

# BIB<sub>T</sub>E<sub>X</sub>ing : BIB<sub>T</sub>E<sub>X</sub> の使い方\* †

Oren Patashnik(訳：松井正一)

January 31, 1988(翻訳版：1991年1月1日)

## 1 概要

[この文書は BIB<sub>T</sub>E<sub>X</sub> version 1.00 が完成した時に拡充される。タイプミス、欠落、不正確な部分、特に、分かりにくい部分について、著者まで連絡されたい (patashnik@SCORE.STANFORD.EDU, 日本語および jBIB<sub>T</sub>E<sub>X</sub> に関する部分は matsui@denken.or.jp)。改善に関する意見を歓迎する。]

これは BIB<sub>T</sub>E<sub>X</sub> version 0.99a<sup>1</sup> の利用者のものであり<sup>2</sup>、参考文献スタイルの設計者(製作者)はこの文書を読んだ後、製作者のためだけに書かれている “Designing BIB<sub>T</sub>E<sub>X</sub> Styles” [4] を読む必要がある。

本稿は以下の3節からなる。第2節で BIB<sub>T</sub>E<sub>X</sub> の version 0.98i と 0.99a の相違点を、各々に対応する標準スタイルファイルの相違点とともに述べる。第3節は L<sup>A</sup>T<sub>E</sub>X book [2] の付録 B.2 の修正であり、第4節で他の文書には書かれていない一般的事項、特殊事項について述べる。本稿の読者は L<sup>A</sup>T<sub>E</sub>X book の関連する節に精通していることが望まれる。

この文書は BIB<sub>T</sub>E<sub>X</sub> を走らせる作業を助けるための入力例としての役割も持つ。多くの文書と同じくこの文書も参考文献の引用を含んでいる。参考文献リストを作るためには、この文書自身を処理するためには、先ず (aux ファイルを作るために) L<sup>A</sup>T<sub>E</sub>X を実行し、次に (bb1 ファイルを作るために) BIB<sub>T</sub>E<sub>X</sub> を実行し、最後に L<sup>A</sup>T<sub>E</sub>X をもう2回実行する (1回は bb1 ファイル中の情報を取り込むためであり、もう1回は前方参照を解決するためである)。めったに起こらないことではあるが、さらにもう1回 BIB<sub>T</sub>E<sub>X</sub>/L<sup>A</sup>T<sub>E</sub>X を走らせなければならないこともある。

BIB<sub>T</sub>E<sub>X</sub> version 0.99a は、closed 参考文献書式を標準としている。L<sup>A</sup>T<sub>E</sub>X version 2.09 とともに使う必要がある<sup>3</sup>。open 参考文献書式にするためにはオプション文書スタイルの openbib を使う必要がある。(open 書式では主ブロックの間で改行が行われるが、closed では行われない。)

**注意：**BIB<sub>T</sub>E<sub>X</sub> 0.99a では古いスタイルファイルは使えない、同様に BIB<sub>T</sub>E<sub>X</sub> 0.98i では新しいスタイルファイルは使えない。(しかし新しい BIB<sub>T</sub>E<sub>X</sub> で古いデータベースファイルを使うことはできる。)

**インプリメンタへの注意：**BIB<sub>T</sub>E<sub>X</sub> には参考文献スタイルファイルの入力ディレクトリのための TEXINPUTS: と、データベースファイルの入力ディレクトリのための BIBINPUTS: の2つの論理名<sup>4</sup> が用意されている。

**jBIB<sub>T</sub>E<sub>X</sub> での注意点：**漢字コード系は BIBTERMCODE, BIBFILECODE の2つの環境変数で変更可能であり、j<sub>T</sub>E<sub>X</sub> の \kanjiterminaltype, \kanjifiletype に指定するのと同じものを指定する。UNIX 版ではサイト毎に標準コードが設定されているはずであり、MS-DOS 版では SJIS が標準である。

\*翻訳の部分は原著者の許可を得て配布するものであり、翻訳について、jBIB<sub>T</sub>E<sub>X</sub> に関する記述については松井に問い合わせられたい。

† jBIB<sub>T</sub>E<sub>X</sub> (version 0.30) の説明も含む。

<sup>1</sup> 訳注：version 0.99a 以降の意味であり、jBIB<sub>T</sub>E<sub>X</sub> の基になっている 0.99c も含まれる。

<sup>2</sup> 訳注：jBIB<sub>T</sub>E<sub>X</sub> 利用者のためでもある。jBIB<sub>T</sub>E<sub>X</sub> については [3] も参照されたい。

<sup>3</sup> 訳注：jBIB<sub>T</sub>E<sub>X</sub> は jL<sup>A</sup>T<sub>E</sub>X とともに使う。

<sup>4</sup> 訳注：UNIX/MS-DOS ではそれぞれ TEXINPUTS と BIBINPUTS の環境変数

## 2 変更点

本節では BibTeX versions 0.98i と 0.99a の違いと、各々のバージョンに対応するスタイルの違いについて述べる。0.98i と 0.99a の間には多くの相違点があるが 0.99 と 1.00 の間ではもつとずっと少なくなるであろう。

### 2.1 BibTeX の新機能

以下に BibTeX の新機能とその使い方を列挙する<sup>5</sup>。

1. 個々の文献毎に `\cite`, `\nocite` を明示的に指定しなくても、`'\nocite{*}'` コマンドひとつで、データベース中のすべての文献を並べたリストを作ることができる。これは、このコマンドを使った場所で、データベース中のすべてのエントリをその順番で `\nocite` することに相当する。
2. フィールド (あるいは `@STRING` の定義中で) 文字列の連結を指定できる。例えば、次のように定義しておくと、

```
@STRING( WGA = " World Gnus Almanac" )
```

