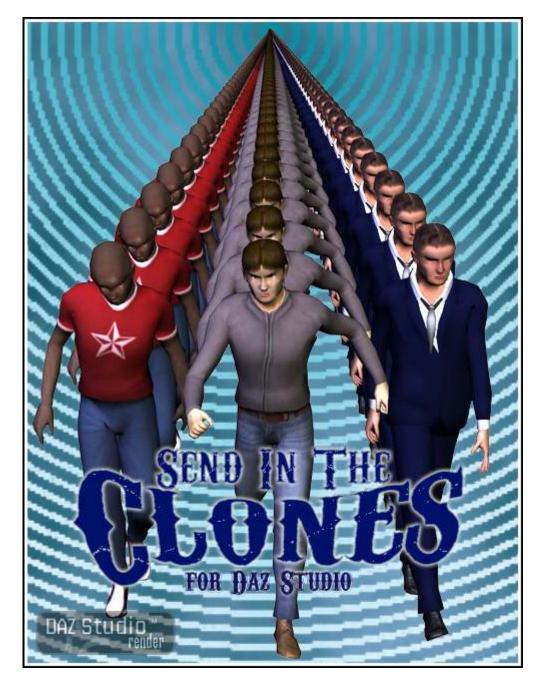
## Send In The Clones Tutorial

By DraagonStorm



Version 1.0

Thank you for purchasing **Send In The Clones**. I know you're going to love using this prop duplicating script in DAZ Studio 2 or DAZ Studio 3.

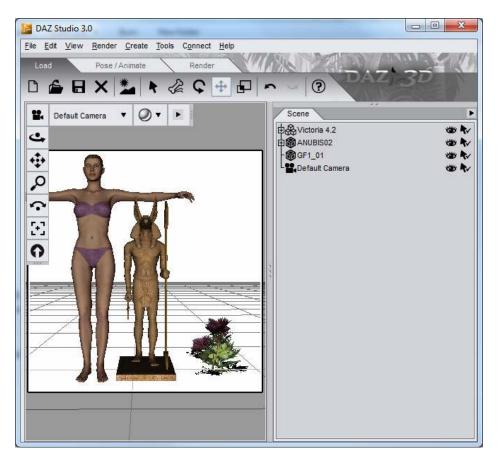
First , here are some notes on this script and your workflow. Please be aware of the limitations of your computer's memory. There is no magic memory saving coding in this script. If you make 100 clones of an object, it affects your scene just as if you loaded an instance of that object manually 100 times. If your computer can't handle you loading them manually, it can't handle you loading them from this script.

There is one memory saving thing that is done which each clone of the parent shares the parents materials. With this you will not be able to change the material setting of the clones individually.

So enough of all this, and on with the tutorial!

## **Knowing What You Can Clone**

The first thing you need to know is that this script will only work with 'single node objects'. "What are single node objects?"



Above you will see that I have a screen shot of a DS3 session with 3 things loaded into the screen. Ignoring the Camera... you see two different icons; the triple box, and the single box.

Firstly, the triple box is a figure, and cannot be used with the **Send In The Clones** product. A figure can be exported as Wavefront.obj, which then converts the figure to a single node object. Then you can import that obj file and use it with **Send In The Clones**. You will not be able to change the pose of this single node object, so be sure to set the pose and morphs before you export. And beware of the size of the object and you computer's limitation. Your mileage will vary.

Second is a single box, which is an object. But in the case of the example above, you would not be able to use the ANUBIS02 object with **Send In The Clones**. Why? To the left of the single box there is a plus sign in a box. This means that the object was loaded with other objects parented to it. Which makes this **NOT** a

single node object. You can 'unparent' the parented objects and then use the object in **Send In The Clones**.

The third item in the single box, GF1\_01, is a single node object, and can be cloned.

## The Send In The Clones Interface

The first box is where you select the type of cloning you want to do. As you select which type, the screen will change. Reading the captions of the selection boxes will guide you to making your choices. Go ahead and experiment. I do suggest to start with, that you use small/low poly objects, until you learn what **Send In The Clones** can do within the limitations of your computer. Also, keep in mind the known issues that are noted at the beginning of this document.

Values are changed by either moving the sliders left or right, or by clicking on the number, and typing in a value. If the slider/number is grayed out, then that value is not used with the type of cloning that has been selected.

