

# **Cross-Platform Font-Encoding Problems in Acrobat PDF Files**

And assorted other stuff

By Steve Rindsberg

with much help, many valuable suggestions and gentle but persistent prodding from  
Arnis Gubins, Aandi Inston, Thomas Phinney and Kathleen Tinkel  
on Compuserve's AdobeApps and DTPForum forums. Thanks, folks.

## Contents

Cross-Platform Font-Encoding Problems in Acrobat PDF Files .....	1
Contents .....	2
What this is all about .....	3
The problem.....	4
Why does it do that? .....	5
Then came Windows. ....	6
Here's what we end up with.....	7
Here's why it happens .....	8
It's supposed to work but ... ..	10
There you are WHERE? .....	11
OK. Now I know where I am. Where's the exit, please? .....	12
You use a PC but want to spare Mac users this abuse? .....	13
About the included files.....	14
What do I do with this embarrassment of riches?.....	17
Who is this "Rindsberg"?.....	18

## **What this is all about**

This all got started when I was researching an article about moving files from PC to Mac.

Part of the article describes what happens to the "high-ascii" characters (all the oddball/symbol/foreign characters you don't find on your everyday keyboard) as they move from one platform to the other. The article series appeared in The Corel Draw Journal, published by The Cobb Group, now part of Ziff Davis. Back issues may be available, in case you're interested.

Later, the subject of cross-platform font encoding arose in the Acrobat section of the AdobeApps forum on CompuServe. It seemed like a natural extension of the work I'd already done so I looked into it a bit further. Most of what follows is a result of that and discussions with the Acrobat forum regulars, particularly Arnis Gubins, Thomas Phinney and Aandi Inston. Without their help and suggestions, none of what follows would ever have gotten sorted out (to the extent that it has).

I don't claim to be an expert at any of this; I'm just passing along the results of the homework I've done. If you find errors or omissions, I'll gladly take the blame for them, so long as you'll take the time to point them out to me. Fair enough?

Mac users will note that this has a markedly "create it on the PC and move it to the Mac" flavor. Guilty as charged. That's what I mostly do, so that's what most interested me. The logic, if not all the specifics, works similarly in the other direction as well, though.

## **The problem**

PDF users find that some characters get changed when you open PC-created PDFs on the Mac, even though this isn't supposed to happen, according to Adobe. In fact, if all goes according to plan, it doesn't happen; PDF does an extraordinary job of preserving PC encoding on the Mac.

But as with so many other well-laid plans, things that go bump in the night can derail the train, to mixmaster a metaphor or three.

Suppose you have a PC document that contains high-ascii characters, French language text with accents, for example. The accented characters will often get converted to something else when you open the same document on a Mac. The same is true going from Mac to PC.

The better cross-platform apps do what they can to correct these mismatches, and generally they do a good job, but some characters may still go astray or disappear altogether.

## Why does it do that?

Computers don't actually deal with characters. They work strictly with numbers.

To display text, computers use one of several conventions that assigns (or "maps") numbers to specific characters. This mapping is called "character encoding" and enables the computer to display an "A" when presented with a text string containing the number 65 and so on.

While we refer to these numbers in decimal or hexadecimal notation for human-readable convenience, the computer actually works with binary numbers, collections of on/off bits. Historically, many computer systems used 7-bit character sets. Since the highest number that can be expressed in seven binary bits is 128, these systems could only deal with 128 characters. One method of mapping these 128 possible numbers to characters came into common use early in compu-history and has become standardized as the ASCII character set. (ASCII = American Standard Code for Information Interchange.)

DOS-based PCs used 8-bit numbers to represent characters, allowing for 256 different characters . Since early PCs had no graphics capabilities, the PC's designers used the newly added character positions for graphics and line-drawing characters.

The Mac came along a few years later. It also used an 8-bit character set, but since it was more than adequately supplied with graphics capabilities, there was no need to waste limited character slots on graphics-drawing characters. Its designers were free to stock the upper 128 (high-ascii) characters with extra punctuation marks and nearly any accented/special characters needed for roman alphabet based languages.

## **Then came Windows.**

I'll leave it to Mr. Gates to explain why he decided to toss out a perfectly workable Mac character-encoding wheel and invent a new (incompatible) one. Windows uses a character set similar to the Mac's but encodes it almost entirely differently once we leave the familiar ASCII characters.

Both systems use accented vowels, for instance, but they're represented by different numbers. In fact, only five high-ascii characters share the same encoding between platforms.

The diagram on the next page shows one example of the chaos that can result. I used Corel Draw to create a drawing that includes all the high-ascii characters in Windows encoding, saved it as an Illustrator file, then opened it in Illustrator on the Mac.

