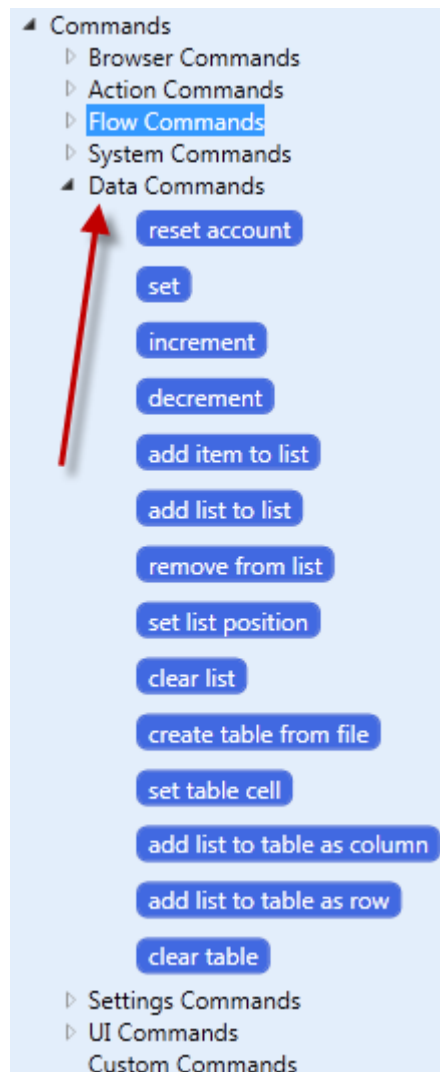


The UBot Studio

SCRIPT REFERENCE

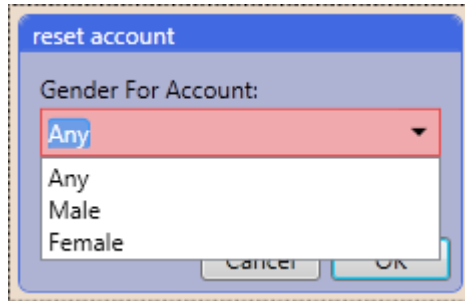
The Data Commands

The data commands are located in the fourth sub-menu under Commands:



Reset Account:

This command resets the account information to a new account. Therefore, if you are creating accounts using UBot's account information (i.e. First name, Last name, etc), for example, inside of a loop, you would place this command at the top of the loop and each cycle would generate new information. You have the option to choose male accounts, female accounts or either (randomly chooses one of those two):



SET:

This command allows you to store a value in a variable. Very simply drag the command into the scripting area, provide a name for the variable and give it a value. In this example I will set a variable called `#email` and provide an email address as its value:



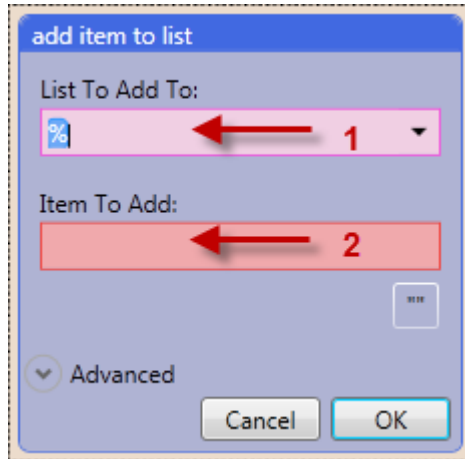
Anytime I need to call my email address in the script, I can simply call the variable that holds that value. For example, I can fill a form field with the variable `#email` instead of typing in the actual email address.

INCREMENT/ DECREMENT:

These commands add one to a variable and subtract one from a variable respectively. So if there is a variable set with the value 1 (we'll call it `#number`), and you increment `#number`, the new value will be 2. Likewise, if you decrement the value of `#number`, the new value will be 0.