別々ではあるが似たような title エントリを簡単に作れる。

```
@BOOK(almanac-66,  
  title = 1966 # WGA,  
  . . .  
@BOOK(almanac-67,  
  title = 1967 # WGA,
```

次のような使い方も可能である。

```
month = "1~" # jan,
```

この時、スタイルファイル中での `jan` の省略形の定義によって異なるが、`bb1` ファイル中では `'1~Jan.'` あるいは `'1~January'` のようになる。(フィールドのデータ長には上限があるが) `'#'` によって任意の数の文字列を連結できる。`'#'` の前後には、スペース<sup>6</sup> か改行を置くことを忘れてはならない。

3. BibTeX に新たな相互参照機能を付け加えた。これを例にそって説明する。文書中に `\cite{no-gnats}` と指定し、データベース中に次の2つのエントリがあるとしよう。

```
@INPROCEEDINGS(no-gnats,  
  crossref = "gg-proceedings",  
  author = "Rocky Gneisser",  
  title = "No Gnats Are Taken for Granite",  
  pages = "133-139")  
.  
.  
@PROCEEDINGS(gg-proceedings,  
  editor = "Gerald Ford and Jimmy Carter",  
  title = "The Gnats and Gnus 1988 Proceedings",  
  booktitle = "The Gnats and Gnus 1988 Proceedings")
```

<sup>5</sup> 訳注: jBibTeX の機能でもある。

<sup>6</sup> 訳注: 漢字コードのスペース (空白) ではダメ!

この時、次に述べる2つのことがおこる。第1に、特別な意味を持つフィールド `crossref` は、`no-gnats` エントリで欠落しているフィールドを、引用している `gg-proceedings` から継承することを  $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$  に伝える。先の例でいえば `editor` と `booktitle` の2つを継承する。ただし、標準のスタイルでは `booktitle` フィールドは `PROCEEDINGS` 型のエントリでは意味を持たない。この例の `gg-proceedings` エントリの `booktitle` フィールドは、これを参照しているものがこのフィールドを継承できるようにするためだけのものである。この会議の論文がどんなに多くデータベース中にあっても、この `booktitle` フィールドは1回だけ書けばよい。

第2に、`gg-proceedings` エントリを `\cite` あるいは `\nocite` で引用している2つ以上の論文が参照していると、このエントリ自身が `\cite` あるいは `\nocite` されていなくても、 $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$  は自動的に `gg-proceedings` エントリを参考文献リストに加える。つまり `no-gnats` の他に1つ以上の論文が `gg-proceedings` を参照していると、自動的に参考文献リストに加えられる。

ただし、上記の結果となることを保証するためには、相互参照されているエントリは、それを引用しているすべてのエントリよりデータベースファイル中で後ろになければならない。従って、相互参照されているエントリは最後に置くことになる。(さらに、残念なことだが、ネストされた相互参照はうまく扱えない。すなわち、相互参照している先がさらに他を参照している場合にはうまく動作しないことがある。) 最後の注意：この相互参照機能は、現在でも利用できる、古い  $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$  の参照の機能とは全く関係がない。すなわち、次のような使い方は新しい機能から何の影響も受けない。

```
note = "Jones \cite{jones-proof} improves the result"
```

- $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$  はアクセント付き文字 (accented character) を扱えるようになった。例えば次のような2つのフィールドからなるエントリがあり、

```
author = "Kurt G{\\"o}del",
year = 1931,
```

参考文献スタイルとして `alpha` を使ったとすると、 $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$  はこのエントリのラベルをあなたの望み通りに `[Göd31]` とする。このような結果を得るためには `{\"o}` あるいは `{\\"o}` のようにアクセント付き文字列を中括弧 (ブレース; `{}`) で囲んでやる必要がある。さらに、このために使う中括弧は、フィールドやエントリの区切り以外の中括弧の中に含まれてはならないし、中括弧の中の一番最初の文字はバックスラッシュでなければならない。従ってこの例でいえば `{G{\\"o}del}` でも `G{\\"o}del` でもうまくゆかない。この機能は  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  book の表 3.1 と表 3.2 にある、バックスラッシュの付かない外国文字 (foreign symbols) を除く、すべてのアクセント付き文字を扱うことができる。この機能は利用者が定義する「アクセント付き文字」でも使える。すぐ後で例を示す。ラベルの文字数を数える場合には、 $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$  は中括弧の中の文字列をまとめて1文字と数える。

- $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$  はハイフン付きの人名も扱うことができる。例えば、次のようなエントリ、

```
author = "Jean-Paul Sartre",
```

で `abbrv` スタイルを使うと、結果は `'J.-P. Sartre'` となる。

- データベースファイル中に `@PREAMBLE` を書くことができる。コマンドは、文字列からなり、名前と等号記号がないことを除けば `@STRING` と同じである。以下に例を示す。

```
@PREAMBLE{ "\newcommand{\noopsort}[1]{ "
            # "\newcommand{\singleletter}[1]{#1} " }
```

(ここでも文字列連結の機能が使われていることに注意しよう)。標準のスタイルでは、ここに書いた文字列(多くの場合 L<sup>A</sup>T<sub>E</sub>X のマクロ)をそのまま bb1 ファイルに書き出す。上に示した `\noopsort` コマンドを例に、使い方を示す。

ソーティングする(アルファベット順に並べる)ことを考えよう。Bib<sub>T</sub>E<sub>X</sub> は割とうまく並べてくれるが、条件によっては Bib<sub>T</sub>E<sub>X</sub> が混乱することがある。データベースの中に同一著者による 2 巻からなる本のエントリがあり、参考文献リストでは第 1 巻が第 2 巻の直前に並んで欲しいとしよう。さらに第 1 巻には 1973 年に発刊された第 2 版があり、第 2 巻には第 1 版しかなくその発刊は 1971 年であったとしよう。標準スタイル `plain` は著者名と発刊年でソートキーを作り、第 2 巻の方が発刊が早いために、Bib<sub>T</sub>E<sub>X</sub> を助けてやらない限り、第 2 巻の方が先にならなってしまう。このためにはこの 2 つの `year` フィールドを次のようにする。

```
year = "{\noopsort{a}}1973"
. . .
year = "{\noopsort{b}}1971"
```

