



Adobe Illustrator Variable Data – The EASY way!

Posted on April 26, 2015

Illustrator Variable Data - The EASY way!



SUPER-DUPER Update: I've completed a 2.5 hour course on how to use the NEW Variable Importer 8.0 script, and it's available on Lynda.com! Click to check out [more info on the Adobe Illustrator Variable Data Lynda.com course!](#)

And now back to our in-progress blog post...

Previously, I wrote up a [tutorial for an easier method to use Adobe Illustrator's Variable Data](#) feature. By easier, I meant easier than the procedure outlined in the official Adobe documentation.

This time, I'm back with an **even easier** method, if you can believe that. Recently, I was contacted by [Vasily Hall](#), an script/automation expert who has developed a script for Illustrator called **Variable Importer**.

I go through it fairly quickly in the video above, but read on for a bit more in-depth walkthrough

WHAT IS VARIABLE IMPORTER?

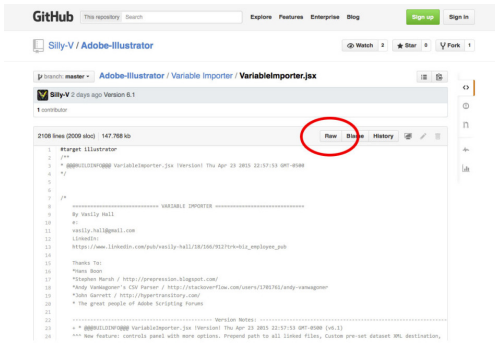
The Variable Importer script enables you to automate many of the previously annoying aspects of the usual methods of importing variable data into Illustrator.

The top issues resolved with the script are:

1. XML is generated automatically by the script, allowing you to import a plain old csv file just like InDesign.
2. Filepath to images automatically linked by the script. No more hunting down the correct syntax to input a filepath into your csv.
3. Variables can be "auto bound" to objects, sparing you yet another annoying step of manual binding
4. You can customize Dataset names at import time.

There are many more little niceties within the script, so let's go ahead and take a look.

GET THE VARIABLE IMPORTER SCRIPT



First we have to get the script installed in Illustrator. To do this, first go to [the VariableImporter script home on GitHub](#).

Once you get there, click on "VariableImporter.jsx", then click the "Raw" button (shown in screenshot above). This will bring up a page that contains the raw code of the script. From here just "Save as..." from your browser to download (*don't just right-click the "VariableImporter.jsx" and save, because that way will introduce errors into the script – make sure to only save the "raw" code*).

You'll see another script there, too called "RenameArtFromText.jsx". If you're interested in doing [multiple data sets up on a sheet](#) then grab this one, as well.

Now place the file in your Illustrator Scripts folder and restart Illustrator (or choose to load it each time using the Other Script command).

On my Mac, the location of the Scripts folder is:

/Applications/Adobe Illustrator CS6/Presets/en_US/Scripts (yours may also be inside a "en_GB" folder or other if you have a different language installed).

On the Windows installs I had access to the path was

Program Files\Adobe\Adobe Illustrator CS6\Presets\Scripts

You can see I'm using CS6 here, but I've tested it in CS5, CC, and CC2014. Plus, I have it on good authority that it works in CS4, too.

Now, once you've restarted Illustrator you'll find the script listed under File > Scripts.

So now we'll need to prepare our data source, which is our csv file, in order to have some variables to import.

SET UP YOUR DATA SOURCE CSV.

A screenshot of a spreadsheet titled 'Super Cards.ods'. The spreadsheet has six columns: A (name), B (number), C (team name), D (publisher), E (@image), and F. The data is as follows:

