JOHNNY COMICS

Johnny Comics typeface is a family of professional comic book fonts (Regular, Italic, and Bold Italic), influenced by the lettering style of some of the best letterers of the Silver and Bronze Ages of Comics, such as Gaspar Saladino and John Costanza. The Regular font employs 7800 alternates in an OpenType feature called "randomize" (rand), which means that basically a 40,000-word text can be typed without repeating a character. This should be quite sufficient for rendering a 120-page graphic novel without repeating a character. This amounts to six 20-page comic books. The Italic contains 50% of this number, and Bold Italic, 20%. Every time a character is typed, in those programs which support the rand feature, such as Serif's Affinity suite of programs (Publisher 2, Designer 2, or Photo 2), a different alternate version of the letter appears onscreen, giving a handwritten appearance. (Affinity programs can be used compatibly in workflow with other graphics and desktop publishing programs, such as Adobe software.)

Being a professional typeface means that Johnny Comics is postscript ready. (Some day, perhaps, comic book text will be printed on postscript printers like other magazines. In comic books the text is normally turned into vector graphics for printing or rasterized. In any event, the fonts are also good for publishing venues or printing modes other than comic books.) Each character has been designed as economically as possible, utilizing few points. However, in order to accomodate postcript printers, more plotting information has been added. This triples the file size. Most graphics and desktop publishing computers these days should have no problem, but the fonts are larger than most.

Postscript Printing Service Provider

Embedding in ebooks or other digital media is prohibited. Embedding the fonts in a file for professional printers is disabled and forbidden by all licensing agreements except the proviso for postscript printing services.

Normally, comics producers will use the typeface according to industry standard protocols for comics production. This entails typesetting in a vector graphics design or desktop publishing program and creating vector outlines before printing, in which case the text will be part of the image. From there the comic producer will need to negotiate with the printer, whether the text can remain vectors or whether it needs to be rasterized, which is normally the case. The industry standard is either an offset process or digital (for Print on Demand, usually).

In most cases the comics producer will want to determine beforehand if he/she wants to use postscript printing services. If a comics producer wants to employ postscript printing, he will need to contact me, and I will send an embeddable version of the typeface to the postscript printing service provider. However, the proscript printing services company will need to pay a nominal fee so that they will be legally liable for ensuring that the font is not pirated or used in an unauthorized manner. The financial agreement between the printer and the comics producer is their own business, but a simple way to handle it is simply increase the printer's fee to compensate for the postscript-printing fee which Johnny Type charges. The fee is nominal: it is paid simply to make the authorization and responsibilities of the printer legal.

If the comics producer determines to use postscript printing services *after* the document(s) has been typset, this should not be a problem, with the proviso that the text has not been vectorized or rasterized. Naturally, in such a case, vectorized or rasterized text should be printed using traditional methods. Or the project must be typeset again: the typesetting must be re-done.

Serif's Affinity Suite

Originally, upon deciding to do an OpenType comic book font, I thought of using the *rand* (Randomize) feature. But then I realized that scarcely any software supported the *rand* feature. Adobe Illustrator and Indesign did not. Nor does Quark Express. Most typographers, in an attempt to make fonts practical for the public and publishing world, opt for the *calt* (Contextual Alternates) feature instead. But for a font that is alternate intensive, *calt* is the stupidest thing in the world (except to accommodate the design software). It is ridicuously labor intensive. This is the reason that all contextual alternate fonts only have a limited number of alternates, three or four of each letter.

When I realized that the *calt* (Contextual Alternates) feature demanded writing one or more lines of code for every single glyph substitution, and doing so for 7000 alternates would mean 10,000 lines of code, I rebelled. Why could you not simply number the alternates and have the user's machine cycle through them. That is exactly what the *rand* feature does.

Finally, I concluded that Adobe and other software developers do not want people using the *rand* feature. But one can work around these ridiculous limitations.

Well, it turns out there is a desktop publishing program that supports the *rand* feature, a suite of three programs, in fact, made by Serif, the Affinity 2 programs: Affinity Photo, Designer, and Publisher. The *rand* feature works flawlessly in them, placing an alternate form each time a letter is typed. All OpenType features employed in the fonts work in them. The programs are comparable with Adobe's suite of programs. They may not have as great a level of functionality, but Affinity is getting there. At any rate, they can be used in workflow with the Adobe design programs. Illustrator and Photoshop files can be opened in them as well as jpeg, tiffs, eps, svg, pdfs, and other file formats. But the best part is that the Affinity programs only costs a fraction of the price of Adobe programs. They are also standalone programs, which is an immense plus for those of us who don't care for the cloud (extortion, eternal payment) method. Not to mention the eternal probing of our computers.

With regard to the Affinity 2 software; comic books should probably be laid out in Affinity Publisher 2, though, rather than Designer. Publisher allows typographer’s quotes (which are also called *smart quotes* or *curly quotes*). Most comics producers will probably want to use typographer’s quotes. Vector balloons can be drawn in Publisher. It has certain text effects as well. And what cannot be created in it can easily be imported from any other Affinity program and most other graphic design software as well.

