorting modes:

- By file name
- By file time
- By file size
- By depth of nested folders

### Sort order

There are two modes: ascending or descending ordering

### Variable

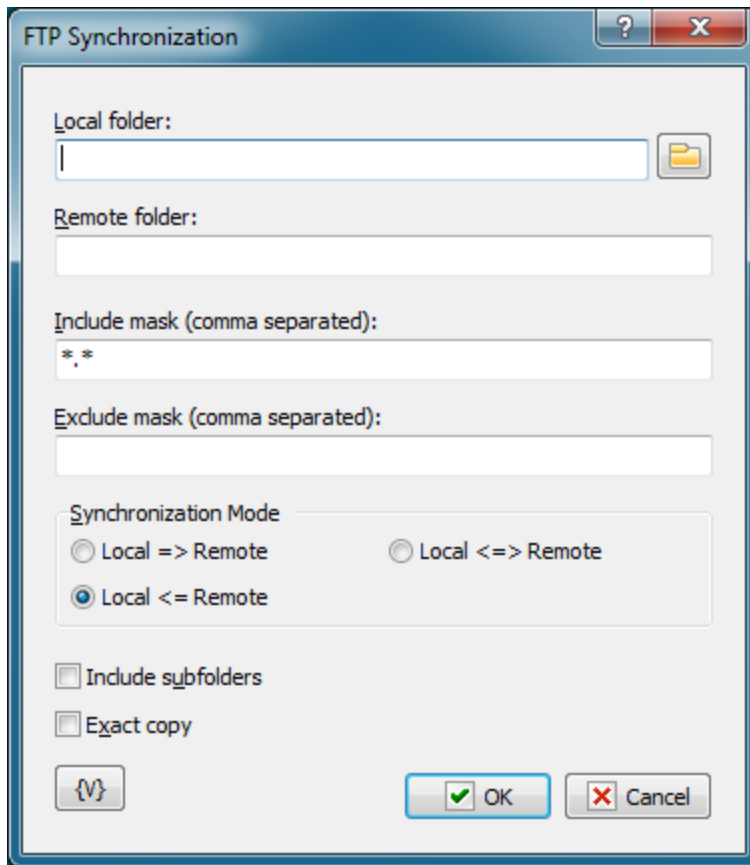
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 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

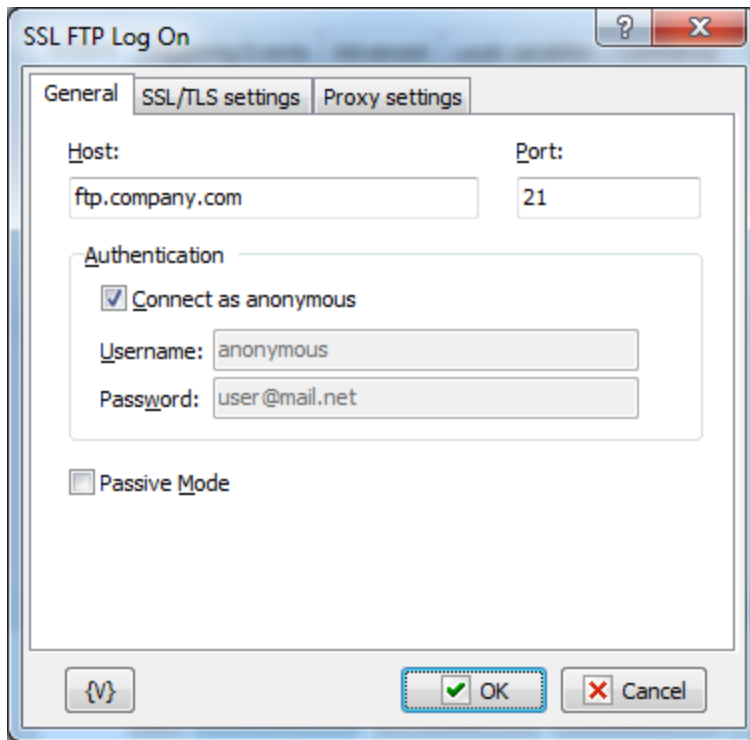
Encryption of traffic lets using of the FTP server in a public network safely (SSL - Secure Sockets Layer). At present most types of servers support the SSL FTP. Also there are freeware servers supporting SSL FTP (for example, FileZilla Server). Plugin SSL FTP allows using an ordinary FTP protocol with 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.

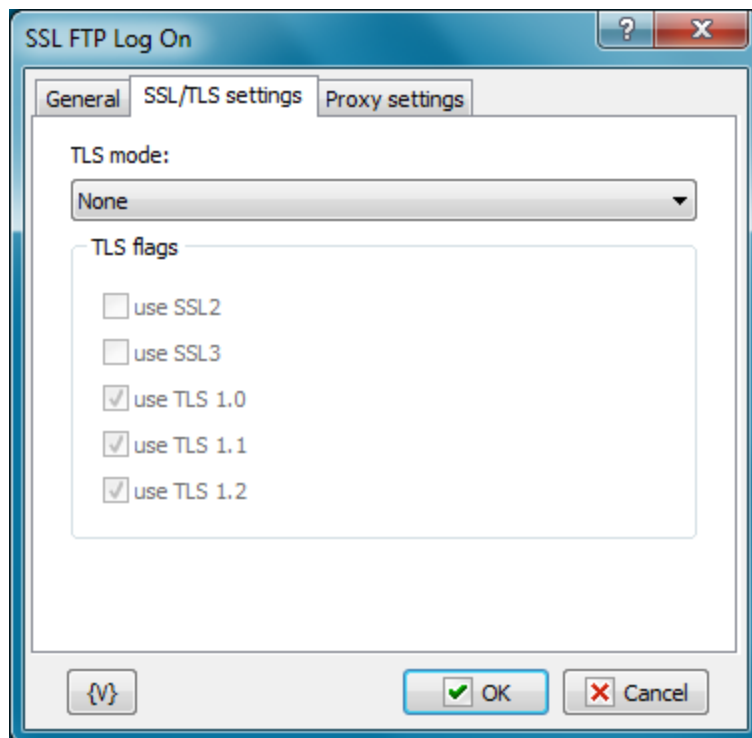
**Password**

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.

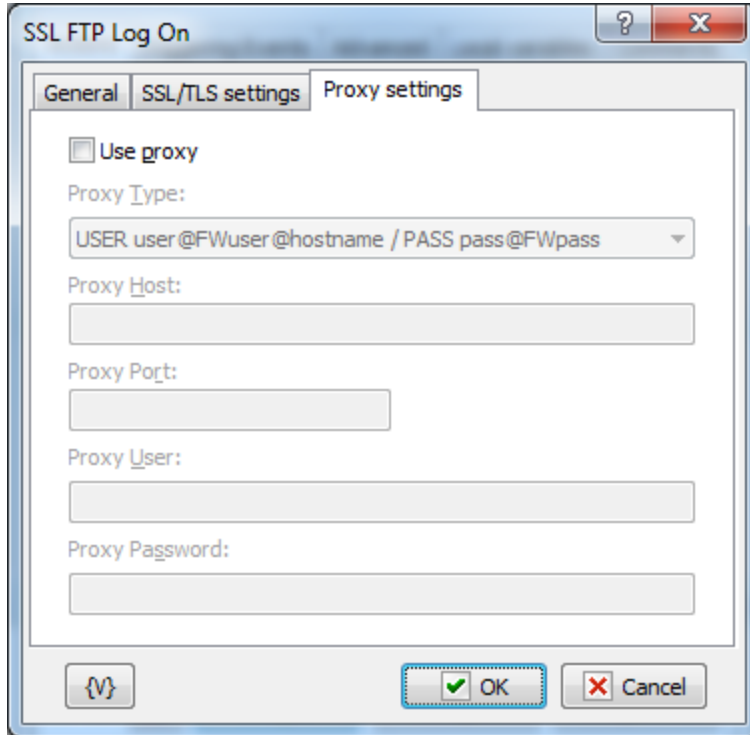
### TLS Flags

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

- Use SSL2
- Use SSL3
- Use TLS 1.0
- Use TLS 1.1

- Use TLS 1.2

### 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.

**Proxy Password**

The password for the authorization in PROXY.

**Variable**

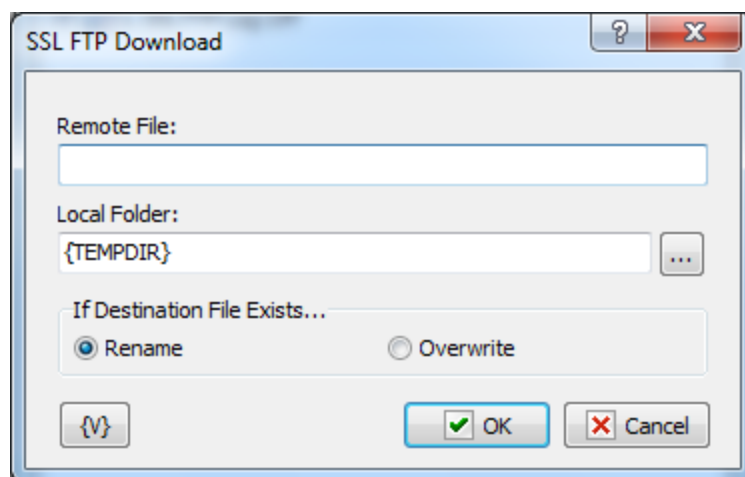
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 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**

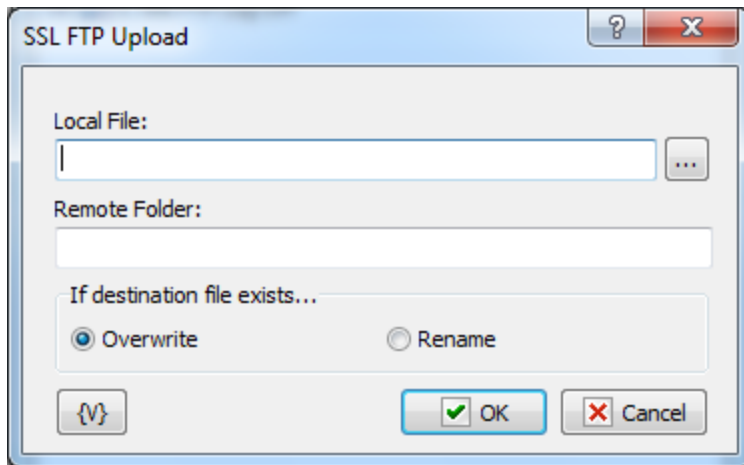
Enter the path and name of the local directory to which you want to copy the remote file.

**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**

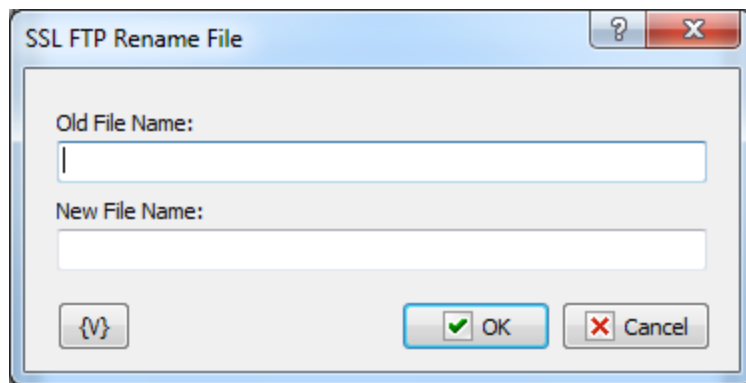
Enter the path and name of the directory on the remote server.

**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**

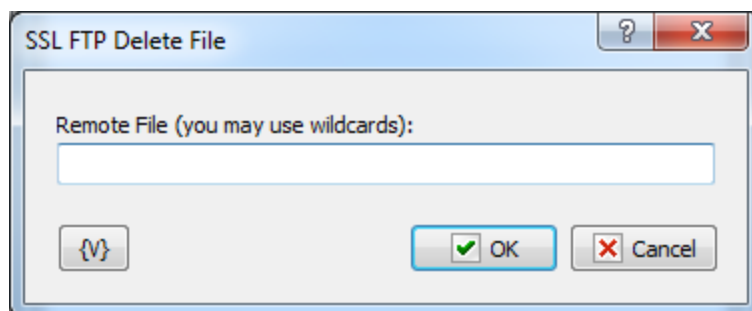
Enter the name that you want to give the file.

**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**

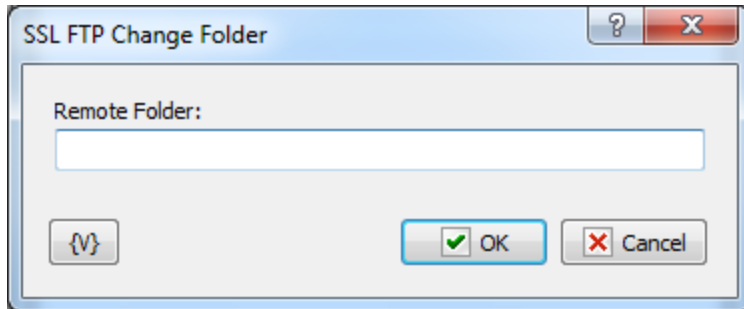
Enter the path and file name that you want to delete on the remote server. Click the **{V}** button to insert a variable that holds the path and name of the file.

**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

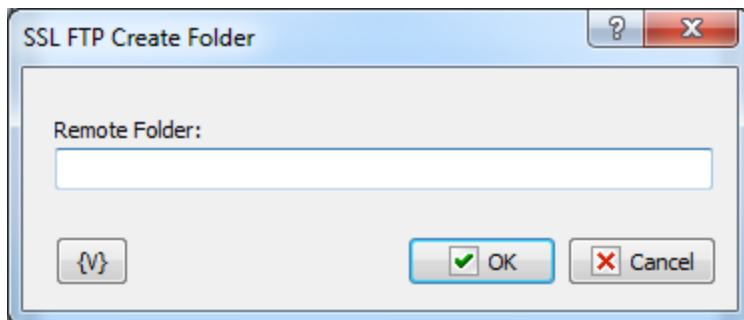
Enter the path or folder name on the remote server that you want to switch to. Click the **{V}** button to insert a variable that holds the name of the folder.

#### 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

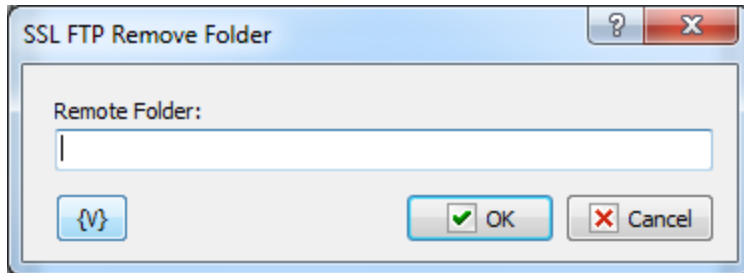
Enter the path or folder name on the remote server that you want to create. Click the **{V}** button to insert a variable that holds the name of the folder.

#### 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.

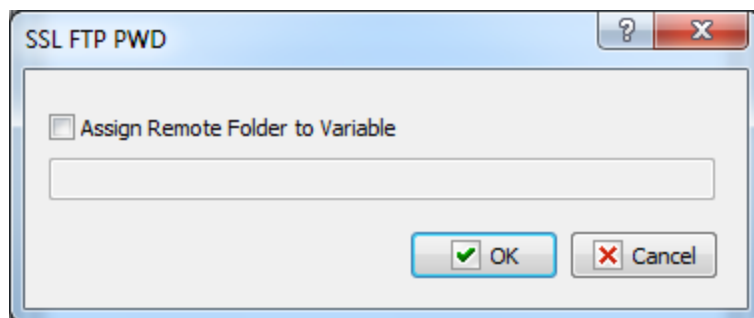
#### Variable

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.

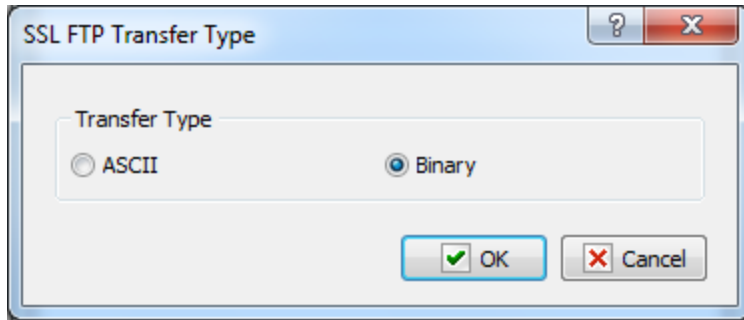


#### Assign Remote Folder to Variable

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.

### General tab



The screenshot shows the 'SSL FTP File Loop' dialog box with the 'General' tab selected. The 'Loop Kind' section has two radio buttons: 'By Files Only' (selected) and 'By Folders Only'. Below this is a text box for 'Remote Folder and File Mask on each folder (e.g. download/\*.zip):'. There is an unchecked checkbox for 'Include Subfolders'. Below that is a text box for 'Exclude Mask (comma separated):'. The 'Save Results' section contains a text box for 'Assign File Name to Variable:', an unchecked checkbox for 'Without Full Path', and two unchecked checkboxes: 'Assign File Size to Variable' and 'Assign Update Time to Variable'. At the bottom, there is a help button with a question mark icon, an 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

**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**

Enter the name of the variable into which it is necessary to save the file name.

**Assign File Size to Variable**

Switch on this checkbox and enter the name of the variable into which it is necessary to save the size of the file. This allows to use the file size data in other actions or tasks.

### Assign Last Update Time to Variable

Switch on this checkbox and enter the name of the variable into which it is necessary to save the date and the time of the file. This allows to use the last update time information in other tasks or actions.

## Date Filter tab

The screenshot shows the 'Date Filter' tab of the 'SSL FTP File Loop' dialog box. The 'All Files' radio button is selected. Below it, the 'Find Files Created or Modified:' section is active, with the 'between' radio button selected. Two date input fields both contain '30.12.1899'. Below these are eight radio button options for filtering by age: 'newer than' and 'older than' for 'month(s)', 'day(s)', 'hour(s)', and 'minute(s)'. Each option has a numeric input field set to '1'. At the bottom, there are buttons for '{V}', 'OK', and 'Cancel'.

### All Files

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

---

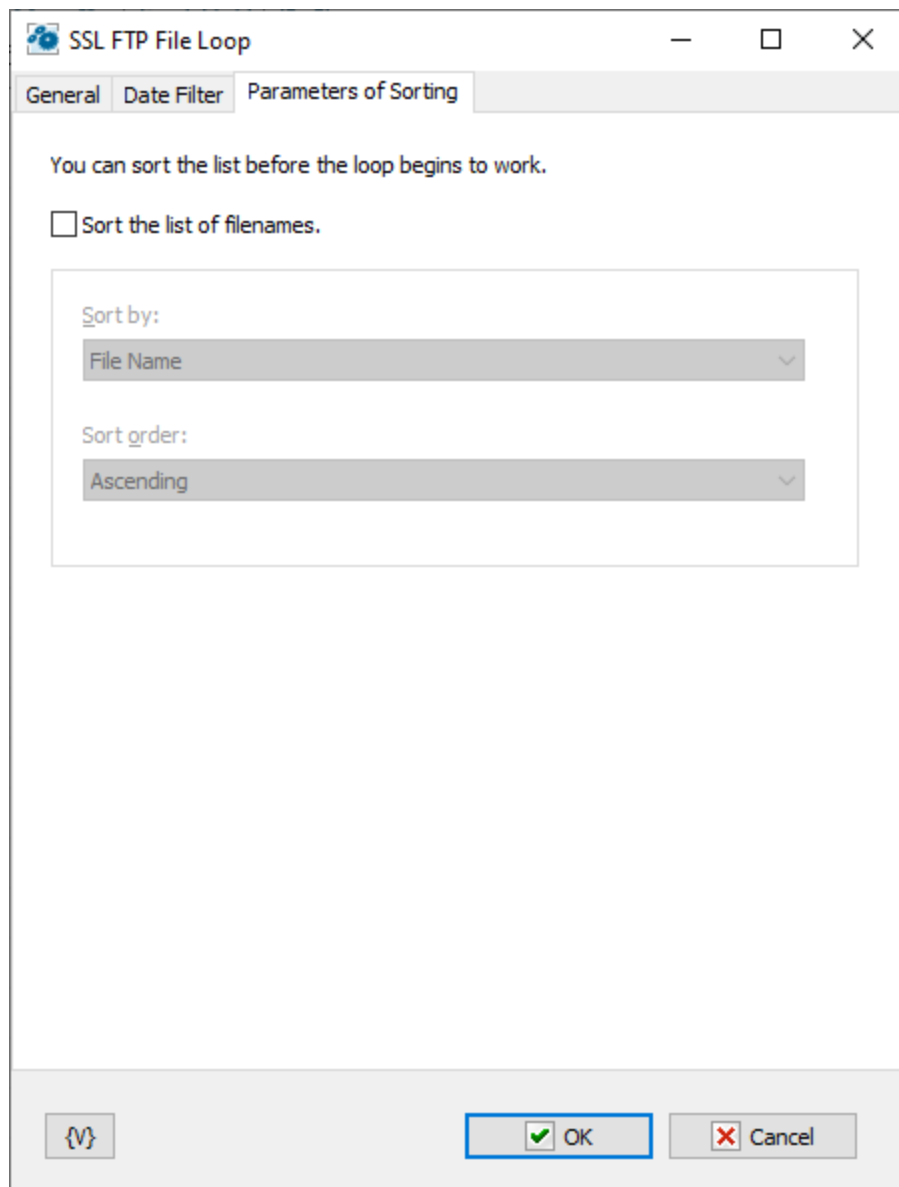
### Find Files Created or Modified

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

- **Between** – You can specify two dates explicitly selecting them from a calendar. Or you can specify the expression using variables. This expression should contain the date in the format set on your computer.
- **Newer Than \_\_\_ Months**
- **Newer Than \_\_\_ Days**
- **Newer Than \_\_\_ Minutes**
- **Older Than \_\_\_ Months**
- **Older Than \_\_\_ Days**
- **Older Than \_\_\_ Minutes**

### Sorting parameters

Parameters of internal sorting of the file list



### Sort the list of filenames

Turn on the checkbox if you want to sort the list

### How to sort

There are four sorting modes:

- By file name
- By file time
- By file size
- By depth of nested folders

### Sort order

There are two modes: ascending or descending ordering

### Variable

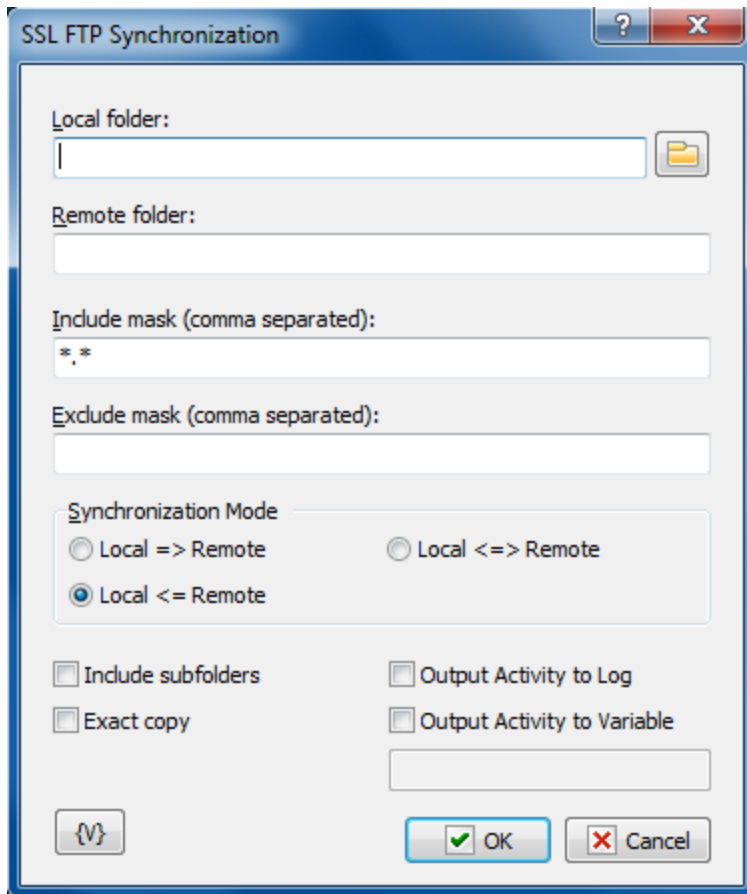
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

#### 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.***

**Output Activity to Log**

If you want to write the log of synchronization into the task log, switch on this checkbox. The action will put the information about downloaded, uploaded and removed files into the task log.

**Output Activity to variable**

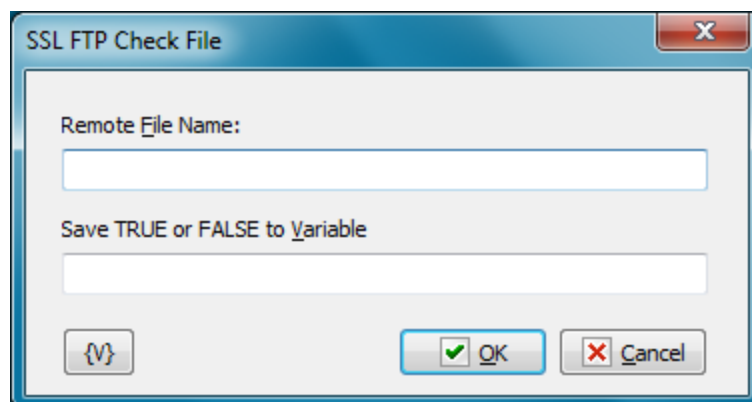
If you want to write the log of synchronization into a variable, switch on this checkbox and put the variable name in the field. The action will put the information about downloaded, uploaded and removed files into the variable as text.

**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.

#### 4.7.14 SSL FTP Check File

The action checks the existence of remote file and save boolean value TRUE or FALSE to specified variable.



### Remote File Name

Specify remote file name to check. You can specify file path from root folder or from current folder.

### Save TRUE or FALSE to Variable

Specify variable name where you want to save the result. *TRUE* - if file exists, *FALSE* - if not.

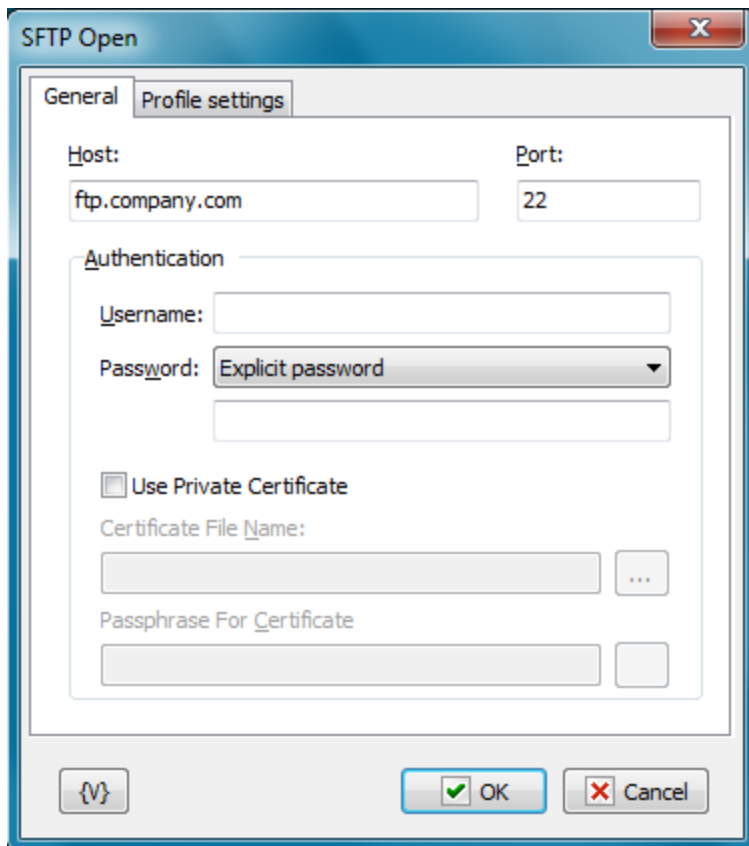
## 4.8 SFTP (SSH FTP)

Plugin unites functions SFTP (according to the protocol SSH). SSH is a very popular protocol on Linux servers. In spite of similarity with FTP protocol, SFTP is an absolutely different protocol of file exchange and has only similarity in appearance with FTP.

The work of the plugin is supported by the console utility PSFTP.EXE from the PUTTY package (free open source software).

### 4.8.1 SFTP Log On

Connecting to the SFTP server. This action have to be done to provide normal work of the rest of SFTP actions.



#### Host

The name or IP-address of the SFTP server. The usage of variables is allowed.



**Port**

The number of a server port. It is 22 by default. The usage of variables is allowed.

**User Name**

The name of a user. The usage of variables is allowed.

**Password**

The password of a user. The usage of variables is allowed.

**Use private certificate**

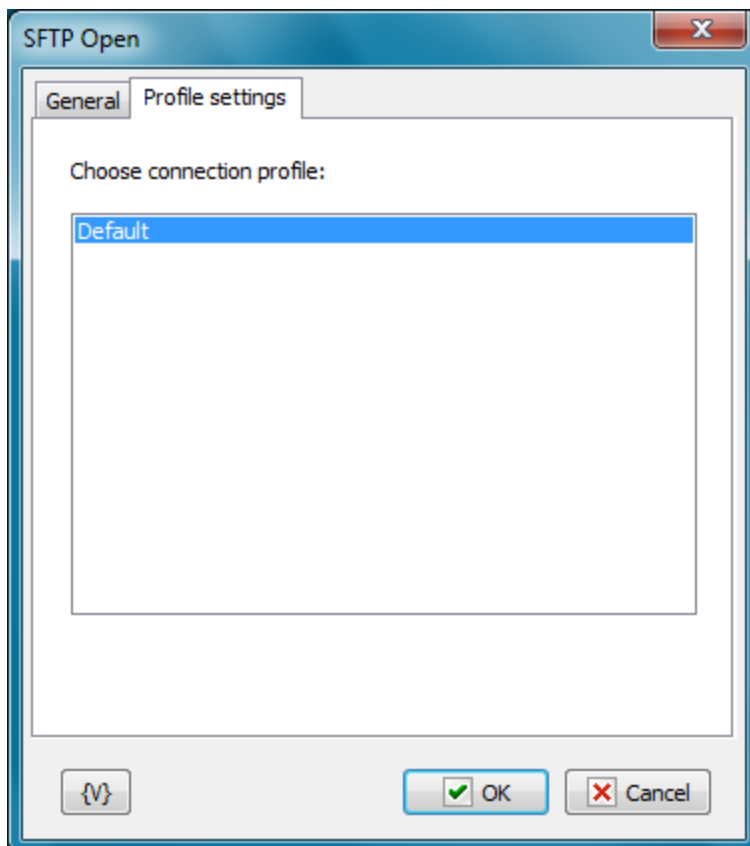
If you want to use the certificate for the authorization on the sFTP server, turn on this checkbox.

**Certificate private key**

Specify the file of the certificate. The certificate should be in the format which is used in the PUTTY. Default extension is PPK.

**Passphrase for certificate**

If the certificate is encoded, specify a password to unpack it. In other case leave this field empty.

**Choose connection profile**

Additional settings of a connection (for example, the usage of a proxy) can be done in a connection profile PUTTY. It is necessary to install PUTTY packet, to do necessary settings and then to save them into the separate profile. This profile can be used when connecting to the server.

### Variable

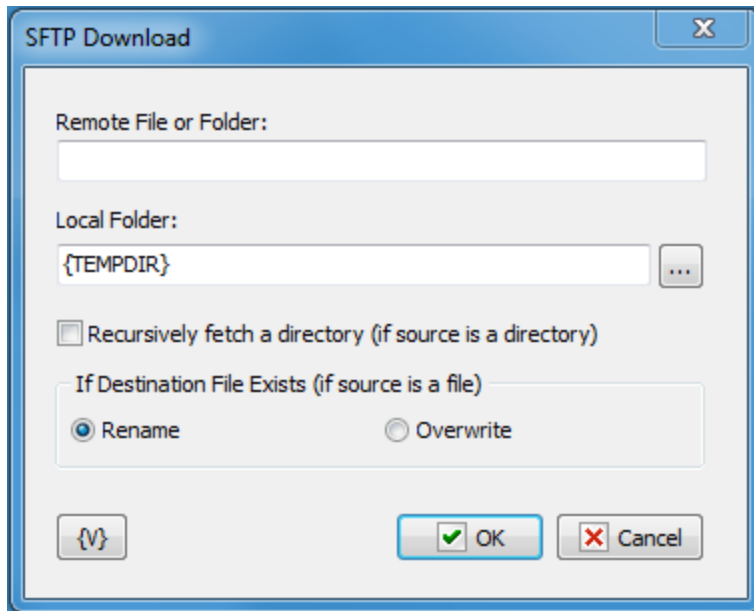
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.8.2 SFTP Log Off

This action completes an SFTP session and closes the connection with a server. The action does not required any additional settings.

## 4.8.3 SFTP Download

This action allows to download a file from a server to a local disc.



### Remote File or Folder

Specify the name of the file or the folder which is necessary to download to a local disc. If the folder is specified, the action downloads all files of the remote folder.

### Local folder

Specify the folder on a local disc or in a local network where is necessary to download a file or a folder from a server.

### Recursively fetch a directory (if a source is a directory)

Switch on this parameter if you want to download the whole folder with all subdirectories. This parameter makes sense only if a folder is specified as a source. If a source is a file, the parameter is ignored.

This mode is similar to [SFTP Synchronization](#) <sup>2011</sup> action.

### If a destination file exists (if a source is a file)

This parameter makes sense only if a file is specified as a source. There are 2 alternatives:

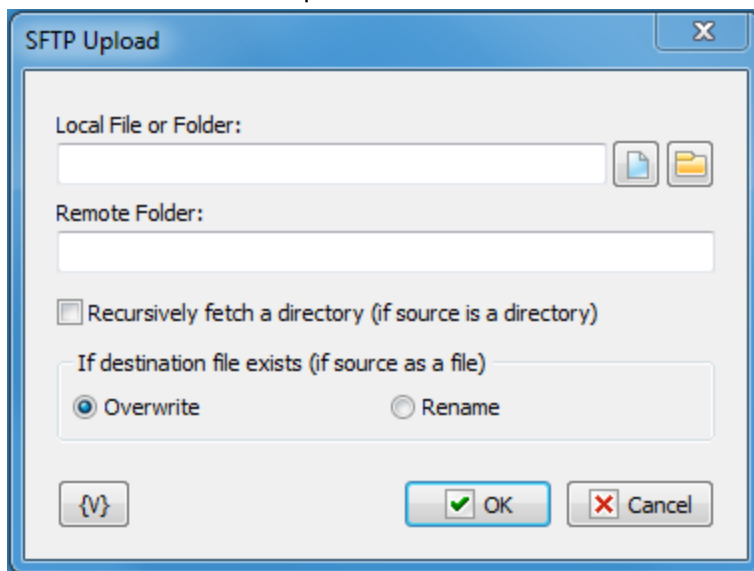
- **Rename** – if such file exists, it will be renamed when downloading. In this case, the existent file will be saved.
- **Overwrite** – a file will be downloaded with an original name. If such file exists, it will be overwritten.

### 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.

## 4.8.4 SFTP Upload

The action allows to upload files and folders from a local disc to the SFTP server.



### Local File or Folder

Specify the name of a file or folder which is necessary to upload from a local disc to a server. If a folder is specified, the action uploads all files from the local folder.

### Remote Folder

Specify a folder on SFTP server where it is necessary to upload the file or the folder from a local disc.

### Recursively fetch a directory (if a source is a directory)

Switch on this parameter if you want to upload the whole folder with all subdirectories. This parameter makes sense only if a folder is specified as a source. If a source is a file, the parameter is ignored.

This mode is similar to [SFTP Synchronization](#)<sup>[2011]</sup> action.

### If a destination file exists (if a source is a file)

This parameter makes sense only if a file is specified as a source. There are 2 alternatives:

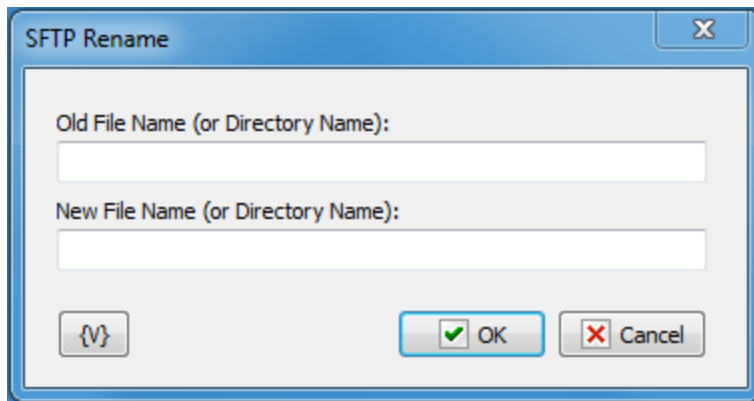
- **Rename** – if such file exists, it will be renamed when downloading. In this case, the existent file will be saved.
- **Overwrite** – a file will be downloaded with an original name. If such file exists, it will be overwritten.

### 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.

## 4.8.5 SFTP Rename File of Folder

The action allows to rename a file or a folder on the SFTP server.



### Old File Name (or Directory Name)

Specify the name of the file or the folder which are necessary to rename.

### New File Name (or Directory Name)

Specify a new name of the file or the folder.

### Usage:

It is possible to rename separate files/folders:

```
OldFile.txt -> NewFile.txt
```

```
OldFile.txt -> SubFolder/NewFile.txt
```

It is possible to move one or several files to another folder. For this it is necessary to specify the file names separated by a space and, in the capacity of a new file name, the name of the folder which the files have to be moved into.

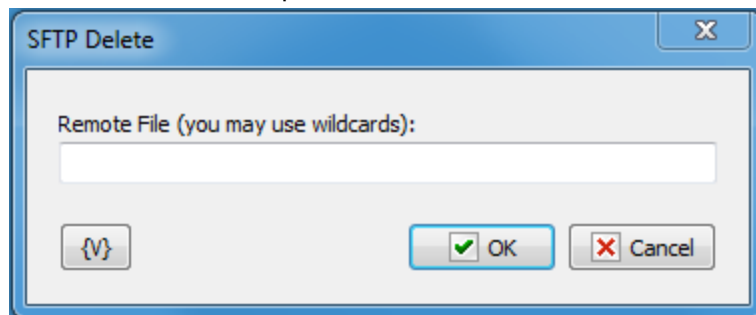
```
File1.txt File2.txt -> /home/user/newfolder
```

### 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.

## 4.8.6 SFTP Delete File

The action deletes specified files from the SFTP server.



### Remote file

Specify the name of the file that has to be deleted. The usage of a mask is allowed. It is also possible to specify several files separated by a space.

For example:

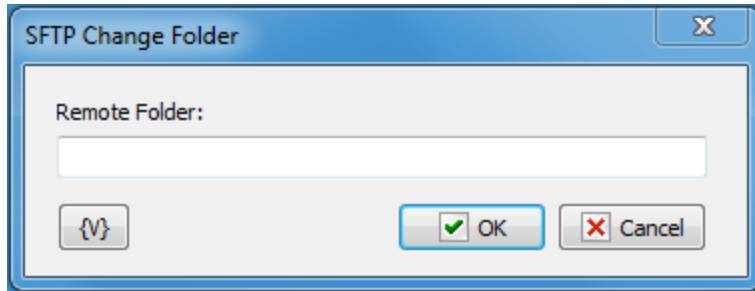
```
file.dat  
file1.txt file2.txt  
*.txt
```

### 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.

### 4.8.7 SFTP Change Folder

The action changes the current work folder on the server.



#### Remote folder

Specify a new work folder. It is possible to specify the folder from the root or current folder. For example:

```
/home/user/test
```

```
test/subfolder
```

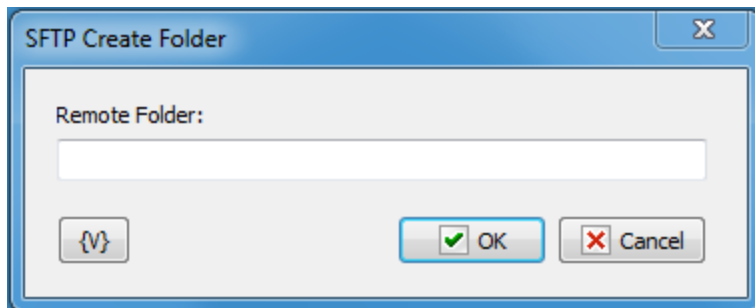
It is possible to get the current folder during (as a result of) the action [SFTP PWD](#)<sup>196</sup>.

#### 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.

### 4.8.8 SFTP Create Folder

The action creates a new folder.



#### Remote Folder

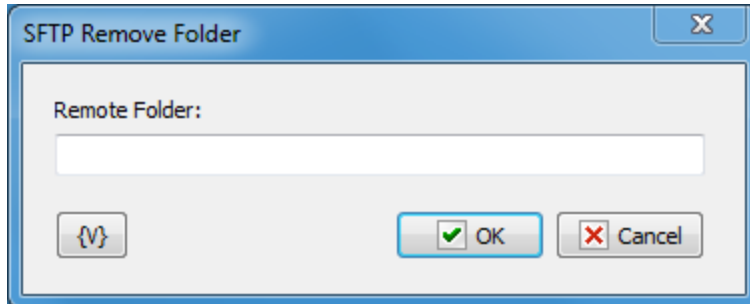
Specify new folder name.

#### 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.

### 4.8.9 SFTP Remove Folder

Allows you to delete a folder on a remote server.



#### Remote Folder

Enter the path or folder name on the remote server that you want to remove.

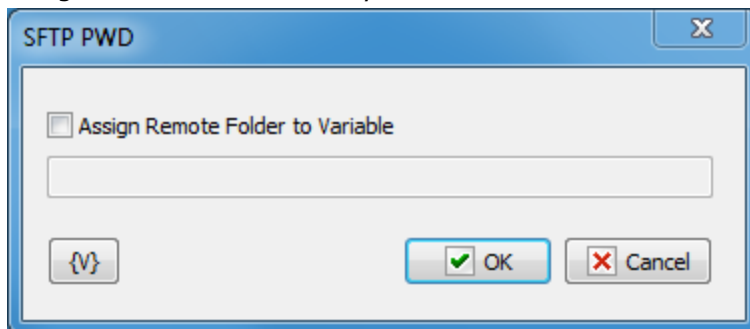
#### Variable

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.

### 4.8.10 SFTP PWD (print working directory)

Allows you to print the current directory on the remote computer.

This action saves the task log to the current directory on the SFTP 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.

#### 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.

### 4.8.11 SFTP File Loop

Allows a task to iterate through the list of files or folders on SFTP server.

#### General tab

The screenshot shows the 'SFTP File Loop' dialog box with the 'General' tab selected. The dialog has three tabs: 'General', 'Date Filter', and 'Parameters of Sorting'. The 'General' tab contains the following elements:

- Loop Kind:** Two radio buttons, 'By Files Only' (selected) and 'By Folders Only'.
- Remote Folder and File Mask (e.g. download/\*.zip):** A text input field with a scroll bar.
- Include Subfolders:** A checkbox that is currently unchecked.
- Exclude Mask (comma separated):** A text input field.
- Save Results:** A section containing:
  - Assign File Name to Variable:** A text input field.
  - Without Full Path:** A checkbox that is currently unchecked.
  - Assign File Size to Variable:** A checkbox that is currently unchecked, with a corresponding text input field below it.
  - Assign Update Time to Variable:** A checkbox that is currently unchecked, with a corresponding text input field below it.

At the bottom of the dialog, there is a button with the {V} symbol, an 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

#### 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.



**Include Subfolders**

Switch on this checkbox if you want to perform the cycle through all the files, including all nested folders.

**Save Results**

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

**Assign File Name to Variable**

Enter the name of the variable into which it is necessary to save the file name.

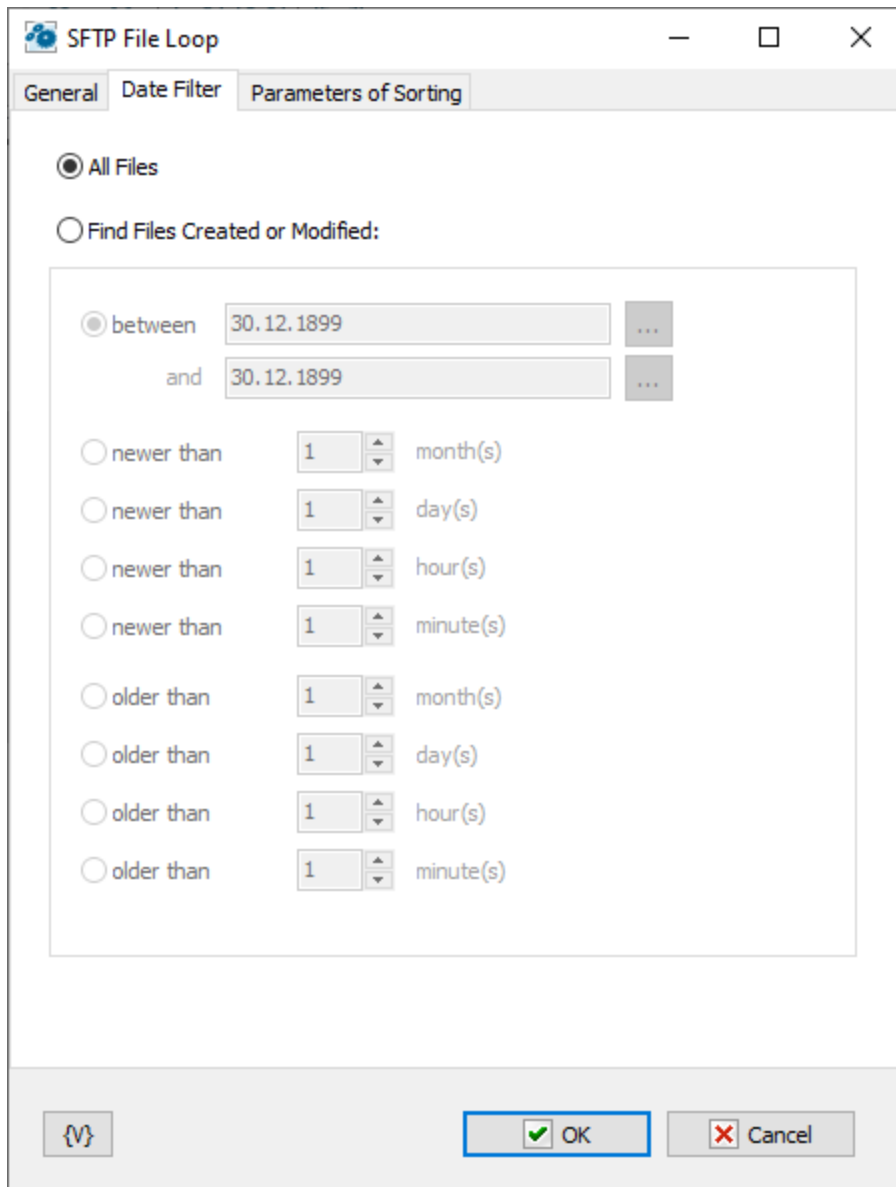
**Assign File Size to Variable**

Switch on this checkbox and enter the name of the variable into which it is necessary to save the size of the file. This allows to use the file size data in other actions or tasks.

**Assign Last Update Time to Variable**

Switch on this checkbox and enter the name of the variable into which it is necessary to save the date and the time of the file. This allows to use the last update time information in other tasks or actions.

**Date Filter tab**



### All Files

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

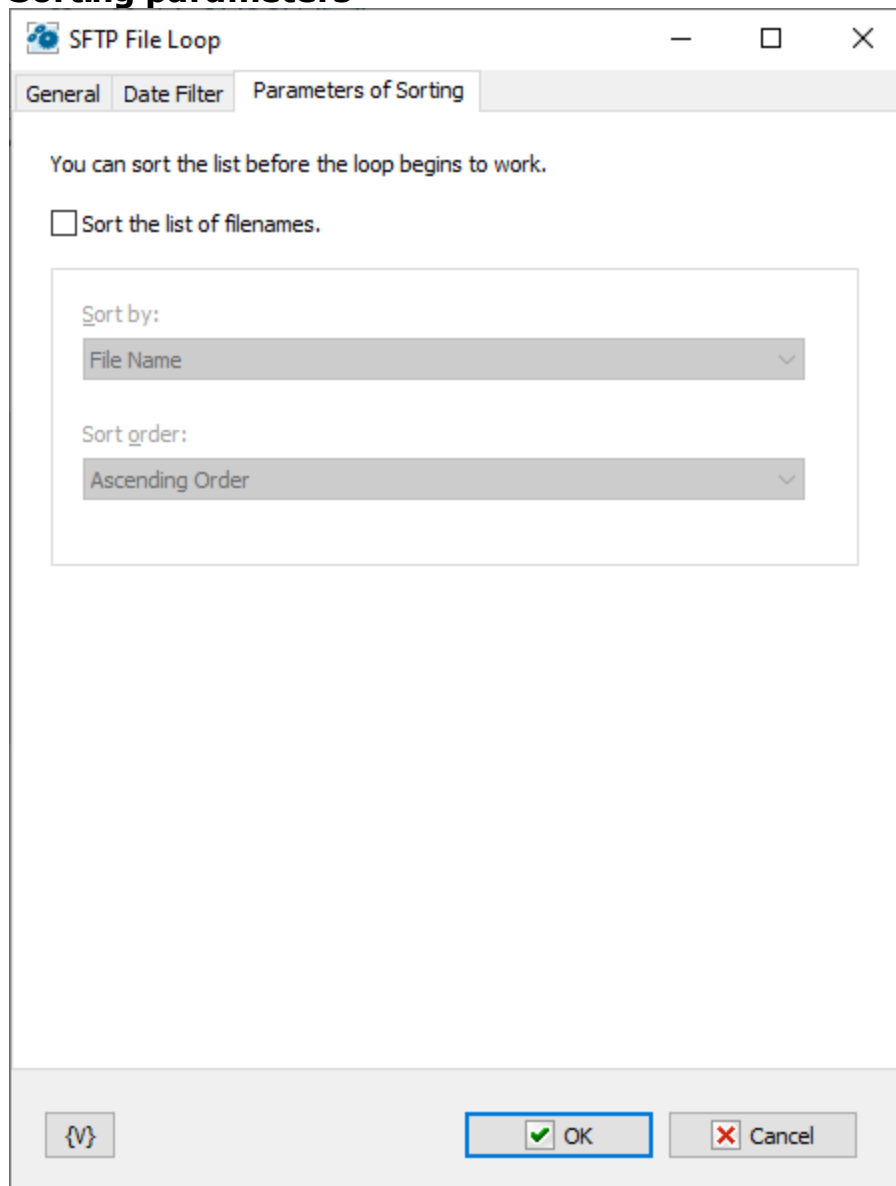
### Find Files Created or Modified

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

- **Between** – You can specify two dates explicitly selecting them from a calendar. Or you can specify the expression using variables. This expression should contain the date in the format set on your computer.
- **Newer Than \_\_\_ Months**
- **Newer Than \_\_\_ Days**
- **Newer Than \_\_\_ Minutes**
- **Older Than \_\_\_ Months**
- **Older Than \_\_\_ Days**

- Older Than \_\_\_\_ Minutes

## Sorting parameters



The screenshot shows a dialog box titled "SFTP File Loop" with three tabs: "General", "Date Filter", and "Parameters of Sorting". The "Parameters of Sorting" tab is active. The dialog contains the following elements:

- A message: "You can sort the list before the loop begins to work."
- A checkbox labeled "Sort the list of filenames." which is currently unchecked.
- A "Sort by:" dropdown menu with "File Name" selected.
- A "Sort order:" dropdown menu with "Ascending Order" selected.
- At the bottom, there are three buttons: a help button with a curly brace icon, an "OK" button with a green checkmark, and a "Cancel" button with a red X.

### Sorting parameters

Parameters of internal sorting of the file list

### Sort the list of filenames

Turn on the checkbox if you want to sort the list

### How to sort

There are four sorting modes:

- By file name
- By file time
- By file size
- By depth of nested folders

**Sort order**

There are two modes: ascending or descending ordering.

**Variable**

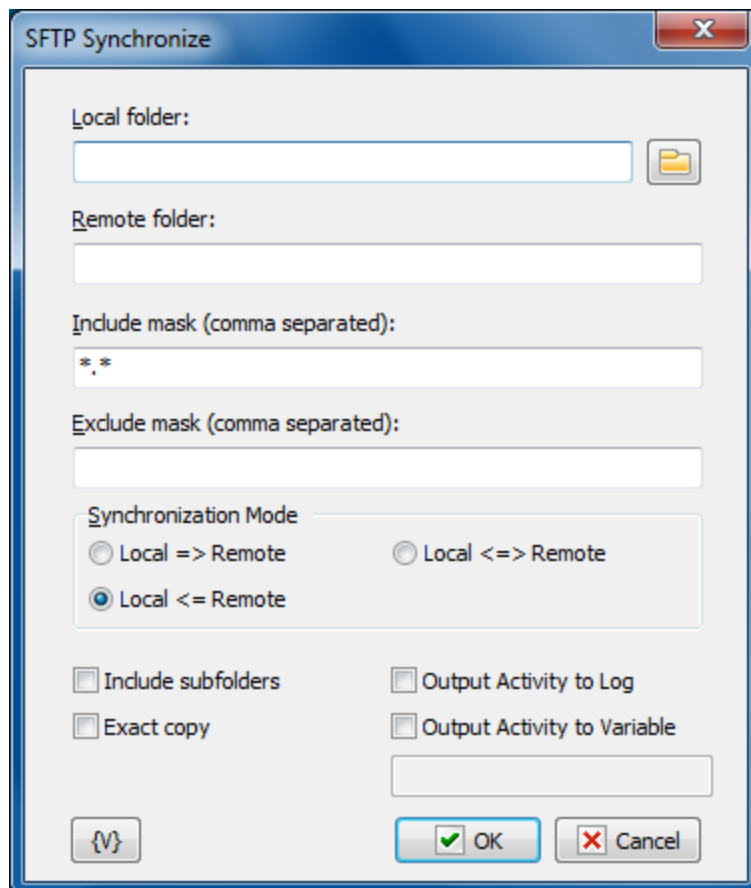
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

**4.8.12 SFTP Synchronization**

Synchronizes files between a local folder and a remote folder on the SFTP 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.
- 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 SFTP server, but still this operation will free you from creating 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.*

**Output Activity to Log**

If to switch on this parameter, the action will display the information about operations with files (download, upload, delete files) during the work.

**Output Activity to variable**

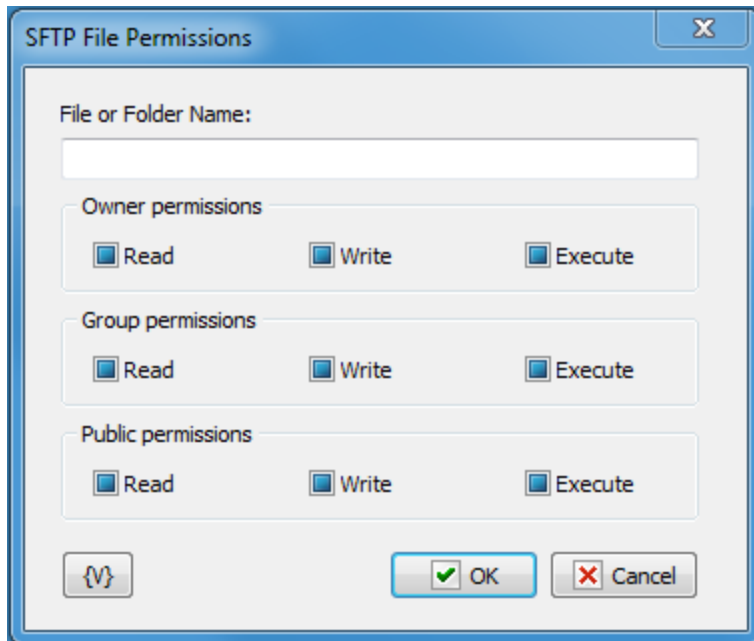
If you want to write the log of synchronization into a variable, switch on this checkbox and put the variable name in the field. The action will put the information about downloaded, uploaded and removed files into the variable as text.

**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.

### 4.8.13 SFTP File Permissions

The action allows you to modify the file permissions on files and directories on the server.



The file permissions on the SFTP server are assigned like in the UNIX systems. Every file or folder has 3 groups of permissions:

- **Owner permissions;**
- **Group permissions.** That is, the permissions of users of the group, which the file owner belong to.
- **Public permissions.** That is, the permissions for all the rest users.

Every group contains 3 types of permissions:

- **Read** - permission to read the file
- **Write** - permission to write to the file
- **Execute** - permission to execute the file

Please, notice that Checkboxes have three positions:

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

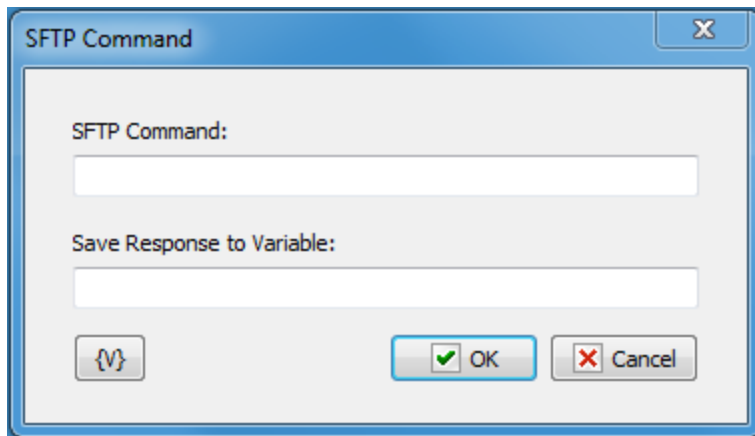
By default, all the checkboxes are in Grayed state.

#### 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.

### 4.8.14 SFTP Command

The action allows to perform a arbitrary SFTP command and save the result of performing into a variable.



### SFTP command

Specify any SFTP command. For example, the command *help*.

### Save Response to Variable

Specify the name of the variable, into which it is necessary to save the result of command performing as a text.

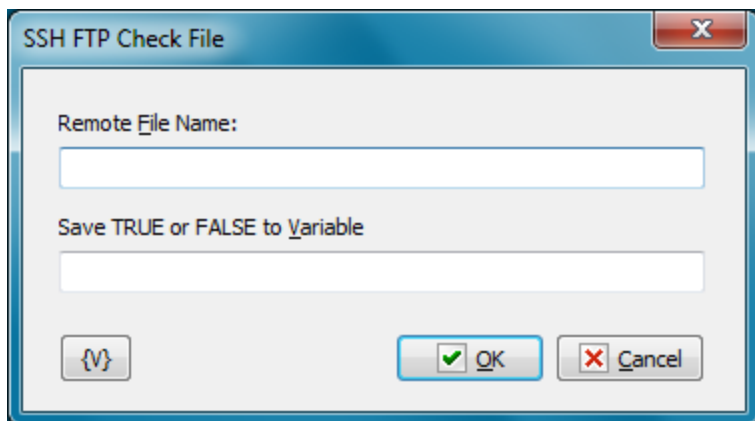
Variable name can be empty. In this case the text of the result will be ignored.

### 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.

## 4.8.15 SFTP File Check

The action checks the existence of remote file and save boolean value TRUE or FALSE to specified variable.



### Remote File Name



---

Specify remote file name to check. You can specify file path from root folder or from current folder.

### **Save TRUE or FALSE to Variable**

Specify variable name where you want to save the result. *TRUE* - if file exists, *FALSE* - if not.

## **4.9 Dial-Up**

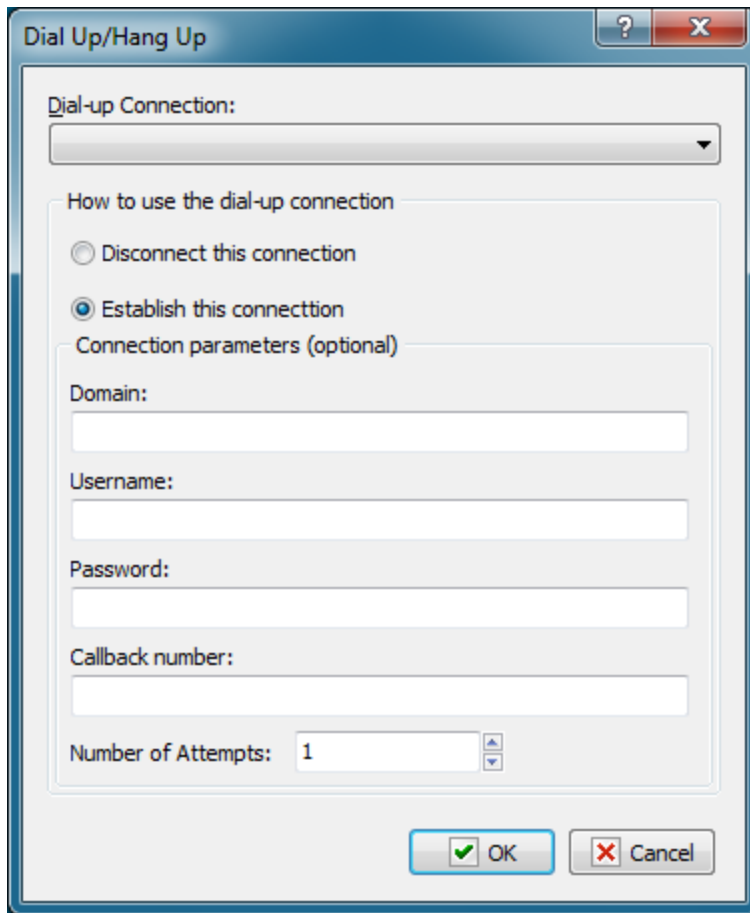
The actions Dial-Up can be used for such Internet connections as these:

- Connection by the use of an usual telephone modem;
- VPN of connections;
- Connections to Internet through mobile net by means of a mobile modem.

These connections use the same system interface.

### **4.9.1 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.

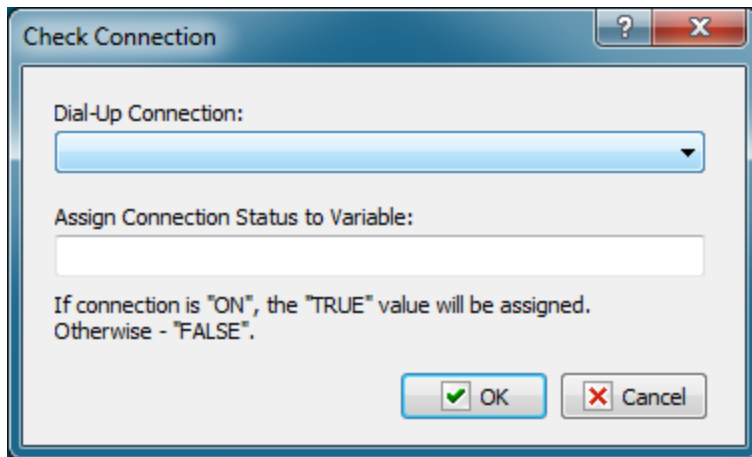
### Connection Parameters

Enter the following optional parameters to establish a connection:

- **Domain**
- **Username**
- **Password**
- **Callback Number**
- **Number of Attempts**

## 4.9.2 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.

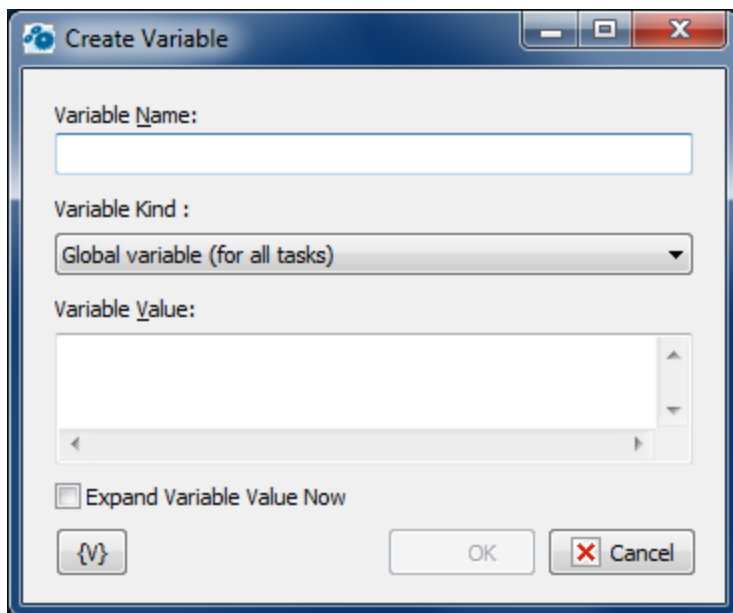
**Assign Connection Status to Variable**

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

## 4.10 Variables

### 4.10.1 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. The action allows to create both a global variable and a local one within a task. Local variables can also be generated by means of creating a list of local variables (see [Local variables tab](#)<sup>[56]</sup> of task editor).



**Variable Name**

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

**Variable Kind**

- **Global variable (for all tasks)** – in this case a global variable will be created. If the variable already exists, a specified value will be assigned to it.
- **Local variable (for this task only)** – in this case a local variable will be created within a task. The local variable will be deleted from the memory automatically when the task is finished. See the chapter [Local Variables](#)<sup>[12]</sup> for details about local variables.

**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.

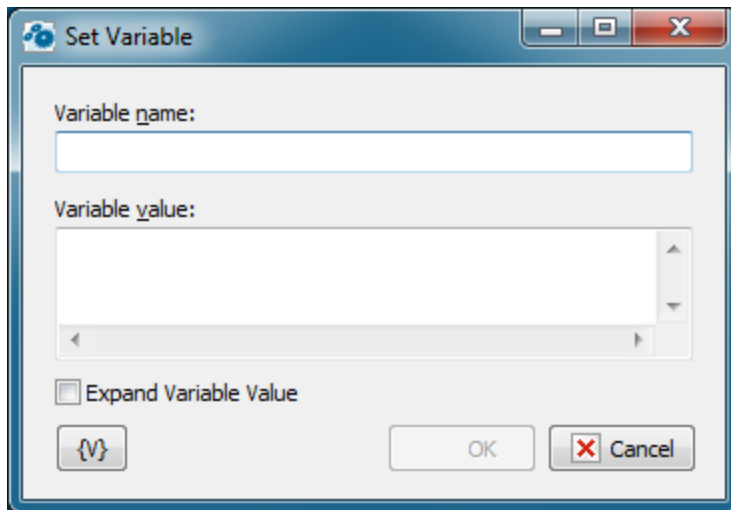
There can be necessity of using both modes in tasks:

- **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.

## 4.10.2 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.

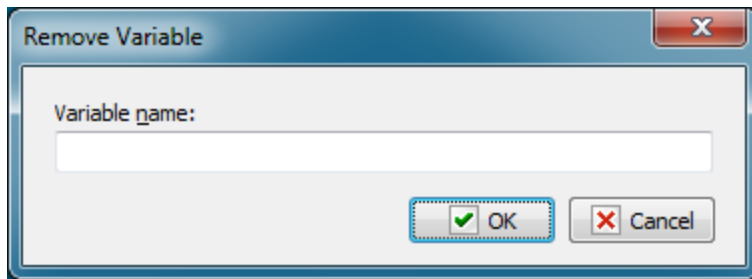
There can be necessity of using both modes in tasks:

- **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.

### 4.10.3 Remove Variable

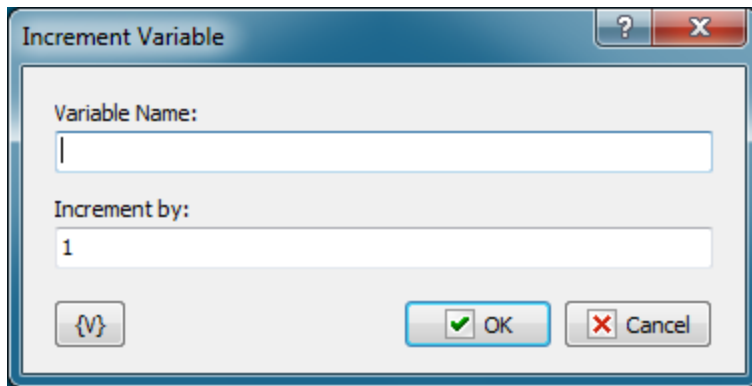
Deletes a variable when you run a task. This ensures that all variables are cleared from memory if a new task starts.

**Variable Name**

Enter the name of the variable you want to delete.

#### 4.10.4 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**

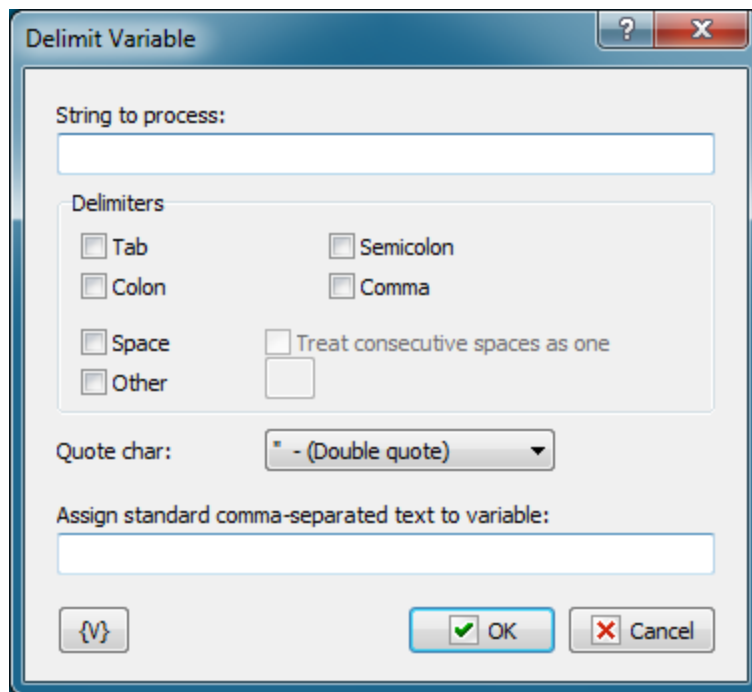
Enter a positive or negative value

**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.

#### 4.10.5 Split text

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>11</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.

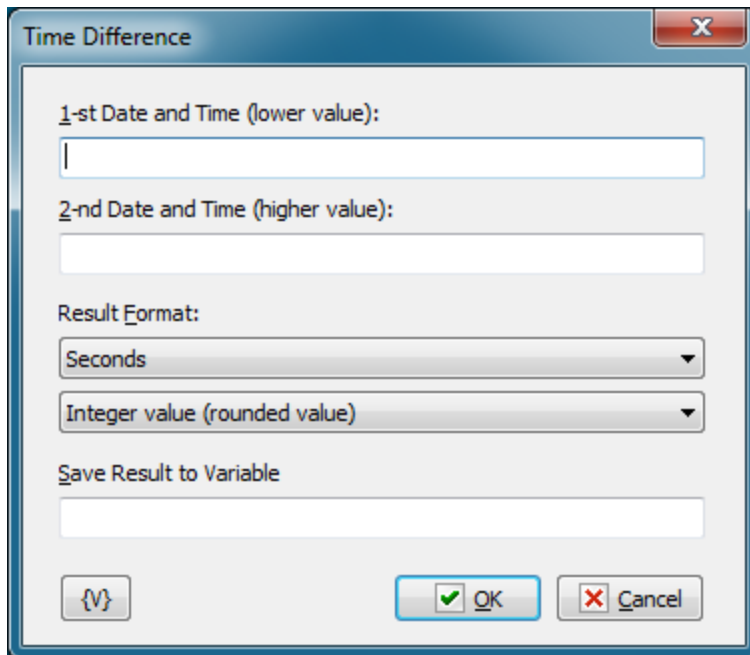
**Variable**

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.

#### 4.10.6 Time Difference

The action is intended to calculate the difference between two values of time quickly. The result can be expressed in days, hours, minutes or seconds.

The value also can be obtained in the compound format: **dd:hh:mm:ss**



### 1st Date and Time (lower value)

The lower value of time. Date and time must be specified in the current system date-time format.

### 2nd Date and Time (higher value)

The higher value of time. Date and time must be specified in the current system date-time format.

**Note:** *If the first value is bigger, the result will be negative.*

### Result Format

You can select in which units to get the result:

- Seconds
- Minutes
- Hours
- Days
- Compound format *dd:hh:mm:ss*

Additionally, you can specify the accuracy of the value:

- Integer value (rounded)
- Real value (floating point value)

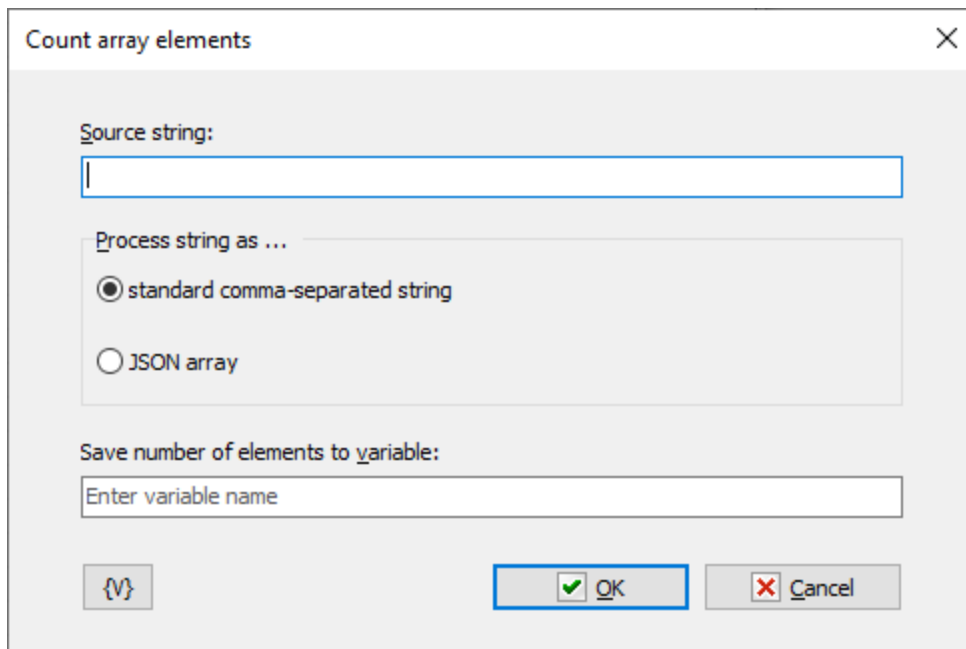
### Save result to Variable

Specify the name of the variable, in which the result must be saved.



### 4.10.7 Count array elements

This action is designed to count elements in comma-separated string or JSON array. This is convenient for looping over elements if the number of elements is not known in advance.



Count array elements

Source string:

Process string as ...

standard comma-separated string

JSON array

Save number of elements to variable:

Enter variable name

{V} OK Cancel

#### Source string

The string in which to count the number of elements. Available formats:

- Standard comma-separated string
- JSON array

#### Process string as...

Specify the processing mode for the source string.

#### Save number of elements to variable

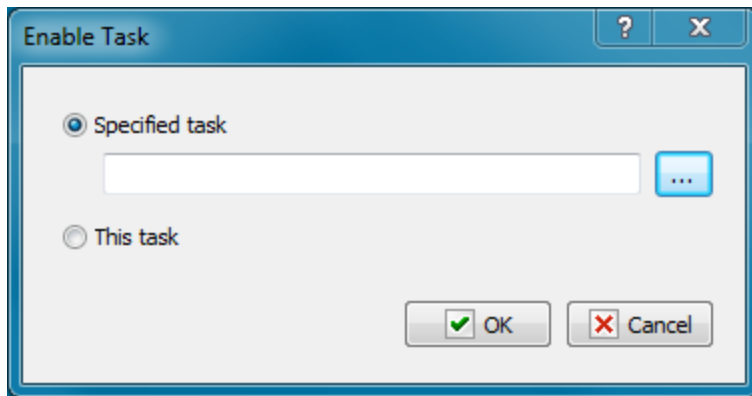
Enter the variable name in which to store the number of elements.

## 4.11 Task Commands

Actions for internal tasks

### 4.11.1 Enable Task

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

**Specified Task**

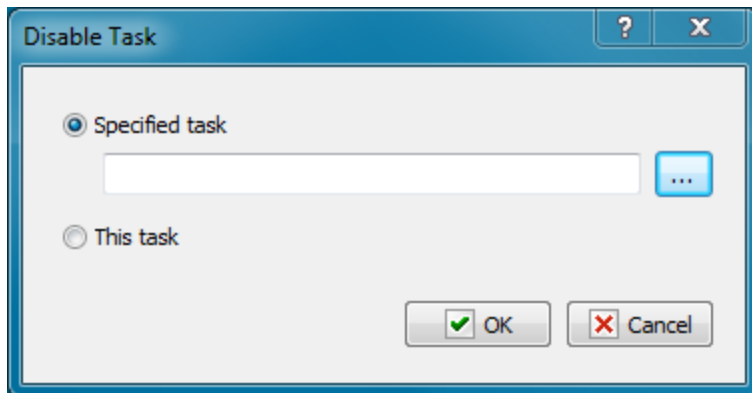
Press Select button and choose the task from the list.

**This task**

If you want to enable this task without selection choose this option.

#### 4.11.2 Disable Task

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

**Specified Task**

Press Select button and choose the task from the list.

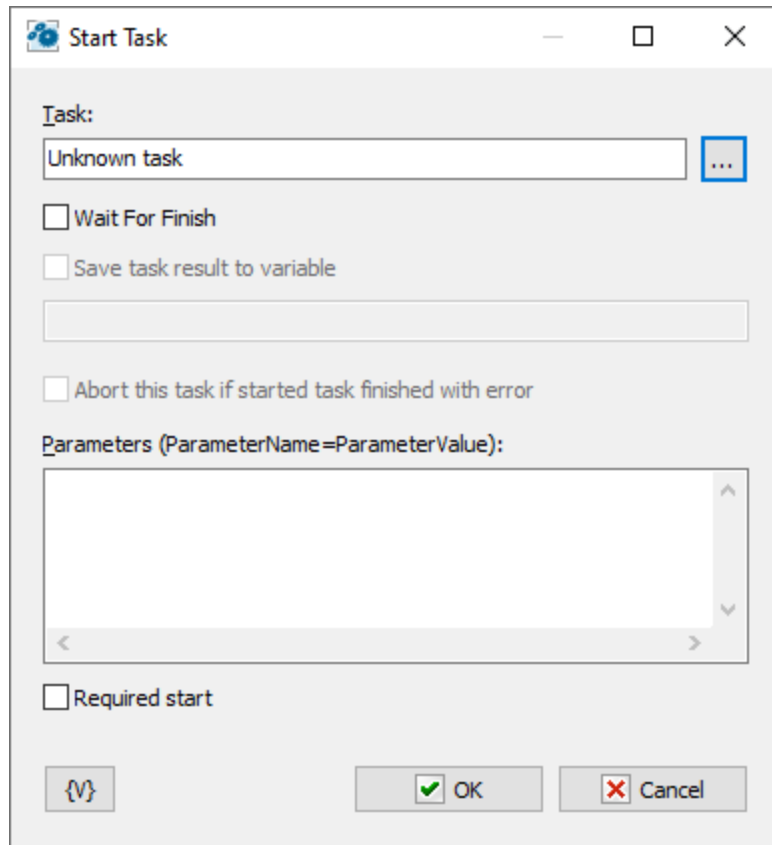
**This task**

If you want to disable this task without selection choose this option.

#### 4.11.3 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.

### Save task result to variable

When the action waits task for finish you can save the task result to variable. The task can return a result. To return some result, simply set the task's variable named *TaskResult* in the task.

Also you can retrieve the last task result anytime. Use system variable

***GetTaskResult(<ExtName>)***

### 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".

#### **Required start**

This parameter guarantees the start of a task. The problem is that the task can be working when you try to start it. In the most cases that doesn't matter because you need only to start the task at the appointed moment. In this case the action just sent the signal to start the task. If the task is idling, it will be started. If the task is already started, the RoboTask ignores the signal of a start because the task works by now.

But a task can be started with parameters and fulfill different actions against parameter values. Switch on this parameter if you need to ensure start up of a task when each effort to start a task. In this case the action will do efforts to start the task until a successful result.

#### **Abort this task is started task finished with error**

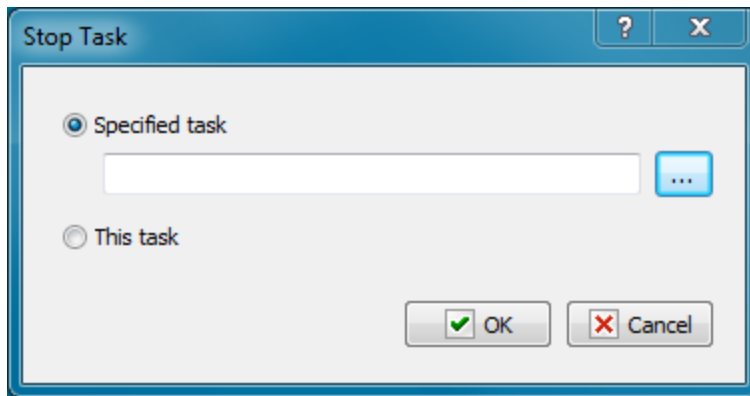
This parameter only makes sense when used together with the "Wait for finish" parameter. A running task may fail for various reasons, which can affect the parent task's algorithm. Therefore, it is possible to interrupt the execution of the current task.

#### **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.

### **4.11.4 Stop Task**

Allows you to stop the selected task.

**Specified Task**

Press Select button and choose the task from the list.

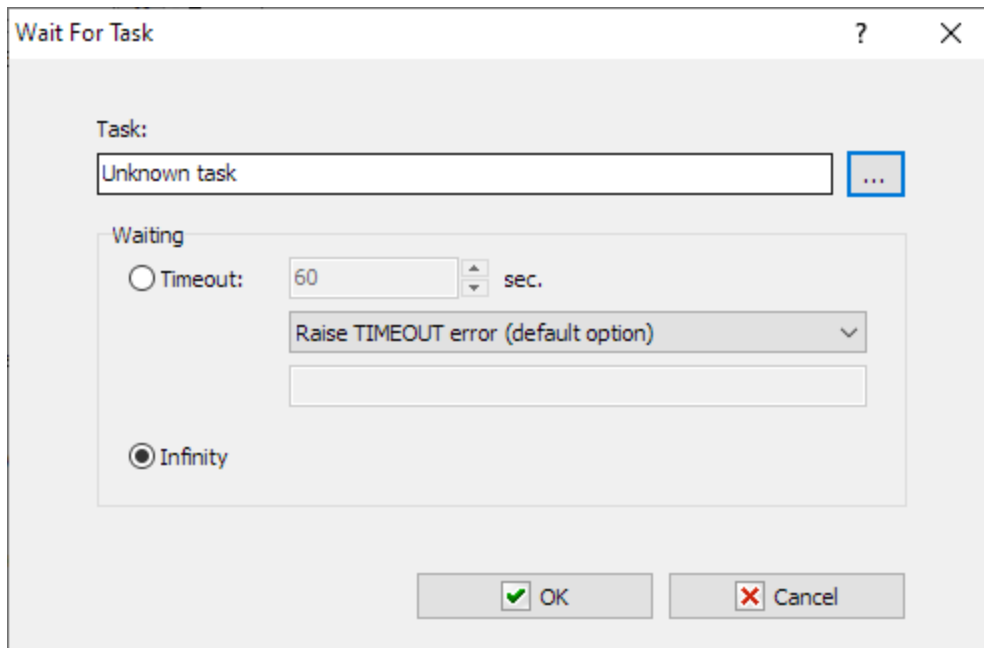
**This task**

If you want to stop this task without selection choose this option.

#### 4.11.5 Wait for Task

Waits until the selected task is finished. This actions 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.

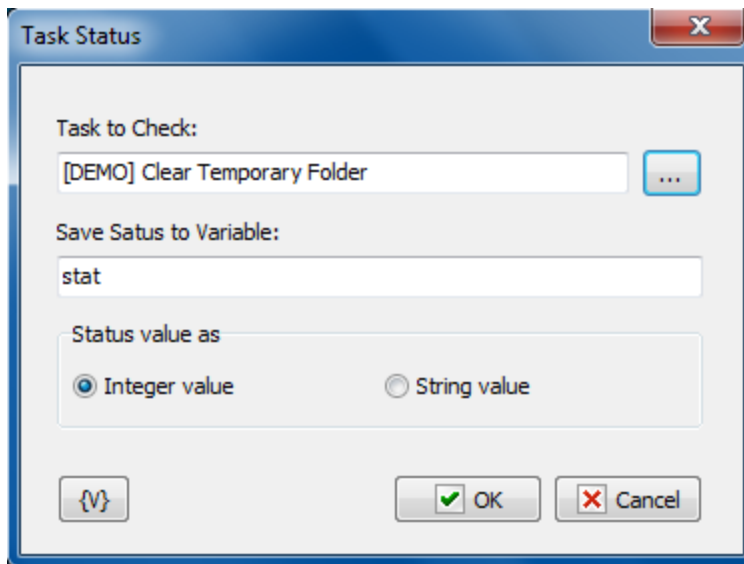
If the timeout is over you can specify an action:

- **Raise Timeout error** - this is default option
- **Set True or False to variable** - in this case the task will continue work without error.

## 4.11.6 Task status

This action reads the status of a specified task and writes it into a variable. The value can be a number from 0 to 4 or as a string:

- **Manual** – (0). A task without triggers or with switched off triggers in the idle mode. It is possible to run the task manually or from another task.
- **Idle** – (1). A task is in the idle mode and has active triggers. It is possible both running manual and automatically when trigger starts the task.
- **Disabled** – (2). A task is busy, it is in the edit mode. The task cannot be started.
- **Running** – (3). The task is in progress. A restart is possible only after the task is completed.
- **Stopping** – (4). A task has got the signal to stop and is waiting for the end of the active step. The task will be stopped only after the end of the active step.



### Task to Check

Specify the task for the reading its status. You can select the task of a list pressing the button to the right of the edit field.

### Save Status to Variable

Specify the name of the variable in which the status must be saved.

### Save status as

**Integer value** – is a status as an integer number from 0 to 4.

**String value** – is a status as a string: *Manual, Idle, Disabled, Running, Stopping*

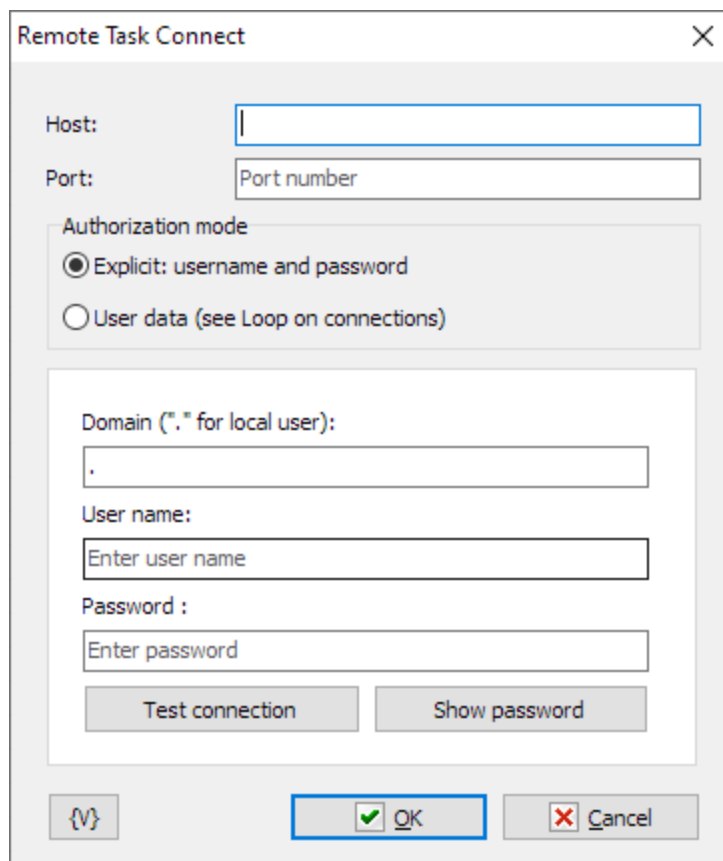
## 4.12 Remote Task

This set of actions is used to manage tasks in the remote RoboTask. A remote RoboTask instance can run on another computer, in another session, or in service mode.

Before you start working with remote tasks downloaded, you need to connect to a remote instance of RoboTask using the [Remote Task Connect](#)<sup>[220]</sup> action. One RoboTask can only support one connection.

### 4.12.1 Remote RoboTask Connect

This action begins work with the tasks of the remote RoboTask instance. One RoboTask can only support one connection. If you need to work with tasks of another RoboTask instance while a task is running, you must first close the current connection with the [Remote Task Disconnect](#)<sup>[222]</sup> action



The screenshot shows a dialog box titled "Remote Task Connect" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Host:** A text input field.
- Port:** A text input field containing the placeholder text "Port number".
- Authorization mode:** A group box containing two radio buttons:
  - Explicit: username and password
  - User data (see Loop on connections)
- Domain ("." for local user):** A text input field containing a single period ".".
- User name:** A text input field containing the placeholder text "Enter user name".
- Password:** A text input field containing the placeholder text "Enter password".
- Test connection:** A button.
- Show password:** A button.
- Footer:** A toolbar containing a help icon (question mark in a square), an **OK** button with a green checkmark, and a **Cancel** button with a red X.

**Host**

Computer name or IP address to connect

**Port**

The port number to connect. By default, the application port is 5500, the service port is 5501. But when configuring the RoboTask network server, the numbers can be changed to any others.

**Authorization mode**

Two authorization modes are available

- **Explicit: user name and password.** - This mode uses classic authorization using a username and password. Variables are allowed.
- **User data** - use an encrypted string that contains encrypted authorization parameters. This data can be obtained in the [Active Connections Loop](#)<sup>[228]</sup>. The data contains the authorization parameters of the currently open connections from the main RoboTask window.

**Domain**

Domain name. Used only if the server is configured for NT authorization. The dot (.) if using the local computer user.

**Username**

Name of registered user

**Password**

Password. This field has a "secure" mode for entering a password. But if you are using variables for the password, then you can click the "Show password" button so that you can edit this field in normal mode.

**Encrypted user info**

This data can be obtained in the [Active Connections Loop](#)<sup>[228]</sup> and saved to a variable. This variable can be used for authorization.

For example: {CurrentUserInfo}

**Test connection**

If you have specified the connection and authorization parameters explicitly, you can check the correctness of the parameters by clicking this button.

**Show password**

You can see the password explicitly or edit the password if you use variables.



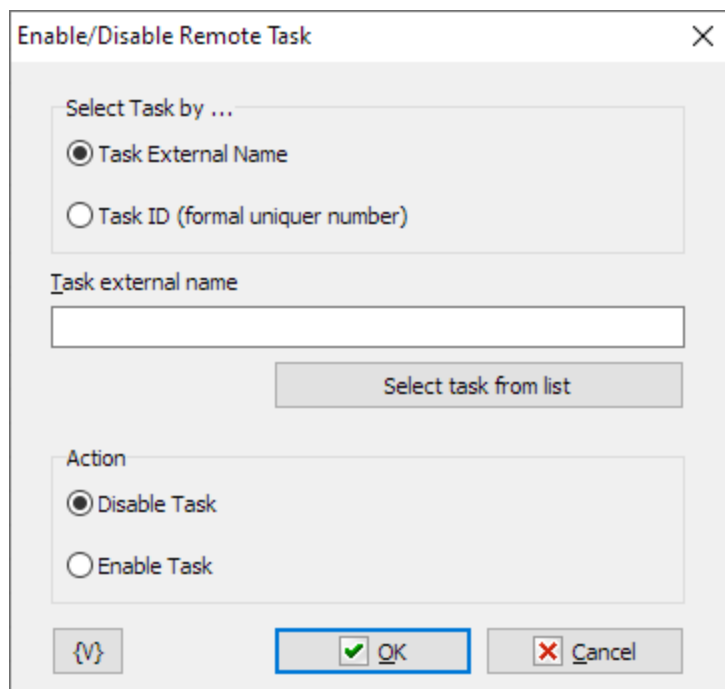
### 4.12.2 Remote RoboTask Disconnect

The action closes the current connection to the remote RoboTask.

The action does not require additional parameters.

### 4.12.3 Enable/Disable Remote Task

Changes the *Enabled* property of a task on a remote RoboTask instance.



#### Select task by ...

You can specify a task in two ways:

- **by External Name** - this method is preferable because you can define the External Name yourself to identify the task (see [Task Editor -> Advanced](#)<sup>54</sup>)
- **by Task ID** - if you know the ID, you can specify the task ID directly. Task ID is a formal unique number. It is assigned once when creating or importing a task and does not change during the "life" of the task.

#### Task External name or Task ID

The title of this field changes depending on the value of the task selection mode. Specify the *External Name* or *TaskID*, respectively. Variables can be used.

