Perforce Birds of a Feather

An introduction to the ‘P4’ Plugin and update on the latest features.

Paul Allen - Perforce Software

Jenkins World
2016

#JenkinsWorld

Introduction

- The Perfect Monorepo
- P4 plugin
- Credentials
- Workspace management
- Perforce Operations
- Polling, triggers and reviews
- P4 Groovy

The Perfect Monorepo

What is a Monorepo?

One Repo

One Repo to rule them all, One Repo to find them,
One Repo to bring them all and in the server bind them.

Credit: http://1nova.com/wallpapers/one-ring-to-rule-them-all-3/

One Store

Store all sources, projects and sub projects;
even artifacts, tooling, docs and test reports...

Credit http://gmunk.com

One History

Boldly go across source and time

Reproduce any source at any point in time

Credit: http://joe.siegler.net/2006/09/the-original-star-trek-to-get-an-upgrade-pt-2

Global Access

Any file any where

Fast Global Distribution

Fine grain protections

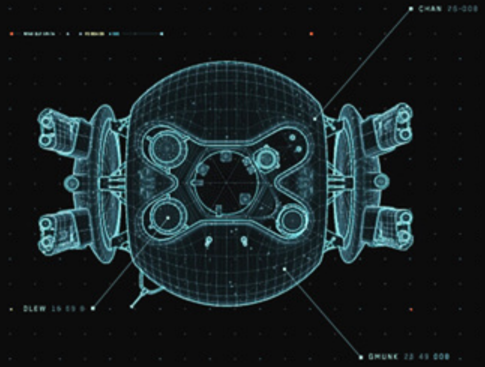
Credit: http://gmunk.com

P4 Plugin

Perforce plugin support for Jenkins

Disambiguation

- Community 'Perforce' plugin
  https://wiki.jenkins-ci.org/display/JENKINS/Perforce+Plugin
- Perforce Supported 'P4' plugin
  https://wiki.jenkins-ci.org/display/JENKINS/P4+Plugin



Credit: http://gmunk.com/OBLIVION-GFX

The ‘P4’ plugin

- Why

  Access the power of Perforce from within Jenkins

- P4Java

  Pure Java solution

  No ‘P4’ executable to install and keep up-to-date.

- Latest features

  Streams, sync and clean up options

Credentials

Connecting to Perforce…

Credentials

- **Perforce Credentials**
  - Shared Credentials
  - Managed from one location
  - Independent from the Job configuration
  - Update connection details in one location

- **Credential Types**
  1. Perforce Password Credential
  2. Perforce Ticket Credential

Credentials

- Connection information
  - Username/Password
  - Perforce address and port
  - `workshop.perforce.com:1666`
- Password credentials
  - Username
  - Password
  - ID (useful to reference in the DSL)
  - Description

Credentials

- **Ticket based security**

  Perforce generated Ticket String

  ```
  $ p4 login -p
  4E034A8812F81B38229BF8FA62B0FEB1
  ```

  Location of Perforce P4TICKET file

  ```
  /home/pallen/.p4ticket
  ```

- **SSL and Trust**

  Check the SSL box to add the `ssl:` part to P4PORT

  Use `p4 trust` or click test to get the fingerprint

Workspace Management

Jenkins and Perforce Workspaces

Workspaces

- Jenkins workspace

  Location of files (on the master or slave) for Jenkins to build

- Perforce workspace

  Location where Perforce will manage the versioned an non-versioned files

- Shared root

  Recommend Perforce and Jenkins workspaces share the same root

  One Jenkins workspace to map to one Perforce workspace

Workspaces

- Workspace Configuration

  Streams Workspace
  Manual Workspace
  Template Workspace
  Static Workspace (predefined)
  Spec Workspace (spec depot or file)

```
workspace: [$class: 'StreamWorkspaceImpl',
  charset: 'none',
  format: 'jenkins-${NODE_NAME}-${JOB_NAME}',
  pinHost: false,
  streamName: '//streams/st1-main']
```

Workspace

View Mapping

```
+---AceProject                //depot/AceProject/...
|    +---src
|    +---test
|    +---~~docs~~                  -//depot/AceProject/docs/...
|    +---~~libs~~                  -//depot/AceProject/libs/...
+---BlastProject
|    +---~~src~~
|    +---~~test~~
|    +---~~docs~~
|    +---libs                //depot/BlastProject/libs/...
|    |    +---blast.jar
```

Workspace

```
//depot/AceProject/...          //{client}/Ace/...


-//depot/AceProject/docs/...    -//{client}/Ace/docs/...
-//depot/AceProject/libs/...    -//(client}/Ace/libs/...



//depot/BlastProject/libs/...   //{client}/Ace/plugins/...
```

