How to Merge

Charles Bailey, Bloomberg LP
Why do we merge?

- We’ve worked in parallel on the same thing
When do we merge?

- On “update” in centralised world
- When we pull, merge or rebase explicitly
When should we merge?

- When we want to incorporate changes from elsewhere
- Merge early; merge often?
Merge mechanics

- Merges happen between directory structures
- Add, remove, change contents, change metadata of directory entries
- Traditional merge activities happen on “change contents” on the same entry
2-way merge
apples
<<<<<<<<< HEAD
bananas
========
cherries
>>>>>>>>> master
dinosaur
apples
bananas
cherries
dinosaur
2-way merge

• Only the state of each side is shown
• No way to infer the changes
3-way merge

• Not a fundamentally different operation
• Merging with less information thrown away
• Ability to compose changes
apples

<<<<<<< HEAD
bananas

||||||| merged common ancestors
bananas
cherries

======

cherries

>>>>>> master
dinosaur
apples
<<<<<<<<< HEAD
bananas
| | | | | | | | merged common ancestors
bananas
cherries
========
cherries
>>>>>>>>> master
dinosaur
apples
<<<<<<<<< HEAD
bananas
| | | | | | merged common ancestors
bananas
cherries
=======
cherries
>>>>>>>>> master
dinosaur
apples
<<<<<<<<< HEAD
bananas
| | | | | | merged common ancestors
bananas
cherries
========
cherries
>>>>>>> master
dinosaur
apples
dinosaur
Manual merge

- Conflict markers
- Git config: `merge.conflictStyle = diff3`
Live Demo!
Lessons

• Don’t try to merge generated files

• Take the easy option
Assisted Merge

- Learn the tool; use the tool
And now, with a tool...
Further lessons

• Take the easy option (always!)
Pitfalls

• Leave enough time

• Test that the result of the merge “works”

• Test that the changes of both sides still have the intended effects
Some Git specifics
Temporary Reversions
revert A

m1

B

x

A

base
Fixing bad merges
rerere

- Records previous merge resolutions
- `mkdir .git/rr-cache` or `git config rerere.enabled`
Questions... and war stories