Git

Git

Repository URL

Enter the location of your git repository. This is the value of the **repository** parameter that gets passed to the clone command line option.

Examples:

- `http[s]://host.xz[:port]/path/to/repo.git/`
- `ssh://host.xz[:port]/path/to/repo.git/`
- `git://host.xz[:port]/path/to/repo.git/`
- file:///path/to/repo.git/
- `ftp[s]://host.xz[:port]/path/to/repo.git/`
- `rsync://host.xz/path/to/repo.git/`
- `/path/to/repo.git/`

Generally, the https and ssh protocols are preferred for remote repositories.

Authentication Method

This is displayed only after a Repository URL starting with https//: (or http://) is entered.

Select one of the following authentication methods:

- **Built-in credential helper (recommended):** The specified username and password are passed securely to Git when requested by the credentials API.
- **None:** No username and password are passed to Git. If the repository requires credentials then Git will request them at a command line prompt and wait for a response. Only use this option if a credentials manager such as Git Credential Manager for Windows is installed and the required credentials are already stored under the profile of the user account running the Server service otherwise the Git repository may stall for several seconds before timing out.
- **Embed username and password in URL:** The username and password are passed as part of the Git repository URL. This option is insecure and **not recommended**. It is only available for only backwards compatibility.

Proxy

Select a Proxy type to open the Proxy tab page for entering proxy details

Recursive

Tick this to pick up submodules when fetching source files from the repository.

> ⓘ **Note:** When using a remote repository, ensure that any local submodule commits are pushed before related commits on the local repository. If a main repository commit is picked up which is pointing tor a missing submodule commit, Continua will not be able to update the repository cache and a popup error message will be displayed against the changeset in the Configuration Changes view. You can resolve this error, after pushing the changes from the submodule, by clicking a link next to the error message.

Fetch other remote refs

Check this to specify refspecs for fetching remote branches . e,g

```
+refs/pull/*/merge:refs/remotes/origin/pr/*
```

See Git Internals - The Refspec for further information

Using

The Using drop down is populated with any property collector whose namespace matches the pattern defined by the Git plugin. The pattern for this action is **`^git\..*`**

If you create a property collector for this action, make sure you select the **Path Finder PlugIn** type and give it a name that will match the pattern above in blue. Example names listed here, search the table's Plugin column for "**Git**".

For more in-depth explanations on property collectors see Property Collectors.

Alternatively, you can select the **Custom** option from the Using drop down list and specify a path in the resulting input field that will be displayed. Please read Why it's a good idea to use a property collector before using this option.

Credentials

Username

If your repository requires a username, enter it here.

Password

If your repository requires a password or access token, enter it here. Note that GitHub requires a personal access token rather than a password. Ensure that you give your access token access permissions to your repository.

Proxy

New Repository

| Repository | Git | Credentials | **Proxy** | Branches | Tags | Scripts | Options | Filtering | Downtime |

Hostname

Port   21

Username

Password

👁 ✅

🔘 Validate   💾 Save   ✖ Cancel   ❓ Help

If your company uses a proxy to access external networks, enter the relevant **Hostname**, **Port**, **Username** and **Password** here.

Branches

Branches To Monitor

There are 3 options to choose from when selecting which branch(es) to monitor.

1. **Single branch**
   This option tells Continua CI to only monitor a single branch of the repository you specify. When you select this option you need to provide a **Branch Name** which is the value of the branch you want Continua to monitor.

2. **All branches**
   This option tells Continua CI to monitor all branches in the repository. When you select this option you need to provide a **Default Branch** value so Continua knows what to use for a build when no branch is selected.

3. **By pattern**
   This option tells Continua CI to monitor any branch that matches a pattern you provide in the **Branch Pattern** field. The Branch Pattern field accepts a regular expression which is matched against all the branches in your mercurial repository. Since the pattern can potentially match more than one branch, you also need to provide a **Default Branch** value so Continua knows what to use for a build when no branch is selected.

Default Branch

Branch to use when not performing a feature branch build.

Use case-sensitive branch matching

Tick to take case into account when matching the branch against the branch pattern.

Maximum Active Branch Age (days)

Enter a non-zero value to hide old branches. Any branches which have no commits or builds within the specified number days will be excluded from the branch list on the Start Build and Trigger dialogs.

