The UBot Studio

SCRIPT REFERENCE

The Professional and Developer Editions

*(Dev License ONLY)

Commands

*(Dev License ONLY)

Constants

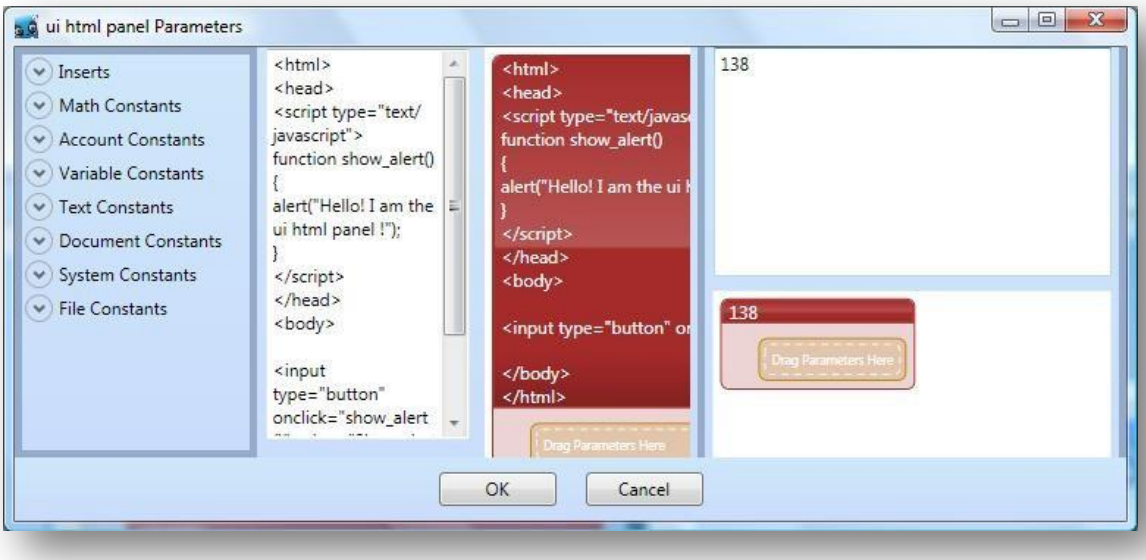
- The Variable Constants
- The Text Constants
- The File Constants

*(Dev License ONLY)

The UI Commands*

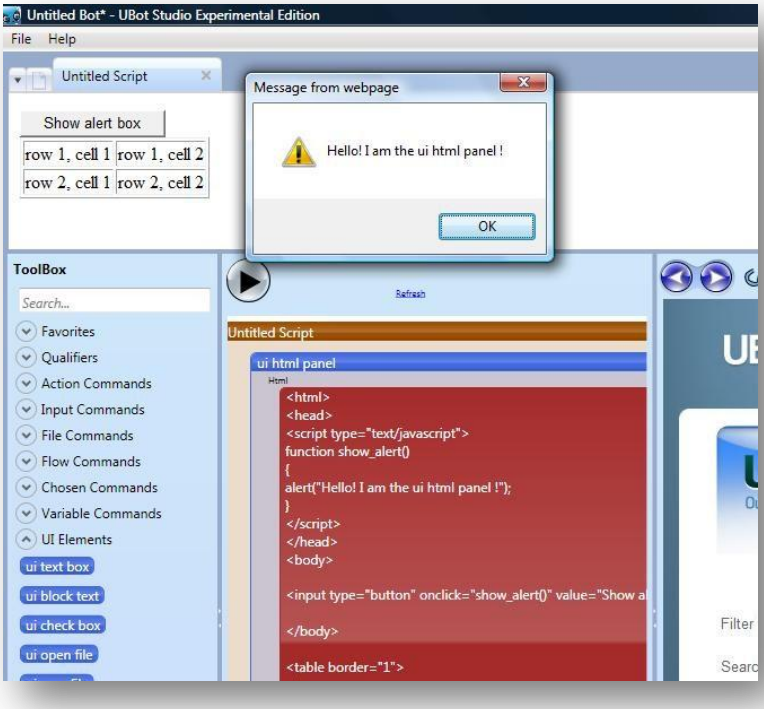
UI html Panel

This command creates a blank panel in the ui area in which you can use html, css, and javascript to design your own ui.
Simply drag the command into the script area to open the parameter window.



Write the html, css or javascript code into the parameter window. Click ok to exit.
The results of the code in the parameter window will appear in the UI area of UBot Studio.

*(Dev License ONLY)



UI Splash Page

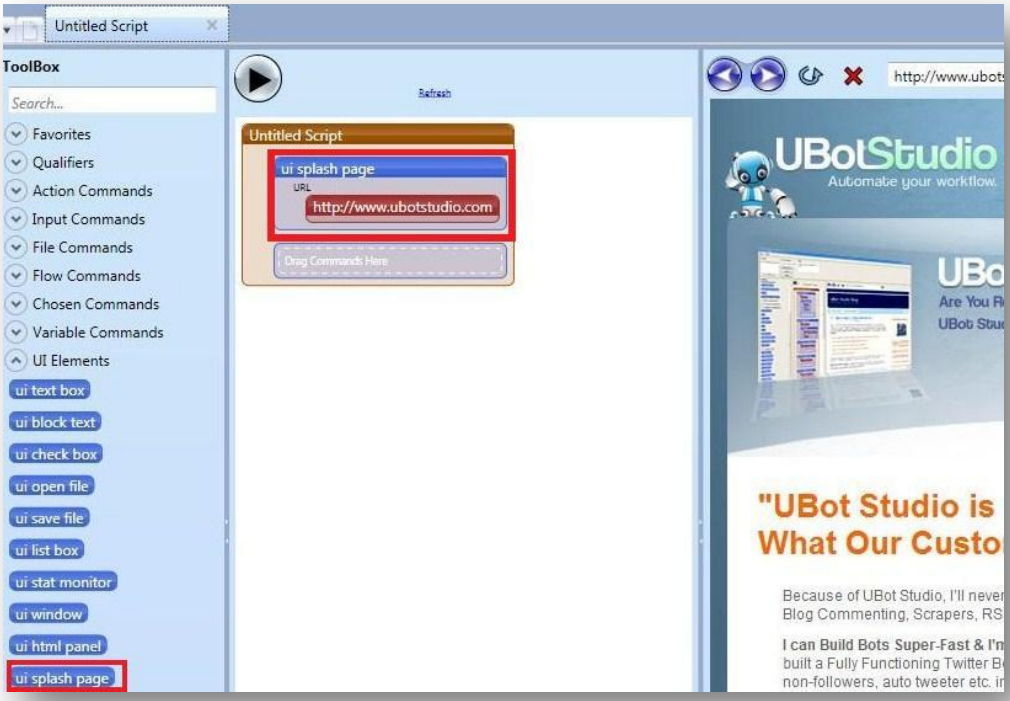
This command allows you to display a webpage within the compiled bot before the bot even starts to run.

To use the UI splash page command, simply drag the command into the script area, type in the full url of your desired splash page in this format:

www.ubotstudio.com

Refresh the script to see the splash page displayed within your browser.

*(Dev License ONLY)



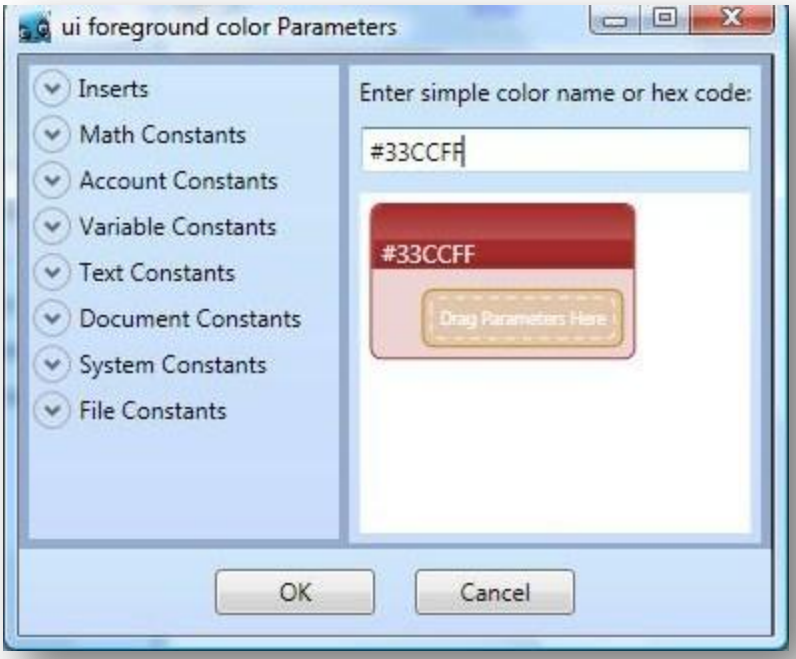
UI Remove Branding

This command will remove the "powered by" link in UBot Compiled Bots.
Simply drag the command into the script area.
The branding will be removed once the script is compiled.

UI Foreground Color

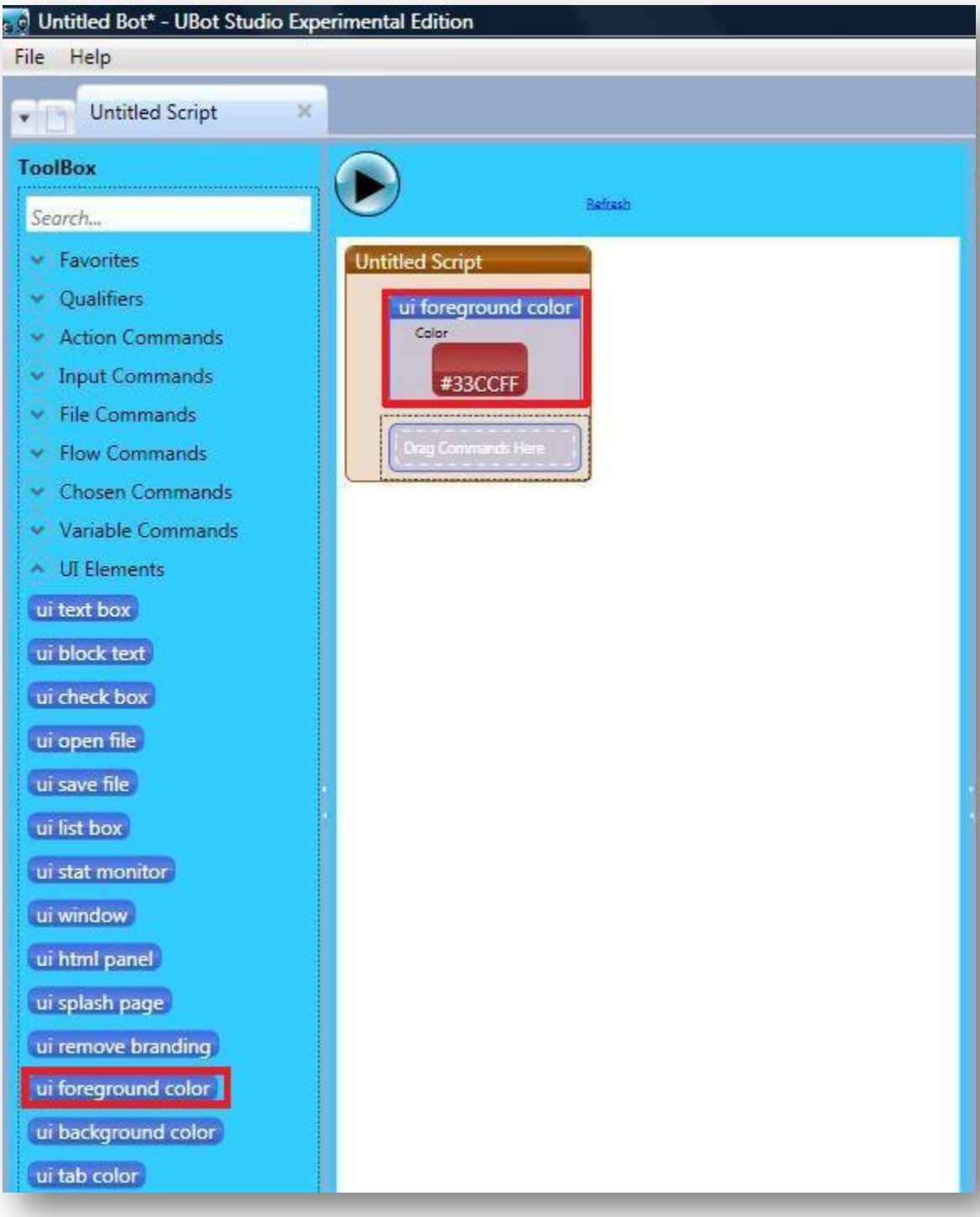
This command will change the foreground color in UBot Studio, UBot Compiled Bots, and UBot Reader.
Simply drag the command into the scripting area. A parameter window will pop up, and you can either write the name of a color (i.e. blue) or use a hex code.

*(Dev License ONLY)



Click ok and the foreground of your bot will now be set to the color you designated.

