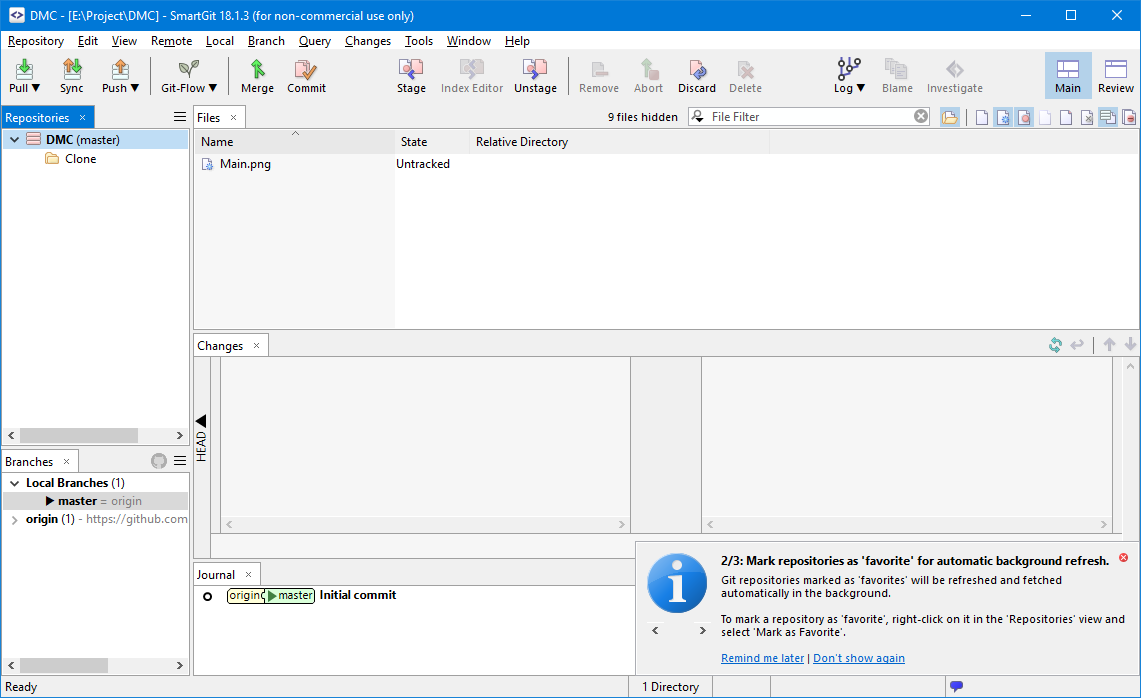
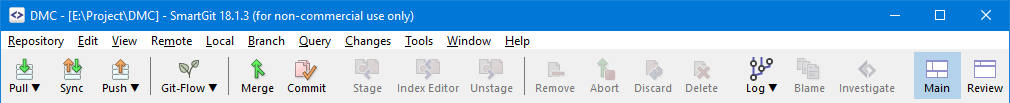
# Màn hình chính

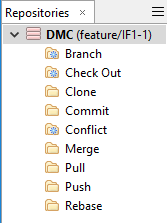


Màn hình chính của SmartGit gồm các phần sau:

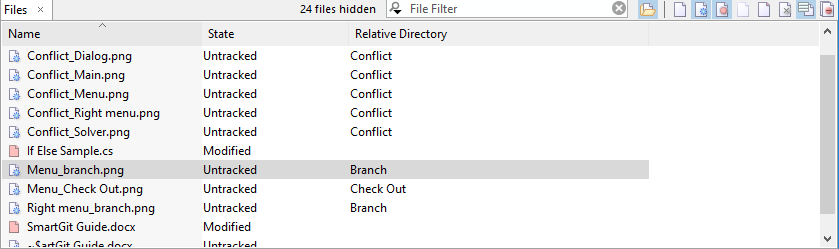
1. Menu bar và toolbar: chứa các tác vụ của Git. 2 nút phía rìa bên phải (**Main** và **Review**) có tác dụng thay đổi kích thước các view trên màn hình chính.



1. View Repositories: hiển thị cấu trúc của repo hiện tại dưới dạng cây

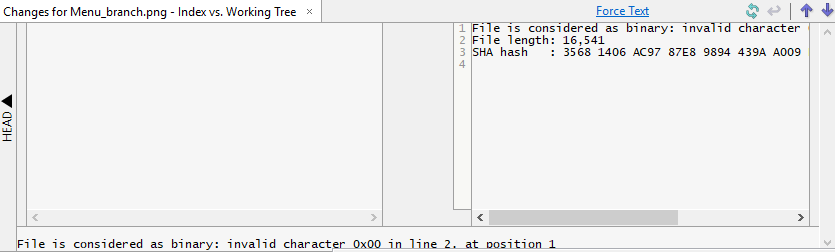


1. View Files: hiển thị các file trong folder và subfolder đang trỏ đến ở view *Repositories*.

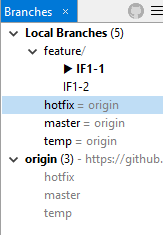


SmartGit cung cấp một số chức năng sau cùng với view:

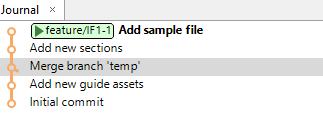
1. Lọc file theo tên: Thông qua input bux *File Filter*.
2. Ẩn hoặc hiện file theo trạng thái trên repo: hang nút góc trên bên phải có rất nhiều tùy chọn ẩn hiện như hiện thị các file được theo dõi bởi git, các file bị ignore, các file trong subfolder…
3. View Changes: hiển thị thay đổi của file đang chọn trong view Files so với commit đang trỏ đến; nếu file không phải dạng text sẽ chỉ có thông tin vắt tắt về binary.



1. View Branches: hiển thị các nhánh trong git hiện hữu ở local và remote, đồng thời hiển thị sự khác biệt giữa local và remote về commit thông qua mã màu.
   1. Branch màu đậm hơn là branch đang check out.
   2. Các branch trùng khớp sẽ có dòng màu xám = origin bên cạnh tên branch trong local (không có nghĩa là branch đó chưa được đẩy lên remote).
   3. Các branch có commit chưa push sẽ có số commit màu cam bên cạnh cùng dấu lớn hơn, ví dụ như 6 > tức là local có 6 commit không có trên remote.
   4. Các branch chưa được cập nhật sẽ có số commit còn thiếu so với remote màu xanh bên cạnh cùng dấu bé hơn, ví dụ như 6 > < 2 tức là local có 6 commit không có trên remote và remote có 2 commit không có trên local.

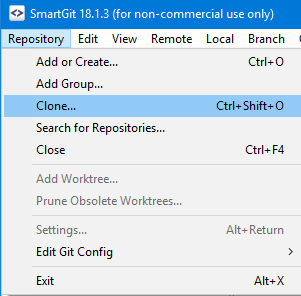


1. View Journal: hiển thị tất cả các commit trên branch đang chọn trong view *Branches*.



# Clone: Sao chép nội dung 1 repo git về máy

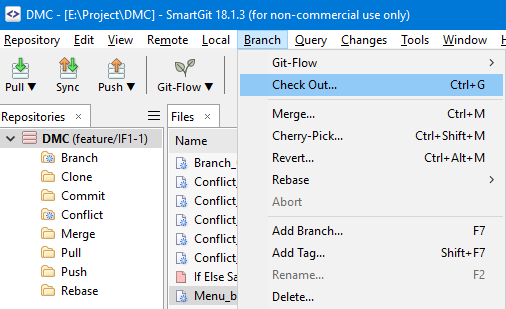
1. Trên thanh menu chọn **Repository** > **Clone…**



1. Cửa sổ *Clone* xuất hiện. Ở bước đầu tiên, *Repository*, chọn **Remote Git or SVN repository** và nhập đường dẫn tới file .git của repo git cần lấy về.
   1. Nếu SmartGit yêu cầu tạo **Master Password**, chọn **Don’t use a master password** để bỏ qua hoặc **Use the following master password** nếu muốn đặt. Master password nếu được đặt sẽ phải nhập 1 lần duy nhất mỗi khi khởi động SmartGit và truy cập vào repo bất kỳ.
   2. Nếu SmartGit yêu cầu đăng nhập vào repository, nhập User Name và Password rồi chọn Login. Tích chọn **Store password** để SmartGit nhớ mật khẩu.
2. Tại bước *Selection*, chọn **Next** nếu không chắc chắn muốn bỏ submodule, head và tag.
3. Tại bước *Local Directory*, chọn vị trí trong máy muốn lưu repo ở **Path** và nhấn **Finish** để hoàn tất.

