「単純な下線」パッケージ uline−−

（2007/07/09 版）

しっぽ愛好家

2009 年 11 月 2 日

1　基本的な使用法

この uline−− パッケージでは，基本的なユーザマクロとして次の 3 個のマクロを提供しています．

- `\uline`: `\uline` の引数としたテキストに下線（縦組み時には左側の傍線）をつけるマクロ
  - `\uline{サンプルテキスト}` → <u>サンプルテキスト</u>
- `\mline`: `\mline` の引数としたテキストに打ち消し線をつけるマクロ
  - `\mline{サンプルテキスト}` → ~~サンプルテキスト~~
- `\oline`: `\oline` の引数としたテキストに上線（縦組み時には右側の傍線）をつけるマクロ
  - `\oline{サンプルテキスト}` → <u>サンプルテキスト</u>

また，線種を破線・波線・点線に変更したマクロおよび「斜線による抹消」風の出力を与えるマクロも用意しています．

- 破線のもの: `\udash`，`\mdash`，`\odash`
  - `\udash{破線の例}\mdash{破線の例}\odash{破線の例}`
    → <u>破線の例</u>~~破線の例~~<u>破線の例</u>
- 波線のもの: `\uwave`，`\mwave`，`\owave`
  - `\uwave{波線の例}\mwave{波線の例}\owave{波線の例}`
    → <u>波線の例</u>~~波線の例~~<u>波線の例</u>
- 点線のもの: `\udotline`，`\mdotline`，`\odotline`
  - `\udotline{点線の例}\mdotline{点線の例}\odotline{点線の例}` → <u>点線の例</u>点線の例<u>点線の例</u>

- 斜線による抹消: `\xout`
  - `\xout{斜線による抹消}` → ~~斜線による抹消~~

これらはいずれも，下線などを付加する対象のテキストの途中での行分割が可能です．禁則処理についても考慮していますが，pLaTeX 使用時の「`\jcharwidowpenalty` を用いた段落末の 1 字孤立の回避」の処理はサポートできていませんので，必要があれば `\uline` などの引数の中で `\nobreak` を用いてください．また，uline−− パッケージは pLaTeX の使用を念頭に置いています．オリジナルの（欧文用の）LaTeX で欧文テキストに下線などを付加する場合にも使えますが，JLaTeX への対応は不完全です（JLaTeX 使用時にはとりあえず処理可能というレベルで，禁則処理などがうまくいきません）．

これらのマクロは組み合わせて用いることもできます．

- `\uline{下線と \mline{打ち消し線}の併用}` → 下線と~~打ち消し線~~の併用
- `\uline{\oline{下線と上線の併用}}` → $\overline{\text{下線と上線の併用}}$

その際，破線・波線・点線を付加するコマンドおよび `\xout` は最大で 5 重にまで重ねて用いることができます（波線でない実線については制限なしに重ねることができます）．やむを得ず破線・波線・点線を付加するコマンドあるいは `\xout` を 6 重以上に重ねる場合には，`\UlineMaxPatterns{`⟨*max_patterns*⟩`}`（⟨*max_patterns*⟩ は 6 以上の整数）という指定をプリアンブルで行うと破線などを付加するコマンドを重ねる段階数の上限を ⟨*max_patterns*⟩ に増やせます．

また，上記の 13 個のマクロはテキストに付加する線などの位置・種類が異なるのみで，実質的には同一のマクロです．そこで，以下の説明は主に `\uline` について行います．

まず，書体変更コマンド，欧文部分，数式についても単純に `\uline` などの引数にできます．version 2.0.0（2007/06/17 版）以降の版では，インライン数式中での自動改行にも対応しています．また，デフォルトでは単語のハイフネーションは自動的には行われません．ただし，（波線でない）実線を付加するマクロ（`\uline`，`\mline`，`\oline`）の場合のみ `\-` を使えます．

- `\uline{\textgt{書体変更}と長めのインライン数式 ($\langle x \rangle = (x_1 + x_2 + \cdots + x_{n-1} + x_n) / n$) を含む sample text}` → 書体変更と長めのインライン数式（$\langle x \rangle = (x_1 + x_2 + \cdots + x_{n-1} + x_n)/n$）を含む sample text
- `\uline{simple {\bfseries sample text with {\itshape long breakable underline}}}` → simple **sample text with *long breakable underline***

なお，パッケージオプション nomath を適用すると，インライン数式中での行分割を行わずに version 1.9.5 以前の版と同様の処理を行います．

通常のコマンドは，原則としてアルファベットと同様に扱われます．また，故意に 1 文字扱いにしたい文字列（行分割させたくない文字列）に対しては，(通常のテキストの記述の場合と同じく) \mbox の引数にするという方法が使えます (なお，本文書では属性 JY1/mc/m/n に対して jis.tfm を用いています)．一方，直前の例からもわかるように，非数式部分で単にグルーピングを行っただけでは 1 文字扱いにはなりません．なお，{，} で囲まれた数式（部分数式）の途中での行分割は起こりません（これも通常の数式の記述と同じです)．

- \uline{\TeX （あるいは \LaTeX ）における下線}
  → TEX（あるいは LATEX）における下線
- \parbox[t]{18zw}{\uline{\texttt{\symbol{92}mbox}を用いた行分割の抑制（例：\mbox{キャット}）}}
  → \mbox を用いた行分割の抑制（例：キャット）
- \parbox[t]{18zw}{\uline{\texttt{\symbol{92}mbox}を用いた行分割の抑制（例：キャット）}}
  → \mbox を用いた行分割の抑制（例：キャット）
- 部分数式を用いた数式中での行分割の抑制：\uline{$a_1 + a_2 + \cdots + a_n + b_1 + b_2 + \cdots + b_n$}
  → 部分数式を用いた数式中での行分割の抑制：$a_1 + a_2 + \cdots + a_n + b_1 + b_2 + \cdots + b_n$
- 部分数式を用いた数式中での行分割の抑制：\uline{${a_1 + a_2 + \cdots + a_n} + {b_1 + b_2 + \cdots + b_n}$}
  → 部分数式を用いた数式中での行分割の抑制：$a_1 + a_2 + \cdots + a_n + b_1 + b_2 + \cdots + b_n$

標準配布のクラスファイルにおける \small といった一部の fragile なコマンドについては \uline などの側で対処していますが，それ以外の一般の fragile なコマンドが \uline などの引数に含まれる場合には，\protect で保護するか，3.2 項で説明する \AddCommandInUline，\AddMathCommandInUline を用いて保護対象として登録してください（\uline などでは，\ref などが与える文字列を正しく処理するといった目的のため，引数を展開してから処理しています)．