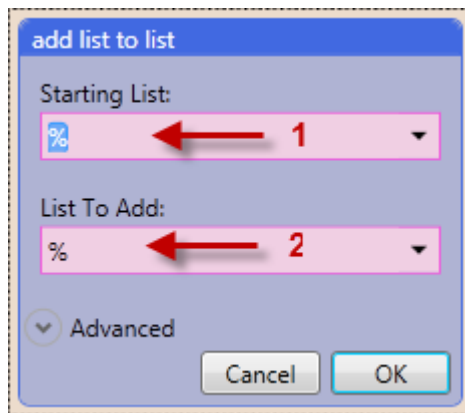
Add Item To List/Add List To List::

These two commands are a slight variation from the v3.5 add to list command. In version 4 there are now two options to add items to a list. You can choose to add a) a single item to a list using `$Add item to list:`



In this case simply choose a list you want to add the item to by typing in a name (if it's a new list) or choosing an existing list from the dropdown (1). Then choose an item to add to the list. Again, you can type in text, type in a variable name (i.e. "#username"). You can also add \$account data as well as other types of data by dragging them into the text field (2) from the parameters menus.

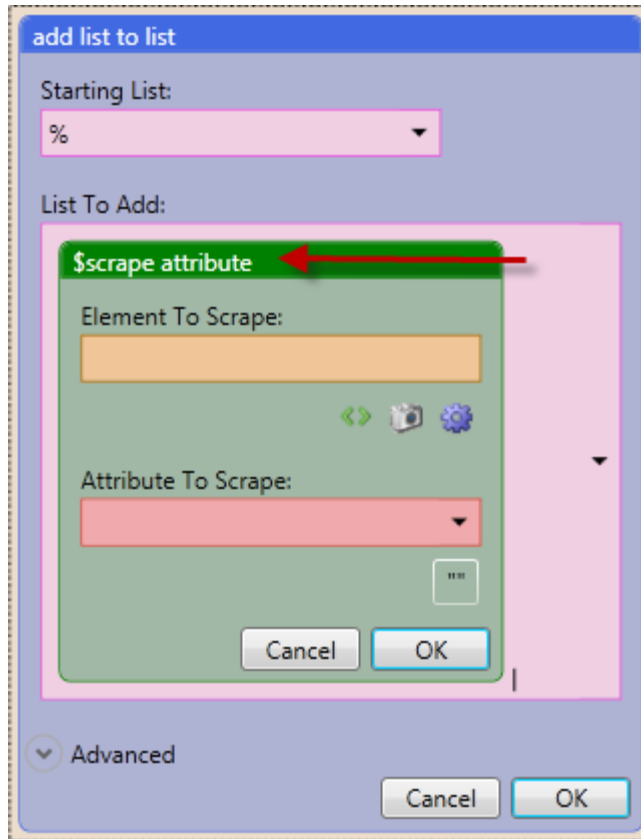
You also have the option to b) add a list to a list. This is a general term that means you can add a list to a list (literally) or to add multiple items to a list at once:




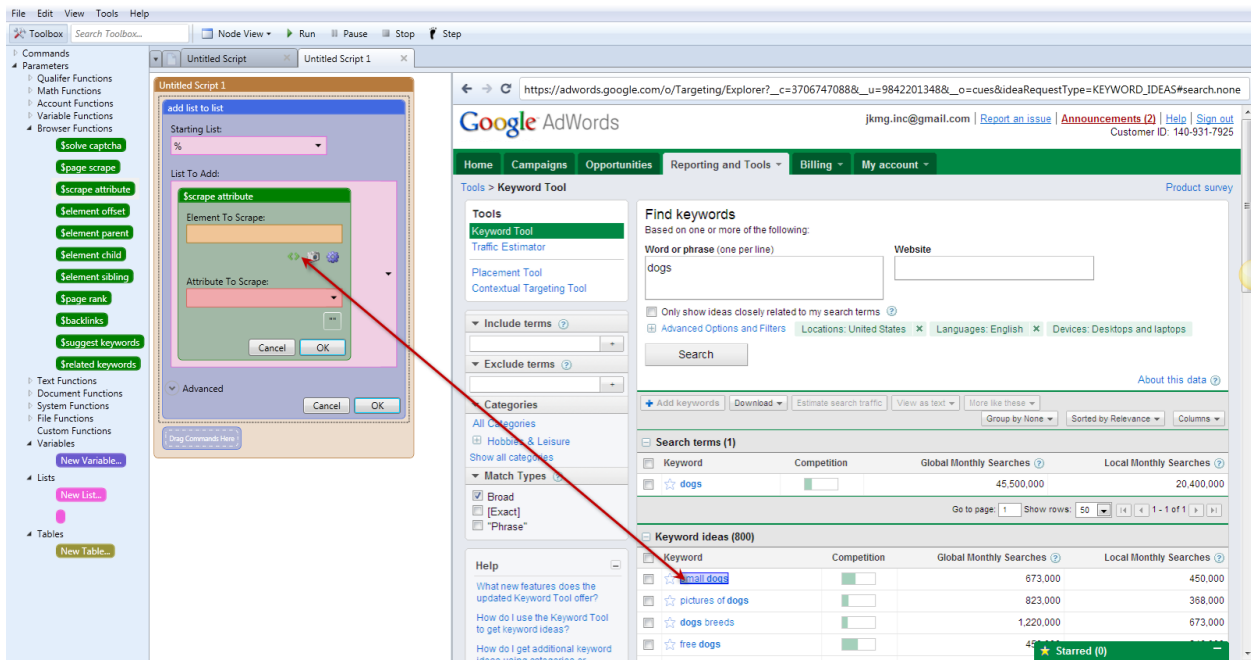
Just like in the add item to list command you will want to either name a new list or choose an existing list (1). In the second parameter ("List to Add") you can choose to add an existing list by typing in the name, selecting it from the dropdown (2) or most commonly drag a new parameter into the option field.

NOTE: This is where you are most likely to use the \$scrape attribute command to scrape data from web pages

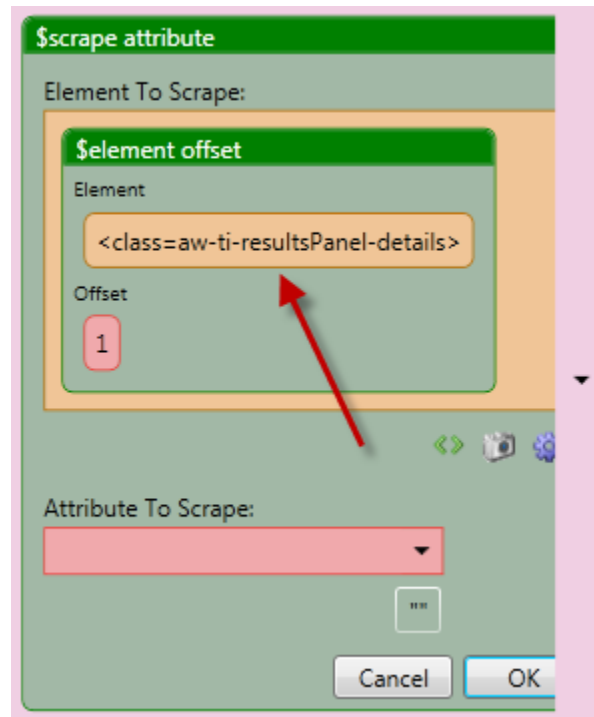
If you are choosing to scrape items to your list simply drag the \$scrape attribute command into the text field from the Browser Functions menu:



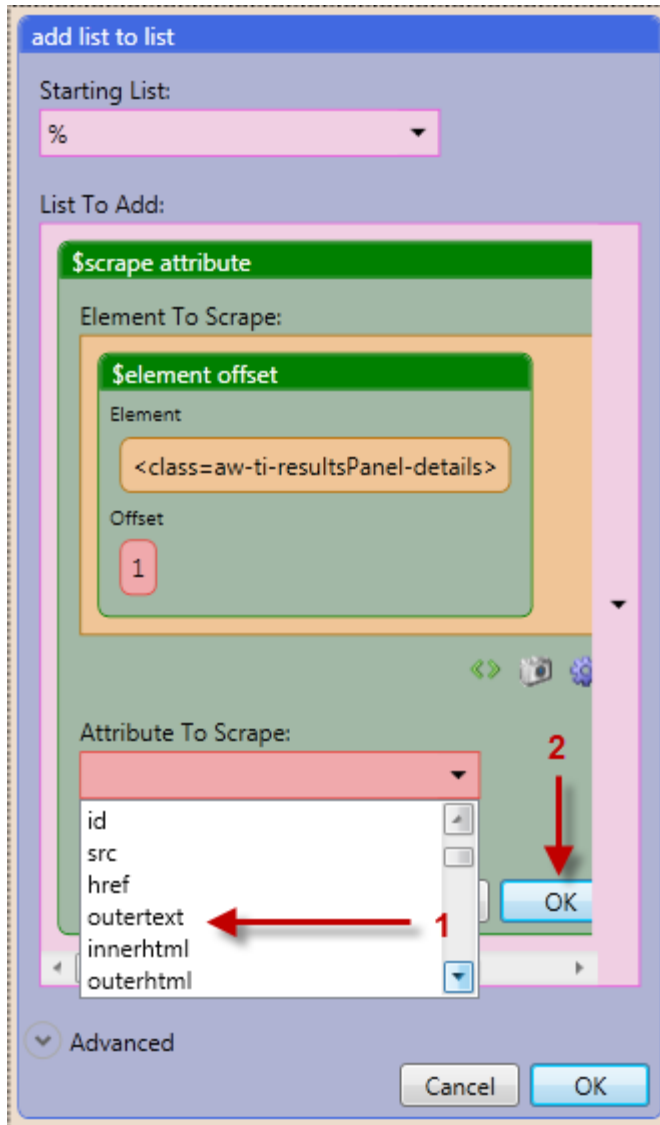
In this case you will need to first choose an element to scrape by selecting the element selector () and then clicking on an element in the browser:



This will load the attribute into the parameter field:

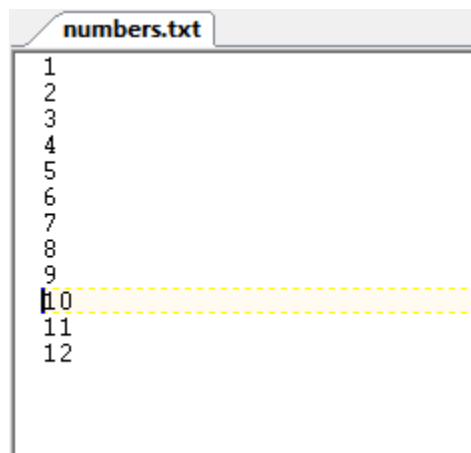


You will now need to choose which attribute you would like to scrape. In this case we will choose outertext. Click OK and you are done with this command:



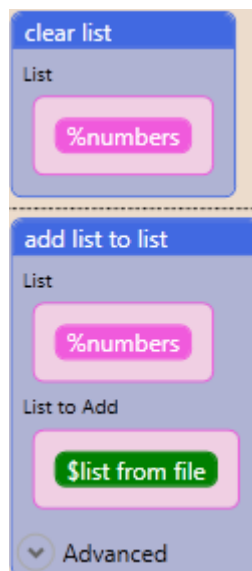
REMOVE FROM LIST:

This command removes a list item at a specified position. In this example I have created a text file with a list of numbers:



```
numbers.txt
1
2
3
4
5
6
7
8
9
10
11
12
```