*(Dev License ONLY)



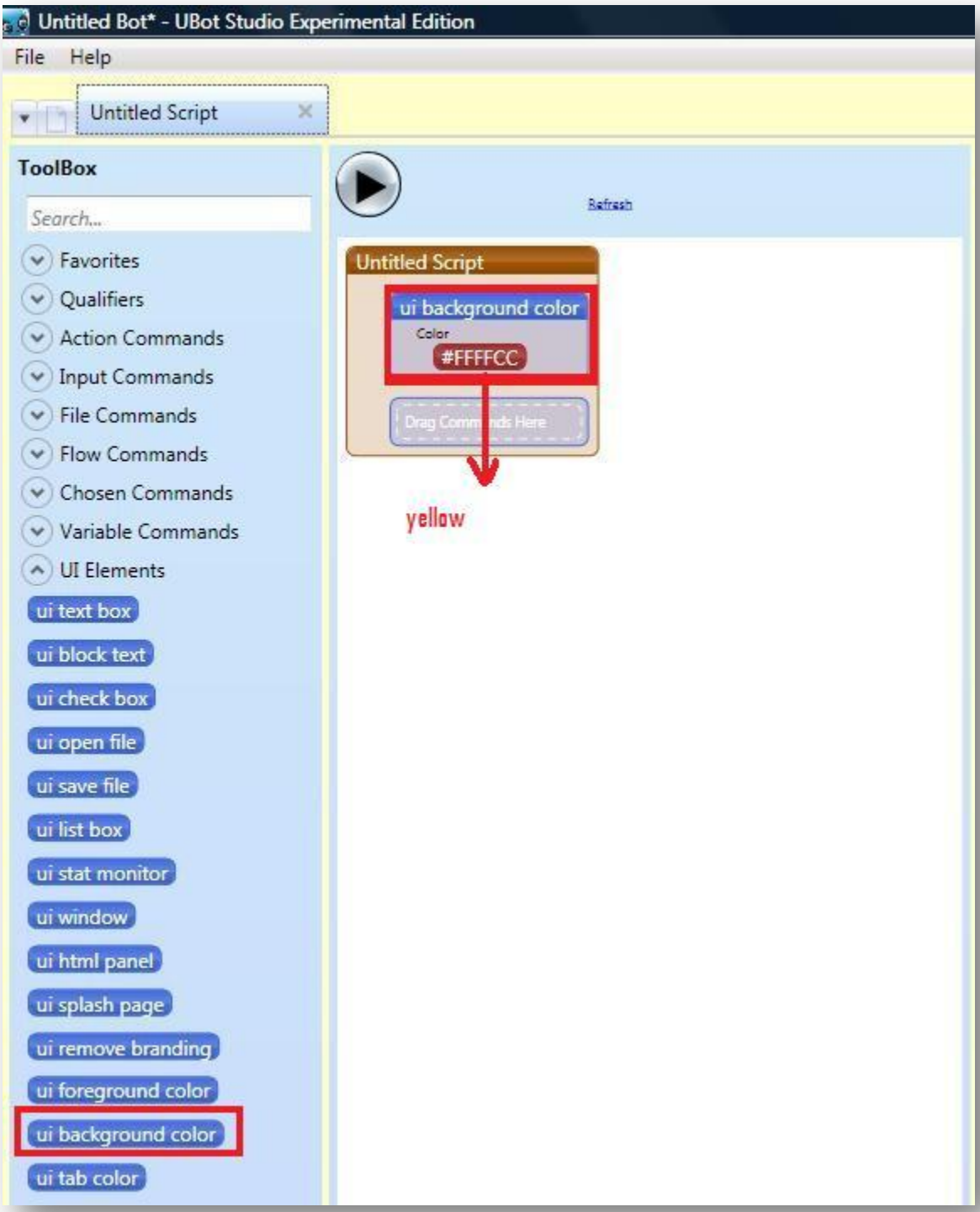
*(Dev License ONLY)

UI Background Color

This command will change the background color in UBot Studio, UBot Compiled Bots, and UBot Reader.

Simply drag the command into the scripting area, type your hex code or color name into the parameter window, and your bot's background color will be changed to the preferred color.

*(Dev License ONLY)



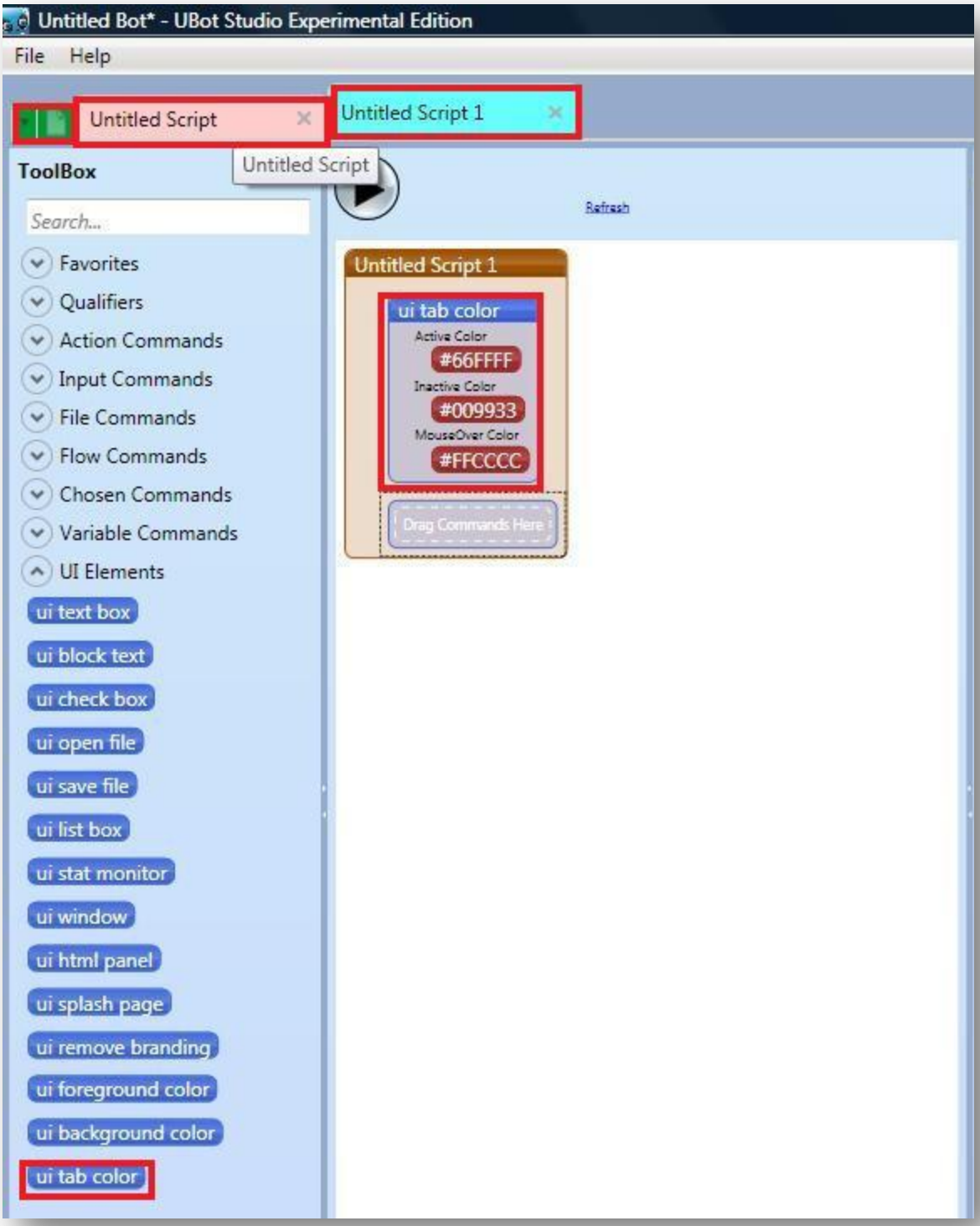
*(Dev License ONLY)

UI Tab Color

This command will change the tab color in UBot Studio, UBot Compiled Bots, and UBot Reader.
Simply drag the command into the scripting area. When the parameter window pops up, you will have the option of changing the color for the inactive, the active and the mouse over color.

Click ok, after typing in the colors you prefer.
The standard tab colors will change to the colors your designated within the parameter window.

*(Dev License ONLY)



*(Dev License ONLY)

The Choose/Chosen Commands

Scrape Chosen Table (often used along with the choose ancestor command)

This command scrapes and creates a table based on a chosen table on a webpage.

An element is first chosen with the choose by attribute command. Afterwards, the table is chosen with the choose by ancestor command, which in this case will be the entire table.

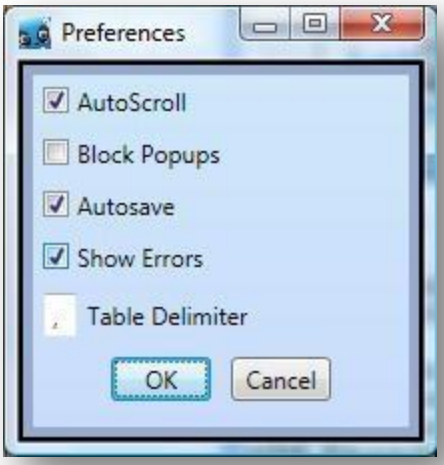
Ex. We have a table with a list of fruits in one column, and percentages in the other. We will choose one of the names first, in this case, Apples, and then we will choose the ancestor of that element, which is the table. The tag you will type into the parameter window will be “table”.

*(Dev License ONLY)

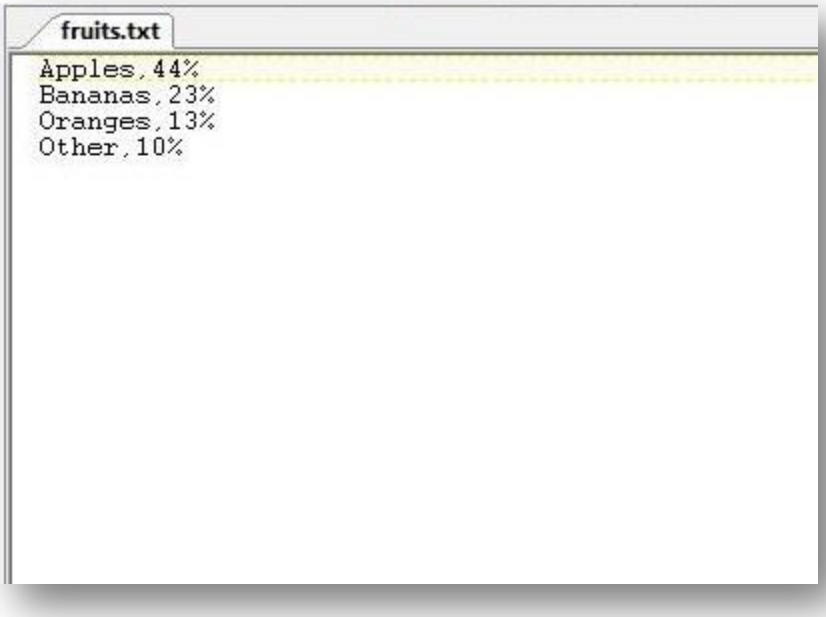
After choosing your table by ancestor, you will right click the table again and choose the scrape chosen table command. When the parameter window pops up, you will give the table you are scraping a name. Click ok to exit.

You can add a save to file command to save the table to a file. You can also add a clear table command from the variable commands menu to clear the table before adding the scraped table. Before you do so, you can designate a delimiter within Preferences in your file menu in UBot Studio.

*(Dev License ONLY)



After you run your script, this will be the resulting scrape.



Notice the commas as the designated delimiter.

Choose by Ancestor (often used along with scrape chosen table)

*(Dev License ONLY)

This command chooses by the ancestor of a selected element on a page.

You would first choose the element with choose by attribute, and then choose the ancestor of the element. This command is perfect for scraping table.

Choose by Parent

This command chooses by the parent of a selected element on a page.

In this example, we have another table to use as an example. We are trying to choose the parent of the element "**Element content**".

The elements "Start tag*, element content, and end tag" together, shown in the image below:

Start tag *	Element content	End tag *
	This is a paragraph	
	This is a link	

Is the parent of the element "Element content"

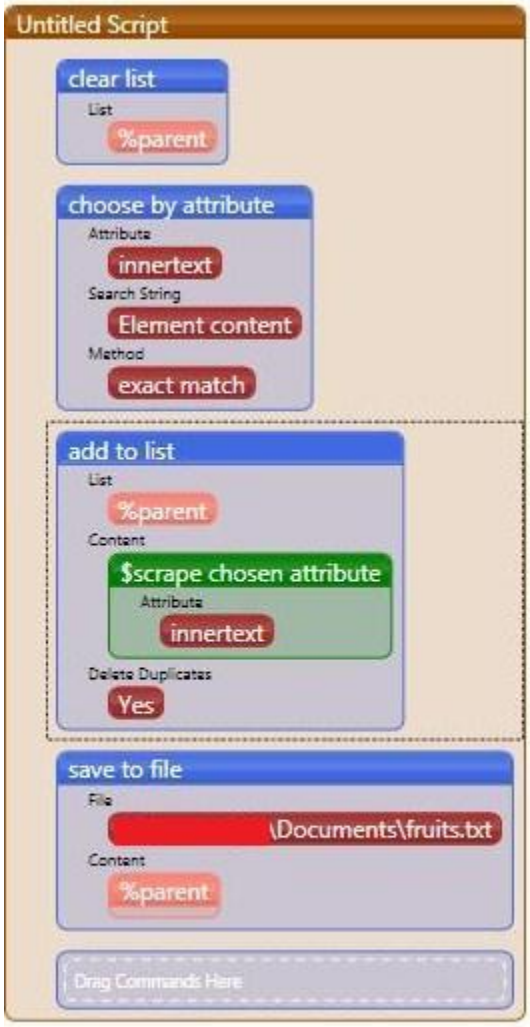
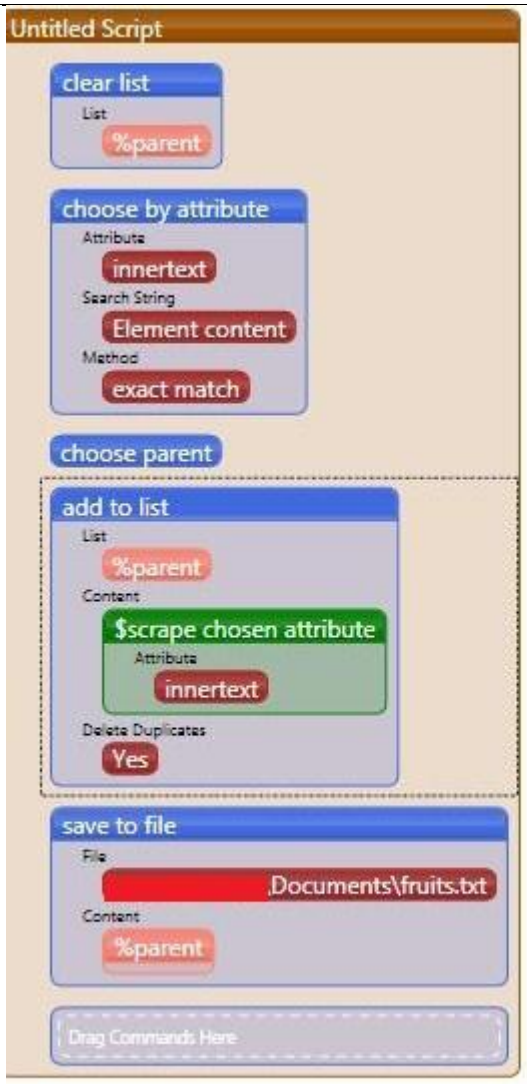
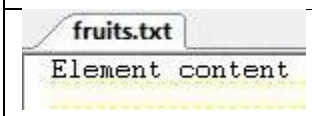
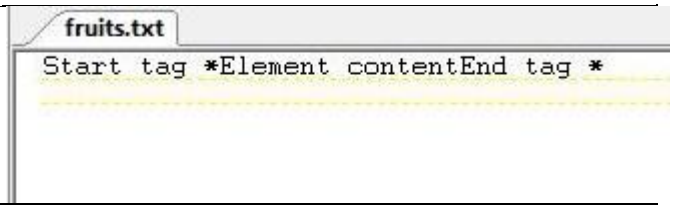
Start tag *	Element content	End tag *
	This is a paragraph	
	This is a link	

Just like the example for the choose by ancestor command example, you will choose Element content by attribute, and then choose it by parent.

