

pxchfon パッケージ

八登崇之 (Takayuki YATO; aka. “ZR”)

v1.7 [2019/11/19]

目次

1	概要	2
2	前提環境	2
3	読込	2
4	機能	5
5	プリセット指定	7
5.1	単ウェイト用の設定	8
5.2	多ウェイト用の設定	8
5.3	ptex-fontmaps 互換のオプション	13
5.4	廃止されたオプション	13
6	ファイルプリセット機能	14
6.1	T _E X Live 用ファイルプリセット機能	14
6.2	単純ファイルプリセット機能	15
7	カスタムプリセット機能	15
8	Unicode 直接指定	16
8.1	Unicode 直接指定に関する注意事項	17
9	dvipdfmx のページ抜粋処理への対応	17
10	欧文フォントの置換の原理	18
11	注意事項	19
付録 A	dvipdfmx 以外の DVI ウェアでの使用	19
A.1	和文フォントだけを置き換えた場合 (noalphabet 指定時)	19
A.2	欧文フォントも置き換えた場合 (alphabet 指定時)	19

1 概要

pL^AT_EX / upL^AT_EX + dvipdfmx のワークフローで PDF 文書を作る場合に、標準の和文フォント（明朝・ゴシック）に対して実際に使われる OpenType フォントをユーザが指定する機能を提供する。使用するフォントを L^AT_EX 文書中で指定するので、一度パッケージをインストールするだけで、任意の日本語フォント（ただし等幅に限る）を使うことができる。欧文部分を同じ日本語フォントで置き換えることも可能である。japanese-otf パッケージ^{*1}にも対応している。

また、(u)pL^AT_EX において広く行われているフォント設定（IPAex フォントの使用等）をパッケージオプション一つで適用する機能（プリセット指定）も備えている。

2 前提環境

- T_EX フォーマット：L^AT_EX
- T_EX エンジン：pT_EX / upT_EX
- DVI ウェア：dvipdfmx
- 前提パッケージ：
 - atbegshi パッケージ（everypage オプション使用時）
 - pxufont パッケージ（unicode オプション使用時）

3 読込

プレアンブルにおいて、`\usepackage` を用いて読み込む。

```
\usepackage[<オプション>,...]{pxchfon}
```

オプションは次のものが用意されている。

- ドライバオプション：dvipdfmx、dvips、dviout、xdvi および nodvidriver^{*2}が指定できる。ただし、本パッケージの主要機能である「フォントマップの文書内での指定」がサポートされるのは dvipdfmx のみである。^{*3}他の DVI ウェアにおける動作については付録 A 節を参照。ドライバオプションの既定値は dvipdfmx である。
- プリセット指定オプション（ipaex、hiragino-pron 等）名前に対応するプリセット指定を有効にする。
※詳細については 5 節を参照。
- カスタムプリセット指定オプション（user:名前）ユーザ定義のプリセット指定を有効にする。

^{*1} japanese-otf パッケージ： <https://www.ctan.org/pkg/japanese-otf>

^{*2} ドライバ依存動作を明示的に無効化するための指定。

^{*3} 現状では、dvipdfmx 以外のドライバ指定は全て nodvidriver と等価である。

※詳細については 7 節を参照。

- ファイルプリセット指定オプション (`tl:名前` または `file:名前`^{*4}) 名前に対応するファイルプリセット指定を有効にする。

※詳細については 6 節を参照。

- **alphabet** : 欧文フォントも指定されたフォントの英数字部分で置き換える。(明朝が `\rmfamily`、ゴシックが `\sffamily` に適用される。)

※技術的制約のため^{*5}、半角等幅のフォントしかサポートされないことに注意してほしい。つまり、この設定を使うと欧文が全て等幅になってしまう。「部分的に欧文フォントを和文フォントに合わせたい」という場合は、後述の **relfont** オプションの使用も検討されたい。

※プリセット指定オプション不使用の場合はこちらが既定で有効になる。すなわち欧文フォントも置き換えられる。

- **noalphabet** : **alphabet** の否定。欧文フォントは変更しない。
※プリセット指定オプション使用の場合はこちらが既定で有効になる。
- **otf** (既定) : **japanese-otf** パッケージの使用時に、そのフォントも置き換えの対象とする。
- **nootf** : **otf** の否定。**japanese-otf** パッケージのフォントは置き換えない。
※この場合、**japanese-otf** パッケージで **noreplace** を指定しない限り、標準の和文フォントは変化しない。
- **prefer2004jis** : pTeX / upTeX の標準和文フォントの CMap を「2004JIS 字形」指定のものに変更する。^{*6}
- **noprefer2004jis** (既定) : **prefer2004jis** の否定。
- **(no)jis2004** : **(no)prefer2004jis** の別名。
※グローバルオプションに **jis2004** を指定して **japanese-otf** と **pxchfon** の両方に適用することを意図している。
- **oneweight** : **japanese-otf** パッケージを単ウェイトで使用する場合に、プリセット設定で使われるフォントの集合を pTeX 標準と同一にする。^{*7}
- **nooneweight** (既定) : **oneweight** の否定。

■上級者向けオプション

- **relfont** : 指定された和文フォントの英数字部分を (pLaTeX の標準機能である) 従属欧文フォントとして設定する。すなわち、既定では欧文フォントは置き換ええないが、`\selectfont` で和文フォントを

^{*4} 旧形式は `+名前` および `*名前`

^{*5} プロポーションル幅のフォントを使うためには、そのフォントに合わせた TeX 側の論理フォント (TFM) を事前に用意する必要があるため。和文が全角幅しか使えないのも同じ理由である。

^{*6} **japanese-otf** パッケージの和文フォントについては、**japanese-otf** パッケージの **jis2004** オプションで 2004JIS 字形指定を行う仕様になっている。そのため、本パッケージの **prefer2004jis** の対象にはならない。ところが 0.7h 版以前の本パッケージでは、upTeX 上の **japanese-otf** の和文フォントにも **prefer2004jis** を適用していた。これは、昔の **japanese-otf** が upTeX 上では **jis2004** オプションに未対応であったためであり、非公式の暫定仕様であった。現在では、この暫定仕様は廃止されている。

^{*7} 多ウェイト用プリセット設定の一部において、このオプションによりゴシックのウェイトが変更される。例えば、小塚フォントのプリセットでは、pTeX 標準のゴシック (**jisg** 等) には M ウェイトを割り当てて一方で、**japanese-otf** パッケージの 3 ウェイトのゴシックには R、B、H ウェイトを割り当てている。従って、**japanese-otf** を単ウェイトで用いる時に M ウェイトを使いたい場合には **oneweight** オプションを指定すればよい。

