

pxjahyper パッケージ

八登崇之 (Takayuki YATO; aka. “ZR”)

v1.2 [2022/05/27]

目次

1	概要	1
2	パッケージの読込	2
3	機能	3
3.1	設定	3
3.2	Unicode 符号値による入力	3
3.3	japanese-otf パッケージの文字入力命令への対応	4
3.4	PDF 文字列用の文字命令のユーザ定義	5
3.5	PDF 文字列用中のテキスト装飾命令の無効化	5
4	hyperref の “unicode モード” での動作	6

1 概要

(u)pL^AT_EX + hyperref + dvipdfmx の組み合わせで日本語を含む PDF 文字列 (“しおり” などの文書情報の文字列) 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 pT_EX の場合は、out2uni/convbkmk (および試験的機能の “force-unicode”) 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 パッケージの読込

`\usepackage` で読み込む。

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

■使用可能なオプション

- **設定キー記述**： `\pxjahypersetup` 命令の引数に書く設定記述をパッケージオプションに書くことができる。例えば
`\usepackage[fallback=delete]{pxjahyper}`
と指定すると、表現不能文字を代替するゲタ文字が出力されなくなる。
- **ドライバオプション**： `dvipdfmx`、`dvips`、および `nodvidriver`（ドライバ依存動作の無効化を明示的に指示、現状ではほとんどの動作が無効になる）が指定できる。
※ 1.0 版において `nodvidriver` の別名の `none` は廃止された。（0.5 版から非推奨だった。）
- **resetdvidriver**（既定）： ドライバオプションを `hyperref` のドライバ指定から自動的に判定する。
※ 1.1 版において名前を `auto` から `resetdvidriver` に変更した。旧名の `auto` は非推奨とする。
- **tounicode**（既定）： 以下の 2 つの設定を行う。
 1. ドライバが `dvipdfmx` である場合は「`tounicode special`」を発行する。
※この設定は `notounicode` オプションにより打ち消される。
 2. 「`tounicode special`」を前提にした出力を行う。
※この設定は `out2uni・convbkmk` オプションと排他である。
- **notounicode**： `dvipdfmx` 用の「`tounicode special`」を発行しない。
- **out2uni**： `out2uni` フィルタを使うことを前提にした出力を行う。
- **convbkmk**： `convbkmk` フィルタを使うことを前提にした出力を行う。
※ドライバが `dvips` の場合は既定で `convbkmk` が有効になる。これにより元の既定値 `tounicode` は（1 は効果がなく 2 は上書きされるため）実質的に無効化される。
- **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。
※詳細は 3.3 節を参照。
※ `otfmacros` を有効にする場合は `otfcid` も有効にする必要がある。
- **nootfmacros**： `otfmacros` の否定。
※ 0.6 版より既定を `otfmacros` に変更した。
- **papersize**（既定）： `papersize special` の補正を有効にする。
- **nopapersize**： `papersize` の否定。

■上級者向けオプション

- **otfutf**（既定）： `japanese-otf` パッケージの `\UTF` 命令を PDF 文字列中で使えるようにする。
※詳細は 3.3 節を参照。

- `nootfutf` : `otfutf` の否定。
- `otfcid` (既定) : `japanese-otf` パッケージの `\CID` 命令を PDF 文字列中で “可能な限り” 使えるようにする。
※エンジンの ε -TeX 拡張および `etoolbox` と `bxjatoucs` パッケージのインストールが必要。
※詳細は 3.3 節を参照。
- `nootfcid` : `otfcid` の否定。
- `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 3.5 節を参照。
- `nodisablecmds` : `disablecmds` の否定。
- `bigcode` (既定) : `upTeX` での ToUnicode CMap として UTF8-UTF16 を用いる。
- `nobigcode` : `bigcode` の否定。 `upTeX` での ToUnicode CMap として UTF8-UCS2 を用いる。
※ 0.3a 版より既定を `bigcode` に変更した。
- `force-unicode` : `hyperref` の `unicode` オプションが有効でかつエンジンが `pTeX` である状況 (0.9 版よりサポートが追加されたが試験的である) において出る警告を抑止する。

