

Changes to P4 Integrate in 2004.2

Jeff Anton

Senior Software Developer
Perforce Software, Inc.



Summary

- Indirect Integration all the time
- Bases for integration are real common ancestors
- Significant Performance improvement for Indirect integration
- Fixes to indirect integration crediting



Motivations

- Rename
- Indirect crediting – better
- Baseless integration issues
- Performance



Direct Integration

- Source revision would be the newest revision in the source revision range (except for previously ignored integrations)
- Base revision would be the revision of the source just prior to the oldest non-credited revision in the source revision range.
- Integrations would be considered 'baseless' if the source revision range started with an *add* or *branch* revision and the base revision would be before that *add* or *branch* revision.



'Cherry Picking'

- Source of integration involves a range limit usually a single change list
- Does not require a common ancestor base revision
- Can be costly in terms of disk space for many integration records and the time needed to consider them

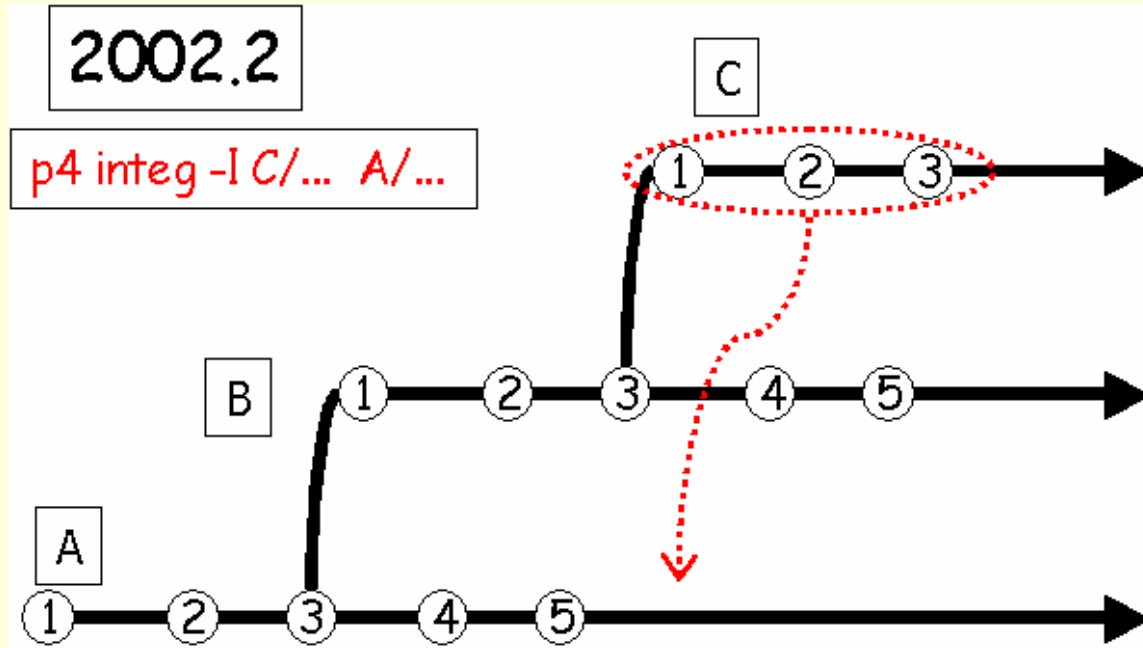


Crediting Defined

- A 'Crediting' integration is one in which we can know that the work of an edit on one file has been applied to another file.
- A 'Non-Crediting' integration means that although an integration has taken place, we can not know that the work of an edit has been applied to another file.



Baseless Indirect problems



Tech Note 065 — Old -I flag implied ‘baseless’ because it would not find a base on the source code line in this case. 2004.2 separated ‘indirect’ from ‘baseless.’ Old base was C1, now it is B1 or A3.

Rename vs. Integrate

➤ File Names vs. File Contents

- To integrate file content properly, merge base revisions will not be on the source code line after a file rename - Integrate must find the right base.
- When integrating between code lines - file renames should be propagated.

➤ How to tell a branch from a rename?



Performance

➤ Path Restrictions

- When searching, do not search past the source filename (we search from target to source) and stop before returning to target.

➤ New Breadth-First Searching

- Exhaust near branches before searching far branches. (Integrations between 'near' branches will be faster)



Indirect Combining Rules

- Integration records combine by overlapping source and target revision ranges.
- The 'kind' of integration (copy, merge etc.) is reduced to the weaker kind and also can be changed by partial overlapping integrations.



