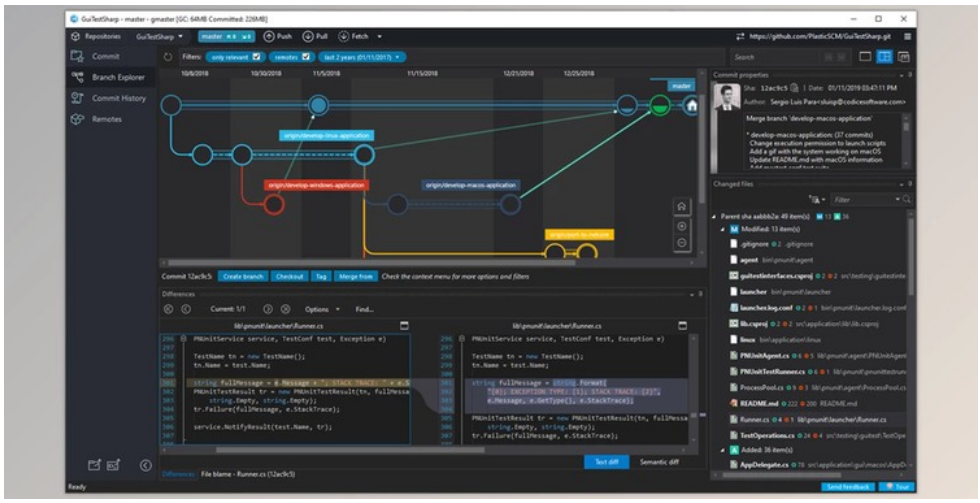
GMaster v1.0.663 (14 Oct 2020)

2025-01-13 22:05:08



Gmaster requires a different approach in contrast with the other Git customers: it's visually rich, comprises excellent diff and merge tools and exploratory technologies.

It turns you into a diff and merges master by parsing the code. It helps you know Git repos with unprecedented visualization. Most tools are only a wrapper around the command line. Instead, we take a visual approach: Branch Explorer as a thorough repo visualization, incorporated side-by-side diff with Semantic capacities, appropriate 3-way merge, and much more.

GMaster Great Features:

Most tools are only a wrapper around the command line.
Instead, we now take a visual approach: Branch Explorer as a thorough repo visualization, incorporated side-by-side diff with Semantic capacities, appropriate 3-way unite, and more.

Gmaster involves a wealthy built-in side-by-side diff, a super-strong 3-way mix, and both are bombarded with semantic abilities to comprehend C#, C++, Java, C, and much more. It's a complete solution from the box.

Why diff and merge code just like it was 1972?
We might not have flying cars yet, but we could do much better than plain text diff.
We emphasize C#, Java, VB.net, C, C++ code (and the third-party contributed parsers) and calculate diffs and merges.
Moved methods? Refactored code? We have you covered.

Gmaster evolves quickly. But we wanted to bring the many magnificent features: the Branch Explorer, Semantic Diff & Merge - the sort of things which make it special.
The strategy would be to continue with attributes like managing hunks, rebasing, and much more.

Tools form what we can really attain. And we create gmaster so we could get more out of Git;-)

Know your Inbox visually

Utilize the Branch Explorer to browse your repository left to right, as you'd do on a blackboard. Checkout branches, diff, and merge out of here.
And identify the commits with more modifications in a glance also.

Diff side-by-side

The easiest and quickest way to comprehend code changes. And, it comprises Xdiff to monitor code fragments that were moved.

Track refactored code

Here comes the ability of Semantic - we parse the code, then compute diffs based on this. It's possible to move a technique, alter this, and semantic diff still understands it's the same.
It's priceless to diff refactored code.

Blend to Be an expert... with no pain

Merge can be a lot simpler.
We split the"commit" in the"mix"; therefore, you receive a complete list of battles and alternatives to fix them rather than a fuzzy interface where we perpetrate and unite resolution ends up being the same thing.

Instantly merge refactored code.

Refactors are great for maintaining code quality, but they're a nightmare to combine.

所有软件资源 100%安全 已检测 无病毒 无篡改
去预览

标签

Tools Applications

Not anymore with Semantic Merge. Move approaches around, alter them, whilst somebody else repairs. The mix will be automatic.

Diff images

It's possible to compare pictures to assess everything changed: side-by-side, mix, swipe, and diff properties.

Only devote

By default, commit adjustments are just one click operation. You diff your adjustments, and you submit. Period. It's not necessary to manage the intermediate indicator or anything. It is as straightforward as it gets.

版本列表

GMaster v1.0.663 (14 Oct 2020)

苏ICP备19032038号 Powered by getShopSys