I begin by mentioning Serif’s Affinity suite of programs because the prospective purchaser of the Johnny Comics fonts needs to understand that, although the font will work in any program that supports OpenType fonts, the cycling alternates feature will only work in programs supporting the *rand* feature, and Serif’s Affinity 2 suite supports it.

Working Procedures

Many serious comic book producers use the Adobe suite of programs (Photoshop, Illustrator, and Indesign), all very good programs. Of course, they are unduly expensive and lock you into a monthly fee. But if a professional comic book producer, having mastered the Adobe software, would like to continue using them in comic production, no problem. Affinity Designer and Publisher will open Photoshop and Illustrator files, with varying degrees of success. Photoshop files import pretty well, whereas Illustrator has more caveots. One must save as PDF, retaining layers and transparencies. If one chooses to use the Adobe programs in the early stages of development, one can letter comics in Affinity in several ways.

Because text and balloon placement is an element of comic book design, one may want to go ahead and typeset the text and make balloons in Illustrator or Photoshop. You can use the Johnny Comics fonts to do this. Text in the Adobe programs will not display alternates but only the main upper and lowercase letters. Since spacing and kerning is comparable in alternates of each letter, any variation should be negligible.

The final art work, aside from text and balloons, should be on one layer, probably a background. It will need to be more or less finished. Better yet, save the text alone without the artwork to a separate file. Save .ai files as PDF Compatable. Open in Affinity and tick “Favor Editable Text.” The *rand* feature will be applied, displaying alternates. Once the *rand* feature has been automatically applied, you can save text as vector outlines or rasterize and bring back into Illustrator, Photoshop, or Indesign, using the former, original text layer as a template for registration.

If one intends to use postscript printing services, in which case text does not need to be vectored or rasterized, one will probably want to limit the number of pages in each document. Each issue or book will be comprised on several documents, which the printer can easily program to print in sequence. In order for the sequence of alternates to be maintained, one document to another, you will need to follow the procedure outlined below. In any event, this process will need to be employed even on multiple documents that one vectorizes or rasterizes.

This said, let us begin:

Overview of Typeface

The style of the characters of the Johnny Comics font have been influenced by the work of various letterers of the Silver and Bronze Ages of Comics, letterers such as Ed Hamilton, house letterer for ACG (American Comics Group) in the 50s;* Stan Starkman of DCs *Challengers of the Unknown* and *Doom Patrol*; Gaspar Saladino, Joe Kubert, Jim Aparo, Sam Rosen, Artie Simek, John Costanza, as well as Frank Thorne, Ray Holloway, Jim Novak, Sonny Trinidad and many others.

Johnny Comics can be used professionally, but a licensing agreement is required when it is used extensively. For specifics

* While not as regular or precise, Hamilton had better letter widths (they were wider) and spacing, a style which was adopted by Rosen, Costanza, and Saladino in their maturity.

see the licensing agreement.

Kerning Note

Kerning is the optical spacing between letters.

Because of the vast number of kerning pairs required for fonts containing thousands of alternates, some economization has been necessary. With few exceptions, the fonts are designed to be typed from the lowercase letters. The great number of alternates are generally accessed by way of the lowercase letters. Therefore, most of the kerning pairs are with respect to letter sequences typed in the lowercase mode. The only substantial exception to this are those special (alternate-design) characters which are typed from the uppercase: such as the capital "I," the "Back-Slanted B" in the Regular font, and the "Toth Y" in all fonts of the Johnny Comics typeface. (Alternate-design characters are discussed below.) In most cases, typing (even inadvertently) an uppercase character may result in letters which are not properly kerned: the spacing may not be visually correct.

Rand Feature

The "*rand*" feature, which is not widely supported, is not really random at all. It displays alternates sequentially as they are laid out in the glyph pallet, giving the impression of randomness, very much like the OpenType feature called "contextual alternates." The *rand* feature uses the respective computer upon which it is running to determine the next character in the sequence to display by ascertaining the name (number) of the last one displayed. Naturally this process only works within single files, since the feature looks back in that file for the last instance of a specific character to ascertain its name (number) in order to continue, displaying the next glyph in the sequence.

To maximize the capability of the font to display numerous unique characters, it is best to ensure that the sequence of programmed alternates is unbroken. If a session of lettering includes many comic book pages in one file, the characters will display in sequence. However, the longer the run of sequential letters the more RAM is necessary because every time the file is opened the computer recalculates which glyphs to display. That is, until they are turned into outlines; which also requires a lot of computer power. These days most computers are capable of handling both, but the best working procedure is to keep files as small as possible. Since the alternates of each glyph (character) are numbered and display in sequence, determining which number to display by ascertaining the number of the last one; when a new file is started the font will begin displaying the alternate glyphs at the beginning of the sequence again, with the respective “number 1,” then “number 2,” and so on. If one wants the letter sequences to continue from one file to the next so that characters are not repeated in the next file, it will be necessary to include the last-used instance of each character in the new file. However, this cannot be done if you turn text into outlines. The gyph’s name information will be lost. Vectorize the letters *after* you have copied the last-used instances, a topic which we begin dealing with now.

