

Integrating SimDiff 4 with Git

Contents

Introduction	2
Diff and merge tools in Git	2
Configuration	3
Configuring SimDiff 4 as a Diff Tool from Windows command prompt	3
Configuring SimDiff 4 as a Merge Tool from Windows command prompt	3
Configuring SimDiff 4 as a Diff Tool from a Git client (like Git Bash)	3
Configuring SimDiff 4 as a Merge Tool from a Git client (like Git Bash)	4

Introduction

Git is a distributed version control system. The primary interface for working with the Git is the command-line, using commands such as `git commit`, `git push`, etc. However, there are a number of GUI tools that have been built on top of the command line that provide a more convenient interface for working with a repository. Some of these interfaces support interactive diff and merge tools - others do not.

Because of the number and variety of Git GUI tools, here we are only providing instructions for configuring SimDiff 4 for use with the command-line Git interface. Please consult your GUI tool's documentation to determine the configuration steps necessary to expand the integration to the GUI tool.

Diff and merge tools in Git

There are two categories of external diff and merge integration that can be configured in Git - tools and drivers. Diff and merge drivers are used to implement a low-level, non-interactive integration with external tools. This type of integration is not suitable for an interactive tool like SimDiff 4. Instead, SimDiff 4 should be configured as a diff and/or merge tool.

Once configured, SimDiff 4 can be launched from Git using the `git difftool` and `git mergetool` commands. Here are a few examples:

Compare the working copy of `my_model.mdl` against the repository copy, using SimDiff:

```
git difftool -t simdiff4 my_model.mdl
```

Compare two repository revisions of `my_model.mdl` against each other, using SimMerge:

```
git difftool -t simdiff4 <commit1> <commit2> my_model.mdl
```

Resolve a merge conflict in a model file `my_model.mdl` after merging a branch:

```
git mergetool -t simdiff4 my_model.mdl
```

Configuration

SimDiff 4 can be configured as diff tools for Git. They can even both be configured at the same time. The configuration commands included below assume default installation locations for SimDiff 4. Modify the commands as necessary if you installed to a different location.

To perform the configuration, use the `git config` command (recommended), or locate the appropriate configuration file and modify it by hand using a text editor. There are three file locations that configuration settings may be saved to, depending on what `--file` flag is provided:

- `[none]` - stores configuration settings in the repository. This is the default behavior if no file flag is specified. This option is not recommended, because not all computers that clone the repository will have SimDiff 4 installed.
- `--global` - stores configuration settings in the user's home directory. Use this option if you are the only user that will use SimDiff 4 with Git, or if you don't have administrator privileges.
- `--system` - stores configuration settings in the Git installation directory. Use this option to configure SimDiff 4 as a diff/merge tool for all users on the machine. This option is recommended, but requires administrator privileges.

Configuring SimDiff 4 as a Diff Tool from Windows command prompt

Use the following commands to configure SimDiff as a diff tool:

```
git config --system difftool.simdiff4.cmd
    "\"C:/Program Files/EnSoft/SimDiff 4/simdiff4.exe\"
    -l \"$LOCAL\" -r \"$REMOTE\"
```

Configuring SimDiff 4 as a Merge Tool from Windows command prompt

Use the following commands to configure SimDiff 4 as a merge tool:

```
git config --system mergetool.simdiff4.cmd
    "\"c:/Program Files/EnSoft/SimDiff 4/simdiff4.exe\"
    -merge -b \"$BASE\" -l \"$LOCAL\" -r \"$REMOTE\" -o
    \"$MERGED\"
```

Configuring SimDiff 4 as a Diff Tool from a Git client (like Git Bash)

Use the following commands to configure SimDiff as a diff tool:

```
git config --system difftool.simdiff4.cmd '"c:/Program
    Files/EnSoft/SimDiff 4/simdiff4.exe' -l "$LOCAL"
    -r "$REMOTE"
```

Configuring SimDiff 4 as a Merge Tool from a Git client (like Git Bash)

Use the following commands to configure SimDiff as a merge tool:

```
git config --system mergetool.simdiff4.cmd '"c:/Program  
Files/EnSoft/SimDiff 4/simdiff4.exe" -merge -b "$BASE"  
-l "$LOCAL" -r "$REMOTE" -o "$MERGED"'
```