Below are two scripts: one with the choose parent command and one without. Each script produces different results.

*(Dev License ONLY)

*(Dev License ONLY)

Without Choose Parent Command	With Choose Parent Command
	
	

*(Dev License ONLY)

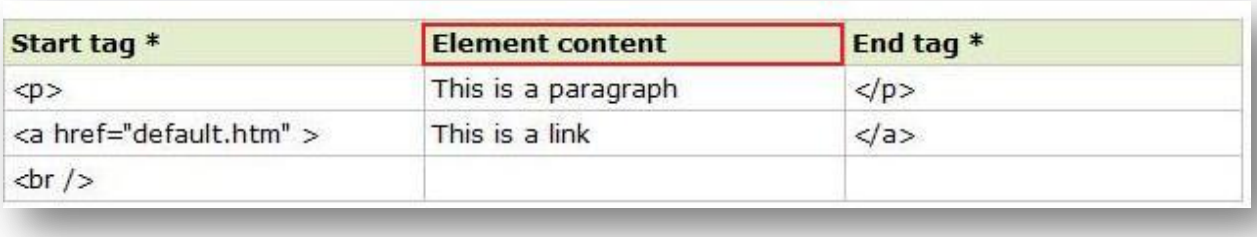
Choose by first Child

This command chooses by the first child of a selected element on a page.

Choose by next sibling

This command chooses by the sibling of a selected element on a page.

Using the same table for the other examples, the next sibling of the element “Element Content” in the table



Start tag *	Element content	End tag *
	This is a paragraph	
	This is a link	

…would be the “End Tag” element within the table. So scraping by the next sibling after choosing the item “Element Content” will produce the word “End tag*” within your file.

*(Dev License ONLY)

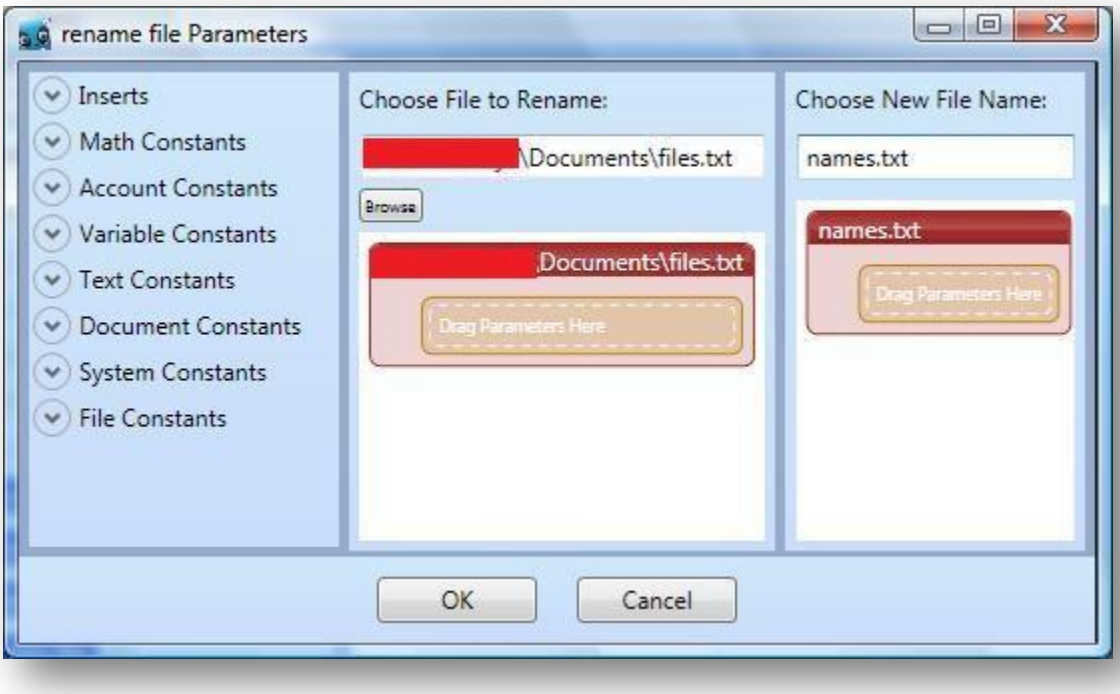
*(Dev License ONLY)

The File Commands

Rename File

Renames a file

The command can be found under the file commands within the tool box. Simply drag the command over, and a parameter window will pop up asking you to browse for the file you are trying to rename and type in a different name for it, along with the extension.



Click ok to exit, and run the script to rename your file. The file name change can be seen here.

*(Dev License ONLY)

Move File

Moves a file to a new location.

The command can be found under the file commands within the tool box. Simply drag the command over, and a parameter window will pop up asking you to browse for the file you are trying to move and to browse for the new location you want to move it to.

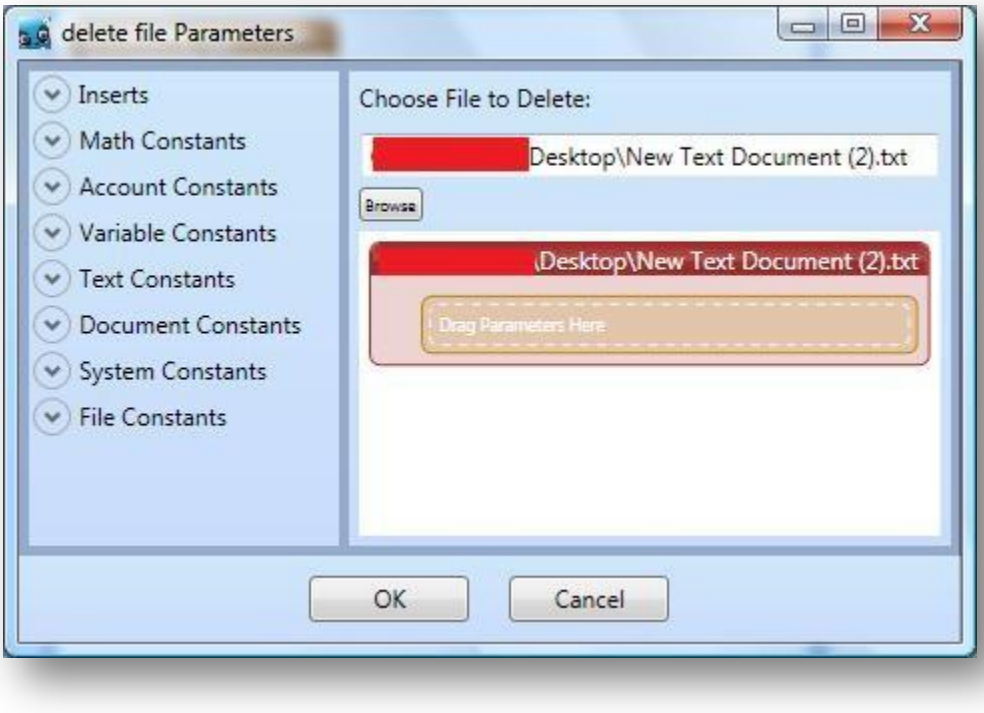
When you run your script, the file will be moved to the location you determined in the parameter window.

*(Dev License ONLY)

Delete File

Deletes a file.

When the command is placed within the scripting area, a parameter window will pop up asking you to browser for the file you are trying to delete.



Once you choose the file, click ok and then run the script, the file will be moved to your recycling bin.

Copy File

Copies a file to a new location.

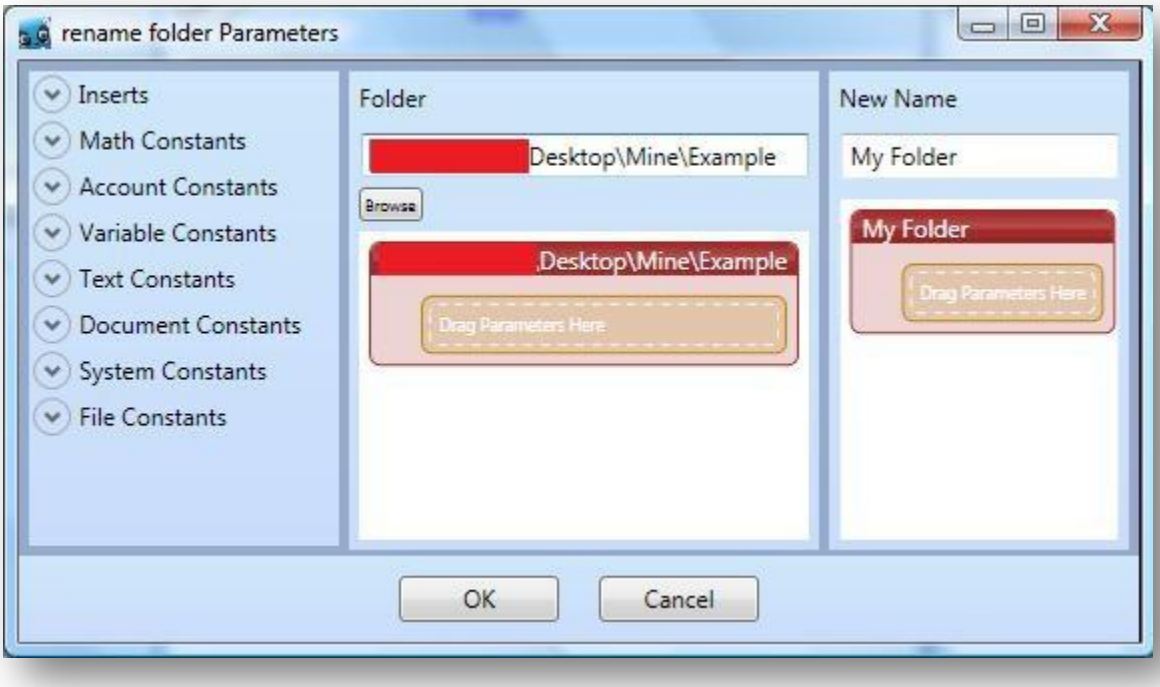
The copy file parameter window will look similar to the move file parameter window. You will browse for the file you would like to copy and then browse for the location you would like to move the file to. Once you run the script, the file will be moved to the location you designated in the parameter window.

*(Dev License ONLY)

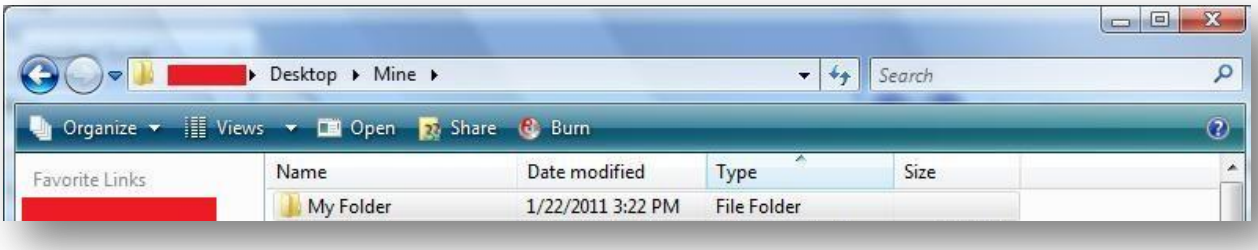
Rename Folder

Renames a Directory (Folder)

The rename folder parameter window will look similar to the copy file parameter window. You will browse for the folder you would like to copy and then type in the new folder name you would like to change the folder name to.



Once you run the script, the folder name will be changed to the name you designated in the parameter window.



*(Dev License ONLY)

Move Folder

Moves a folder to a new location.

The move folder parameter window will look similar to the rename folder parameter window. You will browse for the folder you would like to move and then browse for the location you would like to move the folder to.

Once you run the script, the folder will be moved to the location you designated in the parameter window.

Delete Folder

Deletes a folder.

When the command is placed within the scripting area, a parameter window will pop up asking you to browser for the folder you are trying to delete.

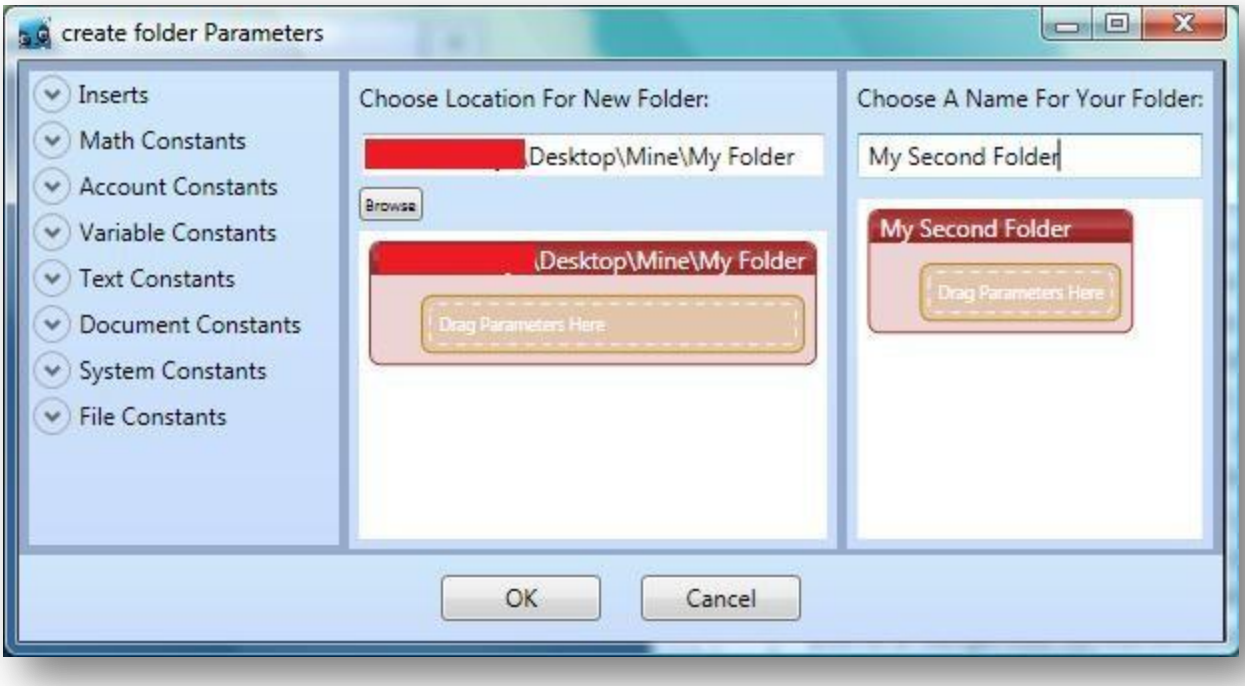
Once you choose the file, click ok and then run the script, the folder will be moved to your recycling bin.