When you are satisfied with the lettering in a file you will probably want turn the text into outlines for printing.* However, the Opentype feature cannot read the names of former glyphs which have been turned into outlines. Thus it cannot continue the letter sequences in another file. One will need to manually establish the sequence to the next file before turning a former file into outlines. Here’s how:

Maintaining Sequence

It is best to restrict lettering files to a few comic book pages each. Perhaps, even, each page as a separate document. At the conclusion of lettering a certain file, before turning text into outlines or rasterizing them, copy the list of characters below

* For electronic comics, one must generate an image file rather than publish with text floating on an image. PDFs allow for embedded text, for example; but hackers can extract the font from such a file. Licensing restrictions for Johnny Comics font forbids embedding for this very reason.

in this documentation and paste them into that same file. The appropriate alternates will be assigned in sequence. Then cut and paste that list of characters, which now employ the “Johnny Comics” font, into the new file. This will assure that the sequence of alternates is contiguous with the former file.

To maintain the glyph sequence:

1. After you have finished lettering a page (or file containing multiple pages), copy the characters below from this “Johnny Comics” documentation PDF.

! ’ , - . 0 1 2 3 4 5 6 7 8 9 ? @ B I Y [ ] a b c d e f g h i j k l m n
o p q r s t u v w x y z {} “ ” —

2. Paste them into the file you are closing out. If “Johnny Comics” font is not assigned automatically to the pasted text, you will need to define the font.

3. Re-copy these letters (list of most-used glyphs) which are now in sequence of the Johnny Comics alternates.

4. Then paste them into the new file of comic book page(s).

5. Delete the ”most-used” characters from the original comic book pages file. Turn the lettering lettering in that file to outlines or close document.

6. If the lettering method employs Regular, Italic, and Bold Italic fonts on each page, or even within the same publication, this procedure will need to be repeated for each font. If you do not use Bold Italic, for example, on one page but use it on subsequent pages in the same issue, care must be taken to maintain the sequence. This can be done by planning ahead of time what needs to be done to maintain letter sequence. The last page upon which Bold Italic (for example) is used, follow the procedure mentioned above and paste the letters onto the first subsequent page in which that font style will be used.

Now you are ready to begin lettering the new comic book file. Move the series of most-used gylphs out of the way so you can letter your comic book images.

Lettering New Comics Pages

After having completed the lettering in that document (a limited number of pages), delete the disposable letters, which you initially pasted into the file. The alternates are then in sequence with the former file, thus employing all the characters in the font without repeating them unduly.

Because there are over 7000 alternates it is doubtful that all alternates will be used even in one book. (I calculate that a 120-page graphic novel, or 40,000 words of text, should not duplicate a letter.) The same process can be used on the next book to continue displaying unique, original letters. Automatically, when one comes to the end of the alternates of any character, the cycling will begin again at the start.

Expanding Font Style Characteristics

Johnny Comics Regular font is somewhat conservative. The variation of each letter is not great, thus giving the impression of a more consistently masterful or smoother style, which actually may seem artificial because such regularity is very difficult for a letterer to maintain. This in itself may seem unnatural. But there are many ways to add variability to Johnny Comics fonts, such as Character Styles and various methods of generating random baseline shift.

Character Styles

A comic producer may like the general design of the Johnny Comics letters, for example, but wish that they were more oblique and cattier. Most desktop publishing programs, such as Affinity Publisher, can Skew (or Shear) any text. Johnny font can be Skewed / Sheared horizontally and/or vertically a few degrees to give a sharper diagonal angle on the oblique horizontals and thus achieve a more italicized appearance to the "Regular" font, a cattier, jazzier style. One can alter the oblique angle of the Italic font similarly to suit one's taste. This can be done on selected text directly, or it can be applied to either selected text or all the text in a document by way of Styles. Most design and desktop publishing software have a "Styles" capability, in which certain presets can be saved and applied to selected text quickly and efficiently through "Paragraph Style" and/or "Character Style." Once one begins to type in a Style, the program can also be configured to remain in that Style every time one types. Some programs do this by default. Scripts can be also downloaded or written to apply these changes randomly throughout selected text.

THIS IS AN EXAMPLE OF TEXT SHEARED/SKEWED
7 (DEGREES). ITALIC IS USUALLY SHEARED
ABOUT 16 DEGREES HORIZONTALLY.

I cannot over-emphasize the ways in which the versatility of desktop publishing programs can expand the look of Johnny Comics fonts. Johnny Comics was designed with somewhat wide letters. (This was the tendency of John Costanza as he ripened at Marvel under the influence of Artie Simek. This was Gaspar Saladino's more mature tendency also.) Perhaps they are too wide for your tastes. No problem. In Serif's Affinity Publisher 2 you can reduce the letter width to 94%, and all lettering done in that session will reflect this change until you specify differently. Of course, upon closing the program, it will no longer remember this setting, unless you save it as a style. However, it is simple enough to set the width every time. But sometimes we forget, in which case a style is nice. There is threshold of alteration at which the distortion of the stroke becomes noticeable, however. One must experiment to determine how much alteration the fonts will visually allow before it begins to look unprofessional.

