

# pBibTeX / upBibTeX マニュアル

日本語 TeX 開発コミュニティ\*

version 0.99d-j0.33, 2022 年 2 月 6 日

pBibTeX と upBibTeX は、それぞれ pTeX と upTeX と組み合わせて使用することを想定して開発された BibTeX の日本語対応版である。

- pBibTeX の開発元：  
<https://github.com/texjporg/tex-jp-build/>
- upBibTeX の開発元：  
<http://www.t-lab.opal.ne.jp/tex/uptex.html>
- 本ドキュメントの開発元：  
<https://github.com/texjporg/pbibtex-manual/>

BibTeX の日本語化は、電力中央研究所の松井正一氏によって jBibTeX という名称で公開されたもの<sup>\*1</sup>がベースであり、その仕様は [1] に詳しい。また、オリジナルの BibTeX 付属ドキュメント `btxdoc.pdf` と `btXHak.pdf` を日本語に訳し、jBibTeX について補足を加えたドキュメントも用意されている [2, 3]。これらのドキュメントは、レガシーエンコーディングの扱いに関する記述（第 3 節を参照）を除き現在の (u)pBibTeX でも有効であるので、参照されたい。

本文書では、オリジナルの BibTeX の仕様を把握している読者を想定し、pBibTeX および upBibTeX における機能の変更・追加点を説明する。

## 目次

1	日本語化の仕様	2
1.1	多バイト文字の扱い	2
1.2	文字種が増えたことへの対応	3
1.3	日本語文字とそれ以外の区別	3
2	コマンドラインオプション	4
3	参考：jBibTeX と pBibTeX の違い	4

---

\* <https://texjp.org>, e-mail: [issue\(at\)texjp.org](mailto:issue(at)texjp.org)

<sup>\*1</sup> 最終版は jBibTeX 0.31 である。1991/01/01 付の jBibTeX 0.30 のパッケージに、1992/10/31 に `fj.comp.texhax` グループへ投稿されたバグ修正パッチ (0.31) を当てて得られる。

# 1 日本語化の仕様

Bi<sub>W</sub>TeX の日本語化の特徴は、大きく分けて 3 つである。

- 多バイト文字の扱い
- 文字種が増えたことへの対応
- 日本語文字とそれ以外の区別

## 1.1 多バイト文字の扱い

(u)pBi<sub>W</sub>TeX は、Bi<sub>W</sub>TeX から最小の変更量で日本語を含む文献情報を扱えるようにした都合上、多バイト文字の扱いは以下のとおりとなっている。

- 文字列の位置や長さは「文字単位」ではなく「バイト単位」でカウントする。
- その結果として、開始位置や終了位置が多バイト文字の途中となる場合は、多バイト文字の途中で切られないように位置を調整して多めに切り出す。

具体的には、以下のスタイルパラメータが該当する。

- `substring$`  
整数値 2 つ（長さ、開始位置）と文字列リテラル 1 つを `pop` し、指定の長さ（バイト長）の部分文字列を `push` する組込関数である。
  - 開始位置が多バイト文字の 2 バイト目以降であれば、1 バイト目から取り出す。
  - 終了位置が多バイト文字の最終バイトでなければ、最終バイトまで取り出す。したがって、pBi<sub>W</sub>TeX（内部コード euc）では最大 2 バイト長く、upBi<sub>W</sub>TeX（内部コード uptex）では最大 6 バイト長い文字列が取り出されうる。
- `text.prefix$`  
整数値（長さ）と文字列リテラルを `pop` し、文字列の先頭から指定の長さ（バイト長）の連続した文字列を `push` する組込関数である。終了位置が多バイトの文字の途中にならないよう調整されるので、pBi<sub>W</sub>TeX（内部コード euc）では最大 1 バイト長く、upBi<sub>W</sub>TeX（内部コード uptex）では最大 3 バイト長い文字列が取り出されうる。

