# bxluatangle パッケージ（v0.2）

八登崇之 （Takayuki YATO; aka. “ZR”）

2012/06/05

## 1 概要

本パッケージは以下の機能を提供する。

- Lua の実行コンテキストの途中で、それまで `tex.print()` 等で TeX 側に書き出したコードの実行を完了させる。

### ■前提環境

- TeX フォーマット： LuaLaTeX

### ■依存パッケージ

- luatexbase パッケージ
- luacode パッケージ

## 2 パッケージの読込

`\usepackage` で読み込む。オプションはない。

```
\usepackage{bxluatangle}
```

## 3 動機

例えば、ある Lua 関数の実行の中で、与えられたテキストを組版した時の幅と与えられた長さとの大小により処理を分ける必要があったとする。この状況を抽出して、次のようなコードを考えてみる。（luacode および bxluafacade の読込を前提とする。）

```
\setlength{\reswidth} % 長さ変数
\newcommand{\widthtest}[2]{\directlua{
  width_test(\luastringN{#1}, \luastringN{#2})
}}
\begin{luacode*}
```

```
-- text: テキスト
-- thr: 長さ
function width_test(text, thr)
  local thr_val = bxlt.to_dimen(thr)
  -- TeX の \settowidth を用いて幅を得る
  tex.print("\\settowidth{\reswidth}{"..text.."}")
  -- 【ここで一旦 TeX の実行を完了させたい】
  if bxlt.length.reswidth.width <= thr_val then
    print("short!")
  else
    pritn("long!")
  end
end
\end{luacode*}
```

上掲のコードはこのままでは正常に動作しない。`tex.print()` で `\settowidth` 命令を書き出しているが、実際にそれが実行されるのは、`width_test()` を呼び出している `\directlua` が終了してからである。

従って、通常は、こういう処理をしようと思うと、「TEX の実行を完了させる必要がある箇所」で一旦 Lua の処理を中断できるようにプログラムを工夫する必要が生じる。仮にその箇所が深いループの中だったりするとその実現は困難を極めることになる。

本パッケージを用いると、今の要望が次のようにして実現できる。

```
\setlength{\reswidth}
\newcommand{\widthtest}[2]{\tangleluaexec{% tangle 付で実行
  width_test(\luastringN{#1}, \luastringN{#2})
}}
\begin{luacode*}
function width_test(text, thr)
  local thr_val = bxlt.to_dimen(thr)
  tex.print("\\settowidth{\reswidth}{"..text.."}")
  bxlt.run_tex()
  if bxlt.length.reswidth.width <= thr_val then
    print("short!")
  else
    pritn("long!")
  end
end
\end{luacode*}
```

すなわち、Lua の実行を開始するときに、通常の `\luadirect` でなく `\tangleluaexec` を用いる（これを「tangle 付で Lua を実行する」と呼ぶ)。「tangle 付」の Lua の実行の中では、`bxlt.run_tex()` を任意の箇所で呼び出して、それまでに `tex.print()` で書き出したコードを実行させることができるようになる。

# 4 基本的な機能

LATEX 側の機能。

- `\tangleluaexec{`⟨Lua コード⟩`}` ： luaexec の `\luaexec` と同じコード記述規定で、tangle 付で Lua コードを実行する。元々 `\directlua` で呼び出していたコードを tangle 付にする場合にもこれを用いる。[*1]
- `tangleluacode` 環境 ： luaexec の `luacode` と同じコード記述規定で、tangle 付で Lua コードを実行する。
- `tangleluacode*` 環境 ： luaexec の `luacode*` と同じコード記述規定で、tangle 付で Lua コードを実行する。

Lua 側の機能。

- `bxlt.run_tex()` ： それまでに書き出された TeX コードの実行を完了させる。

[*1] ただし、`\tangleluaexec` は `\directlua` と異なり展開可能ではない。現状の実装方法では、tangle 付の実行は展開可能にはできない。