Create Folder

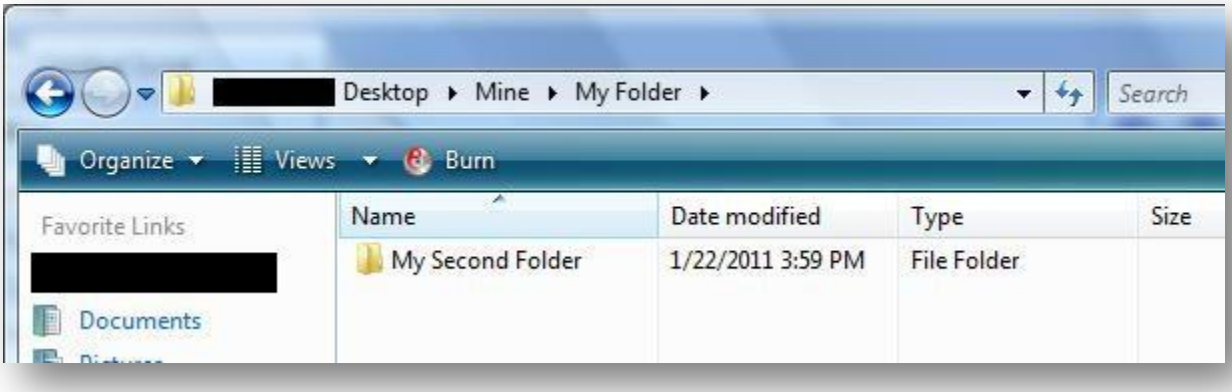
Creates a new folder

This command will allow you to create a new folder to any location on your directory. The parameter window will allow you to browse for the specific location you would like to create the folder to and name the folder.

*(Dev License ONLY)



When you run the script, your new folder will be created in the location you determined within the parameter window.



Copy Folder

Copies a folder to a new location.

*(Dev License ONLY)

The copy folder parameter window will look similar to the copy file parameter window. You will browse for the folder you would like to copy and then browse for the location you would like to move the folder to. Once you run the script, the folder will be moved to the location you designated in the parameter window.

*(Dev License ONLY)

Input Commands

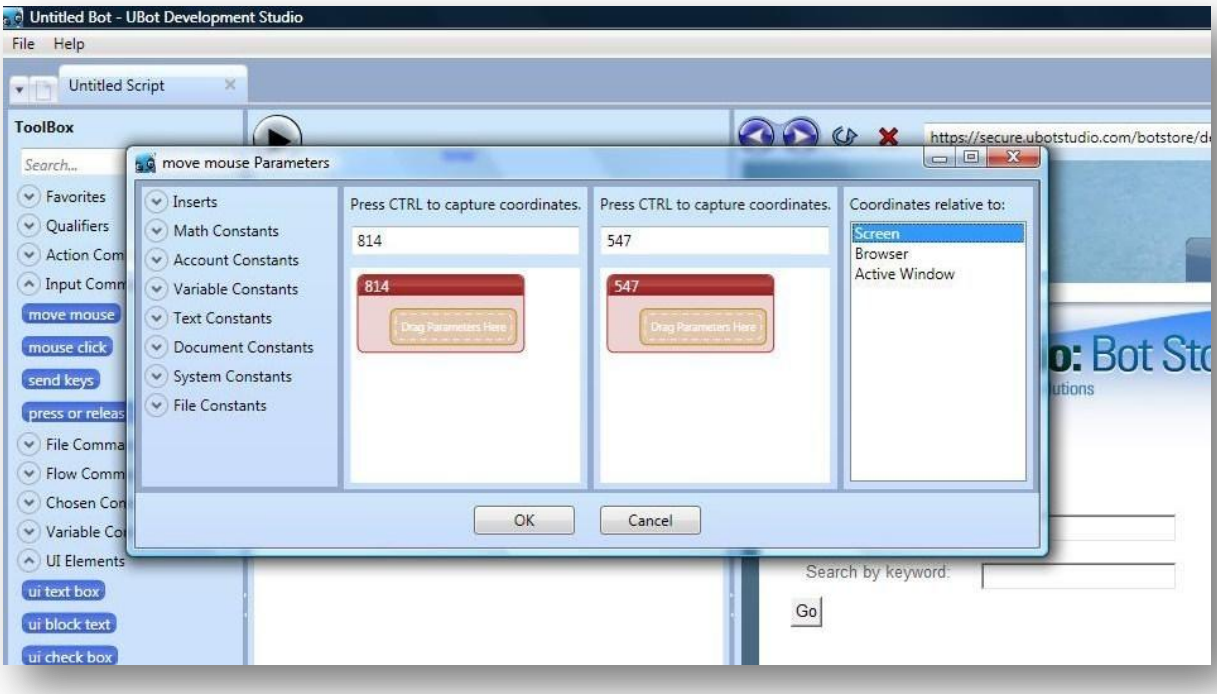
Move Mouse

This command will move the mouse to specified coordinates.

You can choose coordinates relative to the screen, the browser or the active window. Each environment will have different coordinates.

To use the command, simple drag the command into the script area. A parameter window will appear, asking you to choose which environment you are working in.

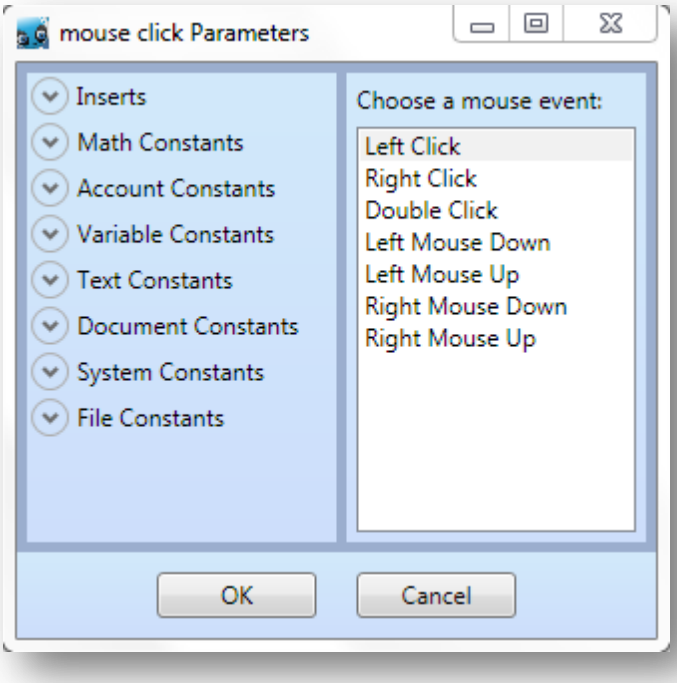
In the far right column, choose what the coordinates are relative to. Once you've chosen the environment, move the mouse to the location you would like the mouse moved to, and hold down the control key on your keyboard to capture the coordinates.



MOUSE CLICK:

*(Dev License ONLY)

This command will simulate a mouse click, a mouse up or mouse down action. There are several choices for mouse clicks:
1) Left Click
2) Right Click
3) Double Click
4) Left Mouse Down
5) Left Mouse Up
6) Right Mouse Down
7) Right Mouse Up



Send Keys

This command simulates typing on the keyboard. It can be very useful for using keyboard shortcuts in addition to using keyboard action keys.
Below is a list of commonly used Send Keys commands you can utilize with this command:

*(Dev License ONLY)

Key	Code
BACKSPACE	{BACKSPACE}, {BS}, or {BKSP}
BREAK	{BREAK}
CAPS LOCK	{CAPSLOCK}
DEL or DELETE	{DELETE} or {DEL}
DOWN ARROW	{DOWN}
END	{END}
ENTER	{ENTER}or ~
ESC	{ESC}
HELP	{HELP}
HOME	{HOME}
INS or INSERT	{INSERT} or {INS}
LEFT ARROW	{LEFT}
NUM LOCK	{NUMLOCK}
PAGE DOWN	{PGDN}
PAGE UP	{PGUP}
PRINT SCREEN	{PRTSC}
RIGHT ARROW	{RIGHT}
SCROLL LOCK	{SCROLLLOCK}
TAB	{TAB}
UP ARROW	{UP}
F1	{F1}

*(Dev License ONLY)

F2	{F2}
F3	{F3}
F4	{F4}
F5	{F5}
F6	{F6}
F7	{F7}
F8	{F8}
F9	{F9}
F10	{F10}
F11	{F11}
F12	{F12}
F13	{F13}
F14	{F14}
F15	{F15}
F16	{F16}

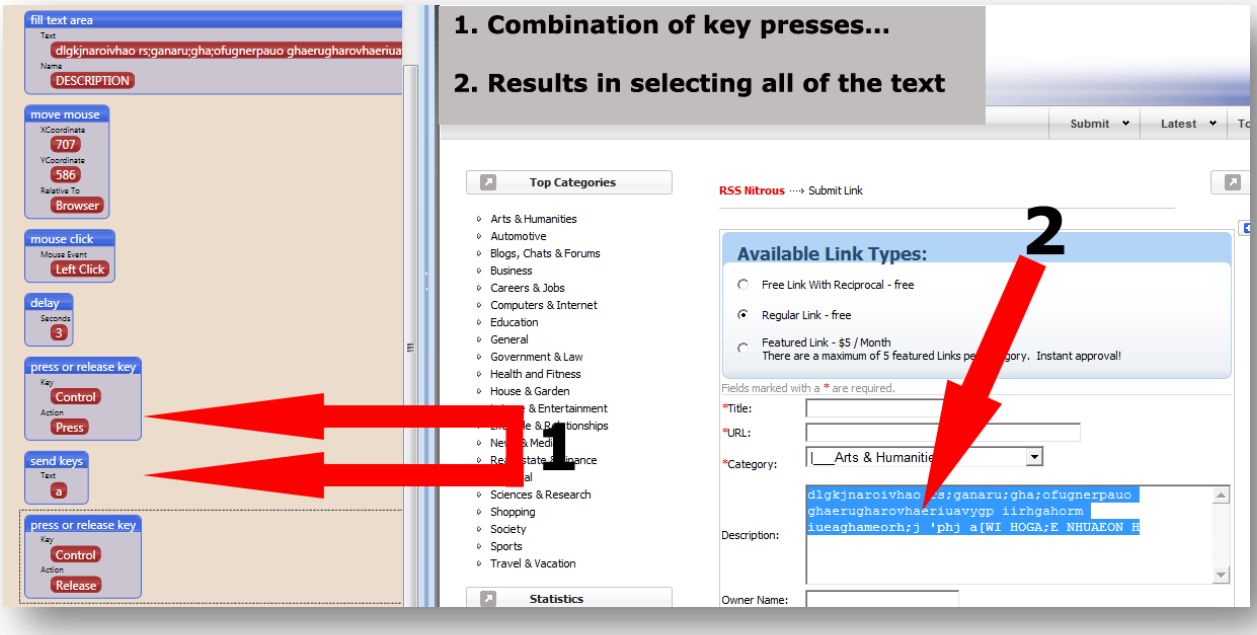
Send keys commands can be combined as shown in the image:

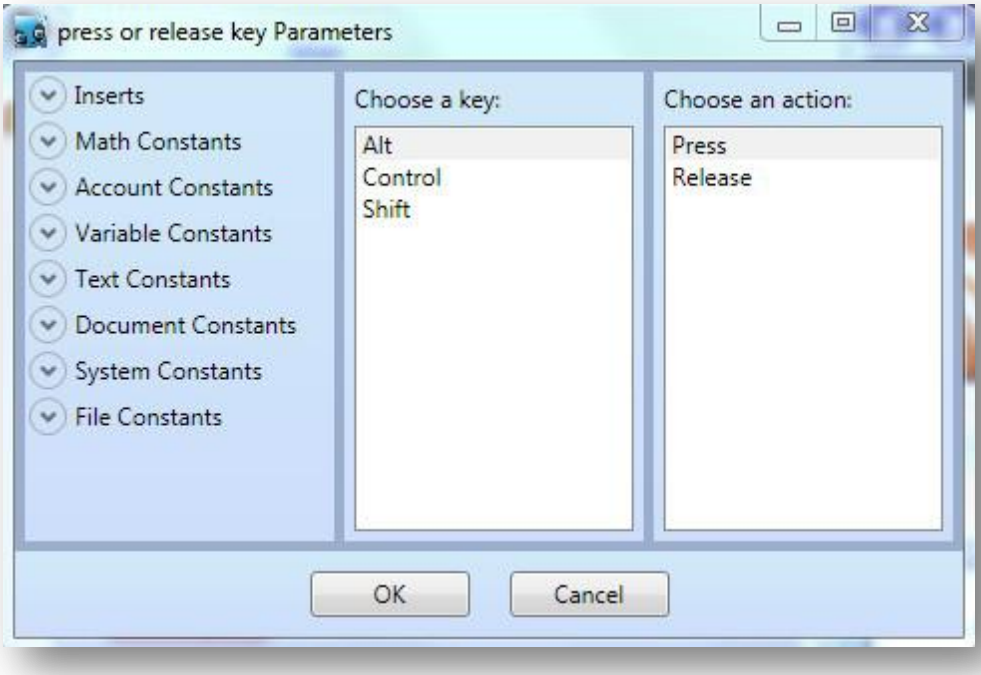
*(Dev License ONLY)

Press and Release Key

This command simulates the pressing (and holding) of the Alt, Shift and Ctrl keys.
There are two parameters to choose from:
1) Which key to press
2) Whether you are pressing or releasing the chosen key
The second parameter is important to understand because once you press one of these keys programmatically it will stay pressed until you release it programmatically.
You can use this command in conjunction with other key presses
(Such as the Send Keys command) to do things like Select All (Ctrl +
a), etc. (see image below)

*(Dev License ONLY)