A	B	C	D	E	F
1	names	number	team_name	publisher	@image
2	1	Justice League	DC Comics	batman.tif	
3	2	Justice League	DC Comics	batman.tif	
4	3	Justice League	DC Comics	black-canary.tif	
5	4	Justice League	DC Comics	black-lightning.tif	
6	5	Avengers	Marvel Comics	black-panther.tif	
7	6	Avengers	Marvel Comics	black-widow.tif	
8	7	Justice League	DC Comics	blue-beetle.tif	
9	8	Justice League	DC Comics	booster-gold.tif	
10	9	Avengers	Marvel Comics	captain-america.tif	
11	10	Justice League	DC Comics	captain-atom.tif	
12	11	Avengers	Marvel Comics	captain-marvel.tif	
13	12	Justice League	DC Comics	cyclops.tif	
14	13	X-Men	Marvel Comics	cyclops.tif	
15	14	Justice League	DC Comics	flash.tif	
16	15	Justice League	DC Comics	flash.tif	
17	16	Justice League	DC Comics	green-arrow.tif	
18	17	Justice League	DC Comics	green-lantern.tif	
19	18	Avengers	Marvel Comics	hawkeye.tif	
20	19	X-Men	Marvel Comics	iceman.tif	
21	20	Avengers	Marvel Comics	iron-man.tif	
22	21	Avengers	Marvel Comics	luke-cage.tif	
23	22	Avengers	Marvel Comics	nova.tif	
24	23	Avengers	Marvel Comics	spider-man.tif	
25	24	Justice League	DC Comics	steel.tif	
26	25	X-Men	Marvel Comics	storm.tif	
27	26	Teen Titans	DC Comics	superboy.tif	
28	27	Justice League	DC Comics	superman.tif	
29	28	Avengers	Marvel Comics	thor.tif	
30	29	X-Men	Marvel Comics	wolverine.tif	
31	30	Justice League	DC Comics	wonder-woman.tif	
32					
33					

We'll revisit the "Super Cards" I did in my last tutorial so we can examine the new process with them.

Checking the screenshot of my csv file you can see it looks pretty average. There's no special tricks needed, and you can use any software that can generate a csv file. I usually use NeoOffice, a more Mac-like version of the OpenOffice software.

A couple of quick observations about the file

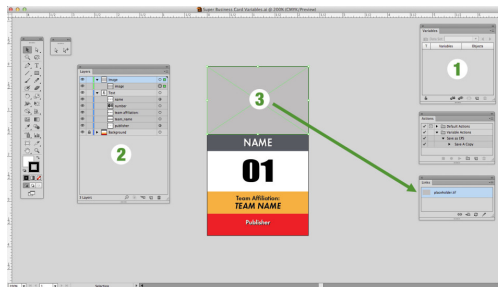
- Note that you don't have to have the first column named "Dataset" anymore. We'll create the Dataset names at import time now, so we don't need a special column to hold those names.

- Note underscore the “team_name” column heading. This isn’t special to the Variable Importer script, but just a reminder that you can’t have any spaces or special characters in heading names. Making your heading names here isn’t strictly necessary (you can also create them via the script itself), but it’s just the way I usually do things.
- Note the image column has the “@” symbol in the front of it. This is similar to InDesign, and it’s the way that the Variable Importer script is alerted that this column contains images.
- Note the image column has no file path information. Again, we’re freed from having to go find this information and place it into our csv, because the script is going to input this info for us at import time.

So basically there’s really not much to do here but fill/paste in your information, or clean up the column headers if the csv is provided to you by someone else.

Then it’s time to get into our Illustrator file.

SET UP YOUR ILLUSTRATOR FILE.



Checking the above screenshot you can see my Super Cards file is back in action.

Everything that I want to be a variable is it’s own separate object. Some things to note here are:

1. You can see in the Variables palette that there are no variables set up yet. Adobe’s official docs have you create these up first. This way we can avoid that.

2. The objects in the layers palette are named to match the header column of my csv. This will allow the auto-binding feature of the script to work. To change the names, just double-click on the object’s name in the Layers palette and then rename it to match your csv header.

You don’t *have* to do this, you can always bind the variables after import as shown in my last tutorial, but for this one we’ll use the auto-binding just because I think it’s cool :)

3. The image at the top is just a placeholder image that is **linked** and not embedded into the file.

If your image is already embedded you’ll need to select it, click the “Relink” button on the Links palette, then find your image again, this time making sure that the “Link” checkbox is checked.

If your image doesn’t have that “x” through it when you select it, it’s embedded and not linked.