- 例えば，何らかの記号を picture 環境で描いたものを与えるマクロを

```
\newcommand*\foo{%
   \begin{picture}(8,8) ... 適当な記述 ... \end{picture}}
```

のような具合に定義した場合，この \foo を \uline の引数中で用いる際には，基本的には \protect\foo のように \protect を前置してください．なお，3.2 項で述べるように，\AddCommandInUline を用いて \foo を \uline などの引数中での保護対象として登録することもできます．

2 オプション指定

この文書の冒頭で挙げた 13 個のマクロのうち，\xout 以外のもの（下線類を付加するもの）については，オプション引数を用いて下線などの位置・太さなどの変更ができます．

- lines=⟨*number*⟩: 線の本数を正整数 ⟨*number*⟩ に設定します．
  - \uline[lines=3]{サンプル}\mline[lines=2]{サンプル 2}
    → サンプル~~サンプル 2~~
- position=⟨*dimen*⟩: 下線などの基準位置を寸法 ⟨*dimen*⟩ のところに設定します（ベースラインの位置から，横組み時は上側を正にし，縦組み時には右側を正にして計ります）．また，この「基準位置」というのは，
  - \uline，\udash，\udotline の場合: 最も上にある線の中心の位置
  - \mline，\mdash，\mdotline の場合: 一連の線の中央の位置
  - \oline，\odash，\odotline の場合: 最も下にある線の中心の位置

  のことです．波線を付加するマクロの場合もこれらに準じています．
  - \uline{サンプル}\uline[position=0pt,lines=2]{サンプル 2}
    → サンプルサンプル 2
  - \mline[lines=2,position=5pt]{サンプル \rule{5pt}{5pt}}
    → ~~サンプル■~~

  なお，この position オプションに限り，寸法を具体的な数値で与える場合には，「position=」を省略して寸法のみで指定できます．例えば，\uline[0pt]{...}と指定した場合は \uline[position=0pt]{...}として扱われます．
- width=⟨*dimen*⟩: 下線などの太さ（点線については，点線を構成する個々の点（小円）の直径）を寸法 ⟨*dimen*⟩ に設定します．
  - \uline{サンプル}\uline[width=1pt]{サンプル 2}
    → サンプルサンプル 2

  ただし，波線については（繰り返しパターンとしてフォント lasy5 の特

定の文字を用いている都合で）0.28 pt が太さの下限になります．

- `linegap=`⟨*dimen*⟩: 2 重以上の下線などでの線の間隔（線の中心間の距離ではなく，線どうしの間隙の大きさ）を寸法 ⟨*dimen*⟩ に設定します．
  - `\mline[lines=2]{サンプル}\mline[lines=2,linegap=2pt]{サンプル 2}` → ~~サンプルサンプル 2~~
- `dashlength=`⟨*size*⟩: 破線の場合専用で，破線の繰り返しパターンの実線部分の長さを ⟨*size*⟩ にします．ただし，個々の行に含まれる破線部分の両端に現れる実線部分の長さは ⟨*size*⟩ の 1/2 となります（つまり，繰り返しパターンは「⟨*size*⟩ の 1/2 の長さの実線部分＋空白＋ ⟨*size*⟩ の 1/2 の長さの実線部分」です）．
  - `\udash{破線のサンプル}\udash[dashlength=5pt]{破線のサンプル}` → 破線のサンプル破線のサンプル
- `dashgap=`⟨*gap*⟩: 破線の場合専用で，破線の繰り返しパターンの空白部分の長さを ⟨*gap*⟩ にします．
  - `\udash{破線のサンプル}\udash[dashgap=3pt]{破線のサンプル}` → 破線のサンプル破線のサンプル
- `dotgap=`⟨*gap*⟩: 点線の場合専用で，点線を構成する個々の点（小円）の間隔（点の中心間の距離ではなく，2 個の点の間の間隙の大きさ）を ⟨*gap*⟩ にします．
  - `\udotline{点線のサンプル}\udotline[dotgap=4pt]{点線のサンプル}` → 点線のサンプル点線のサンプル
- `color=`⟨*colorname*⟩: 下線などの色を ⟨*colorname*⟩ に設定します．この指定を利用する際には color パッケージを別途読み込んでおく必要があります（次項についても同様です）．
  - `\uline[color=blue]{サンプル}` → サンプル
- `color={[`⟨*model*⟩`]`⟨*parameters*⟩`}`: 下線などの色をカラーモデルが ⟨*model*⟩ で，パラメータが ⟨*parameters*⟩ である色に設定します．
  - `\uline[color={[gray].8},width=1pt]{サンプル}` → サンプル
- `foreground`，`background`: `background` を指定した場合，下線などはそれらを付加する対象の文字列の背後に描かれます．`foreground` を指定した場合には，下線などはそれらを付加する対象の文字列の前面に描かれます（`foreground` がデフォルトです）．
  - `\mline[color=red,width=2pt]{\textgt{サンプル}}%`
    `\mline[color=red,width=2pt,background]{\textgt{サンプル}}` → ~~サンプルサンプル~~

- nomath: nomath を指定した場合，数式中での行分割を行いません．
  - \uline{数式（$\langle x \rangle = (x_1 + x_2 + \cdots + x_{n-1} + x_n) / n$）を含む例} → 数式（$\langle x \rangle = (x_1 + x_2 + \cdots + x_{n-1} + x_n)/n$）を含む例
  - \uline[nomath]{数式（$\langle x \rangle = (x_1 + x_2 + \cdots + x_n) / n$）を含む例} → 数式（$\langle x \rangle = (x_1 + x_2 + \cdots + x_{n-1} + x_n)/n$）を含む例

一方，\xout に対しては，上記の各種の指定のうち color=⟨*colorname*⟩，color={[⟨*model*⟩]⟨*parameters*⟩} の形式の色指定のみが有効です．

- \xout[color=blue]{サンプル} → サンプル
- \xout[color={[gray].5}]{サンプル} → サンプル

\uline などの引数の中で別の \uline などを用いた場合，それらに対するオプション指定は独立に適用されます（が，内側の \uline などの引数には外側の \uline などに由来する線も同時に付加される，というのは先の例でみたとおりです）．