# Check Out: chuyển đổi trạng thái repo theo commit

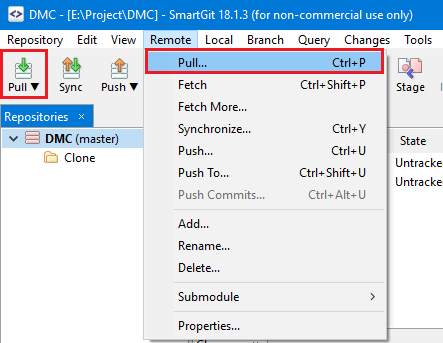
1. Đảm bảo repo đang không có thay đổi nào ngoài thêm file mới.
2. Dịch chuyển tới một branch khác: trong view *Branches*, nhấn đúp chuột vào 1 branch. Branch đó sẽ chuyển thành màu đậm để đánh dấu. Nếu branch chưa có trên local, SmartGit sẽ đề nghị tạo một branch local tương ứng (nếu không tạo, chỉ có thể xem repo mà không thể sửa đổi branch)
3. Dịch chuyển tới một commit khác: về bản chất, các branch cũng chỉ là các header trỏ tới các commit khác nhau, nên việc dịch chuyển tới branch là một phần của dịch chuyển tới commit. Muốn dịch chuyển repo tới trạng thái tương ứng với một commit bất kỳ, trên thanh menu chọn **Branch** > **Check Out…**



1. Trong cửa sổ *Check Out*, chọn commit và nhấn **Check Out**.

# Pull: Lấy phiên bản repo mới nhất và tích hợp vào repo trong máy

1. Trên thanh menu chọn **Remote > Pull…** hoặc nhấn nút Pull ngay trên toolbar

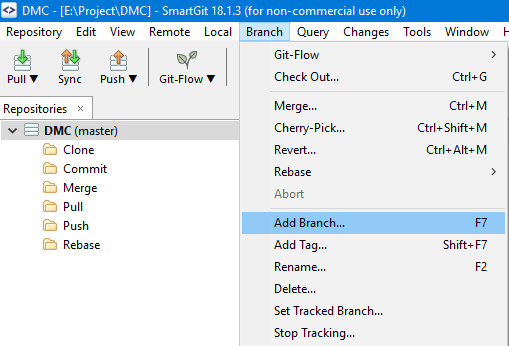


* 1. Nếu cửa sổ *Configure Pull* xuất hiện, chọn **Rebase** hoặc **Merge** và nhấn **Configure** để thiết lập tùy chọn mặc định cho nút Pull trên toolbar cho toàn repo.
* Rebase:
  + Tách tất cả các commit chỉ tồn tại trong local ra.
  + Lấy tất cả các commit mới trên remote đưa về máy.
  + Thực hiện lại từng commit local vừa tách ra theo đúng thứ tự, khiến cho commit local nằm trên các commit mới nhất của remote. Nếu không thực hiện các commit local thành công được, SmartGit sẽ báo có xung đột và yêu cầu xử lý.
* Merge: Ghép các thay đổi trong máy với các commit trên remote lại thành 1 commit riêng.

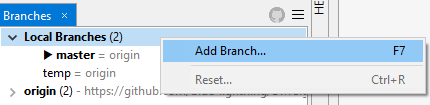
1. Trong cửa sổ *Pull*, chọn **Pull** để kết hợp tất cả các thay đổi trên local và remote lại hoặc **Fetch Only** để lấy các thay đổi trên remote về nhưng không đưa vào local. Mục *More Options* cho phép chọn kiểu pull là rebase hay merge; xem giải thích ở phần 1a phía trên. Nếu có conflict, xem mục ***Conflict*** để giải quyết.