We will add this list of numbers to a UBot list (\$using list from file), loop through the list, and remove any item that contains the number 1 in it. So we start by adding the list to a list in UBot:



clear list

List

%numbers

add list to list

List

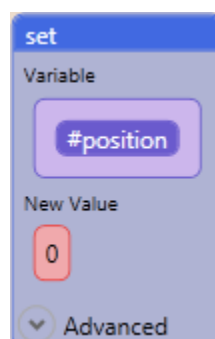
%numbers

List to Add

\$list from file

Advanced

From here we will set up a variable to control the list position and a loop with a condition in it:



set

Variable

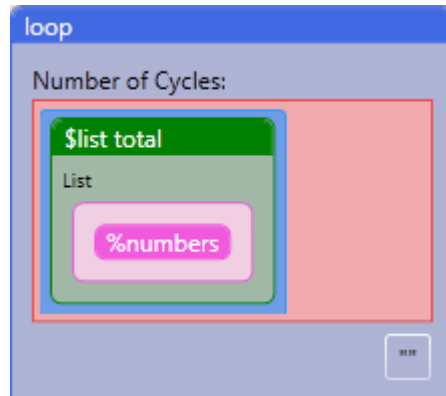
#position

New Value

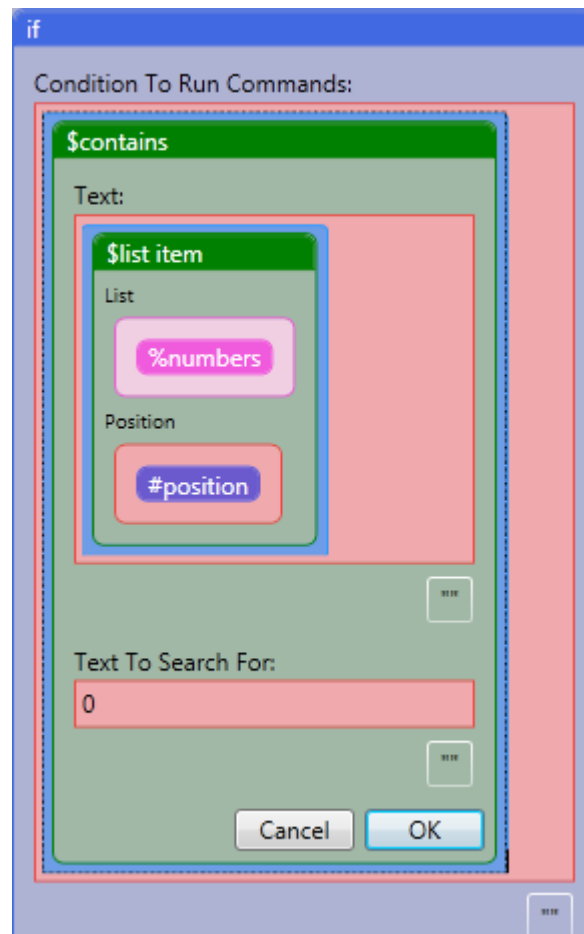
0

Advanced

The number of cycles in the loop will be the \$list total of the list we just created:

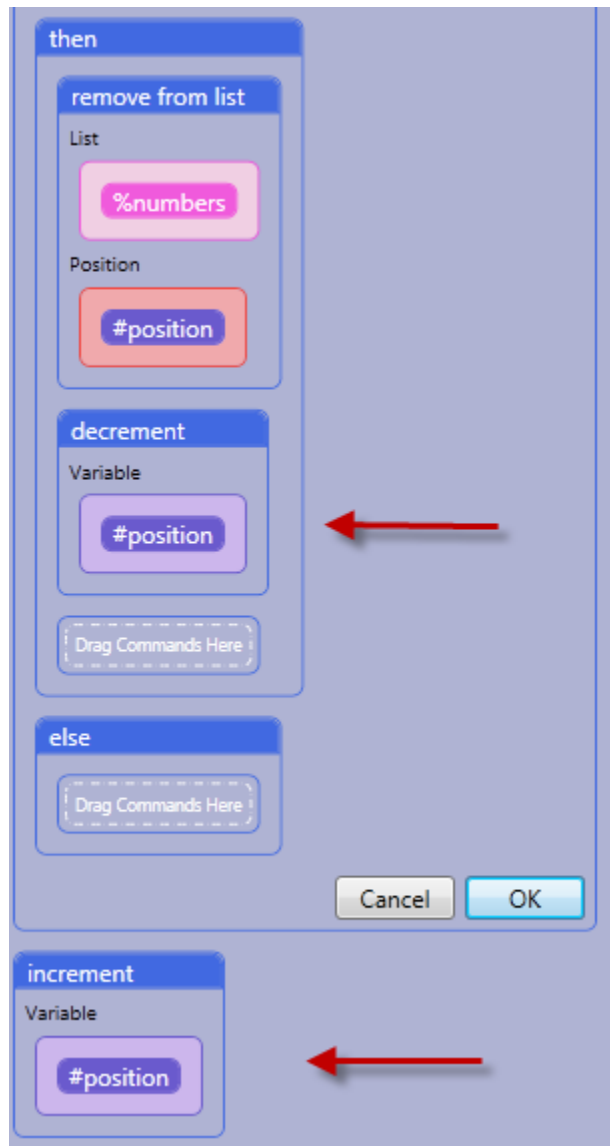


The condition will be a simple one. We will tell the script to check all the list items to see if any of them contain a zero (0). If there is a zero then remove that item from the list. Since our list contains the numbers 1-12 we should see the number 10 removed from the list:



If this condition is met the item will be removed, however, if the item is removed the total number of items on the list changes by one. Therefore, we need to decrement our variable before incrementing to the next list item. If we do not do this, we will skip an item and the script will report an out of range error.

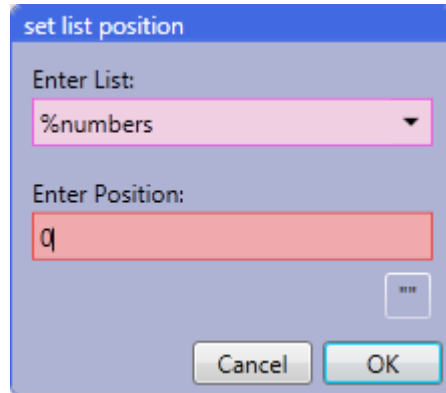
You can think of it like this...If an item in position 5 gets removed, then the item in position 6 becomes position 5. BUT, position five has already been evaluated, so it will move on to position 6 skipping the original position 6 item. By decrementing the variable immediately upon removal of an item, we avoid that issue altogether:



If an item is removed the variable will decrement and then increment. If no item is removed it will simply increment to the next position on the list. When this script is run, the number 10 will be removed from the list.

SET LIST POSITION:

This command sets the position of a list which is used by \$next list item. This command allows you to re-use a list by re-setting the position back to 0 (or any position you desire). Simply choose a list to set the position in and choose the position you desire:



The dialog box has a blue title bar with the text "set list position". Below the title bar, there are two input fields. The first is labeled "Enter List:" and has a dropdown menu with the text "%numbers" and a downward arrow. The second is labeled "Enter Position:" and has a text box containing the number "0". To the right of the text box is a small square button with three dots. At the bottom of the dialog box are two buttons: "Cancel" and "OK".

CLEAR LIST:

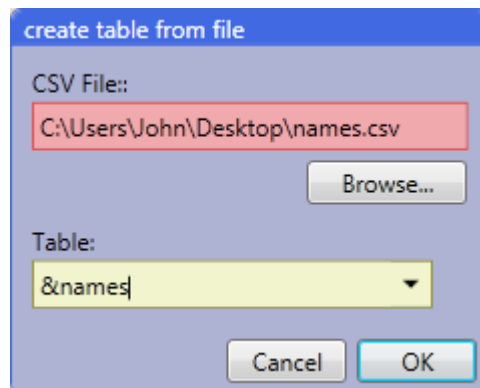
This command empties the contents of a list and resets the position to 0.