- \uline[width=.8pt,color=blue]{オプションの\uline[-4.5pt]{独立性}の例} → オプションの独立性の例

この例では，外側の \uline に対する線の太さ・色の指定は内側の \uline には影響していないことに注意してください．

また，下線などの位置・太さを一律に変更するコマンドを用意しています．例えば，\uline に対しては次の 2 個のコマンドが利用できます．

- \ulineposition{⟨*pos*⟩}: \uline による下線の位置のデフォルト値を寸法 ⟨*pos*⟩ に設定します．
- \ulinewidth{⟨*width*⟩}: \uline による下線の太さのデフォルト値を寸法 ⟨*width*⟩ に設定します．

\mline などの位置・太さのデフォルト値も，\ulineposition などの uline のところを変更した名称のコマンド \mlineposition などを用いて設定できます．\ulineposition などでの設定に関しては，次の 2 点に注意してください．

- \ulineposition などによる設定は局所的なものです（例えば，何らかの環境の内部での \ulineposition などの設定はその環境の外部には及びません）．
- \ulineposition などによる設定は，\uline などの処理開始時における値が用いられます．つまり，\uline の引数の中で \ulineposition または \ulinewidth を用いても「下線の位置・太さを途中で変える」

ことはできません（\mline などについても同様です）.

なお，\uline などに前記のオプション指定（のうちのいくつか）を最初から適用したコマンドが要る場合には，

```
\newcommand*\doubleredunderlines{\uline[lines=2,color=red]}
```

（これは，赤い二重下線を作成するマクロ）という具合に \newcommand で定義してください．あらゆる設定を考えて，それらのひとつひとつに固有の名称をもったマクロを割り振っていてはきりがないので，uline−− パッケージ自身ではこの文書の冒頭で挙げた 13 個のみを用意します.

3 他のマクロを含むテキストに下線などをつける場合の注意

3.1 一般のマクロへの対処法

先に述べたように，\uline などの引数の中では（robust な）マクロはアルファベットと同様の扱いとなります．その一方で，\textbf などの「引数をとるマクロ」は引数を考慮して扱う必要がありますし，\bfseries などのように引数をとらなくとも下線などをつける対象にはならないマクロもあります．定義時に目に付いたマクロに関してはあらかじめ対処を施していますが，もちろん未対処のものもありますし，また，ユーザ定義のマクロにまであらかじめ対処することはできません.

そこで，\uline などの引数の中に（必要に応じて \protect をつけて）含めたときにエラーを生じるか意図通りには処理されていない場合には，次のように対処してください.

- \textcircled のごとく引数を含めて 1 文字扱いにしてもよいもの: 非数式部分では，\mbox{\textcircled{c}} のごとく 1 文字扱いにする範囲を \mbox に入れてください（ただし，\textcircled 自身については対処済みです）．数式中では，1 文字扱いにする範囲とそれにつく上下の添字の全体を {，} で囲んでください.
- \bfseries のごとく下線をつける対象（下線をつける対象はボックスに入れられます）にはできないもの: \nonletter{\bfseries} のごとく \nonletter の引数にしてください（もっとも，\bfseries 自身についても対処済みです）.
- \textbf の類の書体変更範囲を引数とするような書体変更マクロ: \textcommand{\textbf}{書体変更範囲} のごとく，書体変更マクロを \textcommand の引数にしてください（なお，\textbf 自身についてはやはり対処済みです）.

- 数式において，上下の添字やマクロの引数は原則として {，} で囲んでください（単一の数字のように明らかに 1 文字であるものは囲まなくても問題ありませんが，意外なものが複雑な処理をしていることがあります）.

例えば，

```
\DeclareRobustCommand*\textbfit[1]{%
  \leavevmode{\bfseries\itshape#1}}
```

のようにボールド・イタリック体に変更する簡単なマクロ \textbfit を定義したものとします．そのとき，単に \uline{\textbfit{bold italic}}と記述したのでは（\uline 側では \textbfit には対処していないため）エラーが生じます．この場合，\uline{\textcommand{\textbfit}{bold italic}} のように変更すると，***bold italic***という出力が得られます．

さらに，今の \textbfit のような「ただ 1 個の引数をとり，その引数が書体変更範囲となる」タイプの書体変更マクロについては，

```
\AddTextCommandInUline{\textbfit}
```

のように \AddTextCommandInUline を用いて，\textbfit を \uline などに対して書体変更マクロとして登録できます．実際，上記の記述で \textbfit を登録した後では，単に \uline{\textbfit{bold italic}}と記述してもエラーなく ***bold italic***という出力が得られます．

なお，version 2.0.0（2007/06/17 版）以降では，\DeclareTextFontCommand で定義したテキスト用書体変更コマンドに関しては（ユーザ自身が \AddTextCommandInUline を用いなくても）自動的に認識・登録されます．

3.2 保護の必要がある自作マクロなどを多用する場合

3.1 項で述べたように，\uline などの引数中で用いるとエラーが生じるコマンドに対しては一般に \protect を前置したり，適宜 1 文字扱いにしたりすることで処理可能となります．もっとも，そのようなコマンドを多用する場合にいちいち保護のための記述を行うのは面倒ですので，version 2.2.0（2007/06/20 版）以降の版では保護対象のコマンドを登録するためのマクロ \AddCommandInUline，\AddMathCommandInUline を用意しています．

\newcommand，\renewcommand あるいは \DeclareRobustCommand で定義されたマクロを保護対象として登録するには，例えば，

```
\newcommand*\foo[2][n]{...}
```

のようなマクロ定義の \newcommand のところを \AddCommandInUline または \AddMathCommandInUline に取り換えたうえでマクロ定義の本体（置換テ

キスト部分）を空文字列あるいは \foo の属性を指定する文字列（これについては後述します）にした

```
\AddCommandInUline*\foo[2][n]{}
```

という記述を用います（\renewcommand，\DeclareRobustCommand で定義されているマクロについても同様です）．なお，\AddCommandInUline は非数式部分で用いるマクロの登録に用い，\AddMathCommandInUline は数式部分で用いるマクロの登録に用います．\def で定義されたマクロについても，\newcommand あるいは \renewcommand で定義可能ならば，それらで定義した場合を想定したものを上記のように書き換えると登録できます（\newcommand などでは定義できないような複雑な書式指定を伴うマクロやカテゴリーコードの変更などの特殊な処理を伴うマクロは \AddCommandInUline，\AddMathCommandInUline では登録できません）．

\AddCommandInUline，\AddMathCommandInUline での登録の際には，

```
\AddCommandInUline*\foo[2][n]{kanji,*}
```

（これは，\foo には \foo*{...} のように * をつけて用いる場合があり，かつ，\foo とその引数の全体を和文文字扱いにする場合）のように，登録するコマンドの属性を指定できます．可能な指定は次のとおりです（コマンド \foo を登録するものとして説明します）．

- \AddCommandInUline に対して可能な指定:
  - star（別名: *）
    \foo には \foo* のように * つきの形で用いられることもある，ということを表します．
  - skip（別名: s）
    \foo とその引数の全体を，下線などをつける対象にしません．
  - kanji（別名: j）
    \foo とその引数の全体を和文文字扱いにします．
  - noglue（別名: n）
    \foo とその引数の全体とその前後の文字との間に明示的に空白が入っていない場合には，\foo とその引数の全体とその前後の文字とを密着させます．

  skip，kanji，noglue のいずれも用いていない場合には，\foo とその引数の全体をアルファベット 1 文字と同様に扱います．
- \AddMathCommandInUline に対して可能な指定:
  - star（別名: *）
    \foo には \foo* のように * つきの形で用いられることもある，ということを表します．

- \mathop（別名: op）
  \foo とその引数の全体を大型演算子の扱いにします.
- \mathbin（別名: bin）
  \foo とその引数の全体を 2 項演算子の扱いにします.
- \mathrel（別名: rel）
  \foo とその引数の全体を関係演算子の扱いにします.
- \mathopen（別名: open）
  \foo とその引数の全体を開き括弧類の扱いにします.
- \mathclose（別名: close）
  \foo とその引数の全体を閉じ括弧類の扱いにします.
- \mathpunct（別名: punct）
  \foo とその引数の全体を数式用句読点の扱いにします.
- \mathinner（別名: inner）
  \foo とその引数の全体を内部数式として扱います.

\mathop, \mathbin, \mathrel, \mathopen, \mathclose, \mathpunct, \mathinner のいずれも用いていない場合には \foo とその引数の全体を通常の文字として扱います.

これらの指定を複数用いる場合には，先の例のように個々の指定をコンマで区切って列挙します．また，\AddMathCommandInUline に対する \mathop と \mathbin のような両立しない指定を同時に与えた場合には，両立しないもののうちで最後に与えたものが有効になります.

\AddCommandInUline，\AddMathCommandInUline については，次の 3 点にも注意してください.

- \AddCommandInUline, \AddMathCommandInUline で登録したマクロとその引数は 1 文字扱いになります．したがって，\textbf のようなマクロや \newcommand*\NFSS{New Font Selection Scheme} のように定義された \NFSS のようなマクロを登録すると不都合があります.
- \AddCommandInUline はテキスト用コマンドの保護情報のみを更新し，\AddMathCommandInUline は数式用コマンドの保護情報のみを更新するという具合に，これらのコマンドは連動してはいません．数式中と非数式部分のどちらでも用いられるマクロを保護対象として登録する場合には，\AddCommandInUline と \AddMathCommandInUline の両方を用いて登録してください.
- マクロ \foo を保護対象として登録するのは，なるべく \foo が定義された後にしてください.

3.3 \index，\verb などへの対処法

version 1.3.0（2006/12/01 版）より，\uline などの引数の中で \index，\glossary，\verb を用いることが一応できます．具体的には，\uline などが他のマクロ（例えば，\parbox や \footnote）の引数の中に入っていなければ，下線などを付加するテキストの中で，\index，\glossary，\verb が使えます．例えば，「\uline{\verb/\uline/マクロの使用例}」という記述からは，「\uline マクロの使用例」という出力が得られます．ただし，\verb などが「\uline などの処理を行う前に他のマクロの引数として読み取られてしまう」場合には，従来どおり「\verb の代わりに \texttt（と \symbol，\char）で記述」，「\index の引数を他のマクロの引数中で使える形（例えば，特殊文字に \string を前置した形）で記述」といった対処法を用いてください．なお，\verb などへの対処に伴い，\uline などの処理に version 1.2.0 以前の版に比べさらに時間がかかるようになったため，\index，\glossary，\verb 用の処理を取り止めるためのパッケージオプション noverb も用意しています．

3.4 ユーザ定義マクロを用いる場合の注意

uline−− パッケージでは LaTeX 自身あるいは既存のいくつかのパッケージが提供するマクロも \uline の引数中で使えるようにしてはいますが，あくまで **LaTeX または各種パッケージが提供する名称・意味のままで用いた場合**を念頭に置いています．第 2 節で例示したような \newcommand で定義した単純なマクロは（そのマクロの定義中で uline−− パッケージがサポートするコマンドのみを用いている場合には）たいてい \uline などの引数中でも使えますが，うまくいかない場合にはパッケージなどが提供するコマンドを直接用いたうえで，必要に応じて 3.1 項で挙げた対処法を用いてください．

4 色付けに関する補足

\uline などの引数の中では，color パッケージが提供するマクロ \color，\textcolor，\colorbox，\fcolorbox を特別な記述をしなくても使用できます．

- \uline{{\color{blue}色に関する}サンプル}
  → 色に関するサンプル
- \uline{\textcolor{magenta}{色に関する}サンプル}
  → 色に関するサンプル
- \uline{\colorbox{cyan}{色に関する}サンプル}
  → 色に関する サンプル

- `\uline{\fcolorbox{blue}{cyan}{色に関する}サンプル}`
  → 色に関するサンプル

この例では，`\color`，`\textcolor` はテキスト部分の色にしか影響しない（下線の色には影響していない）ことにも注意してください．

また，`\colorbox` の引数の途中での行分割はできないので，行分割可能な背景色付きテキストを下線部に入れるために `\uline` などの引数中でのみ使えるコマンド `\highlight` を用意しています．使用法は，`\colorbox` の代わりに `\highlight` にするだけです（なお，単に背景色だけをつけるには，`\mline` を `\mline[color=cyan,width=12pt,background]{...}` のように使えます）．

さらに，`\highlight*` のごとく `\highlight` に `*` をつけて用いると，背景色をつけた部分のテキストを隠します．単純に背景色で上塗りしたりテキストの色を背景色と同じにするのではなく，`\phantom` を用いて隠しています．

- `\uline{行分割可能な \highlight[gray]{.8}{背景色つきテキスト}の例}` → 行分割可能な背景色つきテキストの例
- `\uline{行分割可能な \highlight*[gray]{.8}{背景色つきテキスト}の例}` → 行分割可能な　　の例

ただし，今の `\highlight` を用いたり，`\uline` などを 2 重以上に用いて複数の色の下線などを重ねたりすると，行分割の際に不具合が生じることがあります．極端な例ですが，pLaTeX 使用時の `\kanjiskip` あるいは JLaTeX 使用時の `\jintercharskip` を 10 pt plus 3 pt に設定した場合，次のような出力が生じます（「単語間スペース」に関しても同様のことが起こります）．

- `\parbox[t]{16zw}{\uline{背景色と \highlight{cyan}{下線を併用した}テキストのサンプル}}`
  → 背景色と下線を併用したテキストのサンプル
- `\parbox[t]{16zw}{\uline[0pt]{複数の色の \uline[color=blue]{下線を用いた}テキストのサンプル}}`
  → 複数の色の下線を用いたテキストのサンプル

このような場合，破線または波線を用いていない場合にはプリアンブルで `\UseAltLeaders` というコマンドを用いる（か，内部的には同じことですが，uline−− パッケージの読み込み時にパッケージオプション `altleaders` を指定する）と出力が改善します（が，処理が非常に重たくなります）．ただし，`\UseAltLeaders` は「非常に細かい繰り返しパターン」を使えることを前提としているので，`\UseAltLeaders` 指定時には（`\udash` などによる）破線・波線・

点線は使えません．uline−− パッケージが提供するマクロを用いるときには，破線・波線・点線を用いる場合には複数の色の線を重ねずに済ませてください．

5　仕様に関する補足

この節の内容は \uline などの「現時点での」仕様に基づくもので，過去の版とは異なっている場合あるいは今後の版で変更となる可能性があります．

5.1　破線・波線・点線の処理についての補足

uline−− パッケージが提供する \udash などによる破線・波線・点線は繰り返しパターン（破線の場合は実線部分と破線部分，波線の場合は「ひとつの波」，点線の場合は個々の点とその両側の空白）を（単に \leaders を用いて）繰り返して作成しています．ここで，破線などをつける部分の長さが繰り返しパターンの幅の整数倍でなかったとしても「破線・波線あるいは点線自体の幅をそれらの線をつける対象の幅に一致させる」という調整は行っていません（また，そのような調整は TEX 自身の機能では困難と思われるので，調整を行う予定もありません）．特に，破線部などの端点においては，次の例のように，最大で繰り返しパターンの幅の 1/2 だけ破線などの開始・終了位置がずれます（したがって，「破線の繰り返しパターンの実線部分・空白部分の長さの和」あるいは「点線における点の間隔」は充分に小さくするのが無難です）．

- `\udash[dashlength=1zw,dashgap=.1zw]{破線のサンプル}`
  → 破線のサンプル
- `\udash[dashlength=.7zw,dashgap=.1zw]{破線のサンプル}`
  → 破線のサンプル

5.2　書体変更あるいは \special とペアカーニングの関係

一般的な（ユーザが直接用いる）和文フォントメトリックにおいては，句読点や括弧類が隣り合う際にはそれらの間隔を調整してあります（例えば，「…），」ではなく，「…），」となります）．しかし，この調整は複数のフォントをまたがる場合にはなされず，「\textgt{…ゴシック），}，」という記述に対しては「…ゴシック），，」という具合に出力されます．また，色の変更がある場合には色の変更・復元に伴って \special が挿入されるため，その \special を挟む 2 個の文字は「隣り合っていない」かのように扱われてしまいます（例えば，「\textcolor{blue}{…括弧類」}，」と記述すると「…括弧類」，」という出力が得られます）（\textcolor に関してはグルーピングの問題もあり

ますが，「…\color{blue}…\normalcolor …」という形にしても状況は変わりません）．

一方，\uline などでは書体変更や色の処理は (p)LaTeX に任せ，一連の文字・コマンドの字面だけを見て処理しています（きわめて大雑把に言えば，例えば「\uline{\textbf{あ}い}」は「\textbf{\underline{あ}}\leaders \vrule... \hskip\kanjiskip\underline{い}」（... のところは \vrule の height と depth の指定）のような感じに変換され，その後実際の組版処理に回されます）．その際，グループの開始・終了箇所をまたぐ文字の組についてもそれらの組み合わせを考慮しています．そのため，下線などをつけた場合とつけない場合とでは次の例に示すような相違が生じます．

- 「下線」と \textgt{「書体変更」}（あるいは \textcolor{blue}{「色変更」}）が関与する例
  →「下線」と**「書体変更」**（あるいは「色変更」）が関与する例
- \uline{「下線」と \textgt{「書体変更」}（あるいは \textcolor{blue}{「色変更」}）が関与する例}
  →「下線」と**「書体変更」**（あるいは「色変更」）が関与する例
- ［比較用の例］
  →「下線」と「書体変更」（あるいは「色変更」）が関与する例

書体変更も色付けも行っていない例と比較するとわかるように，結果的には（jis メトリックを使用している場合）概ね調整不要である方向に変わっています．もちろん，必要があれば \inhibitglue（\<）あるいは \null（= \hbox{}）を用いて手動で補正できますが，単なる空の括弧（{}）は（ペアカーニングに関しては）無視されるので適宜 \null に変更してください．

- \uline{（猫）{}（ねこ）} →（猫）（ねこ）
- \uline{（猫）\null（ねこ）} →（猫）（ねこ）
- ［比較用］（猫）（ねこ）→（猫）（ねこ）
  （猫）{}（ねこ）→（猫）（ねこ）
  （猫）\null（ねこ）→（猫）（ねこ）

5.3 下線などをつけるテキストの先頭・末尾に括弧類・句読点がある場合

現時点では，\uline などの引数の先頭・末尾に括弧類・句読点がある場合，その括弧類・句読点に伴う 1/2 字分の空白は次のように取り扱われます．

- \uline などの引数の先頭の開き括弧類に伴う空白:
  段落の先頭である場合には無視されます．段落の先頭ではない場合は

\hspace を用いて追加されます.

