

p 白鳥マニュアル—Wz4でLaTeX

p 白鳥 WEBSITE【よみがえれ白鳥】

平成 14 年 12 月 19 日

水谷 亘

W.Mizutani@aist.go.jp

フリーソフト「p 白鳥」は、LaTeX の入力支援と pLaTeX の外部コマンド起動を行うための Wz editor 4.0(以下 Wz4、最新版 4.00F) のプラグインマクロです。もともとなった白鳥 (HackTeX) は、広瀬雄二さんが作られた Emacs 用の野鳥 (YaTeX) を竹中浩さんが Windows 上で動く Wz editor に移植されたものですが、Wz4 ではマクロが大幅に変更されてしまい、長い間移植されませんでした。

「p 白鳥 (ぱくちょう)」という名は、白鳥の一部 (part) しか移植できなかった小さい (petit) 白鳥であり、野鳥としては機能が足りない仮の (pseudo) 姿であること、pLaTeX に対応していること、Wz4 の Plug-in マクロであり、pdf ファイルの作成、表示を呼び出せること、竹中さんのコードをいっぱい使わせてもらったことからつけました (竹中さん、どうもありがとうございました)。

機能的にはまだ未完成な p 白鳥ですが、その最大の特徴は、必要なファイルが一個だけ (phacktex.txc) ということです。Wz4 のプラグインマクロとしてとても簡単にインストールできます。白鳥にはなかった機能として、dvi2pdf を呼び出して pdf ファイルを作成できますし、ソーススペシャルにも対応しているので、dviout で表示している文書と Wz4 の間のジャンプもサポートしています。Windows 上で TeX を使う時、mule(+野鳥) は敷居が高いなあ、と思われたら是非試してみてください。

目次

1	インストール・アンインストール・バージョンアップ	3
1.1	インストール	3
1.2	アンインストール	3
1.3	バージョンアップ	3
2	コマンド起動 (タイプセットメニュー)	4
2.1	J)LaTeX (タイプセット)	4
2.2	R)egion (領域タイプセット)	4
2.3	B)ibTeX (参考文献リスト)	4
2.4	make(I)ndex (索引)	5
2.5	P)review	5
2.6	Q)uick	5

2.7	ma(T)h-mode	5
2.8	D)vipdf(pdf ファイルの作成)	5
2.9	V)iewPDF	5
2.10	E)xec %#! Command(外部コマンドを実行)	5
2.11	S)pellCheck(スペルチェック)	6
2.12	O)ptions(設定変更ダイアログ)	6
	2.12.1 Prefix key	6
	2.12.2 各種コマンドの設定	7
	2.12.3 log buffer open	7
	2.12.4 Initialize	7
3	補完入力・入力支援	8
3.1	begin 型補完 (^T+b)	8
3.2	end 補完 (^T+e)	8
3.3	環境名などの変更 (^T+c)	8
3.4	large 型補完 (^T+l)	8
3.5	section 型補完 (^T+s)	8
3.6	maketitle 型補完 (^T+m)	9
3.7	補完入力ダイアログ	9
3.8	アクセント補完 (^T+a)	9
3.9	コマンド、環境、対応する括弧の削除 (^T+k または ^T+K)	9
3.10	イメージ補完	9
4	Jump 機能	10
4.1	対応するラベル、ファイルへ (^T+g)	10
4.2	親ファイルへ (^T+^)	10
4.3	dviout 画面へ (^T+G)	10
5	その他	10
5.1	括弧類の入力 (範囲選択にも対応)	10
5.2	選択範囲のコメント化とコメントアウト	11
5.3	勝手に includeonly	11
6	おまけ	11
6.1	LaTeX エラーへの対応	11
6.2	Wz4 標準ファイラー、高速ファイルオープンの問題	12
6.3	Wz4 の TeX 用アウトライン設定 (おすすめ)	12
6.4	Wz4 の TeX キーワード色分け設定	12