以下に例を示す。pBi<sub>W</sub>TeX と upBi<sub>W</sub>TeX では同じ文字でもバイト長が異なる場合があり、その結果として取り出される見かけの文字数が異なることに注意。

pBi<sub>W</sub>TeX（内部コード euc）の場合

"あいうえお" #-1 #1 substring\$	最後の 1 文字 = 「お」
"あいうえお" #-1 #2 substring\$	最後の 2 文字 = 「お」
"あいうえお" #-1 #3 substring\$	最後の 3 文字 = 「えお」
"あいうえお" #-1 #4 substring\$	最後の 4 文字 = 「えお」
"あいうえお" #-1 #5 substring\$	最後の 5 文字 = 「うえお」
"あいうえお" #-1 #6 substring\$	最後の 6 文字 = 「うえお」

upB<sub>W</sub>TeX (内部コード `uptex`) の場合

"あいうえお" #-1 #1 substring\$	最後の 1 文字 = 「お」
"あいうえお" #-1 #2 substring\$	最後の 2 文字 = 「お」
"あいうえお" #-1 #3 substring\$	最後の 3 文字 = 「お」
"あいうえお" #-1 #4 substring\$	最後の 4 文字 = 「えお」
"あいうえお" #-1 #5 substring\$	最後の 5 文字 = 「えお」
"あいうえお" #-1 #6 substring\$	最後の 6 文字 = 「えお」

## 1.2 文字種が増えたことへの対応

- `add.period$`

文字列リテラルを `pop` し、最後の文字 (} を除く) がピリオド類 `<.>` `<?>` `<!>` のいずれでもないときに `<.>` を最後に加える組込関数である。

pB<sub>W</sub>TeX では、全角の `<!>` `<?>` `<.>` `<。>` (それぞれ U+FF01, U+FF1F, U+3002, U+FF0E) もピリオド類とみなし、これらで終わっても `<.>` を付加しない。

upB<sub>W</sub>TeX ではさらに U+203C, U+2047, U+2048, U+2049 もピリオド類とみなす。

- `format.name$`

文字列 (フォーマット指定), 整数値 (何番めか), 文字列 (名前リスト) を `pop` し, フォーマットされた名前の文字列を `push` する組込関数である。

(u)pB<sub>W</sub>TeX では、日本人の姓名の間のスペースとして全角空白 U+3000 も半角空白と同じとみなし (全角空白は半角空白に変換して処理), また複数の氏名間の区切りとして `and` と同様に全角の読点 `<、>` とコンマ `<、>` (それぞれ U+FF0C, U+3001) も使える。

## 1.3 日本語文字とそれ以外の区別

- `is.kanji.str$`

(u)pB<sub>W</sub>TeX 独自の組込関数である。スタックトップの文字列リテラルを `pop` し、文字列中に「日本語文字」が 1 つでも含まれていれば整数値 1 を、含まれなければ 0 を `push` する。なお、「日本語文字」かどうかの判定は以下のように行う。

– pB<sub>W</sub>TeX の場合

ASCII の範囲 (0–127) に収まらない文字を全て「日本語文字」として扱う。

– upB<sub>W</sub>TeX (内部コード `uptex`) の場合

漢字・かな・ハングルに該当する Unicode ブロックの文字を「日本語文字」として扱う<sup>\*2</sup>。pB<sub>W</sub>TeX とは異なり、記号類 (句読点, 括弧類, ●○■□◆◇など) は「日本語文字」として扱わない (バージョン u1.27 (TeX Live 2021) 以降 [5])。

(u)pB<sub>W</sub>TeX 用のスタイルファイルでは、しばしば「日本人の姓名の間にはスペースを入れない」という挙動を実現するために `is.kanji.str$` 関数が実用されている。

---

<sup>\*2</sup> 実装上は upTeX の `\kcatcode` と同じブロック分けを流用しているののでそれに即して記述すると、既定値が 16 (kanji), 17 (kana), 19 (hangul) のブロックを真, 15 (latin), 18 (CJK symbol) のブロックを偽としている。