Workspace

Perforce Depot

```
+---AceProject
|    +---src
|    +---test
|    +---~~docs~~
|    +---~~libs~~
+---BlastProject
|    +---~~src~~
|    +---~~test~~
|    +---~~docs~~
|    +---libs
|    |    +---blast.jar
```

Jenkins Workspace

```
+---Ace
|    +---src
|    +---test
|    |
|    |
|    +---plugins
|    |    |
|    |    |
|    |    |
|    |    |
|    |    +---blast.jar
```

Streams

- Virtual Streams to manage files
- Configured outside of Jenkins
- Check the Generated Mapping

Variable expansion

- Built in Variables

  /env-vars.html

- Perforce Variables

P4_CHANGELIST	P4_CLIENT
P4_PORT	P4_USER
P4_TICKET	

- Workspace

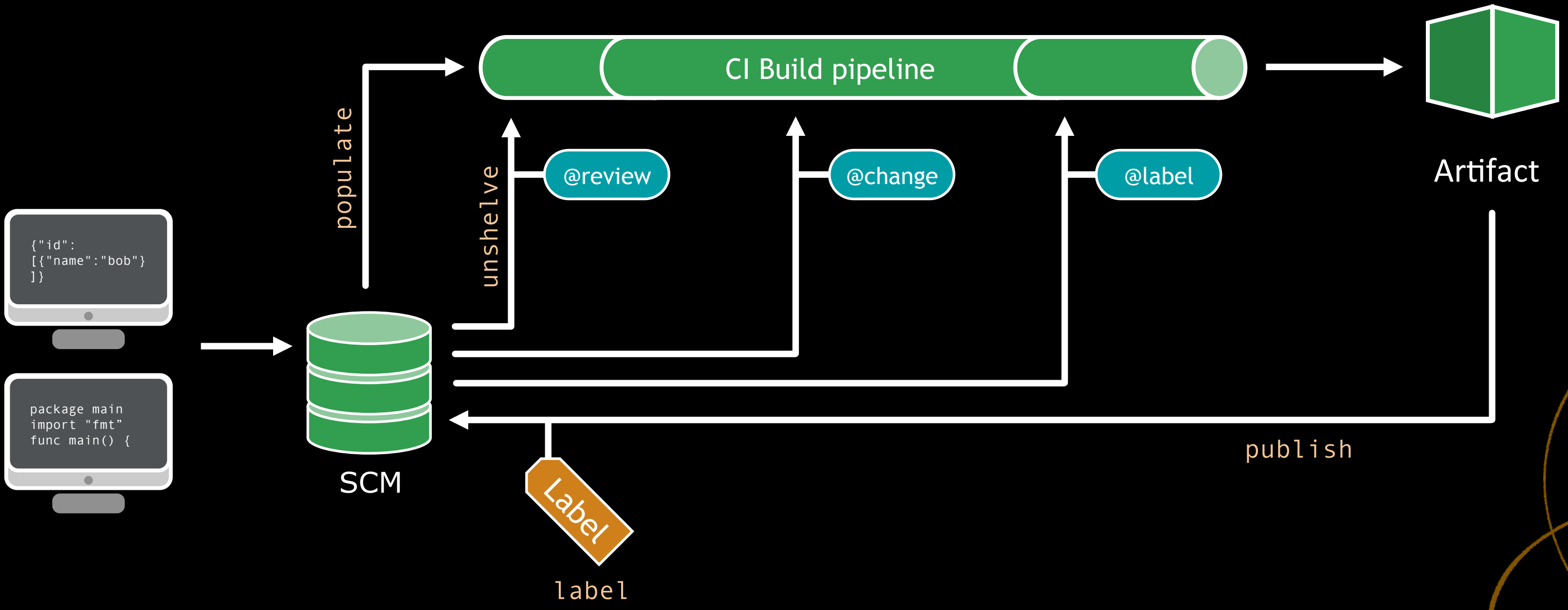
Name	`jenkins-${NODE_NAME}-${JOB_NAME}`
View	`//depot/… //jenkins-${NODE_NAME}-${JOB_NAME}/…`

Perforce Operations

Checkout, Build and Post Build Steps

CD Pipe

- SCM Operations



```
{"id":
[{"name":"bob"}
]}
```

```
package main
import "fmt"
func main() {
```

