

pxjahyper パッケージ

八登崇之 (Takayuki YATO; aka. “ZR”)

v0.7 [2021/02/13]

1 概要

(u)p^LT_EX + 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_TE_X 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)。
- T_EX 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^LT_EX では使用不可。
- `noout2uni` (既定): `out2uni` の否定。
- `otfcid` (既定): `japanese-otf` パッケージの `\CID` を PDF 文字列中で使えるようにする。具体的には、当該の AJ1 のグリフに“対応”する Unicode 文字があればそれを出力し、なければ (警告を出した上で) 削除する。
- `nootfcid`: `otfcid` の否定。
※ `otfcid` の利用には、エンジンの ϵ -T_EX 拡張および etoolbox と bxjatoucs パッケージインストール

が必要。

- `disablecmds` (既定): 「PDF 文字列中のテキスト装飾命令の無効化」を有効にする。
※詳細は 3.3 節を参照。
- `nodisablecmds`: `disablecmds` の否定。
- `otfmacros` (既定): `japanese-otf` 付属の `ajmacros` パッケージが提供する文字入力命令 (`\ajMaru`、`\ajLig` 等) を PDF 文字列中で “可能な限り” 使えるようにする。
※ Unicode 文字で表現可能であればそれを出力し、なければ代替表現を出力する。
※ `otfmacros` を指定する場合は `otfcid` も有効にする必要がある。
※ `ajmacros` パッケージの多くの命令は “脆弱 (fragile)” である。そのため、節見出し (`\section` 等の引数) で `\ajMaru` 等の命令を使いたい場合は、命令の前に `\protect` 付ける必要がある。^{*1}
- `nootfmacros`: `otfmacros` の否定。
※ 0.6 版より既定を `otfmacros` に変更した。
- `bigcode` (既定): `upTeX` 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。
- `nodvidriver`: `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。
※ 0.5 版より、`nodvidriver` の別名の `none` は非推奨の扱いとする。
- `auto` (既定): `hyperref` のドライバが `dvipdfmx` 用ならば `dvipdfmx`、それ以外は `nodvidriver` 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。

3.1 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。^{*2}

- `\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、`japanese-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

^{*1} ちなみに、引数が PDF 文字列として解釈される場合には、`\protect` は全く結果に影響しない。

^{*2} `\Ux` という命令名は `bxbase` パッケージの Unicode 符号値入力用の命令が使っているものである。従って、`bxbase` パッケージを読み込んでいれば、「PDF 文字列と版面出力の両方に使われる」ようなテキストにおいて、`\Ux` で Unicode 符号値入力が可能になる。ただし、Unicode 符号値入力用の命令としては「`japanese-otf` パッケージの `\UTF` 命令」の方が有名であり、`pxjahyper` は `\UTF` も正しく扱えるので、こちらを使う方が無難かもしれない。

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。

3.2 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、PDF 文字列中で `\textschwa`（schwa 記号）や `\d{t}`（`t`）が使えるようになる。

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

3.3 PDF 文字列用中のテキスト装飾命令の無効化

PDF 文字列は単なる Unicode 文字列として扱われるものなので、`\textit` や `\large` 等のテキスト装飾用の命令は意味をなさず、またそれらの命令の実装は PDF 文字列の解釈中は正常に処理できない。PDF 文字列と版面出力の両方に使われるテキスト（節見出し等）についてテキスト装飾命令が支障なく使えるように、`hyperref` では基本的なテキスト装飾命令（多くは `LaTeX` カーネルが提供するもの）について、「PDF 文字列として扱う場合は自動的に無力化^{*3}する」機構を実装している。これにより、例えば節見出しのテキストに“`\textit{text}`”が含まれていたとすると、版面に出力する場合には“*text*”のように装飾が施され、一方で、PDF 文字列としては“`text`”と解釈されることになる。

0.5 版以降の `pxjahyper` では、この無効化の対象に「和文用のテキスト装飾命令（およびそれに準じるもの）」を追加するようになった。以下の命令が対象になる。

- 和文のフォント選択命令： `\textmc \gtfamily \kanjifamily \useroman \userelfont` など
- 次の `pLaTeX` カーネル命令： `\<`
- 次の `pTeX` プリミティブ： `\inhibitglue \noautospace \noautoxspace`
- 次の `plext` の命令： `\bou \kasen \rensuji`
- 次の `japanese-otf` の命令： `\textmg \mgfamily \ltseries \ebseries \propshape`

^{*3} 例えば、“`\textit{text}`”や“`\large text`”は単に“`text`”と書いたものと見なされる。