## 2 コマンドラインオプション

基本的には  $\text{BibTeX}$  と同様であるが、以下が追加されている。

- `-kanji=<encoding>`  
入出力ファイルの文字コードを指定する。利用可能な *<encoding>* の値：
  - $\text{pBibTeX}$  : `eu`, `sjis`, `jis`, `utf8`
  - $\text{upBibTeX}$  : `eu`, `sjis`, `jis`, `utf8`, `uptex`
- `-kanji-internal=<encoding>`  
内部コードを指定する ( $\text{upBibTeX}$  専用)。利用可能な *<encoding>* の値：
  - $\text{pBibTeX}$  : なし (常に `eu` に固定)
  - $\text{upBibTeX}$  : `eu`, `uptex`

## 3 参考： $\text{jBibTeX}$ と $\text{pBibTeX}$ の違い

松井氏による  $\text{jBibTeX}$  0.31 ( $\text{BibTeX}$  0.99c ベース) から現在の  $\text{pBibTeX}$  に至った経緯は以下のとおりである。

- 1994 年、都立大 (のち千葉大) の桜井貴文氏により、 $\text{jTeX}$  1.6 (web2c 6.1) の配布キットに含めるための調整 →  $\text{jBibTeX}$  0.32
- 1995 年、アスキー  $\text{pTeX}$  の配布キットに含めるための調整
- 2002 年、アスキーにより `-kanji` オプションの追加 →  $\text{jBibTeX}$  0.33
- 2009 年、日本語  $\text{TeX}$  開発コミュニティが  $\text{jBibTeX}$  をフォークし、 $\text{pBibTeX}$  に改名
- 2010 年、 $\text{pBibTeX}$  が  $\text{pTeX}$  とともに  $\text{TeX Live}$  へ収録される。のちに  $\text{BibTeX}$  0.99d 対応

$\text{jBibTeX}$  は当初 NTT  $\text{jTeX}$  と組み合わせて使用することを想定して開発されたため、文字コードに関する扱いに NTT  $\text{jTeX}$  由来のものが多かった。具体的には以下の機能があったが、コマンド名が `jbibtex` から `pbibtex` ( $\text{pTeX}$  と同じ接頭辞) に改名された 2009 年に削除されている [4]：

- JIS コードにおいて日本語文字コード開始・終了を示す種々のエスケープ・シーケンス (`ESC$@` と `ESC$B`, `ESC(J` と `ESC(B` など) を区別しない。
- 環境変数 `BIBTERMCODE`, `BIBFILECODE`<sup>\*3</sup> は、 $\text{pBibTeX}$  では参照されない。

---

<sup>\*3</sup> 松井氏のドキュメント `jbibtex.pdf` [1] の「3.3 漢字コードの扱い」および `jbtxdoc.pdf` [2] の「1. 概要」の  $\text{jBibTeX}$  での注意点として言及されているもの。

## 参考文献

- [1] 松井正一, 「日本語  $\LaTeX$ :  $\text{j}\LaTeX$ 」, `./jbibtex.pdf`
- [2] 松井正一, 「 $\LaTeX$ ing:  $\LaTeX$  の使い方」, `./jbtldoc.pdf`
- [3] 松井正一, 「Designing  $\LaTeX$  Styles —  $\LaTeX$  スタイルの作り方」, `./jbtldhak.pdf`
- [4] 土村展之, 「コマンド名問題 - ptexlive Wiki」  
<https://tutimura.ath.cx/ptexlive/?%A5%B3%A5%DE%A5%F3%A5%C9%CC%BE%CC%E4%C2%EA>
- [5] Haruhiko Okumura, 「`upbibtex` で名と姓の間のスペースが消える」, 2020/10/11,  
<https://github.com/texjporq/tex-jp-build/issues/109>