

# WTRef パッケージ (v0.2.1)

ワトソン

2016 年 4 月 8 日

## 概要

WT シリーズは著者が L<sup>A</sup>T<sub>E</sub>X 文書作成にあたってよく利用するマクロを集めたものである。WTRef パッケージはこの WT シリーズを構成するパッケージの 1 つであり、L<sup>A</sup>T<sub>E</sub>X オリジナルの相互参照機能を拡張してスコープの指定と名前空間の分離、参照形式の柔軟な指定を可能にする。サポート対象は任意の T<sub>E</sub>X エンジンと L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> の組み合わせで、動作には xkeyval パッケージが必要である。

## 1 動作要件

WTRef パッケージの動作要件は以下の通りである。

- T<sub>E</sub>X エンジン：任意
- T<sub>E</sub>X フォーマット：L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>
- ドキュメントクラス：任意
- 依存パッケージ：xkeyval

## 2 パッケージ読み込み

読み込みには `\usepackage` 命令を用いる。オプションは存在しない。

```
\usepackage{wtref}
```

## 3 相互参照命令

### 3.1 相互参照命令の新設

`\newref` 命令を用いて相互参照に用いる 2 つの命令を新設することができる。この命令はプリアンブルでのみ使用可能である。

```
\newref[options]{ref types}
```

上記書式中の *ref types* は *ref type* のカンマ区切りリストである。*ref type* は制御綴に使用できる文字のみで構成された文字列で（半角英字のみにしておくことを推奨）、空であってはならない。なお *ref types* に与えるリストは、key-value リストとは異なりカンマ前後の空白も無視されないので注意が必要である。

`\newref` 命令は指定した  $\langle ref\ type\rangle$  に応じて  $\backslash\langle ref\ type\rangle label$ ,  $\backslash\langle ref\ type\rangle ref$  という形をもつ、新しい 2 つの命令を定義する。ここで前者の命令をラベル命令、後者の命令を参照命令と呼ぶことにする。`\newref` 命令は同名の命令が既に存在していても、その定義を上書きするので  $\langle ref\ type\rangle$  は注意して選ぶ必要がある。

また、 $\langle options\rangle$  には以下の内容を key-value リストで指定可能である：

`namespace= $\langle string\rangle$`  与えられた値を用いて  $\langle namespace\rangle$  を “ $\langle string\rangle:$ ” に設定する。`namespace` キーも `nonamespace` キーも指定しない場合、および `namespace` に与えた  $\langle string\rangle$  が空文字列であった場合  $\langle namespace\rangle$  は “ $\langle ref\ type\rangle:$ ” に設定される。

`nonamespace` このキーを指定すると、 $\langle namespace\rangle$  は空となる。すなわち、名前空間機能を無効化できる。なお、`nonamespace` キーに値を指定してもエラーにはならないが、単に無視される。

`scope= $\langle counter\rangle$`  スコープとして利用するカウンタを指定する。 $\langle counter\rangle$  には任意の L<sup>A</sup>T<sub>E</sub>X カウンタを指定することが可能だが、文書内で一意性のあるカウンタを指定することが望ましい。これにより  $\langle scope\rangle$  は “ $\backslash the\langle counter\rangle:$ ” の形に設定される。

これらの設定は、 $\langle ref\ types\rangle$  で指定したすべての  $\langle ref\ type\rangle$  に対応する相互参照命令について適用される。

ここで  $\langle options\rangle$  に何も指定しない場合  $\langle namespace\rangle$  は  $\langle ref\ type\rangle$  が転用されて “ $\langle ref\ type\rangle:$ ” の形となり  $\langle scope\rangle$  は空に設定される。言い換えると、デフォルトでは名前空間機能が有効、スコープ機能は無効に設定されている。

## 3.2 ラベル命令

### 3.2.1 機能と使い方

ラベル命令はラベルを付けるときに用いる。使い方は L<sup>A</sup>T<sub>E</sub>X の `\label` 命令とまったく同様である。以下にラベル命令の例 `\exlabel` を用いるときの書式を示しておく。

```
\exlabel{ $\langle label\rangle$ }
```

### 3.2.2 内部処理

ラベル命令は最終的に次のような形に展開される。

```
\label{ $\langle namespace\rangle\langle scope\rangle\langle label\rangle$ }
```

## 3.3 参照命令

参照命令は対応するラベル命令によってラベル付けされた番号を、指定した書式で印字するための命令である。例として `\exref` の使い方を以下に示す。

```
\exref[ $\langle the\ scope\rangle$ ]{ $\langle label\ list\rangle$ }
```

参照するラベルが同じスコープ内に存在する場合は  $\langle the\ scope\rangle$  は省略可能である。 $\langle the\ scope\rangle$  に参照先のスコープに対応する “ $\backslash the\langle counter\rangle$ ” の出力を直接記入することでスコープの範囲外からラベルを参照することが可能である。なお、スコープ機能が無効 (`\newref` によって相互参照命令を定義する際に  $\langle options\rangle$  で `scope` キーを指定しなかった場合) のときは、 $\langle the\ scope\rangle$  は常に無視される。

また、引数にはカンマ区切りで複数の参照先ラベルを指定することが可能である。〈*label list*〉に与えるリストは、key-value リストとは異なりカンマ前後の空白も無視されないので注意が必要である。複数のラベルを指定した場合、デフォルトでは該当する番号がカンマ区切りで出力される。この出力書式は後述する参照書式変更命令 `\setrefstyle` を用いて柔軟にカスタマイズすることができる。

## 4 参照書式変更命令

`\setrefstyle` 命令を用いると、参照命令による出力を柔軟にカスタマイズすることができる。最初に `\setrefstyle` 命令の書式を示しておく。

```
\setrefstyle{<ref type>}{<options>}
```

これにより 〈*ref type*〉に対応する参照命令が出力する内容の書式を変更できる。この `\setrefstyle` 命令はプリアンブルに限らず L<sup>A</sup>T<sub>E</sub>X 文書ソース中全域で使用可能であり、設定の有効範囲は { や } によるブロックの制御を受ける。

〈*options*〉には以下の内容を key-value リストで指定可能である：

`refcmd`=〈*command*〉 〈*label list*〉に与えたラベルの数だけ、指定した 〈*command*〉が繰り返し実行・印字される。〈*command*〉中に記入された #1 は適切に整形されたラベル名に置換される。規定値は `\ref{#1}`。

`sep`=〈*command*〉 〈*label list*〉に 3 つ以上のラベルが記入されているときに、`refcmd` の各出力の間に印字する区切りの内容を指定する。ただし、最後の区切りは `last sep` に指定した値となる。規定値は `{, \space}`。

`last sep`(=〈*command*〉) 〈*label list*〉に複数のラベルが記入されている際に出力される、最後の区切りを指定する。= 以降を記入せず、単に `last sep` と指定した場合、最後の区切りには `sep` の値が適用される。規定値は値指定なし（つまり `sep` の値に従属）。

`prefix`=〈*command*〉 参照命令を使用したとき、最初に実行・印字される内容を指定する。規定値は {}。

`suffix`=〈*command*〉 参照命令を使用したとき、最後に実行・印字される内容を指定する。規定値は {}。

これらの設定は、明示的に指定したもの以外はそれ以前の設定がそのまま引き継がれる。