# PERFORCE

# P4VS User Guide

2017.1

*June 2017*

# P4VS User Guide
## 2017.1

June 2017

Copyright © 2012-2017 Perforce Software.

Any additional software included within Perforce software is listed in .

# Table of Contents

# About This Manual

This guide describes the installation, configuration, and operation of P4VS, the Perforce plugin for Visual Studio.

# Helix documentation

The following table lists and describes key documents for Helix users, developers, and administrators. For complete information see the following:

[http://www.perforce.com/documentation](http://www.perforce.com/documentation)

| For specific information about… | See this documentation… |
| --- | --- |
| Overview of Helix version control concepts and workflows; Helix architecture, and related products. | *Helix Versioning: An Overview* |
| Administering the Helix versioning engine to support git repos and LFS files. | *Helix4Git* |
| Using the command-line interface to perform software version management and codeline management; working with Helix streams; jobs, reporting, scripting, and more. | *Helix Versioning Engine User Guide* |
| Basic workflows using P4V, the cross-platform Helix desktop client. | *P4V User Guide* |
| Working with personal and shared servers and understanding the distributed versioning features of the Helix Versioning engine. | *Using Helix for Distributed Versioning* |
| **p4** command line (reference). | *P4 Command Reference*, **p4 help** |
| Installing and administering the Helix versioning engine, including user management, security settings. | *Helix Versioning Engine Administrator Guide: Fundamentals* |
| Installing and configuring Helix servers (proxies, replicas, and edge servers) in a distributed environment. | *Helix Versioning Engine Administrator Guide: Multi-site Deployment* |
| Helix plug-ins and integrations. | IDEs: *Using IDE Plug-ins* <br> Defect trackers: *Defect Tracking Gateway Guide* <br> Others: online help from the Helix menu or web site |
| Developing custom Helix applications using the Helix C/C++ API. | *C/C++ API User Guide* |

| For specific information about… | See this documentation… |
|---|---|
| Working with Helix in Ruby, Perl, Python, and PHP. | *APIs for Scripting* |

# Syntax conventions

Helix documentation uses the following syntax conventions to describe command line syntax.

| Notation | Meaning |
|---|---|
| `literal` | Monospace font indicates a word or other notation that must be used in the command exactly as shown. |
| *italics* | Italics indicate a parameter for which you must supply specific information. For example, for a *serverid* parameter, you must supply the id of the server. |
| `[-f]` | Square brackets indicate that the enclosed elements are optional. Omit the brackets when you compose the command.<br><br>Elements that are not bracketed are required. |
| … | Ellipses (…) indicate that the preceding element can be repeated as often as needed. |
| *element1* \| *element2* | A vertical bar ( \| ) indicates that either *element1* or *element2* is required. |

# Please give us feedback

We are interested in receiving opinions on this manual from our users. In particular, we'd like to hear from users who have never used Perforce before. Does this guide teach the topic well? Please let us know what you think; we can be reached at manual@perforce.com.

If you need assistance, or wish to provide feedback about any of our products, contact support@perforce.com.

# Getting Started with P4VS

This chapter provides an overview of P4VS, the Perforce plugin for Visual Studio, as well as instructions for installing and setting it up.

## About P4VS, the Perforce plugin for Visual Studio

P4VS, the Perforce plugin for Visual Studio, enables you to use Perforce as your source control from within Visual Studio.

### Basic Perforce Terminology

*   **Depot**: The shared repository where file revisions are stored and managed by the Perforce versioning service.

*   **Workspace**: The area on your computer where you work with your copies of files that are under Perforce control.

*   **Perforce service**: The application that manages file revisions.

*   **Changelist**: A group of files, with a description explaining how you have changed them (for example, `Fix bug #123`). Changelists are assigned numbers by the Perforce versioning service so you can track them. Changelists enable you to group related files and submit them together.

The following diagram shows the relationship between workspace and depot:
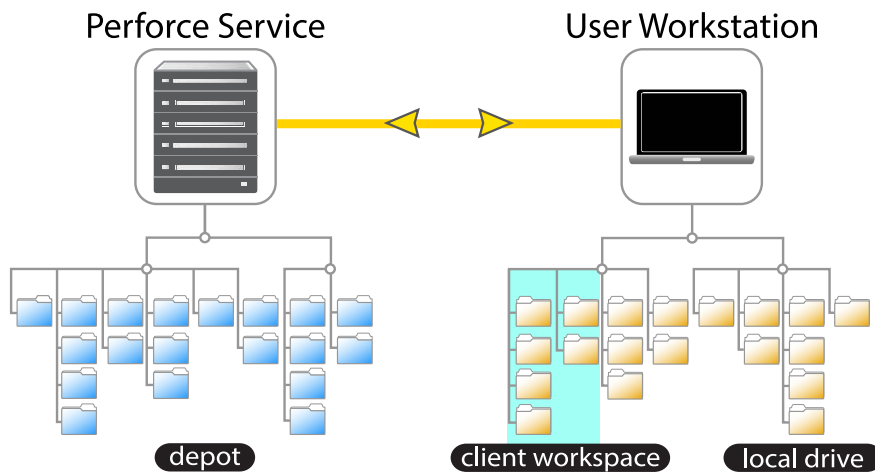


### Basic Tasks

*   **Get revision**: Retrieve a copy of a file version from the depot. Perforce also uses the term *sync* to mean *get revision*.

*   **Check out**: Enables you to change the file.

*   **Mark** for add or delete: indicates that the file is added to or deleted from the depot when the changelist is submitted.

- **Revert** a file: Discard any changes you have made to an open file. If you open a file for edit and make changes, then change your mind and revert the file, Perforce reloads the last version you got from the depot and discards your changes.

- **Submit** a changelist: Update the depot to reflect any changes you have made to files in the changelist. Submitting is an all-or-nothing operation: if there is a problem submitting one file in a changelist, none of the other files in the changelist are updated.

## Using Solution Explorer with P4VS

Solution Explorer provides access to most P4VS functionality and status information.

- When you context-click a file in Solution Explorer, all P4VS actions enabled for that file are available for selection in the context menu.

- Badges on file icons indicate Perforce status:

| | |
|---|---|
| **+** | Marked for add |
| **✕** | Marked for delete |
| **✓** | Checked out by you |
| **✓** | Checked out by another user |
| **🔒** | Locked |
| **▲** | Version in workspace is not latest version |
| **●** | Workspace version is up to date |
| **?** | Needs resolve (conflicting changes have been made) |
| **?** | File not in Perforce depot |
| **↘** | Marked for integrate |
| **●** | Ignored |
| **✳** | File is lazily loaded, whenever operated on by P4VS |

Context-click a file and select **Refresh** in the context menu to refresh Perforce status information for the file and any of its children.

| Note | If you select **Automatically update files status when selection changes** in the P4VS Preferences, the file's Perforce status updates automatically when you click or hover over the file icon, without having to click **Refresh**. |
|---|---|

## Using P4VS toolbars in Visual Studio

P4VS provides two toolbars that you can use with Visual Studio:

- The **P4 Connection** toolbar displays your current Perforce service connection (hostname:port, workspace, and user), as well as the pending changelist you are working in.

- The **P4VS** toolbar provides the same menu of P4VS options as the Solution Explorer context menu.

To enable these toolbars, select **View > Toolbars**.

### For more information

Watch our P4VS tutorial video: [https://www.perforce.com/video-tutorials/plugin-visual-studio](https://www.perforce.com/video-tutorials/plugin-visual-studio)

For more information about how to use Perforce, see *Introducing Helix*.

For the P4VS release notes, see [https://www.perforce.com/perforce/doc.current/user/p4vsnotes.txt](https://www.perforce.com/perforce/doc.current/user/p4vsnotes.txt).

# Installing P4VS and enabling the extension in Visual Studio

To use P4VS with Visual Studio, you must install the plugin and then enable it within Visual Studio.

## Installing P4VS in Visual Studio 2013 And Later Releases

1. Download the P4VS Visual Studio Extension Installer file, `p4vs11.vsix`.

2. Open the installer file.

3. Select the version of Visual Studio that you want the extension to install to.

4. Click **Install**.

5. The installer screen displays an **Installation Complete** message.

## Enabling P4VS in Visual Studio (all supported versions)

| Note | This step is unnecessary when you install a P4VS upgrade. |
|------|----------------------------------------------------------|

1. **In Visual Studio, go to** Tools > Options**.**

2. **Select** Source Control > Plug-in Selection**.**

   Visual Studio 2008 only: if **Source Control** does not appear in the left pane, select **Show all settings**.

3. In the **Current source control plug-in** drop-down list, select **P4VS - Perforce Plugin for Visual Studio**.

# Migrating from P4SCC to P4VS

To migrate your solutions and projects from P4SCC, the Perforce SCC Plug-in, to P4VS, you must do the following:

1. Strip the bindings from the solution (`.sln`) file and all of the project files (`csproj`, `vcxproj`, etc.).

2. Remove `MSSCCPROJ.SCC` and other source-control-specific files.

3. Move all developers to P4VS and have them uninstall the old P4SCC plug-in.

For detailed instructions, see https://www.perforce.com/perforce/doc.current/user/sccpluginnotes.txt.

# Setting Perforce preferences

To set Perforce preferences in Visual Studio, go to **Tools > Options**. You can specify preferences on the following nodes in the **Options** dialog under **Source Control**:

- "Perforce - Connections" on page 4

- "Perforce - Data Retrieval" on page 5

- "Perforce - Diff/Merge" on page 6

- "Perforce - General" on page 7

- "Perforce - Ignoring Files" on page 8

- "Perforce - Logging" on page 9

You can specify keyboard shortcuts for P4VS commands under **Tools > Options > Environment > Keyboard**.

## Perforce - Connections

Set the following preferences to determine how you connect to the shared Perforce service in Visual Studio:

### When opening a project under source control

- `Show the Perforce Connection dialog`: Prompt for connection settings whenever you open a project that is under Perforce source control.

- `Connect to the server using my most recent settings`: Without prompting for connection settings, reconnect to the Perforce service you were connected to during your last session.

- `Connect to the server using solution-specific settings`: Without prompting for connection settings, connect to the Perforce service you last used for the solution or project that you are opening.

- `Connect to the server using my Perforce environment settings`: Connect using Windows environment variables for Perforce connections, which you set using the Perforce Command-Line Client or P4V, the Perforce Visual Client. For more information, see the *P4 User Guide* or the P4V help.

### Opening and closing connections

- **Use IP-specific tickets when logging in**: Specifies whether your login ticket is restricted to the IP address from which you are connecting.

- **Automatically log off when closing a connection**: Specifies whether your ticket is invalidated when you log out.

## Perforce - Data Retrieval

Set the following preferences to determine how P4VS retrieves data from the shared Perforce service:

### Data retrieval:

- **Check server for updates every**: Specifies how often P4VS checks the Perforce service for updated file information. Frequent checks enable P4VS to display current file status but increase the workload on the Perforce service.

- **Maximum number of files displayed per changelist**: Specifies the maximum number of files displayed in a changelist, to help minimize the time required to handle and submit very large changelists. This setting affects only the display of changelists, and does not limit the number of files that a changelist can contain.

- **Maximum size of files to preview**: Limits the size of image files displayed in the Preview tab, to limit the amount of image data sent from the Perforce service to P4VS.

- **Number of changelists, jobs, or labels to fetch at a time**: Specifies the number of specifications read in each batch fetched, to minimize server load and maximize P4VS performance. To retrieve all entries, specify **0**.