選択する際に予め `\userelfont` を実行しておくで欧文も和文と同じ書体になる。^{*8}

※置換が設定されていないウェイトについては無効になる。

※ `alphabet`、`noalphabet`、`relfont` の3つのオプションは排他である。

- `everypage` : 出力 DVI の全ページにフォントマップ設定を書き込む。
※詳細については 9 節を参照。
- `noeverypage` (既定) : `everypage` の否定。出力 DVI の先頭ページにのみフォントマップ設定を書き込む。
- Unicode 直接指定オプション : 一部または全部のフォントについて、エンコーディング指定方式を“CMap 指定”から“Unicode 直接指定”に変更する。`nounicode` (既定 : CMap 設定のまま)、`directunicode`、`directunicode*`、`unicode*`、`unicode` の5種類の指定がある。
※詳細については 8 節を参照。
※一部のプリセット指定 (`sourcehan` 等) は Unicode 直接指定の既定値を変更する。
- `usecmapforalphabet` : `alphabet` または `relfont` を指定して日本語フォントの英数字部分を欧文フォントとして使う際に、そのエンコーディング指定方式を“CMap 指定”にする。
※半角英数字用の `UniJIS-UCS2-HW-H` という CMap が指定される。
※ AJ1 のグリフ集合をもつ OpenType フォントの場合、既定では^{*9}英数字がプロポーショナル幅のものになるがこれはサポートされないで、このオプションを指定して半角幅のものを使う必要がある。
- `nousecmapforalphabet` (既定) : `usecmapforalphabet` の否定。
※和文と異なり、欧文フォントでは実質的に“Unicode 直接指定”の方が既定となっている。
※一部のプリセット指定は `usecmapforalphabet` の指定を強制する。
- `dumpmap` : 「通常マップファイルダンプ出力」を有効にする。すなわち、本パッケージにより文書に設定されるマップ行を、`<ジョブ名>.map` の名のファイルに書き出す。
- `nodumpmap` (既定) : `dumpmap` の否定。
- `dumpmaptl` : 「`TeX` Live マップファイルダンプ出力」を有効にする。すなわち、本パッケージによる設定を再現する `kanji-config-updmap` 用のマップファイルのセット (`ptex-NAME.map`、`otf-NAME.map`、`uptex-NAME.map`、`otf-up-NAME.map` の4つ、ただし `NAME` はジョブ名) を出力する。
※例えば、`japanese-otf` パッケージが使われない場合は `japanese-otf` パッケージ用のマップ行は適用されない。そのため、通常ダンプ出力はそのようなマップ行は書き出されない。これに対して、`TeX` Live 用ダンプ出力は「実際に適用されるか」は無関係で `kanji-config-updmap` の規則に従うため、`japanese-otf` パッケージ用のマップが `otf-*.map` に書き出される。
- `nodumpmaptl` (既定) : `dumpmaptl` の否定。
- `strictcsi` : `Identity-H/-V` の CMap が指定されたマップ行について、CSI 指定を (仕様に厳密に従って) フォントが TrueType グリフの場合にのみ出力する。
※「CSI 指定」とはフォントファイル名の直後に書く“/AJ1”の類のことで、本来は (グリフ集合情報を持たない) TrueType グリフのフォントのためのものである。しかし、CFF グリフのフォントに対して CSI 指定があっても特に問題は起こらず、また、フォントのグリフ種別の判断する処理は少し時間がかかるため、既定では厳密な判定は行わない。

^{*8} 特に「`\userelfont\selectfont`」だけ実行すると、欧文が現在の和文と同じ書体になる。

^{*9} `unicode` オプションが指定されない場合。

- `nostrictcsi` (既定): `strictcsi` の否定。Identity-H/-V に対する CSI 指定は常に出力される。
※さすがにファイルに出力されたマップ行に不備があるのは避けたいので、`dumpmap(tl)` が指定された場合は、既定が `strictcsi` に変更される。
- `expert` (既定): Unicode 直接指定の適用時^{*10}に `japanese-otf` の `expert` オプションの機能を (可能な範囲で) エミュレートする。
※ `japanese-otf` の `expert` が指定されない場合は無意味。
- `noexpert`: `expert` の否定。Unicode 直接指定時には `japanese-otf` の `expert` は無効になる。
- `glyphid`: GID 指定入力 (`\gid` 命令) の機能を有効にする。
※エンジンが `upTeX` でかつ Unicode 直接指定^{*11}が有効の場合にのみ利用できる。
- `noglyphid` (既定): `glyphid` の否定。

4 機能

以下に該当する和文 (CJK) 用の論理フォント (原メトリック TFM) について、それに対応する物理フォント (OpenType フォント) をユーザ指定のものに置き換える。

- `pTeX` の標準のフォント — `rml*/gbm*`
- `upTeX` の日本語フォント — `urml*/ugbm*/uprml*/upgbm*`
- `upTeX` の中国語・韓国語フォント
- UTF パッケージのフォント — `hmr*/hkb*/unij*/cid*`
- `japanese-otf` パッケージの日本語フォント — `{,up}hmin*/{,up}hgoth*/otf-{u,c}j*`
- `japanese-otf` パッケージの中国語・韓国語フォント
- `pxufont` パッケージのフォント — `zur-?j*`

※中国語・韓国語フォントに対するサポートの詳細については[付録 C](#)を参照。

和文フォント置換は、`dvipdfmx` のマップ設定を文書内で (一時的に) 変更するという方法で実現している。欧文フォントについては実現方法が少し異なる ([10 節](#)を参照)。

■単ウエイトの場合の設定 `japanese-otf` パッケージを `deluxe` オプション付きで用いている場合以外、すなわち明朝・ゴシックとも単ウエイトの場合、以下の命令を用いる。

- `\setminchofont[⟨番号⟩]{⟨フォントファイル名⟩}`: 明朝体 (`\mcfamily`) のフォントを置き換えるフォントをファイル名で指定する。TTC 形式の場合の該当のフォントの番号を `⟨番号⟩` に指定する。
- `\setgothicfont[⟨番号⟩]{⟨フォントファイル名⟩}`: ゴシック体 (`\gtfamily`^{*12}) のフォントを置き換えるフォントをファイル名で指定する。`⟨番号⟩` の意味は前項と同じ。
- 以上の 2 つの命令、および以降で紹介するフォント設定命令について、`⟨フォントファイル名⟩` の値を `*` にするとフォント非埋込を指示する。また、この値を空にすると、以前に (当該の命令により) 設定されていた値を取り消して (`dvipdfmx` の) 既定の設定に戻す。

^{*10} 置換されていないウエイトは Unicode 直接指定にならない。 ([8.1 節](#)の注意を参照。)

^{*11} 置換されていないウエイトは Unicode 直接指定にならない。 ([8.1 節](#)の注意を参照。)

^{*12} 単ウエイト設定を用いる多くの場合、明朝体の太字 (`\mcfamily\bfseries`) はゴシック体 (`\gtfamily` と同じもの) で代替される。

以下に設定例を示す。

```
\setminchofont{ipam.ttf}          % 明朝体は"IPA 明朝"
\setgothicfont[0]{msgothic.ttc} % ゴシック体は"MS ゴシック"
\setminchofont{*}                  % 明朝体は非埋込
\setgothicfont{}                   % ゴシック体は既定設定に従う
```

■多ウェイトの場合の設定 `japanese-otf` パッケージを `deluxe` 付きで用いている場合は、明朝・ゴシックともに 3 ウェイトを使う。この時は、各ウェイト毎にフォント指定ができる。またこの場合、丸ゴシック (`\mgfamily`) が使用可能になるが、これに対して置き換えるフォントを指定することができる。

- `\setlightminchofont[番号]{<フォントファイル名>}`: 明朝・細ウェイト (`\mcfamily\ltseries`)
- `\setmediumminchofont[番号]{<フォントファイル名>}`: 明朝・中ウェイト (`\mcfamily\mdseries`)
- `\setboldminchofont[番号]{<フォントファイル名>}`: 明朝・太ウェイト (`\mcfamily\bfseries`)
- `\setmediumgothicfont[番号]{<フォントファイル名>}`: ゴシック・中ウェイト (`\gtfamily\mdseries`)
- `\setboldgothicfont[番号]{<フォントファイル名>}`: ゴシック・太ウェイト (`\gtfamily\bfseries`)
- `\setxboldgothicfont[番号]{<フォントファイル名>}`: ゴシック・極太ウェイト (`\gtfamily\ebseries`)
- `\setmarugothicfont[番号]{<フォントファイル名>}`: 丸ゴシック (`\mgfamily`)

