Preface

About This Manual ....................................................... 7
Please Give Us Feedback .................................................. 7

Chapter 1

Getting Started with P4VS ........................................... 9

About P4VS, the Perforce Plugin for Visual Studio ............... 9
Basic Perforce Terminology .............................................. 9
Basic Tasks ........................................................................ 10
Using Solution Explorer with P4VS .................................... 10
Using P4VS toolbars in Visual Studio ................................. 11
For more information ...................................................... 11
Installing P4VS and enabling the extension in Visual Studio .... 11
Installing P4VS in Visual Studio 2008 ................................. 11
Installing P4VS in Visual Studio 2010 ................................. 12
Installing P4VS in Visual Studio 2012 ................................. 12
Enabling P4VS in Visual Studio (all supported versions) ....... 12
Migrating from P4SCC to P4VS ......................................... 13
Setting Perforce preferences ............................................. 13
Perforce - Connections ..................................................... 13
Perforce - Data Retrieval .................................................. 14
Perforce - Diff/Merge ....................................................... 15
Perforce - General .......................................................... 15
Perforce - Ignoring Files .................................................. 17
Perforce - Logging .......................................................... 18
Keyboard shortcuts ........................................................ 18
Connecting to Perforce services ....................................... 18
Defining a new Perforce service connection ....................... 18
Setting Perforce connection settings using environment variables 19
Opening a defined Perforce service connection ................. 20
Setting Perforce environment variables using P4CONFIG ........ 20
Customizing context menus .......................................... 21
Managing workspace specifications ................................. 22
Creating workspaces .................................................... 22
Changing your workspace .............................................. 24
Viewing workspaces .................................................... 24
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream workspaces</td>
<td>24</td>
</tr>
<tr>
<td>Defining a workspace view</td>
<td>25</td>
</tr>
<tr>
<td>Syntactic view specification</td>
<td>25</td>
</tr>
<tr>
<td>Chapter 2</td>
<td></td>
</tr>
<tr>
<td>Managing Files</td>
<td>27</td>
</tr>
<tr>
<td>Putting a project or solution under Perforce source control</td>
<td>27</td>
</tr>
<tr>
<td>Adding files to the depot</td>
<td>27</td>
</tr>
<tr>
<td>Opening a project or solution in the Perforce depot</td>
<td>28</td>
</tr>
<tr>
<td>Retrieving files from the depot</td>
<td>29</td>
</tr>
<tr>
<td>Checking out and editing files</td>
<td>29</td>
</tr>
<tr>
<td>Checking in files and working with changelists</td>
<td>30</td>
</tr>
<tr>
<td>Checking in files</td>
<td>31</td>
</tr>
<tr>
<td>Displaying changelists</td>
<td>31</td>
</tr>
<tr>
<td>Editing changelists</td>
<td>32</td>
</tr>
<tr>
<td>Restricting access to changelists</td>
<td>33</td>
</tr>
<tr>
<td>Moving a file to another changelist</td>