CREATE TABLE FROM FILE:

This command allows you to create a table based on a csv file. If you have a csv file with rows or columns of data you can load that data into a table to be used by UBot as you desire:

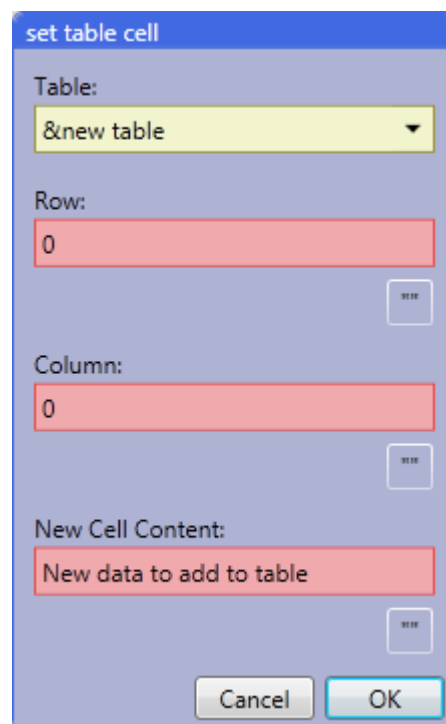
	A	B	C
1	Alans Music Center Inc	8510 Ia	
2	Record City	Serving	
3	Music Trader	3112	
4	Music Rental Co & Pawn	5960 El	
5	Mexicano Musical	2375 Ulric	
6	Access Music	1537	
7	Wal Mart	4840	
8	Tasha's Music City	1853 5th	
9	Apex Music Co	6210 El	
10	Centre City Music	1033 6th	
11	El Portal Musical	2121	
12	Kenn Owen Music	27744 PO	
13			
14			
15			

Simply drag the command into the scripting area, give the table a name and provide the location of the csv file. The table will then be created for you. Remember to use the "clear table command" so you do not append data to the table on subsequent script runs:



SET TABLE CELL:

This command allows you to enter data into any specified cell of any table. This can apply to a pre-existing table, or you can create one using this command:



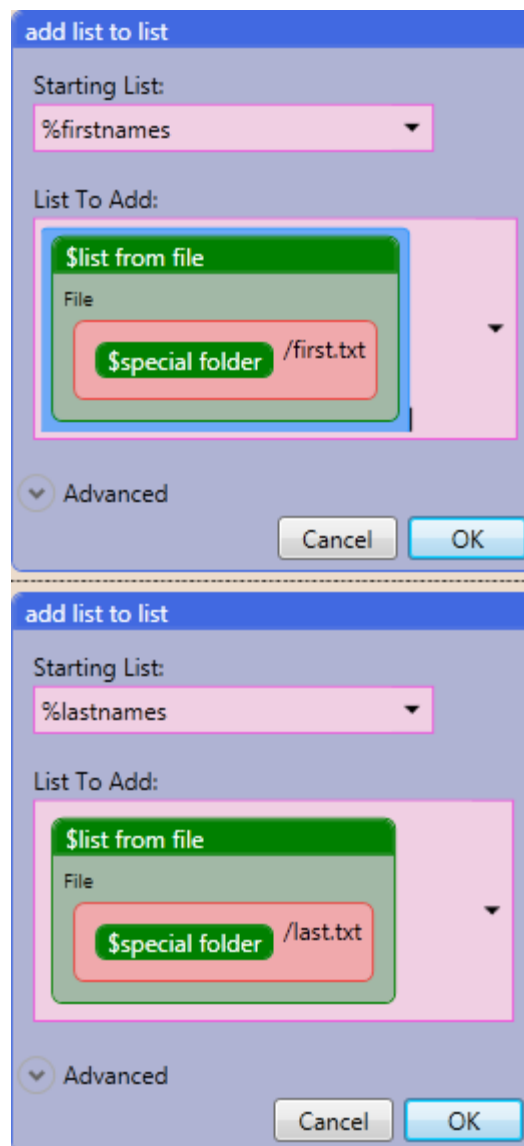
As you can see we provided a name for the table (&new table), the row and column where we want to add the data, and the data itself. Click OK, and run the script, and the new table will contain the data you entered.