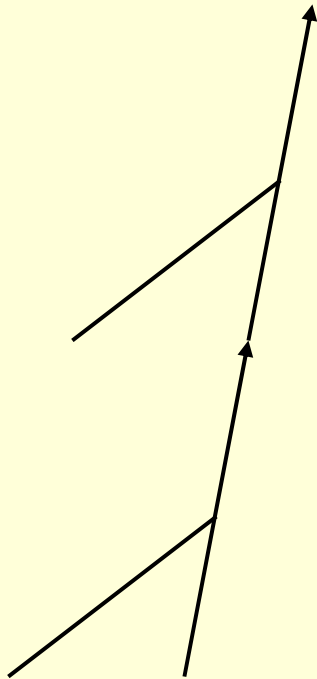
Indirect Combining (cont.)

➤ Examples

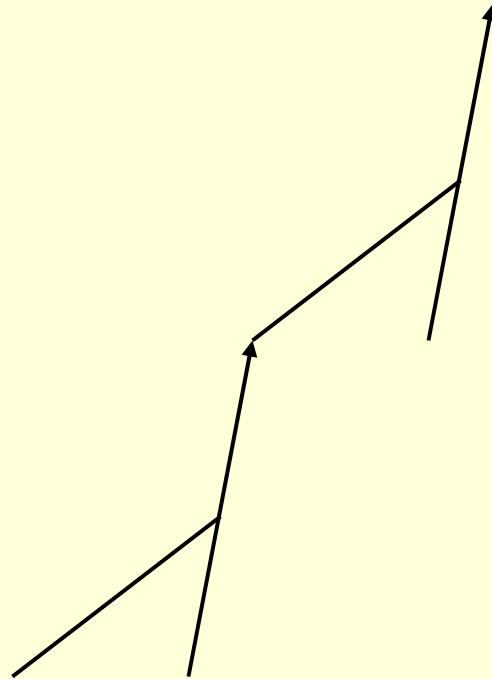
- A Merge combined with a Copy is a Merge
- A Branch combined with a Copy is a Copy
- An Edit combined with a Copy is an Edit
- A Copy to an intermediary which is then edited then that intermediary is then Copied is considered a Merge.



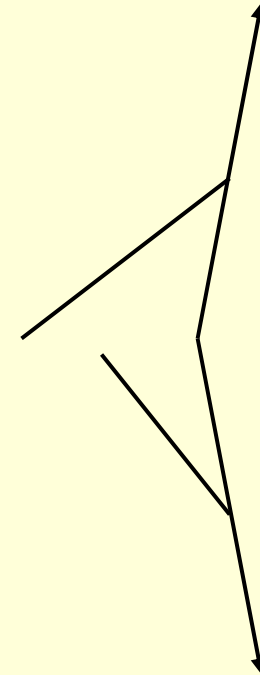
Indirect Combining (cont.)