`\noopsort` の定義から、L<sup>A</sup>T<sub>E</sub>X は年としては本当の年以外は印刷しない。しかし Bib<sub>T</sub>E<sub>X</sub> は `\noopsort` の指定が 'a', 'b' を修飾するものとみなすので、ソートの時にはあたかも 'a1973' と 'b1971' が年のように扱う。'a' は 'b' より前なので、望み通り第 1 巻が第 2 巻の前に並ぶ。ところで、もしこの著者の別の著作がデータベース中にある場合には他の著作との関係を正しくするために `{\noopsort{1968a}}1973` と `{\noopsort{1968b}}1971` といったように書かねばならないこともある。(ここでは第 1 巻の第 1 版の発刊年が 1968 年であることから、これで正しい場所に並ぶものと考えている。)

使える `@PREAMBLE` コマンドの数には上限があるが、`@PREAMBLE` の数を 1 つのデータベースにつき 1 つにしておけば、上限を超えることはない。したがって、項目 2 で述べた、文字列の連結機能を使えばこの制限は大した問題ではない。

7. Bib<sub>T</sub>E<sub>X</sub> のソーティング算法は安定 (stable) なものとなった。つまり、2 つの異なるエントリが同一のソートキーを持つ場合には、引用順に並ぶようになった。(参考文献スタイルがこれらのソートキー、通常は著者名の後に年と表題を付加したものを、を作る<sup>7</sup>)
8. Bib<sub>T</sub>E<sub>X</sub> はファイル名の小文字/大文字変換を行わないようになった。これにより、例えば UNIX システム上に Bib<sub>T</sub>E<sub>X</sub> をインストールするのが簡単になった。
9. コマンドラインの `aux` ファイル名の処理を行うコードを付加することが簡単に行えるようになった。

## 2.2 標準スタイルの変更点

本節では、一般利用者に関係する標準スタイル (`plain`, `unsrt`, `alpha`, `abbrv`) の変更点について述べる<sup>8</sup>。スタイル作成者に関係した変更点は “Designing Bib<sub>T</sub>E<sub>X</sub> Styles” [4] に述べられている。

<sup>7</sup> 訳注：日本語用のスタイルでは、`yomi` というフィールドがあればそれを著者名の代わりに使うことで、読みのアルファベット順に並ぶようになっている。

<sup>8</sup> 訳注：対応する日本語用のスタイルは各々 `jplain`, `junsrt`, `jalpha`, `jabbrv` である。

1. 一般にソーティングは先ず“author”で、次に年、最後に表題で行う。古いバージョンでは年が使われていなかった。(しかし alpha スタイルでは先ずラベルでソートした後、上記のソートを行う)。author の前後に引用符が付いているのは、editor, organization などの他のフィールドが使われることもあるからである。
2. 多くの不必要なタイ (~) を取り除いた。これにより、参考文献リストをフォーマットした時に、L<sup>A</sup>T<sub>E</sub>X から出力される ‘Underfull \hbox’ の数が少なくなった。
3. イタリック指定 (`{\it ...}`) を強調指定 (`{\em ...}`) に置き換えた。出力は変化しないはずである。
4. alpha スタイルでは、ラベルを作る際に著者名のいくつかを省略したことを示す記号を ‘\*’ から ‘+’ に変更した。しかし、以前の形式の方が好き、あるいはこんな記号が不要な場合でもスタイルファイルの修正は不要である。alpha スタイルが (`\thebibliography` environment の直前に) `bb1` ファイルに書き出す `\etalchar` コマンドを、前節の第 6 項目で述べたように、データベース中の `@PREAMBLE` を使って、L<sup>A</sup>T<sub>E</sub>X の `\renewcommand` を用いて定義し直せば、‘+’ の形式を変更できる。
5. `abbrv` スタイルでは、月名として ‘March’ と ‘Sep.’ の代わりに ‘Mar.’ と ‘Sept.’ を使うようになった。
6. 標準スタイルでは、Bib<sub>T</sub>E<sub>X</sub> の新しい相互参照機能を使っている場合には、他の文献を参照しているエントリに対しては、参照先に記述されているフィールドのほとんど全てを省略し、`\cite` を使って参照を示すようになっている。

これは参照元が大部な著作の一部である場合に使う。このような状況としては以下の 5 つが考えられる。(1) `INPROCEEDINGS` (あるいは `CONFERENCE`, どちらも同じこと) が `PROCEEDINGS` を参照している; (2) `BOOK`, (3) `INBOOK`, あるいは (4) `INCOLLECTION` が `BOOK` を参照している (複数巻からなる書物の中のある 1 巻が参照している); (5) `ARTICLE` が `ARTICLE` を参照している。(この場合には、参照されているものは実は論文誌全体であるが、エントリ型として `JOURNAL` がないからこうする。この時には論文誌の `author` と `title` がないという警告メッセージが出力されるが、このメッセージは無視すればよい)。

7. `MASTERSTHESIS` と `PHDTHESIS` エントリ型では、任意フィールドとして `type` を書けるようになった。これにより、

```
type = "{Ph.D.} dissertation"
```

