

Git

[Go up to Terms and Definitions](#)

Discussion

Git is a distributed version control system used to record, organize, compare, and exchange changes to files and directories maintained in a repository.

A Git repository contains the recorded history and associated metadata for a managed body of content. A complete clone normally includes the repository history rather than only the most recent version. This distributed model allows participants to perform many repository operations locally and exchange changes with other repositories. :contentReference[oaicite:1]{index=1}

Git represents recorded repository states as commits. Git workflows commonly use:

- Repositories
- Commits
- Branches
- Tags
- References
- Working trees
- Staging
- Merging
- Rebasing
- Fetching
- Pulling
- Pushing
- Cloning

Git records content snapshots and the relationships among commits. This history enables users and automated processes to identify changes, compare repository states, trace the ancestry of a change, and reproduce an identified repository state. :contentReference[oaicite:2]{index=2}

Within Crucible documentation, Git provides repository and change-history capabilities used by:

- Git-based workflows
- GitOps practices
- [CI/CD Pipelines](#)
- Configuration management
- Automated build operations
- Automated deployment operations
- Automated validation operations
- [Provenance](#)
- [Reproducibility](#)
- [Traceability](#)

Definition

distributed version control system that records repository states and the relationships among changes to managed content

Source

Adapted from the Git Project documentation:

- *Git*
- *git Documentation*
- *Pro Git: About Version Control*
- *Pro Git: What Is Git?*

Note

The name **Git** is not an acronym and should not be written as **GIT**.

Git is distinct from repository-hosting services such as GitHub, GitLab, and Bitbucket. Those services can host and manage Git repositories, but they are not Git itself.

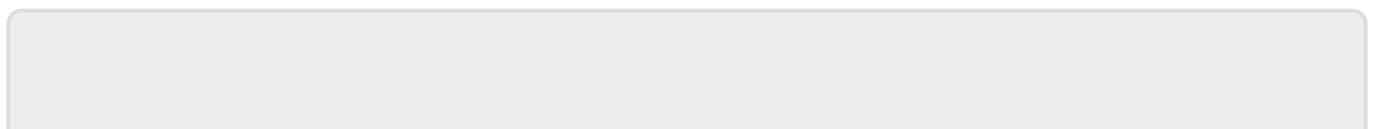
Git does not require a single permanent central repository. A project can designate one repository as an authoritative or shared repository through policy and workflow conventions.

A Git-based workflow uses Git repository states and operations to coordinate change. The existence of a Git repository alone does not define the workflow, approval rules, branch strategy, automation, or deployment behavior.

Example

A contributor commits a Declarative Configuration change to a Git branch. A review process approves and merges the change into the designated branch. A CI/CD Pipeline identifies the resulting commit and initiates automated build, validation, and deployment operations associated with that repository state.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.



From:
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:
https://wiki.didosolutions.com/dido/99_annexes/annex-b-terms-and-definitions/g/git?rev=1785444365

Last update: **2026/07/30 13:46**

