



**Basic Commands**

**Connecting to the Server**

|   |  |
|---|--|
| <code>p4 info</code>  | Displays information about the Perforce server and client configuration.                           |
| <code>p4 login</code>   | Authenticates the user with the Perforce server.   |
| <code>p4 set</code><br><code>P4PORT=</code><br><code>&lt;server:port&gt;</code><br><code>t&gt;</code> | Sets the Perforce server address.<br>Example: <code>p4 set P4PORT=perforce.example.com:1666</code> |
| <code>p4 set</code><br><code>P4USER=</code><br><code>&lt;username&gt;</code>                          | Sets the Perforce username. Example: <code>p4 set P4USER=johndoe</code>                            |
| <code>p4 set</code><br><code>P4CLIENT=</code><br><code>&lt;workspace&gt;</code><br><code>&gt;</code>  | Sets the Perforce workspace (client).<br>Example: <code>p4 set P4CLIENT=johndoe_dev</code>         |

**Workspace Management**

|   |   |
|---|---|
| <code>p4 client</code>                                    | Opens the client specification form for editing. Allows you to configure the workspace. |
| <code>p4 clients</code>                                   | Lists all client workspaces known to the Perforce server.                               |
| <code>p4 sync</code>                                      | Synchronizes the workspace with the latest version of files from the depot.             |
| <code>p4 sync &lt;file(s)&gt;</code><br><code>&gt;</code> | Synchronizes specific files to the latest version.                                      |
| <code>p4 sync @label</code>                               | Synchronizes the workspace to a specific label.   |
| <code>p4 sync #have</code>                                | Synchronizes the workspace to the revisions you had previously synced.                  |

**Adding and Deleting Files**

|   |  |
|---|--|
| <code>p4 add &lt;file(s)&gt;</code><br><code>&gt;</code>    | Marks new files for addition to the depot.   |
| <code>p4 delete &lt;file(s)&gt;</code><br><code>&gt;</code> | Marks files for deletion from the depot.   |
| <code>p4 reconcile</code>                                   | Detects files that have been added, deleted, or modified outside of Perforce and prepares them for submission. |

**Change Management**

**Making Changes**

|  |   |
|--|---|
| <code>p4 edit &lt;file(s)&gt;</code><br><code>&gt;</code>    | Opens files for editing. Required before modifying a file under Perforce control.   |
| <code>p4 revert &lt;file(s)&gt;</code><br><code>&gt;</code>  | Discards changes made to opened files, reverting them to the last synced version.   |
| <code>p4 diff &lt;file(s)&gt;</code><br><code>&gt;</code>    | Displays the differences between the workspace version and the depot version of a file.   |
| <code>p4 resolve &lt;file(s)&gt;</code><br><code>&gt;</code> | Resolves conflicts between the workspace version and the depot version of a file after an update or merge. Important to do before submitting. |

**Submitting Changes**

|   |  |
|---|--|
| <code>p4 change</code>  | Opens a new changelist for grouping related changes.                             |
| <code>p4 submit</code>  | Submits the opened files in the current or specified changelist to the depot.    |
| <code>p4 submit -d " &lt;description&gt; "</code>                 | Submits the changes with a description from command line.                        |
| <code>p4 changes</code>   | Lists recent changelists.  |
| <code>p4 describe &lt;changelist#&gt;</code><br><code>&gt;</code> | Describes a specific changelist, showing the files included and the description. |

**Branching and Merging**

|   |  |
|---|--|
| <code>p4 integrate &lt;fromFile&gt;</code><br><code>&lt;toFile&gt;</code> | Prepares files for merging changes from one branch to another.                 |
| <code>p4 merge &lt;fromFile&gt;</code><br><code>&lt;toFile&gt;</code>     | Merges changes from one branch to another (older command).                     |
| <code>p4 resolve</code>   | Resolves any conflicts after the integration/merge. Crucial before submitting! |
| <code>p4 branch</code>  | Creates a new branch specification.  |

**Advanced Operations**

**Labels and Attributes**

|   |  |
|---|--|
| <code>p4 label</code>                             | Creates or modifies a label specification.   |
| <code>p4 labels</code>                            | Lists available labels.                      |
| <code>p4 tag &lt;file(s)&gt;@&lt;label&gt;</code> | Attaches a label to specific file revisions. |
| <code>p4 attribute</code>                         | Sets or retrieves file attributes.           |

**Streams**

|  |   |
|--|---|
| <code>p4 streams</code>                        | Lists available streams.                      |
| <code>p4 stream</code>                         | Creates or modifies a stream specification.   |
| <code>p4 switch &lt;stream&gt;</code>          | Switches the workspace to a different stream. |
| <code>p4 merge</code> and <code>p4 copy</code> | Used for propagating changes between streams. |

**User and Group Management**

|                       |  |
|-----------------------|--|
| <code>p4 user</code>  | Creates or modifies a user specification (requires admin privileges).  |
| <code>p4 group</code> | Creates or modifies a group specification (requires admin privileges). |
| <code>p4 users</code> | Lists all users.   |

**Troubleshooting and Tips**

## Common Issues

**Conflicts during submit:** Always `p4 resolve` before submitting. Use `p4 diff` to understand the changes.

**Workspace out of date:** Ensure you `p4 sync` regularly to keep your workspace up-to-date.

**Permissions issues:** Check file permissions and ensure the Perforce user has appropriate access.

**Network connectivity:** Verify network connectivity to the Perforce server. Use `p4 info` to check the connection.

## Best Practices

**Atomic Changesets:** Group related changes into a single changelist for better tracking and management.

**Descriptive Comments:** Provide clear and concise descriptions for each changelist.

**Regular Synchronization:** Keep your workspace synchronized with the depot to avoid conflicts and integration issues.

**Branching Strategy:** Develop a clear branching strategy to manage different development streams and releases.

## Command Aliases (Example)

You can create aliases for frequently used commands for efficiency. For example, in bash:

```
alias psync='p4 sync'  
alias psubmit='p4 submit'
```