さらに、この場合、`\setminchofont` と `\setgothicfont` は各々のファミリの 3 ウェイト全てを指定のフォントで置き換える。実質的に単ウェイトになってしまうようで無意味に思えるが、例えば明朝を実際には 2 ウェイトしか使わないという時に、

```
\setminchofont{minchoW3.otf}      % まず 3 ウェイト指定して
\setboldminchofont{minchoW6.otf}  % 太だけ再指定する
```

とする使い方が考えられる。特に、欧文フォントも置き換えたい場合は 3 ウェイトが全て指定されていないと有効にならないので、

```
\setmediumminchofont{minchoW3.otf}
\setboldminchofont{minchoW6.otf}
```

では思い通りにならないことになる。また、この仕様のため、`deluxe` 以外の場合 (既定、`bold`、`noreplace`) は `\setgothicfont` で指定したものが確実にゴシック (単ウェイト) に反映される。

■上級者向け機能

- `\usecmapforalphabet`: `usecmapforalphabet` オプションの設定に切り替える。(3 節を参照。)
- `\nousecmapforalphabet`: `nousecmapforalphabet` オプションの設定に切り替える。
- `\setnewglyphcmapprefix{<文字列>}`: 2004JIS 用の JIS コード系の CMap の名前の接頭辞を指定する。そのような CMap は、`pTeX` の標準和文フォントについて 2004JIS 字形を選択 (`prefer2004jis` 指定) した時に必要となるが、Adobe が配布している CMap ファイルには該当するものがないので、それを適宜用意してそのファイル名をこの命令で指定する必要がある。引数に与えるのは最後の 1 文字 (書字方向の「H」「V」) を除いた部分の文字列である。
CMap 名接頭辞の既定値は「2004-」で、これは最近の `TEX Live` に含まれている「2004-H」等の

CMap ファイルを用いることを意味する。^{*13}

- `\usefontmapfile{<マップファイル名>}`: 指定の dvipdfmx 用のマップファイルの読込を指示する。pdfTeX の `\pdfmapfile` に相当する機能。
- `\usefontmapline{<マップ行>}`: dvipdfmx のマップ行を直接指定して、その読込を指示する。pdfTeX の `\pdfmapline` に相当する機能。

- `\diruni`: 現在の和文フォントを“Unicode 直接入力”（フォントマップを“Unicode 直接指定”にした上でさらに和文 VF をバイパスする）の状態に切り替える（宣言型命令）。これにより、実際のフォントがサポートする任意の文字が出力可能となる。その代わり、この状態では、約物の周りの空き調整が無効になる。

※“Unicode 直接指定”のオプションの何れか（`unicode` 等）が有効であり、かつ現在の和文ファミリーについてフォントの置き換えが有効になっている必要がある。

※全角幅のグリフでないと正常に出力されない。

※この命令自体は単にシェーブを `diruni` という値に変えているだけであり、このシェーブに“Unicode 直接入力”のフォントが予め設定されているわけである。

- `\textdiruni{<テキスト>}`: `\diruni` に対応する引数型命令。
- `\gid{<整数>}`: 現在の和文フォントで、指定の値の GID をもつグリフを出力する。

※全角幅のグリフでないと正常に出力されない。

※エンジンが upLaTeX であり、`unicode` オプションが指定されていて、かつ、現在の和文ファミリーについてフォントの置き換えが有効になっている必要がある。

5 プリセット指定

このパッケージの元々の意図は、標準のフォントを普段使っているものと全く別の書体に変えることであったが、例えば「普段使う設定が複数ありそれを簡単に切り替えたい」という場合にも有用である。そこで、pTeX において広く行われている設定をパッケージ内に組み込んで、パッケージオプションでそれ呼び出すという機能が後になって追加された。^{*14}

パッケージオプションにプリセット名を指定すると予め決められたフォントファイル名が `\setminchofont` 等の命令で設定される。例えば、

```
\usepackage[ipa]{pxchfon}
```

は以下の記述と同等になる。

```
\usepackage[noalphabet]{pxchfon}
\setminchofont{ipam.ttf}
\setgothicfont{ipag.ttf}
```

注意として、プリセット指定を用いた場合は、欧文フォントの置換について `noalphabet`（無効）が既定になる。プリセット指定の場合は和文が「普通の」明朝・ゴシックのフォントとなるので欧文フォントを変更しない場合が多いと考えられるためである。

^{*13} 引数に * を与えた場合は JISX0213-2004-H が指定されたと見なされる（歴史的理由から）。

^{*14} 元々は PXjafont という別のパッケージで提供されていた機能であるが、0.5 版からこのパッケージに組み入れることにした。

5.1 単ウェイト用の設定

後述の「多ウェイト用の設定」で述べられた設定以外で使う場合に使用する。

- **noembed** : フォントを埋め込まない。
 `\setminchofont{*} % 非埋込`
 `\setgothicfont{*} % 非埋込`
- **ms** : MS フォント。
 `\setminchofont[0]{msmincho.ttc} % MS 明朝`
 `\setgothicfont[0]{msgothic.ttc} % MS ゴシック`
- **ipa** : IPA フォント。
 `\setminchofont{ipam.ttf} % IPA 明朝`
 `\setgothicfont{ipag.ttf} % IPA ゴシック`
- **ipaex** : IPAex フォント。
 `\setminchofont{ipaexm.ttf} % IPAex 明朝`
 `\setgothicfont{ipaexg.ttf} % IPAex ゴシック`

5.2 多ウェイト用の設定

japanese-otf パッケージの **deluxe** オプション使用時に有効になる。明朝 3 ウェイト、ゴシック 3 ウェイト、丸ゴシック 1 ウェイトを設定する。

- **ms-hg** : MS フォント + HG フォント。
 ※ HG フォント = Microsoft Office 付属の日本語フォント
 ※ 「HG 丸ゴシック M-PRO」は欧文が等幅でないので **alphabet** オプション指定とともに使うことができない。(後掲の **ipa-hg**、**ipaex-hg** についても同様。)
 `\setminchofont[0]{msmincho.ttc} % MS 明朝`
 `\setboldminchofont[0]{hgrme.ttc} % HG 明朝 E`
 `\setgothicfont[0]{msgothic.ttc} % MS ゴシック`
 `\setmediumgothicfont[0]{hgrgm.ttc} % HG ゴシック M`
 `\setboldgothicfont[0]{hgrge.ttc} % HG ゴシック E`
 `\setxboldgothicfont[0]{hgrsgu.ttc} % HG 創英角ゴシック UB`
 `\setmarugothic{hgrsmp.ttf} % HG 丸ゴシック M-PRO`
- **ipa-hg** : IPA フォント + HG フォント。
 `\setminchofont{ipam.ttf} % IPA 明朝`
 `\setboldminchofont[0]{hgrme.ttc} % HG 明朝 E`
 `\setgothicfont{ipag.ttf} % IPA ゴシック`
 `\setmediumgothicfont[0]{hgrgm.ttc} % HG ゴシック M`
 `\setboldgothicfont[0]{hgrge.ttc} % HG ゴシック E`
 `\setxboldgothicfont[0]{hgrsgu.ttc} % HG 創英角ゴシック UB`
 `\setmarugothic{hgrsmp.ttf} % HG 丸ゴシック M-PRO`
- **ipaex-hg** : IPAex フォント + HG フォント。
 `\setminchofont{ipaexm.ttf} % IPAex 明朝`
 `\setboldminchofont[0]{hgrme.ttc} % HG 明朝 E`
 `\setgothicfont{ipaexg.ttf} % IPAex ゴシック`


```

\setmediumgothicfont[0]{hgrgm.ttc} % HG ゴシック M
\setboldgothicfont[0]{hgrge.ttc} % HG ゴシック E
\setxboldgothicfont[0]{hgrsgu.ttc} % HG 創英角ゴシック UB
\setmarugothic{hgrsmp.ttf} % HG 丸ゴシック M-PRO