Tags

Tag Changes

Choose how to deal with tagged commits. You can choose to **Ignore** tags completely, **Detect** tags and show then next to changesets, or **Detect and list as new changeset** . The later option will create a new changeset in the database representing that a tag change (new tag, tag deletion or modification) was made. This new changeset can be used to trigger a build.

Load any older changesets which have been tagged

Normally only the latest changeset on each branch is loaded into the database when the repository is initialised. When this option is ticked earlier changesets which are the source of a tag are also loaded during initialisation. All tags are then available to be selected when starting a build.

Push Repository URL

The URL for pushing tag changes (made by the Tag Repository Changeset build event handler) to the remote Git repository.

Push Username

An alternative username to use when pushing tag changes to the remote Git repository. Leave blank to use the main repository Username.

Push Password

An alternative password (or personal access token) to use when pushing tag changes to the remote Git repository. Leave blank to use the main repository Password.

Scripts

Post-Checkout Script

The path to a Bash shell script file to execute in the repository working folder directly after each changeset is checked out. This can be used to write Git status and commit details to a file which is then added to the changeset source files. Relative paths are rooted in the repository cache working folder.

> ⓘ Note that post-checkout scripts run on the server and can only be edited by administrators or users with a Manage Server-Side Scripts permission.
>
> The execution of these scripts can also be locked down using the Continua.ExecuteRepositoryServerSideScripts app setting in the server configuration file. This can be set to either:
>
> **None**: No repository scripts can be executed,
> **Anywhere**: Any repository scripts in any location will be executed,
> **ExcludeRepositoryFolders**: Repository scripts in any location outside of the repository cache folder will be executed, or
> **CurrentRepositoryFoldersOnly** : Only repository scripts inside the repository cache folder for the current repository will be executed

Run as Git hook

Tick to copy the script file to the Git hooks folder to be triggered by the Git post-checkout event. Leave this cleared to run the script directly after untracked files are cleaned up.

A script running as a Git post-checkout hook has access to the arguments passed in by Git, including refs of previous and new HEAD and flag indicating checkout type. Note however that the working folder may contain untracked files, which are cleaned up after checkout. You can specify paths to exclude from the clean up in the **Post Checkout Output Paths** field. This field accepts Ant patterns to specify multiple file paths.

Run shell login files

Tick to use the --login shell command option to execute login shell startup files when running the script directly.

Restricted shell

Tick to use the --restricted shell command option to start a restricted shell environment when running the script directly.

Use POSIX standard

Tick to use the --posix shell command option to change the behaviour to match the POSIX standard when running the script directly.

Script Arguments

Enter any arguments to wish to pass to the post-checkout script here.

Options

Use author name

Use the author name instead of committer name as changeset user.

Authentication prompt timeout (in seconds)

How long to wait for Git commands which require authentication to respond with output timeout. This includes clone, fetch and push operations and prevents the git command line hanging if authentication is not set up correctly.

Allow insecure certificates

Allow insecure (such as self-signed) SSL certificates.

Write changeset details to file

Tick to write changeset details to the file "_CI_ChangesetDetails.xml" after checking out each changeset.

Filtering

These filtering operations are applied when new changesets are detected during repository polling. Note that they are not applied retroactively to previously detected changesets unless the repository is reset.

Path Filtering

Specify patterns to match folders or files to include and/or exclude from the repository cache. Type each pattern on a new line using the ANT pattern format or as regular expressions prefixed with '*re:'.*

Changeset Exclude Patterns

Specify patterns to match changesets to exclude. The changeset will only be excluded if all modified files match one or more of the patterns. Type each pattern on a new line using the ANT pattern format.

Downtime

New Repository

| Repository | Git | Credentials | Proxy | Branches | Tags | Scripts | Options | Filtering | **Downtime** |

Specify downtimes when your repository is unavailable. During these times Continua CI will not check your repository for changes.

Note: All times are in 24 hour format and are localised to the server's time. If an earlier time is entered in the *To* input box this will taken to be on the following day. e.g. Saturday from 22:00 to 01:00 means from 22:00 Saturday to 01:00 Sunday.

Every day ▾   From: [hh:mm]   To: [hh:mm]   ⊕

⊘ Validate   💾 Save   ✕ Cancel   ⊙ Help

Specify schedules for times when your repository are unavailable. During these times Continua CI will not poll your repository for changes.