- \uline などの引数の末尾の閉じ括弧類または句読点に伴う空白:
  原則として \hspace を用いて追加されますが，直後に行頭禁則文字（のうち，特に \prebreakpenalty が 10000（以上）であるもの）が続いている場合には空白を入れません．

具体的には，次のようになります．

- 「\uline{（サンプル）}」→「（サンプル）」
- …である \uline{（第 1 節参照）}．→ …である（第 1 節参照）．
- \fbox{\uline{（サンプル）}}→ （サンプル）

これらの例の最初のものの下線部の開き括弧の直前の空白は余分ですが，\uline の直前にあるものを知ることはできないため自動的には対処できません．この場合は \uline の引数の先頭に \inhibitglue（\<）を入れると，次の例のように補正できます．

- 「\uline{\<（サンプル）}」→「（サンプル）」

\fbox の例についても同様で，\uline の引数の先頭・末尾に \inhibitglue（\<）を入れると，次のようになります．

- \fbox{\uline{\<（サンプル）\<}}→ （サンプル）

今までの例では括弧類などに伴う 1/2 字分の空白を下線部などの外に出しましたが，その空白部分も含めて下線を引く場合には，\uline などの引数に適宜 \null を入れてください．例えば，先の \fbox の例の \uline の引数の先頭・末尾に \null を追加すると次のようになります．

- \fbox{\uline{\null（サンプル）\null}}→ （サンプル）

上記のことを用いて空白と下線などの関係を調整すると，次のようなことも一応できます．

- \oline{（上）\uline{（中）}}\uline{（下）}
  → （上）（中） （下）
- \oline{（上）\uline{（中）\<}}\uline{\null（下）}
  → （上）（中）（下）

5.4 フォント lasy5 の形式と波線の出力

波線用のマクロ \uwave などでは波線の繰り返しパターンとしてフォント lasy5 の特定の文字を使用しています．ここで，このフォント lasy5 の表示・印刷に METAFONT ソースから作成した PK フォントを用いるか，（BlueSky Research 社による）Type1 版を用いるかによって波線の「つながり具合」が

異なるように思われます（なお，筆者は BaKoMa フォントについては（所有も使用もしていないため）検証していません）．そこで，type1cm パッケージを読み込んでいる場合（この場合には Type1 版を使用するものとみなしています）とそうでない場合とで波線の繰り返しパターンに対する調整の仕方を変えています．その結果，type1cm パッケージを読み込んでいるか否かによって波線の出力がいくぶん異なってきます．また，type1cm パッケージを用いている状態で処理した波線を PK フォントを使う設定にしたプレビューアで眺める（あるいは，その逆の状況で眺める）と波線の繰り返しパターンの間に小さなすきまがある（あるいは，繰り返しパターンの重なりの部分がいくぶん太くなる）ように見えますので，注意してください．

ただし，パッケージオプション `usepk` を指定すると，type1cm パッケージを読み込んでいるか否かによらず PK フォント用の調整を行います．また，パッケージオプション `usetype1` を指定すると，type1cm パッケージを読み込んでいるか否かによらず Type1 版用の調整を行います．

5.5 点線の個々の点の描画方法について

uline−− パッケージのデフォルトでは，`\udotline` などによる点線の個々の点は，フォント lcircle10 を適当にスケーリングしたものを利用して出力しています．一方，パッケージオプション `tpic` を適用した場合には，点線の個々の点を tpic specials を用いて描画します．

5.6 背景色と破線・波線・点線を併用した場合，破線などと他の線を重ね書きした場合

`\mdash` などで付加した破線・波線・点線が `\highlight` による背景色に重なる場合，破線などの断片の出力と背景色の出力のタイミングの関係で，破線などの一部が途切れます．現時点の実装ではいたしかたありませんので，背景色を設定する部分では（波線ではない）実線のみを用いてください．また，それと同じ理由で，破線などと他の実線・破線などを重ね書きした場合，重ね書きされた破線などの一部が途切れることがあります．

5.7 欧文の自動ハイフネーション処理について

この文書の冒頭で述べたように，uline−− パッケージのデフォルトでは `\uline` などで下線などを付加したテキスト内での自動ハイフネーションは行われません．ただし，（波線ではない）**実線**を付加するマクロ（`\uline`，`\mline`，`\oline`）の引数中の，**アルファベットのみ**からなる語に関しては，パッケージオプション `autohyphenate` を適用すると自動ハイフネーション

を試みます．なお，この処理は実験的なもので，しかも，内部で組版処理を複数回行います．それゆえ，複数回処理に伴う副作用を避けるため，アクセント記号用のコマンドなどのごく一部のコマンド以外のコマンドが含まれる文字列に対しては，自動ハイフネーションを取り止めます．

5.8　数式中で用いた場合

\uline などを数式中で用いた場合，\uline などの引数は「行分割しない，テキストスタイルの単一の数式」として扱われます．例えば，$\uline{x} + y$ は$\mbox{\uline{$x$}} + y$として扱われます．必要があれば，下線などをつける範囲での数式のスタイルを \displaystyle などで明示的に指定してください（「現在処理中の数式のスタイル」はマクロレベルの処理では取得できないので，とりあえずテキストスタイルにしています）．また，下線などをつける範囲はそれ自身がひとつの数式として扱われるため，\uline などの引数に「単独の \left，\right」を含めることはできません（単独の \left，\right を含む部分に下線などをつける場合には，ダミーの \left，\right を補ってください）．

5.9　幅のない項目に下線などをつけようとした場合

例えば，空文字列に数式用アクセントをつけた \uline{$\hat{}$} という記述を行うと，“ˆ” という下線などがつかない出力が得られます．これは，下線をつける対象の（処理上の）幅が 0 pt であるために下線をつけたとしてもその下線の長さが 0 pt となることによります．下線などをつける対象の中に幅をもたない項目が現れたときに，その項目の幅がないということが意図的なもの（例えば，\llap などを適用した場合）であるか否かはパッケージ側では知りようがないので，\uline{$\hat{}$} の類に下線などがつかないのは（現時点では）仕様とします．なお，\uline{$\hat{\ }$} のようにコントロール・スペースに数式用アクセントをつけた場合には（幅が正となり）“ˆ” のように下線などがつきますので，必要があればこの形で記述してください．

5.10　下線部などで脚注・傍注を使用する場合

\uline などで下線などを付加するテキストの中で「\footnote による脚注」や「\marginpar による傍注」および「endnotes パッケージが提供する \endnote による末注」も使用できますが，\uline などでの下線は複数段落にはわたらないことを前提としているため（つまり，\uline などおよびその内部処理に用いるマクロには \long 指定は適用していないため）\uline などの引数中の脚注テキストなどが複数段落にわたるとエラーが生じることがあります．その場合は，脚注テキスト中などでの改段落を \endgraf で行うか，

脚注テキストなどを下線部の外に出してください（必要があれば，下線などを脚注テキストなどの挿入箇所で分割して，挿入箇所の前後で別々に \uline などを用いるとよいでしょう）.