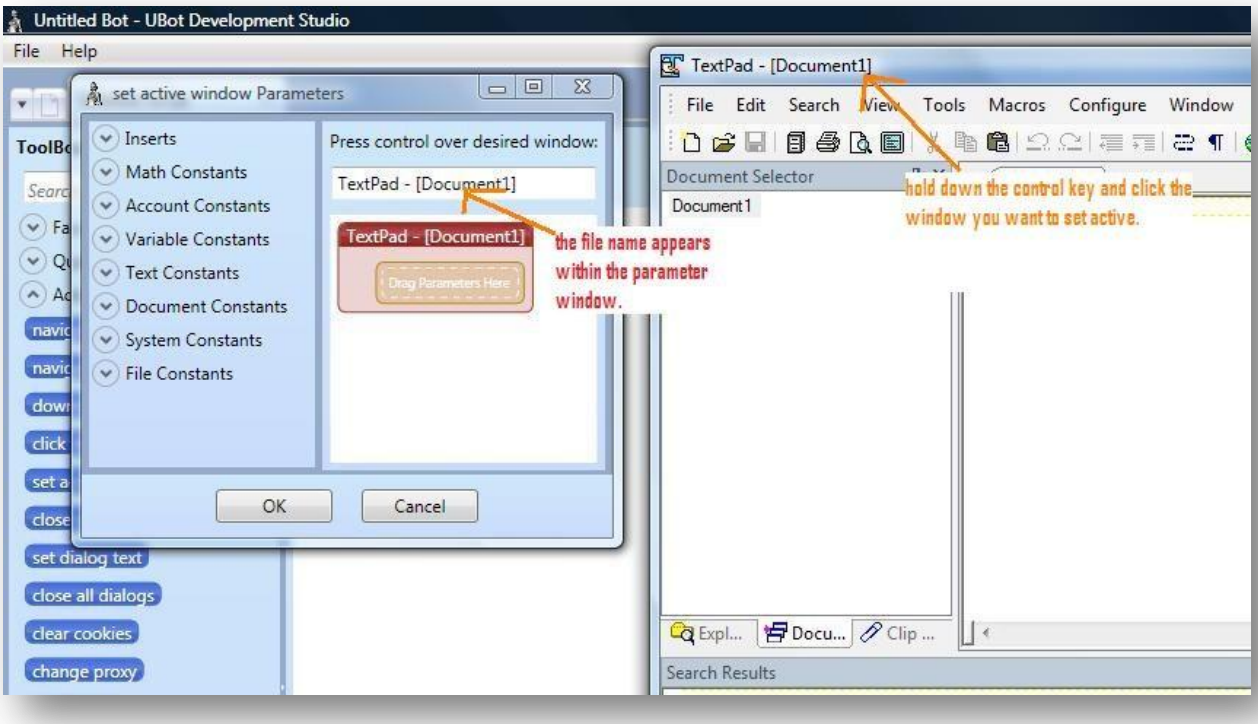
*(Dev License ONLY)

Action Commands

Set Active Window

This command will bring a desired window into focus according to that name of the program window's title.

This command will bring a desired window into focus according to its title. The title of the window is located in the upper left corner.



There is no need to enter the title manually into the node as you can simply click on the desired window once you bring the node into the scripting area.

Close Window

This command will close the active window.
The process works just like the set active window command. You drag the command into the script, and hold CTRL over the window you would like to close, as seen in the preceding image.
Run the script to close the window.

*(Dev License ONLY)

*(Dev License ONLY)

The Pro and Dev License Constants

*(Dev License ONLY)

Variable Constants

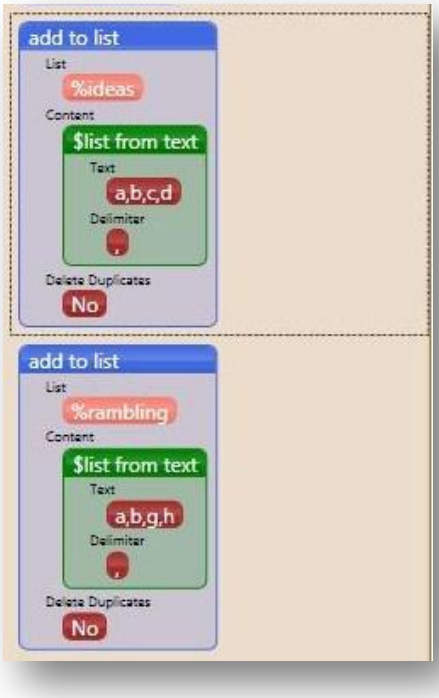
Subtract lists

This command allows you to compare two lists and create a new list containing all items that are unique to the first list.

In this example, we have two lists. List one, which is called $ideas, contains the following: a,b,c,d

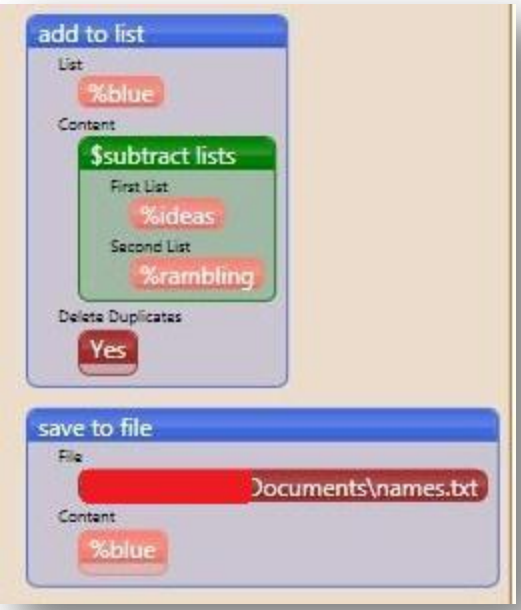
The other list, called $ramblings contains the following: a, b, g,h

Notice that the items in the list are added to the list with the list from text constant, which allows you to type in your list items as opposed to adding a list into UBot from a file on your computer with the list from file constant.

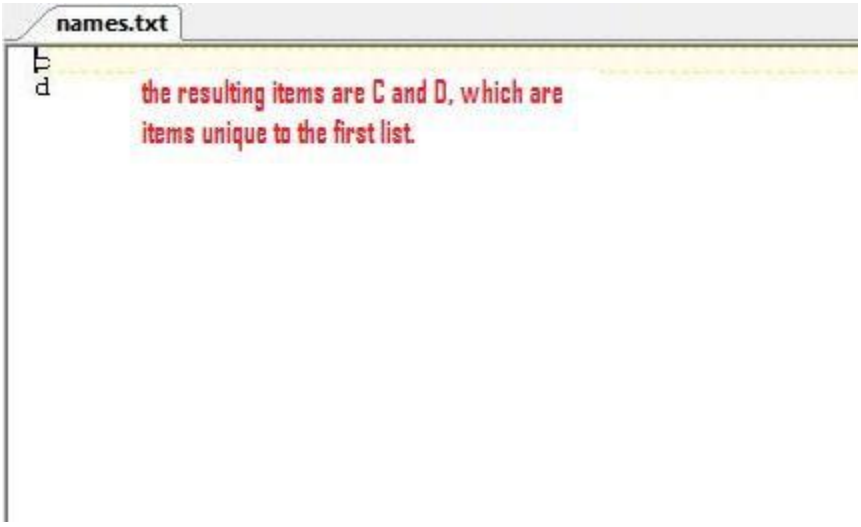


What we want to do is to subtract the list $ideas from the list $ramblings, and leave only the items that are unique to only the first list. We then want to save the unique items to a file on our computer.

*(Dev License ONLY)



When the script is run, the following results are saved to the file:

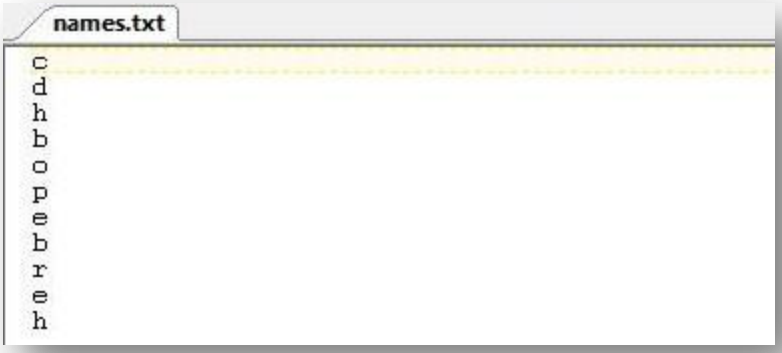


Sort lists

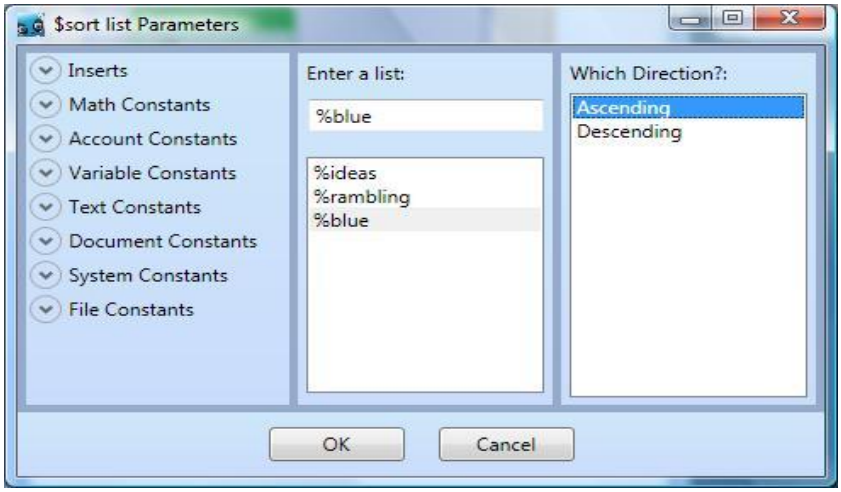
Sort List allows you to sort all items contained in a list in either ascending or descending order.

In this example, I have a list of letters:

*(Dev License ONLY)

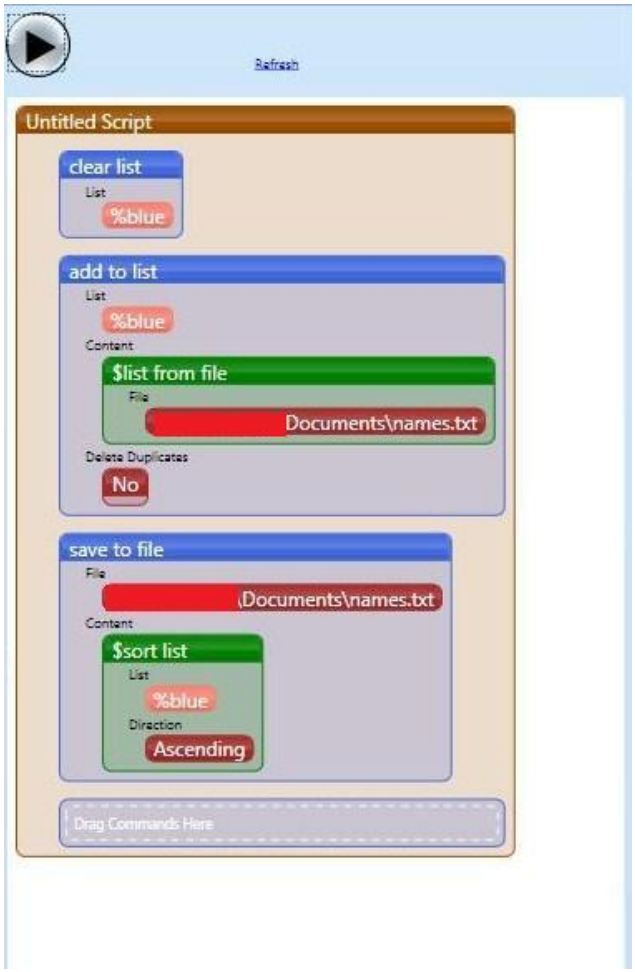


We are going to sort this list in ascending order. The list is added from a file with the list from file constant. We are sorting the file while we are saving the items to a file. In the save to file command, instead of inserting the list directly into the save to file command, you would insert a sort file constant, and then choose the list containing the items you would like to sort.



This is the parameter window for the sort list constant, where a list is chosen for sorting.

*(Dev License ONLY)



When the list is sorted, the result will be saved to the file.

Keep in mind that if you set Delete Duplicates to Yes in your add to list command, duplicate items will be removed from the list.

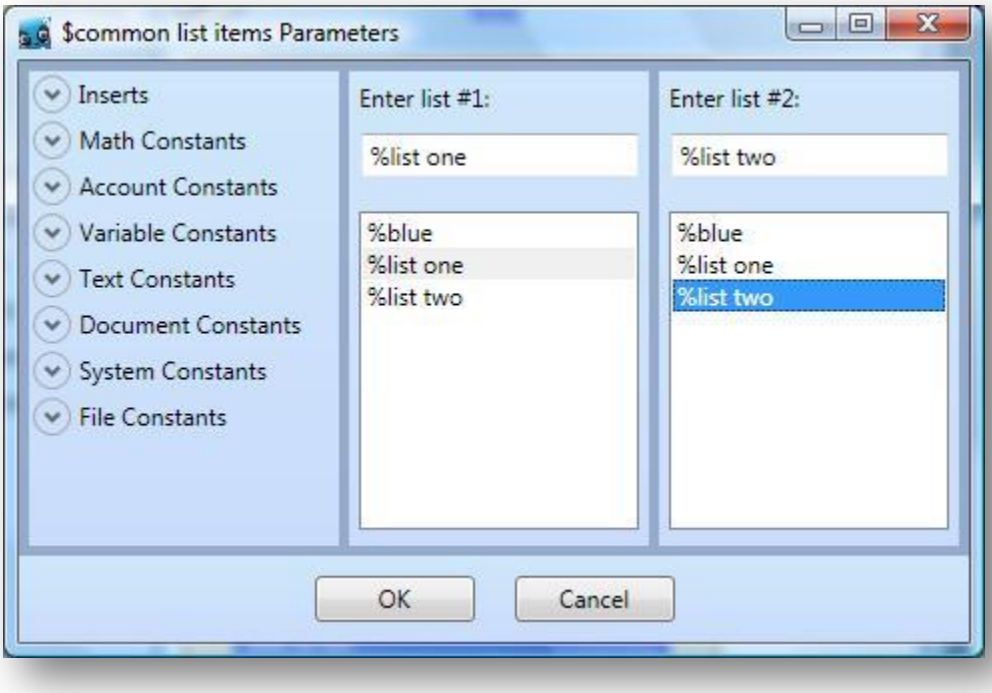
Common list items

*(Dev License ONLY)

This command allows you to compare two lists and create a third list containing all items that are common to both lists.

