

# Using XP8

Macintosh

Version 1.0

by  
Greg  
Swann

## What you need to run XP8

1. An Apple Macintosh computer (SE or above)
2. A means of converting native-format word processor files to Plain Text
3. QuarkXPress 3.0 or above with the XPress Tags filter visible to Quark.

## XP8 is shareware

XP8 is distributed under the shareware system. That means, if you use it, you should pay for it. Considering the amount of money it can save you, the registration fee of \$50 is trivial. Of perhaps greater moment, technical support and product upgrades will be extended ONLY to registered users. Future upgrades to XP8 will not be distributed as shareware. Users who register will receive a disk containing the latest version of XP8, plus copies of all of some other useful text conversion utilities. Registration information is contained elsewhere in this archive, but, in case it is missing, register by sending a check or money order for \$50 payable to Greg Swann; mail it to:

Greg Swann  
P.O. Box 1724  
Andover, MA 01810

Site licenses are available at these rates: 2-10 copies, \$45 each; 11+ copies, \$40 each. Vertical-market and site-specific versions are available at negotiated rates.

## Distribution requirements

XP8 and the other utilities in this archive can be distributed freely PROVIDED THAT all of the original, unmodified files are contained in the archive. These files are: Using XP8 (this manual), Demonstration of XP8, and XP8.

## Typical legal verbiage

This software is not warranted to be useful to anyone for anything. It is provided as is, without warranty of any kind. Anything bad that happens is your own damn fault, and you should have known better. If you attempt to sue me, I will declare personal bankruptcy and run off to sleep in the rain with the Patagonians. So there!

## Atypical legal verbiage

This is MINE. I wrote it, and neither you nor Big Mother nor the Tooth Fairy helped at all (nor would I have wanted you to). It represents hundreds of hours spent staring at a monitor, thinking, for instance, about what other bizarre ways people have come up with for quoting speech. Those hours I could have spent playing with my children or snoozing or having great sex. Instead, I did this. So please don't make me regret making the end product of that labor available to you. I hope XP8 helps you, and I believe that it will. But if it doesn't, please don't attempt to punish me for providing you with a value that you did not provide for yourself.

Likewise, don't rip me off. XP8, related utilities, their source and executable code, and this manual are © 1992 by Greg Swann, and all rights are most emphatically reserved.

## Apology for above verbiage

To the 99.44% of humanity who know the difference between mine and thine, I apologize for the above hectoring disclaimers. It's not you I'm talking to.

## Gratitude

XP8 would not have come to fruition without the support of my wife, Ann Swann, who gracefully endures far more than is just. Stephen Owades had a crucial early influence in 1990. Maureen Tutavilla proofread this manual and provided sound typographic counsel and practical testing. And, among the beta testers, Phil Gaskill's contribution was pivotal.

## The short course...

1. Convert the source file to Text Only (using MacLink or a similar utility)
2. Run the Text Only file through XP8
3. Get Text on the exported file from Quark ( Get Text, Include Style Sheets)
4. Plug and chug...

# The long course...

This is easy. It seems hard, because there are a lot of steps, but really it is just a matter of taking things slowly and logically.

This may not be the best way of doing these; I've been working on conversions for 12 years, and my methods have improved through time. But I can testify that it is a good way: it really works, really fast, really cheap. I know of many methods that take far longer and introduce many more opportunities for error. But, so far, I haven't found a way of doing conversions that beats this one.

There are a lot of little steps involving many software programs and the risk of error is high. So as

a first proviso: be careful. Check your work as you go along, and save the interim steps until you're sure you're right. In most cases, it's faster to go back to the last good step and rework from there than to try to correct a botched file. Our goal is to produce the most work with the least effort.

As with many Mac problems, the shortest distance between two points is two or more applications. As above, the goal is to get it done quickly, and one quick trip through XP8 will save hours of fussing with Find/Change. Likewise, it is usually worthwhile to key substantial amounts of text into the PC and convert from there: Mac keyboards suck, and Macs themselves cost big dollars. A reasonable division of labor, on a big job, is to have one person key into the PC while another sets up the geometry and styling on the Mac. The converted text should spill in with precious little extra labor.

XP8 converts Macintosh or MS-DOS Plain Text files into a form that can be directly imported by the QuarkXPress XPress Tags filter.

To use this software, you need Plain Text files (with or without XPress Tags or XP8 Mark-Up Tags embedded). If you are working from native-format word processor files, you need some way of converting them to Plain Text.

XP8 is multi-finder aware, 32-bit clean, and System 7 compatible. You can leave it open for as long as you need it. Dialog box settings are global, so they will be preserved throughout the entire XP8 session. Likewise, you can leave the help window open as you work through the dialog boxes.

The files XP8 creates are of the type TEXT, so they can be edited by any Macintosh word processor. Moreover, your original source files are not altered in any way.

On the following pages, we'll go through the short course the long way, covering every step and some common variations. In most cases, the short course covers everything. The long course will promote understanding, and will help to deal with oddball cases. At all events, the important points are these: keep it simple, be methodical, keep it cheap. They really are easy.

# The short course elaborated...

## 1. Convert the source file to Text Only

You need Text Only files for the rest of the steps in the process to work. Word processor files are filled with all kinds of garbage that is internally meaningful to the application, but which is useless for typesetting purposes and onerous to remove (if in fact the text itself makes sense in its internal file format). Programs like MacLink convert the WP format to another specified format.

Mac paragraphs versus PC paragraphs: The Macintosh stores a paragraph of text as one long line, with only one carriage return character at the end. PC Text Only paragraphs consist of many individual lines, each of which is terminated with a carriage return and a line feed (line feeds will be discussed later). In consequence, PC paragraphs are paragraphs by interpretation only; there is no logical way for software to know conclusively that a subhead like the one above is a stand-alone paragraph, rather than the first line of the text paragraph that follows it.

If your conversion utility is on the Macintosh, a good format to convert to is Microsoft Word. That way, if you need to do fast edits, you can do them in Word prior to running XP8. To do fast changes in Word, use Change from the Utilities menu (Cmd-H). If you are at the very top of the file, you can Change All very, very quickly. Note that this should only be used for one or two fast changes, since, though it is fast, iterative search and replace is very time consuming in bulk.

Save your file from Word as Text Only (not Text Only With Line Breaks).

Okay, so now you have a file that is ready to go to XP8. If you don't need to edit it for Style Tag calls, you can skip ahead to Step 2. If you do, then hang out.

All of which brings up some interesting points, such as why you'd want to edit a file before running XP8. And all of that will be dealt with in the next section.

## 2. Run the Text Only file through XP8

Simple stuff first. Let's say you have text that is just that: text. Whether it's an ad, a booklet or a book, if it's all just straight running matter, or the head/sub/body illusion of something other than running matter, you can just run it through XP8 and deal with the styling once you have it back in Quark. Styling in Word might be slightly faster, particularly if you have a lot of style changes, but not by much. Just slam the file through XP8, Get Text and gloss and toss. No problems.

But... For jobs where it really makes a difference, XP8 can make a really big difference, saving monstrous amounts of time and allowing functions that would be onerous to the point of financial disaster by any other means. So if your job is just too, too simple, expiate it and get off the road.