Random Baseline Shift

I opted to make the letters of Johnny Comics closely adhere to the baseline. This tends to give a highly professional look, but it may not seem as naturally handwritten as some creators desire. The creator will want to research various methods for automatically applying a random baseline shift within certain programmable parameters. There are various methods for doing this. One is scripting.

A script called *Prime Pussify* (javascript) has a programmable random baseline shift capability for text, which works nicely for Photoshop. But, alas, Adobe products do not support the "rand" feature. At the present time, Affinity software does not support scripting, but the programmers at Serif (makers of the Affinity suite) are feverishly working to add scripting support. Because Affinity 2 programs are written in javascript rather than Python or C++, scripting should be much easier when they add the capability. There are other scripts and programs which will do the same thing, however.

In lieu of scripting for text effects, perhaps the best way to randomly displaced text is to turn the text into vector outlines, in which case each letter (character) will then be a shape. Remember to "Ungroup" if you "Create Outlines" on more than one letter, a necessity of the random shifting of multiple letters. You can use certain programs, plugins, scripts, or capabilities to incrementally shift each letter shape within certain parameters automatically. For example:

Astute Graphics has a suite of Illustrator plugins for a subscription of $150.00 a year, one of which is Randomino. It will randomly shift shapes from the baseline at various programmable parameters. A few free scripts for Illustrator, which can be downloaded on the Internet, will do it, one called *Randomus*. A free Photoshop script called *Randomize* will also. You can find several other free shape displacement scripts on the web. An opensource vector graphics program called *Inkscape* has a feature called "Align and Distribute" which will do it. But this is based on Illustrator's built-in capabilities, which can also do it. Frankly, Illustrator is probably the preferred method.

Here's how in Illustrator. Each line of vectorized text will have to be selected independently. From the main menu select Object / Transform / Transform Each. A panel pops up. Select the checkbox “Random” on the bottom right-hand corner. Then, in the lower section of the panel, Move / manipulate the vertical coordinates, which will distribute above and below the baseline. (-0p2 creates a good random displacement.) This sets the uppermost bounds of the displacement, the various randomized settings applied to each shape being fractional percentages of this. Click “Okay.” Review results of the random shift displacement and manipulate the parameters, if necessary, repeating the process until you are satisfied with the results. This process can then be applied to all text, line by line. It may seem rather time consuming, but it is not as labor intensive as it sounds. The work goes swiftly, and the results are rewarding. Thus the Johnny Comics typeface, which was already far more versatile than other comic book fonts, becomes truly expansive in its capabilities.

APPLY RANDOM BASELINE SHIFT IN ILLUSTRATOR.

NOTE: Illustrator actually seems to create variations on a wave pattern. This may not be as acceptably random as one would like.

Affinity Designer 2 has an “Alignment” capability, the permutations of which can be manipulated a little. It may provide the desired result. But it seems to be a little buggy. Perhaps the best procedure is to vectorize the text, save the file as a .PDF, and, in Illustrator, utilize its random baseline offset capability.

Random Transformations of Scale

Most of the same processes and capabilities can be applied to randomly altering the scale of character shapes (vectors) by small percentages. You may want to experiment with this. But the suggestion goes with a cautionary proviso: Any transformation of the vector shapes (letters, characters, glyphs) will alter the width of the stroke. Only a little transformation will alter it noticeably and unnaturally. The letters will cease to look as though they were executed with one nib, stylus, or pen. And while this may not concern you, it is liable to have a distracting effect upon your audience. There is a certain percentage of scale within which such transformations can be made with negligible distortion. Because the size of the letters of the Johnny Comics typeface adhere closely to the same scale, you may want to experiment. Employ such transformations conservatively.

Performance

Affinity Publisher 2 has the RAM allocation set to about 12 gigabytes. More than likely you will need to increase it. If you do not have sixteen (16) gigabytes of RAM or more, the program may have difficulty keeping up with the alternates. You can configure Publisher to allocate more RAM via the Edit menu: Edit / Settings / Performance. In any event, you will want to work economically, which will entail limiting the number of pages in each file, perhaps even saving each page as a separate file and changing text to outlines/curves at the conclusion of each file. If this is the case, you will need to become adept at saving the last-used alternates and pasting them into the new files to maintain continuity of alternates.

Using Roy Thomas' Crusader story of *Fantastic Four* issue 164 as a gauge, one can expect between 120 and 220 words per page. Three pages would be 600 words. Given the background artwork and vector balloons, this is probably as much as you will want to include in each file, at least until you have experimented to see if your system has enough processing power and RAM to handle more material. If Affinity Publisher 2 begins acting flakey it is probably a sign that the RAM is insufficient, or the program allocation of it is set too low.

Summation on Workflow

Most comics, comic books, or other graphic material in which the Johnny Comics typeface is used will probably want to employ typographer's quotes. Typographer's quotes creates the only real limitation to the Johnny Comics typeface; rather a conflict results when one wants to employ another feature. It is only an issue with regard to the implementation of random baseline shift and typographer's quotes at the same time. Here's why:

1. One will probably want to use the Affinity suite because it provides the OpenType functionality of the "rand" feature, which is one of the great advantages of Johnny Comics. But only Affinity Publisher 2 can apply typographer's quotes. Therefore, the typesetting will probably need to be done in Publisher.

