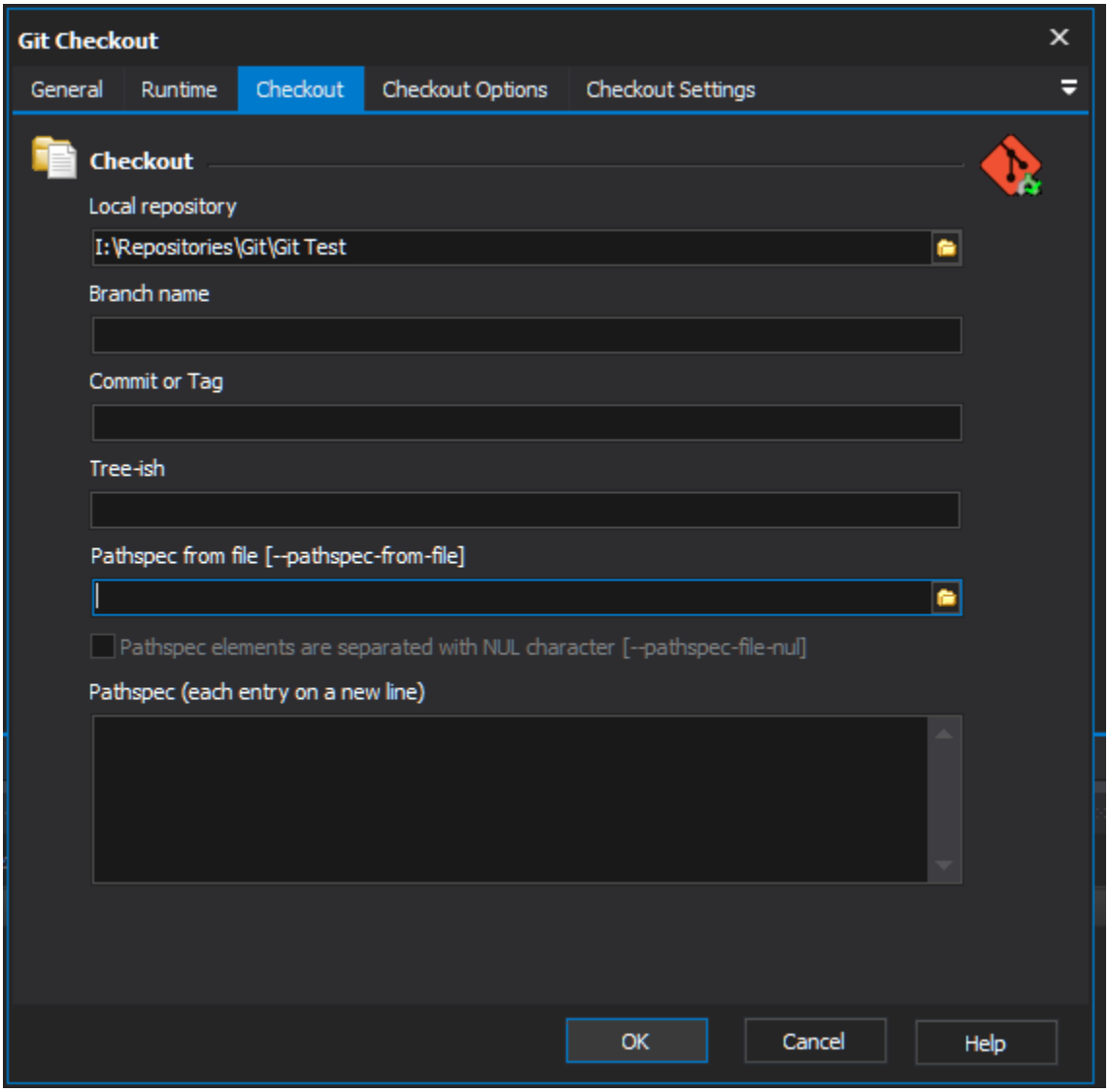
Git Checkout Action

The Git Checkout action allows you to checkout files to the working directory and switch to a different development branch. This action is a wrapper for the git command line. For more information on the use and options for this action, refer to the git checkout command line documentation.