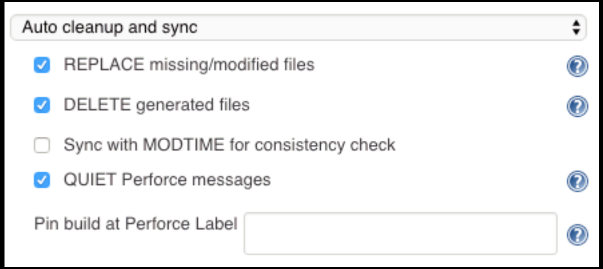
SCM Operations

- **Populate** `p4sync / checkout`
  Synchronize the files in the Workspace prior to build.
- **Unshelve** `p4unshelve`
  Unshelve code into the Workspace prior to build.
- **Publish** `p4publish`
  Submit files back into Perforce, post build.
- **Label** `p4tag`
  Automatic label against the populated files in the Workspace, post build.

Populate

- Auto Cleanup and Sync

```
populate:[$class:'AutoCleanImpl',
   delete:true, replace:true,
   modtime:false, quiet:false, pin:''
]
```



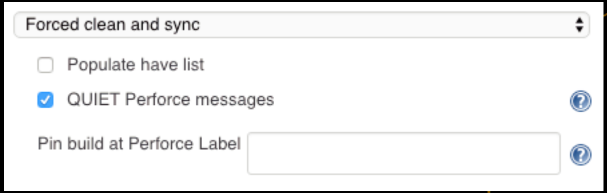
Auto cleanup and sync

- [✓] REPLACE missing/modified files
- [✓] DELETE generated files
- [ ] Sync with MODTIME for consistency check
- [✓] QUIET Perforce messages

Pin build at Perforce Label

- Force Clean and Sync

```
populate: [$class: 'ForceCleanImpl',
   have:false, pin:'', quiet:true
]
```



Forced clean and sync

- [ ] Populate have list
- [✓] QUIET Perforce messages

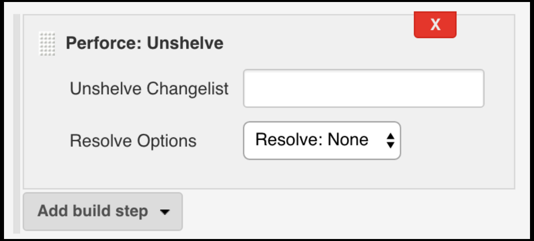
Pin build at Perforce Label

Unshelve

- Unshelve Build Step

  Unshelve the change as a Build step defined in the Job
  Files are unshelved and resolved prior build.

```
p4unshelve resolve:'at', shelf:'12345'
```



Perforce: Unshelve

Unshelve Changelist

Resolve Options: Resolve: None

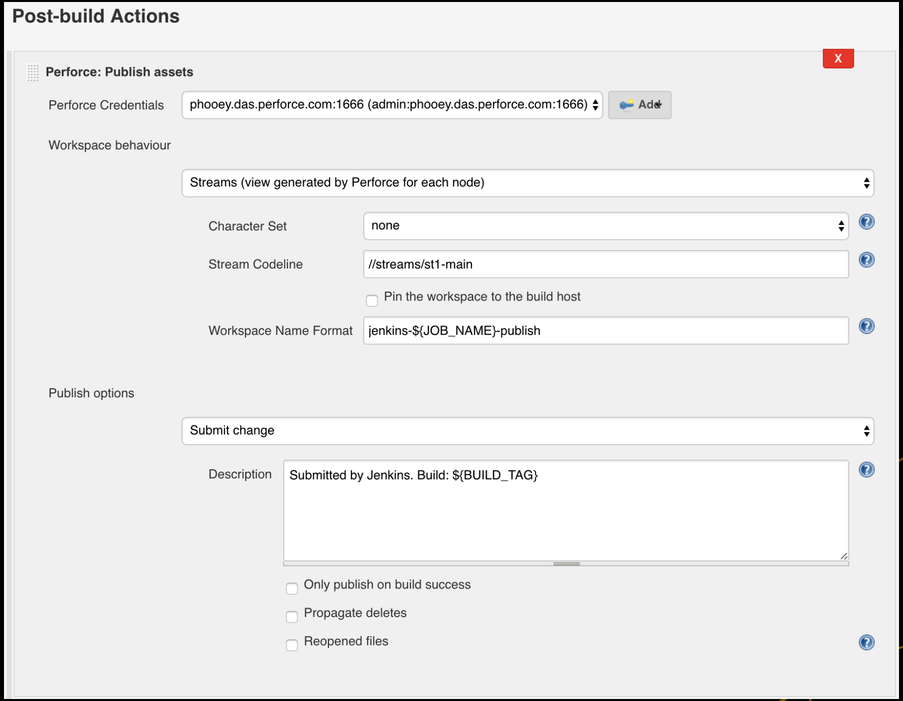
Publish

- Shelve or Submit a change
- Connection & Workspace
- Use a narrow view
- Virtual stream
- Read/Write access for files