5.11 url パッケージを用いる場合

url パッケージが提供する \url コマンドおよび \urlstyle コマンドは \uline の引数中でも用いることができます．また，（ファイル url.sty 自体に解説されているような）\UrlRight，\UrlLeft などを用いたカスタマイズにもある程度対応していますが，次の 4 点には注意が必要です．

- \url（および下記の \email マクロのような「\url 風の」マクロ）は \verb コマンドと類似の処理を必要とします．\url の類を \uline などの引数の中で用いる場合には，uline−− パッケージにパッケージオプション noverb を適用しないでください．
- \UrlLeft/\UrlRight の再定義時に数式用コマンドを用いる場合には，

  ```
  \def\UrlLeft{\ensuremath{\triangleleft}}
  \def\UrlRight{\ensuremath{\triangleright}}
  ```

  という具合に \ensuremath を適宜補ってください．この例のような場合，オリジナルの（すなわち \uline などの引数の中ではないところで用いた）\url に対しては \ensuremath を用いても用いなくても構いません．
- ファイル url.sty には

  ```
  \newcommand\email{\begingroup \urlstyle{rm}\Url}
  ```

  のようなマクロの例があります．この類の \Url を直接用いた「\url 風の」ユーザ定義マクロを \uline などの引数の中でも用いる場合には，あらかじめ

  ```
  \AddUrlLikeCommandInUline{\email}
  ```

  という具合に \AddUrlLikeCommandInUline を用いて「\uline などの引数中で使える \url 風マクロ」として登録してください．
  ただし，（文字 “#” はマクロの処理に関して特殊な挙動を示すため）このような「\url 風」マクロの定義中で引数をもつようなマクロを定義または再定義した場合には，意図通りには処理されません．例えば，ファイル url.sty には

  ```
  \def\UrlLeft#1\UrlRight{%
     \special{html:<a href="#1">}#1\special{html:</a>}}
  ```

という例がありますが，\UrlLeft をこのように再定義するような「\url 風」マクロは \uline などの引数の中では正しく処理されません．

- ファイル url.sty には，

```
\renewcommand\url{\begingroup
   \def\UrlLeft{<url: }\def\UrlRight{>}%
   \urlstyle{tt}\Url}
```

のような \url 自身を再定義する例が載っていますが，uline−− パッケージではこのような再定義を行った場合はサポートしていません（3.4 項で述べたように，uline−− パッケージでは，各種のコマンドを**オリジナルの意味のままで**用いている場合を想定しています）．

6 警告メッセージ

uline−− パッケージが用意している警告メッセージとそれらが表示される状況および（典型的な）対処法を挙げます．なお，この節の記述においては警告メッセージの冒頭の “Package uline−− Warning:” という文字列と警告メッセージの 2 行目以降の行頭の “(uline−−)” という文字列は省略しています．

- Contents in a scratch box register (\box0 etc.) may be broken on input line ....

  \uline などの引数の中でボックスを貼り付ける場合，例えば

```
\newbox\tempbox%%% プリアンブルなどでただ一度行えば充分
\setbox\tempbox\hbox{\verb/(^o^)/}
\uline{sample: \copy\tempbox}
```

のような記述を行う場合に，ユーザ自身で確保したボックスではなく 0 番から 9 番までの「一時使用用」ボックスを用いた場合のメッセージです．つまり，上記の例そのものではなく，例えば

```
\setbox0\hbox{\verb/(^o^)/}
\uline{sample: \copy0}
```

のような記述（ここでは 0 番のボックスを使ってしまっています）を行った場合のメッセージです．

\uline などでも「一時使用用」のボックスを用いるため，ユーザが記述した \copy0 などの記述を実行する前にそれらのボックスの中身が変わっている可能性があります．したがって，この警告が現れた場合には \newbox で割り当てたボックスを用いてください．

- Package “color” is not loaded. Line color is ignored on input line ....
  color パッケージを読み込んでいない状態で下線などの色を指定した場合のメッセージです．color パッケージを読み込むか，下線などに対する色指定を取り止めてください．
- Only solid lines are available under `\UseAltLeaders` or package option “`altleaders`”. Option for leader type is ignored on input line ....
  `\UseAltLeaders` コマンドまたはパッケージオプション `altleaders` の適用時に破線・波線を用いた場合のメッセージです．第 4 節で述べたように，`\UseAltLeaders` コマンドまたはパッケージオプション `altleaders` を適用したときには単純な実線を付加するコマンド（`\uline`，`\mline`，`\oline`）のみを用いてください．
- Too many patterns are used simultaneously. Some patterns will be lost on input line ....
  破線または波線を付加するコマンドを重ねすぎた場合のメッセージです．破線または波線を付加するコマンドを重ねる段階数を減らすか，`\UlineMaxPatterns` を用いて段階数の上限を増やしてください．
- “unbox” operation is not supported.
  `\uline` などの引数の中で `\unhbox`，`\unhcopy`，`\unvbox`，`\unvcopy` を用いた場合のメッセージです．また，上記のメッセージに続いて，“`\unhbox` is replaced with `\box` here” といった状況についての補足も表示されます．`\uline` などは `\unhbox`, `\unhcopy`, `\unvbox`, `\unvcopy` には対応していませんので，`\uline` などの引数の中ではこれらを用いずに済ませてください．

7 更新履歴

- 2007/07/09 (v.2.3.4):
  - v.2.3.0 での追加コマンドの `\udotlineposition` などを `\uline` などの引数中で用いてもエラーが生じないように変更（ただし，点線に関しても「太さなどを途中で変える」ことはできないという点には変更はありません）．
  - 和文の禁則処理に関する penalty の取り扱いを改善．
- 2007/06/25 (v.2.3.2，v.2.3.3):
  - 傍点（plext パッケージによる `\bou`）の類の処理を調整（v.2.3.3）．
  - `\highlight` の内部処理および `\uline` などの引数中の `\mbox` などの中でさらに `\uline` などを用いた場合の処理を調整（v.2.3.2）．

