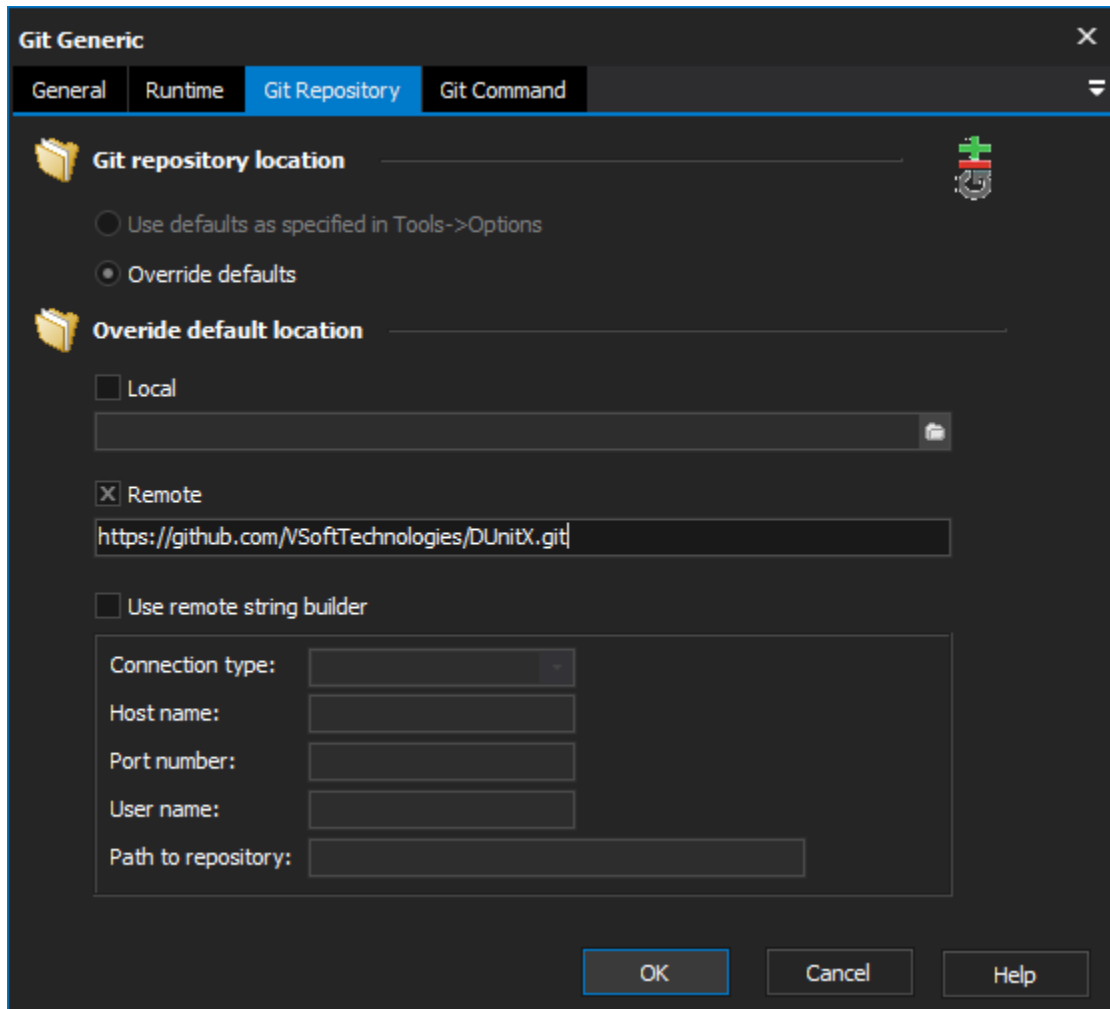


Git Merge Action

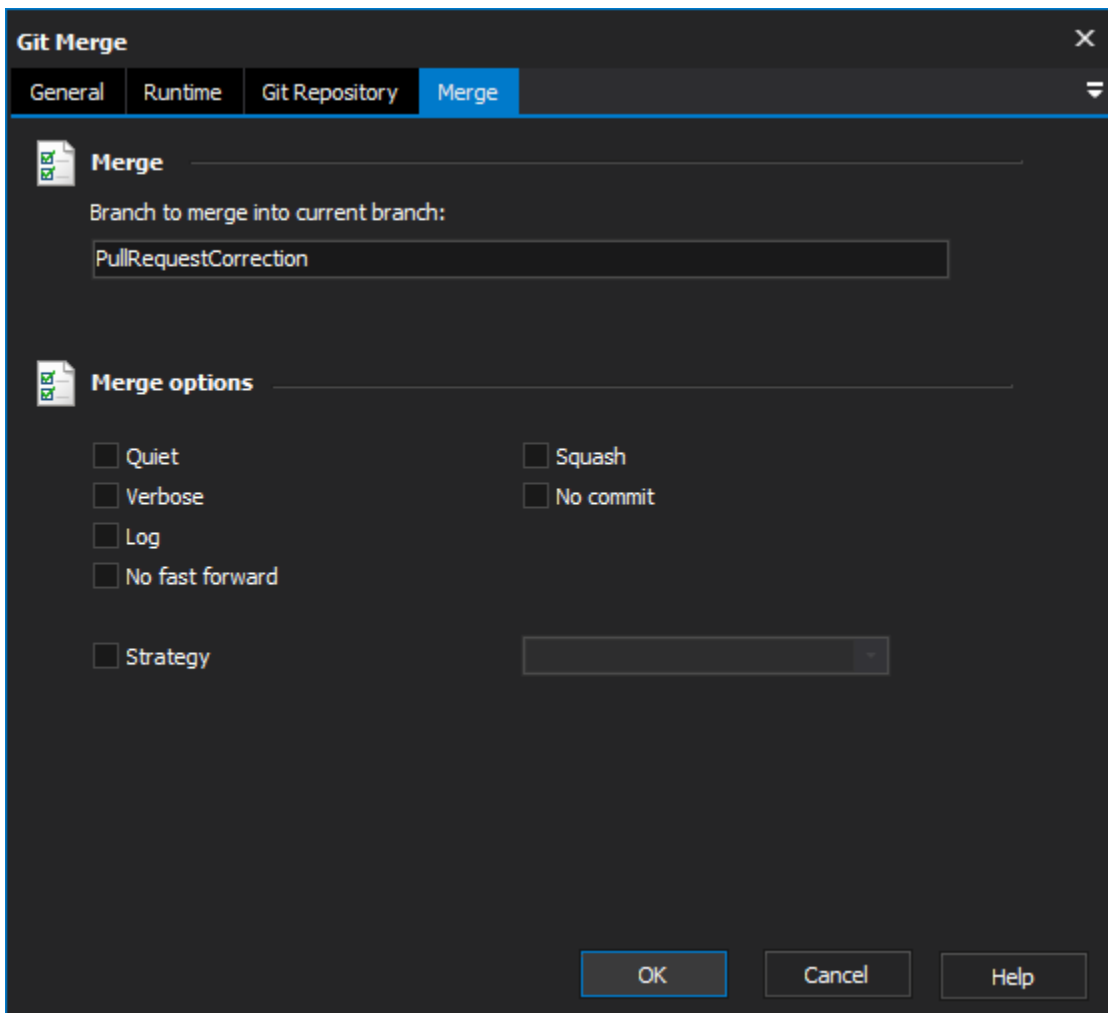
The Git Merge action allows you to join two or more development histories together.

From the Git Repository tab select the location of the local repository where you want to merge the branches.



The screenshot shows the 'Git Generic' dialog box with the 'Git Repository' tab selected. The dialog has four tabs: 'General', 'Runtime', 'Git Repository', and 'Git Command'. The 'Git Repository' tab is active, showing options for repository location. Under 'Git repository location', there are two radio buttons: 'Use defaults as specified in Tools->Options' (unselected) and 'Override defaults' (selected). Below this is the 'Override default location' section, which has two options: 'Local' (unselected) and 'Remote' (selected). The 'Remote' option has a text field containing the URL 'https://github.com/VSoftTechnologies/DUnitX.git'. There is also an unchecked checkbox for 'Use remote string builder'. Below this is a group box containing five fields: 'Connection type:' (a dropdown menu), 'Host name:', 'Port number:', 'User name:', and 'Path to repository:'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Switch to the Merge tab to specify which branches you would like to merge. To merge more than two branches simply type the names into the text field separated by a space.



There are also a number of options available on the Merge tab:

Quiet - Suppress output from Git.

Verbose - Verbose output from Git.

Log - Include single line descriptions from the actual commits that are being merged in message log.

No Fast Forward - Generate a merge commit even if the merge resolved as a fast-forward.

Squash - Produce the working tree and index state as if a real merge happened, but do not actually make the commit so that the next you run a commit it will create the merge commit.

No Commit - Perform the merge but pretend the merge failed and do not auto commit. This will allow user to make further changes before committing the results.

Strategy - Define the merge strategy to be used:

Resolve - Resolves two heads using a three-way merge algorithm. Considered safe and fast.

Recursive - Resolves two heads using a three-way merge algorithm. This option can detect and handle merges where renames are involved. This is the default strategy when pulling or merging a single branch.

Octopus - Resolves cases with more than two heads. Will not perform a complex merge where manual resolution is required. Intended for bundling topic branches together.

Ours - Resolves any number of branch heads. Result of the merge is the current branch head.

Subtree - Similar to the recursive strategy except this strategy adjusts the tree structure of the branches if one branch is a sub tree of another.

For more information on merging in git, see: