

Why I hate rMagick

...and what to do about it!



Simplest Thing That Could Possibly Work?

Not bloody likely



Ginormous

Waaaaay too many dependencies, all independently versioned.



Leaks like a sieve

Just buy more RAM!

Uhhh... I can't buy that much RAM.

Just use minimagick!

Now you can leak PIDs like a sieve... slowly.



Clumsy & Error-prone

Half the projects I work on, it doesn't install correctly or work correctly when it does install.



Maintainer is Developer-Hostile

Doesn't actually compile out of CVS,
bug to that effect was rejected.



**So, what is the
alternative?**



Image*Science*

no more Image VooDoo



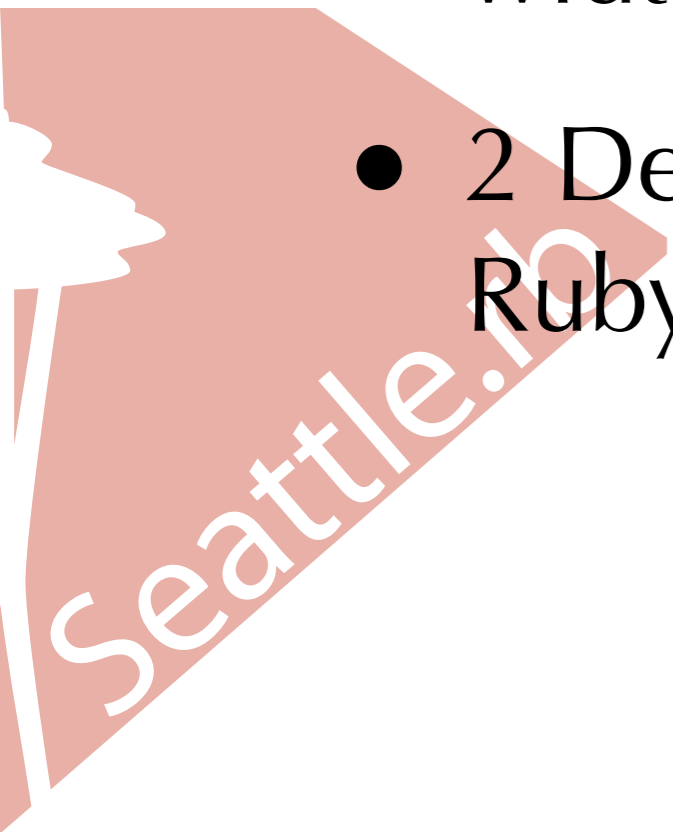
Use Case:

90+% projects use rMagick for thumbnails only.



Simplest Thing That Could Possibly Work?

- Just 1 Class.
- 6 methods: `with_image`, `with_crop`, `height`, `width`, `thumbnail`, `cropped_thumbnail`.
- 2 Dependencies: `FreeImage` and `RubyInline`.



Ginormous?

- FreedImage exports 1 header.
- Shared library is 4 meg.
- ImageScience is 74 lines long.



Leaks like a sieve?

- Doesn't leak a damn thing.
- Thumbnailed 5k images and resident never grew past 7 megs.



Clumsy & Error-prone?

- ImageScience depends on FreeImage and RubyInline only.
- FreeImage has zero dependencies.
 - Everything is bundled and versioned internally.
- By zero, I mean there isn't even a configure step.



```
% time make -f Makefile.osx
```

```
real    2m57.383s  
user    2m23.811s  
sys     0m21.026s
```

```
% time make -f Makefile.osx -j 4
```

```
real    1m47.911s  
user    2m36.636s  
sys     0m24.358s
```

```
% ./image_science.rb
```

```
taking screenshot for thumbnailing benchmarks
```

```
# of iterations = 100
```

	user	system	total	real
null_time	0.000000	0.000000	0.000000 (0.000033)
cropped	7.030000	0.670000	7.700000 (7.822275)
proportional	8.610000	0.490000	9.100000 (9.147624)



Thank You

seattle.rb