

# WTRef Package (v0.3.1)

Watson

August 9, 2016

## Abstract

WT Series collects macros which author frequently use to create  $\text{\LaTeX}$  documents. WTRef package is a part of this WT Series that extend  $\text{\LaTeX}$  original cross-reference system. It makes enable to divide namespace and scope, further arrows users to customise reference formats.  $\text{\LaTeX} 2_{\epsilon}$  on any kind of  $\text{\TeX}$  engine is supported, and `xkeyval` package is required.

## 1 System Requirements

System requirements of WTRef are shown bellow:

- $\text{\TeX}$  engine: any engine
- $\text{\TeX}$  format:  $\text{\LaTeX} 2_{\epsilon}$
- Document class: any class
- Required package: `xkeyval`

## 2 Loading the WTRef Package

To use WTRef package, load `wtrf.sty` file with `\usepackage` command in preamble. No package option is available.

```
\usepackage{wtrf}
```

## 3 Cross-Reference Commands

### 3.1 Definition of New Cross-Reference Commands

`\newref` command create a set of cross-reference commands. This command can only be used in preamble.

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

*\langle ref types \rangle* are comma-separated list of *\langle ref type \rangle*. All characters of *\langle ref type \rangle* must be able to use in control sequence (only ordinary alphabet is recommended) and can not be empty. Notice that leading and trailing spaces and successive spaces around commas are ignored.

`\newref` command defines two commands: `\<ref type>label`, `\<ref type>ref`. In this document, the former are called **label commands** and the latter are called **reference commands**. `\newref` command overwrites existing commands, so `<ref name>` should be decided carefully.

In `<options>`, you can set following parameters by key-value list:

**namespace**=`<string>` Set `<namespace>` to “`<string>`”. If neither **namespace** nor **nonamespace** are specified, or in case `<string>` of **namespace** is empty, `<namespace>` is set to “`<ref type>`”.

**nonamespace** Set `<namespace>` to empty. That is to say, invalidation of function that dividing namespace. It should be noted that you can specify value for **nonamespace** and that will not make any errors, but the value will simply be ignored.

**scope**=`<counter>` Specify counter which used as scope. You can specify any L<sup>A</sup>T<sub>E</sub>X counter to `<counter>` but one which has uniqueness in a document is desirable. This key sets `<scope>` to “`\the<counter>`”.

These optional settings apply to all cross-reference commands relate to `<ref type>` in specified `<ref types>`.

Identically, if any keys do not specified in `<options>`, `<namespace>` is set to “`<ref type>`” and `<scope>` is set to empty. In other words, the function of namespace is active and function of scope is inactive as default.

## 3.2 Label Commands

### 3.2.1 Function and Usage

Label commands are used to create new labels. Usage of those are same to `\label` command of standard L<sup>A</sup>T<sub>E</sub>X. Usage of `\exlabel` is shown bellow as an example:

```
\exlabel{<label>}
```

### 3.2.2 Internal Processing

Label commands finally are expanded to following format:

```
\label{<namespace><scope><label>}
```

## 3.3 Reference Commands

Reference commands print contents of counters which labeled by label commands in specified formats. Usage of `\exref` is shown bellow as an example:

```
\exref[<the scope>]{<label list>}
```

The option argument `<the scope>` can be omitted when referring label exists in the same scope. You can refer outside of scope by writing down the output of proper `\the<counter>`. Notice that if the function of scope is inactive (i.e. in case **scope** key does not specified in `<options>` of `\newref`), this argument is always unnecessary, and in other words it will be ignored all the time.

In argument  $\langle label list \rangle$ , plural labels can be written in comma-separated. Note that leading and trailing spaces and successive spaces around commas are ignored. If actually plural labels are filled in, pertinent counters should be printed out in comma-separate form in default. You can change this format flexibly with `\setrefstyle` command.

## 4 Setting Referece Style

The output format of reference commands can be customised with `\setrefstyle` command. The syntax of `\setrefstyle` is shown bellow:

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

The `\setrefstyle` command can be used any place of  $\text{\LaTeX}$  document (not only preamble), and change reference format locally.

In  $\langle options \rangle$ , you can set following parameters by key-value list:

**refcmd**= $\langle command \rangle$  Specified  $\langle command \rangle$  repeated for the number of labels which filled in  $\langle label list \rangle$  time. String **#1** in  $\langle command \rangle$  may be replaced into appropriate label name. The default value is `\ref{#1}`.

**sep**= $\langle command \rangle$  Specified  $\langle command \rangle$  is output as a separater of each **refcmd** when more than three labels filled in  $\langle label list \rangle$ . Notice that last one separater is given by **last sep**. The default value is `{, \space}`.

**last sep**(= $\langle command \rangle$ ) Specified  $\langle command \rangle$  is output as a last separater when plura labels filled in  $\langle label list \rangle$ . Behind the = can be ommitted, and in that case **last sep** is set to identical value of **sep** (and this is the default).

**prefix**= $\langle command \rangle$  Specified  $\langle command \rangle$  put out first when referece command used. The default value is `{}`.

**suffix**= $\langle command \rangle$  Specified  $\langle command \rangle$  put out last when referece command used. The default value is `{}`.

Parameters which do not set explicitly will not be changed.