



補足情報

- [*1] の Ghostscript 実行においては、Ghostscript 9.15 以上では eps2write デバイスが、それ未満では epswrite が用いられる。
- [*1] の Ghostscript 実行における `-r` オプションの値は、画質優先モードでは 20016 固定、速度優先モードでは解像度レベル設定に従う。
- [*1] の Ghostscript 実行においては、epswrite / eps2write どちらのデバイスであっても、出力される EPS の BoundingBox 値が誤っている場合がある。そこで、eps(2)write で生成された EPS に対しては、Ghostscript の bbox デバイスで取得される BoundingBox 値によって常に上書きするようにしている。
- 「元のページサイズを維持」が ON の場合、[*1] の Ghostscript 実行においては、出力される EPS の BB 値を、変換前の PDF の指定された PageBox の値で上書きする。
- 「元のページサイズを維持」が ON の場合、「pdfTeX でクロップ」の過程では、変換前の PDF の指定された PageBox (余白付与の場合はそれに余白を加えたもの) を MediaBox にした PDF (左下が原点、他の Box は非明示) を生成する。
- [*2] の経路は Ghostscript 9.15 以上の環境で「テキスト形式 EPS 出力」を選んだときのみ実行される。Ghostscript 9.15 未満の場合は epswrite デバイスで得られた EPS をそのまま最終出力とする。
- Ghostscript の eps(2)write デバイスは、TikZ の shadows ライブラリを用いた場合など、特定の種類の図は苦手としている。Ghostscript 9.15 未満の epswrite デバイスの場合は、無数のパスに分割されてエラーを起こす原因となる。Ghostscript 9.15 以上の eps2write デバイスの場合は、ビットマップ化されてしまう。そのような図は、Ghostscript によって綺麗にアウトライン化することはできないので、Ghostscript を経由しない経路 (テキスト保持 PDF, SVG, 速度優先モードでのビットマップ画像) で代用できないか検討してほしい。
- [*3] の epstopdf 実行においては、OS X 10.11 El Capitan 以上では `--hires` オプションありで、それ未満ではなしで実行する。これは El Capitan で修正された Quartz API の不具合に対応するための措置である。OS X 10.10 Yosemite 以下の OS では、`--hires` オプションありで生成された PDF を Quartz API にかけると端が欠けるという現象が発生する。
- 余白は原則として bp 単位が用いられるが、設定で px 単位を選んでいて、かつビットマップ画像出力の場合は、Quartz API によるビットマップ化実行時に px 単位の余白が付与される。