The results aren't the fault of bad programming on anybody's part. Draw (correctly) asks for character number 0174, and expects that the system it's running on will supply a registered trademark symbol. Illustrator on the Mac hands the same 0174 character to the Mac system and gets back the character you start that famous Fable guy's name with.

The fault, dear Brutus, is in the system. Or the difference between the two systems.

## Here's what we end up with

As created in Draw

	0	1	2	3	4	5	6	7	8	9
012										
013	,	f	„	...	†	‡	^	%	Š	‹
014	Œ				‘	’	“	”	•	
015	—	~	™	š	›	œ			ÿ	
016		ı	ç	£	¤	¥	¦	§	¨	©
017	ª	«	¬	–	®	¯	°	±	²	³
018	´	µ	¶	·	¸	¹	º	»	¼	½
019	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç
020	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ
021	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û
022	Ü	Ý	Þ	ß	à	á	â	ã	ä	å
023	æ	ç	è	é	ê	ë	ì	í	î	ï
024	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù
025	ú	û	ü	ý	þ	ÿ				



Different, Mac vs. PC



Not used on Mac



Not used in Windows



Same on Mac and PC

As imported by Mac Illustrator

	0	1	2	3	4	5	6	7	8	9
012									À	Á
013	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã
014	å	ç	é	è	ê	ë	í	ì	î	ï
015	ñ	ó	ò	ô	ö	õ	ú	ù	û	ü
016		°	ç	£	§	•	¶	β	®	©
017	™	´	¨	≠	Æ	∅	∞	±	≤	≥
018	¥	μ	∂	Σ	Π	π	∫	ª	º	Ω
019	æ	ø	¿	ı	¬	√	f	≈	Δ	«
020	»	...		À	Ã	Õ	Œ	œ	–	—
021	“	”	‘	’	÷	◊	ÿ	ÿ	/	¤
022	‹	›	fi	fl	‡	·	¸	„	%	Â
023	Ê	Á	Ë	È	Í	Î	Ì	Ó	Ô	
024		Ò	Ú	Û	Ù	ı	^	~	-	˘
025	·	°	¸	˘	˙	˚				

## Here's why it happens

The illustration gives you some idea what can happen to high-ascii characters when you move them from PC to Mac. As you can see, the only characters that don't get dinged are the five that share the same encoding (that is, are represented by the same number) on both platforms (cent, pound, copyright, plus/minus and micron, by name)

To summarize what we've seen so far:

- The basic alphabetic, numerical and punctuation characters are encoded identically on both platforms, so no cross-platform translation is needed, no cross-platform errors will occur.
- Some characters are used on both platforms but encoded differently. A decimal 0201 on the PC is E-with-acute-accent, but becomes an ellipsis on the Mac. Most modern cross-platform programs try to translate the one character encoding set into the other, but there's a limit to how completely they can do this because ...
- Some characters are part of the Windows character set but are not encoded at all ... that is, they're simply not available ... on the Mac. The broken bar (aka vertical bar or pipe) symbol, some of the fraction characters and a few others, for example.
- Likewise, some Mac characters aren't part of the Windows character set.
- There are also characters on the Mac that, under certain circumstances, get pulled from a completely different font (some of the mathematical symbols). This makes my head hurt, so I'll ignore it from now on.

It's the characters that are available on one platform and not on another that cause most of the ruckus. How are you supposed to translate the broken bar symbol from a Windows document and display it on a Mac when that character doesn't exist on the Mac?

Beats me, but those clever folks at Adobe have it sorted out and working very nicely in the PDF format and the companion Reader software for both PC and Mac. Left to its own devices, Adobe makes it work.

And it's at this point that I first thank you for your patience in bearing with me so far and turn (finally!) to the meat of the matter: Why doesn't it always work?

## It's supposed to work but ...



Sometimes when you open a PDF file on the Mac you'll see a message to the effect that the font could not be re-encoded and that some characters could not be displayed. And in fact, some characters don't appear correctly on screen in Reader or Exchange.

Choose Document Info, Fonts from the File menu and you'll see that the original file calls for e.g. Times-Roman Type 1 but Reader is using Times-Roman TrueType. This can happen even when you've specified that all fonts be embedded when you created the PDF.

Remember I qualified Adobe's perfection with "left to its own devices" a few sentences back? Sometimes a series of little glitches tosses sand in the Acrobat watchworks. For starters, the following fonts are never embedded, no matter what options you choose in PDFWriter/Distiller:

- ❑ Courier (regular, bold, italic and bold-italic)
- ❑ Helvetica (regular, bold, italic and bold-italic)
- ❑ Times-Roman (regular, bold, italic and bold-italic)
- ❑ Symbol
- ❑ ZapfDingbats.