<td>34</td>
</tr>
<tr>
<td>Setting changelist display preferences</td>
<td>34</td>
</tr>
<tr>
<td>Resolving conflicting changes</td>
<td>34</td>
</tr>
<tr>
<td>Resolving multiple files</td>
<td>35</td>
</tr>
<tr>
<td>Resolving individual files</td>
<td>36</td>
</tr>
<tr>
<td>Deleting files</td>
<td>37</td>
</tr>
<tr>
<td>Excluding Files from Perforce Control</td>
<td>38</td>
</tr>
<tr>
<td>Setting Ignore List preferences</td>
<td>39</td>
</tr>
<tr>
<td>Adding a file to an Ignore List</td>
<td>39</td>
</tr>
<tr>
<td>Removing a file from an Ignore List</td>
<td>39</td>
</tr>
<tr>
<td>Editing Ignore Lists</td>
<td>39</td>
</tr>
<tr>
<td>Comparing files using diff</td>
<td>40</td>
</tr>
<tr>
<td>Changing Perforce file types</td>
<td>41</td>
</tr>
<tr>
<td>Renaming and moving Files</td>
<td>41</td>
</tr>
<tr>
<td>Displaying the revision history of a file or folder</td>
<td>43</td>
</tr>
<tr>
<td>Shelving files</td>
<td>44</td>
</tr>
<tr>
<td>Shelving checked-out files</td>
<td>44</td>
</tr>
<tr>
<td>Unshelving files</td>
<td>45</td>
</tr>
<tr>
<td>Submitting shelved files</td>
<td>45</td>
</tr>
<tr>
<td>Chapter 3</td>
<td></td>
</tr>
<tr>
<td>Working with Streams</td>
<td>47</td>
</tr>
<tr>
<td>About streams</td>
<td>47</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Mainline model</td>
<td>47</td>
</tr>
<tr>
<td>Stream types</td>
<td>47</td>
</tr>
<tr>
<td>Stream hierarchy</td>
<td>48</td>
</tr>
<tr>
<td>Stream views</td>
<td>49</td>
</tr>
<tr>
<td>Propagating change between streams</td>
<td>50</td>
</tr>
<tr>
<td>Stream depots</td>
<td>50</td>
</tr>
<tr>
<td>For more information about streams</td>
<td>50</td>
</tr>
<tr>
<td>Using the Streams tool window</td>
<td>50</td>
</tr>
<tr>
<td>Creating and editing streams</td>
<td>51</td>
</tr>
<tr>
<td>Displaying and searching for streams</td>
<td>51</td>
</tr>
<tr>
<td>Using the Stream Graph</td>
<td>52</td>
</tr>
<tr>
<td>Accessing the Stream Graph from P4VS</td>
<td>53</td>
</tr>
<tr>
<td>Setting Stream Graph display options</td>
<td>53</td>
</tr>
<tr>
<td>Displaying stream status</td>
<td>54</td>
</tr>
<tr>
<td>Working in a stream</td>
<td>54</td>
</tr>
<tr>
<td>Other actions you can perform with the Stream Graph</td>
<td>55</td>
</tr>
<tr>
<td>Merging down and copying up between streams</td>
<td>55</td>
</tr>
<tr>
<td>Merging down</td>
<td>55</td>
</tr>
<tr>
<td>Copying up</td>
<td>56</td>
</tr>
<tr>
<td>Propagating change between unrelated streams</td>
<td>56</td>
</tr>
<tr>
<td>Viewing integration history in the Revision Graph</td>
<td>59</td>
</tr>
<tr>
<td>Launching Revision Graph</td>
<td>59</td>
</tr>
<tr>
<td>Reading the Revision Graph</td>
<td>59</td>
</tr>
<tr>
<td>Navigating the Revision Graph</td>
<td>60</td>
</tr>
<tr>
<td>Filtering the Revision Graph</td>
<td>60</td>
</tr>
<tr>
<td>Displaying details</td>
<td>61</td>
</tr>
<tr>
<td>Viewing file history with Time-lapse View</td>
<td>61</td>
</tr>
<tr>
<td>Displaying Time-lapse View</td>
<td>61</td>
</tr>
<tr>
<td>Controlling the display</td>
<td>62</td>
</tr>
<tr>
<td>Viewing a project in P4V, the Perforce Visual Client</td>
<td>63</td>
</tr>
<tr>
<td>Using jobs (defect tracking)</td>
<td>63</td>
</tr>
<tr>
<td>Creating jobs</td>
<td>63</td>
</tr>
<tr>
<td>Editing jobs</td>
<td>64</td>
</tr>
<tr>
<td>Displaying jobs</td>
<td>64</td>
</tr>
<tr>
<td>Associating changelists with jobs</td>
<td>64</td>
</tr>
<tr>
<td>Filtering Expressions</td>
<td>65</td>
</tr>
</tbody>
</table>
Table of Contents