But if it's studly, hang out: XP8 is too totally awesome.

What is XP8? XP8 is a piece of file conversion software that I wrote two years ago, specifically because I was so disgusted with what I was having to do to convert files into the Macintosh. What it does is clean up Text Only files and make them Mac-ready. It does most of what it can do transparently, without having to be told. For instance, it soaks up multiple carriage returns, cleans up all the quotes and dashes, makes bullets of commonly used simulated bullets, turns double spaces into single spaces and turns strings of three or more spaces into tabs, converts strings of five or more periods into double-tabs (which can be styled with leaders in XPress), reformat's PC-like paragraphs into Mac-like paragraphs (intelligently guessing which type of file it is looking at), strips leading and trailing spaces, and on and on and on. It does everything that makes manual Mac conversions such a drag, and does them all at about 5K per second. More to the point, XP8 has a mark-up language that allows you to pre-code a file for Quark, such that the converted file will come into Quark needing little if any formatting, and only aesthetic tweaking to be ready to travel. It is this last feature that makes it worthwhile to edit files bound for XP8.

XP8 makes use of the QuarkXPress XPress Tags language. This language is documented in the Quark manuals, to which you are referred.

In addition, XP8 has its own mnemonic language, which uses brackets to flag commands. This is the XP8 mnemonic language. At the time that I began XP8, I was still importing a great deal of text from the Quadex, so I built in support for the lower 32 Quadex characters. In the list below, the codes in the left column denote their Quadex keyboard equivalents; the text to the right denotes what they will be converted to. To use this feature, you need to make a Quadex Y-Translate table that searches for the literal code in the Quadex file and replaces with the text in the left column.

XP8 Command	Quadex Literal Character
[cbx]	Closed box
[tr]	Shifted return
[ct.s]	Cents sign
[cpt]	Copyright
[ddgr]	Double dagger
[degr]	Degree
[divs]	Divided by sign
[embul]	Em bullet
[emsp]	Em space
[enbul]	En bullet
[ensp]	En space
[figsp]	Figure space
[lar]	Left arrow
[lks]	English pound sign
[ftr]	Hard return; paragraph ending
[mdsh]	Em dash
[ndsh]	En dash
[obx]	Open box

[para]	Paragraph symbol
[ran]	Right arrow
[regm]	Registration mark
[rg]	Shifted return
[sagr]	Single dagger
[secmk]	Section mark
[thsp]	Thin space
[timz]	Times sign
[tm]	Trademark
[uques]	Upside down question mark
[uxclm]	Upside down exclamation point

These are the XP8 mark-up tags. Many mnemonics are added here, using square brackets. These codes emulate common Quadex functions. Additionally, dingbat support is built in with calls to 26 dingbats fonts

XP8 Dingbat Mnemonic Code	Result	
[a0]	[a255]	The character in the Carta font
[b0]	[b255]	The character in the Bundesbahn Pi 1 font
[c0]	[c255]	The character in the Bundesbahn Pi 2 font
[d0]	[d255]	The character in the Bundesbahn Pi 3 font
[e0]	[e255]	The character in the European Pi 1 font
[f0]	[f255]	The character in the European Pi 2 font
[g0]	[g255]	The character in the European Pi 3 font
[h0]	[h255]	The character in the European Pi 4 font
[i0]	[i255]	The character in the New Cent Fracs Reg
font		
[j0]	[j255]	The character in the New Cent Fracs Bld font
[k0]	[k255]	The character in the Helvetica Fracs Reg font
[l0]	[l255]	The character in the Helvetica Fracs Bld font
[m0]	[m255]	The character in the MICR font
[n0]	[n255]	The character in the Universal News font
[o0]	[o255]	The character in the OCR A font
[p0]	[p255]	The character in the OCR B font
[q0]	[q255]	The character in the Chicago font
[r0]	[r255]	The character in the Sonata font
[s0]	[s255]	The character in the Symbol font
[t0]	[t255]	The character in the Lucida Math Ext font
[u0]	[u255]	The character in the Lucida Math Italic font
[v0]	[v255]	The character in the Lucida Math Symbol
font		
[w0]	[w255]	The character in the Wood Type Orn 1 font
[x0]	[x255]	The character in the Wood Type Orn 2 font
[y0]	[y255]	The character in the Universal Math Pi font
[z0]	[z255]	The character in the Zapf Dingbats font

These are one-character dingbat calls, strictly to make it convenient to drop out of the style s font for one character. If you need a font that is not supported here, you can always convert to one of these, then Find/Change from within Quark. Alternatively, you can use the <f"font name"> call, but you must address the font by its true system name (viz., <f"CenturyExpanded">).

XP8 Mnemonic Code	Result
[...]	Spacey 3-dot ellipsis (w/space at either side)
[...]	Spacey 4-dot ellipsis (w/space at either side)
[cbax]	Closed box
[crt \$]	Cents sign
[cprt]	Copyright
[ctr]	Shifted return

[ddgr]	Double dagger
[degr]	Degree
[divs]	Divided by sign
[embox]	Em box
[embul]	Em bullet
[emobox]	Em open box
[embul]	Em open bullet
[emsp]	Em space
[enbul]	En bullet
[endbul]	En open bullet
[ensp]	En space
[figsp]	Figure space
[foot]	Foot mark (math single prime)
[fra/xxx/xxx]	Built-up fraction, xxx is any text
[h]	Discretionary hyphen
[inch]	Inch mark (math double prime)
[inf/xxx]	Inferior, xxx is any text
[larr]	Left arrow
[lcs]	English pound sign
[¶]	Hard return; paragraph ending
[mdsh]	Em dash
[ndsh]	En dash
[obox]	Open box
[para]	Paragraph symbol
[p/m]	Plus or minus sign
[qm]	Tab only
[qx]	Tab only
[rarr]	Right arrow
[regm]	Registration mark
[rft]	Shifted return
[f]	Return to normal face for style
[sdgr]	Single dagger
[secmk]	Section mark
[sup/xxx]	Superior, xxx is any text
[thbul]	Thin bullet
[thdbul]	Thin open bullet
[thsp]	Thin space
[times]	Times sign
[tm]	Trademark
[uques]	Upside down question mark
[uxclm]	Upside down exclamation point
[yen]	Japanese Yen symbol

#### Case Conversion Commands

#### Result

[caps/xxx]	CONVERT TO ALL CAPS
[lowr/xxx]	convert to all lower case
[sent/xxx]	Convert to sentence caps
[upst/xxx]	Convert To Upstyle Initial Caps
[dnst/xxx]	Convert to Downstyle Initial Caps

These commands will change the case of copy; xxx is any copy, all in one paragraph. These are hard (permanent) conversions, not styles like <K> or <H>. [caps/xxx] is redundant given <K>, but I've added it because a hard conversion is forevermore secure from being lost to the No Style black hole. [lowr/xxx] is obvious. [sent/xxx] will convert any text to all lower case, with the first character in the string and the first alphabetical character after a period converted to caps. [upst/xxx] converts the initial character of every word (delimited by space or punctuation) to caps, all else lower case. [dnst/xxx] is the same as [upst/xxx] except that words with three or fewer characters are turned down. The latter four can make use of the <H> toggle (e.g., <H>[dnst/sPaStIc Is thE



InPuT] <H> would turn into <H>Spastic is the Input<H>). And I expect that it would be the rare case that the latter four will do exactly what is needed; they'll simply put you closer than you were to what you ultimately need.

Most of these tags are useless most of the time. They're there on the presumption that it's better to have a nuke and not need it, etc. I use the calls to style sheets (@style: ) all the time. Of the others, the ones I use most commonly are font changes (<I>, <B>, etc.), [fra 1/2] and the copyright, trademark and registered trademark dingbats. For most of these, you can just do a search in Word. If there are only a few fractions in your source file, a search will do fine. Or you can fall back on XP8, which will do them all, including fractions whose set-ups vary (as they of ten do, viz., 1-1/2, 1 1/2).

Here's something cool. XP8 is making a large number of stylistic decisions about the source copy as it dugs. For instance, all hard hyphens are preceded and succeeded by discretionary hyphens. That way, compound words will break only at the hard hyphen. Likewise, the last word in every paragraph is preceded by a discretionary hyphen, so an expiated file will never break into a widow. Em and en dashes are preceded and succeeded by discretionary returns, so that they'll break up if they will fit; both kinds of dashes also have a <k6> on either side to open them up a bit. Slashes with four or more alphabetical characters following them are followed by a discretionary return, to permit the construct to break after the slash. A <k 11> will appear at either side of every numeral 1 (except in tables). Strings of five or more consecutive periods are converted to two tabs; from this, you can create a leader tab; the first of the two tabs is tracked to <t41>, to get a nice, open leader. Strings of five or more underbars are converted to a tab, which can be styled in Quark to an underbar character.

What's the point of all this? Well, for a first thing, XP8 is doing all it can to save you time. Because a certain number of typographic modifications should always (or almost always) be done, and since these can be unambiguously coded, XP8 takes care of them. For a second, doing these modifications manually takes monstrous amounts of time. For a third, it is entirely too easy to miss them (or decide to overlook them) when doing them manually. For a fourth—and this is really cool

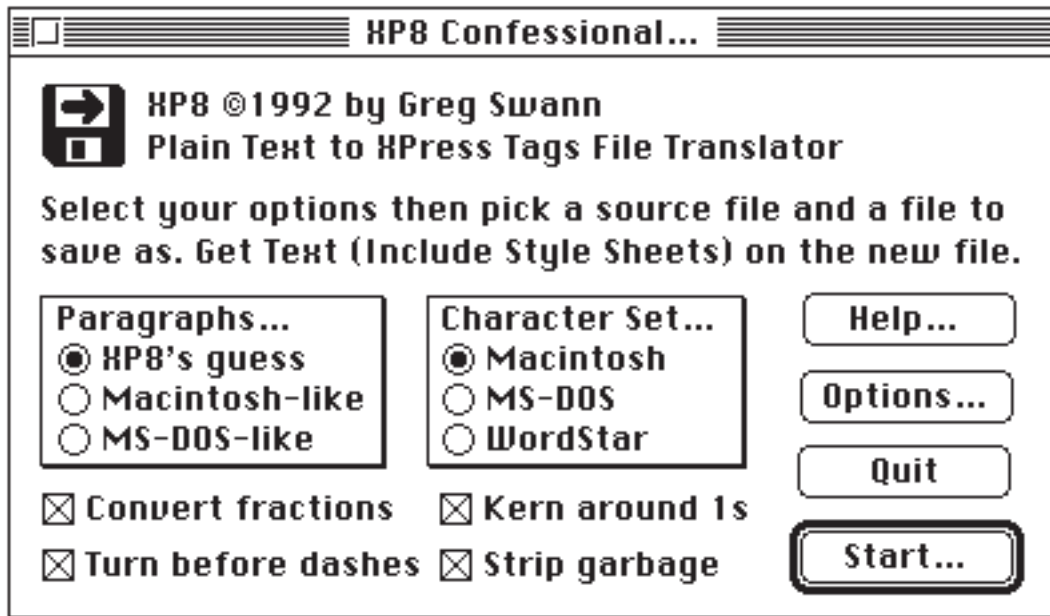
XP8 can perform these modifications on standing Quark text. The how-to of this is discussed later). For a fifth, it's free: really fast, really cheap. XP8's type manipulations will give a high-budget look to low-budget work, and they will make high-budget work easier to modify.

Now, then, to the heart of the matter:

To use XP8, double-click on the application. You will be presented with dialog boxes offering a number of options.

The XP8 Confessional... and XP8 Options... dialog boxes list features of the program that are user-selectable. But there are many XP8 functions that are performed automatically. For example, XP8 intelligently converts or omits leading, trailing and midline spaces; it converts runs of periods and underbars to tabs, which can be styled in XPress as leader or underbar tabs; it parses its own Mark-Up Tagging language that adds significantly to the number of effects that can be achieved with XPress Tags; it converts quotes better than XPress Tags, and better than any quote converter we've seen; it converts dashes, bullets and other punctuation; and it exerts a gentle influence on the typographic perfection of the rag. Many of these functions are switchable, and there are other functions that are not always needed that can be turned on for special cases. Examples of the latter include intelligent conversions of ligatures, figures and financial tables.

On the following pages, we'll be going through the XP8 dialog boxes in detail.



#### PARAGRAPHS...

Macintosh paragraphs consist of one long line terminated by a carriage return. MD-DOS paragraphs consist of a number of individual lines, each of which is terminated by a carriage return (and, of ten but not always, a line feed). XP8 s guess , the default, interrogates the source file and makes an intelligent guess about what kind of file it is processing. It is accurate in large measure, but it is not at all of fended if you tell it whether the paragraphs in your file are Macintosh-like or MS-DOS-like . If you don t know, let XP8 guess. But if you do, check the appropriate radio button and the conversion will be faster. Note that you can use this control to force an interpretation that is not literally true. If you have a Mac-like listing of bulleted items that you would like to turn into a single paragraph, tell XP8 that the file is DOS-like. If you have a DOS-based financial table, tell XP8 that the paragraphs are Mac-like, even though this isn t literally true.

#### CHARACTER SET.

The Macintosh and MS-DOS character sets differ in the eight-bit characters. In Macintosh ASCII, decimal 165 is a bullet ( ), while in DOS, that same slot is occupied by the character . As a third complication, that seemingly ubiquitous senior citizen of DOS-dom, WordStar , uses the eighth bit to flag word endings. You can use this group of radio buttons to force the character set interpretation XP8 uses when translating your file. In most cases, you ll be using Macintosh files, so you should leave this alone. If you have a DOS-file, though, selecting MS-DOS will cause XP8 to convert any eight-bit characters found to their closest Macintosh equivalent. If you select WordStar , the eighth bit will be stripped, revealing the true seven-bit character hidden under that umbrella. The styling attributes of WordStar files will also be converted to the appropriate XPress Tags.

