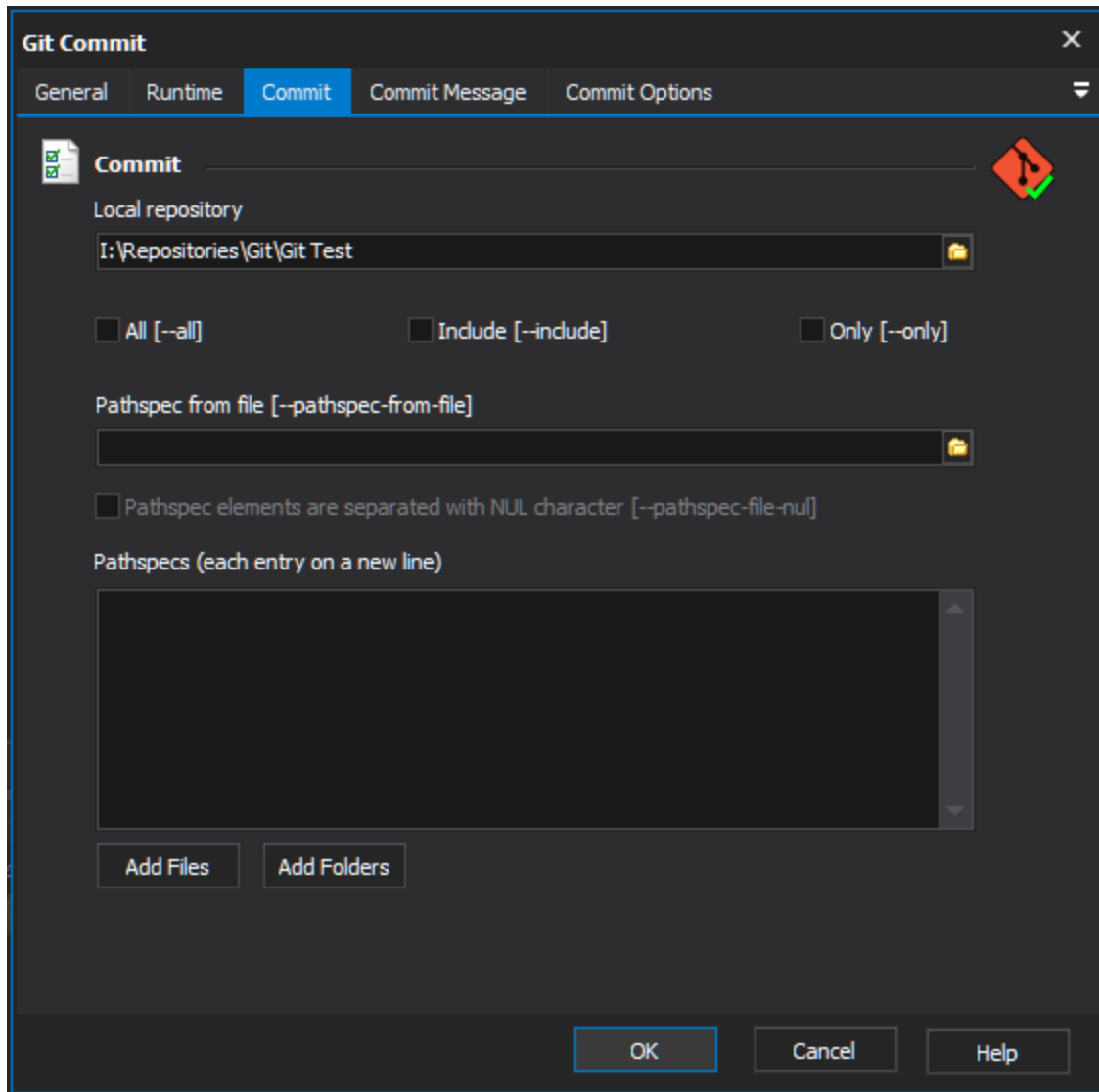# Git Commit Action

The Git Commit action allows you to record a snapshot of currently staged changes to a git local repository. 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 commit command line documentation.

To run this command, you will need to specify the working directory of the local repository to commit to.

Local repository - The working directory of the local repository.

