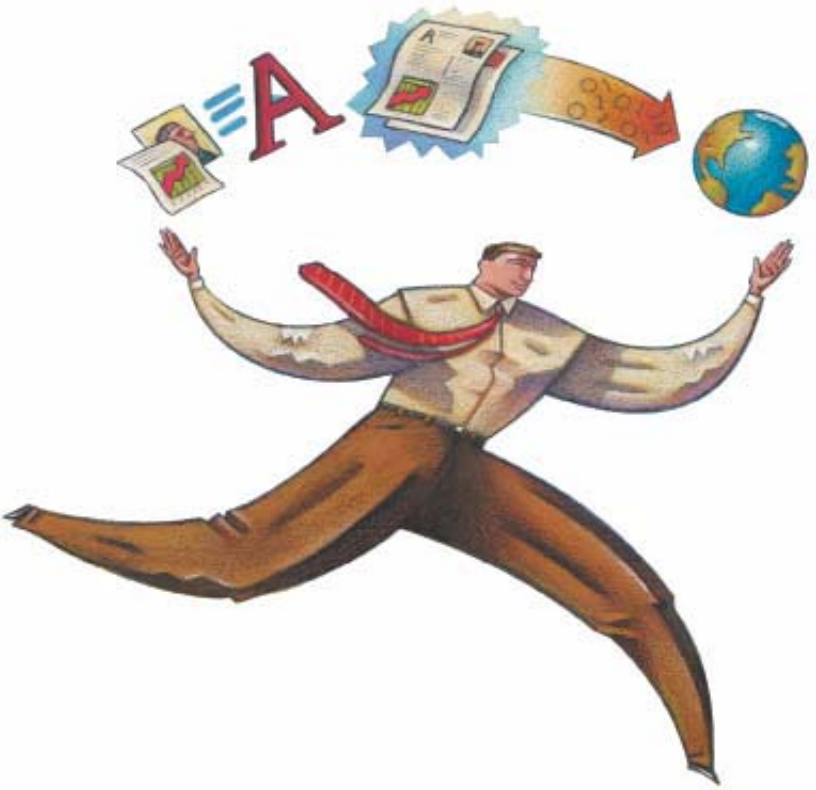
Acrobat Widths-Only CIDFont Tutorial

Technical Note #5412



ADOBE SYSTEMS INCORPORATED
Corporate Headquarters
345 Park Avenue
San Jose, CA 95110-2704
(408) 536-6000
http://partners.adobe.com

April 12, 2001

Contents

Contents

1 Acrobat Widths-Only CIDFont Tutorial

1.1 Introduction

Widths-only CIDFont files are almost identical to genuine CIDFont files, except that they have a special */WidthsOnly keyword* in the header, and provide only advance widths for each CID. Hence, they are referred to as widths-only CIDFonts.

Widths-only CIDFonts are used by Acrobat Distiller, Version 4.0 and later, when it encounters a PostScript file that contains a reference to a font whose outlines are not otherwise available. Widths-only CIDFonts allow Acrobat Distiller to create a metrically-correct PDF files.

Detailed information on how to build genuine CIDFont files can be found in Adobe Technical Note #5014, “Adobe CMap and CID Font Files Specification Version 1.0,” which is available from the ASN (Adobe Solutions Network) Developer Program at the following URL:

http://partners.adobe.com/asn/developer/technotes/main.html

1.2 Differences Between CIDFonts & Widths-Only CIDFonts

Widths-only CIDFonts and CIDFonts differ in a small number of ways, enumerated as follows:

- There is a new keyword in the CIDFont header, which is set to *true*:

  ```
  /WidthsOnly true def
  ```

  This keyword, when set to *true*, tells Acrobat Distiller that this is a widths-only CIDFont, and needs to be treated accordingly.