# Add branch: thêm nhánh mới cho repo

1. Trên thanh menu chọn **Branch** > **Add Branch…** hoặc nhấn chuột phải vào dòng chữ **Local Branches** trên view *Branches*.

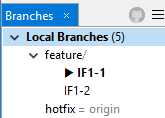


*Sử dụng menu*

**

*Sử dụng menu chuột phải*

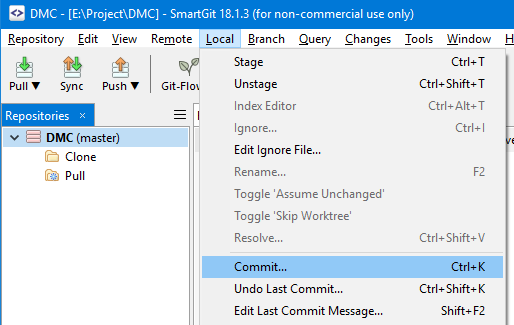
1. Trong cửa sổ *Add Branch*, nhập tên branch và chọn **Add Branch** hoặc **Add Branch & Checkout** nếu muốn chuyển sang làm việc trên nhánh này luôn.
   1. Git cho phép trong tên branch có một dấu gạch chéo */*. Các branch có phần trước dấu gạch chéo giống nhau sẽ được SmartGit gộp lại làm một nhóm.



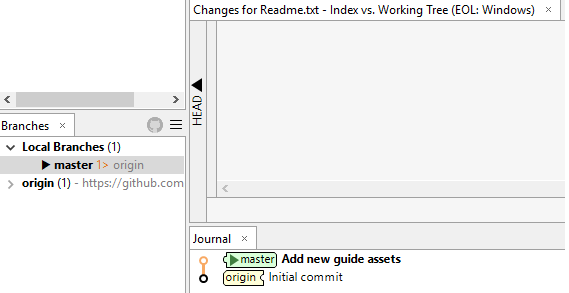
*2 nhánh feature/IF1-1 và feature/IF1-2 được SmartGit nhóm lại*

# Commit: Lưu các thay đổi vào repo trong máy

1. Pull repo để đảm bảo trong máy đã có phiên bản mới nhất của nhánh đang làm việc.
2. Trong view *Files* ở màn hình chính*,* chọn các thay đổi muốn lưu vào repo. Chọn nhiều file tương tự như chọn trong Windows Explorer với các phím Ctrl, Shift hay Ctrl+A…
3. Trên thanh menu chọn **Local** > **Commit…**

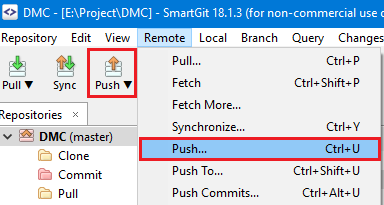


1. Trong cửa sổ *Commit*, kiểm tra lại và chọn tất cả các thay đổi muốn lưu vào repo, nhập nội dung vắn tắt thay đổi vào *Commit Message* và nhấn **Commit**. Sử dụng **Commit & Push** nếu muốn kết hợp đẩy thay đổi lên repo trên remote; nên hạn chế vì commit trên repo sẽ không thể sửa đổi dễ dàng.
   1. Nếu trong local có commit chưa đẩy lên remote, có thể tích chọn **Amend…** để gộp các thay đổi này vào commit của local gần nhất.
2. Sau khi commit thành công, repo sẽ đánh dấu số lượng commit chênh lệch giữa local và remote bằng màu cam trong view *Branches* và *Journal*.



# Push: Đẩy commit từ repo local lên remote

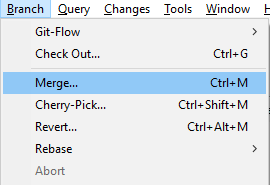
1. Trên thanh menu chọn **Remote** > **Push…** hoặc nhấn nút **Push** trên toolbar



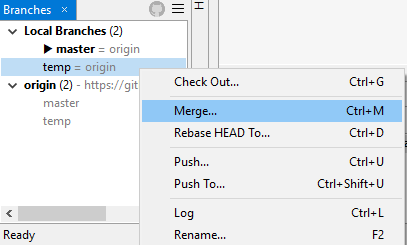
1. Cửa sổ *Push* hiện ra, chọn **Push** để xác nhận.