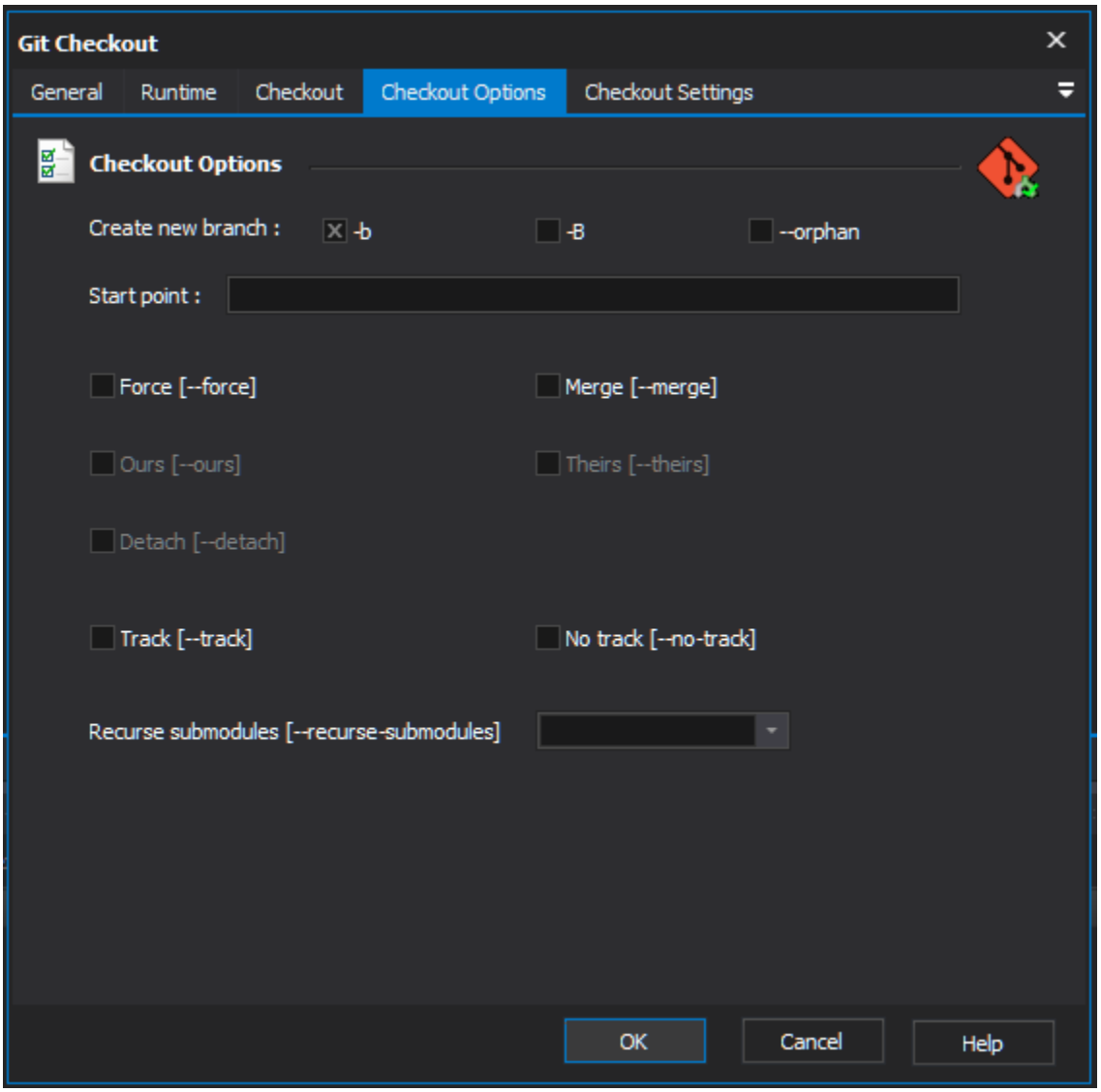
In the **Local repository** field of the **Checkout** tab, enter the path to a local Git working folder for the local repository that you want to checkout to.



You can then either:

- enter a branch name to switch to or add,
- enter a commit or tag to checkout,
- or enter a tree-ish (reference to commit) and optional list of files to checkout. The list of files may be entered into the **Pathspec** field or loaded from a pathspec file by specifying the file path in the **Pathspec from file** field.
  When adding files to the list, ensure each entry is placed on a new line. The asterisk can be used as a wildcard character, which saves having to manually enter every file that you want to add.

The **Checkout Options** tab allows you to specify some options to pass to the Git Checkout command line.



Create new branch - Select **-b** to create a new branch and checkout out to that branch. Select **-B** to do the same but also reset to any existing the branch., Alternatively you can use the **--orphan** option create an orphaned branch based on **Start Point** and switch to it.

Start point - The commit at which to start a new branch.

Force - Allow switching branches when the index or the working tree is different to the head, which will throw away local changes.

Merge - Allows to you perform a three-way merge between the current branch, working tree contents and the new branch in order to preserve modifications. This occurs when attempting to switch branches where the current branch has local modifications that differ from the branch that is being switched to.

Force - Allow switching branches when the index or the working tree is different to the head, which will throw away local changes.

Ours - For unmerged paths, check out the original working branch.

Theirs - For unmerged paths, check out the branch that you are merging into.