  Set Workspace option `ALLWRITE` or use filetype `+w`

Publish

```
p4publish credential:'phooey1666',
 publish: [
  $class:'SubmitImpl',
  delete:false,
  description:'Build: ${BUILD_TAG}',
  onlyOnSuccess:false,
  reopen:false],
 workspace: [
  $class:'StreamWorkspaceImpl',
  charset:'none',
  format:'jenkins-${JOB_NAME}-publish',
  pinHost:false,
  streamName:'//streams/st1-main']
```



Post-build Actions

Perforce: Publish assets

Perforce Credentials: phooey.das.perforce.com:1666 (admin:phooey.das.perforce.com:1666)

Workspace behaviour: Streams (view generated by Perforce for each node)

Character Set: none

Stream Codeline: //streams/st1-main

☐ Pin the workspace to the build host

Workspace Name Format: jenkins-${JOB_NAME}-publish

Publish options: Submit change

Description: Submitted by Jenkins. Build: ${BUILD_TAG}

☐ Only publish on build success

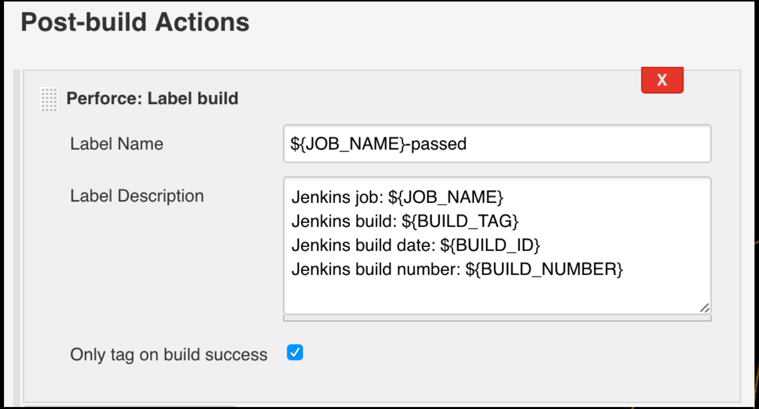
☐ Propagate deletes

☐ Reopened files

Label

- Automatic label
  - Label on success option
  - Uses Populate Client's View
- Name & Description

```
p4tag rawLabelName:'${JOB_NAME}-passed',
rawLabelDesc:'''Jenkins job: ${JOB_NAME}
Jenkins build: ${BUILD_TAG}
Jenkins build date: ${BUILD_ID}
Jenkins build number: ${BUILD_NUMBER}'''
```



Post-build Actions

Perforce: Label build

Label Name: ${JOB_NAME}-passed

Label Description:
Jenkins job: ${JOB_NAME}
Jenkins build: ${BUILD_TAG}
Jenkins build date: ${BUILD_ID}
Jenkins build number: ${BUILD_NUMBER}

Only tag on build success ✓

Polling, Triggers and Reviews

Still polling?

Polling… if you must

- Polling build Filters
  - Exclude changes from Depot path
  - Exclude changes from user
  - Exclude changes outside view mask
  - Poll on Master using Latest Build
  - Polling per Change
- Workspace modes
  - Workspace to check the build - Preview check Only (sync –k)

Triggers

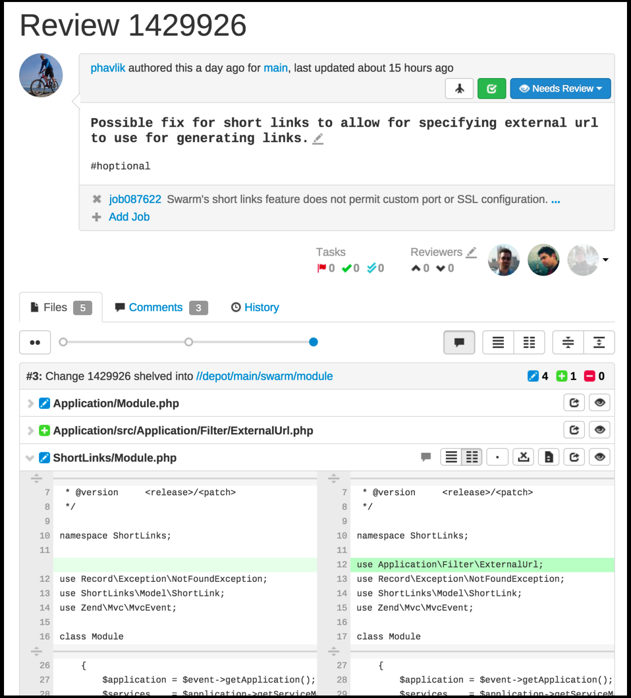
- Perforce triggered build
- Subscribe Job (P4 Trigger)

```
curl --header 'Content-Type: application/json' \
  --request POST \
  --data "payload={change:200,p4port:\"perforce.com:1666\"}" \
  http://jenkins:8080/p4/change
```

