

TOOLS



Three-Way Merging: A Look Under the Hood

By Pablo Santos, December 24, 2013

[Post a Comment](#)

Automating three-way code merges requires considerable sophistication from the version control system.

For several years, Pablo was a blogger for Dr. Dobb's covering Agile technologies and best practices. With this article, he returns as a regular blogger. His next post will appear in the left panel of our home page, along with our regular bloggers. —Ed.

When I run a demo of our [source code merge program](#), a developer occasionally raises a hand and asks, "What do you mean by 'three versions' of the file being merged?" To answer this and explain how three-way merges work and why they are important, let's start by taking a look at the traditional two-way merge.

What Is a Two-Way Merge?

Suppose we're modifying the same file concurrently. You go and make some changes to the file and then I make some more changes.

At some point in time, someone looks at the two copies of the file and they see something like Figure 1:



Figure 1.

This third person looking at the files sees there's a difference on line 30 but:

- How can he tell whether you modified line 30 or if I modified it?
- What if we both modified the line? How can he tell?

He can't.

He will have to call both of us and trust our memories to find out who modified what. And yes, fortunately we programmers never forget, right? :-)

This "third person" is likely to actually be the version control system trying to do a simple two-way merge, which just compares two versions of a file and tries to merge them. But here, the VCS will require user intervention because it won't be able to figure out what to do.

It is better to avoid concurrent modifications on a file if you have to rely on two-way merge because it will be a slow manual process. Imagine 300 files requiring a simple merge...it would take ages to complete!

The Three-Way Merge

Let's forget for a second about version control, and go back to this "third person" looking into the two files we modified: How can he figure out what happened by himself?

He can look into the original version of the file we both used as starting point (Figure 2):

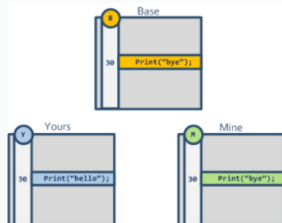


Figure 2.

Then, looking at how the file was originally ("base common ancestor" or simply "base"), he can figure out what to do.

Based on Figure 2, only one developer actually changed line 30, so the conflict can be manually resolved: Just keep "Yours" as the solution, so it will say: `Print("hello");`

This is how three-way merge helps: It turns a manual conflict into an automatic resolution. Part of the magic here relies on the VCS locating the original version of the file. This original version is better known as the "nearest common ancestor." The VCS then passes the common ancestor and the two contributors to the three-way merge tool that will use all three to calculate the result.

Using two-way merge only, the lines modified by two developers will require manual intervention, while everything else will be automatically merged. With three-way merge, it is possible to run a painless merge involving hundreds of files.

A Manual Merge Scenario

Let's now check a slightly more complex case, as shown in Figure 3:

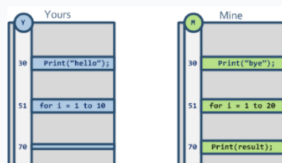
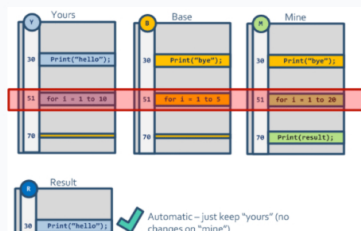


Figure 3.

Comparing the two files side by side, we can see that there are three lines with differences:

- Line 30: with the same conflict we had before.
- Line 51: with a for loop being modified.
- Line 70: where we don't know whether "yours" removed some code or "mine" added it.

Let's now look at the common ancestor to be able to properly solve the conflicts (Figure 4):



Tools Recent Articles

- Dr. Dobb's Archive
- Jolt Awards 2015: Coding Tools
- Thriving Among the APIs
- Building Node.js Projects in Visual Studio
- Building Portable Games in C++

Most Popular

Stories [Blogs](#)

- RESTful Web Services: A Tutorial
- Developer Reading List: The Must-Have Books for JavaScript
- iOS Data Storage: Core Data vs. SQLite
- Building OUI Applications in PowerShell
- MongoDB with C#: Deep Dive

Upcoming Events

Live Events [WebCasts](#)

- [Video] Shaping the Future of IT: CIO Lightning Series - Interop 19 - Interop 2019
- [Video] An (Irrational) Approach to Interoperability - Interop19 - InteropTX 2018
- Get Insights: Cisco vs Microsoft & More at EC20 Orlando - Enterprise Connect 2020
- Cisco & Amazon Web Services to Keynote at Enterprise Connect 2020 - Enterprise Connect 2020

Featured Reports

What's this?