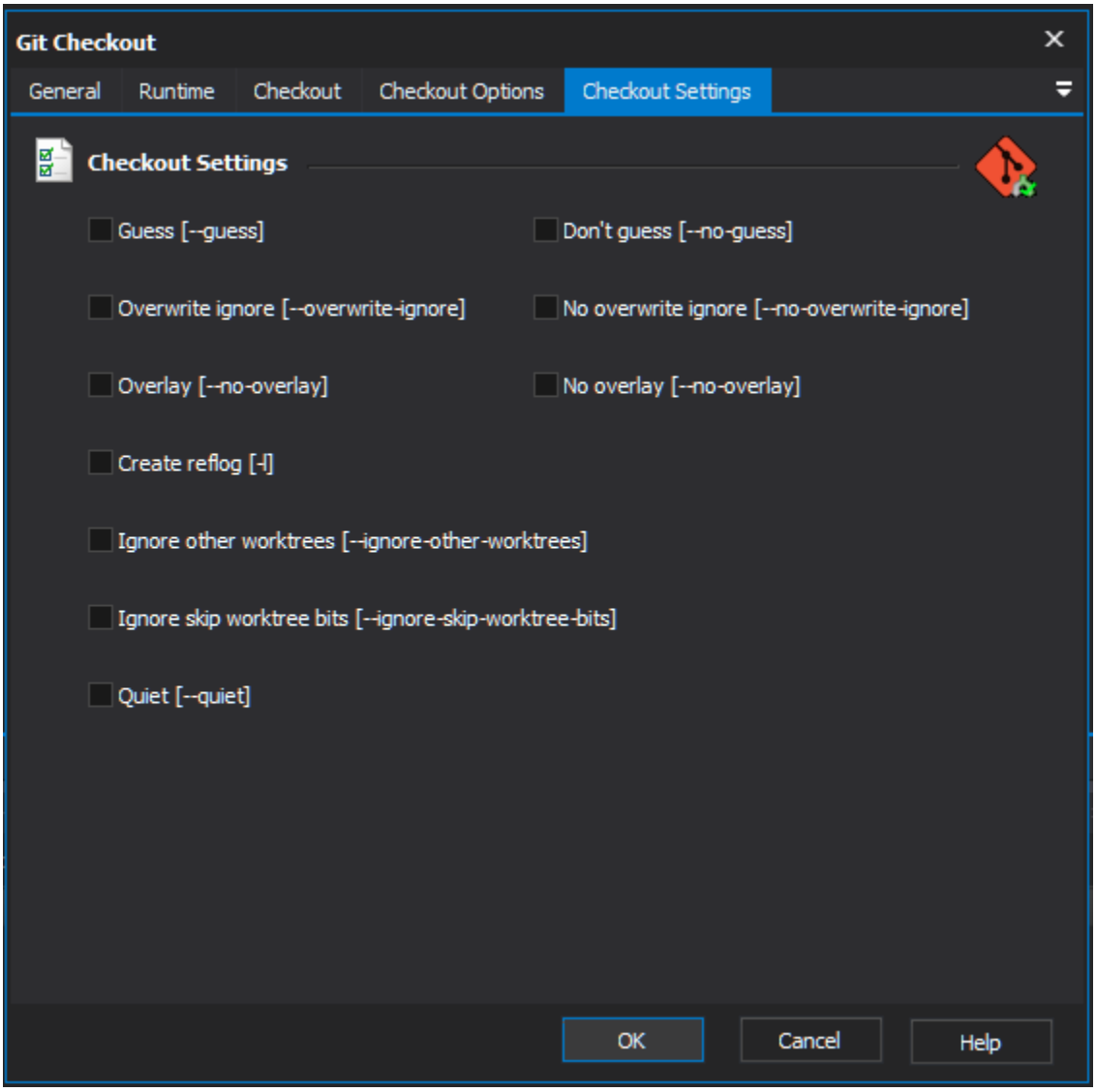
Detach - Rather than checking out a branch to work on it, check out a commit for inspection and discardable experiments.

Track - When creating a new branch, set up "upstream" configuration.

No track - Do not set up "upstream" configuration, even if the `branch.autoSetupMerge` configuration variable is true.

Recurse submodules - Update the content of all active submodules according to the commit recorded in the superproject.

The **Checkout Settings** tab allows you to specify some additional settings to pass to the Git Checkout command line.



Guess - If branch is not found but a tracking branch exists with a matching name, use that branch;

Don't Guess - Don't use a different remote branch with matching name.

Overwrite ignore - Silently overwrite ignored files when switching branches.

No overwrite ignore - Abort the operation when the new branch contains ignored files.

Overlay - Never remove files from the index or the working tree.

No overlay - Remove files that appear in the index and working tree, but not in tree-ish.

Create reflog - Create the new branch's reflog

Ignore other worktrees - Allow checkout when the commit is already checked out to another working folder

Ignore skip worktree bits - Ignore any the sparse checkout patterns

Quiet - Suppress all output.

For more information on adding files to git, see the git checkout command line documentation.