Congratulations on taking the first step to moving from Subversion to Perforce Software Version Management. Subversion-to-Helix Enterprise migrations vary in scale and complexity. Many migrations are straightforward and can be self-service and completed in hours. For larger and more complex Subversion installations, migrations will take longer and a more formal approach is recommended.

With proper planning and testing, you will easily migrate and start enjoying the benefits, speed, and scale that Helix Enterprise provides. This playbook describes some of the considerations necessary to make your migration timely and successful from the beginning.

“When we had a big project, for somebody to do a get of about 120 GB and 5-6 million files, it would take about 2 hours. When we switched to Helix Enterprise, that get took less than 5 minutes.”

– Michael Austin, CTO and co-founder, Hidden Path Entertainment
MIGRATION MILESTONES

There are six recommended milestones for your Subversion-to-Helix Enterprise migration. These milestones are crucial for larger migration projects, and if you consider each in your planning, even for smaller projects, you’ll cover all the bases.

1. Select and train transition team

Your transition team is fundamental to getting your migration right. Participants will not only define the migration and make sure everything is included, tested, and completed, but they will also play a critical role in communicating and supporting the move to Helix Enterprise. Transition team members should include software developers, your Subversion admin (or developers who act as admins), and, if possible, a person with Subversion-to-Helix Enterprise migration experience, such as a consultant or someone who’s made the move with another company.

2. Develop migration plan

There are several components to your migration plan, and we’ll discuss these in the next sections. Each component requires your transition team to evaluate the needs of your organization, both to support current and older products and releases, and to prepare for, and improve, the status quo. Not only does Helix Enterprise bring new capabilities to improve your software change management system, but the migration also is the perfect opportunity to examine what works today and what doesn’t, why and how you do things, and what changes you want and need as you move forward.

Migration plan components include:

- Selecting your migration approach
- Defining your cut-over strategy
- Identifying your transition strategy
- Creating your transition schedule
- Planning your branching strategy
3 Complete dry runs

Each milestone is important, but completing dry runs is absolutely essential for success. Do a dry run of everything. You should perform full dry runs against various stresses and loads, test your backup systems, test integration with your developers’ tools, have developers do runs from their desktops, and so on. Look out for and address any data and/or process snags and make appropriate corrections. Also consider a production simulation via pilot or proof of concept.

4 Get Helix Enterprise environment ready

At this point you should be prepared to make sure everything migrated completely and the source code is verified in your Helix Enterprise system. This is also the time to do any necessary retooling on your build systems, such as custom automation that needs to be changed to talk to Helix Enterprise instead of Subversion. Then verify the build results—make sure you can do a build, complete a build, and run it through your test suites.

5 Train users

Getting everybody to use Helix Enterprise is the next step. This should be done as close to your cut-over date as possible; build in a little leeway in case any new or missed opportunities result from the training.

6 Move to Helix Enterprise

When each milestone has been completed, you’re ready to cut over to Helix Enterprise.
CONSIDERATION AND OPTIONS

There are several options to consider as you work your way through the milestones. Let’s look at a few of those.

Branching strategy

You need to think about your flow of change and what you want it to look like. What does your process look like and where do you want changes flowed? Helix Enterprise has vastly more powerful merging capabilities than Subversion, and you want to get the best and the most out of Helix Enterprise by taking advantage of its power. This is an opportunity to improve and automate processes and to do more sophisticated things with less human effort.


Migration approaches

Determining what and how much history to import requires careful consideration, and you may have different approaches for different products, depending on where they are in their lifecycle and importance to your organization. Also consider that most customers never use the history they’ve imported, so take a good look at what you’ll really need and why before you take the time and effort required for imports.

Tips only

This approach involves getting the latest file versions, or “tips,” from Subversion and adding them to Helix Enterprise. No history is preserved. This approach is sometimes appropriate for documentation but is rarely ideal for source code, except for prototype and demo code. This is, in essence, a “start over” approach.