1 インストール・アンインストール・バージョンアップ

Windows9X, me, 2000 上で動作確認しています。DOS 窓のコマンドラインで pLaTeX、jBibTeX などが使えるようにインストールされている必要があります。また、dviout をインストールして、dvi ファイルに dviout を関連づけておいてください。dvipdfm が使えると、pdf ファイルを作製できます。

1.1 インストール

必要なファイルは、phacktex.txc だけです。phacktex.txc を Wz4 の Macro ホルダーにコピーし、Wz4 のメニューから

ツール ⇒ マクロ ⇒ TX-C の設定 ⇒ 新規登録

とし、phacktex をプラグインとして使える状態にしてください。その後、一旦すべての Wz4 を閉じてから、*.tex(*.sty) ファイルを読み込めば使えます。

1.2 アンインストール

アンインストールするには phacktex を「TX-C の設定」で削除してください。その他の痕跡として、辞書を Wz フォルダの _wzhistu.dat というファイルに保存しています。完全にアンインストールしたい場合には _wzhistu.dat 内の phacktex_initialized, phacktex_singlecmd, phacktex_environ, phacktex_fontsize, phacktex_section という 5 つのエントリーを消去してください。残っていても悪さはしないと思います。他にユーザヒストリー機能を使っていないようでしたら、_wzhistu.dat ごと消してしまえば簡単でしょう。Permanent 変数を使っていますので、マクロのホルダーに phacktex.cfg というファイルができます。phacktex をアンインストールするときは削除してください。(白鳥あとを濁さず) この他にはインストール、アンインストールによって設定を変えることはありません。

1.3 バージョンアップ

p 白鳥のバージョンアップは基本的にマクロの変更と同じ手順でできます。

新しい phacktex.txc を Wz4 の Macro ホルダーにコピーし、Wz4 で phacktex.txc を開いてください。その状態でメニューから

ツール ⇒ マクロ ⇒ コンパイル

とし、コンパイルが成功すると新しい phacktex がインストールされます。その時、警告がでると思いますが、一旦すべての Wz4 を閉じてから、*.tex(*.sty) ファイルを読み込めば使えるようになります。

エラーが出た場合、

ツール ⇒ マクロ ⇒ TX-C の設定

でプラグインの状態をチェックしてください。コンパイルに成功しても一度エラーになると状態が「エラー」となっていることがあります。その場合は状態ボタンを押して、OKを選び、Wz4 を閉じてファイルを読み直します。

2 コマンド起動(タイプセットメニュー)

LaTeX 関連コマンドの多くは、コンソール画面で起動するので、ソースを編集し直すたびに Windows のコマンド (DOS) 窓に移動するのは効率が良くありません。そこで p 白鳥は、よく使うコマンドを Wz4 の中から呼び出して起動します。p 白鳥をインストールし、拡張.tex のファイルを編集している状態で、“ ^T T ”というキー操作をすると現れるタイプセットメニューの説明をします。(p 白鳥を起動するキー (Prefix key)“ ^T(ctrl-T) ”は変更可能です。)

2.1 J)LaTeX(タイプセット)

マウスでクリックするか、カーソルの上下で選択してリターンするか、キーボードで “ j ”を押すと、platex を起動して dvi ファイルを作ります。(選択する操作は他のメニュー項目も同様です。)

2.2 R)egion(領域タイプセット)

ソースファイルの一部がちゃんと書けているかチェックするために、その範囲をタイプセットし、dviout を起動して表示します。

タイプセットする領域を指定する方法は二つあります。

1. %#BEGIN-%#END で挟む (指定し直すまで領域は固定される)

^T+% で、選択範囲を %#BEGIN-%#END で囲み、他の %#BEGIN, %#END を消去します。

2. マウスなどで反転させて選択する (そのたびに指定しなければならない)

以下の説明で範囲選択とか選択範囲に対する操作が出てきますが、これは、エディタ上で文字をマウスでドラッグしたり、Shift キーを押しながらカーソル移動させることで **このように反転表示** させた部分のことです。

