**AutoHotkey programming with DRAKON Editor**

**Warning: At the moment code generation for AutoHotkey is on beta stage of development and some details is subject to change!**

This is a short tutorial of using AutoHotkey together with DRAKON Editor. Please note that DRAKON Editor is currently tested only with AutoHotkey\_L .

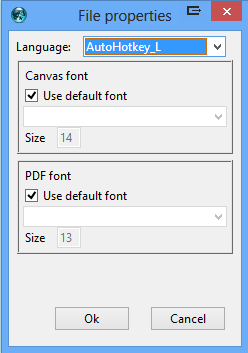
More advanced examples can be found in examples/AutoHotkey subfolder in the DRAKON Editor folder.

**Contents**

* The basics
* Hello, world
* Adding your own code to header and footer to generated file
* Types of diagrams and commenting diagrams
* Function parameters
* Flow control
* Flow control with Fork macroicon
* Flow control with Switch macroicon
* Loops
* Loops with Arrow
* C-style For loop
* Foreach loop
* Debugging tips
* Reserved Labels and Variables

**Hello, world**

1. Create a new file with DRAKON Editor.
2. Select AutoHotkey\_L as the programming language. In the main menu, go to **File / File properties...** and select *AutoHotkey\_L* in the *Language* combo box:

****

1. Create a new diagram called *Hello*:



1. Add your own code. In the main menu, go to **File / File description...** and add the following text:

**=== header ===**

**Hello()**

1. Generate the source code. In the main menu, go to **DRAKON / Generate** code or just press Ctrl-B (Command-B on Mac).

DRAKON Editor will generate a new .ahk file and put it next to the .drn file. The generated code will be equivalent to (**Note that here and after in this document code in examples are not exactly the same that generated code, but both will be logically equivalent**):

**Hello()**

**Hello()**

**{**

**msgbox, Hello, World!**

**return**

**}**

**Adding your own code to header and footer to generated file**

Sometimes it is necessary to add some hand-written code to the generated file. DRAKON Editor allows you to do that using the so called sections in the *File description*.

To edit the file description, go to **File / File description...**

To add some code at the beginning (header) of generated file add this code to *File description*:

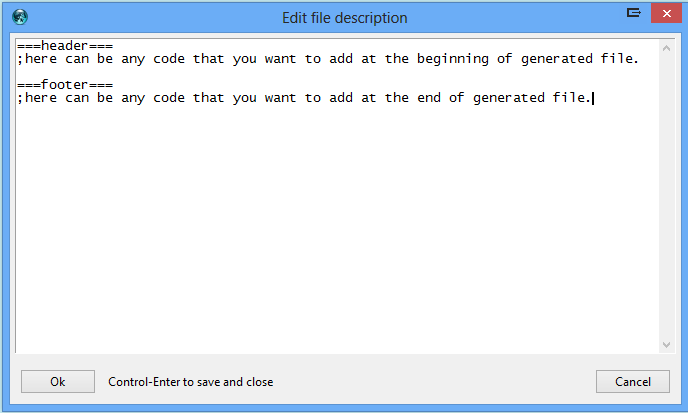
**===header===**

**; Some code here**

To add code at the end (footer) of generated file add this code to *File description*:

**===footer===**

**; Some code here**

For example:

We used the header section in the *Hello, World!* script example above to actually run the function we created.

The text in the file description that goes before any sections is ignored.

In AutoHotkey using footer in almost any cases is not necessary. So best practice is to use only Header and to use it only for calling one diagram that start execution of the script. Avoid writing any other code directly to Header. All code should be in to diagrams. So best *File description* looks like:

**===header===**

**Gosub, Free\_Start**

If you need to use both Header and Footer, best *File description* looks like:

**===header===**

**Gosub, Free\_Start**

**Goto, Footer**

**===footer===**

**Footer:**

**Gosub, Free\_Footer**

**Types of diagrams and commenting diagrams**

Currently for generating AutoHotkey code DRAKON Editor supports three types of diagrams: Free type, Hotkey Type, Function type.

*Free* type of diagram is a diagram that begins with label which has the same name as *Free* type diagram name and ends with **return**. So you can use anywhere in your code **gosub** command to call *Free* type diagram.

To make *Free* type of diagram add **Free\_** (not case sensitive) at the beginning of diagram name.

