



Introduction to PDepend

Presented by Eli Taft
14 September 2017

www.quoininc.com

Boston

Charlotte

New York

Washington DC

Managua

Introduction

PDepend is a simple dependency management script written in Python. It might be useful to you if:

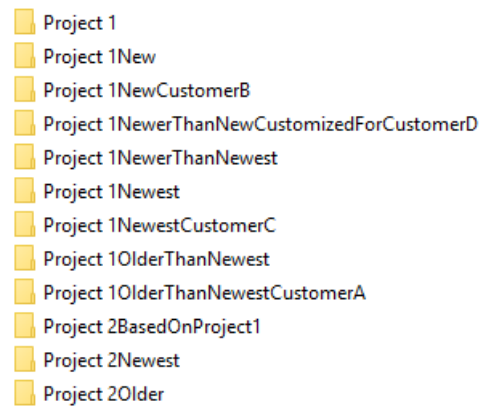
- You are working with a programming language / environment that does not have an easy way to share code across multiple applications already.
- You would rather stop pretending to like submodule or git subtree.

Background

- 25 software company
- One primary, self-trained developer who didn't like change
- Code duplication
- Project duplication
- Visual Basic 6

Background

- 25 software company
- One primary, self-trained developer who didn't like change
- Code duplication
- Project duplication
- Visual Basic 6



- Project 1
- Project 1New
- Project 1NewCustomerB
- Project 1NewerThanNewCustomizedForCustomerD
- Project 1NewerThanNewest
- Project 1Newest
- Project 1NewestCustomerC
- Project 1OlderThanNewest
- Project 1OlderThanNewestCustomerA
- Project 2BasedOnProject1
- Project 2Newest
- Project 2Older

Example Problem

- Bug found in the settings used to communicate with a medical device. Multiple projects use the same code, which has been copied and pasted. Over the years, modifications have been made to these copies independently, creating gloriously diverged code.

Example Problem

- As long as you're the only developer and it's all in your head, it's manageable.



The Need

- Single project for each project, with feature flags that can be toggled on/off
- Share code between projects.
 - Example: Device communication

The Options Considered

- DLL
- Git Submodule
- Git Subtree

Option 1 - DLL

- Tedious development process

Option 1 - DLL

- What version of the DLL was used?

```
$ git log
```

```
commit 62cf4912c97bcc64dac3b2f92729a71922bffdcc
```

```
Author: Eli Taft <eli@domain.net>
```

```
Date: Thu Aug 11 15:19:28 2016 -0400
```

```
Updating frmTrend32new and frmTrend75 to use new CGZipLibrary.dll for extracting the zip file from  
oximeter.commit
```

```
$ git log -p
```

```
diff --git a/project.vbp b/project.vbp
```

```
index 5a70844..d2e34e0 100644
```

```
-- a/project.vbp
```

```
+++ b/project.vbp
```

```
+Reference=*\G{293364AC-43F8-11D3-BC2D-4000000A2806}#2.0#0#CGZipLibrary.dll#CodeGuru  
Zip Library
```

Option 1 - DLL

- How can I debug?

```
'Unzipping all files
Call WriteLog("Using cGUnzipFiles dll...")
Dim oUnzip As CGUnzipFiles
Set oUnzip = New CGUnzipFiles
With oUnzip
    .ZipFileName = sFileSource
    .ExtractDir = sFileDest
    .HonorDirectories = True
    .OverWriteFiles = True
    If .UnZip <> 0 Then
        MsgBox .GetLastMessage
    End If
End With
Set oUnzip = Nothing
```

Option 2 - Git submodule

- Difficult to use
- Easy to forget to push submodules and push parent code depending on it
- Can get ugly with multiple developers pushing changes to the parent depending on different versions of the submodules
- Difficult to train new developers who are just learning Git to begin with

Option 3 - Git subtree

- Difficult to use
- Learning a new merge strategy
- Can get ugly with multiple developers making sure no one mixes super and sub-project code in commits.
- Difficult to train new developers who are just learning Git to begin with

What I Really Want...

- `pip install -r requirements.txt`

```
pylibmc==1.5.2  
pyparsing==2.0.3  
pypdf2==1.26.0  
pytz==2014.10  
regex==2016.3.2
```

```
-e git+https://github.com/jazzband/django-flatblocks.git@9717219d50e6082156efa85608842caa2574bbb4#egg=django_flatblocks-dev
```

What I Really Want...

- pip install -r requirements.txt

```
pylibmc==1.5.2  
pyparsing==2.0.3  
pypdf2==1.26.0  
pytz==2014.10  
regex==2016.3.2
```

```
-e git+https://github.com/jazzband/django-flatblocks.git@9717219d50e6082156efa85608842caa2574bbb4#egg=django_flatblocks-dev
```



What I Really Want...

- Share code across multiple projects
- Share in plain-text, not compiled
- Don't muddy the git commits
- Don't confuse developers
- Enforce documentation of versions at built-time

My Solution

- PDepend
 - <https://github.com/elimtaft/pdepend>
- Usage examples

TODO:

- Use Git's native command-line functionality to handle cloning and credentials in various ways.