- **Automatically update file status when selection changes**: Select to enable the Perforce status badges in Solution Explorer to update automatically when you click or hover over the file icon, without having to click **Refresh**. Deselect to improve performance.

### Optimize file status retrieval:

| Note | The **Treat Solution/Projects as directories when selected**, **Preload file state**, and **Lazy load file state** options are used to tune the performance of P4VS for your environment. If none of these options are selected, P4VS will load the metadata for each file individually. Unless you have a small project you should look to use one of these options. |
| --- | --- |

- **Optimize file state retrieval**: Select to apply optimizations on retrieving file state. Subordinate optimization options include:

  - **Treat Solution/Projects as directories when selected**: Select to treat solutions and projects as directories when P4VS runs Perforce commands.

    Use this option to improve performance when working with solutions that contain a large number of projects or files. Do not use this option if the directories in the solution contain a large number of other Perforce controlled files that are not included in the solution.

This option does not require that all the files and directories referenced by the solution are under the solution directory.

- **Preload file state**: Select to preload the metadata for all of the files in the perforce depot in or under the directory containing the solution file.

  Use this option to improve performance when loading small to medium sized solutions where all the files that make up the solution are under the solution root. Only use this when there are few if any files under the solution root that are Perforce controlled that are not part of the solution.

  This option tends to work best with solutions and projects created and managed by Visual Studio. Do not use this option if the files for the solution are intermixed in directories with large numbers of other Perforce controlled files. This option will provide little improvement in performance if most of the files composing the solution are outside of the solution root.

- **Lazy load file state**: Select to only load the Perforce metadata files in the solution as they are operated on using P4VS.

  With this option, Perforce metadata for a file is only retrieved from the server when you select a P4VS operation on a file. At that point, the Perforce metadata will be obtained from the server, the operation performed, and the Perforce metadata updated to reflect the results of the operation. When this option is selected, the file will by badged in the solution explorer to indicate that its status is unknown (in Visual Studio 2010, this will be a blank badge). After a Perforce operation is performed on a file it will be badged to indicate its current state.

  Use this option to improve performance loading large to very large sized solutions where you are interested in only working on a few select files. This is best for solutions and projects which include large amounts of code from libraries or frameworks, large numbers of asset files such as graphical elements for a game, or large numbers of files that are generated by another development system or plugin. This option is also useful in situations where the connection to the Perforce server is over a slow network or VPN.

  - **Full menu**: Select to allow the full P4VS menu to be displayed on files that have not had their metadata loaded.

    This option is displayed if the **Lazy load file state** option is enabled. This allows you to perform an operation on a file before P4VS loads its state. If this option is not selected, you are given the option to refresh the file which will load the metadata of the file from the server and then enable the appropriate P4VS operations on that file. Please note that if you choose this option and perform P4VS operations on a file that are not valid based on its current state, you are very likely to get error messages back from those operations.

- **Do not optimize**: Select this to disable all optimizations related to file state retrieval.

## Perforce - Diff/Merge

To set the default diff application, select one of the following:

- **P4Merge**: Perforce's companion diff tool.

- **Other application**: Browse to your preferred diff tool.

To specify arguments for third-party diff applications, enter **%1** for the name of the first file and **%2** for the name of the second file in the **Arguments** field. Perforce replaces these placeholders with the actual filenames when calling the diff application.

To set the default merge application, select one of the following:

- **P4Merge**: Perforce's companion merge tool.

- **Other application**: Browse to your preferred merge tool.

To specify arguments for third-party merge applications, enter the following replaceable parameters in the Arguments field:

- Base file: **%b**

- Their/Source file: **%1**

- Your/Target file: **%2**

- Result file: **%r**

Perforce replaces these placeholders with the actual filenames when calling the merge application.

# Perforce - General

Set the following display and file behavior preferences:

## Display:

- **Use OS format for dates**: Use the date format that the operating system uses.

- **Format dates using Perforce standard (yyyy/mm/dd hh:mm:ss)**: Use the Perforce format.

## Files and folders:

- **Warn before reverting files**: Specifies whether P4VS displays a prompt before reverting files.

- **Lock files on checkout**: Select to lock files every time you check them out. Locks prevent other users from checking in changes while you work on a file.

- **Prompt for changelist when checking out or adding files**: Specifies whether P4VS prompts you to choose a changelist when you add or check out files.

- **Automatically add new files to Perforce**: Select if you want P4VS to mark new files for add in a pending changelist.

  This option works together with the **Prompt for changelist when checking out or adding files** option to determine prompting behavior when you add new files to a project or solution that is under Perforce control:

  - If both this option and the **Prompt for changelist** option are enabled, P4VS prompts you to mark new files for add.

- If this option is enabled and the `Prompt for changelist` option is disabled, new files are automatically added to the default changelist without any prompt.

- If this option is disabled and the `Prompt for changelist` option is enabled, no prompts will appear; you must manually mark new files for add.

- `Update related projects when reverting moved files`: Select to revert file renames or moves in Visual Studio when you revert a Perforce rename/move operation.

  If you do not select this option, Visual Studio continues to show the new file name or location despite the fact that Perforce has reverted the file to its original name or location.

  | **Note** | If you revert a folder rename/move in P4VS, you must manually revert the name or location in Visual Studio, regardless of your preference selection. |
  |---|---|

- `Use Visual Studio to view file versions`: Select to view previous revisions of a file (from the **File History** dialog, for example) in a Visual Studio editor window.

  You can use this option to view file revisions or shelved files the same way that you would view an editable file in the Visual Studio IDE.

### Project and solution files:

- `Tag project and solution files as controlled by P4VS`: Select to enable P4VS to write tags to the solution and project files that are under Perforce control. The default is not to tag project and solution files; P4VS does not need to tag these files to know that they are under Perforce control.

- `Set the location of new projects to the current workspace root directory`: Select to put new projects in the current workspace root directory. The **Location** field in the **New Project** dialog will be populated by default with the current workspace directory.

## Perforce - Ignoring Files

To avoid adding and checking in files that do not belong in the repository, you can exclude individual files or file types (for example, build or release artifacts) from source control using **Ignore Lists**. For more information, see <u>"Excluding Files from Perforce Control" on page 26</u>.

You can set the following preferences:

- `Enable Ignore Lists for specifying files to ignore when marking for add`: Select to enable P4VS to use **Ignore Lists** to keep individual files or file types from being added to the Perforce repository.

  If you have already set an Ignore List file as the `P4IGNORE` environment variable on your local machine, that file name appears by default in the **Name** field. If not, enter a file name or accept the standard default, `.p4ignore.txt`. The first time you select **Edit Ignore List** or **Add to Ignore List** for a file in a folder in Solution Explorer, P4VS adds an **Ignore List** file with this file name to that folder.

  | **Note** | Your local `P4IGNORE` environment variable will be updated with the file name that you enter here. If other Perforce clients (such as `p4` or P4V) on your local |
  |---|---|

> machine use **Ignore List** files, be sure to use the same file name as you use with those clients.

- `Automatically add new Ignore Lists to solution or project`: Select to have new **Ignore Lists** appear in the Solution Explorer.

  If you do not select this option, the **Ignore List** file will be hidden in Solution Explorer.

- `Prompt when creating a new Ignore List`: Select to have P4VS prompt you when you select **Add to Ignore List** in Solution Explorer to add an **Ignore List** to a folder that does not yet have one.

  If you do not select this option, the system creates the **Ignore List** without prompting.

- `Automatically ignore new Ignore Lists (add Ignore Lists to themselves)`: Select to have P4VS automatically add the **Ignore List** file to itself to prevent the **Ignore List** from being added to the Perforce depot.

- `Automatically add new Ignore Lists to Perforce`: Select to have P4VS automatically add new **Ignore Lists** to the Perforce depot.

  This option works only when the `Automatically ignore new Ignore Lists (add Ignore Lists to themselves)` option is not selected. If that option is not selected, and you do not select the `Automatically add new Ignore Lists to Perforce` option, then you must manually mark the **Ignore List** file for add to add it to the repository.

## Perforce - Logging

Set the following logging preferences. You can view P4VS log messages in the **Output** window in Visual Studio if you select `Perforce Source Control` in the **Show output from** drop-down list in the **Output** window.

- `Show p4 reporting commands`: Specifies whether the Output window in Visual Studio displays all commands issued by P4VS, including commands issued by P4VS to obtain status information from the Perforce service.

- `Show p4 command output for file operations`: For verbose log messages, enable this option.

- `Enable logging to file`: Logs P4VS activity to the specified file.

  - `Name`: Specifies the name and location of the log file.

  - `Size`: Specifies the maximum size of the log file.

## Keyboard shortcuts

Go to **Tools > Options > Environment > Keyboard.** You can find P4VS commands by entering `P4VS` in the **Show commands containing** field.

For more information about creating keyboard shortcuts in Visual Studio, see the Microsoft Visual Studio help.

# Connecting to Perforce services

Connections enable you to access the shared Perforce service to submit and obtain access to files under Perforce control. You use the **Open Connection** dialog both to define connections and to open them in P4VS.

## Defining a new Perforce service connection

To define a new Perforce service connection in Visual Studio:

1. Open the **Open Connection** dialog. You can open this dialog the following ways:

   - Add a new project in Visual Studio in the **New Project** dialog.

     The **Open Connection** dialog appears unless you've set the connection settings in **Tools > Options > Source Control** to default to the last Perforce connection or to connection settings defined in your environment variables.

   - Open a solution or project under Perforce source control in Visual Studio.

     The **Open Connection** dialog appears unless you've set the connection settings in **Tools > Options > Source Control** to default to the last Perforce connection or to connection settings defined in your environment variables.

   - Go to **File > Open Connection to a Perforce Depot** in Visual Studio.

2. **Enter the service name and port number for this connection using `service_host:port_number`.**

   If your Perforce service is enabled for SSL (Secure Sockets Layer) encryption, use the following syntax: `ssl:service_host:port_number`

   | **Important** | If you attempt to connect to an SSL-enabled Perforce service and you see a warning about an untrusted SSL connection or altered SSL fingerprint, contact your Perforce administrator before completing the connection. |
   |---|---|

3. In the **User** field, enter your user name.

   - To browse for a particular user, click the **Browse...** button and select the user from that list.

   - To create a user, click the **New...** button and fill in the appropriate information.

4. (Optional) In the **Workspace** field, specify the name of your client workspace.

   - To browse for a particular client workspace, click the **Browse...** button and select the workspace from that list.

   - To create a client workspace, click the **New...** button.

   - In the **New Workspace** dialog, enter a workspace name and click **OK**.

   - In the **Workspace** dialog, entered the required information.

For more information on setting up client workspaces, see "Managing workspace specifications" on page 13.

5. **Click** OK.

P4VS connects to the specified Perforce service.

## Setting Perforce connection settings using environment variables

You can set Windows environment variables for Perforce connection settings, which makes the settings available to P4VS and other Perforce client applications (for example, P4EXP, the Perforce plug-in for Windows Explorer). Set Perforce connection settings as environment variables and configure your connection preferences in **Tools > Options > Source Control** to default to the environment variables.

Another approach is to create a configuration file that stores your Perforce environment variables. You can then point to the configuration file using the environment variable `P4CONFIG`. P4VS searches the current working directory and its parents for the file. If the file exists, then P4VS uses the variable settings within the file. `P4CONFIG` makes it easy to switch Perforce settings when switching between different solutions or projects. If you place a configuration file in each of your client workspaces and set `P4CONFIG` to point to that file, your Perforce settings change to the settings in the configuration files automatically as you move from directories in one workspace to another.

For more information about how to use `P4CONFIG` with P4VS, see "Setting Perforce environment variables using P4CONFIG" on page 11.

For more information about Perforce environment variables, see the *P4 Command Reference* and "Setting Perforce preferences" on page 4.

## Opening a defined Perforce service connection

To open a Perforce service connection that you have already used, select the connection from the drop-down list in the **Open Connection** dialog. You can also configure P4VS to automatically open the connection that you used most recently when you open a project. See "Setting Perforce preferences" on page 4.

# Setting Perforce environment variables using P4CONFIG

`P4CONFIG` is an environment variable that you can use to point to a file that stores other Perforce environment variables. The current working directory and its parents are searched for the file. If the file exists, then the variable settings within the file are used.

`P4CONFIG` makes it easy to switch Perforce settings when switching between different solutions or projects. If you place a configuration file in each of your client workspaces and set `P4CONFIG` to point to that file, your Perforce settings change to the settings in the configuration files automatically as you move from directories in one workspace to another.

To use `P4CONFIG` to switch settings between client workspaces on P4VS, you must create separate Visual Studio shortcuts for each workspace, setting the `Start in` property as the workspace directory (which is also the directory where the configuration file resides). If you launch Visual Studio using

a shortcut defined this way, P4VS will read the Perforce settings from the configuration file in that workspace's directory. This is required because Visual Studio otherwise uses its own directory as the current working directory.

To use `P4CONFIG` with P4VS:

1. Create a configuration file that contains the Perforce environment variable settings you want, and put it in the workspace directory for the relevant Visual Studio solution.

2. Using `p4`, the Perforce Command Line Client, unset the `P4CLIENT`, `P4PORT`, and `P4USER` environment variables and set `P4CONFIG` to the configuration file name.

3. Using P4VS, go to **Tools > Options > Source Control > Perforce - Connections** and select **Connect to the server using my Perforce environment settings**.

4. Create a Windows desktop shortcut for Visual Studio that is dedicated to the workspace with which you want to use the configuration file.

   a. Right-click on the desktop and select **New > Shortcut**.

   b. Enter the location of the Visual Studio executable and click **Next**.

   c. Enter a shortcut name and click **Finish**.

   d. In the shortcut properties, under **Start in**, enter the workspace directory where the configuration file is located, and click **OK**.

   Repeat for each workspace for which you want a different configuration file.

5. **Whenever you want to work in that workspace using the configuration file settings, use the shortcut to launch Visual Studio.**

For more information about `P4CONFIG` and Perforce environment variables, see the *P4 Command Reference* and "Setting Perforce preferences" on page 4.

# Customizing context menus

You can use Visual Studio customization functionality to add or remove P4VS commands in context menus.

To add or remove a P4VS command using Visual Studio 2010:

1. Go to **Tools > Customize** and open the **Commands** tab.

2. In the **Menu bar** drop-down, select the menu you want to customize.

   The P4VS menus begin with `File | Perforce`.

3. **Under** Controls**, select a command to delete or move, or select** Add Command… **to select a command to add to the menu.**

   Many of the P4VS commands are under the **File** and **View** categories.

| **Note** | There are many P4VS commands with names that are similar to native Visual Studio or other plug-in commands. If you have any questions about which commands belong to P4VS, contact your Perforce administrator. |
|----------|---|

For more information, or for Visual Studio 2008 instructions, see the Microsoft Visual Studio help.

# Managing workspace specifications

A workspace specification defines which portion of the depot can be accessed from that workspace and specifies where local copies of files in the depot are stored. This location is called the workspace. A computer can contain multiple workspaces. A workspace is required when connecting to a Perforce service if you intend to work with files.

The mapping of depot files to local files is called the workspace view. If you are working with streams, the workspace view is generated by Perforce, based on the structure of the stream. If the structure of the stream changes, the workspace view is updated automatically. (In fact, you cannot manually edit the view of a stream workspace.) If you use classic depots, you must define and maintain the workspace view manually.

## Creating workspaces

To create a new workspace in P4VS:

1. Open the **Open Connection** dialog.

   For more information, see <u>"Connecting to Perforce services" on page 10</u>.

2. Click the **New…** button next to the **Workspace** field to open the **New Workspace** dialog.

3. Enter a workspace name and click **OK**.

4. In the **Workspace** dialog, view or enter the following settings:

| Setting | Description |
|---------|-------------|
| Workspace | Workspace name. Defaults from the **New Workspace** dialog. |
| Owner | The user who created the specification. Defaults to you when you create a new workspace. |
| Host | (optional) The computer where the workspace resides. To enable the workspace to be used from any machine, leave this field blank. |
| Submit options | Configures what happens when users submit files. |
| Line endings | The line-end convention used for storing text files on the workspace computer:<br><br>• Local: Uses the workspace platform default |

| Setting | Description |
| --- | --- |
| | • Unix: `LF` |
| | • Mac: `CR` |
| | • Windows: `CRLF` |
| | • Share: Line endings are `LF`. Any `CR` prior to a line ending is removed for storage or syncing (for disks shared between UNIX and Windows) |
| `Description` | Your own explanation of the purpose of the workspace, or any related information you want to specify. |
| `Root` | Workspace root directory where you want local copies of depot files stored. |
| `Alt Roots` | For workspace specifications used from hosts on different platforms, a list of workspace roots in host-platform-specific syntax. |
| `Options` | • `allwrite`: All files in the workspace are writable (can be modified). |
| | • `clobber`: Syncing files overwrites writable files on the workspace. |
| | • `compress`: Compresses data sent between the workspace and the Perforce service. |
| | • `locked`: Only the owner of the workspace can use, change, or delete the workspace specification. |
| | • `modtime`: Modification time for files edited in the client workspace is set to the time when the file is submitted to the depot. |
| | • `rmdir`: Deletes a workspace folder if all the files contained in the folder are removed. |
| `Stream Root` | Root directory for a workspace associated with a mainline stream. For more information on streams and how the Perforce service handles stream workspaces, see ["Stream workspaces" on page 15](#). |
| **View** | The workspace view determines which portions of the depot are visible in your **Workspace Tree** and where local copies of depot files are stored in your workspace. If you use streams, the workspace view is generated and updated automatically. For more information on workspace views, see ["Defining a workspace view" on page 15](#). |

5. Click **OK** to save your entries and create the workspace specification.

## Changing your workspace

To change the workspace you are using, use the **Open Connection** dialog and specify the workspace in the **Workspace** field.

For more information, see "Connecting to Perforce services" on page 10.

## Viewing workspaces

To view all of the workspaces for the service to which you are connected, do either of the following:

- Go to **View > Workspaces** in the Visual Studio menu bar to open the **Workspaces** tool window.

  Click a workspace row to display the details of the client workspace specification.

  To change the order in which columns are displayed, drag the column headings right or left to the desired position. To sort by column, click the sort arrow on a column heading.

- Open the **Open Connection** dialog and click the **Workspace Browse…** button to open the **Workspace Browser** dialog.

  Click a workspace row to display the details of the client workspace specification.

  For more information, see "Connecting to Perforce services" on page 10.

## Stream workspaces

If you work with streams, P4VS uses workspaces differently than it does with classic depots. For more information, see the "Streams" chapter in the *Helix Versioning Engine User Guide*>.

## Defining a workspace view

The workspace view (sometimes called a *client* view) determines which portions of the depot are available for you to work with in P4VS and where local copies of depot files are stored in your workspace. If you use streams, the workspace view is generated and updated automatically. If you use classic depots, you must maintain the view manually, as described in this topic.

To define or change the workspace view for an existing workspace:

1. Select **View > Workspaces**. The **Workspaces** tab is displayed.

2. Context-click the workspace and select **Edit Workspace**. The **Workspace** form is displayed.

3. Edit the **View** field. Define the view as described under "Syntactic view specification" on page 16.

4. When you have finished editing, save your changes.

To define the workspace view for a new workspace:

1. Open the **Open Connection** dialog.

   For more information, see "Connecting to Perforce services" on page 10.

2. Click the **New…** button next to the **Workspace** field to open the **New Workspace** dialog.

3. Enter a workspace name and click **OK**.

4.  In the **Workspace** dialog, edit the **View** field. Define the view as described under <u>"Syntactic view specification" on page 16</u>.

## Syntactic view specification

Type your view specification using Perforce client view syntax. Views consist of *mappings*, one per line. The left-hand side of the mapping specifies the depot files and the right-hand side specifies the location in the workspace where the depot files reside when they are retrieved from the depot. Example:

```
//depot/...        //bruno/depot/...
//user_depot/...   //bruno/user_depot/...
//projects/...     //bruno/myprojects/...
```

For details about client view syntax, refer to the *P4 User Guide*.

# Managing Files

This chapter discusses how to manage files using P4VS.

## Putting a project or solution under Perforce source control

P4VS requires that your work be included in a project or solution file.

To put existing projects or solutions solutions under Perforce source control:

1. Make sure that your projector solution and all files included in it are in the workspace (your client directory) being used by your Perforce service connection.

2. In Solution Explorer, select the project or solution that should be placed under source control.

   Follow the procedure described in .

To put new projects or solutions under Perforce source control:

1. When you create a new project using the **New Project** page, select **Add to Perforce SCM** under the **Browse** button.

2. When you click **OK** (and after you run any wizards), the **Open Connection** dialog appears.

3. Enter your Perforce connection settings.

   Make sure that your project or solution files are in the workspace (your client directory) being used by your Perforce service connection.

4. In Solution Explorer, select the project or solution.

   Follow the procedure described in .

## Adding files to the depot

To add a file to the depot, you must perform two actions:

1. Open the file for add, which places the file in a changelist.

2. Submit the changelist, which copies the file to the depot.

To open a file for add:

1. In the Solution Explorer, browse to the file you want to add.

   If a file does not reside in the depot, its icon is marked with a blue question mark.

2. Context-click the file and select **Mark for Add**.

   A P4VS dialog opens asking you to add the files to the Perforce depot.

3. Select the pending changelist you want to use for submitting the file.

4. Click **OK**.

   The file icon in Solution Explorer displays a red plus sign indicating that it is open for add.

To submit the changelist:

1. In the Solution Explorer, context-click the file and select **Submit**.

2. In the **Submit Files** dialog, enter a description of the change and click **Submit**.

   The new file is added to the depot.

| Note | If you add a file to a solution that is already under Perforce control, you are prompted to put the new file under Perforce control. If you enabled the `Automatically add new files to Perforce` option and disabled the `Prompt for changelist when checking out or adding files` option under **Tools > Options > Source Control**, the file is marked for add and placed in a changelist without any prompting. For more information about these options, see <u>"Setting Perforce preferences" on page 4</u>. |
|------|---|

For more information, see <u>"Checking in files and working with changelists" on page 20</u>.

# Opening a project or solution in the Perforce depot

To open a project or solution that has been checked into a Perforce depot:

1. Go to **File > Perforce > Open Solution/Project in Perforce Depot**.

2. In the **Choose Solution/Project in Depot** dialog, expand the tree to find the solution or project you want to open.

   Select **Filter by client workspace** to limit the depot tree to the solution and project files that are included in the current workspace view.

   If you cannot expand and view the contents of the depot tree, you are not connected to a Perforce service. Click **Open Connection** to connect.

3. Click the file and click **OK** to open it in Visual Studio.

# Retrieving files from the depot

You can retrieve the most recent revision or any previous revision of a file from the depot to your workspace. In the Solution Explorer, open the folder containing the file you want to retrieve. The icons indicate the status of the files; see <u>Chapter 1, "Getting Started with P4VS" on page 1</u> for details.

To get the latest revision:

1. Context-click the file or folder in the Solution Explorer.

2. Select **Revisions > Get Latest Revision**.

To get a previous revision:

1. Context-click the file in the Solution Explorer and select **Revisions > Get Revision…**

2. In the **Get Revision** dialog, specify the revision you want:

   • Under **Get or replace the following files/folders**, you can select specific files or folders to retrieve.

   • To to specify a revision by changelist number, label, workspace, or date, choose the method from the **Specify revision using**: drop-down list and specify the value in the edit field.

   • Select **Force Operation** to retrieve the selected revision into your workspace even if the workspace already has the file. This option does not affect open files.

   • Select **Only get revisions for files listed in changelists** to retrieve only those files that are included in changelists.

   • If you are specifying a revision by label, you can ensure that your workspace contains only the labeled file revisions by selecting **Remove files from workspace if they are not in label.**

3. Click **Get Revision** to retrieve the files to your workspace.

# Checking out and editing files

Before you edit a file, you must check it out of the Perforce depot.

To check out and edit a file:

1. In the Solution Explorer, find the file that you want to edit.

   If necessary, retrieve the correct revision to your workspace. For more information, see <u>"Retrieving files from the depot" on page 18</u>.

2. Context-click the file and choose one of the following:

   • **Checkout** *filename* to check out only the selected file

   • **Checkout All in Project** to check out the project file and all files in the project

   • **Checkout All in Solution** to check out the solution file and all files in the solution

   When you check a file out, it is placed in a changelist.

3. Make your changes.

4. To check your revised version back into the depot so that other users can view your changes and edit it, context-click the file and choose **Submit…**

   In the **Pending Changelist** dialog, enter a description of your changes and submit the changelist that contains the file. For more information, see <u>"Checking in files" on page 20</u>.

To display a file without checking it out, double-click the file icon. It opens in read-only mode.

To lock a file to prevent others from checking it out while you are working on it, context-click the file icon and select **Manage Files>Lock**. To unlock it, context-click and select **Manage Files > Unlock**.

| Note | When you try to edit or save a file that is checked into Perforce, P4VS asks if you want to check it out (and save it, if you are attempting a save). It also gives you the following options: |
|------|---|
| | • **Don't show this dialog again (always use the default changelist)**: always check out (when opening for edit) or check out and save (when saving edits) and add to the default changelist, without prompting from P4VS. |
| | • **Do this for all files being saved** (or **edited**): if your save or edit operation involves multiple files, select this option to check out (when opening for edit) or check out and save (when saving) all files in the current operation without having the P4VS dialog prompt you for each file individually. |

# Checking in files and working with changelists

To check in a file, you must submit a *changelist*. Whenever you mark files for add or delete, check them out, integrate (merge or copy), or schedule them for resolve, the files are added to changelists. Perforce changelists are lists of actions to be performed on files. The actions in the changelist are performed when you *submit* the changelist. *Pending* changelists are changelists that have yet to be submitted. Changelists are assigned unique numbers by the Perforce service. In addition, a default changelist is maintained for each client workspace. If submission of the default changelist fails, the Perforce service assigns it a number.

## Checking in files

To check in files (submit a changelist):

1. Open the **Submit** dialog by doing one of the following:

   • Context-click the file icon of a file that is checked out, marked for add, or marked for delete, and choose **Submit...** to open the **Submit Files** dialog.

   • Go to **View > Pending changelists**, context-click a changelist, and choose **Submit...** to open the **Submit Files** dialog.

   Note that the **Submit Changelist** and **Submit Files** dialogs are functionally identical; they differ only in how you access them.

2. In the **Submit Changelist** or **Submit Files** dialog, enter a description or edit the existing description, and select the files you want to check in.

   You can also perform the following actions:

   • Remove files from the changelist.

   • Revert unchanged files in the changelist (removing the unchanged files from the changelist, canceling the check-out, and leaving them synced to the version you originally checked out) or

submit only changed files (moving the unchanged files to the default changelist after the current changelist is submitted).

- Check out submitted files after you submit them.

- Associate the changelist with a job and set the job status upon submit. For more information about jobs, see <u>"Using jobs (defect tracking)" on page 44</u>.

- Perform a diff on a file pending submission by context-clicking the file and selecting **Diff Against Have Revision**. For more information, see <u>"Comparing files using diff" on page 28</u>.

3. Optional) Click **Save** to save your changelist options without checking in files.

4. Click **Submit** to check in your files.

## Displaying changelists

To display changelists:

1. Go to **View > Pending changelists** or **View > Submitted changelists** to open the **Pending** or **Submitted** tool windows.

   To change the order in which columns are displayed, drag the column headings right or left to the desired position. To sort by column, click the sort arrow on a column heading.

2. Optional) Filter the displayed changelists:

   ```
   Enter your filter criteria in the **Folder/file, User**, and **Workspace** fields.
   ```

   To filter by file, enter the full path of the file in the workspace. The filtering process is case-sensitive.

   ```
   Click **Filter**.
   ```

3. View changelist details by doing one of the following:

   - Select a changelist to display details in the fields below the changelist viewer, including description, files, jobs, and user.

   - Click the arrow next to the changelist row to expand the changelist row and view the files included in the changelist.

## Editing changelists

You can edit and perform actions on a pending changelist using the **Pending** tool window and the **Pending Changelist** dialog.

To edit a changelist from the **Pending** tool window:

1. Go to **View > Pending changelists**.

2. Context-click the changelist or file row in the viewer.

3. Perform any of the following actions or edits on a changelist or file within a changelist:

   - Submit the changelist

   - Move files to another changelist

   - Revert files

   - Shelve, unshelve, delete, or view shelved files

   - Remove or view associated jobs

   - Diff files

   - Change filetype

   - Lock and unlock files

   - Change owner and workspace

   - Create a new pending changelist

To edit a changelist from the **Pending Changelist** dialog:

1. Go to **View > Pending changelists**.

2. Open the **Pending Changelist** dialog by context-clicking the changelist in the viewer and selecting **Edit Pending Changelist** *changelist name*.

3. Perform any of the following actions or edits:

   - Edit the changelist description

   - Restrict access

   - Select files for inclusion

   - Move files to another changelist

   - Revert files

   - Unshelve, delete, or view shelved files

   - Attach or view associated jobs

   - Perform a diff on a file pending submission by context-clicking the file and selecting **Diff Against Have Revision**. For more information, see <u>"Comparing files using diff" on page 28</u>.

4. Click **OK** to save your changes.

## Restricting access to changelists

By default, all users can view a pending or submitted changelist, regardless of whether they are permitted access to the files in the changelist by the protections table. To prevent users from seeing a

changelist, check the `Restrict Access to Changelist` option when you edit a pending or submitted changelist.

This option enables the following restrictions:

- Pending changelists: visible only to the owner, regardless of whether other users have access to checked-out files.

- Pending changelists containing shelved files: visible only to users who have access to one or more of the shelved files.

- Submitted changelists: visible only to users who have access to one or more of the files that were submitted in the changelist.

## Moving a file to another changelist

To move a file from its current changelist to another one, do one of the following:

- Context-click the file in Solution Explorer and select **Manage Files > Move to another Changelist…**.

- Context-click the file in the **Pending** tool window and select **Move to another Changelist…**.

In the dialog that opens, select the changelist you want to move the file to.

## Setting changelist display preferences

To minimize the time it takes P4VS to handle very large changelists, limit the number of files displayed in a changelist by setting the **Maximum number of files displayed per changelist** field in the Perforce preferences under **Tools > Options > Source Control.** See "Setting Perforce preferences" on page 4.

You can still submit changelists with more than the specified number of files, but the file lists are displayed as follows:

- **Pending** and **Submitted** tabs display `There are ### files in this changelist.`

- **Details** tab displays the list of files in a simple text box (with no P4VS file badges).

# Resolving conflicting changes

Conflicts occur when you attempt to integrate a file into an existing codeline or to submit a changelist containing a file that another user has edited and submitted while you had the file checked out. When the conflict occurs, Perforce schedules the file for resolve. Conflicts must be resolved before you can submit the changelist that contains the conflicting file.

When you attempt to submit a changelist containing a file that must be resolved, a Perforce Command Error is returned: `Merges still pending—use 'resolve' to merge files.` When you return to the Solution Explorer, you will see a red question-mark badge next to the file icon in Solution Explorer  : (You may need to context-click the file icon and select **Refresh** to see the question-mark badge).

If there is a yellow triangle badge on any file, get the latest revision of that file by context-clicking it and selecting **Revisions > Get Latest Revision**. This will not overwrite the copy of the file that is

in your workspace. After you have the latest revision, you can resolve the file. You can resolve files individually or attempt to resolve multiple files at once.

> **Note**　In the P4VS Resolve dialog, **Target** is the file in your workspace and **Source** is the file in the depot.

## Resolving multiple files

When there are multiple files in a changelist that need to be resolved, it is recommended that you first try to resolve them automatically.

To resolve multiple files at once, automatically:

1.  Select the files in Solution Explorer, then context-click and select **Copy/Merge > Resolve…**.

2.  In the **Resolve** dialog, select **Auto resolve multiple files**.

    The dialog displays the **Files to Resolve**. As files are resolved, they are removed from this list.

3.  Select whether to **Merge binary files as text when resolving content**.

    If you select this option, P4VS treats binary files like text files and attempts a textual merge between the source and target files.

4.  Select a **Resolve method**:

    * **Safe automatic resolve (no merging)**: Accepts the source file (the file in the depot) if it has the only changes. Accepts the target file (the file in your workspace) if it has the only changes. Doesn't resolve if both the source and target have changed.

    * **Automatic resolve (allow merging)**: Accepts the source if it has the only changes. Accepts the target file if it has the only changes. Merges changes if both the source and target have changed and there are no conflicts.

    * **Accept Source**: Replaces the copy of the file in your workspace with the version that is in the depot, discarding your changes.

    * **Accept Target**: Accepts the file that is in your workspace, overwriting the version that is in the depot when you submit the file.

    * **Automatic resolve (allow merging with conflicts)**: Accepts the source if it has the only changes. Accepts the target file if it has the only changes. Creates a merged file if both the source and target have changed, even if there are conflicts. Where there are conflicts, both versions are included with text notations indicating the conflicts.

5.  (Optional) Select **Set as Auto Default** to set your selections as the default for auto-resolving multiple files.

6.  Click **Auto Resolve**.

7.  To check in the changes, submit the changelist that includes the resolved files.

To resolve multiple files one at a time (recommended when there are conflicts):

1. Select **Interactively resolve files one at a time**.

2. Follow the procedure described in .

## Resolving individual files

To resolve an individual file:

1. Select the file in Solution Explorer, then context-click and select **Copy/Merge > Resolve…**.

2. Select **Interactively resolve files one at a time**.

   The **Resolve** dialog displays the **Files to Resolve**. If you are resolving multiple files one at a time, select the file you want to resolve. The files are removed from this list as they are resolved.