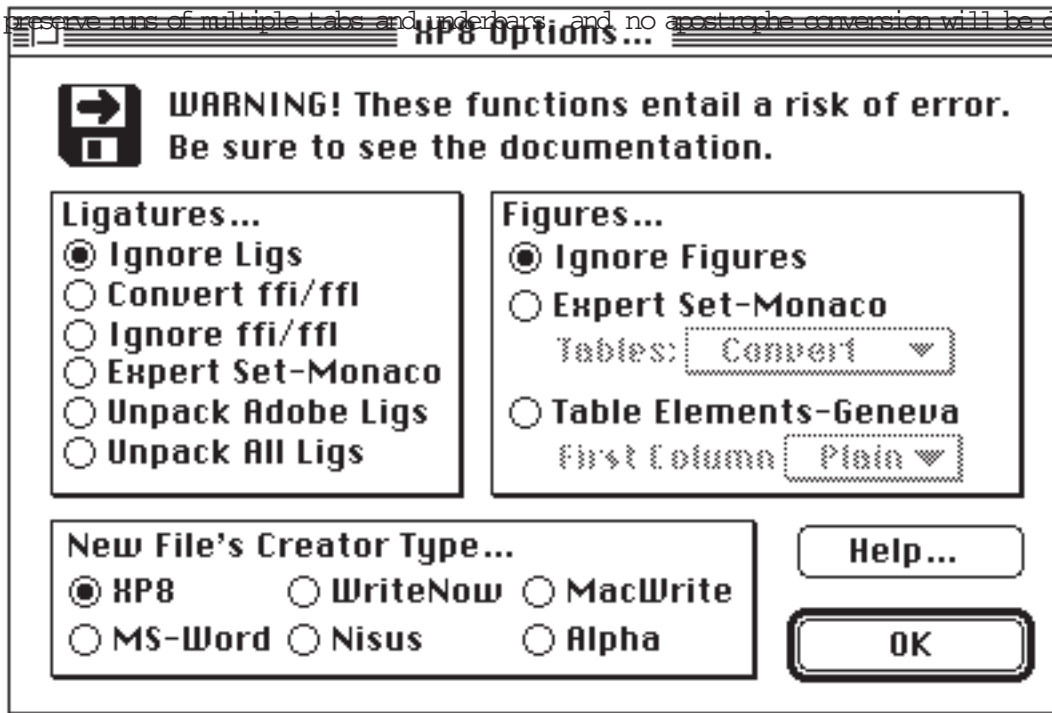
#### CHECK BOXES...

If you accept the default for Convert fractions , XP8 will search for common word processor fraction-euphemisms and replace them with the appropriate XPress Tags coding. There is a bug in the XPress Tags filter that shipped with XPress 3.0 that will make this function fail to work. If you are not using XPress 3 1 (with version 1.5+ of the XPress Tags filter), turn this feature of f

Kern around 1s will put a kern of 11 units at either side of every numeral 1 in your text, EXCEPT IN TABLES. In most Macintosh fonts, this is a big help. The value 11 was chosen to make it easy to search for and change (about which more later).

Strip garbage is what XP8 is all about. People put a lot of crap in their files, and most of it is contra-utitarian. XP8 is stripping most of this junk without so much as a by your leave . But even the best of us has hit two returns when what we really should have done was put a Space Before of the amount of the leading. So, XP8 gives you the option to preserve this kind of garbage. You should only use this on files that are already free of all unwanted double returns. Unchecking Strip garbage will also

preserve runs of multiple tabs and underbars, and no apostrophe conversion will be done.



XP8 S OPTIONS...  
are dangerous!

Seriously! Nothing is buggy. Everything works as documented. But what is being done is inherently risky. If you are not a finicky typographer with a strong technical bent, accept the defaults in this box and save yourself some grief.

That being said, there is much benefit to be realized from the judicious use of the XP8 Options dialog box. For example, you can take advantage of old style figures from Expert Set or Small Caps and Old Style Figures fonts without doing 10 laborious searches. Or you can swap the tabular elements in a financial table to another font without dragging through the text a line at a time. Or you can do both simultaneously!

The watchword is: Be Careful! Make sure that what you ask for is what you really want. And remember: your original file is unchanged. If you make a mistake, you can always go back to the original file and expiate it again.

#### LIGATURES...

The default, `Ignore Ligs`, will pass untranslated all ligatures and ligature candidates found in the source. `Convert ffi/ffl` will convert every `fi` and `fl` pair in the source to the appropriate standard Adobe-font-layout ligature. `Ignore ffi/ffl` leaves the triplets `ffi` and `ffl` unconverted.

If you select `Expert Set-Monaco`, `ff`, `fi`, `fl`, `ffi`, and `ffl` (and any Adobe-layout ligatures found) will be converted to the appropriate Expert Set ligature character, WITH THE TYPEFACE SET TO MONACO. You can use your word processor or Font Usage in Quark to change Monaco to the actual Expert Set font you will be using.

Note that ligatures have no Small Caps or All Caps counterparts. If you ask for ligatures and you're using `<K>` or `<H>`, be prepared for some unsightly typography.

`Unpack Adobe Ligs` will convert the Adobe-layout ligatures to their Plain Text counterparts, while `Unpack All Ligs` will do the same for both Adobe-layout and Expert Set ligatures. In the latter case, the leading and trailing Xpress Tags coding is stripped.

FIGURES...

Ignore Figures will pass all numbers in the source text untranslated.

Expert Set-Monaco will pass numbers untranslated, BUT THE TYPEFACE WILL BE SET TO MONACO. You can use your word processor or Font Usage in Quark to change Monaco to the actual Expert Set or Small Caps and Old Style Figures font you will be using. The associated pop-up menu gives you a choice of what to do about the numbers and punctuation making up the elements of a table. Convert Tables will put the table elements in Monaco as well; this has a high potential for error, since Expert Set character slots are not perfectly consistent. Ignore Tables will pass the tabular portions of your text untranslated. Tables To Geneva will pass tabular elements untranslated, BUT THE TYPEFACE WILL BE SET TO GENEVA. By checking this button, you will make all of the figures in running matter and table stubs convertible to one font, with all of the tabular matter convertible to another (using Font Usage or your word processor).

The last button, Table Elements-Geneva, will pass tabular elements untranslated, BUT THE TYPEFACE WILL BE SET TO GENEVA. This is a fast, cheap way to get table stubs in one font (the Style Sheet's default), with the figures and related punctuation in another (that Garamond 3 with Helvetica 93 thing that would be 96 tears with the mouse). The associated pop-up menu lets you specify whether the first tabular column should be set in Plain or Bold.

For the record, a tabular element is triggered by any character from decimal 35 to 57 THAT IS PRECEDED BY A TAB. The condition persists up to but not including the first character found that is NOT in that range.

NEW FILE S CREATOR TYPE...

This function does nothing more than let you establish the Creator Type of the newly created file. If you routinely edit in Word, for instance, selecting the MS-Word button will create the new file as type TEXT with a Creator Type of MSWD. If you double-click on the file, Word will be launched and the file will be opened.