2.3 B)ibTeX(参考文献リスト)

BibTeX を起動して参考文献リスト (bbl ファイル) を作ります。

2.4 make(I)ndex(索引)

mindex を起動して、索引を作ります。

2.5 P)review

すでに作成してある dvi ファイルを dviout で表示します。

2.6 Q)uick

タイプセットと表示をひとつのコマンドで実行できます。ちょっと直しては、プレビューするのに便利。

2.7 ma(T)h-mode

\$\$内など math 環境で、;、:を使って呼び出すギリシャ語、数学記号の入力支援(イメージ補完)をオン/オフします。

2.8 D)vipdf(pdf ファイルの作成)

すでに作成してある dvi ファイルを元にして、dvi2pdf を起動して pdf ファイルを作成します。AcrobatReader がすでに開いている場合は、表示している書類を全部閉じてから新しく作成した書類を表示します。この機能を使うためには、dvi2pdf、AcrobatReader がインストールされていて、AcrobatReader が pdf ファイルに関連づけられている必要があります。

2.9 V)iewPDF

AcrobatReader を起動します。すでに開いている書類がある場合は、全部閉じてから表示します。

2.10 E)xec %#! Command(外部コマンドを実行)

ファイルの頭から検索して最初に出会う %#! で始まる行に書かれている外部コマンドを実行します。このとき、%FN は現在編集しているファイル名(拡張子抜き)になります。例えば、

```
%#! dvi2pdf -l %FN
```

```
%#! dvipsk -P pdf %FN.dvi
```

野鳥では%#!で始まる行の最後の単語を親ファイルと推測するようになっていますが、編集以外以外のファイルをコンパイルしたくない場合は、この行のどこかに%NOPFと書いておくと、親ファイルはないと明示的に示すことができます。

逆に明示的に親ファイルを示すには、

```
%#! %PFN: parent_file_name
```

と書いておくと、タイプセット命令で `parent_file_name` をコンパイルします。%PFN: は、コマンドを DOS に渡すときには無視されます。(%PFN:の後ろに書かれたファイル名は残ります)。

大きな文書を分割して作成する場合、親ファイルで `¥input`, `¥include` として入力するファイルを指定して呼び出します。実際には呼ばれる側(子ファイル)を編集することが多いのですが、タイプセットする場合は、親ファイルに `tex` をかけなければなりません。子ファイルから親ファイルがどれだかわからないので、%#! 行の中で指定するようになっていきます。

2.11 S)pellCheck(スペルチェック)

`ispell` を起動してスペルチェックをします。テンポラリファイルとして、`_spell...i` を作製します(泥臭い方法)。範囲選択をしていると範囲内の単語だけをチェックします。この機能を使うためには、`ispell` がインストールされている必要があります。

2.12 O)ptions(設定変更ダイアログ)

Prefix Key、外部アプリケーション、log buffer などは、この設定変更ダイアログから変更できます。Permanent 変数を使っていますので、マクロのホルダーに `phacktex.cfg` というファイルができます。上記設定を初期化する場合、あるいは `phacktex` をアンインストールするときは削除してください。

2.12.1 Prefix key

p 白鳥を起動するキー(Prefix key)は変更可能です。デフォルトは、(Prefix key)“ ^T(ctrl-T) ”になっています。野鳥(YaTeX)の標準では“ ^C ”なのですが、Windows ではコピーに使われるためです。Prefix Key の変更後は、一旦すべての Wz4 を閉じてください。次に*.tex(*.sty) ファイルを読み込めば更新されます。

Prefix Key の指定には、

- ^: Ctrl
- %: Alt

- +: Shift
- F6

などが使えます。Alt+英文字を指定する場合は、Wz のメニューから

表示 → キャプション → キー

で、「Alt キーをコマンドに割り当てる」をチェックしてください。Prefix Key を何にしたか忘れてしまったときは、マクロのホルダーにある phacktex.cfg を削除して立ち上げると、初期状態 (^T) に戻ります。