# Merge: Kết hợp thay đổi của các nhánh

1. Trên thanh menu chọn **Branch** > **Merge…** hoặc nhấn chuột phải vào nhánh muốn tích hợp và chọn **Merge…**

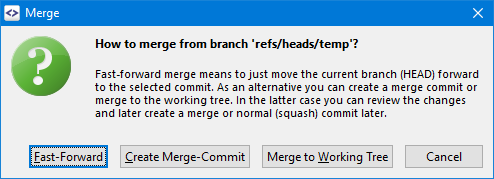


*Sử dụng menu*



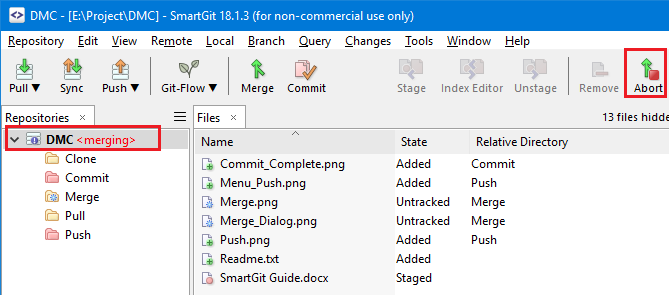
*Sử dụng menu chuột phải*

1. Nếu dùng menu, sẽ có 1 cửa sổ *Merge* hiện ra để chọn nhánh và cách ghép. Nếu dùng chuột phải, chỉ có các lựa chọn cách ghép hiện ra.



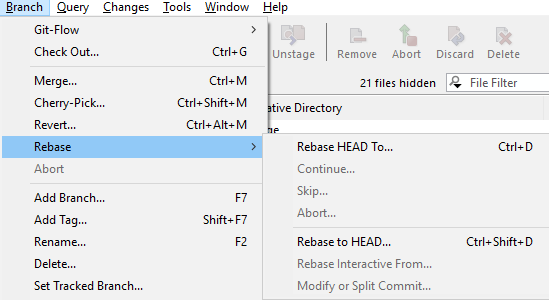
* 1. Fast-Forward: Nếu nhánh hiện tại là một phần của nhánh sắp được ghép vào (nhánh được ghép chính là nhánh hiện tại cộng thêm vài commit) thì không cần thực hiện commit. SmartGit sẽ dịch con trỏ nhánh hiện tại về commit cuối của nhánh được ghép đó; commit đó sẽ được 2 con trỏ của 2 nhánh trỏ vào.
  2. Create Merge-Commit: SmartGit sẽ ghép các thay đổi từ điểm chung cuối cùng của 2 nhánh từ nhánh được ghép vào nhánh hiện tại và commit thay đổi vào repo local với message “Merge branch X”.
  3. Merge to Working Tree: SmartGit sẽ ghép các thay đổi từ điểm chung cuối cùng của 2 nhánh từ nhánh được ghép vào nhánh hiện tại nhưng không commit. **Đây là cách nên làm** vì sẽ có thể xem các đoạn ghép tự động của SmartGit có sai lệch với ý định của mình không trước khi commit để sửa lại. Việc này nhằm đảm bảo code khi commit lên nhánh chắc chắn sẽ chạy được.

1. Nếu chọn Merge to Working Tree, view *Repositoríes* sẽ chuyển trạng thái thành <merging> để chờ commit. Chọn **Abort** trên toolbar để hủy bỏ hoặc thực hiện thay đổi thêm và **Commit** để đưa thay đổi lên repo. Nếu có conflict, xem mục ***Conflict*** để giải quyết.