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 設定

パッケージの動作の設定を `\pxjahypersetup` 命令で行える。

```
\pxjahypersetup{<キー>=<値>,...}
```

有効な設定キーは以下の通り。

- `fallback=<値>` : PDF 文字列中で表現不可能な文字についての代替出力の方法を指定する。
 - `geta` (既定) : 表現不可能な文字をゲタ記号 (■) に置き換える。
 - `delete` : 表現不可能な文字を削除する。

3.2 Unicode 符号値による入力

PDF 文字列入力中で、命令 `\Ux` が以下の意味に変更される。PDF 文字列以外では `\Ux` は以前の定義 (または未定義) に戻る。^{*1}

- `\Ux{<Unicode 符号値 16 進>}` : その符号値の文字を出力する。具体的な動作は以下の通り :
 - `out2uni` または `convbkmk` が有効の場合は、エスケープ表記 (`\0xUUUU`) を出力する。

^{*1} `\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` もサポートするので、こちらを使う方が無難かもしれない。

- エンジンが upLaTeX の場合、あるいは“unicode モード” (4 節) が有効の場合は、当該の Unicode 文字自体を書いたのと同等になる。
- 上記以外で、 TeX Live 2022 以降の pLaTeX の場合^{*2}は、当該の Unicode 文字に対応する JIS 符号系の文字を書いたのと同等になる。JIS 符号系にない文字の場合は出力できないので警告を出した上でゲタ記号 (■) を出力^{*3}する。
- どの条件にも当てはまらない場合は、 $\backslash\text{Ux}$ は無効 (定義されない) になる。

3.3 japanese-otf パッケージの文字入力命令への対応

■ **$\backslash\text{UTF}$ 命令** `japanese-otf` パッケージの $\backslash\text{UTF}$ 命令は、PDF 文字列中では `out2uni` 用の出力を行う。本パッケージで `otfutf` オプションを有効にした場合は、PDF 文字列中の動作が以下のように変更される。

- $\backslash\text{Ux}$ 命令 (3.2 節参照) が有効の場合は、 $\backslash\text{Ux}$ と同じ動作^{*4}になる。
- それ以外の場合は、常に (警告を出した上で) ゲタ記号を出力する。

※ $\backslash\text{UTF}$ 命令の多言語版、すなわち $\backslash\text{UTFC}$ ・ $\backslash\text{UTFK}$ ・ $\backslash\text{UTFM}$ ・ $\backslash\text{UTFT}$ 命令も $\backslash\text{UTF}$ と同じ扱いになる。

■ **$\backslash\text{CID}$ 命令** `japanese-otf` パッケージの仕様では $\backslash\text{CID}$ 命令は、PDF 文字列中ではサポートされない (未定義動作となる)。本パッケージで `otfcid` オプションを有効にした場合は、PDF 文字列中で $\backslash\text{CID}$ が“可能な限り”使えるようにする。具体的な仕様は以下の通り。

- 当該の AJ1 のグリフに“対応”する**単独の** 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) ゲタ記号を出力する。^{*5}
 ※例えば、 $\backslash\text{CID}\{8226\}$ (ローマ数字 12) は Unicode 文字の U+217B に“対応”するので $\backslash\text{Ux}\{217B\}$ と同等になるが、 $\backslash\text{CID}\{8297\}$ (ローマ数字 15) については“対応”する単独の Unicode 文字がないので、ゲタ記号に置き換えられる。
- ただし $\backslash\text{Ux}$ 命令 (3.2 節参照) が無効になる場合は、そもそも Unicode 文字も出力できないため、常に (警告を出した上で) ゲタ記号を出力する。結局情報は欠落するが、それでも未定義動作 (エラーになりにえる) よりは好ましいであろう。

※ `otfcid` の利用には、エンジンの $\varepsilon\text{-TeX}$ 拡張および `etoolbox` と `bxjatoucs` パッケージが必要。

※ $\backslash\text{CID}$ 命令の多言語版はサポートされない。

■**`ajmacros` パッケージの命令** 本パッケージで `otfmacros` オプションを有効にした場合は、`japanese-otf` 付属の `ajmacros` パッケージが提供する文字入力命令 ($\backslash\text{ajMaru}$ 、 $\backslash\text{ajLig}$ 等) を PDF 文字列中で“可能な限り”使えるようにする。具体的な仕様は以下の通り。

- 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。
- Unicode 文字を出力する場合の仕様は $\backslash\text{CID}$ と同じ。 ($\backslash\text{Ux}$ が無効の場合はゲタ記号になる。) 代替表現