When you hit the Start... button, you will be prompted for a file to convert. You will then be prompted for a file to Save As, with XP8 suggesting a name. This new file will be created where you specify and your original file remains unaltered.

### Time out for important technical information:

Eight-bit characters: Modern computers base their notions of characters on a code called ASCII (American Standard Code for Information Interchange). ASCII was defined as a seven-bit code, which means that it can contain up to 128 unique codes ( $2^7$ ). The lower 32 are used for communications and printer control and for space modifier characters. Qualex violates the ASCII standard by using these for its own internal uses, which is why XP8 translates these to their functional equivalents in Mac-ASCII. The upper 96 are the punctuation, numeric, and upper and lower case alphabetical characters. A byte, the lingua franca of modern computing, uses eight bits, not seven, so it can contain up to 256 unique codes ( $2^8$ ). Both the PC and the Macintosh use that eighth bit to make an extended character set, and these codes are called eight-bit characters. Unfortunately, the two code sets are different.

Now, back to XP8.

One last bit, and then we'll drop out of this endless sidebar: XP8 makes a guess about what type of file it's looking at. Before it starts translating the file, it looks to see if the source is composed of Mac-like or PC-like paragraphs. As I said before, software can only guess, because DOS uses the same characters to denote line endings and paragraph endings. In translating, XP8 looks for firm clues. If it has decided the paragraphs are Mac-like, then every carriage return is treated as a paragraph ending. If it has decided the paragraphs are PC-like, then any paragraph preceded by a double carriage return or beginning with a space-modifier (tab or space), commonly-used bullet character (\* or o) or style tag is treated as a new paragraph. The guess XP8 makes before translating is as follows: if the average line length is below 40 characters and no line exceeds 200 characters, then Mac-like paragraphs are presumed (in this case, the assumption is of a listing, like a phone-book listing or baseball stats). If the average line length exceeds 200 characters, Mac-like paragraphs are presumed (since PC-like paragraphs are usually from 50 to 80 characters

per line). Everything in the middle is treated as PC-like paragraphs. But: what if you have a financial job and now have PC-like paragraphs composed of extremely long lines? Force the interpretation by telling XP8 that the paragraphs are Mac-like, even though this isn't literally true. Likewise, suppose you have a file that contains DOS eight-bit characters. XP8's presumption is that all eight-bit characters it finds in the source are from the Macintosh code set. You can force XP8 to convert DOS eight-bit characters to their closest Mac equivalents by using the MS-DOS Character Set radio button.

So that's the deal with XP8. I've devoted so much attention to it, because it is my closest approach to date to the be-all, end-all file conversion utility. Make it your friend, and it will save you many hours of time, and, potentially, embarrassing errors. For example, the quote-conversion utility is so far beyond Quark's Convert Quotes function that its error rate is measured in the parts per million. But: most of what XP8 does is done either from the pseudo-certainty of static rules or is a guess. The guesses can be very sophisticated, but they are no substitute for human certainty and its local consequence, a file that is prepared with all ambiguities resolved. So, for example, every instance of \* (asterisk, space) in your file will be converted to (en bullet, space); that combo is used more often for typewriter bullets than for footnotes, and so XP8, which can't read or reason, after all, presumes that a bullet is wanted. If you want to resolve all doubt, convert the asterisks that should remain as asterisks to their Mac equivalent (<#42>). (The same thing can be done for any character that you want to be passed untranslated.)

Okay, so the deal is, we're going to edit the file we converted prior to running it past XP8. Our

reasons for doing this can be manifold: we may want to deal with certain situations that will be ambiguous to XP8, we may want to change some illogical constructs in the source, we may just want to take advantage of the fast search and replace in Word. Most likely, we'll be editing the file in order to put in calls to style tags. This is the most valuable thing we can do by editing a file prior to running XP8. First, we can code a file far more quickly in Word than we can in Quark. And second, bringing the file into Quark pre-coded will make the overall process quicker; for example, we won't create extra pages that we'll then have to delete. Note that if local changes will occur in the text, whether put in by editing or by XP8, style tags are most emphatically recommended.

**Local changes:** A Style Sheet in Quark is a set of character and paragraph attributes. Any one specific attribute in any of those components can be specified, but only one. When we code for a variation upon a defined style, as, for example, the italics in this paragraph, we are introducing a local change. Whether we do this with keyboard or mouse commands at the Mac or with Word or XP8, a local change is not part of the Style Sheet. Often, when you change from one style sheet to another in Quark, local changes are lost. This is warranted to happen if the change is from No Style to a named style, but it often happens that a style sheet change will not take effect without changing to No Style first. Any calls to font changes, to the Zapf Dingbat or Symbol fonts, or to fractions, among others, will introduce local changes; even if you have not explicitly coded for local changes, XP8 may introduce some (for example, for the fractions that XP8 automatically converts). In consequence, it is very intelligent to code all of your style sheets in Word prior to running XP8.

Note that if you do not declare Style Sheets in your source file, XP8 will set everything to the Normal style. Quark's Get Text defaults unstyled text to No Style, which will cause local changes to be lost when a style is applied to the text. Consequently, XP8 makes sure that all text comes into Quark with a style applied to it.

Okay, so we're going to edit the file in Word. We're going to style tag it, using macros, which will whip through the file way quick. We'll do searches to resolve any ambiguities in the source. One other thing we'll need to do is put in any calls to italic, bold or bold italic (or

other type style variations). When we saved Text Only from Word, we lost any of these that may have been in the original file. That's life: XP8 will get rid of all the stupid crud people put into their word processor files, but local style variations are one of the prices we have to pay for this. This is not too terrible, since it's quick enough to search through the source and type in a <I> here and an <B> there (remembering to toggle any of these off again). If there are a great many, or if you're typing, you can write a macro to do the job.

A better way: This is not for the faint of heart. You can keep local style changes that were in the original file by doing this: before saving the converted Word file as Text Only, take it into Quark (Get Text, Include MS-Word Style Sheets). In Quark, select all, set the Style to No Style, then edit the Formats dialog box to lose all indents, spaces before and after; set the lead to one value, and set the alignment and H&J to one value (in other words, no fields in the dialog box should be blank or greyed out). Save the text to a file (Save Text, XPress Tags). Edit the file in Word, eliminating the paragraph styling in the first paragraph. Find on < to eliminate local styling you don't want to keep. Finally, save the Word file as Text Only and run XP8. If that's confusing, play with it a while; don't do it on a live job until you're sure you can make it work.

Take note that it is possible to fully-code a style sheet in XPress Tags. You can establish all of your specs blind, just like we did in the bad old days. I think this is a bad idea. First, it's very easy to code style sheets in Quark, where you can see what you're doing. Second, the XPress Tags language is very intricate and verbose, and completely unforgiving of errors.

Error trapping: The XPress Tags filter does not trap for errors. This is unkind, since errors in an XPress Tags stream can cause irreparable damage to Quark files. In consequence, XP8 is exhaustively trapping for XPress Tags errors. Any text that would result in an error when you Get Text is converted to its harmless form. You'll be able to see it, but it won't damage your file.