2. Affinity Publisher 2, although providing typographer's quotes, does not allow the text to be randomly shifted much. One can vectorize text, however: Menu / Layer / Convert to Curves, which is usually necessary for random baseline shift. Publisher has a rudimentary "Distribute Vertically" function under Layer / Alignment, but it does not have programmable parameters. The Alignment / Align Vertically function is a little more versatile, but it does not seem to produce random results but the same pattern of distribution every time, based more or less upon a wave.

3. Typesetting in Publisher 2 can be vectorized, exported as a PDF, and re-opened in Illustrator, in which its random baseline shift capability can be utilized, which is more versatile. Files can also be exported as a Photoshop document (PSD), but my old CS 5.1 version rasterizes the curves. It rasterizes the text but maintains them on layers.

4. Some planning needs to be done before typesetting. The project needs to be diagrammed, identifying those pages which use typographer's quotes. But since everything with regard to balloon-drawing and typesetting can be done in Publisher, then, Publisher seems to be the go to. If random baseline shift is desired, export as .PDF and do it in Illustrator.

5. This is the state of affairs presently. I expect the functionality of the Affinity programs to increase, removing these limitations.

Regarding “aalt” Feature

I did not write an “aalt” feature into Johnny Comics font. The “aalt” feature allows all of the alternates of any certain character to be ready at the typographer’s fingertips, so to speak. Depending upon the design/publishing software to employ the feature, this option amounts to the alternates of a certain character displaying when the cursor hovers over that respective letter. Since all the glyphs display in the Adobe software anyway, and the option of simply cycling through variants of each character is so simple with the “rand” feature, the “aalt” feature was deemed superfluous to comic book lettering.

Special Characters in “Johnny Comics”

The character style of the Exclamation Point in the “Johnny” font is basically that used by most comic book letterers. The nib is turned flat, more or less perpendicular to the direction of the stroke, drawn backward and pulled up for diagonal Exclamation Points.

John Costanza did not make diagonal Exclamation Points. He drew then vertically. Consequently, he began the stroke at the top rather than at the bottom. But, for a right-handed person, this is as difficult as drawing a diagonal stroke, perhaps more so.

There are stylistic variations of the Exclamation Point, which most letterers use at one time or another; such as varying the angle of the stroke, which changes the thickness of the Exclamation Point and often some rotation of the stroke. Jim Aparo was pretty regular in his treatment of the single-stroke diagonal Exclamation Point. This is the style of Exclamation Point (and its cycling alternates) which are accessed in the Johnny Comics font by the keyboard’s Exclamation Mark key.

Rosen Exclamation Point in @ Key

However, there is another type of Exclamation Point, which some letterers use from time to time. In *The Brave and the Bold* 119, Aparo tried his hand successfully at this more design-intensive Exclamation Point. Sam Rosen used it regularly, more than 50% of the time. He often made a hand-drawn Exclamation Point, which resembles the baseball-bat style of Exclamation Points in serif fonts and type. Drawing it requires a looping stroke (Rosen) or two strokes altogether (Rosen and Aparo). Sometimes, in a rather heavy-handed way, Rosen even drew a short horizontal line, capping off two acute diagonal lines, which converged at the bottom: three strokes for an acute triangle. Sometimes he simply allowed the ink to blob, thus covering the distinction between two lines. Aparo's lettering was cleaner, and the separate strokes can be identified, but he knew that reduction would make the two marks visually blend and did not bother with a cap. In any event, the process was more time consuming. Rosen believed that variety and weight were necessary for the type of dynamism Marvel was promoting at the time. Aparo drew it almost exclusively for twelve issues of *The Brave and the Bold* and seems to have been influenced by Rosen. (Simek and Rosen were powerful forces in the industry: even Saladino and Costanza were influenced by them, arguably the greatest letterers of their generation.)

A great number of Rosen baseball-bat Exclamation Points can be accessed via the @ symbol key. Humorous cartoons may choose this Exclamation Point exclusively. They are also good for titling.

Brackets

Usually the comic book symbol for sighing or exasperation, which sets off a word like "Whew!" is typed using the "brace" brackets on the keyboard. I have included several. But I have also experimented with an original type of punctuation symbol, perhaps to be used in more frenetic exclamations. A few of these can be accessed by typing the regular bracket symbol.