2.12.2 各種コマンドの設定

起動するコマンドは DOS 窓で編集している文書のディレクトリから呼び出せるよう、path が設定してある必要があります。デフォルトで、platex -src とソーススペシャル・オプションをつけています。ソース・スペシャルを使わない人は -src をはずしてください。必要なときだけ使う場合は、ソース (tex ファイル) に

```
%#! platex -src %FN %NOPF
```

と書いて、タイプセットメニューから Exec %#! Command で使うこともできます。

ispell にも -t -j -d american というオプションがついていますが、この意味については ispell の help を参照してください。

2.12.3 log buffer open

このボックスをチェックしておくと、Wz4 の編集画面の下に log 窓 (ログバッファ) が開き、コマンド実行時の log を表示します。^T+(アポストロフィ) でログバッファに表示してあるエラー行にジャンプします。ログバッファ内では、p, n, .(ピリオド), (スペース) を入力すると、前後のエラーメッセージにジャンプしたり、本体のエラー行へ移動します。q, ESC でログバッファを閉じます。

2.12.4 Initialize

Initialize のボタンは、本体 (phacktex.txc) の辞書に新しいコマンドを手で追加したとき、追加したコマンドを含めて初期化するためのものです。_wzhistu.dat を消去するか、Initialize をしないかぎりもとの履歴を読み込んでしまいます。逆に、これまで_wzhistu.dat に記録していたコマンドは消えてしまうので、注意してください。

3 補完入力・入力支援

LaTeX のコマンドには、長いものや複雑な構造を持っているものがあるので、正確に覚えて入力したり、削除するのは大変です。p 白鳥はその手間を軽減するためのいろいろな機能を持っています。

begin 型コマンドとは \forall begin ~ \forall end の形式のコマンド、
section 型コマンドとは \forall section{ 題 } のように引数を取るコマンド、
large 型コマンドとは { \forall tt } のような中括弧で囲まれたフォント/サイズ
指定子、
maketitle 型コマンドとは \forall maketitle のように引数を伴わないコマンドを
指します。
(広瀬雄二「『野鳥』クイックリファレンス」より)

3.1 begin 型補完 (^T+b)

補完入力ダイアログが現れます。リストで選ぶかコマンド名を途中まで入力して、スペースキーを押して下さい。マッチしたコマンドが表示されます。リターンで入力を確定するか ESC でダイアログは消えます。

範囲選択した状態で ^T+b すると、その範囲を挟むように \forall begin ~ \forall end の型のコマンドが挿入されます。

3.2 end 補完 (^T+e)

\forall begin{***} に対応する \forall end{***} がないとき、カーソル位置に自動的に \forall end{***} を挿入します。

3.3 環境名などの変更 (^T+c)

\forall begin{***}、 \forall end{***} 行で ^T+c と入力すると、対応する \forall begin、 \forall end の***の部分を変更することができます。

3.4 large 型補完 (^T+l)

補完入力ダイアログでコマンドを入力できます。

3.5 section 型補完 (^T+s)

補完入力ダイアログでコマンドを入力できます。

3.6 maketitle 型補完 (^T+m)

補完入力ダイアログでコマンドを入力できます。

3.7 補完入力ダイアログ

p 白鳥の補完入力ダイアログでコマンド名を途中まで入力して、スペースキーを押して下さい。マッチしたコマンドが表示されます。学習機能もあります。辞書から選択したコマンド名をリストの上に持ってくる履歴機能もあります。辞書をユーザ履歴として Wz フォルダの `_wzhistu.dat` というファイルに保存しています。

3.8 アクセント補完 (^T+a)

ドイツ語やフランス語などで使うウムラウト、アクセントなどの入力を支援します。メニューが表示されるので、入力方法を覚えてなくても調べなくて済みます。

3.9 コマンド、環境、対応する括弧の削除 (^T+k または ^T+K)