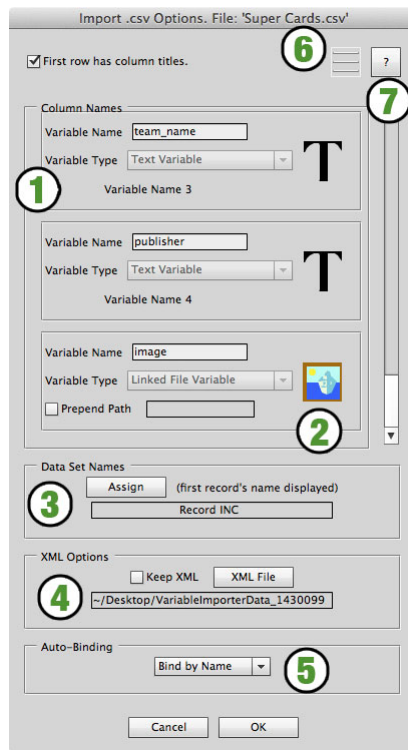
4. Special CS5 note for images: I mentioned last time I had trouble getting images to import using CS5. I’ve had *some* success overcoming this with this script. My home version of CS5 still doesn’t work, but the CS5 on a few machines I work with on locations work fine. I’m not sure what the issue is with that, but if the error pops up at import time you may have to switch to a different version.

You probably already have your desired file set up, but just make sure to give it the old once-over before you begin.

RUN THE VARIABLE IMPORTER SCRIPT

Ok, time to make the magic happen.

First, go to File > Scripts > VariableImporter then choose your csv file in the window that appears. Once you choose your csv file, you’ll see the below window:



This is the *Import Options* window, and you'll see you have quite a few options here (see screenshot).

1. The top part of the box is a scrolling frame with all column names listed. Just check through here and make sure all your columns are showing up as intended. If you have a space or special character in your column heading in your csv file then the variable won't be listed correctly here, and you'll need to go back and fix it.

2. Note that image variables have a colorful, different icon and a **Prepend Path** checkbox underneath. Checking this box will pop open a dialogue box allowing you to choose the folder where your images are located. It will then prepend that path into the XML file it generates. A huge amount of hassle is solved with this feature. After you choose the folder you'll see the path filled into the box.

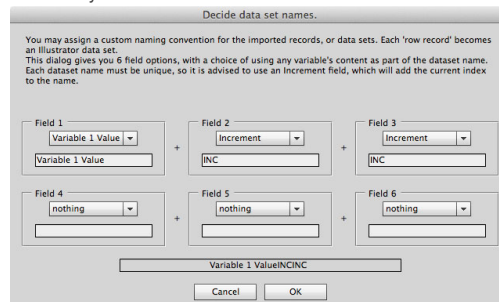
3. The next box is the area where you control the Data Set names and name preview. As I mentioned earlier, this feature allowed us to skip using the first column of our csv for the Dataset name as I showed in my last tutorial. Clicking on the "Assign" button will allow you to customize the name of the Datasets using six pulldown menus.

One thing to always remember is that the name of each Dataset **must be unique**. So if you're doing business cards and you have two Datasets based on the last name of "Thomas", then the import will fail because those two names are the same.

That's why you'll notice in the Assign Dataset names box that the "Increment" feature is on by default in Field 3. This is enough to spare you from that error, but you can customize further.

If you'd like to use one of the variables in the Dataset name you can do that, too. Just remember to keep the increment field on and it should all work fine.

At the bottom of the window there should be a preview of what your Dataset names will look like. When it all looks good just click the "OK" button. Honestly I skip this most of time in my own work, but if you need those names to be precise, then the script makes it easy.



4. Another cool feature of this script is the ability to keep the XML file that it generates. Depending on your workflow, it could be prudent to save this file and keep it as a backup just in case. Again, this isn't something I use, but I'm betting there will be those who can really make use of this, even if it's just as an example of what a properly formatted Illustrator XML file *should* look like.

5. At the bottom of the Import Options window is the “auto-binding” feature I mentioned earlier. By default it’s set on “No Auto-Binding”, which means you’ll have to select your text or image object, then select the variable in the Variables palette, and then click the “Make Text Dynamic” or “Make Linked File Dynamic” button at the bottom of the palette.

The other options are:

Bind by Name: With this option, the variable will auto bind if the variable name (column header in your csv) matches the name of your object (which you can find and change in the Layers palette).