This is by design. If these fonts aren't already on your system, the Reader or Exchange installer installs them. Since this base set of fonts will (in theory) always be present, there's no need to embed them in PDFs, and leaving them out makes the PDF files smaller.

However, the Acrobat installer evidently doesn't distinguish between Type 1 and TrueType versions of fonts. If the Helvetica on the Mac happens to be TrueType, the installer doesn't install the Type 1 version and there you are.

## **There you are WHERE?**

In Funny Character City is where.

The Type 1 versions of the fonts contain all the characters needed to display anything you can enter on a PC or a Mac. Though the computer can only cope with 255 of them at a time, the font itself can contain many more glyphs (character descriptions).

Acrobat's clever enough to re-encode and display the extra glyphs in Type 1 fonts when a PDF needs them, so that even though there's no broken bar or S-with-caron on the Mac, Acrobat can display them if the PDF requires it.

Neat trick. But to pull it off, Acrobat has to have access to a Type 1 version of the font.

Perhaps the TrueType versions of the base fonts lack these extra glyphs or perhaps TrueType fonts can't be re-encoded/remapped at will by Acrobat, but in any case, if the PDF needs one of these special characters and Acrobat finds itself having to use TrueType rather than the Type 1 font it expects, you'll see the "can't re-encode" message and there will be character substitutions.

## **OK. Now I know where I am. Where's the exit, please?**

To cure the problem on the Mac, check your Fonts folder. Double click each of the Courier, Helvetica, Times-Roman, Symbol and ZapfDingbats suitcases to see if there's a TrueType font lurking within. I'm told by my betters that these fonts are included with the OS installation on newer Macs. Here's how you can tell:

- Bitmap (and for our purposes, innocuous) screen fonts in the suitcase have numbers next to their names indicating point size. TrueType fonts don't.
- The usual bitmap screen fonts are around 7k or so in size. TrueType fonts are in the 60k size range.
- The icons for bitmap screen fonts are a single "A". TrueType icons are a "zoomed" series of three "A"s.

Remove the TrueType versions of these fonts. You probably know a lot more about your Mac than I know about mine, so handle this however you think best. Just be sure to lose the TrueType versions of the base Acrobat fonts listed earlier. Personally, I'd lose the TrueType versions of all the base PostScript fonts and any other Type 1 font I plan to use, but that's your call. In any case, it's always Bad Medicine to have both TrueType and Type 1 versions of the same font installed.

Then reinstall Reader to the same folder it already lives in. No need to uninstall it first. Since the TrueType fonts are gone, it'll install Type 1 fonts if necessary.

Now look at this PDF again. It uses Helvetica and Times-Roman both, is that Rindsberg a sneaky devil or what? If you get a "can't re-encode" message, something's not right. If File, Document Info, Fonts shows that Reader is using a TrueType font or AdobeSanMM/AdobeSerifMM, ditto.

## **You use a PC but want to spare Mac users this abuse?**

Good on you! May you be amply rewarded in the next life or sooner for your consideration and sterling moral qualities. Here's how to make life easier for Mac users who read your PDFs:

The coward's way out: Don't use any funny characters or at least stick to the ones that are used on both PC and Mac. Yeah. I know. "Not practical, Steve."

A little trickier, but very effective: If you can live with slightly larger PDFs and aren't constrained by design guidelines, use a font that isn't part of the standard 35 base fonts and embed the font in the PDF. Acrobat will use the embedded font on the Mac and there should be no problems. Embedding the font shouldn't add more than the size of the font itself to the resulting PDF, and will probably add quite a bit less, especially if you use subsetting.

If you insist on making life difficult for yourself (by using TrueType fonts): You'll have to do some experimenting. I did my tests with Corel Draw, which allows me optionally to convert TrueType fonts to Type 1 at print time. Which is exactly what I did when testing with TrueType fonts. These opened in Reader on the Mac without incident, and a peek at the Document Fonts info confirmed that Reader was using the embedded subset of the font from the PDF in Type 1 format. Because the font is subsetted, I probably wouldn't be able to do any text editing in Exchange on the Mac (or on another PC). That suits my immediate needs, so I didn't pursue it further. You may want to achieve different results with different originating applications. If so, a little testing is called for. You didn't have anything better to do this weekend anyhow, did you?



## About the included files

The original for all this started in CorelDraw8. The drawing contains all the high-ascii (> 127 decimal) characters in Helvetica. All the characters have been duplicated, converted to curves, given a red fill and sent to the back of the drawing. In other words, each character in fonted text sits atop a graphics-only version of the same character. You can easily see which characters have been remapped or changed when the file is imported into a different app, since the red/graphic-only "character" never changes, even though the fonted text may.

The truly eagle-eyed among you will note that I've goofed in at least one place: the OE character in caps appears twice. As noted on the drawing, some apps "helpfully" capitalize letters at the beginning of sentences for you. I thought I'd thrashed this foolishness out of Draw and corrected the resulting screwups, but I missed this one. You'll have to cope as best you can. Corrected in the other PDFs.