```

- moga-mobo : Moga フォント + Mobo フォント。

※「丸ゴシック」ファミリに MoboGothic を充てている。

※ Moga/Mobo フォントは CID アクセス非対応であるが、フォント実体を変えることで prefer2004jis オプションに対応させている。

prefer2004jis 非指定時

```

\setminchofont[3]{mogam.ttc} % Moga90Mincho
\setboldminchofont[3]{mogamb.ttc} % Moga90Mincho Bold
\setgothicfont[2]{mogag.ttc} % Moga90Gothic
\setboldgothicfont[2]{mogagb.ttc} % Moga90Gothic Bold
\setxboldgothicfont[2]{mogagb.ttc} % Moga90Gothic Bold
\setmarugothic[2]{mobog.ttc} % Mobo90Gothic

```

prefer2004jis 指定時

```

\setminchofont[0]{mogam.ttc} % MogaMincho
\setboldminchofont[0]{mogamb.ttc} % MogaMincho Bold
\setgothicfont[0]{mogag.ttc} % MogaGothic
\setboldgothicfont[0]{mogagb.ttc} % MogaGothic Bold
\setxboldgothicfont[0]{mogagb.ttc} % MogaGothic Bold
\setmarugothic[0]{mobog.ttc} % MoboGothic

```

- moga-mobo-ex : MogaEx フォント + MoboEx フォント。

※「丸ゴシック」ファミリに MoboExGothic を充てている。

※フォント実体を変えることで prefer2004jis オプションに対応させている。

prefer2004jis 非指定時

```

\setminchofont[4]{mogam.ttc} % MogaEx90Mincho
\setboldminchofont[4]{mogamb.ttc} % MogaEx90Mincho Bold
\setgothicfont[3]{mogag.ttc} % MogaEx90Gothic
\setboldgothicfont[3]{mogagb.ttc} % MogaEx90Gothic Bold
\setxboldgothicfont[3]{mogagb.ttc} % MogaEx90Gothic Bold
\setmarugothic[3]{mobog.ttc} % MoboEx90Gothic

```

prefer2004jis 指定時

```

\setminchofont[1]{mogam.ttc} % MogaExMincho
\setboldminchofont[1]{mogamb.ttc} % MogaExMincho Bold
\setgothicfont[1]{mogag.ttc} % MogaExGothic
\setboldgothicfont[1]{mogagb.ttc} % MogaExGothic Bold
\setxboldgothicfont[1]{mogagb.ttc} % MogaExGothic Bold
\setmarugothic[1]{mobog.ttc} % MoboExGothic

```

- moga-maruberi : Moga フォント + モトヤ L マルベリ 3 等幅。

※ moga-mobo と以下を除いて同じ。

```

\setmarugothic{MTLmr3m.ttf} % モトヤ L マルベリ 3 等幅

```

- ume : 梅フォント。

```

\setminchofont{ume-tmo3.ttf} % 梅明朝

```

- \setgothicfont{ume-tgo5.ttf} % 梅ゴシック 05
 - \setmediumgothicfont{ume-tgo4.ttf} % 梅ゴシック
 - \setmarugothic{ume-tgo5.ttf} % 梅ゴシック 05
- kozuka-pro : 小塚フォント (Pro 版)。
 - \setminchofont{KozMinPro-Regular.otf} % 小塚明朝 Pro R
 - \setlightminchofont{KozMinPro-Light.otf} % 小塚明朝 Pro L
 - \setboldminchofont{KozMinPro-Bold.otf} % 小塚明朝 Pro B
 - \setgothicfont{KozGoPro-Medium.otf} % 小塚ゴシック Pro M
 - \setmediumgothicfont{KozGoPro-Regular.otf} % 小塚ゴシック Pro R
 - \setboldgothicfont{KozGoPro-Bold.otf} % 小塚ゴシック Pro B
 - \setxboldgothicfont{KozGoPro-Heavy.otf} % 小塚ゴシック Pro H
 - \setmarugothicfont{KozGoPro-Heavy.otf} % 小塚ゴシック Pro H
- kozuka-pr6 : 小塚フォント (Pr6 版)。
 - \setminchofont{KozMinProVI-Regular.otf} % 小塚明朝 Pro-VI R
 - \setlightminchofont{KozMinProVI-Light.otf} % 小塚明朝 Pro-VI L
 - \setboldminchofont{KozMinProVI-Bold.otf} % 小塚明朝 Pro-VI B
 - \setgothicfont{KozGoProVI-Medium.otf} % 小塚ゴシック Pro-VI M
 - \setmediumgothicfont{KozGoProVI-Regular.otf} % 小塚ゴシック Pro-VI R
 - \setboldgothicfont{KozGoProVI-Bold.otf} % 小塚ゴシック Pro-VI B
 - \setxboldgothicfont{KozGoProVI-Heavy.otf} % 小塚ゴシック Pro-VI H
 - \setmarugothicfont{KozGoProVI-Heavy.otf} % 小塚ゴシック Pro-VI H
- kozuka-pr6n : 小塚フォント (Pr6n 版)。
 - \setminchofont{KozMinPr6N-Regular.otf} % 小塚明朝 Pr6N R
 - \setlightminchofont{KozMinPr6N-Light.otf} % 小塚明朝 Pr6N L
 - \setboldminchofont{KozMinPr6N-Bold.otf} % 小塚明朝 Pr6N B
 - \setgothicfont{KozGoPr6N-Medium.otf} % 小塚ゴシック Pr6N M
 - \setmediumgothicfont{KozGoPr6N-Regular.otf} % 小塚ゴシック Pr6N R
 - \setboldgothicfont{KozGoPr6N-Bold.otf} % 小塚ゴシック Pr6N B
 - \setxboldgothicfont{KozGoPr6N-Heavy.otf} % 小塚ゴシック Pr6N H
 - \setmarugothicfont{KozGoPr6N-Heavy.otf} % 小塚ゴシック Pr6N H
- hiragino-pro : ヒラギノフォント基本 6 書体セット (Pro/Std 版) + 明朝 W2。
 - \setminchofont{HiraMinPro-W3.otf} % ヒラギノ明朝 Pro W3
 - \setlightminchofont{HiraMinPro-W2.otf} % ヒラギノ明朝 Pro W2
 - \setboldminchofont{HiraMinPro-W6.otf} % ヒラギノ明朝 Pro W6
 - \setgothicfont{HiraKakuPro-W3.otf} % ヒラギノ角ゴ Pro W3
 - \setboldgothicfont{HiraKakuPro-W6.otf} % ヒラギノ角ゴ Pro W6
 - \setxboldgothicfont{HiraKakuStd-W8.otf} % ヒラギノ角ゴ Std W8
 - \setmarugothicfont{HiraMaruPro-W4.otf} % ヒラギノ丸ゴ Pro W4
- hiragino-pron : ヒラギノフォント基本 6 書体セット (ProN/StdN 版) + 明朝 W2。
 - \setminchofont{HiraMinProN-W3.otf} % ヒラギノ明朝 ProN W3
 - \setlightminchofont{HiraMinProN-W2.otf} % ヒラギノ明朝 ProN W2
 - \setboldminchofont{HiraMinProN-W6.otf} % ヒラギノ明朝 ProN W6
 - \setgothicfont{HiraKakuProN-W3.otf} % ヒラギノ角ゴ ProN W3
 - \setboldgothicfont{HiraKakuProN-W6.otf} % ヒラギノ角ゴ ProN W6
 - \setxboldgothicfont{HiraKakuStdN-W8.otf} % ヒラギノ角ゴ StdN W8
 - \setmarugothicfont{HiraMaruProN-W4.otf} % ヒラギノ丸ゴ ProN W4
- hiragino-elcapitan-pro : ヒラギノフォント (Mac OS X El Capitan 搭載 ; Pro/Std 版) + 明朝