コマンド、環境の `\begin`、`\end` 行の上にカーソルを合わせて `^T+k` または `^T+K` で削除することができます。`^T+k` ではコマンド、環境のみ、`^T+K` では内容も含めて削除されます。`{...}`、`[...]`、`(...)` の括弧の上でも同様の動作をします。

3.10 イメージ補完

`math mode` が ON の状態で、数式環境で `;()` を入力するとイメージ補完 (ギリシャ文字、数学記号の入力支援) のキー・シーケンス待ちになります (ステータスバーに `Math` か `Greek` と表示されます)。この状態で `TAB` キーを押すとキー・シーケンスとコマンドのリストを表示します。入力したいコマンドを選択してリターンキーを押せば、そのコマンドが入力されます。

例えば、`→` を入力したいときは、`;` に続いて `-->` (マイナス、マイナス、より小さい) とタイプすると、下の行にイメージが表示されますので、リターンで決定すると、`'\longrightarrow'` と入力されます。ギリシャ文字の場合は、`:` に続いて `b` と入力すると `'\beta'` と入力されます。

`math mode` は、タイプセットメニュー `^T+T+T` でトグルします (数式環境で `:` ; としでも、そのまま入力されるようになります)。数式環境でも `\label{...}` の中では `:` はそのまま入ります (`\label{eq:1}` などと入力する場合があるので)。

`^T+;()` では常にイメージ補完に入ります。この場合は、`TAB` による一覧表示はできません。

4 Jump 機能

4.1 対応するラベル、ファイルへ (^T+g)

対応する \forall ref と \forall label 間、環境の \forall begin と \forall end、 \forall bibitem と \forall cite 間などでジャンプします。bbl ファイルにもジャンプできます。

4.2 親ファイルへ (^T+^)

親ファイルへジャンプします。親ファイルの中に書かれた

```
 $\forall$ input{child_file}
 $\forall$ include{child_file}
```

の上で ^T+g とすると child_file にジャンプします。

4.3 dviout 画面へ (^T+G)

ソーススペシャルが設定されていれば、dviout と DDE 通信してソースファイル (*.tex) のカーソル位置が含まれるページを表示します。この機能を使うためには、-src オプションをつけて dvi ファイルを作成しておく必要があります。

dviout の

Option → Setup Parameters → Common

の src の欄に、

```
c: $\forall$ wz4 $\forall$ wzeditor.exe^s/j%d "%s"
```

という感じ (c: \forall wz4 \forall は、実際の path に変えてください) で Wz を指定しておけば dviout の表示画面からソースファイルの対応箇所にジャンプできますので、お試し下さい (方法は、dviout の画面でソースが見たい部分を Shift を押しながらマウスの左クリック。下に出てきた補助画面から、→Src をクリックする。詳しくは dviout の Help 参照)。

5 その他

p 白鳥をインストールすると、入力時の動作が変わります。最初は戸惑うかも知れませんが、慣れると便利です。

5.1 括弧類の入力 (範囲選択にも対応)

日本語入力がオフの状態では、(, {, [を入力すると、自動的に括弧が閉じてカーソルが括弧の中に移動します。数式環境の \$ や \forall [も自動的に環境が閉じられ、カーソルが環境内に移動します。選択してから入力すると、選択部分を囲むように括弧が入力されます (\forall [は範囲選択には対応していません)。

5.2 選択範囲のコメント化とコメントアウト

複数の行を選択した状態で `^T+>` とすると選択開始部分と行頭に `%` が挿入されてコメントになります。複数のコメント行の冒頭から `%` を取り除くには、選択して `^T+<` とします。

5.3 勝手に includeonly

親ファイルが指定してある子ファイルを編集していて、親ファイルが編集集中の子ファイルを `\includeonly` していない時、タイプセットの前に自動的に (勝手に) 親ファイルの `\includeonly` を書き換えて編集集中の子ファイルをタイプセットできるようにします。「勝手に includeonly」が起動すると、ダイアログボックスが現れて、

```
A)dd R)eplace C)omment Cancel
```