Detailed History Import (DHI)

This approach is the most complete case of conversion, and it captures as much detailed branch/merge information as possible so that comprehensive historical research can occur in the new system without requiring the old one. A DHI captures elements such as properties, user descriptions, and check-in comments. With this approach, you can see your old history in a new light with Helix Enterprise’s powerful visualization tools. Depending on your plans, you may or may not need all these details. Many customers will do a BBI first, get started with Helix Enterprise, and then, if needed, proceed with a DHI.

Baseline and Branch Import (BBI)

The BBI approach offers a lightweight migration alternative that is more sophisticated than the tips approach and avoids the technical complexity and schedule and budget risks of detailed history imports. With the BBI approach, the “interesting history” to be imported is specified using a branch diagram that shows the baselines (snapshots of a directory structure at a specified time) and major branching operations. It imports branching operations at a high level, capturing the sum of merge operations. A BBI lets you get started quickly and is very reliable.
Transition and cut-over strategies

There are two approaches you can take to cutting over to Helix Enterprise: a “grand opening” strategy or a “phased by team” strategy. The grand opening strategy cuts over everything at once, with all team members on Helix Enterprise—for example, using Subversion on a Friday and starting with Helix Enterprise on a Monday. With a solid migration plan and execution against the milestones, this approach should not be a major effort and is recommended if possible.

If the grand opening strategy doesn’t work for you, perhaps due to large projects or individual team readiness, then the phased-by-team strategy is necessary. This can be a more time-consuming and complicated cut-over. As various teams are migrated, you need to consider any software dependencies across teams, any dependencies between collaborating teams, and, if you have a very large repository, limitations of the migration utilities themselves.

MIGRATION TIPS

As you move from Subversion to Helix Enterprise, consider the following best practices and tips to help your migration go smoothly:

- There can be case issues, so convert on the same platform as the server. For example, migrate from Windows to Windows or Linux to Linux, and then change platforms if desired.
- Make sure you have enough hardware. Because migration can be hardware intense and require much more resources than a Subversion or Helix Enterprise server in normal operation, you may want a separate migration server. This applies mostly for DHI migrations.
- Consider typemap, file types, and Unicode. Typemap is a Helix Enterprise feature, so you’ll want to plan according to the data types you have.
- Subversion has empty directories and Helix Enterprise does not; use .d or README files so these directories are no longer empty. Or move the responsibility for those empty directories to your makefiles or to your build/deploy scripts.
- Use the flexibility of workspace mapping to handle subversion externals.
- Keep Subversion around as a point of reference or history.
THE GREAT MIGRATION

Migrating to Helix Enterprise is the right decision. Helix Enterprise provides the benefits you need from a state-of-the-art software change management system, including:

- Fast, powerful, and flexible branching and powerful merge
- Distributed and centralized workflows with full Git support
- Ability to store files of any size or type
- Permanent history combined with universal security policies
- Built-in replication and caching technology for global development sites, at no additional cost
- Scalability to petabytes and millions of concurrent transactions
- “More agile development and faster time-to-market,” according to Forrester Research

And Helix Enterprise outperforms Subversion in important areas:

- Vastly more powerful branching and merging
- Much more scalable
- Better visualizations for improved and faster merges and detection of a need to merge
- Much faster, including 5 to 10 times faster when syncing large numbers of files

Don’t put off your migration from Subversion any longer. You need to position your software development efforts for success. Helix Enterprise is the fast, modern, scalable platform that allows you to version everything. Get started now on your migration to Helix Enterprise.

1. The Total Economic Impact™ Of Perforce, Forrester Research, Inc., May 2014, p.6
Get Expert Assistance
Not sure where to start with your migration? Contact Consulting Services.
info.perforce.com/ondemand-webinar-legacy-scm-migration.html

See How It’s Done
Thousands have migrated from a legacy SCM to Perforce.
Learn how you can too. Get a Migration Kit.
info.perforce.com/migration-kit-overview.html

Customized ROI Analysis
Gauge the increased productivity and bottom line benefits of moving off your legacy SCM system. Get Custom Report.
info.perforce.com/customized-ROI-analysis.html