- Swarm (P4 Review)

  Build triggered by Perforce Swarm.
  Review or Change unshelved into workspace prior to build

Swarm



Review 1429926

phavlik authored this a day ago for main, last updated about 15 hours ago

Needs Review

Possible fix for short links to allow for specifying external url to use for generating links.

#hoptional

job087622 Swarm's short links feature does not permit custom port or SSL configuration. ...

Add Job

Tasks 0 0 0 Reviewers 0 0

Files 5 Comments 3 History

#3: Change 1429926 shelved into //depot/main/swarm/module 4 1 0

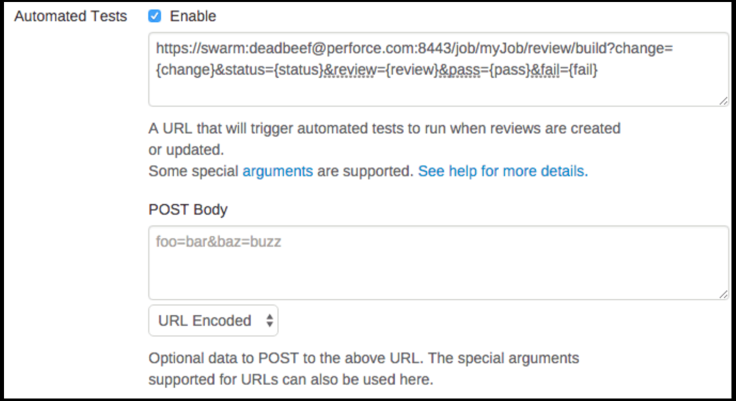
Application/Module.php

Application/src/Application/Filter/ExternalUrl.php

ShortLinks/Module.php

```
 * @version     <release>/<patch>
 */

namespace ShortLinks;

use Application\Filter\ExternalUrl;
use Record\Exception\NotFoundException;
use ShortLinks\Model\ShortLink;
use Zend\Mvc\MvcEvent;

class Module

    {
        $application = $event->getApplication();
```

```
[POST]
https://swarm:deadbeef@perforce.com:8443
    /job/myJob/review/build
    ?change={change}&status={status}&review={review}
    &pass={pass}&fail={fail}
```



Automated Tests ✓ Enable

https://swarm:deadbeef@perforce.com:8443/job/myJob/review/build?change={change}&status={status}&review={review}&pass={pass}&fail={fail}

A URL that will trigger automated tests to run when reviews are created or updated.
Some special arguments are supported. See help for more details.

POST Body

foo=bar&baz=buzz

URL Encoded

Optional data to POST to the above URL. The special arguments supported for URLs can also be used here.

P4 Groovy

P4 command access for Groovy

P4Groovy

- P4Groovy object

  Credential

  Workspace

```
ws = [$class: 'StreamWorkspaceImpl',
  charset: 'none', format: 'jenkins-${JOB_NAME}',
  pinHost: false, streamName: '//streams/projAce']

p4 = p4(credential: 'phooey', workspace: ws)
```

P4Groovy Methods

- Run

  Requires: command, arguments (‘,’ separated String)
  Returns: tagged output (specifically Map<String, Object>[])

```
p4.run(’changes', '-m5, //...')
```

- Getters

```
p4.getUserName()
p4.getClientName()
```

P4Groovy Methods

- Fetch

  Requires: spec type, spec id
  Returns: a spec as a Map

```
client = p4.fetch('client', 'my_ws')
```

- Save

  Requires: spec type, the spec as a Map
  Returns: tagged output (specifically Map<String, Object>[])

```
p4.save('client', client)
```

P4Groovy

- Examples/Demo

P4Groovy

- Combine Steps with P4Groovy

  Populate is made up of several steps not just sync
  Use P4Groove for custom operations e.g.

```
node() {
    ...
    job = p4.fetch('job', 'job000006')
    desc = job.get('Description')
    desc = desc + env.BUILD_URL
    job.put('Description', desc)
    p4.save('job', job)
}
```

Questions

?

# Jenkins World

2016

#JenkinsWorld