RbTeX

Steven Rosendahl
Christopher Newport University

Abstract

RbTeX is a package that provides support for running Ruby code during compilation of LaTeX documents. LaTeX is generally regarded as a static language (although packages that provide dynamic features do exist), but Ruby adds a very powerful set of features and libraries that can be used inside of LaTeX, analogous to the relationship between JavaScript and HTML.

Motivation

Modern LaTeX distributions include a tool called **lualatex** that allows users to dynamically produce content via use of Lua code. Unfortunately, the Lua standard libraries do not have as much functionality as other popular scripting languages, such as Ruby. The goal of this project is to incorporate Ruby into LaTeX in a manner similar to **lualatex**, but with the power and simplicity of Ruby over Lua.

Ruby was chosen over various other languages due to its widespread availability, its powerful yet simple package manager, **gem**, and its easy to understand syntax. Statements in other scripting languages, such as Python, require specific indentations or are very verbose. Ruby manages to cut away a lot of the code that needs to be written in other languages. Ruby is also built to support the modern web; as a result many of the gems provide one-line access to online APIs and endpoints that may be of interest to mathematicians, scientists, and anyone else who uses LaTeX.

The RbTeX Package

Since the program unites Ruby and LaTeX, it is hard to find a common place to host the package. RubyGems's policies are more lax than CTAN's, so the package is hosted as a gem. However, the gem is not enough by itself (although it is possible to have just the gem and to require it in a user's standalone Ruby project). Since the **rbtex** gem only comes with the ruby side, users need to get the install script from the official repository and run that instead. The install script provides users with both the Ruby gem *and* the LaTeX package. Users who want this package need to have an up-to-date Ruby version and TeXLive.

Working Examples

1 Getting a Word Count

In Rubylatex

```
\begin{rbtex}
def printWordCount
    numwords = `detex poster.tex`
    Tex.print "This file contains #{numwords} words"
end
\end{rbtex}
```

In Lualatex

```
\begin{luacode}
function printWordCount()
    local exitcode = os.execute("detex ".."\jobname".." | wc -w > count.txt")
    if exitcode == 0 then
        local file = io.open("count.txt")
        if file ~= nil then
            tex.print(""..file:read())
            os.remove("count.txt")
            -- Now we have to read from the file :(
        end
    end
end
\end{luacode}
```

2 Grabbing Data from Twitter

```
client.search("#", result_type: 'recent', lang: 'en').take(10).each do |tweet|
    ntw = tweet.text.gsub('#','\\#')
    ntw = ntw.gsub('_','\\_')
    ntw = ntw.gsub('&','\\&')
    ntw = ntw.gsub('$','\\$')
    Tex.print ntw
    Tex.print ""
end
```

@Muzachan I'm planning to be in Japan this December. For now going through your blog for extra helps

Goodmorning!

@_theBROKE1 don't act brand new bye

@kaylapurnell12 come watch me pole vault

s/o to God for the free carwash

RT @clawhammer36: RT/FALLOW @worldclas_babes @Sapphire_Blue69 @xxshowgirlsx https://t.co/kRBUBhubpj

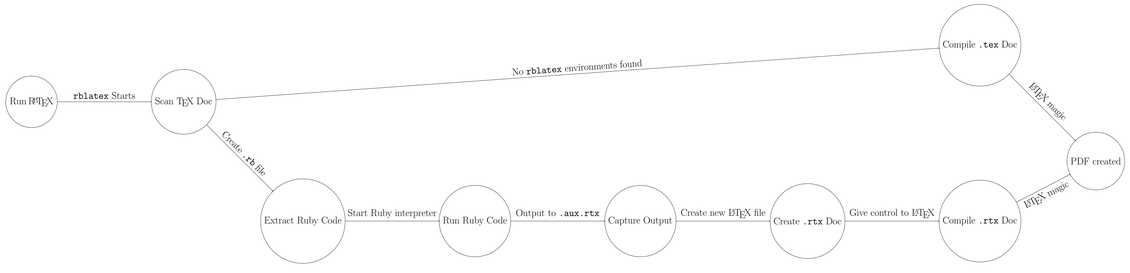
I nominate #JuliaBarretto for 100 Most Beautiful Faces of 2016 #TBWorld2016 DTopbeautyworld https://t.co/PYDZCMZtn7

RT @ilovenature: Need to go camping here https://t.co/7PG65z0pvJ

tell you what, @selftalkband are very impressive. Gaslight Anthem guitars here: https://t.co/jTLa3QZHGr https://t.co/t5hBk2avja

How It Works

In order to correctly compile a document using RbTeX, one must run the **rblatex** command on a valid LaTeX document. The workflow of RbTeX looks like



Unlike many LaTeX packages, RbTeX runs outside of TeX. This allows a user to have the ability to do what they are used to, rather than having to obey the rules of LaTeX. In fact, the **rblatex** program is removed from Ruby as well. The **rblatex** program parses both Ruby code and LaTeX code into it's own code, which it is then able to parse back into LaTeX code. In case a user forgets to use **pdflatex** and uses **rblatex** on a TeX document not containing any **rbtex** environements, **rblatex** just runs **pdflatex** and skips all the extra steps.

Basic Usage

In order to use RbTeX, a user needs to do three things:

1. Include the **rubylatex** package in their TeX document.
2. Wrap all Ruby code in **rbtex** environments.
3. Run **rblatex** in the Terminal/Command Line to compile the document and produce a pdf.

A minimal working example look something like this:

```
\documentclass{article}
\usepackage{rubylatex}

\begin{document}
\noindent Here, have some \LaTeX\\
\begin{rbtex}
Tex.print "Here, have some #{Tex.logo}"
\end{rbtex}
\end{document}
```

The result:

Here, have some LaTeX
Here, have some RbTeX

References

References

[1] Shane Kilkelly Henry Oswald James Allen. *Writing Your Own Package*. 2013. URL: **https://www.sharelatex.com/learn/Writing_your_own_package**.

[2] Scott Pakin. "How to Package Your LaTeX Package". In: (2015).

[3] Rubygems. *Make Your Own Gem*. 2016. URL: **http://guides.rubygems.org/make-your-own-gem/**.