The value ranges are:

- Inline Cloning:
  - Number of Copies (up to 250)
  - $\circ$  xAxis, yAxis and zAxis offset to range of -5000 to 5000
  - Overall Scale, xScale, yScale and zScale offset to range of 0 to 1000
  - xRotation, yRotation, and zRotation to range of -180 to 180
- Matrix Cloning:
  - Number of Copies along xAxis, yAxis and zAxis (up to 60 on each Axis)
  - xAxis, yAxis and zAxis offset to range of -5000 to 5000
  - xScale, yScale and zScale offset to range of 0 to 1000
  - xRotation, yRotation, and zRotation to range of -180 to 180
- Random Cloning:
  - Number of Copies (up to 250)
  - Overall Scale, xScale, yScale and zScale offset to range of 0 to 1000
  - Minimum xAxis, yAxis and zAxis offset to range of -5000 to 5000
  - Minimum xRotation, yRotation, and zRotation to range of -180 to 180
  - Maximum xAxis, yAxis and zAxis offset to range of -5000 to 5000
  - Maximum xRotation, yRotation, and zRotation to range of -180 to 180

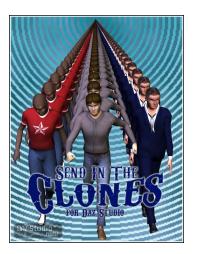
\*Note if the user enters a minimum that is larger then the maximum, the script will swap the values.

Screen Shot of In Line selected:

	and the owner where the									
SELECT TYPE OF CLONING	<u>1</u>									
In Line			O Matrix				O Randor	n		
UMBER OF CLONES [Set the	number of clone	es to make]								
		N/	A					N/A		
- 0	+	1	- 0		+		- 0		+	
RANSLATION [The pixel tran	slation from the p	previous cl	one]							
		ут						zTran		
	+	3		0		3		0	+	3
OTATION [Rotation from the	zero point on all o	clones]								
		yR	otation					zRotation-		
	• •	0.00				0.00		-0	+	0.00
CALE (*Zero value means n	o change to curre	ent scale)								
	-xScale				yScale				zScal	e
- 0	0.00 -	0	- +	0.00	- 0		+ 0.00	- 0	+	0.00
ANDOM MINIMUM [Only use										
	d for random clon	ing]								
	d for random clon	ung)	-	0					+	1:
0			-	0		,		-0-	• •	ţ.
0				0		1			+	5. 0.00
0000		99) (200)		0				0	••	1. 0.00
() () ANDOM MAXIMUM [Only us		99) (200)		0				0	•	4. 0.07
0000		99) (200)		0				0	•	ti ana f
0000		99) (200)		0					•	ti Dino Ti
0000		90 (200)		0				0 0	•	1; 0:00 1; 0:00
0000	+ + ed for random clo	21) (200) (ning)		0	••	ā.co		0 0	+	ñ

This will make the number of clones input, offset by the values input in the TRANSLATION box of the previous clone.

Rotation and Scale changes are constant.

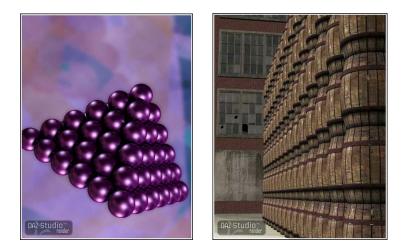


Screen shot of Matrix selected:

	11-12-12-12										
SELECT TYPE OF	CLONING -										
				🛞 Matrix				O Rando	m		
UMBER OF CLONE	S [Set the nu	umber of clor	nes to mak	e at each axis]							
			)						zAxis		
- 0			1	- 0		• •	1	- 0		+	1
TRANSLATION [The	pixel transla	ation from the	previous	clone]							
			)	Tran					zTran		
	-0	+	3		-0-		3			• •	3
ROTATION [Rotation	from the zer	ro point on al	l clones]								
			)	Rotation					zRotation-		
-	-0	+	0.00		-0-		0.00		-0	•+	0.00
SCALE (*Zero value	means no c	hange to cu	rrent scale	:)							
·	xs	cale				vScale				zSca	le
0											
- 0	+	0.00	- 0	-+	0.00	- 0		+ 0.00	- 0	+	0.00
	+ Only used fr			•	0.00	- 0		+ 0.00	- 0	•	0.00
	Only used fo			-+	0.00	- 0		+ 0.00	- 0	•	0.00
	+ Only used fo				0.00		8	+ 0.00	- ()	•	0.00
	Only used fo		on <mark>ing</mark> ]	•				+ 0.00		•••	
	Only used fo		oning]		0		8	+ 0.00		+	
	Only used fo	or random ck	on <mark>ing</mark> ]	-	0			+ 0.00	0	+	
	••	or random ck	oning]	-	0		8	+ 0.00	0	+	
RANDOM MINIMUM [	••	or random ck + +	oning]		-0-0	•	ъ 9.00	-	0	+	(4)
RANDOM MINIMUM [	••	or random ck	oning]		-0-0		8	+ 0.00 	0	•	
RANDOM MINIMUM [	-() -() -() -()	for random ck	oning]		0 0	•	2 (7.00		0 0		(9) 2005
RANDOM MINIMUM [	-() -() -() -()	or random ck + +	oning]		0 0	•	ъ 9.00	-	0 0	•	(4)
RANDOM MINIMUM [	-() -() -() -()	for random ck	oning]		0 0	•	2 (7.00		0 0		(9) 2005