While some letterer/type-setters might have a use for the regular brackets here or there (although I cannot remember seeing them used in comic books), I have included four alternates at the beginning of the string of alternates assigned to brackets. Sixteen of the original exlamation symbols follow. If one has no need of regular brackets but would like to use the new exclamation brackets, you will need to open the character palette from the main menu. In Adobe Illustrator this is "Type / Glyphs"; or a comparable panel in some other software. Within the Glyphs panel locate the alternates and insert the first one manually. This is the only instance in the font in which the letterer/type-setter must have recourse to the Glyphs panel. In an effort to keep the file size of the alternate-intensive font small, I have opted to not include the OpenType feature "aalt" for "Alternates," which would provide a special access pallet for each gyph/letter form, making the process of finding the brackets easier. But I do not think that it poses an undue difficulty in view of streamlining the font. Besides, one can simply cycle through the four normal brackets, deleting them as one goes, to quickly access the originally designed expressive brackets. Everything else should be easy typing, rather intuitive. It is unfortunate, but I had to prioritize. I thought it best to have a few regular brackets, and this is the workaround. This represents the only instance in which different characters can be typed from the same key.

In any event, if one uses the new exclamation symbols, you will need to go through the same process with "close" brackets, the right-hand bracket. Because the brackets have been designed in pairs so that the open and close brackets work together aesthetically, it is best if the numbers of the open (left-hand brackets) and close (right-hand) brackets correspond since they have been designed to be partners.

Em Dash

Em dashes are usually rendered with two short dashes, each being a little shorter than the regular dash. But together, they are wider than the hyphen-dash. This is a convention in comic book lettering. There was a good reason for treating the em dash in this manner. It was a "writing" convention to start with, the em dash was a "printing" convention. Letterers did not want the single wide dash, which is the em dash represented in most typography, to be mistaken for a hyphen. In other words, they did not want to confuse the reader. One great letterer, however, broke with convention a few times and used the regular wide em dash rather than the two short dashes. Gaspar Saladino did this to good effect, I think. It fit seamlessly into his smooth, curvy, catty style. However, he did not do so for long, which suggests that he also came to the conclusion that the practice could be confusing.

I personally like the regular unbroken wide em dash. So I have included some. But most typesetters/letterers will probably want to keep with tradition. Conventional double-dashed em dashes can be inserted from the glyphs/special characters panel or by using the keyboard command for inserting them, which is (hold) ALT + 0151. Affinity Designer has a method of doing so from the menu, which is quite sufficient. If, however, you opt for the unbroken em dash, these can be accessed by the caret key, "^," which is also called a "spacing circumflex," to distinguish it from the circumflex accent. The spacing circumflex is used as an editorial notation and has little application for a comic book font.

Although it is easier to simply press SHIFT + ^ to insert the broken em dash, the style of em dash that most letterers will probably use, doing so is unconventional in typing and typography. Therefore, rather than including the broken em dash there, I have included them so that they are accessed in the conventional way. The typesetter/letterer will need to get familiar with the keyboard command ALT + 0151.

Personally, I think that both can be used effectively. Broken em dashes can be used within sentences or lines of text so that there is no confusion with a hyphen. And the longer single em dashes can be used at the end of a sentence or line of text with no misunderstanding.

Mathmatical Symbols

In typography and typesetting, mathmatical symbols are usually separated from numbers with a thin space or a hair space. A regular space is too wide. I could have selected a key to insert a thin space or a hair space, but I was running out of keys as it was. Besides, the typesetter/letterer is already required to remember a few special characters and keys in the font. And mathmatical symbols are not often used in comic books. Certainly, a person lettering with Johnny Comics typeface should not be expected to insert a thin space or hair space from the glyphs/special characters panel. Doing so would be unduly laborious. One would have to search through thousands of glyphs to find it. So I opted to simply make the left and right side bearings of the mathmatical symbols wide enough to simulate a thin space or hair space. This system requires the letterer to type mathmatical symbols consecutively, without any space before or after a number. For example, when writing "2 + 2 = 4," rather than spacing between the numbers and mathematical symbols, one types them without spaces, and they should look fine. Allow me to restate: rather than include a thin space, which would require an unduly complex system, I have simply made the glyph space wide enough to accommodate the impression of a thin space, which is about 1/5th the width of a normal space. If you want to type 6 + 7, you should be able to simply type all the glyphs consecutively without a space at all. Doubtless, the letterer will be reminded of the system when he or she types an equation since the additional spaces will not look visually correct.

Back-Slanted "B"

Normally, comic book lettering is a tad oblique. That is, the vertical strokes are slanted a little to the right-hand side, italicized. This conveys a sense of dynamism. Some letterers, such as Gaspar Saladino and Jim Aparo, were more oblique than others. Sam Rosen or John Costanza were usually more conservative. That said, some letterers sometimes treat certain characters in a more overtly hand-written style, which violates the general system of obliqueness. Saladino is a good example of a natural, quickly rendered hand-written style. Although the reader may not be aware of it, the uppercase "B," often, in contravention to the general obliqueness of the font, can actually be found slanting backward. It is counter-intuitive to the general geometric system of the style, but it looks quite natural. Costanza did this periodically, particularly when the “B” was the first letter of a word. More regularly when the letter “B” was the first letter of the first word of a sentence.

The “Johnny” Regular font contains back-slanted “B”s, which can be typed from the uppercase “B” on the keyboard (Shift - B), alternates being assigned as one types. I have not included the back-slanted “B”s in the Italic or Bold Italic fonts.