Using Labels .................................................................................................................. 66
Creating and editing labels ....................................................................................... 66
Labeling files ............................................................................................................. 66
Displaying and searching for labels ........................................................................ 66
Retrieving files by label ............................................................................................. 67
Using P4Sandbox ....................................................................................................... 67
Prerequisite .................................................................................................................. 67
Creating a local P4Sandbox ...................................................................................... 67
About This Manual

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

This manual assumes some familiarity with Perforce. For basic information on Perforce, see the guide *Introducing Perforce*.

All of our documentation, along with tutorial videos, eLearning courses, release notes, and knowledge base articles, is available from our web site at http://www.perforce.com.

Please Give Us Feedback

We are interested in receiving opinions on it 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.
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 (Visual Studio 2012 only)
  - Ignored (Visual Studio 2012 only)
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**

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

For the P4VS release notes, see [http://www.perforce.com/perforce/doc.current/user/p4vsnotes.txt](http://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 2008**

1. Download the P4VS Installer for Visual Studio 2008, `p4vs.exe`, from the Perforce downloads site.
2. Launch the installer.
3. On the **Select Features** page, select whether to install P4VS alone or with P4, the Perforce Command-Line Client.
4. Accept or change the installation directory.
5. On the **Client Configuration** page, enter the following information:
   - Hostname and port number for your primary Perforce service connection, using the following syntax: `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`

- Default text-editing application.
- User name for your primary Perforce service connection.

These settings determine the default Perforce connection the first time you open P4VS. You can override them once you start using P4VS.

On the Ready to Start page, click Start to begin the installation.

The installer screen displays an Installation Complete message. Click Finish to exit the installer.

**Installing P4VS in Visual Studio 2010**

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.

**Installing P4VS in Visual Studio 2012**

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.

   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 www.perforce.com/perforce/doc.current/user/p4vs_p4sc_migration.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 14
- “Perforce - Data Retrieval” on page 15
- “Perforce - Diff/Merge” on page 16
- “Perforce - General” on page 16
- “Perforce - Ignoring Files” on page 18
- “Perforce - Logging” on page 19

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 *Perforce User’s 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:

• **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.
Perforce - Diff/Merge

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

1. **P4Merge**: Perforce’s companion diff tool.
2. **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:

1. **P4Merge**: Perforce’s companion merge tool.
2. **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.

• **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 loading solutions with a large number of projects or projects with a large number of files.

**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 “Excluding Files from Perforce Control” on page 38.

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.

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 21.

For more information about Perforce environment variables, see the Perforce Command Reference and “Setting Perforce preferences” on page 14.

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 14.

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 Perforce Command Reference and “Setting Perforce preferences” on page 14.

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 “Connecting to Perforce services” on page 19.
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:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workspace</td>
<td>Workspace name. Defaults from the New Workspace dialog.</td>
</tr>
<tr>
<td>Owner</td>
<td>The user who created the specification. Defaults to you when you create a new workspace.</td>
</tr>
<tr>
<td>Host</td>
<td>(optional) The computer where the workspace resides. To enable the workspace to be used from any machine, leave this field blank.</td>
</tr>
<tr>
<td>Submit options</td>
<td>Configures what happens when users submit files.</td>
</tr>
</tbody>
</table>
### Line endings

The line-end convention used for storing text files on the workspace computer:

- **Local**: Uses the workspace platform default
- **Unix**: LF
- **Mac**: CR
- **Win**: 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.
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 19.

### 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 19.

### Stream workspaces

If you work with streams, P4VS uses workspaces differently than it does with classic depots. For more information, see “About streams” on page 47.
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 26.
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 19
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 “Syntactic view specification” on page 26.

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’s Guide.
Chapter 2  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 “Adding files to the depot” on page 27.

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.
   - When you click **OK** (and after you run any wizards), the **Open Connection** dialog appears.
   - 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.
2. In Solution Explorer, select the project or solution. Follow the procedure described in “Adding files to the depot” on page 27.

**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 “Setting Perforce preferences” on page 14.

For more information, see “Checking in files and working with changelists” on page 30.

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.
Chapter 2: Managing Files

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 “This chapter provides an overview of P4VS, the Perforce Plugin for Visual Studio, as well as instructions for installing and setting it up.” on page 9 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 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 “Retrieving files from the depot” on page 29.
Chapter 2: Managing Files

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 “Checking in files” on page 31.

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 “Using jobs (defect tracking)” on page 63.
   - Perform a diff on a file pending submission by context-clicking the file and selecting Diff Against Have Revision. For more information, see “Comparing files using diff” on page 40.

