

## AOZ Studio Beta - Bug #697

### Screen copy for high screen numbers shifts the copy left.

03/22/2021 11:00 AM - Paul Davis

<b>Status:</b>	Rejected	<b>Start date:</b>	03/22/2021
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Francois Lionet	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Target version:</b>			
<b>Affected version:</b>	1.0.0 (B7)		

#### Description

When I run the code below, the content of the copied screen moves left.  
The higher the screen number, the greater the shift.  
Unless I am doing something wrong...  
This effects Beta6 (and earlier versions) there is no option for that version here yet.

Thanks  
Davo.

(I also go by the name Davo.Sydney, but I had to re-register here in the bug system as I couldn't log on using my other account.)

```
#splashScreen:false
SHidden=10:SVisible=2           // The higher the screen numbers, the higher the movement
Screen Open SHidden,800,600,32,Lowres // Screen 1 is the hidden background (Source)
Screen Open SVisible,800,600,32,Lowres // Screen 2 is visible (Output)
Screen SHidden : Ink 1 : Cls 0 : Screen Hide SHidden // Clear Screen 1 (Hidden)

Print "This text shouldn't move"
Print "(Press Left Mouse a few times)"

Screen SVisible : Ink 1 : Cls 0 // Clear Screen 2 (Visible)
Screen Copy SHidden,SVisible // Copy Screen 1 (Hidden) to Screen 2 (Visible)

Do
Screen SVisible // Do everything on Screen SVisible (If this is removed. The screen copy doesn't shift)
If Mouse Key = 1
While Mouse Key = 1
' This is where I was adding graphics to the source screen
Wait 0.01 // Acts as a replacement for the old WaitVBL, (works on my computer)
Wend
Screen Copy SVisible,SHidden // Apparently the screen copy shifts the screen left
// Eg: Screen Copy x,y
// The higher the source screen (x) the greater the movement/offset.
// If in line 2 SHidden is set to 0, there is no movement.
// I guess in the Java Routine, the source screen number (eg: SHidden) is used
// in the screen copy loop. (aka added to the start of the loop creating a
// shift left of x pixels)
//I hope this makes sense
Wait 0.01
End If
Loop
```

#### History

#1 - 03/31/2021 02:38 PM - Francois Lionet

- Status changed from New to Resolved

- Assignee set to Francois Lionet

Fixed!  
Thank you for your help! :)

## #2 - 04/01/2021 10:46 AM - Brian Flanagan

- Status changed from Resolved to Feedback

- Affected version changed from 1.0.0 (B5) to 1.0.0 (B7)

Still fails. Screen Copy still shifts the screen to the left. (Remember to click the mouse to test.)

Perhaps adding an X across the screen will help illustrate the problem:

```
#splashScreen:false
SHidden=10:SVisible=2 // The higher the screen numbers, the higher the movement
Screen Open SHidden,800,600,32,Lowres // Screen 1 is the hidden background (Source)
Screen Open SVisible,800,600,32,Lowres // Screen 2 is visible (Output)
Screen SHidden : Ink 1 : Cls 0 : Screen Hide SHidden // Clear Screen 1 (Hidden)
Ink 1 : Draw 0,0 To Screen Width-1,Screen Height-1 : Draw 0,Screen Height-1 To Screen Width-1,0 // BJB added t
his

Print "This text shouldn't move"
Print "(Press Left Mouse a few times)"

Screen SVisible : Ink 1 : Cls 0 // Clear Screen 2 (Visible)
Screen Copy SHidden,SVisible // Copy Screen 1 (Hidden) to Screen 2 (Visible)

Do
Screen SVisible // Do everything on Screen SVisible (If this is removed. The screen copy doesn't shift)
If Mouse Key = 1
While Mouse Key = 1
' This is where I was adding graphics to the source screen
Wait 0.01 // Acts as a replacement for the old WaitVBL, (works on my computer)
Wend
Screen Copy SVisible,SHidden // Apparently the screen copy shifts the screen left
// Eg: Screen Copy x,y
// The higher the source screen (x) the greater the movement/offset.
// If in line 2 SHidden is set to 0, there is no movement.
// I guess in the Java Routine, the source screen number (eg: SHidden) is used
// in the screen copy loop. (aka added to the start of the loop creating a
// shift left of x pixels)
//I hope this makes sense
Wait 0.01
End If
Loop
```

FYI Paul, Wait Vbl still works. (It's two words.)

```
Wait Vbl
```

## #3 - 04/05/2021 01:20 PM - Paul Davis

Brian, Thanks for the information, I will use Wait Vbl in my programs. I haven't spent as much time on AOZ as I had planned, Hopefully I get some spare time soon...