と動作を選択するように催促します。Add を選ぶと、`\includeonly` の引数に現在編集集中のファイル名を書き加えます。他の動作についてもお試しください。

通常、親ファイルは複数の子ファイルを読み込んでいます。特定の子ファイルだけをタイプセットするために、親ファイルの中で `\includeonly` を使って子ファイルを指定します。しかし、現在編集しているファイルが親ファイルの `\includeonly` で指定されていないとタイプセットされません。ただし、`\includeonly` が指定されてなければ、すべての子ファイルがタイプセットされるので、この機能は働きません。また、すでに編集集中のファイルが `\includeonly` で指定されている場合もこの機能は働きません。

6 おまけ

6.1 LaTeX エラーへの対応

厳密には p 白鳥の説明ではありませんが、タイプセットコマンドを起動して DOS 窓の中で下のようなメッセージが出て止まってしまうことがよくあります。

```
This is pTeX, Version p2.1.11, based on TeX, Version 3.14159 (SJIS) (Web2C 7.3.3)
(./texput.tex
pLaTeX2e <2000/11/03>+0 (based on LaTeX2e <2000/06/01> patch level 0)
(c:/pTeX/share/texmf/inputs/article.cls
Document Class: article 1999/09/10 v1.4a Standard LaTeX document class
(c:/pTeX/share/texmf/tex/latex/base/size10.clo)) (./texput.aux)
! Misplaced alignment tab character &.
1.13 ...nced Materials \& Devices Lab, Toshiba R &
D Center
?
```

‘?’ の後ろに ‘X’(大文字のエックス) を打ち込んでリターンすると、Wz4 に制御が戻ります。

ちなみに上の例では、tabular 環境などで使われる ‘&’ が生のまま出てきたために LaTeX が怒っているので、エスケープしてあげれば (¥をつけて ‘¥&’ にする) 解決します (同じセンテンスの前の部分ではエスケープしていたのに、二個目をし忘れていて)。

ログバッファが開いていれば、^T+’(アポストロフィー) でエラーが生じた行にジャンプします。

6.2 Wz4 標準ファイラー、高速ファイルオープンの問題

Wz4 の標準ファイラーからファイルを開くと、外部コマンド起動の後でファイラー画面がアクティブになってしまう不具合がありました。Wz4 のメニューで

表示 ⇒ オプション ⇒ ファイルを開く動作

を高速から標準に戻したら不具合は出なくなりました。

6.3 Wz4 の TeX 用アウトライン設定 (おすすめ)

これは Wz4 の使い方ですが、

表示 ⇒ 文書の設定 ⇒ 見出し

で、

見出し 1 ¥section

見出し 2 ¥subsection

見出し 3 ¥includegraphics

見出し 4

のような要領で設定しておく、Shift+ESC すると文書の構造が見渡せて便利です。

この他、アウトライン機能の使い方としては、bib ファイルを編集するとき、見出し語を@(アットマーク)にするのもおすすめです (項目ごとの移動ができます)。

6.4 Wz4 の TeX キーワード色分け設定

これも Wz4 の使い方ですが、メニューから

表示 ⇒ 文書の設定 ⇒ 色分け

に入ります。ここで、キーワードのキーワード欄は空欄にして、キーワードファイルを指定します。

キーワードファイルはWZフォルダに、拡張子 .TKW のファイルとして作成しておきます。キーワードファイルの拡張子を除いたファイル名を、C.TKW なら C のように指定します。(Wz4のHELPより)

例えば、tex.tkw というファイルを作って(キーワードファイルの欄には‘tex’と入力します)色分けしたいキーワードを

¥item
¥bibitem
¥¥
¥footnote
...

のように列挙しておくとしソースが見やすくなるかもしれません。