3. Select whether to **Merge binary files as text when resolving content**.

   If you select this option, P4VS treats binary files like text files and attempts a textual merge between the source and target files.

4. View the **Recommended action**.

   P4VS recommends an action, based on the differences and conflicts in the files selected. It also displays:

   • The common base file

   • The number of differences between the source and base file

   • The number of differences between the target and base file

   • The number of conflicts that would be present in the merged result.

5. Select a **Resolve method**:

   • **Accept Source**: Replaces the copy of the file in your workspace with the version that is in the depot, discarding your changes.

   • **Accept Target**: Accepts the file that is in your workspace, overwriting the version that is in the depot when you submit the file.

   • **Accept Merged**: Replaces the file in your workspace with the merged result of the target file (in your workspace) and source file (in the depot).

   • **Run merge tool**: Opens your chosen merge tool, enabling you to edit the file and save the merged result.

6. Select any **Additional Actions** that apply:

   • **Open File**: Enables you to open either version of the file individually or the merged result file in any editor.

   • **Diff**: Opens your diff tool to diff the files with each other or with the base file. It also enables you to diff the source, target, and base file with the merged file.

- **File History**: Displays the revision history of either file.

- **Time-lapse View**: Displays the history of either file using the **Time-lapse View** tool.

- **Revision Graph**: Displays the history of either file using the **Revision Graph** tool.

7. When the resolve is complete, check in the changes by submitting the changelist that includes the resolved file.

| Note | The default diff and merge tool for P4VS is P4Merge. You can set diff and merge preferences, including configuring the diff and merge tool of your choice, on the **Perforce - Diff/Merge** node under **Tools > Options > Source Control**. |
|------|------|

# Deleting files

To delete a file from the depot, you must delete it using Visual Studio, mark it for delete using P4VS, then submit the changelist containing the marked file. When you delete a file, a new revision marked *deleted* is stored in the depot and the file is removed from your workspace. Previous revisions in the depot are not affected.

To delete a file:

1. Right-click the file and choose **Delete**.

   P4VS asks if you want to mark the file for delete.

2. On the P4VS dialog, select the default pending changelist or a new changelist

3. Click **Yes**.

   P4VS marks the file for delete and it is placed in a changelist.

4. Submit the changelist containing the file. The file is deleted from the depot and your client workspace.

If you want to keep a file in your project but avoid adding it to Perforce control, use Ignore Lists. For more information, see .

# Excluding Files from Perforce Control

Your workspace may include files that you do not want to add to the Perforce repository, such as files used or generated by automated build processes.

You can use Visual Studio to exclude a file from a solution by context-clicking the file in Solution Explorer and selecting **Exclude from Project** in the context menu. If the file is under Perforce control, P4VS prompts you to mark the file for delete, and after submitting the changelist that includes the deletions, the file is removed from both the project and the Perforce repository.

You can also use *Ignore Lists* in P4VS to specify files or filetypes that you want to keep in your project but do not want to add to the Perforce repository. An **Ignore List** is a file in your local workspace

directory that contains a list of file names or file types to ignore. For example, you can create an Ignore List called `.p4ignore` in your project folder that contains the following:

```
*.swp
*~
tmp/*
.p4ignore.txt
```

(Note that the **Ignore List** file itself is included in the list.)

You can add an **Ignore List** file at any level of the solution hierarchy in your workspace. If you set your `P4IGNORE` environment variable to the file name of the **Ignore List** file, P4VS will not mark the listed files and filetypes for add, nor will it prompt you to do so.

**Ignore List**s only affect commands that search for and *add* new files. If you have already marked a file for add, P4VS will no longer ignore it, even if it or its filetype appear in an Ignore List.

You can add Ignore Lists at any folder level in your workspace (or solution). P4VS applies the rules in the **Ignore List** at the deepest folder level relative to the file being checked, along with the rules in any **Ignore List**s found in parent folders (although you can use the `!` character to override higher-level rules.)

The syntax for ignore rules is not the same as Perforce syntax. Instead, it is similar to that used by other versioning systems:

• Files are specified in local syntax

• `#` at the beginning of a line denotes a comment

• `!` at the beginning of a line excludes the file specification

• `*` wildcard matches substrings

For example:

| | |
|---|---|
| `foo.txt` | Ignore files called `foo.txt` |
| `*.exe` | Ignore all executables |
| `!bar.exe` | Exclude `bar.exe` from being ignored |

While you can set your local `P4IGNORE` environment variable and add **Ignore List**s manually, P4VS provides preferences and context menu options to simplify the process of adding and editing **Ignore List**s.

## Setting Ignore List preferences

Go to **Tools > Options > Source Control > Perforce - Ignoring Files** to set Ignore List preferences, including the **Ignore List** file name. The file name you enter in your preferences is set by P4VS as the local `P4IGNORE` environment variable and used for all of your **Ignore List**s. For more information about setting **Ignore List** preferences, see <u>"Setting Perforce preferences" on page 4</u>.

### Adding a file to an Ignore List

To add a file to an **Ignore List** in Solution Explorer, context-click the file and select **Manage Files > Add to Ignore List**. P4VS adds the file to the **Ignore List** in the current folder. If there is no **Ignore List** file in the current folder, P4VS creates one. P4VS denotes an ignored file with a gray circle glyph next to the file icon.

### Removing a file from an Ignore List

To remove a file from an **Ignore List** in Solution Explorer, context-click the file and select **Manage Files > Remove from Ignore List**. P4VS adds an exclusionary (**!**) line for the file in the **Ignore List** in the current folder, which overrides any **Ignore List**s in parent folders.

### Editing Ignore Lists

To edit an **Ignore List** in Solution Explorer, context-click any file in the same folder and select **Manage Files > Edit Ignore List**. P4VS opens the **Ignore List** file for edit. If there is no **Ignore List** file in the current folder, P4VS creates one. Use **Edit Ignore List** when you want to add file types using wildcard expressions.

# Comparing files using diff

You can compare file revisions using the diff tool associated with P4VS. The default diff tool is P4Merge, which is included with P4V. To associate a different diff tool, go to **Tools > Options > Source Control > Perforce - Diff/Merge**. For more information, see [“Setting Perforce preferences” on page 4](#).

To diff two files or file revisions:

1. In the Solution Explorer, **Submitted** tool window, **Submit Changelist** dialog, **Pending** tool window, or **Pending Changelists** dialog, context-click the file whose revisions you want to diff.

   You can also diff two file revisions from the **File History** tool window by dragging one revision row onto another.

2. Select one of the following:

   - **Diff > Diff Against…**: compare any two files or revisions of a file.

   - **Diff > Diff Against Have Revision**: compare the file version in your workspace against the depot revision that you retrieved most recently. This selection opens P4Merge (or your preferred diff tool, if it is not P4Merge) without first opening the **Diff** dialog.

   - **Diff Against Previous Revision** (from **Submitted** tool window only): compare the revision you selected against the version in the previous changelist. This selection opens P4Merge (or your preferred diff tool, if it is not P4Merge) without first opening the **Diff** dialog.

3. In the **Diff** dialog, specify the revisions of the files you want to diff:

   - **Path**: the two files you want to diff. If you choose **Workspace version on local disk**, you can ensure that all files in the workspace (including files within the client mapping that are not under

Perforce control) are displayed by using local syntax. To display only files under Perforce control, use depot syntax

- **Workspace version on local disk**: the file revision in your client workspace, including any changes you made after retrieving it from the depot and editing it.

- **Latest revision**: the revision that was most recently submitted to the depot (the head revision).

- **Have revision**: the revision you most recently retrieved. Does not include any edits you made after retrieving it from the depot.

- **Specify revision**: enables you to designate the desired revision using a revision number, changelist number, date, label, or workspace.

4.  Click **Diff**. P4VS launches P4Merge (or your preferred diff tool, if it is not P4Merge), displaying the differences between the files at the specified revision.

For more information about diffing files with P4Merge, see the P4Merge help.

# Changing Perforce file types

Perforce file types determine how a file is stored in the depot and synced (retrieved) to the workspace, and whether it can be diffed.

To change a file's Perforce file type (or other storage attributes):

1.  Context-click the file and choose **Manage Files > Change Filetype…**.

    The **Change Filetype** dialog is displayed.

2.  Set the desired type and attributes and click **OK** to dismiss the dialog.

    If the file was not checked out, P4VS checks it out and makes the change.

3.  Submit the changelist containing the file.

For details about file types and attributes, see the *P4 Command Reference*.

# Renaming and moving Files

When you rename or move a file using the Visual Studio **Rename** option, P4VS prompts you to add the renamed or moved file to a changelist. When you do, P4VS automatically marks the new file name or location for add and the old file name or location for delete. When you submit the changelist, the Perforce service creates an integration record that links the renamed or moved object to its deleted predecessor, preserving its history.

To rename a file:

1.  In Solution Explorer, context-click the file or folder you want to rename and select **Rename**.

    The file name in Solution Explorer becomes writable.

2.  Type the new name.

3. When you leave the edit box, P4VS prompts you to add the renamed file to a changelist:
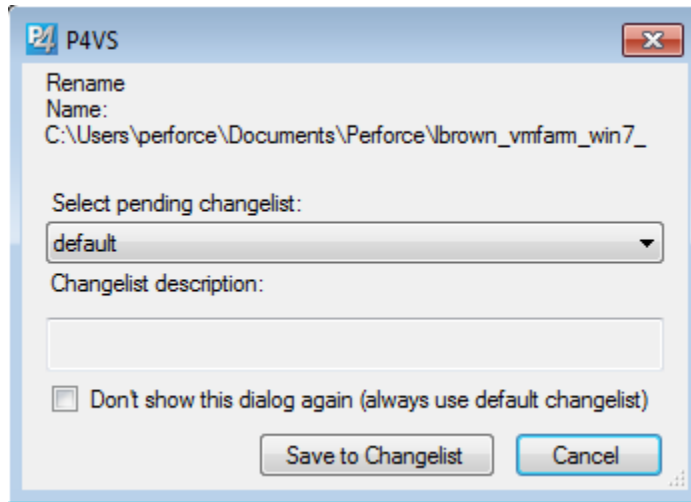


4. Select a pending changelist. The description defaults to the following, but you can change it:

```
\__path__\__old_filename__ to:
\__path__\__new_filename__
```

5. Click **Save to Changelist** to save the changes.

6. If there are code references to the renamed file in your project, Visual Studio asks if you want to rename all references.

   If you click **Yes**, Visual Studio renames all references, and P4VS prompts you to add the changes to a changelist. Follow the steps listed here to complete the process of submitting those changes to the Perforce depot.

7. Submit the changelist.

   The changelist includes **Add** operations for the new file name and **Delete** operations for the old file name.

   For more information about submitting changelists, see "Checking in files and working with changelists" on page 20.

To move a file from one location to another:

1. Context-click the file you want to move and drag it to the new location.

2. P4VS prompts you to add the moved file to a changelist.

3. Select a pending changelist. The description defaults to the following, but you can change it:

```
\__path__\__old_filename__ to:
\__path__\__new_filename_+
```

4. Click **Save to Changelist** to save the change.

5. Submit the changelist.

   The changelist includes **Move/Add** operations for the new file location and **Move/Delete** operations for the old file location.