- Assessing Cybersecurity Risk in Today's Enterprise
- The Definitive Guide to Managed Detection and Response (MDR)
- 2019 SANS Incident Response Survey Report
- 2019 Threat Hunting Report
- Getting Started With Emerging Technologies

[More >>](#)

Featured Whitepapers

What's this?

- Getting Started With Emerging Technologies
- [Infographic] Are You Maximizing Value of the Cloud?
- 2019 State of DevOps
- Shed Light on the Darkspace with East-West Traffic Monitoring
- 2019 State of the Internet / Security: Phishing - Baiting the Hook

[More >>](#)

Most Recent Premium Content

Digital Issues

2014

- Dr. Dobb's Journal
- November - Mobile Development
- August - Web Development
- May - Testing
- February - Languages

Dr. Dobb's Tech Digest

- DevOps
- Open Source
- Windows and .NET programming
- The Design of Messaging Middleware and 10 Tips from Tech Writers
- Parallel Array Operations in Java 8 and Android on x86: Java Native Interface and the Android Native Development Kit

2013

- January - Mobile Development
- February - Parallel Programming
- March - Windows Programming
- April - Programming Languages
- May - Web Development
- June - Database Development
- July - Testing
- August - Debugging and Defect Management
- September - Version Control
- October - DevOps
- November - Really Big Data
- December - Design

2012

- January - C & C++
- February - Parallel Programming
- March - Microsoft Technologies
- April - Mobile Development
- May - Database Programming
- June - Web Development
- July - Security
- August - ALM & Development Tools
- September - Cloud & Web Development
- October - JVM Languages
- November - Testing
- December - DevOps



Figure 4.

- The conflict on line 30 can be automatically solved and the "yours" (source contributor) will be kept as result because only one contributor modified.
- The conflict on line 70 can also be automatically solved to "mine" (destination contributor) because it is clear now that the line has been added and it wasn't there before.
- The conflict on line 51 needs manual resolution; You need to decide whether you want to keep one of the contributors, the other, or even modify it manually.

And this is basically how three-way merge works.

12 Next

Related Reading

News | Commentary

Did Barcode Reading Just Get Interesting?
SmartBear Supports Selenium WebDriver
Jelastic Docker Integration For Orchestrated Delivery
Parasoft DevTest Shifts To Continuous

[More News >>](#)

Slideshow | Video

2014 Developer Salary Survey
Jolt Awards: Mobile Development Tools
Developer Reading List
Jolt Awards: Coding Tools

[More Slideshows >>](#)

Most Popular

Hadoop: Writing and Running Your First Project
iOS Data Storage: Core Data vs. SQLite
Unit Testing with Python
What Exactly is DevOps?

[More Popular >>](#)

More Insights

White Papers

Getting Started With Emerging Technologies
2019 State of DevOps

[More >>](#)

Reports

Rethinking Enterprise Data Defense
The Definitive Guide to Managed Detection and Response (MDR)

[More >>](#)

Webcasts

How to Prevent 81% of Phishing Attacks from Sailing Right into Your Inbox with DMARC
Cloud Sprawl: It's Worse Than You Think

[More >>](#)

INFO-LINK

[Login or Register to Comment](#)

DISCOVER MORE FROM INFORMA TECH

InformationWeek
Dark Reading
Network Computing

Interop
Data Center Knowledge
IT Pro Today

WORKING WITH US

Contact Us
About Us
Advertise

FOLLOW DR. DOBB'S ON SOCIAL



[Home](#) [Cookie Policy](#) [Privacy](#) [Terms](#)

Copyright © 2019 Informa PLC. Informa PLC is registered in England and Wales with company number 8860726 whose registered and head office is 5 Howick Place, London, SW1P 1WG.