IBM i Git Source Management Made Easy

Implement Git for your Classic IBM i Source Version Control in Minutes

Git is the most popular source control versioning tool available for developers today. Git has been available natively on IBM i for IFS source management for several years. The biggest challenge with using Git for traditional IBM i developers has been the lack of seamless integration and management of SEU, PDM and source physical files with a Git repository for classic 5250 based source version control. RDi developers have tried iProjects and other methods to manage source in the IFS, but the result is often a complex and frustrating edit, compile and commit workflow experience that doesn't feel natural because of the many manual steps.

iForGit solves the problem of awkward IBM i Git source management with a native IBM i Git client interface allowing Git to be easily integrated with 5250 development workflows using SEU, PDM and RDi editors and custom user options. iForGit commands also work with SSH terminal sessions to integrate with editors such as VS Code and others that support SSH connectivity.

IBM i developers can learn to use iForGit within minutes or source version commits can be fully automated so developers initially have zero learning curve.

For more complete source management and automated build processes, iForGit commands can be used in conjunction with open source and commercial IBM i enterprise source and project management toolsets if a development team has already implemented a custom build solution but needs to add Git source version control to the process.

Highlights

• Install and start versioning IBM i source with Git in minutes.

• Immediately add audited Git IBM i source management and compliance.

• Source code versioning can be made 100% automatic and transparent for developers.

• Works with PDM, SEU, RDi and VS Code to manage source versioning.

• Git repository backups can be scheduled for automatic hands-off source version commits.

• Local Git source repositories are managed in the IFS to augment existing source backups.

• Git repositories can live on-premise or in the cloud using a remote Git host such as:
  GitHub, Bitbucket, GitLab, GitBucket, Azure DevOps, Bonobo Git Server or any Git project hosting environment.

• Descriptive source member information can be stored in source member headers.

• Line numbers and date sequences can be retained to support SEU and RDi editing environments.

• Source change and audit info can be viewed with git log, show, diff or blame.

• Old source member versions can be easily restored to source library from a Git repository.

• Git source control can enhance and augment existing source control and build processes.