Along the same lines: in general you're better off toggling local changes than defaulting back to plain. The code <I>italic<I> is better than <I>italic<P>; the reason is this: if you have some other local change in effect, both will be turned off by the <P>, whereas the <I> will just toggle the italics off.

### 3. Get Text on the expiated file from Quark

This is the fun part. I love to watch text spill into Quark pages. That little percentage odometer just cranks over so nice and fast, and I know that I'm saving a gallon of sweat with each fleeting number.

Here's how to maximize the satisfaction.

When you Get Text, check the Include Style Sheets box. Do this even if you haven't declared any styles. If you leave this box unchecked, the formatting XP8 has added will come through as angle-bracketed garbage.

A word (or many) about Style Sheets

We said previously that it's a good idea to declare Style Sheets before expiating. We also noted that, though possible, it's not a good idea to write Style Sheets in the Quark mark-up language. You can do literally anything with the mark-up language that you can do directly in Edit Style Sheets, but: it isn't easy, it isn't safe, and it really isn't very smart, considering how easy it is to make Style Sheets in Quark.

The ideal way of working with XP8 is to make all of your Master Pages and Style Sheets before Getting Text. (Where possible, I prefer to work with linking Masters, since I can just put my cursor in the first page, Get Text, then finger through to force page breaks. It happens sometimes that I can't do this. Text needs to be cut out and pasted in all over the place. On those occasions, I make a bogus Master Page with a single big text box in it (often split up in columns). I pull that master down next to the first page into which I'll be placing text, Get Text into the bogus page, and chop and drop into the good page. When I'm finished with the first page, I drop the bogus page down next to the second page, and so on. When I'm done, I ditch the bogus page and its master.) The beauty of working this way is that, if your logic was good, you're almost done as soon as you Get Text. It's certainly possible to write your Style Sheets after you Get Text, but it will take a lot longer, since you have to iteratively Edit Style Sheets, and since the changes you make will take time to ripple through the text. Moreover, you can make Style Sheets very quickly by inheritance.

Try this: type some text. Change the font, change the size and lead, change the indents, give it a rule under and a drop cap, run wild. Now select that one paragraph and Edit Style Sheets. Select new. Give the new Style Sheet a name and a key. Save. Voil! You have created a new Style Sheet by inheritance. You have inherited the properties of the selected text into the new Style Sheet. Very quick. Even better (when you're doing this for real), you can work the text on the screen to make sure that it is working, rather than going back to Edit Style Sheets again and again (a procedure that I heartily discourage). If you have sub-styles that are similar to a style you've already created, you can either create new ones Based On the old one or, as I do, edit the text still on the screen to the new specs and inherit again to a new Style Sheet. (I do this because I've had bad luck in the past with Based On; I also know from experience that it's a bad idea to use spaces in Style Sheet names.)

That's it: create your Masters and Styles, Get Text (remembering to Include Style Sheets), go home early (grin).

VOCATIONAL EXPIATION...

XP8 is intended to make fast work of what would otherwise be very slow work. Out of the box, it does this with alacrity. But there are some tricks you can play that will make it even faster.

1. Hear the Word. If you are converting a word processor file from its native format, try this: instead of converting to Plain Text, convert to Microsoft Word 4.0 for the Macintosh. Get Text (Include Style Sheets) into an empty Quark document. Select All the text and edit the Paragraph Formats dialog box to make all the values the same (viz., nothing should be blank or greyed out). Save Text as XPress Tags. Edit the XPress Tags text to remove any tags you don't want and to add any that you do. Finally, run the edited file through XP8, and Get Text (Include Style Sheets) into your live Quark document. By doing this, you will preserve any bolds and italics the word processor was thoughtful enough to include.

2. Expiate Xdata. You can do the same sort of thing with text that has been processed by Em Software's Xdata, PROVIDED THAT YOU HAVE NOT USED XDATA 2.0'S EXCELLENT INLINE-BOX CREATION TOOLS. Process your Xdata prototype as normal, then Select All text, Save Text as XPress Tags, expiate the tagged file, then Get Text (Include Style Sheets) into the live document (if you leave the selection highlighted, the expiated text will replace that which is there). In this way, you get Xdata's remarkable data processing abilities, plus XP8's typographic finessing.

Note that text processed by Xdata can confuse XP8's Guess. Often, the line-lengths of the paragraphs fall within the range XP8 regards as being DOS-like. So, when you expiate Xdata, be sure to force the paragraph interpretation to Mac-like.

3. Expiate Thyself! As has been indicated, XP8 can process either Plain Text or XPress Tags text. One of the best uses you can make of the program is to expiate text that has Already Been Quarked. The ls will look better, the rags will improve, and you can save yourself many dollars and many headaches in the XP8 Options... dialog box. As stated above, when you Get Text on a highlighted selection, the new text replaces the old. Painless.

4. Code for XP8. If people type for you, get them to type for XP8. And go ye and do likewise. If you write text conversion macros or software, write for XP8. With XPress Tags and the XP8 mark-up language, you can do many things that are difficult (or even impossible) to do without retyping. Whether you use QuicKeys/Tempo macros, HyperCard scripts, AWK/GREP/SED or a high-level language, XP8 gives you a simple, easy-to-remember language to code to.



## 4. Plug and chug...

Last licks: XP8 can't think for you. It can do a lot of cool stuff that is loathsome and onerous to do by hand, but it can't tell you if your Master Page geometry is wrong. It can't tell you if your Style Sheets conform to the specifications or the standards of good art. It doesn't know what face or size you're using, so it can't kern for you. It doesn't count lines, so it can't do linebreaks and pagebreaks for you. It spares you a lot of monkey labor, but it defaults on you for the human labor. XP8 will save most of the time you used to spend on conversion, but it will save very little of the time you spend on typography. Keep your eyes open and your mind turned on and XP8 will save you huge quantities of time. But if you expect it to do the whole job for you, you'll be disappointed.

But: there are some cool (and dangerous) things you can do to expiate text. As mentioned above, it is possible to globally edit style-tagged text after you have brought it into Quark. This is very dangerous, so Save your file, Save As under a new name, and work only in that copy when trying this stuff

What you need to do is select the text you want to edit and Save Text as XPress Tags. Leave the selection active on the Quark page, since you'll be replacing it in due course. Launch Word and open the text you saved. Using Change in Word, change anything you want. For example, XP8 tracks the first tab in a leader tab sequence as <t41>. You may think that's too much. If so, Change for <t41>^t and replace with, e.g., <t31>^t. You can do global kems in the same way, remembering that the <k###> command affects the next two characters, not the pair it happens to be between. (This is really dangerous, since it will add kems everywhere, including inside XPress Tag commands; if you do this, do not Change All; look at each one to make sure it's text and not a command.) When you're done, save the file as Text Only and pop back to the Quark document. With the selection still highlighted, Get Text (Include Style Sheets). Eureka! The new text replaces the old. Check carefully for unintended side-effects. If all is well, you can Save As under the old name.

(Incidentally, you can play this game with Quark, too. If you need to run the Spelling Checker on a chunk of text, you can copy what you need to massage, paste it into a bogus document, run the utilities, copy it again, then paste it back in over the standing selection in the original Quark file. Very fast, particularly on big jobs.)

