



VS



PERFORCE

COMPARISON

Helix Core vs. Microsoft TFS/TFVC

In this document, we will compare Perforce Helix Core to Microsoft Team Foundation Server (TFS.) Specifically, we will look at the TFVC (Team Foundation Version Control) functionality. With TFVC, your code is kept on the TFS server, usually on-premises. Developers often work in the popular Microsoft Visual Studio application on their own Windows workstation. They connect to the TFVC server to collaborate with other developers.

Changes in the Microsoft World

When TFS version control originally shipped, it was exclusively an on-premises solution - Visual Studio 2005 Team System. Since then, of course, [Microsoft has made regularly made significant updates.](#)

In 2016, Microsoft released a cloud-based service, Visual Studio Team Services (VSTS). This marked a different direction for TFS on-premises customers. More recently, VSTS has been rebranded as Azure DevOps, with a variety of developer tools available on Microsoft's Azure cloud platform. Although Azure DevOps continues to provide support for TFVC repositories, the Microsoft vision has a new iteration.

Move Over TFS

Microsoft's acquisition of GitHub – the highly-adopted open source code-hosting and collaboration site – may provide clues to the company's version control strategy. But analysts believe that the actual VCS features of GitHub are less important to Microsoft.

What's more important? Winning the hearts and minds of some 31 million+ GitHub users, and keeping them away from the very popular AWS cloud.

Because Microsoft's new Azure and Git strategy is so different from the way TFS and TFVC users work today, teams have a big decision looming. Should they move to the cloud and embrace Microsoft's DevOps pipelines and other Azure-based technologies? Or should they choose another VCS that will work on-premises and deliver on their DevOps priorities? Microsoft will continue mainstream support of TFS through 2023. Although some older versions (pre 2016) that many teams are still using are no longer supported.

TFS has several applications in addition to version control. These Application Lifecycle Management (ALM) features include task tracking, build management, and testing. Here we will focus entirely the VCS functionality. But if you are looking to replace your entire TFS solution, [investigate upgrading to Helix ALM + Helix Core.](#)

Contents

SUMMARY	3
WHY SWITCH TO HELIX CORE	3
COMPARISON MATRIX	4
PERFORMANCE AND SCALABILITY	5
BRANCHING AND MERGING	5
WORKING WITH CODE	6
CODE REVIEW	6
RELEASE MANAGEMENT	7
WORKING WITH GIT	7
GLOBAL TEAMS	8
SECURITY	8
VISUAL CLIENT	9
CONCLUSION	9

Summary

Version control has become a key component for the foundation for DevOps. A highly performant and scalable version control system is critical to achieving accelerated Continuous Integration and Continuous Delivery (CI/CD) workflows. Legacy TFS lacks the features and performance to provide the foundation for DevOps success. And new Azure DevOps is joined at the hip with the Microsoft cloud, limiting your deployment choices and development tools.

Common frustrations with TFVC include:

- **Performance and Scalability Limitations:** The system slows down as the number of users, repositories, files, branches, and revisions increases.
- **Total Cost of Ownership:** Lack of tooling and infrastructure, and the cost in time of system administration, increases the total cost of ownership.
- **Time-Consuming Operations:** Developers, admins, and build engineers spend too much time on low value tasks with TFS.
- **Difficult Release Management:** CI/CD is harder than it should be, especially with multiple release branches. This can lead to drawn-out code freezes and potentially delayed releases.
- **Coexistence with Git:** The new Azure DevOps has both TFVC and Git support, but the two are very separate. In a nutshell, you must choose between TFVC and Git, not both.
- **Lack of Support For Global Teams:** TFS is a single-site installation with no geographic replication. Teams need to rely on infrastructure to support geographically diverse sites. And even with a robust network, daily tasks can cause performance delays. Developers waiting means productivity suffers.

Why Switch to Helix Core?

This document examines Helix Core (version 2018.2) and Microsoft Team Foundation Server 2018 TFVC. Major VCS features are compared. The resulting analysis shows that Perforce's robust version management system can enhance team productivity with significant advantages in scalability and total cost of ownership. Helix Core easily handles basic and advanced version management operations more reliably. Perforce and Helix Core give you DevOps at scale with innovative tools, simpler installation and administration, without deployment, workflow and tool selection limitations.