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 “Comparing files using diff” on page 40.

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.
Chapter 2: Managing Files

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 14.

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 on terminology: 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.”

**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.
Chapter 2: Managing Files

- **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” on page 38.
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 Lists 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 Lists 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:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>foo.txt</td>
<td>Ignore files called foo.txt</td>
</tr>
<tr>
<td>*.exe</td>
<td>Ignore all executables</td>
</tr>
<tr>
<td>!bar.exe</td>
<td>Exclude bar.exe from being ignored</td>
</tr>
</tbody>
</table>

While you can set your local `P4_IGNORE` environment variable and add Ignore Lists manually, P4VS provides preferences and context menu options to simplify the process of adding and editing Ignore Lists.

### 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 `P4_IGNORE` environment variable and used for all of your Ignore Lists. For more information about setting Ignore List preferences, see “Setting Perforce preferences” on page 14.

### 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.

In Visual Studio 2012, P4VS denotes an ignored file with a gray circle glyph next to the file icon. In Visual Studio 2008 and 2010, P4VS denotes an ignored file by removing the blue question mark glyph: 🕵.

### 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 Lists 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 14.

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 *Perforce 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 30.

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 30.

**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 14.

---

**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 differencing 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**...
Chapter 3  Working with Streams

This chapter discusses how to work with streams using P4VS.

About streams

Perforce streams are “branches with brains,” a containerized approach to managing bodies of related files such as codelines. Streams associate these bodies of related files with rules that define how you can work with those files, including how you can move changes between them. Most notably, streams are defined hierarchically using the mainline model, and Perforce generates the views for workspaces that are associated with a stream based on strictly inherited rules. Among the advantages of this approach are (1) change is propagated in a controlled way through the hierarchy that you define, and (2) you can compose the contents of a stream by defining its view, thereby providing all users who work in the stream with a consistent view of its contents.

Mainline model

The mainline model of software configuration management defines a structure based on the stability of the contents of the stream, from softest (unstable or experimental) to firmest (high-quality, releasable). Typically, the mainline is required to be fairly stable (for example, code must build), development codelines less stable, and release codelines the most stable.

Stream types

Streams codify the soft-to-firm model of stability in its standard stream types:

- **Mainline:** 🌊 A stream with no parent. Expects merging down from more stable child streams. Expects copying up from less stable child streams. Used as the stable trunk of a stream system.

- **Release:** 🔝 A stream that is more stable than its parent. Expects merging down from more stable child streams. Does not expect copying up from its parent stream. Useful for ongoing stabilization, bug fixing and release maintenance.

- **Development:** 🌠 A stream that is less stable than its parent. Expects merging down from its parent stream. Expects copying up from its less stable child streams. Does not expect to have more stable child streams. Useful for long-term projects and major new features.

There are two additional stream types with special characteristics:
Chapter 3: Working with Streams

- **Task** streams are lightweight, short-term branches that you can use for work that affects a small portion of a full project branch. Task streams enable you to work privately, switch contexts quickly, and reduce the amount of metadata managed by the Perforce service.

- **Virtual** streams provide users with the ability to restrict the workspace view of a real stream. Virtual streams act as a filter. They are useful when you want to:
  - Submit changes directly to a stream but do not want to sync all of the files in the stream view to your workspace.
  - Develop against the same stream but switch imported libraries, such as when you develop across multiple platforms.

**Stream hierarchy**

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.

The Stream graph displays soft streams below the mainline and firm streams above it. For example, the following diagram shows a typical software development structure, consisting of a stable mainline, with experimental streams below it and release-quality streams above:

![Stream Graph Diagram](image)

In the stream graph, the most stable streams are always at the top and the least stable at the bottom. The stream graph also displays the parent-child relationships that control the propagation of change between codelines. All streams are children of the mainline, and all children can be parents of yet more children of the same type, enabling specific development and release codelines while ensuring that changes are propagated properly between codelines. The stream graph represents those paths of change propagation using arrow connectors.
Chapter 3: Working with Streams

The Streams tool window shows streams within a parent-child hierarchy, with more stable release streams and less stable development streams both appearing below the mainline stream. Stream type icons indicate the role of the stream within the hierarchy.