   For more information about submitting changelists, see "Checking in files and working with changelists" on page 20.

   | **Note** | When you revert a rename or move operation in P4VS, Visual Studio continues to show the new file name or location despite the fact that Perforce has reverted it to the original name or location, unless you select **Update related projects when reverting moved files** in the P4VS preferences. For more information, see "Setting Perforce preferences" on page 4. |
   | --- | --- |

# Displaying the revision history of a file or folder

To display a file's revision history:

1. Open the **File History** tool window by doing one of the following:

   • Context-click the file or folder icon in Solution Explorer and choose **Revisions > Show History**.

   • Go to **View > File History**.

2. View file revisions in the **File History** tool window by clicking the triangle to the left of the file name.

3. To view details, including changelist descriptions, select **Details**.

4. To view integration history, select **Integrations**.

5. To view label history, select **Labels**.

6. To diff two file revisions, drag one revision row and drop it onto the other.

   This launches P4Merge (or your preferred diff tool, if it is not P4Merge), which displays the differences between the two file revisions. For more information about diffing files with P4Merge, see the P4Merge help.

# Shelving files

Shelving enables you to store copies of open files temporarily in the Perforce repository without checking them in. Shelving is useful for a variety of purposes, including taking and restoring snapshots of in-progress work and reviewing other users' code before it's checked in. When you shelve a file, a copy is placed in a pending changelist from which other users can unshelve it. Pending changelists that contain shelved files are indicated by a red triangle marked by a file icon: ▲ . When the changelist is expanded, shelved files are indicated by a file icon with a check-mark: ☑ .

When managing shelved files, note the following:

- **Basics**: To be shelved, a file must be checked out. However, you cannot unshelve a checked-out file.

- **Submitting shelved files**: As of Perforce Server 2013.1, you can submit a shelved file directly. For previous versions of Perforce Server, you must first unshelve a file to submit it, then delete the shelved copy. (Unshelving does not delete the shelved copy.)

- **Managing changelists**: You cannot move a shelved copy to another pending changelist. If you revert a file after shelving it, the copy remains shelved in the changelist until you delete it. Only the changelist owner can reshelve or delete files that are shelved in the changelist. For Perforce Server releases that predate version 2013.1, you cannot submit a changelist that contains shelved files; you must delete the shelved copies before submitting. Starting with Perforce Server 2013.1, you can submit shelved files directly, but your changelist must contain only shelved files.

- **File history**: No file history is created when you shelve or unshelve files.

- **Diffing**: You can diff shelved copies by right-clicking the shelved file in the **Pending** dialog (**View > Pending Changelists**) and selecting **Diff Against Source Revision** or **Diff Against Workspace File**.

## Shelving checked-out files

To shelve checked-out files in a pending changelist:

1. Open the **Shelve** dialog by doing one of the following:

    - Go to **View > Pending Changelists**. On the **Pending** dialog, context-click the changelist and select **Shelve…**.

    - In the Solution Explorer, context-click a file that is in a pending changelist and select **Shelve…**.

2. In the the **Shelve** dialog, select the files you want to shelve.

3. Select any of the following options that apply:

    - **Revert checked-out files after they are shelved**: The files in your workspace will revert to the head revision in the depot.

    - **Clear changelist of all previously shelved files before shelving**

4. Click **Shelve**.

5. When prompted, enter a description and click **OK**.

    P4VS shelves the file in the selected changelist or, if you are shelving files in the default changelist, creates a new changelist.

## Unshelving files

After shelving a file, you (or another user) can unshelve it, which restores the shelved copy to your workspace and opens it in the changelist of your choice. Unshelving does not remove files from the shelf. To unshelve a file that was shelved by another user, you must have permission to check out the file. When you unshelve a file that was shelved by another user, it is copied to one of your changelists, from which you can edit and submit the file.

To unshelve files in a pending changelist:

1. Context-click the file in the changelist and select **Unshelve…**. P4V displays the **Unshelve** dialog.

2. Check the files you want to unshelve and click **Unshelve** and any other desired options. The shelved file is copied to your workspace and opened in the specified changelist.

Shelved files remain shelved until you delete them from the pending changelist. **To delete a shelved file from a pending changelist**, context-click the file and select **Delete**. You can also context-click the pending changelist and select **Delete Shelved Files…**.

## Submitting shelved files

As of Perforce Server 2013.1, you can submit shelved files directly.

| **Note** | If there are non-shelved files along with shelved files in a pending changelist, you must first revert the non-shelved files or move them to another changelist. You cannot submit shelved files from a task stream. |
|---|---|

To submit shelved files in a pending changelist, context-click the changelist and choose **Submit Shelved Files…**.

# Working with Streams

This chapter explains how to use Helix P4VS with Helix streams.

Before reading this chapter, review the "Streams" chapter in the *Helix Versioning Engine User Guide* and the "Basic Concepts" chapter in *Introducing Helix*, which explain fundamental stream concepts.

You may also find it helpful to see the P4V help (launch P4V from the context menu using **Views > View in P4V**, and click **Help** on the P4V toolbar).

# Using the Streams tool window

P4VS provides two ways to view streams graphically: you can use the **Streams tool window** directly in P4VS, or you can call the **Stream Graph**, a P4V component, from within P4VS. This topic discusses how to use the Streams tool window.

In the **Streams** tool window, status indicators between streams tell you which streams have changes to contribute and where the changes can be copied or merged:

| | |
|---|---|
| ⬇ | Merge indicator |
| ⬆ | Copy indicator |

The arrows are color-coded to indicate status:

• Gray: no merge or copy required

• Green: a merge or copy operation is available

• Orange: stream must be updated, after which merge or copy is available

For example, the following arrows next to a stream indicate that you must update it by merging down from its parent, after which you can copy up changes to the parent.

⬇ ⬆

Context-clicking on a stream in the **Streams** tool window shows the available copy and merge actions that you can perform. If you need to work in another stream to complete an action, you are prompted to switch workspaces, create a new workspace, or select a workspace from an available list depending on the existing workspaces that are available for use with the target stream. From there you can preview the copy or merge operation and complete it. After the copy or merge is done, you are prompted to select a changelist (if the preference is set for changelist prompts) and then to save or submit that changelist. When the merge or copy workflow is complete, your connection changes back to the original workspace that was in use if the workspace was switched during the merge or copy process.

## Creating and editing streams

You must use P4V, the Perforce Visual Client, or **p4**, the Perforce Command-Line Client, to create and edit streams. For more information, see the P4V help or the *P4 User Guide*.

## Displaying and searching for streams

To display the streams defined for the Perforce depot to which you are connected in P4VS:

1. Go to **View > Streams** to open the **Streams** tool window.

2. Search for streams using the filter fields.

   You can filter by any combination of the following:

   - Depot (requires an entry)

   - Owner

   - Name

   - Parent

   - Type

Use standard Perforce syntax (`//streamdepot/stream`). For more information, see the *P4 User Guide*.

Note that because this tool window provides a hierarchical view of streams, you may see parent streams that do not match the filter. These are included in the list to show the hierarchy of the streams all the way to the related mainline, but are grayed out.

To change the order in which columns are displayed, drag the column headings right or left.

The details of a selected stream are displayed in the lower pane.

# Using the Stream Graph

P4VS provides two ways to view streams graphically: you can use the **Streams** tool window directly in P4VS, or you can call the **Stream Graph**, a P4V component, from within P4VS. This section discusses how to use the **Stream Graph**.

The **Stream Graph** provides a graphical view of stream relationships and provides tools and shortcuts for working with streams.

The graph uses location and color to depict stream types: mainline streams are gray and placed in the middle of the graph, release streams are orange and appear above the mainline, and development streams are blue and appear below. For example:
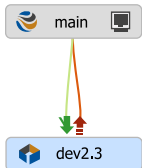


Status indicators between streams tell you which streams have changes to contribute and where the changes can be copied or merged:

| | |
|---|---|
| ⬇ | Merge indicator |
| ⬆ | Copy indicator |

The arrows are color-coded to indicate status:

- Gray: no merge or copy required

- Green: a merge or copy operation is available

- Orange: stream must be updated, after which merge or copy is available

For example, the following arrows next to a stream indicate that you must update it by merging down from its parent, after which you can copy up changes to the parent.



The workspace icon indicates the stream you are currently working in.

## Accessing the Stream Graph from P4VS

Go to **File > Perforce > Views > Stream Graph** or right-click in the Solution Explorer and select **Views > Stream Graph**.

| Note | The **Stream Graph** is a P4V component. When you are working in the **Stream Graph**, you are working in P4V. |
| --- | --- |

## Setting Stream Graph display options

Select display options in the **Graph View Options** dialog:

1. In the **Depot** drop-down list, select the depot containing the streams you want to view. By default, the graph shows the stream containing the files you are currently working in.

2. To select the streams you want displayed in the graph, click **Select Streams** and choose the display option, or check the individual streams that you want displayed in the graph. You may need to expand the tree within the dialog pane to view the streams you want to select.

3. Click **Apply Filter**. The stream graph displays the streams that you specified.

4. (Optional) In the **Graph Navigator** dialog, configure the size of the stream graph display and select which portion of the stream graph to view. Use your mouse or cursor keys on the navigator pane to select the portion of the image you want to view.

## Displaying stream status

Double-click a stream to view a pop-up that contains status details:

## Working in a stream

To work in a stream or switch from one stream to another using the **Stream Graph**, do one of the following:

- Double-click the stream and select **Work in this stream**.

- Drag the workspace icon (▣ ) from the stream you are working in to the one you want to work in.

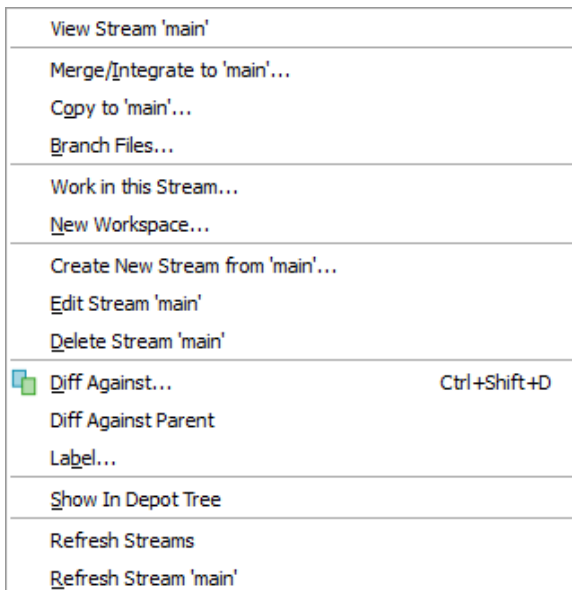| **Important** | In order to switch streams in P4VS using the **Stream Graph**, you must set your P4V stream operations preference to **Use the same workspace and switch it between streams**. |
|---|---|

If you have not set this preference, a warning dialog pops up when you try to switch streams and asks you to switch workspaces or create a new one. If you then click the **Switch Workspaces** button, the dialog closes, as does the **Stream Graph**, with the workspace unswitched. If you click the **New Workspace** button, the **Workspace:New** dialog opens. You can create a new workspace, but the dialog and **Stream Graph** close without switching workspaces in P4VS.

If you do not want to use the same workspace when switching streams in P4VS, you must open a new connection to the Perforce service to select a new workspace.

For more information about setting P4V preferences, see "Configuring P4V Preferences" in the P4V help (launch P4V from the context menu using **Views > View in P4V**, and click **Help** on the P4V toolbar).