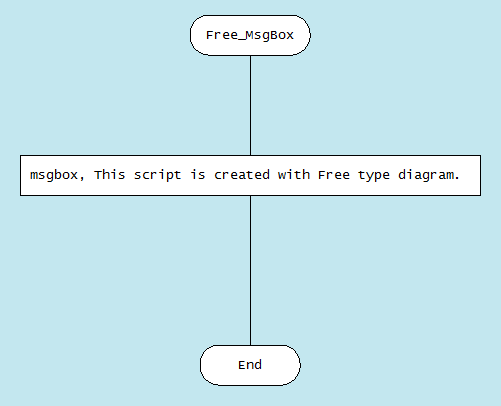
For example, let’s create a script that uses *Free* type diagram and pops up message box “This script is created with Free type diagram.”:

Type in *File description* :

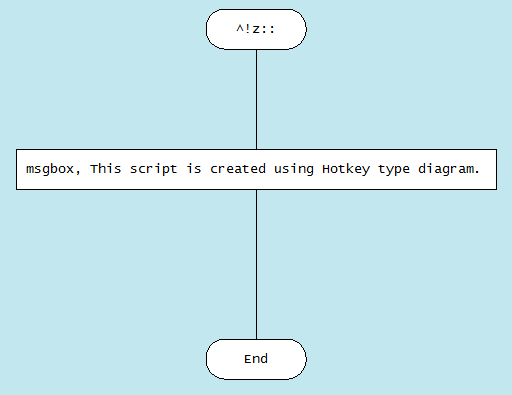
**===header===**

**gosub, Free\_MsgBox**

and create diagram:



To make *Hotkey* type diagram, use hotkey in the format of AutoHotkey for the name of diagram. DRAKON Editor will generate code for *Hotkey* type diagram, which begins with the name of diagram and ends with **return**. For example, for hotkey *Control+Alt+z* use **^!z::** for the name of the diagram.

Let’s create *Hotkey* type diagram, which pops up message box with “*This script is created using Hotkey type diagram.”* when *Control+Alt+z* is pressed:

**Note:** When DRAKON Editor sees **::** (two colons) in the name of the diagram, it generates *Hotkey* type of diagram, so don’t use two colons in the name of other type of diagrams.

Diagram that is commented is excluded from code generation, so code is not generated for that diagram. To make any diagram commented, type **Comment\_** (not case sensitive) in the beginning of the name of the any type of diagram. For example, the diagram that has name **Comment\_Free\_Start** is commented and will not be generated as AutoHotkey code. Any type of diagram can be commented.

Diagrams that have no **Free\_** or **Comment\_** at the beginning of the name or **::** (two colons) in the name, will be generated as *Function* type diagrams. Code generated from *Function* type diagram will be placed inside function with the same name as diagram name. So *Function* type diagram name and function generated from it will have same name. Example of *Function* type diagram is *Hello, World!* script that we have created before.

**Advices:**

1. Use *Hotkey* type diagram when you are creating hotkey.
2. Use *Function* type diagram everywhere when possible.
3. Use *Free* type diagram only in case when it is not possible to place part of the code inside function.

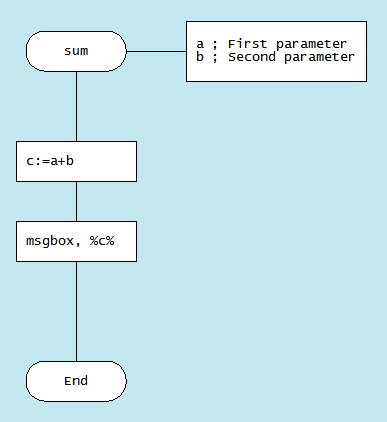
**Function parameters**

The *Function* type diagram can have parameters for function. The parameters of the function are placed in the special *Parameters* icon, to the right from the *Begin* icon.

How to create a Parameters icon:

1. Add a new *Action* icon and put it next to the right side of the *Begin* icon.
2. Connect the *Begin* icon with the new *Action* icon with a horizontal line.
3. Add parameters, one parameter per line.

