Version control

Since projects only contain SQL scripts and Visual Studio configuration files, they can be committed to version control without any extra configuration.

When committing to version control you should ignore the following:

bin folder
obj folder
[project name].dbmdl file
[project name].jfm file
[project name].sqlproj.user file

It is especially important to ignore the user file as this contains user-specific settings which can cause unexpected behavior for other users.

Working with Git

Sample .gitignore file for SQL Change Automation

A *.gitignore* file containing rules to ignore the files listed above can be downloaded here.

Place your .gitignore file in the same folder as your SQL Change Automation project file (.sqlproj file).

Git Branching Strategies for SQL Change Automation

Git encourages workflows that branch and merge often. We encourage Git users to embrace branching and merging for their database code for the same reasons many people do this for application code. Read more about this in "The Managers Guide to Git Training for Database Administrators."

SQL Change Automation works well with many popular branching strategies, including Git Flow, Release Flow, and other patterns. If you are not sure which pattern to begin using, we often advise:

- If a Git branching strategy is already working well for a team you work closely with, consider adopting this strategy and asking a member of that team to serve as a coach for your group
- If this is not the case, the Azure DevOps Release Flow model works well for many teams who are first starting out

Learn more

- Switching branches
- Using version control in SSMS
- Using Git to share your changes in SSMS
- Setting up a mergetool when working in SSMS