# Exercises - 1

## Basic File Operations

*You may complete these exercises using P4V or the command line client (p4.exe). The default is assumed to be P4V.*

*In the scenario for these exercise sets, you are a new Perforce user, starting to work on files in an existing Perforce database. Your system administrator has already set up your local environment variables so you can connect to your Perforce server and has assigned you the user name "bruno," and the client name "bruno_ws". You have a Perforce server running on your local machine, and you will be working on the 'jam' project data files.*

**Your objectives for this exercise:**

- To be able to perform fundamental operations required for manipulating files in your local workspace and submit your completed work to your Perforce depot.

1. Open a P4V connection, logging in to Perforce. Select No if you'd like to skip the Connection Setup Wizard and go to the Open Connection dialog to log into Perforce. The password for bruno (if your are prompted) is **brunopass**.

2. Show your connection information:

   - What is the port address of your current Perforce server?
   - What is your effective Perforce user name?
   - What is the name of your current Perforce client workspace?

3. Fill your workspace with files from the depot ("get latest" or "sync" them).

4. In P4V select a file under //depot/Jam/MAIN/src and right click > Show in Explorer (towards the bottom of the options) to launch Windows Explorer in the appropriate directory.

5. In the c:\bruno_ws\Jam\MAIN\src (equivalent to //depot/Jam/MAIN/src) directory, check out jam.c and jam.h (CLI - open them for editing). What has changed to the attributes of the files (use Windows Explorer, Right Click > Properties to check and compare with other files that are not checked out).

6. Mark variable.c and variable.h for deletion (open for delete). What's happened to the local copy of the files (use Windows Explorer or the Workspace View Tab)?

7. Create a new file under c:\bruno_ws\Jam\MAIN\src using Windows Explorer or notepad, and mark it for add in P4V from Workspace View (P4Win/CLI - open them for add).

8. What command would remove jam.c from the list, and return it to its original state (undo the checkout)?

9. Submit your changelist (check in). What is the changelist number you have just submitted?

10. Check out jam.h (open for edit), and make a small change. Show how your workspace file differs from the depot revision of jam.h which you last got from the repository.

 

# Exercises - 2

## File Information

**Your objectives for this exercise:**

- To be able to retrieve information about files in the Perforce depot, including determining the current state of files and finding out what and when changes were made to files.

11. Sync (get revision) all the revisions of the files in //depot/Jam/MAIN/src to your client workspace in the state they were in as of changelist 165. How can you find out what a get latest (sync to head) would do **without doing it**?

12. Sync (get latest) all your files again.

13. How does the March 24, 2002 version of jambase.c differ from the May 1, 2002 version?

14. What files changed in the //depot/Jam/MAIN area during March 2002 (hint: think differences between start and end of the month)? Extra credit – what changelists occurred during that time? (You can “see” this in p4v, but actually there is a command line command “p4 changes” which will tell you directly with an appropriate revision range parameter)

# Exercises - 3

## Changelist Management

**Your objectives for this exercise:**

- To retrieve all or selected submitted changelists in order to report information about files in your depot
- To create a pending changelist so you can efficiently organize your work in your client workspace.

15. What files were affected by changelist 350 (and how do you see this in P4V)?

16. List only those changelists that affected the file “//depot/Jam/MAIN/src/command.h”.

17. Create a pending changelist or two, move some files into them and submit them as you wish (practice using pending changelists).

18. Practice shelving and unshelving a couple of files, experimenting with the various options.

 

# Exercises - 4

## Workspace Management

*Up until now you have been working on Perforce depot files using a workspace created for you by your Perforce Administrator*

*Perforce works on files. If you are specifying a directory in the following exercises, be sure to add "/..." at the end of the file pattern to specify all the files in that directory and its subdirectories, e.g., //depot/Jam/MAIN/....*

**Your objectives for this exercise:**

- To make changes to your client workspace specification
- To create a new client workspace specification for a different project.

19. Change your bruno_ws client workspace to set the "rmdir" option (from "normdir").

20. Change your bruno_ws client workspace view so that instead of mapping all files in the whole of //depot, the only files in your workspace are those under //depot/Jam/MAIN. You may find it easier to remove the existing view (of //depot/…) and then add in the specific view you want.

21. Sync (get latest) your workspace.

22. Modify your workspace view to add //depot/Talkhouse/main-dev files to the view, and yet exclude //depot/Talkhouse/main-dev/system files (and get latest/sync the workspace)

23. In this exercise we will create a new workspace bruno_projb and add some new files (any files) such that they will go into the repository in //depot/projb/MAIN/… This makes sure you understand how workspace views map from the local file system to Perforce as well as vice versa.

 

This will require you to:

- Create the new workspace “Bruno_projb”
- Set its root directory, for example: c:\bruno_projb
- Set the workspace view appropriately, e.g. using Text View set to:
  - //depot/projb/MAIN/… //Bruno_projb/… (Do not cut and paste this line, type it in the view.)
- Switch to using this new workspace (if this wasn’t done automatically when you saved it)
- Using Windows Explorer, create the workspace root directory, c:\bruno_probj
- Create some files in that directory (you can copy them from c:\bruno_ws or create new ones)
- Finally add the new files into Perforce from the P4V Workspace View tab – you may find it useful to Refresh (F5) the current view.
- Don’t forget to submit!