For example, let’s create script that adds two parameters and pops up sum in message box:



The equivalent AutoHotkey code for that diagram is:

**Sum(1,2)**

**Sum(a,b)**

**{**

**c:=a+b**

**msgbox, %c%**

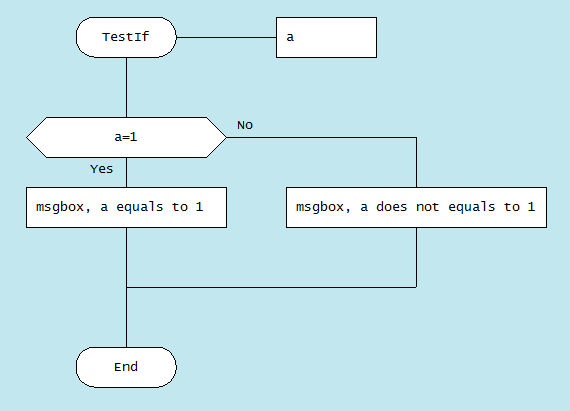
**return**

**}**

**Note:** Function parameters in DRAKON Editor can have comments. For making comments after parameter use **;** (semicolon) and after write a comment.

**Flow control with Fork macroicon**

The If (question) icon produces the if statement:



The equivalent AutoHotkey code for that diagram is:

**TestIf(a)**

**{**

**If (a = 1)**

**{**

**msgbox, a equals to 1**

**}**

**Else**

**{**

**msgbox, a does not equals to 1**

**}**

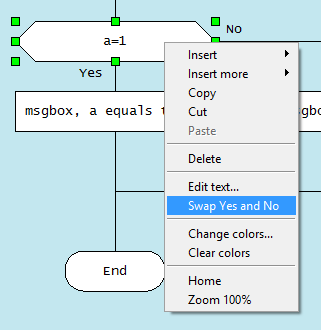
**return**

**}**

**Important!** The left and the right branches of an *If* icon are not equal in DRAKON.

**Rule: the further to the right, the “worse”.** It means that the happy path should go straight down. If we have an *If* icon and one of the outcomes is “worse” than the other, the “worst” one must go to the right.

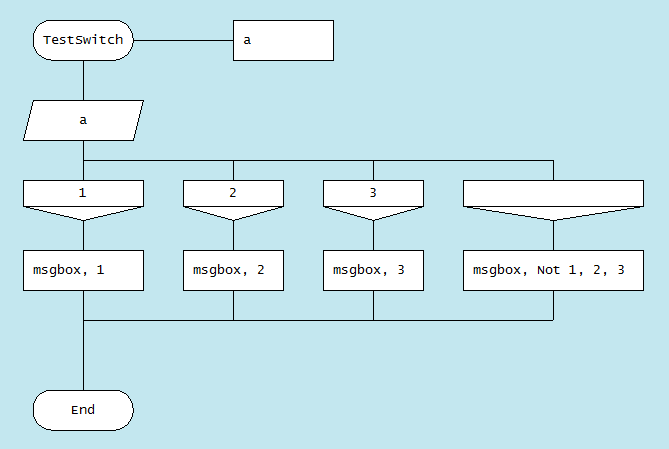
It is possible to swap *YES* and *NO* exits of an *If* icon. Right-click on the *If* icon and choose **Swap YES and NO** :



**Flow control with Switch macroicon**

Imagine you need to make a choice out of several options based on some value. In other words, there is a question that has several answers. In DRAKON, you can use the Select (Choice) and Case icons. Switch macroicon has two modes.

In First mode, left part of condition write in *Select* icon and right part of condition in *Case* icons. Between left and right part **=** sign is automatically inserted in code generation process. For example:



The equivalent AutoHotkey code for that diagram is:

**TestSwitch(a)**

**{**

**If (a=1)**

**{**

**msgbox, 1**

**}**

**Else If (a=2)**

**{**

**msgbox, 2**

**}**

**Else If (a=3)**

**{**

**msgbox, 3**

**}**

**Else**

**{**

**msgbox, Not 1, 2, 3**

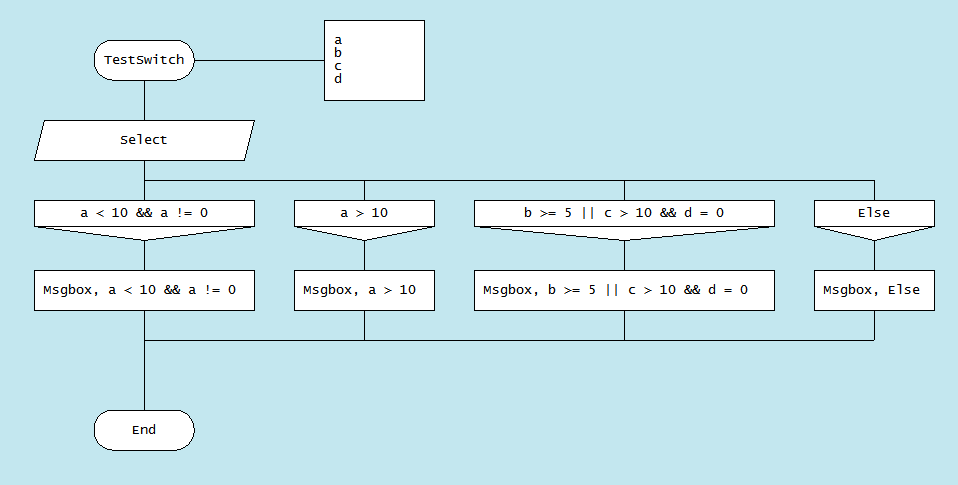
**}**

**return**

**}**

**Note:** Empty *Case* icon means *else*. To give icon *else* meaning you can also write in *Case* icon **Else** (not case sensitive).

In Second mode, in *Select* icon write first word **Select** (not case sensitive). So text in *Select* icon can be **Select** or **select** or **SelectFrom** or **Select From** or some other variant. In *Case* icon, write condition. Condition can have any complexity. For example:



The equivalent AutoHotkey code for that diagram is:

**TestSwitch(a)**

**{**

**If (a < 10 && a != 0)**

**{**

**msgbox, a < 10 && a != 0**

**}**

**Else If (a > 10)**

**{**

**msgbox, a > 10**

**}**

**Else If (b >= 5 || c > 10 && d = 0)**

**{**

**msgbox, b >= 5 || c > 10 && d = 0**

**}**

**Else**

**{**

**msgbox, Else**

**}**

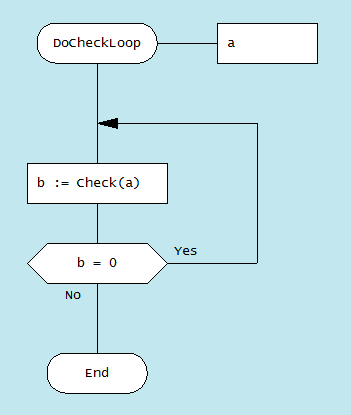
**return**

**}**

**Loops with Arrow**

When an *Arrow* is added to an *If* or *Select*, a loop is created.

1. Do-Check loop:



The equivalent AutoHotkey code for that diagram is:

**DoCheckLoop(a)**

**{**

**LpStart:**

**b := Check(a)**

**If (b = 0)**

**{**

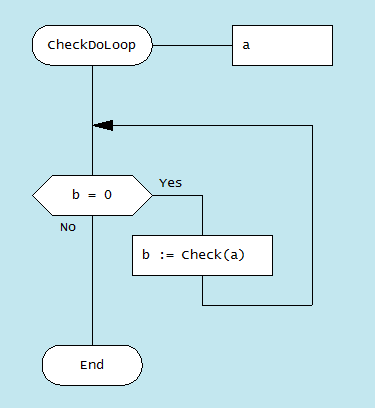
**Goto, LpStart**

**}**

**return**

**}**

1. Check-Do loop:



The equivalent AutoHotkey code for that diagram is:

**CheckDoLoop(a)**

**{**

**LpStart:**

**If (b = 0)**

**{**

**b := Check(a)**

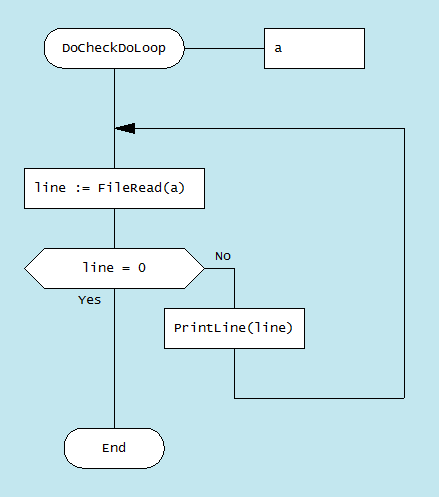
**Goto, LpStart**

**}**

**return**

**}**

1. Do-Check-Do (hybrid) loop:



The equivalent AutoHotkey code for that diagram is:

**DoCheckDoLoop(a)**

**{**

**LpStart:**

**line := FileRead(a)**

**If (line = 0)**

**{**

**}**

**Else**

**{**

**PrintLine(line)**

**Goto, LpStart**

**}**

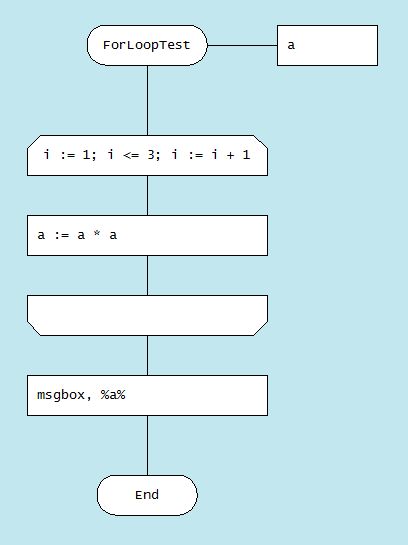
**return**

**}**

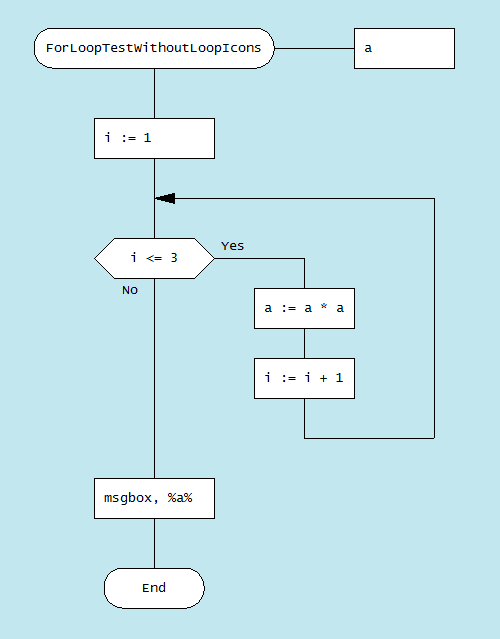
**C-style For loop**

There is no C-style *For* loopin AutoHotkey, but with DRAKON Editor, it is possible to write C-style *For* loops in AutoHotkey.

Example of *For* loop:



Which is a “sugared” version of the below diagram:



The equivalent AutoHotkey code for last two diagrams is:

**ForLoopTest(a)**

**{**

**i := 1**

**LpStart:**

**If (i <= 3 )**

**{**

**a := a \* a**

**i := i + 1**

**Goto, LpStart**

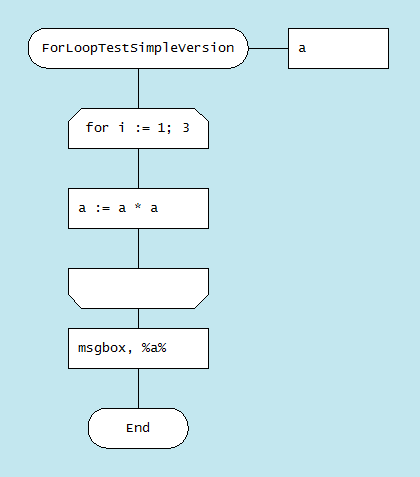
**}**

**msgbox, %a%**

**return**

**}**

Also there is simplified version of *For* loop:



The equivalent AutoHotkey code for that diagram is the same as for *For* loop.

**Foreach loop**

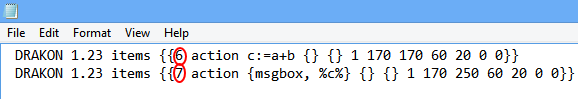
There is no working *Foreach* loop in beta version. It can be added later if there will be demand for it. **Don’t try to use it until it officially added to prevent unexpected behaviour!**

**Debugging tips**

Note the item comments (like *; item 6* and *; item 7*) in the generated code from previous diagram. They tell from which icon of the diagram the next piece of code is generated from.

To jump to the specific icon if you know its number, press Ctrl-I and enter the number. This is convenient for debugging.

To get number of icon: Select and copy icon in DRAKON Editor and paste it in any text editor. Here I pasted two icons (each line each pasted icon) from previous diagram:



First number after braces in text(shown in red circles) is number of icon. Number of first pasted icon is 6 and second icon 7.

**Reserved Labels and Variables**

It is not recommended to create *Labels* manually. Right way is to create *Free* type diagram that will automatically create *Label* corresponding to its name (more in “Types of diagrams and commenting diagrams” section of this manual). However, in case there is need to create *Label* manually, it must not be the same as the name of any *Free* type diagram in your file.

Variable **\_next\_item\_** is reserved and is used in some cases by DRAKON Editor in code generation process.