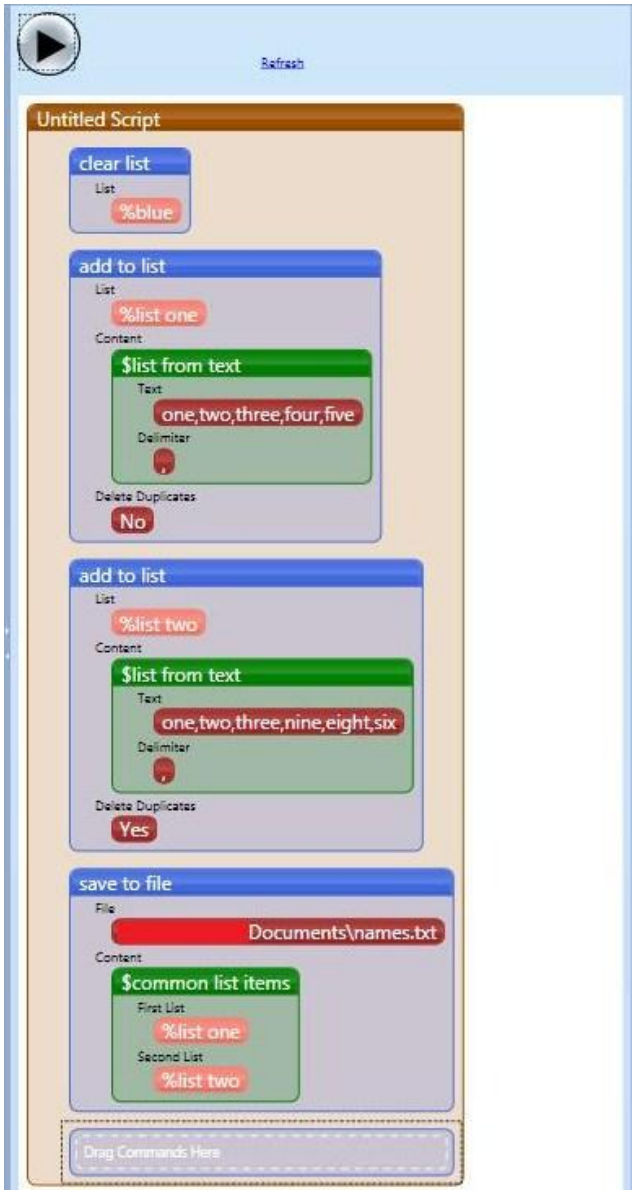
This command works in the opposite way that the subtract list command works. Instead of comparing two lists and finding items only unique to the first list, it finds list items common to both lists.

In this example, we have two lists, list one and list two.

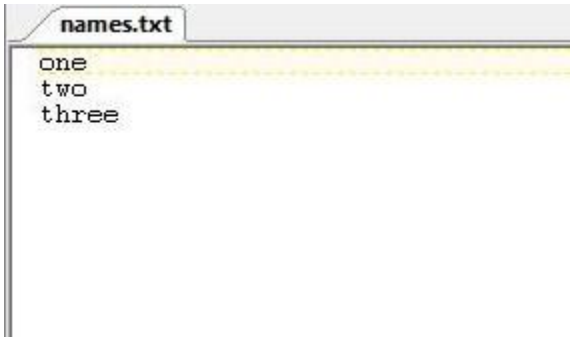


We are then finding the common list items in the save to file command, and then saving the common items in a file.

*(Dev License ONLY)



And this is the resulting items when the common items in the list are done.



Because the words one, two and there are common to both lists.

*(Dev License ONLY)

*(Dev License ONLY)

Text Constants

Text to uppercase

Changes all text in a string to uppercase letters.

You can turn any text item to uppercase, whether is a list item from a list or simply a string you are trying to fill a field with. Here are some examples of how to use the constant.

This example shows the text to uppercase constant in a fill field command. This will change the string “variables” to uppercase.

*(Dev License ONLY)

This example shows how you can use the text to uppercase command with a next list item command. The list tem being changed here is the phrase “Passwords Here Please”. After passing through the text to uppercase constant, the phrase is now “PASSWORDS HERE PLEASE”.

In this example, you simply drag the field you are trying to fill into the scripting area. When the parameter window pops up, go under the text constants and select the Text to Uppercase constant. Another parameter window will pop up, where you will either type in the phrase you want changed or insert the next list item constant from the list you are using. The constant will change any string into uppercase form.

Text to lowercase

Changes all text in a string to lowercase letters.

The constant works exactly like the text to uppercase constant, except it changes the string to all lowercase characters. You can use it either with list items, as seen in this example:

*(Dev License ONLY)

Or with a typed in string, as seen in this example:

*(Dev License ONLY)

Text to proper case

Capitalizes the first letter of each word in a string which is also known as Title Case.

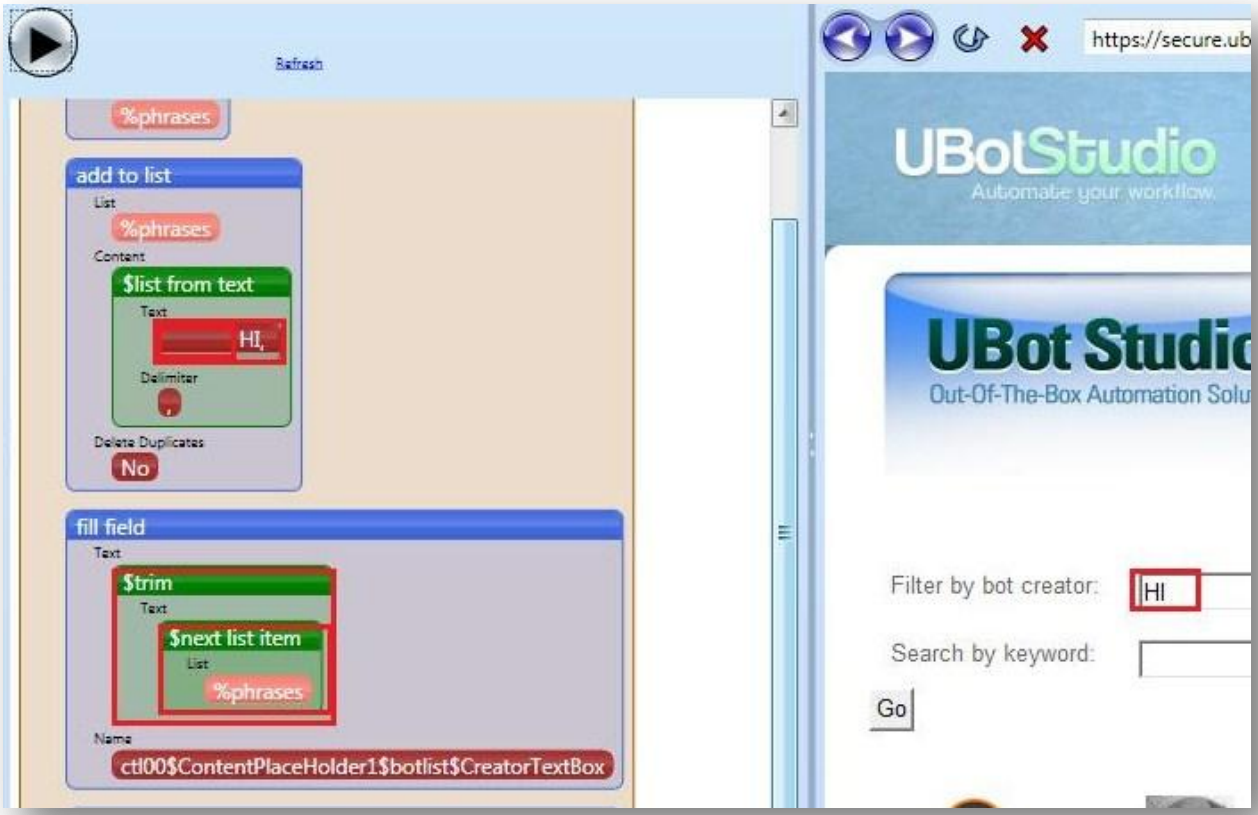
This constant also works like the first two text constants. You can use it the same way as the text to uppercase constant and the text to lowercase constant.

Trim

Removes any leading or trailing blank spaces from a string.

The constant can be used the same way you would use the other two constants, except it removes extra spaces in a list item. In this example, I am trying to post a list item that has a large amount of space preceding it. With the trim constant, we are able to remove the blank space before the list item.

*(Dev License ONLY)

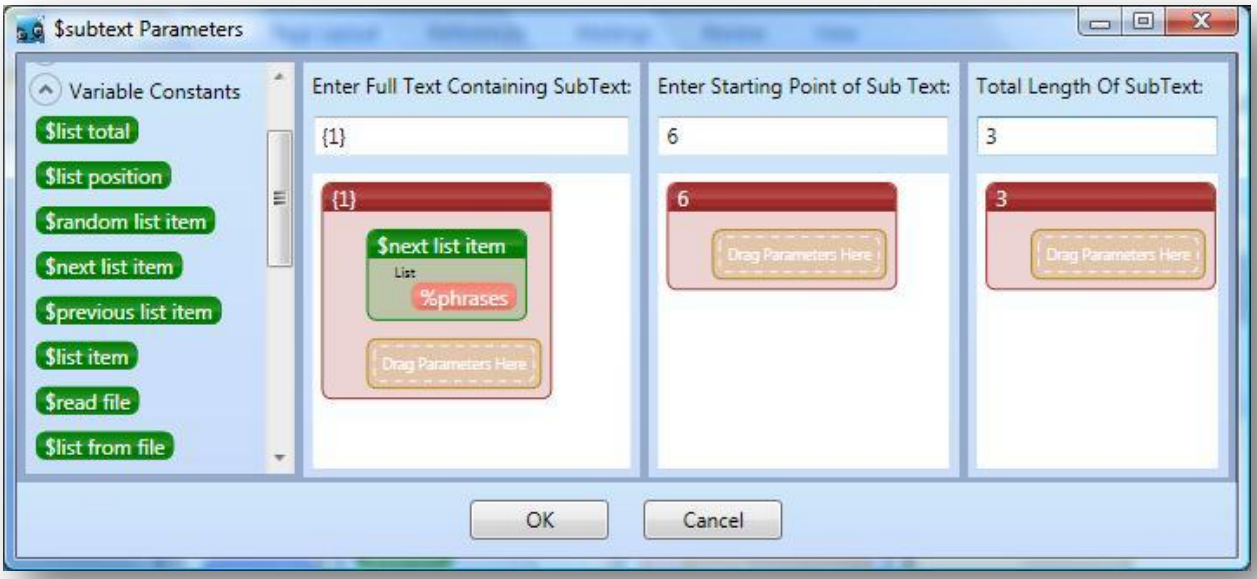


Subtext

Returns a new string that is a substring of an original string. The substring begins at a given starting point and extends up to the given length.

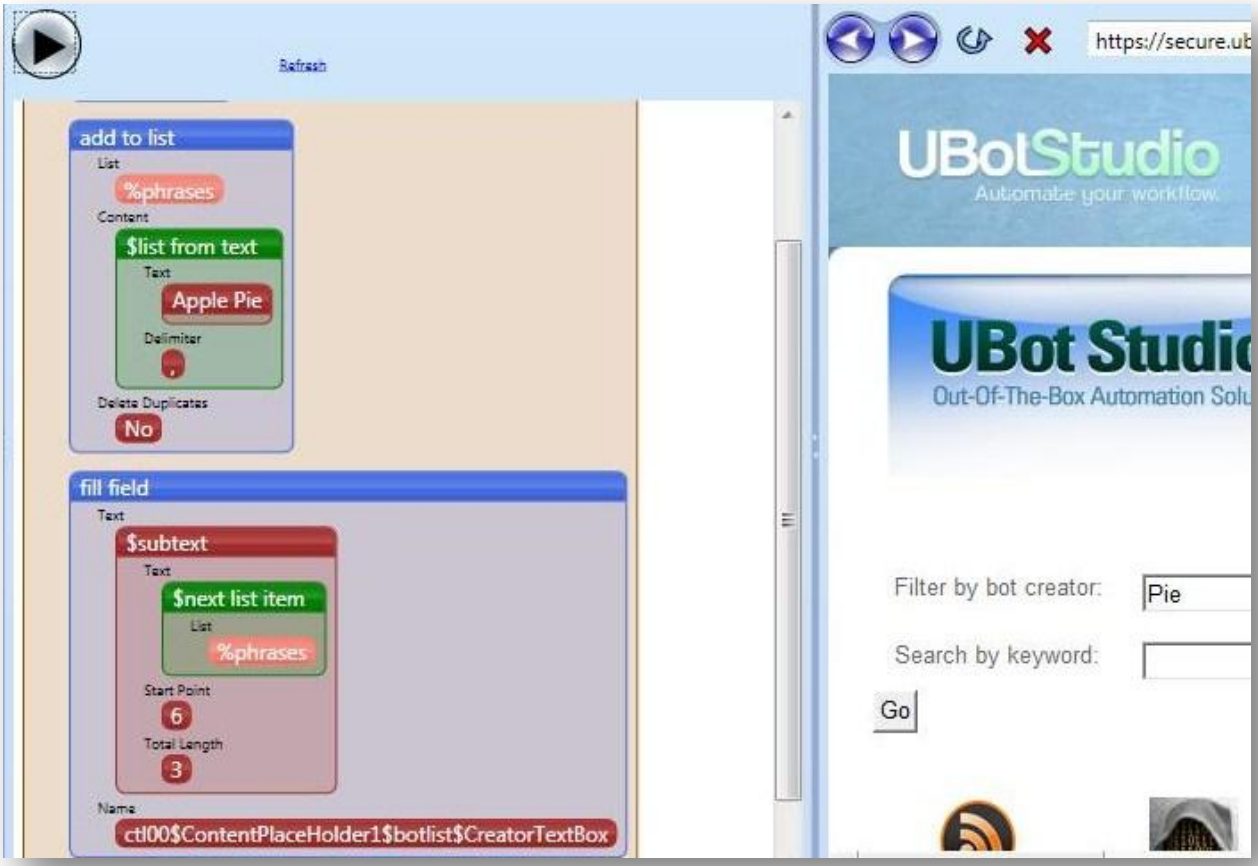
Again, this constant can be used the way that the other constants can be used, except this finds the determined subtext of a string. After the second list item pops up, you will be asked to determine where the subtext starts and what the total length of the subtext is within the string.

*(Dev License ONLY)



In this case, our string is the phrase “Apple Pie”. We are trying to grab the subtext Pie, which is three letters long and starts at position 6 of the string. When we run the script after clicking of, the field will be filled with the subtext “Pie”.