#### Select task from list

*External Name* or *TaskID* can be selected interactively from the list. To do this, the first time you need to enter the parameters for connecting to the remote RoboTask to get the list. The connection parameters are saved for the next sessions.

**Action**

There are two options to choose from

- **Enable** - "enables" the task. In other words, it starts all connected task triggers.
- **Disable** - stops all task triggers. In this case, the task will not start automatically.

**4.12.4 Start Remote Task**

Launches a task on a remote RoboTask instance.

Remote Task Start

Select Task by ...

Task External Name

Task ID (formal unique number)

Task external name

Select task from list

Wait For Finish

Save task result to variable

Abort this task if started task finished with error

Parameters (ParameterName=ParameterValue):

{ } OK Cancel

**Select task by ...**

You can specify a task in two ways:

- **by External Name** - this method is preferable because you can define the External Name yourself to identify the task (see [Task Editor -> Advanced](#)<sup>[54]</sup>)

- **by Task ID** - if you know the ID, you can specify the task ID directly. Task ID is a formal unique number. It is assigned once when creating or importing a task and does not change during the "life" of the task.

### Task External name or Task ID

The title of this field changes depending on the value of the task selection mode. Specify the *External Name* or *TaskID*, respectively. Variables can be used.

### Select task from list

*External Name* or *TaskID* can be selected interactively from the list. To do this, the first time you need to enter the parameters for connecting to the remote RoboTask to get the list. The connection parameters are saved for the next sessions.

### Wait for finish

Enable this checkbox if you need to wait for the task to complete. In this case, the execution of the current task is stopped until the running task on the remote RoboTask is completed.

### Save task result to variable

When the action waits task for finish you can save the task result to variable. The task can return a result. To return some result, simply set the task's variable named *TaskResult* in the task.

### Parameters (ParameterName = ParameterValue)

When starting the task, you can pass some values in the parameters. Within a task, they can be used as local variables. Parameters can be specified as text, each line as *ParameterName=ParameterValue*.

For example:

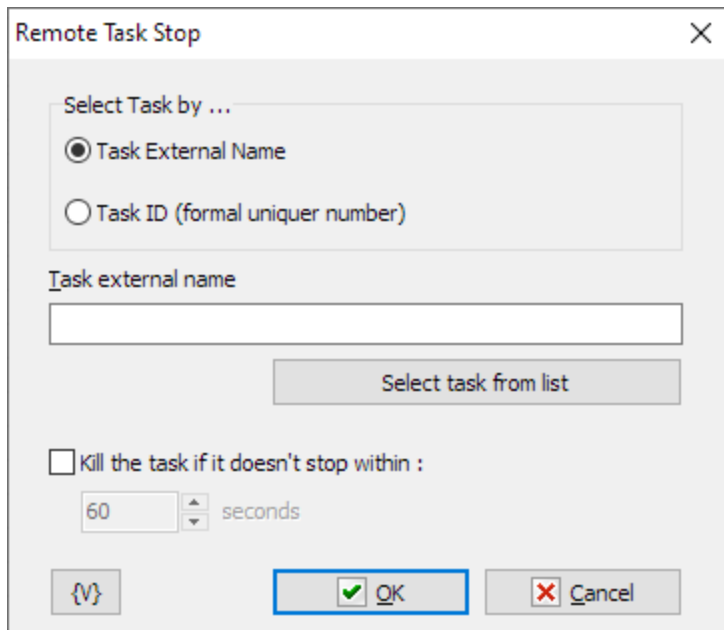
```
Order={orderID}
Date={orderDate}
```

### Abort this task is started task finished with error

This parameter only makes sense when used together with the "Wait for finish" parameter. A running task may fail for various reasons, which can affect the parent task's algorithm. Therefore, it is possible to interrupt the execution of the current task.

## 4.12.5 Stop Remote Task

Stops the specified task.



### Select task by ...

You can specify a task in two ways:

- **by External Name** - this method is preferable because you can define the External Name yourself to identify the task (see [Task Editor -> Advanced](#)<sup>[54]</sup>)
- **by Task ID** - if you know the ID, you can specify the task ID directly. Task ID is a formal unique number. It is assigned once when creating or importing a task and does not change during the "life" of the task.

### Task External name or Task ID

The title of this field changes depending on the value of the task selection mode. Specify the *External Name* or *TaskID*, respectively. Variables can be used.

### Select task from list

*External Name* or *TaskID* can be selected interactively from the list. To do this, the first time you need to enter the parameters for connecting to the remote RoboTask to get the list. The connection parameters are saved for the next sessions.

### Kill the task if it doesn't stop within ...

You can force abort the task if needed. Turn this switch on and specify the desired timeout in seconds.

## 4.12.6 Wait for Remote Task

Waits for the specified task to complete within the specified timeout.

Wait for Remote Task

Select Task by ...

Task External Name

Task ID (formal uniquer number)

Task external name

Select task from list

Waiting

Timeout: 60 sec.

Set True of False to variable

Input variable name

Infinity

OK Cancel

### Select task by ...

You can specify a task in two ways:

- **by External Name** - this method is preferable because you can define the External Name yourself to identify the task (see [Task Editor -> Advanced](#)<sup>[54]</sup>)
- **by Task ID** - if you know the ID, you can specify the task ID directly. Task ID is a formal unique number. It is assigned once when creating or importing a task and does not change during the "life" of the task.

### Task External name or Task ID

The title of this field changes depending on the value of the task selection mode. Specify the *External Name* or *TaskID*, respectively. Variables can be used.

### Select task from list

*External Name* or *TaskID* can be selected interactively from the list. To do this, the first time you need to enter the parameters for connecting to the remote RoboTask to get the list. The connection parameters are saved for the next sessions.

### Waiting

Standby mode. There are two options to choose from:

- **Timeout** - specify the time in seconds. If the task does not complete during the timeout, the action will generate an error.

If the timeout is over you can specify an action:

- **Raise Timeout error** - this is default option
- **Set True or False to variable** - in this case the task will continue work without error.
- **Infinite** - endless waiting for completion. We recommend using this option very carefully.

#### 4.12.7 Remote Task Status

This action allows you to save the state of the specified task to a variable.

The value can be a number from 0 to 4, or as a string:

- **Manual - (0)**. Task without triggers, or with triggers disabled in standby mode. It is possible to launch the task manually or start it from another task.
- **Idle - (1)**. The task is idle and has active triggers. Both manual and automatic start is possible when the trigger is triggered.
- **Disabled - (2)**. The task is in edit mode. The task cannot be started.
- **Running - (3)**. The task is being completed. Restarting is possible only after the completion of the task.
- **Stopping - (4)**. The task has received a halt signal and is waiting for the current step to complete. The task will only stop after the current step is completed.

Remote Task Status

Select Task by ...

Task External Name

Task ID (formal uniquer number)

Task external name

Select task from list

Save Satus to Variable:

Status value as

Integer value  String value

Integer value – is a status as an integer number from 0 to 4.  
String value – is a status as a string: Manual, Idle, Disabled, Running, Stopping

{}

OK Cancel

Select task by ...

You can specify a task in two ways:

- **by External Name** - this method is preferable because you can define the External Name yourself to identify the task (see [Task Editor -> Advanced](#)<sup>[54]</sup>)
- **by Task ID** - if you know the ID, you can specify the task ID directly. Task ID is a formal unique number. It is assigned once when creating or importing a task and does not change during the "life" of the task.

### Task External name or Task ID

The title of this field changes depending on the value of the task selection mode. Specify the *External Name* or *TaskID*, respectively. Variables can be used.

### Select task from list

*External Name* or *TaskID* can be selected interactively from the list. To do this, the first time you need to enter the parameters for connecting to the remote RoboTask to get the list. The connection parameters are saved for the next sessions.

### Save status to variable

Specify the name of the variable where you want to store the status value.

### Status value as

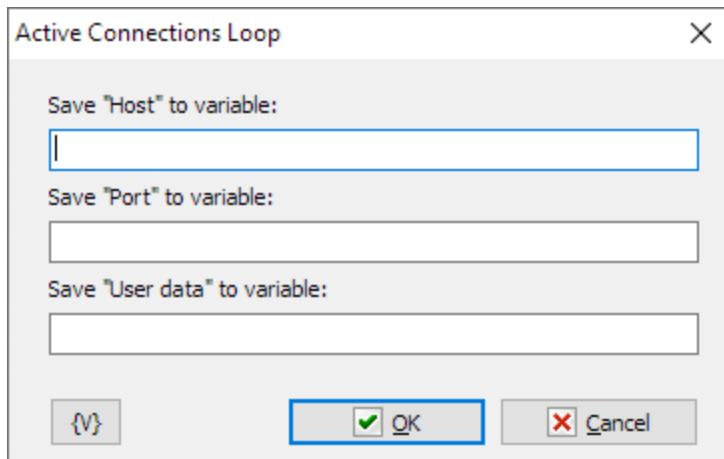
The value can be either an integer or a string:

- **Integer value** - is a status as an integer number from 0 to 4.
- **String value** - is a status as a string: *Manual, Idle, Disabled, Running, Stopping*

## 4.12.8 Active Connections Loop

This action allows you to loop over open connections with other RoboTask instances. On each iteration, the action stores the connection parameters into variables. You can use these variables in the [Remote task connect](#)<sup>[220]</sup> action.

The action does not use open connections. It only fetches the parameters for each open connection and stores them in variables. You can use this loop, for example, to run a task on all RoboTask instances with which a connection is open.

**Save "Host" to variable**

Specify the variable where you want to store the Host value. The Host is the computer name or IP address

**Save "Port" to variable**

Specify the variable where you want to store the Port value. The Port is an integer value for the port number. The Host-Port pair make up the connection parameters.

**Save "User data" to variable**

User authorization parameters in encrypted form. *User Data* is a base64 string. This string can be used in the [Remote Task Connect](#)<sup>[220]</sup> action

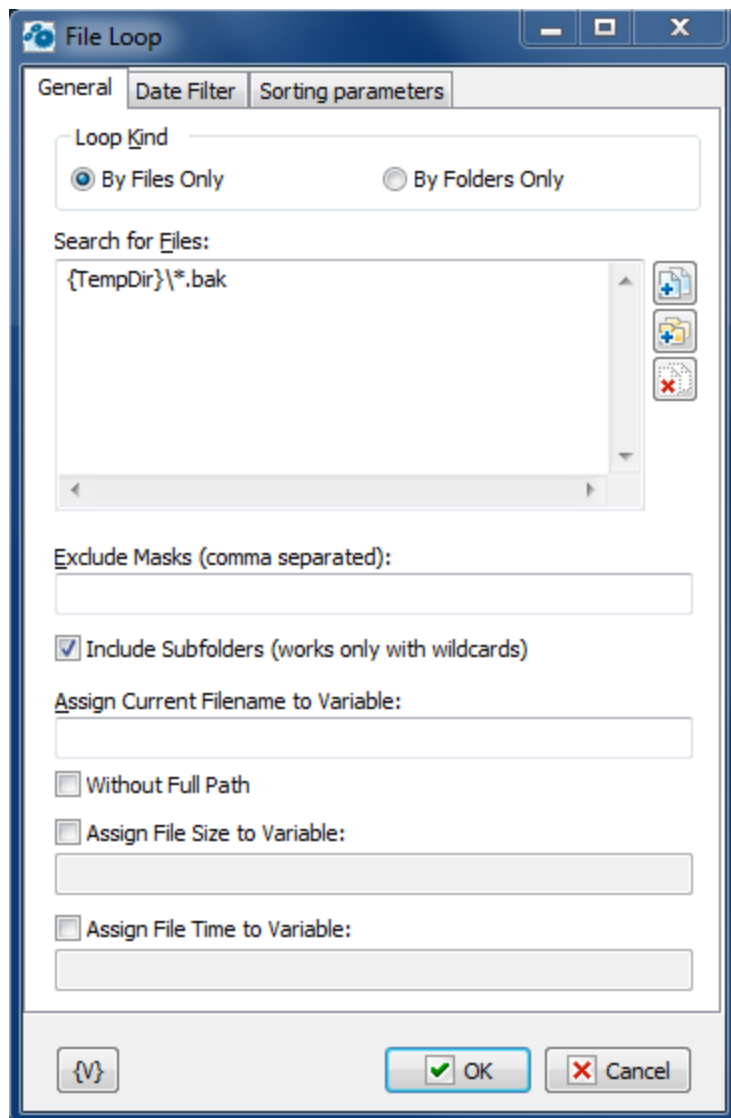
## 4.13 Loops and Flows

### 4.13.1 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.

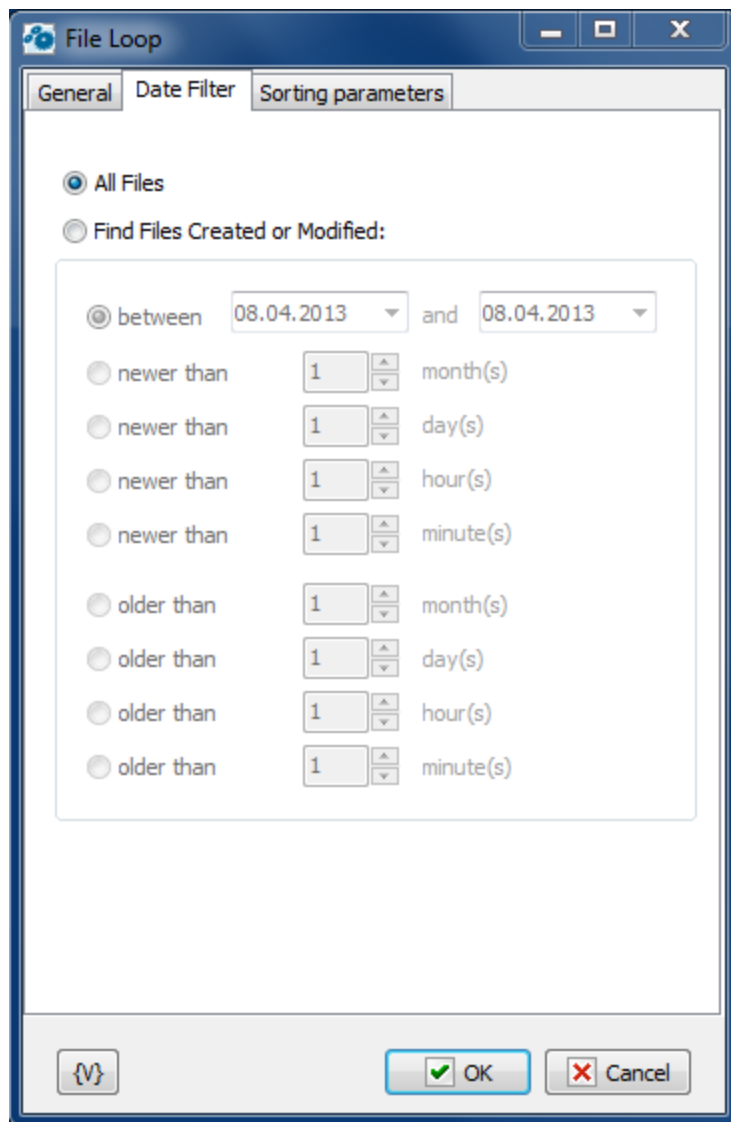
### **Assign File Size to Variable**

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.

### **Date Filter tab**



### All Files

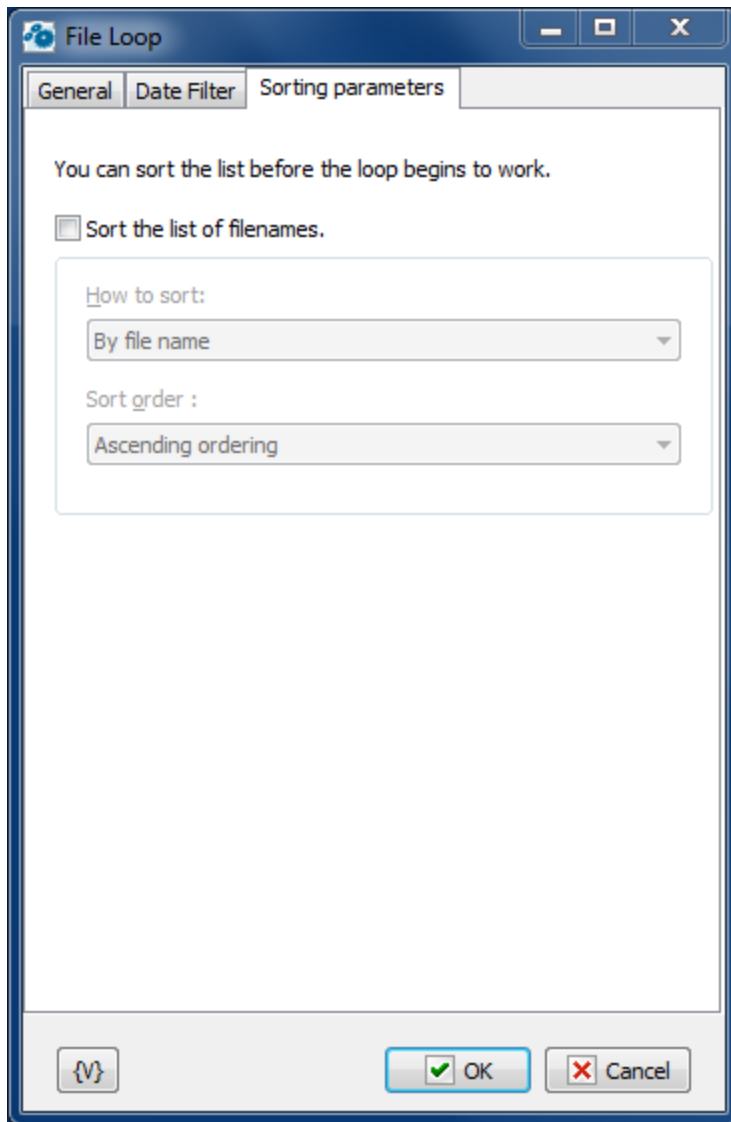
Click this option to include all files in the file loop.

### 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**

### Sorting parameters



Parameters of internal sorting of the file list

### **Sort the list of filenames**

Turn on the checkbox if you want to sort the list

### **How to sort**

There are four sorting modes:

- By file name
- By file time
- By file size
- By depth of nested folders

### **Sort order**

There are two modes: ascending or descending ordering

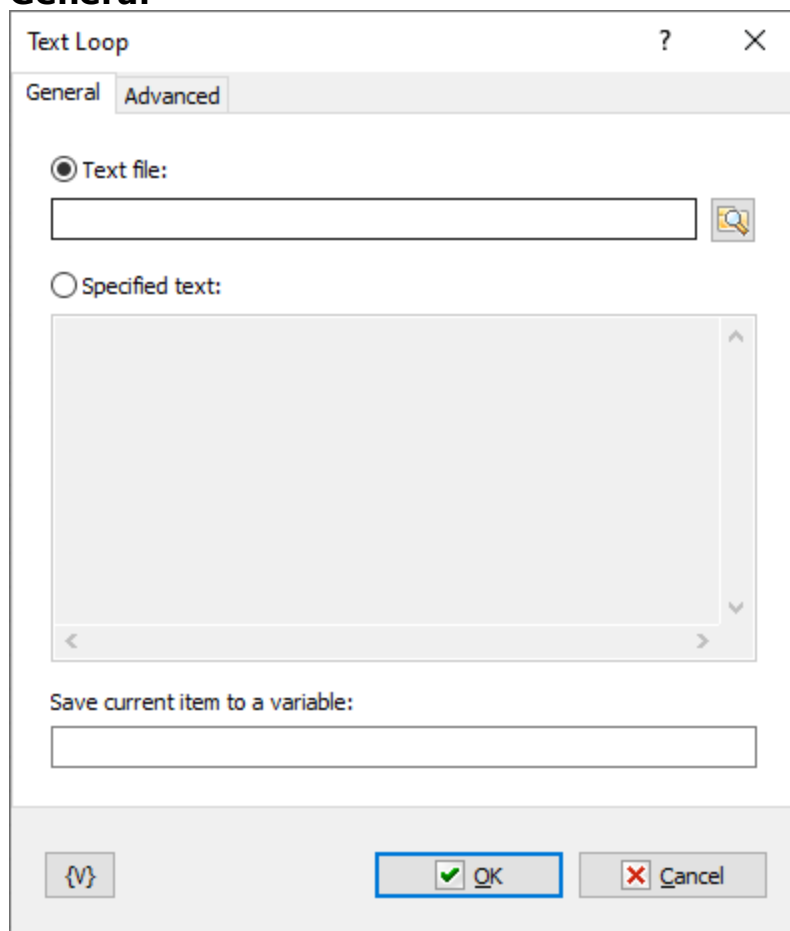
## Variable

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 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.

### General



The screenshot shows the 'Text Loop' dialog box with the 'General' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are two tabs: 'General' and 'Advanced'. The 'General' tab contains the following elements:

- A radio button labeled 'Text file:' is selected. Below it is a text input field and a browse button (magnifying glass icon).
- A radio button labeled 'Specified text:' is unselected. Below it is a large text area with scrollbars.
- A label 'Save current item to a variable:' is followed by a text input field.
- At the bottom left is a button with the '{V}' symbol.
- At the bottom center is an 'OK' button with a green checkmark icon.
- At the bottom right is a 'Cancel' button with a red 'X' icon.

### External Text File

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

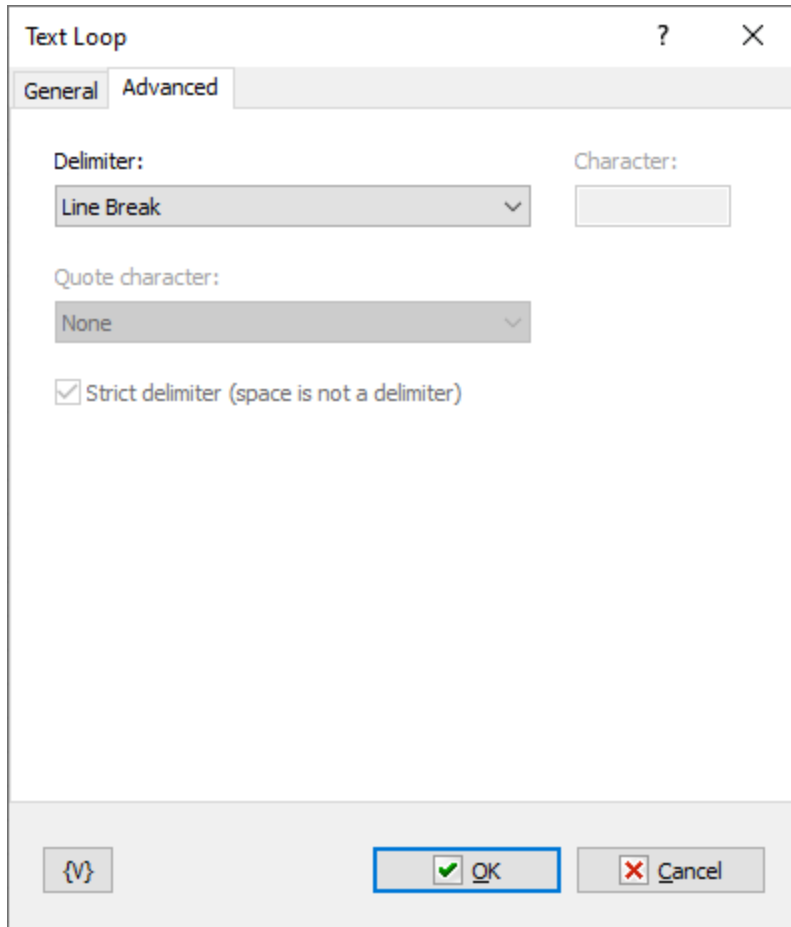
### Specified Text

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.

## Advanced



The screenshot shows the 'Text Loop' dialog box with the 'Advanced' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are two tabs: 'General' and 'Advanced'. The 'Advanced' tab contains the following settings:

- Delimiter:** A dropdown menu set to 'Line Break'.
- Character:** An empty text input field.
- Quote character:** A dropdown menu set to 'None'.
- Strict delimiter (space is not a delimiter)**

At the bottom of the dialog, there are three buttons: a button with a curly brace icon, an 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

By default, the Text Loop processes the text line by line as regular multi-line text.

### Delimiter

By default, newline characters are used as separators. In this case, the action treats the text as regular multi-line text.

If you choose a different separator, you have the option to select the CSV text format. In this case, the Text Loop treats the text as a delimiter-separated string (CSV).

### Variable

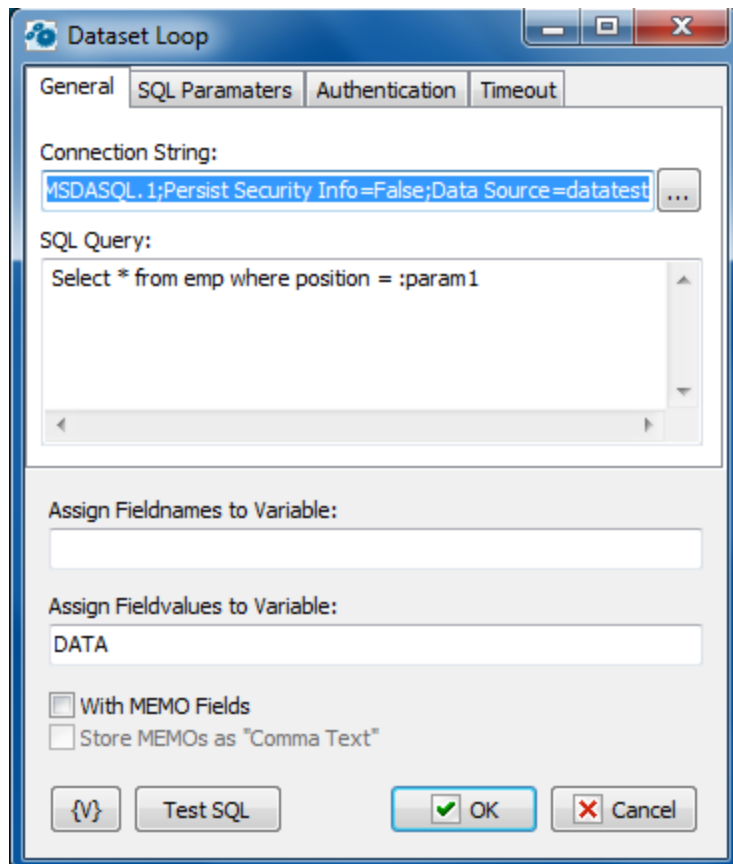
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

### 4.13.3 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.

**Test SQL**

Allows you to test the SQL syntax of your query.

**SQL parameters**

Sometimes using of parameters of a query is more convenient instead of generation of the query by means of variables. For example, when the variable value contains quotes or another special symbols. If you use variables (or macro substitution) when forming SQL inquiry, availability such symbols can "break" the syntax of the query and you will get a runtime error.

Parameters SQL are designated in the inquiry by means of colon (:) before the parameter name.

For example:

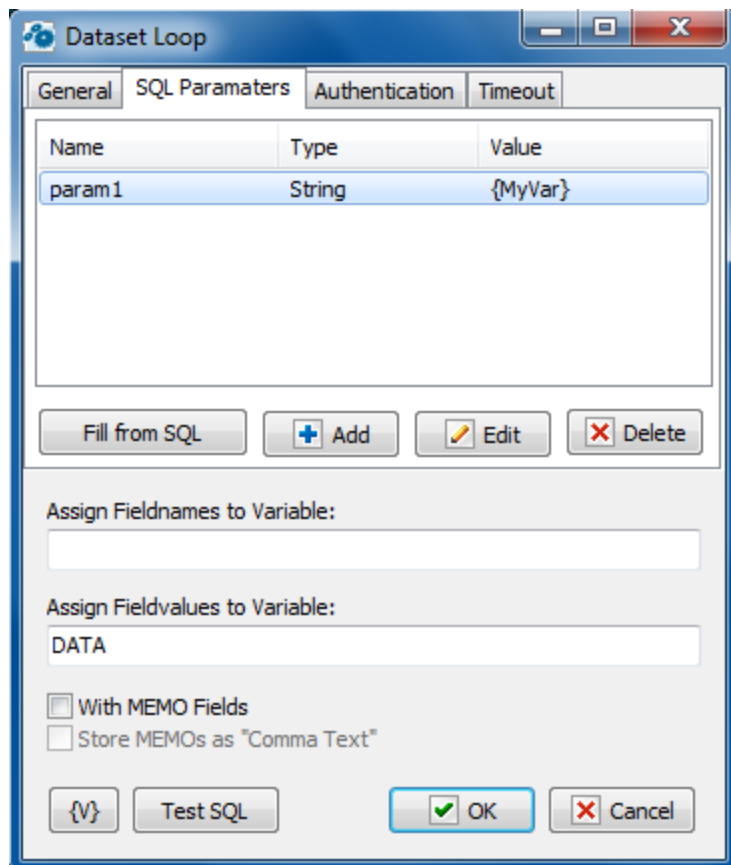
```
Select * from employee where hiredate > :dt and salary >= :amount
```

*:dt* and *:amount* are parameters of the query.

Parameters can be several types:

- **String** – any string
- **Integer** – an integer number
- **Float** – a real number or the number with a floating dot.
- **Boolean** – a logical value True or False.
- **DateTime** – a date and time. The value of such parameter must be defined in current system format of date and time (see the system regional settings). Notice that using of date-time can have its own characteristics for different databases. For example, it is better specify such parameters for the MySQL as a string parameter in the format YYYY-MM-DD or YYYY-MM-DD hh:nn:ss.





### Fill from SQL

Fill the list of parameters from the current query. If a parameter is already in the list, this parameter will be without change. The function fills only the missing parameters in the list.

### Add

Add a parameter manually. You will be proposed to type or select from the list the parameter name, select the parameter type and input an expression as the parameter value.

Using of variables in a parameter name and in its value is allowed. The RoboTask will substitute variable values instead of variables before performing the SQL.

### Edit

Edit the properties of the selected parameter.

### Delete

Delete the selected parameter from the list.

If you specify the parameter which is absent in the query, the action ignores this parameter when performing and writes the proper warning in the log of the task.

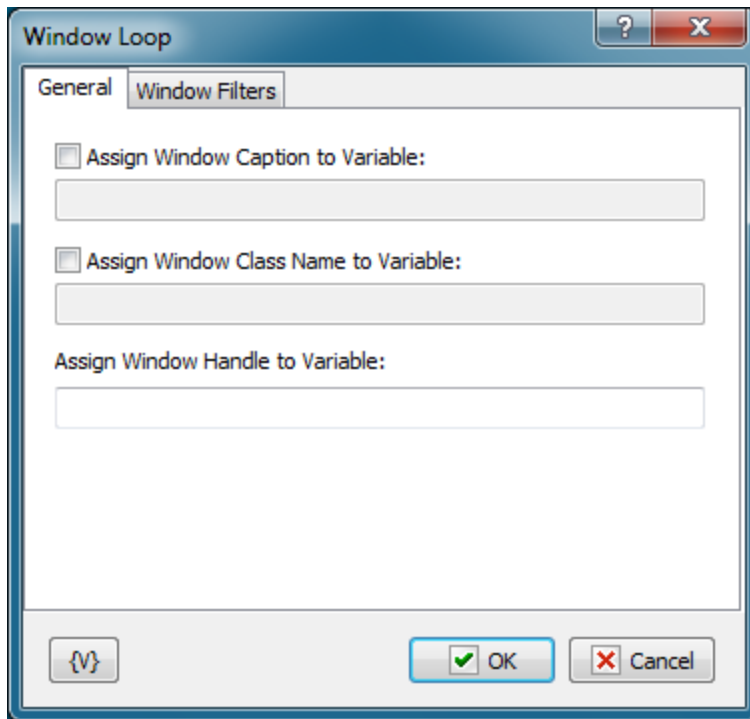
### Variable

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.

#### 4.13.4 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.

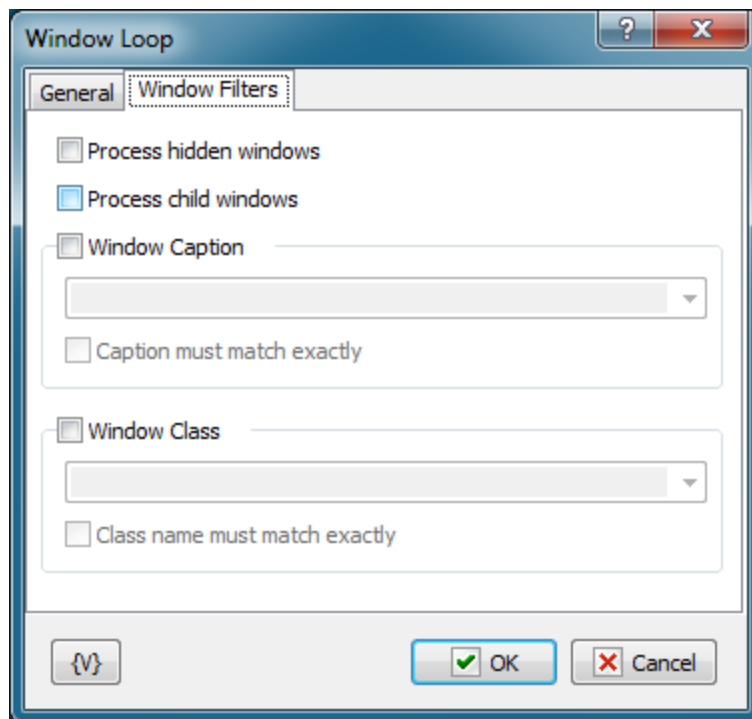
##### **Assign Window Class to variable**

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.

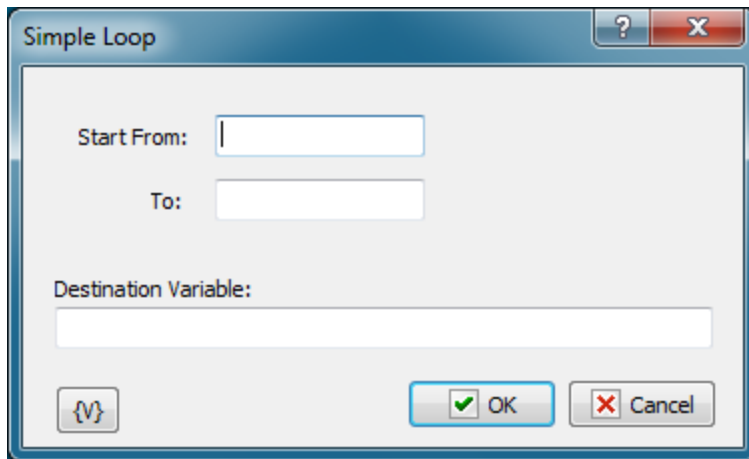
The "Window caption" and "Window class" lists depend on the "Process Hidden Windows" and "Process Child Windows" checkboxes.

### 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.

## 4.13.5 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.

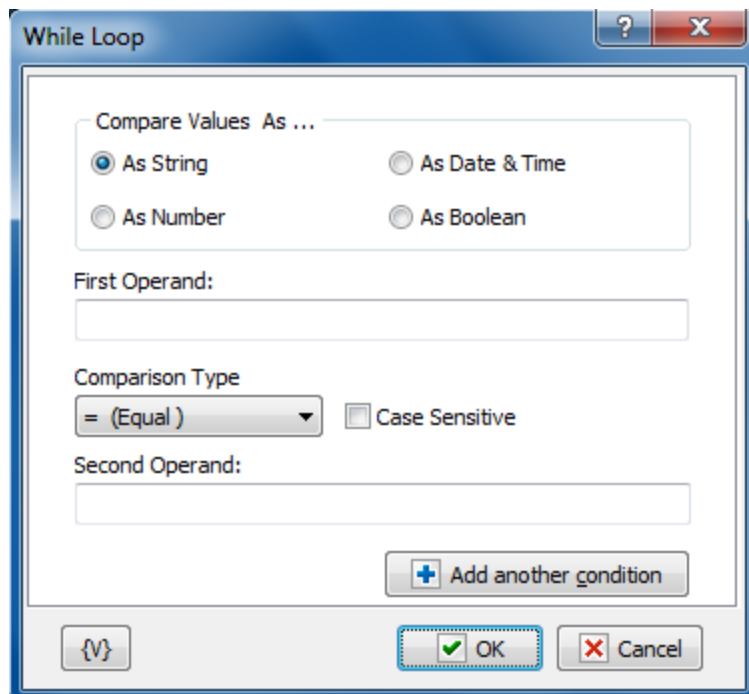
**Destination Variable**

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.

#### 4.13.6 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

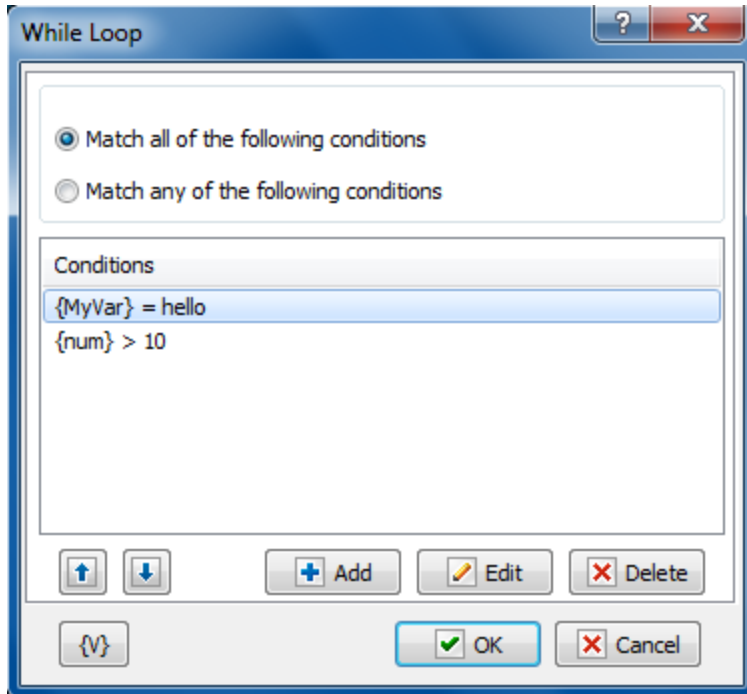
Click to make the comparison dependent on whether upper case or lower case characters are used in the expressions.

## Second Operand

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

It is possible to use a combination from several conditions in the actions [If..Then](#)<sup>[245]</sup>, [ElseIf..Then](#)<sup>[248]</sup> and *While Loop*. To add a condition, press the button *Add Another Condition*. You will see the similar dialog of editing of the comparison condition.

If you use more than one condition, you will see the list of conditions instead of parameters of conditions.



## The combination of conditions

Two variants of the combination of conditions are possible.

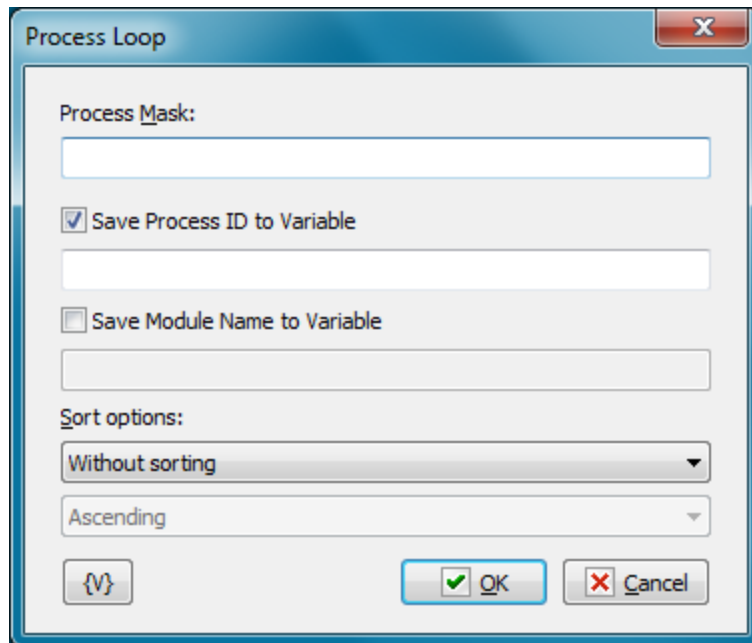
- **Match all of the following conditions** – the common result of the comparing will be True if all enumerated conditions return the result True (logical AND).
- **Match any of the following conditions** - the common result of the comparing will be True if at least one of the enumerated conditions returns the result True (logical OR).

## 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.

### 4.13.7 Process Loop

This loop allows you to enumerate running processes and save the process ID (PID) and the module name into variables at each iteration.



### Process Mask

Specify the name of the EXE file. An empty string is equivalent to a mask \*. \*. You can specify the module name explicitly.

### Save Process ID to Variable

Switch on this checkbox and enter the variable name to store the PID value

### Save Module Name to Variable

Switch on this checkbox and enter the variable name to save the module name. If possible, the action retrieves the full file name (with the full path).

### Sort Options

The list of processes can be sorted if necessary. Available options:

- Without sorting
- By process ID (PID)
- By process Name
- Direction of sorting – Ascending or Descending

## 4.13.8 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.

### 4.13.9 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.

### 4.13.10 Continue

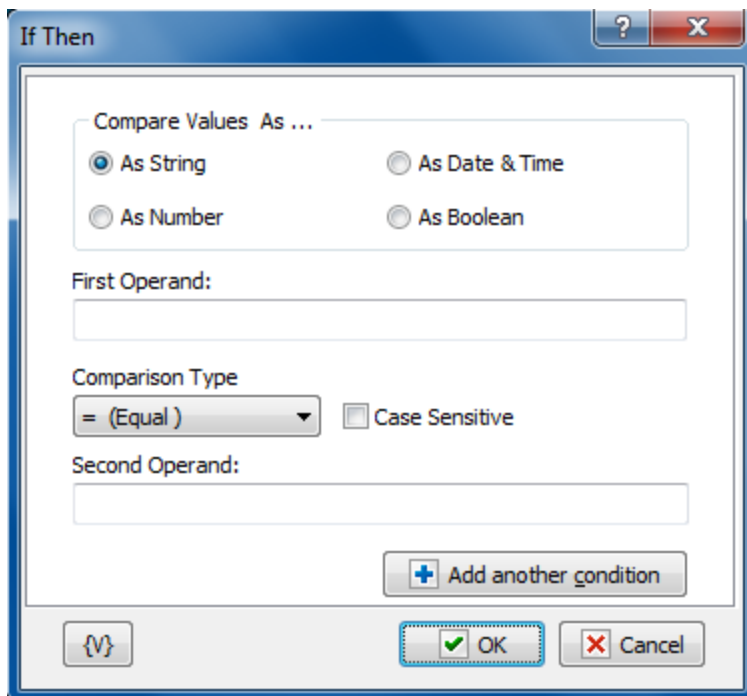
Skips all the actions between it and the **End Loop** action and begins a new iteration. No specific settings are associated with this action. The **Continue** action is used in conjunction with the **If Then** action.

### 4.13.11 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

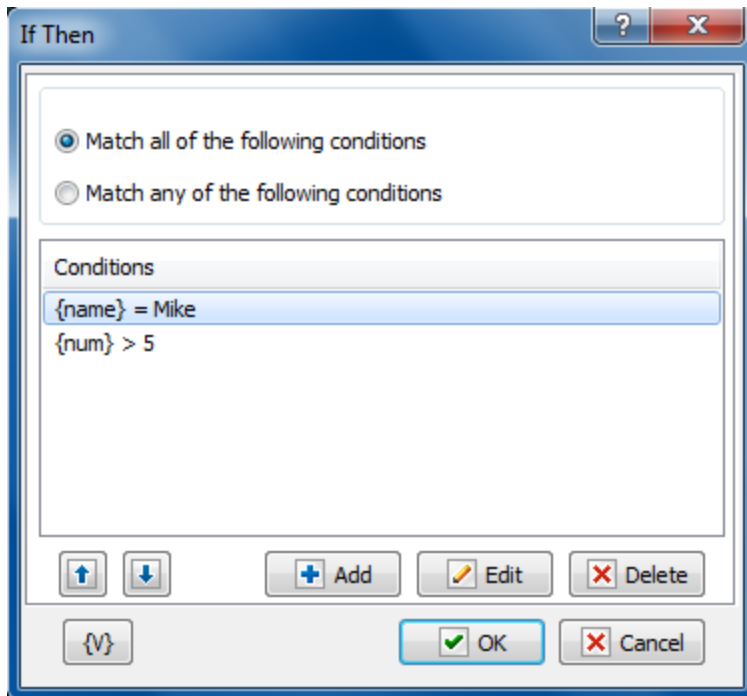
Click to make the comparison dependent on whether upper case or lower case characters are used in the expressions.

### Second Operand

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

It is possible to use a combination from several conditions in the actions *If..Then*, [ElseIf..Then](#)<sup>[248]</sup> and [While Loop](#)<sup>[241]</sup>. To add a condition, press the button *Add Another Condition*. You will see the similar dialog of editing of the comparison condition.

If you use more than one condition, you will see the list of conditions instead of parameters of conditions.



### The combination of conditions

Two variants of the combination of conditions are possible.

- **Match all of the following conditions** – the common result of the comparing will be True if all enumerated conditions return the result True (logical AND).
- **Match any of the following conditions** - the common result of the comparing will be True if at least one of the enumerated conditions returns the result True (logical OR).

### 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.

#### 4.13.12 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.

#### 4.13.13 End If

Indicates the end of the **If Then** block.

#### 4.13.14 Elself Then

This action allows to build a construction of a complex selection with several conditions. For example:

```
If <condition1> then
  DoSomething1
ElseIf <condition2> then
  DoSomething2
ElseIf <condition3> then
  DoSomething3
.....
Else
  DoSomethingN
End if
```

This action can be put in the block "If Then - EndIf" (between [If..Then](#)<sup>[245]</sup> and [End If](#)<sup>[247]</sup>).

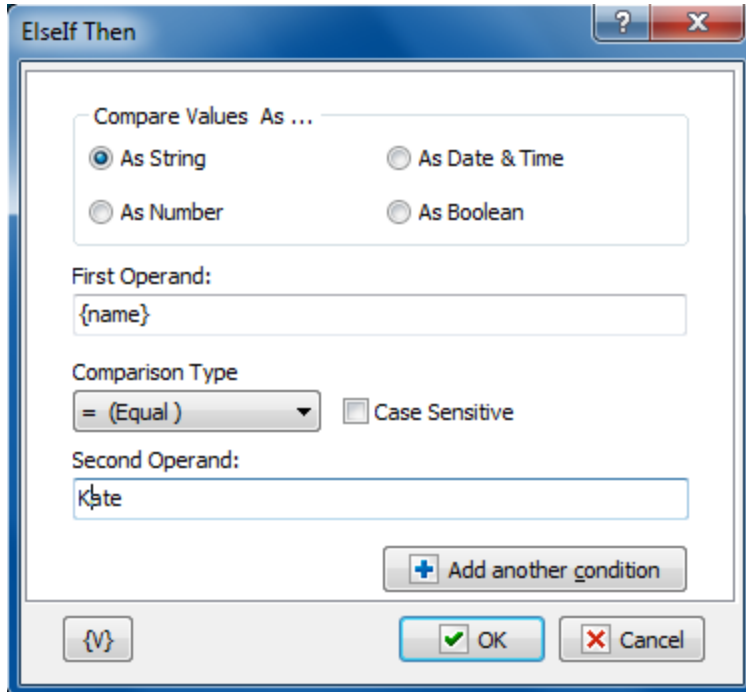
Such construction works in the following way:

- RoboTask checks the condition *<Condition1>*. If the condition is fulfilled (the logical value true) then RoboTask performs the steps *DoSomething1*.
- Otherwise, RoboTask checks the next condition *<Condition2>*. If the condition is fulfilled (the logical value True) then RoboTask performs the steps *DoSomething2*
- Otherwise, RoboTask checks the next condition *<Condition3>*. And so on.
- If none of enumerated conditions is fulfilled, the performing of the task moves to the block Else and then RoboTask performs the steps *DoSomethingN*.

Of course, it is possible to realize this algorithm by means of actions If..Then, Else, and End If. But apparently such construction is less convenient and less demonstrable. An algorithm with three conditions will look like this:

```
If <condition1> then
  DoSomething1
Else
  If <condition2> then
    DoSomething2
  Else
    If <condition3> then
      DoSomething3
    Else
      DoSomethingN
    End If
  End If
End if
```

The parameter editing dialog of the action **ElseIf Then** is completely identical to the window of configuration for the action If Then.



### 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

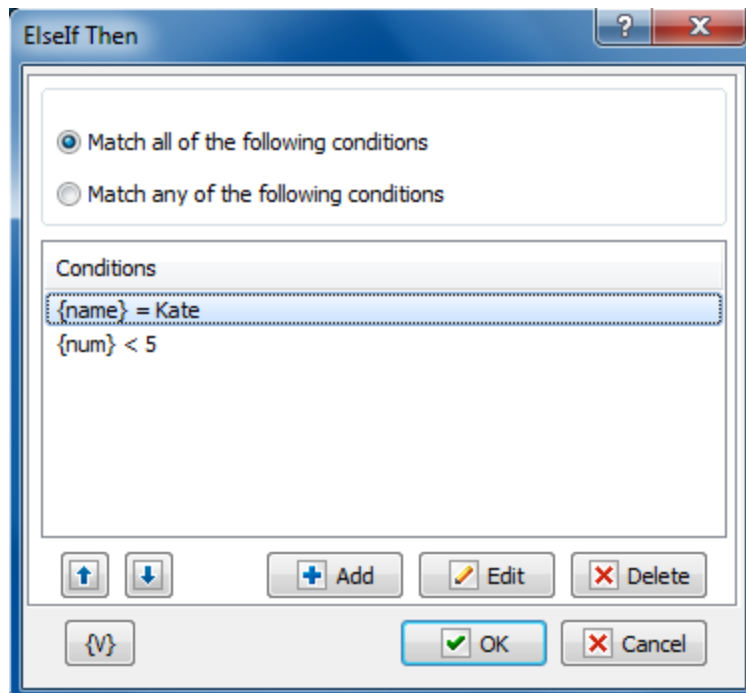
Click to make the comparison dependent on whether upper case or lower case characters are used in the expressions.

### Second Operand

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

It is possible to use a combination from several conditions in the actions [If..Then](#)<sup>[245]</sup>, [ElseIf..Then](#) and [While Loop](#)<sup>[241]</sup>. To add a condition, press the button *Add Another Condition*. You will see the similar dialog of editing of the comparison condition.

If you use more than one condition, you will see the list of conditions instead of parameters of conditions.



### The combination of conditions

Two variants of the combination of conditions are possible.

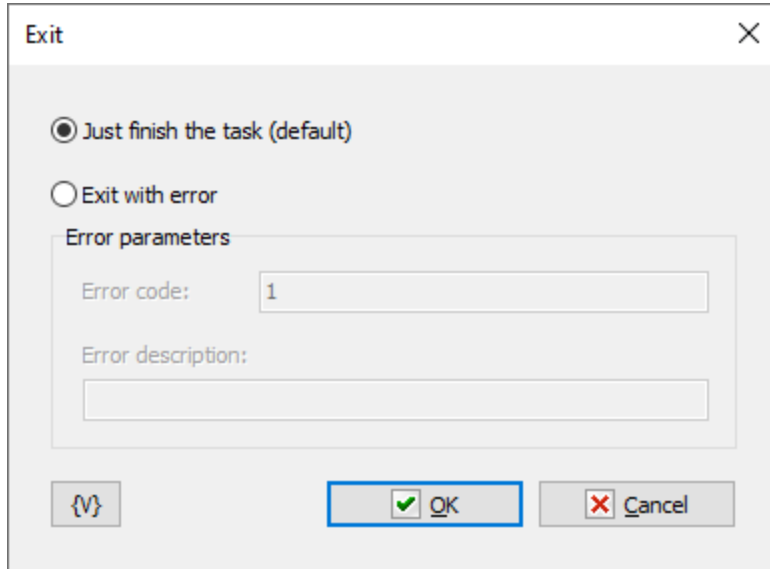
- **Match all of the following conditions** – the common result of the comparing will be True if all enumerated conditions return the result True (logical AND).
- **Match any of the following conditions** - the common result of the comparing will be True if at least one of the enumerated conditions returns the result True (logical OR).

### 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.

### 4.13.15 Exit

This action makes an immediate exit from the current task. The action allows to raise an error if it's necessary.



#### **Just finish the task**

Exit from the task. This is default option.

#### **Exit with error**

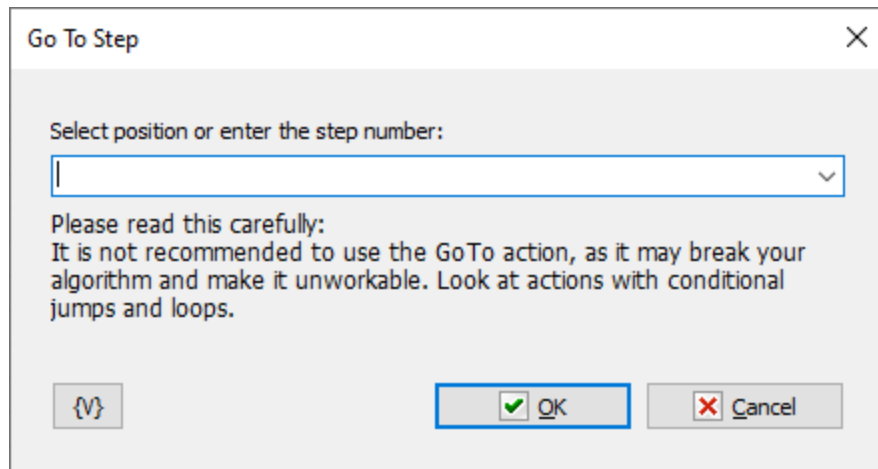
If you need to raise an error choose this option. Sometimes this is useful.

In this case the task will be closed with specified error. You can set the error code and error message.

### 4.13.16 GoTo

This action provides an unconditional transition of task execution to the specified step.

We do not recommend using the GoTo action, despite its simplicity. You can easily break your algorithm and make it non-functional. It is better to look towards [conditional transitions and loops](#)<sup>[229]</sup>.



### Select position or enter the step

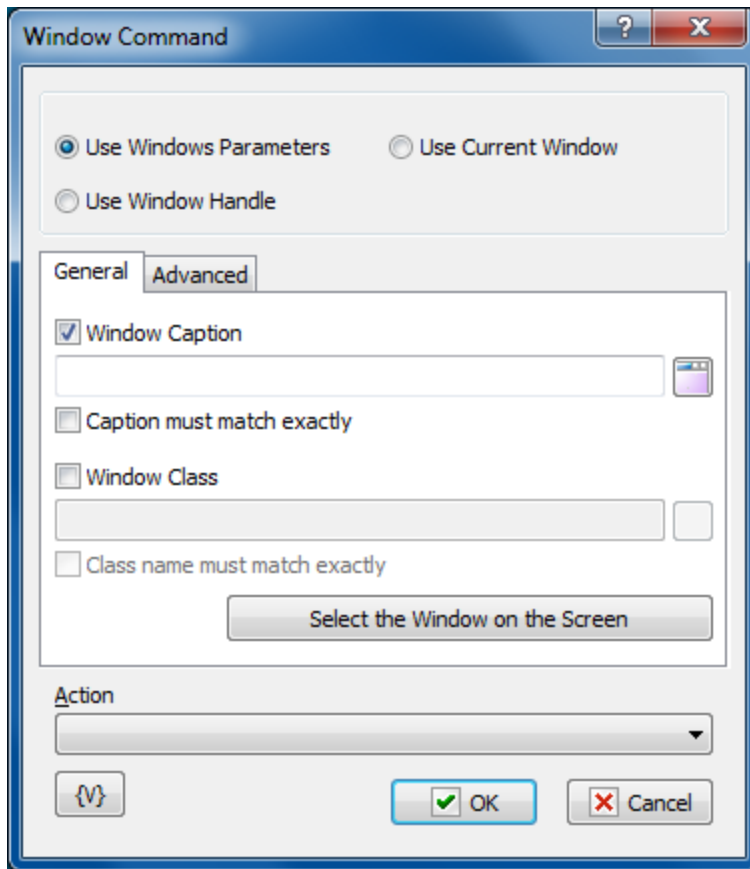
You can choose the step from drop-down list or enter the step number. Variables are allowed. Step must be greater than 0.

If the step number is greater than the number of steps in the task, then the task will simply finish its execution (like the action [Exit](#)<sup>[251]</sup>)

## 4.14 Window

### 4.14.1 Window Command

Manipulates certain windows that you specify.



### Window Selection Mode

- **Use Window Parameters.** In this case it is necessary to specify parameters of a required window: Window Caption, Window Class, compliance mode, etc. (see the description below).
- **Use Window Handle.** In this case it is necessary to specify a Window Handle. Handle is a number, an identifier of a window. It can be got, for example, in the action "Window loop". The Window Handle uniquely identifies the window in a system, so the rest of window parameters have no sense.
- **Use Current Window.** In this case a current active window is used. For example, if the necessary window was activated in the previous actions of a task, there is no need to specify window parameters again. Just use the current active window.

### 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>.



**Window Class.**

If you want to search the window by a window class, switch on this checkbox and input the expected name of the window class. You can input only part of the name. In this case the action will check availability of the input line in the name of the window class.

**Class Name Must Match Exactly.**

If you need the exact coincidence of a window class, switch on this checkbox. Usually it is worthwhile to search by the whole class name, because a window class is permanent and does not change its value during the "life" of the window, unlike a caption.

**Select Window on the Screen.**

When pressing this button, you turn on the mechanism of selection a window on the screen. The window, which the mouse pointer points on, highlights with a red frame. Meanwhile, you can use the mouse to perform some actions. To select the necessary window, press *Ctrl*. To cancel - press *Esc*. When selecting a window, the fields Window Caption and Window Class are filled in automatically.

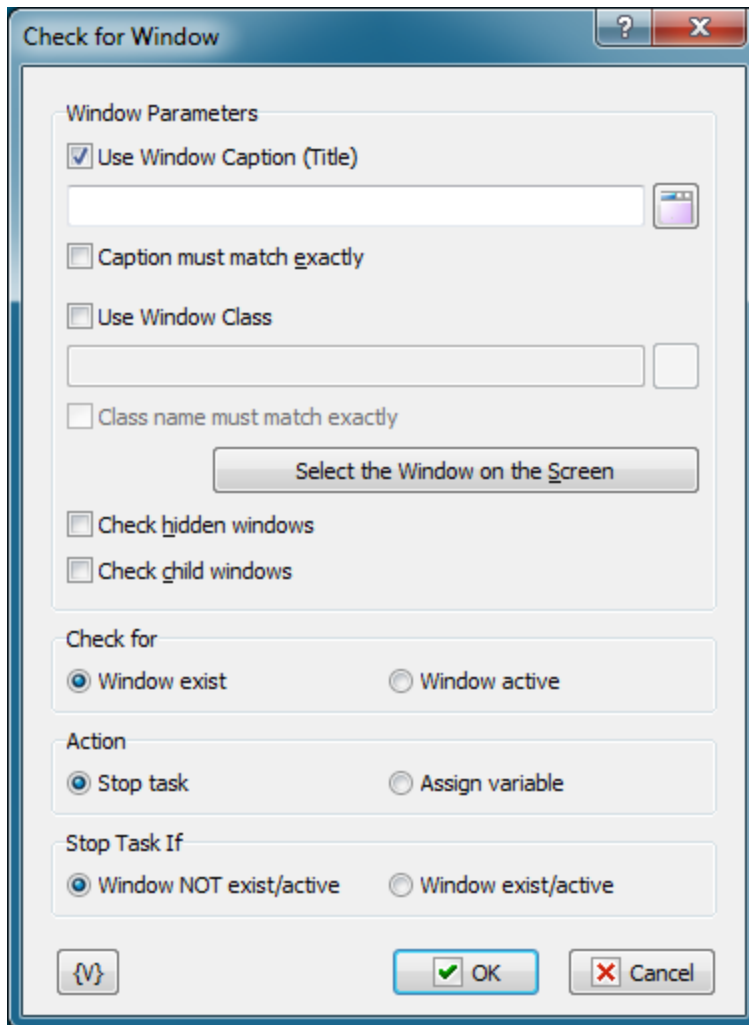
**Action**

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.

**4.14.2 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.

### Window Class.

If you want to search the window by a window class, switch on this checkbox and input the expected name of the window class. You can input only part of the name. In this case the action will check availability of the input line in the name of the window class.

### Class Name Must Match Exactly.

If you need the exact coincidence of a window class, switch on this checkbox. Usually it is worthwhile to search by the whole class name, because a window class is permanent and does not change its value during the "life" of the window, unlike a caption.

### Select Window on the Screen.

When pressing this button, you turn on the mechanism of selection a window on the screen. The window, which the mouse pointer points on, highlights with a red frame. Meanwhile, you can use the mouse to perform some actions. To select the necessary window, press *Ctrl*. To cancel - press *Esc*. When selecting a window, the fields Window Caption and Window Class are filled in automatically.

**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.

### Check For

Select one of the following options to check for existing windows or windows that are open.

- **Window Exist**
- **Window Active**

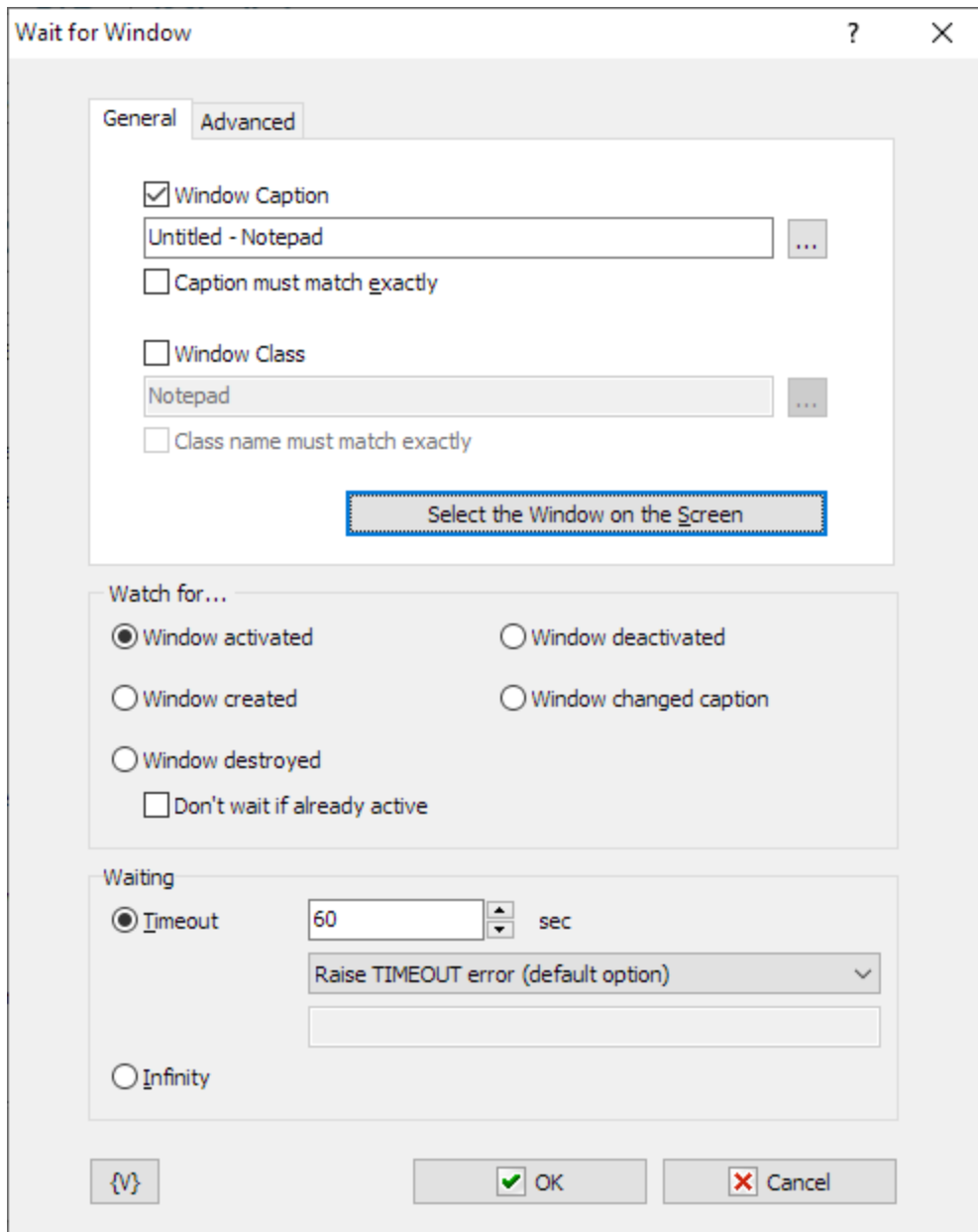
### Abort Task If

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.

## 4.14.3 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 of 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 of 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.

### Select the Window on the Screen

Select existing window interactively. This function highlights the window pointed by mouse. Press CTRL key to select current window or ESC to cancel operation

Wait for Window

General Advanced

Watch hidden windows

Watch child windows

Assign window handle to variable

Watch for...

Window activated  Window deactivated

Window created  Window changed caption

Window destroyed

Don't wait if already active

Waiting

Timeout 60 sec

Raise TIMEOUT error (default option)

Infinity

{V} OK Cancel

### 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.

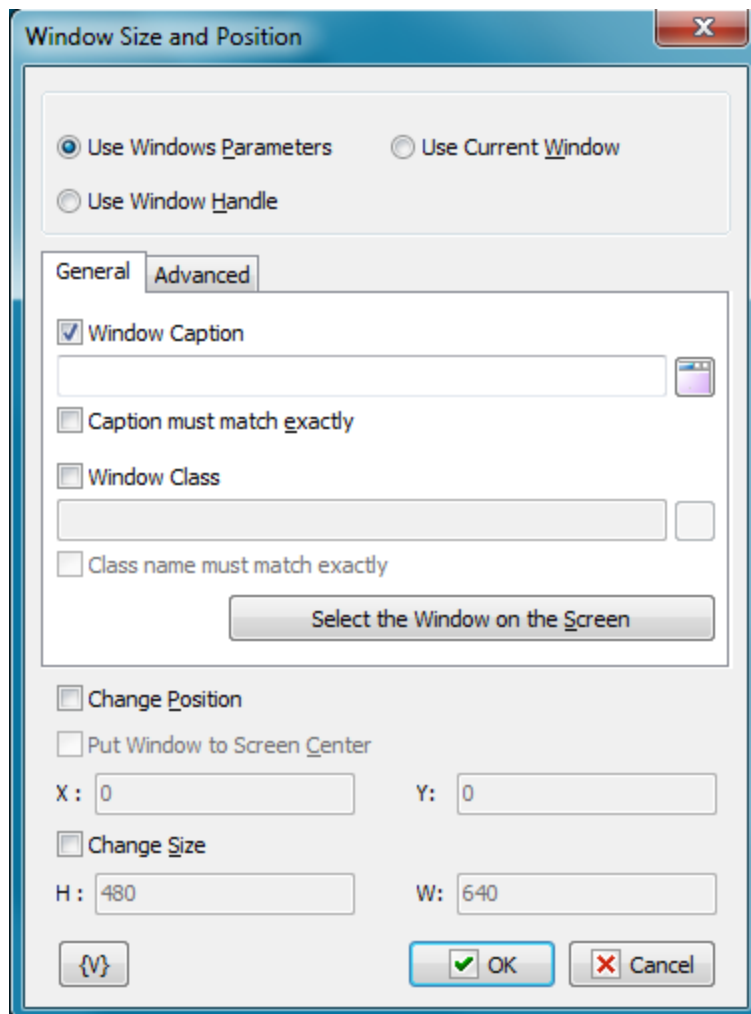
**Waiting** – specify the timeout in seconds or select the Infinity option.

If the timeout is over you can specify an action:

- **Raise Timeout error** - this is default option
- **Set True or False to variable** - in this case the task will continue work without error.

## 4.14.4 Window Set Size and Position

This action allows to change sizes and position of a specified window. The dialog of parameter editing resembles the editor of parameters of the action "[Window command](#)<sup>252</sup>"



### Window Selection Mode

- **Use Window Parameters.** In this case it is necessary to specify parameters of a required window: Window Caption, Window Class, compliance mode, etc. (see the description below).
- **Use Window Handle.** In this case it is necessary to specify a Window Handle. Handle is a number, an identifier of a window. It can be got, for example, in the action "Window loop". The Window Handle uniquely identifies the window in a system, so the rest of window parameters have no sense.
- **Use Current Window.** In this case a current active window is used. For example, if the necessary window was activated in the previous actions of a task, there is no need to specify window parameters again. Just use the current active window.

### 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>.

### **Window Class.**

If you want to search the window by a window class, switch on this checkbox and input the expected name of the window class. You can input only part of the name. In this case the action will check availability of the input line in the name of the window class.

### **Class Name Must Match Exactly.**

If you need the exact coincidence of a window class, switch on this checkbox. Usually it is worthwhile to search by the whole class name, because a window class is permanent and does not change its value during the "life" of the window, unlike a caption.

### **Select Window on the Screen.**

When pressing this button, you turn on the mechanism of selection a window on the screen. The window, which the mouse pointer points on, highlights with a red frame. Meanwhile, you can use the mouse to perform some actions. To select the necessary window, press *Ctrl*. To cancel - press *Esc*. When selecting a window, the fields Window Caption and Window Class are filled in automatically.

### **Change position**

Switch on this checkbox if you want to change a window position. After that, fill in parameters X and Y.

**X** – Specify the horizontal position of the left window edge. The usage of variables is allowed. The result of the expression must be an integer. Otherwise, the action uses default value, which is 0.

**Y** – Specify the upright position of the upper window edge. The usage of variables is allowed. The result of the expression must be an integer. Otherwise, the action uses default value, which is 0.

### **Put Window to Screen Center**

Action calculates new coordinates and put the window to the center.

### **Change size**

Switch on this checkbox if you want to change a window size. After that, fill in parameters H and W.

**H** – Specify a new height of the window. The usage of variables is allowed. The result of the expression must be an integer. Otherwise, the action uses default value, which is 480.

**W** - Specify a new width of the window. The usage of variables is allowed. The result of the expression must be an integer. Otherwise, the action uses default value, which is 640.

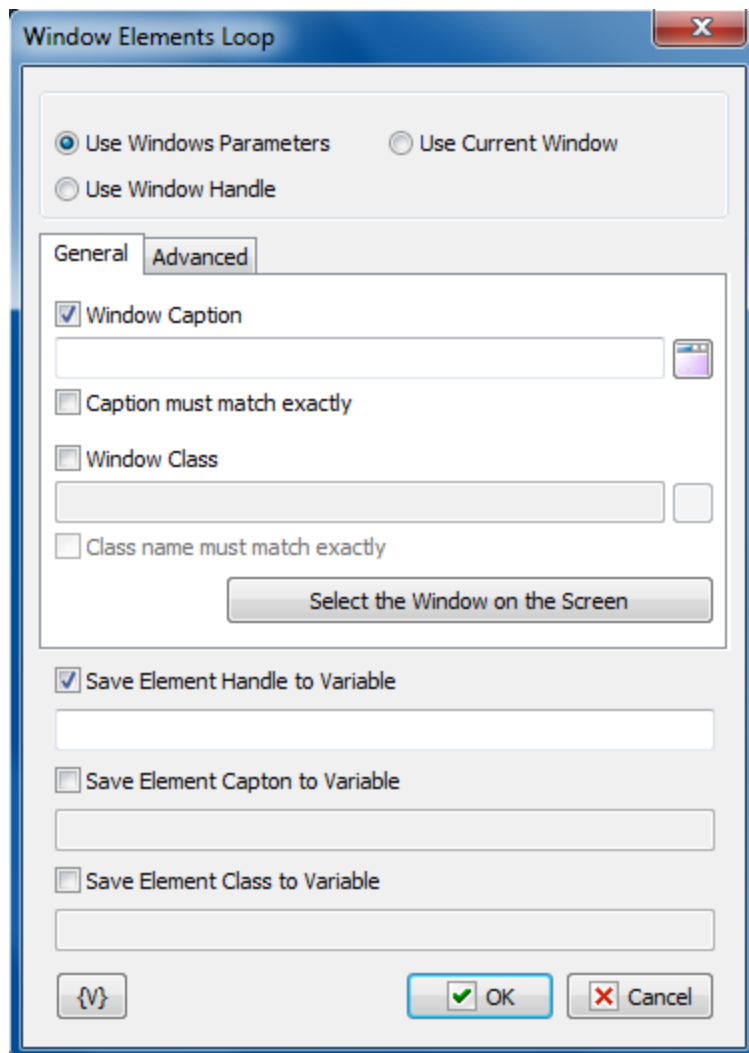


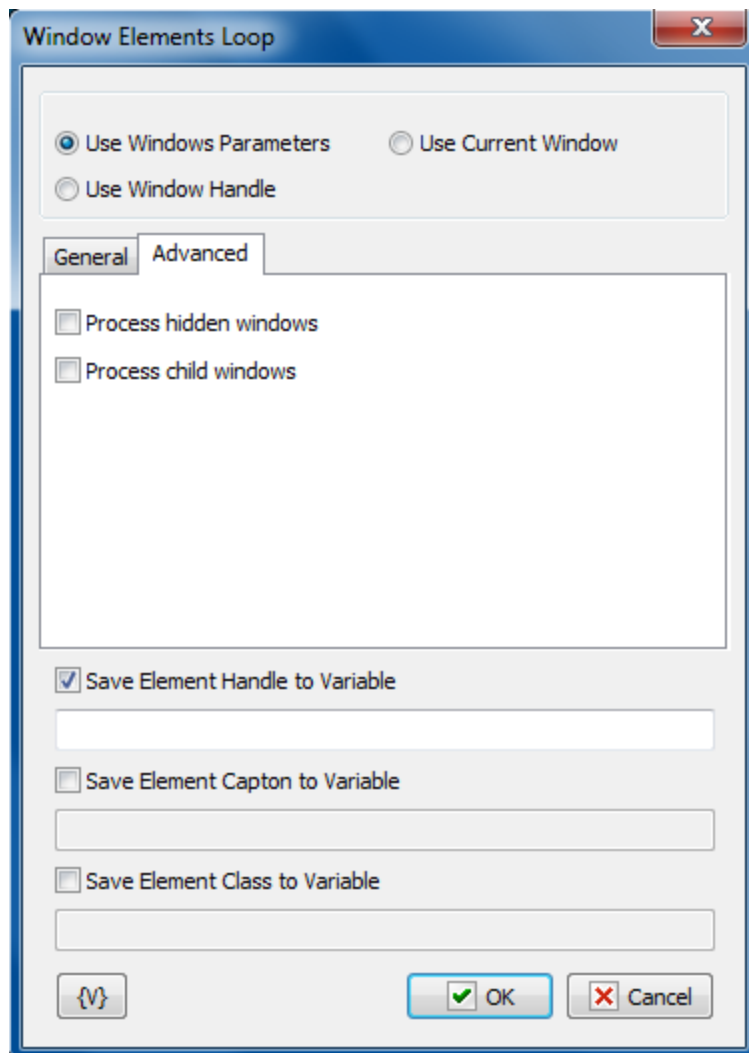
---

#### 4.14.5 Window Elements Loop

This is a loop by window elements. Sometimes it is possible to extract and analyze some information from window elements. The loop enumerates all window elements registered in the system. When every iteration, you can save the characteristics of the current element into variables:

- Handle
- Caption
- Class





### Window Parameters Mode

There are three options:

**Use Window parameters** - Use the Window Caption and the Window Class to look for the necessary window.

**Use Window Handle** - This is a search for a window by a handle. The handle can be got in the loop Window Loop, from the trigger Wait for Window or some other way.

**Use current window** - This is a current window in a moment of the executing of the action. This mode does not require any additional parameters.

### Use Window parameters

#### Window Caption.

If you want to search for a window by means of its caption, switch on this checkbox and input the expected window caption. You can input only part of the caption. In this case the action will check availability of the input part in the window caption.

#### Caption Must Match Exactly.

If you need the exact coincidence of a window caption, switch on this checkbox.

**Window Class.**

If you want to search the window by a window class, switch on this checkbox and input the expected name of the window class. You can input only part of the name. In this case the action will check availability of the input line in the name of the window class.

**Class Name Must Match Exactly.**

If you need the exact coincidence of a window class, switch on this checkbox. Usually it is worthwhile to search by the whole class name, because a window class is permanent and does not change its value during the "life" of the window, unlike a caption.

**Select Window on the Screen.**

When pressing this button, you turn on the mechanism of selection a window on the screen. The window, which the mouse pointer points on, highlights with a red frame. Meanwhile, you can use the mouse to perform some actions. To select the necessary window, press *Ctrl*. To cancel - press *Esc*. When selecting a window, the fields Window Caption and Window Class are filled in automatically.

**Window Handle.**

Input the expression containing a window handle. This parameter is used for the mode Use Window Handle. A window handle is just a number, identifier of a window in a current session of system work.

**Save Element Handle to Variable**

If you want to save the value of the element handle, switch on this checkbox and input the value name in the entry field.

**Save Element Caption to Variable**

If you want to save the element caption, switch on this checkbox and input the value name in the entry field.

**Save Element Class to Variable**

If you want to save the class of the element, switch on this checkbox and input the value name in the entry field.

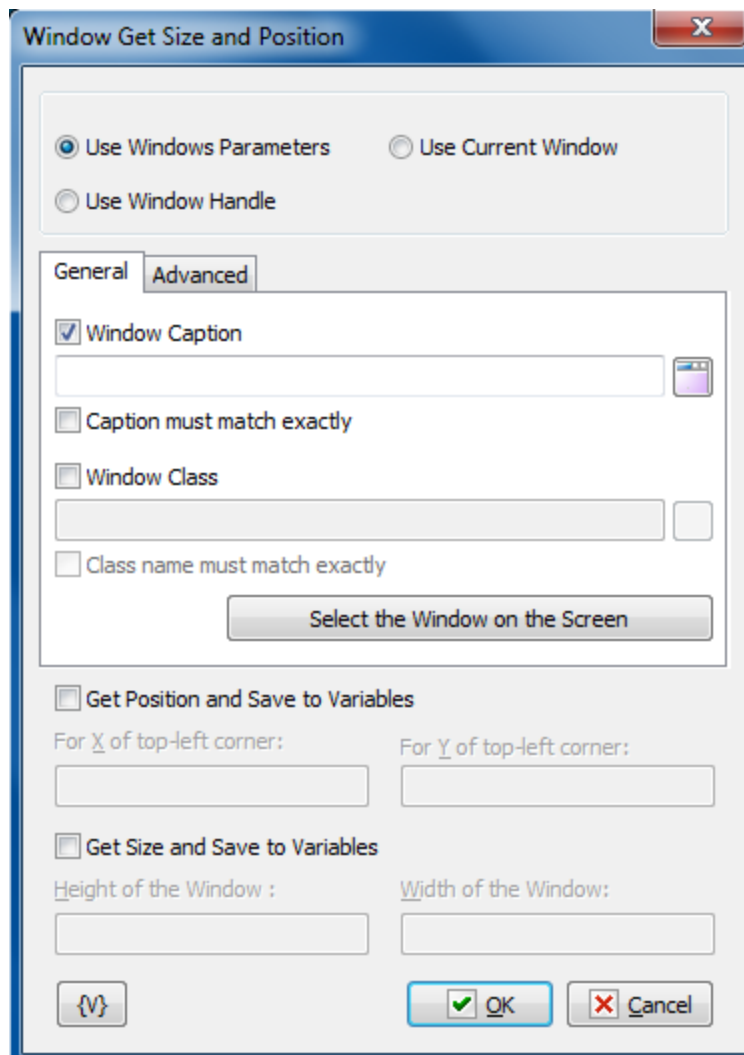
#### 4.14.6 Window Get Size and Position

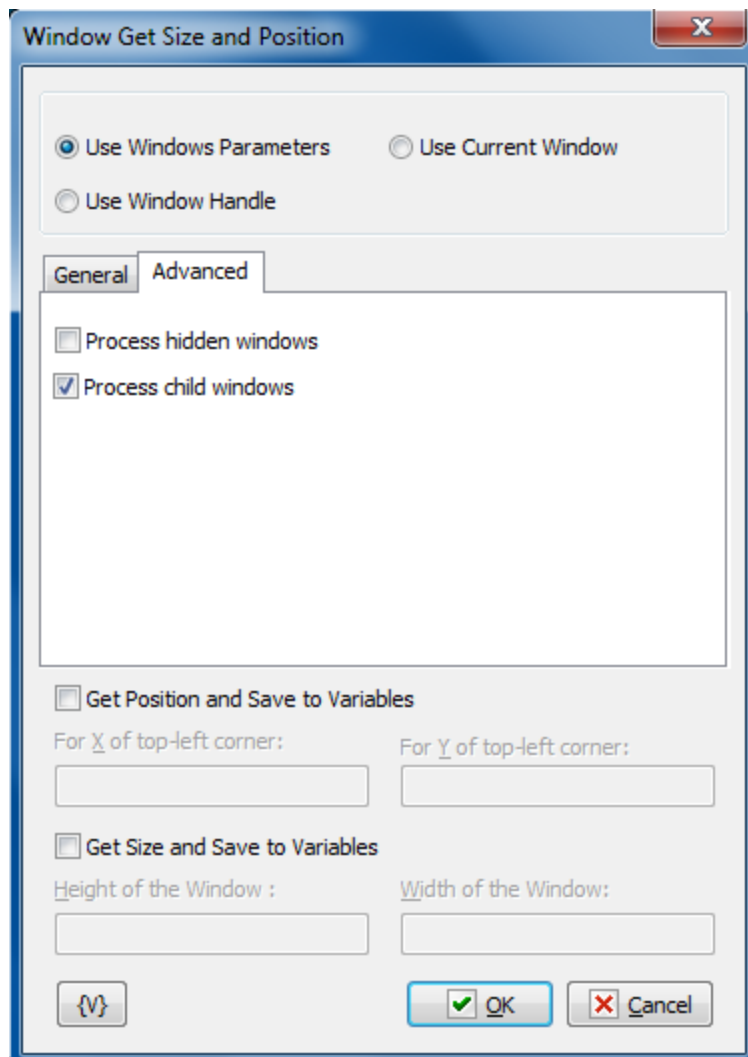
This action allows to save the sizes and position of the specified window into variables for further processing.

The action can save into separate variables such values as:

- The horizontal coordinate of the left top corner.
- The vertical coordinate of the left top corner
- The height of the window.

- 
- The width of the window.





### Window Selection Mode

- Use Window Parameters. In this case it is necessary to specify parameters of a required window: Window Caption, Window Class, compliance mode, etc. (see the description below).
- Use Window Handle. In this case it is necessary to specify a Window Handle. Handle is a number, an identifier of a window. It can be got, for example, in the action "Window loop". The Window Handle uniquely identifies the window in a system, so the rest of window parameters have no sense.
- Use Current Window. In this case a current active window is used. For example, if the necessary window was activated in the previous actions of a task, there is no need to specify window parameters again. Just use the current active window.

### Use Window parameters

#### Window Caption.

If you want to search for a window by means of its caption, switch on this checkbox and input the expected window caption. You can input only part of the caption. In this case the action will check availability of the input part in the window caption.

**Caption Must Match Exactly.**

If you need the exact coincidence of a window caption, switch on this checkbox.

**Window Class.**

If you want to search the window by a window class, switch on this checkbox and input the expected name of the window class. You can input only part of the name. In this case the action will check availability of the input line in the name of the window class.

**Class Name Must Match Exactly.**

If you need the exact coincidence of a window class, switch on this checkbox. Usually it is worthwhile to search by the whole class name, because a window class is permanent and does not change its value during the "life" of the window, unlike a caption.

**Select Window on the Screen.**

When pressing this button, you turn on the mechanism of selection a window on the screen. The window, which the mouse pointer points on, highlights with a red frame. Meanwhile, you can use the mouse to perform some actions. To select the necessary window, press *Ctrl*. To cancel - press *Esc*. When selecting a window, the fields Window Caption and Window Class are filled in automatically.

**Window Handle.**

Input the expression containing a window handle. This parameter is used for the mode Use Window Handle. A window handle is just a number, identifier of a window in a current session of system work.

**Get Position and Save to Variables**

Turn on this checkbox to save the position of the left top corner of a window.

**For X of Top-left corner**

Input the variable name for the horizontal coordinate (X) of the corner from the left edge of the screen.

**For Y of Top-left corner**

Input the variable name for the vertical coordinate (Y) of the corner from the left edge of the screen.

**Get Size and Save to Variables**

Turn on this checkbox to save the height and the width of the window.

**Height of the window**

Input the variable name for the height of the window.



**Width of the window**

Input the variable name for the width of the window.

## 4.15 System Functions

### 4.15.1 Start Service

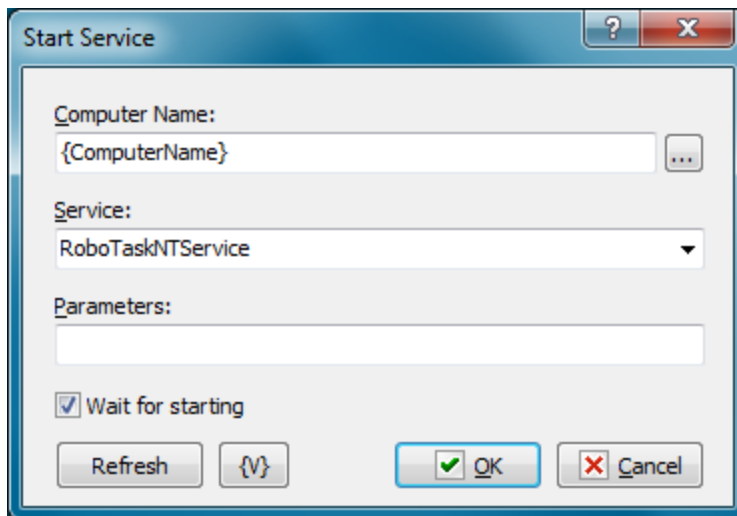
**Important**

- This action is not available on Windows 9x/Me.

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 have not 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**

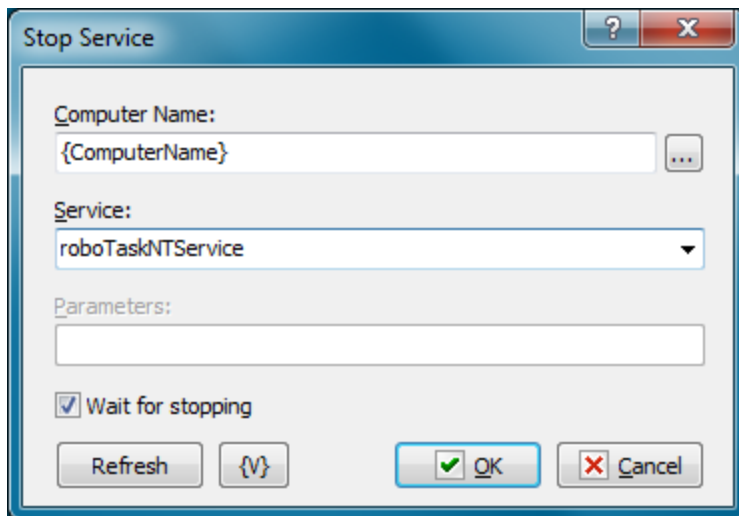
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.15.2 Stop Service

**Important**

- This action is not available on Windows 9x/Me.

Stops a service already installed and started on the local or remote server or workstation. For more information, see [Start Service](#).<sup>270</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.

### 4.15.3 Service status

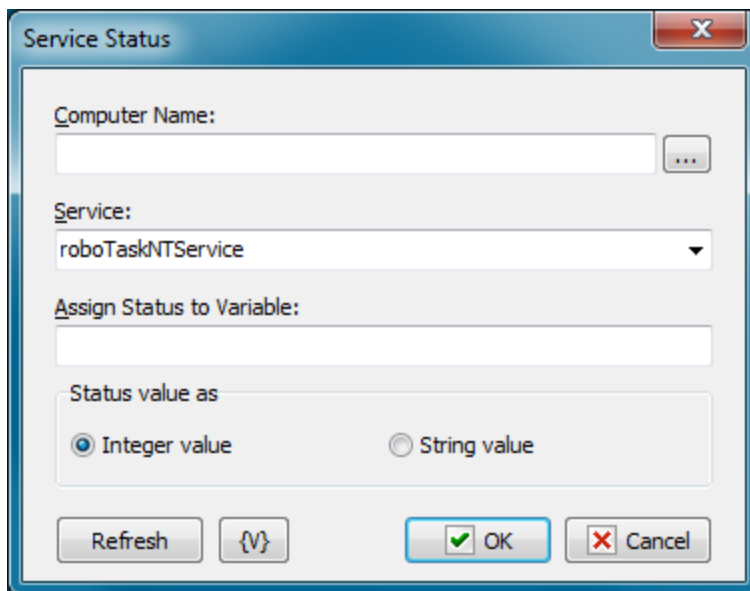
#### Important

- This action is not available on Windows 9x/Me.

This action allows to read the status of a specified system service and to assign a value to a variable.

The value of a service status can be assigned as a line or a number. The values correspond to the Microsoft documentation. There are 7 statuses:

- **SERVICE\_STOPPED = 1** - The service is not running.
- **SERVICE\_START\_PENDING = 2** - The service is starting
- **SERVICE\_STOP\_PENDING = 3** - The service is stopping.
- **SERVICE\_RUNNING = 4** - The service is running.
- **SERVICE\_CONTINUE\_PENDING = 5** - The service continue is pending.
- **SERVICE\_PAUSE\_PENDING = 6** - The service pause is pending.
- **SERVICE\_PAUSED = 7** - The service is paused



#### Computer Name

Specify the computer name of a local network. The usage of variables is allowed. For example, {ComputerName}. If the name of computer is not specified (an empty line), a local computer is exploited.

#### Service

Specify the name of a service. The name can be typed or selected from a dropdown list. Usage of variables in the name of a service is allowed.

### Assign Status to Variable

Assign the name of the variable where it is necessary to write the service status.

### Status Value as

Specify the form of the value which is necessary to get.

- **Integer value** – the number from 1 to 7.
- **String value** – one of possible values:
  - SERVICE\_STOPPED,
  - SERVICE\_START\_PENDING,
  - SERVICE\_STOP\_PENDING,
  - SERVICE\_RUNNING,
  - SERVICE\_CONTINUE\_PENDING,
  - SERVICE\_PAUSE\_PENDING,
  - SERVICE\_PAUSED

### Variable

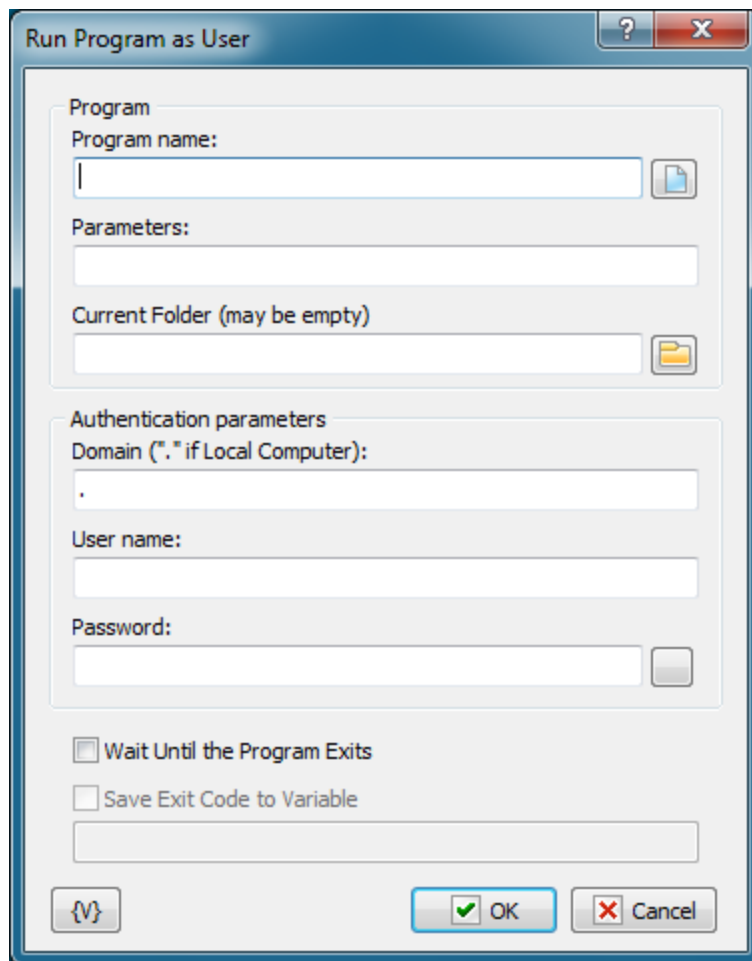
Allows you to add a variable to the action or triggering event that you selected. You must place the cursor in a edit box in the settings dialog window of the action or event and then click the {V} button.

## 4.15.4 Lock Workstation

This action submits a request to lock the workstation's display. Locking a workstation protects it from unauthorized use.