とデータベース中に書くことで、標準の ‘PhD thesis’ の代わりに ‘Ph.D. dissertation’ とできるようになった。

8. 同様に `INBOOK` と `INCOLLECTION` エントリ型では、任意フィールドとして `type` を書けるようになった。これを使えば、次のようにデータベースに書いておけば、標準の ‘chapter 1.2’ の代わりに、‘section 1.2’ とできる。

```
chapter = "1.2",
type = "Section"
```

9. `BOOKLET`, `MASTERSTHESIS`, `TECHREPORT` エントリ型では、`BOOK` の `title` のようにではなく、`ARTICLE` の `title` のように、表題をフォーマット (タイプセット) するようになった。
10. `PROCEEDINGS` と `INPROCEEDINGS` エントリ型では、`address` フィールドを出版社、主催者の住所を示すために使うのではなく、会議の開催された場所を示すために使うようになった。出版社、開催者の住所を入れたいのであれば、`publisher` か `organization` フィールドに入れること。

11. BOOK, INBOOK, INCOLLECTION, PROCEEDINGS エントリ型では, volume だけを許すのではなく, volume か number のいずれか一方を許すようになった.
12. INCOLLECTION エントリ型では, series と edition フィールドを許すようになった.
13. INPROCEEDINGS と PROCEEDINGS エントリ型では, volume または number と series を許すようになった.
14. UNPUBLISHED エントリ型では, 日付情報を後に付けて, note フィールドを1つのブロックとして出力するようになった.
15. MANUAL エントリ型では, author フィールドが空であれば, organization フィールドを第1ブロックとして出力するようになった.
16. MISC エントリ型では, 全ての任意フィールド (すなわち全てのフィールド) が空の時には警告メッセージを出力するようになった.

## 2.3 jBIBTEX の標準スタイル

jBIBTEX の標準スタイルとしては plain, alpha, abbrev, unsrt に対応して jplain, jalpha, jabbrv, junsrt が作成されている. また情報処理学会論文誌 tipsj, 情報処理学会欧文論文誌 jipsj, 電子情報通信学会論文誌 tieice, 日本オペレーションズリサーチ学会論文誌 jorsj, 人工知能学会 jsai, ソフトウェア学会誌 jssst 用のスタイルも作成されている. これらのスタイルで行なっている日本語対応は以下の通りである (jipsj は英語なので, 変更は必ずしも日本語対応のためだけではない).

1. 著者名が日本語かどうかで名前のフォーマットの方法を変える.
2. 著作名に日本語が含まれる場合には強調指定を付けない.
3. ページ範囲指定を Page, Pages から p., pp. に変更した. ナンバー指定, ボリューム指定を No., Vol. に変更した.
4. ソーティングキーを作る時には, yomi フィールドがあればその情報を著者名, 編集者名の代りに使う.

## 3 データベースのエントリ

本節は L<sup>A</sup>T<sub>E</sub>X book [2], © 1986, by Addison-Wesley の付録 B.2 の単なる修正版である. 基本構成は同じであり, 細目が少し変更されている.

### 3.1 エントリの型

参考文献をデータベースに登録する場合に一番最初に決めなければならないことは, その型 (種類) である. 完全な決まったやり方はないが, BIBTEX ほとんどすべての文献を合理的に扱うためのエントリの型を用意している.

文献の出版形態が異なればそれが持っている情報も異なる. 論文誌に発表されたものには論文誌の巻数 (volume) と番号 (number) があるが, 本の場合にはこれらは通常意味がない. 従って異なったエントリ型毎に異なったフィールドがある. エントリの型毎にフィールドは以下の3つに分類される.

**必須** これが欠落していると警告メッセージが出力され、まれにはあるが文献リストが変な形に出力される。もし必要な情報が意味を持たないのであれば、間違った型を指定しているのである。必要な情報が意味を持つのであるが、他のフィールドにそれが記述されている場合には警告メッセージを無視すればよい。

**任意** このフィールドは、もしあれば使われるが、何の問題もなしに省略できる。もしこれらのフィールドの情報が読者を助けるのであれば、これを省略しない方がよい。

**無視される** このフィールドは無視される。BIBTEX は必須あるいは任意のフィールド以外は無視するから、bib ファイルには好きなフィールドを書いてよい。参考文献リストには出力されないかもしれないが、文献に関連するすべての情報を bib ファイルに書いておくことよい。例えば、論文のアブストラクトをコンピュータのファイルにとっておくのであれば、bib ファイル中の abstract フィールドに入れておけばよい。bib ファイルはアブストラクトを入れて置くにちょうどよい場所であり、アブストラクト付きの文献リストを作成するスタイルを作ることもできる。

**注意：**フィールド名をミススペルするとそのフィールドは無視される。特に任意フィールドの場合には BIBTEX は警告メッセージを出さないから、タイプミスに注意しなければならない。

以下で各々のエントリの型について説明を行う。同時に、文献リストの項目として並べられる順番に (エントリ型によっては特定のフィールドのあるなしで若干順番が異なることもあるが) 必須フィールド、任意フィールドを示す。これらエントリの型は、Scribe システムのために Brian Reid が採用した van Leunen [5] の分類と同じものである。個々のフィールドの意味については次節で説明する。標準以外のスタイルでは任意フィールドのいくつかを無視することがある。bib ファイル中ではエントリの型の前に@文字を付けるのを忘れてはならない。

**article** 論文誌中の論文、雑誌の記事。必須フィールド：author, title, journal, year. 任意フィールド：volume, number, pages, month, note.

**book** 出版主体が明確な本。必須フィールド：author または editor, title, publisher, year. 任意フィールド：volume または number, series, address, edition, month, note.

**booklet** 印刷、製本されているが、出版者あるいはスポンサーの名前がないもの。必須フィールド：title. 任意フィールド：author, howpublished, address, month, year, note.

**conference** INPROCEEDINGS と同じ。Scribe との互換性のため。

**inbook** 章 (あるいは節などの) 本の一部かつ/または (and/or) 本のあるページ範囲。必須フィールド：author あるいは editor, title, chapter かつ/または pages, publisher, year. 任意フィールド：volume または number, series, type, address, edition, month, note.