*(Dev License ONLY)



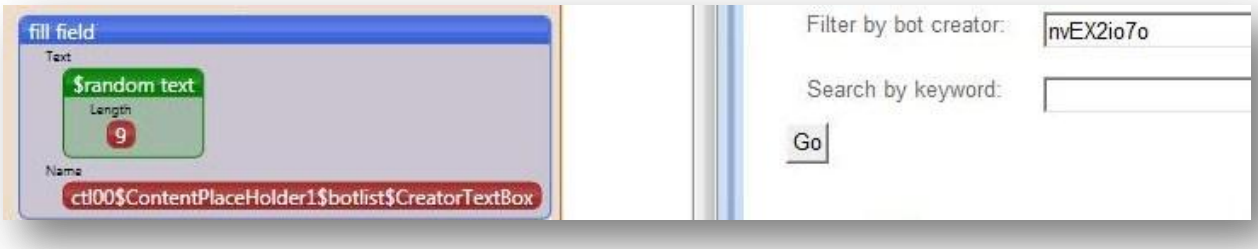
The subtext of “Apple Pie” is “Pie”.

Random text

Generates a random alpha-numeric string of user defined length, including upper and lowercase letters

This constant will fill a field with a bunch of random text. It is usually used as a space filler.

*(Dev License ONLY)



Pad text

Adds a fixed number of characters to the left or right side of a string based on the total length of the string selected.

In this example, we are padding the text “Apple Pies”. You would drag the field you want populate into the script window. When the parameter window pops up, choose the pad text constant from under the text constants. You will be allowed to choose the character you are trying to pad the original text with. The original text we are trying to pad in this case is a list item from a list. Total length refers to how long you want the

*(Dev License ONLY)

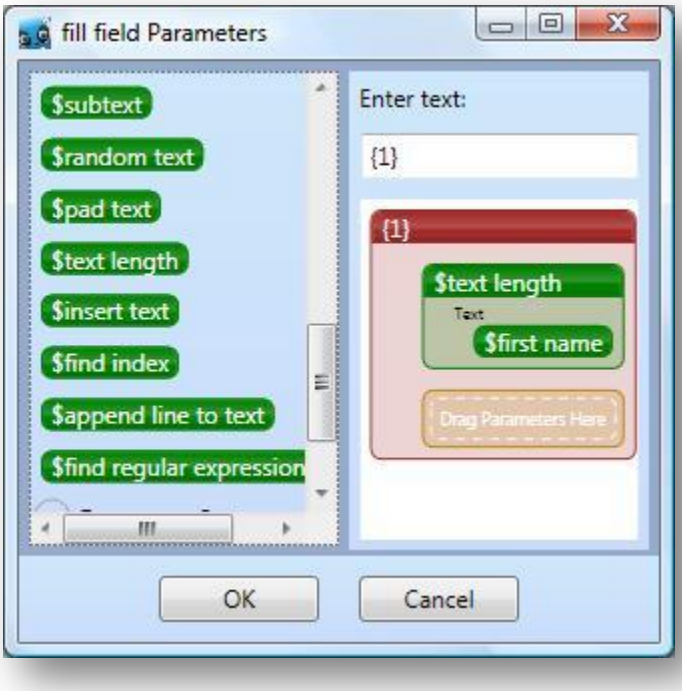
text to be after padding. The direction refers to whether you want the text padded from the left or the right.

In this case the total length of the original text was 9. After padding the test with one “s” the total length will be 10. I can pad the text with many more “s”’s by increasing the total length of the text after padding.

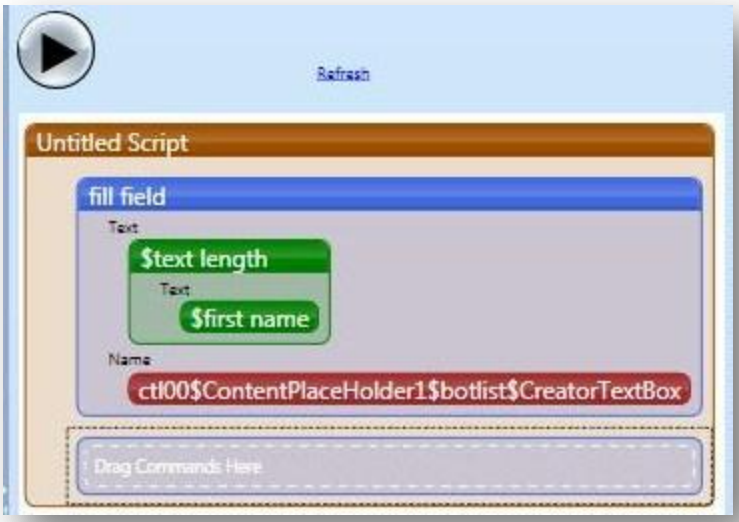
Text length

Returns the total length of some text.

You can fill a field with the constant to find the text length of either an account constant (username, first name, password, etc), a variable, or a piece of text.



*(Dev License ONLY)



Insert text

Inserts a substring into a string based on the specified position. In this example, we are trying to follow up a first name with a bunch of numbers, so we can try to create a username like: Penny42534

***(Dev License ONLY)**

We set the first name constant first and named my variable. We then dragged in the field we want to fill with the result. When the parameter pops up, we drag in the insert text constant. Another parameter window will pop up, and there will be three fields you will need to fill:

Original Text: Where our variable will go

Text to Insert: We will drag in a zipcode variable

Position: The position will be the text length of the variable containing the original text.

The resulting item will be a first name with numbers following it.

Find index

Finds the index of any subtext within a text string. This constant is perfect for use within the insert text constant. You can find the index of an element and then use that index number to determine where you want a new item inserted. In this example, we would like to insert the name “Manning” into the item “Jayne333” between Jayne and

*(Dev License ONLY)

333. The first thing we did was find the index of where we would like to insert the word. So we are finding the index of 333 within Jayne 333. The index is 5.

So then we move on to fill the field. After dragging in the field we would like to fill, you will go under account constants and choose the Insert Text constant.

Original Text: Where Jayne333 will go

Text to Insert: We are trying to insert the word “Manning”

Position: The position will be the variable with the index set to it.

After running the script, “Jayne333” turns to “JayneManning333”.

Append line to text

Appends a new line to a piece of text.

The constant simply adds one piece of text to another. So, in this example, we have a sentence being spun and set to a variable. We are then going to fill the field with spun content in the variable, which will be appended to a first name constant.

*(Dev License ONLY)

When you drag in the append line to text constant, you will notice there are two columns:

Original Text: The text you are trying to append another text to.

Text: Which refers to the text you are trying to append into the original text.

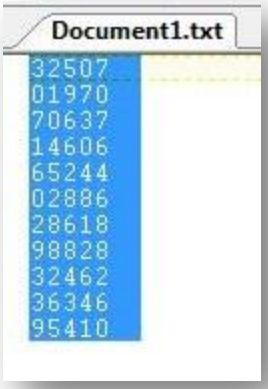
Click ok and click ok again to get back to the scripting area. Once you run your script, the field will be filled with the resulting appended text.

Find regular expression

Find regular expression allows users to conduct specific searches on text strings using regular expressions.

This constant is used for lists, variables, or whatever piece of information you are trying to scrape. In our examples, we have created a list of random numbers and saved them to a list in a .txt file.

*(Dev License ONLY)



Next, we are going to write a simple regex code to find all the numbers that start with the numbers between the numbers [2-3]. The numbers should also end with the numbers between [0-6].

Here is our regex for that criteria:

[2-3]...[0-6]

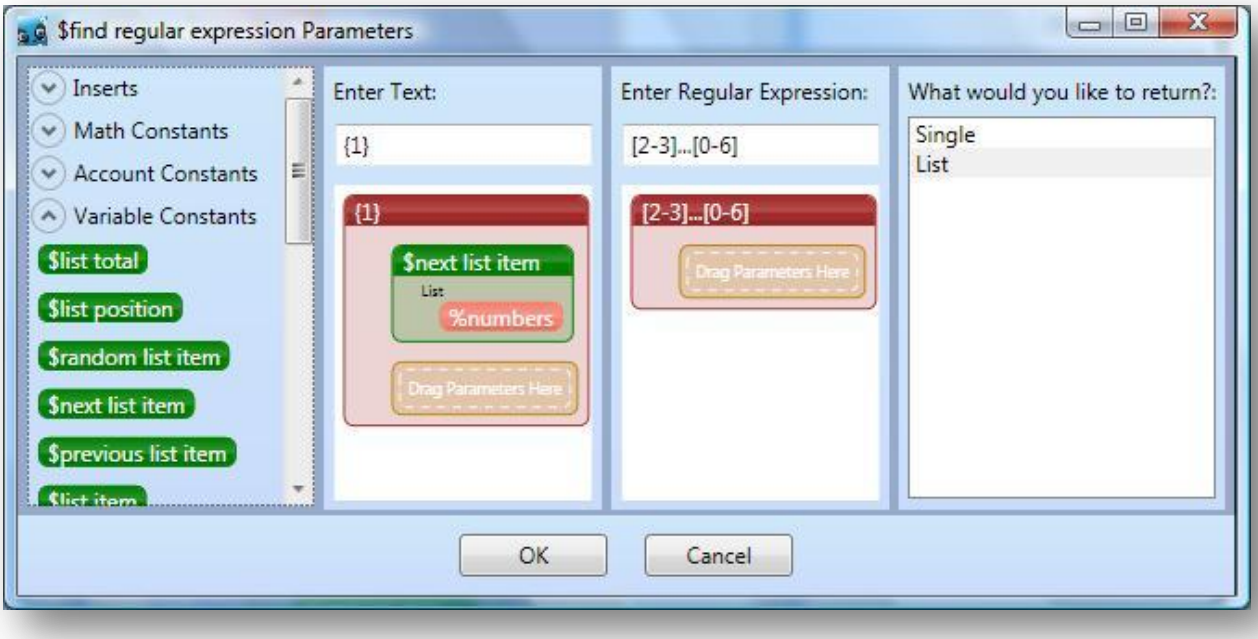
We are going to add the numbers that meet that criteria into a new list, and then display the list in the browser. We have a loop going which will go through each list item and analyze them based on our regex criteria. We drag in the add to list command, and when the parameter window pops up, we will select the find regular expression constant from under the text constants.

Another window will pop up. Under Text will go the item you are trying to analyze with the regex. Under Expression will go your regex, in this case:

[2-3]...[0-6]

Under return type, you will get to determine if you want the results return as a single item or as a list. We will select list for this example.

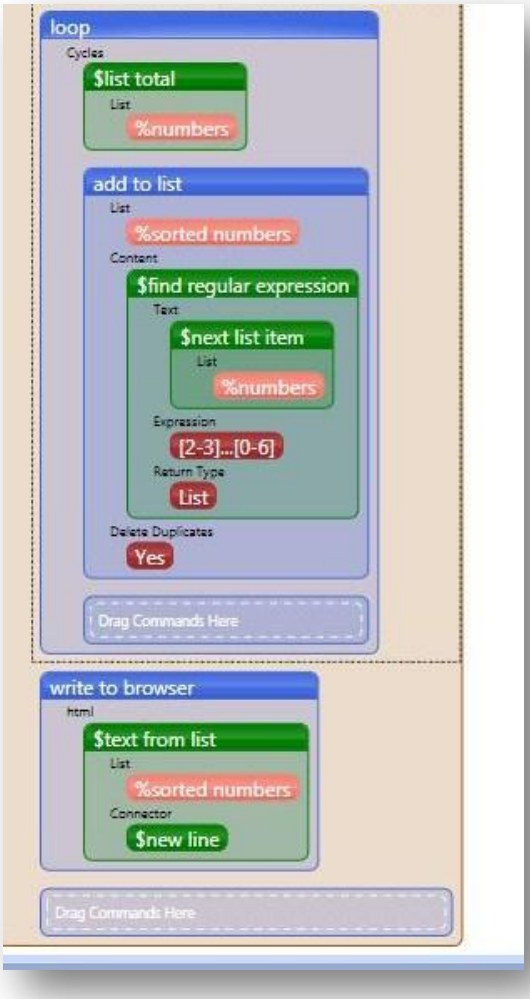
*(Dev License ONLY)



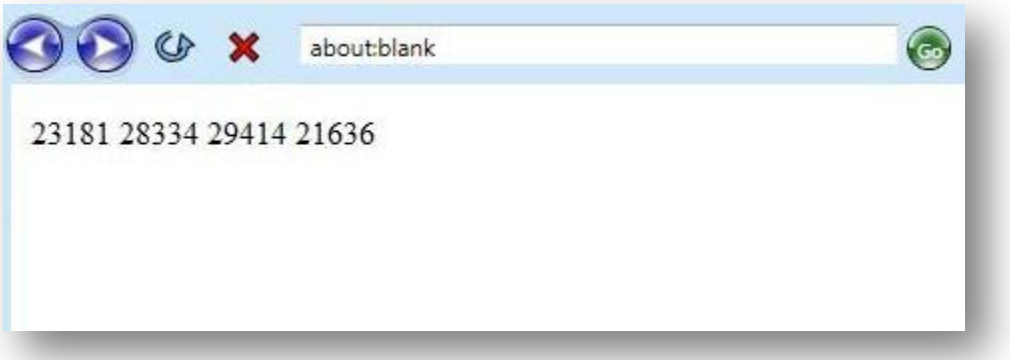
Click the appropriate oks to get back to the scripting area.

Add a write to browser command with a text from list constant. The connector in this case will be the new line constant (kind of like pressing enter).

*(Dev License ONLY)



When the entire script is run, we get the following results that meet regex criteria.



Replace Regular Expression

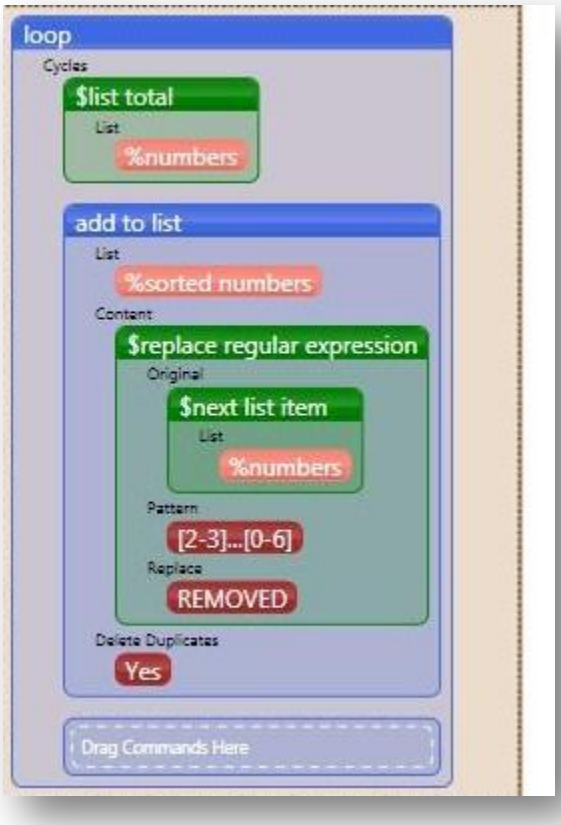
*(Dev License ONLY)

This allows users to find a specific element within a list or a simple word and replace the item using regular expression. This command works just like the find regex constant.