Stream views

A stream is defined not just by its type, but also by its view, which specifies the files and folders that the stream contains and whether they can be edited, merged down, copied up, or branched to a new stream. Stream views define the files that you can work on in your workspace and what you can do with them. They also restrict the files that a child stream inherits from its parent.

For example, let’s say you have a mainline stream that includes the following directories of files:

```bash
//Acme/Main
  --apps
  --api
  --resources
  --docs
```

You want to branch to a development stream. You only need to work in the `apps` folder, but you need resources from the `api` and `resources` folders. You can create a development stream as a child of `main` that includes the files in the `apps` folder, excludes the files in the `docs` folder, and imports the contents of the `api` and `resources` folders so you can use them but cannot edit and submit any changes to the depot. You assign a stream view to the child stream that enacts these rules. Perforce generates the workspaces -- and enforces the submit and integration rules -- from the stream view. Any children branched from this new stream inherit those rules. You can assign a more restrictive stream view to subsequent children, but you cannot assign a less restrictive one. Child streams cannot branch more than their parent streams are willing to share.
Propagating change between streams

The primary principle of change propagation is *merge down, copy up*. The goal is to keep less stable streams up to date with their more stable parent or child, so that when change is propagated from a less stable stream to a more stable one, no work is overwritten. This approach keeps the resolve operation as simple as possible. Perforce’s merge/copy and resolve features are the means by which you propagate change between streams. When you use streams, all changes are propagated between parent and child streams. If you want to merge or copy between peer streams (for example, to merge another developer’s changes to your development stream), you must first reparent your stream (that is, edit the stream specification and change the Parent field). This approach is a key benefit of streams: the ability to configure the flow of change.

For more information, see “Merging down and copying up between streams” on page 55.

Stream depots

Streams are stored in a stream depot, which is displayed like this: 🌃. You cannot add a stream to a “classic” Perforce depot. You can add streams depots through P4Admin or the Perforce command-line client.

For more information about streams

For more information about these topics, 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’s 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’s 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:

![Stream Graph Example](image)

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 above 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.
Chapter 3: Working with Streams

Displaying stream status

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

![Stream: dev-2.1M2](stream_image)

- Merge changes
- Nothing to copy
- Work in this stream

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: ![workspace icon](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:

![Stream Graph Options]

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 right-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:

![Revision Graph Example]

**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 View 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.
  - Single revision: one revision at a time is displayed
  - Incremental diffs: two adjacent revisions are displayed, with changes highlighted
  - 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 change

**Line Ending**
- Specifies how line endings and whitespace are treated to determine differences
Chapter 4: Using Other Perforce Features

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.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Slider Appearance</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single revision</td>
<td><img src="image" alt="Single slider" /></td>
<td>Move it to the right to display the next file revision or left to display the previous revision.</td>
</tr>
<tr>
<td>incremental diffs</td>
<td><img src="image" alt="Incremental diffs" /></td>
<td>Move it to the right to display the next pair of file revisions, or left to display the previous pair of file revisions.</td>
</tr>
<tr>
<td>multiple revisions</td>
<td><img src="image" alt="Multiple revisions" /></td>
<td>Move the right and left halves separately, to control how many revisions are displayed.</td>
</tr>
</tbody>
</table>

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.

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 System Administrator’s Guide.
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 System Administrator’s Guide.
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 65.
   
   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 “Filtering Expressions” on page 65.
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 “Editing jobs” on page 64.

## Filtering Expressions

Valid job filtering expressions are as follows:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>word word word</code></td>
<td>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 &quot;and&quot;.</td>
<td>filter file mailbox</td>
</tr>
<tr>
<td>`word</td>
<td>word</td>
<td>word`</td>
</tr>
<tr>
<td><code>^word</code></td>
<td>Displays jobs that do not contain the specified word. The 'not' (^) operator cannot be used alone or with the 'or' operator (</td>
<td>), only with the 'and' operator (&amp; or space).</td>
</tr>
<tr>
<td><code>fieldname=value</code></td>
<td>Displays jobs that include the specified value in the specified field.</td>
<td>status=open owner=edk</td>
</tr>
<tr>
<td><code>^fieldname=value</code></td>
<td>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 (</td>
<td>), only with the 'and' operator (&amp; or space).</td>
</tr>
<tr>
<td><code>fieldname = value*</code></td>
<td>Displays jobs that contain the specified value in the specified field, including any combination of characters in the position of the asterisk wildcard.</td>
<td>owner=<em>ed</em></td>
</tr>
</tbody>
</table>

The 'not' (^) operator cannot be used alone or with the 'or' operator (|), only with the 'and' operator (& or space).
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’s 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’s 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’s 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.

