Power Tooling for DevOps at Scale

PERFORCE

Using Helix Version Control with Git

Enterprise scaling in an asymmetric VCS environment creates challenges at every stage of your Continuous Delivery pipeline.

The Perforce Helix Version Control System (Helix VCS) allows you to manage your source code and build assets on a global level, while also supporting Git environments at scale.

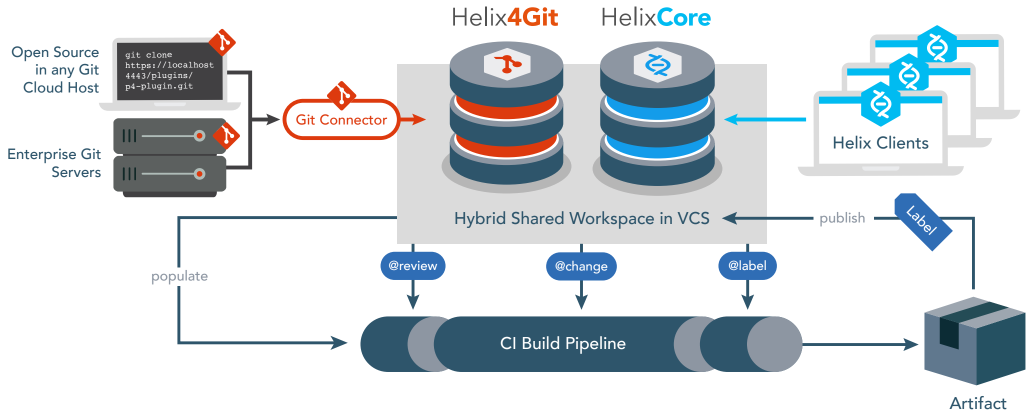
Helix VCS consists of Helix Core, the versioning engine known for its enterprise-class speed and scalability, and a new solution called Helix4Git, which allows enterprise teams to scale projects with multiple Git repositories.

Helix4Git enables faster performance and scale without the large investments in infrastructure that many Git enterprise environments require. With Helix4Git, you can achieve:

- **40-80%** faster builds than any other Git solution without infrastructure upgrades.
- **Unified, hybrid workspace** for Git source code and binary files.
- **High-performance** parallel connection of hybrid workspaces to your CI tool.
- **Simplified build process** for large projects by syncing multiple Git repos.

Better Git for CI/CD

Helix4Git solves common Git scalability issues and lets you execute high-performance builds across multiple Git repos. Leverage edge/commit technology to scale your build farm, offloading the burden from master CI servers.



```
git clone
https://localhost
4443/plugins/
p4-plugin.git
```

When Helix4Git is combined with Helix Core, you can also incorporate assets that are too large for effective handling in Git–such as audio, video, large graphics files, and other binaries. The result is a simplified build process that encompasses all your product development needs.

If you are using third-party source code hosting solutions or working with open source, Helix4Git lets you mirror the code from any Git solution, whether Cloud or On-Premises.

High-Performance Multi-Repo Builds with Jenkins

Jenkins and Helix VCS are a strong combination for solving enterprise CI/CD challenges.

Perforce offers a P4 Plugin for Jenkins that provides full Helix VCS support and is designed to let you easily implement Jenkins builds and automation, incorporating the full Helix VCS solution, including Helix Core and Helix4Git.

With both solutions, you can combine code and artifacts managed in Helix Core alongside Git assets, and implement a high-performance build environment that integrates nicely with Jenkins.

P4 PLUGIN

The P4 Plugin for Jenkins is written using P4Java, a fully native Java interface to the Helix VCS server. The plugin allows users to easily populate their workspaces with source files and assets from Helix VCS. The plugin supports the Jenkins scm-api, allowing users to write Jenkinsfile, Pipeline scripts, and enabling support for Jenkins MultiBranch.