## 4.15.5 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.

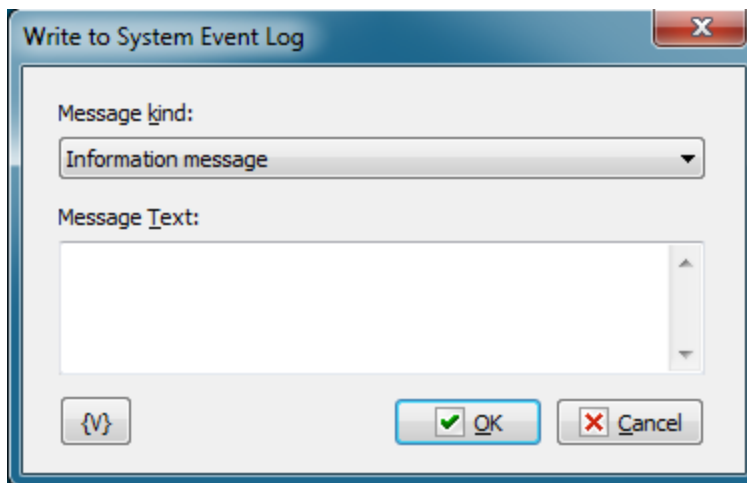
### Variable

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.

## 4.15.6 Write to Event Log

The action writes specified message to the System Event Log.

For application mode, the source of the message is the *RoboTask Application*. For system service mode, the source of the message is the *RoboTask Service*.



### Message Kind

Select the message text from the drop-down list:

- Information message
- Warning message
- Error message

### Message Text

Enter the text of the message text as a multi-line text. You can use RoboTask variables in the text.

---

## 4.16 RoboTask Commands

These actions perform RoboTask commands.

### 4.16.1 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.

### 4.16.2 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.

### 4.16.3 Exit RoboTask

Closes RoboTask. No specific settings are associated with this action.

### 4.16.4 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.

### 4.16.5 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.

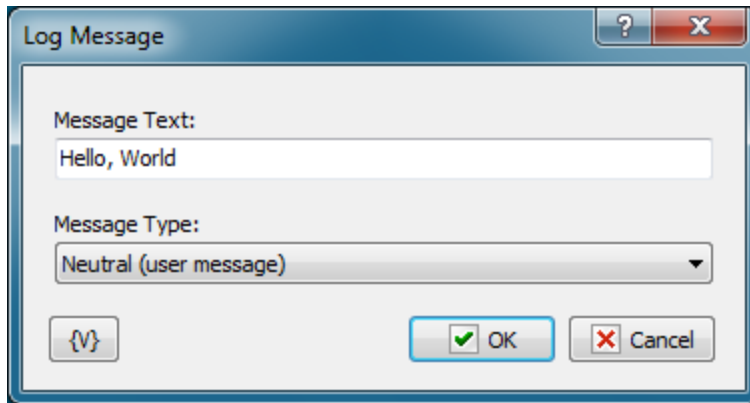
### 4.16.6 Log Message

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 NNNNNNNN.tsk where NNNNNNNN is its ID number. The log file name is NNNNNNNN.log.

### Important

- You can set options for logging on the [Advanced tab](#) 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.

## 4.16.7 Restart RoboTask

Sometimes you need to restart RoboTask to free up system resources. RoboTask has a restart feature in the main window menu (Task -> Restart application). But this action is designed to restart automatically in a task.

This action syncs with app mode and works according to different algorithms for app mode and system service mode:

- In app mode, it restarts itself.
- In service mode, it restarts RoboTask Service.

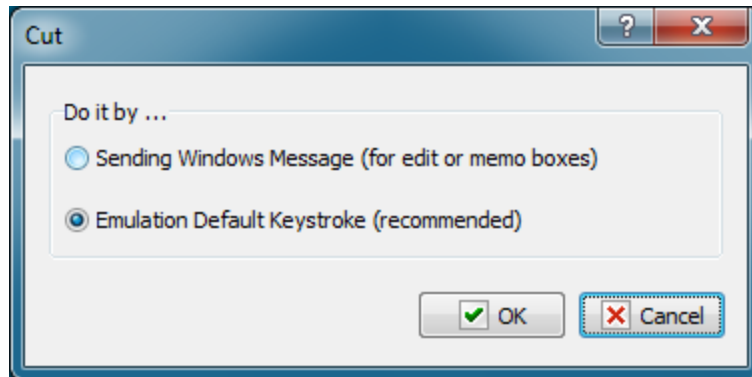
This action doesn't require additional configuration.



## 4.17 Clipboard

### 4.17.1 Cut to Clipboard

Cuts the selected text fragment (image) into the system clipboard from the active Windows application.



#### Do it by...

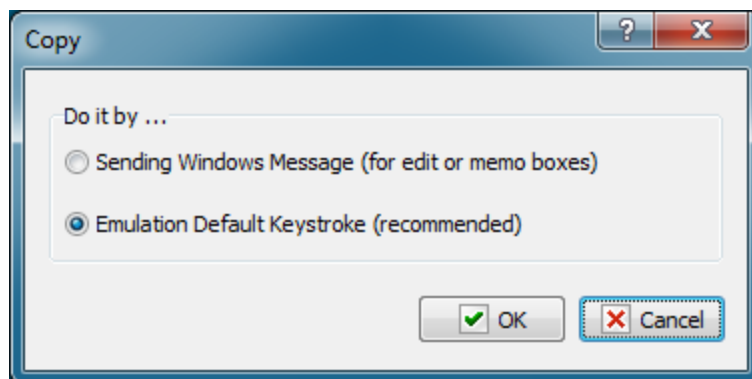
There are two ways to cut a fragment:

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.

### 4.17.2 Copy to Clipboard

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:

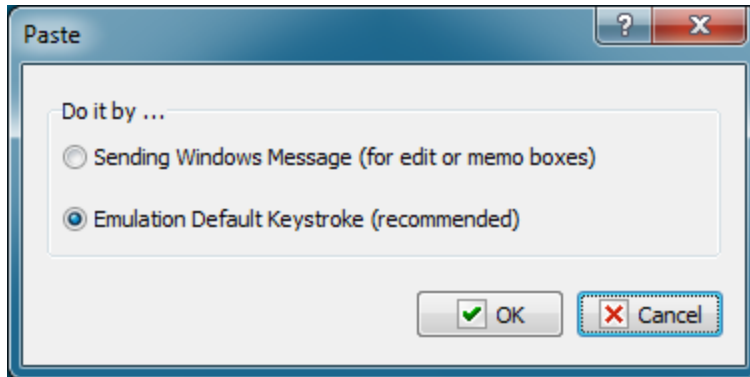
1. by sending a Windows message to the corresponding edit element

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.

### 4.17.3 Paste from Clipboard

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

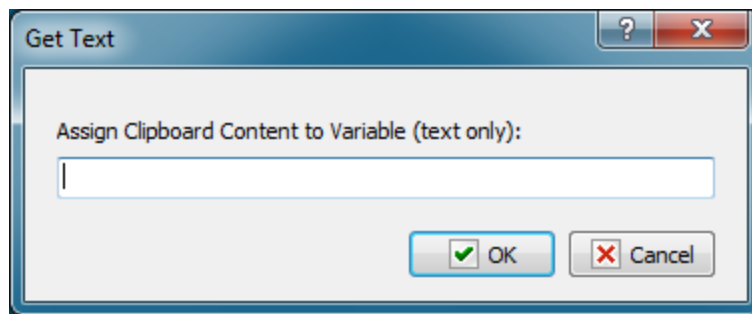
It is recommended to use the second method because it is a universal mechanism for any application and any data type.

### 4.17.4 Clear Clipboard

Clears the system clipboard. It has no parameters.

### 4.17.5 Get Text from Clipboard

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.

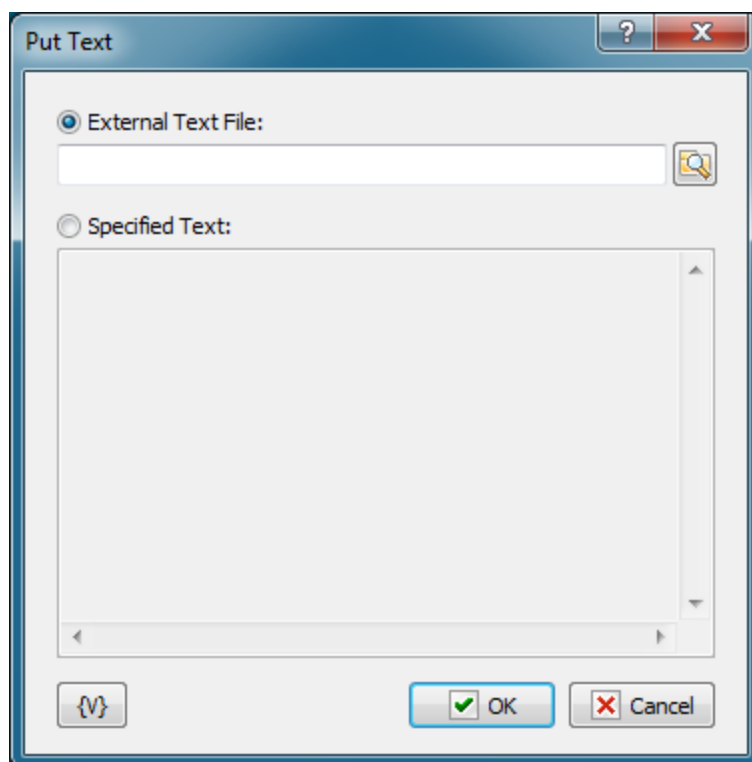


### Assign Clipboard Content to variable

Specify the variable the text from the system clipboard should be assigned to.

## 4.17.6 Put Text to Clipboard

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

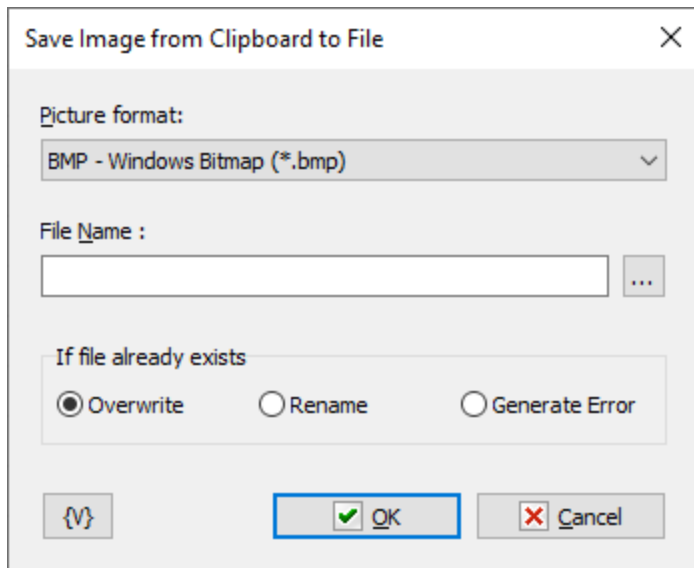
Type the fragment that should be copied into the clipboard.

### 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.

#### 4.17.7 Save Image from Clipboard

You can save image form system clipboard to file. Of course this is possible when the clipboard contains some image. Otherwise the action raises the error "Clipboard doesn't contain an image"



##### Picture format

There are four formats available:

- **BMP** – standard Windows bitmap;
- **Jpeg** – Joint Photographic Experts Group format;
- **PNG** – portable network graphics format
- **GIF** – Graphics Interchange Format

##### File name

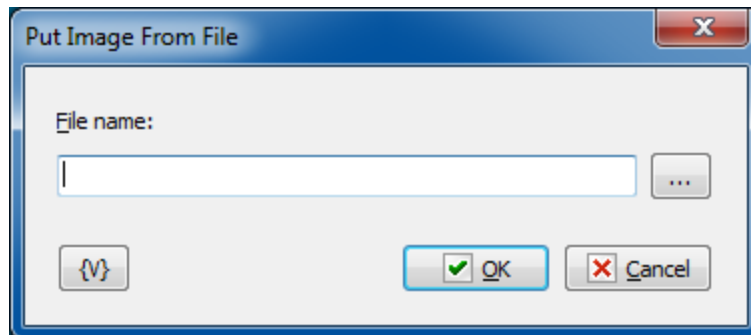
Input the file name where you want to save the image. Variables are allowed.

##### If file already exists

The action if file already exists.

#### 4.17.8 Put Image from File to Clipboard

This action put the image from file to system clipboard as image.

**File name**

Input the file name of the image. You can use these kinds of images:

**BMP** - windows bitmap

**JPEG** - Jpeg files. Such files may have the extension *JPG* or *JPEG*

**PNG** - portable network graphics

**GIF** - Graphics Interchange Format

**ICO** - system icon file

## 4.18 Mouse

Actions for simulation of mouse movements and clicks.

### 4.18.1 Move to an 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 press **"Select window and element manually"** button and select the object you need. While you are moving the mouse pointer, elements recognized by RoboTask are highlighted with a frame. To select press CTRL key, to cancel operation press ESC key.

When you selected the object 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

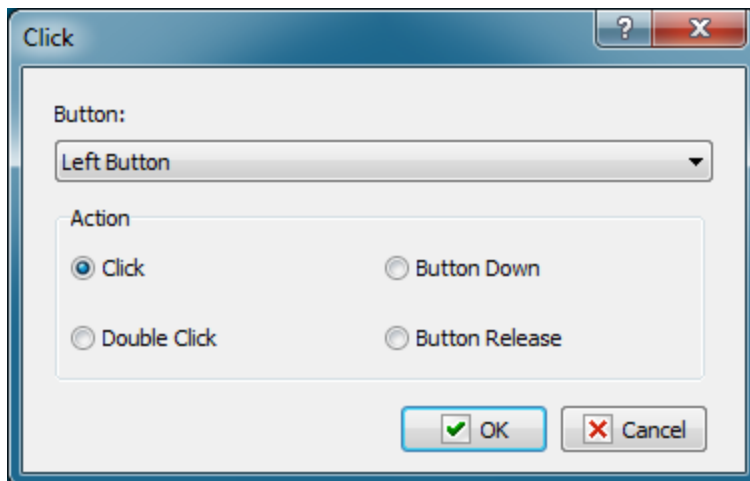
Specify the position of the mouse pointer relatively to the control element.

### 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.

## 4.18.2 Click

Emulates a mouse click in the current position.



### Button

Select the mouse button:

- Left Button
- Right Button
- Middle Button.

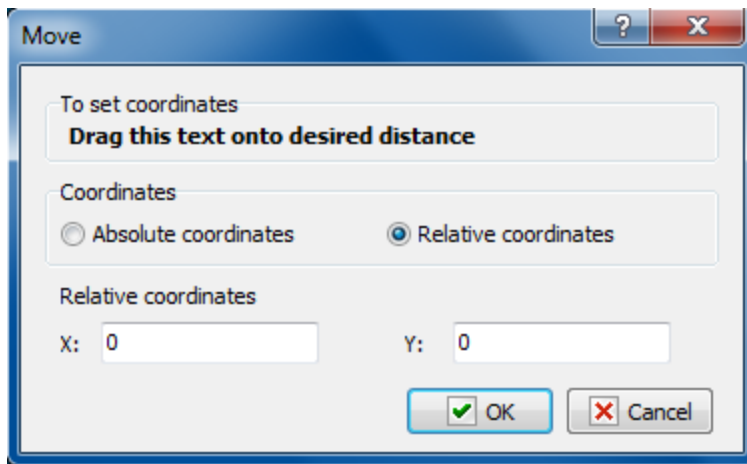
### Action

Select the action. There are 4 variants to choose from:

- Click
- Double Click
- Button Down
- Button release

### 4.18.3 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.

### 4.18.4 Move Mouse to Image

Sometimes it is impossible to find the necessary element by the Caption or ClassName. The classic example is the web page in your browser. From point of view of the system, the web page is a single block with drawn on it elements and a text.

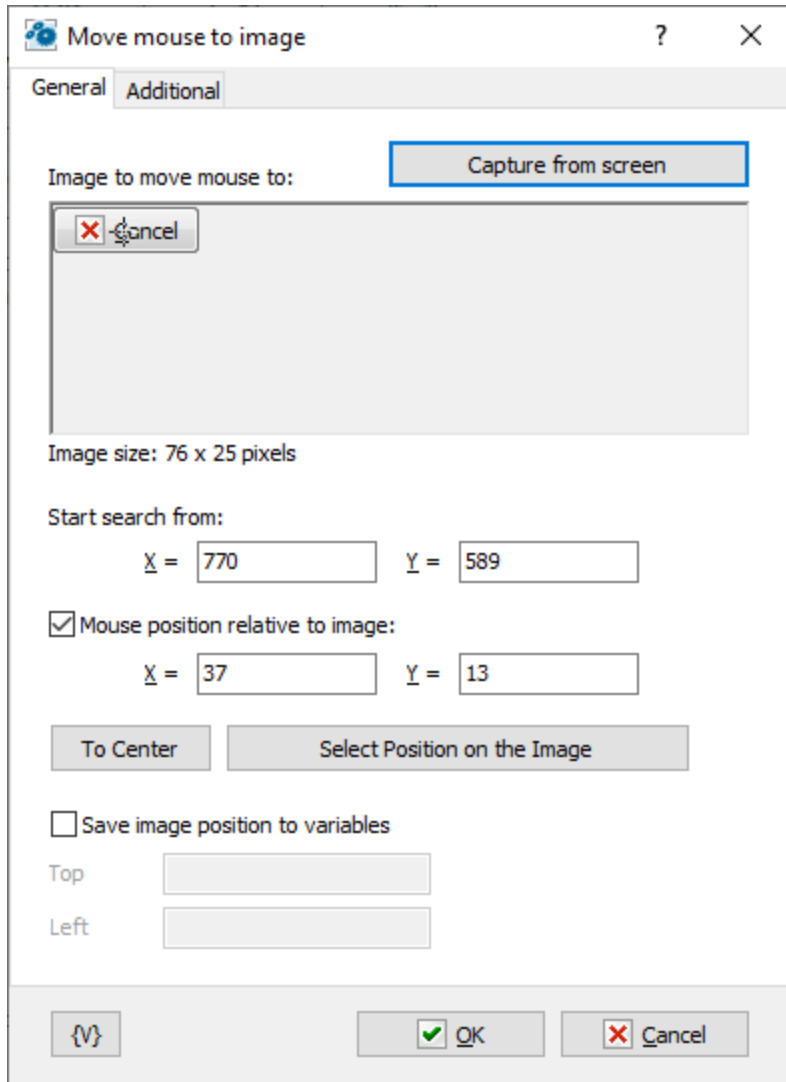
This action allows you to find a visible element on the screen by its image and move mouse pointer in the necessary dot of the found element.

Notice: it is necessary the searched element should be visible wholly on any place of a desktop. In another case, the action will not find it. For example: if this element is overlaid



with another window or is outside the desktop. In this case, to correct the situation, it is necessary to use the actions [Window Command](#)<sup>[252]</sup> and/or [Window Set Size and position](#)<sup>[259]</sup>, to make the necessary element visible.

## General tab



### The search Image.

The image of the element which you need to find on the screen. You can select the necessary area on the screen pressing the button "Select the Image on the Screen". The editor will take a screenshot of the current screen and propose to select the necessary area on the received image.

### Initial position of Element for Searching.

The position of the top-left corner of an element. The action begins to search the image on the desktop (screen) from this position. The nearer the element is to the initial position, the

less time it takes to search for it. Generally speaking, the initial position can be any. If the element is on the screen, the action finds it, but the searching can take several seconds.

### Put Mouse Cursor to Position on the Element

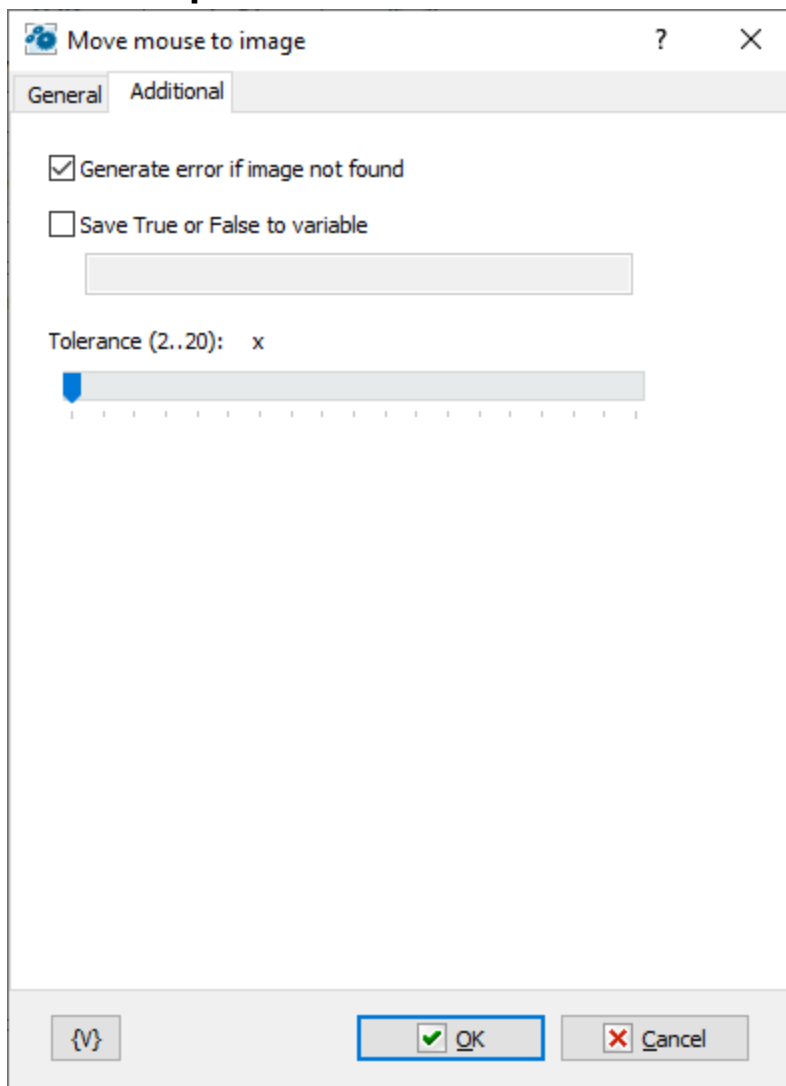
Specify the coordinates of the position of the mouse cursor relative to the upper left corner of the found element. You can specify the coordinates in the entry fields or on the image directly by right clicking in the necessary position on the element. You can also select the coordinates of mouse cursor pressing the button "Select Position On the Image".

The selected position is indicated on the image with a cross of an inverse color.

### Save image position to variables

Check this box and enter variable names to save the position of the top left corner of the image.

### Additional parameters



### Generate Error if Object Is Not Found

Switch on this checkbox if you want to generate an error when an object is not found. This option is enabled by default, but you can change it when you need, if it is necessary.

### Save True or False to variable

If you want to save the result into a variable, switch on this checkbox and input the variable name. The action will save the value *True* if the element is found and *False* if not.

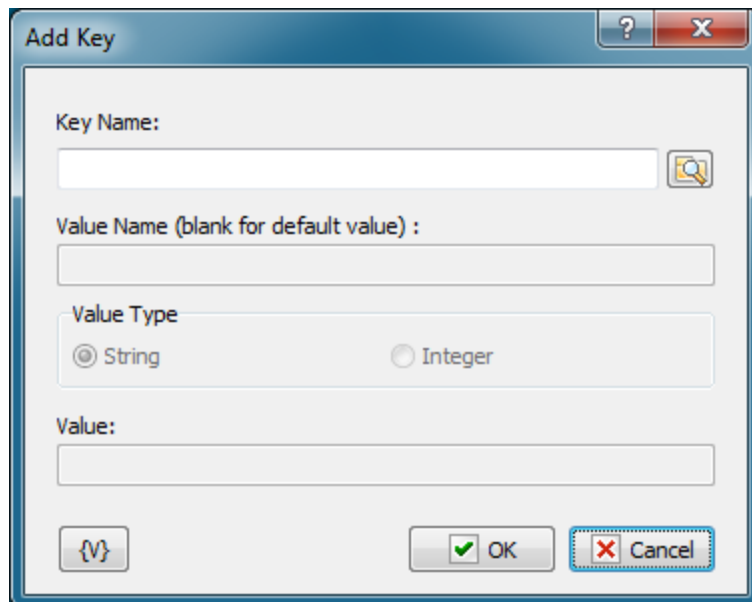
### Tolerance

Match accuracy. The smaller the value, the more accurate the match. You need to choose the optimal value. The tolerance must always be greater than zero due to the inaccuracy of displaying the elements on the screen.

## 4.19 Registry

### 4.19.1 Add Registry Key

Adds a key (branch) to the registry.



#### Key Name

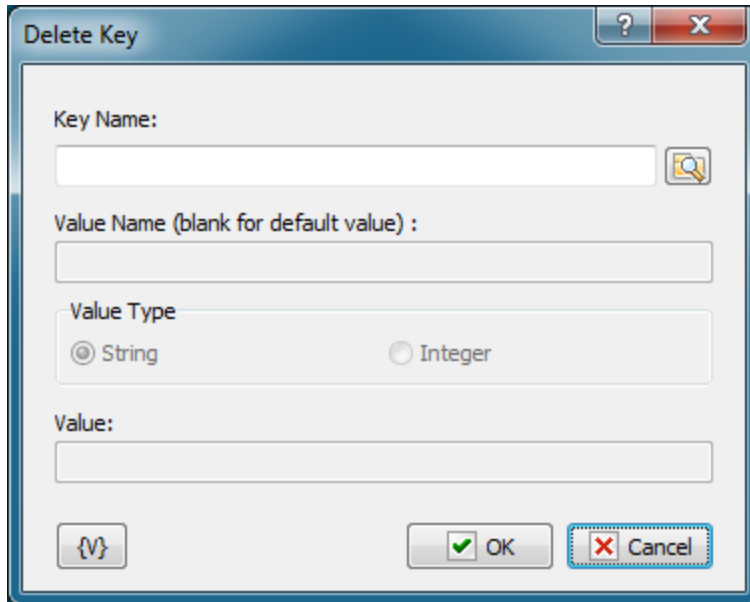
Name of the key to be added. You can use the dialog box to select the key interactively; to open it click the button to the right of the text box.

#### 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.

### 4.19.2 Delete Registry Key

Deletes the specified key (branch) from the registry. The specified key will be deleted together with all its subkeys and entries.

**Key Name**

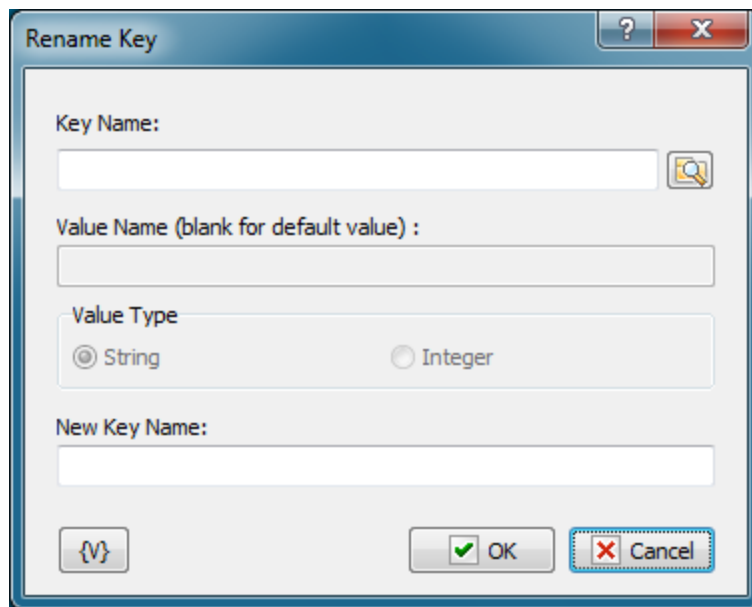
Name of the key to be deleted. You can use the dialog box to select the key interactively; to open it click the button to the right of the text box.

**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.

### 4.19.3 Rename Registry 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**

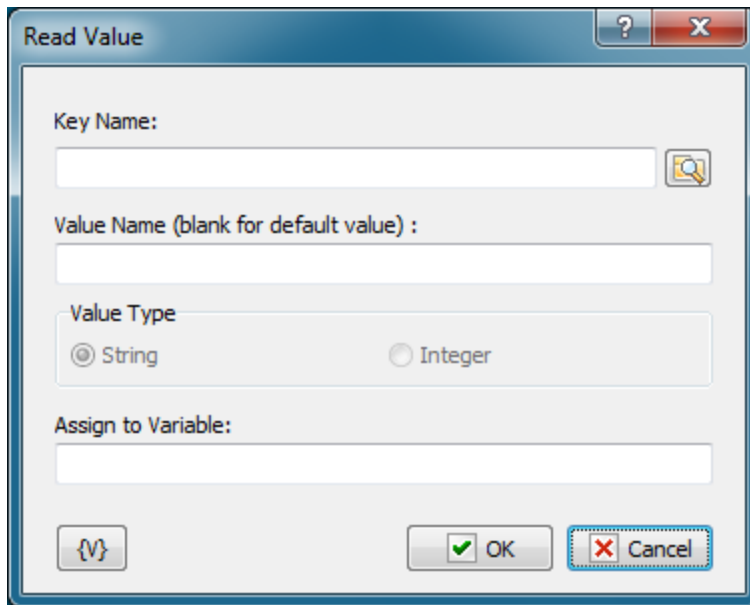
A new name that will be given to the specified key.

**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.

#### 4.19.4 Read Registry 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**

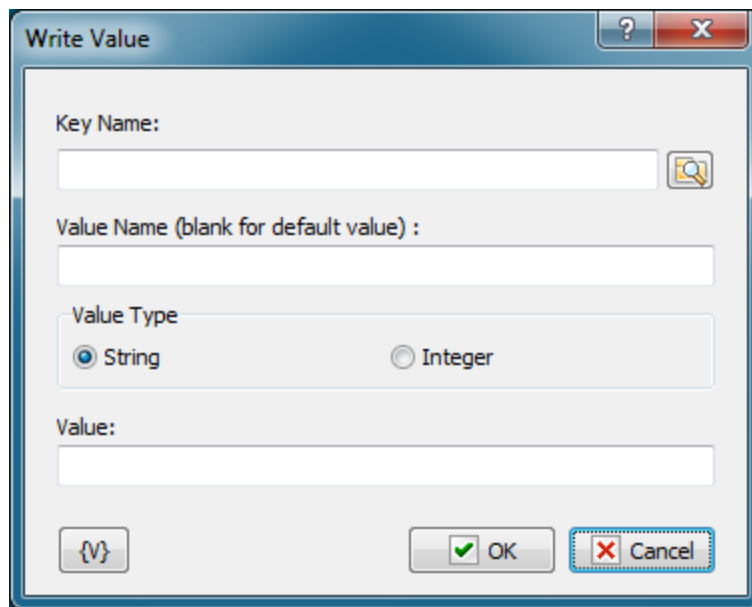
Name of the user variable that should be assigned the value we have read.

**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.

### 4.19.5 Write Registry 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**

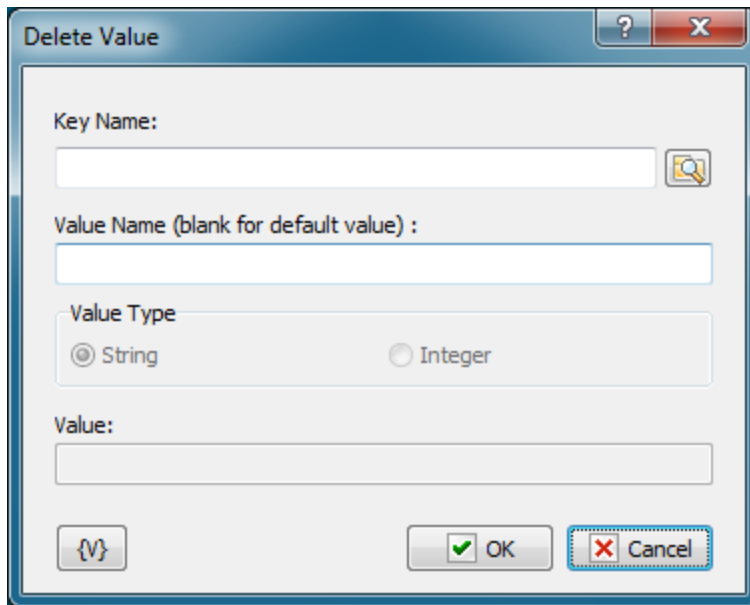
The value to be written.

**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.

### 4.19.6 Delete Registry 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.

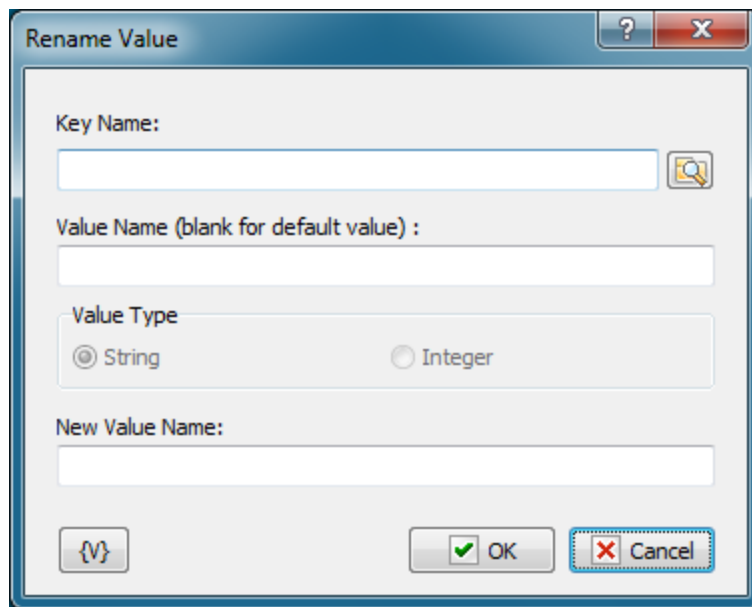
**Variable**

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.

#### 4.19.7 Rename Registry 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**

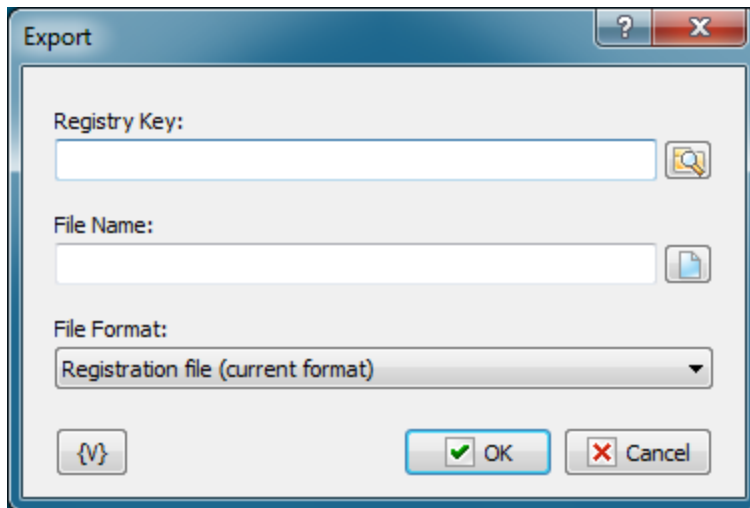
New name that will be assigned to the specified value.

**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.

#### 4.19.8 Export from Registry

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>[295]</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.

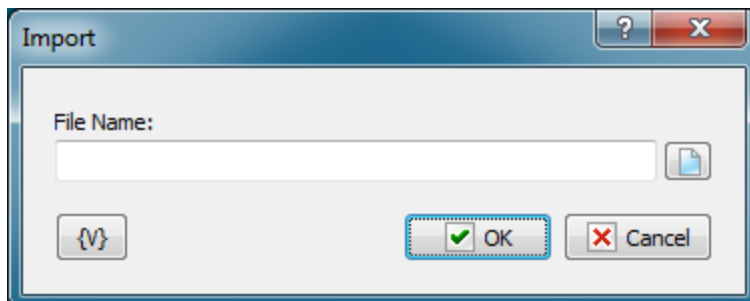
- Registration file (current format) - current default format for the installed operating system. For Windows 2000 and higher the format is Unicode text.
- Win9/NT4 Registration File - old format that provides compatibility with Windows 9x/NT4. This format has a structure of a regular text file.

### 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.

## 4.19.9 Import to Registry

Imports the export file to the registry. The export file should be prepared with the **REGEDIT.EXE** system utility or the [Export](#)<sup>[294]</sup> action.



**File Name.**

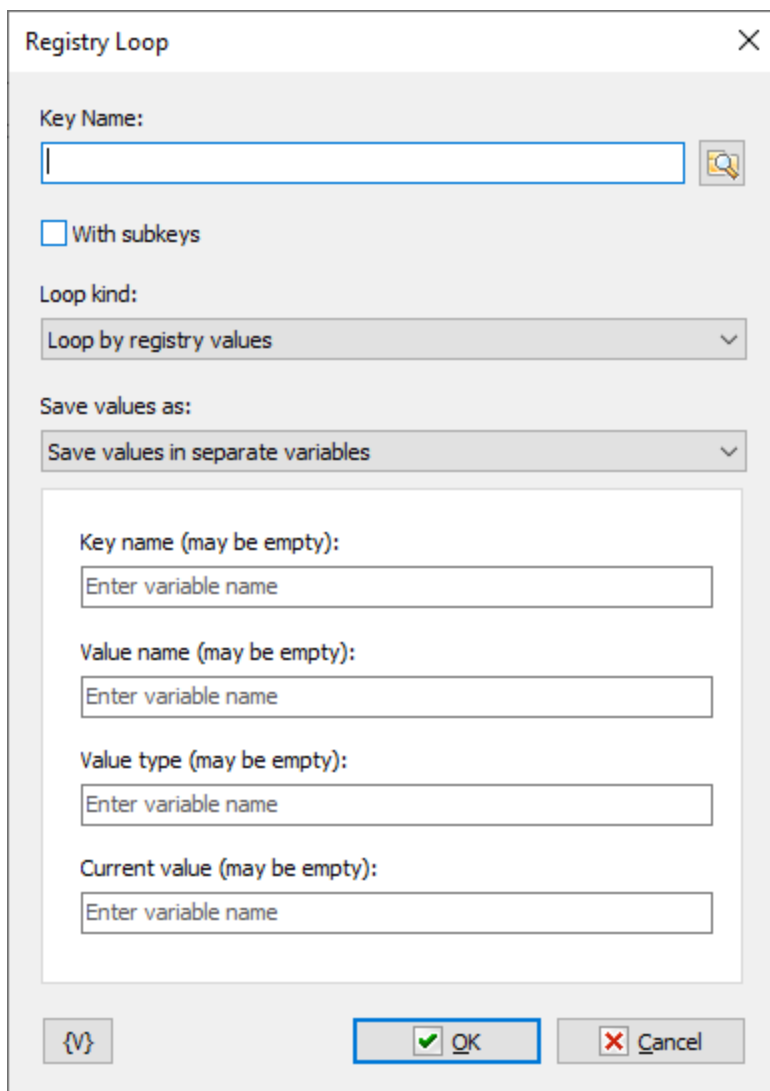
Name of the file to be imported.

**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.

### 4.19.10 Registry loop

The action allows you to enumerate keys or values from the registry.



The image shows a dialog box titled "Registry Loop" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Key Name:** A text input field with a search icon (magnifying glass) to its right.
- With subkeys**
- Loop kind:** A dropdown menu currently showing "Loop by registry values".
- Save values as:** A dropdown menu currently showing "Save values in separate variables".
- A large white-bordered area containing four text input fields, each with a placeholder "Enter variable name":
  - Key name (may be empty):**
  - Value name (may be empty):**
  - Value type (may be empty):**
  - Current value (may be empty):**
- At the bottom left, a button with the symbol {V}.
- At the bottom center, a button with a green checkmark and the text "OK".
- At the bottom right, a button with a red X and the text "Cancel".

**Key name**

The start key name. You can use the dialog box to select the key interactively. To open it click the button to the right of the text box.

### With subkeys

If you want to process all sub-keys recursively switch on this checkbox

### Loop kind

There are two options

- Loop by registry values
- Loops by keys only

### Save values as

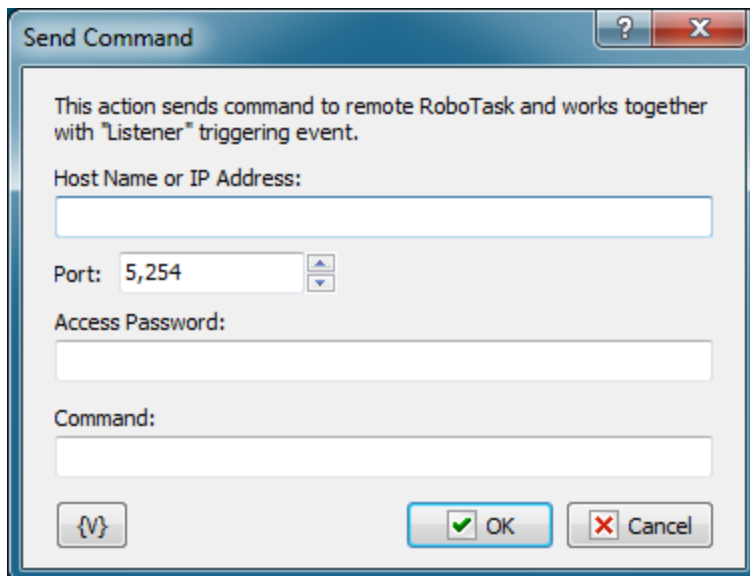
You can save values of the loop in separate variables or in one variable as JSON

## 4.20 Network

### 4.20.1 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>[545]</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.*



The image shows a Windows-style dialog box titled "Send Command". The dialog has a title bar with a question mark and a close button (X). The main content area contains the following text: "This action sends command to remote RoboTask and works together with 'listener' triggering event." Below this text are four input fields: "Host Name or IP Address:" (a text box), "Port:" (a spinner box with "5,254" displayed), "Access Password:" (a text box), and "Command:" (a text box). At the bottom of the dialog, there is a button with a curly brace icon, and two buttons labeled "OK" (with a green checkmark) and "Cancel" (with a red X).

#### 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**

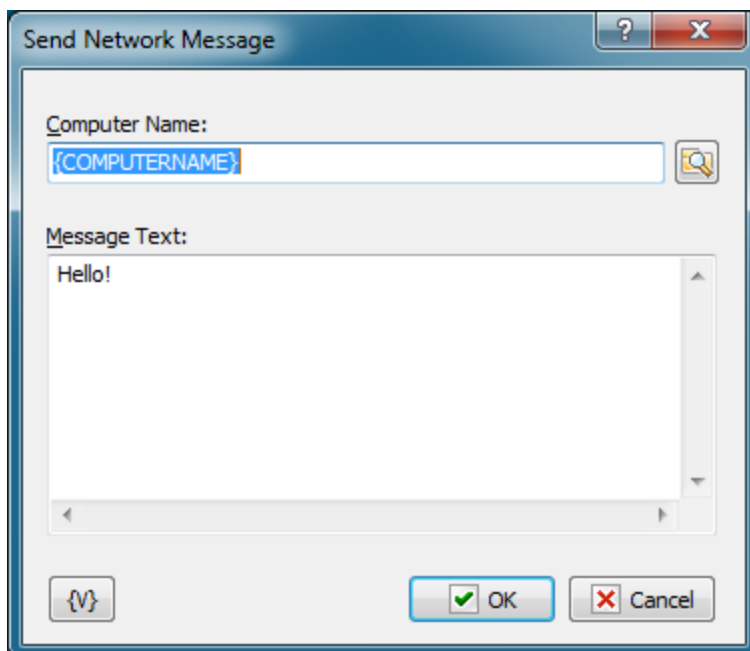
Specify the string (command) that should be sent to remote RoboTask. It can be any string. The "Listener" triggering event on the remote computer will analyze the string.

**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.

## 4.20.2 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**

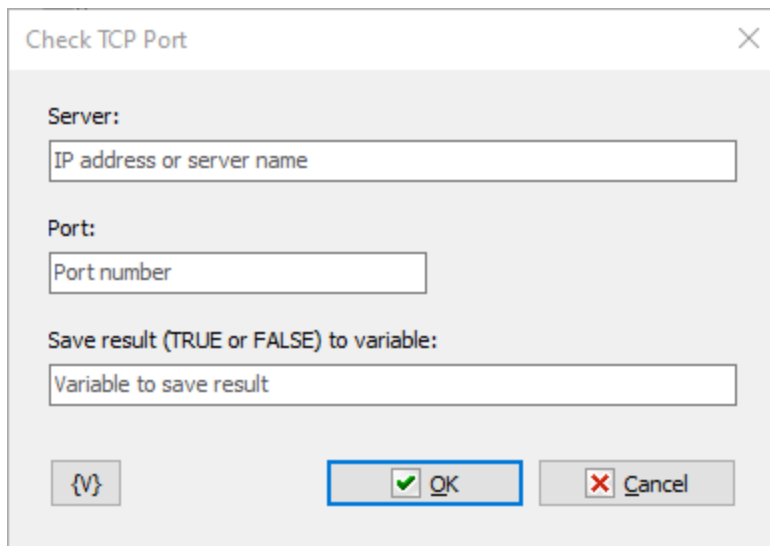
Specifies the text of the message to be sent.

**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.

### 4.20.3 Check TCP port

The action checks TCP port and returns TRUE if the port is open or FALSE if not.

**Server**

Enter IP address or server name

**Port**

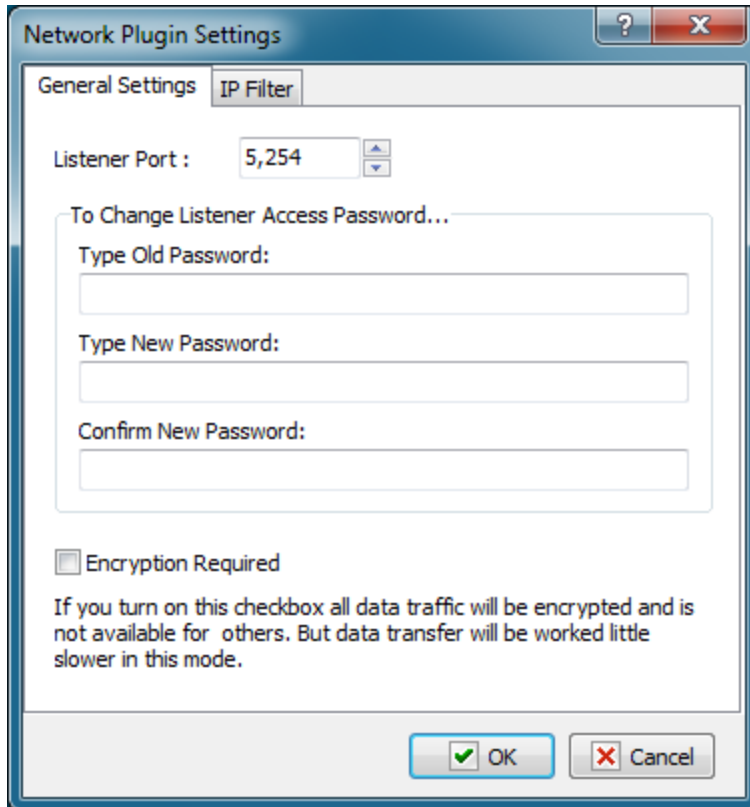
Enter port number

**Save result to variable**

Enter the name of the variable in which you want to save the result.

## 4.20.4 Network Plugin settings

The settings of the Network plugin are related to the "[Listener](#)<sup>545</sup>" triggering event and set the global parameters of the "[Listener](#)<sup>545</sup>".



### Listener port

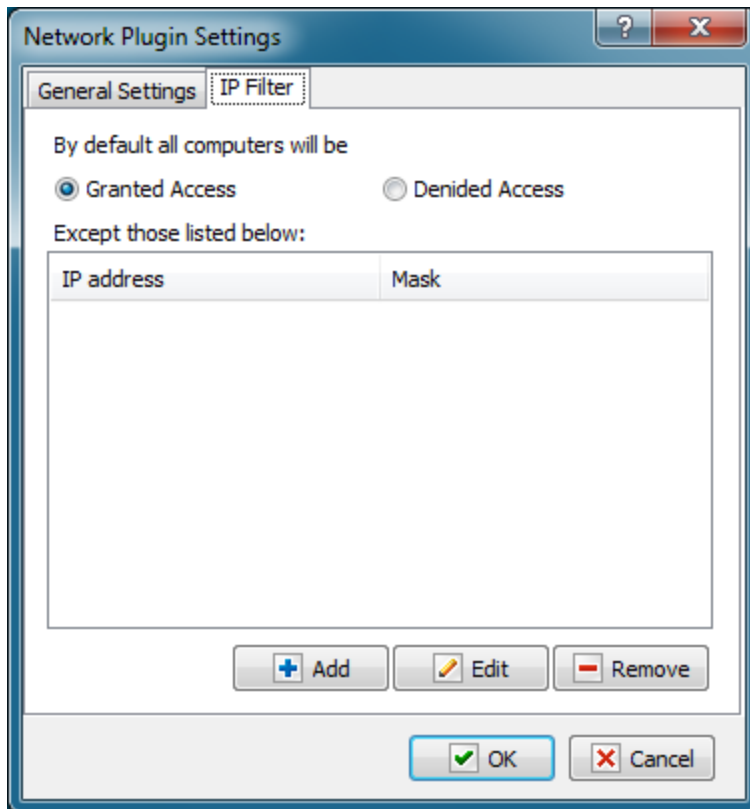
Specify the port the "Listener" will listen to. The default is 5254, but it can be changed for any other available port.

### 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.

## 4.21 Email

This plugin contains email action which work with POP3 and IMAP protocols.

### 4.21.1 POP3 Open

Opens a connection with the POP3 server. This action should be performed before you perform the "[POP3 Mail Loop](#)<sup>[303]</sup>", "[POP3 Download Email](#)<sup>[306]</sup>" and "[POP3 Delete Email](#)<sup>[307]</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>[303]</sup>" action).



The screenshot shows a 'POP3 Open' dialog box. It has a title bar with a question mark and a close button. The main area contains the following elements:

- Host:** A text input field.
- Port:** A text input field containing '110' and a 'Use SSL' checkbox.
- SSL/TLS flags:** A group box containing five checkboxes: 'SSL2' (unchecked), 'SSL3' (unchecked), 'TLS 1.0' (checked), 'TLS 1.1' (checked), and 'TLS 1.2' (checked).
- User Name:** A text input field.
- Password:** A text input field with a small icon to its right.
- Buttons:** At the bottom, there is a button with '{V}', an 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

**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.

**SSL/TLS Flags**

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

- Use SSL2
- Use SSL3
- Use TLS 1.0
- Use TLS 1.1
- Use TLS 1.2

**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.

#### **4.21.2 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.

#### **4.21.3 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.

POP3 Mail Loop

Genetral    Advanced parameters

Assign Message Parameter to Variable

Order number:

Subject:

Sender (FROM):

Recipient (TO):

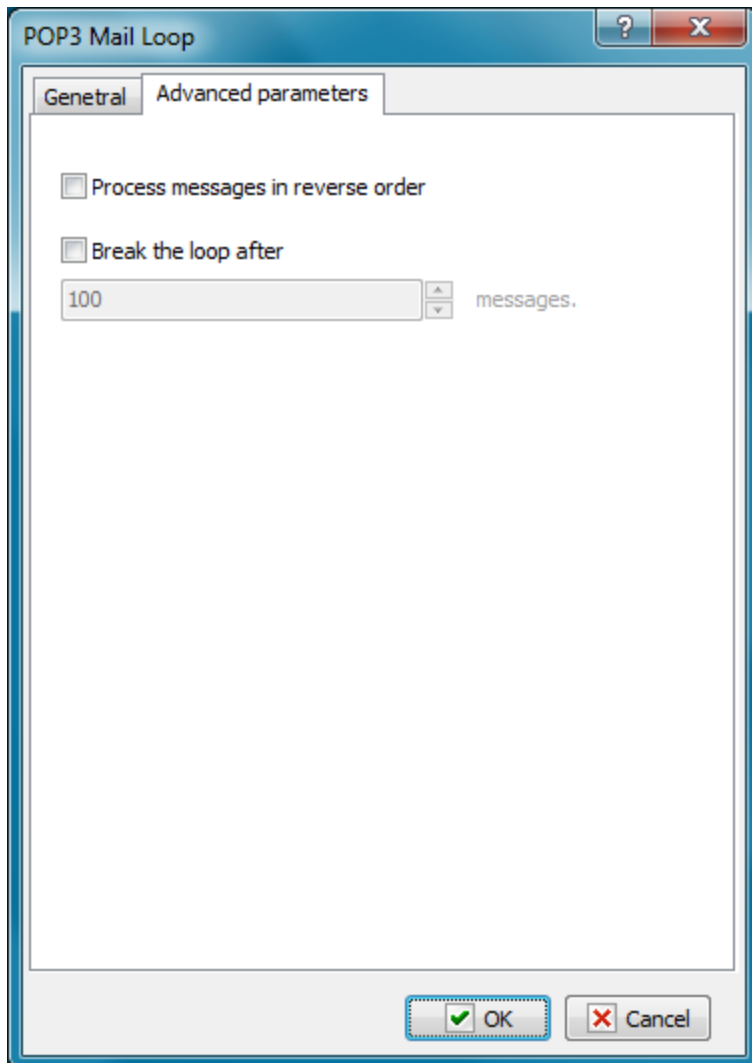
Carbon Copy (CC):

Message Date:

Message Size (bytes):

Message Priority (Low/Normal/High):

OK     Cancel

**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>[306]</sup>" and "[POP3 Delete Email](#)<sup>[307]</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.

**Process messages in reverse order**

You can process the list of messages in reverse order. This is useful when you work with a large list, because new messages are at the end of the list.

**Break the loop after**

Turn on this checkbox and specify the number of messages if you want to limit the cycle.

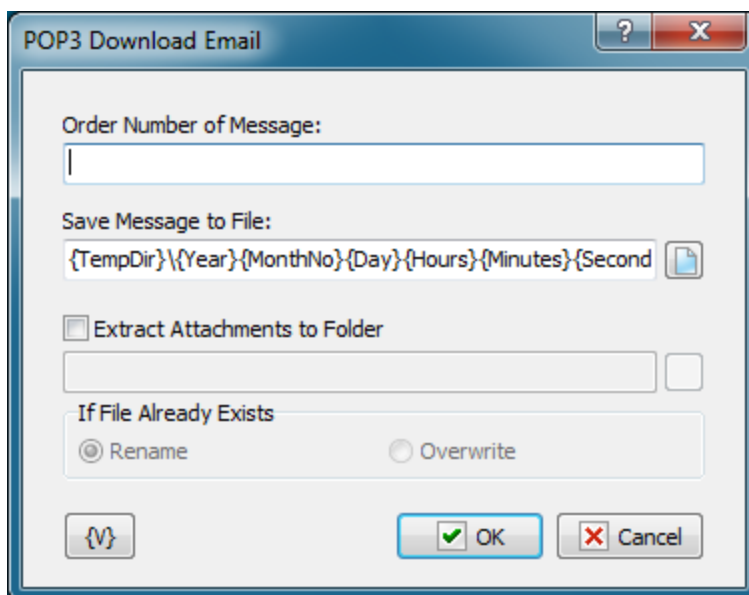
**Note:**

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>

#### 4.21.4 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.

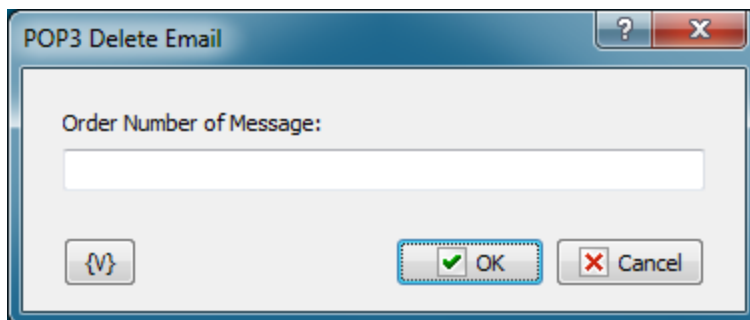
- Rename - in this case a suffix with an order number will be added to the file name;
- Overwrite - the file will be saved with the original name and the old file will be overwritten.

### 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.

## 4.21.5 POP3 Delete Email

Deletes the email message from the POP3 server.



### Order Number of Message

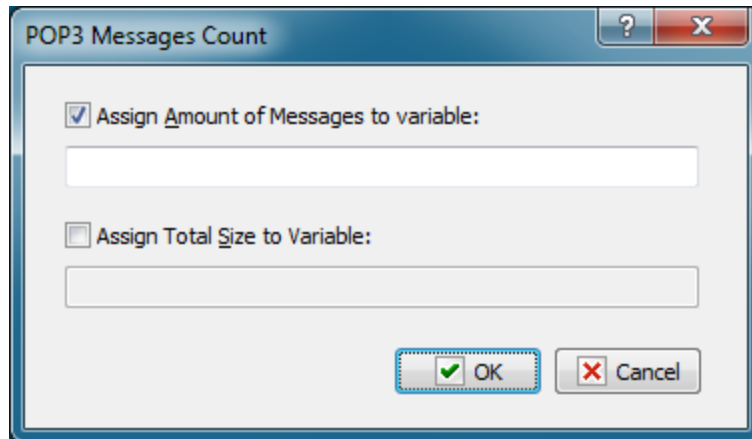
Order number of the message to be deleted. It can be generated in the "[POP3 mail loop](#)<sup>303</sup>" using a variable or you can set it manually. If the message number is invalid, a "No such message" error will occur when is action is carried out.

### 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.

### 4.21.6 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>301</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

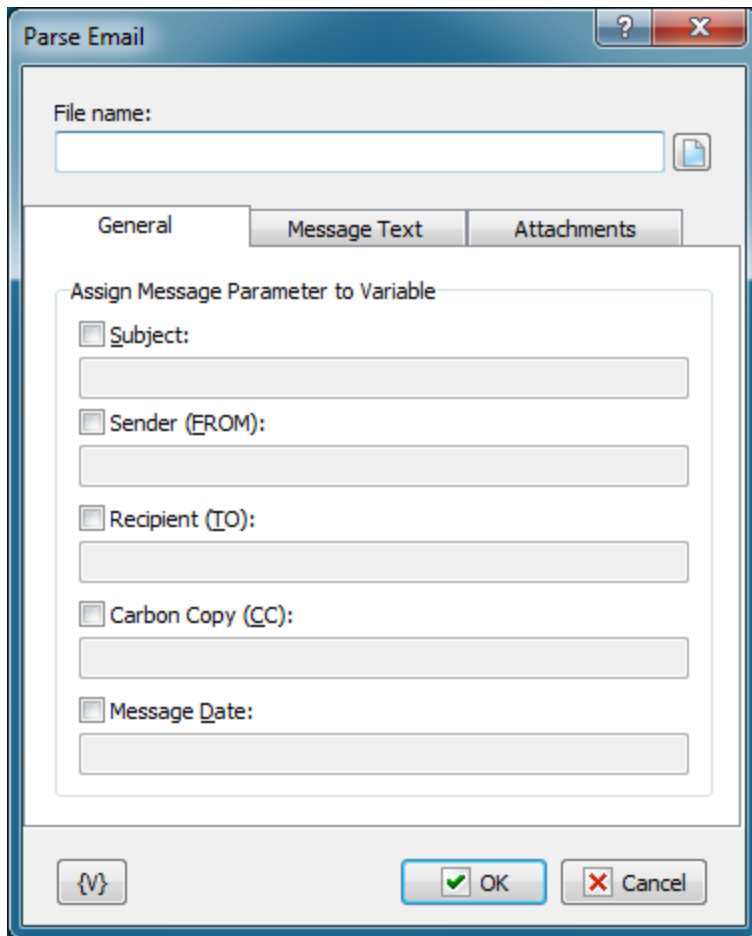
Select this checkbox if you want to save the total size of messages (in bytes) to a variable. You should enter the variable name into the field below.

While editing the step, you should specify at least one variable.

### 4.21.7 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)**

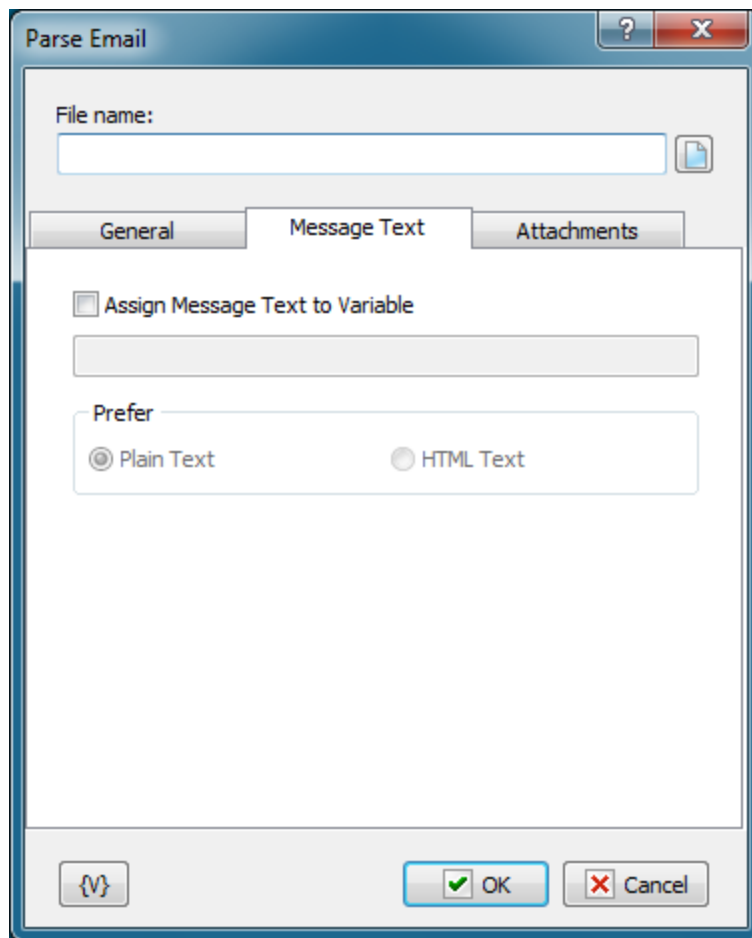
Specify the name of variable where the recipient's address should be saved. This field can contain several addresses separated with a comma or a semicolon.

**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.

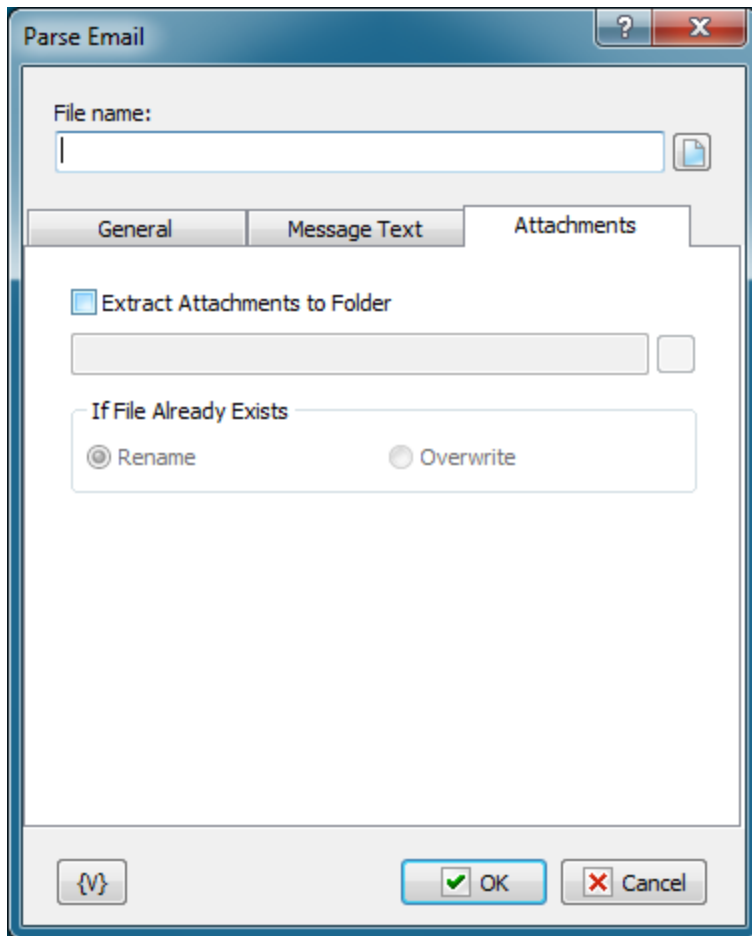
### Prefer

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>

### 4.21.8 IMAP Open

The action opens the connection with the IMAP server. All actions with the IMAP server have to begin from the opening of the connection. Otherwise it is impossible to use the actions of the IMAP. It is possible to perform only one connection with the IMAP within one task. If you need to switch over another IMAP server within a task, at first, you should close the current connection (see the [IMAP Close](#)<sup>313</sup>).

The screenshot shows a dialog box titled "IMAP Open". It contains the following fields and controls:

- Server:** A text input field.
- Port:** A text input field containing "143".
- Use SSL:** A checkbox.
- SSL/TLS flags:** A group box containing five checkboxes:
  - SSL2 (unchecked)
  - SSL3 (unchecked)
  - TLS 1.0 (checked)
  - TLS 1.1 (checked)
  - TLS 1.2 (checked)
- User Name:** A text input field.
- Password:** A text input field with a visibility icon (eye) to its right.
- Select Mail Folder:** A text input field containing "INBOX" and a dropdown arrow icon.
- Buttons:** A help icon (question mark in a circle), an "OK" button with a green checkmark, and a "Cancel" button with a red X.

#### Server

The name or the IP address of a server.

#### Port

The number of a port for connection. On default, the port of the IMAP is 143 for a simple connection and 993 for the SSL connection.

#### Use SSL

Need for using the SSL (Secure Socket Layer)

**SSL/TLS Flags**

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

- Use SSL2
- Use SSL3
- Use TLS 1.0
- Use TLS 1.1
- Use TLS 1.2

**User Name**

A username for authentication.

**Password**

A password for authentication.

**Select Mail Folder**

Specify the folder which is necessary to be open after opening the connection. The folder can be specified manually or selected from the list pressing the button on the right of entry field. On default, the folder INBOX (the folder of incoming mail) is specified. The composition and names of folders depend on a mail server.

Almost all IMAP actions work with the current folder of a mail server. So it is necessary to select the current folder before to start working with IMAP messages.

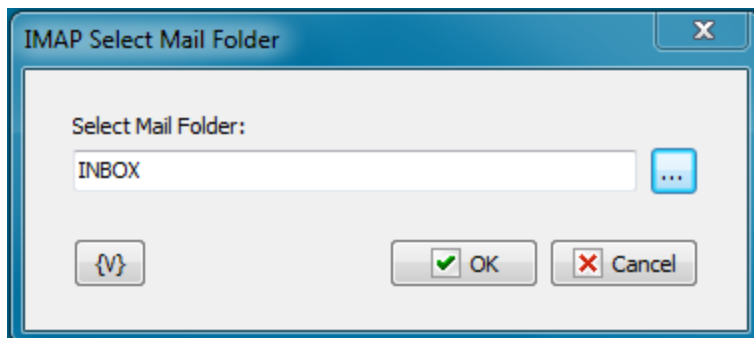
**4.21.9 IMAP Close**

The action closes the previously opened connection. If the connection has not opened, the action is ignored.

The action does not need any additional settings.

**4.21.10 IMAP Select Mail Folder**

The action is designed for changing of the current folder of the mail server during execution of a task.



---

### Select Mail Folder

Specify the name of a folder of the IMAP server. The folder name can be typed manually explicitly or using variables. Also the folder name can be selected from the list pressing the button on the right of entry field.

#### 4.21.11 IMAP Mail loop

This action allows to process by cycle all messages from server. As with any another cycle, the group of cycle steps should be finished with the action End Loop. The IMAP Mail Loop uses a current folder on the server. The folder can be set when the connection is opened or with the action IMAP Select Mail Folder.

In each iteration of the cycle you can save various characteristics of the current message into variables for processing by steps inside of the cycle.

IMAP Mail Loop

General Advanced parameters

Assign Message Parameter to Variable

Order number:

Subject:

Sender (FROM):

Recipient (TO):

Carbon Copy (CC):

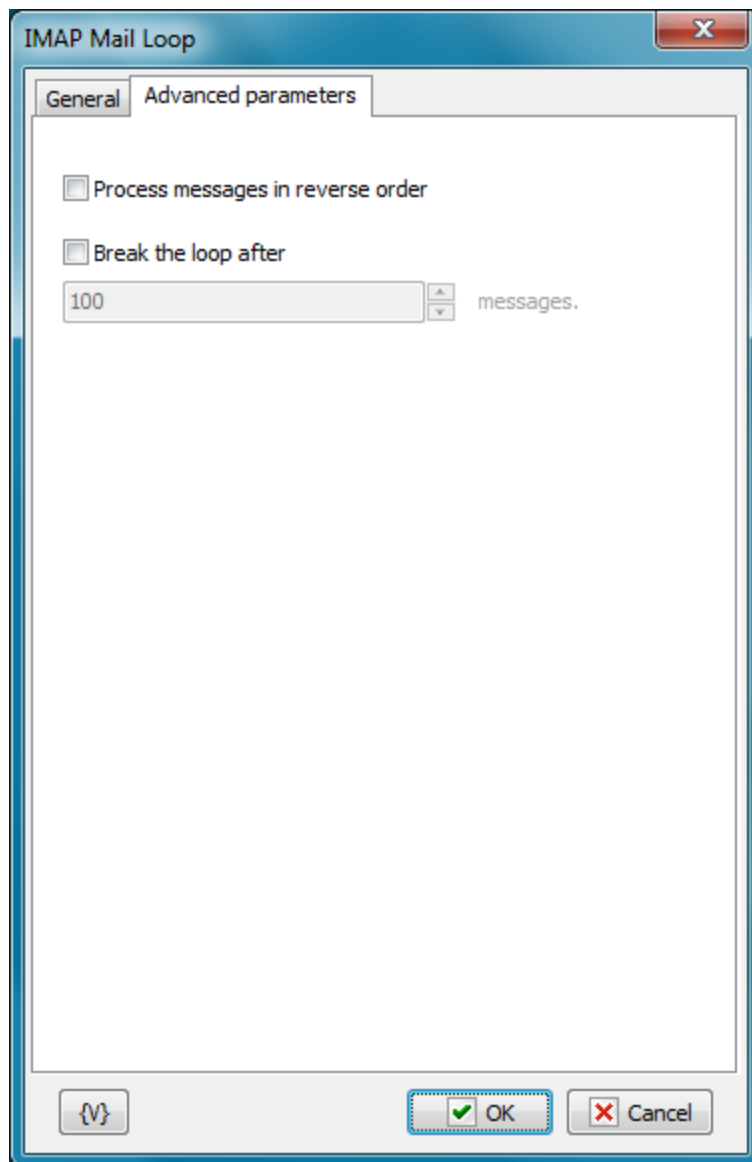
Message Date:

Message Size (bytes):

Message Flags

Message Priority

{V} OK Cancel



### Order Number

Specify the name of the variable into which it is necessary to save the ordinal number of the current message. Next actions with messages will execute according to the number of the message in the folder (for example, [IMAP Download Message](#)<sup>[317]</sup>, [IMAP Set Message Flags](#)<sup>[318]</sup>, etc).

### Subject

To save the subject of the current message, switch on this checkbox and specify the variable name.

### Sender (FROM)

To save the address and the sender name of the current message, switch on this checkbox and specify the variable name.

**Recipient (TO)**

To save the list of recipients of the current message, switch on this checkbox and specify the variable name.

**Carbon Copy (CC)**

To save the list of recipients of the carbon copy of the current message, switch on this checkbox and specify the variable name.

**Message Date**

To save the date and time of sending of the current message, switch on this checkbox and specify the variable name.

**Message Size**

To save the size in bytes of the current message, switch on this checkbox and specify the variable name.

**Message Flags**

To save flags (attributes) of the current message, switch on this checkbox and specify the variable name. Flags of the message will be saved as string. Every letter corresponds to existence of a flag of the message.

- A - Answered
- F - Flagged
- D - Deleted
- S - Seen
- T - Draft
- R - Recent

For example,

AS - means that the message is answered and seen.

R - recent message.

**Process messages in reverse order**

You can process the list of messages in reverse order. This is useful when you work with a large list, because new messages are at the end of the list.

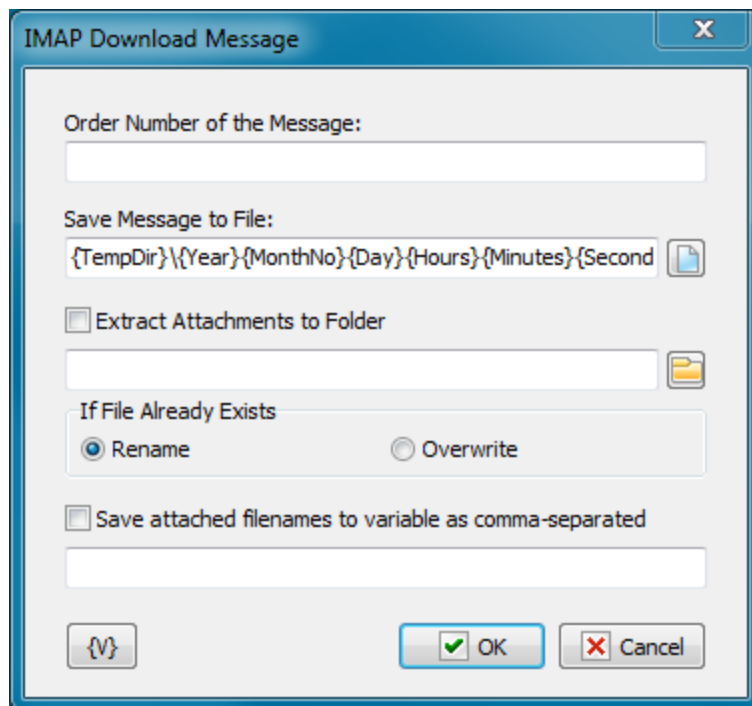
**Break the loop after**

Turn on this checkbox and specify the number of messages if you want to limit the cycle.

**4.21.12 IMAP Download Message**

The action permits to download a whole message from the server and save it into a file.





### Order Number of the Message

An order number of a message. This number can be got in a cycle or specified explicitly. Numbers of messages begin from number 1.

### Save Message to File

Specify the name of a file into which it is necessary to save the message. On default, it is proposed a file name based on a system date and time:

```
{TempDir}\{Year}{MonthNo}{Day}{Hours}{Minutes}{Seconds}{mSeconds}.eml
```

Such file name enables you to avoid name clashes. Also, you get a possibility to sort files according to the receiving time.

### Extract Attachments to Folder

If you want to extract attached files, switch on this checkbox and specify the folder name.

### If File Already Exists

Specify a necessary action if the file already exists when saving attachments:

- Rename
- Overwrite

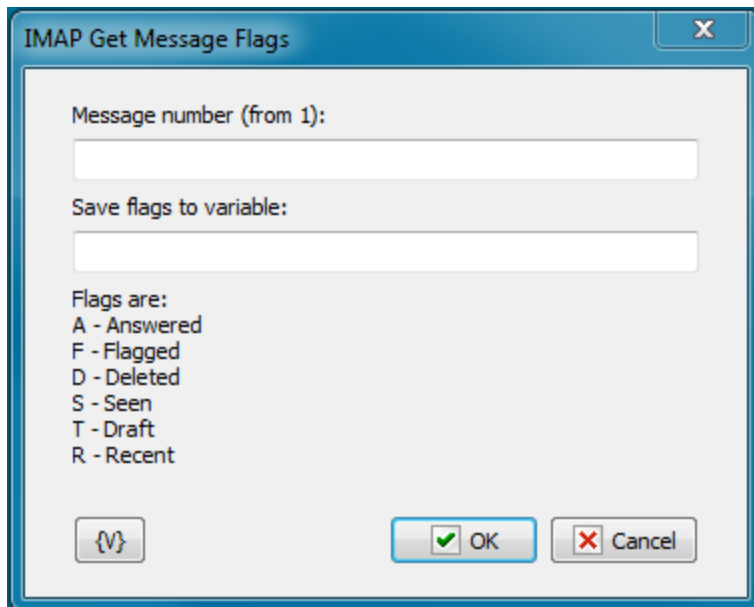
### Save attached file names to variable as comma separated

To save the list of attached file names into a variable, switch on this checkbox and specify the name of the variable.

### 4.21.13 IMAP Get Message Flags

The action enables to get message flags according to their ordinal number and save them into a variable. The message flags are saved as a line. Each letter corresponds to existence of the flag of the message.

- A - Answered
- F - Flagged
- D - Deleted
- S - Seen
- T - Draft
- R - Recent



#### Order Number of the Message

An order number of a message. This number can be got in a cycle or specified explicitly. Numbers of messages begin from number 1.

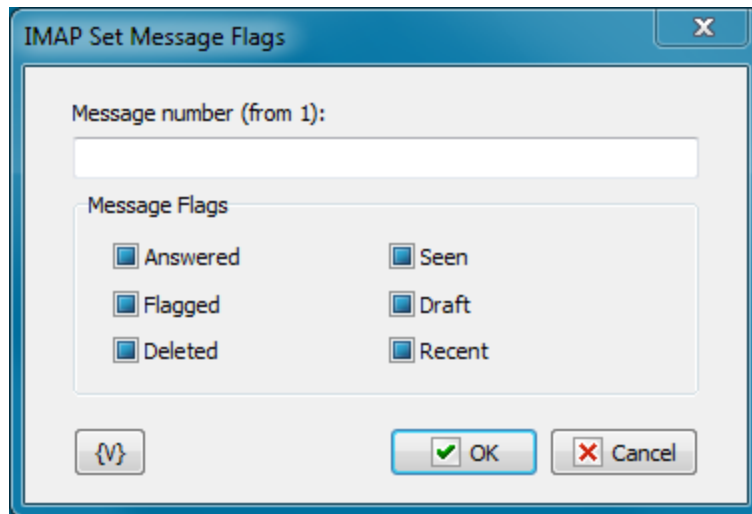
#### Save Flags to Variable

Specify the name of the variable into which it is necessary to save message flags.

### 4.21.14 IMAP Set Message Flags

Sets up message flags according to the number in the current folder. Flags are set by means of checkboxes. Notice that each checkbox has 3 statuses:

- Off. This means that the flag should be resetted (cleared).
- On. This means that the flag should be set.
- Intermediate status. This means that the flag should be left unchanged.



### Message Number

The ordinal number of the message in the current server folder.

### Message Flags

Set up new message flags by means of checkboxes. There are 6 checkboxes which correspond to the message flags:

- Answered
- Flagged
- Deleted
- Seen
- Draft
- Recent

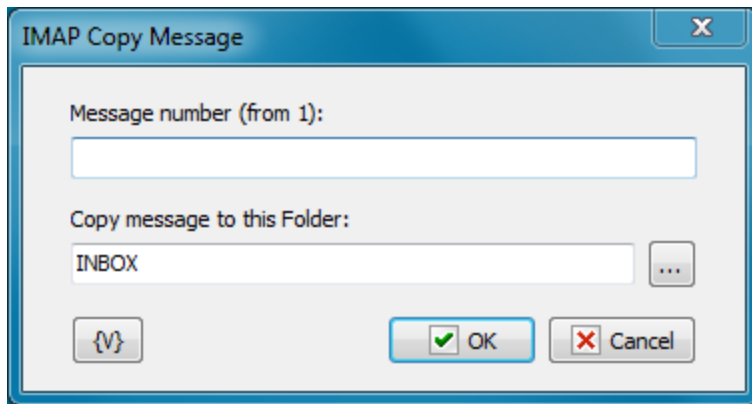
Notice, that to mark the message for deletion, it is enough to mark it with the flag Deleted. After all necessary messages are marked, the action IMAP Delete Marked Messages needs to be performed.

## 4.21.15 IMAP Delete Marked Messages

Performs physical deletion of all messages which are marked with the flag Deleted ( see IMAP Set Message Flags). The action performs for messages in the current folder and does not require additional parameters.

## 4.21.16 IMAP Copy Message

The action performs copying of a message from the current folder into other folder of the server.



### Message Number

The number of a message in the current folder. The numbers begin from 1. It is allowed to use variables.

### Copy Message to this Folder

Input the name of the destination folder. The folder name can be input manually or selected from the list pushing the selection button on the right of the field. Also it is allowed to use variables.

Note:

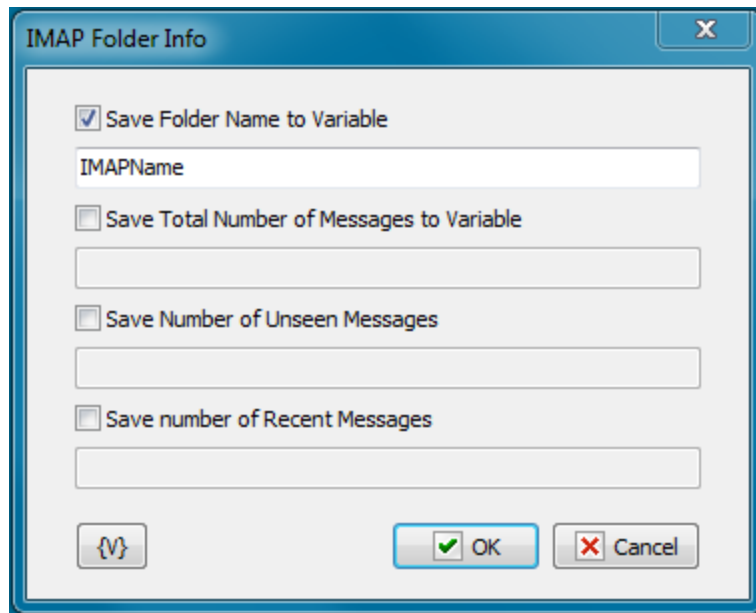
If you want to MOVE the message into another folder you have to copy the message and then mark it as DELETED (see [IMAP Set Message Flags](#)<sup>[319]</sup> action). Also you can put the [IMAP Delete Marked Messages](#)<sup>[320]</sup> after moving all necessary messages.

## 4.21.17 IMAP Folder Info

The action enables to read parameters of the current folder and to save their values into variables for the next processing.

The folder parameters which can be extract:

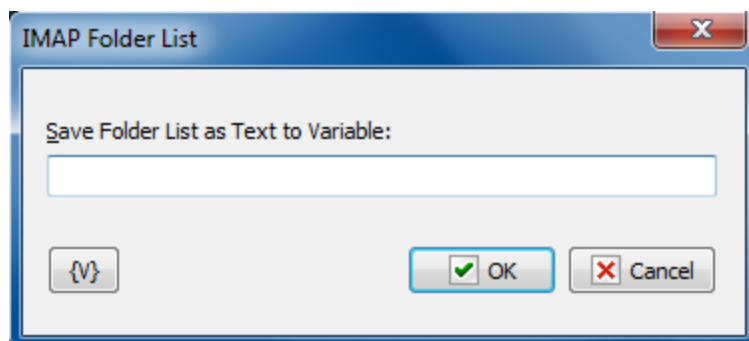
- The folder name
- Amount of messages
- Amount of not shown messages
- Amount of resent messages



To extract the parameter value, it is necessary to switch on the appropriate checkbox and to input the variable name.

#### 4.21.18 IMAP Folder List

This action saves the folder list of an open IMAP server as a text into a variable. Every folder name is on the new line. You can process the text line by line, for example. In the loop Text Loop.



#### Save Folder List as Text to Variable

Input the name of the variable into which you need to save the folder list.

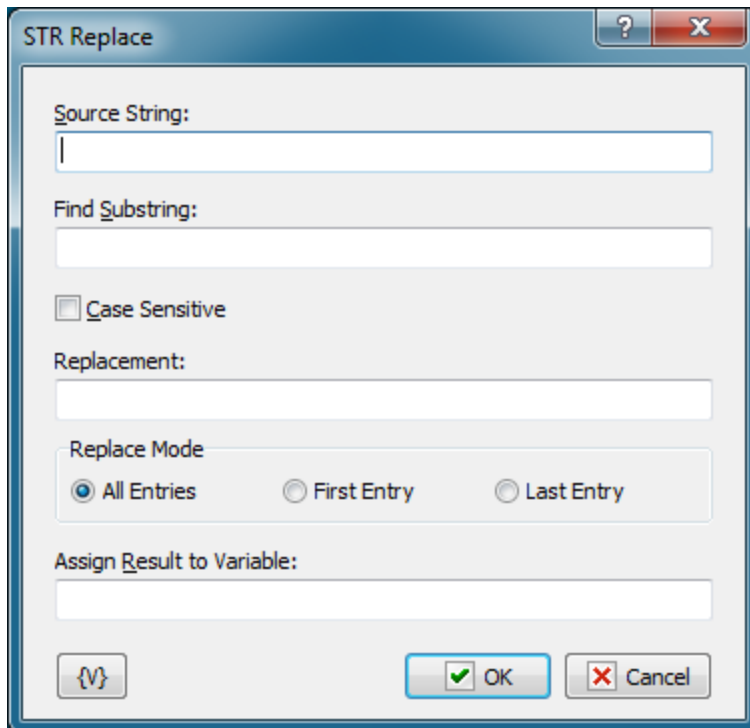
## 4.22 Strings

### 4.22.1 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

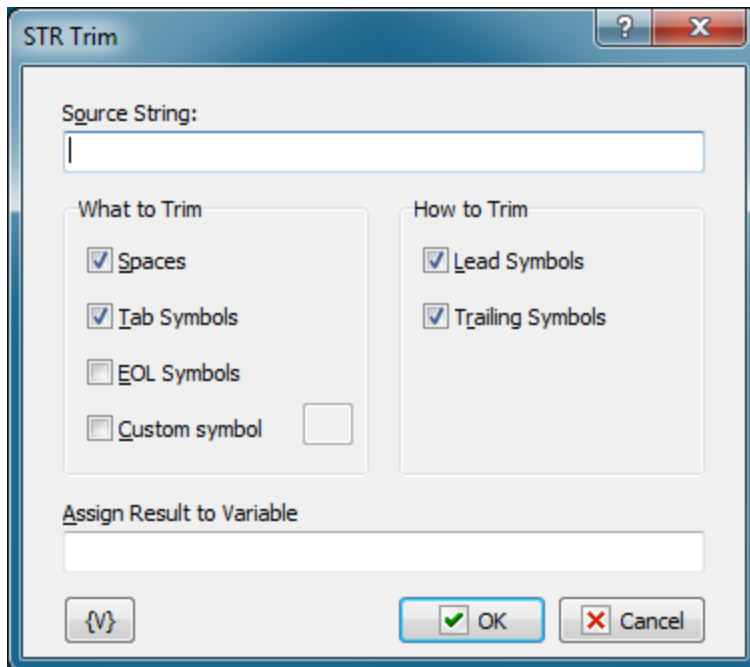
#### Variable

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.

### 4.22.2 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

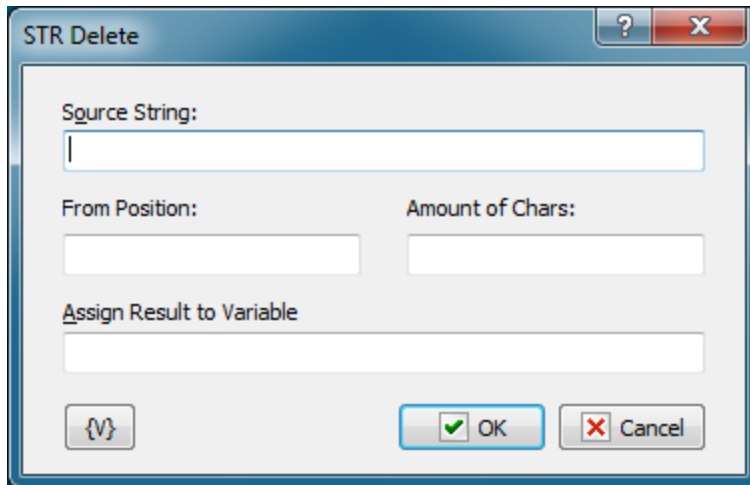
The name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

**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.

**4.22.3 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

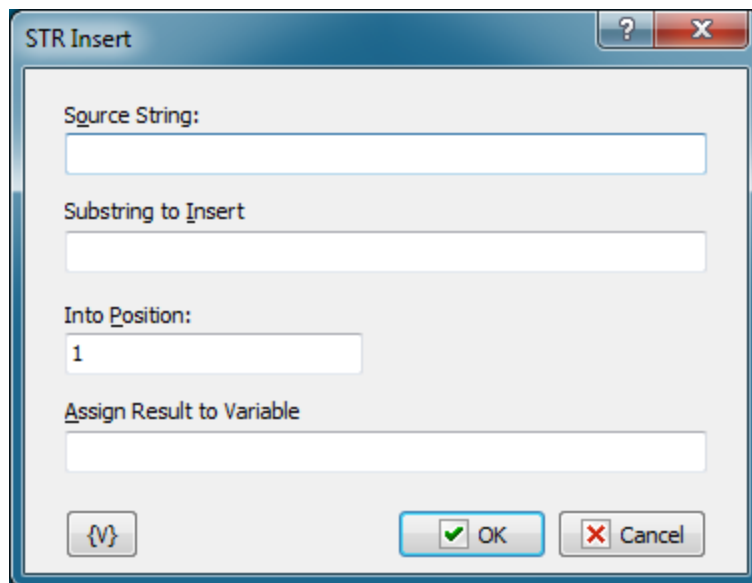
**Variable**

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.