**incollection** それ自身の表題を持つ本の一部。必須フィールド：author, title, booktitle, publisher, year. 任意フィールド：editor, volume または number, series, type, chapter, pages, address, edition, month, note.

**inproceedings** 会議録中の論文。必須フィールド：author, title, booktitle, year. 任意フィールド：editor, volume または number, series, pages, address, month, organization, publisher, note.

**manual** マニュアル。必須フィールド：title. 任意フィールド：author, organization, address, edition, month, year, note.

**mastersthesis** 修士論文。必須フィールド：author, title, school, year. 任意フィールド：type, address, month, note.

**misc** 他のどれも当てはまらない時に使う。必須フィールド：なし。任意フィールド：author, title, howpublished, month, year, note.

**phdthesis** 博士論文。必須フィールド：author, title, school, year。任意フィールド：type, address, month, note.

**proceedings** 会議録。必須フィールド：title, year。任意フィールド：editor, volume または number, series, address, month, organization, publisher, note.

**techreport** 学校などで発行されているテクニカルレポートであり、通常は通番を持つ。必須フィールド：author, title, institution, year。任意フィールド：type, number, address, month, note.

**unpublished** 正式には主版されていないが、著者、表題のある著作物。必須フィールド：author, title, note。任意フィールド：month, year.

上に示したフィールドに加えて、各々のエントリ型では任意フィールドとして key があり、スタイルによっては並べ替え、相互参照、あるいは \bibitem を作るのに使われる。また、「著者」情報が欠けているエントリに対しては必ず key フィールドを入れておかないといけない。「著者」は通常は author フィールドに書くが、エントリの型によっては、editor フィールドであったり、あるいは organization フィールドであったりもする(第4節でもっと詳しく説明する)。key フィールドを \cite に現れるデータベースのエントリの最初に書くキーと混同してはならない。この項目が“key”という名前なのは Scribe との互換性のためである。

**JB<sub>B</sub>TEX での注意**：JB<sub>B</sub>TEX の標準スタイルでは以下に示されているフィールドの他に、任意フィールドとして、漢字コード表記された著者名の「読み」を書くためのフィールドとして yomi がある。

## 3.2 フィールド

標準スタイルで認識されるフィールドを以下に示す。標準スタイルでは無視されるが、この他の任意のフィールドを書いてもよい<sup>9</sup>

**address** 通常は publisher などの住所を書く。van Leunen は、大きな出版社に対してはこれを省略することを勧めている。他方、小さな出版社に対しては、読者の便を考えて、完全な住所を書くといい。

**annote** 注釈。標準のスタイルでは使われないが、他の注釈付きの参考文献スタイルで使われる。

**author** 著者名であり、L<sub>A</sub>T<sub>E</sub>X book に説明されているように書く<sup>10</sup>。

**booktitle** 一部分を引用する本の表題。記述の方法については L<sub>A</sub>T<sub>E</sub>X book を参照のこと。エントリ型が本 (book) の場合には title を使う。

**chapter** 章 (あるいは節などの) 番号。

**crossref** 参照する文献のデータベースキー。

**edition** 本の版—例えば “Second”。(日本語以外では) 順序数で書くこと、また最初の文字は上記の例のように大文字のこと。必要であれば標準スタイルが自動的に小文字に変換する。

<sup>9</sup> 訳注：JB<sub>B</sub>TEX の標準スタイル用に yomi フィールドが追加されている。

<sup>10</sup> 訳注：JB<sub>B</sub>TEX では、漢字コード著者名の姓と名の間にはスペース (半角でも全角でも) を置くことが望ましい (スペースがないと jabbrv スタイルなど省略形が基本のスタイルの時に姓だけにならない)。

**editor** 編集者の名前を L<sup>A</sup>T<sub>E</sub>X book の説明のように書く。author フィールドもある場合には、ここには論文が載っている本あるいは論文集の編集者名を書く<sup>11</sup>。

**howpublished** この奇妙な著作がどのようにして出版されたか。最初の文字は大文字でなければならない。

**institution** テクニカルレポートのスポンサー名。

**journal** 論文誌の名前。多くの論文誌の省略形が用意されている。Local Guide を参照のこと。

**key** 著者に関する情報(第4節参照)がない時にソーティング、相互参照、ラベル作成の処理に使われる。このフィールドを、データベースエントリの最初の項目である \cite で使うキーと混同してはならない。

**month** 著作物が出版された月。未出版であれば書かれた月。L<sup>A</sup>T<sub>E</sub>X book の付録 B.1.3 に示されている3文字の省略形を使うこと<sup>12</sup>。

**note** 読者を助ける付加的な情報を記述する。最初の文字は大文字でなければならない。

**number** 論文誌、雑誌、テクニカルレポートあるいは一連の著作の番号。通常、論文誌、雑誌の特定の号は巻数と番号で識別され、テクニカルレポートを発行している機関はレポートに番号を振っている。またシリーズの本にも番号がある。

**organization** 会議のスポンサーあるいはマニュアル manual を出版した機関の名称。

**pages** 42--111, 7,41,73--97, 43+などの(いくつかの)ページあるいはページ範囲(最後の例は後続ページが単純な範囲でないことを示す)。Scribe 互換のデータベースの維持を簡単にするため、標準のスタイルでは7-33のようなダッシュ1つの形式を T<sub>E</sub>X での数値範囲の指定形式である7--33のようなダッシュ2つに自動的に変換する。