Let's examine why Helix Core might be a better choice. Making the switch can give you:

SUPERIOR PERFORMANCE

GLOBAL SCALABILITY

MODERN CODE REVIEW WORKFLOWS

DEVOPS AT SCALE

ADDED GIT SUPPORT

BETTER SECURITY WITH GRANULAR PERMISSIONS

Comparison Matrix

CAPABILITY	 Team Foundation Server	 HelixCore
PERFORMANCE AND SCALABILITY	TFVC is installed on two to five Windows servers, depending on features and performance needs. For teams with multiple terabytes of assets, this is a severe limitation.	Helix Core effortlessly supports thousands of users and millions of daily transactions. And it accelerates CI/CD with lightning-fast commits and concurrent builds.
BRANCHING AND MERGING	It's easy to branch with TFVC's user interface. Merging is also easy, but can be slow. The inability to handle non-contiguous changesets can lead to delayed deployments. And, over time, this can create a less stable codebase.	Helix Core uses Streams – a simple, intelligent, and automatic branching mechanism. Advanced merging provides flexibility, while at the same time prevents developers from making mistakes. Rules are enforced based on the intended purpose of the branches.
WORKING WITH CODE	In TFVC, code is copied to a folder (workspace) on a workstation. It's a 1:1 relationship. Code can then be edited and when submitted, a changelist is created.	Helix Core's Client Workspace gives you a window into the code on the server. It empowers development and provides flexibility. Code is not constrained to a single folder. And submitted work is organized using uniquely numbered changelists.
CODE REVIEW	Code reviews are invoked using Visual Studio's Team Explorer. Code is shared in TFVC. Then a review is requested in Microsoft Visual Studio.	Helix Swarm is a browser-based code review and collaboration interface that is included at no additional cost with Helix Core. It is also integrated with Microsoft Visual Studio.
RELEASE MANAGEMENT	With TFVC, the relationship between branches and trunk is not always clear. It is up to your team to create a naming scheme and documentation to keep track of the changes.	Streams identifies branches as mainline, development, and release. This hierarchy helps to easily identify changes that need to be propagated into multiple codelines.
WORKING WITH GIT	With Visual Studio, you can work with either Git, or TFVC– but the two are separate. Having teams collaborating on code requires selecting one or the other, not both.	Helix4Git is an add-on option that allows you to manage Git repos in the high-performance Helix Core server. Developers can use Git seamlessly, while Helix4Git provides high performance DevOps integrations for CI/CD.
GLOBAL TEAMS	TFVC doesn't have a native capability for teams to collaborate across multiple sites. There is no concept of replication.	Helix Core offers an unparalleled set of technologies including caching and advanced replication options. With Perforce Federated Architecture, global teams can access and manage files at LAN speeds.
SECURITY	TFS security is based on Windows authentication, users, and groups.	Helix Core has powerful file-level access control. It integrates with your preferred Identity Provider (IdP) to secure your IP via clients, plugins, and even the command line.
VISUAL CLIENTS	TFVC seamlessly integrates with Visual Studio.	P4VS is a plugin for Visual Studio that has been downloaded more than 350,000 times. The plugin supports the most often used features of Helix Core.

Performance and Scalability

TFS

TFS requires multiple Windows servers to support its components. Installation includes TFS itself and your Microsoft SQL server installed as a single instance. Because of this, many companies have experienced limitations trying to support a large user base, multiple teams, high frequency of automation requests resulting from CI/CD and delivering high volume of data to users.

Despite the complexity of a TFS deployment, simple data replication is not supported. For teams with multiple terabytes of assets, sharing may be limited. It may be necessary to simply copy code from one server to another for reuse, which is an administrative burden.

HELIX CORE

Helix Core outperforms TFS in several performance benchmarks, such as the number of cores, memory, OS, and IO – even in a single server configuration. Speed has always been a key part of Helix Core. The ability to scale linearly ensures that no matter the number of revisions, or the file size, performance never dips.

Helix Core has been deployed successfully in environments with tens of thousands of users, petabytes of versioned content, and tens of millions of revisions. And Perforce Federated Architecture provides many options for scaling out a local and global deployments.

