

pxjahyper パッケージ

八登崇之 (Takayuki YATO; aka. “ZR”)

v0.3a [2017/10/17]

1 概要

(u)p \LaTeX + hyperref + dvipdfmx の組み合わせで日本語を含む「しおり」をもつ PDF 文書を作成する場合に必要な機能を提供する。

- dvipdfmx 用の「tounicode special」について、内部漢字コードに応じて適切なものを出力する。
- PDF 文字列の中で LICR (`\a` や `\textsection` 等の文字出力の命令) が正しく機能するようにする。ただし、エンジンが p \TeX の場合は、out2uni を利用する場合を除き、JIS X 0208 にない文字は出力できない (hyperref の警告が出る)。
- \TeX の版面拡大機能が使われている (`\mag` が 1000 でない; 典型的には jsclasses の文書クラスで基底フォントサイズが 10 pt 以外である) 場合に、hyperref が出力する papersize special の紙面サイズの値が不正になるが、この場合に `\mag` 値を考慮して求めた正しいサイズによる papersize special を改めて出力する。

2 パッケージの読み込み

`\usepackage` で読み込む。

```
\usepackage[オプション,...]{pxjahyper}
```

使用可能なオプションは以下の通り。

- `tounicode` (既定) : dvipdfmx 用の「tounicode special」を出力する。
- `notounicode` : `tounicode` の否定。
- `out2uni` : out2uni フィルタ (角藤氏製作) を使うことを前提にした出力を行う。(tounicode が無効になる。) up \LaTeX では使用不可。
- `noout2uni` (既定) : `out2uni` の否定。
- `bigcode` (既定) : up \TeX での ToUnicode CMap として既定の UTF8-UCS の代わりに UTF8-UTF16 を用いる。(当該のファイルが存在する必要がある。)
- `nobigcode` : `bigcode` の否定。
※ 0.3a 版より既定を `bigcode` に変更した。
- `dvipdfmx` : dvipdfmx を前提とした動作を行う。

- `nodriver/none` : `dvipdfmx` を前提とした動作を抑止する。現状では、この場合には本パッケージは実質的に何の動作も行わない。
- `auto` (既定) : `hyperref` のドライバが `dvipdfmx` 用ならば `dvipdfmx`、それ以外は `none` の動作。

3 機能

「概要」で述べた機能は（オプション設定に応じて）自動的に実施される。

■Unicode 符号値による入力 PDF 文字列入力中で、`\Ux` が以下の意味になる。PDF 文字列以外では `\Ux` は以前の定義（または未定義）に戻る。

- `\Ux{Unicode 符号値}` : その符号値の文字を出力する。

符号値は 16 進数で指定する。

なお、OTF パッケージの `\UTF` 命令は、PDF 文字列中では `out2uni` 用の出力を行うように設計されているが、本パッケージを `upLaTeX` で用いた場合は、`\UTF` も（PDF 文字列中では）`\Ux` と同じ動作（つまり `tounicode` 用の出力）になるように変更される。

■PDF 文字列用の文字命令の定義 以下の命令が提供される。（プリアンブルでのみ使用可能。）

- `\pxDeclarePdfTextCommand{\制御綴}{<JIS 符号値>}{<Unicode 符号値>}` : PDF 文字列中の `\制御綴` の動作として、指定した符号値の文字を出力することを指定する。
- `\pxDeclarePdfTextComposite{\制御綴}{<引数>}{<JIS 符号値>}{<Unicode 符号値>}` : PDF 文字列中の `\制御綴`（アクセント命令）+ `<引数>` の動作として、指定した符号値の文字を出力することを指定する。

これらの命令において、符号値は 16 進数で指定する。「JIS 符号値」は `upLaTeX` では使われないので省略して（空にして）もよい（或いはそもそも JIS X 0208 にない文字の場合は省略する）。逆に「Unicode 符号値」は `pLaTeX` の動作でかつ「JIS 符号値」が指定されている場合は省略してよい。

例えば、以下のように定義しておくと、PDF 文字列中で `\textschwa`（schwa 記号）や `\d{t}`（`t̥`）が使えるようになる。

```
\pxDeclarePdfTextCommand{\textschwa}{}{0259}
\pxDeclarePdfTextComposite{\d}{t}{}{1E6D}
```