Helix Coreを用いた分散バージョン管理環境（DVCS）

説明

Helix Coreを用いた分散バージョン管理を行うための環境構築から利用手順を紹介します。
以下は、Helix Coreサーバ 2017.2、Helix Visual Clietn P4V 2017.3を使用した例で説明しています。

概要

DVCSとは

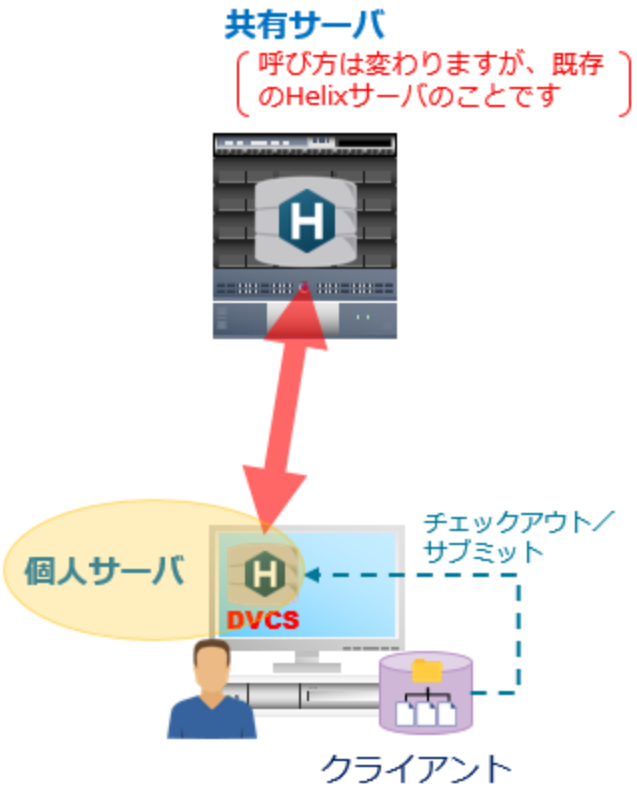
DVCSは、Gitに近いの開発環境を実現する機能です。
※DVCS： Distributed Version Control System

既存のHelixサーバに保持しているファイルの一部、あるいはすべてをサブセットとして切り出し、個人用のサーバでバージョン管理します。

- ファイルの切り出し元にしたHelixサーバを、共有サーバと呼びます。
- 新たに構築した個人用のHelixサーバを、個人サーバと呼びます。

個人サーバに接続したクライアントで、チェックアウトやサブミットを実行できます。

共有サーバ上で更新された情報を個人サーバに取り込んだり、個人サーバで行った変更を共有サーバに登録したりできます。



DVCSがもたらすメリット

共有サーバから必要な情報(ファイル)を切り出し、個人サーバ上でバージョン管理します。
これにより、以下のメリットがもたらされます。

- 各ユーザは、試行錯誤の過程で作成されるファイルをローカルマシン内でバージョン管理しながら開発できます。
- 共有したい変更結果や成果物だけを共有サーバに戻すことにより、共有サーバは整理された状態を維持できます。
- 共有サーバが地理的に離れた拠点に存在する場合、共有サーバに接続することなく開発作業を進められます。
- すべてのユーザが共有サーバに常時接続する必要はないため、共有サーバの負荷を低減できます。
- 必要なファイルだけを保持することにより、個人サーバのディスク容量を最小限に抑えられます。

典型的なユースケース

DVCS環境は、次のようなケースにおいて有効です。

- **1つの修正タスクに対して、個人的には細かい変更ごとに差分を登録しておきたいが、Helix Coreサーバ(共有サーバ)上には細かい中間作業の記録を残したくない場合**
- Helix Coreサーバ(共有サーバ)は地理的に離れた拠点に存在し、クライアントの拠点からアクセスするには低速のWANを経由する必要があり、パフォーマンス上の問題を抱えている場合
- 数百名のユーザがHelixサーバ(共有サーバ)に常時アクセスすることにより、Helixサーバの負荷が定常的に高くなり、パフォーマンスの低下が見られる場合
- Gitを利用しているが、Gitリポジトリでは全ファイルを保持したクローンしか作成できず、必要以上にローカルリポジトリのディスク領域を使用することに困っている場合