Bind by Note: Choosing this option auto-binds the variables of the objects have a note (set in the “Attributes” palette) that matches the name of the column header in your csv.

Bind by Tag: As far as I know, tags aren’t available through the normal Illustrator GUI, but can only be set via script anyway, so most of us can safely ignore this option.

6. If you click the “Hamburger” menu at the top right of the Import Options box (to the left of the Question Mark help button) you’ll gain access to a few more settings you can change. Most notable is the Preset Options section. Here you can save and load your settings, which is a huge time saver if you running specific jobs over and over. Or even if you have similar jobs where the same settings will mostly apply.

Another section of vital importance is the “Prepend Path Options” area. This allows you to choose your folder and it will add this path to ALL image variables in your csv. This is huge because otherwise you’d have to select the folder for *each image variable* in your csv. For 10 images it’s an annoyance, but if you have 20, 30, 50 images it becomes a problem.

Other stuff in this window include the ability to predetermine where you want the XML file to be saved, and checking the top button allows you to pop up a warning if the increment field in the Dataset naming window is removed. Vasily added this at my request because I worked with a couple of people who removed that field and ended up causing their import to fail.

Now, if that field is removed, seeing that quick error will hopefully allow them to fix that before their import is potentially ruined.

To get out of this area click the Hamburger menu button again and it will take you back to the main Import Options window.

7. Finally, the “Question Mark” button at the top right has some quick instructions on how to use the script. This is good for giving someone a quick rundown if you were to pass the script on to them.

So once you’ve adjusted all these options to your liking, you just click the “OK” button at the bottom and the import will proceed. If all went well, you’ll get a dialogue pop-up declaring how many records and variable names have been imported.



CHECK YOUR RECORDS



After this, it will probably look like nothing has changed. This is because Illustrator doesn’t actually select the first record for you, so you have to choose your first record from the Data set dropdown in the Variables palette. Once you choose that first one, the correct information should now be showing in your file. Then you can use the arrows at the top of the Variables palette to quickly cycle through all your sets and make sure they are ok.

At this point all you have to do is export each Data set as it’s own file.

EXPORT ALL DATASETS VIA ACTION

Now that you have the variables loaded in and working, you're going to have to output these to separate files at some point. This is accomplished by using an export action. Basically you have to save the file first, record the saving of that file, then play it back so Illustrator can save all the files itself without user input.

If it's not already open, select "Window > Actions" from the top menu. Select "New Action..." from the right flyout menu, or click the New Action button on the bottom of the Actions palette.

In the window that appears, you can name the action something like "Save EPS" or "Save AI" or anything that helps you remember what it does.

Once you click the "Record" button in that window, the action will begin recording. From this point, do a "Save As Copy...", and then choose your desired format and location. Don't worry about what you name the file at this point.

It's important to use "Save As Copy..." because if you simply use "Save As..." then for some reason Illustrator will not save the file extension. This may be fine for some, but the Rips I work with won't process an image without an extension, so I gotta have it.

Another important note about "Save As Copy...". In my experience I found you **should not change the name of the file** when you save it. If you do, then the extension will again be lost, and you'll have to add it back using Bridge or some other method.

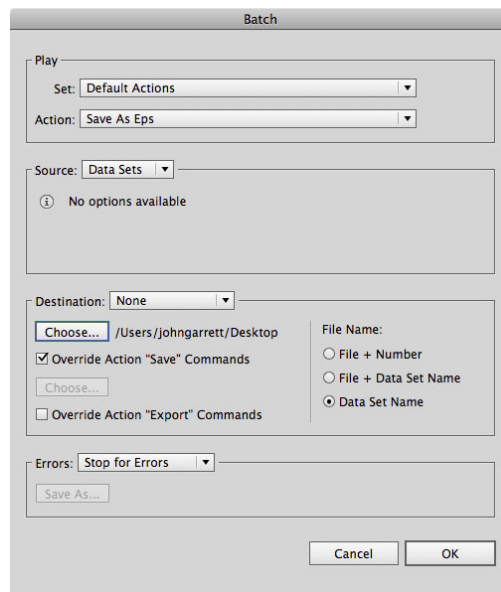
When the file is saved, make sure to click the "Stop" button at the bottom of the Actions palette. You now have a complete action you can use to export all of the datasets. You can delete this file you saved, since you won't need it anymore.