W2。

```
\setminchofont[1]{HiraginoSerif-W3.ttc}
\setlightminchofont{HiraMinPro-W2.otf}
\setboldminchofont[1]{HiraginoSerif-W6.ttc}
\setgothicfont[3]{HiraginoSans-W3.ttc}
\setboldgothicfont[3]{HiraginoSans-W6.ttc}
\setxboldgothicfont[2]{HiraginoSans-W8.ttc}
\setmarugothicfont[0]{HiraginoSansR-W4.ttc}
```

- hiragino-elcapitan-pron: ヒラギノフォント (Mac OS X El Capitan 搭載; ProN/StdN 版) + 明朝 W2。

```
\setminchofont[0]{HiraginoSerif-W3.ttc}
\setlightminchofont{HiraMinProN-W2.otf}
\setboldminchofont[0]{HiraginoSerif-W6.ttc}
\setgothicfont[2]{HiraginoSans-W3.ttc}
\setboldgothicfont[2]{HiraginoSans-W6.ttc}
\setxboldgothicfont[3]{HiraginoSans-W8.ttc}
\setmarugothicfont[1]{HiraginoSansR-W4.ttc}
```

- morisawa-pro: モリサワフォント基本 7 書体 (Pro 版)。

```
\setminchofont{A-OTF-RyuminPro-Light.otf} % A-OTF リュウミン Pro L-KL
\setboldminchofont{A-OTF-FutoMinA101Pro-Bold.otf} % A-OTF 太ミン A101 Pro
\setgothicfont{A-OTF-GothicBBBPro-Medium.otf} % A-OTF 中ゴシック BBB Pro
\setboldgothicfont{A-OTF-FutoGoB101Pro-Bold.otf} % A-OTF 太ゴ B101 Pro
\setxboldgothicfont{A-OTF-MidashiGoPro-MB31.otf} % A-OTF 見出ゴ MB31 Pro
\setmarugothicfont{A-OTF-Jun101Pro-Light.otf} % A-OTF じゅん Pro 101
```

- morisawa-pr6n: モリサワフォント基本 7 書体 (Pr6N 版^{*15})。

```
\setminchofont{A-OTF-RyuminPr6N-Light.otf} % A-OTF リュウミン Pr6N L-KL
\setboldminchofont{A-OTF-FutoMinA101Pr6N-Bold.otf} % A-OTF 太ミン A101 Pr6N
\setgothicfont{A-OTF-GothicBBBPr6N-Medium.otf} % A-OTF 中ゴシック BBB Pr6N
\setboldgothicfont{A-OTF-FutoGoB101Pr6N-Bold.otf} % A-OTF 太ゴ B101 Pr6N
\setxboldgothicfont{A-OTF-MidashiGoPr6N-MB31.otf} % A-OTF 見出ゴ MB31 Pr6N
\setmarugothicfont{A-OTF-Jun101Pro-Light.otf} % A-OTF じゅん Pro 101
```

- yu-win: 游書体 (Windows 8.1 搭載版)。

```
\setminchofont{yumin.ttf} % 游明朝 Regular
\setlightminchofont{yuminl.ttf} % 游明朝 Light
\setboldminchofont{yumindb.ttf} % 游明朝 Demibold
\setgothicfont{yugothic.ttf} % 游ゴシック Regular
\setboldgothicfont{yugothib.ttf} % 游ゴシック Bold
\setxboldgothicfont{yugothib.ttf} % 游ゴシック Bold
\setmarugothicfont{yugothic.ttf} % 游ゴシック Regular
```

- yu-win10: 游書体 (Windows 10 搭載版)。

※フォントの性質のため、この設定では欧文引用符“ ” ‘ ’ の出力が不正になる。この不具合は unicode オプションを指定する (dvipdfmx の 20170918 版が必要)、または代わりに yu-win10+ プリセットを指定する (dvipdfmx の 20170318 版が必要) ことで回避できる。詳細については 8 節を参照されたい。

```
\setminchofont{yumin.ttf}
\setlightminchofont{yuminl.ttf}
```

*15 「じゅん」は Pr6N 版が存在しないため Pro 版が使われる。

- ```

\setboldminchofont{yumindb.ttf}
\setgothicfont[0]{YuGothM.ttc}
\setmediumgothicfont[0]{YuGothR.ttc}
\setboldgothicfont[0]{YuGothB.ttc}
\setxboldgothicfont[0]{YuGothB.ttc}
\setmarugothicfont[0]{YuGothM.ttc}

```
- **yu-osx** : 游書体 (Mac OS X 搭載版)。

```

\setminchofont{YuMin-Medium.otf} % 游明朝体 ミディアム
\setboldminchofont{YuMin-Demibold.ttf} % 游明朝体 デミボールド
\setgothicfont{YuGo-Medium.otf} % 游ゴシック体 ミディアム
\setboldgothicfont{YuGo-Bold.otf} % 游ゴシック体 ボールド
\setxboldgothicfont{YuGo-Bold.otf} % 游ゴシック体 ボールド
\setmarugothicfont{YuGo-Medium.otf} % 游ゴシック体 ミディアム

```
  - **sourcehan-otc** : Source Han Serif (源ノ明朝) + Source Han Sans (源ノ角ゴシック)、OTC 版。

```

\setminchofont[0]{SourceHanSerif-Regular.ttc}
\setlightminchofont[0]{SourceHanSerif-Light.ttc}
\setboldminchofont[0]{SourceHanSerif-Bold.ttc}
\setgothicfont[0]{SourceHanSans-Medium.ttc}
\setmediumgothicfont[0]{SourceHanSans-Regular.ttc}
\setboldgothicfont[0]{SourceHanSans-Bold.ttc}
\setxboldgothicfont[0]{SourceHanSans-Heavy.ttc}
\setmarugothicfont[0]{SourceHanSans-Medium.ttc}

```
  - **sourcehan** : Source Han Serif (源ノ明朝) + Source Han Sans (源ノ角ゴシック)、言語別 OTF 版。

```

\setminchofont{SourceHanSerif-Regular.otf}
\setlightminchofont{SourceHanSerif-Light.otf}
\setboldminchofont{SourceHanSerif-Bold.otf}
\setgothicfont{SourceHanSans-Medium.otf}
\setmediumgothicfont{SourceHanSans-Regular.otf}
\setboldgothicfont{SourceHanSans-Bold.otf}
\setxboldgothicfont{SourceHanSans-Heavy.otf}
\setmarugothicfont{SourceHanSans-Medium.otf}