ADD LIST TO TABLE AS COLUMN/ROW:

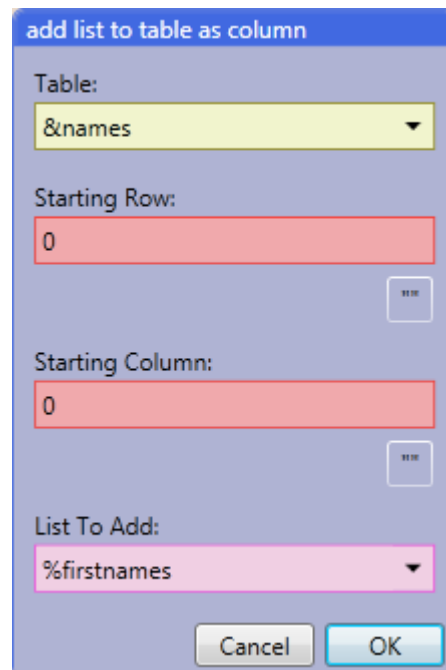
These commands add lists to tables as either columns or rows in a specified location. In this example we will create two lists and add them as both columns and rows in a table, save them to a file and check the results. The first list will contain 5 first names and the second list will contain 5 last names:

John	Smith
Pam	Jones
Susan	Peters
Carolyn	Rogers
Robert	Johnson

Now add the file contents to their respective lists:



Click OK on both nodes and drag the Add List to Table as Column command into the scripting area:



add list to table as column

Table:
&names

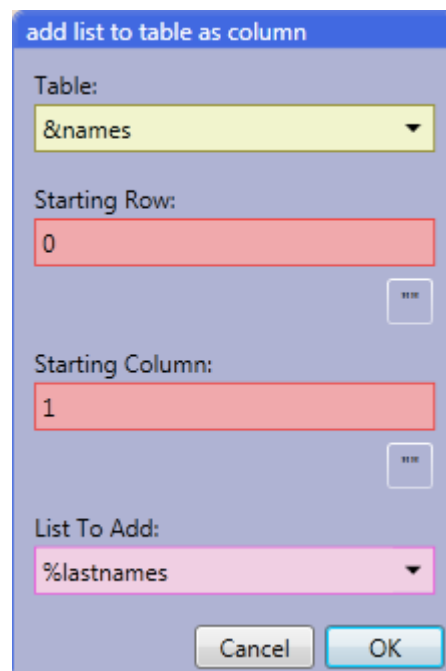
Starting Row:
0

Starting Column:
0

List To Add:
%firstnames

Cancel OK

The second list will be added the same way except it will be added to column 1, not column 0 (Tables are 0 based so the first column and row are always 0 not 1):



add list to table as column

Table:
&names

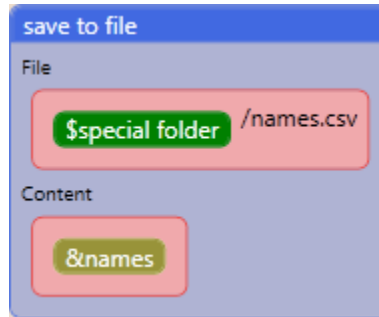
Starting Row:
0

Starting Column:
1

List To Add:
%lastnames

Cancel OK

Now we will save the results to a csv file and check them:



Run the script and open the file you just created on your desktop. You should see the following:

	A	B	C
1	John	Smith	
2	Pam	Jones	
3	Susan	Peters	
4	Carolyn	Rogers	
5	Robert	Johnson	
6			
7			

If You added the lists to the table as rows instead of columns, your results would look like this:

	A	B	C	D	E
1	John	Pam	Susan	Carolyn	Robert
2	Smith	Jones	Peters	Rogers	Johnson
3					
4					

CLEAR TABLE:

This command empties the contents of a table.