# Exercises - 5

# Handling File Conflicts

**Your objectives for this exercise:**

- To create a situation where you are editing files that are out of date (as you would be when someone else edits the same file you are working on and submits their work before you)
- To resolve your files before you submit your work to your depot.
- To back out your change after you submit your completed work, because the project lead has rejected your work.

24. For the following exercises, start by doing get revision (sync) of your “bruno_ws” workspace to **changelist** 350 (not **revision** 350). Some files will now be out of date.

25. Check out //depot/Jam/MAIN/src/RELNOTES and //depot/Jam/MAIN/src/patchlevel.h (if you are warned, then select Don’t Get Latest).

    - Edit patchlevel.h and add some text at the beginning of the file (You can use the ‘notepad’ program to edit the files.)

    - Edit RELNOTES and add some text. (For the most dramatic effect, change the line “PATCHLEVEL 0” to “PATCHLEVEL 2a” or similar.)

    -

26. Confirm that the only files you have in your pending changelist are patchlevel.h and RELNOTES. If you have any others opened, revert them.

27. Try and submit the changelist - in p4v you will not be able to – it will tell you to do a Get Latest, so cancel submit, select the files and do Get Latest.

 

28. In P4V, right-click files and resolve them (CLI run **p4 resolve** to start up the interactive dialog) and run the merge program. Make any appropriate changes and accept the merge result.

29. Submit your changelist.

# Exercises - 6

## Branching & Integration

*Perforce works on files. If you are specifying a directory in the following exercises,* ***be sure to add "/..."*** *at the end of the file pattern to specify all the files in that directory and its subdirectories,* ***e.g., //depot/Jam/MAIN/....***

**Your objectives for this exercise:**

- To create a personal branch/codeline, since the development work on the 'jam' project has reached a point where it is appropriate.

30. In P4V, right click //depot/Jam/MAIN and **Branch Files…** the files to //depot/Jam/<YOURNAME> (**p4 populate** on command line)**.** Use the auto-submit option.

31. Change your bruno_ws client workspace view so that the view maps everything under //depot/Jam/MAIN and //depot/Jam/<YOURNAME> (e.g. view contains just "//depot/Jam/MAIN/… //Bruno_ws/Jam/MAIN/…" and similarly for the other path) and do a get latest.

32. Checkout (open for edit) a couple of files in //depot/Jam/<YOURNAME>, make a couple of edits and submit your changelist.

 

33. How can you show what changes made to //depot/Jam/<YOURNAME> need to be integrated back into the //depot/Jam/MAIN branch? (Hint: think integrate preview)

34. Integrate all changes made to //depot/Jam/<YOURNAME> back into the //depot/Jam/MAIN branch using automatic resolve options as appropriate.

35. Integrate all changes made to //depot/Jam/MAIN to //depot/Jam/<YOURNAME> branch using safe automatic resolve options as appropriate.

# Exercises - 7

## Streams

These exercises take you through the basics of using the new streams feature.

36. Edit your Preferences > Streams options to reuse the current workspace and automatically update the workspace file when switching between streams.

37. Create a new streams depot called Jam (Tools > Administration and then File > New > Depot)

38. Create a stream “Main” of type mainline, located in depot Jam.

39. Right click the stream and “work in this stream” – so create a new Workspace (accept the defaults)

40. You will be prompted that the stream is empty – select option Branch files from depot and select //depot/Jam/MAIN as the source. Submit the files.

41. Right click Main stream and create a new stream called Dev which is of type Development

42. Change a file in Dev and Copy it to Main

43. Change a file in Main (and submit it) and notice that the icon changes to show it should be Merged to Dev (don't do this yet)

44. Change another file in Dev (and submit it) – notice the icons change. You should not be able to Copy to Main without a warning.

45. Merge from Main down to Dev. On the Filter menu – show the changelists needing to be merged down. Perform the merge.

46. Copy from Dev up to Main!

# Exercises - 8

## Labels

**Your objectives for this exercise are to tag files using two different methods:**

- Tag files under the //depot/Jam/MAIN/src directory that were current as of a changelist without syncing the files to your client workspace.
- Tag files in your client workspace based on the following criteria:
  - tag files current in the //depot/Jam/MAIN/src directory when a particular file was submitted, with a label
  - exclude the '*.c' and '*.h' files from the label.

44. Tag files under the //depot/Jam/MAIN/src directory with a label named "jam-NT-build" that were current as of changelist 392, regardless of which versions of the files are in your client workspace.

45. List the versions of the files you have just tagged with label jam-NT-build.

46. Sync your bruno_ws client workspace with the file versions that were current when changelist 395 was submitted.

47. Create a label specification called jam-2.2.1. Restrict the label's view to //depot/Jam/MAIN/src files.

48. Tag the files in your client workspace with the jam-2.2.1 label.

49. What version of //depot/Jam/MAIN/src/jam.h is tagged with the “jam-2.2.1” label? Is the version tagged the head revision?

50. Remove all the *.c and *.h files from the jam-2.2.1 label.

51. Sync your client workspace with the file versions in the “jam2.2.1” label.

Congratulations! You have completed the one day Perforce Training Course. We hope this course material has prepared you to use Perforce with confidence and ease.