## Other actions you can perform with the Stream Graph

When you right-click a stream in the **Stream Graph**, you see the following options:

```
View Stream 'main'

Merge/Integrate to 'main'...
Copy to 'main'...
Branch Files...

Work in this Stream...
New Workspace...

Create New Stream from 'main'...
Edit Stream 'main'
Delete Stream 'main'

▣ Diff Against...                        Ctrl+Shift+D
  Diff Against Parent
  Label...

  Show In Depot Tree

  Refresh Streams
  Refresh Stream 'main'
```

To learn about these streams options, see the P4V help (launch P4V from the context menu using **Views > View in P4V**, and click **Help** on the P4V toolbar).

# Merging down and copying up between streams

Before changes made in a less stable stream can be copied up to its more stable child or parent, any changes in the more stable stream must be merged down to the less stable.

## Merging down

To merge changes down to a less stable stream:

1. Go to **File > Perforce > Copy/Merge > Merge to Stream…** or context-click in the Solution Explorer and select **Copy/Merge > Merge to Stream…**.

   When you merge down or copy up, you must be working in the target stream.

2. In the **Merge** dialog, select the **Source Stream** (the stream you want to merge down changes from). This must be a parent of the target stream.

3. (Optional) Click **Preview** to view the merge results.

4. Click **Merge**.

5. If necessary, resolve the merges manually, then submit the resulting changelist.

If you want to merge changes between streams without working in the target stream, open the **Streams** tool window, context-click a stream that shows a pending **Merge** indicator, and select **Merge to 'streamname' from parent**.

If you want to use advanced options when merging changes between streams, launch the Stream Graph, context click the stream you want to merge down to, and select **Merge/Integrate to 'streamname'**. You can also use P4V or the Perforce Command-Line Client. For more information about the full set of **Merge** options, see the P4V help (launch P4V from the context menu using **Views > View in P4V**, and click **Help** on the P4V toolbar).

## Copying up

When you copy changes up to a more stable stream, you are propagating a duplicate of the less stable stream.

To copy changes up to a more stable stream:

1. Go to **File > Perforce > Copy/Merge > Copy to Stream…** or right-click in the Solution Explorer and select **Copy/Merge > Copy to Stream…**.

   When you merge down or copy up, you must be working in the target stream.

2. In the **Copy** dialog, select the **Source Stream** you want to copy from.

3. (Optional) Click **Preview** to view the copy results.

4. Click **Copy**.

5. Submit the resulting changelist.

If you want to copy changes between streams without working in the target stream, open the **Streams** tool window, context-click a stream that shows a pending **Copy** indicator, and select **Copy to 'streamname' from parent** or **Copy to parent from 'streamname'**.

If you want to use advanced options when copying changes between streams, launch the **Stream Graph**, context click the stream you want to copy up to, and select **Merge/Integrate to 'streamname'**. You can also use P4V or the Perforce Command-Line Client. For more information about the full set of **Copy** options, see the P4V help (launch P4V from the context menu using **Views > View in P4V**, and click **Help** on the P4V toolbar).

## Propagating change between unrelated streams

To propagate change between streams that are not directly connected, use P4V or the Perforce Command-Line Client.

You can also reparent a stream to create the relationship. To reparent a stream in the **Stream** graph, drag the stream to the new parent stream. Note that you cannot reparent a task stream.

For more information, see "Merging Down and Copying Up between Streams" in the P4V help (launch P4V from the context menu using **Views > View in P4V**, and click **Help** on the P4V toolbar).

# Chapter 4 — Using Other Perforce Features

This chapter discusses how to take advantage of other Perforce features available from within P4VS.

# Viewing integration history in the Revision Graph

The **Revision Graph** displays file integration history, showing when a file was added, branched, edited, integrated, and deleted.
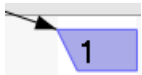
## Launching Revision Graph

Right-click a file or folder in the Solution Explorer or go to **File > Perforce** and select **Views > Revision Graph**.

| Note | The **Revision Graph** is a P4V component. When you are working in the **Revision Graph**, you are working in P4V. |
|------|---|

## Reading the Revision Graph

Each revision of a file is represented by a shape. The shape denotes the action that created the revision. For example, the following shape indicates that the revision was created by branching the file:



When multiple revisions contribute to an integration, **Revision Graph** displays a bracket below the contributing revision, as shown in the following figure:



To display details about the meaning of the shapes and the lines that connect them, click the **Legend** tab in the lower right pane.

The top bar of the revision graph displays the changelist that created the file revision. To view the changelist (or sync to it or integrate it), context-click the changelist number.

## Navigating the Revision Graph

To select revisions, click them or use the arrow keys. Details about the selected revision are displayed in the lower right-hand pane. To select multiple revisions, control-click them.

For files that have a large history, **Revision Graph** displays a portion of the graph in its main window, and a map of the graph in the lower left **Navigator** tab. A box in the **Navigator** outlines the portion displayed in the main window.

To navigate the diagram:

• drag the box in the Navigator pane, or

• use the main window scrollbars, or

• in the main window, use the mouse wheel or middle button.

To zoom in or out, move the slider in the toolbar or hold down the **CTRL** key and use the mouse wheel.

Highlighting shows the revisions that have contributed content to the selected revision or received content from it. To highlight file revisions, select the revision of interest and choose an option from the **Highlight** menu.

To diff two revisions, drag one revision to another or select the revisions, then context-click and choose **Diff Revisions**.

To move a line of revisions up or down: select it and click **CTRL**+**up arrow** or **CTRL**+**down arrow**.

## Filtering the Revision Graph

To reduce the detail displayed in the main window, you can filter the information. To remove a file or folder from the main window, uncheck it in the **File Filter** tree.

To enter a more precise file filter, click **Advanced…** and enter the file specification for the files and folders you want to retain in the main window (or, for files and folders you want to exclude, exclusionary lines preceded by "-"), check any filtering options you want to apply, then click **Filter**. To retain this filter in effect for future invocations of **Revision Graph**, click **Set as Default**.

To further compress the detail displayed in the main window, toggle the options on the **View** menu as follows:

• **File Renames Collapsed**: displays renamed files on a single line instead of multiple lines.

• **Compressed Integration History**: displays only revisions that were branched or integrated.

To compress file rename operations by omitting intervening revisions, choose **View > File Renames Collapsed**. **Revision Graph** displays the original and renamed file, indicating the operation with an angled arrow, as follows:
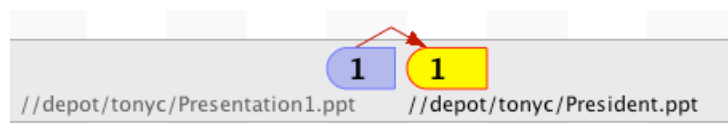


## Displaying details

To display details about a file revision, click the revision in the main window. Details are displayed in the lower left pane.

Related revisions are listed on the **Integrations** tab. To get the revision, diff it, or display its history, context-click the revision on the **Integration** tab. To view integrated revisions in the main window, click the corresponding icon on the **Integrations** tab.

# Viewing file history with Time-lapse View

**Time-lapse Viewe** provides an interactive graphical representation of a file's history, showing when lines were added, changed, and deleted, who made the changes, and when the changes were made. **Time-lapse View** enables you to browse forward and back through changes dynamically, enabling you to locate changes of interest. Detail panes at the bottom of the window provide more information about selected chunks.

## Displaying Time-lapse View

Right-click in the Solution Explorer or go to **Files > Perforce** and select **Views > Time-lapse view**.

| Note | **Time-lapse View** is a P4V component. When you are working in **Time-lapse View**, you are working in P4V. |
|------|------|
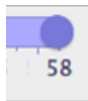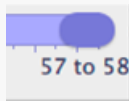
## Controlling the display

The following options are available on the toolbar:

| Mode | Determines how many revisions are displayed. Options are:<br><br>• Single revision: one revision at a time is displayed<br><br>• Incremental diffs: two adjacent revisions are displayed, with changes highlighted<br><br>• Multiple revisions: a range of revisions is displayed, with changes highlighted |
|------|------|
| **Content Range** | Specifies the starting and ending revision displayed. |
| **Scale** | Specifies the unit used: changelist number, date, or revision number. |
| **User** 👤 | Toggles display of the user that made the change. |
| **Aging** 🟩 | Displays color coding to indicate how recently a change was entered. The darker the shading, the more recent the change. |
| **Line Numbers** | Toggles display of line numbers. |
| **Lifetimes** | Toggles display of lifetimes, which are graphics that indicate by their width how long the adjacent chunk of text has been in the file. |
| **Branch History** 🌿 | Toggles inclusion of branching (integration) history. |

| | |
|---|---|
| **Find** 🔍 | Search text |
| **Go To** 🔧 | In single revision mode, go to specified line number. |
| **Next/Previous Diff** ⬅ ➡ | Go to next or previous modification |
| **Line Ending** 🔧. | Specifies how line endings and whitespace are treated to determine differences |

The slider enables you to browse rapidly through file revisions. The appearance of the slider corresponds to the mode you select. The unit by which the slider advances is specified by the mode you select (date, changelist, or revision). The revision, date, or changelist number is displayed under the slider.

| Mode | Slider Appearance | Description |
|---|---|---|
| **Single revision** | 58 | Move it to the right to display the next file revision or left to display the previous revision. |
| **Incremental diffs** | 57 to 58 | Move it to the right to display the next pair of file revisions, or left to display the previous pair of file revisions. |
| **Multiple revisions** | 53 55 58 | Move the right and left halves separately, to control how many revisions are displayed. |

# Viewing a project in P4V, the Perforce Visual Client

P4V is the dedicated visual client application for Perforce. It provides a rich interface for managing your projects under source control.

To view a project or file in P4V, context-click the project or files you want to view in P4V and select **Views > View in P4V**.

FIXME For more information about P4V, see *Getting Started with P4V* and the P4V online help.

# Using jobs (defect tracking)

Jobs enable you to record requests for work. You can associate jobs with changelists to track the work done to fulfill the request. When you submit the changelist, the job can be closed.

## Creating jobs

1. Go to **View > Jobs**.

2. In the **Jobs** tool window, context-click anywhere in the job list pane and select **New Job…**

3. Fill in the **Job** form.

   The fields that appear on the **Job** form depend on the customizations set up by your Perforce administrator. For more information, see the *Perforce Server Administrator Guide: Fundamentals*.

4. Click **OK**.

## Editing jobs

1. Go to **View > Jobs**.

2. In the **Jobs** tool window, context-click a job row and select **Edit Job…**.

3. Update the **Job** form.

   The fields that appear on the Job form depend on the customizations set up by your Perforce administrator. For more information, see the *Perforce Server Administrator Guide: Fundamentals*.

4. Click **OK**.

## Displaying jobs

To view jobs:

1. Go to **View > Jobs**.

   In the **Jobs** tool window, enter search terms in the **Keywords** field or the depot directory path in the **Folder/file** field.

   For keyword syntax, see "Filtering Expressions" on page 46.

   Use the **Folder/file** field when you know the location of a file that is included in an associated changelist. Enter the directory path using Perforce syntax (`//depot/folder/folder/filename` or `//depot/folder/…`)

2. Click **Filter**.

3. Click a job row to view details about the job.

To change the order in which columns are displayed, drag the column headings right or left to the desired position. To sort by column, click the sort arrow on a column heading.

## Associating changelists with jobs

To add a job to a pending changelist:

1. Open the **Submit** dialog.

2. Select a changelist in the **Link jobs to changelist** list.

If the job you want is not on the list, add it by clicking **Browse…**. In the **Jobs Browser**, find and select the job you want. For keyword syntax, see .

3. Specify the **Job status upon submit**: `open`, `suspended`, or `closed`.

You can also add a changelist to a job by editing the job. For more information, see .

# Filtering Expressions

Valid job filtering expressions are as follows:

| Syntax | Description | Example |
|---|---|---|
| `word word word` | Words separated by spaces indicate that the job must contain all the words in the string in any of the job fields to be included in the filter. Spaces represent the boolean "and". | `filter file mailbox`<br><br>Displays jobs containing all the words "filter", "file", and "mailbox" in any of the job fields. |
| `word \| word \| word` | Displays jobs that contain any of the specified words. Pipes represent the boolean "or". | `filter\|file\|mailbox`<br><br>Displays jobs containing the words "filter", "file" or "mailbox". |
| `^word` | Displays jobs that do not contain the specified word. The 'not' (^) operator cannot be used alone or with the 'or' operator (\|), only with the 'and' operator (& or space). | `filter ^file`<br><br>Displays jobs that contain "filter" and do not contain "file". |
| `fieldname = value` | Displays jobs that include the specified value in the specified field. | `status=open owner=edk`<br><br>Displays open jobs owned by `edk`. |
| `^fieldname = value` | Displays jobs that do not include the specified value in the specified field. The 'not' (^) operator cannot be used alone or with the 'or' operator (\|), only with the 'and' operator (& or space). | `^status=closed& subsystem=parser`<br><br>Displays unclosed jobs affecting the `parser` subsystem. |
| `fieldname = value +*` | Displays jobs that contain the specified value in the specified field, including any combination of characters in the position of the asterisk wildcard. | `owner=*ed*`<br><br>Displays jobs in which the value of the field `owner` contains the substring `ed`, including such values as `Ted`, `Edk`, and `Fred`. |

# Using Labels

Labels can be used to mark important file revisions, such as the set of file revisions used to build a particular software release. You can use labels to specify groups of related file revisions when you get file revisions (sync), compare file revisions (diff), and integrate (merge, copy, and branch).

To use labels, you first define the label and then apply the label to file revisions in the depot.

## Creating and editing labels

You must use P4V, the Perforce Visual Client, or **p4**, the Perforce Command-Line Client, to create and edit labels. For more information, see the P4V help or the *P4 User Guide*.

## Labeling files

You must use P4V, the Perforce Visual Client, or **p4**, the Perforce Command-Line Client, to apply labels to files. For more information, see the P4V help or the *P4 User Guide*.

## Displaying and searching for labels

To display the labels defined for the Perforce depot to which you are connected in P4VS:

1. Go to **View > Labels** to open the **Labels** tool window.

2. To search for labels, use the filter fields.

   You can filter by any combination of the following:

   • Owner

   • Label name

   • File path

   Use standard Perforce syntax (**//depot/folder/folder/filename** or **//depot/folder/...**). For more information, see the *P4 User Guide*.

   To change the order in which columns are displayed, drag the column headings right or left to the desired position. To sort by column, click the sort arrow on a column heading.

3. To view details about a label, such as the owner, description, and view, select the label row and click **Details** in the lower pane.

4. To view a list of files in a label, select the label row and click **Files** in the lower pane.

## Retrieving files by label

To retrieve a file revision in a label:

1. Context-click the file in Solution Explorer and select **Revisions > Get Revision…**.

2. Select **Specify revision using: Label** and browse for the label.

3. (Optional) Select **Remove files from workspace if they are not in label** to ensure that your workspace contains only the labeled file revisions.

# Working With Reviews in Swarm

Perforce Swarm is a powerful and flexible code review and collaboration solution that helps teams ship quality software faster. Swarm enables review of code and other assets before or after commit and can be customised to fit into various workflows. Swarm stores all of its metadata including reviews, projects and comments in Perforce, which makes it an attractive solution since it doesn't require backing up an external database. For more about using and installing Swarm, please see the *Swarm User Guide*.

## Workflow of a review

Below is the happy path workflow for a Swarm review. There are more permutations and variations that are described in the Swarm documentation.

1. **Make local changes to files**: Swarm reviews can follow either a pre-commit or post-commit workflow. In both models, the author would make some local content changes to one or more files and then get those content changes into Perforce.

2. **Request a review**: For pre-commit code reviews, the Swarm solution uses Perforce shelving technology to get the content to Perforce. For post-commit code reviews, content committed to Perforce is added to a review. In both cases, a Swarm review is created with an id, a description, a set of files and other meta-data including the author, reviewers and comments made on the review.

3. **Provide review feedback**: Reviewers can comment on files or on individual lines of files using Swarm. Reviewers can also add follow-up tasks that the author would be asked to address before the review could be closed.

4. **Request revisions**: If the reviewers find the review needs more work, which is often the case, they can change the state of a review to **Needs Revision**, thereby notifying the author that the review is back in their court.

5. **Request further review**: Authors can request further review of their review content changes and update any of the tasks they were asked to complete, thereby notifying the reviewers that they are ready for more of their feedback.

6. **Approve or reject review**: Reviews can be approved or rejected using Swarm. Once a review is approved or rejected, it is considered closed.

7. **Commit the review.** For pre-commit reviews, authors can commit reviews using their Perforce clients such as P4V or P4VS. For this scenario, committing a pre-commit code review is synonymous with submitting the changelist associated with the review. They can also optionally use Swarm to commit pre-commit reviews.

## Setting up the Swarm integration

A minimum requirement for the P4VS integration is to run Swarm version 2014.4.

None of the new features for Swarm are available unless the Swarm integration is turned on. This integration needs to be turned on for each Perforce server. In order to make P4VS enable the Swarm features, the Perforce administrator must run the **p4 property** command for the Swarm URL. This will tell the Perforce server the Swarm URL. The P4VS integration uses this URL when making API requests to the Swarm server.

Example **p4 property** command to run:

```
p4 property -a -n P4.Swarm.URL -v https://_swarm.yourcompanydomain.com_
```

where _https://swarm.yourcompanydomain.com_ is the URL for the Swarm server.

If you are testing the Swarm integration, you may wish to set the property for a specific user. For example, to enable the Swarm integration for the user _username_:

```
p4 property -a -u _username_ -n P4.Swarm.URL -v https://_swarm.yourcompanydomain.com_
```

Similarly, you can enable the Swarm integration for a specific group of users. For example, to enable the Swarm integration for the group _group_:

```
p4 property -a -g _group_ -n P4.Swarm.URL -v https://_swarm.yourcompanydomain.com_
```

## Authentication with Swarm

P4VS uses the user's existing Perforce ticket to communicate with Swarm. If you get authorization errors, ensure that the **Use IP-specific tickets when logging in** is disabled. This is synonymous with using the **-a** option with the **p4 login** command so that the ticket can be used on any machine.

## Swarm integration features

Once the Swarm integration is enabled a number of new features are available in P4VS including new context menus, review request and update dialogs, badging on pending and committed changes, as well as **Review ID** and **Review State** columns.

## Request a review

Reviews can be requested from either pending or submitted changelists. Note that a changelist cannot be associated with more than one review, however a review can have more than one changelist associated with it.

Pre-commit code reviews are a more popular approach since they allow validating of code and correcting defects before they become a part of the committed code-base. Swarm supports pre-commit code reviews via pending changelists.

Post-commit code reviews allow reviewers to provide feedback on the submitted content and they warrant that the author follow on with more submitted changes when wanting to make the updates

recommended by the reviewers. Development branches are well-suited for the post-commit review process.

## Request a review from a pending changelist

To request a review from a pending changelist, go to **View > Pending changelists**, select the changelist and choose the **Request New Swarm Review…** from the context menu. Note that if the changelist is already part of a Swarm review, this option is not available.

The **Request New Swarm Review** dialog displays a list of files to be shelved in order to request the review. If the changelist already has shelved files, the dialog also lists these already shelved files. The aggregate of the shelved files comprises the review. The review must have a description, which defaults to the changelists' description. The dialog offers additional options including: reviewers, reverting checked out files after they are shelved, not shelving unchanged files, and opening the review in Swarm.

Once the review has been requested, the pending changelist is badged with a Swarm icon and P4VS updates the **Review ID** and the **Review State** fields with their values from Swarm.

It is a best practice for the author to keep this pending changelist for subsequent updates to the review. This same changelist can be used by the author to submit the review. If the review is rejected or the review is committed from Swarm, then the author should manually discard this pending change so that it does not get accidentally committed.

## Request a review from a submitted changelist

To request a review from a submitted changelist, go to the **Submitted changelist** tab, select the changelist and choose the **Request New Swarm Review…** option from the context menu. Note that if the changelist is already part of a Swarm review, this option is not available.

The **Request New Swarm Review** dialog displays the files that to be added to the review. The review must have a description, which defaults to the changelists' description. The dialog offers additional options including: reviewers and opening the review in Swarm.

Once the review has been requested, the pending changelist is badged with a Swarm icon and P4VS updates the **Review ID** and the **Review State** fields with their values from Swarm.

# Update Swarm Review

If you need to update the files in a review for any reason, such as to respond to the feedback you received from the reviewers, P4VS provides an option to update an existing Swarm review.

## Update a Swarm review from a pending changelist

To update a review from a pending changelist that is associated with the review, go to **View > Pending changelists**, select the changelist and choose the **Update Swarm Review '*xxxx*"…** option from the context menu, where *xxxx* is the review id.

The **Update Files in Review** dialog displays a list of files to be shelved in order to update the review. If the changelist already has shelved files, the dialog also lists these already shelved files. The aggregate of the shelved files comprises the updated review. You can also update the review description at this

time. The dialog offers additional options including: reverting checked out files after they are shelved, not shelving unchanged files, and opening the review in Swarm.

### Update a Swarm review from a submitted changelist

To associate a submitted changelist with an existing Swarm review, select the submitted changelist and choose **Add to Swarm Review** context menu option.

The **Add to a Swarm Review** dialog displays a list of files to be added to a review. The dialog has a field where you can enter the review id of the review to which you'd like to add these files. Type in the review id in the **Update Review** field and click the **View Review Description** button if you want to see a preview of the review's description in order to verify that this is in fact the review you'd like to add these files to. The dialog offers an additional options to open the review in Swarm.

## Open review in Swarm

If you leave the **Open Review in Swarm** checkbox option selected in the **Review Request** or **Review Update** dialogs, then P4VS launches Swarm to the review page in your default browser. This serves as confirmation that the review has been created or updated.

If a pending or submitted changelist is already associated with a review, context click the changelist and select **Open Review 'xxxx' in Swarm…** to open the associated review in your default web browser, where *xxxx* is the id of the associated review.

## Review Id and Review State columns

P4VS will add a **Review Id** and **Review State** column to both the submitted and pending changelist tabs for connections that have the Swarm integration enabled.

If you are connected to a Perforce server with the Swarm integration enabled and do not see the columns, right click on the header row and select these fields.

**Appendix** # License Statements

Perforce software includes software developed by the University of California, Berkeley and its contributors. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

Perforce software includes software developed by the OpenLDAP Foundation (http://www.openldap.org/).

Perforce software includes software developed Computing Services at Carnegie Mellon University: Cyrus SASL (http://www.cmu.edu/computing/).

Perforce software includes software from the NLog project (http://nlog-project.org/), available under the terms of BSD license. (https://github.com/NLog/NLog/blob/master/LICENSE.txt)