Integrating Simulink and Stateflow with Git using SimDiff

Contents

Last updated 2024-11-12

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 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. Instead, SimDiff should be configured as a diff and/or merge tool.

Once configured, SimDiff 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 simdiff my_model.mdl
```

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

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

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

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

Configuration

Git can be configured to use SimDiff for diff and merge. To perform the configuration, use the git config command with one of the options below. EnSoft recommends using *`--system`* if possible, or *`--global`*.

- `--system`: Configures for all users on a machine but requires administrator privileges.
- `--global`: Configures for a single user, affecting all repositories.
- `--local`: Configures a single repository, storing the settings in the repository. Note that repository settings take precedence over global and system settings, therefore they will affect other users that clone the repository.

The configuration commands included below assume default installation locations for SimDiff. Modify the commands as necessary if you installed to a different location.

Configuring SimDiff as a Diff Tool from Windows command prompt

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

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

Configuring SimDiff as a Merge Tool from Windows command prompt

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

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

Configuring SimDiff 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.simdiff.cmd '"C:/Program
     Files/EnSoft/SimDiff/simdiff4.exe" -l "$LOCAL"
     -r "$REMOTE"'
```

Configuring SimDiff 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.simdiff.cmd '"C:/Program
     Files/EnSoft/SimDiff/simdiff4.exe" -merge -b "$BASE"
     -l "$LOCAL" -r "$REMOTE" -o "$MERGED"'
```