You can then either commit all staged files, additionally commit all modified and deleted files, specify a list of files to commit either by entering them directly or via a pathspec files.
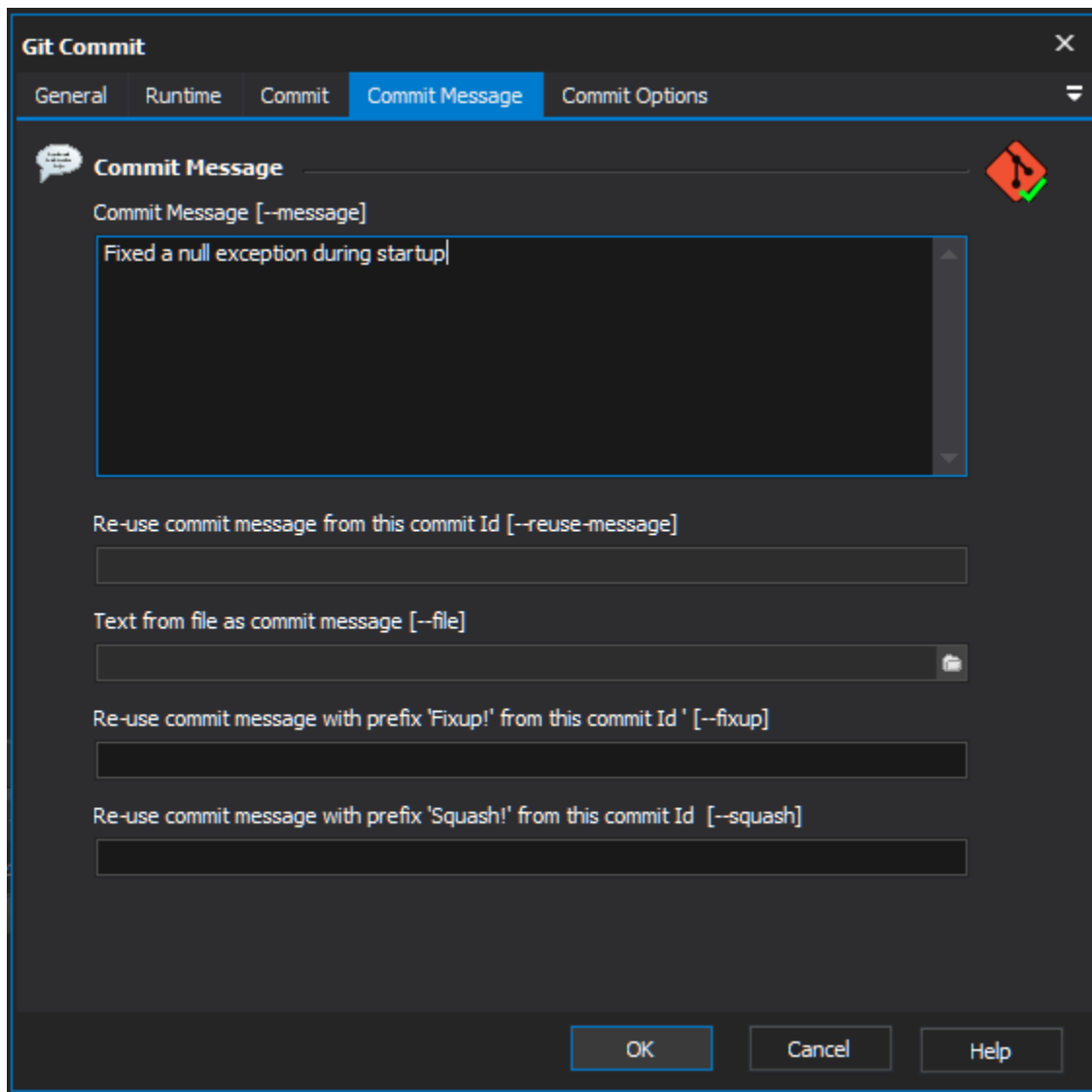
All - Automatically stage files that have been modified and deleted, but not any new untracked files before committing. You need to use a separate git add to stage untracked files.

Include - Stage the files listed in the Pathspecs before making the commit. The resultant commit will include both existing staged files and the files listed in Pathspecs. Note that you can only include modified or deleted files with this option, not new untracked files.

Only - Only commit the files listed in the Pathspecs, ignoring any other currently staged files. If this option is used in conjunction with the Amend option then no paths need to be specified and this can be used to amend the previous commit without committing changes that are currently staged.

The list of files for the above options may be entered into the **Pathspecs** field or loaded from a pathspec file by specifying the file path in the **Pathspec from file** field. When adding files/folders to the PathSpecs, 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.

To perform a commit a message is required. Switch to the **Commit Message** tab to provide a message. The message can be entered directly, loaded from a file or reused from a previous commit,

Commit Message - The message to associated with the commit.

Re-use commit message from this commit id - Enter the id of an existing commit. The message and the authorship information (including the timestamp) will be reused from this commit from when creating the new commit.

Text from file as commit message - Enter the path to a file containing text to be used for the commit message.

Re-use commit message with prefix 'Fixup!' from this commit id - Enter the id of an existing commit. The message will be reused from this commit from when creating the new commit. The message will be prefixed with "Fixup!" and will be added to any commit message defined using other fields.

Re-use commit message with prefix 'Squash!' from this commit id - Enter the id of an existing commit. The message will be reused from this commit from when creating the new commit. The message will be prefixed with "Squash!" and will be added to any commit message defined using other fields.

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

Switch to the Commit Options tab to see the list of available options that can be executed with this action.

Quiet - Suppress output from Git.

Verbose - Verbose output from Git.

Amend - Creates the new commit as an amendment which replaces the tip commit on the current branch.

Allow empty - Allows you to override the safety mechanism that prevents you from making a commit where the tree is exactly the same as its sole parent commit.

No verify - Bypass running any pre-commit and commit-msg hooks.

No post rewrite - Bypass running any post-rewrite hook.

Sign off - Add the text "Signed-off-by : <committer name>" to the end of the commit message.

Reset author - When reusing or amending an existing commit, specify that the authorship of the resulting commit now belongs to the committer.

Author - Override the author name used in the commit. Specify an explicit author using the standard "A U Thor <author@example.com>" format. Otherwise the text entered is assumed to be a pattern and is used to search for an existing commit by that author; the commit author is then copied from the first such commit found.

Date - Override the author date used in the commit.

Clean Up - Determines how the commit message is cleaned up:

- **Strip** - Removes leading and trailing empty lines, trailing whitespace, commentary and collapse consecutive empty lines.
- **Whitespace** - Same as the Strip option except the commentary is not removed.
- **Verbatim** - Do not change message at all.
- **Default** - Same as the Strip option if the message is to be edited. Otherwise same as Whitespace.

Untracked Files - Specifies how untracked files are listed in output:

- **No** - Do not show untracked files.
- **Normal** - Show untracked files and directories.
- **All** - Show untracked files and individual files within untracked directories.

For more information on committing changes to git, see the git commit command line documentation.