Handwritten “Y”

Some letterers, particularly, it seems, those in syndication, make a form of the letter “Y” which is different than most comic book letterers. It is a curved, handwritten form, whereas comic book letterers usually make a straight-line geometric “Y.” One good example of a comic book letterer who penned a curved, handwritten-style “Y” was Alex Toth. Of course, there are varying ways to approach the task of making an alternate “Y,” different curvaceous forms. My version can be typed via the uppercase “Y” key (Shift - Y) on the keyboard.

Naturally, one will want to be consistent. If you choose to use the alternate “Y,” you will need to get in the habit of typing the uppercase “Y” every time that a “Y” appears in your text. However, there is another method that you may choose to employ. One can type all lowercase normally; then, after a typing session, run a “Find/Search and Replace (All)” function. This will replace every instance of the lowercase “Y” with the uppercase. Of course, this will only work with text which has not been turned into outlines. Furthermore, the same procedures mentioned above will need to be employed to maintain the sequence of alternates when another page is started..

There are over 300 alternates of this form of “Y,” which can be typed as an uppercase “Y.”

Toggling Typographer’s Quotes

For most purposes Smart Quotes (also called “Curly Quotes”) can be turned on in the design or publishing software from the Preferences menu. (In the Affinity suite only Publisher has this capability at the present time.) However, one may want to toggle back and forth between the different types of quotation marks. Normally, the straight quotes display. If Typographer’s Quotes are not selected in Preferences, they can be typed with a keyboard command: In Windows:

’	opening single quote	alt 0145
’	closing single quote	alt 0146
”	opening double quote	alt 0147
”	closing double quote	alt 0148

Commands for Other Special Characters

...	Ellipsis	alt 0133
--	Em Dash	alt 0151

Double-Letter Ligature Substitution

In Opentype programming, contextual ligatures are written “clig.” The only contextual ligatures in the font are “CC,” “FF,” and “LL.”

The ligatures that I have included in the font should display automatically in Adobe Indesign and most desktop publishing software. Adobe Illustrator may need to be configured to do so, a simple task of turning on “Ligatures.” (However, the alternate ligatures will not be typed, owing to the lack of “rand” functionality in Adobe software.) Affinity software should show them automatically. When a word is typed which has two instances of the same letter consecutively, such as the word “effective,” certain letter pairs which I have programmed into the font will be replaced with the ligature automatically. This is desirable because, when hand lettering, a letterer will often make the first instance of the letter narrower than the second one so that, together, they do not take up too much room. Without the ligatures, two instances of the same alternate, typed consecutively, might not look good. Otherwise, although there is some variation in the widths of each alternate, it could happen that the wider versions appears first and then a narrower instance of the alternate. Visually, this seems unnatural. Therefore, in an attempt to compensate, if that letter form is among the ligatures, when the second letter is typed the computer will recognize the letter pair and replace both with the ligature which contains both letters, the first one being narrower and the second one, wider. The ligatures included in the font are: CC, FF, LL. I did not include an alternate "L + Space" for wider Ls on the end of a word since my "L"s are pretty wide as it is. Nor did I include an alternate of a "T" with a longer right-hand width on the end of a word or sentence, for the same reason. I needed to economize and prioritize, given the size of the font generally. I did not include a "TT" ligature, either, because, after some experimentation, it seemed superfluous.

Letter Frequency

Originally, while designing the characters, I conceived of the font containing, first, 200 alternates of each letter, then, 300, and, as I continued to work, even more. When time came to actually create the font, I began to realize, because of the magnitude of the work involved, that it might be best to economize on letters that are not used often. This led me to research letter frequency. *Wikipedia* online encyclopedia, Oxford University, and Cornell University all have published studies which show virtually the same percentage of incidence of characters of the alphabet in numerous sample texts, such as novels. Other studies reflect similar results, some using larger sample texts, others smaller. Below is the average that I came up with. While the number of times certain letters are used is dramatically different than the lesser-used characters, I have more or less averaged them toward the center, adding a greater percentage to the lesser-used end of the scale and a lesser percentage for the more-used end. Some of my reasoning was practical and expedient. In a projected number of 6,000 alternates, for example, the most frequent letters would have, some, over 500 alternates, which seemed to me a little excessive; and the lesser-used letters less than sixty or so. In any event, this is more or less the manner that I proceeded in determining how many alternates of each character in relation to the others that I chose.

The percentages of incidence are roughly as follows:

A	8%
B	2%
C	3%
D	4%
E	11%
F	2%
G	2%
H	3%
I	7%
J	1%
K	1%
L	5%
M	3%
N	7%
O	7%
P	3%
Q	1%
R	7%
S	6%
T	7%
U	4%
V	1%
W	1%
X	1%
Y	2%
Z	1%

Vowels are used more often than consonants. The relative numbers of letters reflects this.