One block over: How smart is XP8? We know it's brawny, because it can deal with all of those creepy word processor files, even DOS word processor files. But XP8 is also brawny enough to clean up text that is already in a Quark file, without wrecking it. This is important: there are a lot of really bad Quark files (none of them done by you, of course). In most cases, the bulk of the clean-up those files need can be done by XP8 and should be. The hyphenation control, line break control, and kerning are just too good to pass up. To expiate a file that has Already Been Quarked, Save Text as XPress Tags, run XP8, then Get Text (Include Style Sheets), replacing the old text. If you have garbage in the file that you'd rather preserve than replace, uncheck Strip Garbage when you run XP8.

As I said before, I think it's a very bad idea to iteratively re-Edit Style Sheets. That box is okay if

I only see it once in a while, but every ten minutes is too often (particularly if I have to go to the Character box). If I really, truly need to Edit Style Sheets, I tend to keep a list of changes to be made, then make them all at once. Saves time and aggravation.

In the same neighborhood, I tend not to make separate Style Sheets for every little varia-

tion in the spex. Of ten, I will force a change with a manual edit. This is contra-doctrinaire, according to Quark. Manual edits result in problems when the base style is radically changed. I can live with that, but you may not be able to. If it s a book, then, definitely, every change in spex is a different Style. If it s an ad get real! In between, let your confidence in the specifier and your level of frustration be your guide. The point is this: Edit Style Sheets is, potentially, a bad time sink. It s meant to save time; if it doesn t, don t use it.

Lastly, don t forget our good friends Shift-Option-Click and Inherit Tabs. If you put the insertion

bar into a paragraph (or highlighted range of paragraphs), then Shift-Option-Click on another paragraph in the same text link, the selected paragraphs will inherit the paragraph attributes of the paragraph clicked upon. Paragraph attributes are all of those below the dotted line in the Style menu: Alignment, Leading, Formats, Rules, Tabs and Style Sheets. If you got a sub but wanted a hed, Shift-Option-Click your troubles away. Likewise, if you are violating Quark s advice and locally formatting tabs, you can inherit them very quickly by putting a paragraph with good tabs above your paragraphs with bad tabs. Select all of them, Style Tabs, hit okay, and all the paragraphs selected will inherit the tabs used in the first paragraph. Play with these two until they re second nature. They re mildly dangerous, and they re not good practice in cases where you need later to globally Edit Style Sheets. But for small jobs, financial jobs, and jobs that you ll never see again, they re huge time-savers.

And that s the point. This whole procedure is complicated. It runs from moderately to extremely risky at various steps along the way. And it s no day at the beach, no matter what. But it s fast. It doesn t eliminate all the labor of doing conversions, but it gets rid of most of it, the worst of it, and goes a long way toward containing the rest. But the rest is up to you. You can waste every minute you ve saved, or you can find better and better ways to do the work that remains, applying the time you save to a better use (like a day at the beach!). So, as a last word, if you find yourself flailing, tired, angry, frustrated, convinced that, There s got to be a better way there is. It s not obvious, and it s probably not trivial, but it s there. Here s hoping XP8 buys you some time to look for it...

Greg Swann

2/28/92

# Appendix

## Macintosh (text font) character set

0		32 Sp	64 @	96 '	128 ~	160		192 ÷	224
1		33 !	65 A	97 a	129 -	161		193 ÷	225 •
2		34 "	66 B	98 b	130 '	162 φ		194 <	226
3		35 #	67 C	99 c	131	163 f		195 >	227
4		36 \$	68 D	100 d	132	164 §		196	228
5		37 %	69 E	101 e	133	165		197 †	229 '
6		38 &	70 F	102 f	134	166 ¶		198 ‡	230 •
7		39 '	71 G	103 g	135 Æ	167		199 «	231 `
8		40 (	72 H	104 h	136	168 fi		200 »	232 `
9	Tab	41 )	73 I	105 i	137	169 '		201	233 `
10		42 *	74 J	106 j	138	170		202	234 `
11		43 +	75 K	107 k	139 °	171 •		203	235 `
12		44 ,	76 L	108 l	140	172 α		204 ^	236 `
13		45 -	77 M	109 m	141	173 /		205	237
14		46 .	78 N	110 n	142 Ø	174 ~		206	238
15		47 /	79 O	111 o	143 Ł	175		207	239
16		48 0	80 P	112 p	144 Œ	176		208	240 —
17		49 1	81 Q	113 q	145 °	177 -		209	241
18		50 2	82 R	114 r	146	178		210	242
19		51 3	83 S	115 s	147	179		211	243
20		52 4	84 T	116 t	148	180 ¥		212	244
21		53 5	85 U	117 u	149	181		213	245 f
22		54 6	86 V	118 v	150 æ	182		214	246
23		55 7	87 W	119 w	151	183		215 "	247
24		56 8	88 X	120 x	152	184		216	248 fi
25		57 9	89 Y	121 y	153	185		217	249
26		58 :	90 Z	122 z	154	186		218 ...	250
27		59 ;	91 [	123 {	155 ı	187 "		219	251
28		60 <	92 \	124	156 œ	188 "		220	252 ,
29		61 =	93 ]	125 }	157 ø	189		221	253
30		62 >	94 ^	126 ~	158 ß	190		222 %	254
31		63 ?	95 _	127	159	191 ‡		223	255

## Univ NewsCommPi character set:

0		32 Sp	64 @	96 '	128 ~	160		192 ÷	224
1		33 !	65 A	97 a	129 -	161		193 ÷	225 •
2		34 "	66 B	98 b	130 '	162 φ		194 <	226
3		35 #	67 C	99 c	131	163 f		195 >	227
4		36 \$	68 D	100 d	132	164 §		196	228
5		37 %	69 E	101 e	133	165		197 †	229 '
6		38 &	70 F	102 f	134	166 ¶		198 ‡	230 •
7		39 '	71 G	103 g	135 Æ	167		199 «	231 `
8		40 (	72 H	104 h	136	168 fi		200 »	232 `
9	Tab	41 )	73 I	105 i	137	169 '		201	233 `
10		42 *	74 J	106 j	138	170		202	234 `
11		43 +	75 K	107 k	139 °	171 •		203	235 `
12		44 ,	76 L	108 l	140	172 α		204 ^	236 `
13		45 -	77 M	109 m	141	173 /		205	237
14		46 .	78 N	110 n	142 Ø	174 ~		206	238
15		47 /	79 O	111 o	143 Ł	175		207	239
16		48 0	80 P	112 p	144 Œ	176		208	240 —
17		49 1	81 Q	113 q	145 °	177 -		209	241
18		50 2	82 R	114 r	146	178		210	242
19		51 3	83 S	115 s	147	179		211	243
20		52 4	84 T	116 t	148	180 ¥		212	244
21		53 5	85 U	117 u	149	181		213	245 f
22		54 6	86 V	118 v	150 æ	182		214	246
23		55 7	87 W	119 w	151	183		215 "	247
24		56 8	88 X	120 x	152	184		216	248 fi
25		57 9	89 Y	121 y	153	185		217	249
26		58 :	90 Z	122 z	154	186		218 ...	250
27		59 ;	91 [	123 {	155 ı	187 "		219	251
28		60 <	92 \	124	156 œ	188 "		220	252 ,
29		61 =	93 ]	125 }	157 ø	189		221	253
30		62 >	94 ^	126 ~	158 ß	190		222 %	254
31		63 ?	95 _	127	159	191 ‡		223	255