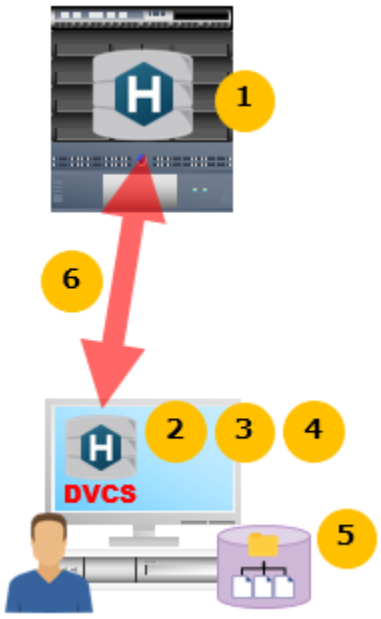
ファイルのサブセットを切り出し

個人サーバでバージョン管理したいファイルセットを、既存のHelix Coreサーバから切り出します。
複数箇所を選択することも可能です。
切り出したファイルセットは、個人サーバ上のストリームとして構築されます。

ここでは、ユースケース「1つの修正タスクに対して、個人的には細かい変更ごとに差分を登録しておきたいが、Helixサーバ(共有サーバ)上には細かい中間作業の記録を残したくない場合」に基づいて個人サーバを構築し、利用する際の手順をご説明します。

個人サーバの構築／利用手順

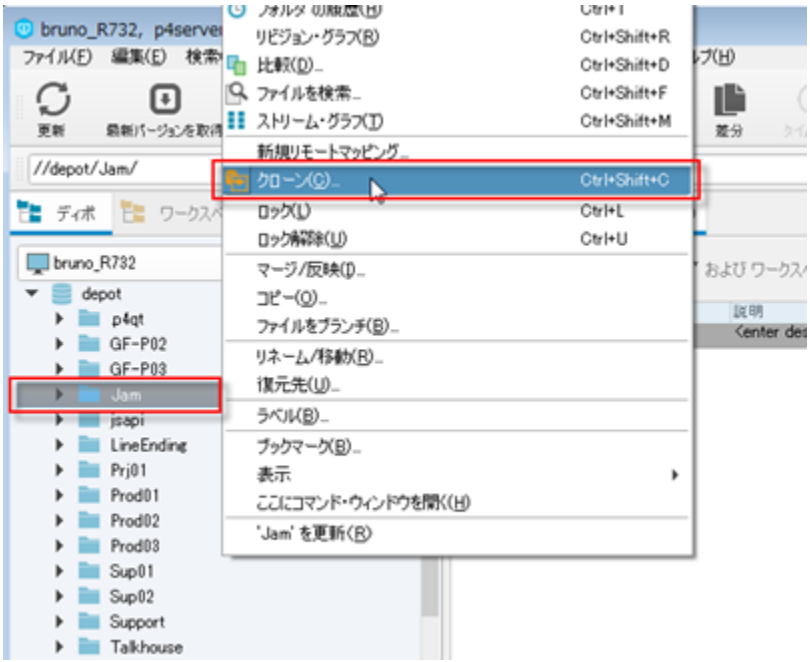
個人サーバを構築し、構築後に利用する手順となります。
作業の流れは以下の通りです。



1. 切り出すファイルセットを選択します。

共有サーバに接続して操作します。
共有サーバから切り出して、個人サーバ上に保持したいディレクトリを選択して右クリックし、[クローン]を選択します。

▼共有サーバ上の" //depot/Jam/..."を切り出す図



2. 個人サーバの構成を決定します。

[クローン]ダイアログで、リモートマッピングの[新規]ボタンをクリックします。
個人サーバの構成を決定するための、[リモートマッピング]フォームが表示されます。

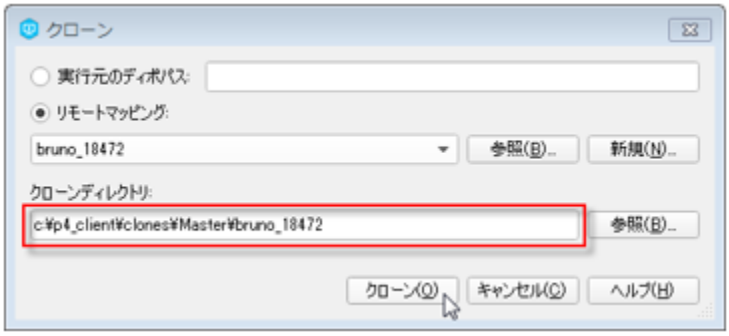
表示されたリモートマッピングに、[**ディポマッピング**]フィールドの設定をします。
個人サーバのパス、リモートサーバのパス(共有サーバ)を記述します。