- 2007/06/23 (v.2.3.1):
  - v.2.2.0 への変更の際に生じたバグ（数式部分と非数式部分のどちらでも用いられるコマンドの一部，例えば `\mbox`，が数式中で使えなくなっていた点）の修正．
  - txfonts/pxfonts パッケージ使用時の `\not` の処理時の無限ループ回避処理を強化．
- 2007/06/22 (v.2.3.0):
  - `\udotline`, `\mdotline`, `\odotline`, `\xout` を導入し，`\highlight` に `*` 形式を追加．
  - 「`\uline` などの引数の中の `\mbox` などによって 1 文字扱いにした箇所」の中で用いた `\highlight` がエラーを引き起こすというバグを修正．
- 2007/06/20 (v.2.2.0):
  - `\AddCommandInUline`，`\AddMathCommandInUline` を導入．
- 2007/06/18 (v.2.1.0):
  - `\uline` などの引数中の `\verb` の適用範囲に文字 '`$`' があると，そこが数式の開始と誤認されるというバグを修正．
  - `\uline` などの個々のコマンドに nomath オプションを導入．
  - mathabx パッケージ使用時の `\not`，`\varnot` の処理を修正．
- 2007/06/17 (v.2.0.0，v2.0.1):
  - 空文字列にテキスト用アクセント記号をつけるとエラーが生じるというバグを修正（v.2.0.1）．
  - インライン数式中での行分割に対応．それに伴い，パッケージオプション `nomath` を導入（v.2.0.0）．
  - 標準的なテキスト用書体変更コマンドの自動認識処理の導入（v.2.0.0）．
  - 内部処理の高速化（v.2.0.0）．
- 2007/05/26 (v.1.9.5):
  - `\ignorespaces` の処理を追加．
- 2007/05/01 (v.1.9.4):
  - イタリック補正を伴う語の直後で行分割が生じた場合の行分割位置での体裁を調整．
- 2007/04/29 (v.1.9.3):
  - `\textit` などに伴う自動イタリック補正の（`\uline` などでの内部での）取り扱いにあった，余分な括弧をつけるとイタリック補正が欠けることがあるというバグを修正．

- 2007/04/28 (v.1.9.2):
  - v.1.1.1，v.1.1.2 への変更で行った「色つきの下線などの禁則処理の調整」の際に処置し忘れていた箇所（「下線の色をカラーモデルとパラメータを直接記述して指定した場合の禁則処理の調整」のところ）を修正.
- 2006/12/29 (v.1.9.0，v.1.9.1):
  - パッケージオプション autohyphenate 適用時のハイフネーション処理の調整（v.1.9.0，v.1.9.1）.
  - \gtfamily，\mcfamily に対処し忘れていたので，それらに対処（v.1.9.0）.
  - \UlineMaxPatterns の導入（v.1.9.0）.
- 2006/12/28 (v.1.8.0–v.1.8.2):
  - \uline などの引数の中で脚注・傍注と \url コマンド（url パッケージによるもの）を併用した場合の処理を修正（v.1.8.2）.
  - \discretionary の内部処理を調整（v.1.8.1）.
  - url パッケージ（の主要な機能）に対応（v.1.8.0）.
  - パッケージオプション autohyphenate 適用時に，アクセント記号付きアルファベットを含む語も原則としてハイフネーション処理の対象とするように変更（v.1.8.0）.
- 2006/12/22 (v.1.7.1，v.1.7.2):
  - パッケージオプション autohyphenate 適用時に，幅がハイフンの幅以下であるような文字が単語末にある語のハイフネーション可能位置の取得を誤るというバグを修正（v.1.7.2）.
  - パッケージオプション autohyphenate 適用時に penalty などの処理の際にエラーが生じるというバグを修正（v.1.7.1）.
- 2006/12/21 (v.1.7.0):
  - パッケージオプション autohyphenate を導入.
- 2006/12/14 (v.1.6.2，v.1.6.3):
  - \text... の形の書体変更マクロによる自動イタリック補正に伴い下線などが一部消えてしまうことがあるバグを修正.
- 2006/12/08 (v.1.6.0，v.1.6.1):
  - \uline などの引数の中での一部の代入の記述の処理の修正（v.1.6.1）.
  - 単語間グルーの処理の調整および \setlength などのサポート（v.1.6.0）.

- 2006/12/06 (v.1.5.0–v.1.5.3):
  - \uline などをネストさせたときに内側の \uline などの引数中の \nonletter の引数まで下線などを付加する対象になっていたバグを修正（v.1.5.3）.
  - 縦組み時の段落の先頭における一部の文字（開き括弧類の場合？）の寸法の取得に失敗することへのその場しのぎの対処（v.1.5.1, v.1.5.2）.
  - \ulineposition，\ulinewidth などのカスタマイズコマンドの導入（v.1.5.0）.
  - kunten2e パッケージが提供するユーザマクロをサポート（v.1.5.0）.
- 2006/12/05 (v.1.4.0):
  - \dotfill，\hrulefill および，\Pifill（pifont パッケージが提供するもの）をサポート.
- 2006/12/02 (v.1.3.1，v.1.3.2):
  - \hrule，\vrule のパラメータ読み取りの処理のバグの修正（v.1.3.2）.
  - v.1.3.0 での変更点に関する動作調整および \hrule，\vrule のサポート（ただし，\vrule の処理は不完全）（v.1.3.1）.
- 2006/12/01 (v.1.3.0):
  - \marginpar を一応サポート.
  - \index/\glossary/\verb を一応サポート.
  - 下線などをつけたテキスト内に下線類を含む脚注・傍注を用いたときに，脚注などの中の下線部などが脚注などの周囲の線種の影響を受けていたバグを修正.
  - 各種マクロのオプション引数の取り扱いに関するバグを修正.
- 2006/11/19 (v.1.2.0):
  - グルーピングの終了処理のバグを修正.
  - sfkanbun，furikana，furiknkt の各パッケージで提供されるマクロ（の一部，ユーザマクロと思われるもの）に対応.
- 2006/11/18 (v.1.1.0–v.1.1.3):
  - 破線・波線の途中では \- を無視するように変更（v.1.1.3）.
  - 色つきの下線などの禁則処理の調整およびその変更に伴うバグの修正（v.1.1.1，v.1.1.2）.
  - amsmath パッケージが \, などを再定義していることに対処（v.1.1.0）.
  - 一応，\uline などを数式中でも使えるように変更（v.1.1.0）.

- 2006/11/11 (v.1.0.2):
  - `\pfmtname` が定義されていない場合，各種の寸法のデフォルト値の中では単位 zw，zh が用いられないように変更.
  - `\bou`（plext パッケージによるもの）の処理を修正.
- 2006/11/09 (v.1.0.1):
  - `\ref`，`\pageref` を robust なマクロにしている場合に対処.
  - `foreground` 指定と `background` 指定が混在する場合の文字間・単語間グルー部分の処理を修正.
- 2006/10/25 (v.1.0.0): 公開