Use export action in Batch process to export all files.

From here we'll use the "batch" feature to run the action we just made on every one of the datasets.

To start, open the flyout menu on the Actions palette and choose "Batch..." from all the way at the bottom.

In the window that pops up, make sure your action is chosen in the "Action" dropdown. Set the "Source" to "Data Sets", and choose the folder you saved to in the "Destination" area.



On the right of the "Destination" area is where I usually set my File Name to use "Data Set name". You don't have to use that option, it's just the one I prefer.

As soon as you click the OK button, Illustrator will begin switching through all the data sets and saving them out according to your action. Depending on the speed of your computer and the number of records this might take several minutes. It will definitely be faster than doing it by hand, though.

At this point you're essentially finished, you have all your files exported and ready to go. Obviously you should take a quick look at all the files using bridge and make sure no errors jump out at you, but basically it should be all done.

If all you needed were separate files, then mission accomplished. But, if you need to get those files laid out on one big sheet for printing, then you'd have to place them all back into Illustrator and group them by hand. In an upcoming tutorial I'll talk about placing multiple Data sets up on a page.

Congrats, we just imported variable data – the **easy** way!

LINK ROUNDUP

Here's a few quick links for convenience:

- [The VariableImporter script home on GitHub](#)

This is the official home for the script, so always go here if you need to update it or grab it for another install of Illustrator.

- [In-Depth tutorial on VariableImport by Stephen Marsh of Prepression](#)

I know we just got done with a rather involved tutorial here, but I **urge** you to head over to Prepression and read Stephen's detailed write-up of the script.

Where my tutorial walks you through a project, Stephen outlines pretty much every little feature of the script, so make sure you check that out.

- [Vasily Hall on LinkedIn](#)

Vasily is the author of this script, and he's actually available for more script work. So if you need some automation in your workflow or know someone who needs some scripting work you should definitely give him a shout.

THE WRAP UP

So that's all for this tutorial. I hope this provides a much easier method for working variable data in Illustrator.

Stay tuned for another tutorial I'm working on detailing how to add multiple Data sets to a sheet in a semi-painless manner.

(EDIT: You can see this tutorial here:

https://www.hypertransitory.com/blog/2016/04/11/adobe_illustrator_variable_data_multiple_datasets/

For a final announcement, earlier I mentioned that version 8 of the Variable Importer is already out. Well, I've actually created a video course that walks through all the bells and whistles of using that new script for variable data, and ~~it will be available on Lynda.com very soon~~ **IT'S HERE!** So those of you interested in checking it out can get [more info on the Adobe Illustrator Variable Data Lynda.com course](#) at that link.

With that said, I'm outta here. Leave me a comment or otherwise contact me if you have any questions or issues.



Posted in [Graphic Design Tutorials](#), [Hypertransitory](#), [Illustrator Tutorials](#), [Tutorials](#)
tagged [adobe illustrator](#), [illustrator](#), [illustrator tutorials](#), [variable data](#)


[Previous » John Garrett Interviewed on Renderstreet](#)

[Next » Adobe Illustrator Variable Data – Multiple Datasets With VariableImporter Script](#)


51 THOUGHTS ON “ADOBE ILLUSTRATOR VARIABLE DATA – THE EASY WAY!”

1.  **Jason** says:
[May 22, 2015 at 12:30 PM](#)


Vasily does it again

2.  **Maurice Mitchell** says:
[May 22, 2015 at 3:35 PM](#)

What a cool feature of Illustrator and it looks like a much easier way to automate a tedious task.

3.  **João Faraco** says:
[August 25, 2015 at 7:02 PM](#)

This is beautiful ! Much easier than the other method that involved having to mess with the xml code. Great job on the written and video tutorials, and thank you Vasily for the script !

-  **JG** says:
[August 26, 2015 at 11:13 PM](#)

Thanks, João!

I think Vasily has started a revolution when it comes to handling variable data in Illustrator lol.

All the best, man!

4.  **Carrie Dils** says:
[September 5, 2015 at 1:57 PM](#)