3. 個人サーバの配置場所を指定します。

[クローン]ダイアログに戻り、[クローンディレクトリ]フィールドに配置場所を入力します。
指定したクローンディレクトリに、個人サーバでは次の情報を保持します。

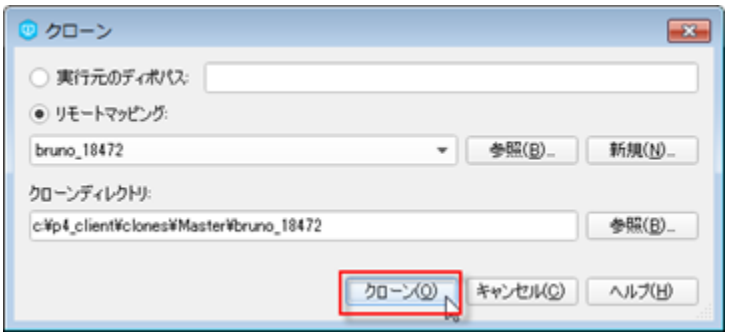
- 個人サーバのディポとデータベース
- ワークスペースのルート

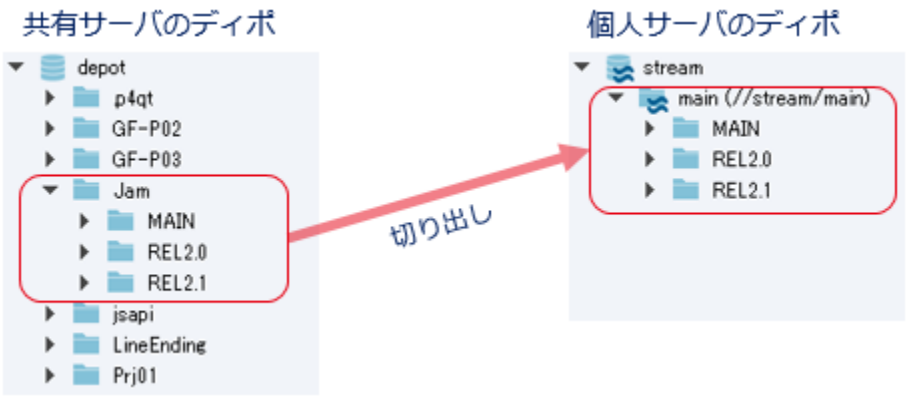
パフォーマンスの観点から、ローカルのディスク領域を指定することをお勧めします。



4. 個人サーバを構築します。

[クローン]ボタンをクリックして、個人サーバを構築します。

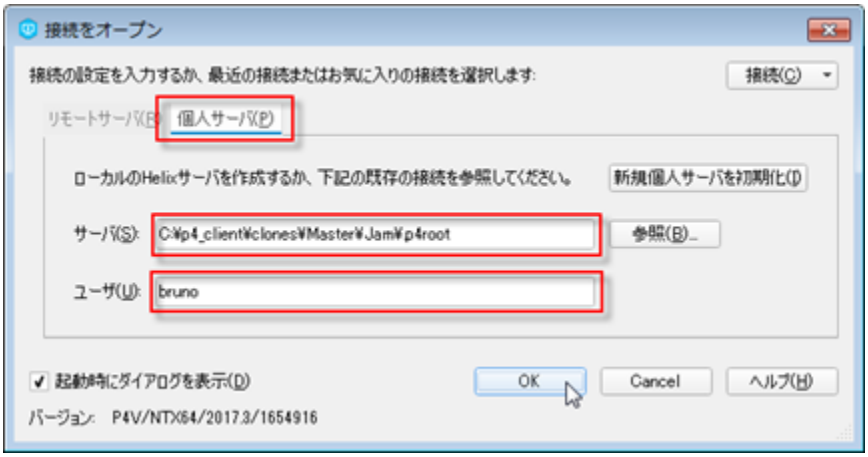




構築した個人サーバの" //stream/dev/..." には、共有サーバの" //depot/Jam/..." が切り出され、保持されます

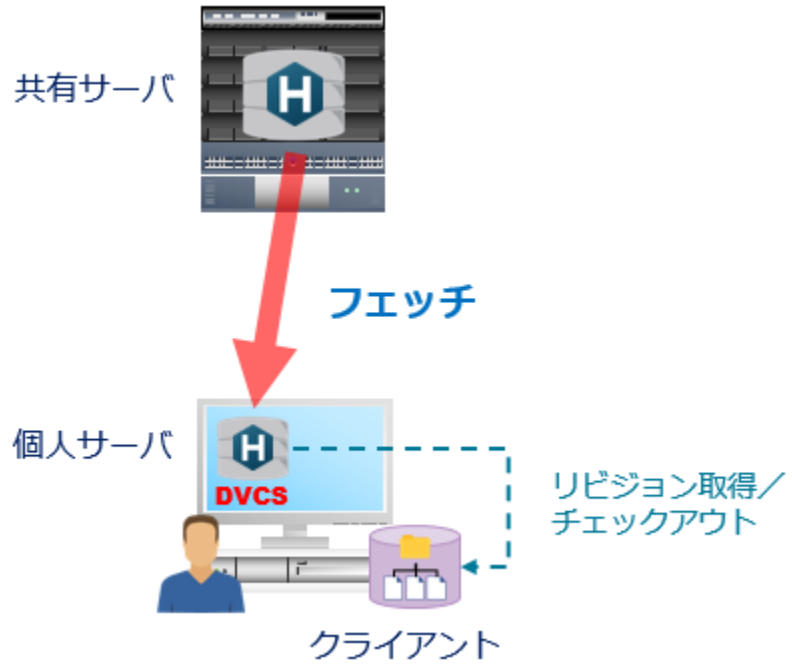
5. 個人サーバへの接続を確認します。

1. P4Vを起動します。
2. P4Vの[接続をオープン]ダイアログにおいて、[個人サーバ]タブを開きます。
3. [サーバ]フィールドにクローンディレクトリを指定します。
4. [ユーザ]フィールドに自分のユーザ名を指定します。
5. [OK]ボタンをクリックして、P4Vのメイン画面が立ち上がることを確認します。

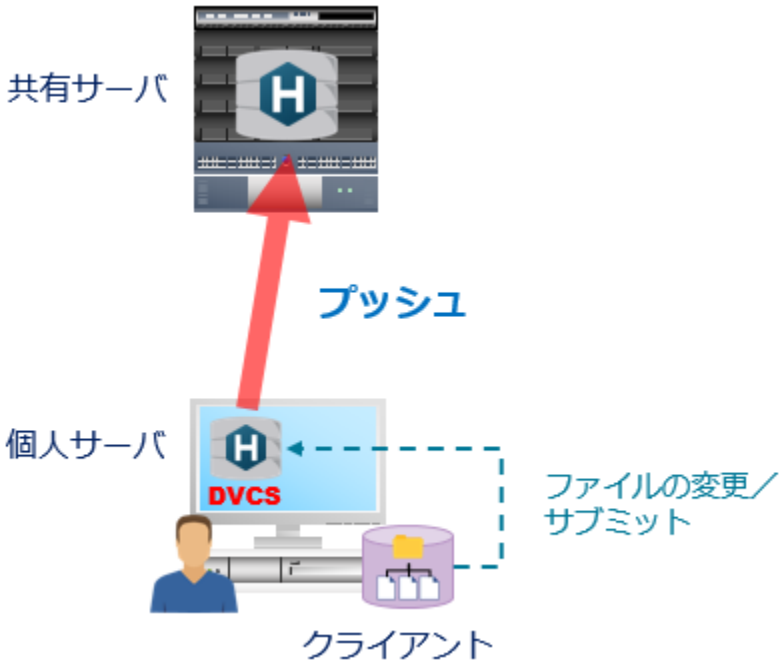


6. 個人サーバ構築後の利用手順

他のユーザが変更した共有サーバ上のファイルを、個人サーバに取り込みます。(この操作をフェッチと呼びます)
取り込んだファイルを、個人サーバ上で変更(チェックアウト／サブミット)します。