This will make the number of clones input along x, y and z axis, offset by the values input in the TRANSLATION box of the previous clone.

Rotation and Scale changes are constant.



Screen shot of Random selected:

2011 (2011) (2011) (2012) (201	and the second second										
SELECT TYPE OF	CLONING -										
O In Line				O Matrix				Rando	m		
UMBER OF CLONE	S [Set the n	umber of clon		-							
- 0		+	1					- ()	N/A	+	ŝ
RANSLATION [Not	used for rai	ndom cloning]									
	-0	+	Q.		0		3	-		•	<u>(</u> 3)
OTATION [Not used	d for randon	n cloning]									
-	-0	• •	0.60		0	< <b>+</b>	0.00		0	• +	0.00
SCALE (*Zero value		and the second	or an	6 C		80 - 67					<b>2</b> .5
- 0	+	Scale				yScale		+ 0.00	- ()	zSca	
- 0	+	0.00	0		• 0.00	- 0		+ 0.00	- 0	zSca	0.00
	Farthest lef	0.00 • t/down/back p	- () placement,	and rotate left	• 0.00 /forward/circle	- () right]				zSca	
	Farthest lef	0.00 • t/down/back p	- () placement,	and rotate left	• 0.00	- 0			zTran		
•	Farthest lef	0.00 · t/down/back p	- ()y placement, y 0	and rotate left Tran	• 0.00 /forward/circle	- () right] 	0		zTran	•	0.00
- () RANDOM MINIMUM ( 	Farthest lef	0.00 · t/down/back p	- ()y placement, y 0	and rotate left Tran	• 0.00 /forward/circle	- () right] 	0		zTran	•	0.00
•	Farthest lef	0.00 +	- ()	and rotate left Tran Rotation -	+ 0.00 /forward/circle 	- () right] +	0		zTran	•	0.00
RANDOM MINIMUM [	Farthest lef	0.00 +	- ()	and rotate left Tran Rotation  nt, and rotate i	+ 0.00 /forward/circle 	- () right] + +	0		zTran	•	0.00
RANDOM MINIMUM [	Farthest lef	0.00 +	- ()	and rotate left Tran Rotation - - - - - - - - - - - - - - - -	+ 0.00 /forward/circle 	- () right] + + + + + + + + + + + + + + + + + + +	0		2Tran0 	•	0.00
RANDOM MINIMUM [	Farthest lef	0.00 + t/down/back p + ght/up/forwar	- ()	and rotate left Tran Rotation - - - - - - - - - - - - - - - -	+ 0.00 /forward/circle 	- () right] + + + + + + + + + + + + + + + + + + +	0.00		2Tran0 	+ -+ -+	0.00
RANDOM MINIMUM [	Farthest left	0.00 + t/down/back p + ght/up/forwar	- ()	and rotate left Tran Rotation nt, and rotate r Tran Rotation	+ 0.00 /forward/circle 	- () right] + !eft] +	0.00			+ -+ -+	0.00
RANDOM MINIMUM [	Farthest left	0.00 t/down/back p + ght/up/forwar	- ()	and rotate left Tran Rotation nt, and rotate r Tran Rotation	+ 0.00 /forward/circle 	- () right] + !eft] +	0.00			+ + +	0.00

This will randomly place the cloned object within the minimum and maximum range input, with no consideration of where the parent object or previous clone is.

Each random value is independent of any other value.





Thank you again for purchasing **Send In The Clones**.

DraagonStorm