We can take a look at the last example. The only thing we would need to change in the previous example is the section within the code where the find regex constant is located within the add to list command.

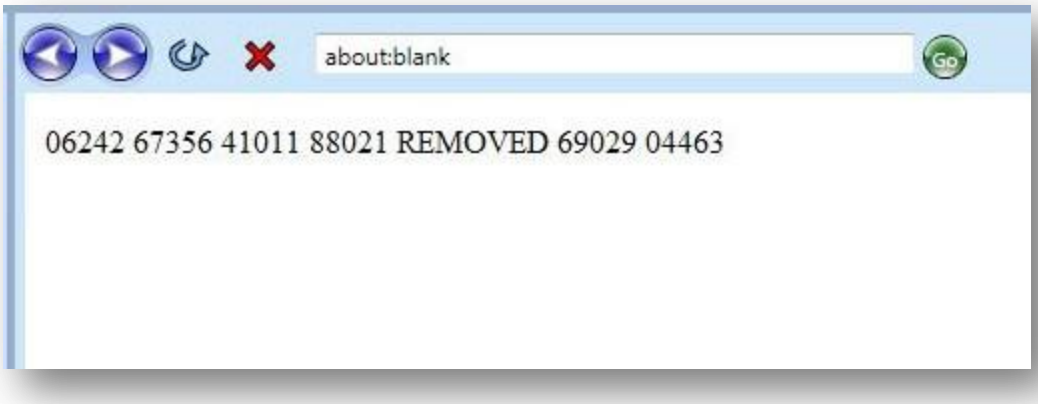
In the parameter window for the replace regex command, we will use the same regular expression (which will be inserted into the column labeled Pattern), and in the column under replace will go whatever you are trying to replace the items that meet the regex criteria with. Under original will still be the next list item constant.

So this is the only change we will see in the script:



When the script is run, the numbers that meet the regex criteria will be replace with the word “REMOVED”:

*(Dev License ONLY)



File Constants

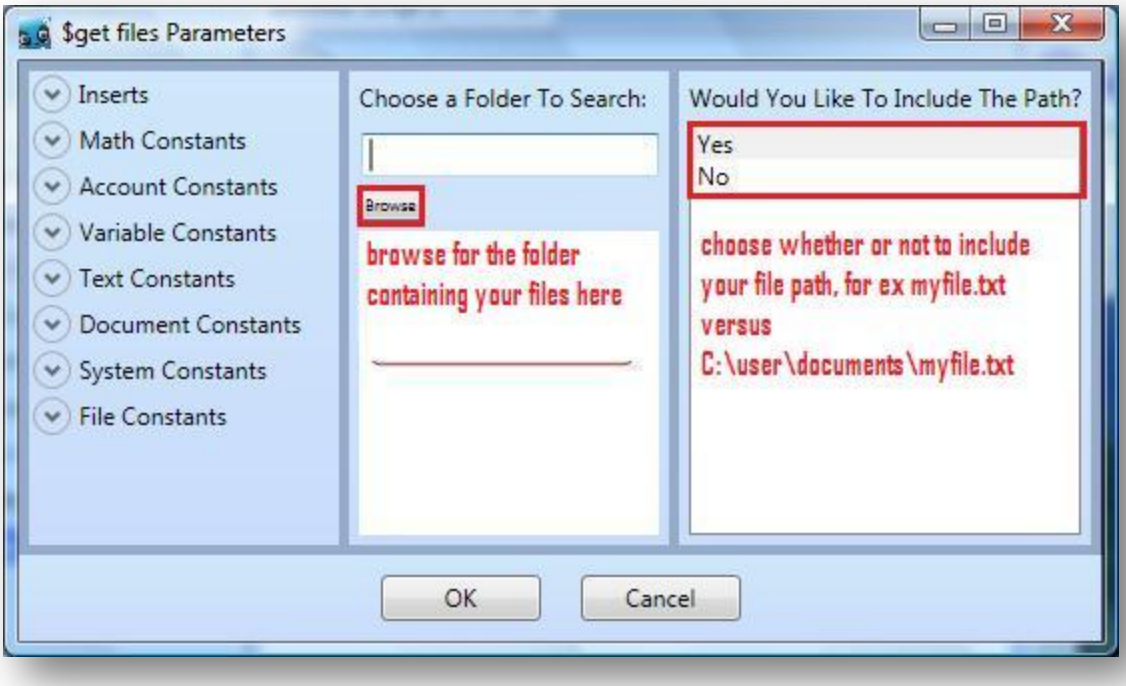
Get files

The Get Files constant returns a list of all files in any given directory.

In this example, we are looking into a folder called my folder and getting a list of all the files in the folder. I have added the get files constant to the save to file command, so that the list of the files gets saved to a .txt file for later use.

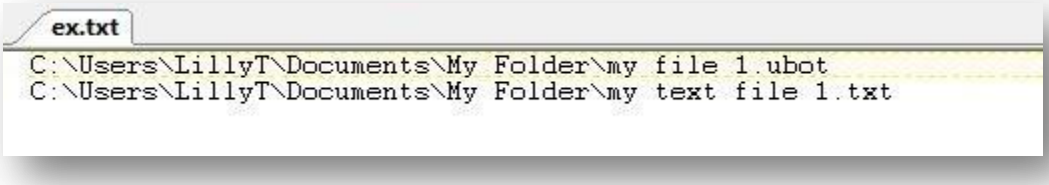
In the parameter window for the get files constant, you will browse for the folder you want the files from, and then select whether or not you want the file path included into the list of files.

*(Dev License ONLY)



Choosing whether or not to use the file path makes a big difference.

Selecting “Yes” for include file path option will give this result:



Selecting “Yes” for include file path option will give this result:

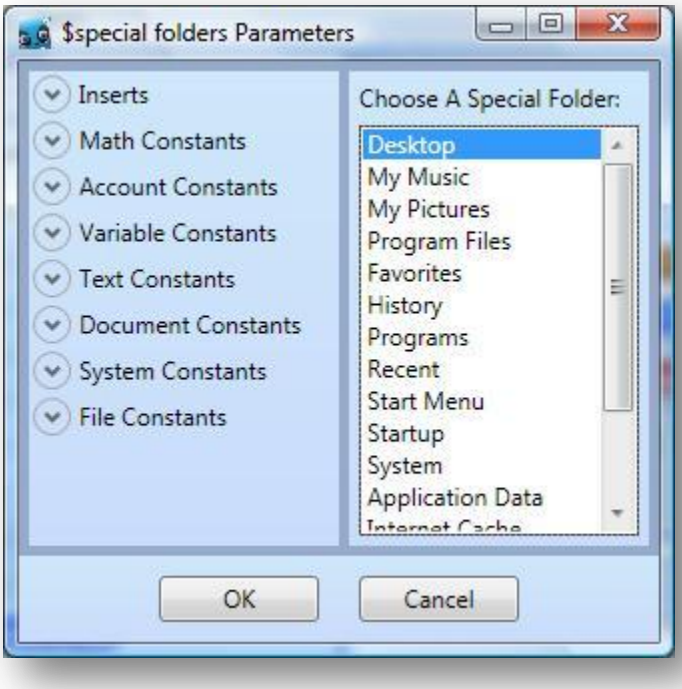
You can add the list of files to a list and then use each file as you need it with a variable constant.

*(Dev License ONLY)

Special folders

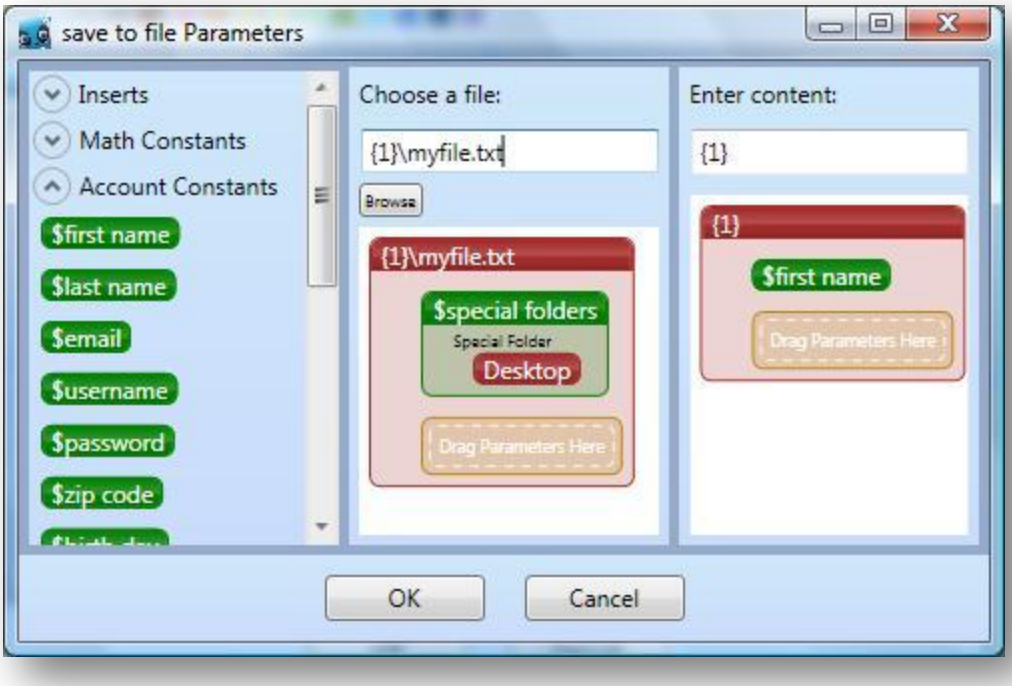
Shows the current directory of the computer's special folder.

This constant will use the special folder you designate in the parameter window after compiling. In this example, we are saving an item to a .txt file in a special folder. Our special folder in this example will be the desktop.



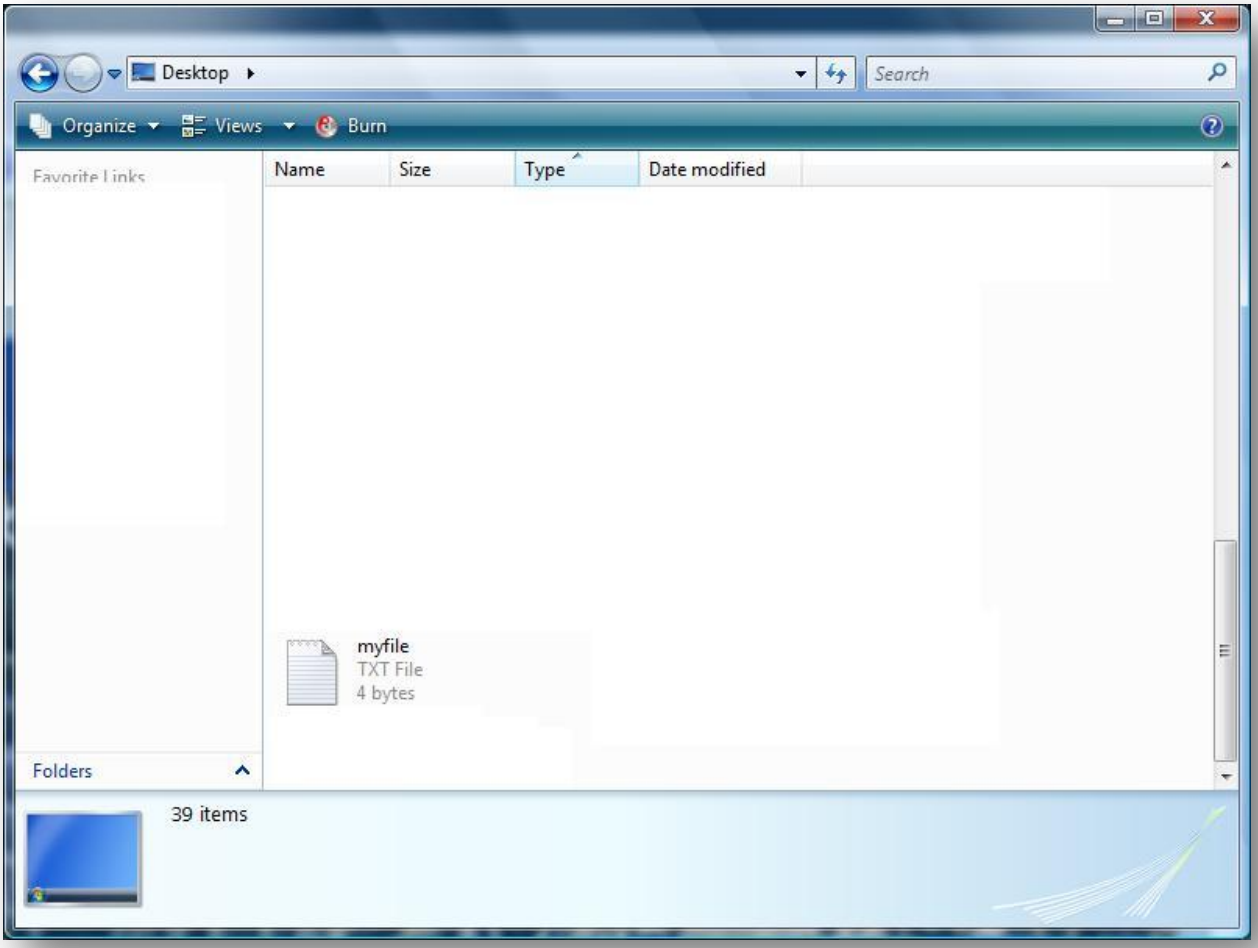
In the save to file parameter, we will drag in the special folder constant and then after we have designated what the special folder will be, we will name the file in this format:

*(Dev License ONLY)

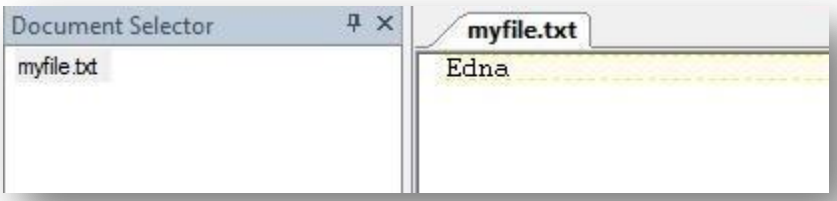


After inserting the item, which in this case is a random first name, we will click ok. When the script run, a file will be created on the desktop of the computer the compiled bot is in:

*(Dev License ONLY)



And then the list item will be added into the file:



*(Dev License ONLY)