## #4 - 04/06/2021 03:21 PM - Francois Lionet

- Status changed from Feedback to Rejected

When I said "Fixed", I meant up to the limit possible with the possibilities of Javascript Canvas. Precision goes away in the Draw Image process, impossible to correct without doing everything by hand, and it will be MUCH slower min Aoz resolutions.

Rejected for the moment.

## #5 - 04/14/2021 11:31 AM - Brian Flanagan

Paul, I found the problem! (Sorry I can't mark this "Resolved".)

The syntax you're using on Screen Copy is incorrect, but is being parsed anyway.

Since the code immediately following the screen ID is numeric instead of the **To** keyword, the transpiler is assuming you're using a different form of the command.

The correct syntax you want is:

```
Screen Copy sourceScreen To destScreen
```

(Use the keyword **To** instead of the comma.)

In the other forms, the parameter immediately following the screen ID is the starting X position for the copy, so that's why you're seeing the screen

shift! (The transpiler is just assuming the default values for all of the other parameters.)

```
Screen Copy sourceScreen,sourceX,sourceY,sourceWidth,sourceHeight To destScreen,destX,destY,destWidth,destHeight
```

In other words, it's doing exactly what you told it to! ;-)

Here's your code modified to work properly:

```
#splashScreen:false
SHidden=100:SVisible=2 // The higher the screen numbers, the higher the movement
Screen Open SHidden,800,600,32,Lowres // Screen 1 is the hidden background (Source)
Screen Open SVisible,800,600,32,Lowres // Screen 2 is visible (Output)
Screen SHidden : Ink 1 : Cls 0 : Screen Hide SHidden // Clear Screen 1 (Hidden)
Ink 1 : Draw 0,0 To Screen Width-1,Screen Height-1 : Draw 0,Screen Height-1 To Screen Width-1,0 // BJJ added this

Print "This text shouldn't move"
Print "(Press Left Mouse a few times)"

Screen SVisible : Ink 1 : Cls 0 // Clear Visible screen
Screen Copy SHidden To SVisible // Copy Hidden screen to Visible

i=0
Do
  Screen SVisible // Do everything on Screen SVisible (If this is removed. The screen copy doesn't shift)
  If Mouse Key = 1
    Locate 1,3 : Print "Iteration: ";i

    Repeat // Wait for click loop
      Wait Vbl
    Until Mouse Key <> 0 // Allow ANY mouse button.
    Repeat Until Mouse Key = 0 // Release click
// In 1.0.0 (B7) You'll be able to use:
// Wait Click // instead!

    Inc i
/* NOTE: This code does NOT wait for a mouse click, so the program will do the Screen Copy continuously!
   (Use Repeat statements shown above instead.)
   While Mouse Key = 1
' This is where I was adding graphics to the source screen
    Wait 0.01 // Acts as a replacement for the old WaitVBL, (works on my computer)
  Wend
*/

// Screen Copy SVisible,SHidden // <<< The INCORRECT SYNTAX causes the problem.
// Screen Copy SVisible To SHidden // <<< Use THIS instead. ;-)
// Try both ways to see the difference! BJJ

// Eg: Screen Copy x,y
// The higher the source screen (x) the greater the movement/offset.
// If in line 2 SHidden is set to 0, there is no movement.
// I guess in the Java Routine, the source screen number (eg: SHidden) is used
// in the screen copy loop. (aka added to the start of the loop creating a
// shift left of x pixels)
//I hope this makes sense
  Wait Vbl
End If
Loop
```

**Francois:** Please change the transpiler to ensure that *if* the keyword **To** is missing that a proper error message is displayed instead of attempting to parse the command. ;-)

In other words, IMO, when using the form:

```
Screen Copy sourceScreen,sourceX,sourceY,sourceWidth,sourceHeight To destScreen,destX,destY,destWidth,destHeight
```

...at the least, **"To screenDest"** should be mandatory.

(The defaults could be assumed for the rest of the parameters, as they are now.)