## Symbol character set:

0		32 Sp	64 @	96 '	128 ~	160		192 ¿	224
1		33 !	65 A	97 a	129 -	161		193 i	225 •
2		34 "	66 B	98 b	130 ' .	162 φ		194 <	226
3		35 #	67 C	99 c	131	163 ₣		195 >	227
4		36 \$	68 D	100 d	132	164 \$		196 †	228
5		37 %	69 E	101 e	133	165		197 ‡	229 ' .
6		38 &	70 F	102 f	134	166 ¶		198 ‡	230 ' .
7		39 '	71 G	103 g	135 Æ	167		199 «	231 ' .
8		40 (	72 H	104 h	136	168 fi		200 »	232 ' .
9	Tab	41 )	73 I	105 i	137	169 '		201	233 ' .
10		42 *	74 J	106 j	138	170		202	234 ' .
11		43 +	75 K	107 k	139 º	171 .		203	235 ' .
12		44 ,	76 L	108 l	140	172 π		204 ^	236 ' .
13		45 -	77 M	109 m	141	173 /		205	237
14		46 .	78 N	110 n	142 Ø	174 ~		206	238
15		47 /	79 O	111 o	143 Ł	175		207	239
16		48 0	80 P	112 p	144 €	176		208	240 —
17		49 1	81 Q	113 q	145 °	177 -		209	241
18		50 2	82 R	114 r	146	178		210	242
19		51 3	83 S	115 s	147	179		211	243
20		52 4	84 T	116 t	148	180 ¥		212	244
21		53 5	85 U	117 u	149	181		213	245 f
22		54 6	86 V	118 v	150 æ	182		214	246
23		55 7	87 W	119 w	151	183		215 "	247
24		56 8	88 X	120 x	152	184		216	248 fi
25		57 9	89 Y	121 y	153	185		217	249
26		58 :	90 Z	122 z	154	186		218 ...	250
27		59 ;	91 [	123 {	155 ı	187 "		219	251
28		60 <	92 \	124	156 œ	188 "		220	252 ,
29		61 =	93 ]	125 }	157 ø	189		221	253
30		62 >	94 ^	126 ~	158 ß	190		222 %	254
31		63 ?	95 _	127	159	191 Ł		223	255

## Zapf Dingbats character set:

0		32 Sp	64 @	96 '	128 ~	160		192 ¿	224
1		33 !	65 A	97 a	129 -	161		193 i	225 •
2		34 "	66 B	98 b	130 ' .	162 φ		194 <	226
3		35 #	67 C	99 c	131	163 ₣		195 >	227
4		36 \$	68 D	100 d	132	164 \$		196 †	228
5		37 %	69 E	101 e	133	165		197 ‡	229 ' .
6		38 &	70 F	102 f	134	166 ¶		198 ‡	230 ' .
7		39 '	71 G	103 g	135 Æ	167		199 «	231 ' .
8		40 (	72 H	104 h	136	168 fi		200 »	232 ' .
9	Tab	41 )	73 I	105 i	137	169 '		201	233 ' .
10		42 *	74 J	106 j	138	170		202	234 ' .
11		43 +	75 K	107 k	139 º	171 .		203	235 ' .
12		44 ,	76 L	108 l	140	172 π		204 ^	236 ' .
13		45 -	77 M	109 m	141	173 /		205	237
14		46 .	78 N	110 n	142 Ø	174 ~		206	238
15		47 /	79 O	111 o	143 Ł	175		207	239
16		48 0	80 P	112 p	144 €	176		208	240 —
17		49 1	81 Q	113 q	145 °	177 -		209	241
18		50 2	82 R	114 r	146	178		210	242
19		51 3	83 S	115 s	147	179		211	243
20		52 4	84 T	116 t	148	180 ¥		212	244
21		53 5	85 U	117 u	149	181		213	245 f
22		54 6	86 V	118 v	150 æ	182		214	246
23		55 7	87 W	119 w	151	183		215 "	247
24		56 8	88 X	120 x	152	184		216	248 fi
25		57 9	89 Y	121 y	153	185		217	249
26		58 :	90 Z	122 z	154	186		218 ...	250
27		59 ;	91 [	123 {	155 ı	187 "		219	251
28		60 <	92 \	124	156 œ	188 "		220	252 ,
29		61 =	93 ]	125 }	157 ø	189		221	253
30		62 >	94 ^	126 ~	158 ß	190		222 %	254
31		63 ?	95 _	127	159	191 Ł		223	255

# MS-DOS character set (as translated by XP8):

0		32		64		96		128	·	160	Æ	192		224	a
1	m	33	!	65	A	97	a	129		161		193		225	b
2	l	34		66	B	98	b	130	∅	162		194		226	G
3	'	35	#	67	C	99	c	131		163	œ	195		227	p
4	fi	36	\$	68	D	100	d	132		164	æ	196		228	S
5		37	%	69	E	101	e	133		165		197		229	s
6		38	&	70	F	102	f	134		166	"	198		230	m
7		39		71	G	103	g	135		167	"	199		231	t
8		40	(	72	H	104	h	136	Ⓔ	168	ç	200		232	F
9		41	)	73	I	105	i	137	°	169		201		233	Q
10		42		74	J	106	j	138	Ł	170	<	202		234	W
11		43	+	75	K	107	k	139		171	<sup>1</sup> ... <sup>2</sup>	203		235	d
12		44	,	76	L	108	l	140		172	<sup>1</sup> ... <sup>4</sup>	204		236	
13		45		77	M	109	m	141		173	i	205		237	f
14		46	.	78	N	110	n	142	~	174	«	206		238	e
15	Y	47	/	79	O	111		143	-	175	»	207		239	«
16	%	48	0	80	P	112	p	144		176		208		240	
17		49	1	81	Q	113	q	145		177		209		241	-
18	"	50	2	82	R	114	r	146	~	178		210		242	
19		51	3	83	S	115	s	147		179		211		243	
20	¶	52	4	84	T	116	t	148		180		212		244	
21	\$	53	5	85	U	117	u	149		181		213		245	f
22		54	6	86	V	118	v	150	β	182		214		246	
23	"	55	7	87	W	119	w	151	∅	183		215		247	t
24	/	56	8	88	X	120	x	152		184		216		248	
25		57	9	89	Y	121	Y	153		185		217		249	
26	~	58	:	90	Z	122	z	154		186		218		250	
27	¤	59	;	91	[	123	{	155	φ	187		219		251	>
28		60		92	\	124		156	£	188		220		252	h
29	·	61	=	93	]	125	}	157	¥	189		221		253	<sup>2</sup>
30	s	62	>	94	^	126	~	158	P <sub>t</sub>	190		222		254	l
31	t	63	?	95	_	127		159		191		223		255	