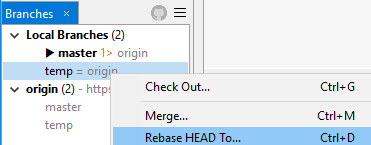


# Rebase: đưa các commit chồng lên trên một nhánh khác

1. Đảm bảo repo hiện tại trên local đang không có thay đổi nào chưa được lưu lại.
2. Trên thanh menu chọn **Branch** > **Rebase…** hoặc nhấn chuột phải vào nhánh muốn tích hợp và chọn **Rebase…** Nếu sử dụng menu sẽ có 2 lựa chọn trong **Rebase…**: **Rebase HEAD to** và **Rebase to HEAD.**



*Sử dụng menu*

**

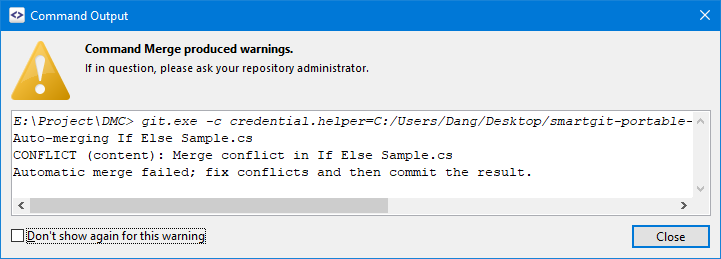
*Sử dụng menu chuột phải*

* 1. Rebase HEAD to: lấy ra từ nhánh hiện tại toàn bộ các commit tính từ điểm chung gần nhất giữa 2 nhánh. Đặt chúng lên trên commit cuối cùng của nhánh được rebase và chuyển con trỏ của nhánh hiện tại lên đó.
  2. Rebase to HEAD: sao chép toàn bộ nội dung các commit từ nhánh bị rebase và đặt chúng lên trên commit cuối cùng của nhánh hiện tại.

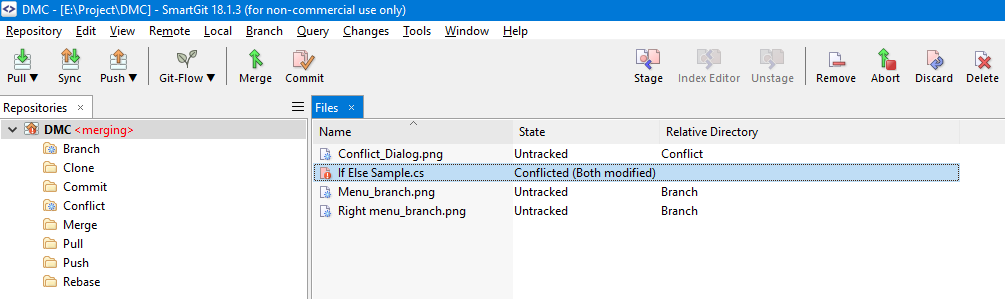
1. Chọn **Rebase HEAD to** hoặc **Rebase to HEAD** trong cửa sổ hiện ra.
2. Nếu Rebase thành công, các commit sẽ được đặt lại vị trí như mong muốn. Nếu có conflict, view *Repositoríes* sẽ chuyển trạng thái thành <merging> để chờ commit. Chọn **Abort** trên toolbar để hủy bỏ hoặc thực hiện thay đổi thêm và **Commit** để đưa thay đổi lên repo.

# Conflict: Giải quyết các xung đột khi commit code

1. Trong quá trình commit, merge, rebase… các xung đột giữa code trong local và code trên repo rất dễ xảy ra khi có 2 sửa đổi trên cùng một vị trí trên repo. Những lúc này các file xung đột sẽ chuyển trạng thái sang Conflicted.