**4.22.4 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**

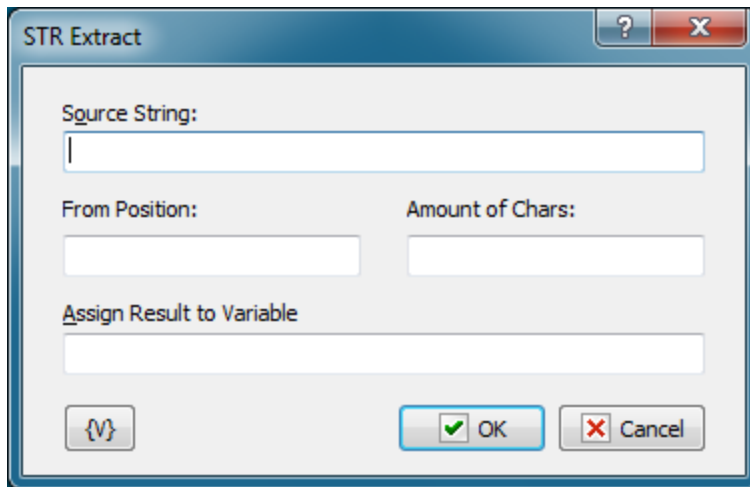
The name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

**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.

#### 4.22.5 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**

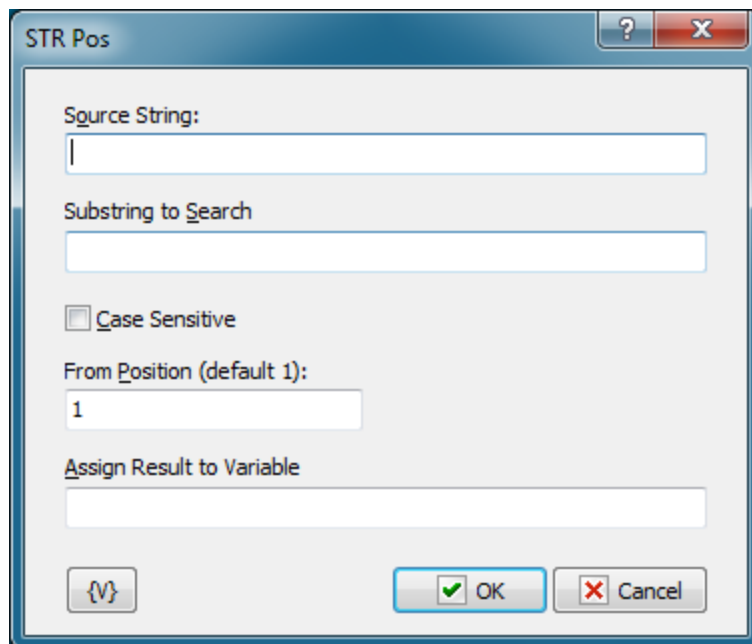
The name of the variable the result of the action will be assigned to. It is impossible to use macros with variables

**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.

#### 4.22.6 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.

## 4.22.7 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:

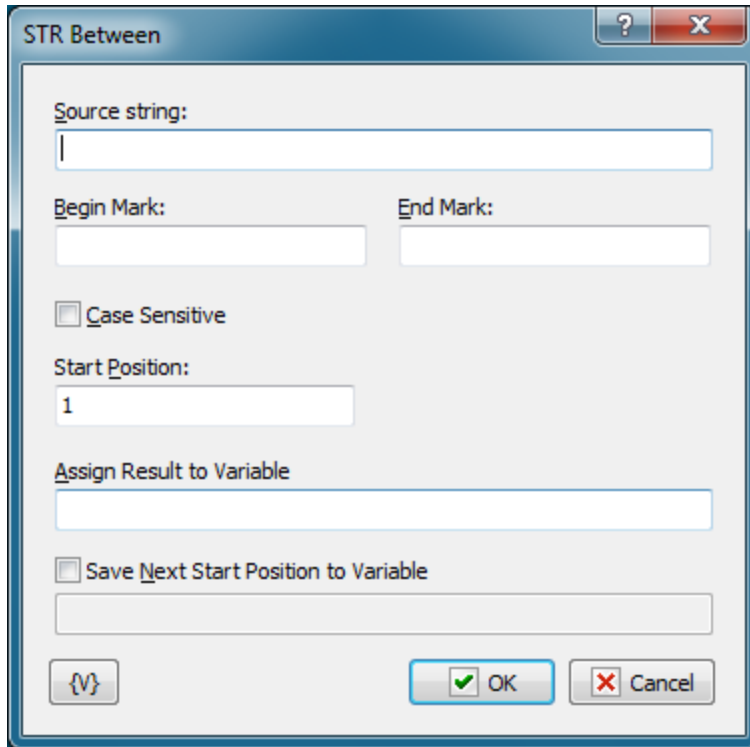
```
<h3><b>Another news - at last week</b></h3>
```

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
```

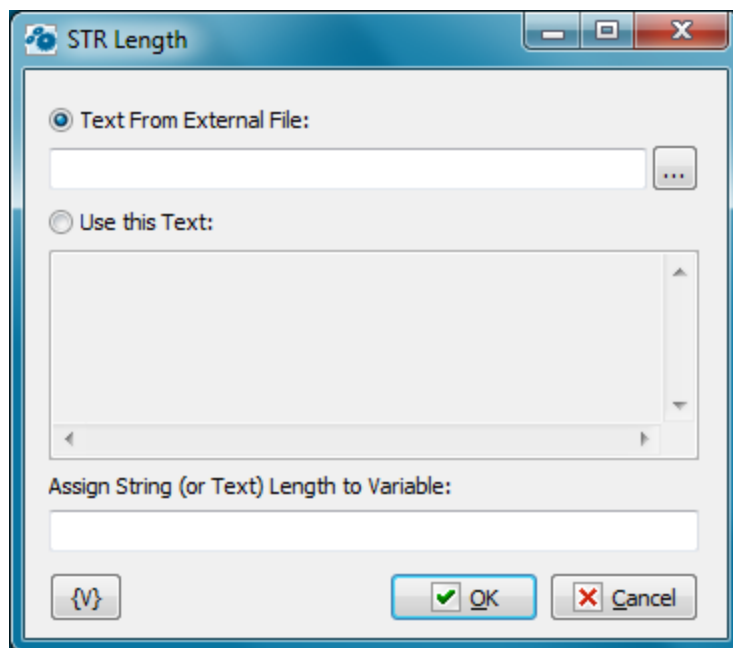
You can easily do it using a loop. See the following example: "**[DEMO] STR Between - extract all entries**"

### 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.

## 4.22.8 STR Length

The action measures the length of a string or a text in symbols. The RoboTask has the system variable **{StrLength(Sample String)}**. But this variable fits for small simple strings. Sometimes it is impossible to use this function because the string can contain special symbols and brackets what can break the syntax of the expression with variable and then you get an error. In such cases it is better to use this action.



### Text From External File

Enter the name of the file from which you want to take a text. You can select the file clicking the button on the right of the entry field.

### Use This Text

Specify the text explicitly or by using the variables of the RoboTask.

### Assign String Length to variable

Enter the name of the variable in which you want to save the length of a string or a text.

#### 4.22.9 STR format CSV line

Action allows to change CSV format of string.

STR Format CSV line

Standard format of comma-separated string

Delimiter: single comma char (,)

Quote char: double quote char (")

Custom format

Delimiter: , - comma

Enter symbol:

Quote char: " - (Double quote)

Source string:

Enter source comma-separated string

Direction: Standard format ==> Custom format

Save result to variable:

Enter variable name

{V} OK Cancel

**Standard format** is static and can not be changed

#### Custom format

Select *delimiter* and *quote char*

#### Source string

Enter source string to transform.

**Direction**

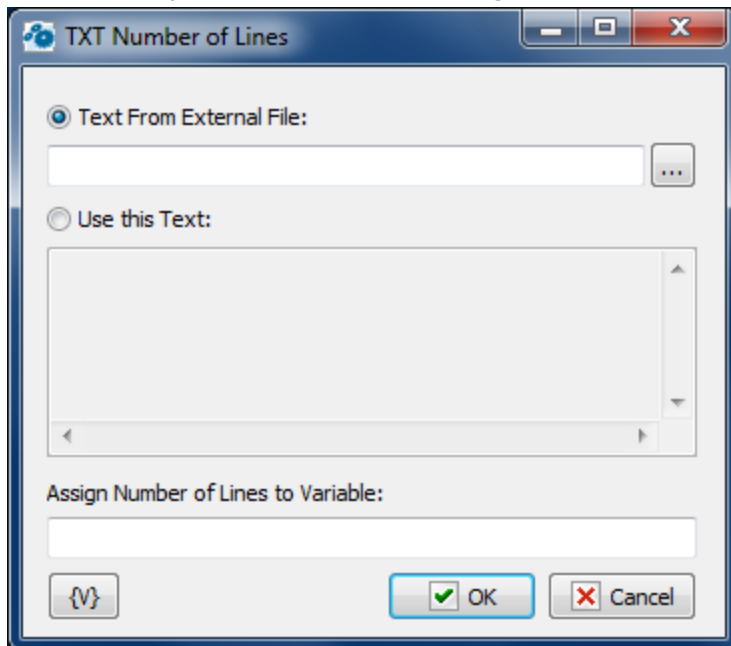
Select the conversion direction: standard to custom or backwards

**Save result to variable**

Enter the name of the variable where you want to store the result.

#### 4.22.10 TXT Number of Lines

The action is intended to count the number of lines in a text. It allows to count the number of lines in a specified file as well as in a specified text with variables. The action assigns the result to a specified variable for using it in the following steps.

**Text from External File**

Specify the file name where it is necessary to count the number of lines. It is possible to select the name using the dialog of file selection. Using variables in the file name is possible.

**Use this Text**

You can set a text in the editing field explicitly. The text can also contain variables.

**Assign Number of Lines to Variable**

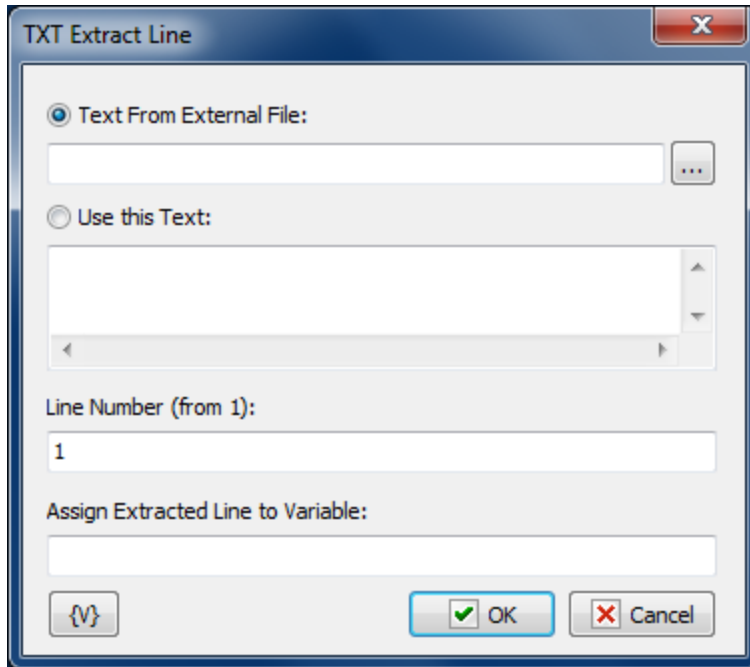
Specify the name of a variable where the number of lines is to be assigned.

**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.

### 4.22.11 TXT Extract Line

This action allows extracting a line by number from a text. The number has to be specified from 1 to the quantity of lines in a text. If the number is incorrect, the action will return an empty line. The action assigns the specified line to a variable for using it in the following steps.



#### Text from External File

Specify the file name where it is necessary to count the number of lines. It is possible to select the name using the dialog of file selection. Using variables in the file name is possible.

#### Use this Text

You can set a text in the editing field explicitly. The text can also contain variables.

#### Line Number

Specify the number of a line which is necessary to extract from a text. Using variables is possible. If non-numeric value is specified, the action generates a run-time error.

#### Assign Extracted Line to Variable

Specify the name of the variable where the demanded line will be assigned.

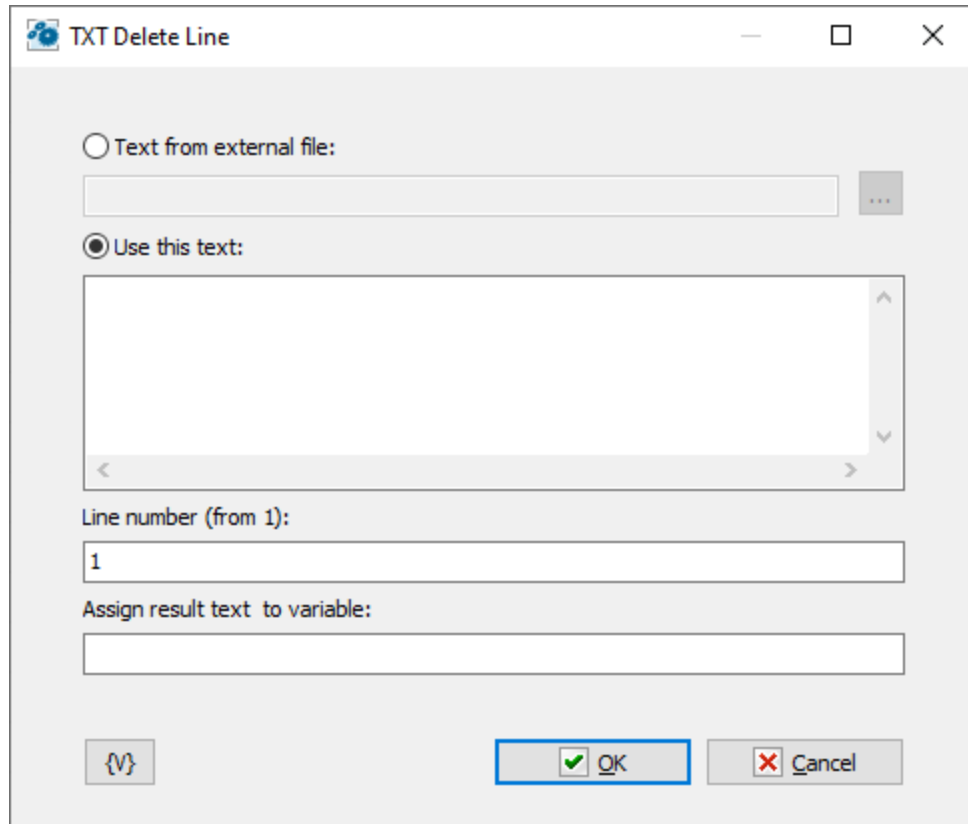
#### Variable

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.



### 4.22.12 TXT Delete line

The action allows to remove one line from the text by line number.



#### **Text from external file**

Enter file name or select file from dialog. You can use variables.

#### **Use this text**

Enter the text into this field. You can use variables in the text

#### **Line number**

Enter the number of the line to be removed from the text. The line number starts at 1. If the line is less than 1 or greater than the number of lines, then the action writes a warning to the task log.

#### **Assign result to variable**

Enter variable name where you want to save the result.

### 4.22.13 TXT Insert line

Enter topic text here.

The action allows you to insert a line in the text at the specified position.

The screenshot shows a dialog box titled "TXT Insert Line". It features two radio buttons: "Text from external file:" (unselected) and "Use this text:" (selected). Below the first radio button is a text input field with a file selection icon. Below the second radio button is a large text area. Below the text area is a text input field labeled "Insert this line". Below that is a dropdown menu labeled "Where to add:" with the text "Enter line number to insert". Below the dropdown is a text input field labeled "Line number (from 1):" with the value "1". Below that is a text input field labeled "Assign result text to variable:". At the bottom are three buttons: a variable icon, an "OK" button with a green checkmark, and a "Cancel" button with a red X.

#### **Text from external file**

Enter file name or select file from dialog. You can use variables.

#### **Use this text**

Enter the text into this field. You can use variables in the text

#### **Insert this line**

Enter a string to insert

#### **Line number**

---

Enter the number of the line where you want to insert string. The line number starts at 1. If the line is less than 1 or greater than the number of lines, then the action writes a warning to the task log and appends the string to the end of the text.

**Assign result to variable**

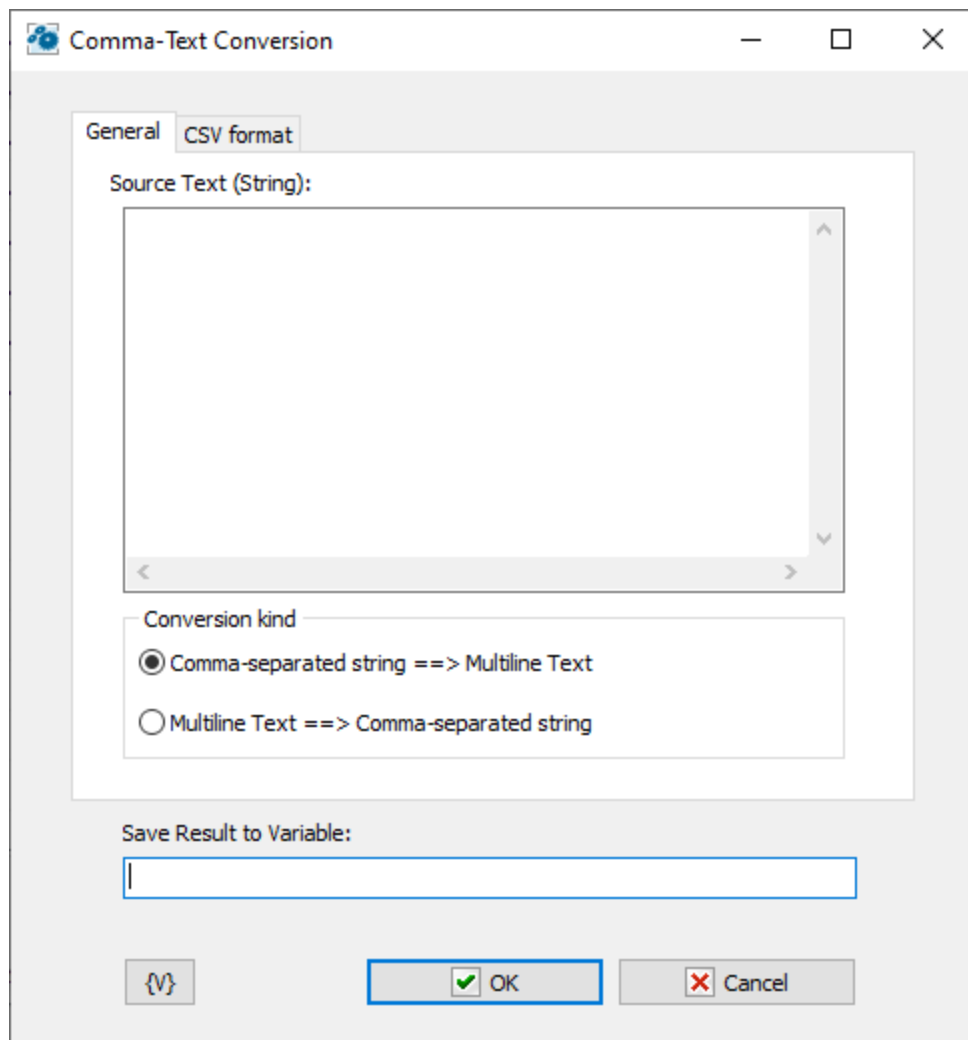
Enter variable name where you want to save the result.

**4.22.14 TXT Comma-Text Conversion**

This action is designed for conversion of a comma-separated string in a multiline text and vice versa.

Often data for machine processing is given as comma-separated string. At the same time a multiline text is easier for processing.

- Processing of strings in the cycle (Text Loop)
- Calculation of quantity of strings and extraction of a string by its number.
- etc.



### Source Text (string)

Specify a source text or a comma-separated string for conversion.

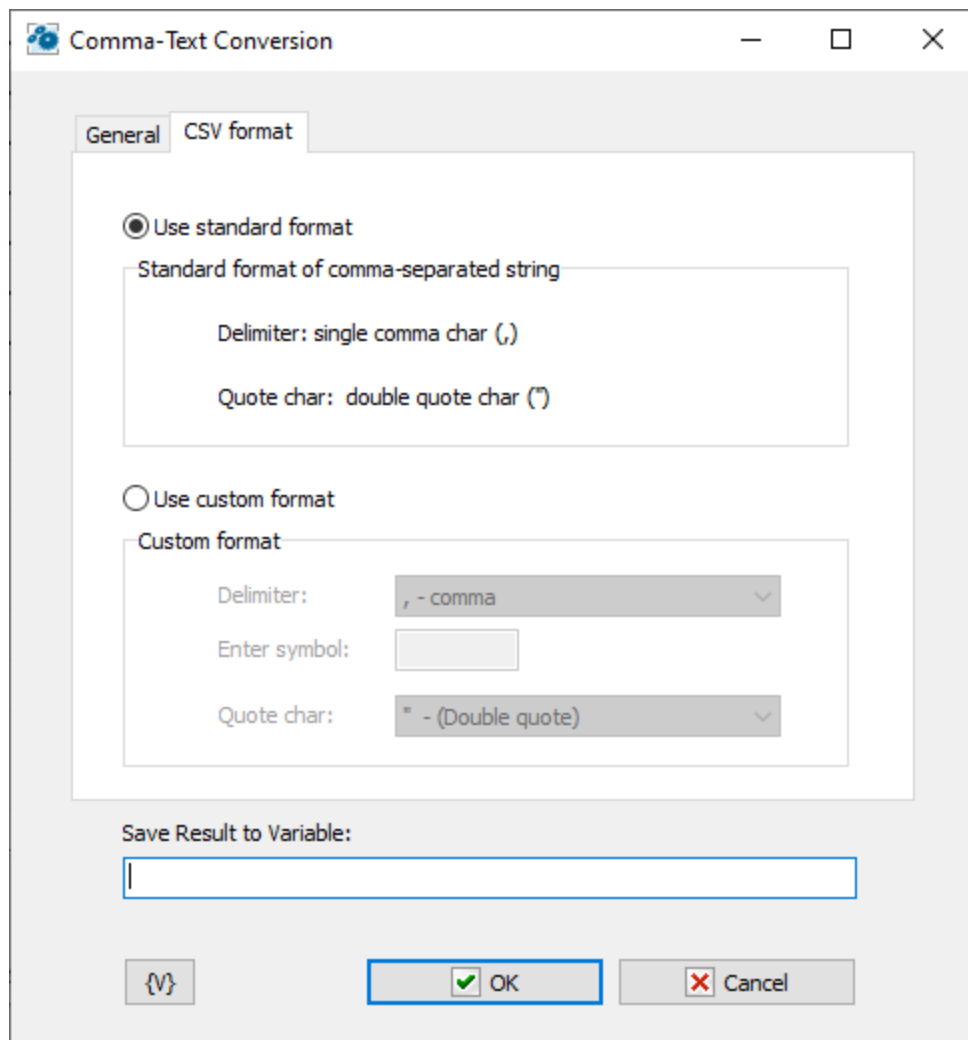
### Conversion kind

Select one of two conversion options

- Comma-Separated String ==> Multiline Text
- Multiline Text ==> Comma-Separated String

### CSV format

You can use more than just the standard CSV string format.



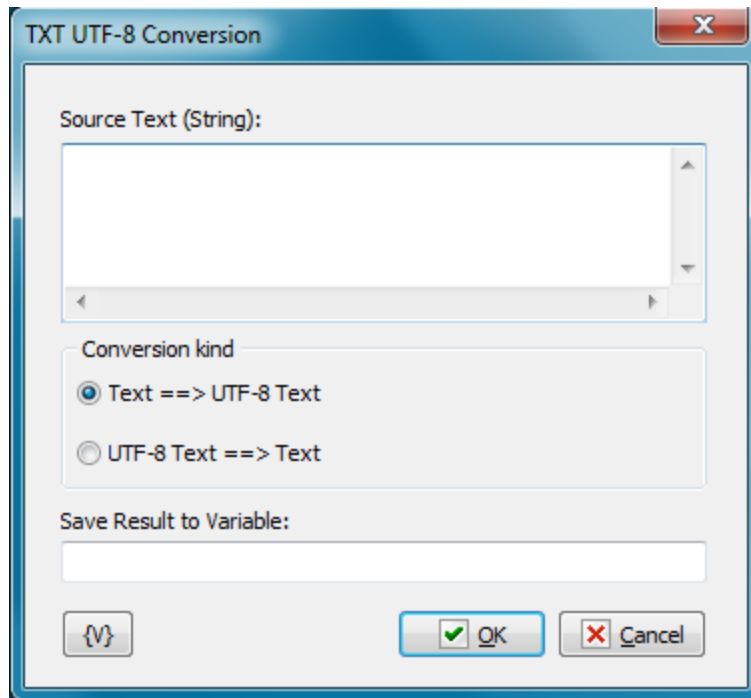
Choose *Custom format* if you want to use non-standard CSV. Then select *delimiter* and *Quote char*

### Save Result into Variable

Input the name of the variable into which you need to save the result of the conversion.

## 4.22.15 TXT UTF-8 Conversion

The action permits to converse a text from Unicode to the UTF-8 and vice-versa. The RoboTask contains the variables [UTF8Decode and UTF8Encode](#) <sup>[556]</sup> permitting to do such conversions for small lines. But, unfortunately, some problems can appear when trying to converse a big text. The text can contain the symbols breaking the variable syntax. This action permits to avoid such problems.

**Source Text (string)**

The source text. You can specify as a text or a string explicitly or with using variables.

**Conversion kind**

The direction of conversion

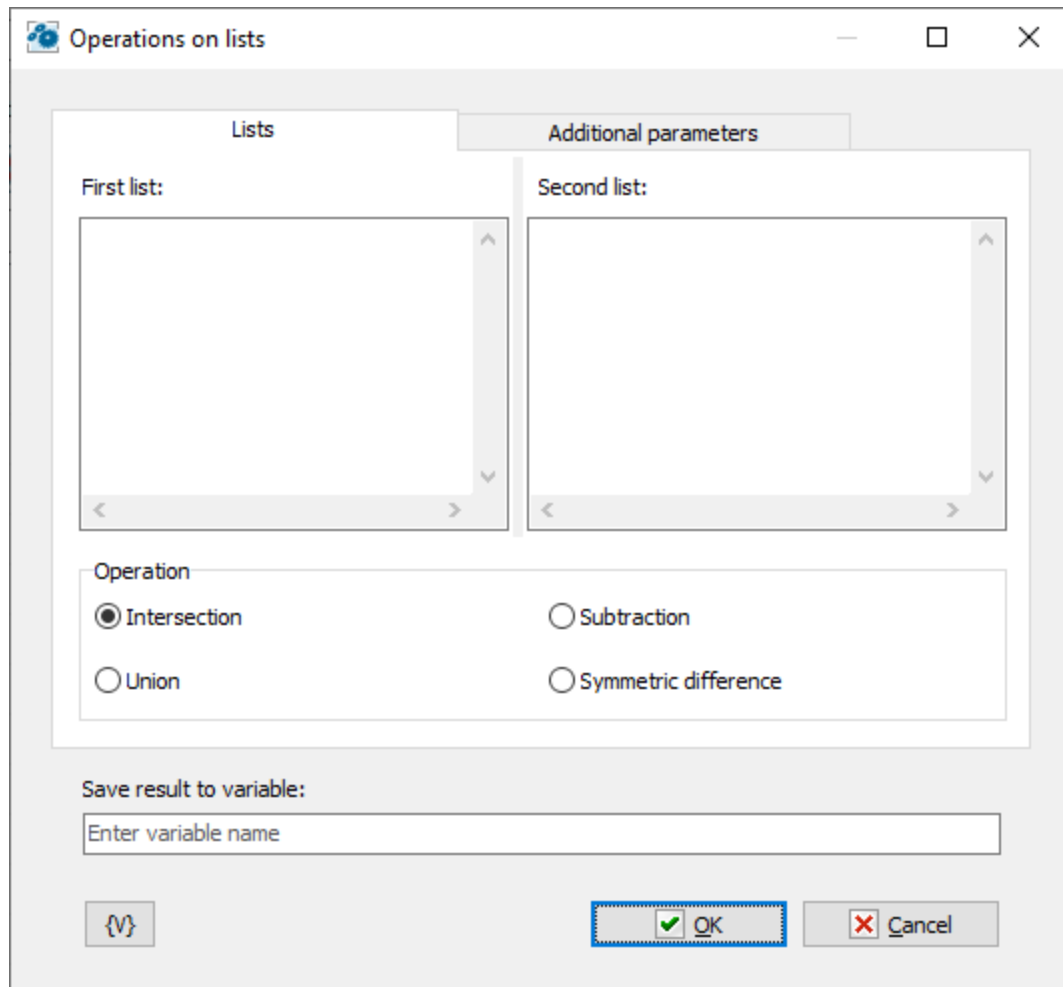
- Text ==> UTF-8 Text
- UTF-8 Text ==> Text

**Save Result to Variable**

Specify the name of user's variable in which you need to save the result.

#### 4.22.16 Operations on lists

The action allows you to perform mathematical operations on two lists: intersection, union, subtraction and symmetric difference and save result to variable



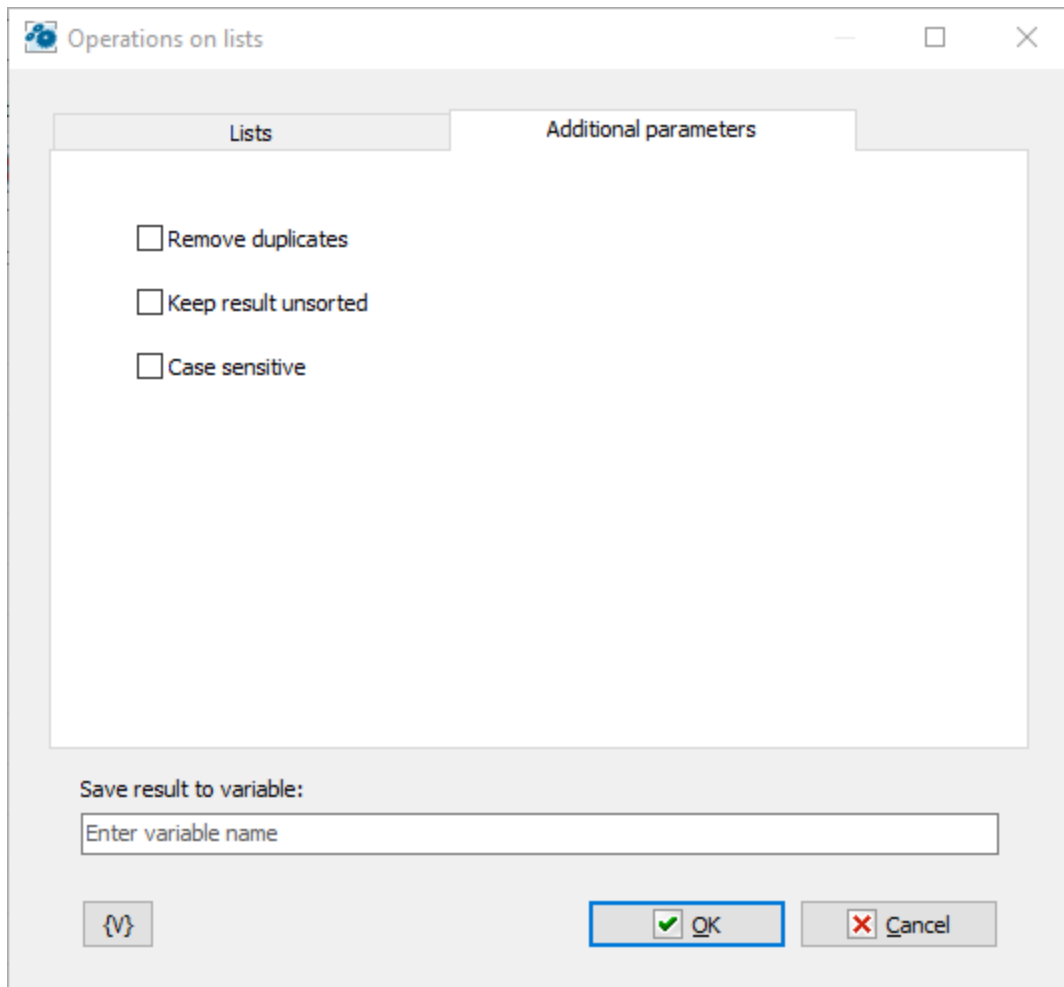
### First list, Second list

Lists for processing. The list is simple text where each line is the element of the list. For example the list of file names.

### Operation

There are four operations:

- *Intersection* - elements that are in both lists
- *Union* - all elements from both lists
- *Subtraction* - elements from the 1st list and missing from the 2nd list (1-st list minus 2-nd list)
- *Symmetric difference* - the *union* minus *intersection*



### Remove duplicates

The action removes duplicate elements from result

### Keep result unsorted

If you want to keep the order of the elements, then turn on this switch

### Case Sensitive

By default, the action is case-insensitive when comparing elements.

## 4.23 Embedded Chromium Browser

This set of actions is for automated web page management. The embedded web browser runs on the *Chromium Embedded Framework* (CEF). Unfortunately, the actions of *Web Documents* are somewhat outdated, because the used engine does not support many innovations of the HTML format.



### 4.23.1 Browser Fill Web Form

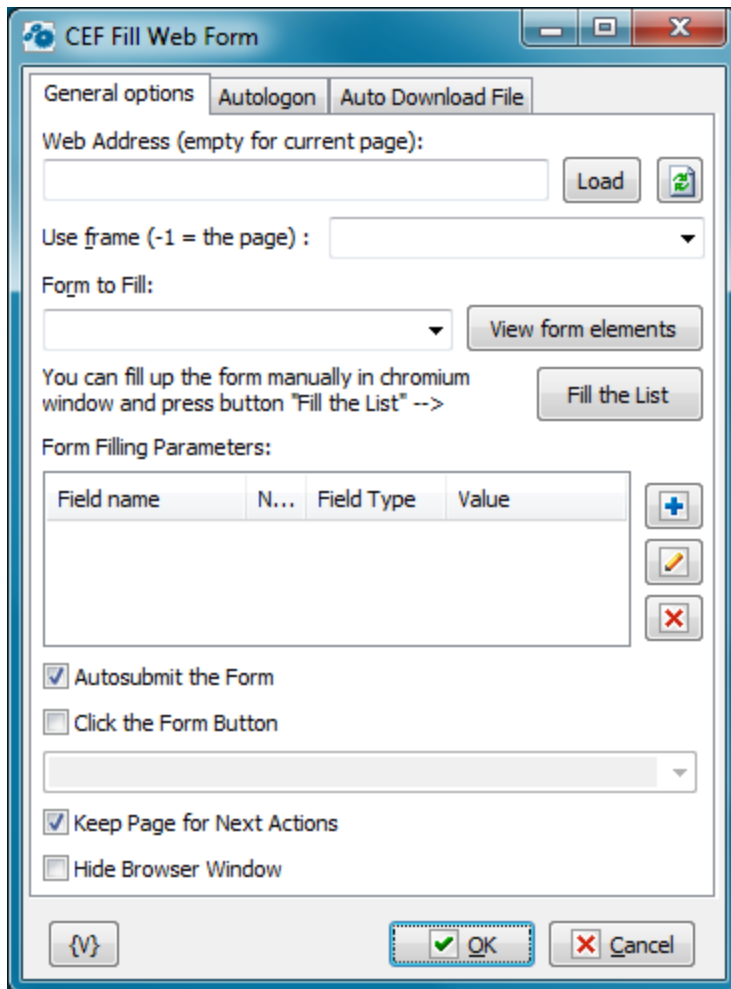
This action is destined for automation of filling web forms. Of course you can automate filling forms by "Send Keystrokes" and mouse actions. But unfortunately, this is much more difficult and unreliable, because RoboTask tries to influence your external application (your web browser) and can't control 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 "[CEF Wait For Done](#)<sup>357</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



### 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

### Use Frame

Specify frame number (from 0) . "-1" corresponds to the page. The action uses any non-numeric string as "-1". If you input invalid number then you will get the error "Frame not found".

### 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, and so on.

### 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>[347]</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.

### Click the Form Button

To reproduce a button click on a web form, switch on this checkbox and select the button name from dropdown list or input it by hand.

### 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.

### Hide Browser Window

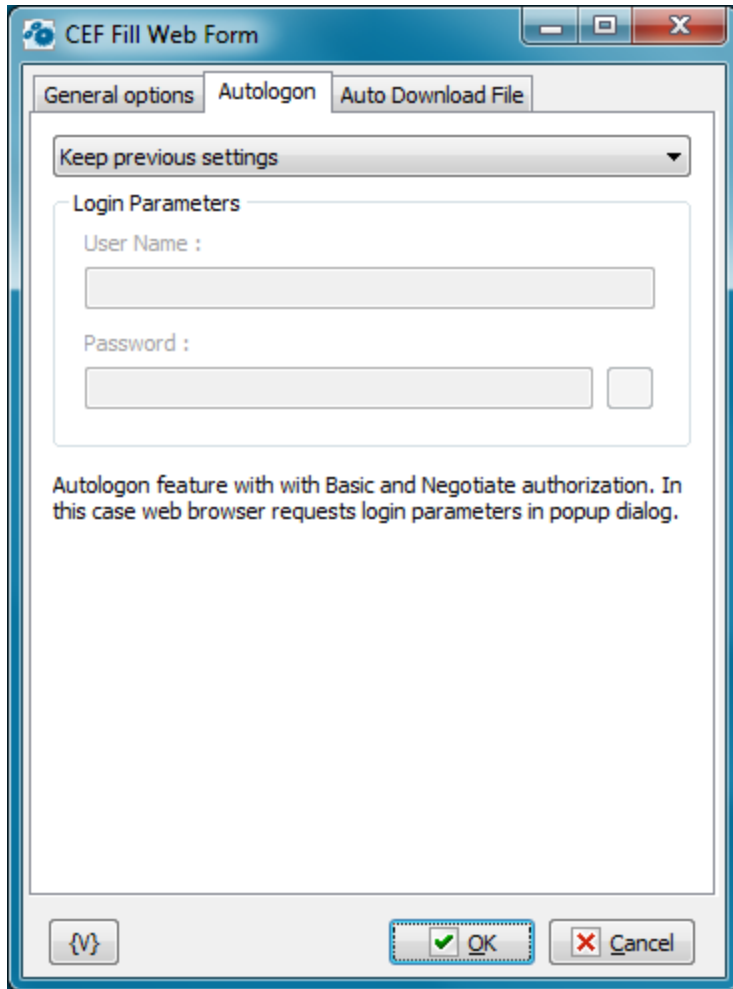
To ensure hidden work of the web browser window, switch on this checkbox. In this case the browser window is not visible on the screen.

### 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.



### Autologon Mode

There are three options

- **Keep previous settings**
- **No autologin (manual login)** - user have to login manually when web browser ask *username* and *password*
- **Enable autologin. Set username and password.**

### User name

A name of a user for authorization. Variables can be used.

### Password

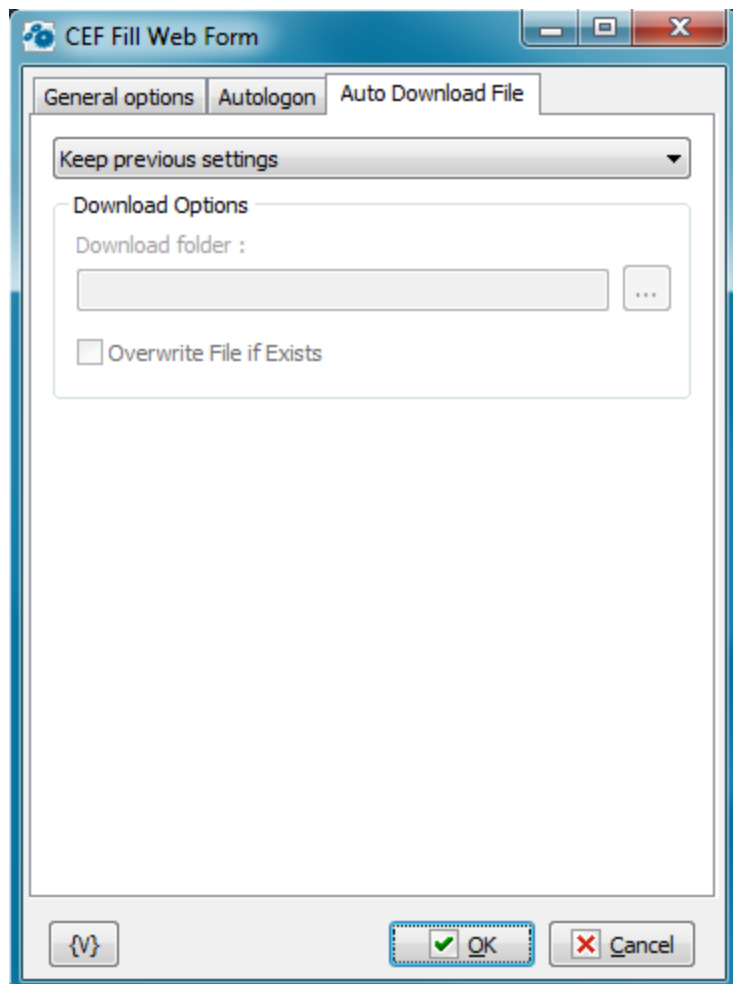
A password for authorization. Variables can be used.

## 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.



### Auto-download mode

There are two options

- Keep previous settings

- Change download options - set the *folder name* and *overwrite* parameter.

Default download folder is **%temp%\cfdownloads**

### Download Folder

Specify the folder in which the file will be downloaded automatically. It will be downloaded with the file name by default. Default value is **%temp%\cfdownloads**

### 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. Default value is **OFF**

#### 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

The screenshot shows a dialog box titled "Edit Webform Parameter". It contains the following fields and controls:

- Frame: page; Form:
- Field name: (dropdown menu)
- Field Number : 0 (spin box)
- Field type: (dropdown menu showing "Edit")
- Value for "Edit" field: (text input field)
- Bottom buttons: {V}, OK (with green checkmark), and Cancel (with red X).

### 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. Also you can specify field order number.

### 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. This parameter is ignored if you specify field number as field name.

### 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.
- **Password** - as edit field but the field value is hidden from user

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.
- **Password** - 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 Browser Web Page Information

This action simulates a visit to the specified web page by means of built-in web browser. If the address of a page is not specified, the action uses the current page which has been loaded in the previous steps. Also the action allows to extract the following information from the web page:

- **HTML code** – the source code of the page as HTML text.
- **Text** – the plain text of the page without HTML tags
- **Links** – the list of links as a text on the page : one link per line.

The extracted information can be assigned to variables for the following processing.

#### General options

CEF Web Page Information

General Options Autologon Auto Download File

Web Address (Empty for Current Page):

Keep the Page for Next Actions

Save Current Address as a Referrer

Hide Browser Window

Get Frame count to variable

Get Current URL

Use Page or Frame for the Parameters Below (-1 = page):

-1

Get HTML Code of the Page

Get Text of the Page

Get Links from the Page

{V} OK Cancel



### **Web Address**

The address of a web page which is necessary to download. If the address is not specified, the action uses the current page which has been downloaded in the previous steps.

### **Keep the Page for Next Actions**

If you need the downloaded page to be available in the following steps, switch on this parameter. Otherwise, the page will be deleted from the memory immediately after performing this action.

### **Hide Browser Window**

To ensure hidden work of the web browser window, switch on this checkbox. In this case the browser window is not visible on the screen.

### **Use Page or Frame for Parameters Below**

Specify frame number (from 0). "-1" or any non-numeric value corresponds to main page. If you input invalid number then you will get the error "Frame not found".

### **Get Frame Count to Variable**

Assign frame count to specified variable

### **Get Current URL**

Assign current page URL to variable. The URL can be changed during task performing.

### **Get HTML Code of the Page**

Switch on this checkbox if you want to get an HTML code. Below specify the name of the variable in which the HTML code will be assigned.

### **Get Text of the Page**

Switch on this checkbox if you want to get extract the text from the page. The action extracts the text only, without any HTML tags.

Below specify the name of the variable in which the extracted text will be assigned.

### **Get Links from the Page**

If you want to get a list of links on the page, switch on this checkbox and specify the name of the variable in which you need to assign the received list. The action extracts the list of links as a text, one link per line. Then you can process this list in a loop (e.g. in Text Loop).

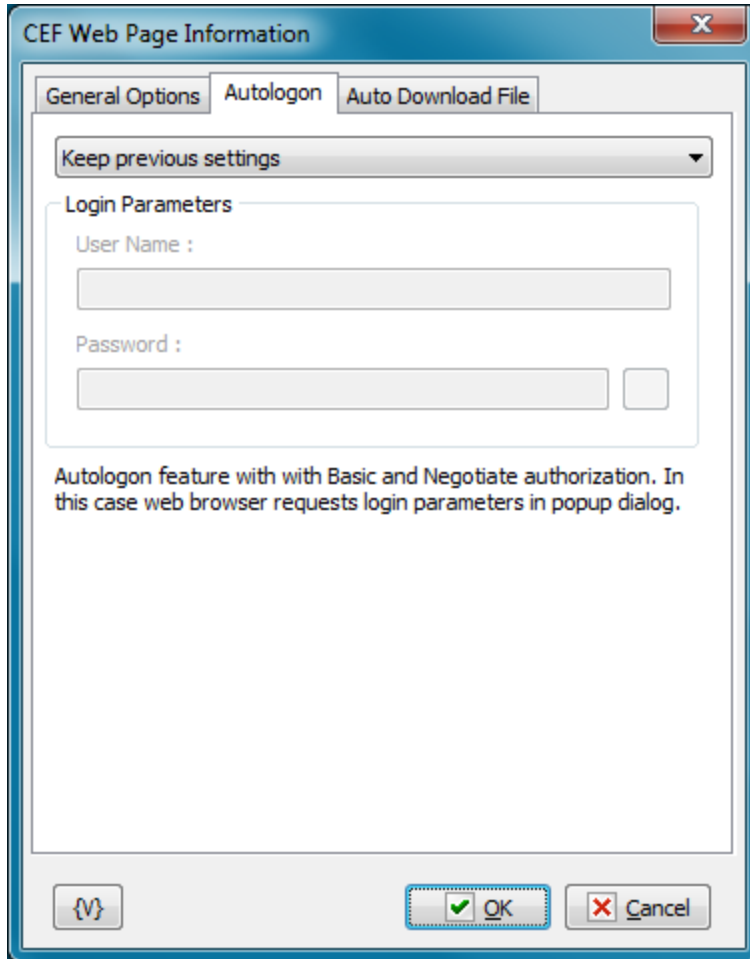
The list of links is a comma-separated text with two fields: *Link address* and *Link Text*

### **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.



### Autologon Mode

There are three options

- **Keep previous settings**
- **No autologin (manual login)** - user have to login manually when web browser ask *username* and *password*
- **Enable autologin. Set username and password.**

### User name

A name of a user for authorization. Variables can be used.

### Password

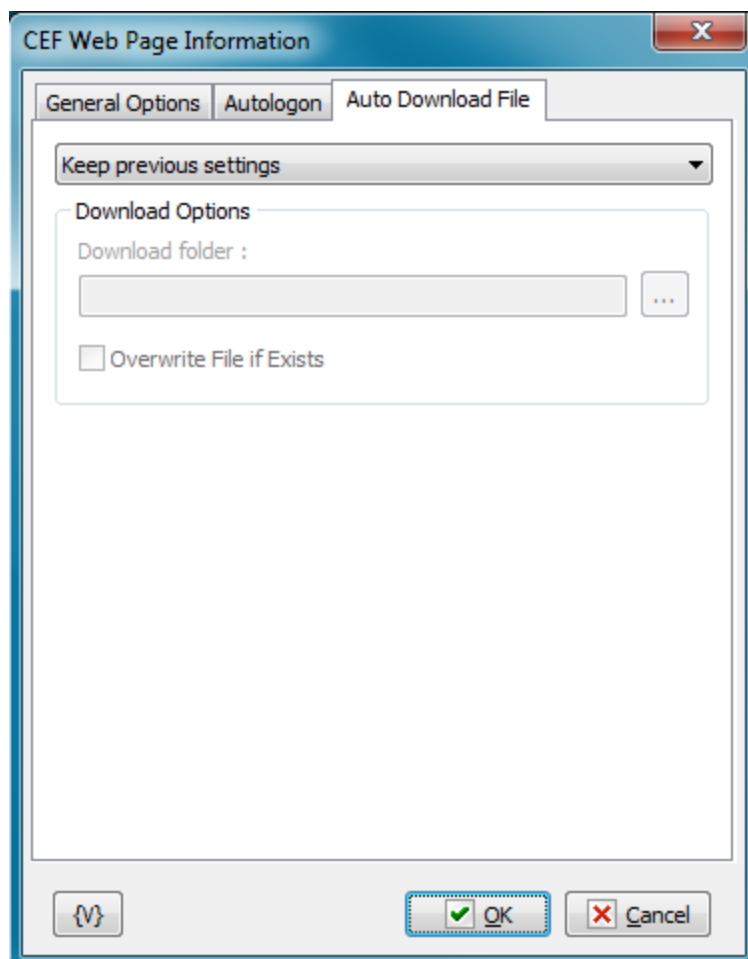
A password for authorization. Variables can be used.

## 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.



### Auto-download mode

There are two options

- Keep previous settings
- Change download options - set the *folder name* and *overwrite* parameter.

Default download folder is ***%temp%\cefdownloads***

### Download Folder

Specify the folder in which the file will be downloaded automatically. It will be downloaded with the file name by default. Default value is **%temp%\cefdownloads**

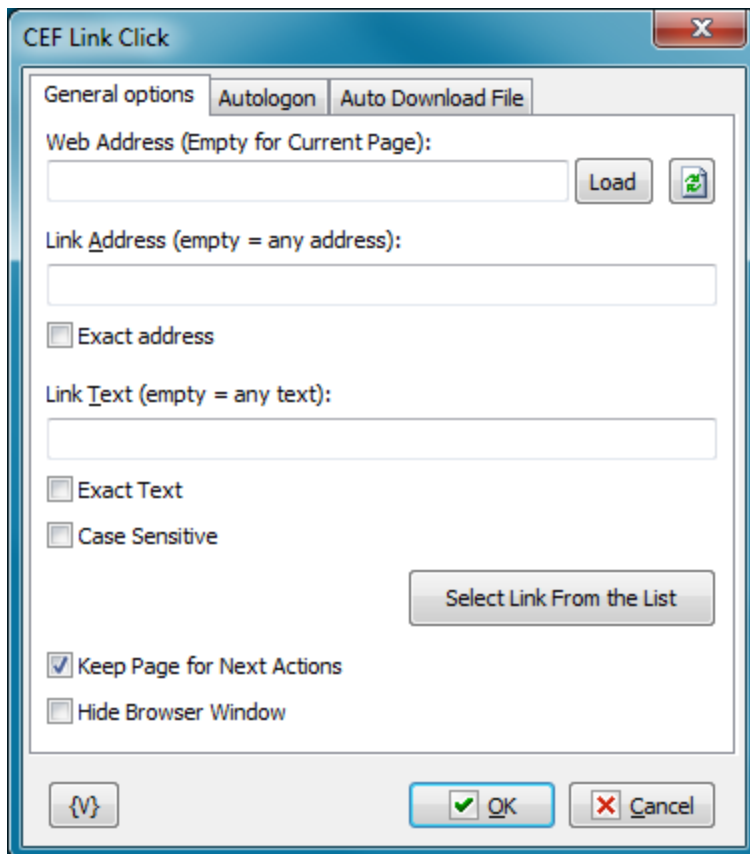
### Overwrite File If Exists.

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. Default value is **OFF**

### 4.23.3 Browser Click Link

The action simulates a click on the hyperlink on the page. Often on web pages it is used the implementation of the built-in script at click on the hyperlink. In this case, loading the page via a direct link may not give the desired result. It is also possible that the links on the page are dynamically generated and can only be identified by the key phrase of the link.

The action searches hyperlink on the page by text and link address (or by part of the text and/or address) and makes an imitation click on the link. If the required link is not found, then the action generates the error "Link not found".



### 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.

**Load**

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 the page (or a blank page) is loaded, you can manage the page interactively: click on the links, fill in the input fields, click on the form buttons, or enter a new address in the address line.

**Refresh button**

Updates the list of links from the current page, because the current loaded page may differ from the original page

**Link address**

Enter the address of the link (or part of the address) to search. You can select a link and a text from the list by clicking the "Select Link from the List" button.

You can leave the field blank. In this case, the address will not be taken into account when searching (any address). You can also use variables.

**Exact address**

Switch on this checkbox if you need to search for an address by exact match. In most cases, it is enough to specify a part of the address to search.

**Link Text**

Enter the text of the link (or part of text) to search. You can select a link and a text from the list by clicking the "Select Link from the List" button.

You can leave the field blank. In this case, the address will not be taken into account when searching (any address). You can also use variables.

**Exact Text**

Switch on this checkbox if you need to search for a text by exact match. In most cases, it is enough to specify a part of the text to search.

**Case sensitive**

Switch on this checkbox if you need to consider case when searching. In most cases, this is not necessary.

**Select Link from the List**

You can select the necessary link from the list of links on the page. Each link has 2 parameters to identify:

- Link Address
- Link text

When selecting, the Link Address and Link Text fields are automatically filled.

### Keep page for next actions

If you need the downloaded page to be available in the following steps, switch on this parameter. Otherwise, the page will be deleted from the memory immediately after performing this action. Default value is ON.

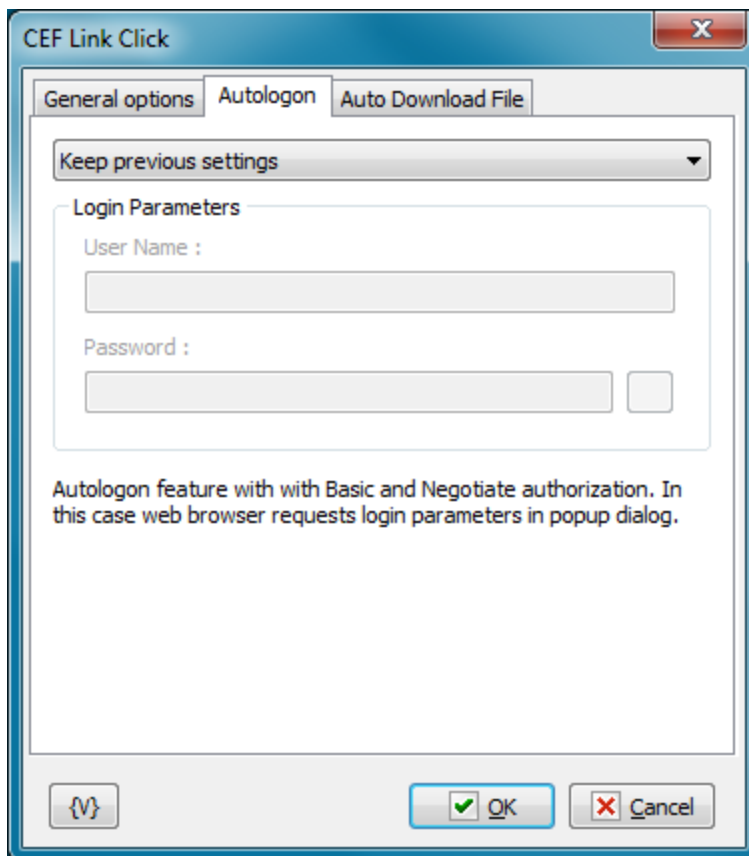
### Hide Browser Window

To ensure hidden work of the web browser window, switch on this checkbox. In this case the browser window is not visible on the screen.

## 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.



### Autologon Mode

There are three options

- **Keep previous settings**
- **No autologin (manual login)** - user have to login manually when web browser ask *username* and *password*
- **Enable autologin. Set username and password.**

### **User name**

A name of a user for authorization. Variables can be used.

### **Password**

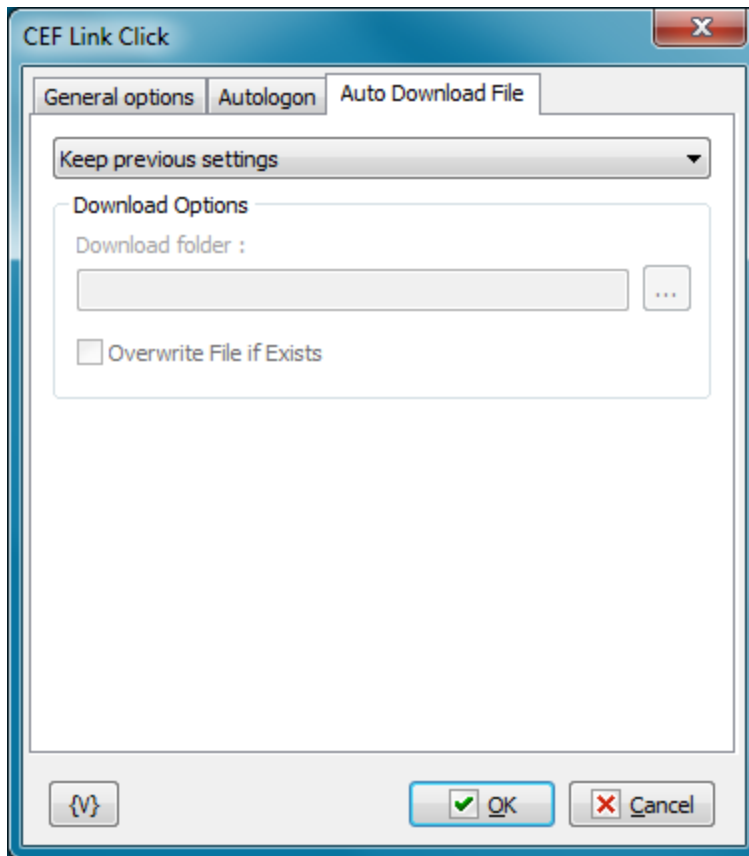
A password for authorization. Variables can be used.

## **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.



### Auto-download mode

There are two options

- Keep previous settings
- Change download options - set the *folder name* and *overwrite* parameter.

Default download folder is ***%temp%\cefdownloads***

### Download Folder

Specify the folder in which the file will be downloaded automatically. It will be downloaded with the file name by default. Default value is ***%temp%\cefdownloads***

### 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. Default value is ***OFF***

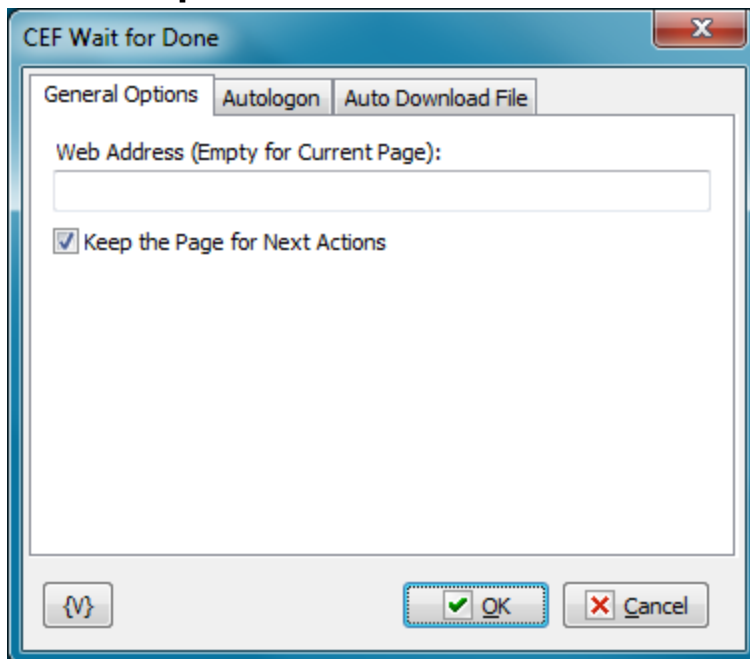
## 4.23.4 Browser Wait for Done

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.

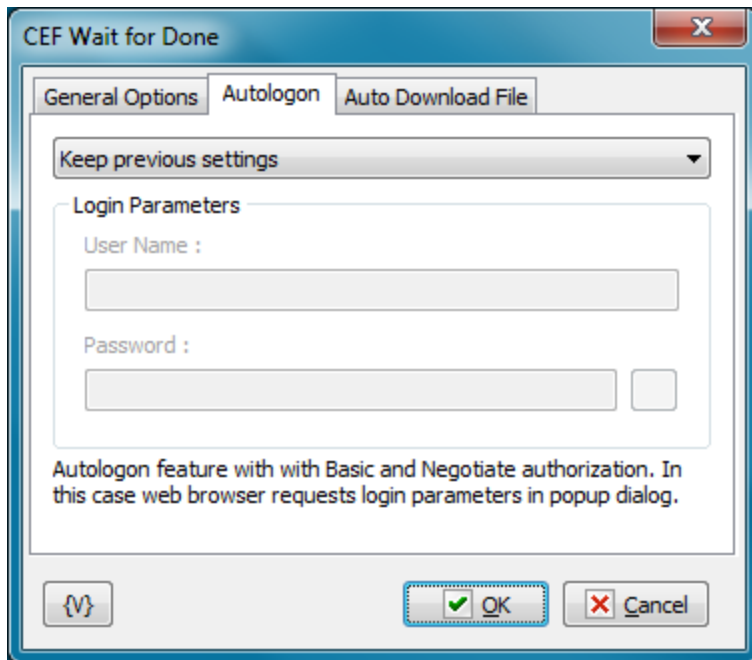
### Keep Page for Next Actions

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

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.



### Autologon Mode

There are three options

- **Keep previous settings**
- **No autologin (manual login)** - user have to login manually when web browser ask *username* and *password*
- **Enable autologin. Set username and password.**

### User name

A name of a user for authorization. Variables can be used.

### Password

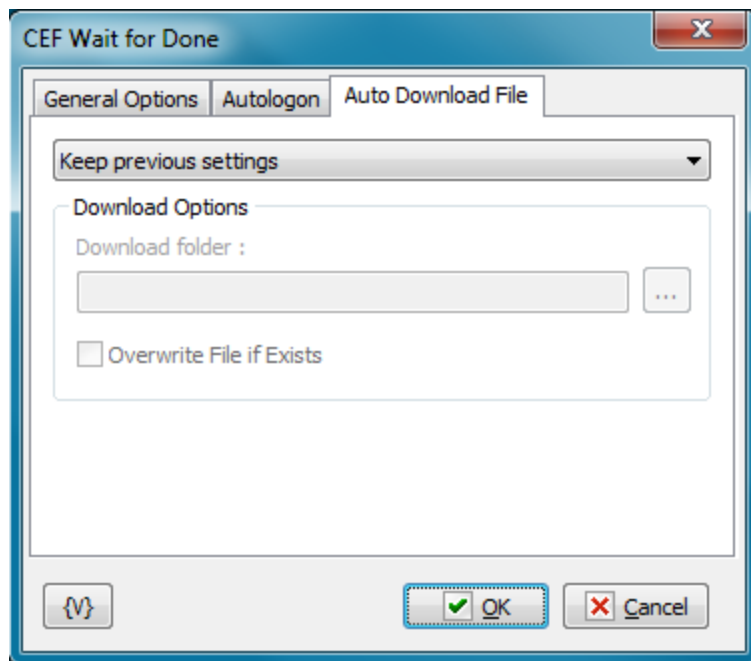
A password for authorization. Variables can be used.

### 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.



### Auto-download mode

There are two options

- Keep previous settings
- Change download options - set the *folder name* and *overwrite* parameter.

Default download folder is ***%temp%\cefdownloads***

### Download Folder

Specify the folder in which the file will be downloaded automatically. It will be downloaded with the file name by default. Default value is ***%temp%\cefdownloads***

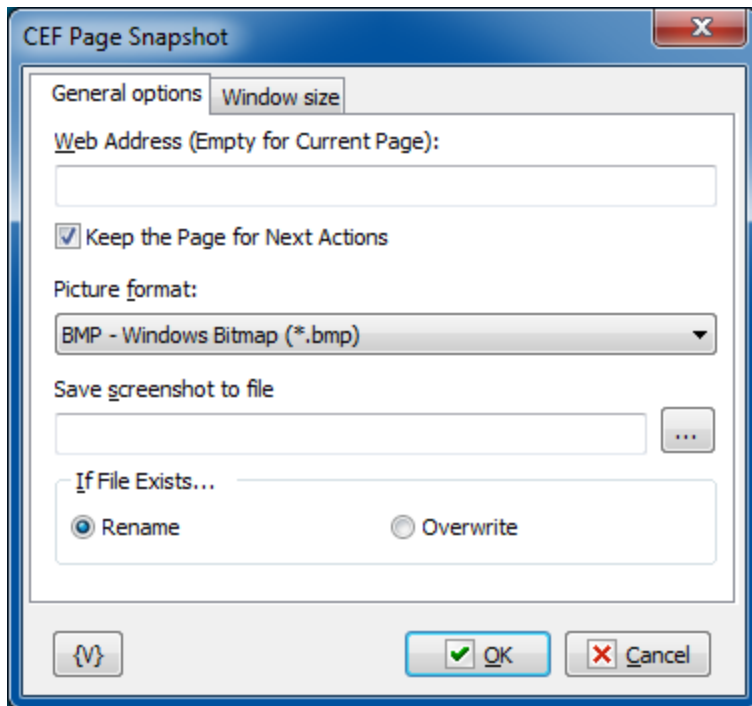
### Overwrite File If Exists.

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. Default value is ***OFF***

## 4.23.5 Browser Page Snapshot

The problem is that most of the browsers do not allow you to take a snapshot of the browser window correctly. The CEF Page Snapshot action solves this problem because the page is loaded under the control of RoboTask in the built-in browser.

### General tab



### 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.

### Keep page for next actions

If you need the downloaded page to be available in the following steps, switch on this parameter. Otherwise, the page will be deleted from the memory immediately after performing this action. Default value is ON.

### Picture Format

Screenshot is a graphic image. It can be saved in the different formats:

- **BMP** – standard Windows bitmap;
- **Jpeg** – Joint Photographic Experts Group format;
- **PNG** – portable network graphics format
- **GIF** – Graphics Interchange Format

### Save Screenshot to File

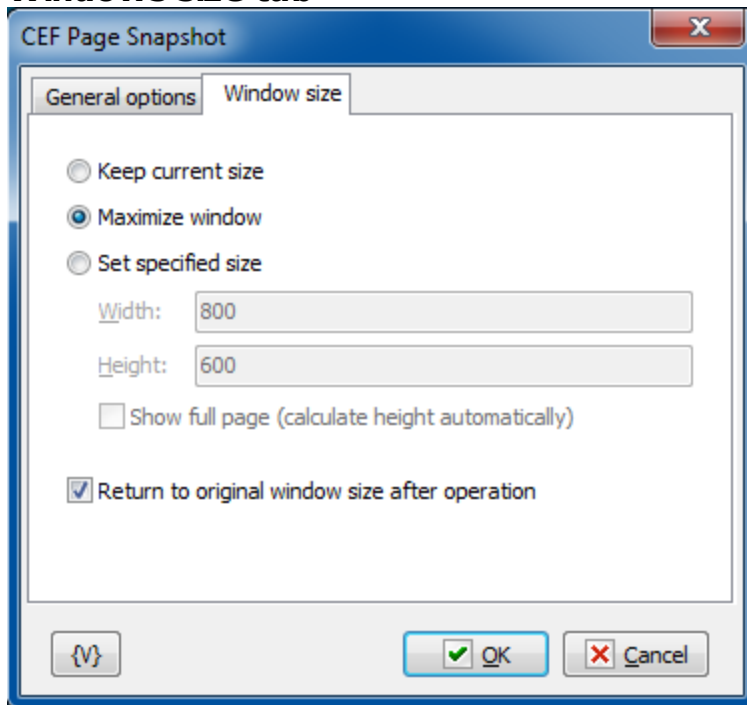
Specify the file name into which the snapshot needs to be saved.

### If File Exists

Action if the specified file exists:

- **Rename** - saves the file under a different name by adding the number to the file name. Thus the existing file will be left unchanged.
- **Overwrite** - overwrites the specified file.

### Windows size tab



Use necessary parameters if you want to change browser window size before taking a snapshot.

**Keep current size** - the window size will not change

**Maximize window** - action maximizes the browser window before taking a picture (default value)

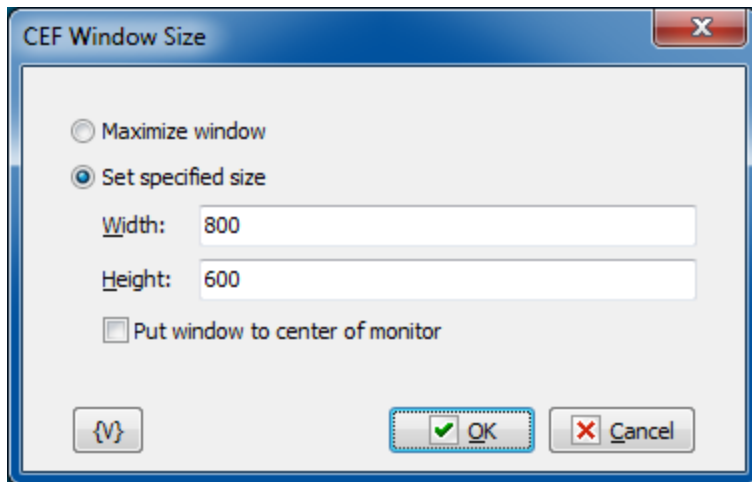
**Set specified size** - specify necessary size of the window

**Show full page** - the browser calculates the full height of the page and displays the entire page using the current width

**Return to original window size** - the action returns previous size of the window after operation

#### 4.23.6 Browser Window Size

This action allows to change internal browser window size.



**Maximize window** - maximize the browser window

**Set specified size** - set necessary height and width of the browser window

**Put window to center** - put window to center of current monitor after resizing

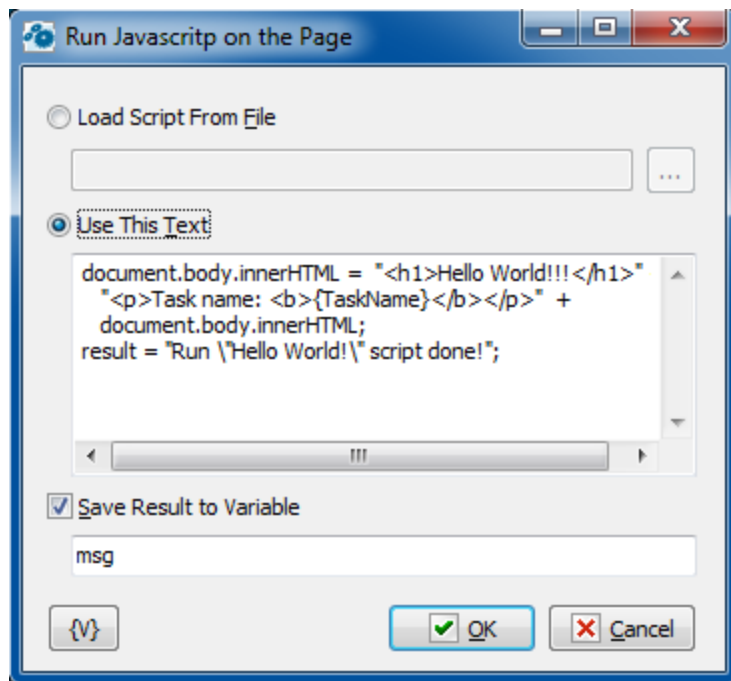
#### 4.23.7 Browser Run Javascript

**Note:** *This feature is available for Business license only*

This action allows to run some Javascript in opened page in Chromium browser. You can change the page, get some information from page or perform some actions on the page. Of course you have to know Javascript and object model of browser and structure of the page.

Web page must be loaded in internal browser before performing of this action. You can use [CEF Web Page Information](#)<sup>[349]</sup> or [CEF Fill WebForm](#)<sup>[342]</sup>

If you want to save some result from script you have to set this value to variable **result**. By default the result contains string "Result is undefined".



**Load script from file** - load Javascript from external text file

**Use this text** - type the text of script here in text editor

**Save result to variable** - assign the value of *result* variable into some user variable. Of course you have to assign necessary value in the script.

## 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:

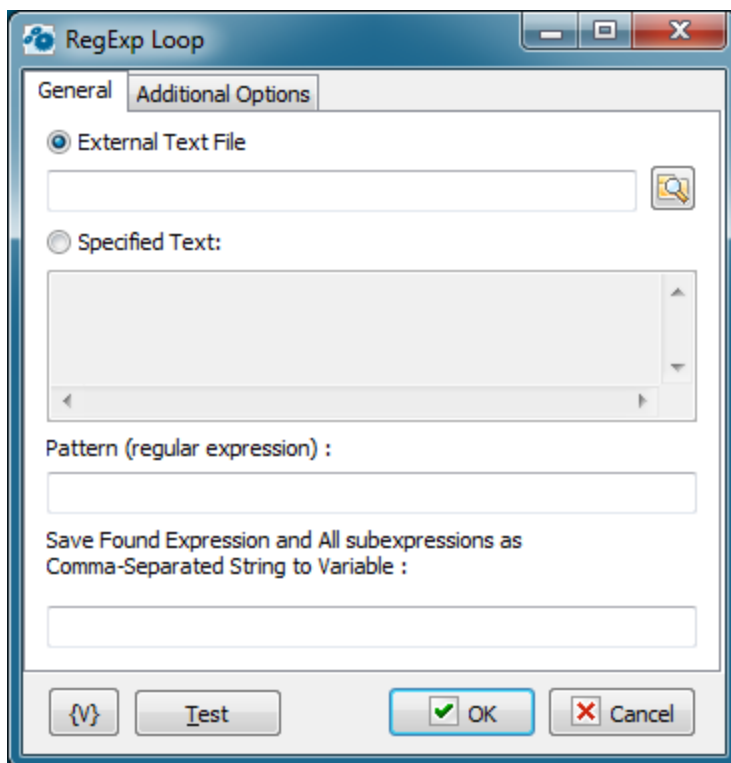
- The link address
- The link description

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)}

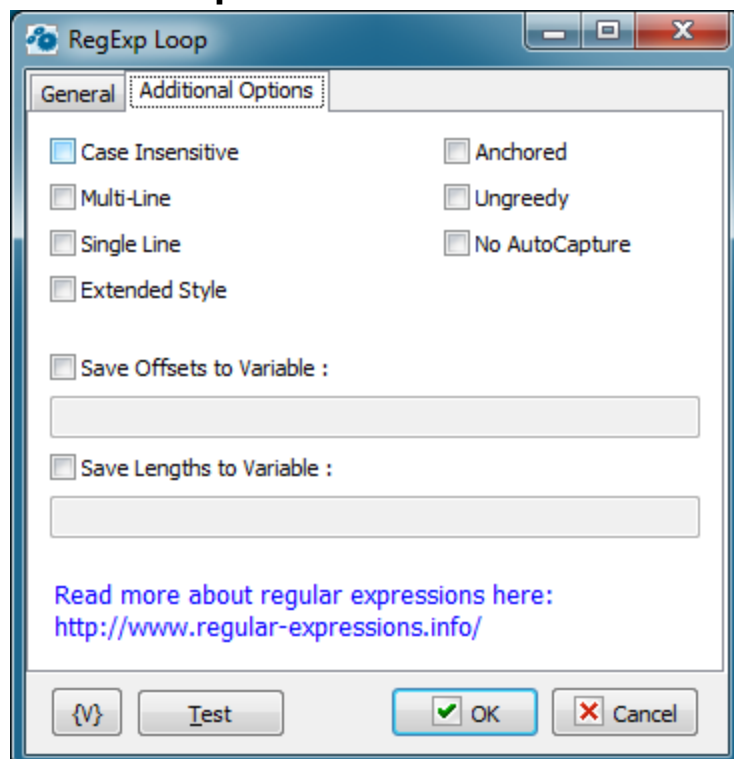
{MyVar(1)}

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.

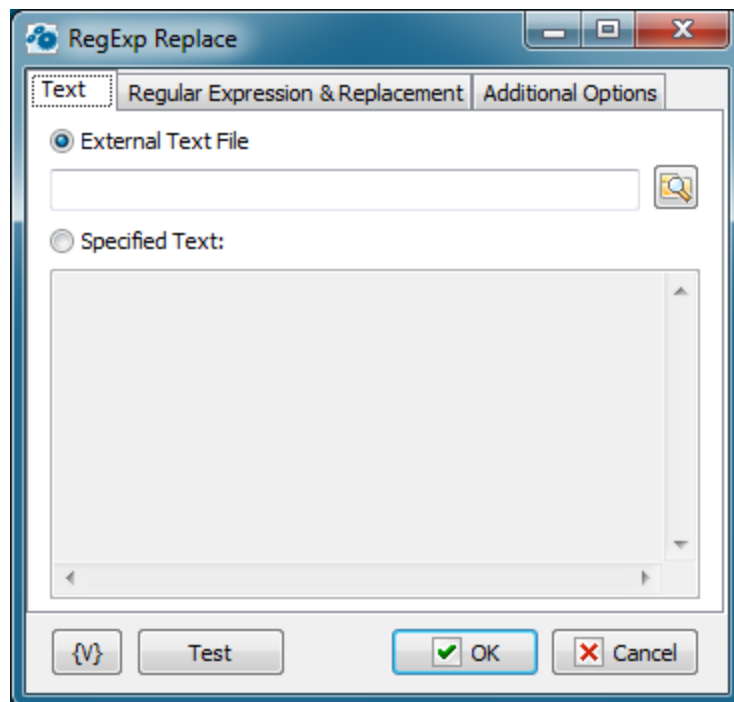
**Save Lengths to Variables**

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



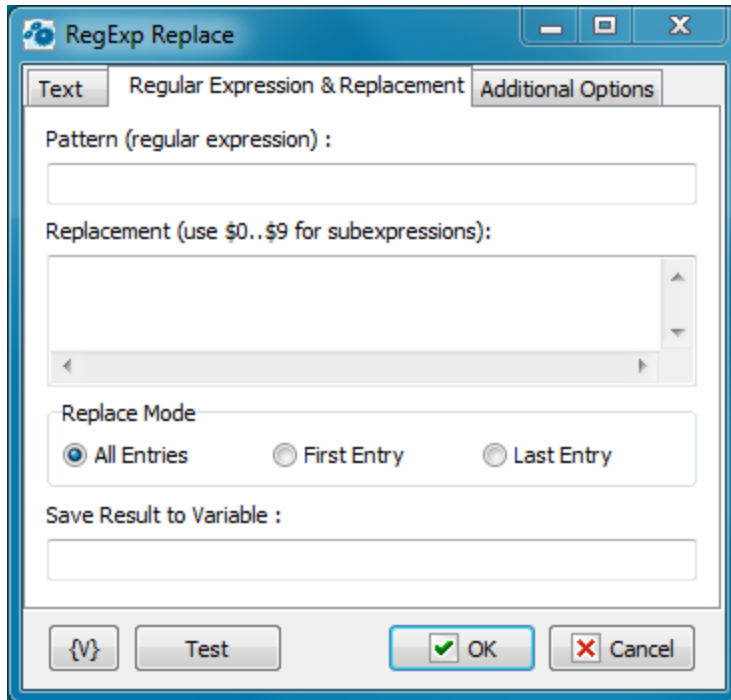
### External Text

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"*

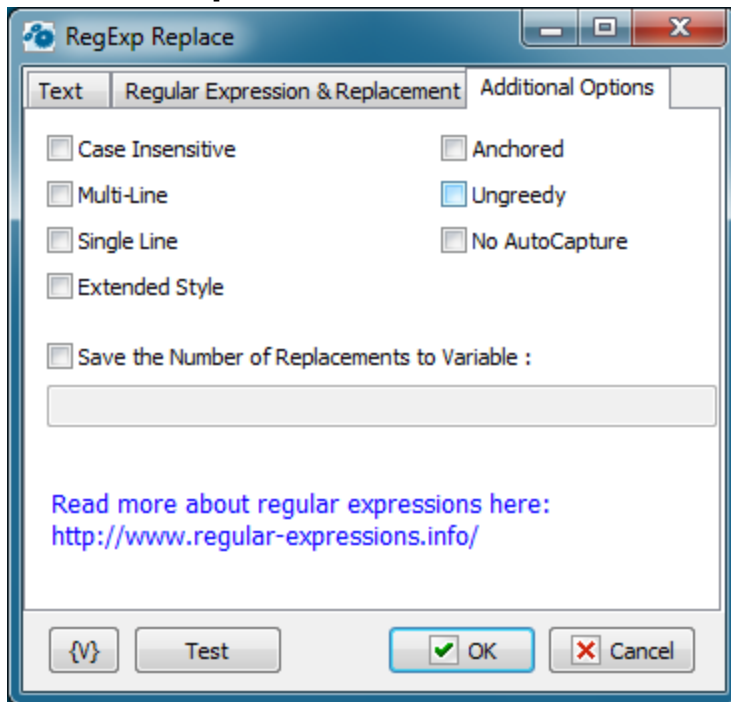
### Replace Mode

- *All entries* - all matched entries will be replaced
- *First entry* - the action make replacement only at first entry
- *Last Entry* - the action make replacement only at last entry

### 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.

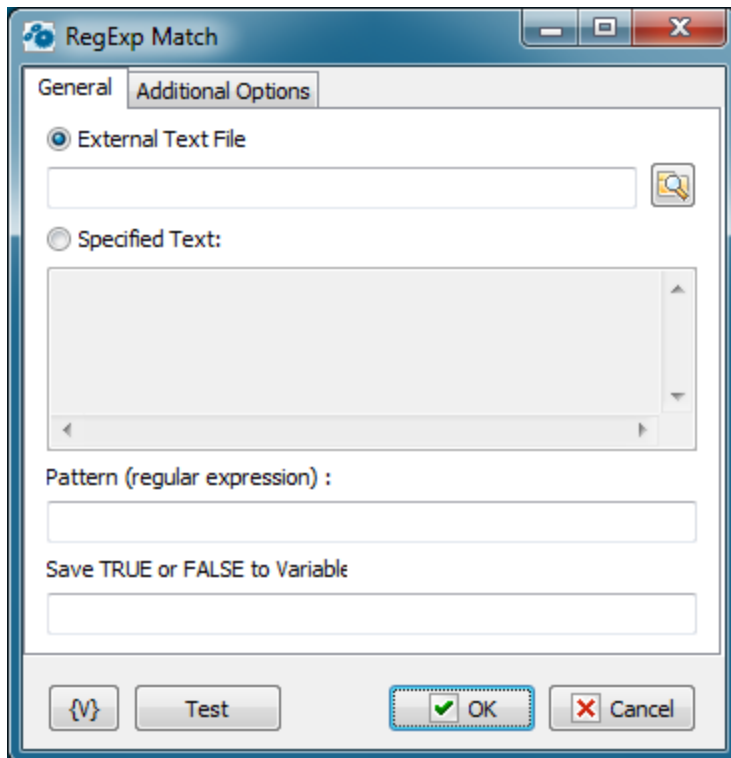
### Save the number of Replacements to Variable

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.

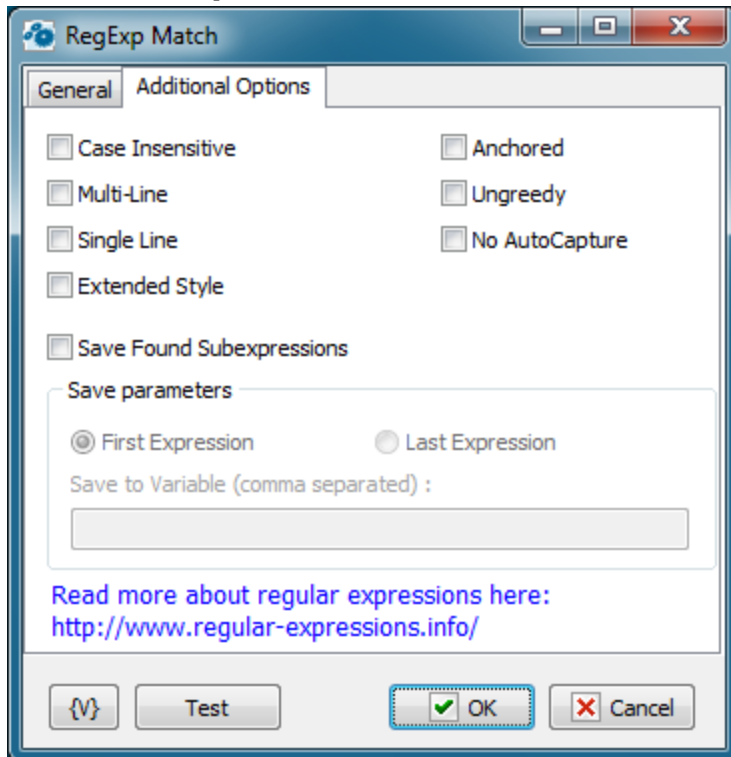
#### Pattern (Regular Expression)

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.

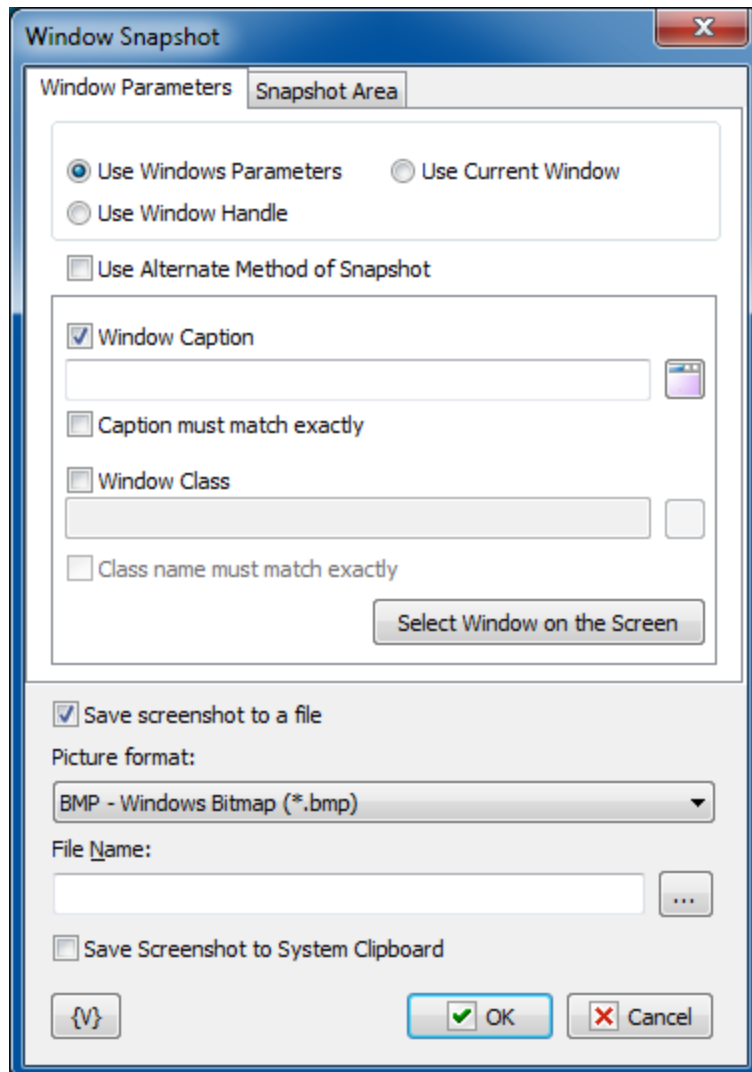
## 4.25 Screenshot

This plug-in unites actions of taking a screenshots, snapshots of necessary window or color of pixel on the screen or in the window.

### 4.25.1 Window Snapshot

This action allows to take a snapshot of a window and save the window image into a file.

**Window parameters.**



This action allows to use three modes of window selection to take a snapshot:

- **Use Window Parameters** – search of a window according to parameters (Window Caption, Window class);
- **Use Windows Handle** – according to the Handle of the window. Each window in a system has its unique Handle. It can be got in the loop Window Loop. For example, the following task can be done: to take all snapshots of windows which have the word Notepad in their captions.
- **Use Current window** – to use the current active window.

When using the mode Use Window Parameters, the parameters of the window for a search are available to the user.

### Use alternate method of Snapshot

Generally, getting a window image works correctly. But, unfortunately, not always. Some applications do not allow you to take a picture directly from the window. For example, *Google Chrome* or applications with the *Metro user interface*. To get the image, you must use an alternative method in this case.

---

An alternative method is that the application cuts a window image from a desktop snapshot. But please note that this method cannot get the correct image if the desired window (or part of the window) is covered by another window or goes beyond the desktop. You need to ensure the visibility of the window before taking a snapshot.

### **Window Caption**

Switch on this checkbox if you want to use the Window Caption to search a window. Then fill in the field for the search of the Window Caption. The field can be filled in automatically when you select the window from the window list. You need to click the button to the right of the entry field. The usage of variables is allowed.

### **Window caption must match exactly**

If you want a Window Caption to be exactly matched to the input string, switch on this checkbox.

### **Window Class**

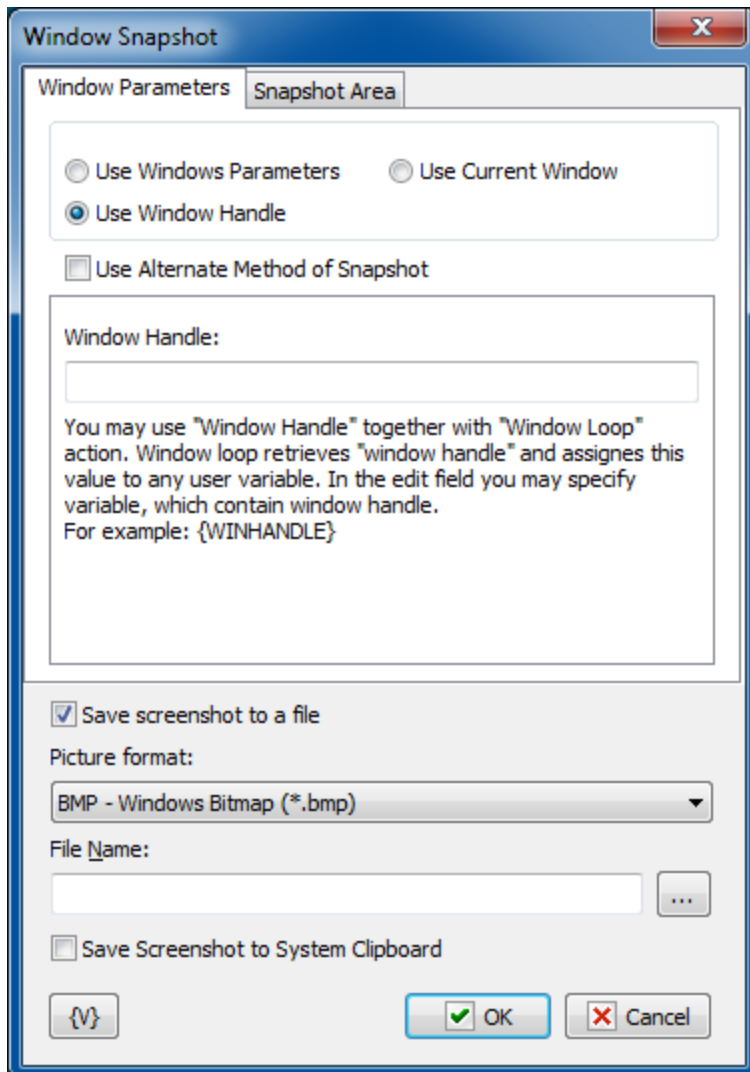
Switch on this checkbox if you want to use the Window Class for a search of a window. Then, fill in the field for the search of the Window Class. The field can be filled in automatically when you select the window from the window list. You need to click the button to the right of the entry field. The usage of variables is allowed.

### **Window class must match exactly**

If you want a Window Class to be exactly matched to the input string, switch on this checkbox.

### **Select Window on the screen**

Using this button you can select any window in a screen and fill in fields automatically. Just direct the mouse cursor onto the necessary window and click Ctrl to select the window or the Esc to cancel the operation. When moving the mouse cursor, the current window is highlighted in a red frame. You can use the mouse to manipulate with windows.

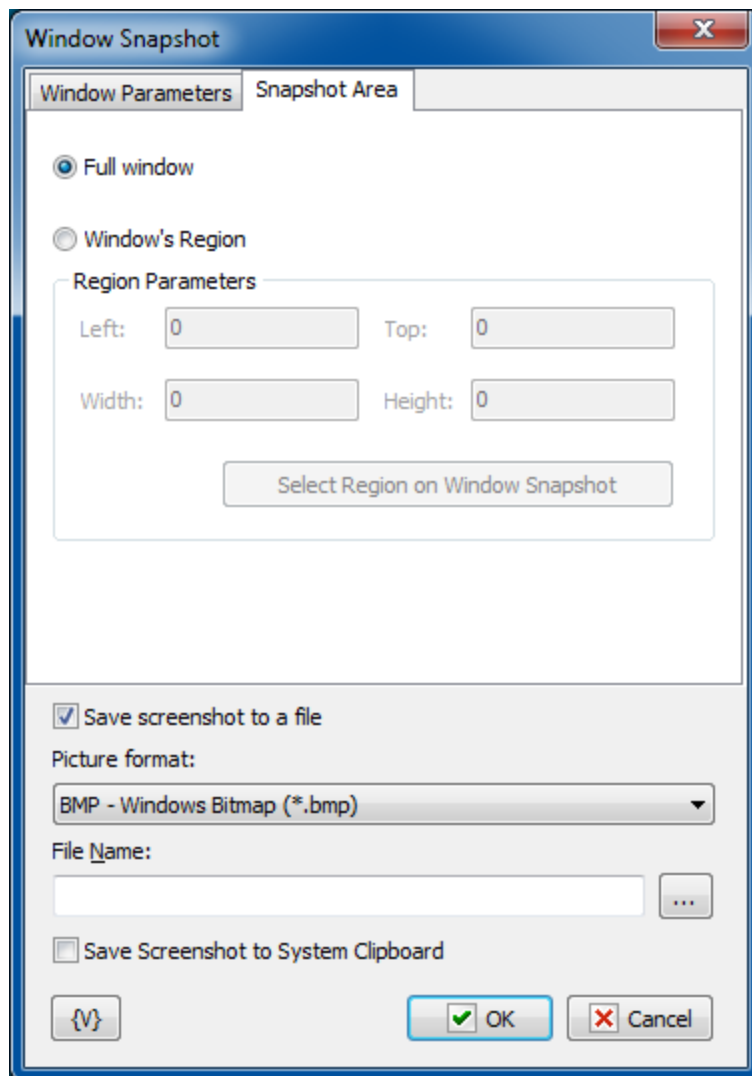


### Window Handle

When using of the mode Use Window Handle, this parameter is available to the user. The Window Handle identifies the window explicitly. It is convenient to use this mode in the Window Loop. The loop allows to perform the selection of windows according to necessary criteria and save the window handle into a variable. By means of the window handle it is possible to take a snapshot

### Capture mode

The action allows to take snapshot of the whole window or of a part of him.



### Full Window.

Select this point if you want to take the screenshot of the whole window.

### Window Region.

If you want to take the screenshot only of a part of a window, select this point and input the position data of a rectangular region of the window: **Left**, **Top**, **Height**, **Width**. You can also specify the rectangular region on a screenshot interactive clicking the button **Select Region on Window Snapshot**.

### Picture Format

Snapshot of a window is a graphic image. It can be saved in the different formats:

- **BMP** – standard Windows bitmap;
- **Jpeg** – Joint Photographic Experts Group format;
- **PNG** – portable network graphics format
- **GIF** – Graphics Interchange Format

### Save the snapshot to file

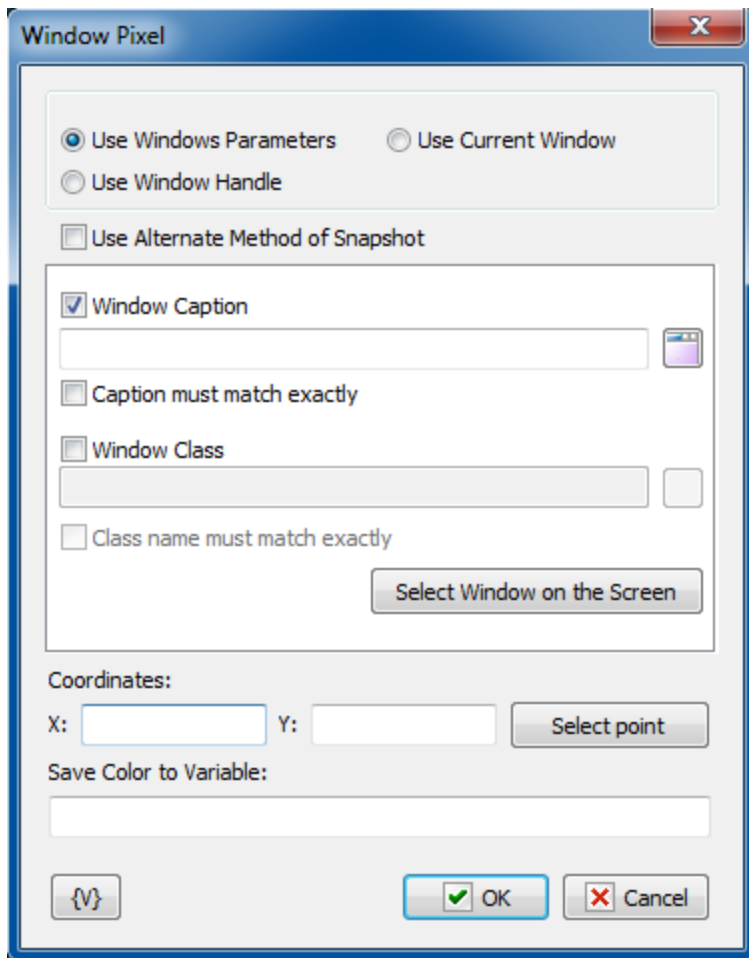
Specify the file name into which the snapshot need to be saved.

### Save screenshot to system clipboard

Turn on this checkbox if you want to save the snapshot into system clipboard

## 4.25.2 Window Pixel

This action allows to get the color of a window pixel according to the specified coordinates and save it into a variable. The color of a pixel is a number. It can be compared with a sample like an ordinary number or with another color using a tolerance.



### Window Selection Mode

The action allows to use three selection modes of a window to take a screenshot:

- **Use Window Parameters** – the search of a window according parameters (Window Caption, Window class);



- **Use Window Handle** – according the Window Handle. Each window in a system has the unique Handle. It can be got in the loop Window Loop. For example, the following task can be made: to take all screenshots, which have the word Notepad in their captions.
- **Use Current window** – use the current active window.

### Use alternate method of Snapshot

Generally, getting a window image works correctly. But, unfortunately, not always. Some applications do not allow you to take a picture directly from the window. For example, *Google Chrome* or applications with the *Metro user interface*. To get the image, you must use an alternative method in this case.

An alternative method is that the application cuts a window image from a desktop snapshot.

But please note that this method cannot get the correct image if the desired window (or part of the window) is covered by another window or goes beyond the desktop.

You need to ensure the visibility of the window before taking a snapshot.

### Window Caption

Switch on this checkbox if you want to use the Window Caption for a search of a window. Then, fill in the field for the search of the Window Caption. The field can be filled in automatically when you select the window from the window list. You need to click the button to the right of the entry field. The usage of variables is allowed.

### Window caption must match exactly

If it is necessary the Window Caption to be match exactly with the entry string, switch on this checkbox.

### Window Class

Switch on this checkbox if you want to use the Window Class for a search of a window. Then, fill in the field for the search of the Window Class. The field can be filled in automatically when you select the window from the window list. You need to click the button to the right of the entry field. The usage of variables is allowed.

### Window class must match exactly

If it is necessary the Window Class to be match exactly with the entry string, switch on this checkbox.

### Select Window on the screen

Using this button you can select any window in a screen and fill in fields automatically. Just direct the mouse cursor on the necessary window and click Ctrl to select the window or Esc to cancel the operation. When moving the mouse cursor, the active window is highlighted in a red frame. You can use the mouse to manipulate with windows.

### Window Handle

When using of the mode Use Window Handle, this parameter is available to the user. The Window Handle identifies the window explicitly. It is convenient to use this mode in the Window Loop. The loop allows to perform the selection of windows according to necessary

criteria and save the window handle into a variable. By means of the window handle it is possible to take a snapshot

### Coordinates

Specify the coordinates of a pixel from the top left corner of a window snapshot. The usage of variables in the expression is allowed. You can specify the coordinates interactive clicking the button Select Point.

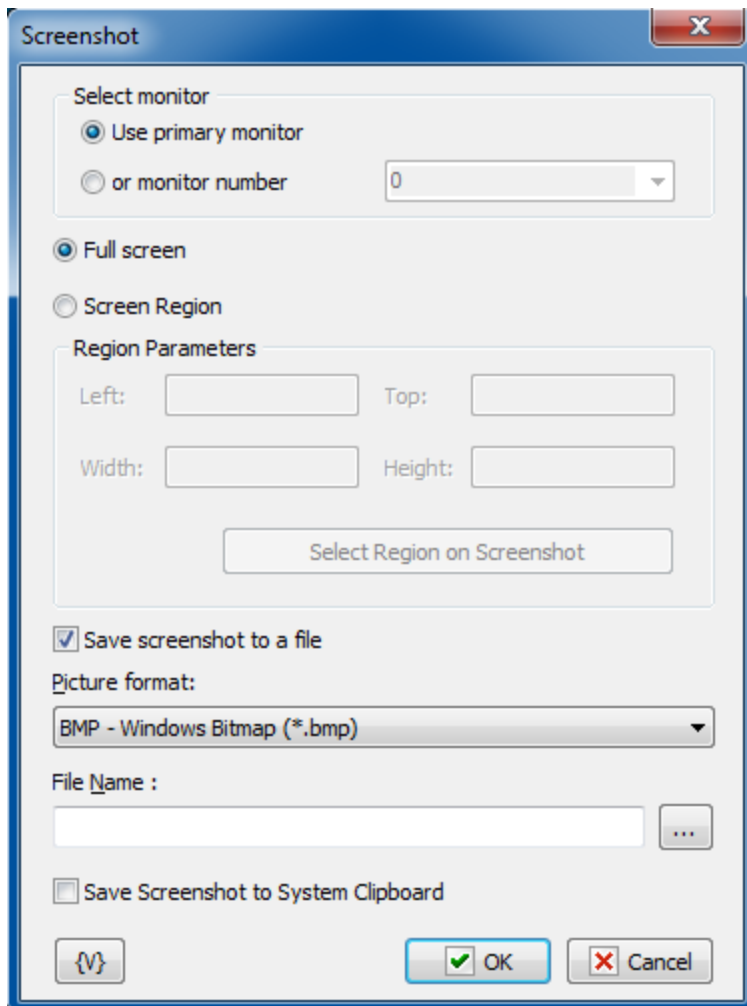
### Save Color To Variable

Specify the name of the variable in which it is necessary to save the color. The color of a pixel is a number. For the convenience of sensing, the action saves color in a hexadecimal look.

For example, \$FFFF00, \$800000

## 4.25.3 Screenshot

The action allows to take a snapshot of the whole screen (or of a part of it) and save it into a file.



### Select monitor

There are two options

**Use primary monitor** - take a snapshot from primary monitor

**Monitor number** - specify the number of monitor (from 0)

### Full screen

Select this option if you need to take the snapshot of a whole screen.

### Screen region

Select this option if it is necessary to save the snapshot of a rectangular region of the screen.

Determine the coordinates of the region: **Left, Top, Height, Width.**

You can determine the coordinates of a rectangular region interactive clicking the button "Select Region on Screenshot".

### Picture Format

Screenshot is a graphic image. It can be saved in the different formats:

- **BMP** – standard Windows bitmap;
- **Jpeg** – Joint Photographic Experts Group format;
- **PNG** – portable network graphics format
- **GIF** – Graphics Interchange Format

### Save screenshot to file

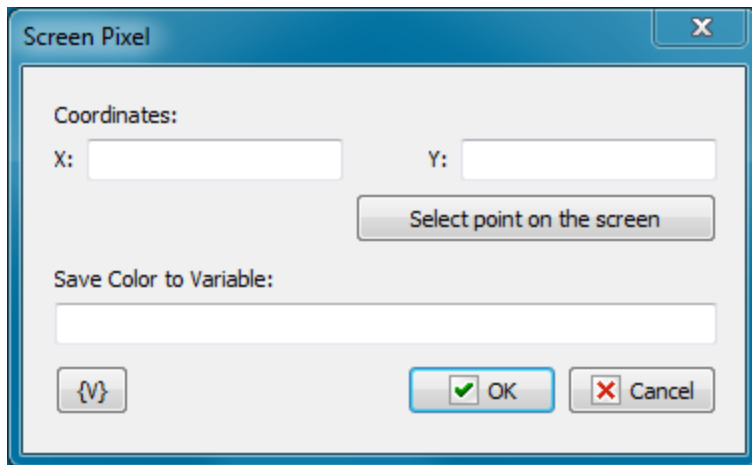
Specify the file name into which the snapshot need to be saved.

### Save screenshot to system clipboard

Turn on this checkbox if you want to save the snapshot into system clipboard

## 4.25.4 Screen Pixel

This action allows to get the color of a window pixel according to the specified coordinates and save it into a variable. The color of a pixel is a number. It can be compared with a sample like an ordinary number or with another color using a tolerance.



### **Coordinates**

Specify the coordinates of a pixel. The usage of variables is allowed. You can specify the coordinates interactive in the picture clicking the button "Select Point on the screen".

### **Save Color to Variable**

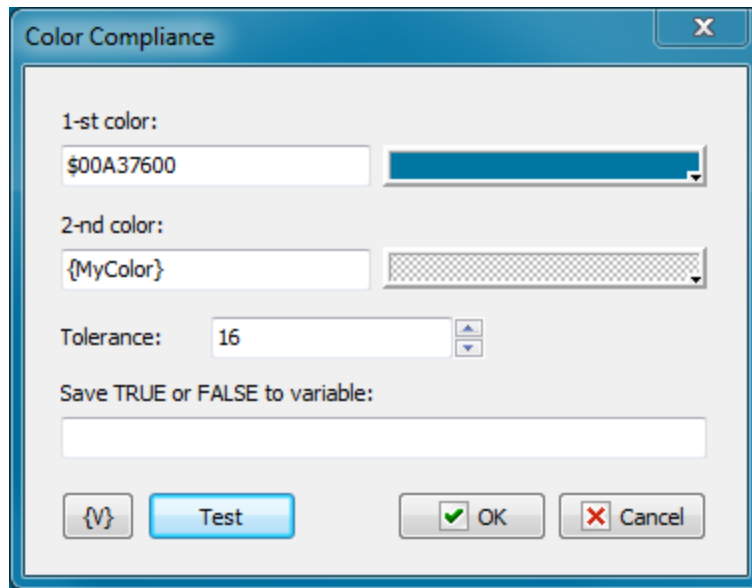
Specify the name of the variable in which it is necessary to save the color. The color of a pixel is a number. For the convenience of sensing, the action saves color in a hexadecimal look.

For example, \$FF80E0, \$808080

## **4.25.5 Color Compliance**

This action allows to compare a color with the standard sample color. To identify color compliance it can be specified some tolerance. 0 means exact compliance. The maximum value of a tolerance is 255. The maximum value of a tolerance means that the result of the compliance check of any two color always is TRUE.

The compare result can be saved into a variable for the next processing in following steps of the task.



### 1-st color, 2-nd color

Specify two colors for comparing. You can specify the color as an ordinary number or in a hexadecimal look (beginning from the symbol \$). The color can be selected from a palette or from any point of a screen clicking the button to the right of the edit field. The usage of variables is allowed.

### Tolerance

Specify the necessary tolerance to compare two colors. On default, it is 16. 0 corresponds to the exactly coincidence.

255 is the maximum value. When a value is maximum the result of comparing of any two colors is TRUE.

### Save TRUE or FALSE to variable

Input the name of the variable in which it is necessary to save the result of the comparing.

## 4.25.6 Wait for Pixels

The action waits for the change of pixels color on a screen or in an appointed window. This is a standard way to identify finishing some operation in a system: a picture or an inscription will change. Or some area of a window (or screen) will change its color.

The action pauses the executing of a task and waits for changing of pixels color on a screen or in a selected window. The action will continue to implement the task when the condition of correspondence of colors of the selected pixels is fulfilled.

You can specify a set of pixels (one or several) in this action. It is necessary to specify the required color and the tolerance of color comparing for each pixel.

You can specify the condition of color coincidence for a set of pixels:

- Match ALL of the following pixels (AND)

- Match ANY of the following pixels (OR)

Thereby, you can specify the exact enough condition of changes on a screen.

If the action has not waited for implementation of the condition for a predetermined time, the action generates the error "*Timeout error*".

If the required window has not been found to the moment of a start of a task, the action generates the error "*Window is not found*"

## **Source tab**

Wait for Pixels

Source Pixels

Window  Screen

Use Windows Parameters  Use Current Window

Use Window Handle

Use Alternate Method of Snapshot

Window Caption

Caption must match exactly

Window Class

Class name must match exactly

Select Window on the Screen

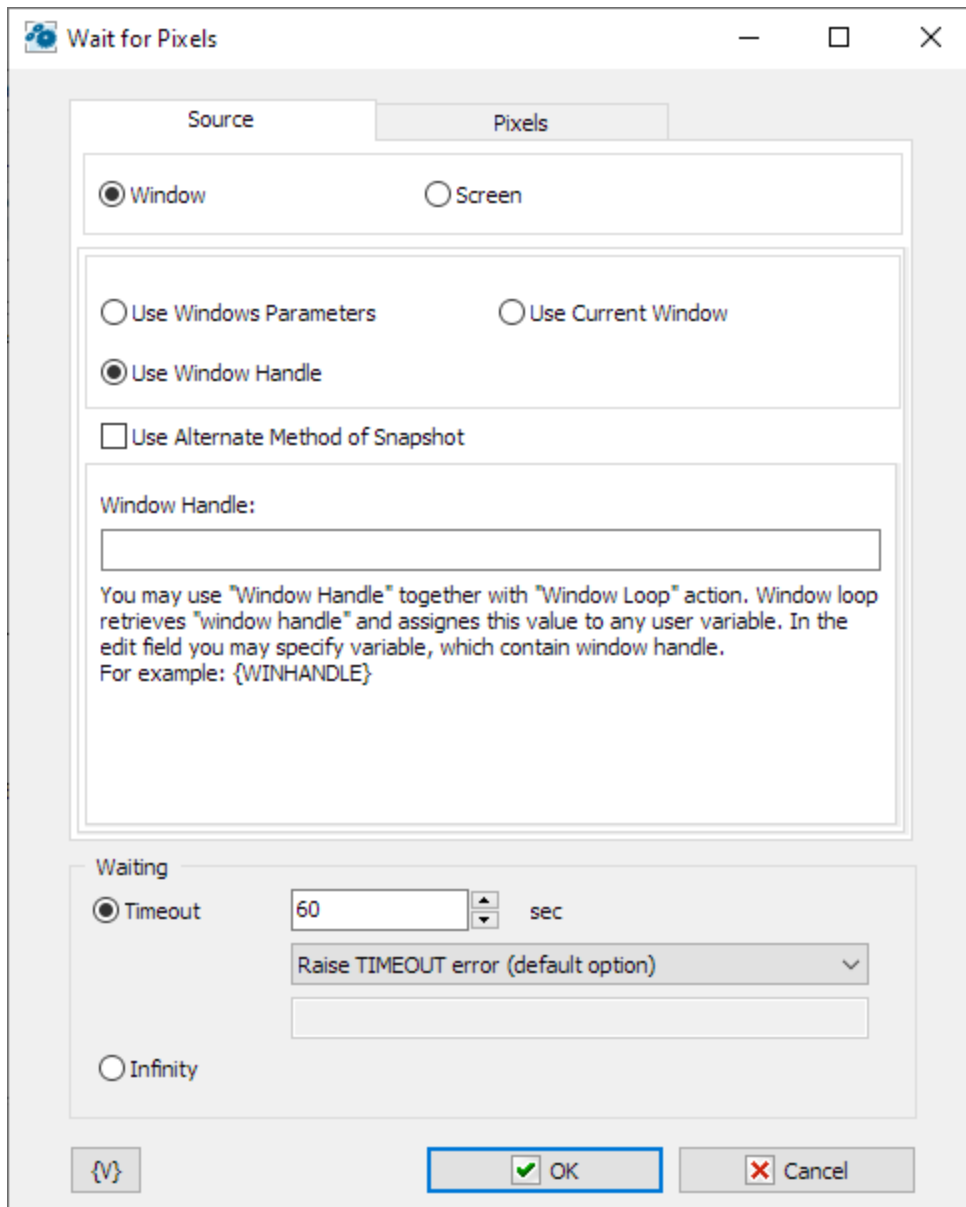
Waiting

Timeout  Infinity

60 sec

Raise TIMEOUT error (default option)

{V} OK Cancel



When you determine parameters of an action, at first you need to specify the source of a picture:

- **Window** - This is a separate window. All coordinates of pixels will be determined concerning the upper left corner of this window. If you have selected the window as a source, you need to specify the window parameters.
- **Entire screen** - This is a screen with all visible windows.

The window parameters are similar to the parameters in the actions [Window Snapshot](#)<sup>[376]</sup>, [Window Command](#)<sup>[252]</sup>, [Window Size and Position](#)<sup>[259]</sup>, [Window Elements loop](#)<sup>[262]</sup>.

### Window Parameters Mode

- **Use Window parameters** - Use the Window Caption and the Window Class to look for the necessary window.



- **Use Window Handle** - This is a search for a window by a handle. The handle can be got in the loop Window Loop, from the trigger Wait for Window or some other way.
- **Use current window** - This is a current window in a moment of the executing of the action. This mode does not require any additional parameters.

### Use alternate method of Snapshot

Generally, getting a window image works correctly. But, unfortunately, not always. Some applications do not allow you to take a picture directly from the window. For example, *Google Chrome* or applications with the *Metro user interface*. To get the image, you must use an alternative method in this case.

An alternative method is that the application cuts a window image from a desktop snapshot.

But please note that this method cannot get the correct image if the desired window (or part of the window) is covered by another window or goes beyond the desktop.

You need to ensure the visibility of the window before taking a snapshot.

### Use Window parameters

#### Window Caption.

If you want to search for a window by means of its caption, switch on this checkbox and input the expected window caption. You can input only part of the caption. In this case the action will check availability of the input part in the window caption.

#### Caption Must Match Exactly

If you need the exact coincidence of a window caption, switch on this checkbox.

#### Window Class.

If you want to search the window by a window class, switch on this checkbox and input the expected name of the window class. You can input only part of the name. In this case the action will check availability of the input line in the name of the window class.

#### Class Name Must Match Exactly.

If you need the exact coincidence of a window class, switch on this checkbox. Usually it is worthwhile to search by the whole class name, because a window class is permanent and does not change its value during the "life" of the window, unlike a caption.

### Select Window on the Screen.

When pressing this button, you turn on the mechanism of selection a window on the screen. The window, which the mouse pointer points on, highlights with a red frame. Meanwhile, you can use the mouse to perform some actions. To select the necessary window, press *Ctrl*. To cancel - press *Esc*. When selecting a window, the fields Window Caption and Window Class are filled in automatically.

### Window Handle.

Input the expression containing a window handle. This parameter is used for the mode Use Window Handle. A window handle is just a number, identifier of a window in a current session of system work.

## Pixels tab

You need to specify the pixel set for checking on this tab.

The screenshot shows the 'Wait for Pixels' dialog box with the 'Pixels' tab selected. The dialog is divided into several sections:

- Match Logic:** Two radio buttons are present: 'Match ALL of the following pixels (AND)' (which is selected) and 'Match ANY of the following pixels (OR)'.
- Pixel List Table:** A table with three columns: 'Coordinates', 'Color', and 'Tolerance'. The table is currently empty. Below the table are three buttons: '+ Add', 'Edit', and 'Remove'.
- Waiting Section:** Contains two radio buttons: 'Timeout' (selected) and 'Infinity'. The 'Timeout' section includes a text input field with the value '60' and the unit 'sec', and a dropdown menu currently showing 'Raise TIMEOUT error (default option)'.
- Bottom Buttons:** A 'Cancel' button with a red 'X' icon, an 'OK' button with a green checkmark icon (highlighted with a blue border), and a button with a curly brace icon.

### List of pixels

You can edit a list using the buttons *Add*, *Edit* and *Remove*.

By means of an edit window you can change pixel parameters both manually and by means of interactive instruments.

Pixel parameters:

**X, Y** - It is coordinates of a pixel from the left corner. If a source of a picture is a window, coordinates are determined from the upper left corner of the screen window.

**Pattern color** - This is a pattern of the color with which the action will compare the color of a pixel. Color is just a number. For comfort it is written in hexadecimal.

For example, \$0066FF00 or \$66FF00.

In hexadecimal you can write color using 6 symbols (as RGB): the first couple (younger) is red, the second couple is green, the third couple (elder) is blue color.

**Tolerance** - This is an accuracy of color coincidence. The error can be changed from 0 to 255 (in hex from 00 to FF). When it is 0, it is exact color coincidence. When it is 255, it is a coincidence with any color.

Tolerance is an acceptable difference between two colors for any component RGB (Red - Green - Blue). Every component of the color can be changed from 0 to 255 (from 00 to FF in hexadecimal).

On default, the tolerance is 16.

### Waiting

There are two options:

- **Timeout** - It is waiting time in seconds. On default, it is 60 seconds.

If the timeout is over you can specify an action:

- **Raise Timeout error** - this is default option
- **Set True or False to variable** - in this case the task will continue work without error.
- **Infinity** - This is not limited waiting time. Pay attention, if you specify an illegal condition of the checking of pixels, the task can "hang down" and it can be finished by means of the function "Kill task" only.

### 4.25.7 Find Pixel on Image

The action finds the specified color on the image. The source image can be the area on desktop, image from clipboard or an image from file.

Find pixel on image ✕

Choose source image Color matching and saving the result

Source image

From screen  From file

From clipboard

Top:  Left:

Height:  Width:


Select area from screen

{V}  OK  Cancel

Find pixel on image ✕

Choose source image Color matching and saving the result

Find color:



Tolerance:

Save TRUE or FALSE to variable:

Save coordinates as JSON ({"top"="yy", "left"="xx"}):

{V}  OK  Cancel

### Source image

You can choose

- *From screen* - you must specify coordinates of area on your desktop. Also you can select area interactively, just press "Select area from screen" button.

- *From clipboard* - in this case system clipboard must contain an image. Otherwise the action will throw an error.
- *From file* - the action load the image from file.

### Find color

Specify the necessary color. You can specify the color as an ordinary number or in a hexadecimal look (beginning from the symbol \$). The color can be selected from a palette or from any point of a screen clicking the button to the right of the edit field. The usage of variables is allowed.

### Tolerance

Specify the necessary tolerance to compare two colors. On default, it is 16. 0 corresponds to the exactly coincidence.

255 is the maximum value. When a value is maximum the result of comparing of any two colors is TRUE.

### Save TRUE or FALSE to variable

Enter the name of the variable in which you want to save the search result. *TRUE* if specified color is found, *FALSE* if not.

### Save coordinates as JSON

You can store the coordinates of the first occurrence of a given color as JSON string.

## 4.26 INI files

INI files have been used as configuration files since the earliest Windows versions. These are plain text files that can be viewed and edited in any text editor and have the following format:

```
[section1]
var1=value1
var2=value2

[section2]
var1=value1
var2=value2

[section3]
var1=
var2=
```

The file consists of sections, which are enclosed in square brackets. The section names must be unique and must not be repeated. Each section contains parameters that consist of a name

and a value and are separated by an "=" (equal) character. Parameter names must be unique within their section. Values can be empty, i.e. a blank line after the "equal" character.

Section names, parameter names and parameter values must be single-line. Multiline values can be combined into a single line by using a delimiter line when saving. During reading such values can be restored to multiline values again. For example, *string1{eol}string2{eol}string3*, where *{eol}* is the separator.

It is still common practice to use INI files to save/read local configurations, which (in contrast with the registry) can be copied and moved around.

### 4.26.1 INI read value

Reads the value of one parameter from a specified INI file and saves it to a user-specified variable.

The screenshot shows a dialog box titled "INI Read Value" with a close button (X) in the top right corner. The dialog has two tabs: "General" and "Advanced". The "General" tab is selected. The dialog contains the following fields and controls:

- INI file name :** A text input field with a browse button (...).
- INI Section :** A dropdown menu with "Input section name" selected.
- Value name :** A dropdown menu with "Input key name" selected.
- Save value to variable :** A text input field with "Input variable name" entered.
- Default value :** A text input field with "Input default value" entered.

At the bottom of the dialog, there are three buttons: a folder icon, an "OK" button with a green checkmark, and a "Cancel" button with a red X.

## General

### INI file name

The full path to the file or file name in the current folder (please see Set Current Folder action).

### INI section

The name of a section. If absolute path is given in file name and file exists, can be selected existing name from dropdown list.

**Value name**

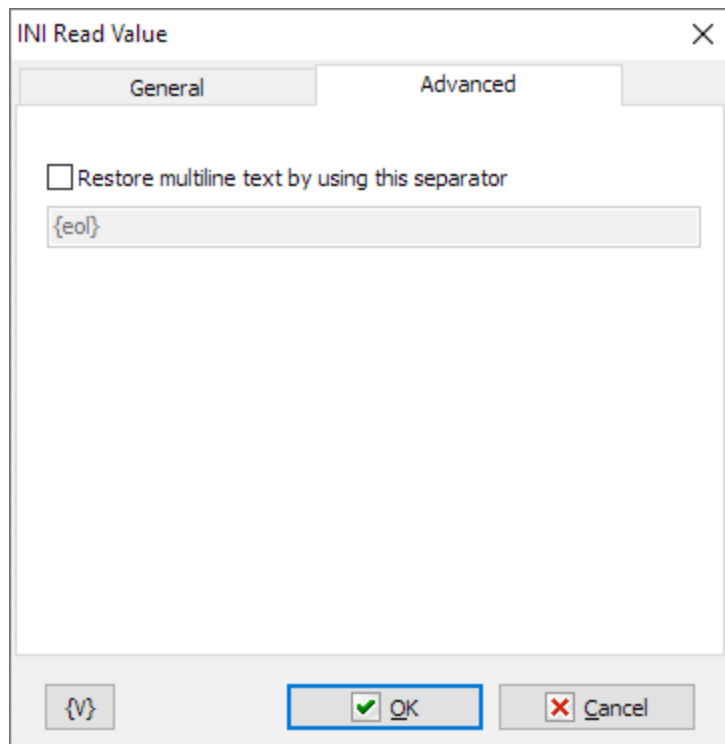
Parameter name. If absolute path is given in file name and file exists, can be selected existing name from dropdown list.

**Save value to variable**

The variable name, in which the value of the parameter will be saved.

**Default value**

The default value that will be saved in a variable, if the specified parameter does not exist.

**Advanced****Restore multiline text by using this separator**

A separator string for restoring multiline values.

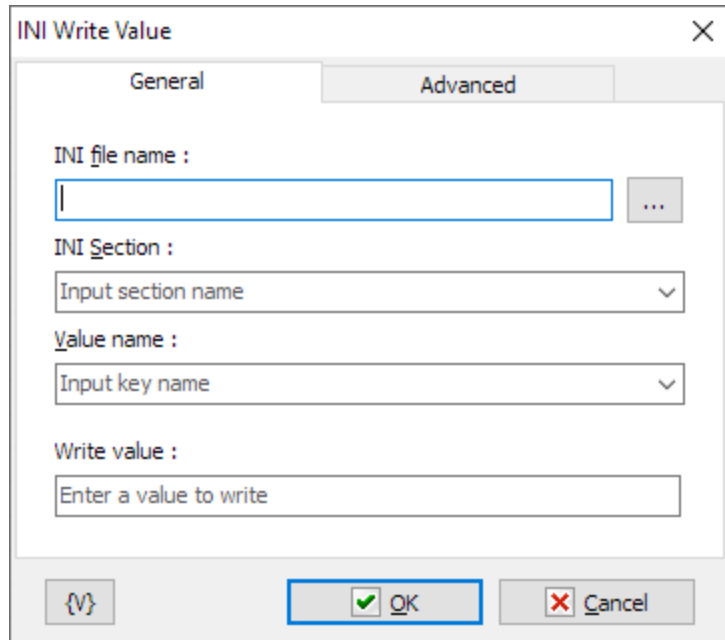
The INI file name, INI section, Value name fields can contain both absolute values and variables that will be replaced by its value when the action is performed.

**The {V} button.**

Allows you to select a system or user variable to insert into the current field.

## 4.26.2 INI write value

Saves the value of a single parameter in a specified INI file.



### General

#### **INI file name**

The full path to the file or file name in the current folder (please see Set Current Folder action).

#### **INI section**

The name of a section. If absolute path is given in file name and file exists, can be selected existing name from dropdown list.

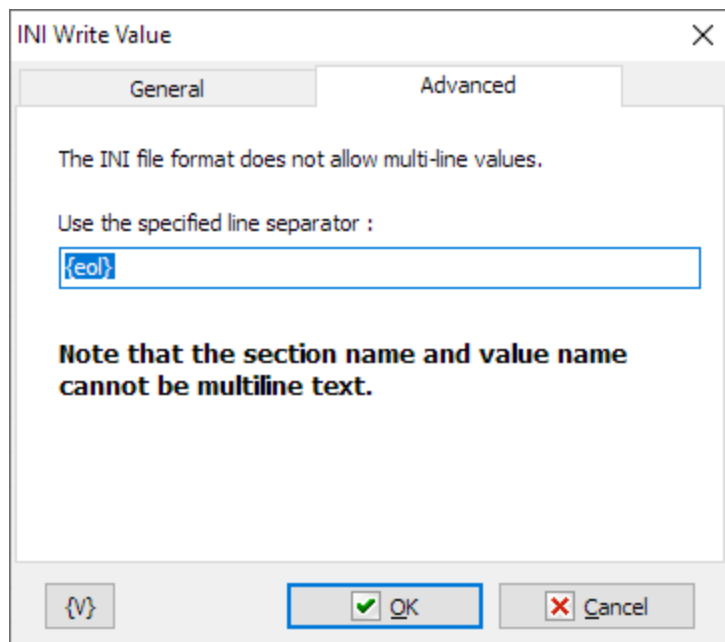
#### **Value name**

Parameter name. If absolute path is given in file name and file exists, can be selected existing name from dropdown list.

#### **Write value**

Enter the value to write.





## Advanced

### Use the specified line separator

The separator string to use when saving multi-line values is `{eol}` by default.

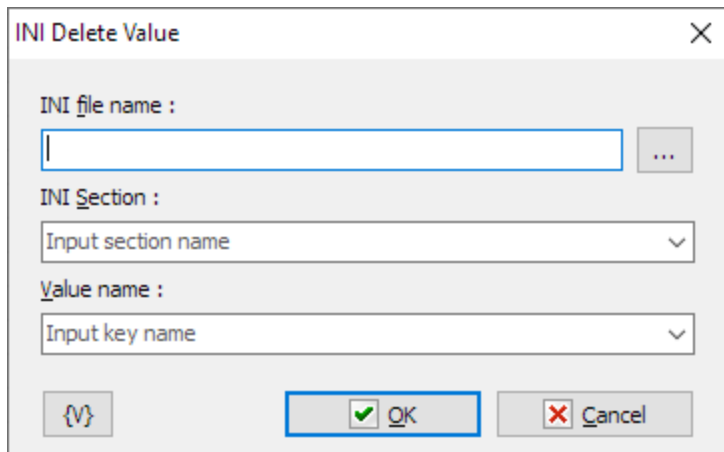
The INI file name, INI section, Value name fields can contain both absolute values and variables that will be replaced by its value when the action is performed.

### The {V} button.

Allows you to select a system or user variable to insert into the current field.

### 4.26.3 INI delete value

Deletes the specified parameter from an INI file.



INI Delete Value

INI file name :  
[Text box] ...

INI Section :  
[Dropdown menu: Input section name]

Value name :  
[Dropdown menu: Input key name]

{V} [OK] [Cancel]

**INI file name**

The full path to the file or file name in the current folder (please see Set Current Folder action).

**INI section**

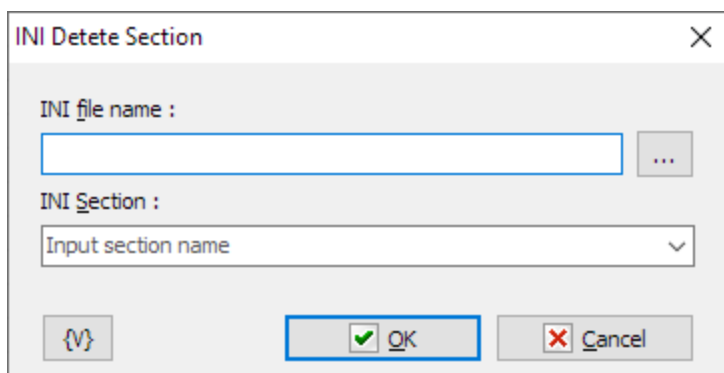
The name of a section. If absolute path is given in file name and file exists, can be selected existing name from dropdown list.

**Value name**

Parameter name. If absolute path is given in file name and file exists, can be selected existing name from dropdown list.

#### 4.26.4 INI delete section

Removes the specified section from the INI file. All parameters from the section will also be deleted.



INI Detete Section

INI file name :  
[Text box] ...

INI Section :  
[Dropdown menu: Input section name]

{V} [OK] [Cancel]

**INI file name**

The full path to the file or file name in the current folder (please see Set Current Folder action).

### INI section

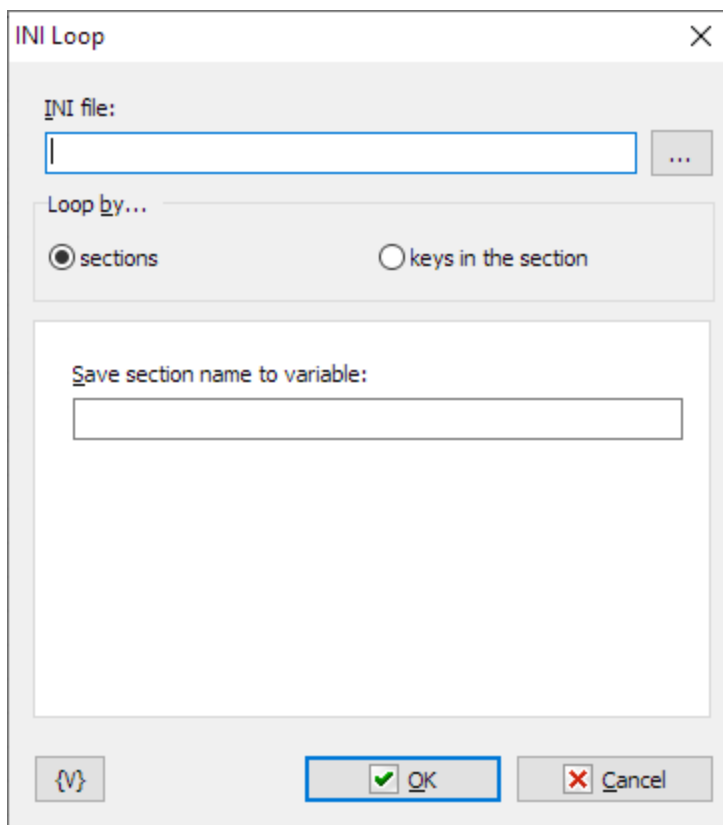
The name of a section. If absolute path is given in file name and file exists, can be selected existing name from dropdown list.

## 4.26.5 INI loop

Loops through all sections or parameters in the specified INI file. This action allows you to read all values whose names are not known beforehand.

The action can operate in one of two modes:

1. **Loop by sections.** In this mode, at each iteration, the action saves the next section name to the specified variable.
2. **Loop by the values in the specified section.** In this mode, at each iteration, the action saves the next name and the parameter value to the specified variables.



### INI file name

The full path to the file or file name in the current folder (please see Set Current Folder action).

### Save section name to variable

Specify a variable name to store the section name

The screenshot shows a dialog box titled "INI Loop". It features a text input field for "INI file:" with a browse button (...). Below this is a "Loop by..." section with two radio buttons: "sections" and "keys in the section" (which is selected). Underneath are three text input fields: "Section name:", "Save key name to variable:", and "Save key value to variable (may be empty):". At the bottom, there are buttons for "{V}", "OK", and "Cancel".

### Section name

Specify the name of the section where you want to list the values

### Save key name to variable

Specify a variable name to store the key (or parameter) name

### Save key value to variable

Specify a variable name to store the key (or parameter) value.

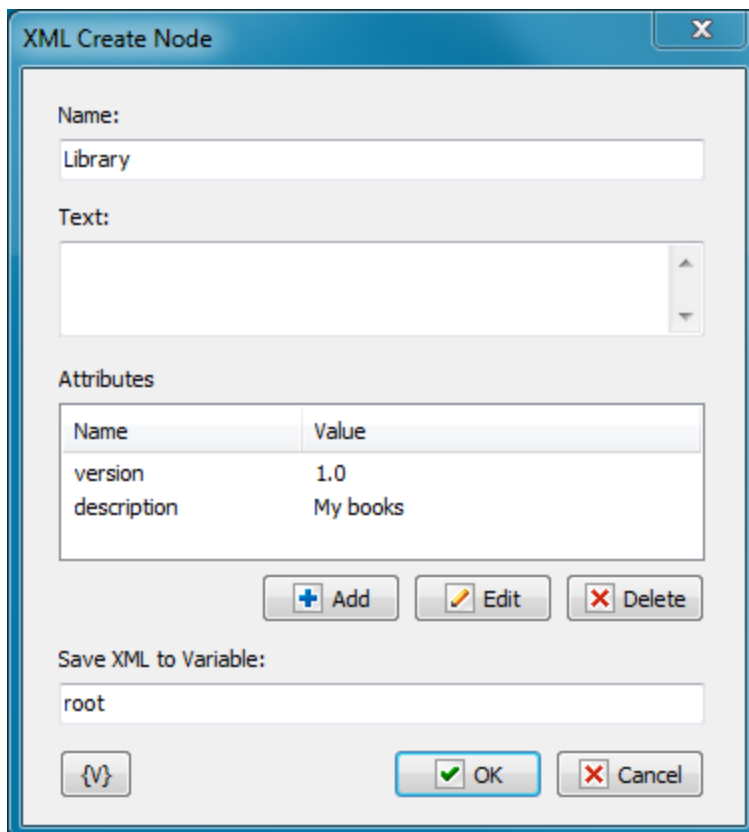
## 4.27 XML Documents

By means of these actions can be created an XML document or analyze a ready one. XML document is a text file written according to the syntax XML. Generally speaking, XML file can be created or opened with any text editor. But sometimes it is laboriously enough to follow all features of the XML syntax in a manual mode.

### 4.27.1 XML Create Node

XML node is a fundamental element of XML document. Every XML node can contain some value or a set of nested XML nodes. Also every node can have arbitrary number of attributes. Roughly speaking, XML document is an XML node which contains inserted nodes with information or with another inserted nodes.

That is, an XML document can contain any structured information.



The screenshot shows a dialog box titled "XML Create Node". It has a close button in the top right corner. The dialog contains the following elements:

- Name:** A text input field containing the text "Library".
- Text:** A text area for entering the node's content.
- Attributes:** A table with two columns: "Name" and "Value".

Name	Value
version	1.0
description	My books
- Buttons:** Three buttons labeled "Add", "Edit", and "Delete" are located below the attributes table.
- Save XML to Variable:** A text input field containing the text "root".
- Final Buttons:** At the bottom, there is a button with curly braces "{ }", and two buttons labeled "OK" and "Cancel".

#### **Name**

The name of a node. This is an identifier of the node. It can contain Latin symbols and numbers without any whitespace or another delimiters.

#### **Text**

This is a value of a node. Every node can contain either some value or inserted nodes.

#### **Attributes**

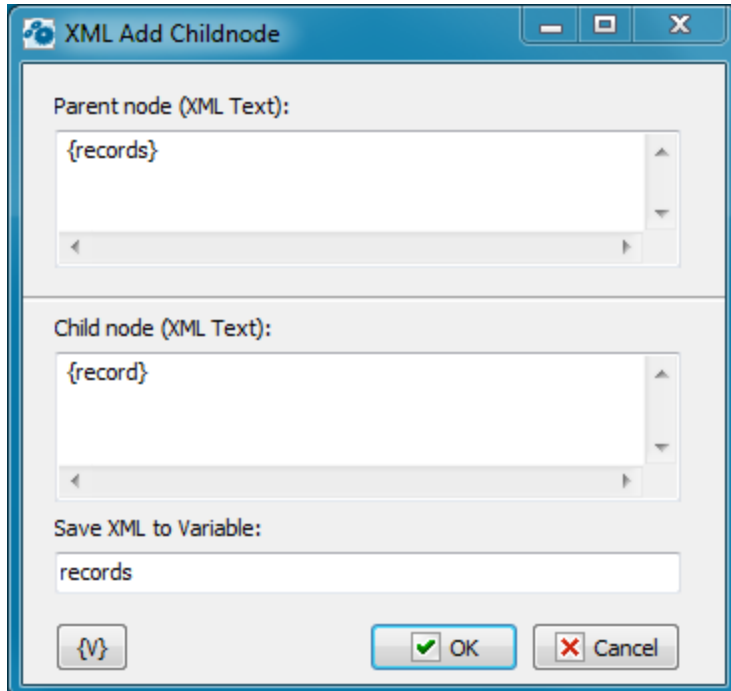
Attributes of a node. Every identifier has the name and the value. The composition of attributes depends on purpose of a documents and defined tasks. The order of attributes does not matter. A node can be without any attributes.

#### **Save XML to Variable**

Enter the name of a variable into which created node will be saved. The node will be saved as a text in the XML format.

### 4.27.2 XML Add Childnode

The action allows adding a child node to another node.



#### Parent Node (XMLText)

Specify a parent node. The node can be specified as an XML text explicitly. For example:

```
<books description="My books">
  <book>
    <author>George Orwell</author>
    <name>1984</name>
  </book>
  <book>
    <author>George Orwell</author>
    <name>Animal Farm</name>
  </book>
</books>
```

Also the node can be specified by means of the variable in which the node is written down. For example:

```
{record}
```

#### Child node (XML text)

Specify the child node which is necessary to add into parent node. The node can be also specified as a XML text or by means of variables.

### Save XML to Variable

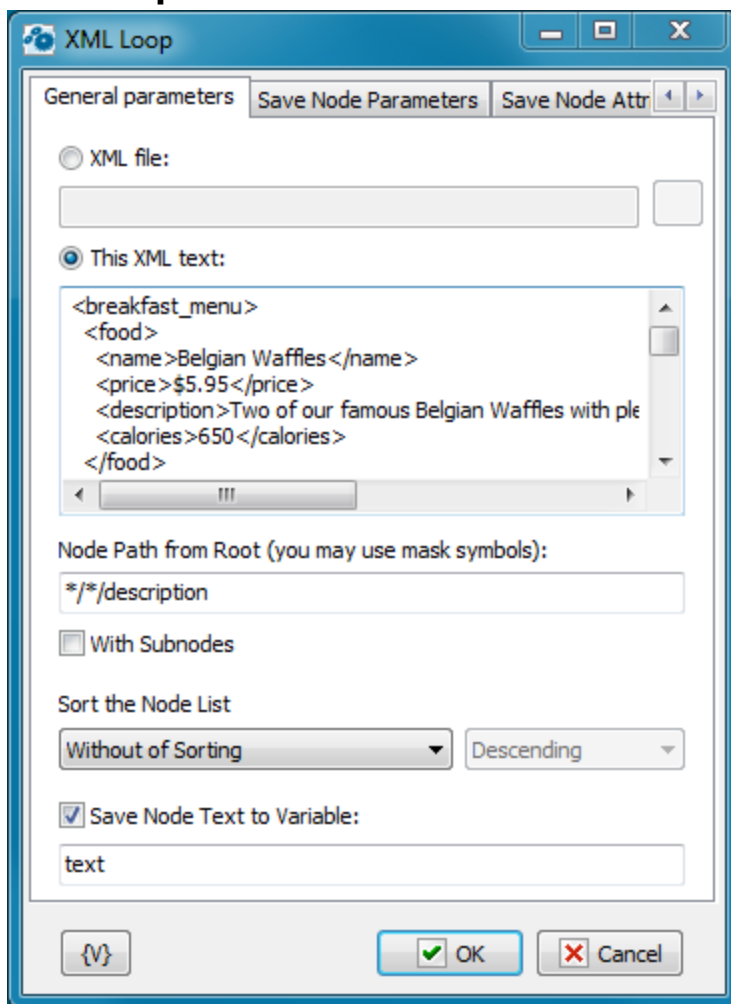
Specify the variable into which the result is necessary to save. The result is saved as a text in XML format.

## 4.27.3 XML Loop

The XML Loop allows to get the information about nodes of the document in a loop. When of every cycle iteration it is possible to save various parameters of the node and the node itself into variables. By means of the XML Loop it is possible to analyze or to process the information enclosed in XML.

If you want to retrieve and process some single node use [XML Get Single Node](#)<sup>4081</sup> action

### General parameters



**XML File**

Read XML from file. This option is recommended if XML contain encoded strings. When the action reads it from file it decodes all encoded strings automatically.

The most popular *UTF-8* encoding. XML file should have corresponding header. For example:

```
<?xml version="1.0" encoding="utf-8"?>
```

**XML**

Specify the XML text which you want to analyze. The text can be specified explicitly or with using of variables.

**Node Path from Root**

The Node Path is a criterion of selecting of loops from an XML document.

Specify a path of nodes from/of a root that you want to enumerate. It is possible to use mask symbols:

- ? - means any single symbol.
- \* - means arbitrary number of symbols.

Instances of usage:

```
*\*\description  
breakfast_menu\*\description  
breakfast_menu\food\description  
breakfast_menu\food\*
```

If you want to enumerate all nodes beginning from a root, specify just the symbol "\*" with switched parameter *With Subnodes*.

**With Subfolders**

If you want to enumerate all nodes including nested nodes, switch on this checkbox.

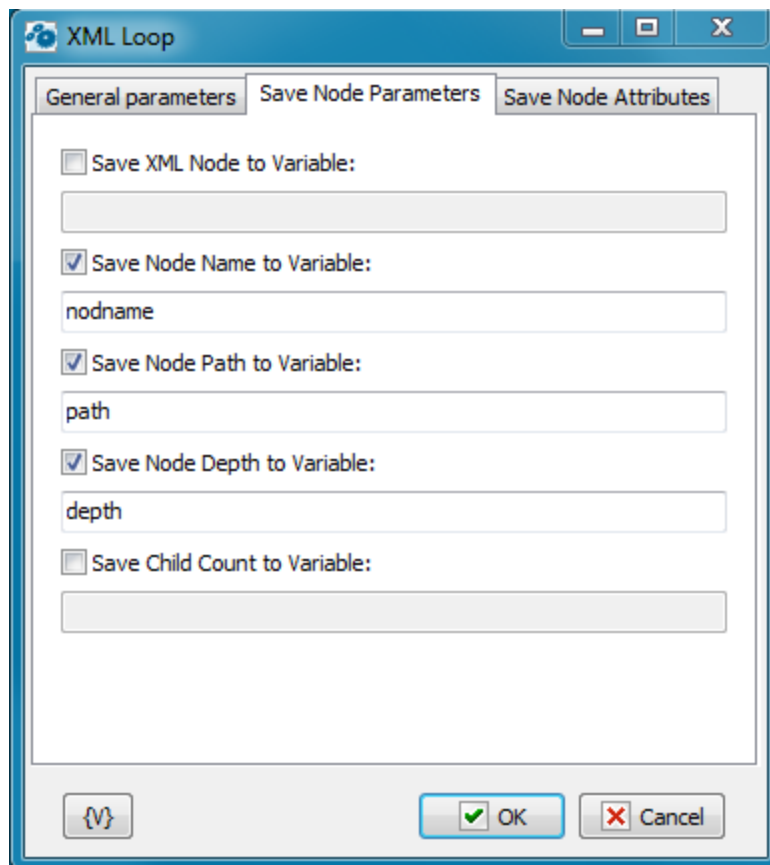
**Save Node Text to variable**

If you need to save a node text, switch on this checkbox and enter the variable name.

**Save node parameters**

The rest of node parameters are collected in this group of parameters.





### Save XML Node to Variable

To save the whole node, switch on this checkbox and enter the variable name.

### Save Node Name to Variable

To save the node name, switch on this checkbox and enter the variable name.

The fact of the matter is that sometimes you cannot know a node name beforehand. For example, this is possible when you select nodes according a mask:

```
breakfast_menu\food\*
```

### Save Node Path to Variable

To save the path of a node, switch on this checkbox and enter the variable name.

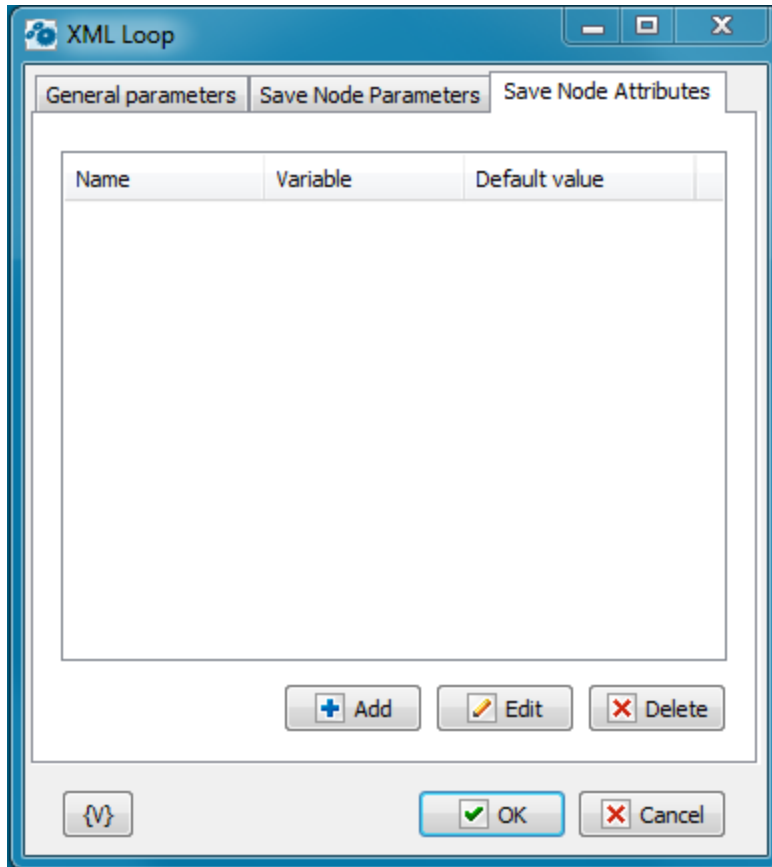
### Save Node Depth to Variable

To save the depth of enclosure of the node, switch on this checkbox and enter the variable name. A node depth is a number, 0 corresponds to a root element.

### Save Child Count to Variable

To save quantity of child nodes, switch on this checkbox and enter the variable name.

## Save Node Attributes



On this tab one can specify the attributes which values are necessary to be saved into variables. It is necessary to specify parameters for every attribute:

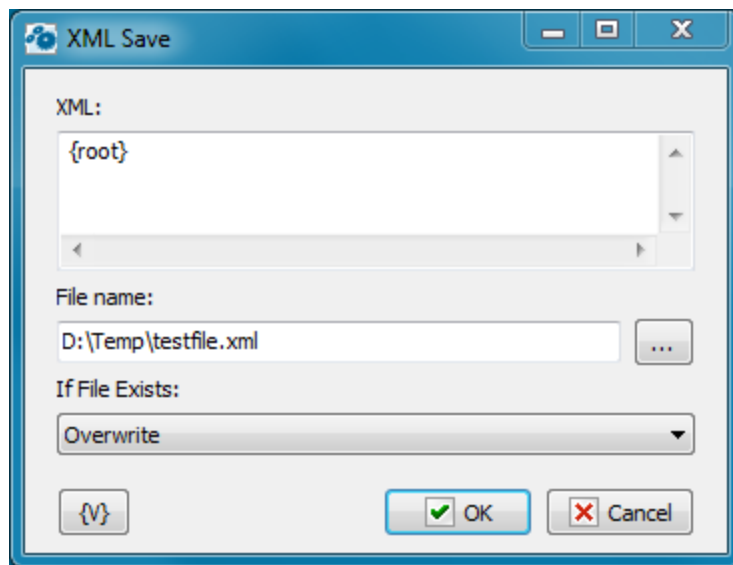
- **Name** - The attribute name
- **Variable** - The name of the variable in which it is necessary to save the attribute value.
- **Default value** - The value which is written into a variable if the attribute is absent.

### 4.27.4 XML Save

The action saves an XML text into a file. In some cases it possible to use the action Write Text File. But it is better to use the XML Save because this action automatically corrects the text format when saves it:

- Adds a header of XML
- Automatically converts non-ASCII symbols into the UTF8 format.

The action always saves the text as a common ASCII text with necessary conversions. Such conversions are not need to save in the memory because the XML document is always saved in the memory in the *Unicode* mode.



## XML

Specify the text of the XML document which you want to save. The text can be specified explicitly or with using of variables.

### File Name

Specify the name of the file into which is necessary to save the XML document.

### If File Exists

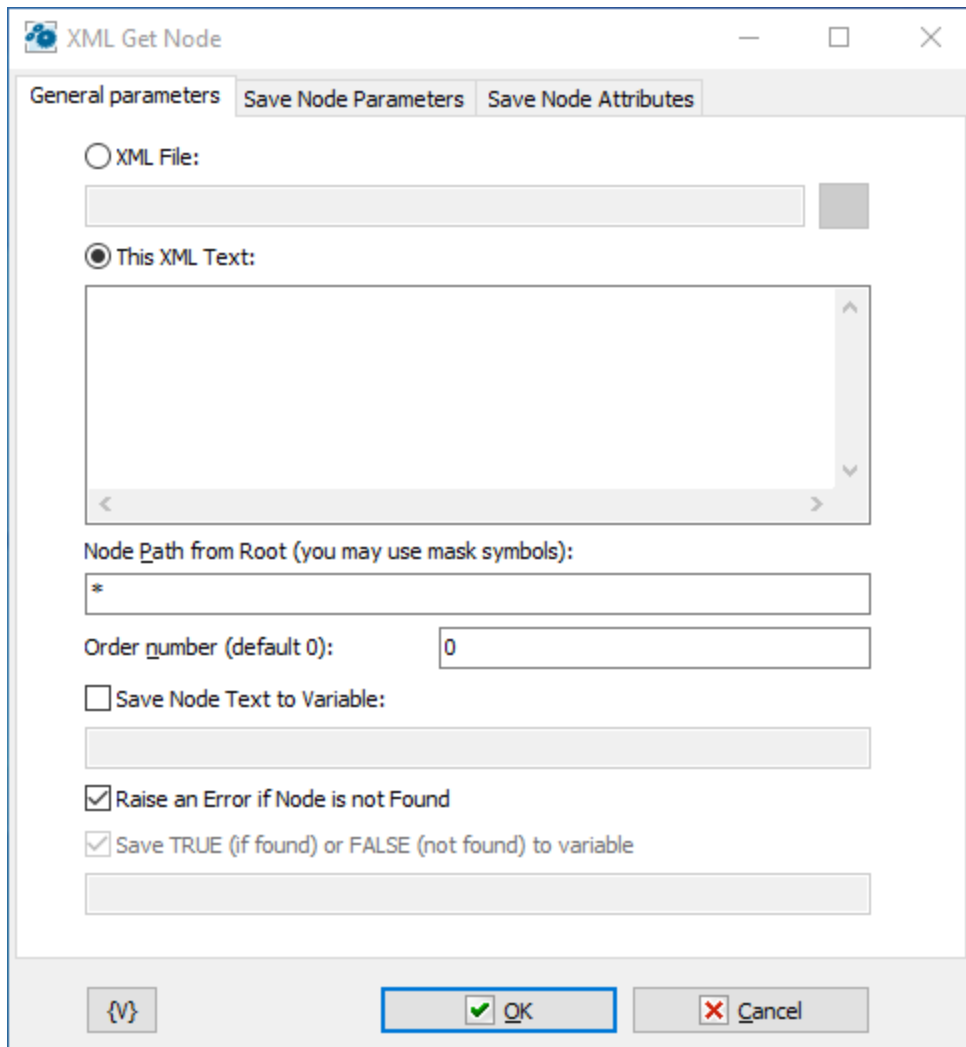
The action if the specified file exists already:

- **Rename** - The file will be saved under a different name
- **Overwrite** - The file will be overwritten.
- **Raise an Error** - The action generates the error File Already Exists.

## 4.27.5 XML Get Single Node

The action permits to extract a single node from the XML without using a cycle. You can specify a mask as a name of the node. If several nodes correspond to selected criteria, the action extracts the first node. The parameters of the action are like the parameters of [XML Loop](#)<sup>404</sup>.

### General parameters



### XML File

Read XML from file. This option is recommended if XML contain encoded strings. When the action reads it from file it decodes all encoded strings automatically.

The most popular UTF-8 encoding. XML file should have corresponding header. For example:

```
<?xml version="1.0" encoding="utf-8"?>
```

### XML

Specify the XML text which you want to analyze. The text can be specified explicitly or with using of variables.

### Node Path from Root

The Node Path is a criterion of selecting of loops from an XML document.

Specify a path of nodes from/of a root that you want to enumerate. It is possible to use mask symbols:

- ? - means any single symbol.

- \* - means arbitrary number of symbols.

Examples of usage:

```
*\*\description
```

```
breakfast_menu\*\description
```

```
breakfast_menu\food\description
```

```
breakfast_menu\food\*
```

### **Order number**

Enter the order number of node. Default is 0 (first node). XML file can contain nodes with the same names. Like records in database table. By default the action returns first matched element but you can specify another ordinal.

### **Save Node Text to variable**

If you need to save a node text, switch on this checkbox and enter the variable name.

### **Raise an Error if Node is not Found**

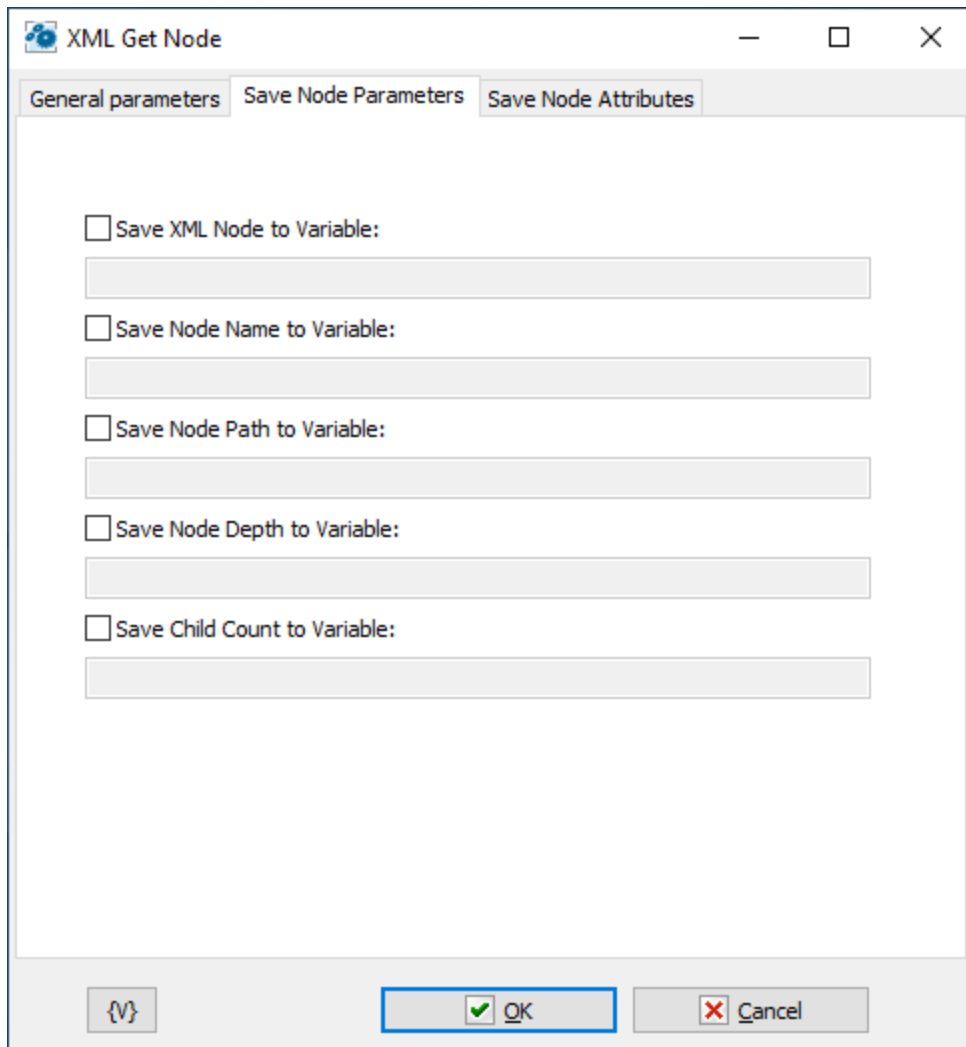
Switch on this checkbox if you want to raise an error. Otherwise the action put the warning message into the log.

### **Save TRUE (if found) or FALSE (not found) to variable**

If you want to save search of necessary note to variable switch on this checkbox and input variable name below.

### **Save node parameters**

The rest of node parameters are collected in this group of parameters.



### Save XML Node to Variable

To save the whole node, switch on this checkbox and enter the variable name.

### Save Node Name to Variable

To save the node name, switch on this checkbox and enter the variable name.

The fact of the matter is that sometimes you cannot know a node name beforehand. For example, this is possible when you select nodes according a mask:

```
breakfast_menu\food\*
```

### Save Node Path to Variable

To save the path of a node, switch on this checkbox and enter the variable name.

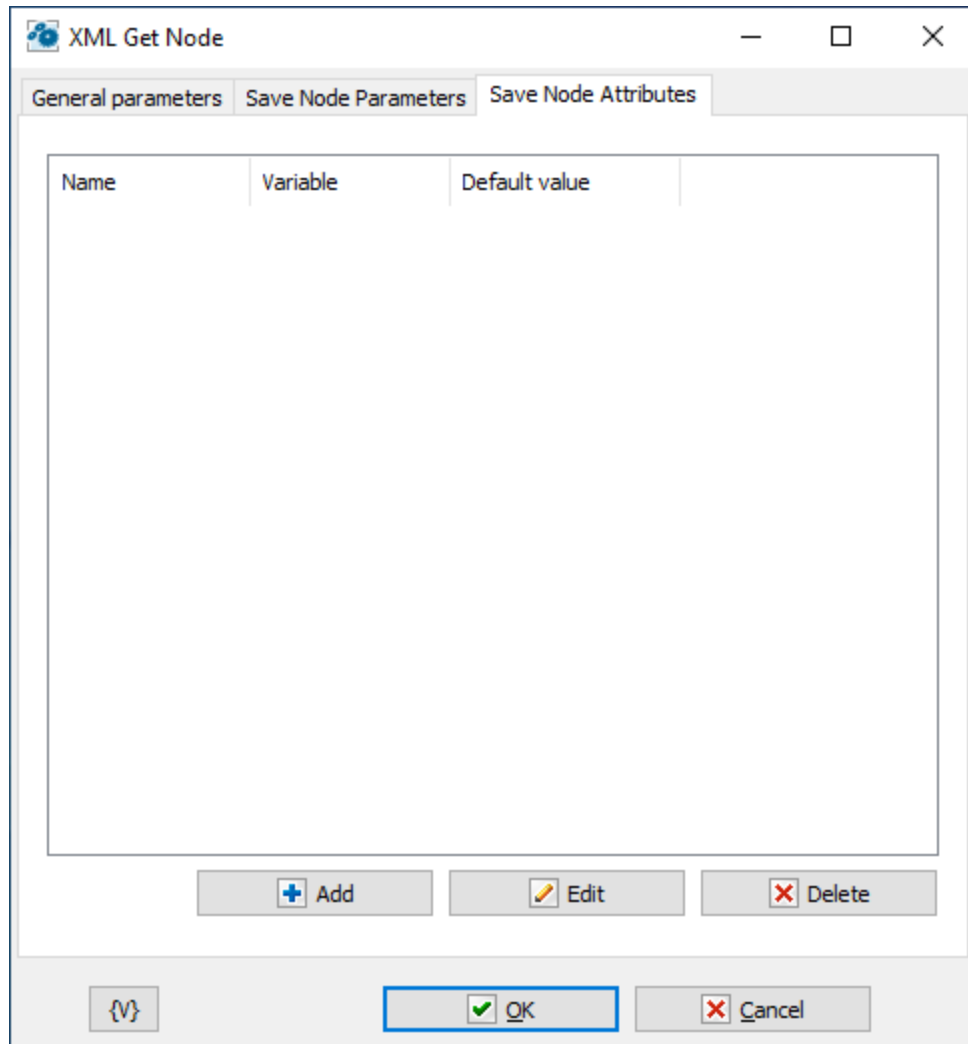
### Save Node Depth to Variable

To save the depth of enclosure of the node, switch on this checkbox and enter the variable name. A node depth is a number, 0 corresponds to a root element.

### Save Child Count to Variable

To save quantity of child nodes, switch on this checkbox and enter the variable name.

### Save Node Attributes



On this tab one can specify the attributes which values are necessary to be saved into variables. It is necessary to specify parameters for every attribute:

- **Name** - The attribute name
- **Variable** - The name of the variable in which it is necessary to save the attribute value.
- **Default value** - The value which is written into a variable if the attribute is absent.

## 4.28 Script tools

This plug-in uses a system script engine and allows to apply program inserts directly in a task. Scripts can be written in JS Script (JS) or VB Script (VBS)

Syntax of JS Script language is described in Microsoft documentation:

[http://msdn.microsoft.com/en-us/library/hbxc2t98\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/hbxc2t98(v=vs.84).aspx)

Syntax of VB Script language is described in Microsoft documentation:

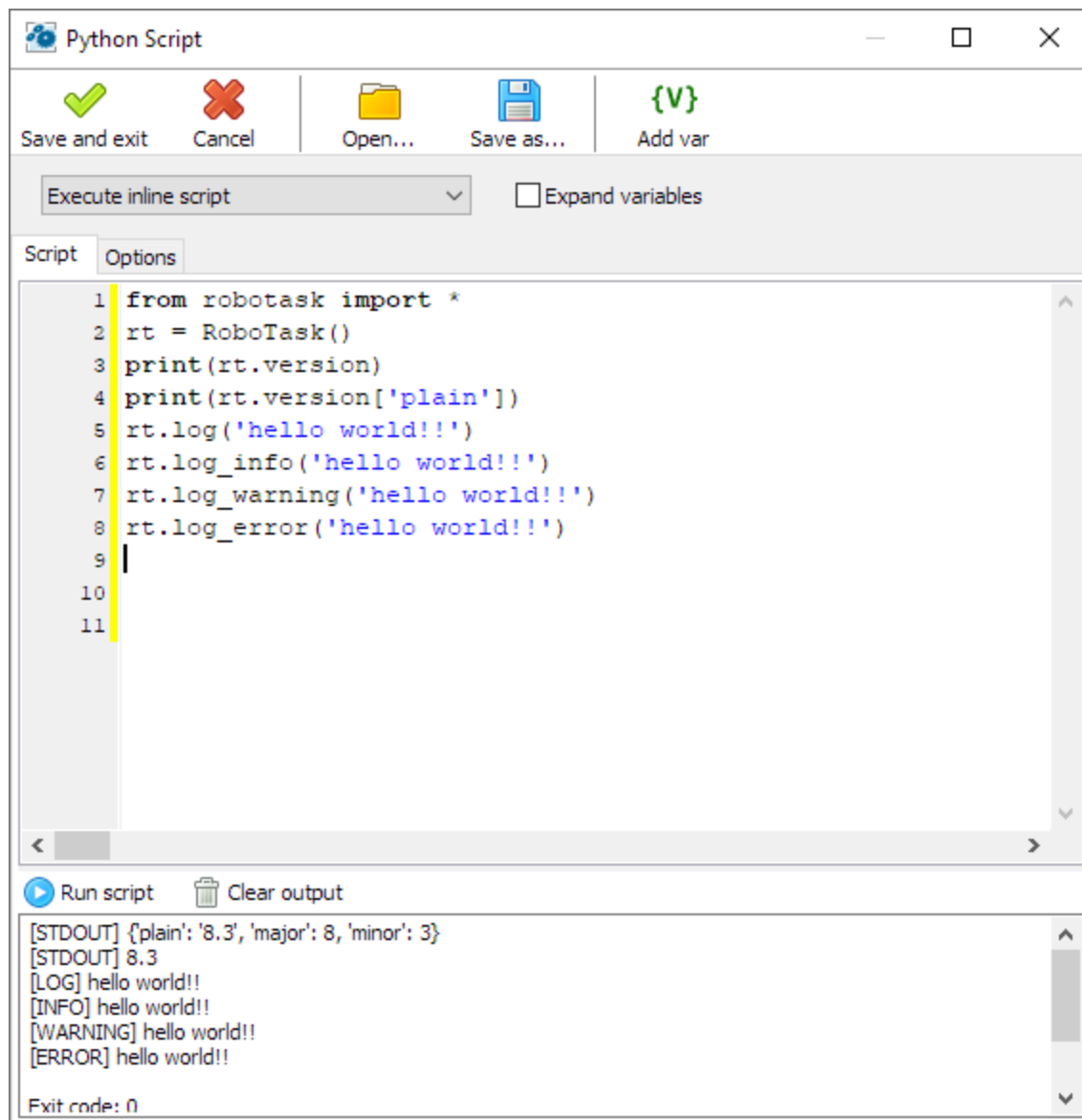
[http://msdn.microsoft.com/en-us/library/t0aew7h6\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/t0aew7h6(v=vs.84).aspx)

### 4.28.1 Python Script

Python is one of the most popular scripting languages today. It can be used for almost any task, including system administration and automation. There are many optional installation packages for a wide variety of tasks. You can find online documentation on Python here: <https://www.python.org/doc/>.

By default, Python Script action uses the preinstalled Python engine that comes with RoboTask. You can also use any other Python package installed and customized to your needs. The script can interact with RoboTask using the [RoboTask object](#)<sup>[416]</sup>.





### Toolbar buttons

*Save and Exit* - saves task settings and closes the editor

*Cancel* - closes the editor without saving

*Open* - opens the script text from the external text file and places it into the script text editor.

*Save as* - saves the script text to an external file.

*Add var* - calls the variable selection window and places the variable call at the current cursor position.

### Script Mode

There are two options

- *Execute inline script* - execute the script you have written in the script editor. In this case, you will see the text editor.
- *Execute external script* - run the script from an external file.

### Expand variables

Check this box if you want to expand RoboTask variables in the script body before running the script. With variables, you can create dynamic text depending on different conditions.

### Script

In "Execute inline script" mode, it is a script text editor.

In "Execute External script" mode, it is a field to enter the file name and a file selection button.

Python Script

Save and exit Cancel Open... Save as... Add var

Execute inline script  Expand variables

Script Options

Assign exit code to variable:

Assign output of script to variable:

Python interpreter

Use embedded Python

Use specified Python

Path to Python interpreter (leave empty for system-wide installation):  ...

Python version: 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)]

Module search paths (one per line):  +

## Options

### Assign exit code to a variable

Enter the variable name to save the script exit code. By default, the exit code is 0, but it can be changed in the script.

### Assign script output to a variable

Enter a variable name in this field to write the script's output text stream to a variable. You will then be able to process this text in your task.

### Python interpreter

- *Use Embedded Python* - use the default Python engine that comes with RoboTask.
- *Use Specified Python* - use your Python package. You can have several Python packages configured for different tasks.

### Module search paths

Here you can set the search paths to additional Python modules (which are included in the script with the **import** command).

#### 4.28.1.1 RoboTask object for Python

RoboTask object is similar in functionality to the built-in COM object [RoboTaskApp](#)<sup>[562]</sup>, but there are major differences. Please note that [Python](#)<sup>[413]</sup> is case-sensitive.

Here is an example of how to use the object:

```
from robotask import *
rt = RoboTask()
print(rt.version)
print(rt.version['plain'])
```

The basic Python package doesn't contain a Windows COM object support module. However, you can use your Python with pywin32 support ([see here](#)) and use the [RoboTaskApp](#)<sup>[562]</sup> object directly. Code example:

```
import win32com.client as win32
rt = win32.Dispatch('RoboTask.App')
print(rt.Version)
print(rt.TaskCount)
```

### RoboTask()

Constructor.

Creates RoboTask object.

Use:

```
rt = RoboTask()
```

**version**

Property.

Returns dictionary with fields:

*plain* - RoboTask text version. For example, "8.3"

*major* - major version (8)

*minor* - minor version (3)

**task\_count**

Property.

Returns the number of registered tasks in RoboTask.

**expand\_text(text)**

Method.

Expands a string containing RoboTask variables and returns it.

For example, expression

```
rt.expand_text('Date and time now: {DateTimeToFormat({DateTime},yyyyy-mm-dd  
hh:nn:ss)}')
```

Will return a value like this:

```
Date and time now: 2021-04-01 17:57:35
```

**log(text)**

Method.

Outputs the specified text to the task log in user message mode.

**log\_info(text)**

Method.

Outputs the specified text in the log in system message mode.

**log\_warning(text)**

Method.

Outputs the specified text in the log in warning mode.

**log\_error(text)**

Method.

Outputs the specified text to the task log in error message mode

**add\_user\_var(name, value)**

Method.

Creates RoboTask variable with *NAME* name and *VALUE* value.

**set\_user\_var(name, value)**

Method.

Sets the *VALUE* to RoboTask variable with name *NAME*.

### **del\_user\_var(name)**

Method.

Deletes RoboTask variable named *NAME*.

### **task\_info(num)**

Method.

Returns task information by number or name (external name).

Method returns dictionary with fields:

- *name* - task name (title).
- *ext\_name* - external name. You can use this name to refer to the task in the script
- *enabled* - True or False
- *state* - task state. Can have 5 values:
  - manual* - the task has no triggering events, or its triggering events are disabled (see the enabled property). It waits to be run manually or from another task;
  - idle* - the task has some enabled triggering events. It waits to be run automatically,
  - disabled* - the task is disabled while editing,
  - running* - the task is started and in progress,
  - stopped* - the task received the stop command, but it is yet running. After the task is completely stopped, its status becomes either "manual" or "idle".

### **current\_task()**

Method.

Returns information about the current task.

The result is exactly the same as *task\_info* method except that it returns information about the current task.

### **start\_task(ext\_name, params = "")**

Method.

Start task by number or *External Name*.

Parameters

- *ext\_name* - task number or external name
- *params* - parameters of the task to be launched. Empty string by default. Parameters can be set as text, one parameter per line :

```
parameter_name=parameter_value
```

### **stop\_task(ext\_name)**

Method.

Stops task with a specified number or *External Name*

Note that RoboTask only sends a signal to stop the task. The task will stop when the step that is currently in progress is completed.

**enable\_task(ext\_name)**

Method.

Changes task state by number or *External Name* to "enabled" and starts all "enabled" task triggers.

**disable\_task(ext\_name)**

Method

Changes task state by number or *External Name* to "disabled" and stops all active task triggers.

**service\_mode**

Property

Returns logical value.

True - RoboTask is running in service mode.

False - RoboTask is running in normal app mode.

**window\_handle**

Property

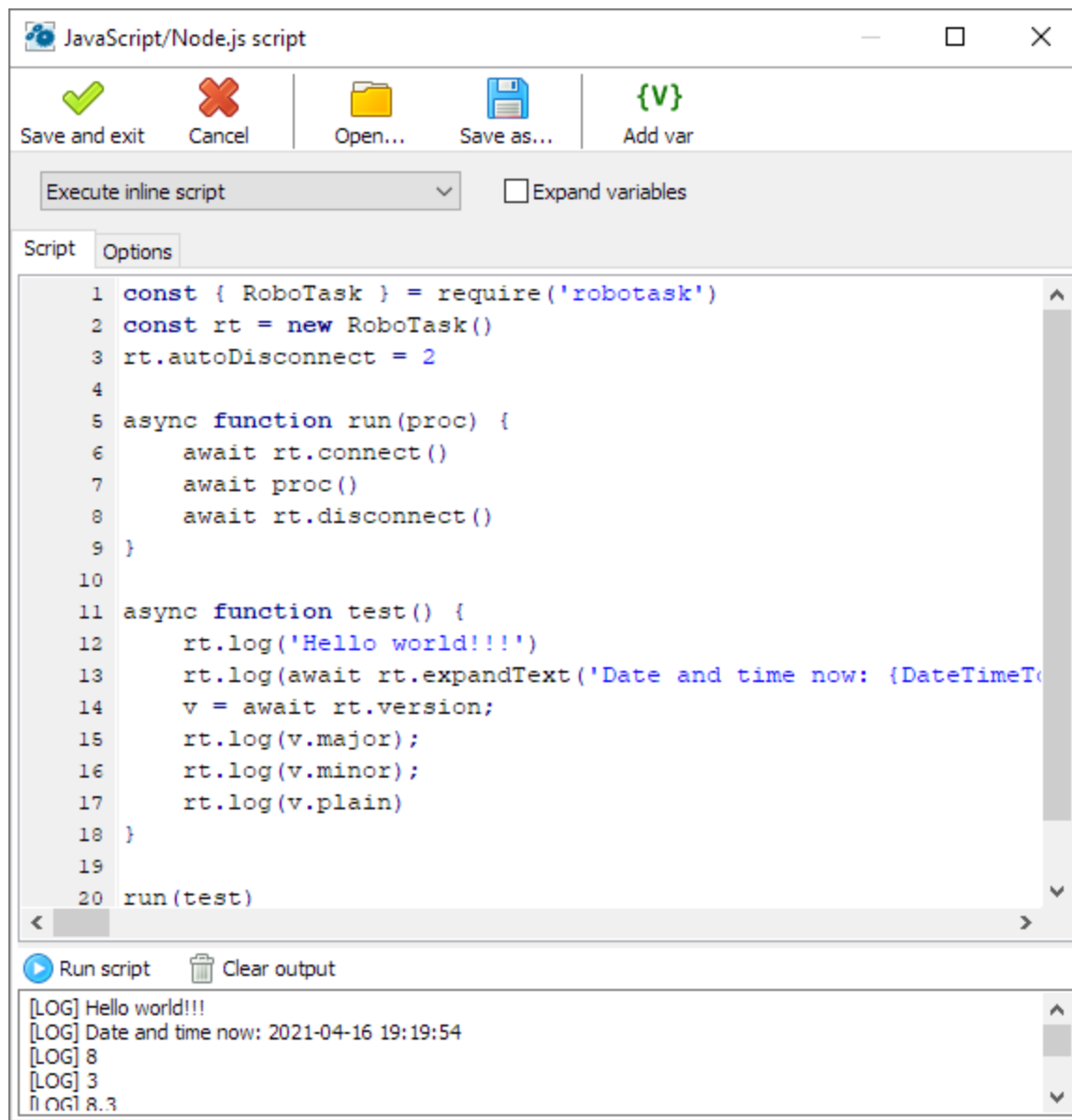
Returns the RoboTask's window handle that receives system messages.

## 4.28.2 JavaScript/Node.js script

This action supports the full JavaScript language, unlike [JS Script](#)<sup>[430]</sup> and [JS Evaluate](#)<sup>[432]</sup> actions. [JS Script](#)<sup>[430]</sup> and [JS Evaluate](#)<sup>[432]</sup> actions use the system script engine and support a fairly simplified language.

By default, the JavaScript/Node.js script action uses the preinstalled node.js engine that comes with RoboTask. You can also use any other node.js kit installed and customized to your needs. The script can interact with RoboTask using the [RoboTask object for node.js](#)<sup>[422]</sup>

You can find the language description [here](#) and the node.js objects description [here](#)<sup>[520]</sup>.



### Toolbar buttons

*Save and Exit* - saves task settings and closes the editor

*Cancel* - closes the editor without saving

*Open* - opens the script text from the external text file and places it into the script text editor.

*Save as* - saves the script text to an external file.

*Add var* - calls the variable selection window and places the variable call at the current cursor position.

### Script Mode

There are two options

- *Execute inline script* - execute the script you have written in the script editor. In this case, you will see the text editor.
- *Execute external script* - run the script from an external file.

### Expand variables

Check this box if you want to expand RoboTask variables in the script body before running the script. With variables, you can create dynamic text depending on different conditions.

### Script

In "Execute inline script" mode, it is a script text editor.

In "Execute External script" mode, it is a field to enter the file name and a file selection button.

The screenshot shows a window titled "JavaScript/Node.js script" with a toolbar containing icons for "Save and exit" (green checkmark), "Cancel" (red X), "Open..." (yellow folder), "Save as..." (blue floppy disk), and "Add var" (green curly braces). Below the toolbar is a dropdown menu set to "Execute inline script" and a checkbox for "Expand variables". The main area has two tabs: "Script" and "Options". Under "Options", there are two input fields: "Assign exit code to variable:" and "Assign output of script to variable:". Below these is a section for "Node.js interpreter" with radio buttons for "Use embedded Node.js" (selected) and "Use specified Node.js". A text field for "Path to Node.js interpreter (leave empty for system-wide installation):" is followed by a file selection button. Below that, it shows "Node.js version: v14.16.0". At the bottom, there is a section for "Module search paths (one per line):" with a text area and a "+" button to add paths.



## Options

### Assign exit code to a variable

Enter the variable name to save the script exit code. By default, the exit code is 0, but it can be changed in the script.

### Assign script output to a variable

Enter a variable name in this field to write the script's output text stream to a variable. You will then be able to process this text in your task.

### Node.js interpreter

- *Use Embedded Node.js* - use the default Node.js engine that comes with RoboTask.
- *Use Specified Node.js* - use your Node.js package. You can have several Node.js packages configured for different tasks.

### Module search paths

Here you can set the search paths to additional Node.js modules (which are included in the script with the **require** command).

#### 4.28.2.1 RoboTask object for node.js

The RoboTask object for node.js is similar in functionality to the built-in COM object [RoboTaskApp](#)<sup>[562]</sup>, but there are major differences. Note that the JavaScript/node.js language is case-sensitive.

Example of the object usage:

```
const { RoboTask } = require('robotask')
const rt = new RoboTask()
rt.log('Hello world!').then(()=>rt.disconnect())
```

The basic JavaScript/Node.js package also doesn't support Windows COM objects. You can use your node.js package with [win32ole](#) installed. In this case, you will be able to use the built-in [RoboTask.App](#)<sup>[562]</sup> object directly.

### RoboTask()

Constructor.

Creates RoboTask object.

Use:

```
rt = RoboTask()
```

### autoDisconnect

Read-write property

Timeout in seconds. Timeout of automatic disconnection from RoboTask.

0 means permanent connection. In this case, the script won't finish until *disconnect()* is executed. The default is 3 seconds.

When calling methods or properties of the RoboTask object, the connection is automatic.

**connect()**

Method.

Connecting to the RoboTask

**disconnect()**

Method.

Disconnecting from RoboTask

**connected**

Read-only property

Returns RoboTask connection status: True or False

**log(text)**

Method.

Outputs the specified text to the task log in user message mode.

**logInfo(text)**

Method.

Outputs the specified text in the log in system message mode.

**logWarning(text)**

Method.

Outputs the specified text in the log in warning mode.

**logError(text)**

Method.

Outputs the specified text to the task log in error message mode

**expandText(text)**

Method.

Expands a string containing RoboTask variables and returns it.

For example, expression

```
rt.expand_text('Date and time now: {DateTimeToFormat({DateTime},yyyyy-mm-dd  
hh:nn:ss)}')
```

Will return a value like this:

```
Date and time now: 2021-04-08 16:33:21
```

**addUserVar(name, value)**

Method.

Creates RoboTask variable with *NAME* name and *VALUE* value.

### **setUserVar(name, value)**

Method.

Sets the *VALUE* to RoboTask variable with name *NAME*.

### **delUserVar(name)**

Method.

Deletes RoboTask variable named *NAME*.

### **version**

Property.

Returns dictionary with fields:

*plain* - RoboTask text version. For example, "8.3"

*major* - major version (8)

*minor* - minor version (3)

### **taskCount**

Property.

Returns the number of registered tasks in RoboTask.

### **taskInfo(num)**

Method.

Returns task information by number or name (external name).

Method returns dictionary with fields:

- *name* - task name (title).
- *ext\_name* - external name. You can use this name to refer to the task in the script
- *enabled* - True or False
- *state* - task state. Can have 5 values:
  - manual* - the task has no triggering events, or its triggering events are disabled (see the enabled property). It waits to be run manually or from another task;
  - idle* - the task has some enabled triggering events. It waits to be run automatically,
  - disabled* - the task is disabled while editing,
  - running* - the task is started and in progress,
  - stopped* - the task received the stop command, but it is yet running. After the task is completely stopped, its status becomes either "manual" or "idle".

### **currentTask**

Property.

Returns information about the current task.

The result is exactly the same as *taskInfo* method except that it returns information about the current task.

### **startTask(ext\_name, params)**

Method.

Start task by number or *External Name*.

Parameters

- *ext\_name* - task number or external name
- *params* - parameters of the task to be launched. Empty string by default. Parameters can be set as text, one parameter per line :

```
parameter_name=parameter_value
```

### **runTask**

Alias of *startTask* method

### **stopTask(ext\_name)**

Method.

Stops task with a specified number or *External Name*

Note that RoboTask only sends a signal to stop the task. The task will stop when the step that is currently in progress is completed.

### **enableTask(ext\_name)**

Method.

Changes task state by number or *External Name* to "enabled" and starts all "enabled" task triggers.

### **disableTask(ext\_name)**

Method

Changes task state by number or *External Name* to "disabled" and stops all active task triggers.

### **serviceMode**

Property

Returns logical value.

True - RoboTask is running in service mode.

False - RoboTask is running in normal app mode.

### **windowHandle**

Property

Returns the RoboTask's window handle that receives system messages.

### 4.28.3 PowerShell script

PowerShell is a powerful scripting and command-line tool for managing your system. You can find more information about the PowerShell script [language and the commands description here](#).

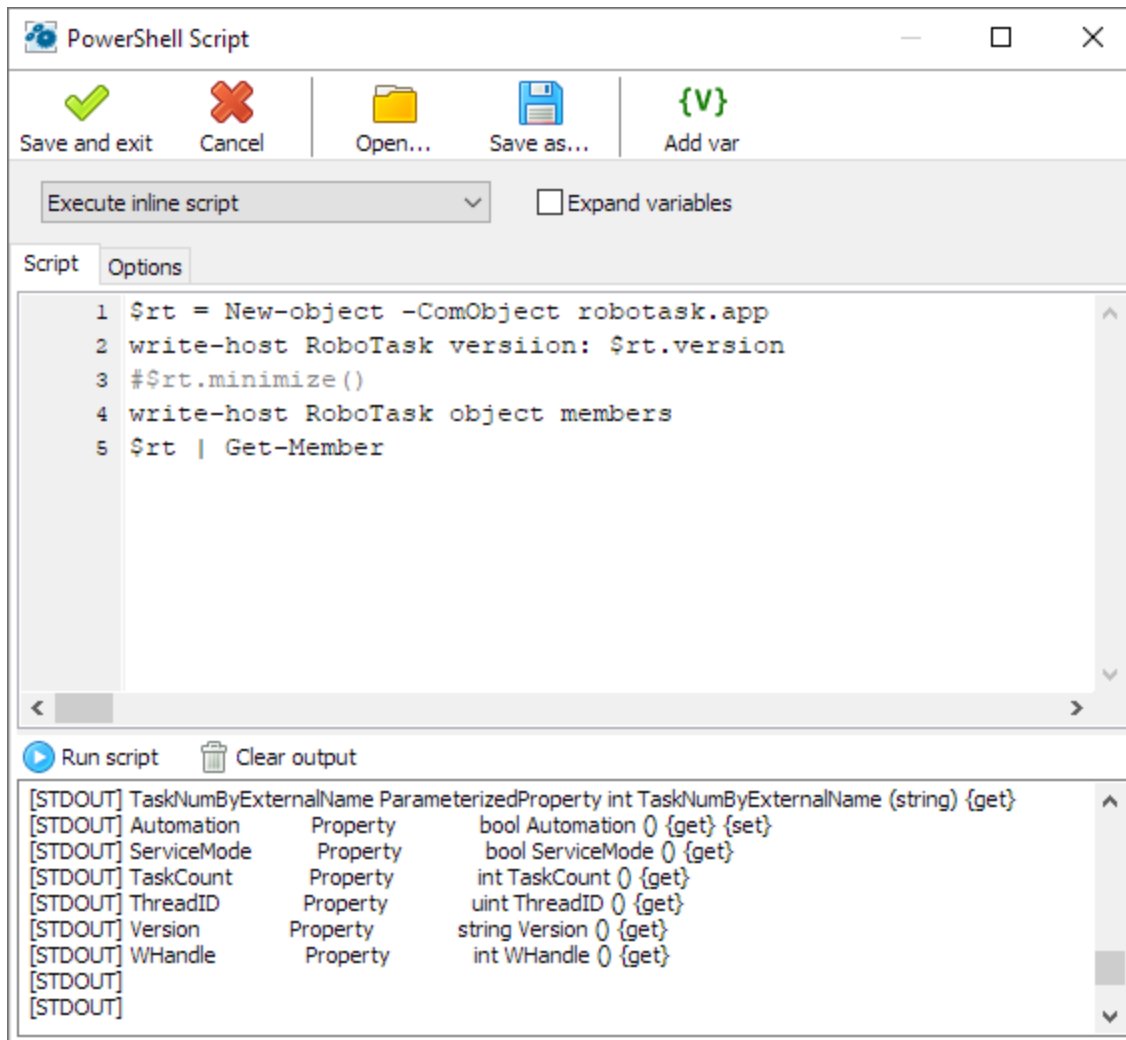
You can use the built-in [RoboTaskApp COM object](#) to communicate with RoboTask. For example:

```
$rt = New-object -ComObject RoboTask.App
write-host RoboTask version: $rt.version
write-host RoboTask object members
$rt | Get-Member
```

For a PowerShell script, the current ThreadId of the task is passed through the `ROBOTASK_SCRIPT_ID` environment variable.

Example of the script:

```
$rt = New-object -ComObject RoboTask.App
#write-host $env:ROBOTASK_SCRIPT_ID
$rt.SetThreadId($env:ROBOTASK_SCRIPT_ID)
$rt.WriteToLog(0, "Hello", 0)
```



### Toolbar buttons

*Save and Exit* - saves task settings and closes the editor

*Cancel* - closes the editor without saving

*Open* - opens the script text from the external text file and places it into the script text editor.

*Save as* - saves the script text to an external file.

*Add var* - calls the variable selection window and places the variable call at the current cursor position.

### Script Mode

There are two options

- *Execute inline script* - execute the script you have written in the script editor. In this case, you will see the text editor.
- *Execute external script* - run the script from an external file.

## Expand variables

Check this box if you want to expand RoboTask variables in the script body before running the script. With variables, you can create dynamic text depending on different conditions.

## Script

In "Execute inline script" mode, it is a script text editor.

In "Execute External script" mode, it is a field to enter the file name and a file selection button.

The screenshot shows a dialog box titled "PowerShell Script". At the top, there are five icons: a green checkmark for "Save and exit", a red X for "Cancel", a yellow folder for "Open...", a blue floppy disk for "Save as...", and a green curly brace for "Add var". Below these icons is a dropdown menu set to "Execute inline script" and a checkbox labeled "Expand variables". The dialog has two tabs: "Script" and "Options". The "Options" tab is active and contains several fields: "Assign exit code to variable:" with an empty text box, "Assign output of script to variable:" with an empty text box, a "PowerShell" section with "Path to Powershell (leave empty for system-wide installation):" and an empty text box with a browse button "...", and "PowerShell version: 5.1". At the bottom, there is a "Module search paths (one per line):" label and an empty text area with a "+" button and a scroll bar.

## Options

### Assign exit code to a variable

Enter the variable name to save the script exit code. By default, the exit code is 0, but it can be changed in the script.

### Assign script output to a variable

Enter a variable name in this field to write the script's output text stream to a variable. You will then be able to process this text in your task.

### PowerShell

Path to PowerShell - empty by default. This means you can use system default PowerShell utility. But you can use alternate PowerShell and specify path to it. (for example PWSH.EXE)

### Module search paths

Here you can set the search paths to additional cmdlets, script files, or operable program.

## 4.28.4 Script extensions

The *RoboTask* gives some extensions of the language (for [VB Script](#)<sup>[434]</sup> and [JS Script](#)<sup>[430]</sup>): several additional functions. You can use these functions freely in your texts of scripts. Notice that the JavaScript is case sensitive. But the functions of extensions work insensitive to a using register. Thereby, the expressions

```
LogMessage("Hello World!!!");  
logmessage("Hello World!!!");
```

are identical.

### function LogMessage(AMessage)

The function outputs a message to a log. In edit mode of a script, the function outputs a message to the log of the editor. When a task works, a message is output to a log of the task in the mode *User Message*.

*AMessage* - the string to output

### function include(ScriptFileName)

This function returns the text of specified script file.

*ScriptFileName* - The name of file which is necessary to load.

You can execute this text in your script. Just put these into your script:

for [VB Script](#)<sup>[434]</sup> action use

```
execute include("d:\myscripts\script.txt")
```

For [JS Script](#)<sup>[430]</sup> action use