Using P4Sandbox

You can use P4VS to work with P4Sandbox, which enables distributed, connection-independent versioning and branching within the context of a shared Perforce service. P4Sandbox provides a local service to work with or without an active connection to a shared Perforce service; you connect to a shared service only when you want to merge down changes or share your work.

You can use P4VS as your P4Sandbox client application, and you can create a local P4Sandbox by launching the P4Sandbox Configuration Wizard directly from P4VS.

Prerequisite

To launch the P4 Sandbox Configuration Wizard directly from P4VS, you must first install P4Sandbox into your Program Files/Perforce or Program Files (x86)/Perforce folder.

Creating a local P4Sandbox

1. Launch the P4Sandbox Configuration Wizard using one of the following options:
   - Click the Perforce Sandbox Configuration icon in your Start>All Programs>Perforce folder.
   - In P4VS, go to the Open Connection dialog and click Configure Sandbox.
2. Complete the wizard pages:

<table>
<thead>
<tr>
<th>Page</th>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to the Sandbox</td>
<td>Create a new or edit an existing P4Sandbox.</td>
<td>This page does not appear when you launch the wizard directly from P4VS, although you can go back to it.</td>
</tr>
<tr>
<td>Setup Wizard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specify Central Server</td>
<td>Specify connection to the shared service or a local P4Sandbox only.</td>
<td>This page does not appear when you launch the wizard directly from P4VS, although you can go back to it. The wizard populates your existing P4VS connections settings by default, although you can change them. The connection setting you specify here determines whether the local P4Sandbox is a connected or standalone implementation. This selection determines the pages that the Configuration Wizard displays next.</td>
</tr>
<tr>
<td>Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replication Settings</td>
<td>Define pulling interval and select files.</td>
<td>When you launch the wizard from P4VS, this page comes up first, populated with your current P4VS connection settings. Choose the files to copy from the shared Perforce service to the local P4Sandbox. These files occupy space on your hard disk and require time to transfer across the network. Minimize the number of folders you copy to minimize disk space and increase performance. Note that if you do not define a pulling interval, you must manually perform this action. P4Sandbox prompts you to pull changes, as needed, before allowing you to push changes.</td>
</tr>
</tbody>
</table>
3. When the configuration wizard finishes creating the P4Sandbox, P4VS does one of the following:

   - (Initial P4Sandbox launch only) Opens a dialog indicating that P4VS has launched a P4Sandbox connection window.
     
     Select **Don't show again** to prevent the dialog from popping up the next time you create a P4Sandbox.
     
     - Opens a local P4Sandbox connection window in P4VS.

For more information, see the *Perforce Sandbox User’s Guide*. 

<table>
<thead>
<tr>
<th>Page</th>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify Local Sandbox Server and Workspace Settings</td>
<td>Review connection settings for a local P4Sandbox.</td>
<td>Specify the <strong>Server</strong>, <strong>Port</strong>, <strong>User</strong>, <strong>Workspace Name</strong>, <strong>Workspace Root</strong>, and <strong>Charset</strong> (character set) for <strong>Unicode Servers</strong> fields. You cannot specify the same port number for use by multiple P4Sandboxes. When you attempt to re-use an existing port number, the following message appears: The specified port number is already in use by another P4Sandbox.</td>
</tr>
<tr>
<td>Settings Summary</td>
<td>Review P4Sandbox settings before finalizing. Shows settings and number of files to be replicated.</td>
<td>Review and click <strong>Finish</strong> to complete the configuration.</td>
</tr>
</tbody>
</table>