For those of you who worry about these things: I did the original drawing and the exports under NT4/SP3 and used the MS version of the PostScript driver and the Distiller Assistant PPD to make all of these PDFs.

I've exported it in several formats, using the options shown in the table on the following page:

File	Created by
Charmap.EPS	Exported as EPS, text as text (i.e. NOT converted to curves), fonts not included, 72dpi 1-bit TIFF preview
CharmapP.AI	Exported as Adobe Illustrator/AI 3.x format, text as text, For PC
CharmapM.AI	Exported as Adobe Illustrator/AI 3.x format, text as text, For Mac
1.PDF - 8.PDF	<p>Later in this document there's a spreadsheet that summarizes what these are about, the options I chose at each step in producing them, and the results when viewed on PC and Mac.</p> <p>In all cases, I printed to the Adobe Distiller Assistant driver then distilled to PDF in Distiller v 3.02, running NT4/SP3 and using the Microsoft-supplied PostScript printer driver.</p>

The For PC or For Mac option in Draw8 affects character mapping, so the two AI files are likely to produce different results, depending on the app and platform you import them into. As nearly as I can determine, the For Mac option maps the PC encodings for characters to their Mac equivalents if an equivalent exists on the Mac. Otherwise it substitutes nulls or leaves out the character.

Other AI-producing apps don't offer this PC vs Mac encoding option. Most likely they'll map characters as appropriate to the platform they're running on. Another exercise for the reader. You weren't busy next weekend either, were you?

Since I told Draw NOT to download fonts when creating 1.PDF, the file ends up being rather large, since all fonted text is converted to graphics. I didn't include it for that reason, and because with no fonts in it, it isn't particularly relevant in any case. (Translation: I was too lazy to rename all the files and re-write all the documentation)

For PDFs 2-8, I created PDFs from the same high-ascii character set. I used a variety of fonts, including Helvetica and Times-Roman (both Type 1 and printer resident as well as being treated specially by Distiller), Barmeno-Regular (Type 1, non-printer resident), ErasMediumITC (TrueType), Veljovik (Type 1, non-printer-resident) and Palatino (Type 1, resident on all but very old PS printers).

An interesting note: later testing suggests that you can get very good cross-platform file exchanges by creating PDF on the originating side, taking the PDF to the opposite side of the Great Platform Gulch, opening it in Exchange and exporting to EPS with fonts included.

## **What do I do with this embarrassment of riches?**

Try opening the various PDF files on your own computer to see whether there are any error messages and to verify (via Files, Document Info, Fonts) what fonts are actually used on your system. Here, so long as the caveats above are observed, they all open correctly on my Mac.

As to the EPS/AI versions of CharMAP, it's instructive if sometimes dismaying to open, import, or view these in different apps on different platforms. If nothing else, it proves that cross-platform compatibility is still a dream yet to be realized, but the PDFs show that it's getting close. Very close.

It can also show that even if a particular app happens to be running under Windows, it may still reveal its MacHeritage. The Windows version of Photoshop 4, for example, doesn't know what to do with characters that don't have a Mac equivalent. It rasterizes them incorrectly. The rasterizer that generates previews from EPS and AI images in Windows PageMaker has similar difficulties. Neither handles high-ascii characters correctly in all cases, as compared to a printout of the same file on a true Adobe PS printer or a very good clone PS interpreter.

You can use these files to help you get a sense of what happens when files move from app to app, or PC to Mac. You can learn from them which characters will get munged along the way.

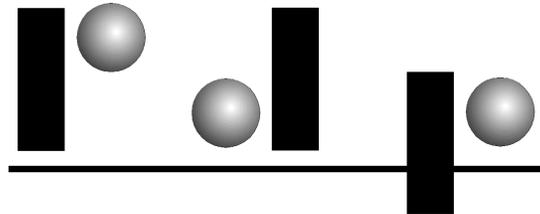
If you have questions or comments about any of this, please step forward and speak up. You can usually locate me on Compuserve's DTPFORUM or ADOBEAPPS forums. I check the comp.text.pdf newsgroup fairly frequently as well.

Do me, yourself and everybody else a favor, though. Please don't email me privately with requests for technical support. We all benefit from sharing our knowledge and experiences on a public forum, but we make me cranky by trying to do it privately. We may not get a response if we try. ;-)

Enjoy!

## Who is this "Rindsberg"?

Steve Rindsberg is President of RDP.  
Also CEO, CFO, COO, LOAIS, SPQR, LSMFT  
and any other titles that passing gusts of whimsy  
leave wrapped around his ankles



<http://www.rdpslides.com>