**publisher** 出版主体の名称。

**school** 修士論文、博士論文が書かれた(提出された)大学名。

**series** 一連の書物のシリーズ名、セット名。本全体を引用する場合には、title フィールドにその本の表題を書き、任意フィールドである series にシリーズ名あるいは複数巻の書物の題名を書く。

**title** 著作の表題。L<sup>A</sup>T<sub>E</sub>X book の説明のようにタイプセットされる。

**type** テクニカルレポートの種類—例えば“Research Note”。

**volume** 論文誌あるいは複数巻の書物の巻数。

**year** 著作物が出版された年。未出版であれば書かれた年。標準スタイルは、‘(about 1984)’のような、どんな形でも year フィールドに許すが、区切り文字以外の最後の4文字は、通常は1984のような4つの数字から構成されていなくてはならない。

**yomi** jBIB<sub>T</sub>E<sub>X</sub> のみで意味を持つ。著者(編集者)名が日本語文字の時にソート、ラベル作成がうまくいくように漢字コード氏名の代りに使うフィールドであり、ローマ字表記した姓名を著者名などと同じ書き方で記述する(ひらがなで書いて五十音順にソートしたりもできる)。例を示す。

```
author="松井 正一 and 高橋 誠",
yomi="Shouichi Matsui and Makoto Takahasi"
```

<sup>11</sup> 訳注：jBIB<sub>T</sub>E<sub>X</sub> では、編集者名の姓と名の間にはスペースを置くことが望ましい(スペースがないと jabbrv スタイルの時に姓だけにならない)。

<sup>12</sup> 訳注：念のために列挙すると jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec

## 4 ヒント

本節では、他では詳しく説明されていない事項を解説する。簡単なものから順に説明する。その前に先ず口上を述べたい。

参考文献のタイプセットスタイルの選択の余地は少ない。論文誌  $X$  は  $Y$  のスタイルを指定するし、また別の論文誌は別のものを指定する。しかし、選択の余地があるのであれば、plain 標準スタイルを使うことを勧める。このスタイルは、他のものより具体的に生き生きした、より良い著作となると van Leunen [5] は主張している。

他方で、*The Chicago Manual of Style* [1], は引用が本文中に ‘(Jones, 1986)’ のように現れる「著者-年月日形式 (author-date)」を推奨している。しかし、私はこのような方式は関連する/しない情報が散乱するだけでなく、本文が漠然とし、たいしたことではないかも知れないが受動態や曖昧な文が多くなると考える。さらにコンピュータタイプセットの出現により、「これが最も実用的な方法である」といった議論はまったく意味のないものとなったと考える。例えば、*Chicago Manual* の 401 ページの中ごろに述べられている、「plain のようなスタイルの欠点は、タイプした後では、引用文献の追加・削除のためには、参考文献リストの修正だけでなく、本文中の番号の修正が必要なことである」といった主張はまったくの時代錯誤といえる。L<sup>A</sup>T<sub>E</sub>X を使えば欠点ではないことは明らかである。

最後に、「著者名-年月日形式」の論理的欠陥はそのためのプログラムを作成してみると非常に明白になる。例えば、多数の参考文献リストを普通にアルファベット順に並べた時に、‘(Aho et al., 1983b)’ が ‘(Aho et al., 1983a)’ の半ページも後に並ぶことがある。これを修正するともっと悪くなることもある。なんてこった。(悲しいことに、私はそういったスタイルをプログラムしたことがある。阿呆な出版者のせいで、あるいは私の主張に同調せずに、このようなスタイルを使いたいのであれば、この方式のスタイルは Rochester style collection から入手可能である。)

前口上が長くなってしまったが、私の気分は良くなり、血圧も正常に戻った。以下に標準スタイルで BibT<sub>E</sub>X を使う時のヒントを示す(標準以外のスタイルでも多くの事柄があてはまる)。

1. BibT<sub>E</sub>X のスタイル設計言語を使えば、参考文献のリスト作成以外にも、一般的なデータベース操作のためのプログラムを作ることができる。これを使ってデータベース中のすべての文献のデータベースキー/著者の索引を生成するプログラムを作るとは、この言語に精通していれば簡単なことである。どんなツールが用意されているかについては *Local Guide* を参照されたい。
2. 標準スタイルで使っている 13 個のエントリ型でほとんどの文献をうまくフォーマットできるが、13 個だけでなんでもうまくできるわけではない。したがって、エントリ型の使い分けを工夫するとよい(しかし、余りにいろんな型を試してみないとうまくいかないのは、エントリの型が間違っていることが多い)。
3. フィールド名は厳格に考えなくてよい。例えば、出版者の住所を address フィールドではなく、名前とともに publisher フィールドに入れることもできる。あるいは、複雑な形式の文献には note フィールドをうまく使うことで対処できる。
4. 警告メッセージは深刻に考える必要はない。例えば *The 1966 World Gnus Almanac* のように、表題に年が入っている文献では、year フィールドは省略した方がよく、BibT<sub>E</sub>X の警告メッセージは無視してよい。
5. author, editor の名前が余りに多い時には名前リストを “and others” で終りにしておけば、標準スタイルでは “et al.” に自動的に変換される<sup>13</sup>。

