# RoslynDOM Release: aka, you are (I am) community

In April 2014, I gave an all day workshop on Roslyn metaprogramming. The morning was scheduled for metaprogramming basics, and the afternoon was Roslyn.

Everyone arrived after lunch all bright eyed and excited. I started explaining how to work with Roslyn, with frequent insertions of “I’m sure the community will develop libraries.” Crushed is the only word I can use for the change in demeanor on the faces of almost everyone in the room. Excitement went to “too hard for my team” and no amount of “the community will develop libraries” could compensate.

I taught a lot of Roslyn basics that day. But what kept rolling around in my head is that some of the most excited programmers about Roslyn, folks that had paid for my workshop, arrived at “too hard for my team.”

And “the community will develop libraries” kept rolling around in my head. “I am community” is comfortable when it involves speaking, writing, lobbying for features and training. I planned to release the research I’m doing. But a general purpose Roslyn library based on my beliefs of what we need? Well, I’m still not sure how much hubris is involved, but I think it’s a cliff I need to dive off.

Instead of moving headlong to releasing my new stuff, I stepped sideways and began yanking my utilities out into a separate, highly tested library. Life intervened as well, so it’s been more weeks than I hoped for, but I’m releasing the first alpha of my library on NuGet (search for RoslynDOM) and the code’s available here on GitHub.

I’ll follow up with the code first support library I showed in the workshop (the release was delayed to split out the library) and hopefully the new templates, assuming no brick walls appear in that research.

Thanks to the folks in my workshop for inspiring me.

## Why Roslyn is hard

## A couple of solutions

## Mentors, helpers and friends

Twice before I’ve released something to the wild, and it died.

I understand a lot more about Open Source than I did then, most particularly, I now know that I don’t know much about the mechanics of OSS.

# RoslynDOM Goals

* Hide Roslyn roundtrip details (attributes, fields, namespace nesting)
* Retain access to full Roslyn details, by providing direct access to syntax and symbol
* Provide simplified tree interface that is platform/compiler/underlying tree agnostic

# RoslynDOM Features

# RoslynDOM Samples