個人サーバ上で変更したファイルを、共有サーバに登録します。(この操作をプッシュと呼びます)
共有サーバに登録した情報は、他のユーザから参照できるようになります。



個人サーバ構築／応用例

特定のディレクトリの除外

個人サーバ上に保持するファイルセットから、特定のディレクトリやファイルを除外することができます

ディポマッピングにおける除外マッピング

- 個人サーバのパスの行頭にマイナス記号 "−" を付けます
- ワークスペースビューのように、指定したディレクトリやファイルを除外できます

記述例：

//stream/dev/...	//depot/Jam/...
-//stream/dev/test/...	//depot/Jam/test/...
-//stream/dev/....iso	//depot/Jam/....iso

上記の意味は、" //depot/Jam/... " 配下のファイルを保持しますが、以下については、保持の対象から除外する記述例です。

//depot/Jam/test/...
すべての .iso ファイル

保持するリビジョン数を制限

個人サーバ上に保持するファイルにおいて、保持するリビジョン数を制限することができます。

アーカイブ制限でリビジョン数を制限

- パスとリビジョン数を指定します。
- ファイルタイプ修飾子 +S のように、保持するリビジョン数を制限できます。

記述例：

//stream/dev/bin/...	10
//stream/dev/....zip	1

上記の意味は、以下です。

" //depot/Jam/bin/... " については、最新の10リビジョンのみを保持します。
すべての .zip ファイルについては、最新リビジョンのみを保持します。