<sup>13</sup> 訳注：jBibT<sub>E</sub>X では、漢字コード著者なら “ほか” に変換される。

6. 一般に、BibTeX が大文字を小文字に自動的に変換するのを止めさせたかったら、中括弧 ({ }) で囲めばよい。しかし、左中括弧に続く最初の文字がバックスラッシュであると、変換されることもある。「特殊文字」のところで再び説明する。
7. *Scribe* との互換性のため、データベースファイル中に @COMMENT コマンドを書くことができる。BibTeX はデータベースファイル中ではエントリの内側以外ならどこにでもコメントを書くことを許すから、実際にはこのコマンドは不要である。あるエントリをコメントアウトするには、単にエントリの型の前の@文字を取り除けばよい。
8. 標準スタイルファイル中には計算機科学関係の論文誌の省略形がいくつか入れているが、これはあくまでも例である。これと異なる論文誌の省略形は @STRING コマンドを使って、個人のデータベースで定義するのがよい。そして L<sup>A</sup>T<sub>E</sub>X の \bibliography 指定の最初の引数として、この省略形の入ったデータベースを記述すればよい。
9. 「月」に関しては、自分で月名を書かずに、3文字の省略形を使うのがよい。そうすれば整合的な形式となる。「日」の情報も含めたい場合には month フィールドに書く。例えば、

```
month = jul # "~4,"
```

とすれば、望みの結果が得られる。

10. (参考文献が引用順に列挙される) unsrt スタイルを \nocite{\*} (データベースのすべての文献が列挙される) と共に使う場合には、文書中の \nocite{\*} コマンドの位置で引用文献の並ぶ順番が決まる。第 2.1 節で示した規則によれば、コマンドが文書の先頭であれば、データベース中の順番そのものになり、最後にあれば、\cite あるいは \nocite で明示的に引用した文献が引用順に並び、その後にはそれ以外のものがデータベース中の順番で並ぶ。
11. 学位を授与するのは大学であり学部ではないから、学位論文に対しては、van Leunen は学位の後に学部名を書かないことを推奨している。学部名を書いた方が読者が論文を見つけやすいと考えるのであれば、学部名は address フィールドに書くといい。
12. MASTERSTHESIS と PHDTHESIS のエントリ形は *Scribe* との互換性のために付けた名称であり、MINORTHESIS と MAJORTHESIS の方が良い名前かも知れない。従ってアメリカ以外での学位論文ではこのことを念頭におくこと。
13. ある著者の 2 つの著作の著者名が若干異なる場合の対処の方法。2 つの論文が次のようであったとしよう。

```
author = "Donald E. Knuth"
. . .
author = "D. E. Knuth"
```

2 つのやり方が考えられる。(1) このままにする、(2) これは同一人物であるであることがはっきりと分かっているので、著者の好みの形式 (例えば、'Donald E. Knuth') に統一する。最初の方法では文献の並ぶ順番が正しくなくなるかもしれないし、2 番目の方法では名前を少し変えたことで、誰かが電子的に検索した時にへまをやるかもしれない。私好みの第 3 の方法がある。2 番目の論文のフィールドを次のように変換する。

```
author = "D[onald] E. Knuth"
```

BIB<sub>T</sub>E<sub>X</sub> はかぎ括弧がないとしてソートするし、かぎ括弧によって読者は完全な「名」は元の論文にはなかったと考えるから、前述の問題が解決できる。もちろん ‘D[onald] E. Knuth’ はみっともないという別の問題が生ずるが、私はこの場合には正確さのためには美しさを犠牲にしてよいと考える。

14. L<sub>A</sub>T<sub>E</sub>X のコメント文字 ‘%’ はデータベースファイル中ではコメント文字ではない。
15. 前節でふれた「著者」についてもっと詳しく説明しよう。ほとんどすべてのエントリでは「著者」は単に author フィールドの情報である。しかしながら、BOOK と INBOOK エントリ型では、author に著者がなければ editor フィールドが使われ、MANUAL では、author に著者がなければ organization フィールドが使われ、PROCEEDINGS では、editor に著者なければ organization フィールドが使われる。
16. ラベルの作成の時に、alpha スタイルでは上述の「著者」を使うが、MANUAL と PROCEEDINGS エントリ型では key フィールドが organization フィールドより優先する。これは次のような場合に役立つ。

```
organization = "The Association for Computing Machinery",
key = "ACM"
```

key フィールドがないと alpha スタイルは organization フィールドの情報から 3 文字のラベルを作る。alpha スタイルは ‘The’ を取り除くが、それでもラベルは ‘[Ass86]’ となってしまう、仕組み通りではあるが、有益なものではない。上のように key フィールドを入れることで、もっとよいラベル ‘[ACM86]’ とできる。

しかし、organization フィールドを無効にするために key フィールドが常に必要なわけではない。例えば、

```
organization = "Unilogic, Ltd.",
```

としておけば、alpha スタイルは ‘[Uni86]’ といった合理的なラベルを生成する。

17. 第 2.1 節でアクセント付き文字を説明した。BIB<sub>T</sub>E<sub>X</sub> にとってはアクセント付き文字は、トップレベルの左中括弧の直後の文字がバックスラッシュである中括弧で括られている文字列、すなわち「特殊文字」の特別な場合にすぎない。例えば、

```
author = "\AA{ke} {Jos{\'}{e}} {\'}{E}douard} G{\"}del"
```

には ‘{\'}{E}douard}’ と ‘{\"}’ の 2 つの特殊文字がある。(上の例で、フィールドの区切り文字であるダブルクォートを中括弧で置き換えたものでも同じことである)。一般に BIB<sub>T</sub>E<sub>X</sub> は特殊文字中の T<sub>E</sub>X あるいは L<sub>A</sub>T<sub>E</sub>X コマンドに対しては何の操作もほどこさないが、そうでない場合には操作を**ほどこす**。したがって表題を小文字に変換するスタイルでは

```
The {\TeX BOOK\NOOP} Experience
```

を

```
The {\TeX book\NOOP} experience
```

に変換する。(しかし、‘The’ は表題の最初の単語であるのでそれはそのままである)。

特殊文字はアクセント付き文字を扱うのに役立つ、BIB<sub>T</sub>E<sub>X</sub> にあなたの望み通りの順番で文献を並べさせるのに役立つ。また BIB<sub>T</sub>E<sub>X</sub> は特殊文字をすべてをまとめて 1 文字と数えるから、ラベルの中に文字を