Aligned



Demote



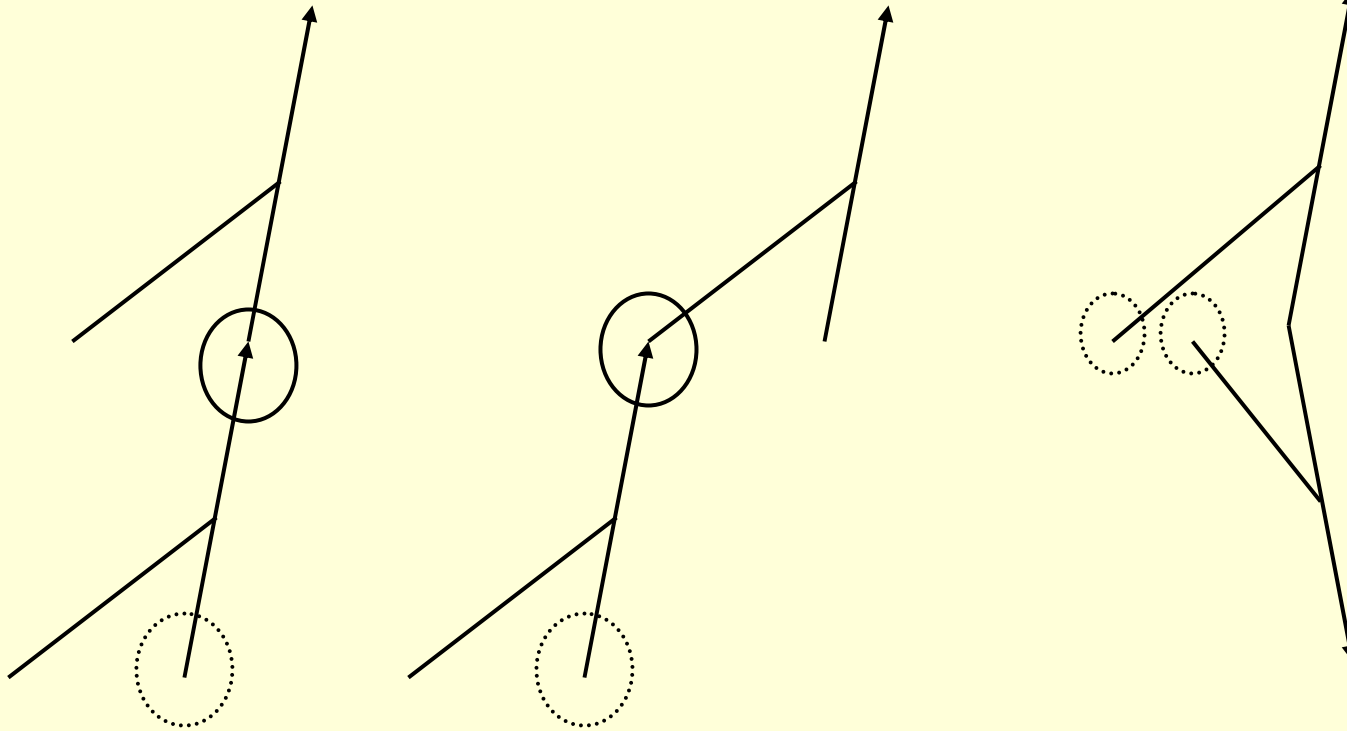
Demote

Base Finding

- The Base revision used to come from the last Source revision credited
- Still true, but the base revision comes from the combined integration record which gave the last credit
- Except, if there are no credits but some 'relationship' then the base is found on the earliest combined integration record



Base Finding (cont.)



Dirty Branching

- Opening a newly branched file for edit produces a 'Dirty Branch' because the newly branched file may have differences from the branch source file.
- Back integration of a dirty branch would not transfer the edit of the branch due to the base selection.

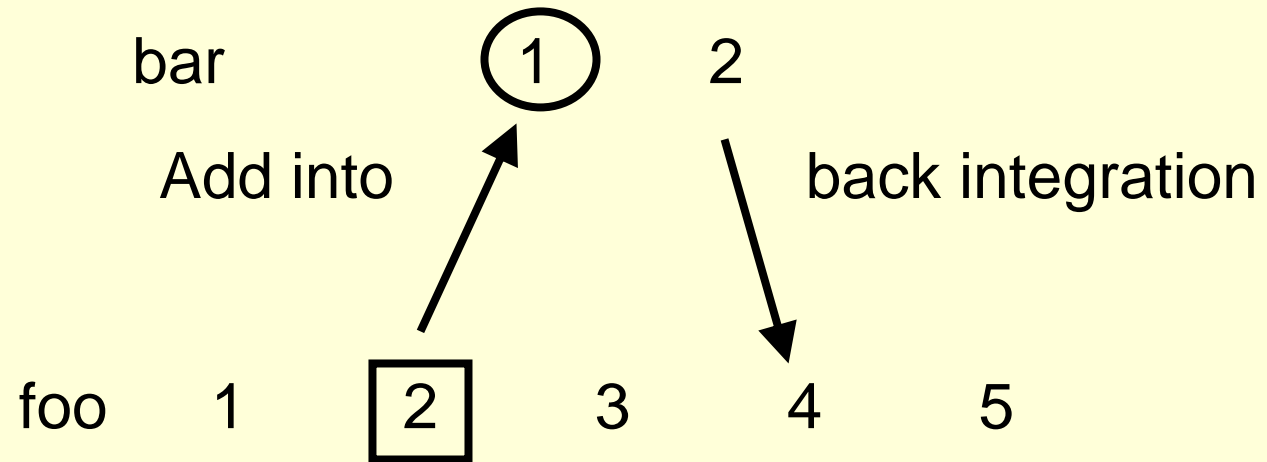


Dirty Branching (cont.)

- Diagram setup (create the dirty branch)
(With foo at revision #2)
 - p4 integrate foo bar
 - p4 edit bar (reopen for edit)
 - p4 submit
- Later
 - p4 integrate bar foo



Dirty Branching (cont.)



Pre-2004.2 back integration base was bar#1 now foo#2

Dirty Branch base change



Frequent Integrations

- Credit searching looks at many integration records
- Frequent integrations will increase the amount of work by '*p4 integrate*'
- Occasional *Copy* integrations give credit for all past revisions which reduces the work load of '*p4 integrate*'



Expensive Integrations

- Normally, an integration between branches will only affect a much smaller number of files in those branches
- Edits which touch all files (such as a company name or copyright change) will force all files to be integrated



Expensive Integrations (cont.)

- Breadth-First search is not a help if an integration is needed
- All revisions, direct and indirect, will have been looked at before determining that an integration is needed.
- Use undocumented '-1' flag to force direct integration only and 'cherry pick'



P4 Interchanges

- Undocumented command shares the crediting system with 'p4 integrate'
- Cherry picked integrations may be missed

Foo	...	2	3	4
Bar		3	4	5

A diagram showing a mapping between two rows of data. The top row is labeled 'Foo' and contains the values '...', '2', '3', and '4'. The bottom row is labeled 'Bar' and contains the values '3', '4', and '5'. A vertical arrow points from the '3' in the 'Foo' row down to the '4' in the 'Bar' row, indicating a specific interchange or relationship between these two values.