^{*2} 正確にいうと、 $\backslash\text{Uchar}$ と $\backslash\text{ucs}$ プリミティブをもつ $\varepsilon\text{-pTeX}$ エンジンである場合。

^{*3} 設定キー `fallback` の値が `delete` の場合は何も出力しない。以降で「ゲタ記号を出力」とある場合についても同様。

^{*4} もし $\backslash\text{Ux}$ の出力がゲタ記号になる場合は、 $\backslash\text{UTF}$ もゲタ記号になる。

^{*5} 旧版では削除していたが、他の同様の場合と合わせるため 1.0 版よりゲタ記号を出力する仕様を変更した。

の場合は「普通の文字の出力に置き換えられる」可能性がある。

※例えば、`\ajLig{ドル}`（“ドル”の組文字）は Unicode 文字の U+3326 に“対応”するので `\Ux{3326}` と同等になるが、`\ajLig{ウルシ}`（“ウルシ”の組文字）は Unicode に“対応”する文字がないため単に“ウルシ”と書いたのと同等になる。

※ `otfmacros` を有効にする場合は `otfcid` も有効にする必要がある。（従って `otfcid` と同じ前提条件が課される。）`otfcid` が無効な場合は `otfmacros` も無効になる。

※ `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。^{*6}

3.4 PDF 文字列用の文字命令のユーザ定義

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。Unicode 文字の出力が可能な状況（エンジンが `upLaTeX` の場合は常に該当する）では「JIS 符号値」は使われないので省略して（空にして）もよい^{*7}（或いはそもそも JIS X 0208 にない文字の場合は省略する）。

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}`（`t`）が使えるようになる。

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

3.5 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^{*8}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）」

^{*6} 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。

^{*7} 一応、「JIS 符号値」が使われることが確実な状況では「Unicode 符号値」も省略可能であるが、そのような状況であるかの判断は困難であるため、「Unicode 符号値」の省略は推奨されない。

^{*8} 例えば、“`\textit{text}`”や“`\large text`”は単に“`text`”と書いたものと見なされる。

を追加するようになった。以下の命令が対象になる。

- `hyperref` での無効化の対象である「フォント選択命令」の和文版に相当するもの。例えば、`\textmc\gtfamily\kanjifamily\romanshape\usekanji\useroman\userelfont` 等が該当する。
- `pLATEX` カーネル命令： `\<`
- `pTEX` プリミティブ： `\(dis)inhibitglue\\(no)autospadding\\(no)autoxspacing`
- `plext` の命令： `\bou\kasen\rensuji`
- `japanese-otf` の命令： `\textmg\mgfamily\ltseries\ebseries\propshape`
- `jsclasses` のクラスの命令： `\maybeblue\HUGE`
- `jlreq` クラスの命令： `\jafontsize\tatechuyoko\jidori`

4 `hyperref` の“unicode モード”での動作

`hyperref` パッケージの `unicode` オプションが有効である場合（これを“unicode モード”と呼ぶことにする^{*9}）で動作している場合は、PDF 文字列の Unicode への変換は（DVI ドライバ等でなく）`hyperref` 自身により行われる。`hyperref` が“unicode モード”である場合には、`pxjahyper` はそれを自動的に検知してそれに適応した動作に切り替える。

- `\Ux` 命令は `hyperref` の `\unichar` 命令を利用して出力する。このため、`pLATEX` でも Unicode 文字の出力が可能になる。
- PDF 文字列中の和文文字や `LICR` 命令の処理は `hyperref` の側に任せられる。
※ただし現状では「`tounicode special`」の発行は（特に害はないため）無効化されない。

ただし、現状での“unicode モード”対応動作には以下の制限がある。

- `hyperref` の（`pdftitle` 等の）パッケージオプション中での和文文字の処理は失敗する。このため、文書情報は `\hypersetup` 命令で指定する必要がある。
- `pLATEX` における“unicode モード”対応動作は試験的機能の扱いであるため、その旨の警告が表示される。この警告は `force-unicode` オプションを指定すれば抑止できる。

^{*9} “unicode モード”を有効にする方法は `unicode(=true)` の指定以外にも存在する。また、`hyperref` の 7.00g 版 [2021-02-04] より、`pLATEX` 以外のエンジン（`upLATEX` も含む）について“unicode モード”は既定で有効になっていることに注意。