Branching and Merging

TFS

TFVC branching and merging has evolved over the years. Although it has become more robust, merging remains slow. Like many VCS systems built a few years ago, development and release management models are built by convention. A directory structure is needed to define different branch types. This can limit the adoption of automated release management processes due to ineffective labeling.

Visualization for branches is available as a graph in TFS, but it does not indicate the stability of different branches or the intended merge pathways. For example, there is no indication that a “rel-1.0 branch” could be more stable than “my-task branch.” And there is no indication whether the rel-1.0 branch should accept a change from its parent. There are also no visual indicators to show that merges are pending between branches. Limited visibility can create merge conflicts that soak up developers’ time and severely impact productivity.

HELIX CORE

Helix Core’s “next generation” branching technology, Streams, is both powerful and flexible. It provides a lightweight model capable of branching thousands of files rapidly, while retaining a complete branch and merge history. Instead of manually tracking all changes across branches, users can rely on Helix Core. Using Streams, a product architect can define the relationship between streams, the modules of a product, and the direction of merges. This simplifies and automates many routine user operations.

Automatically merging file changes across multiple branches creates fewer conflicts to resolve. This enables a variety of development scenarios such as client-specific versions, experimental branches, personal or task branches, and classic release branching patterns. Revision Graph, a built-in graphical tool, displays the detailed branching history of each file for easy visualization.

(CONTINUED ON NEXT PAGE)

HELIX CORE (CONT.)

Helix Core's merge engine can easily handle the most complex merge scenarios including refactoring, indirect merges, and non-content changes. Because of this robustness, automation can confidently be incorporated into most branching and release management workflows.

Working with Code

TFS

TFVC workspaces consist of simple 1:1 mapping between repository directories and the local file system. New workspaces can be created using an existing workspace as a template. TFS's changesets are roughly equivalent to Helix Core changelists.

They show atomic operations and sync points in the repository. However, TFVC does not allow the creation of pending changesets to organize work. Changesets cannot be created until check-in. Shelvesets can be used to store pending work. But submitting a shelveset checks in the file as stored in the shelf. The file then sits on the client, which can create problems for other developers looking to make changes on the server.

HELIX CORE

Helix Core workspace views are flexible and granular, allowing the selection, exclusion, or remapping of files. For example, a workspace could exclude all build artifacts (.obj) and route region-specific translation files to a different folder for localized builds.

Workspaces can be created manually or use existing workspaces as templates. When working with Streams, workspaces can be generated automatically with only the correct set of files needed for the project/task. And when creating a pending changelist, work is effectively organized into several tasks in a workspace. This simple feature helps users avoid submitting unrelated files by mistake.

Code Review

TFS

Depending on the TFS/TFVC version, users may initiate code reviews from either Visual Studio, or a work item in TFS. Code reviews consist of comments on code from team members. One downside to TFS on-premises installation is the lack of browser-based code review for projects. While the VSTS interface does support web-based code reviews for TFVC and for Git projects, they are separate.

HELIX CORE + HELIX SWARM

Helix Swarm is a code review and collaboration tool with modern features such as code browsing, inline commenting, suggested tasks, review "upvoting," and final work submission. Swarm's web-based interface supports extremely large teams, and a large number of simultaneously open reviews.

It provides a strong cornerstone for Continuous Integration (CI). Key features include workflow management at scale, support for build-runners, agile project management integration, customizable dashboards, and notifications. Helix Swarm part of the Helix Core license and is accessed within Visual Studio via the P4VS plugin.

Release Management

TFS

With TFVC, you can tag/label each release branch or bug fix. However, the relationship between branches and the trunk is not always clear. It is up to the team to create a naming scheme and documentation to keep track of the flow of change.

HELIX CORE

Helix Core also supports labeling. It is a popular feature for use with automation, such as Jenkins. But Helix Core's strongest release management feature is Streams. Developing using Streams simplifies automation because branches are easily identified as mainline, development, and release.

This hierarchy makes it possible to automatically determine changes that need to be propagated and to which codeline(s). Streams' structure supports agile/DevOps environments. It streamlines release management for complex product development cycles and empowers high release frequencies.