```
  - **sourcehan-jp** : Source Han Serif JP (源ノ明朝) + Source Han Sans JP (源ノ角ゴシック)、地域別サブセット OTF 版。

```

\setminchofont{SourceHanSerifJP-Regular.otf}
\setlightminchofont{SourceHanSerifJP-Light.otf}
\setboldminchofont{SourceHanSerifJP-Bold.otf}
\setgothicfont{SourceHanSansJP-Medium.otf}
\setmediumgothicfont{SourceHanSansJP-Regular.otf}
\setboldgothicfont{SourceHanSansJP-Bold.otf}
\setxboldgothicfont{SourceHanSansJP-Heavy.otf}
\setmarugothicfont{SourceHanSansJP-Medium.otf}

```
  - **noto-otc** : Noto Serif CJK JP + Noto Sans CJK JP、OTC 版。

```

\setminchofont[0]{NotoSerifCJK-Regular.ttc}
\setlightminchofont[0]{NotoSerifCJK-Light.ttc}
\setboldminchofont[0]{NotoSerifCJK-Bold.ttc}
\setgothicfont[0]{NotoSansCJK-Medium.ttc}
\setmediumgothicfont[0]{NotoSansCJK-Regular.ttc}

```

- ```

\setboldgothicfont[0]{NotoSansCJK-Bold.ttc}
\setxboldgothicfont[0]{NotoSansCJK-Black.ttc}
\setmarugothicfont[0]{NotoSansCJK-Medium.ttc}

```
- noto : Noto Serif CJK JP + Noto Sans CJK JP、言語別 OTF 版。

```

\setminchofont{NotoSerifCJKjp-Regular.otf}
\setlightminchofont{NotoSerifCJKjp-Light.otf}
\setboldminchofont{NotoSerifCJKjp-Bold.otf}
\setgothicfont{NotoSansCJKjp-Medium.otf}
\setmediumgothicfont{NotoSansCJKjp-Regular.otf}
\setboldgothicfont{NotoSansCJKjp-Bold.otf}
\setxboldgothicfont{NotoSansCJKjp-Black.otf}
\setmarugothicfont{NotoSansCJKjp-Medium.otf}

```
 - noto-jp : Noto Serif JP + Noto Sans JP、地域別サブセット OTF 版。

```

\setminchofont{NotoSerifJP-Regular.otf}
\setlightminchofont{NotoSerifJP-Light.otf}
\setboldminchofont{NotoSerifJP-Bold.otf}
\setgothicfont{NotoSansJP-Medium.otf}
\setmediumgothicfont{NotoSansJP-Regular.otf}
\setboldgothicfont{NotoSansJP-Bold.otf}
\setxboldgothicfont{NotoSansJP-Black.otf}
\setmarugothicfont{NotoSansJP-Medium.otf}

```

5.3 ptex-fontmaps 互換のオプション

ptex-fontmaps のプリセット名を別名として用意した。

- noEmbed : noembed の別名。
- kozuka : kozuka-pro の別名。
- hiragino : hiragino-pro の別名。
- hiragino-elcapitan : hiragino-elcapitan-pro の別名。
- morisawa : morisawa-pro の別名。

5.4 廃止されたオプション

以下に挙げるのは、0.5 版以降で非推奨となっていたプリセット設定である。これらは 1.0 版において廃止されたため、現在は使用するとエラーが発生する。

- ipa-otf : 「拡張子が .otf の」IPA フォント。
※代替のプリセットはない。
- ipa-otf-dx : 「拡張子が .otf の」IPA フォント + HG フォント。
※代替のプリセットはない。
- kozuka4 : 小塚フォント (Pro 版) の単ウェイト使用。
※ kozuka-pro + oneweight オプションで代替可能。
- kozuka6 : 小塚フォント (Pr6 版) の単ウェイト使用。

- ※ `kozuka-pr6 + oneweight` オプションで代替可能。
- `kozuka6n` : 小塚フォント (Pr6n 版) の単ウェイト使用。
 ※ `kozuka-pr6n + oneweight` オプションで代替可能。
- `hiragino` : ヒラギノフォントの単ウェイト使用。
 ※ `hiragino-pro + oneweight` オプションで代替可能。
 ※ 1.2a 版以降で、`hiragino-pro` の別名として再定義された。
- `ms-dx` : `ms-hg` の別名。
- `ipa-ttf` : `ipa` の別名。
- `ipa-ttf-dx` : `ipa-hg` の別名。
- `ipav2` : `ipa` の別名。
- `ipav2-dx` : `ipa-hg` の別名。
- `ipa-dx` : `ipa-hg` の別名。
- `hiragino-dx` : `hiragino-pro` の別名。

6 ファイルプリセット機能

ファイルプリセット機能を利用すると、既存の `dvipdfmx` 用のマップファイルの読込を文書内で指定することが可能になる。パッケージオプションに次の何れかの形式の文字列を指定すると、ファイルプリセットの指定と見なされる。

- `t1`:名前: `TEX` Live 用ファイルプリセット。
- `file`:名前: 単純ファイルプリセット。

※ 1.6a 版以前では、`TEX` Live 用が「+名前」、通常が「*名前」という形式であった。これらの古い形式も当面の間はサポートされるが、新しい形式の使用を推奨する。

6.1 `TEX` Live 用ファイルプリセット機能

`TEX` Live では (u)p`LATEX` のフォントの設定を `kanji-config-updmap` というユーティリティで行うことができる。ここでは、決まった形式のファイル名をもつ `dvipdfmx` 用のマップファイルを用意していて、ユーザが要求したプリセット名に対応したファイルを `updmap` の機構を用いて有効化することで、`dvipdfmx` の既定の設定を切り替えている。

パッケージオプションとして `t1`: で始まる文字列（仮に `t1:NAME` とする）を与えると、`kanji-config-updmap` 用のマップファイルの読込が指示される。具体的には、以下の名前のマップファイルが読み込まれる。

- p`LATEX` の場合 :
 - `ptex-NAME.map`
 - `otf-NAME.map`
- up`LATEX` の場合、上記のものに加えて以下のもの :
 - `uptex-NAME.map`

– otf-up-NAME.map

例えば、pL^AT_EX 文書において以下のようにパッケージを読み込んだとする。

```
\usepackage[tl:yu-win]{pxchfon}
```

この場合、ptex-yu-win.map と otf-yu-win.map の 2 つのマップファイルが dvipdfmx 実行時に読み込まれる。

6.2 単純ファイルプリセット機能

パッケージオプションとして file: で始まる文字列（仮に file:NAME とする）を与えると、NAME.map という名前のマップファイルの読込が指示される。

例えば、以下のようにパッケージを読み込んだとする。

```
\usepackage[file:yu]{pxchfon}
```

この場合、yu.map というマップファイル^{*16}が dvipdfmx 実行時に読み込まれる。

7 カスタムプリセット機能

1.7 版で新設された「カスタムプリセット機能」とは、ユーザが独自にプリセットを用意するためのものである。ユーザが定義したプリセット（カスタムプリセット）を「pxchfon.cfg」という名前のカスタムプリセット定義ファイルに記述してそのファイルを T_EX が読める場所に置くと、pxchfon パッケージ読込時にオプションとして指定することでカスタムプリセットを利用できるようになる。