```
eval(include("d:\myscripts\script.txt"));
```

### function Pause(AMilliseconds)

Pauses and stops execution of the script for the specified time.

*AMilliseconds* - pause time in milliseconds.

### function MessageBox(AMessage)

Displays a dialog box with the specified message



*Amessage* - the message as a string

### function RoboTaskApp()

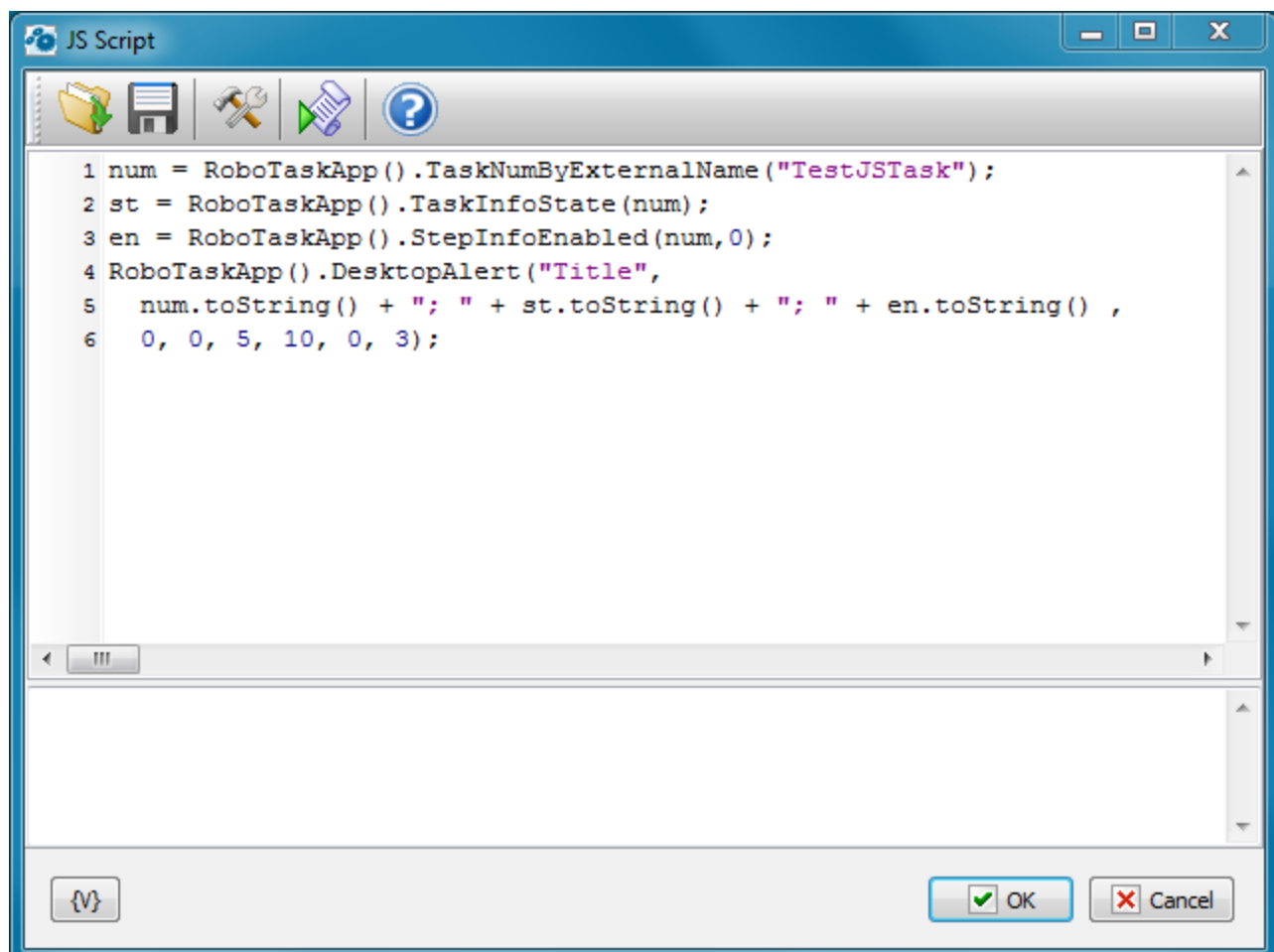
This function returns the COM object RoboTask. This object allows to use some functions of the *RoboTask* directly from a script. The description of the object is in the chapter ["RoboTaskApp Object"](#)<sup>562</sup>.

## 4.28.5 JS script

This action allows to perform a program, which is written in JavaScript. Notice that the languages Java and JavaScript are not the same, although they are very alike. The documentation for the language can be read on the Microsoft site:

[http://msdn.microsoft.com/en-us/library/hbxc2t98\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/hbxc2t98(v=vs.84).aspx)

The action editor is a specialize text editor adapted for editing a program code.



**Popup proposals.**

When pressing the Ctrl-Space, a popup menu with some functions and objects of JavaScript shows up on the screen. One can choose a necessary point from the list and then the editor will insert a proper line/string into a text.

If you began to enter a function name and pressed the Ctrl-Space, only alternatives corresponding to the input line will be left in the popup menu.

### Autocomplete.

This function allows to insert standard language constructions into a text automatically. For this you need to enter 2-5 key constructions (identifier of construction) and press the combination Ctrl-J. The editor inserts the whole construction instead of the key word.

The editor supports the next constructions:

Identifier	Description	Insert Statement
if	Simple "if"	<pre>if (   ) { }</pre>
ife	"if .. else" statement	<pre>if (   ) { } else { }</pre>
for	Simple "for" loop	<pre>for (i=0; i&lt;  ; i++) { }</pre>
func	Function declaration	<pre>function   ( ) { return 0; }</pre>
try	"try .. catch .. finally" block	<pre>try { } catch(e) { } finally { }</pre>
do	"do .. while" loop	<pre>do { } while (   )</pre>
while	"while" loop	<pre>while ( ) { }</pre>
with	"with" keyword	<pre>with ( ) { }</pre>
swi	"switch" statement	<pre>switch ( ) { case 0: break; case 1:</pre>

```
break;  
default:  
}
```

## Interface

### Load from file

Load the text of a script from a file.

### Save script as file

Save the text of a script into a file.

### Additional options

*Expand Variables in the script* - If to switch on this mode, the RoboTask expands all inserts of variables in a text before running of the script.

*Save Final script into variable* - If it is necessary to monitor the final text of the script, switch on this checkbox and enter the variable name. This mode is need only for control and checkout. The fact of the matter is that the substitution of variable values can break down the syntax of a script text. By means of this mode it is possible to get the final script and analyze it if it is necessary.

The field of a log is under the editor. The messages about runtime errors of the script are output into this field. Also the text of messages LogMessage is output here.

### Run script

Running of a script from the editor to test. If a runtime error arises, you can see a message about the error in the field of the edit log.

### Help

Activation of a help system.

## 4.28.6 JS Evaluate

This action allows to calculate an expression value and then save it into a variable. The expression has to be written according to the syntax of JS language. The documentation for the language can be read on the Microsoft site: [http://msdn.microsoft.com/en-us/library/hbxc2t98\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/hbxc2t98(v=vs.84).aspx)

The action also allows to connect the function library from an external file or directly in an additional script.

For example, try to calculate the expression

```
sum(5,6) + multi(3,4)
```

And we get the error. The JavaScript language doesn't contain functions sum and multi.

To solve this, it is possible to specify a little library of user functions:

```
function sum(a, b)
```

```
{
  return a+b;
}

function multi(a, b)
{
  return a*b;
}
```

In this case the action calculates the result without any troubles. The given example is too simple and is cited for the demonstration solely.

Another example. Calculate the *factorial* of a number. Such function is absent in JavaScript. But it is easy to evade. It takes only to create the simply function:

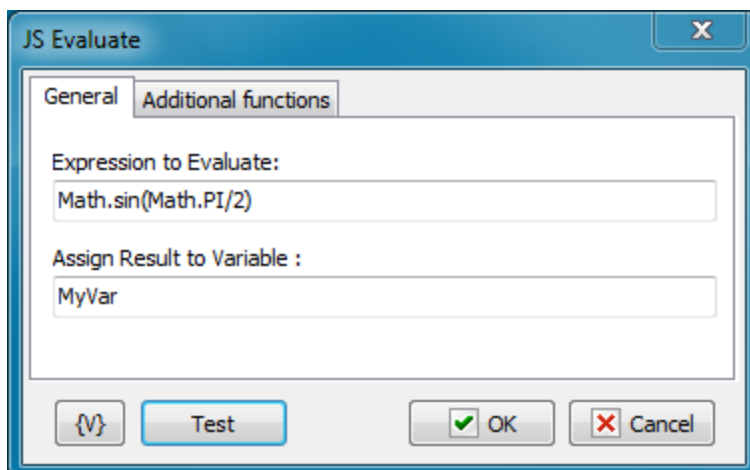
```
function factorial(a)
{
  if (a>30)
  { //this is synthetic limitation
    throw new Error(1, "Parameter more than 30");
  }
  res = 1;
  for (i=1; i<=a; i++)
  {
    res *= i;
  }
  return res;
}
```

Now we easy can calculate the expression:

```
factorial(15)
```

Given possibility allows to do some specific calculations, that are difficult to be written in one line.

## General Tab



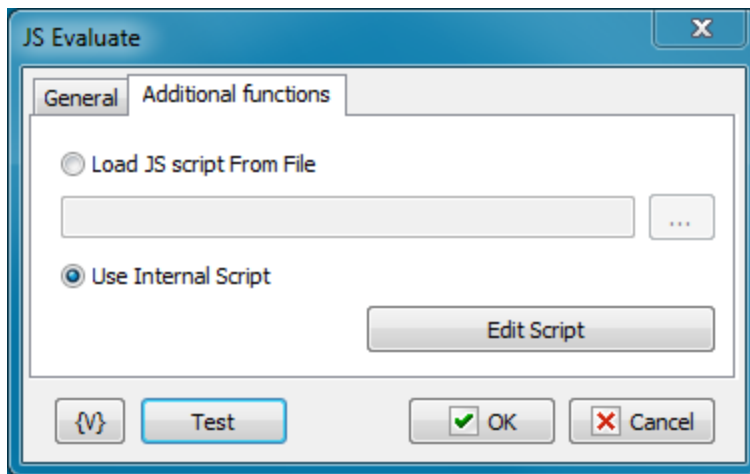
### Expression To Evaluate

Write down the expression whose value needs to be calculated. The expression has to be made according to the syntax of JavaScript.

### Assign Result to Variable

Enter the name of the variable into which it is necessary to save the result.

## Additional Functions



### Load JS Script from File

Load the library of additional functions from an external file. Enter a file name or choose it interactively.

The file name can be empty. In this case the action works only with standard functions and objects of JS.

### Use Internal Script

If you want to specify functions directly in a task, choose this option. To edit a script, click the button Edit Script. You can see the editor like in the action [JS Script](#)<sup>[430]</sup>.

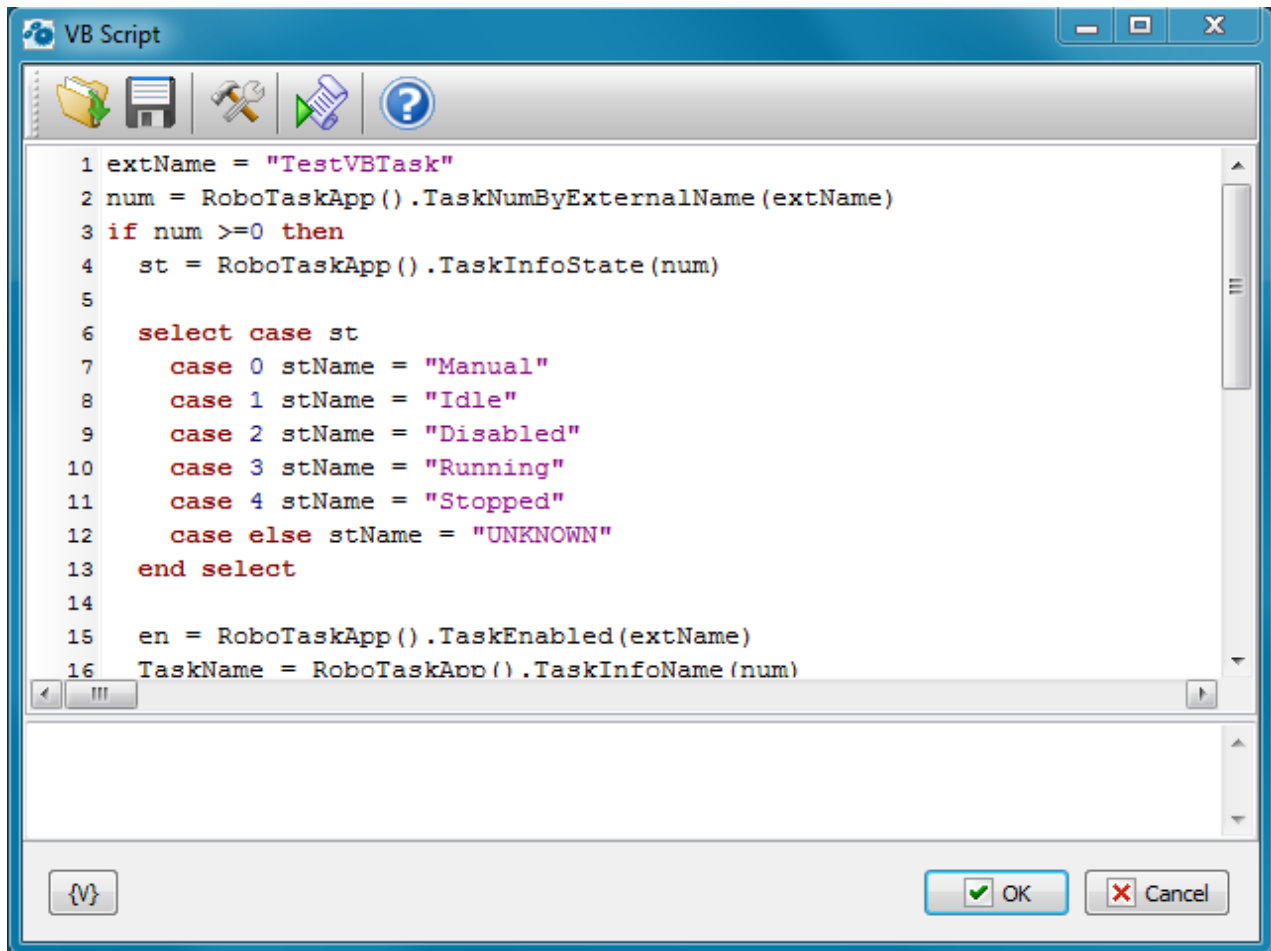
The text of a script can be empty. In this case the action works only with standard functions and objects of JS.

## 4.28.7 VB Script

This action allows to perform a program, which is written in VB Script. Notice that the languages Visual Basic and VB Script are not the same, although they are very alike. The documentation for the language can be read on the Microsoft site:

[http://msdn.microsoft.com/en-us/library/t0aew7h6\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/t0aew7h6(v=vs.84).aspx)

The action editor is a specialized text editor adapted for editing a program code.



### Popup proposals.

When pressing the Ctrl-Space, a popup menu with some functions and objects of VB Script shows up on the screen. One can choose a necessary point from the list and then the editor will insert a proper line/string into a text.

If you began to enter a function name and pressed the Ctrl-Space, only alternatives corresponding to the input line will be left in the popup menu.

### Autocomplete.

This function allows to insert standard language constructions into a text automatically. For this you need to enter 2-5 key constructions (identifier of construction) and press the combination Ctrl-J. The editor inserts the whole construction instead of the key word.

The editor supports the next statements:

Identifier	Description	Insert statement
do	"do" loop	do  loop

foreach	"For each" loop	for each   in  next
for	Simple "for" loop	for i=0 to   step 1  next
func	Function declaration template	function  ()  end Function
if	Simple "IF"	if   then  end if
ife	IF .. Else statement	if   then  else  end if
select	Select case statement	select case   case case case end select
sub	Sub procedure declaration	sub  ()  end sub
while	"While" loop	while    wend
with	"With" statement	with    end with

## Interface

### Load from file

Load the text of a script from a file.

### Save script as file

Save the text of a script into a file.

### Additional options

*Expand Variables in the script* - If to switch on this mode, the RoboTask expands all inserts of variables in a text before running of the script.

*Save Final script into variable* - If it is necessary to monitor the final text of the script, switch on this checkbox and enter the variable name. This mode is need only for control and checkout. The fact of the matter is that the substitution of variable values can break down the syntax of a script text. By means of this mode it is possible to get the final script and analyze it if it is necessary.

The field of a log is under the editor. The messages about runtime errors of the script are output into this field. Also the text of messages LogMessage is output here.

### Run script

Running of a script from the editor to test. If a runtime error arises, you can see a message about the error in the field of the edit log.

### Help

Activation of a help system.

## 4.28.8 VB Evaluate

This action allows to calculate an expression value and then save it into a variable. The expression has to be written according to the syntax of JS language. The documentation for the language can be read on the Microsoft site: [http://msdn.microsoft.com/en-us/library/t0aew7h6\(v=vs.84\).aspx](http://msdn.microsoft.com/en-us/library/t0aew7h6(v=vs.84).aspx)

The action also allows to connect the function library from an external file or directly in an additional script.

For example, try to calculate the expression

```
sum(5,6) + multi(3,4)
```

And we get the error. The VB Script language doesn't contain functions *sum* and *multi*.

To solve this, it is possible to specify a little library of user functions:

```
function sum(a, b)
  sum = a + b
end Function

function multi(a, b)
  multi = a * b
end function
```

In this case the action calculates the result without any troubles. The given example is too simple and is cited for the demonstration solely.



Another example. Calculate the *factorial* of a number. Such function is absent in VB Script. But it is easy to evade. It takes only to create the simply function:

```
function factorial(a)
  if a > 30 then
    'this is synthetic limitation
    err.raise 1, , "Parameter more than 30"
  end if

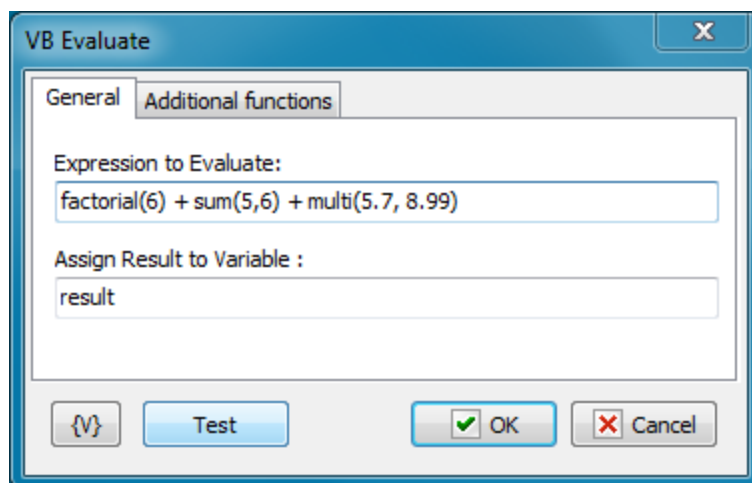
  res = 1
  for i = 1 to a
    res = res * i
  next

  factorial = res
end function
```

Now we easy can calculate the expression:

```
factorial(15)
```

Given possibility allows to do some specific calculations, that are difficult to be written in one line.

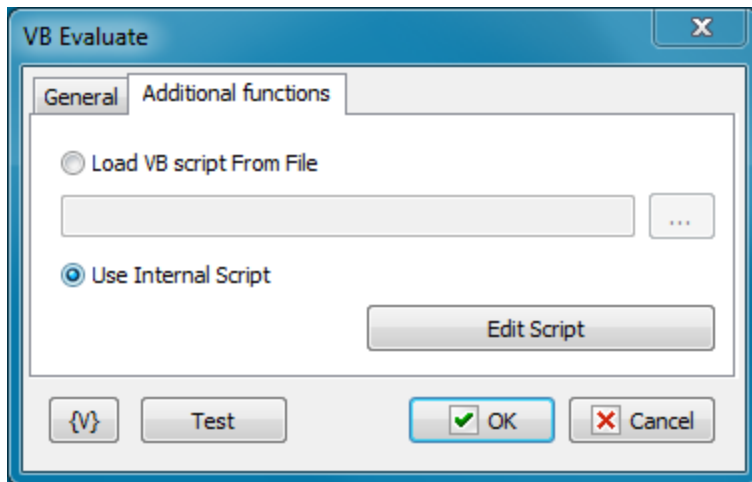


### Expression To Evaluate

Write down the expression whose value needs to be calculated. The expression has to be made according to the syntax of VB Script.

### Assign Result to Variable

Enter the name of the variable into which it is necessary to save the result.



### Load VB Script from File

Load the library of additional functions from an external file. Enter a file name or choose it interactively.

The file name can be empty. In this case the action works only with standard functions and objects of VB Script.

### Use Internal Script

If you want to specify functions directly in a task, choose this option. To edit a script, click the button Edit Script. You can see the editor like in the action [VB Script](#)<sup>[434]</sup>.

The text of a script can be empty. In this case the action works only with standard functions and objects of VB Script.

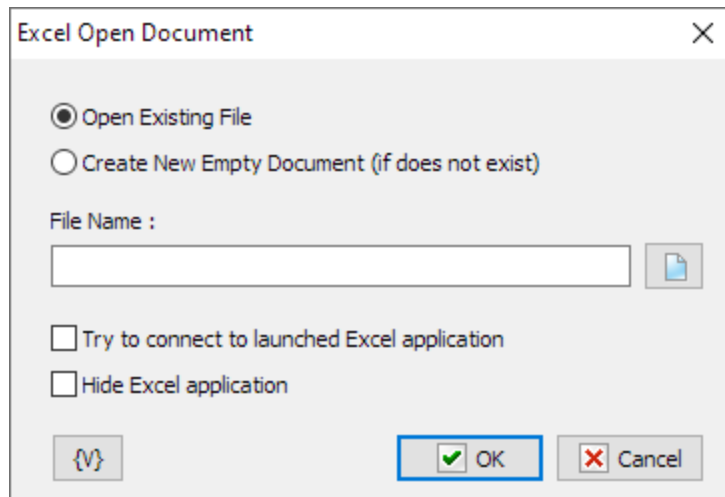
## 4.29 Excel

This group of actions is for working with MS Excel documents. All actions ensure the work by means of application *MS Excel*. Thereby, the MS Office package must have been installed on your computer.

### 4.29.1 Excel Open Document

This action starts the application MS Excel and opens an existent document or creates a new one. You have to open document first before using other Excel actions.

Note that you can open only one document in the task simultaneously. If you want to open another document, you must first [close the open document](#)<sup>[446]</sup>.



### Open Existing File

Open the specified file. If the file does not exist, the action generates a task error.

### Create New Empty Document

Create a new file with a specified name. If such file is already exist, the action just opens it.

### File Name

Specify the file name. Usage of variables is permitted.

### Try to connect to launched EXCEL application

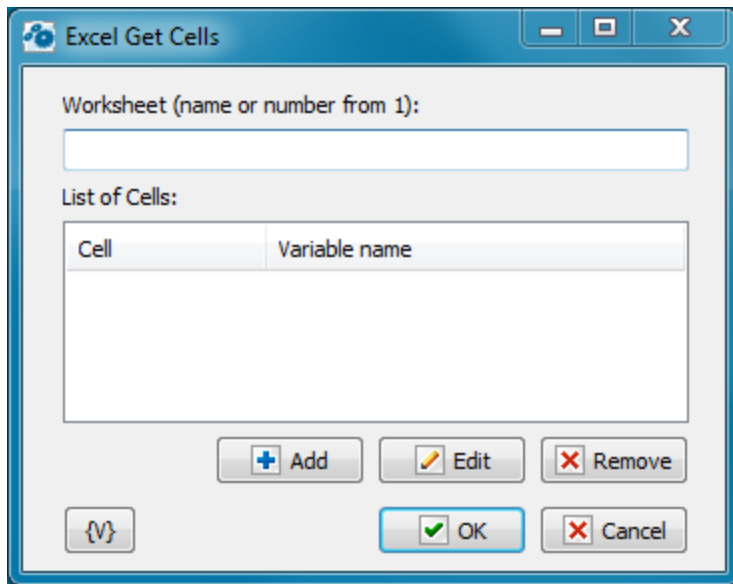
Click this option if you want to try to connect to an already running Excel application first.

### Hide Excel Application

If you need all actions with the document to occur in a hide mode, switch on this checkbox.

## 4.29.2 Excel Get Cells

This action permits to read the values of table cells and write them into variables of RoboTask. This action demand the document to be opened by means of the [Excel Open Document](#)<sup>439</sup>.



### Worksheet

Specify the name or the number of a worksheet of a document. The ordinal number begins from 1.

### List of Cells

The list of cells and variables. The coordinates of every cell can be specified with two ways:

- In a style **A1**, that is an address of a column is written with letters but an address of a row is written with numbers. For example: *C3*, *D5*, *BD15*.

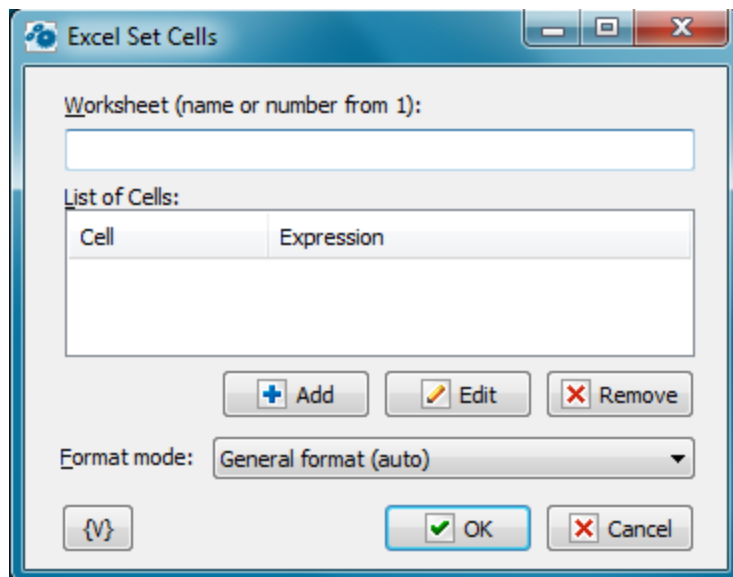
The next column after Z is the column *AA*, *AB*, etc.

- In a style **R1C1**. After the letter R is the number of a row beginning from 1, after the letter C is the number of a column beginning from 1.

Each cell corresponds to the variable name in which the action saves the cell value. The variable name can be specified in an explicit form or by using of another variables.

### 4.29.3 Excel Set Cells

This action permits to change the cell value. This action demand the document to be opened by means of the [Excel Open Document](#)<sup>[439]</sup>



### Worksheet

Specify the name or the number of a document worksheet. The ordinal number begins from 1.

### List of Cells

The list of cells and variables. The coordinates of every cell can be specified with two ways:

- In a style A1, that is an address of a column is written with letters but an address of a row is written with numbers. For example: C3, D5, BD15.  
The next column after Z is the column AA, etc.
- In a style R1C1. After the letter R is the number of a row beginning from 1, after the letter C is the number of a column beginning from 1.

Each cell corresponds to the expression which will be written into the cell. The expression is a string which can be written explicitly or by using of variables.

### Format mode

When writing a value to an Excel Spreadsheet cell, the action can change the format of the cell

- **General Format (auto)** - automatic default format
- **Do not change cell format** - keep cell format unchanged
- **Set value as text** - change the cell format to text

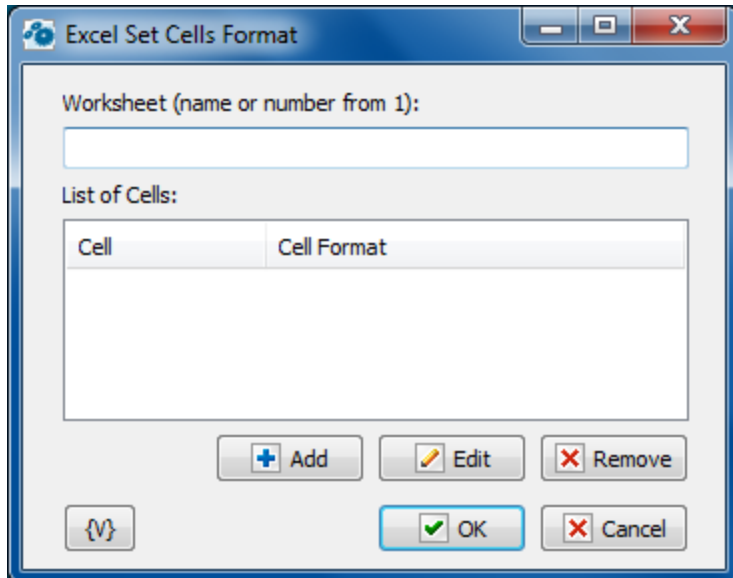
## 4.29.4 Excel Cell Format

MS Excel presents given cells according to the cell format. The *Default* format of data display fits in the most cases. But sometimes it is necessary to change the format forcibly.

For example, a number (an integer or a floating-point number) can be displayed as a number or as a date and time. And both presentation can be correct.

An empty format string corresponds to the default format.

The form for editing of parameters is similar to action editor [Excel Get Cells](#)<sup>[440]</sup> and [Excel Set Cells](#)<sup>[441]</sup>.



### Worksheet

Specify the name or the number of a document worksheet. The ordinal number begins from 1.

### List of Cells

The list of cells and format strings. The coordinates of every cell can be specified with two ways:

- In a style *A1*, that is an address of a column is written with letters but an address of a row is written with numbers. For example: *C3*, *D5*, *BD15*. The next column after Z is the column AA, etc.
- In a style *RIC1*. After the letter R is the number of a row beginning from 1, after the letter C is the number of a column beginning from 1.

Each cell corresponds to the format string. The format string is a string which can be written explicitly or by using of variables.

### ***How to specify the necessary presentation format.***

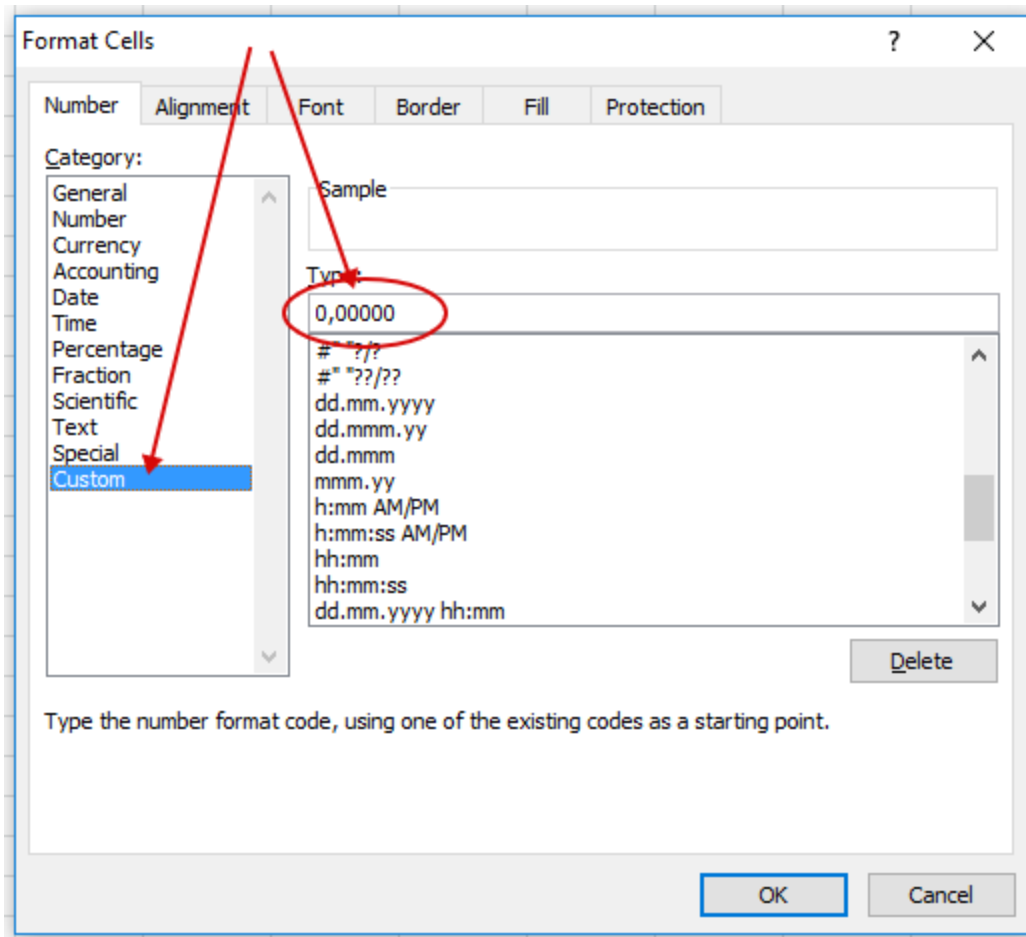
Unfortunately, a format string depends on international settings of Windows™ and language localization of the MS Office™.

For example, what symbol is used as a decimal delimiter: period or comma. The format of data presentation is also different for different countries.

To specify the string of format of data presentation, you need do the next actions:

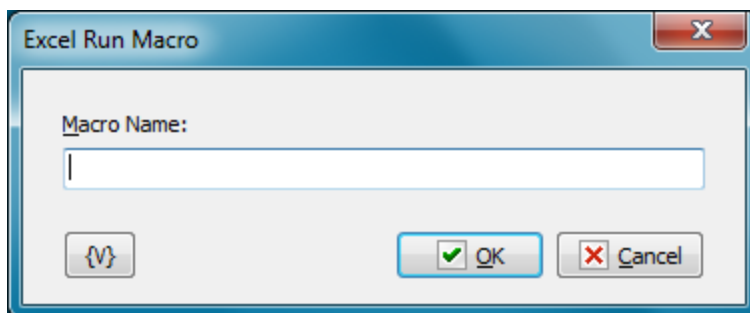
- Open the document or generate an empty document in Excel.
- Set up the necessary format of a cell (right mouse click -> *Format Cells...*, select *Number* tab). You can specify the format interactively selecting the predefined format from the list and change its parameters if it is necessary.

- Then select the *Custom* in the list Category. The cursor will point to a current format string in the field Type (see screenshot).



#### 4.29.5 Excel Run Macro

The Excel can contain macros which need be run to get the necessary result.



#### Macro Name

Specify the name of the macro which is necessary to run. You can use variables.

For example:

TestMacro

The macro name can be specified using the document name.

For example:

```
Book1.xlsm!TestMacro
```

If you need to pass some parameters use such expressions

```
'TestMacro "Hello World! "'
```

or

```
'TestMacro("Hello World!")'
```

If you need to pass some cell value to the macros use *Evaluate* function with cell address.

For example:

```
'TestMacro(evaluate("$A$1"))'
```

or

```
'TestMacro evaluate("$A$1")'
```

Read more detail explanation [here](#).

#### 4.29.6 Excel Save Document

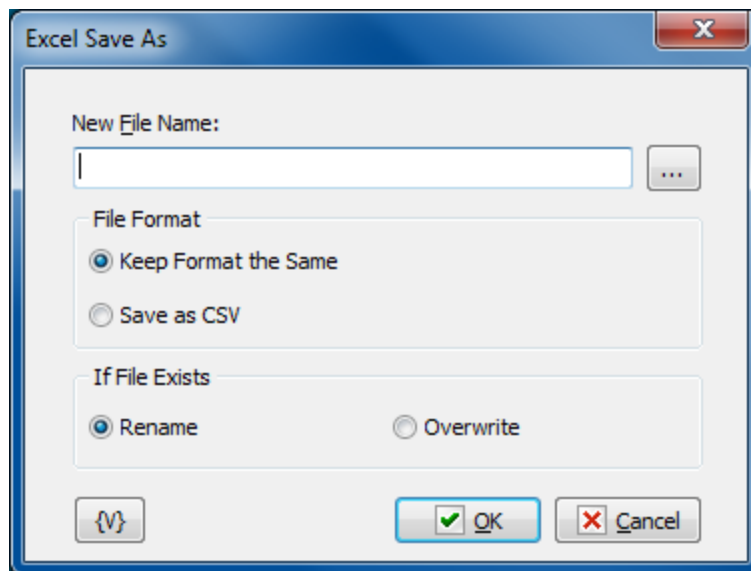
This action saves the changed document in the disk. If you need to save the changing in a file, use this action without fail. *The automatic saving does not occurs when a task ends or a document is closed.*

This action does not demand additional settings and saves the file with the name which is specified in the action [Excel Open Document](#)<sup>439</sup>.

#### 4.29.7 Excel Save As

The action is used to save the Excel document under a different name. It is possible to save in the CSV format (comma separated text).





### New File Name

Input the file name. You can use the Save dialog to select the file interactive. To do it, press the button from the right of the input field.

### File Format

There are two variants available:

- Keep format the same
- Save as CSV

### If File Exists

The action if such file exists already:

- **Rename** – saves under the different name, adds a number to the file name.
- **Overwrite** – overwrites the file if it is possible.

## 4.29.8 Excel Close

This action closes the opened document. The opened document will be closed automatically when a task is executed. As it was said before, the automatic saving does not occur. If you need to save the changes in the document, use the action [Excel Save Document](#)<sup>[445]</sup> before the closing of the document or the ending of the task.

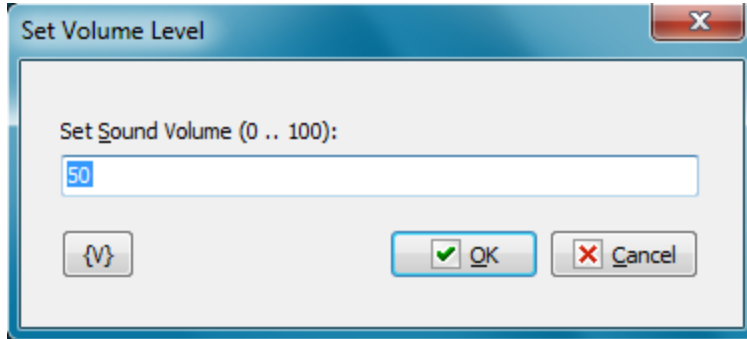
If you want to open two different documents in the task, at the beginning close the first document and then open the second one by means of the [Excel Open Document](#)<sup>[439]</sup>.

## 4.30 Sound Volume

This plugin allows controlling system sound.

### 4.30.1 Set Sound Volume

The action sets system sound volume level.

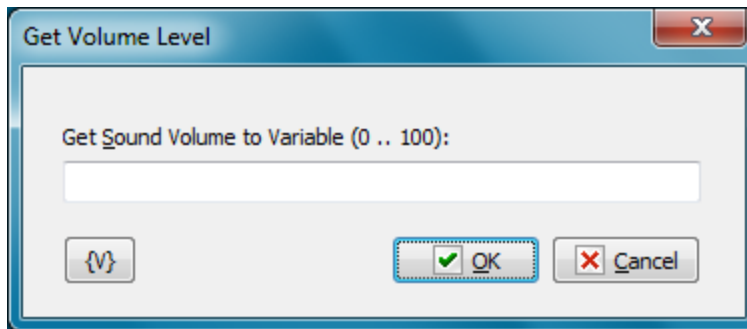


#### **Set sound Volume (0 .. 100)**

Specify the number from 0 up to 100. You can use variables.

### 4.30.2 Get Sound Volume

The action allows retrieving current sound volume level and saving it in variable. The value is the integer number from 0 up to 100.

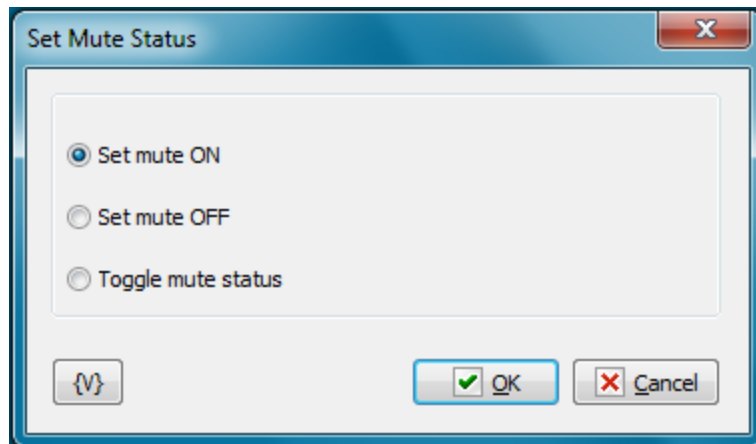


#### **Get Sound Volume to Variable (0 .. 100)**

Specify the variable name where you want to save the result.

### 4.30.3 Set Mute Status

The action can mute the sound, turn on the sound or switch mute status.



#### Mode

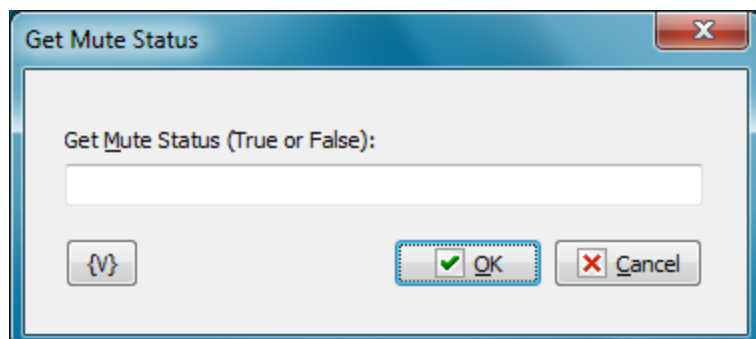
- **Set mute ON** – turn off the sound
- **Set mute OFF** – turn on the sound
- **Toggle mute status** – switch the mute status
- 

#### 4.30.4 Get Mute Status

The action retrieves the mute status and saves the logical value to variable.

*True* – mute is ON

*False* – mute is OFF



#### Get Mute Status (TRUE or FALSE)

Specify the variable name where you want to save the result.

#### 4.30.5 Sound Volume Up

The action raises the sound volume level on 2%. The action has not any parameters.

## 4.30.6 Sound Volume Down

The action lowers the sound volume level on 2%. The action has not any parameters.

## 4.31 DataBase

The actions for working with a database are collected in this group. In contrast to actions [SQL Query](#)<sup>[91]</sup> и [Dataset Loop](#)<sup>[236]</sup>, all these actions work in one session. At first you have to open a database ([DB Open](#)<sup>[449]</sup>), perform necessary actions and then close the database ([DB Close](#)<sup>[450]</sup>). Thereby, you can execute a complex sequence of actions with a database as a single block.

The actions [SQL Query](#)<sup>[91]</sup> and [Dataset Loop](#)<sup>[236]</sup> open their session for processing query and serve to perform single queries.

The group of actions DataBase works on basis of the mechanism ADO (ActiveX Data Objects) and can work with any database if the system has a proper driver.

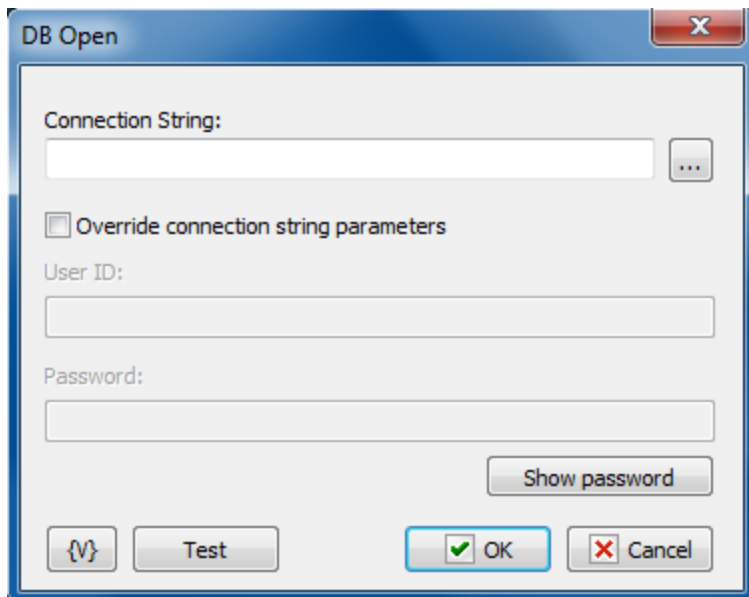
**Note:** This feature is available for Business license only

### 4.31.1 DB Open

The action opens a database. All actions with a database must begin from opening the database. If you try to process any actions with the database without its opening, you will get the error *Database Is Not Opened*.

Every task permits only one opened database (DB). If you need open another DB, you should close the previous session.

If you try to open the DB when the session is opened, the action will give the warning *Database Connection Is Already Opened* and ignores the operation of opening.



### Connection String

---

Input a connection string to connect to a database. You can use a system constructor pressing the button on the right of the entry field.

### **Override connection string parameters**

If you need to override authorization parameters, turn on this switch and specify the User ID and Password. These parameters will be used while opening the DB.

#### **User ID**

Input a new user name.

#### **Password**

Input a password.

***Note:** This feature is available for Business license only*

### **4.31.2 DB Close**

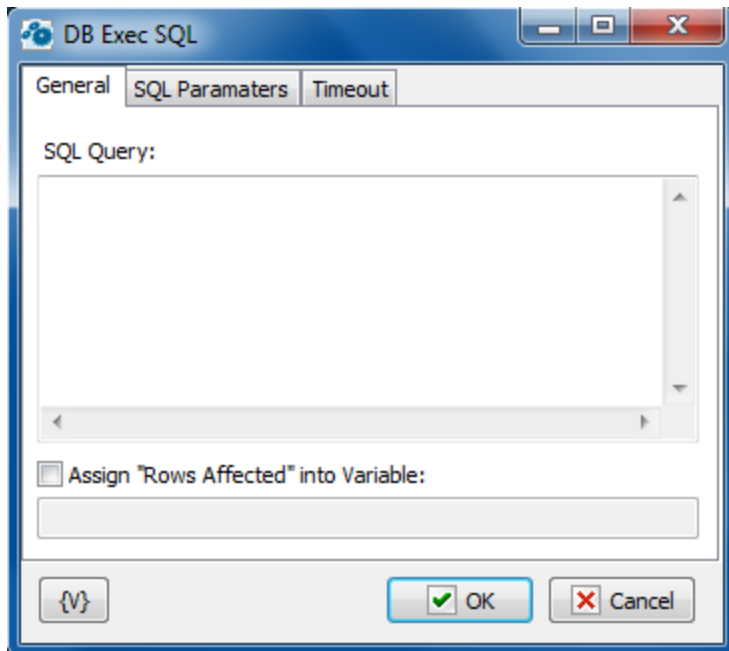
Closes the currently work session with the database. Any additional parameters are not required.

***Note:** This feature is available for Business license only*

### **4.31.3 DB Exec SQL**

Processes an SQL query. The action is similar to the SQL Query action. But this action is performed within an opened database session. Thereby, you can process a series of queries in a single block.

#### **General**



### SQL Query

Input an SQL query according to syntax. You can use parameters. It is necessary to put the colon (:) in front of the parameter name.

For example:

```
Select * from dept where city = :citypmt
```

*citypmt* is a parameter of the query.

### Assign Rows affected to variable

Turn on this checkbox and input a variable name if you need to save the number of processed records into a variable.

### SQL Parameters

Sometimes using of parameters of a query is more convenient instead of generation of the query by means of variables. For example, when the variable value contains quotes or another special symbols. If you use variables (or macro substitution) when forming SQL inquiry, availability such symbols can "break" the syntax of the query and you will get a runtime error.

Parameters SQL are designated in the inquiry by means of colon (:) before the parameter name.

For example:

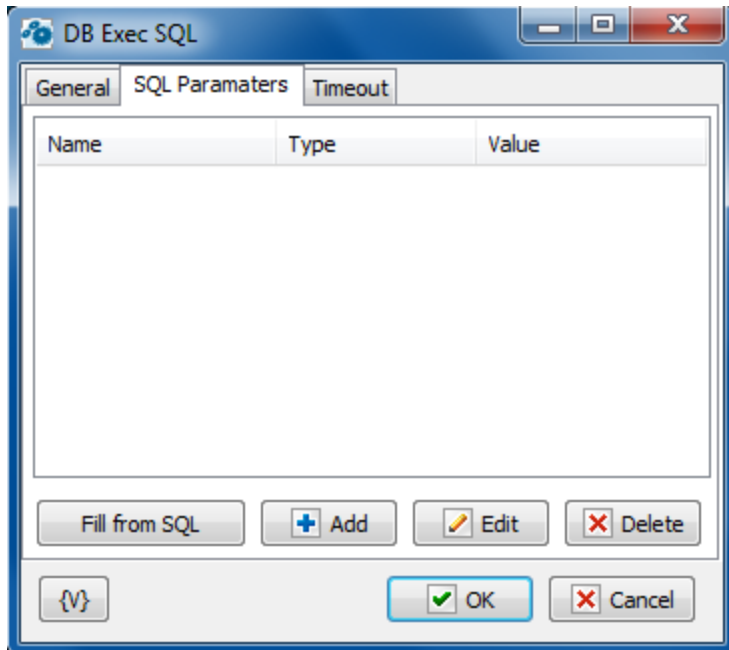
```
Select * from employee where hiredate > :dt and salary >= :amount
```

*:dt* and *:amount* are parameters of the query.

Parameters can be several types:

- **String** – any string
- **Integer** – an integer number
- **Float** – a real number or the number with a floating dot.

- **Boolean** – a logical value True or False.
- **DateTime** – a date and time. The value of such parameter must be defined in current system format of date and time (see the system regional settings). Pay attention that using of date-time can have its own characteristics for different databases. For example, it is better specify such parameters for the MySQL as a string parameter in the format **YYYY-MM-DD** or **YYYY-MM-DD hh:nn:ss**.



### Fill from SQL

Fill the list of parameters from the current query. If a parameter is already in the list, this parameter will be without change. The function fills only the missing parameters in the list.

### Add

Add a parameter manually. You will be proposed to type or select from the list the parameter name, select the parameter type and input an expression as the parameter value.

Using of variables in a parameter name and in its value is allowed. The RoboTask will substitute variable values instead of variables before performing the SQL.

### Edit

Edit the properties of the selected parameter.

### Delete

Delete the selected parameter from the list.

If you specify the parameter which is absent in the query, the action ignores this parameter when performing and writes the proper warning in the log of the task.

**Timeout**

There you can change the time of waiting of a query process. On default, timeout is 60 seconds. But in case of need, it is possible to change this parameter.

**Note:** *This feature is available for Business license only*

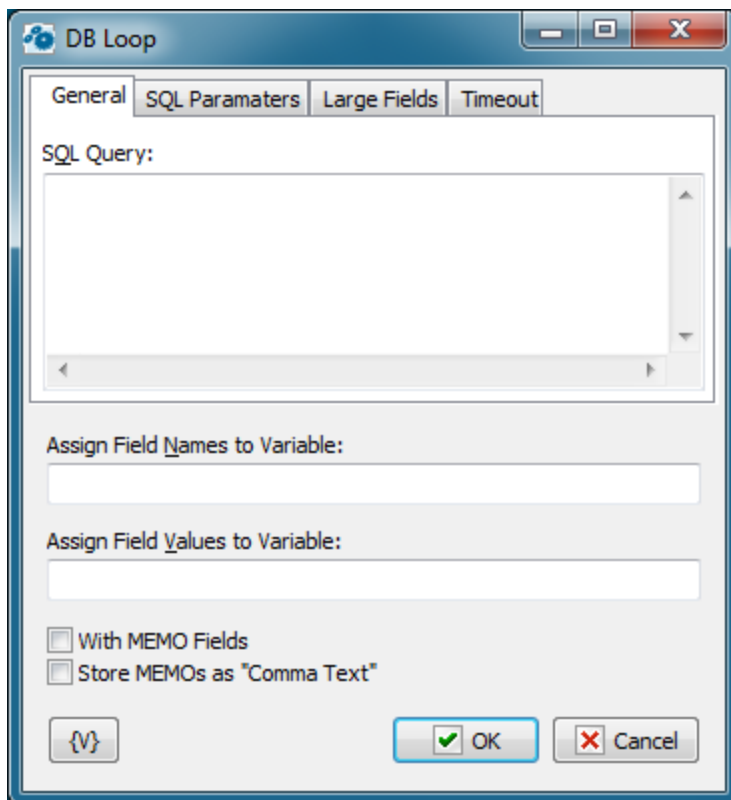
**4.31.4 DB Loop**

This action is similar to the [Dataset Loop](#)<sup>[236]</sup>. The difference is that the action works within already open session.

You need to use as a query the expressions which return a set of records or strings. For instance, queries *SELECT*. DB Loop sorts out the whole set of records in a loop sequentially.

**Important**

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

**General****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.

## SQL parameters

Sometimes using of parameters of a query is more convenient instead of generation of the query by means of variables. For example, when the variable value contains quotes or another special symbols. If you use variables (or macro substitution) when forming SQL inquiry, availability such symbols can "break" the syntax of the query and you will get a runtime error.

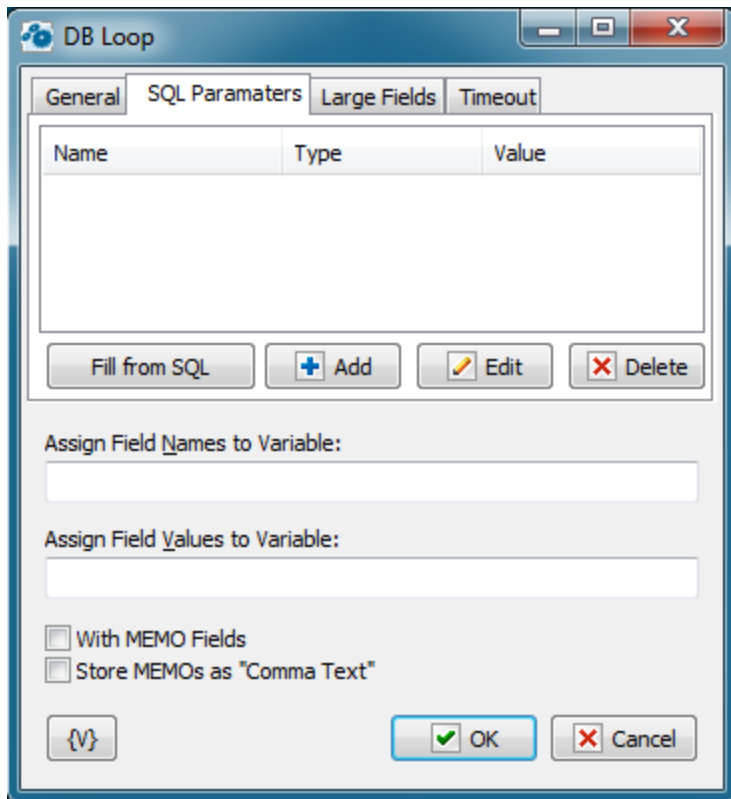
Parameters SQL are designated in the inquiry by means of colon (:) before the parameter name.

For example:

```
Select * from employee where hiredate > :dt and salary >= :amount  
:dt and :amount are parameters of the query.
```

Parameters can be several types:

- **String** – any string
- **Integer** – an integer number
- **Float** – a real number or the number with a floating dot.
- **Boolean** – a logical value True or False.
- **DateTime** – a date and time. The value of such parameter must be defined in current system format of date and time (see the system regional settings). Notice that using of date-time can have its own characteristics for different databases. For example, it is better specify such parameters for the MySQL as a string parameter in the format YYYY-MM-DD or YYYY-MM-DD hh:nn:ss.



### Fill from SQL

Fill the list of parameters from the current query. If a parameter is already in the list, this parameter will be without change. The function fills only the missing parameters in the list.

### Add

Add a parameter manually. You will be proposed to type or select from the list the parameter name, select the parameter type and input an expression as the parameter value.

Using of variables in a parameter name and in its value is allowed. The RoboTask will substitute variable values instead of variables before performing the SQL.

### Edit

Edit the properties of the selected parameter.

### Delete

Delete the selected parameter from the list.

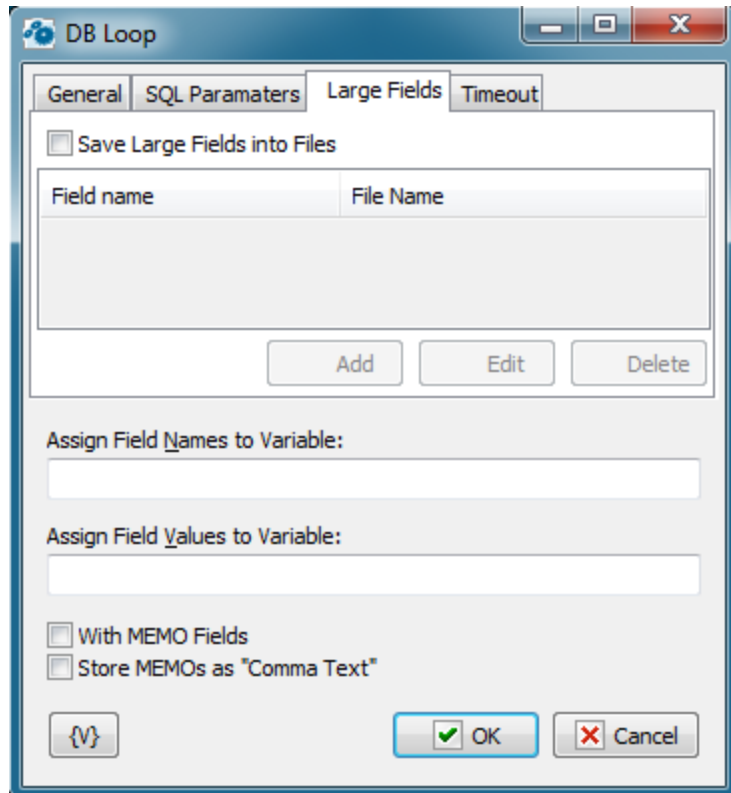
If you specify the parameter which is absent in the query, the action ignores this parameter when performing and writes the proper warning in the log of the task.

### Large fields

You can tune up the saving of the field value to file on this tab. As a rule, the necessity of export to file exists for large fields (BLOB, CLOB, Memo). These fields do not have a fixed size

and can contain unexpectedly big information content. The BLOB field can contain a binary massive which cannot be saved to RoboTask variable.

The action also allows you to save to a text file a value of any type, not only large fields.



### Save Large Fields into Files

Switch on this checkbox to save field values to a file.

After that you need to fill in the list of fields to save. Each field must correspond to the file name to save.

You can use variables for the field name and file name.

**But pay attention** that the field name is calculated when the loop is initialized and stay permanent during the whole cycle. The file name is calculated at each iteration before saving information to the file. Therefore, for a file name, you can use any variables which change their value at each iteration.

### Add, Edit, Delete

Buttons for editing a field list.

### Timeout

There you can change the time of waiting of a query process. On default, timeout is 60 seconds. But in case of need, it is possible to change this parameter.

**Note:** *This feature is available for Business license only*

### 4.31.5 DB Begin Transaction

The action begins the transaction of a database. The transaction can be finished ([DB Commit](#)<sup>[457]</sup>) or canceled ([DB Rollback](#)<sup>[457]</sup>). Other users cannot see any changes of database from the moment of the beginning of the transaction to the finish of it.

In case of commit, all changes of the data base from the moment of the beginning of the transaction will be written in the database and accessible to other users of the database.

In case of rollback of the transaction, all changes of the data base from the moment of the beginning of the transaction will be canceled.

The transaction acts in a current session with a data base. The action do not require any additional parameters.

**Note:** *This feature is available for Business license only*

### 4.31.6 DB Commit

The action saves all changes of a data base from the beginning of a transaction. Other users of the data base can see all the changes

The action do not require any additional parameters.

**Note:** *This feature is available for Business license only*

### 4.31.7 DB Rollback

The action cancels all changes of a data base from the beginning of a transaction.

The action do not require any additional parameters.

**Note:** *This feature is available for Business license only*

## 4.32 CSV Data

This group contains actions for working with CSV texts. CSV data (comma-separated values) is a popular format for data exchange between different systems. However, manual processing of this data can be challenging because CSV data is simply text. Generally, to perform any actions with this data, you need to load it into a database or Excel. These actions will help you avoid that.

To manipulate the data, you need to first open the data. Use the "[CSV Open Data](#)"<sup>[458]</sup> action. Additionally, almost every action allows you to independently load data from a file or variable.

All actions are designed in such a way that opened CSV data is stored in memory until the task is completed.

Note that the data is not automatically exported to a file or variable. To save the data, you must use the "[CSV Save Data](#)"<sup>[479]</sup> action.

**Note:** This feature is available for Business license only

### 4.32.1 CSV Open Data

This action opens CSV data for processing in subsequent steps.

#### CSV data

CSV Open Data

CSV data | Source format

CSV source

File  Text

Skip first lines:

0  Skip empty lines

Skip lines starting with:  First row is column headers

#

File name:

Encoding:

Default

{V} OK Cancel

#### CSV source

There are two options to choose from:

- *File*: Read data from a file. Specify the file name below and select the encoding format. The most popular encodings are available for selection.
- *Text*: Load data directly from the specified text. You can provide the text explicitly or using variables.

If CSV data has been loaded previously in the previous task steps, the data will be overwritten. You can only use one set of CSV data within one task.

#### Skip first lines

Specify how many lines need to be ignored from the beginning of the text.

### Skip empty lines

Enable this switch if you want to ignore empty lines. This parameter is enabled by default.

### Skip lines starting with

Ignore comments. There is no standard for comments in CSV texts. The action will consider a line as a comment if it starts with the specified combination of characters. You can choose an option or specify your own combination.

### First row is column headers

The CSV text may start with a row of column headers. Enable this switch if the first row of the data array is the header.

## Source Format

The screenshot shows a dialog box titled "CSV Open Data" with a "Source format" tab selected. The "Delimiter:" dropdown is set to "Semicolon - ;". The "Character:" field is empty. The "Quote character:" dropdown is set to "Double quote - \"". The "Strict delimiter (space is not a delimiter)" checkbox is checked. There are two buttons: "Reset to standard format" and "Reset to Excel format". At the bottom, there is a "Cancel" button and an "OK" button with a green checkmark icon.

Specify the format of the CSV data: delimiter, quote character.

Note that if the "Strict delimiter" switch is not enabled, spaces are also considered delimiters. And all fields containing spaces must be enclosed in quotes.

### Reset to standard format

---

Sets the standard parameters:

- Delimiter: Comma (,)
- Quotes: Double quotes (")
- Non-strict delimiter (spaces are also considered delimiters)

#### **Reset to Excel format**

Sets the parameters used by MS Excel by default:

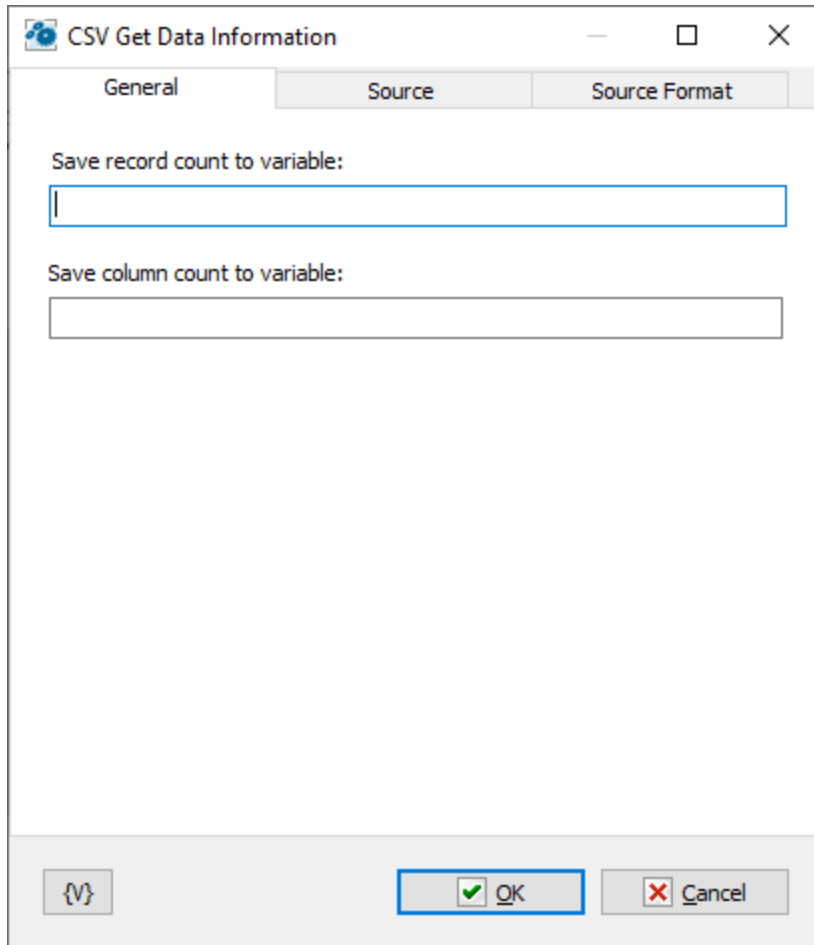
- Delimiter: Semicolon (;)
- Quotes: Double quotes (")
- Strict delimiter.

**Note:** *This feature is available for Business license only*

### **4.32.2 CSV Get Data Information**

This action retrieves general information about the loaded data: the number of records and the number of columns.

## General



The screenshot shows a dialog box titled "CSV Get Data Information" with three tabs: "General", "Source", and "Source Format". The "General" tab is selected. It contains two text input fields. The first is labeled "Save record count to variable:" and is currently empty. The second is labeled "Save column count to variable:" and is also empty. At the bottom of the dialog, there are three buttons: a help button with a question mark icon, an "OK" button with a green checkmark icon, and a "Cancel" button with a red X icon.

Enter the variable names to store the data. Fields can be left empty. In that case, the information will not be saved.



## Source

The screenshot shows a dialog box titled "CSV Get Data Information" with three tabs: "General", "Source", and "Source Format". The "Source" tab is active. It contains the following elements:

- CSV source:** Three radio buttons: "Prev. CSV action" (unselected), "File" (selected), and "Text" (unselected).
- Skip first lines:** A numeric input field set to "0" and a checked checkbox "Skip empty lines".
- Skip lines starting with:** An unchecked checkbox and a dropdown menu showing "#".
- First row is column headers:** An unchecked checkbox.
- File name:** A text input field with a browse button "...".
- Encoding:** A dropdown menu set to "Default".
- Buttons:** A refresh button (circular arrow), an "OK" button (highlighted with a blue border), and a "Cancel" button.

### CSV Source

There are three options to choose from:

- *Previous CSV actions* - use the data opened in previous actions. If you need to load new CSV data, select the File or Text options.
- *File* - read data from a file. Specify the file name below and select the encoding format. The most popular encodings are available for selection.
- *Text* - load data directly from the specified text. You can provide the text explicitly or using variables.

Note that if you choose *File* or *Text* as the source and CSV data has been loaded previously in the previous task steps, the data will be overwritten. You can only use one set of CSV data within one task.

### Skip first lines

Specify how many lines need to be ignored from the beginning of the text.

### Skip empty lines

Enable this switch if you want to ignore empty lines. This parameter is enabled by default.

### Skip lines starting with

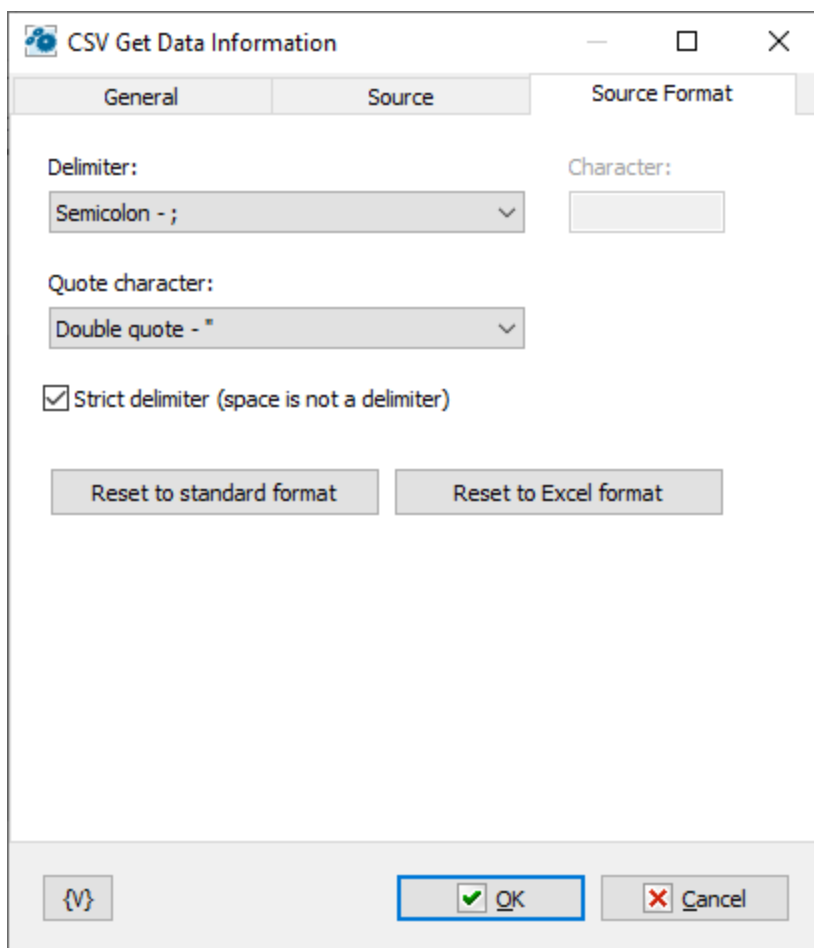
Ignore comments. There is no standard for comments in CSV texts. The action will consider a line as a comment if it starts with the specified combination of characters. You can choose an option or specify your own combination.

### First row is column headers

The CSV text may start with a row of column headers. Enable this switch if the first row of the data array is the header.

## Source Format

Editing the CSV format for loading is available if File or Text is selected as the source.



The screenshot shows the 'Source Format' tab of the 'CSV Get Data Information' dialog box. It features three tabs: 'General', 'Source', and 'Source Format'. The 'Source Format' tab is active and contains the following controls:

- Delimiter:** A dropdown menu set to 'Semicolon - ;'.
- Character:** An empty text input field.
- Quote character:** A dropdown menu set to 'Double quote - "'.
- Strict delimiter (space is not a delimiter)**
- Two buttons: 'Reset to standard format' and 'Reset to Excel format'.
- At the bottom, there are three buttons: a help icon '{V}', a highlighted 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

Specify the format of the CSV data: delimiter, quote character.

Note that if the "Strict delimiter" switch is not enabled, spaces are also considered delimiters. And all fields containing spaces must be enclosed in quotes.

### Reset to standard format

Sets the standard parameters:

- Delimiter: Comma (,)
- Quotes: Double quotes (")
- Non-strict delimiter (spaces are also considered delimiters)

### Reset to Excel format

Sets the parameters used by MS Excel by default:

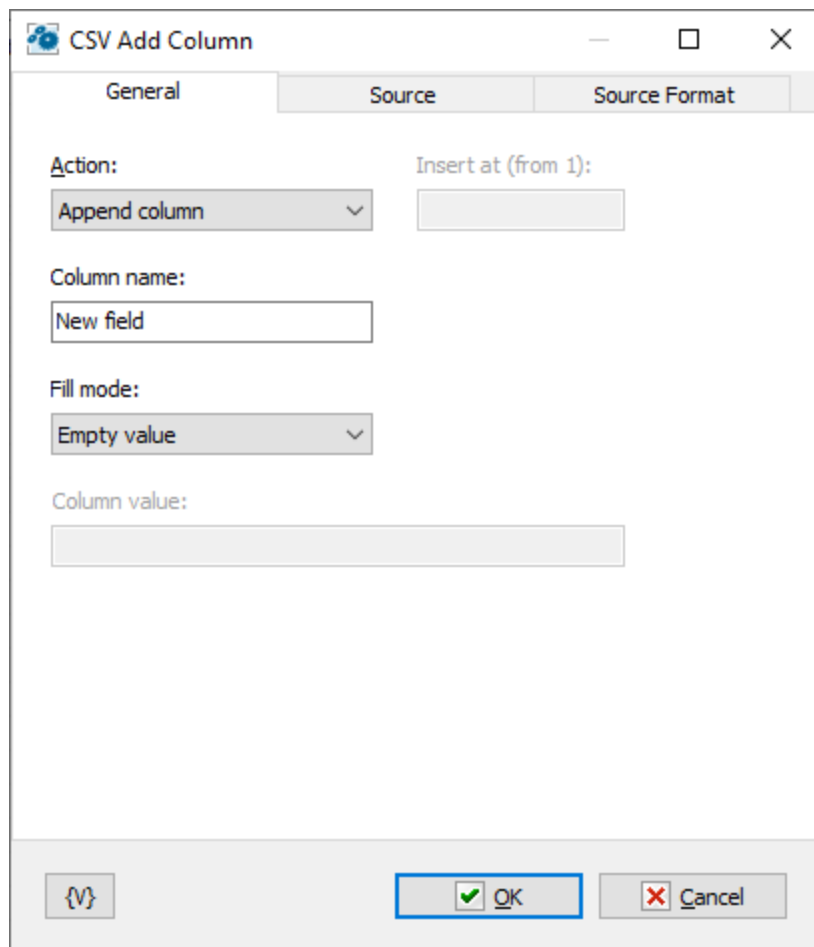
- Delimiter: Semicolon (;)
- Quotes: Double quotes (")
- Strict delimiter.

**Note:** *This feature is available for Business license only*

### 4.32.3 CSV Add Column

This action adds a column to the CSV data.

#### General



The screenshot shows a dialog box titled "CSV Add Column" with three tabs: "General", "Source", and "Source Format". The "General" tab is active. It contains the following fields:

- Action:** A dropdown menu set to "Append column".
- Insert at (from 1):** An empty text input field.
- Column name:** A text input field containing "New field".
- Fill mode:** A dropdown menu set to "Empty value".
- Column value:** A large empty text area.

At the bottom of the dialog, there are three buttons: a placeholder button with "{V}", an "OK" button with a green checkmark, and a "Cancel" button with a red X.

**Action**

Column addition mode:

- *Append column* - add the column at the end.
- *Insert column* - insert the column at the specified position. Note that the column numbering starts from 1. If the specified position is less than 1 or greater than the number of columns, the column will be added at the end.

**Column name**

Enter the name of the new column.

**Fill mode**

Fill mode for the new column. Three options are available:

- *Empty value*
- *Sequence number* - the action fills the field with the row number, starting from 1.
- *Static value* - the action fills all entries of the new field with the specified value.

**Source and Source format**

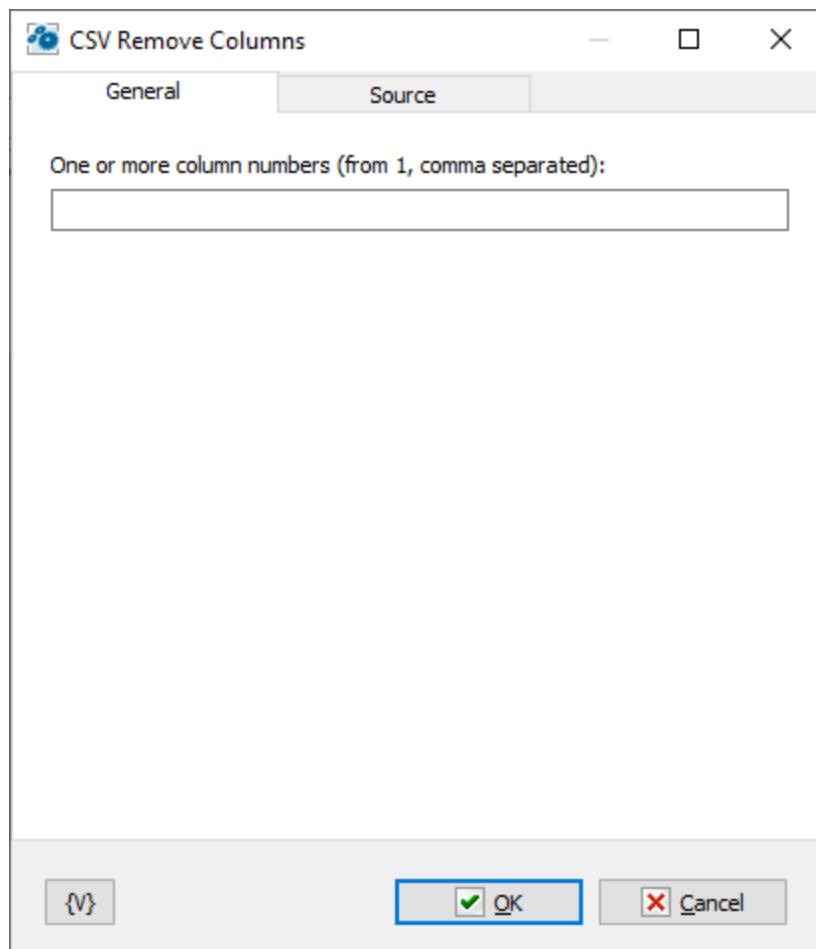
[Source](#)<sup>[462]</sup> and [Source format](#)<sup>[463]</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>[460]</sup> chapter

**Note:** *This feature is available for Business license only*

#### 4.32.4 CSV Remove Column

This action removes the specified column by numbers. Column numbers should be listed separated by commas. If a column number is less than 1 or greater than the number of columns, the action logs a warning in the task log. If it is not possible to remove any column, the action generates an error.

**General**



One or more column numbers

List the column numbers to be removed, separated by commas. Column numbers start from 1 and should not exceed the total number of columns.

### Source and Source format

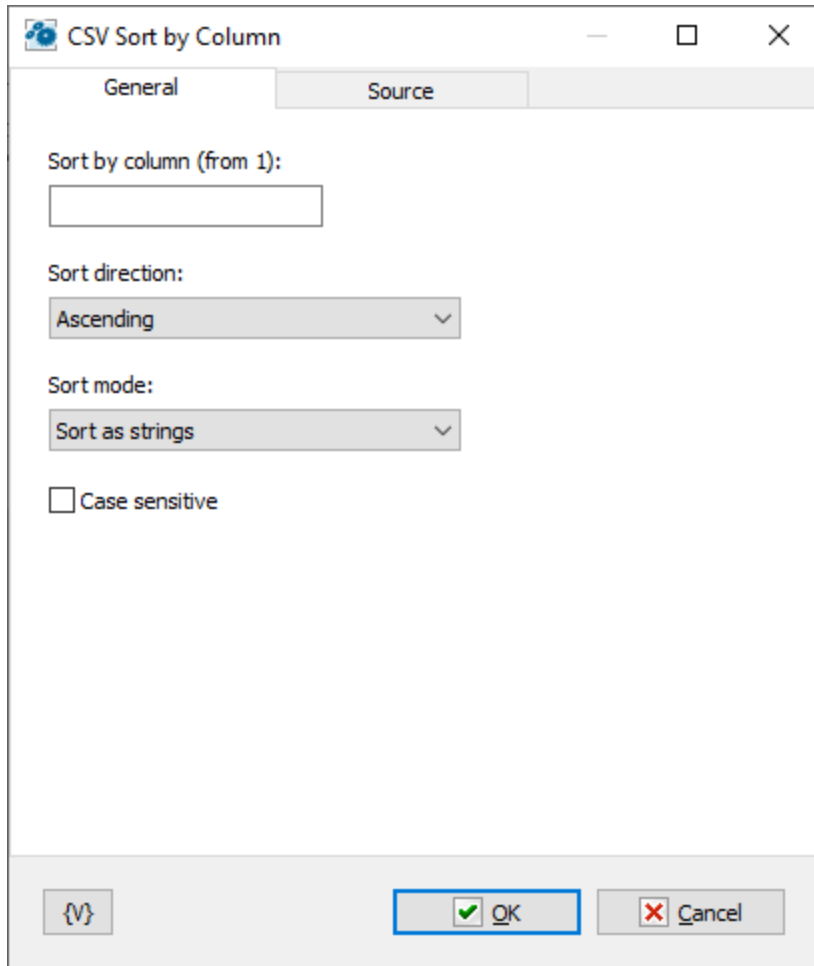
[Source](#)<sup>[462]</sup> and [Source format](#)<sup>[463]</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>[460]</sup> chapter

**Note:** *This feature is available for Business license only*

### 4.32.5 CSV Sort by Column

This action sorts the CSV data based on the specified column.

## General



CSV Sort by Column

General Source

Sort by column (from 1):

Sort direction:

Ascending

Sort mode:

Sort as strings

Case sensitive

{V} OK Cancel

### Sort by column

Specify the column number to sort the data by.

### Sort direction

Select the sorting direction: ascending or descending.

### Sort Mode

Comparison mode for sorting: as strings or as numbers.

### Case Sensitive

Enable this toggle if you want to compare strings in a case-sensitive manner.

### Source and Source format

[Source](#)<sup>[462]</sup> and [Source format](#)<sup>[463]</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>[460]</sup> chapter

**Note:** This feature is available for Business license only

### 4.32.6 CSV Move Column

This action moves a column to a new position.

#### General

The screenshot shows a dialog box titled "CSV Move Column". It has two tabs: "General" and "Source". The "Source" tab is selected. Inside the dialog, there are two input fields: "Column number (from 1):" and "Move to:". At the bottom of the dialog, there are three buttons: a help button with a question mark icon, an "OK" button with a green checkmark, and a "Cancel" button with a red X.

#### Column number

Specify the column number to be moved. Column numbers start from 1.

#### Move to

Specify the new position for the column.

#### Source and Source format

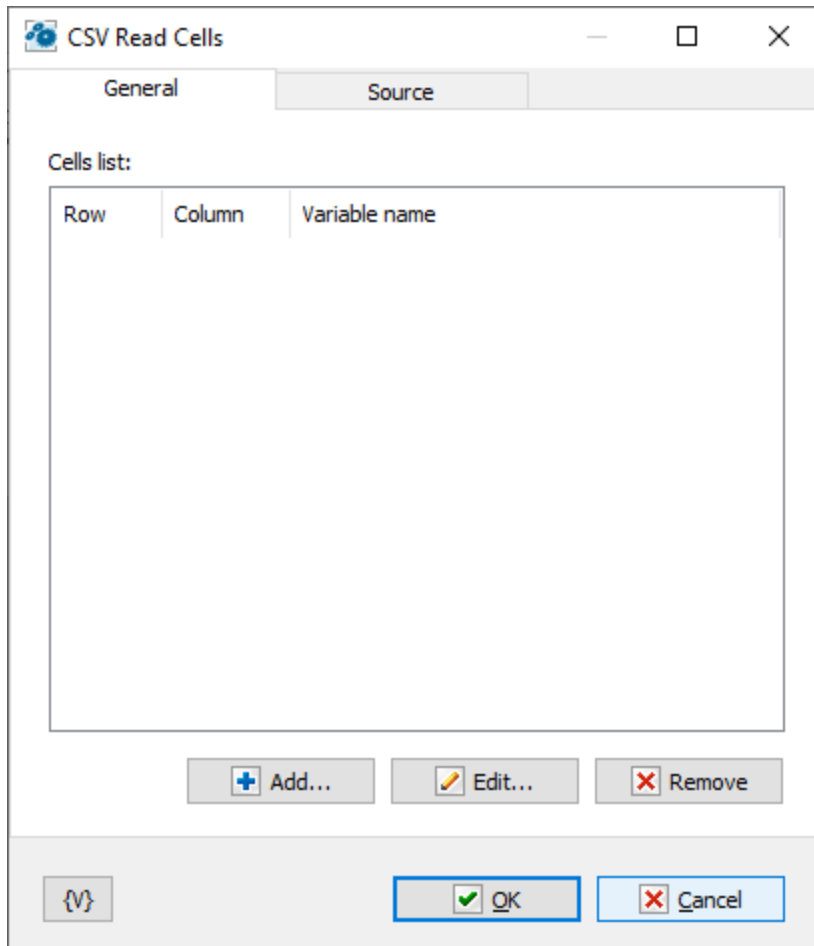
[Source](#)<sup>462</sup> and [Source format](#)<sup>463</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>460</sup> chapter

**Note:** This feature is available for Business license only

### 4.32.7 CSV Read Cells

This action reads CSV data fields and saves them into the specified variables.

#### General



#### Cells list

This is a list of fields in the data array that need to be read, specified by their coordinates (row, column). For each cell, you should indicate the variable in which the read value should be stored. The row and column numbers start from 1.

#### Source and Source format

[Source](#)<sup>[462]</sup> and [Source format](#)<sup>[463]</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>[460]</sup> chapter

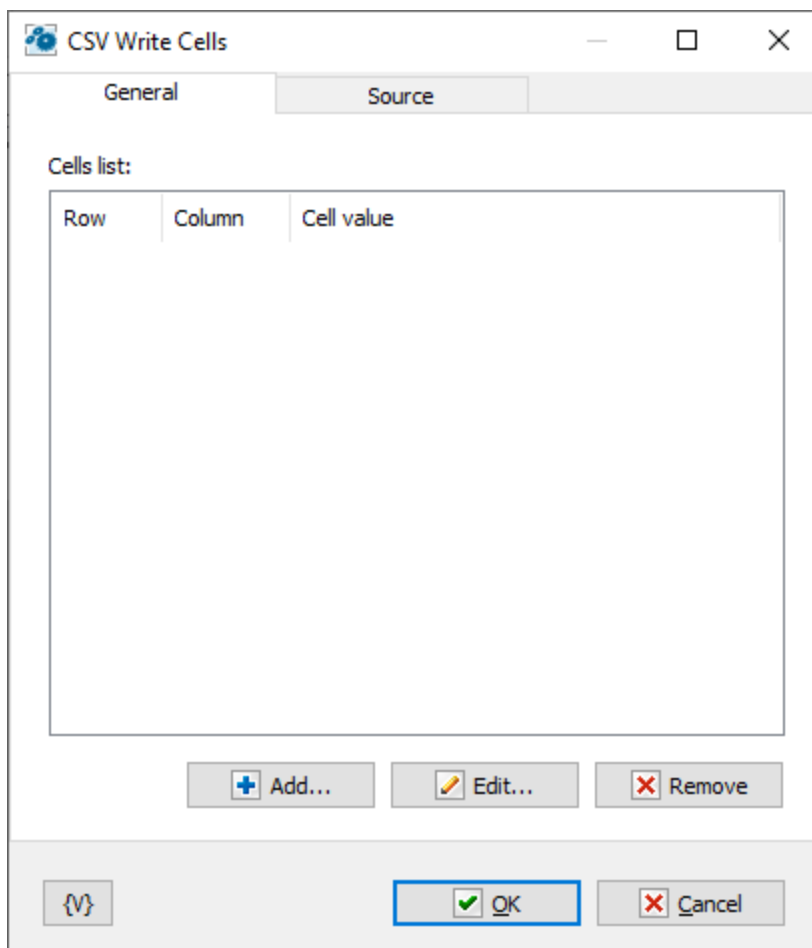


**Note:** This feature is available for Business license only

### 4.32.8 CSV Write Cells

This action allows you to write new values to CSV data fields.

#### General



#### Cells list

This is a list of fields in which new values should be written, specified by their coordinates (row, column). For each cell, you should indicate the value to be written. The row and column numbers start from 1.

#### Source and Source format

[Source](#)<sup>[462]</sup> and [Source format](#)<sup>[463]</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>[460]</sup> chapter

**Note:** This feature is available for Business license only

### 4.32.9 CSV Merge Data

This action allows you to merge two CSV data arrays into one. Data merging is done by adding data from another source: file, variables, or explicitly specified data to the already opened CSV array.

#### Data to add

Data to be added to already opened CSV data.

#### CSV source

There are two options to choose from:

- *File*: Read data from a file. Specify the file name below and select the encoding format. The most popular encodings are available for selection.
- *Text*: Load data directly from the specified text. You can provide the text explicitly or using variables.

If CSV data has been loaded previously in the previous task steps, the data will be overwritten. You can only use one set of CSV data within one task.

#### Skip first lines

Specify how many lines need to be ignored from the beginning of the text.

### Skip empty lines

Enable this switch if you want to ignore empty lines. This parameter is enabled by default.

### Skip lines starting with

Ignore comments. There is no standard for comments in CSV texts. The action will consider a line as a comment if it starts with the specified combination of characters. You can choose an option or specify your own combination.

### First row is column headers

The CSV text may start with a row of column headers. Enable this switch if the first row of the data array is the header.

### Data format

The screenshot shows a dialog box titled "CSV Merge Data" with three tabs: "Data to add", "Data Format", and "Advanced". The "Data Format" tab is active. It contains the following settings:

- Delimiter:** A dropdown menu set to "Semicolon - ;".
- Character:** An empty text input field.
- Quote character:** A dropdown menu set to "Double quote - ""
- Strict delimiter (space is not a delimiter):** A checked checkbox.
- Two buttons: "Reset to standard format" and "Reset to Excel format".

At the bottom of the dialog, there are three buttons: a button with "{V}", a highlighted "OK" button with a green checkmark, and a "Cancel" button with a red X.

Specify the format of the CSV data: delimiter, quote character.

Note that if the "Strict delimiter" switch is not enabled, spaces are also considered delimiters. And all fields containing spaces must be enclosed in quotes.

### Reset to standard format

Sets the standard parameters:

- Delimiter: Comma (,)
- Quotes: Double quotes (")
- Non-strict delimiter (spaces are also considered delimiters)

### Reset to Excel format

Sets the parameters used by MS Excel by default:

- Delimiter: Semicolon (;)
- Quotes: Double quotes (")
- Strict delimiter.

## Advanced

The screenshot shows the 'CSV Merge Data' dialog box with the 'Advanced' tab selected. The dialog has three tabs: 'Data to add', 'Data Format', and 'Advanced'. The 'Advanced' tab contains two sections of options:

- If the number of columns is different ...**
  - Raise an error
  - Adjust the number of columns
- Merge mode**
  - Intersection
  - Subtraction
  - Union (default)
  - Symmetric difference

At the bottom of the dialog, there are three buttons: a help icon (question mark), an 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

### If the number of columns is different

Choose the action to take if the number of columns differs. The default action is to generate an error.

## Merge mode

There are four modes:

- *Intersection* - elements that are in both lists
- *Union* - all elements from both lists. Default option
- *Subtraction* - elements from the 1st list and missing from the 2nd list (1-st list minus 2-nd list)
- *Symmetric difference* - the union minus intersection

**Note:** *This feature is available for Business license only*

### 4.32.10 CSV Filter Data

This action allows you to filter CSV data records based on specified conditions. All records that do not meet your criteria will be deleted. This can be useful to reduce data for further processing if needed.

#### General

The screenshot shows the 'CSV Filter Data' dialog box with the 'General' tab selected. The 'Delete rows that ...' section has two radio buttons: '...don't match filters' (selected) and '...match filters'. The 'Filter options' section has a dropdown menu set to 'All conditions must be met (AND)'. Below this is a table with columns 'Col #', 'Substring or regex', 'Case...', and 'Rege'. The table is currently empty. At the bottom of the table are three buttons: '+ Add ...', 'Edit ...', and 'Remove'. At the bottom of the dialog are three buttons: '{V}', 'OK', and 'Cancel'.

Col #	Substring or regex	Case...	Rege
-------	--------------------	---------	------

**Delete rows that...**

Record deletion mode:

- ...don't match filters - keep only the records that meet your conditions. The remaining records will be deleted.
- ...match filters - delete all records that meet your conditions.

**Filter options****Filter mode:**

Choose the condition combination mode: AND (all listed conditions must be met) or OR (at least one condition must be met).

**Filter list**

- Each condition corresponds to a specific column.
- Substring or regular expression - The condition string. By default, the action checks for the presence of the specified substring in the field value. You can indicate that the action should use this expression as a regular expression. Regular expressions are more flexible for defining conditions but require some skills.
- Case sensitive - yes or no
- Regular expression - yes or no. How to use condition string: as regular expression (yes) or as substring (no)

**Source and Source format**

[Source](#)<sup>[462]</sup> and [Source format](#)<sup>[463]</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>[460]</sup> chapter

**Note:** *This feature is available for Business license only*

**4.32.11 CSV Data Loop**

This action iterates through CSV data records in the loop, considering specified conditions.

## General

The screenshot shows a dialog box titled "CSV Data Loop" with two tabs: "General" and "Source". The "General" tab is active. It contains the following fields and controls:

- "Save row number to variable:" followed by an empty text input field.
- "Save line to a variable:" followed by an empty text input field.
- "Save format:" followed by a dropdown menu showing "Standard comma-separated string".
- "Filter Options" section with a dropdown menu showing "All conditions must be met (AND)".
- A table with columns: "Col #", "Substring or regex", "Case ...", and "Regex". The table is currently empty.
- Buttons: "+ Add...", "Edit...", and "Remove".
- At the bottom: a variable placeholder "{V}", a highlighted "OK" button, and a "Cancel" button.

### Save row number to variable

Specify the variable name to store the row number. Row numbers start from 1. This field can be left empty.

### Save current row to variable

Specify the variable name to store the current record. This field can be left empty.

If you want to save the current record for processing, specify the format:

- Standard comma-separated string (default format)
- JSON array

### Filter options

#### Filter mode:

Choose the condition combination mode: AND (all listed conditions must be met) or OR (at least one condition must be met).

### Filter list

- Each condition corresponds to a specific column.
- Substring or regular expression - The condition string. By default, the action checks for the presence of the specified substring in the field value. You can indicate that the action should use this expression as a regular expression. Regular expressions are more flexible for defining conditions but require some skills.
- Case sensitive - yes or no
- Regular expression - yes or no. How to use condition string: as regular expression (yes) or as substring (no)

### Source and Source format

[Source](#)<sup>462</sup> and [Source format](#)<sup>463</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>460</sup> chapter

**Note:** *This feature is available for Business license only*

#### 4.32.12 CSV Remove Duplicates

This action removes duplicate rows that contain the same data.



## General

CSV Remove Duplicates

General Source

Case sensitive

Save number of deleted rows to variable:

Duplicate removal mode

Complete match (default)

Match on specific columns

Specify column numbers from 1 separated by commas:

{V} OK Cancel

### Case sensitive

Enable this checkbox for case-sensitive comparison.

### Save number of deleted rows to variable

You can save the number of deleted rows to a variable if needed. This field can be left empty.

### Duplicate removal mode

- *Complete match* - the default option. Only rows that are exact duplicates are removed.
- *Match on specific columns* - rows that are duplicates only in the specified columns are removed. You need to list the column numbers separated by commas for control. Column numbers start from 1. If you leave the field with numbers empty, the action will check the entire row (complete match). The action will ignore incorrect column numbers.

### Source and Source format

[Source](#)<sup>[462]</sup> and [Source format](#)<sup>[463]</sup> tabs are identical for most SCV actions. Read description in "[CSV Get Data Information](#)"<sup>[460]</sup> chapter

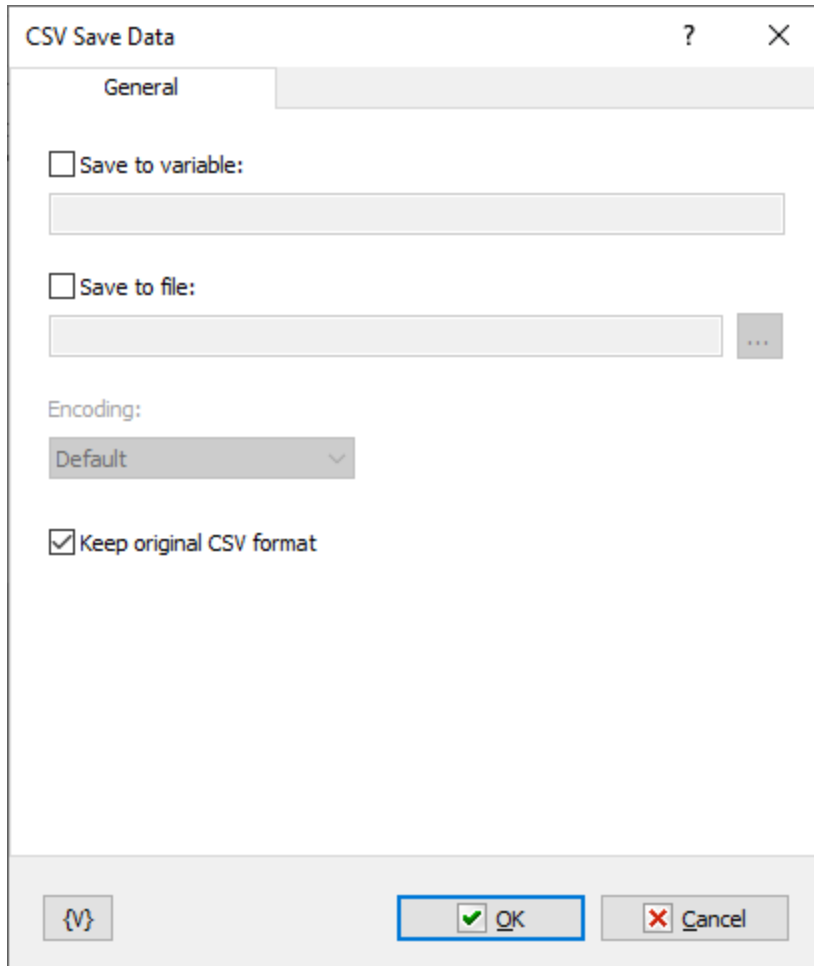
**Note:** This feature is available for Business license only

### 4.32.13 CSV Save Data

This action saves CSV text to a variable or file.

**Note** that CSV data array is not automatically saved to a file or variable.

#### General



The screenshot shows a dialog box titled "CSV Save Data" with a "General" tab. The dialog has a title bar with a question mark and a close button. The main area contains the following options:

- Save to variable:  
A text input field for specifying a variable name.
- Save to file:  
A text input field for specifying a file name, followed by a browse button (three dots).
- Encoding:  
A dropdown menu currently set to "Default".
- Keep original CSV format

At the bottom of the dialog, there are three buttons: a help button (question mark in a box), an "OK" button (with a green checkmark icon), and a "Cancel" button (with a red X icon).

#### Save to variable

Turn this switch on and specify the variable name to save the CSV text.

#### Save to file

Turn this switch on and specify the file name to save the CSV text. Also, specify the text encoding if needed.

Note that UTF-8 is a universal format for any language.

#### Keep original format

By default, the action suggests keeping the CSV text formatting (delimiter, quotes, etc.). But you can choose a different format. Simply turn this switch off and specify the new format on the CSV Format tab.

## CSV format

The screenshot shows a dialog box titled "CSV Save Data" with a "CSV Format" tab selected. The "Delimiter" dropdown is set to "Semicolon - ;". The "Character" field is empty. The "Quote character" dropdown is set to "Double quote - \"". The "Strict delimiter (space is not a delimiter)" checkbox is checked. There are two buttons: "Reset to standard format" and "Reset to Excel format". At the bottom, there are three buttons: a button with a curly brace icon, a highlighted "OK" button, and a "Cancel" button.

Specify the format of the CSV data: delimiter, quote character.

Note that if the "Strict delimiter" switch is not enabled, spaces are also considered delimiters. And all fields containing spaces must be enclosed in quotes.

### Reset to standard format

Sets the standard parameters:

- Delimiter: Comma (,)
- Quotes: Double quotes (")
- Non-strict delimiter (spaces are also considered delimiters)

### Reset to Excel format

Sets the parameters used by MS Excel by default:

- Delimiter: Semicolon (;)
- Quotes: Double quotes (")
- Strict delimiter.

**Note:** *This feature is available for Business license only*

#### 4.32.14 SQL Query to CSV File

This action is used to create a CSV file based on an SQL query from a database. RoboTask allows you to export data from SQL to CSV using other actions, but the algorithm would be more cumbersome. Using a single action is much simpler.

##### Dataset

The screenshot shows a dialog box titled "SQL Query to CSV File". It has four tabs: "Dataset", "SQL parameters", "Target CSV file", and "CSV Format". The "Dataset" tab is selected. The dialog contains the following fields and controls:

- Connection String:** A text input field with a dropdown arrow to its right.
- Override connection string parameters**
- User ID:** A text input field.
- Password:** A text input field.
- Show password:** A button located below the password field.
- SQL Query:** A large text area with vertical and horizontal scrollbars.
- Buttons:** At the bottom, there are four buttons: a button with a curly brace icon, "Test SQL", "OK" (highlighted with a blue border), and "Cancel".

##### Connection string

The database connection string. You can either manually write the connection string or generate it using the system dialog.

##### Overwrite connection string parameters

You can specify authentication parameters (username and password) directly in the connection string. If you want to override these parameters, enable this checkbox and fill in the "User ID" and "Password" fields.

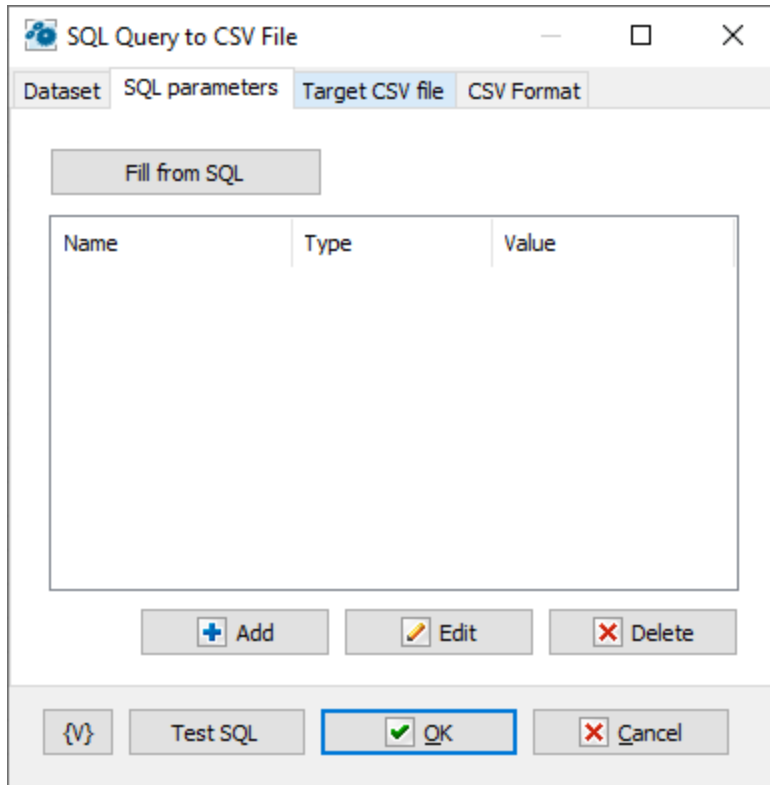
### Show password

Show the password in plain text for ease of editing, as the entered string may contain RoboTask variables.

### SQL query

The SQL query text. Please note that the syntax may vary slightly depending on the type of database.

### SQL Paramaters



Sometimes it's more convenient to use query parameters instead of generating a query using variables. For example, when the value of a variable contains quotes or other special characters. If you use variables (or macro substitution) when forming a SQL query, the presence of such characters can "break" the syntax of the query and you will get a run-time error.

Parameters SQL are designated in the inquiry by means of colon (:) before the parameter name.

For example:

```
Select * from employee where hiredate > :dt and salary >= :amount
```

**:dt** and **:amount** are parameters of the query.

Parameters can be several types:

- *String* – any string
- *Integer* – an integer number

- *Float* – a real number or the number with a floating dot.
- *Boolean* – a logical value True or False.
- *DateTime* – a date and time. The value of such parameter must be defined in current system format of date and time (see the system regional settings). Notice that using of date-time can have its own characteristics for different databases. For example, it is better specify such parameters for the MySQL as a string parameter in the format *YYYY-MM-DD* or *YYYY-MM-DD hh:nn:ss*.

**Fill from SQL**

Fill the list of parameters from the current query. If a parameter is already in the list, this parameter will be without change. The function fills only the missing parameters in the list.

**Add**

Add a parameter manually. You will be proposed to type or select from the list the parameter name, select the parameter type and input an expression as the parameter value.

Using of variables in a parameter name and in its value is allowed. The RoboTask will substitute variable values instead of variables before performing the SQL.

**Edit**

Edit the properties of the selected parameter.

**Delete**

Delete the selected parameter from the list.

If you specify the parameter which is absent in the query, the action ignores this parameter when performing and writes the proper warning in the log of the task.

## Target CSV file

The screenshot shows a dialog box titled "SQL Query to CSV File" with four tabs: "Dataset", "SQL parameters", "Target CSV file" (selected), and "CSV Format". The "Target CSV file" tab contains the following controls:

- File name:** A text input field with a browse button (three dots) to its right.
- Encoding:** A dropdown menu currently set to "Default".
- Use field names
- With memo fields (as comma separated string)
- Limit field length to:
- Use custom date-time format

At the bottom of the dialog are four buttons: a brace icon, "Test SQL", "OK" (highlighted with a blue border), and "Cancel".

### File name

The name of the CSV file.

### Encoding

Choose the desired file encoding. The following popular encodings are available for selection:

- *Default* – Use the system encoding.
- *UTF-7*
- *UTF-8*
- *Unicode*
- *ASCII*
- *ANSI*

### Use field names

Enable this checkbox if you want to save column headers from the SQL query.

### With memo fields

If you want to save MEMO fields in the CSV file, enable this checkbox. The MEMO field text will be saved as a comma-separated string. Please note that it doesn't make sense to save long texts in CSV format, as CSV format is not suitable for this purpose.

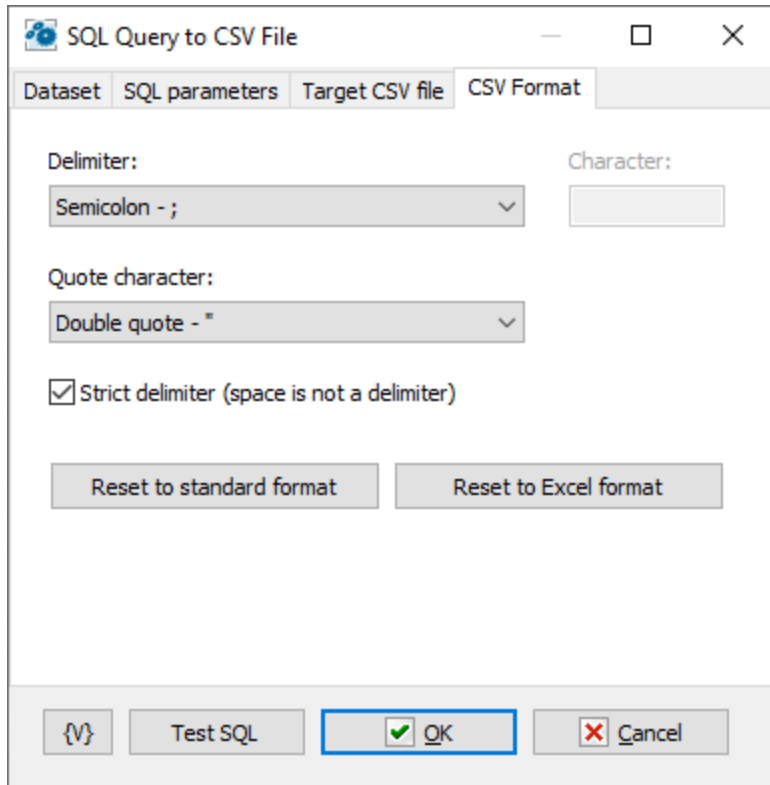
### Limit field length to

Limit the field length to the specified size. The default is 100 characters. You can change this value or disable the limitation.

### Use custom date-time format

This action writes the DateTime field values in the current system format. You can specify any format that suits your needs. The default is the universal format: yyyy-mm-dd hh:nn:ss.

## CSV format



Specify the format of the CSV data: delimiter, quote character.

Note that if the "Strict delimiter" switch is not enabled, spaces are also considered delimiters. And all fields containing spaces must be enclosed in quotes.

### Reset to standard format

Sets the standard parameters:

- Delimiter: Comma (,)
- Quotes: Double quotes (")
- Non-strict delimiter (spaces are also considered delimiters)

### Reset to Excel format

Sets the parameters used by MS Excel by default:



- Delimiter: Semicolon (;)
- Quotes: Double quotes (")
- Strict delimiter.

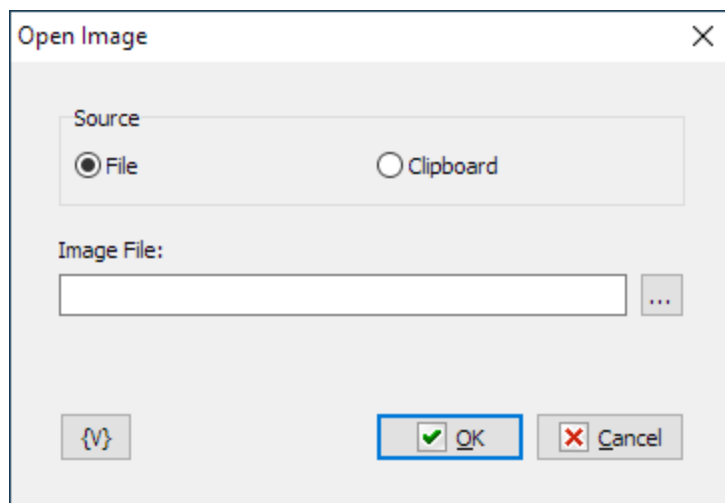
## 4.33 Graphics

This set of actions is intended for batch processing of images. In order to process the image in any way, you need to first [open the image](#)<sup>486</sup>, process it, and save this image. If you attempt to process the image without opening it, you will get an execution error message.

### 4.33.1 Open Image

All image processing operations should begin with the opening of an image.

This action allows opening an image from file or clipboard. If the source does not contain an image, you will get an execution error message.



#### Source

You can choose the image source:

- File
- Clipboard

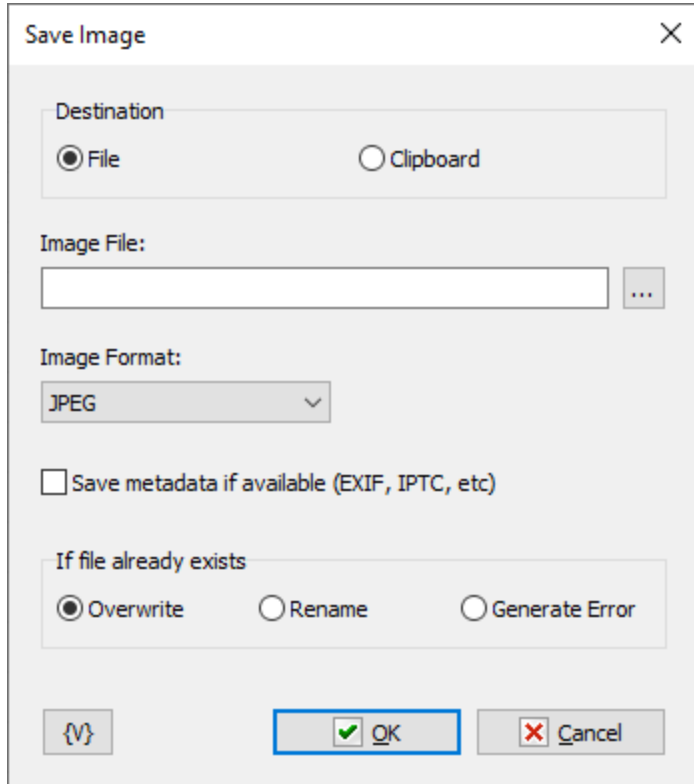
#### ImageFile

Specify a file name when a file has been chosen as the image source.

If the file or clipboard does not contain an image, you will get an execution error message.

### 4.33.2 Save Image

This action saves the current image to file or system clipboard.



#### Destination

Choose the file save destination:

- File
- Clipboard

#### Image File

Enter the file name if you choose to save to file.

#### Image Format

When saving to file, choose the file format. When saving to clipboard, this parameter is omitted.

#### Save Metadata if available.

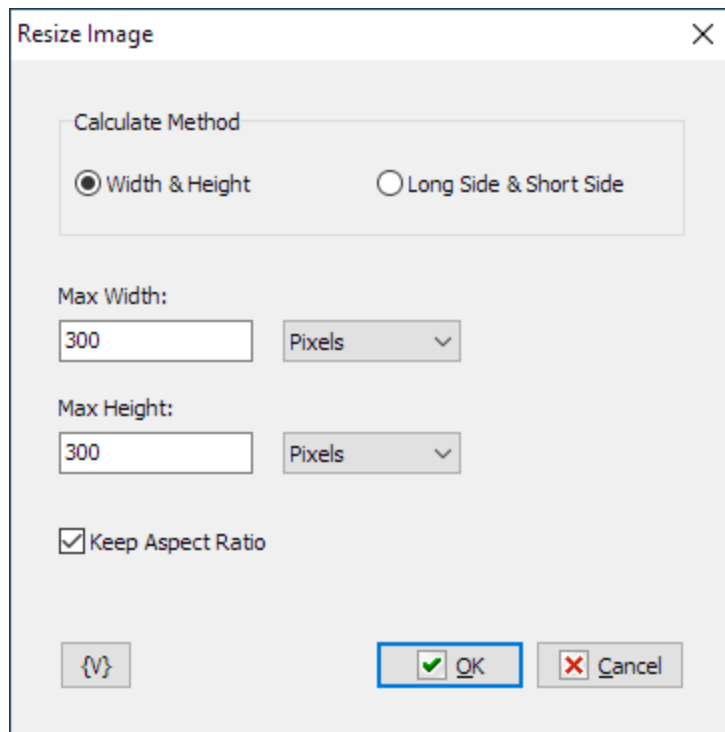
Select this checkbox if you need to save metadata (additional image info). Metadata (if any) are loaded when the image is opened. Please note that an image in clipboard cannot contain metadata.

#### If file already exists

The action if file already exists.

### 4.33.3 Resize Image

This action changes the image dimensions.



#### Calculate method

Two image size recalculation methods are available:

- Width & Height – in this mode, the horizontal size (Width) and the vertical size (Height) are specified.
- Long side & Short Side – the long side and the short side are used in this case, since the image can have either album or portrait orientation.

#### Width & Height

New image dimensions. The meaning of these parameters changes depending on the Calculate Method and Keep Aspect Ratio parameter:

- Width & Height
- Max Width & Max Height
- Long side & Short Side
- Max Long side & Max Short Side

The dimensions can be specified in pixels and in percentage points.

#### Keep Aspect ratio

Enable this parameter if you need to keep the image aspect ratio.

#### 4.33.4 Crop Image

This action crops the image according to the parameters specified.

Crop Image

Left:  
0 Pixels

Top:  
0 Pixels

Right:  
0 Pixels

Bottom:  
0 Pixels

HINT: Negative values meaning that they will be deduced from actual image width/height.

{V} OK Cancel

#### **Left, Top, Right, Bottom**

Crop boundaries. The boundaries can be specified in both pixels and percentage points.

If a positive number is specified, it represents the absolute coordinates of the crop boundaries.

If a negative number is specified, this means a space from the relevant width or height of the image.

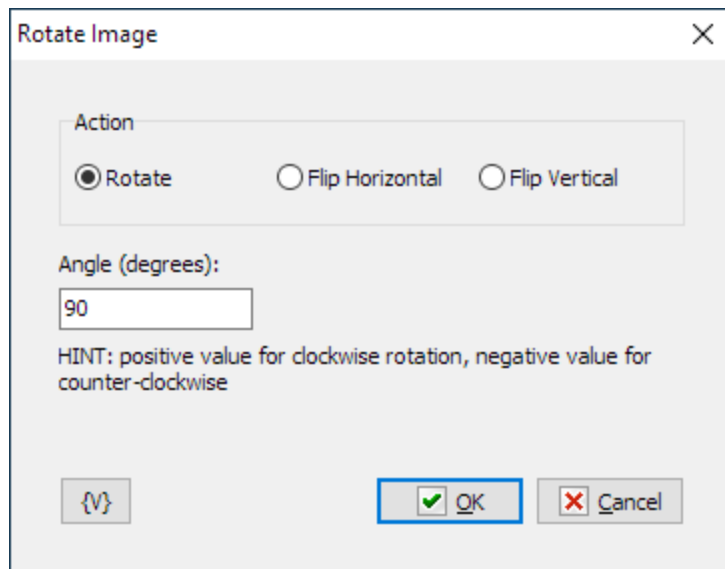
For example if you want to cut off 10 pixels from each side you have to specify such parameters:

```
Left   = 10px
Top    = 10px
Right  = -10px
Bottom = -10px
```

The use of variables is allowed.

### 4.33.5 Rotate Image

This action rotates the image by the angle specified. This action also mirrors the image about the vertical or horizontal axis.



#### Action

Select an action to be performed on the image:

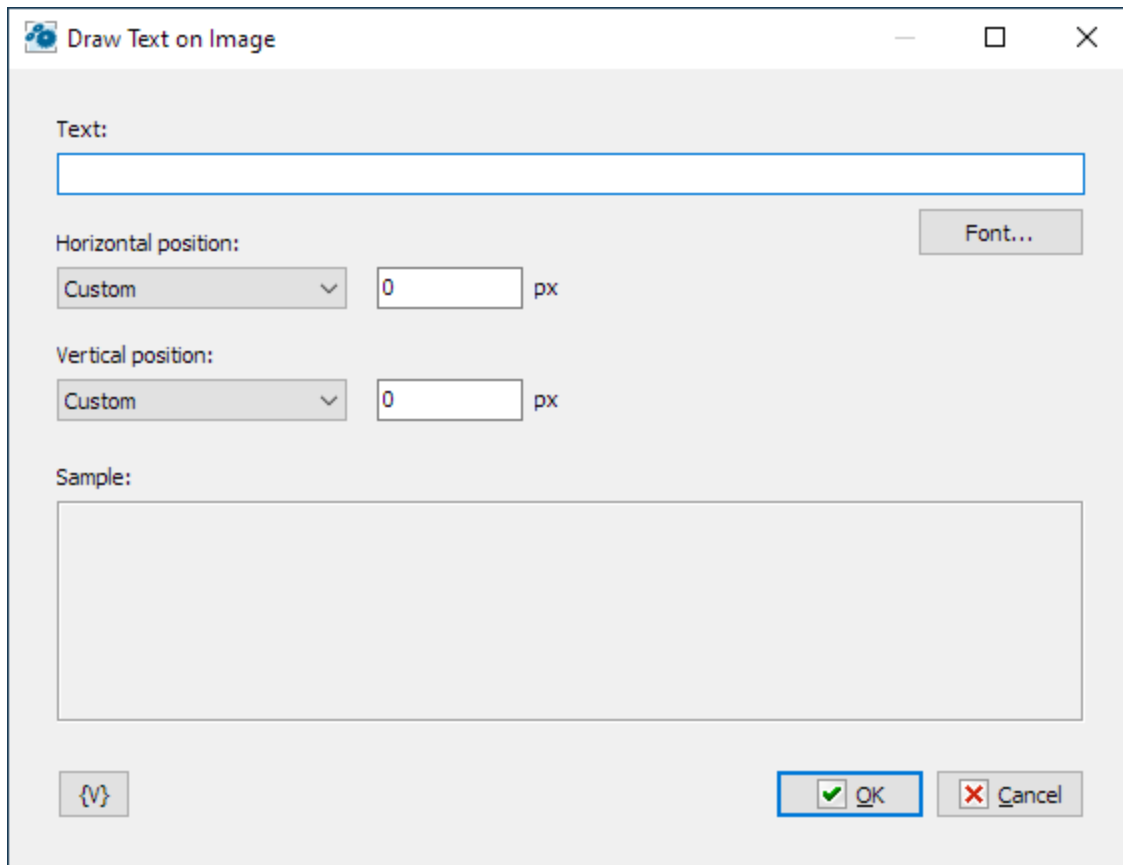
- Rotate
- Flip Horizontal
- Flip Vertical

#### Angle (degrees)

Enter an angle in degrees by which you want to rotate the image. A positive number means clockwise rotation. A negative number means counterclockwise rotation. This parameter is omitted during mirroring.

### 4.33.6 Draw Text on Image

This action allows applying text to the image at the specified coordinates.

**Text**

Type the text that you want apply to the image.

**Font**

You can choose a font, font style, and size.

**Horizontal position**

Choose one of the options

- Left
- Center
- Right
- Custom – specify the position in pixels in this case

**Vertical position**

Choose one of the options

- Top
- Center
- Bottom
- Custom – specify the position in pixels in this case

### 4.33.7 Draw Image on Image

This action places another image on top of the current image. For example: a logo or watermark.

Draw Image

Image File:

Horizontal position:

Custom 0 px

Vertical position:

Custom 0 px

Opacity (0-100):

100 %

{V} OK Cancel

#### Image file

Enter or choose the file name of the image that you want to insert.

#### Horizontal position

Choose one of the options

- Left
- Center
- Right
- Custom – specify the position of the left edge in pixels in this case

#### Vertical position

Choose one of the options

- Top
- Center
- Bottom
- Custom – specify the position of the top edge in pixels in this case

**Opacity**

Specify the image opacity in percentage points.

100 – the image is fully non-transparent

0 – fully transparent.

**4.33.8 Add Border to Image**

This action adds borders to an image. The border width and color can be specified in the parameters.

The screenshot shows a dialog box titled "Add Border to Image" with a close button (X) in the top right corner. The dialog contains the following controls:

- Left:** A text input field containing the number "3" and a dropdown menu labeled "Pixels".
- Top:** A text input field containing the number "0" and a dropdown menu labeled "Pixels".
- Right:** A text input field containing the number "0" and a dropdown menu labeled "Pixels".
- Bottom:** A text input field containing the number "0" and a dropdown menu labeled "Pixels".
- Color:** A dropdown menu showing a black square icon and the text "Black".
- Buttons:** A reset button with a curly brace icon, an "OK" button with a green checkmark, and a "Cancel" button with a red X.

**Top, Left, Right, Bottom**

Border width at the top, on the right, on the left, and at the bottom. The border width can be specified either in pixels or as a percentage of the image size.

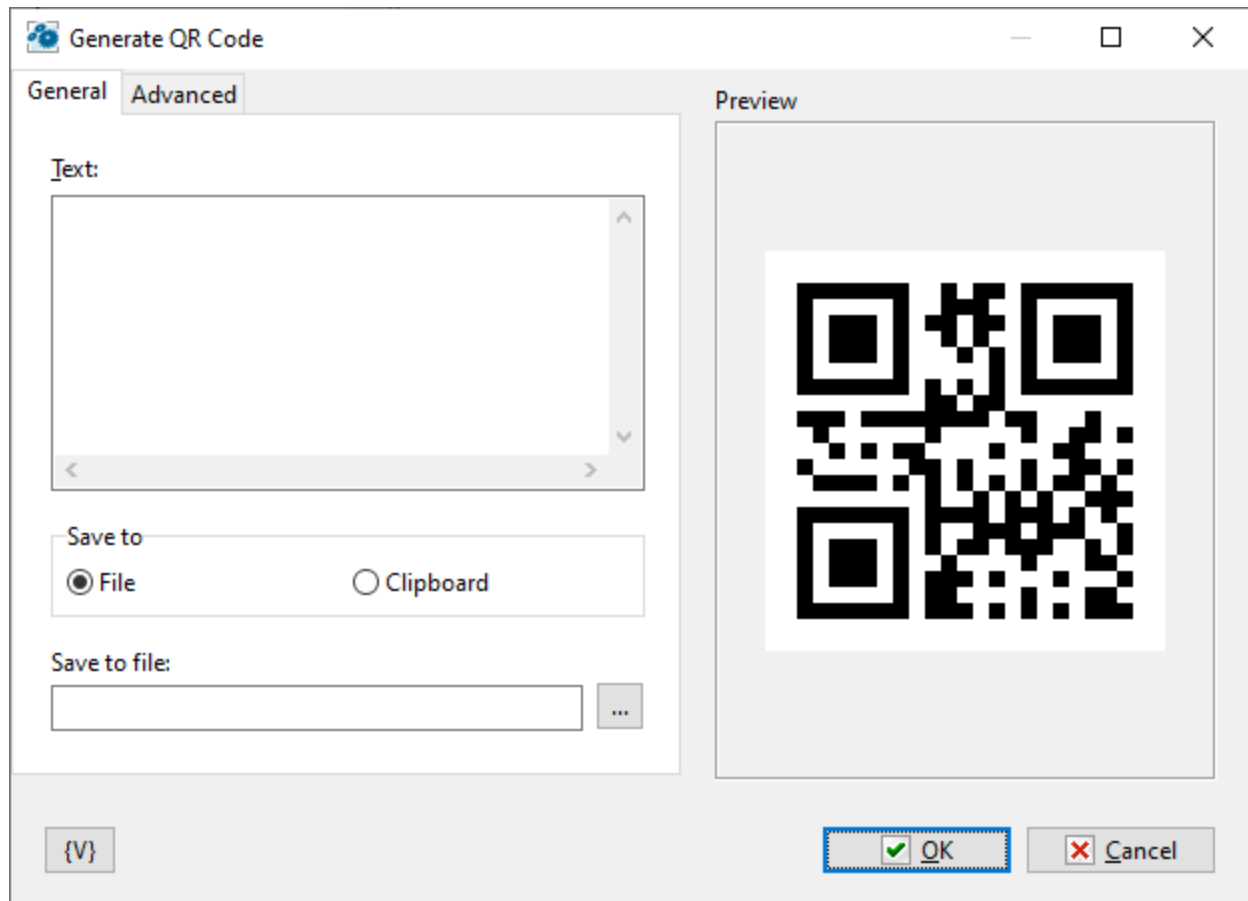
**Color**

Choose the field fill color.



### 4.33.9 Generate QR Code

The action allows you create QR code and save it as image to a file or clipboard

**Text**

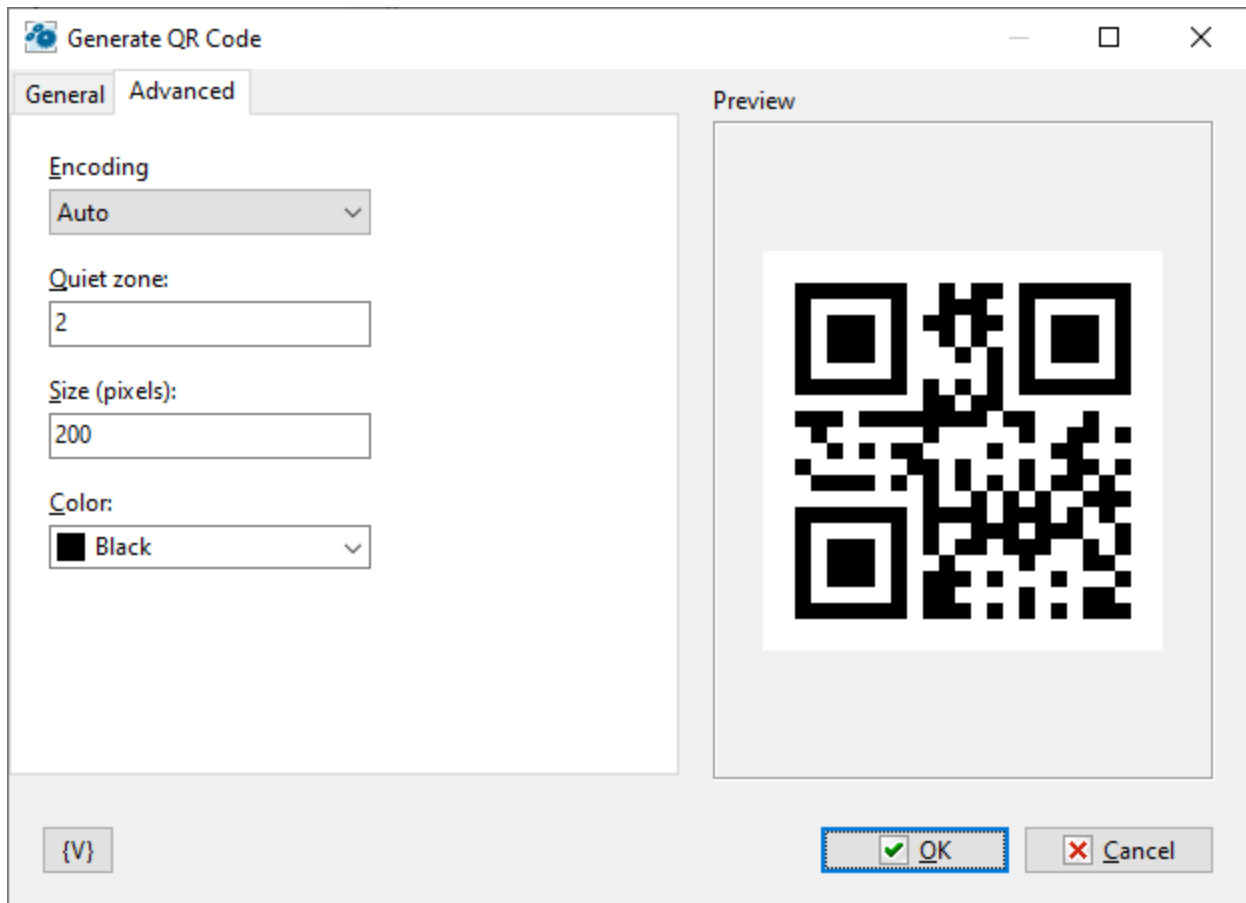
Enter the text you want to encode in the QR

**Save to**

Choose the target: file or system clipboard.

**Save to file**

Enter the file name if you want to save image to file



### Encoding

Choose character encoding. If it's hard for you to choose, choose *Auto* or *UTF-8 with BOM*. This is suitable for most cases.

### Quiet zone

The size of border of QR code

### Size (pixels)

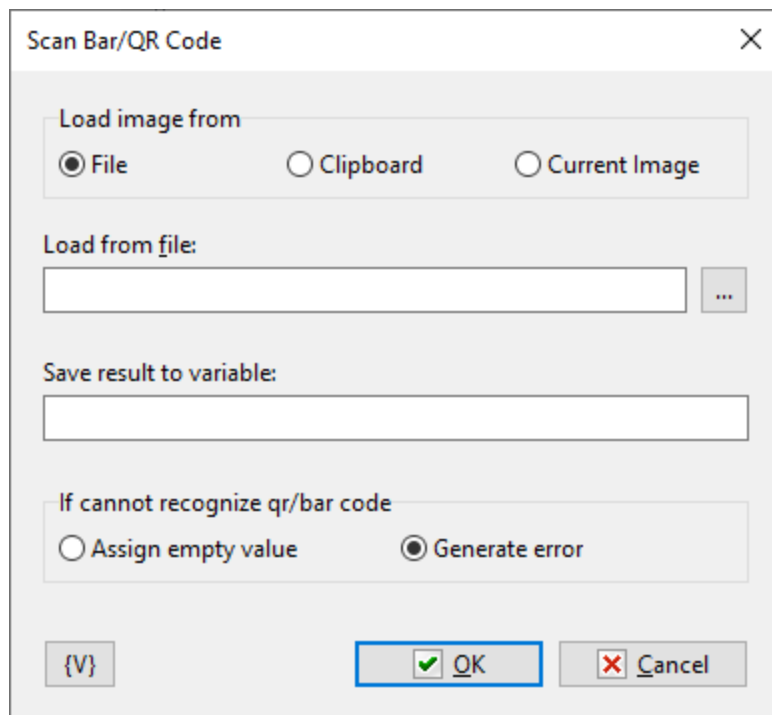
Size of the result image in pixels

### Color

Choose the color of QR blocks.

## 4.33.10 Scan QR/Bar Code

The action reads QR or Bar code from the image and saves result to variable



### Load image from

There are three options

- File - specify file name below
- Clipboard - load image from system clipboard.
- Current image - use the the image opened in the action [Open image](#)<sup>486</sup>

If source does not contain an image then action generates an error.

### Load from file

Specify file name to load image from file

### Save result to variable

Save the result as string to specified variable.

### If cannot recognize QR/Bar code

There are two options

- Assign empty value to variable
- Generate error

## 4.34 WebSockets

WebSocket is a TCP communication protocol designed to exchange messages between applications in real time. Most browsers support the protocol, including Google Chrome,

Microsoft Edge, Internet Explorer, Firefox, Safari and Opera. [You can read more about this protocol here.](#)

This plugin enables RoboTask communication with other applications that use WebSocket protocol or with other RoboTask instances.

The WebSocket protocol mechanism requires a permanent client-server connection and the capability to exchange messages asynchronously. The source of the message can be either the client or the server. If the client loses connection for some reason, by default it tries to re-establish the connection.

To get started, create the required set of Websocket Servers and Clients. All actions and triggers of this plugin work with this set of servers and clients in one way or another. Their combination and parameters depend on your requirements.

To define WS Servers and WS Clients, use [WS Connection Manager](#)<sup>497</sup>.

**Note:** *This feature is available for Business license only*

#### 4.34.1 WS Connection Manager

You can open WS Connection Manager via the plugin settings (menu *Options->Settings->Plugins tab*) or from the editor of WebSocket actions or WebSocket triggers.

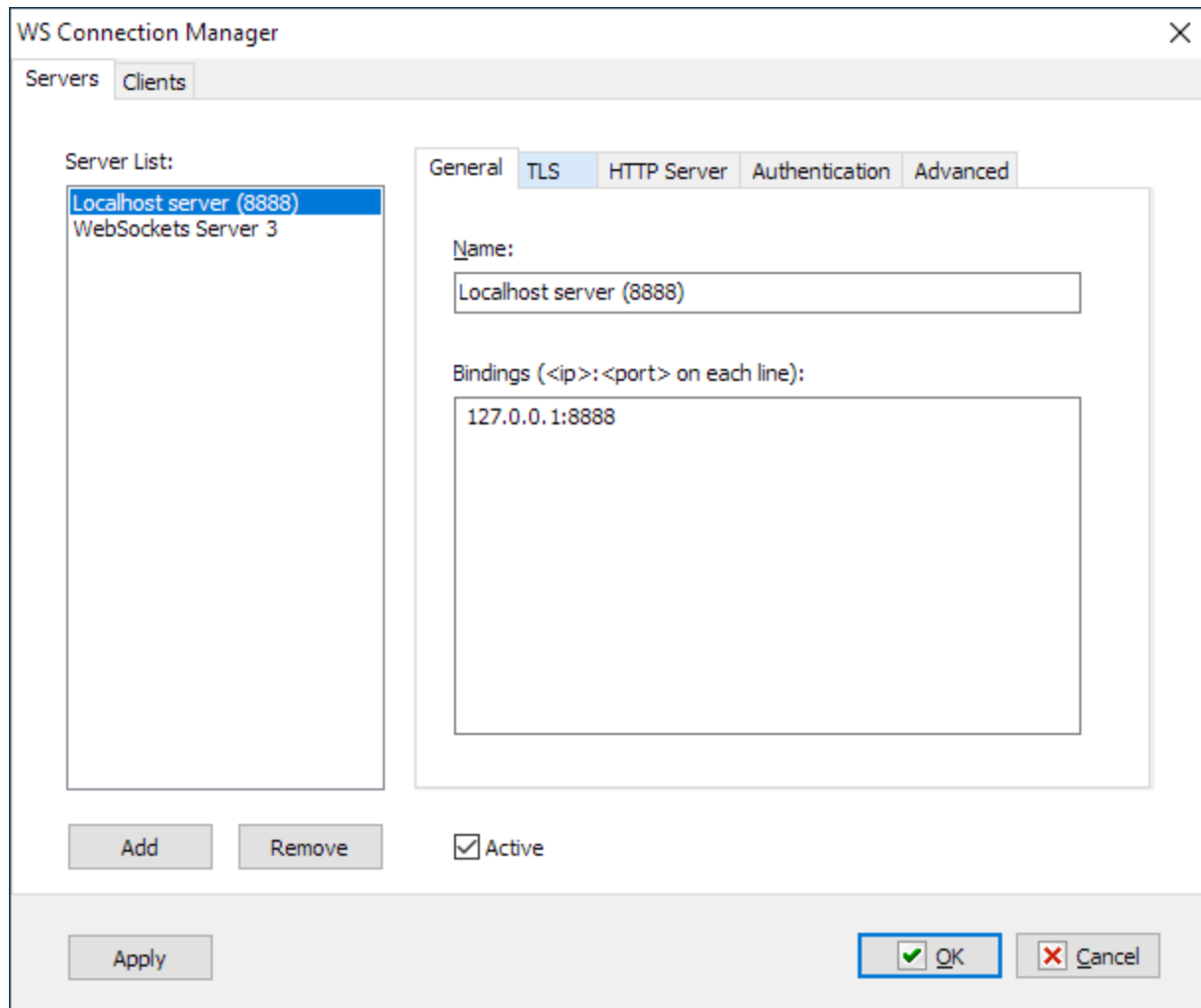
Here you can define the required set of servers and clients. The editor is divided into two tabs, respectively:

- Servers
- Clients

**Note:** *This feature is available for Business license only*

### Servers

You can create a list of required WebSocket Servers and define required parameters for each server separately.



### Active

Tick this checkbox if you want this server to be active when RoboTask is loaded and when parameters are saved.

### General

#### Name

Server name. When selecting a server in the actions and triggers editor, you can specify any string for visualization.

#### Bindings

Binding to IP address and port. Binding to multiple IP addresses and ports is supported. The binding is specified in **<IP\_address>:<port>** format

If you specify **0.0.0.0** as the address, the server will bind to all available computer addresses of the specified port.

If you specify local address of **127.0.0.1**, the server will only accept local connections within the local computer

## TLS

TLS (transport layer security) – traffic encryption mode.

If you access WS Server via the wide area network, it is recommended to use traffic encryption. To do this, you need to specify the certificate and the private key for encryption. You can use paid or free certificates ([for example, here](#)).

You can also create the so-called "self-signed" certificates. You can make self-signed certificates using *OpenSSL* or *Powershell* package. Read how to do it [here](#) or [here](#)

If you intend to use Websocket locally on your computer or in the office LAN, traffic encryption is not necessary.

If you specify that the server needs to encrypt traffic, you must also specify the encryption certificate and the private key of the certificate. These are text files in PEM format. They can have any extensions, but as a rule, PEM or CRT extensions are used for the certificate. PEM or KEY extensions are used for the key.

## HTTP Server

Websocket server can act as a simple HTTP server.

### Empty Response

A default option. WS Server sends an empty response to the HTTP request.

### HTML page

WS Server returns the specified HTML page when it receives the HTTP request.

### Serve files from folder

WS Server acts as a simple HTTP server built on static HTML pages, CSS files and image files  
Specify Content folder and Default document

## Authentication

### Allows anonymous connections

If you enabled this option, no user rights check is carried out. The server accepts connections without authorization

### HTTP basic authorization

Standard authorization using a username or password.

### URL Authentication

This authorization method is mainly used in scripts. In this case, the username and password are passed via the query string:

```
wss://<host>:<port>/auth/<username>/<password>
```

---

When configuring WS Client in RoboTask, you need to specify the following string in the Parameters field

```
/auth/<username>/<password>
```

### User rights

Defining a list of users for authorization

- **System Administrators** – in this case, the system check of user rights is used. Authentication is considered successful, if the authentication parameters are correct and the user belongs to the Administrators system group.
- **Custom User list** – in this case, you need to define the user as text. Every string should look like:

```
Username=Password
```

### Advanced

#### ID

WS server ID. By default, *WS\_SERVER\_N* string, where N is a number (ordinal number), is assigned as the identifier.

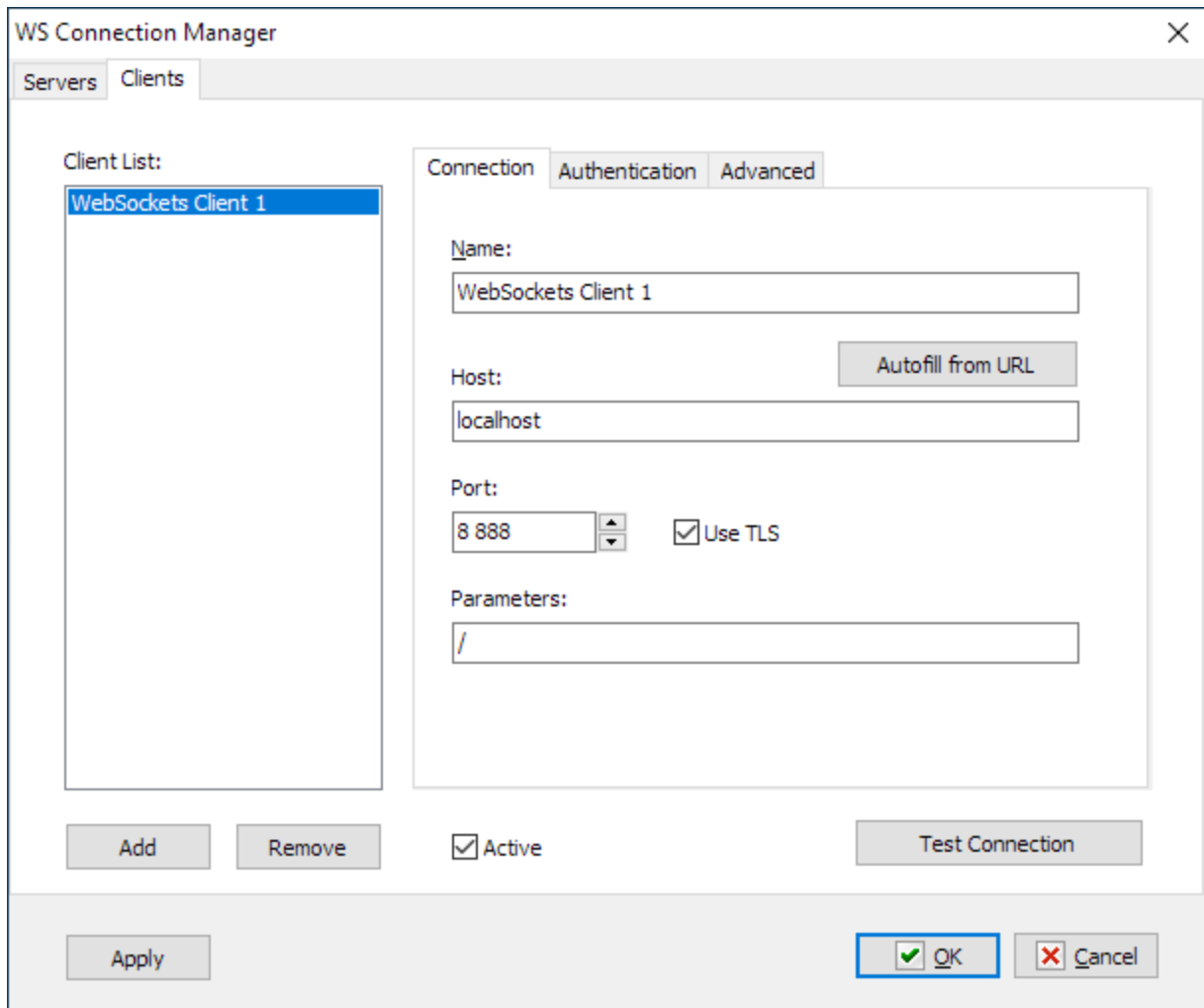
The identifier must consist of letters, numbers and an underscore and must begin with a letter or an underscore. Other characters are not allowed.

The identifier is case insensitive.

When editing a network task you must know the WebSocket server ID and specify it explicitly, because you won't be able to choose the desired server from the list.

### Clients

Here you can create a list of required WebSocket clients, and define required parameters for each Client separately.



### Active

Tick this checkbox, if you want this client to be active when RoboTask is loaded and when parameters are saved.

### Test connection

When clicking this button, the client settings are saved and an attempt is made to connect with the current client parameters. After the operation is complete, you will see a message about the result of the connection attempt.

### Connection

General parameters of client connection

#### Name

The client name for visualization. Enter any name that is easy to understand.

#### Autofill from URL



---

This function allows to fill in the joining parameters automatically, if you know full URL of the WebSocket server.

URL starts with `ws://` or `wss://` prefix

For example:

```
wss://myhost.com:12345/auth/user/psw
ws://myhost.com/report?info=common&date=20201201
```

### Host

WebSocket server name or IP address

### Port

WebSocket server port. By default, the ports used are the same as those of HTTP server:

- 80 – without using TLS/SSL
- 443 – when using TLS/SSL.

### Use TLS

Tick this checkbox, if you want to use encryption. Please note that the encryption mode is determined by the WS server. If the client's encryption mode does not match the server setting, the client will not be able to connect.

### Parameters

A string of the path and URL parameters. For example:

```
/auth/user/psw
/report?info=common&date=20201201
```

## Authentication

### HTTP Basic Authentication

Tick this checkbox, if you need basic authorization

### Username

A user name.

### Password

An explicit password.

## Advanced

### ID

WS Client ID. By default, `WS_CLIENT_N` string, where N is a number, is assigned as the identifier.

The identifier must consist of letters, numbers and an underscore and must begin with a letter or an underscore. Other characters are not allowed.

The identifier is case insensitive.

When editing a network task you must know the WebSocket client ID and specify it explicitly, as you won't be able to choose the desired server from the list.

#### **Ping server every X seconds to maintain connection alive**

If you want to regularly check connectivity even without performing any action, then tick this checkbox. It is disabled by default.

Some applications with a built-in WebSocket server can forcibly disconnect when there is no client's activity. In this case, ping will help keep the connection active.

#### **Interval (ping)**

Specify the check interval.

#### **Auto reconnect to server if connection is lost**

Tick this checkbox, if it is necessary to re-establish the lost connection. It is disabled by default.

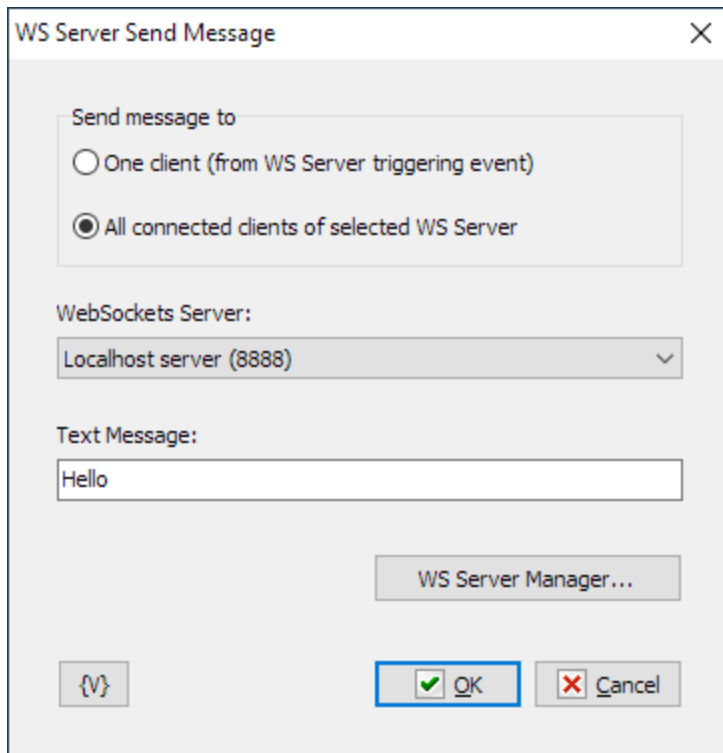
#### **Interval (reconnect)**

Specify the interval of connection attempts

### **4.34.2 WebSocket Server Send Message**

Action enables to send a message to active clients.

**Note:** *This feature is available for Business license only*



WS Server Send Message

Send message to

One client (from WS Server triggering event)

All connected clients of selected WS Server

WebSockets Server:

Localhost server (8888)

Text Message:

Hello

WS Server Manager...

{V}

OK Cancel

### Send message to

Two options are possible

- One client – this option works, if the WebSocket Server On Message trigger is connected to the task. When this trigger is activated, the specified WS server sends a message only to the client, from which it received the message that activated the trigger.
- All connected clients – the specified WS server sends a message to all connected clients.

### WebSocket server.

Choose the required WebSocket server from the list. Please note that you cannot choose the server when editing a network task. Instead, you must enter the server ID. You must know the server ID in advance (see *WS Connection manager->Servers->Advanced*). If an invalid ID is specified, an "Invalid WS Server ID" error is returned when executing.

### Text message

Enter the text of the message you want to send. You can use variables.

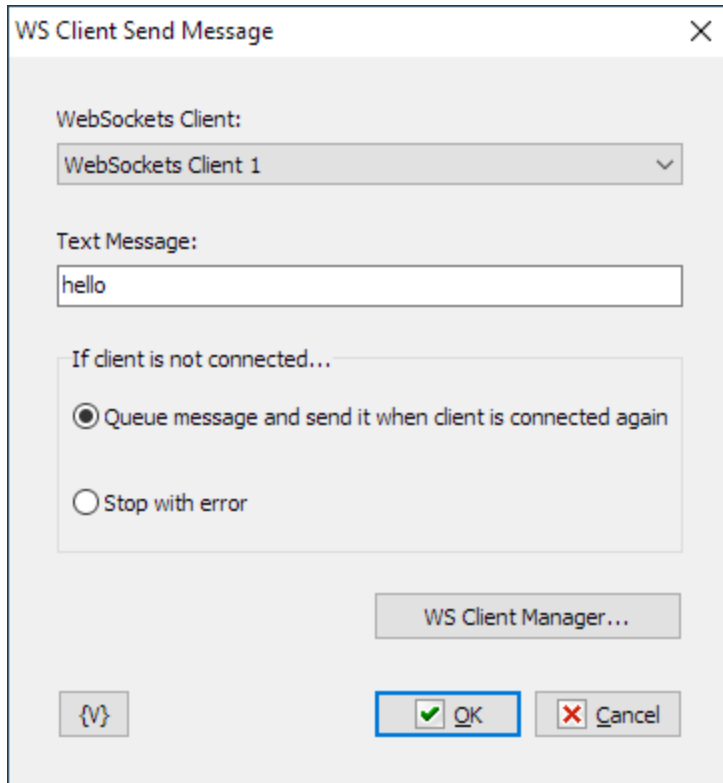
### WS Server Manager

You can view or edit lists of registered Websocket servers and clients (see [WS Connection Manager](#)<sup>497</sup>).

### 4.34.3 WebSocket Client Send Message

The action sends a text message via the specified WS client to the connected WebSocket server

**Note:** This feature is available for Business license only



WS Client Send Message

WebSockets Client:  
WebSockets Client 1

Text Message:  
hello

If client is not connected...

Queue message and send it when client is connected again

Stop with error

WS Client Manager...

{V} OK Cancel

#### WebSocket Client

Choose the WebSocket client from the list. If you edit a network task, the list is not available and you have to enter the client ID manually (see *WS Connection manager->Clients->Advanced*). If the client ID is incorrect, the action will get an "Invalid WS Client ID" run-time error.

#### Text message

The text message you want to send

#### If client is not connected

If the specified client is not connected at the time the action is called, two execution options are offered:

- Queue message and send it when client is connected again;
- Stop with error

#### WS Client Manager

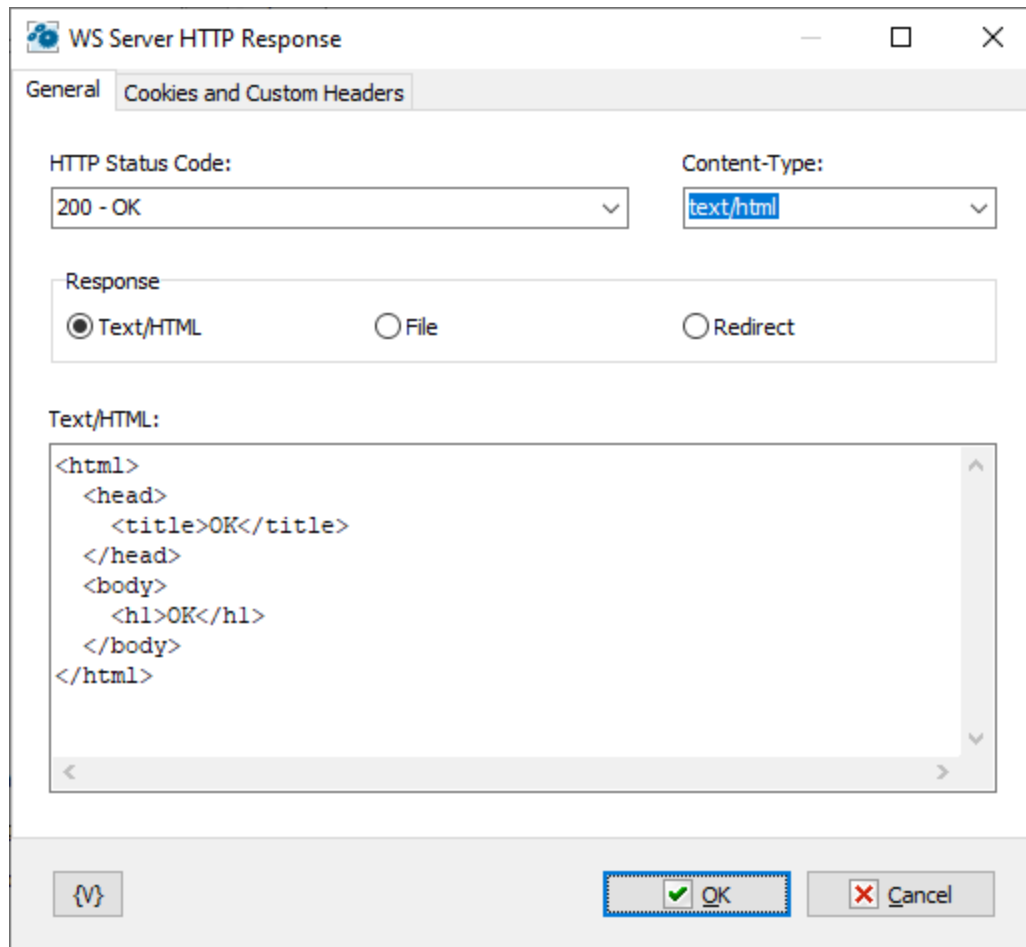
---

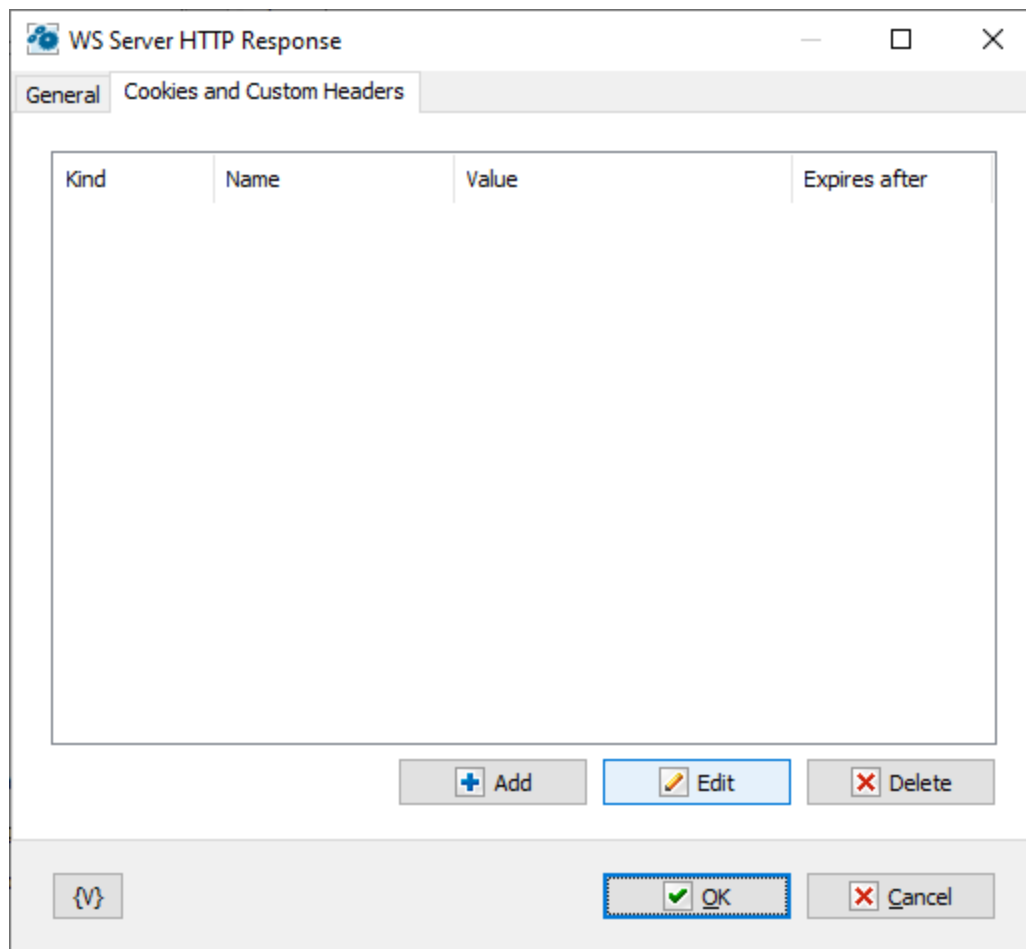
You can view or edit lists of registered WebSocket servers and clients (see [WS Connection Manager](#)<sup>[497]</sup>).

#### 4.34.4 WebSocket Server HTTP Response

This action works in conjunction with the [WebSocket Server On HTTP Request](#)<sup>[553]</sup> trigger. It doesn't make sense to use an action without this trigger, as having received the request the trigger starts the task and passes the request parameters to it for processing. Thus, the action enables building a simple web server.

**Note:** *This feature is available for Business license only*





### HTTP Status code

Choose from the list of standard HTTP statuses.

### Content type

Choose a content type from the list or enter your own option.

### Response

There are three options to choose from:

- **Text/HTML** – in this case, enter the response text as the plain text or HTML text
- **File** – choose the file to send to the client
- **Redirect** – redirect the request to another URL

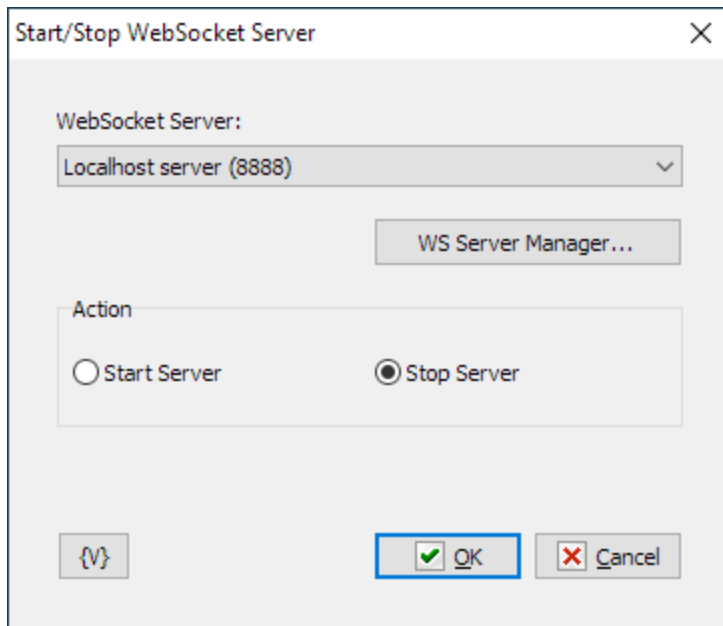
### Cookies and Custom Headers

You can set your own cookies and custom fields in HTTP response.

#### 4.34.5 Start/Stop WebSocket Server

This action allows to stop or start the WebSocket server. Please note that this action affects the global configuration of the active WebSocket of the server (see [WS Connection manager](#)<sup>[497]</sup>).

**Note:** *This feature is available for Business license only*



##### **WebSocket server.**

Choose the required Websocket server from the list. Please note that you cannot choose the server when editing a network task. Instead, you must enter the server ID. You must know the server ID in advance (see *WS Connection manager->Servers->Advanced*). If an invalid ID is specified, an "Invalid WS Server ID" error is returned when executing.

##### **WS Server Manager**

You can view or edit lists of registered Websocket servers and clients (see [WS Connection Manager](#)<sup>[497]</sup>).

##### **Action**

Choose the action:

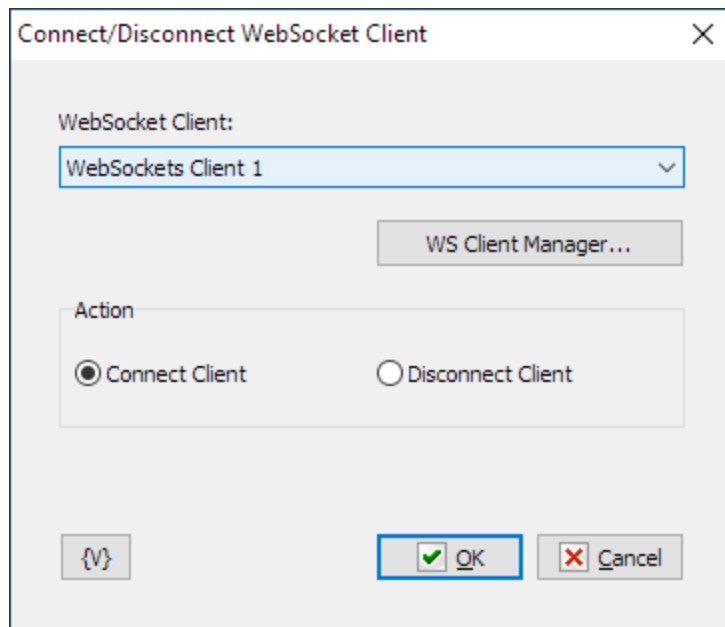
- Start Server
- Stop server

#### 4.34.6 Connect/Disconnect WebSocket Client

This action allows to enable or disable the WebSocket client. Please note that this action affects the global configuration of the active WebSocket of the client (see *WS Connection manager*).



**Note:** This feature is available for Business license only



### WebSocket Client

Choose the Websocket client from the list. If you edit a network task, the list is not available and you have to enter the client ID manually (see *WS Connection manager->Clients->Advanced*). If the client ID is incorrect, the action will get an "Invalid WS Client ID" run-time error.

### WS Client Manager

You can view or edit lists of registered Websocket servers and clients (see [WS Connection Manager](#)<sup>[497]</sup>).

### Action

Choose the action:

- Connect Client
- Disconnect Client

## 4.35 Console Application

This plugin contains actions to automate interactive console applications. For example, *cmd.exe*, *powershell.exe*, etc.

Please note that if it is possible to write a script (or batch file) and execute it from the command line, this option is preferable. CMD and POWERSHELL also allow you to execute a previously prepared script from the command line. Look at [Run Console Application](#)<sup>[80]</sup> action

In addition, some console utilities don't allow I/O to be intercepted in interactive mode, which makes such automation impossible.

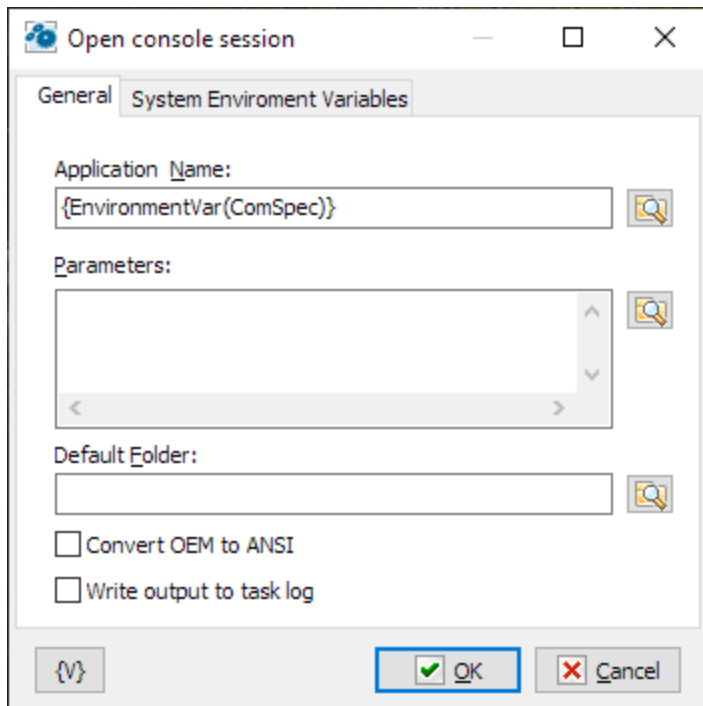
**Note:** *This feature is available for Business license only*

### 4.35.1 Open Console Session

Working with a console application should start with this action. First, start the application (open a session). All other steps are meaningless without opening a session.

There can only be one console session within a task.

#### General



#### Application name

This is the full name of the application to run. The default suggested expression is `{EnvironmentVar(ComSpec)}`. This is the full path to the system command prompt CMD.EXE

#### Parameters

Parameters are given as text. When the application starts, all line breaks will be replaced with spaces.

#### Default Folder

When you start an application, you can indicate the folder that will be active for the launched application. An empty value corresponds to the current RoboTask folder.

### Convert OEM to ANSI

In many operating system localizations, OEM and ANSI encodings are different. Console applications usually use OEM encoding, while GUI applications use ANSI encoding.

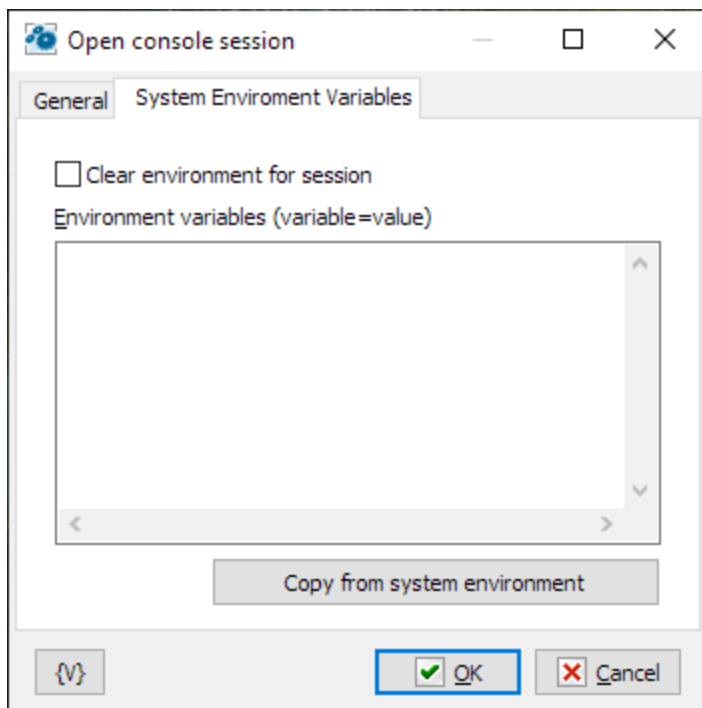
Check this box to avoid encoding conflict.

### Write output to log

Check this box if you need the parallel output to the task log when reading the output stream.

## System environment variables

You can change the system environment for the application when it launches.



### Clear environment for session

Check this box if you want to replace the environment completely. The action completely replaces the environment variables with the values of the Environment Variables parameter.

### Environment variables

Specify the environment variables that you want to replace or add. The variables are shown as text, and each line looks like the following:

```
variable_name=value
```

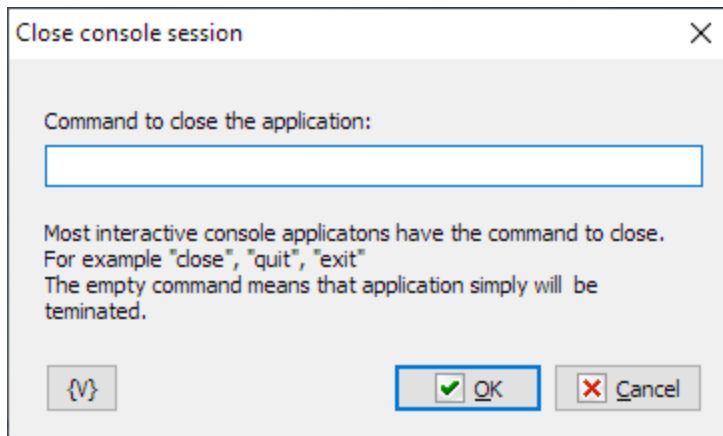
### Copy from system environment

This button allows you to fill the list with the current values of the environment variables.

**Note:** *This feature is available for Business license only*

## 4.35.2 Close Console Session

The action closes the console session and the application. You can specify a command to close the application. Typically, these commands are 'exit', 'quit', and 'close'.



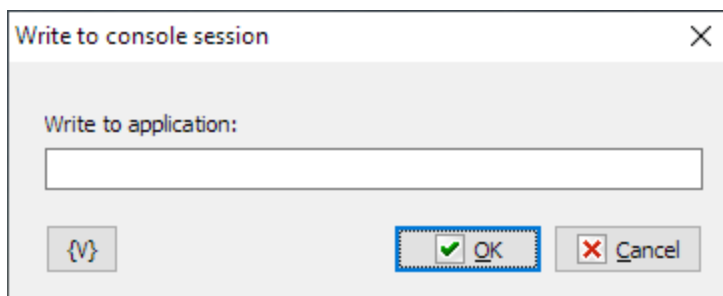
### Command to close the application

Specify a command to terminate the application. If no command is specified, the action will forcibly close the application with the *terminate* command.

**Note:** *This feature is available for Business license only*

## 4.35.3 Write To Console Session

The action sends a command to the console application. The command ends when pressing the 'Enter' key.



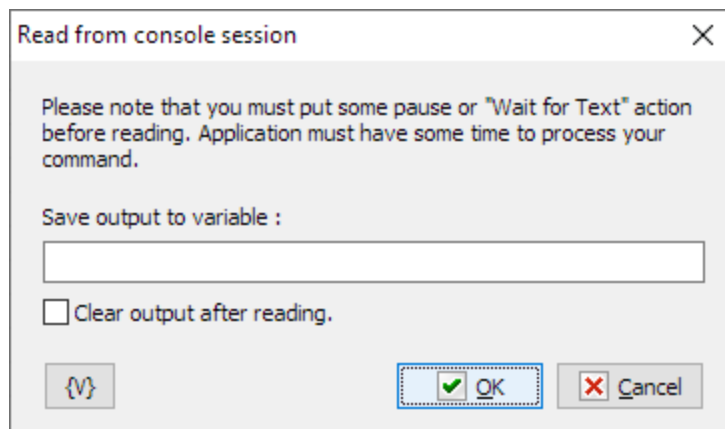
### Write to application

Specify the command as a string.

**Note:** *This feature is available for Business license only*

## 4.35.4 Read from Console Session

The action reads the output stream of the console application into a variable. Note that after sending a command, the application must be given some time to output the result of the command. You must insert a pause (100-200 milliseconds or more) or wait for a key phrase to be output before reading (look at the action [Wait for Text](#)<sup>[514]</sup>)



### Save output to variable

Enter the name of the variable where you want to save the read value.

### Clear output after reading

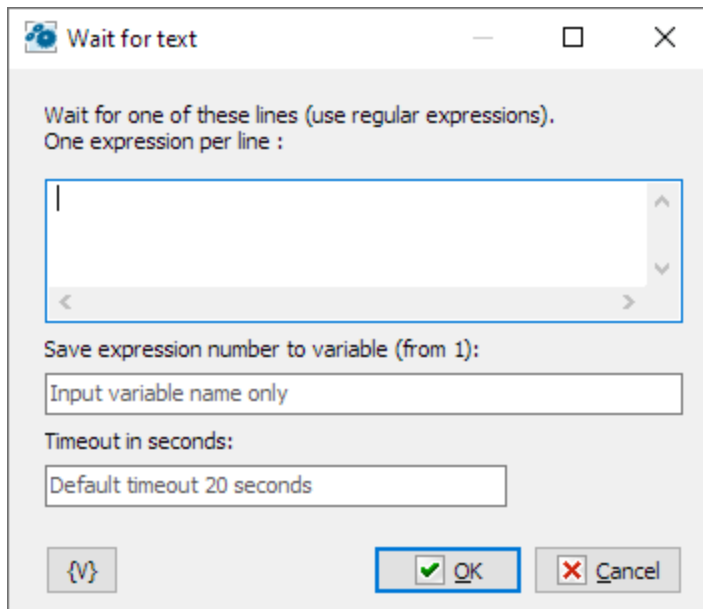
If you want to clear the output buffer after reading, check this box.

**Note:** *This feature is available for Business license only*

## 4.35.5 Wait for Text

This action expects a key phrase in the output stream. Some applications may output different values depending on some conditions. In this case, you can specify several phrases, with one phrase per line.

The key phrase should be written using regular expression syntax. The explicit string is a special case of a regular expression.



### Wait for one of these lines

Write one or more waiting phrases using regular expressions. One expression per line.

### Save expression number to variable

Specify the name of the variable you want to store the number of the expression you found. In some cases, more than one answer is possible for the application. The number of the expression starts with 1. If you get 0, it means that no specified expression was found during the timeout.

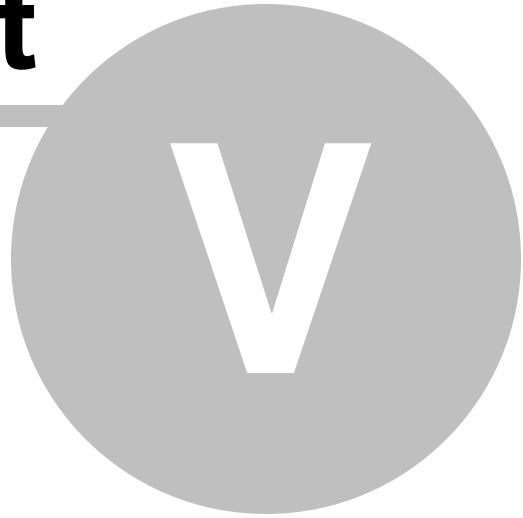
### Timeout in seconds

Specify the timeout in seconds. If a specified value cannot be converted to an integer; the default value is set to 20 seconds.

**Note:** *This feature is available for Business license only*

# Part

---



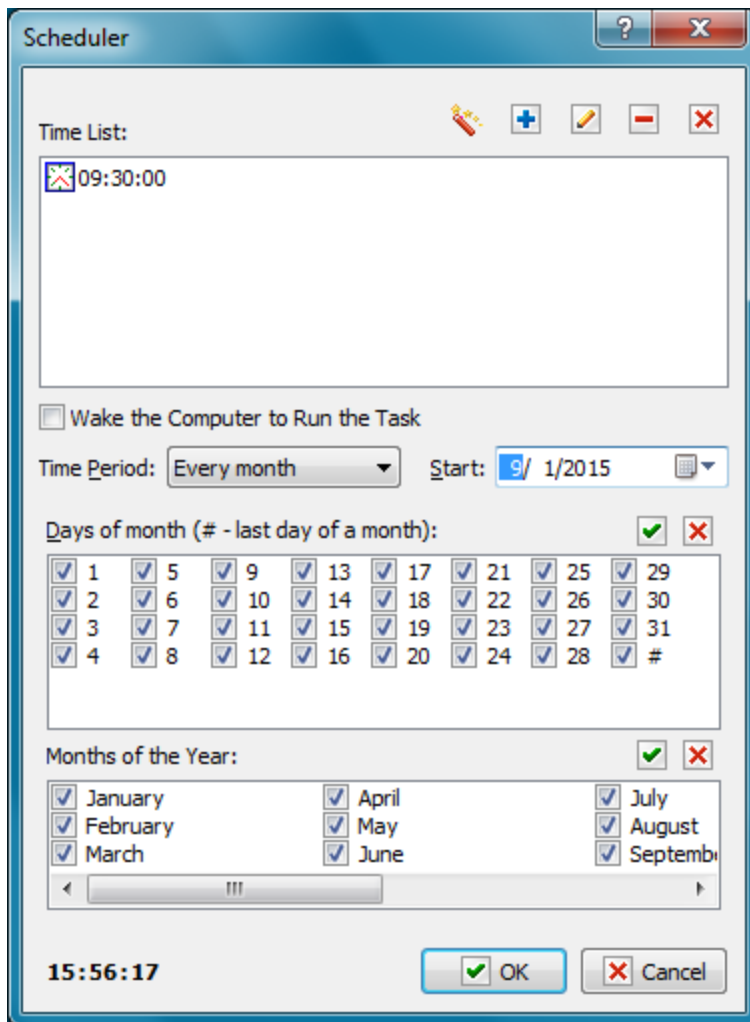
## Task Triggers

## 5 Task Triggers

Triggers for automation of your tasks.

### 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

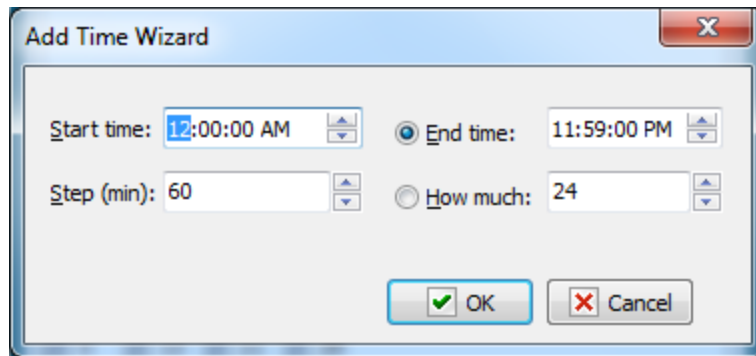
Enter a time to execute the action and then click the Plus button to add the time to the list.

#### 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.

### Wake the Computer to Run the task

Switch on this checkbox if you want to wake computer from "sleep" or "hibernate" mode. System timer will wake the computer 20 seconds before specified time and then the trigger will start the task.

#### Note:

Your system must have "Allow wake timers" enabled if you want to use this feature. This parameter is disabled by default in system installed on notebook. [See here how to enable it](#) 518

### 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.

## 5.1.1 How to enable wake timer

1. Open Control panel
2. Open Power options
3. Click on "Change plan settings" link

## Preferred plans

 **Balanced (recommended)**

Automatically balances performance with energy consumption on capable hardware.

[Change plan settings](#) **High performance**

Favors performance, but may use more energy.

[Change plan settings](#)

## Hide additional plans

 **Power saver**

Saves energy by reducing your computer's performance where possible.


[Change plan settings](#)

4. Click on "Change Advanced power settings"

### Change settings for the plan: High performance

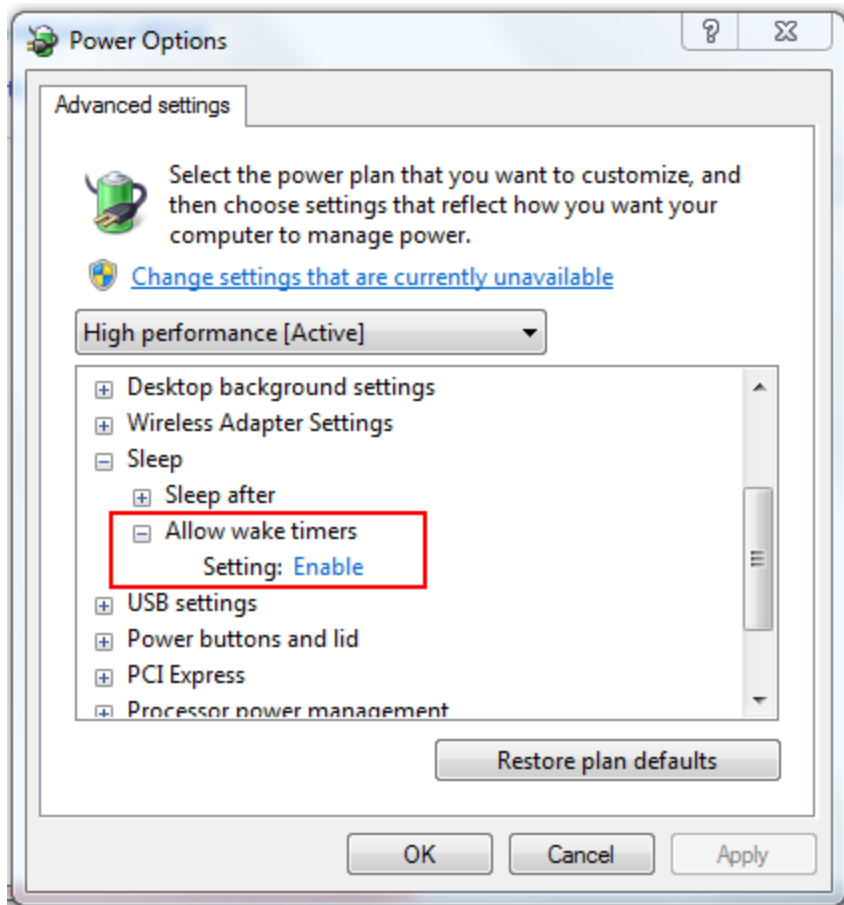
Choose the sleep and display settings that you want your computer to use.

 Turn off the display:

 Put the computer to sleep:

[Change advanced power settings](#)[Restore default settings for this plan](#)

5. Enable "Allow wake timers" parameter for your power plan

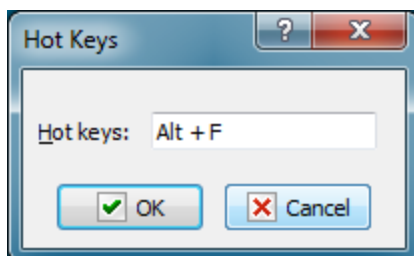


## 5.2 Hot Keys

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.

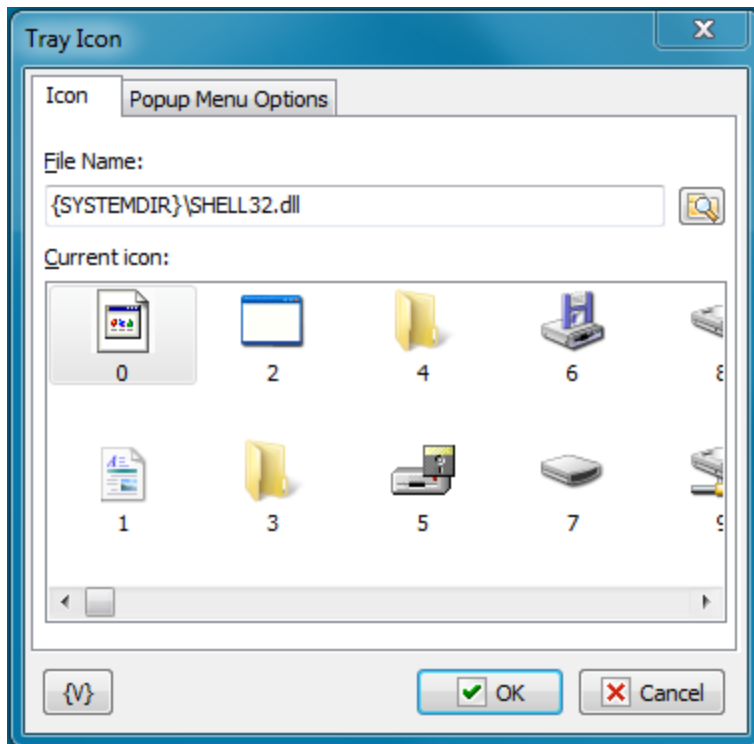
## 5.3 Tray Icon

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.

### Icon



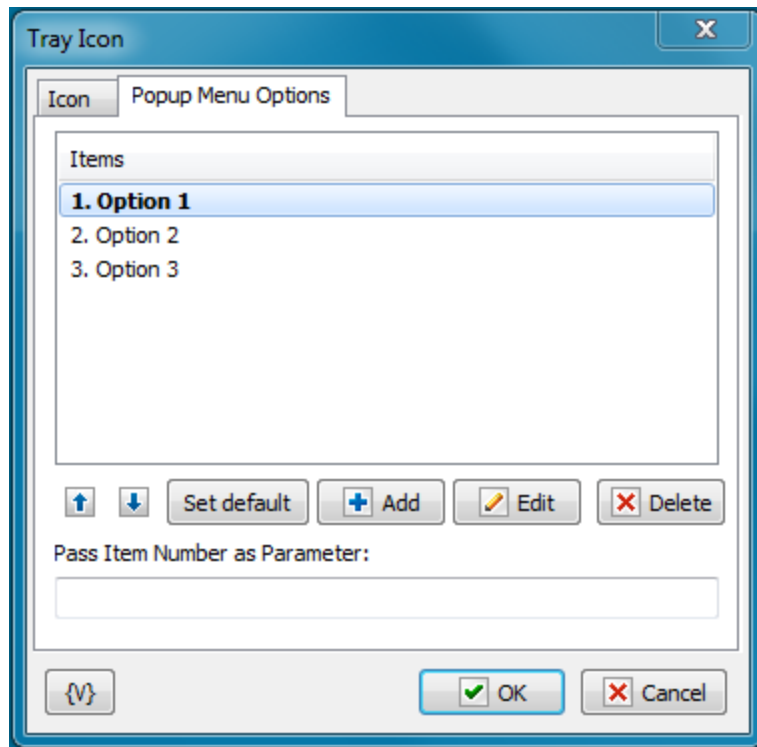
### File Name

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.

### Popup Menu Options



One can define the popup menu items for an icon on this tab. In this case the ordinal number (beginning from 1) of selected will be passed into a/the task as a parameter.

#### **Items**

The list of menu items. This list can be edited with buttons *Add*, *Edit* and *Delete*. Also it is possible to change the order of options using buttons *Move Up* and *Move Down*.

To select Default Menu Item use the button *Set Default*.

#### **Pass Item Number as Parameter**

Specify the name of parameter in which the number of menu item must to be passed. On double click on the icon, the default item number of menu will be passed.

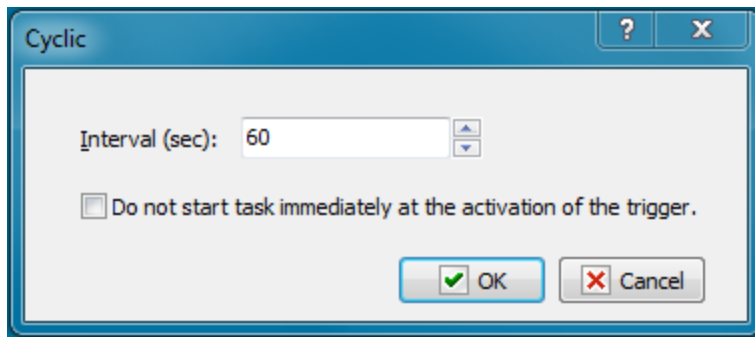
If the menu is empty, the parameter is 0 always.

#### **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)

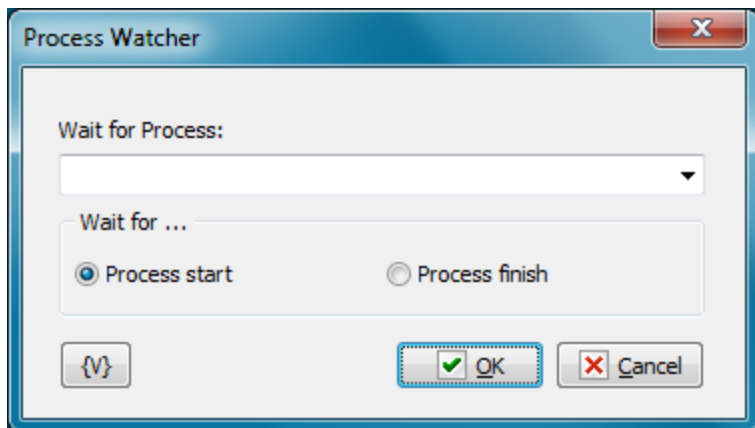
Enter the number of seconds that you want the task or actions to wait between loops.

### Do not start immediately at the activation of the trigger

By default Cyclic trigger starts the task immediately when the trigger is being started. switch ON this checkbox if you want to prohibit the first running of the task. In most cases this doesn't matter, but may be helpful sometimes.

## 5.5 Process Watcher

The trigger responds to the start or stop of the specified process.



### Wait for process

Select from the list or write the name of the process. For example, Notepad.exe

### Wait for

- **Process start** - waiting for start the specified process.
- **Process finish** - waiting for completion of the specified process

## 5.6 File Monitor

Executes a task when a file is created, deleted, or changed.

- [General tab](#)<sup>524</sup>
- [Save to File tab](#)<sup>526</sup>
- [Save to Task Variables tab](#)<sup>528</sup>
- [Advanced tab](#)<sup>529</sup>

### General tab

The screenshot shows the 'File Monitor' dialog box with the 'General' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are four tabs: 'General', 'Save to File', 'Save to Task Variables', and 'Advanced'. The 'General' tab contains the following elements:

- Folders:** A text area for listing folders to monitor, with a plus button (+), an edit button (pencil), a minus button (-), and a delete button (X) to its right.
- Include Masks (comma separated):** A text field containing the wildcard pattern '\*.\*'.
- Exclude Masks (comma separated):** An empty text field.
- Filter condition:** A group box containing several checked options:
  - Watch for new files
  - Watch for deleted files
  - Watch for changed files:
  - Monitor files
  - Monitor folders
  - Time changed (>)
  - Time changed (<)
  - Size changed (>)
  - Size changed (<)
  - Attributes changed
- Poll Interval (sec):** A spin box set to the value '60'.
- Buttons:** A help button (V), an 'OK' button with a green checkmark, and a 'Cancel' button with a red X.

### 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)**

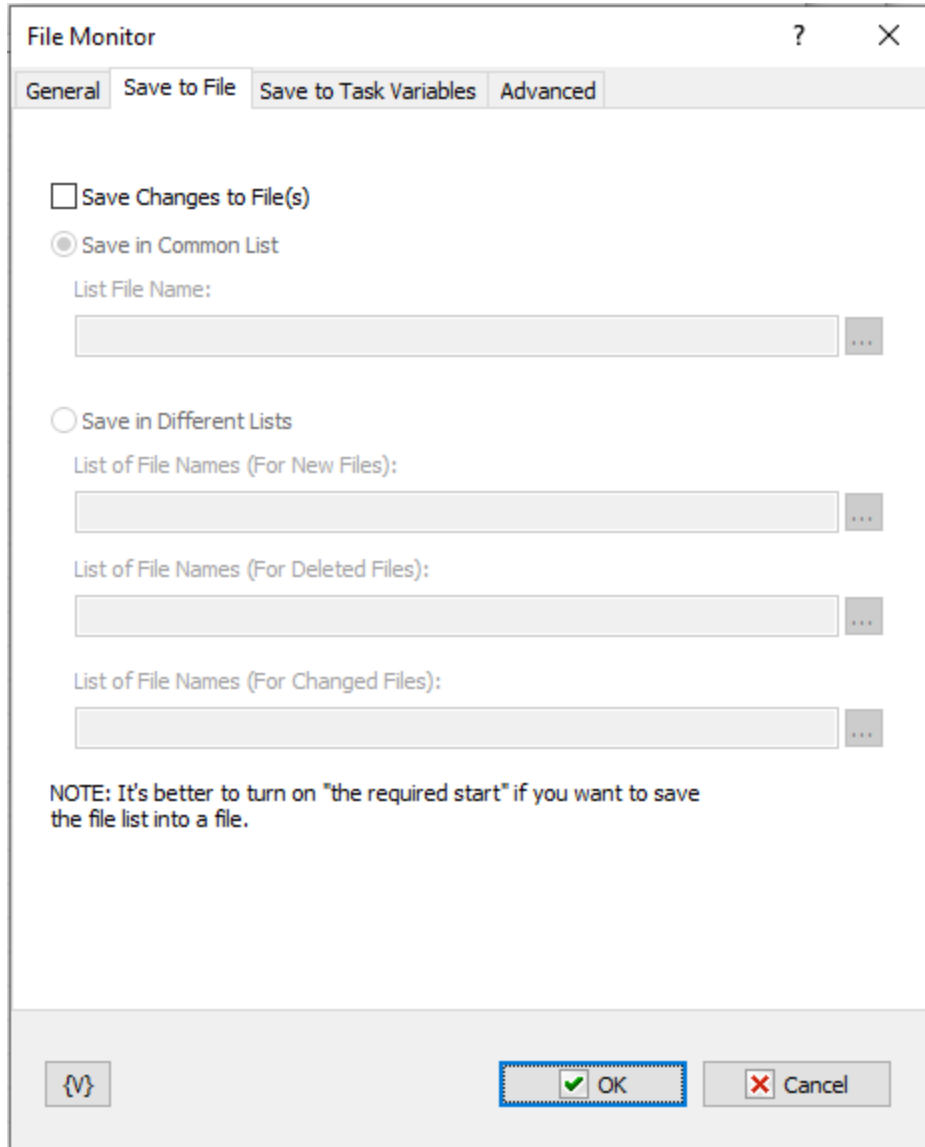
Enter the time in seconds that you want the event to wait between requests for data.

**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 to File tab



The screenshot shows the 'File Monitor' dialog box with the 'Save to File' tab selected. The dialog has four tabs: 'General', 'Save to File', 'Save to Task Variables', and 'Advanced'. The 'Save to File' tab contains the following options:

- Save Changes to File(s)
- Save in Common List
  - List File Name:  
 ...
- Save in Different Lists
  - List of File Names (For New Files):  
 ...
  - List of File Names (For Deleted Files):  
 ...
  - List of File Names (For Changed Files):  
 ...

NOTE: It's better to turn on "the required start" if you want to save the file list into a file.

At the bottom of the dialog, there is a refresh button (circular arrow icon), an 'OK' button (highlighted with a blue dashed border), and a 'Cancel' button.

### 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.

## Save to Task Variables tab

The screenshot shows a dialog box titled 'File Monitor' with a close button (X) and a help button (?). The 'Save to Task Variables' tab is selected. The dialog contains the following options and fields:

- Save Changes to Task Local Variables
- Save in Common List
  - Variable Name:
- Save to Separate Variables
  - Variable Name for New Files:
  - Variable Name for Deleted Files:
  - Variable Name for Changed Files:

NOTE: It's better to turn on "the required start" if you want to save file list to task variables.

At the bottom, there is a refresh button (circular arrow icon), an OK button with a green checkmark, and a Cancel button with a red X.

### Pass Changes into the task as parameters

#### Save in Common list

To pass the common list of changed, new and deleted files to the task select this option and input the parameter name for common list of files.

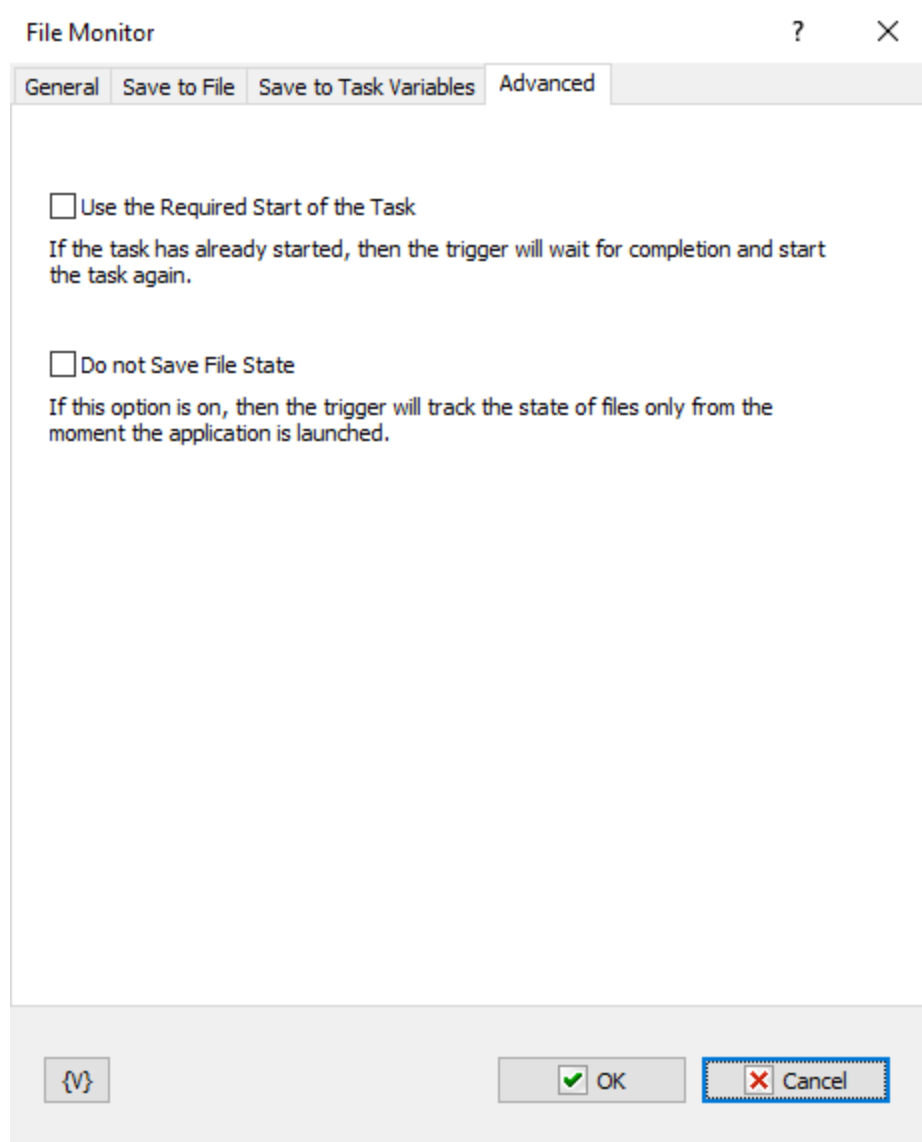
#### Variable Name

Variable name for common list of files

#### Save to Separate Variables

- Variable Name for New Files
- Variable Name for Deleted Files
- Variable Name for Changed Files

## Advanced tab



### Use Required Start of the Task

This parameter guarantees the start of a task. The problem is that the task can be working when trigger tries to start it. In the most cases that doesn't matter because you need only to start the task at the appointed moment. In this case the trigger just send the signal to start the task. If the task is idling, it will be started. If the task is already started, the RoboTask ignores the signal of a start because the task works by now.

But a task can be started with parameters and fulfill different actions against parameter values. Switch on this parameter if you need to ensure start up of a task when each effort to start a task. In this case the action will do efforts to start the task until a successful result.

### Do not save file state

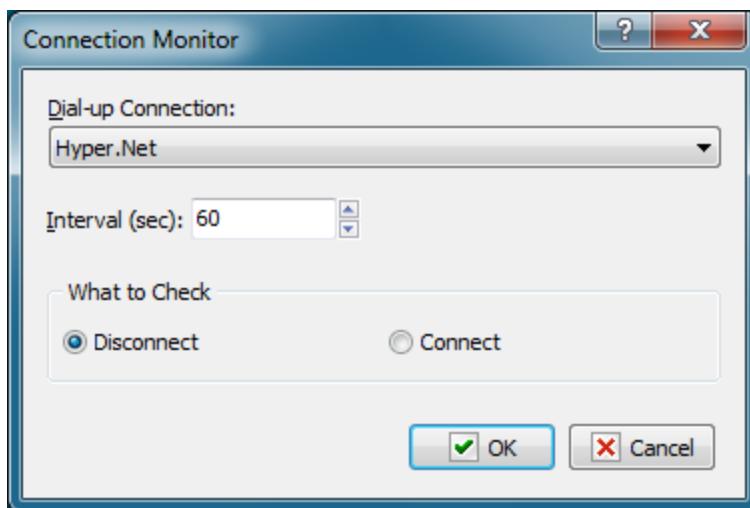
If this option is on, then the trigger will track the state of files only from the moment the application is launched.

### 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.

## 5.7 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

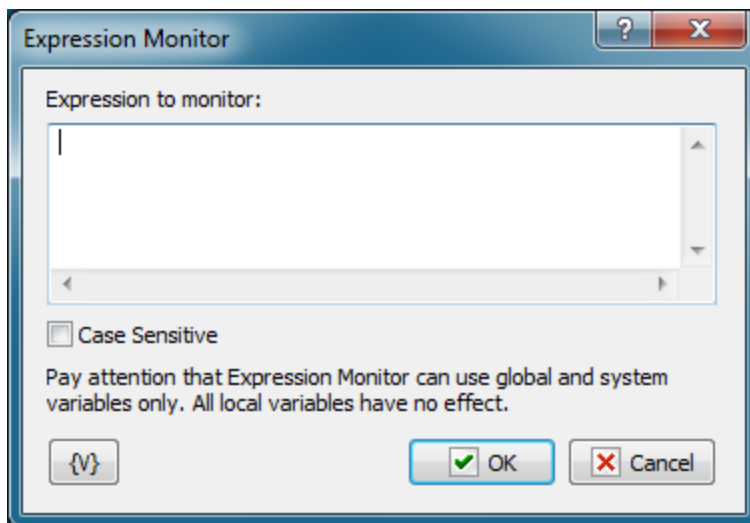
## 5.8 Expression monitor

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>522</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.

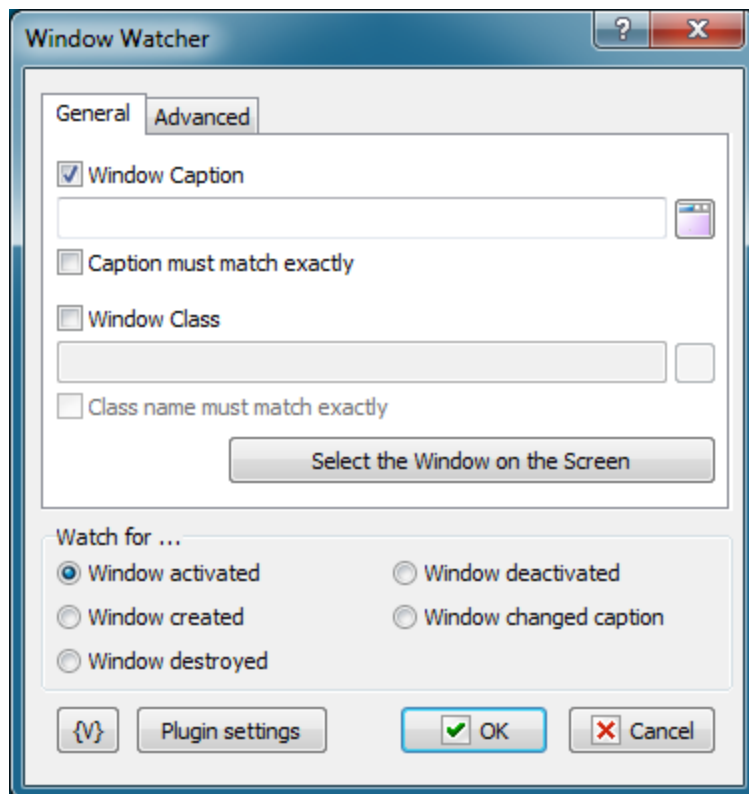
### Case Sensitive

Expression Monitor is insensitive to a register By default. If it is important to take into account the register, switch on this checkbox.

## 5.9 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.

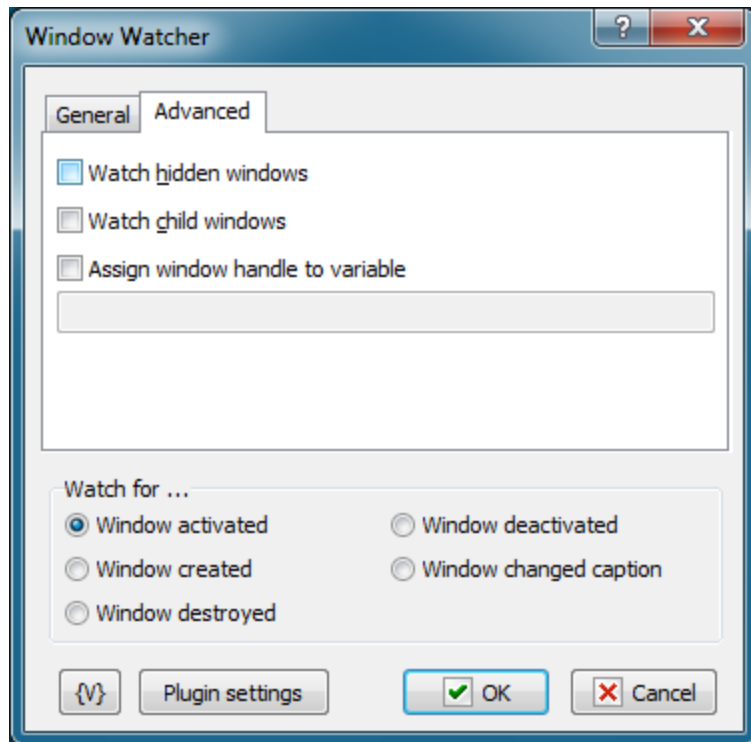
### Class must match exactly

Select this checkbox if the window class string should match exactly.

### Select the Window on the Screen

Select existing window interactively. This function highlights the window pointed by mouse. Press CTRL key to select current window or ESC to cancel operation

## 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>[252]</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.



## 5.10 On Time Changed

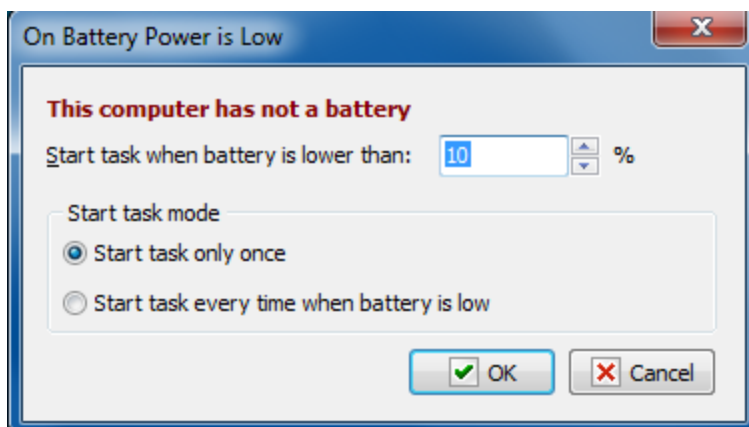
Executes a task when the computer system time changes. No specific settings are associated with this triggering event.

## 5.11 On Battery Power is Low

This trigger reacts to the low battery to the specified level.

**Note:** *this trigger makes sense only for the computers which have autonomous battery (laptop) or for computer connected to Smart UPS.*

The trigger is activated if the computer is running on battery and battery charge has dropped to the specified level. If battery charge is low but the computer is running on a network, the trigger is not activated. The parameter editor will notify you if there is not set a battery on your computer.



### Start task when battery is lower than

Specify the battery level from 3% to 100%

### Start task mode

- **Start task only once.** A task is started only once when the specified level has reached.
- **Start Task every time when battery is low.** A task is started approximately one time a minute if the battery level is lower or equal to the specified one

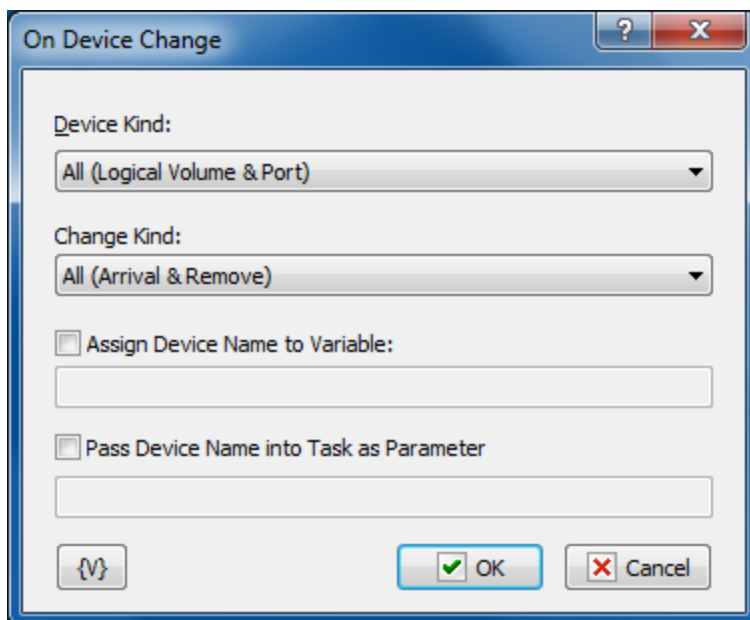
When power supply is restored, this trigger goes to the initial waiting status.

## 5.12 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.

## 5.13 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

Switch on this checkbox and specify a name of variable where you want to save the name of the device that has been changed.

### Pass Device Name into the Task as Parameter

Switch on this checkbox and specify a name of parameter which will be passed into the task at starting.

## 5.14 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.

## 5.15 On Display Changed

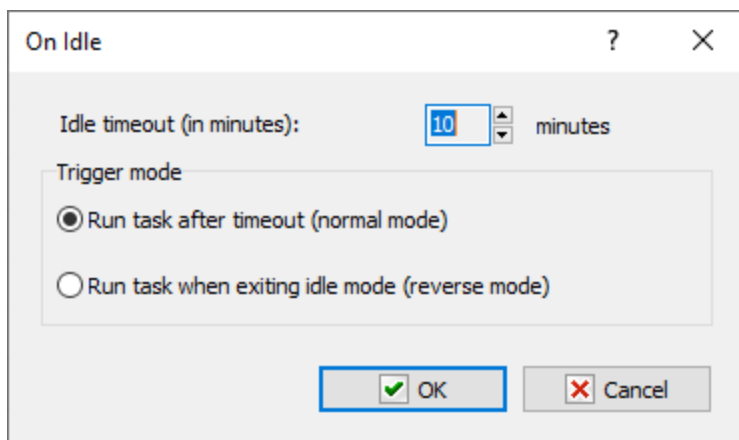
Executes a task when the display on the computer screen changes. No specific settings are associated with this triggering event.

## 5.16 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.

## 5.17 On Idle

Executes a task when the computer has been idle for a specified period of time.



### Idle timeout in minutes

Enter the time that you want in minutes.

### Trigger mode

*Run task after timeout* - normal mode;

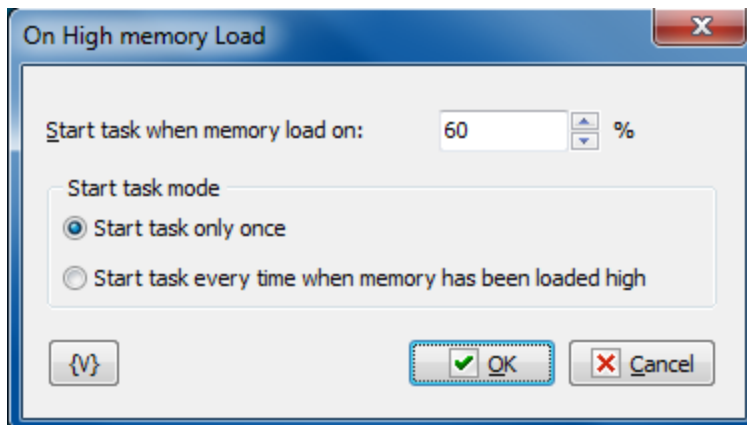
*Run task when exiting idle mode* - the trigger is activated after the specified timeout and fires when the keyboard or mouse is touched.

## 5.18 On Resume

This trigger starts a task when the computer "awakes" out of the Sleep Or Hibernate mode. It does not demand any additional settings.

## 5.19 On High Memory Load

This trigger starts the task when the RAM loading is exceeded the specified value. For example, to send a message or write a warning into a log-file.



### Start task when memory load on

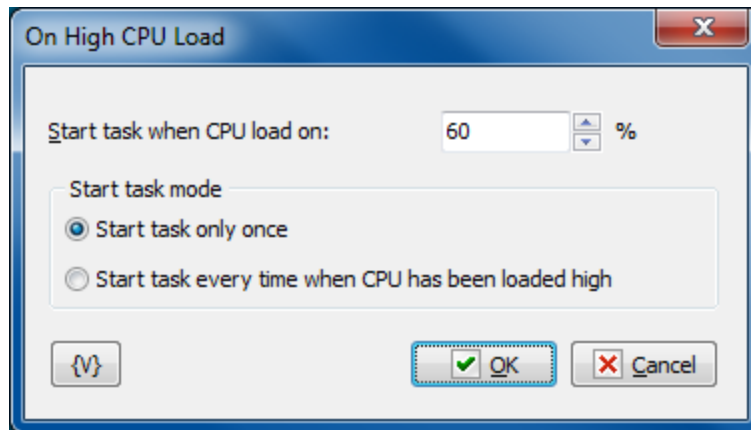
Specify a boundary value in percents. If the RAM loading exceeds this value, the trigger will start the task.

### Start task mode

- **Start task only once.** This option starts the task only once at the moment when the value is exceeded. If the memory loading falls below the value, the trigger will start the task again when the value is exceeded the next time.
- **Start task every time when memory has been loaded high.** In this mode the trigger will start the task approximately once a minute while the memory loading is higher than the specified value.

## 5.20 On High CPU Load

This trigger starts the task when the loading of the computer CPU is higher than the specified value. For example, to send a message or write a warning into a log-file.



### Start task when memory load on

Specify a boundary value in percents. If the CPU loading exceeds this value, the trigger will start the task.

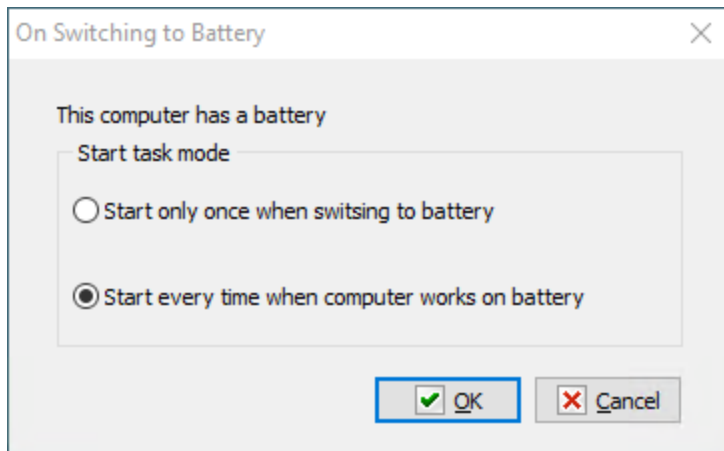
### Start task mode

- **Start task only once.** This option starts the task only once at the moment when the value of loading is exceeded. If the CPU loading falls below the value, the trigger will start the task again when the value is exceeded the next time.
- **Start task every time when memory has been loaded high.** In this mode the trigger will start the task approximately once a minute while the CPU loading is higher than the specified value.

## 5.21 On Switching to Battery

The trigger is activated when the power is switched to battery. Naturally, this trigger makes sense, if the computer has standby power supply from the recharging battery. For example: laptop or computer connected to the Smart UPS.

If you try to activate the trigger on a computer without a battery, the trigger will display the *"This computer has no battery"* warning in the RoboTask system log. In this case the trigger will not work.



### Start Task Mode

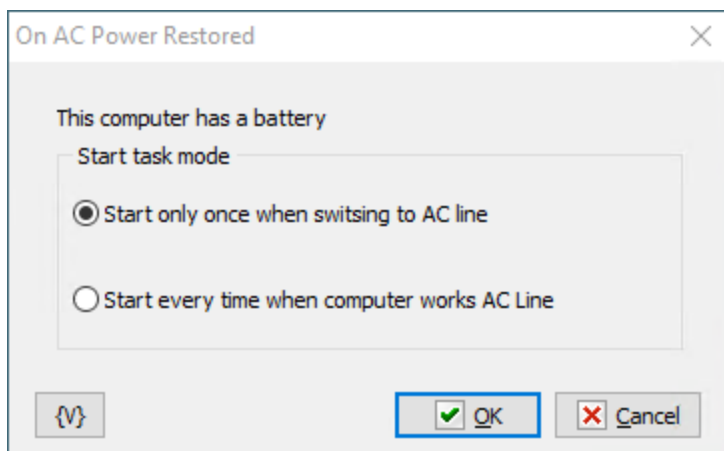
Two trigger modes are possible:

- **Start only once when switching to battery** – the trigger is activated once when switching to the battery.
- **Start every time when computer works on battery** - the trigger is activated when switching to the battery. Then the trigger is activated intermittenly about once a minute when on battery power

## 5.22 On AC Power Restored

The trigger is activated when AC power of the computer is restored. Naturally, this trigger makes sense, if the computer has standby power supply from the recharging battery. For example: laptop or computer connected to the Smart UPS.

If you try to activate the trigger on a computer without a battery, the trigger will display the *"This computer has no battery"* warning in the RoboTask system log. In this case the trigger will not work.



### Start Task Mode

Two trigger modes are possible:

- **Start only once when switching to AC line** – the trigger is activated once when switching to the mains supply.
- **Start every time when computer works AC line** - the trigger is activated when switching to AC mains. Then the trigger is activated intermittently about once a minute when on the mains supply.

## 5.23 On Local/Remote Session Changes

This trigger responds to opening and closing a user session on your PC. The trigger responds to any other session, no matter in what session RoboTask is running. This can be useful when working on Terminal Server.

When a trigger responds and starts the task, it can write to a task variable information about the session in JSON format, with fields:

- *event* - which event occurred. Possible options:
  - CONSOLE\_CONNECT
  - CONSOLE\_DISCONNECT
  - REMOTE\_CONNECT
  - REMOTE\_DISCONNECT
  - SESSION\_LOGON
  - SESSION\_LOGOFF
  - SESSION\_LOCK
  - SESSION\_UNLOCK
- *client\_name* - the client computer name. For local sessions, this field is always empty
- *client\_address* - the client network address. For local sessions, this address is always 0.0.0.0
- *domain* - the domain name of the client computer. For local sessions, this field is always empty
- *user\_name* - The client user name. For local sessions, this field is always empty.
- *hres* - horizontal dimension, in pixels, of the client's display
- *vres* - vertical dimension, in pixels, of the client's display
- *color\_depth* - color depth of the client's display.

On local/remote session changes

On console connect

On console disconnect

On remote connect

On remote disconnect

On session logon

On session logoff

On session lock

On session unlock

Save session info to variable:

{V} OK Cancel

### Trigger mode

Enable one or more events for the trigger to respond:

- Console connect
- Console disconnect
- Remote connect
- Remote disconnect
- Session logon
- Session logoff
- Session lock
- Session unlock

### Save session info to variable

Enter the variable name to save the session information.

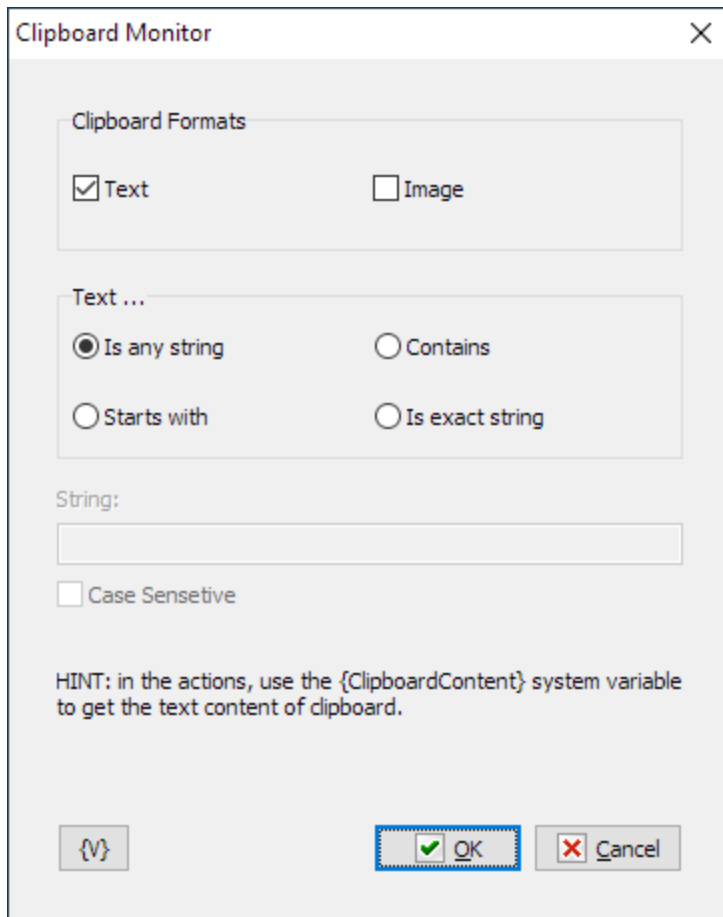


## 5.24 Clipboard Monitor

This trigger causes a task to be launched when the content of the system clipboard changes. The trigger keeps track of two formats: *text* and *image*.

The text tracking mode can be specified.

When an image is tracked, the trigger is activated following any image change.



The screenshot shows the "Clipboard Monitor" dialog box. It has a title bar with "Clipboard Monitor" and a close button (X). The dialog is divided into several sections:

- Clipboard Formats:** Contains two checkboxes: "Text" (checked) and "Image" (unchecked).
- Text ...:** Contains four radio buttons for text tracking modes: "Is any string" (selected), "Contains", "Starts with", and "Is exact string".
- String:** A text input field for specifying a string.
- Case Sensitive:** A checkbox that is currently unchecked.
- HINT:** A text area containing the text: "HINT: in the actions, use the {ClipboardContent} system variable to get the text content of clipboard."
- Buttons:** At the bottom, there are three buttons: a button with a curly brace and "V" symbol, a "OK" button with a green checkmark, and a "Cancel" button with a red X.

### Clipboard formats

Specify the format to be tracked. Two formats are available: *Text* and *Image*.

Choose at least one format or two formats at once.

### Text

Choose the text tracking mode. This parameter is omitted when images are tracked.

- **Is any string** – the trigger is activated when clipboard contains any text
- **Starts with** – the trigger is activated when the text in clipboard begins with the string specified in the "String" parameter
- **Contains** – the trigger is activated when the text in clipboard contains the string specified in the "String" parameter

- **Is exact string** – the trigger is activated when the text in clipboard fully matches the string specified in the "String" parameter

**String**

Enter a string for text tracking.

**Case sensitive**

Select this checkbox if an exact match (case-sensitive) is needed.

## 5.25 Registry Monitor

This trigger responds to changes in the system registry. Lists of registry changes can be saved in the task's variables.

**General**

The screenshot shows the 'Registry Monitor' dialog box with the 'General' tab selected. The dialog has a title bar with a question mark and a close button. Below the title bar are two tabs: 'General' and 'Save changes'. The 'General' tab contains the following elements:

- A label 'Registry key:' followed by a text input field and a browse button (three dots).
- A label 'Value names to monitor (comma separated):' followed by a text input field containing the text 'Wildcards allowed.'
- A checkbox labeled 'With subkeys' which is currently unchecked.
- A checkbox labeled 'Save changes to task variables' which is currently checked.

At the bottom of the dialog, there are three buttons: a help button (question mark in a square), an 'OK' button (green checkmark and 'OK' text), and a 'Cancel' button (red X and 'Cancel' text).

**Registry key**

The registry branch (key) to monitor for changes.

**Value names**

Specify the names of values to monitor, separated by commas. You can use value masks with "?" and "\*" characters. An empty field value means "all values."

**With subkeys**

If you want to monitor changes throughout the subtree starting from the specified branch, enable this checkbox. Please note that you should not monitor excessively large parts of the registry, as it can slow down the trigger's operation.

### Save changes to task variables

If you want to save the lists of changes in task variables, enable this checkbox and fill in the variable names on the "Save Changes" tab.

### Save changes

All changes are saved in variables as multiline text. Typically, multiple values can change at once. Therefore, you should process these variables as multiline text, for example, using a [Text Loop](#)<sup>[234]</sup>. If no variable name is specified (empty input field), no saving occurs.

For "Removed keys" and "Added keys" each line saves the full key path.

For values (*Removed values*, *Changes values*, *Added values*), each line contains a comma-separated string with two elements:

- The zeroth element contains the full key name.
- The first element contains the value name.

### Removed keys

The variable name in which the list of removed keys is saved.

### Added keys

The variable name in which the list of added keys is saved.

**Removed values**

The variable name in which the list of removed values is saved.

**Changes values**

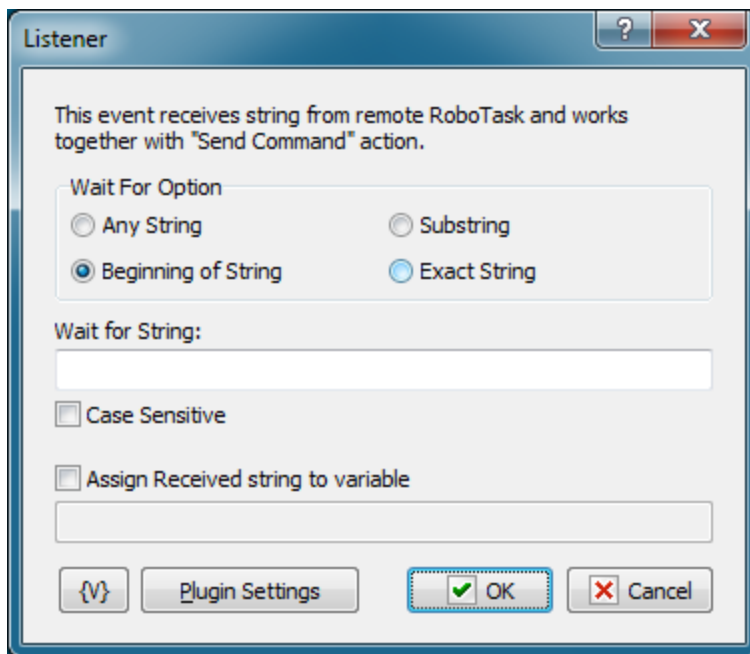
The variable name in which the list of changed values is saved.

**Added values**

The variable name in which the list of added values is saved.

## 5.26 Listener

Allows you to start a task when a command from another computer is received. Works together with the [Send Command action](#)<sup>[297]</sup>. 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>[300]</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.

## 5.27 Pixels Monitor

This event starts the task when the condition of comparison of pixels on the screen or in the specified window is fulfilled. The parameter editor is very similar to the editor of the action "[Wait for Pixels](#)"<sup>[388]</sup>.

You can specify a set of pixels (one or several) in this trigger. You need to specify the necessary color and the tolerance of the comparison of pixels for every pixel.

You can specify the condition of color coincidence for a pixel set:

- Match ALL of the following pixels (AND)
- Match ANY of the following pixels (OR)

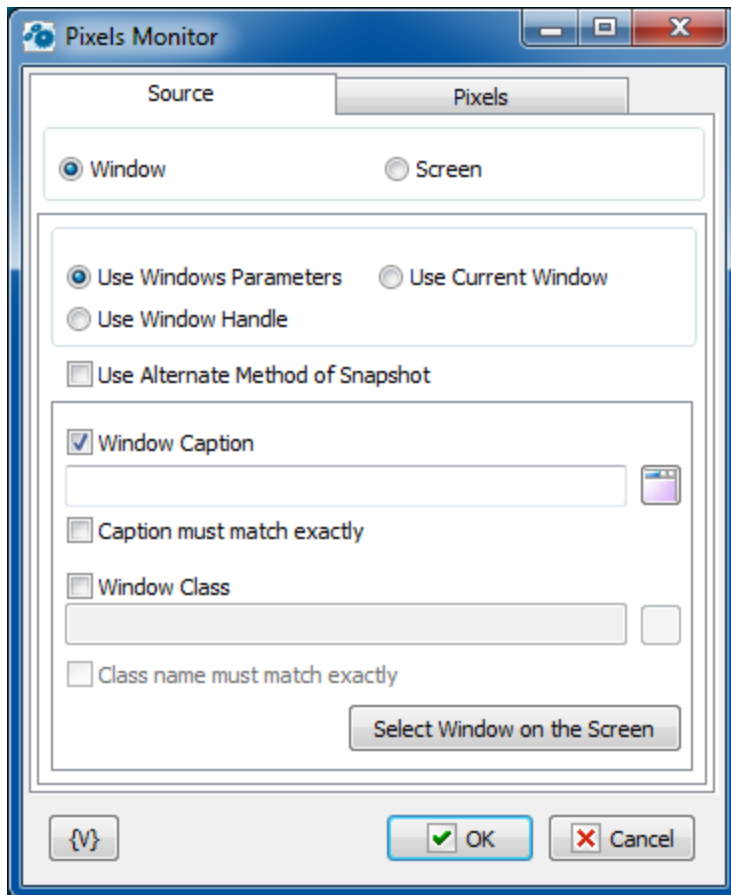
Thereby, you can specify the exactly enough condition of changes on the screen.

### Source tab

When defining parameters, at first you need to specify the source of an image:

- **Window** - This is a separate window. All coordinates of pixels will be determined concerning this window. If you have selected the window as a source, you need to specify the window parameters.
- **Entire screen** - This is a screen with all visible windows.

The window parameters are similar to the parameters in the actions [Window Snapshot](#)<sup>[376]</sup>, [Window Command](#)<sup>[252]</sup>, [Window Size and Position](#)<sup>[259]</sup>, and [Window Elements loop](#)<sup>[262]</sup>.



### Window Parameters Mode

There are three options:

- **Use Window parameters.** Use the Window Caption and the Window Class to look for the necessary window.
- **Use Window Handle.** This is a search for a window by a handle. The handle can be got in the loop Window Loop, from the trigger Wait for Window or some other way.
- **Use current window.** This is a current window in a moment of the executing of the action. This mode does not require any additional parameters.

### Use Window parameters Window Caption.

If you want to search for a window by means of its caption, switch on this checkbox and input the expected window caption. You can input only part of the caption. In this case the action will check availability of the input part in the window caption.

### Caption Must Match Exactly.

If you need the exact coincidence of a window caption, switch on this checkbox.

### Use alternate method of Snapshot

Generally, getting a window image works correctly. But, unfortunately, not always. Some applications do not allow you to take a picture directly from the window. For example, *Google Chrome* or applications with the *Metro user interface*. To get the image, you must use an alternative method in this case.

An alternative method is that the application cuts a window image from a desktop snapshot.

But please note that this method cannot get the correct image if the desired window (or part of the window) is covered by another window or goes beyond the desktop.

You need to ensure the visibility of the window before taking a snapshot.

### **Window Class.**

If you want to search the window by a window class, switch on this checkbox and input the expected name of the window class. You can input only part of the name. In this case the action will check availability of the input line in the name of the window class.

### **Class Name Must Match Exactly.**

If you need the exact coincidence of a window class, switch on this checkbox. Usually it is worthwhile to search by the whole class name, because a window class is permanent and does not change its value during the "life" of the window, unlike a caption.

### **Select Window on the Screen.**

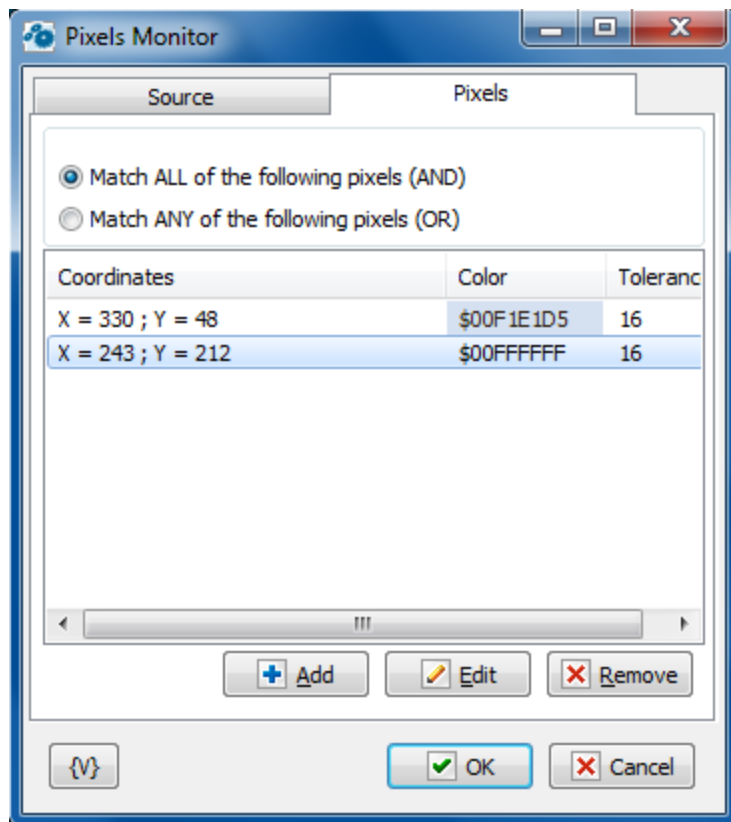
When pressing this button, you turn on the mechanism of selection a window on the screen. The window, which the mouse pointer points on, highlights with a red frame. Meanwhile, you can use the mouse to perform some actions. To select the necessary window, press *Ctrl*. To cancel - press *Esc*. When selecting a window, the fields Window Caption and Window Class are filled in automatically.

### **Window Handle.**

Input the expression containing a window handle. This parameter is used for the mode Use Window Handle. A window handle is just a number, identifier of a window in a current session of system work.

### **Pixels tab**

You need to specify the pixel set for checking on this tab.



### Comparison mode

- Match ALL of the following pixels (AND)
- Match ANY of the following pixels (OR)

### List of pixels

You can edit a list using the buttons *Add*, *Edit* and *Remove*.

By means of an edit window you can change pixel parameters both manually and by means of interactive instruments.

#### Pixel parameters:

**X, Y** - It is coordinates of a pixel from the left corner. If a source of a picture is a window, coordinates are determined from the upper left corner of the screen window.

**Pattern color** - This is a pattern of the color with which the action will compare the color of a pixel. Color is just a number. For comfort it is written in hexadecimal.

For example, \$0066FF00 or \$66FF00.

In hexadecimal you can write color using 6 symbols (as RGB): the first couple (younger) is red, the second couple is green, the third couple (elder) is blue color.

**Tolerance** - This is an accuracy of color coincidence. The error can be changed from 0 to 255 (in hex from 00 to FF). When it is 0, it is exact color coincidence. When it is 255, it is a coincidence with any color.



Tolerance is an acceptable difference between two colors for any component RGB (Red - Green - Blue). Every component of the color can be changed from 0 to 255 (FF in hexadecimal).

On default, the tolerance is 16.

## 5.28 WebSocket Server On Message

This trigger responds to the messages sent by different clients. This can be a WebSocket Client send message or any other applications or scripts that support WebSocket connections.

**Note:** *This feature is available for Business license only*

WebSocket Server On Text Message

WebSocket Server:  
Localhost server (8888) [v]

WS Server Manager...

Message ...

Is any string       Contains  
 Starts with       Is exact string

String:  
[ ]

Case Sensitive

Save received message to variable:  
MSG

[ { } ]      [ OK ]      [ Cancel ]

### WebSocket server.

Choose the required Websocket server from the list. Please note that you cannot choose the server when editing a network task. Instead, you must enter the server ID. You must know the server ID in advance (see *WS Connection manager->Servers->Advanced*). If an invalid ID is specified, an "Invalid WS Server ID" error is returned when executing.

### WS Server Manager

You can view or edit lists of registered WebSocket servers and clients (see [WS Connection Manager](#)<sup>[497]</sup>).

### Message...

Several options for analyzing the message received are supported

- **Is any string** – in this case, the trigger launches the task when any string is received;
- **Starts with** – the message must start with the string specified in the String field;
- **Contains** – the message must contain the string specified in the String field
- **Is exact string** – the message must exactly match the specified string.

### String

Enter the string to compare it with the message

### Case Sensitive

If you need the case-sensitive comparison, then tick this checkbox.

### Save received message to variable

Specify the name of the local variable of the task, where you want to save the received message. If the variable name is empty, the trigger doesn't save anything.

## 5.29 WebSocket Client On Message

This trigger responds to messages sent by the server the client is connected to. This can be a WebSocket server send message or any other applications or scripts that support WebSocket connections as a server.

**Note:** *This feature is available for Business license only*

WebSocket Client On Text Message

WebSocket Client:  
WebSockets Client 1

WS Client Manager...

Message ...

Is any string       Contains

Starts with       Is exact string

String:

Case Sensitive

Save received message to variable:  
MSG

{}

OK Cancel

### WebSocket Client

Choose the WebSocket client from the list. If you edit a network task, the list is not available and you have to enter the client ID manually (see *WS Connection manager->Clients->Advanced*). If the client ID is incorrect, the action will get an "Invalid WS Client ID" run-time error.

### WS Client Manager

You can view or edit lists of registered WebSocket servers and clients (see [WS Connection Manager](#)<sup>[497]</sup>).

### Message...

Several options for analyzing the message received are supported

- **Is any string** – in this case, the trigger launches the task when any string is received;
- **Starts with** – the message must start with the string specified in the String field;
- **Contains** – the message must contain the string specified in the String field
- **Is exact string** – the message must exactly match the specified string.

### String

Enter the string to compare it with the message

**Case Sensitive**

If you need the case-sensitive comparison, then tick this checkbox.

**Save received message to variable**

Specify the name of the local variable of the task, where you want to save the received message. If the variable name is empty, the trigger doesn't save anything.

## 5.30 WebSocket Server On HTTP Request

The WebSocket server can serve as an HTTP server at the same time. WS server distinguishes between simple messages via WebSocket connection and a full-fledged HTTP request. HTTP request can be sent from a browser or scripting languages. In other words, you can launch a RoboTask from your web-browser with this trigger.

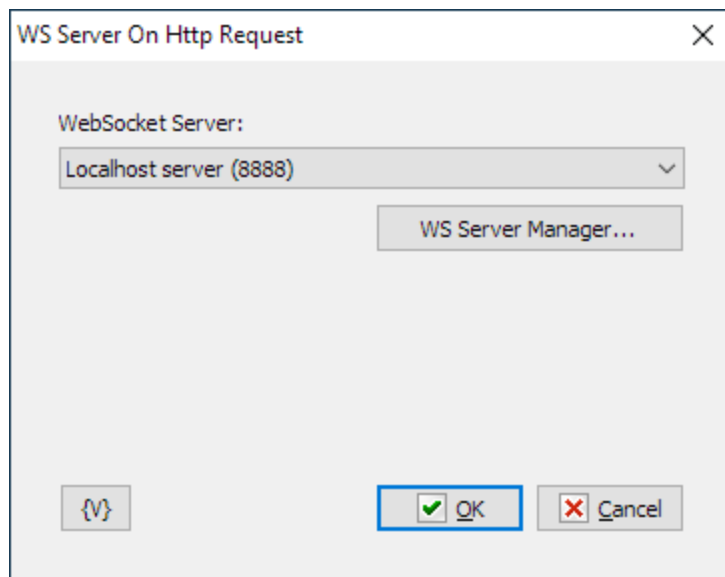
If the task is launched by this trigger, you can use the following variables in the task to analyze the HTTP request:

- **HttpFilename** - Return the file name of an HTTP request
- **HttpHeader(Header-Name)** - Returns the header of an HTTP request. In the parameter you must record the name of the header. For example: accept-encoding, user-agent, referer etc.
- **HttpMethod** - Returns the method of an HTTP request (GET, POST, etc.)
- **HttpPath** - Return the path of an HTTP request.
- **HttpPostParam(ParamName)** - Return the post parameter value of an HTTP request
- **HttpQuery** - Return the query string of an HTTP request
- **HttpQueryParam(ParamName)** - Return the query parameter value of an HTTP request

Actually, you can use these variables to analyze HTTP request and generate the required HTTP response (see action [WebSocket server HTTP response](#)<sup>[506]</sup>).

Please note that if the task is launched otherwise (e.g. manually or by another trigger), all these variables will contain empty values.

**Note:** *This feature is available for Business license only*



### WebSocket server.

Choose the required WebSocket server from the list. Please note that you cannot choose the server when editing a network task. Instead, you must enter the server ID. You must know the server ID in advance (see *WS Connection manager->Servers->Advanced*). If an invalid ID is specified, an "Invalid WS Server ID" error is returned when executing.

### WS Server Manager

You can view or edit lists of registered Websocket servers and clients (see [WS Connection Manager](#)<sup>497</sup>).

# Part

---



## Appendix

## 6 Appendix

### 6.1 Full list of system variables

#### Example of usage

```
{ActiveRTConnections}
{Add(17,8)}
{AddressOnly(John Smith <j.smith@mydomain.com>)}
{Asc(©)}
{AutomationStatus}

{BatteryLife}

{BatteryPower}

{CapsLockState}
{Chr(169)}

{ClipboardContent}
{CommaSeparatedToText("item 1","item 2","item 3")}

{ComputerName}
{CPULoad}
{CurrentWindowCaption}

{CurrentWindowClass}

{CurrentWindowHandle}
{Date}
{DateIncDays(-7)}
{DateTime}
{DateTimeToFormat({Date},yyyy-mm-dd)}

{Day}
{DayOfTheYear({Date})}

{DayOfWeek}
{DayOfWeekNo}
{DiskLoad(c:\)}
{Divide(36,5)}
{DocumentsDir}
{DownloadsDir}
{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)}
```

#### Description

```
Returns active RoboTask connections as JSON
Adds the specified numbers
Returns address only from full address
Returns ASCII code of specified character.
Returns automation status of RoboTask (TRUE or FALSE)
The number of seconds of battery life remaining.
The percentage of full battery charge remaining.
Checks CapsLock state
Returns the symbol with specified ASCII code.
Clipboard content as Text
Converts commaseparated string to multiline text
Computer name
CPU load in percent
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
Returns formatted date and time in specified format (*)
Current day
Returns the day of the year of specified or current date
Current day of week
Current day of week number
Returns the disk load in percents
Divides the first number by the second
Returns user documents folder
Returns user downloads folder
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
```

<code>{ExtractFilePathNoDrive(C:\Folder\File.ext)}</code>	Extracts file path from root folder
<code>{FileAttributes(C:\AUTOEXEC.BAT)}</code>	Returns attributes of file as string, like this: "AHRs"
<code>{FileCreateTime(c:\autoexec.bat)}</code>	Date and time of file creation
<code>{FileExists({TempDir}\file.tmp)}</code>	Check file existence
<code>{FileLastAccessTime(c:\autoexec.bat)}</code>	Date and time of the last file access
<code>{FilesCount({TempDir})}</code>	Returns the amount of files in specified folder
<code>{FileSize(c:\autoexec.bat)}</code>	File size in bytes
<code>{FileTime(c:\autoexec.bat)}</code>	Returns the time of last update of the file
<code>{FolderExists({TempDir})}</code>	Checks folder existence
<code>{FoldersCount({TempDir})}</code>	Returns the amount of folders in specified folder
<code>{FolderSize({TempDir})}</code>	Returns the total size of files in specified folder (in bytes)
<code>{FolderTime({TempDir})}</code>	Returns the time of last update of the folder
<code>{FormatToDateTime(2014-02-19,yyyy-mm-dd)}</code>	Returns standard date and time from formatted string (*)
<code>{FreeDiskSpace(c:)}</code>	Free disk space in bytes
<code>{GetCurrentFolder}</code>	Returns full path of current folder
<code>{GetTaskCurrentStatus(&lt;ExtName&gt;)}</code>	Returns task status as number by ID or External name. 0-Manual, 1-Idle, 2-Busy, 3-Running, 4-Stopping
<code>{GetTaskExtNameById({TaskId})}</code>	Returns the task external name by task ID
<code>{GetTaskIdByExtName({TaskExtName})}</code>	Returns the task ID by external name
<code>{GetTaskLastError(&lt;ExtName&gt;)}</code>	Returns last task error message by external name or task ID
<code>{GetTaskLastFinish(&lt;ExtName&gt;)}</code>	Returns last task finish time by external name or task ID
<code>{GetTaskLastStartReason(&lt;ExtName&gt;)}</code>	Returns the task last start reason by external name or task ID
<code>{GetTaskLastStartTime(&lt;ExtName&gt;)}</code>	Returns the task last start time by external name or task ID
<code>{GetTaskLastSuccess(&lt;ExtName&gt;)}</code>	Returns last task finish state by external name or task ID. 0 - not started, 1 - success, 2 - not success
<code>{GetTaskLastSuccessFinish(&lt;ExtName&gt;)}</code>	Returns last success finish time by external name or task ID
<code>{GetTaskResult(&lt;ExtName&gt;)}</code>	Returns the result of task execution by external name or task ID
<code>{GMTDateTime({DateTime})}</code>	Returns GMT date and time
<code>{Hours}</code>	Current hours
<code>{HttpClientIP}</code>	Returns client's IP address. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpCookie(CookieName)}</code>	Returns the cookie value of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpFilename}</code>	Returns the file name of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.



<code>{HTTPHeader(Header-Name)}</code>	Returns the header of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpMethod}</code>	Returns the method of an HTTP request (GET, POST, etc.). Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpPath}</code>	Returns the path of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpPostParam(ParamName)}</code>	Returns the post parameter value of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpQuery}</code>	Returns the query string of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpQueryParam(ParamName)}</code>	Returns the query parameter value of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{HttpRAWHeaders}</code>	Returns RAW headers of an HTTP request. Works only with the "WebSocket Server HTTP On Request" trigger.
<code>{IncDays(-1)}</code>	Current day ± number of days
<code>{IncMonths(-1)}</code>	Current month ± number of months
<code>{IncYears(-1)}</code>	Current year ± number of years
<code>{IPAddress({ComputerName})}</code>	Returns IP address by computer name
<code>{IsError}</code>	Returns "True" when error occurred at previous step
<code>{IsFileFree(c:\autoexec.bat)}</code>	Checks file for free access
<code>{IsProcessExists(notepad.exe)}</code>	Checks existence of process in memory
<code>{IsWindowActive(Notepad)}</code>	Check if the window is active
<code>{IsWindowHidden(Notepad)}</code>	Returns "True" if window invisible
<code>{IsWindowMaximized(Notepad)}</code>	Returns "True" if window maximized
<code>{IsWindowMinimized(Notepad)}</code>	Returns "True" if window minimized
<code>{IsWindowVisible(Notepad)}</code>	Returns "True" if window visible
<code>{IsWSClientActive(ID)}</code>	Returns if WebSocket client is active
<code>{IsWSServerActive(ID)}</code>	Returns if WebSocket server is active
<code>{LastDateOfTheMonth({Date})}</code>	Returns the last date of the month of specified date
<code>{LastErrorCode}</code>	Error code. Parameter of error handling task.
<code>{LastErrorDescription}</code>	Error description. Parameter of error handling task.
<code>{LastErrorStep}</code>	Error step number. Parameter of error handling task.
<code>{LastErrorTaskExternalName}</code>	External name of error task. Parameter of error handling task.
<code>{LastErrorTaskID}</code>	Error task ID. Parameter of error handling task.
<code>{LastErrorTaskName}</code>	Error task name. Parameter of error handling task.

{LogFileName}	Log file name (only works in the context of a task)
{LogFolder}	Folder of logs
{LowerCase(Sample String)}	Converts a string to lowercase.
{MemoryLoad}	Memory load in percent
{Minutes}	Current minutes
{MonitorCount}	Number of Monitors
{MonthIncDays(-1)}	Current month $\pm$ number of days
{MonthName}	Current month name
{MonthNo}	Current month number
{MouseLeftBtn}	The status of LEFT mouse button
{MouseMiddleBtn}	The status of MIDDLE mouse button
{MouseRightBtn}	The status of RIGHT mouse button
{MouseX}	X-pos of mouse pointer
{MouseY}	Y-pos of mouse pointer
{mSeconds}	Current milliseconds
{Multiply(3,4)}	Multiplies the specified numbers
{NumLockState}	Checks NumLock state
{OnBattery}	True - if computer on battery; False - power is online
{PicturesDir}	Returns user pictures folder
{ProcessID}	Self process ID of RoboTask
{RandomString(32)}	Returns random string of the specified length (default length 32)
{RandomValue(1000)}	Returns random integer value between 0 and parameter (default 1000)
{RoboTaskVersion}	Returns RoboTask product version
{ScreenBottom(0)}	Bottom edge of specified screen
{ScreenH(0)}	Height of specified screen
{ScreenHeight(0)}	Height of specified screen
{ScreenLeft(0)}	Left edge of specified screen
{ScreenRight(0)}	Right edge of specified screen
{ScreenTop(0)}	Top edge of specified screen
{ScreenW(0)}	Width of specified screen
{ScreenWidth(0)}	Width of specified screen
{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
{StartTime}	RoboTask start date and time
{StringLength(Sample String)}	Returns the length of specified string.
{Subtract(24,7)}	Subtracts the second number from the first
{SystemDir}	Windows system directory
{TaskCategoryID}	Category ID of current task (only works in the context of a task)
{TaskComment}	Task comment (only works in the context of a task)
{TaskExtName}	External name of current task (only works in the context of a task)
{TaskFileByID({TaskId})}	Task filename by ID

{TaskFileName}	Task file name (only works in the context of a task)
{TaskFolder}	Folder of tasks
{TaskID}	ID of current task (only works in the context of a task)
{TaskLogFileById({TaskId})}	Task log filename by ID
{TaskName}	Name of current task (only works in the context of a task)
{TaskStatus}	Returns task status as number (works only in task context). For internal use
{TempDir}	Temporary directory
{TextFile(file_name)}	Contents of text file
{TextFileANSI(file_name)}	Reads the text file as ANSI text and returns the text
{TextFileUNICODE(file_name)}	Reads the text file as UNICODE text and returns the text
{TextFileUTF8(file_name)}	Reads the text file as UTF8 text and returns the text
{TextToCommaSeparated(Line 1{EOL}Line 2)}	Converts multi-line text to comma separated string
{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.
{UpTime}	Up time of RoboTask
{UrlFilename(URL)}	Return the file name of a specified URL
{UrlHost(URL)}	Returns the host of a specified URL
{UrlPath(URL)}	Returns the path of a specified URL
{UrlPort(URL)}	Returns the port of a specified URL
{UrlProtocol(URL)}	Returns the protocol of a specified URL
{UrlQuery(URL)}	Returns the query part of a specified URL
{UrlQueryParam(URL, ParamName)}	Returns the query parameter value of a specified URL
{UserName}	User name
{UTF8Decode(<Put UTF8 string here>)}	Decodes string from UTF8 format
{UTF8Encode(           !)} !)	Encodes string to UTF8 format
{ValidFileName(c: "My<Invalid:File*Name>.txt")}	Returns valid file name
{ValidVariableName(Some wrong String)}	Returns valid variable name from custom string
{VolumeLabel(c:\)}	Returns the label of specified volume
{WeekOfTheYear({Date})}	Returns the week of the year of specified or current date
{WinDir}	Windows directory
{WindowCaption({CurrentWindowHandle})}	Returns the title (caption) of specified window handle
{WindowClass({CurrentWindowHandle})}	Returns the class name of specified window handle
{WindowExists(Notepad)}	Check window existence
{WorkingSet({ProcessID})}	Memory working set of process (KBytes)
{Year}	Current year
{YearIncDays(-1)}	Current year ± number of days

**(\*) Format options for format variables:**

<b>d</b>	Displays the day as a number without a leading zero (1-31).
<b>dd</b>	Displays the day as a number with a leading zero (01-31).
<b>mm</b>	Displays the month as a number with a leading zero (01-12). If the mm specifier immediately follows an h or hh specifier, the minute rather than the month is displayed.
<b>yy</b>	Displays the year as a two-digit number (00-99).
<b>yyyy</b>	Displays the year as a four-digit number (0000-9999).
<b>h</b>	Displays the hour without a leading zero (0-23).
<b>hh</b>	Displays the hour with a leading zero (00-23).
<b>n</b>	Displays the minute without a leading zero (0-59).
<b>nn</b>	Displays the minute with a leading zero (00-59).
<b>s</b>	Displays the second without a leading zero (0-59).
<b>ss</b>	Displays the second with a leading zero (00-59).
<b>am/pm</b>	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'am' for any hour before noon, and 'pm' for any hour after noon. The am/pm specifier can use lower, upper, or mixed case, and the result is displayed accordingly.
<b>a/p</b>	Uses the 12-hour clock for the preceding h or hh specifier, and displays 'a' for any hour before noon, and 'p' for any hour after noon. The a/p specifier can use lower, upper, or mixed case, and the result is displayed accordingly.

## 6.2 Command Line Parameters

**Command-line options for *RoboTask.EXE*:*****/ASADMIN***

Start RoboTask with full rights. For operating systems with UAC (i.e. Windows Vista and later).

***/REGSERVER***

Register built-in COM server in your system.

***/UNREGSERVER***

Unregister built-in COM server from your system.

**Command-line options for *RTServiceGuard.exe******/INSTALL***

Install RoboTask service

### **/UNINSTALL**

Remove RoboTask service

### **Command-line option for *InputRegKey.EXE*:**

#### **activate <your\_actual\_license\_key>**

Silent activation if no error. Otherwise it shows the alert with error message.

### **Command-line option for *RoboTaskRuntime.exe***

RoboTaskRuntime.exe <Task\_File\_Name> [/stop]

**<Task\_File\_Name>** - full name of task file

**/stop** - (optional parameter) if you want to stop the application before exiting anyway to read the log

## **6.3 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. You can use it in external scripts by using the name ***RoboTask.App***. For example:

#### For VB script

```
set obj = CreateObject("RoboTask.App")
```

#### For JS Script

```
var obj = new ActiveXObject("RoboTask.App");
```

### **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.DesktopAlert(ByVal Title as String, ByVal Msg as String, ByVal IconKind as Long, ByVal AlertKind as Long, ByVal Duration as Long, ByVal Width as Long, ByVal Height as Long, ByVal Corner as Long)**

Method.

Shows desktop alert or balloon of tray icon with specified text

*Title* - [in] Title of the alert

*Msg* - [in] Message of the alert

*IconKind* - [in] The number of icon

0 - Information

1 - Warning (Exclamation)

2 - Error

3 - Default icon. When *IconKind* contains incorrect number the method shows default icon

*AlertKind* - [in] 0 - Alert window; 1 - Tray icon balloon.

*Duration* - [in] Duration in seconds

*Width* - [in] Width in pixels. If the value less than 330 the method uses default value (330).

This parameter is being ignored if *AlertKind*=1

*Height* - [in] Width in pixels. If the value less than 76 the method uses default value (76). This parameter is being ignored if *AlertKind*=1

*Corner* - [in] The corner of alert window (This parameter is being ignored if *AlertKind*=1):

0 - top-left

1 - top-right

2 - bottom-left

3 - bottom right

### **RoboTaskApp.EventInfoEnabled(ByVal TaskNum as Long, ByVal EventNum as Long)**

Read-only property. Value type - boolean.

Returns event status (enabled or disabled) by number of the task and event number.

*TaskNum* - [in] task number

*EventNum* - [in] event number

### **RoboTaskApp.EventInfoEventID(ByVal TaskNum as Long, ByVal EventNum as Long)**

Read-only property. Value type - String.

Returns event identifier by number of the task and event number. For example:

E\_NET\_LISTENER, E\_WINDOW\_WATCHER, E\_GENERAL\_CYCLIC, etc.

*TaskNum* - [in] task number

*EventNum* - [in] event number

### **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.SetThreadId(By Val ID as long)**

Method

This is an auxiliary method that allows accessing local task variables from an external script that uses the *RoboTask.App* COM object (see the methods *ExpandText*, *SetUserVariable*, *AddUserVariable*, *DelUserVariable*). If you are using actions such as [JS script](#)<sup>[430]</sup>, [JS Evaluate](#)<sup>[432]</sup>, [VB Script](#)<sup>[434]</sup>, [VB Evaluate](#)<sup>[437]</sup>, you can use the built-in *RoboTaskApp* object, which is already properly formed. The use of this method is not required.

When using [Python Script](#)<sup>[413]</sup>, [JavaScript/Node.js script](#)<sup>[419]</sup>, use the *RoboTask* object. Examples are given [here](#)<sup>[416]</sup> and [here](#)<sup>[422]</sup>. The use of this method is not required.

However, the [PowerShell script](#)<sup>[426]</sup> action does not contain a built-in object for accessing RoboTask functions. You must create an object inside the script and initialize it with the correct *ThreadID*. An example is given [here](#)<sup>[426]</sup>.

For a PowerShell script, the current ThreadId of the task is passed through the `ROBOTASK_SCRIPT_ID` environment variable.

Example of the script:

```
$rt = New-object -ComObject RoboTask.App
#write-host $env:ROBOTASK_SCRIPT_ID
$rt.SetThreadId($env:ROBOTASK_SCRIPT_ID)
$rt.WriteToLog(0, "Hello", 0)
```

If you want to run an external script using the [Run Console Application](#)<sup>[80]</sup> or [Run Program/Open Document](#)<sup>[79]</sup> actions, you can obtain the task's ThreadID using the `{TaskThreadId}` variable and pass this value as a parameter via environment variable, parameter, or file.

*ID* - Task thread ID

**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.StepInfoActionID(ByVal TaskNum as long, ByVal StepNum as long)**

Read-only property. Value type - string.

Returns the action identifier by task number and the step number.

Returned value is an action type identifier (for example: A\_TASKS\_START, A\_TASK\_WAITCOMPETE, A\_GENERAL\_RUN\_PROG, etc.)

*TaskNum* - [in] task number

*StepNum* - [in] step number. Step number begins from 0.

**RoboTaskApp.StepInfoName(ByVal TaskNum as long, ByVal StepNum as long)**

Read-only property. Value type - string.

Returns step name by task number and the step number.

*TaskNum* - [in] task number

*StepNum* - [in] step number. Step number begins from 0.

**RoboTaskApp.StepInfoEnabled(ByVal TaskNum as long, ByVal StepNum as long)**

Read-only property. Value type - boolean.

Returns step status (enabled or disabled) by task number and the step number.

*TaskNum* - [in] task number

*StepNum* - [in] step number. Step number begins from 0.

**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. Step number begins from 0.

**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.TaskInfoExtname(ByVal NumTask as Long)**

Read-only property, value type - string

Returns External name of the task by number of task (NumTask)

*NumTask* - [in] task number

Example of usage (Basic):

```
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
        extNam = RoboTaskApp.TaskInfoExtName(i)
        nam = RoboTaskApp.TaskInfoName(i)
        id = RoboTaskApp.TaskInfoID(i)
        RoboTaskApp.WriteToLog(3, extNam+"; "+nam+"; "+Str(id), RoboTaskApp.ThreadID)
    Next
End Sub
```

**RoboTaskApp.TaskInfoLastStart(ByVal NumTask as Long)**

Read-only property, value type - DateTime

Returns the date and the time of the last running of a task. If the task has not been running from the moment of start of the RoboTask, the result will be zero.

*NumTask* - [in] task number

**RoboTaskApp.TaskInfoName(ByVal NumTask as Long)**

Read-only property, value type - string

Returns the name of the task by number of task (NumTask)

*NumTask* - [in] task number

**RoboTaskApp.TaskInfoID(ByVal NumTask as Long)**

Read-only property, value type - long integer

Returns folder ID of the task by number of task (NumTask)

*NumTask* - [in] task number

**RoboTaskApp.TaskInfoFolderID(ByVal NumTask as Long)**

Read-only property, value type - long integer

Returns ID of the task by number of task (NumTask)

*TaskNum* - [in] task number

**RoboTaskApp.TaskInfoState(ByVal NumTask as Long)**

Read-only property, value type - long integer  
Returns status of the task by number of task (NumTask)  
Possible returned values are:

- 0 - (tsManual) 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.
- 1 - (tsIdle) the task has some enabled triggering events. It waits to be run automatically
- 2 - (tsDisabled) the task is disabled while it is being edited.
- 3 - (tsRunning) the task has been launched and it is running.
- 4 - (tsStopping) the task received the stop command, but it is yet running. After the task is completely stopped, its status becomes either tsManual or tsIdle.

*TaskNum* - [in] task number

### **RoboTaskApp.TaskNumByExternalName(ByVal ExternalName as String)**

Read-only property, value 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
```

## Deprecated methods

These methods are not preferred, because they are not supported by scripting languages: Javascript and VB Script.

### **RoboTaskApp.EventInfo(ByVal TaskNum as Long, ByVal EventNum as Long, EventID as String, Enabled as Boolean)**

Method. *(deprecated 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.StepInfo(ByVal TaskNum as long, ByVal StepNum as long, ActionID as String, Enabled as Boolean, Name as String)**

Method. *(deprecated 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.TaskInfo(By Val NumTask as Long, ExtName as String, Name as String, ID as Long)**

Method. *(deprecated method)*

Sets the properties of the task with the specified number ("*NumTask*").

*NumTask* - [in] task number

*ExtName* - [out] 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 (*deprecated 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.

*NumTask* - [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*.

# Index

## - A -

### Actions

- Add key 288
- Break 245
- Browse for Computers 105
- Browse for Folders 103
- Change Folder 128
- Check connection 207
- Check for E-mail 146
- Check for Window 254
- Check HTTP address 153
- Color Compliance 385
- comma-separated 336
- comment 96
- Continue 245
- Copy/Move File 117
- Create Folder 128
- Create Variable 208
- Dataset Loop 236
- Delete File 120
- Delete key 289
- Delete Value 292
- Delimit variable 211
- Dial Up/Hang Up 206
- Disable Automation 276
- Download Files 151
- Else 247
- Elseif Then 248
- Enable Automation 276
- Enable Task 214, 215
- End If 247
- End Loop 244
- Excel Run Macro 444
- Exit RoboTask 276
- Export 294
- Extract Files 144
- File Attributes 137
- File Loop 229
- FTP Change Folder 163
- FTP Command 166
- FTP Create Folder 164
- FTP Delete File 163
- FTP Download 161
- FTP File Loop 166
- FTP Log Off 161
- FTP Log On 160
- FTP PWD 165
- FTP Remove Folder 164
- FTP Rename File 162
- FTP Upload 161
- If Then 245
- Import 295
- Increment Variable 211
- Input box 106
- Lock Workstation 273
- Log Message 276
- Log Off 89
- Minimize RoboTask 276
- multiline text 336
- Open Dialog 100
- Pack Files 142
- Pause 90
- Ping Host 152
- Print Document 83
- query 91
- Question Dialog 99
- Read Value 290
- Remove Folder 129
- Remove Variable 210
- Rename File 119
- Rename Folder 129
- Rename key 289
- Rename Value 293
- Restore RoboTask 276
- Run as User 273
- Run Command-Line Utility 80
- Run Program/Open Document 79
- Save Dialog 102
- Screen Pixel 384
- Screenshot 383
- Search for Files 123
- Send command 297
- Send Email 147
- Send Keystrokes 84
- Send Network Message 298
- Set Variable 209
- Shif Keys State 88
- Show Message/Play Sound 97
- Simple Loop 240
- Sort Text 135
- Split text 211

## Actions

- Start Service 270
- Start Task 215
- Stop Service 271
- Stop Task 217
- Synchronize Folders 130
- Task status 219
- Terminate Process 95
- Text Loop 234
- Time difference 212
- Transfer Type 165
- TXT Comma-Text Conversion 336
- Unique File Name 136
- Wait for File 134
- Wait for pixel 386
- wait for process finish 93
- wait for process start/finish 94
- Wait for Task 218
- Window Command 252
- Window Loop 239
- Window Pixel 381
- Window position 259
- Window size 259
- Window snapshot 376
- Write Text File 121
- Write to Event Log 275
- Write Value 291
- administrative commands 276
- alarm clock 517
- archiving
  - files 142
- Array elements 214
- Ascending 135
- assistance
  - getting 76
- automation
  - disabling 276
  - enabling 73, 276

**- B -**

## Backup

- application tasks 75
- service tasks 28

## Basic

- RoboTaskApp object 562
- script 562

Basic script 373

## Battery

- AC power restored 539
- Switching to battery 538
- battery power
  - triggering event 534
- beginning
  - RoboTask 3
- Break 245
- breaking
  - loops 245
- browse dialog box
  - displaying 103, 105
- bugs
  - reporting 76
- buying
  - RoboTask 6

**- C -**

- Catch the output text 80
- Change Folder 195
- changing 128
  - directories 128
  - folders 128
- Check connection 207
- Check email 308
- checking
  - e-mail 146
  - for windows 254
- choice 109, 520
- Clear Clipboard 279
- Click the button 284
- Client-Server 42
- Clipboard 542
  - copy 278
  - Cut 278
  - get 279
  - monitor 517
  - paste 279
  - put 280
  - Put Image 281
  - Save Image 281
- Clipboard
  - clear 279
- Close session 513
- closing
  - windows 252
- Code page 140

- Color 392
  - Color Compliance 385
  - commands
    - using ftp 166
  - comma-separated string 211, 214, 336
  - comment 96
  - computer display
    - triggering event 536
  - computer idle
    - triggering event 536
  - computers
    - locking 273
    - shutting down 89
  - conditional statements
    - using 245, 247, 248
  - Connect 220
  - Connect to 44
  - connecting
    - dial-up connections 206
  - Console Application 510
  - Continue 245
  - Conversion
    - comma-text 336
    - UTF8 338
  - Copy 278
  - copying
    - files 117
    - tasks 60
  - CPU 537
  - CPU Load
    - events 537
  - Create Folder 195
  - creating 14, 128
    - directories 128
    - expressions 14
    - file lists 123
    - folders 128
    - tasks 5, 50
    - text files 121
    - variables 14, 208
    - Zip files 142
  - CSV
    - add column 464
    - data loop 475
    - filter data 474
    - information 460
    - merge data 471
    - move column 468
    - open data 458
    - read cells 469
    - remove column 465
    - remove duplicates 477
    - save 479
    - sort 466
    - write cells 470
  - CSV actions 457
  - CSV File 481
  - Custom variables 12
  - customizing
    - options 6
  - Cut 278
  - cyclic
    - triggering event 522
- ## - D -
- Database 91, 449
    - Begin transaction 457
    - close 450
    - Commit 457
    - Exec SQL 450
    - Loop 453
    - open 449
    - Rollback 457
  - datasets
    - looping 236
  - date-time 212
  - DB 449
  - Define connections 44
  - delaying
    - tasks 90
  - delete email 307
  - Delete file 194
  - Delete key 289
  - Delete Value 292
  - Deleted tasks 73
  - deleting 163
    - files 120, 163
    - ftp 163
    - tasks 58
  - Descending 135
  - device changed
    - triggering event 535
  - Dial Up
    - Check connection 207

Dialogs 97, 99, 100, 102, 103, 105, 106, 109, 111, 113, 114

dial-up connection monitor  
triggering event 530

dial-up connections  
making 206

directories  
changing 128, 163  
creating 128, 164  
printing 165  
removing 129, 164  
renaming 129  
synchronizing 130  
transferring 165

disabling  
tasks 73, 215

Disconnect 222

display  
triggering event 536

documents  
opening 79  
printing 83

Download 191

Downloading  
email 306  
files 151  
ftp 161

duplicating  
tasks 60

## - E -

editing  
tasks 6, 58

Else statements  
using 247

Elseif Then  
using 248

e-mail  
checking 146  
sending 147

enabling  
automation 73  
tasks 73

End If statements  
using 247

End Loop 244

error

handling 41  
processing 41  
raise 251

Evaluate  
JS expression 432  
VB expression 437

Event log 275

events  
Cyclic 522  
Dial-Up Connection Monitor 530  
Expression Monitor 531  
File Monitor 524  
Hot Keys 520  
On Battery Power is Low 534  
On Device Changed 535  
On Display Changed 536  
On Idle 536  
On Log Off 536  
On Palette Changed 536  
On Resume 537  
On Spooler Changed 535  
On Time Changed 534  
Scheduler 517  
Tray Icon 521  
Window Watcher 531

Excel  
macro 444

exiting  
RoboTask 60, 276

Export Registry 294

Expression Monitor  
triggering event 531

expressions 210  
creating 14, 208  
deleting 210  
modifying 71  
overview 10  
removing 210  
setting 209  
starting 215

## - F -

File Attributes 137

file lists  
creating 123

File Loop 197, 229

file loops



- file loops
    - ftp 166
  - file monitor
    - triggering event 524
  - File Permissions 204
  - files 117, 142, 144
    - archiving 142
    - copying 117
    - creating 121
    - deleting 120, 163
    - downloading 151, 161
    - extracting 144
    - looping 229
    - moving 117
    - renaming 119, 162
    - searching 123
    - synchronizing 130
    - transferring 165
    - unzipping 144
    - uploading 161
    - zipping 142
  - Fill
    - Web Form 342
  - Fill web form 341
  - Finish task 251
  - focusing
    - windows 252
  - folders
    - changing 128, 163
    - creating 128, 164
    - printing 165
    - removing 129, 164
    - renaming 129
    - synchronizing 130
    - transferring 165
  - ftp
    - changing folders 163
    - creating folders 164
    - deleting 163
    - download 161
    - file loops 166
    - log off 161
    - log on 160
    - PWD 165
    - removing folders 164
    - renaming files 162
    - transfer 165
    - uploading 161
    - using commands 166
  - FTP File Loop 166
- ## - G -
- Get Text 279
  - getting started 3
  - Graphics actions 486
- ## - H -
- Help
    - opening 76
  - hiding
    - windows 252
  - hosts
    - pinging 152
  - hot keys
    - triggering event 520
  - HTTP
    - downloading 151
    - request 153
    - response 153
    - result 153
- ## - I -
- idle
    - triggering event 536
  - If Then statements
    - using 245
  - Image
    - add border 493
    - crop 489
    - Draw Image 492
    - draw text 490
    - Generate QR 494
    - open 486
    - resize 488
    - rotate 490
    - save 487
    - Scan QR 495
  - IMAP 312, 313, 314, 317, 319, 320, 321
    - close connection 313
    - delete marked messages 320
    - download message 317
    - folder info 321
- 
- © 2004-2023 Neowise software

IMAP 312, 313, 314, 317, 319, 320, 321  
   get flags 319  
   mail folder 313  
   mail loop 314  
   move message 320  
   open connection 312  
   set flags 319  
 Import Registry 295  
 Incoming email 301, 303, 306, 307, 308  
 increment 211  
 INI Files 394, 395, 397, 398, 400  
 Input box 106  
 Input dialog 114  
 Intersection 339  
 introduction 2  
 IsError 41

## - J -

Java script 419  
 Javascript 363, 430, 432, 562  
 jobs 215  
   adding variables 14  
   creating 5, 50  
   creating shortcuts 60  
   deleting 58  
   disabling 73, 215  
   duplicating 60  
   editing 6, 58  
   running 6, 58  
   starting 214, 276  
   stopping 58, 215, 276  
 JS Evaluate 432  
 JSON 16  
 JSON array 214

## - K -

keyboard  
   hold down keys 88  
   release keys 88  
   strokes for 84  
 keystrokes  
   sending 84, 88

## - L -

LastErrorCode 41  
 LastErrorDescription 41  
 LastErrorStep 41  
 LastErrorTaskExternalName 41  
 LastErrorTaskID 41  
 Legal notice 7  
 Link click 353  
 listener 545  
 Lists  
   Operations 339  
 local variables 12, 21  
 locking  
   computers 273  
 Log Dialog 117  
 log off 191  
   ftp 161  
   triggering event 536  
 Log on 189  
   ftp 160  
 logging  
   messages 276  
 logging off  
   computers 89  
 logoff 89  
 logs  
   viewing 67  
 Loop 303  
   on connections 228  
   on registry 296  
 looping 236  
   datasets 236  
   files 229  
   queries 236  
   simple 240  
   text 234  
   Window 239  
 loops 241  
   breaking 245  
   continuing 245  
   ending 244  
   iterating 166

## - M -

- Macro recorder 18
- Mail loop 308
- maximizing
  - windows 252
- Memory Load
  - events 537
- messages
  - displaying 97
  - logging 276
  - sending 298
- Messages count 308
- minimizing
  - RoboTask 276
  - windows 252
- Mouse 18
  - click 284
  - move pointer 285
  - move to object 282
- Move mouse pointer 285
- Move pointer to object 282
- Move to Object 282
- moving
  - files 117
- Multichoice 111

## - N -

- Network
  - Client 44
  - Connect to 44
  - Control remote 42
  - Defined connections 44
  - listener 545
  - plugin settings 300
  - send command 297
  - Server 43
- Network Client 220, 222, 223, 224, 225, 227, 228
- network messages
  - sending 298
- networking
  - dial-up connections 206
- Node.js Script 419

## - O -

- open dialog box
  - displaying 100
- Open session 511
- options
  - customizing 6
  - setting 67
- overview 2

## - P -

- packing
  - Zip files 144
- Page snapshot 360
- palette
  - triggering event 536
- Parse 308
- Paste 279
- pinging
  - hosts 152
- Pixel 392
- Pop3 Server 301, 303, 306, 307, 308
- port 299
- PowerShell 426
- practical recommendations 22
  - custom variables 12
  - Debugging 21
  - task synchronization 22
  - tasks 20, 22
- print working directory 196
- printing
  - documents 83
  - folders 165
- process 93, 94
- programs
  - running 79
- purchasing
  - RoboTask 6
- Put Image
  - Clipboard 281
- Put Text 280
- Python Script 413

**- Q -**

QR code  
 Scan 495  
 QR core  
 Generate 494  
 queries  
 looping 236  
 query 91  
 querying  
 loops 236  
 question dialog box  
 displaying 99  
 quitting  
 RoboTask 276

**- R -**

RAM 537  
 RDC 36  
 Read from session 514  
 Read Text File 140  
 Read Value 290  
 Record a macro 18  
 Recycle bin 73  
 RegExp 364  
 loop 365  
 match 371  
 replace 367  
 RegExp Com object 373  
 registering  
 RoboTask 6  
 Registry  
 Add key 288  
 Delete key 289  
 Delete Value 292  
 Export 294  
 Import 295  
 Read Value 290  
 Rename key 289  
 Rename Value 293  
 Write Value 291  
 Registry loop 296  
 Regular Expressions 364, 365, 367, 371, 373  
 Remote  
 RoboTask 220, 222, 223, 224, 225, 227, 228  
 task 220, 222, 223, 224, 225, 227  
 remote servers  
 checking 152  
 Remove Folder 196  
 removing 129  
 directories 129  
 folders 129  
 variables 210  
 Rename file 193  
 Rename key 289  
 Rename Value 293  
 renaming 129, 162  
 directories 129  
 files 119, 162  
 folders 129  
 ftp 162  
 reporting  
 bugs 76  
 Resore  
 application tasks 75  
 service tasks 28  
 REST api 156  
 restoring  
 RoboTask 276  
 windows 252  
 RoboTask  
 exiting 276  
 minimizing 276  
 restoring 276  
 stopping 60  
 RoboTask Object  
 for Node.js 422  
 for Python 416  
 internal AciveX 562  
 RoboTask Service  
 Advantages 23  
 Authentication 27  
 configure service tasks 37  
 Disadvantages 23  
 get access 37  
 Install 26, 28  
 Settings 28  
 Start 26, 28  
 Stop 26, 28  
 UNC names 27  
 Run as User 273  
 running  
 tasks 6, 58

## - S -

- save dialog box
  - displaying 102
- Save Image 281
- Scale 73
- schedulers
  - triggering event 517
- Screen Pixel 384
- Screenshot 383
  - Color Compliance 385
  - Screen Pixel 384
  - Window Pixel 381
  - Window snapshot 376
- Script 413
  - JS Evaluate 432
  - JS Script 430
  - Node.js 419
  - PowerShell 426
  - Python Script 413
  - VB Evaluate 437
  - VB Script 434
- searching
  - files 123
- Send command 297
- Send Keystrokes 79
- sending
  - e-mail 147
  - keystrokes 84, 88
- Service
  - start 270
  - status 272
  - stop 271
- Set size of browser window 362
- setting
  - options 67
  - variables 209
- SFTP 189
  - Change Folder 195
  - Create Folder 195
  - Delete file 194
  - Download 191
  - File Loop 197
  - File Permissions 204
  - Log off 191
  - Log on 189
  - print working directory 196
  - PWD 196
  - Remove Folder 196
  - Rename file 193
  - Synchronization 201
  - Upload 192
- SFTP Command 204
- shortcuts
  - creating 60
- Show
  - balloon 106
  - message 106
  - notification 106
- Show Text 113
- shut down
  - triggering event 536
- shutting down
  - computers 89
- Simple Loop 240
- simple loops
  - using 229, 240
- sleep 537
- Sort Text 135
- sound
  - playing 97
- spooler
  - triggering event 535
- SQL
  - loops 236
  - query 91
- SQL Query 481
- SSH FTP 189
- starting
  - services 270
  - tasks 214, 215
- Stop Task 217
- stopping
  - RoboTask 60, 276
  - services 271
  - tasks 58
- String
  - between 328
  - CSV format 331
  - delete 325
  - extract 326, 328
  - Length 330
  - pos 327
  - replace 323
  - trim 324

Strings 323, 324, 325, 326, 327, 328, 330, 331  
 Synchronization 201  
 synchronize tasks 218  
 Synchronize workstations 297, 545  
 synchronizing  
   directories 130  
   files 130  
   folders 130  
 System services 270, 271, 272  
 system time  
   triggering event 534  
 System variables 11, 12, 556  
   displaying 71

## - T -

Task  
   disable 222  
   enable 222  
   start 223  
   status 219, 227  
   stop 224  
   wait 225  
 task synchronization 22  
 tasks 18, 22, 60, 214, 215  
   adding variables 14  
   copying 60  
   creating 5, 18, 50  
   creating shortcuts 60  
   delaying 90  
   deleting 58  
   disabling 73, 215  
   duplicating 60  
   editing 6, 18, 58  
   enabling 73, 214  
   running 6, 58  
   starting 214, 215, 276  
   stopping 58, 215, 217, 276  
   waiting 218  
 TCP port 299  
 technical support 76  
 Terminate Process 95  
 Text 121, 332, 333, 334, 335, 336, 338  
   looping 234  
 Text file 135  
   creating 121  
   read 140  
   write 121

Time difference 212  
 toolbars  
   viewing 67  
 Total size 308  
 transferring  
   files 165  
 Trash bin 73  
 tray icon  
   triggering event 521  
 Triggers  
   Clipboard Monitor 542  
   Cyclic 522  
   Dial-Up Connection Monitor 530  
   Expression Monitor 531  
   File Monitor 524  
   Hot Keys 520  
   On Battery Power is Low 534  
   On CPU load 537  
   On Device Changed 535  
   On Display Changed 536  
   On Idle 536  
   On Log Off 536  
   On memory load 537  
   On Palette Changed 536  
   On resume 537  
   On Spooler Changed 535  
   On Time Changed 534  
   Scheduler 517  
   Tray Icon 521  
   Window Watcher 531

## - U -

unhiding  
   windows 252  
 Union 339  
 Unique File Name 136  
 unzipping  
   files 144  
 Upload 192  
 uploading  
   ftp 161  
 user variables  
   displaying 71  
 user-defined variables 14  
 Using Basic 22  
 UTF8 338  
 Utilities 47

## - V -

- Variable usage 18
- Variables 11, 12
  - adding to tasks 14
  - creating 14, 208
  - deleting 210
  - Delimit variable 211
  - increment 211
  - list 556
  - modifying 71
  - overview 10
  - prefix 16
  - removing 210
  - setting 209
  - Split text 211
  - Time difference 212
- VB Script 434
- viewing
  - logs 67
  - toolbars 67
- Vista 36

## - W -

- Wait
  - for File 134
  - for pixel 386
  - for process finish 93
  - for process start 94
  - for Task 218
  - for text 514
  - for window 256
- Web
  - Document 341, 349, 357
  - Form 341, 342, 349, 357
  - Javascript 363
  - Link 349, 353
  - page 349
- Web browser 341, 342, 349, 353, 357, 360, 362, 363
- WEB service 156
- WebSocket 496
  - Connect/Disconnect client 509
  - Connection manager 497
  - HTTP Response 506

- On HTTP request 553
- On message 550, 551
- Send message 503, 505
- Start/Stop server 509
- WS Client 505, 509, 551
- WS Server 503, 506, 509, 550, 553
- While loop 241
- Window 256
- Window loop 239
- Window pixel 381
- Window snapshot 376
- window watcher
  - triggering event 531
- Windows
  - Change Position 259
  - Change Size 259
  - checking for 254
  - focusing 252
  - hiding 252
  - manipulating 252
  - maximizing 252
  - minimizing 252
  - restoring 252
  - unhiding 252
- Windows 2008 Server 36
- Windows 7 36
- WinPopup
  - sending 298
- Working as system service 35
  - differences 36
  - RDC 36
  - Vista 36
- Write to session 513
- Write Value 291

## - X -

- XML Document 402, 403, 404, 407

## - Z -

- Zip files
  - creating 142
  - extracting 144