

AOZ Studio Beta - Bug #719

Hex\$(n) can't handle negative numbers.

04/11/2021 01:40 PM - Brian Flanagan

Status:	Feedback	Start date:	04/11/2021
Priority:	High	Due date:	
Assignee:	Francois Lionet	% Done:	0%
Category:		Estimated time:	1:00 hour
Target version:			
Affected version:	1.0.0 (B10) u16		
Description			
Example:			
<pre>Print Hex\$(-1) // FAILS. Produces illegal function call.</pre>			

History

#1 - 04/11/2021 06:39 PM - David Baldwin

Fixed to work as 'AMOS' used to. Hex\$(-1) = "FFFFFFFF".

#2 - 09/15/2021 02:38 AM - Brian Flanagan

- Status changed from New to Feedback

- Affected version changed from 1.0.0 (B7) to 1.0.0 (B10) u16

Re-tested in 1.0.0 (B10) u16. It's failing in both AOZ and Amiga manifests.

Yes, in Amiga mode, \$FFFFFFFF should be -1, and any number with the high bit set would be negative. This, however, must have been re-broken, since it's not working in ether manifest now.

The Hex\$() function can't print negative hex numbers.
There is also a bug in returning a very large integer via inline JavaScript.

ALSO, keep in mind that for AOZ mode, we should allow for much larger (and smaller) numbers. We already have the ability to handle very large integers (48 to 53 bits), and AOZ already handles these fine except for the hex functions.

IMO, we should have an Option in Amiga mode to use extra-long Integers. It looks like we could safely set a limit at 48-bits. We could go up to 53-bits, but it could get tricky, and there may be limitations.

Try this example:

```
Print "53-bit Integers (max/min): (MAX_SAFE_INTEGER / MIN_SAFE_INTEGER) "  
MaxSafeInt = { Number.MAX_SAFE_INTEGER }  
MinSafeInt = { Number.MIN_SAFE_INTEGER }  
Print MaxSafeInt, Hex$(MaxSafeInt)  
Print MinSafeInt ' Hex$(MinSafeInt)  
Print  
Print "48-bit Integers (max/min): "  
Fmt$="-#####"  
X=$FFFFFFFF : Print Using Fmt$,X, : Print Hex$(X,12)  
Y=-$FFFFFFFF : Print Using Fmt$,Y, ' Bug in Hex$, can't print negative  
Print  
Print  
Print "Calculation on large integers:"  
Z=9000000000000000 : W=543 : V=-123456789012345  
Print Using Fmt$, (Z+V)/W, : Print Hex$(Z+V/W,12)  
Print  
Print "32-bit Integers:"  
Print $FFFFFFF ' should be -1 in the Amiga manifest.  
          ' or 4294967295 in the AOZ manifest.  
Print $7FFFFFFF ' should be 214783647 in either manifest.
```

#3 - 09/21/2021 08:43 AM - Francois Lionet

- Status changed from Feedback to Resolved

- Assignee set to Francois Lionet

Fixed.

#4 - 09/22/2021 12:11 AM - Brian Flanagan

- Status changed from Resolved to Feedback

THANK YOU! It's great to have the extra long integers functional on the Hex\$() function, *however*, in the Amiga Manifest, *by default*, they should be limited to 32-bit integers in the same method that the Amiga did.

I see you have the extra long integers working in Amiga too, and that's great, but that should really be **optional** with a tag or an instruction to turn them on.

For example:

```
#longIntegers: true
```

Could turn them on, but again, by default, it should follow the Amiga standard of a 32-bit signed integer. (Of course, it would be the opposite in AOZ mode. Long integers would be on by default.)