- The **/FontBBox** array in the CIDFont header should be set the same as the genuine CIDFont. For the widths-only KozMinPro-Light CIDFont, it is set as follows:

  ```
  /FontBBox [ -191 -272 1104 1071 ] def
  ```

  This information is used by Distiller, and helps a widths-only CIDFont behave much like the genuine CIDFont.

- The charstrings for each CID shall look like those in the following three examples, shown in human-readable form:

```
0 1000 hsbw 0 callsubr endchar
0 280 hsbw 0 callsubr endchar
0 1000 hsbw 0 callsubr endchar
```

  Note how the first and third charstrings are identical. These example charstrings are taken from CIDs 0, 1, and 1200 of the CIDFont KozMinPro-Light. CID=0 has a full width (1,000 units), CID=1 has a proportional width (280 units), and CID=1200 has a full width (1,000 units). Compare the above with the charstrings in the genuine CIDFont:

```
100 1000 hsbw 0 50 vstem 750 50 vstem -120 50 hstem 830 50 hstem
  -120 vmoveto 800 hlineto 1000 vlineto -800 hlineto closepath 400
  -459 rmoveto -318 409 rlineto 636 hlineto closepath -286 -450
  rmoveto 318 409 rlineto -818 vlineto closepath -668 -41 rmoveto
  318 409 rlineto 318 -409 rlineto closepath -668 859 rmoveto 318
  -409 rlineto -318 -409 rlineto closepath endchar
0 280 hsbw endchar
42 1000 hsbw 0 917 vstem 378 29 hstem 8 378 rmoveto 854 hlineto 30 0
  15 6 10 20 rrcurveto -112 78 rlineto -59 -75 rlineto -746 hlineto
  closepath endchar
```

  Specifically, the hints and contour descriptions are replaced by the same subroutine call (“0 callsubr”), but the width information remains. Note how the value immediately before the specified width, called the left side bearing value, is changed to 0 (zero) for all charstrings.

- Each hint dictionary shall contain exactly one subroutine, defined as follows:

```
0 -120 rmoveto 1000 hlineto 1000 vlineto -1000 hlineto closepath
return
```

  The `"0 callsubr"` in each charstring references the above subroutine definition.

- The */lenIV* in each hint dictionary shall be set to a value of –1, meaning that all charstrings are clear, thus lacking Type 1 encryption.

NOTE: It is not absolutely necessary to subroutinize the charstrings as described above, but it is recommended for space savings. The basic requirement is that the charstrings must encode the same set width as the corresponding charstring in the genuine CIDFont, and must contain a path that draws a solid em-box relative to the origin.

Lastly, a widths-only CIDFont is named exactly the same as its genuine counterpart.

1.3 Installing & Using Widths-Only CIDFonts

Widths-only CIDFonts must be installed into Acrobat Distiller's private CIDFont directory in order to be recognized and used. That directory, for both Macintosh and Windows, is located as follows:

Mac OS: Distiller:Data:psdisk:Resource:CIDFont:

Windows: Distillr\Data\psdisk\Resource\CIDFont\

When widths-only CIDFonts are used, PDF files with font references are created. It's not possible to embed widths-only CIDFonts, because the only meaningful information that they contain are advance widths.

Bear in mind that there are limitations in how Acrobat substitutes CJKV fonts. If the original font is not available when the PDF file is printed or displayed, then a substitution font is selected that may or may not match the original font in weight or style. Substitution of glyphs that use proportional widths is only supported for proportional Latin glyphs (that is, kana glyphs that use proportional widths will not be displayed or printed correctly when a substitution font is used).

When thumbnails are automatically created by Distiller, the use of widths-only CIDFonts that conform to these specifications will result in boxes appearing in the thumbnails. If the thumbnails are generated (or regenerated) using the Acrobat Viewer, this problem does not occur, because Acrobat Viewer does not have access to the widths-only CIDFonts.

To ensure that a PDF file will be printed and displayed as the original document, the original font, containing all needed outline glyphs, must be embedded in the PDF file.