カスタムプリセットを定義する命令は以下の通りである。これらの命令は pxchfon.cfg の中でのみ利用できる。

- `\pxchfonDeclareOneWeightPreset{<名前>}{<明朝>}{<ゴシック>}`: 単ウェイトのカスタムプリセットを定義する。プリセットの名前は <名前> の文字列の前に「user:」を前置したものになる。第 2 引数以降は使用するフォントファイル名（TTC 形式の場合は番号も含めて :番号:ファイル名 の形式）を指定する。
- `\pxchfonDeclareMultiWeightPreset{<名前>}{<明朝・細>}{<明朝・中>}{<明朝・太>}{<ゴシック・中>}{<ゴシック・太>}{<ゴシック・極太>}{<ゴシック・単>}{<丸ゴシック>}`: 多ウェイトのカスタムプリセットを定義する。引数の意味は前項と同様である。<ゴシック・単> は「oneweight オプション指定時のゴシック体」を表す。

例えば、全ての和文フォントを（Microsoft Office 付属の）「HG 創英角ポップ体」に置き換えるような単ウェイトのカスタムプリセット user:soeikakupoptai を定義するには次の命令を実行する。

```
\pxchfonDeclareOneWeightPreset{soeikakupoptai}
{:0:HGRPP1.TTC}{:0:HGRPP1.TTC}
```

このカスタムプリセットを利用したい場合はパッケージ読込を以下のようにする。

^{*16} 例えば W32T_EX では yu.map というマップファイルが用意されている。

```
\usepackage[user:soeikakupoptai,...]{pxchfon}
```

8 Unicode 直接指定

dvipdfmx のフォントマップ設定において、和文フォントのエンコーディングを指定する方法は“CMap 指定”と“Unicode 直接指定”の 2 種類がある。^{*17}かつては、Unicode で包摂されている異体字を区別するためには CMap 指定の利用が必須であったため、慣習的に、dvipdfmx のフォントマップ設定においては CMap 指定が主に用いられてきた。

しかしこの CMap 指定は、Adobe-Japan1 (AJ1)^{*18}のグリフ集合に対応した OpenType フォントにしか適用できない、という欠点がある。近年は、“AJ1 でない” OpenType フォント^{*19}が普及しつつあり、そのようなフォントでは異体字の切替などの付加機能を専ら“OpenType 属性の指定”により行うことを想定している。これに対応するため、dvipdfmx のマップ指定において“OpenType 属性の指定”がサポートされるようになった。

pxchfon では和文フォントのエンコーディングに対する Unicode 直接指定をサポートしている。特に 1.0 版から、新しい dvipdfmx の OpenType 属性の指定を積極的に利用することで、“AJ1 でない”フォントを使用した場合でも、CMap 指定の場合の機能性を可能な限り保つことを目指している。

※ Unicode 直接指定に対するサポートは発展途上であるため、過渡的な要素が多く混ざっていてやや煩雑になっていることに注意してほしい。

■ “Unicode 直接指定” オプション 以下のパッケージオプションを指定することで Unicode 直接指定の有効・無効を切り替えられる。

- **unicode** : 全般的に Unicode 直接指定を利用する。最も理想的な設定であるが、 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Live 2018 以降の dvipdfmx (20170918 版以降) が必要である。
- **unicode*** : $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Live 2017 の dvipdfmx (20170318 版) 専用の過渡的な設定。
※ 1.5 版以降では非推奨であり、将来的に廃止される予定である。
- **directunicode*** : 全般的に Unicode 直接指定を利用するが、古い ($\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Live 2016 以前の) dvipdfmx に対応するために、OpenType 属性の指定を全く行わない。つまり、入力 of Unicode 文字に対する既定のグリフが常に出力され、異体字の区別は全て無効になる。
- **directunicode** : `japanese-otf` の `\UTF` 入力のフォントに限って Unicode 直接指定を利用する。
directunicode* と同じく OpenType 属性の指定を全く行わない。
※前述の通り **directunicode*** はデメリットが強いため、適用範囲を限定したもの。

■ Unicode 直接指定専用プリセット 以下に挙げるプリセット設定は“AJ1 でない” OpenType フォントを利用するものである。そのため、これらのプリセットを指定した場合は、自動的に **unicode** が (既定として) 指定される^{*20}。

^{*17} 詳細については dvipdfmx のマニュアルを参照されたい。

^{*18} または各々の CJK 言語の“Adobe 標準”のグリフ集合、例えば簡体字中国語なら Adobe-GB1。

^{*19} 例えば、Adobe 開発のフリーフォントの「Source Han Serif (源ノ明朝)」など。

^{*20} 1.0~1.1b 版では $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ Live 2017 のための暫定措置として「+ 付の特殊プリセットへの自動振替」が行われていたが、1.2 版から本来の仕様が適用される。

- sourcehan-otc
- sourcehan
- sourcehan-jp
- noto-otc
- noto
- noto-jp

■特殊プリセット指定(非推奨) 以下に挙げるプリセット設定は T_EX Live 2017 (dvipdfmx 20170318 版) のために用意された特殊な設定である。

1.5 版においてこれらのプリセットは非推奨となった。将来的には廃止される予定である。そのため、これらのプリセットを利用している場合は、T_EX システムを更新 (dvipdfmx を 20170918 版以降に) した上で、“本来の正しい設定”に移行することを推奨する。

- sourcehan+, sourcehan-otc+, noto+, noto-otc+ :
→代わりに + 無しの名前 (sourcehan 等) を指定する。
※前述の通り、unicode は自動的に補われる。
※日本語版以外のフォントはもはや不要である。
- yu-win10+ :
→代わりに yu-win10 と unicode を指定する。

8.1 Unicode 直接指定に関する注意事項

- プリセット指定または `\set...font` 命令でフォントを置換していないウェイトについては、Unicode 直接指定は無効になる。従って、そのようなウェイトは想定する出力にはならず、また Unicode 直接指定を必要とする機能 (expert や glyphid の各オプション) も使えない。
- 特に、expert については、非置換のウェイトの動作が非常に複雑で解りにくい。従って、expert を指定する場合は、全ファミリー・ウェイトについて置換を指定することを推奨する (これに反する場合は警告が出る)。
- 非埋込を指定したウェイトについては Unicode 直接指定は無効になる。この動作も非常に解りにくい。ため、Unicode 直接指定を用いる場合は、非埋込の指定をしないことを推奨する (これに反する場合は警告が出る)。

9 dvipdfmx のページ抜粋処理への対応

dvipdfmx には元の DVI 文書の一部のページだけを抜粋して PDF 文書に変換する機能がある (`-s` オプション)。ところが、本パッケージではユーザが指定したフォントマップ情報を DVI の先頭ページに書き出すという処理方法をとっている (すなわち「ページ独立性」を保っていない) ため、先頭ページを含まない抜粋を行った場合は、フォント置換が効かないという不具合が発生する。

この問題を解決するのが `everypage` パッケージオプションである。このオプションが指定された場合は、DVI 文書の全てのページにフォントマップ情報を書き出すので、ページ抜粋を行っても確実にフォント置換

が有効になる。ただし、このオプションを指定する場合は `atbegshi` パッケージが必要である。

10 欧文フォントの置換の原理

指定された和文フォントの半角部分からなる欧文フォントファミリーとして `cfjar` (明朝)、`cfjas` (ゴシック)、`cfjam` (丸ゴシック) の 3 つ (以下では CFJA ファミリーと総称する) を定義している。その上で、CFJA ファミリーに対するマップ指定を和文と同じ方法で行っている。

■**alphabet オプション指定時** オプション `alphabet` を指定した場合、CFJA ファミリーを既定の欧文ファミリーとして設定する (`cfjar → \rmdefault`; `cfjas → \sfdefault`)。従って、例えば一時的に従来の CM フォントを使いたい場合は、適宜ファミリーの変更 (`\fontfamily{cmr}` 等) を行えばよい。