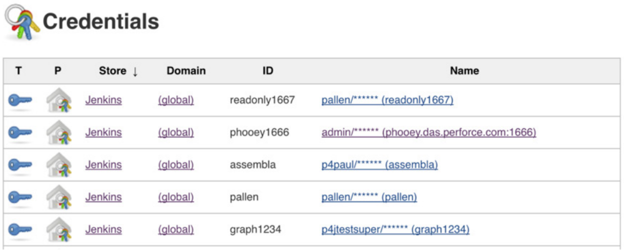
The P4 Plugin for Jenkins supports:

- Credential authentication
- Client workspace management
- Synchronization behavior
- Polling and filtered builds
- External review tools (such as Helix Swarm)
- Change list browsing
- Labeling builds (tagging)
- External repository browsers (P4Web, Helix Swarm)
- Submit assets back to Perforce Helix (Publish)

P4 PLUGIN CONFIGURATION

Setup and configuration is easy with the use of the P4 Plugin credential support to help manage your connection to Helix VCS edge-commit or proxy servers. Simply provide Helix VCS server port details and login information, then reuse the credential for all your Jenkins Jobs.

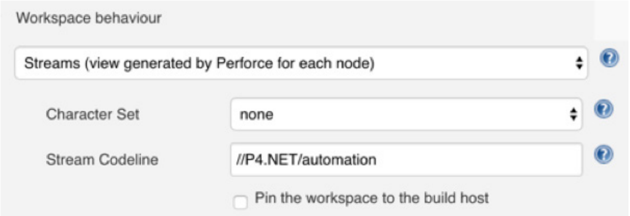
Helix VCS workspace configuration is simple with dropdown options for basic use cases where a single path is mapped and flexibility for more complex view mapping. Helix VCS sync behavior can be easily controlled through the populate options, where workspaces can be synced incrementally or wiped clean.



Credentials

T	P	Store ↓	Domain	ID	Name
🔑	🏠	Jenkins	(global)	readonly1667	pallen/****** (readonly1667)
🔑	🏠	Jenkins	(global)	phooey1666	admin/****** (phooey.das.perforce.com:1666)
🔑	🏠	Jenkins	(global)	assembla	p4paul/****** (assembla)
🔑	🏠	Jenkins	(global)	pallen	pallen/****** (pallen)
🔑	🏠	Jenkins	(global)	graph1234	p4jtestsuper/****** (graph1234)

Credentials dialog in Jenkins



Workspace behaviour

Streams (view generated by Perforce for each node)

Character Set: none

Stream Codeline: //P4.NET/automation

□ Pin the workspace to the build host

Workspace dialog in Jenkins

PIPELINE SUPPORT

The P4 Plugin provides DSL wrappers for common workflows such as ‘p4sync’ to sync files from the Helix VCS server, ‘p4tag’ to label/tag source files used in a build, ‘p4publish’ to shelve or submit content back to the Helix VCS server, ‘p4unshelve’ to unshelve files outside of a review, and ‘p4cleanup’ to remove a Helix VCS client workspace and clean up its files and metadata.

ABOUT P4GROOVY

For advanced users who require access to more Helix VCS commands, there is P4Groovy, a set of objects for performing tasks with Jenkins that are not possible with the P4 Plugin. P4Groovy is a Groovy interface to P4Java that allows you to write Helix VCS commands in the pipeline DSL.

P4GROOVY SETUP

To allow P4Groovy code to be executed, uncheck the ‘Use Groovy Sandbox’ checkbox under the Pipeline script.

The P4Groovy setup creates a ‘p4’ object. For example:

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

// Create object
def p4 = p4(credential: ‘phooey’, workspace: ws)
```

This code can be created for you using the ‘p4: P4 Groovy’ pipeline syntax snippet generator.

RUNNING A COMMAND

The run command takes a comma separated list of Helix VCS commands and then arguments. For example:

```
p4.run(‘changes’, ‘-m5’, ‘//depot/path/...’)
```

The command will return an array of Maps (specifically Map<String, Object>[]). For example:

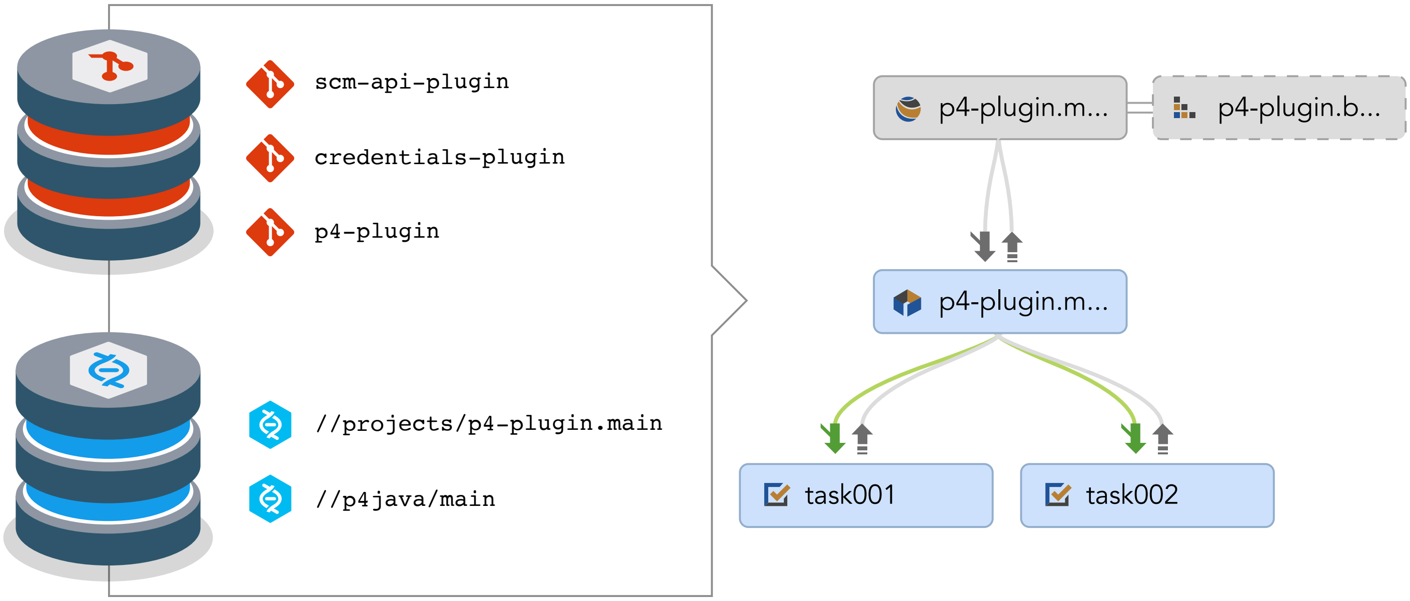
```
def changes = p4.run(‘changes’, ‘-m5’, ‘//depot/path/...’)
```

You can iterate through the response by iterating through each array item and extracting the keys. For example:

```
for(def item : changes) {
 for (String key : item.keySet()) {
    value = item.get(key)
    println
    (“Key: “ + key + “ Value: “ + value)
    }
}
```

FOR MORE INFORMATION

To find out how to install and use P4 Plugin for Jenkins, or if you are an advanced user who wants more access to P4 commands by using P4Groovy, access the documentation by visiting https://github.com/jenkinsci/p4-plugin.



ADDING HELIX4GIT

Helix4Git is free to use for up to 10 Git repos. Just add the Git repos to your stream or map them to your workspace view, then select the GraphHybrid option for Populate.

FREE TO GET STARTED

Helix Core and Helix4Git are free for small teams up to 5 users and 10 Git repos. To get started today, visit www.perforce.com/free-small-teams.

About Perforce

Enterprises across the globe rely on Perforce to build and deliver complex digital products faster and with higher quality. Perforce is best known for its highly scalable version management and collaboration platform that securely manages change across all digital content – source code, art files, video files, images, libraries - while supporting the developer and build tools your teams need to be productive, such as Git, Visual Studio, Jenkins, Adobe, Maya and many others. Perforce also offers complete project lifecycle management tools to accelerate a project's delivery cycle by linking the requirements, test plans, source code, and helpdesk in an integrated platform. Perforce is trusted by the world's most innovative brands, including Pixar, NVIDIA, Scania, Ubisoft, and VMware. The company has offices in the US, the United Kingdom, Germany, Canada and Australia, and sales partners around the globe. For more information, please visit www.perforce.com