Number of Alternates in Regular Font

Exclamation	120
Quote Double Straight	30
Pound	6
Dollar Sign	6
Percent	9
Ampersand	12
Apostrophe	30
Parenthesis Left	8
Parenthesis Right	8
Asterix	8
Plus Sign	6
Coma	40
Hyphen	18
Period	50
Slash	6
Zero	20
One	20
Two	20
Three	20
Four	20
Five	20
Six	20
Seven	20
Eight	20
Nine	20
Colon	13
Semi-Colon	8
Less	6
Equal	6
Greater	6
Question Mark	80
Rosen Exclam (At Symbol)	80
B (Back-Slanted B)	30
Uppercase I	240
Uppercase Y	208
Left Bracket	20
Forward Slash	4
Right Bracket	20
Single Em Dash (Space Circumflex)	56
Underscore	8

A	360
B	200
C	240
D	280
E	400
F	200
G	208
H	240
Lowercase I	60
J	100
K	132
L	240
M	240
N	340
O	360
P	240
Q	100
R	340
S	320
T	340
U	280
V	100
W	144
X	100
Lowercase Y	200
Z	100
Brace Left	40
Brace Right	40
Ellipsis	22
Typographer’s Quotes Left	40
Typographer’s Quotes Right	40
Typographer’s Quotes Double Left	50
Typographer’s Quotes Double Right	50
Em Dash	50

Johnny Comics Italic font contains exactly half this number of alternates relative to each character group; less certain optional characters which I did not include in it, such as the “emphasis” brackets, which I uniquely designed in the Regular font.*

* If, for instance, one wants to use the original “emphasis” brackets in a

Accented Characters

Because a great number of alternates requires such a vast number of kerning pairs, the kerning of accented characters was deemed too laborious This should not particularly pose a problem, however. Generally, in English, accented characters are rarely used in comics anyway. When they are: insert the character from the Glyphs ("Special Characters" or whatever the respective software calls the panel to access the alternates) panel manually. After typing the sentence that they appear in, you will need to go back, select that character, and manually kern it, utilizing the design (typesetting) or desktop publishing software's capability for doing so. Better still, having once manually kerned a letter pair containing an accented character (this might appear regularly in a Latinate name, for example), copy it to the clip board, one by one select all instances in which you inserted a dummy character in lieu if it, and paste the kerning pair containing the accented character.

If you want variation in the accented characters (for example, a name, which contains an accented character and appears often), type the word with the regular base character, without applying any accent. Each instance of the base character will be an alternate (if the *rand* feature is utilized) Then, in a different place or on a different layer, insert only the accent from the "Glyphs" or "Special Characters" panel (or corresponding panel). Select the accent and drag it over the base character. Copy and paste it, repeating the process in each instance the accented character appears in the text. If you, further, want the accent to seem unique, transform or distort it slightly, utilizing the capabilities in the design/publishing software to do so.

Letter-Number Combinations

string of Italic text, simply change to the Regular font and insert them. If they are not oblique enough to suit your taste, skew/shear them in the desktop publishing or design software.

Nor did I kern letters and numbers together; as in the name "Apartment 3G" or as they might appear together in certain codes. The permutations are so vast and the potential usage so small that, here also, I deemed it an unneccessary labor. As with accented characters, the typesetter/letter will need to manually kern the glyphs in the desktop publishing software.

Random Shift of Bold Italic

I have more or less arranged the letters along the baseline and kerned them naturally. But oftentimes in comic books, Bold Italic letters, which are used for emphasis, are also used for "extreme" emphasis, such as shouting; in which case the letters are more informally juxtaposed, jumping above or dropping below the baseline. If the letterer/typesetter desires more erratic or fluid handwritten letter placement, this will need to be accomplished using the capabilities in the publishing or design software. Letters can be selected and nudged upward or downward, and kerning can be altered. For more variability, letters can be distorted: scaled (stretched horizontally or vertically) and skewed/sheared, as well as other vector effects applied for distortion. Otherwise, one will need to use a random baseline shift script or capability and apply it with a Style.

If one is working between softwares that supports the *rand* (Randomize) feature and design or photo software which does not, you will need to turn the text into outlines. One can also transform the outline text similarly. Specifically, vectorized individual letters can be shifted, moved, scaled, skewed/sheared, and other distortion effects, such as warps, applied.

Point Size and Printing

Here are some dimensions of Gaspar Saladino's letters for DCs 1950 *Strange Adventures*.

- Image dimension per page: 8 5/16" x 6 1/16". Decimal image dimension: 8.3125" x 6.0625".

- Text height from top of letters one sentence to top of letter below: 1/8″ or .125. Actual letter height is over 1/16″ or .0625″: it is 2 millimeters, which is 5/64″ or .078125″.

- Image height divided by full sentence height with leading: 8.3125″ / .0625″ = 133 lines per page. Space: 1 millimeter = 3/64″ or .046875″.

- 4 millimeters plus 1 millimeter space equals 5 millimeters total from top of letters one sentence to top of letters in sentence below. 5 millimeters equals 13/64″ or .203125″.

Notice that Gaspar Saladino letters are .78″ tall with . 047″ space. With regard to 300 DPI printing, 9 point Johnny Comics text is .074″ tall. Therefore, Saladino's lettering is a little larger than 9 points. 9.5 point text is .079″ tall, and 10 point is .0825.″ Personally, I typeset at 9 pt with leading 8.5 pt.