Working with Git

TFS

Microsoft officially began supporting Git in TFS and Visual Studio back in 2013, alongside TFVC. Git came to the forefront with the release of VSTS and Visual Studio 2017. Today, Azure DevOps has replaced VSTS. And while Git is in first-class, TFVC takes a backseat. There is still support for TFVC repos and workflows, but no ability to co-exist with Git. The primary difference to developers? Git has a pull-request workflow versus the shelving/code review workflow in TFVC.

HELIX CORE

Helix Core has a unique add-on technology option called Helix4Git, which stores Git repos natively in a high-performance Helix Core server. It lets development teams take advantage of the benefits of both the Helix Core workflow and Git. Teams can build products using both solutions.

Developers can use Git seamlessly, while Helix4Git provides high-performance DevOps integrations. Whether it is binary artifacts, graphics, or video from Helix Core, it can be seamlessly combined with code from Git in build pipelines. Perforce also has a product called Helix TeamHub, which offers an enterprise-grade web code hosting and collaboration platform for Git.

Global Teams

TFS

TFS is a single-site installation with no built-in replication. Enterprises need to rely on infrastructure to support geographically diverse teams. Teams are dependent on network infrastructure to access TFS and TFVC servers. For teams with multiple terabytes or petabytes of assets, this is a severe limitation in all dimensions of scale.

HELIX CORE

Perforce Federated Architecture is robust native replication technology. It seamlessly scales to meet global demands. This provides a real “single source of truth” that lets developers work together, even when teams are located in diverse geographic locations.

Helix Core provides options such as proxy servers, brokers, and a variety of replica types that can be tailored to satisfy the most complex environments. In addition to supporting teams, replica servers can also support advanced automation processes for accelerated DevOps, whether locally or in remote locations.

Security

TFS

TFS offers granular permissions and groups, and it is integrated into Microsoft Active Directory (AD) for identifying users. TFVC permissions are administered separately after your TFS project has been configured. Administrators have explicit permissions to control the use of features, for example branching and merging, by contributors.

HELIX CORE

One of the most compelling security features of Helix Core is that you can control access down to a single file. Individual and group contributors can also have restricted functionality on a granular basis. Helix Core can also be integrated with your organization’s Identity Provider (IdP) using Helix Authentication Service (HAS). It currently supports the OpenID Connect and SAML 2.0 authentication protocols. It can be used in conjunction with Helix Core clients and plugins, and the command line. This service is internally certified with Microsoft Azure Active Directory (AAD), Okta, and Google Identity. It is also known to be compatible with other IdPs such as Auth0, OneLogin, and Google G-Suite.

Visual Client

TFS

Microsoft Visual Studio is the visual client used by TFVC developers. It is arguably the most popular and powerful IDE in the software development world.

HELIX CORE

Helix Core provides seamless integration with Microsoft Visual Studio. The P4VS plugin is available in the Visual Studio Marketplace, and provides access to all the major functions of Helix Core from within the IDE. This includes branching and merging, exclusive checkouts with file locking, code reviews, and more. In addition, P4V, Helix Core's multiplatform visual client, provides an easy-to-use interface across Windows, Linux, and Mac.

Conclusion

TFS is like an old friend. But it's one who wants to go give you handwritten directions, rather than plug an address into Google maps. It's probably holding you back at this point, or at least delaying your arrival. Helix Core is a robust solution to TFS/TFVC that supports your DevOps and automation initiatives. It can easily handle your version management operations more reliably. And Helix Core delivers satisfaction to your organization with faster performance, global scalability, better security, and support for DevOps at scale.

Want to learn more about how Helix Core stacks up against TFS? [We're here to help!](#)

About Perforce

Perforce is a leading provider of enterprise scale software solutions to technology developers and development operations ("DevOps") teams requiring productivity, visibility and scale during all phases of the development lifecycle. Enterprises across the globe rely on its agile planning and ALM tools, automated test management, static code analysis, and version control solutions as the foundation for successful DevOps at scale. Perforce is trusted by the world's most innovative brands, including NVIDIA, Pixar, Scania, Ubisoft, and VMware. For more information, please visit www.perforce.com