追加するのに役立つ。BibTeX とともに配布されている XAMPL.BIB ファイル中にすべての使い方の例がある。

18. 本節の最後の項目として (author, editor フィールドの中に記述する) 名前について、L<sup>A</sup>T<sub>E</sub>X book の付録 B より少し詳しく説明する。以下では「名前」は個人に対応するものとしよう。(複数の著者を1つのフィールドに入れるには、中括弧に入れなくて、前後にスペースを入れて “and” でつなぐ。ここでは1つの名前だけを考える)。

名前は、姓 (Last), von, 名 (First) と Jr の4つの部分からなり、それぞれは (空かもしれないが) 文字列 (トークン) のリストからなる。空でない名前の姓は空でない、すなわち1つの文字列からなる名前は姓だけのものとなる<sup>14</sup>。

Per Brinch Hansen の名前は次のようにタイプしなければならない。

```
"Brinch Hansen, Per"
```

この名前の名は “Per” の1つのトークンからなり、姓は “Brinch” と “Hansen” の2つのトークンからなり、von, Jr の部分はない。もし次のようにタイプしたとすると、

```
"Per Brinch Hansen"
```

BibTeX は (誤って)、“John Paul Jones” の名トークンが “Paul” であるのと同じように “Brinch” が名のトークンであると考え、2つの名トークンと1つの姓トークンからなるものとする。

もう1つ例を示そう。

```
"Charles Louis Xavier Joseph de la Vall{\`e}e Poussin"
```

この名前には4つの名トークンと、2つの von トークンと、2つの姓トークンがある。von パートは小文字で始まることから、BibTeX は各々の部分がどこから始まり、どこで終わるかがわかる。

一般に中括弧のレベルが0のところでは小文字で始まるものは von トークンとされる。技術的には「特殊文字」は中括弧のレベル0であるから、TeX のコマンド文字列の大文字、小文字を保つようなダミーの特殊文字を使うことで、BibTeX が von トークンとして扱うように/扱わないようにできる。

まとめると、BibTeX は名前の書き方として次の3つの形式を許す。

```
"First von Last"
```

```
"von Last, First"
```

```
"von Last, Jr, First"
```

Jr パートがある場合、あるいは姓が複数の構成要素からなる場合以外では一番最初の書き方を使うのが普通である。

19. jBibTeX では漢字コード名の場合には姓と名の間スペース (半角でも全角でも) を置くことを標準とするが、スペースがなくても、jabbrv スタイルなどで出力される名前が姓のみにならないなどの問題しか生じない。また、複数の氏名を並べる時には `and` の代わりに、全角の句点 “、” あるいは “、” を使うこともできる。

<sup>14</sup> 訳注: 漢字コード表記された名前では、姓と名にスペース (半角でも全角でも) を入れておけば、Family name と Last-token からなるものとされ、スペースがなければ Last-token のみからなるものとされる。

20. アルファベット順でなく、五十音順にソートしたければ `yomi` フィールドに「ひらがな」で「読み」を書いておけばよい。 `jplain`, `jabbrv` でアルファベット表記の著者はアルファベット順に並べ、その後に日本語著者名のものを五十音順に並べたい場合などに、ひらがな表記の読みを使えばよい。

ただし `jalpha` スタイルではラベルをうまく作り出すには工夫が必要である。 `jalpha` スタイルを使った時には、五十音順にソートすることはないであろうが、次の例のようにすればよい。

```
@preamble{ "\newcommand{\noop}[1]{}" }
@BOOK{sym,
  editor="Janusz S. Kowalik",
  title="Coupling Symbolic and Numerical Computing in Expert Systems",
  publisher="North-Holland", year=1986}
@BOOK{dss,
  editor="Clyde W. Holsapple and Andrew B. Whinston",
  title="Decision Support Systems: Theory and Application",
  publisher="North-Holland", year=1987}
@INCOLLECTION{goto,
  author="後藤英一", title="計算機による数式処理とは",
  yomi="{\noop{ごとう}後}藤",
  crossref="reduce", pages="4--6", year=1986 }
@UNPUBLISHED{磯崎,
  author="磯崎 秀樹",title="How To Use {\JLaTeX}",
  yomi="{\noop{いそぎ}磯}崎",
  note="memo for {\JLaTeX}", year=1987}
```

こうしておけば、各々のラベルは [Kow86], [HW87], [後藤 86], [磯崎 87] となり、アルファベット順に [HW87], [Kow86] と並んだ後に、五十音順に [磯崎 87], [後藤 86] と並ぶ。

## 参考文献

- [1] *The Chicago Manual of Style*, pp. 400–401. University of Chicago Press, thirteenth edition, 1982.
- [2] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*. Addison-Wesley, 1986. (邦訳:『文書処理システム L<sup>A</sup>T<sub>E</sub>X』 Edgar Cooke, 倉沢 良一 監訳, 大野 俊治, 小暮 博道, 藤浦 はる美 訳, アスキー, 1990 年).
- [3] 松井正一. 日本語 B<sub>I</sub>B<sub>T</sub>E<sub>X</sub>:jB<sub>I</sub>B<sub>T</sub>E<sub>X</sub>. jB<sub>I</sub>B<sub>T</sub>E<sub>X</sub> C Version キットと共に配布されている文書 (ver.0.20 用, 1989 年), 1991.
- [4] Oren Patashnik. Designing B<sub>I</sub>B<sub>T</sub>E<sub>X</sub> styles. The part of B<sub>I</sub>B<sub>T</sub>E<sub>X</sub>'s documentation that's not meant for general users, January 1988.
- [5] Mary-Claire van Leunen. *A Handbook for Scholars*. Knopf, 1979.