■**relfont オプション指定時** オプション `relfont` を指定した場合、CFJA ファミリーを和文ファミリーの従属欧文フォントに設定する (`cfjar → \mcfamily`; `cfjas → \gtfamily`; `cfjam → \mgfamily`)。

■**利用可能な欧文エンコーディング** 現状で、CFJA ファミリーが対応している欧文エンコーディングは OT1、T1、TS1 の 3 つである*²¹ (TS1 は事前に `textcomp` を読み込んだ場合にのみ有効)。このため、欧文フォントの置換 (`alphabet` や `relfont` オプション) を利用する文書は、欧文エンコーディングが OT1 か T1 である必要がある*²²。

何れのエンコーディングを用いる場合でも、様々な理由により、利用できる文字の種類が限られている。

1. AJ1 対応の OpenType フォントを指定して、かつ `usecmaphoralphabet` オプションを指定した場合、以下の文字 (でかつ当該の欧文エンコーディングに含まれるもの) が使える。

※ AJ1 対応フォント用のプリセット (`kozuka-pr6n` 等) を指定した場合は `usecmaphoralphabet` は自動的に有効になる。

- a) ASCII 文字*²³および en-ダッシュ (—)。

※これらは半角幅である。

- b) クオート (“ ” ‘ ’)。

※これらは和文用の全角幅の字形を、空きを詰めて半角幅で出力する。

- c) em-ダッシュ (—)。

※これらは U+2015 の全角幅の字形をそのまま全角幅で出力する。

2. 1 以外の場合、以下の文字 (でかつ当該の欧文エンコーディングに含まれるもの) が使える。

※この場合、既定では欧文は `directunicode*` 相当の状態になる。

- 1 項の a、b、c の文字はフォントが正しい幅 (a は半角、b と c は全角) の字形を持っていれば使える。
- それ以外でも、Unicode 文字の大部分は、フォントが半角幅の字形を持っていれば使える。
※半角幅のまま出力される。
- `unicode` オプションを指定すると、正しい幅の字形を優先して使おうとするため、既定の場合より

*²¹ 1.4 版で T1 に“暫定対応”し、1.5 版で T1 と TS1 に正式に対応した。

*²² TS1 は記号用のエンコーディングであり、本文の既定のエンコーディングとして使うものではない。

*²³ 出力される文字の話であることに注意。例えば、OT1 や T1 で ' (U+0027) を入力したときに出力される文字は ' (U+2019) であり、これは ASCII 文字ではない。なお、TS1 は U+0027 を含む。

も多くの文字が使える。

11 注意事項

- 指定できるフォントは等幅のものに限られる。実際に使われるメトリックは置換前と変わらない。(例えば jsarticle の標準設定なら JIS メトリック)
- 欧文部分を置き換えた場合、残念ながら欧文も等幅（半角幅）になってしまう。
- japanese-otf パッケージ使用時に \UTF や \CID で指定した文字が出力されるかは、指定したフォントがその文字を持っているかに依存する。
- deluxe 付きの japanese-otf パッケージと alphabet 付きの pxchfon を同時に使う場合には、japanese-otf パッケージを先に読み込む必要がある。(これに反した場合はエラーになる。)
- 単ウエイトの場合は、明朝の太字はゴシックになるという一般的な設定に欧文フォントの置換の際にも従っているが、明朝のみが置換されている場合は、明朝の置換フォントが太字にも適用される。

付録 A dvipdfmx 以外の DVI ウェアでの使用

本パッケージの核心の機能である「使用フォントを文書中で指定する」ことの実現には dvipdfmx の拡張機能を利用している。従って、dvipdfmx の利用が必須となるのだが、(プレビュー等の目的で) 文書中で指定したフォントが反映されなくてもよいのなら、他の DVI ウェアでも本パッケージを利用した DVI 文書を扱える可能性がある。

A.1 和文フォントだけを置き換えた場合 (noalphabet 指定時)

この設定で生成される DVI ファイルを dvipdfmx 以外の DVI ウェアに読ませた場合、フォント置換が無視され、そのソフトウェアで設定されたフォントで出力されるはずである。

A.2 欧文フォントも置き換えた場合 (alphabet 指定時)

欧文フォントを置き換えた DVI ファイルは、独自の欧文フォント (r-cfja* という形式の名前) を含んでいるので、少なくともそれに関する設定をしない限りは dvipdfmx 以外の DVI ウェアで処理することができない。さらに、このフォントを扱うためには DVI ウェアがサブフォント (sfd) に対応している必要がある。文書中での設定を dvipdfmx 以外の DVI ウェアで活かすことはできない。しかし、「独自部分の欧文フォントを常に特定の代替フォントで表示させる」ということは、sfd 対応の DVI ウェアであれば可能である。

以下に、ttf2pk について、「常に IPA フォントで代替する」ための設定を掲げておく。この記述を ttf2pk のマップファイル (ttfonts.map) に加えると、例えば、dviout で本パッケージ使用の DVI ファイルを閲覧できるようになる。

r-cfjar-l-@PXcjk0@	msmincho.ttc FontIndex=0
r-cfjar-r-@PXcjk0@	msmincho.ttc FontIndex=0
r-cfjar-b-@PXcjk0@	msmincho.ttc FontIndex=0
r-cfjas-r-@PXcjk0@	msgothic.ttc FontIndex=0

```

r-cfjas-b-@PXcjk0@    msgothic.ttc FontIndex=0
r-cfjas-x-@PXcjk0@    msgothic.ttc FontIndex=0
r-cfjam-r-@PXcjk0@    msgothic.ttc FontIndex=0

```

付録 B pxjafont パッケージ

現在の版の pxchfon パッケージは旧来の pxjafont の機能を取り込んでいるため、pxjafont は不要である。古い環境との互換性のため pxjafont を残していたが、1.0 版において pxjafont の使用を非推奨の扱いとした。(近い将来に廃止予定。)

pxjafont を利用しているユーザは、5 節を参照して現在の pxchfon 用の適切な設定に書き直す必要がある。
※古いプリセット名の中に廃止されたものがあるので注意。

付録 C 中国語・韓国語フォントへの対応

0.7c 版で japanese-otf パッケージ (multi オプション指定) および upTeX 標準の中国語・韓国語フォントについてのサポートを始めた。以下の命令で、実フォントの置換指定ができる。

- `\setkoreanminchofont` [`<番号>`] {`<フォントファイル名>`} : 韓国語・明朝体。
- `\setkoreangothicfont` [`<番号>`] {`<フォントファイル名>`} : 韓国語・ゴシック体。
- `\setschineseminchofont` [`<番号>`] {`<フォントファイル名>`} : 簡体字中国語・明朝体 (宋体)。
- `\setschinesegothicfont` [`<番号>`] {`<フォントファイル名>`} : 簡体字中国語・ゴシック体 (黒体)。
- `\setttchineseminchofont` [`<番号>`] {`<フォントファイル名>`} : 繁体字中国語・明朝体 (明体)。
- `\setttchinesegothicfont` [`<番号>`] {`<フォントファイル名>`} : 繁体字中国語・ゴシック体 (黒体)。

注意事項。

- プリセット指定は中国語・韓国語のフォントについては何も指定しない。従って、上記の命令を用いない場合は、これらのフォントのマップ再設定が行われることはない。
- 「Unicode 直接指定オプション」は中国語・韓国語のフォントに対しても有効である。`directunicode` の場合は「japanese-otf パッケージの Unicode 入力命令」(`\UTFK`、`\UTFM`、等) が対象となり、それ以外はこれに加えて upTeX 標準のフォントも対象になる。