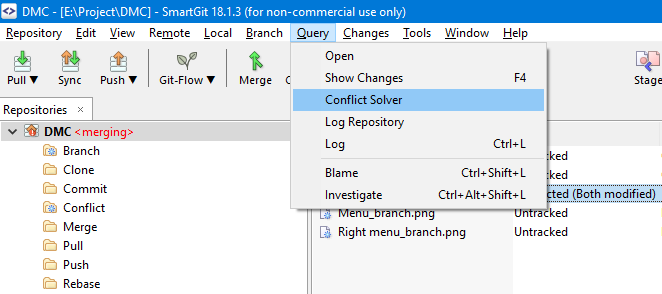


*Một thông báo lỗi điển hình của SmartGit khi có xung đột*

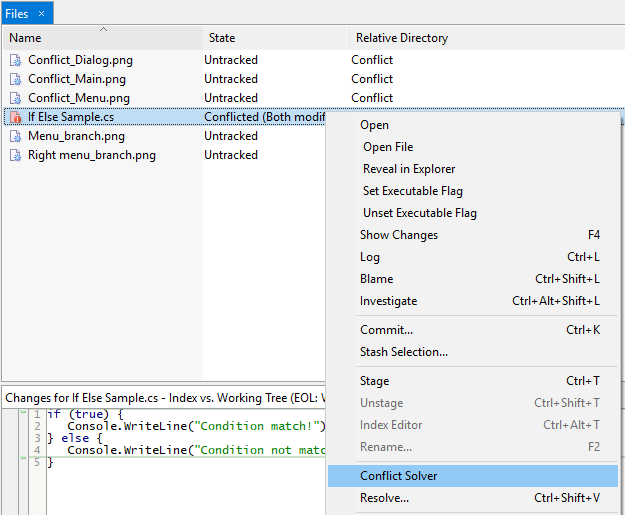
**

*Màn hình chính khi có xung đột; chú ý trạng thái* ***Conflicted*** *trong cột State*

1. Trên thanh menu chọn **Query** > **Conflict Solver…** hoặc nhấn chuột phải vào file muốn xử lý và chọn **Conflict Solver…**

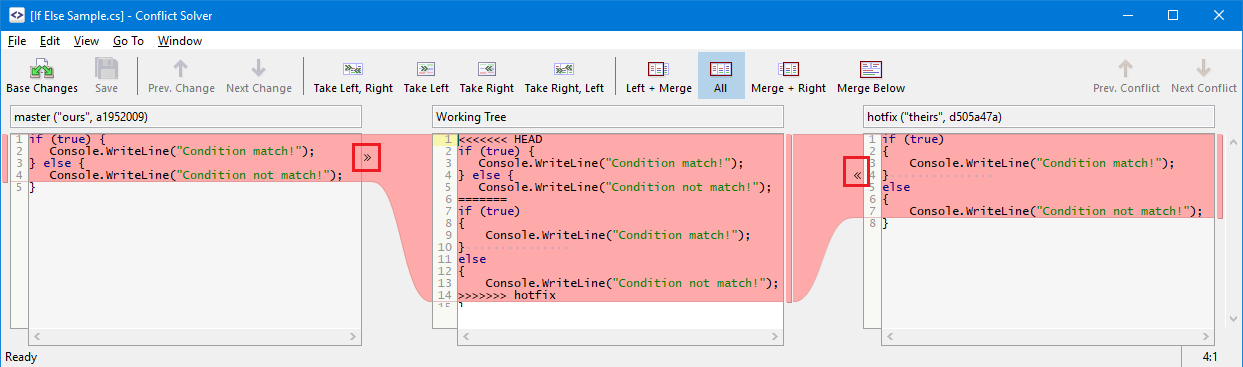


*Sử dụng menu*

**

*Sử dụng menu chuột phải*

1. Trong cửa sổ *Conflict Solver* sẽ có các lựa chọn lấy các thay đổi bên trái, bên phải hoặc trực tiếp kết hợp các thay đổi bằng tay. Khung bên trái luôn là thay đổi trên repo local hiện tại (“ours”), khung giữa là tình trạng thực tế của file trên máy (“Working Tree”) và khung bên phải là thay đổi từ nơi khác (“theirs”).



1. Có thể thay đổi ở khung giữa, cũng có thể sửa file ở ngoài SmartGit bằng công cụ khác tùy chọn. Sau khi đã lưu các thay đổi (với cửa sổ *Conflict Solver* nút **Save** nằm ở rìa trái thanh công cụ), **stage** lại file để đánh dấu thay đổi vào Index.
2. Lặp lại các bước trên cho đến khi không còn xung đột, sau đó thực hiện commit.