Reclaiming mannequins for GitHub Enterprise Importer

In this article

After your migration, you can assign the history of a placeholder identity, or mannequin, to a member of your organization.
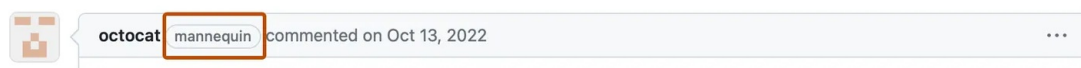
> **Who can use this feature**
> Organization owners can reclaim mannequins.

About mannequins

After you run a migration with GitHub Enterprise Importer, all user activity in the migrated repository (except Git commits) is attributed to placeholder identities called mannequins.

Each mannequin only has a display name, which comes from the display name in the source repository. Mannequins do not have organization membership or repository access. Mannequins always use the same avatar, a ghost octocat, and include a mannequin label following the display name.

You can reattribute the history for each mannequin to an organization member with the GitHub CLI or in your browser. If you use the GitHub CLI, you can reclaim mannequins in bulk.

Reclaiming is optional and can happen any time after a migration is finished. For this reason, you can allow your team to begin working in migrated repositories before reclaiming.

Mannequins are not used for Git commit attribution. For more details about commit attribution, see "[Managing authorship for commit commits](#)" below.

Reclaiming mannequins

You can reclaim mannequins with GitHub CLI (recommended) or the browser.

- [Reclaiming mannequins with the GitHub CLI (recommended)](#)
- [Reclaiming mannequins in your browser](#)

By default, reclaiming a mannequin will send an attribution invitation to the target user.

The target user can choose to accept or reject the invitation. After a user accepts an attribution invitation, all contributions previously attributed to the mannequin will be attributed to the user instead. In future migrations to the same organization, any contributions from the same mannequin will be automatically reclaimed for the same user.

If your organization uses Enterprise Managed Users and you choose to reclaim mannequins with the GitHub CLI, you can optionally skip the invitation process, immediately reclaiming the mannequin without the user's approval.

> **Notes:**
>
> - You cannot reclaim mannequins after you have transferred a repository to another organization. If you wish to transfer a repository to another organization after your migration, you must reclaim the mannequins before the transfer.
> - When reclaiming mannequins, you can only target existing organization members. Before attempting to reclaim a mannequin, verify that the GitHub user you want to invite is already added to the organization.

GitHub Enterprise Importer does not migrate user access to repositories. After reclaiming mannequins, if any of the users do not already have appropriate access to the repository via team membership, you must separately give the users access to the repository. For more information, see "Managing an individual's access to an organization repository."

Reclaiming mannequins with the GitHub CLI (recommended) 🔗

You can use the GitHub CLI to reclaim mannequins individually or in bulk. For more information about installing and updating migration extensions for the GitHub CLI, see "Migrating repositories with GitHub Enterprise Importer."

The exact command you need to use depends on which extension of the GitHub CLI that you're using.

- Reclaiming mannequins with the GEI extension
- Reclaiming mannequins with the ADO2GH extension

Reclaiming mannequins with the GEI extension 🔗

If your migration source is a GitHub product, you can reclaim mannequins with the GEI extension of the GitHub CLI.

If you don't already have a `GH_PAT` environment variable set for a personal access token with access to the destination organization, add `--github-target-pat TOKEN` to each of the commands below, replacing `TOKEN` with the personal access token. For personal access token requirements, see "Managing access for GitHub Enterprise Importer."

1. Optionally, to reclaim mannequins in bulk, create a CSV file that maps mannequins to organization members.

   - To generate a CSV file with a list of mannequins for an organization, use the `gh gei generate-mannequin-csv` command, replacing DESTINATION with the destination organization and FILENAME with a file name for the resulting CSV file.

     Optionally, to include mannequins that have already been reclaimed, add the `--include-reclaimed` flag.

     ```shell
     gh gei generate-mannequin-csv --github-target-org DESTINATION --output FILENAME.csv
     ```

- Edit the CSV file, adding the username of the organization member that corresponds to each mannequin.
- Save the file.

2. To reclaim mannequins, use the `gh gei reclaim-mannequin` command.

   - To reclaim mannequins in bulk with the mapping file you created earlier, replace DESTINATION with the destination organization and FILENAME with the file name of the mapping file.

     Shell

     ```shell
     gh gei reclaim-mannequin --github-target-org DESTINATION --csv
     FILENAME.csv
     ```

   - To reclaim an individual mannequin, replace DESTINATION with the destination organization, MANNEQUIN with the login of mannequin, and USERNAME with the username of the organization member that corresponds to the mannequin.

     If there are multiple mannequins with the same login, you can replace `--mannequin-user MANNEQUIN` with `--mannequin-ID ID`, replacing ID with the ID of the mannequin.

     If your organization uses Enterprise Managed Users and you want to skip the attribution invitation to reclaim the mannequin immediately, add the `--skip-invitation` argument.

     Shell

     ```shell
     gh gei reclaim-mannequin --github-target-org DESTINATION --mannequin-user MANNEQUIN --target-user USERNAME
     ```

By default, the organization member will receive an invitation via email, and the mannequin will not be reclaimed until the member accepts the invitation.

Reclaiming mannequins with the ADO2GH extension 🔗

If your migration source is Azure DevOps, you can reclaim mannequins with the ADO2GH extension of the GitHub CLI.

If you don't already have a `GH_PAT` environment variable set for a personal access token with access to the destination organization, add `--github-target-pat TOKEN` to each of the commands below, replacing `TOKEN` with the personal access token. For personal access token requirements, see "Managing access for GitHub Enterprise Importer."

1. Optionally, to reclaim mannequins in bulk, create a CSV file that maps mannequins to organization members.

   - To generate a CSV file with a list of mannequins for an organization, use the `gh ado2gh generate-mannequin-csv` command, replacing DESTINATION with the destination organization and FILENAME with a file name for the resulting CSV file.

     Optionally, to include mannequins that have already been reclaimed, add the `--include-reclaimed` flag.

```Shell
gh ado2gh generate-mannequin-csv --github-org DESTINATION --output
FILENAME.csv
```

- Edit the CSV file, adding the username of the organization member that corresponds to each mannequin.
- Save the file.

2. To reclaim mannequins, use the `gh ado2gh reclaim-mannequin` command.

- To reclaim mannequins in bulk with the mapping file you created earlier, replace DESTINATION with the destination organization and FILENAME with the file name of the mapping file.

```Shell
gh ado2gh reclaim-mannequin --github-org DESTINATION --csv FILENAME.csv
```

- To reclaim an individual mannequin, replace DESTINATION with the destination organization, MANNEQUIN with the login of mannequin, and USERNAME with the username of the organization member that corresponds to the mannequin.

  If there are multiple mannequins with the same login, you can replace `--mannequin-user MANNEQUIN` with `--mannequin-ID ID`, replacing ID with the ID of the mannequin.

  If your organization uses Enterprise Managed Users and you want to skip the attribution invitation to reclaim the mannequin immediately, add the `--skip-invitation` argument.
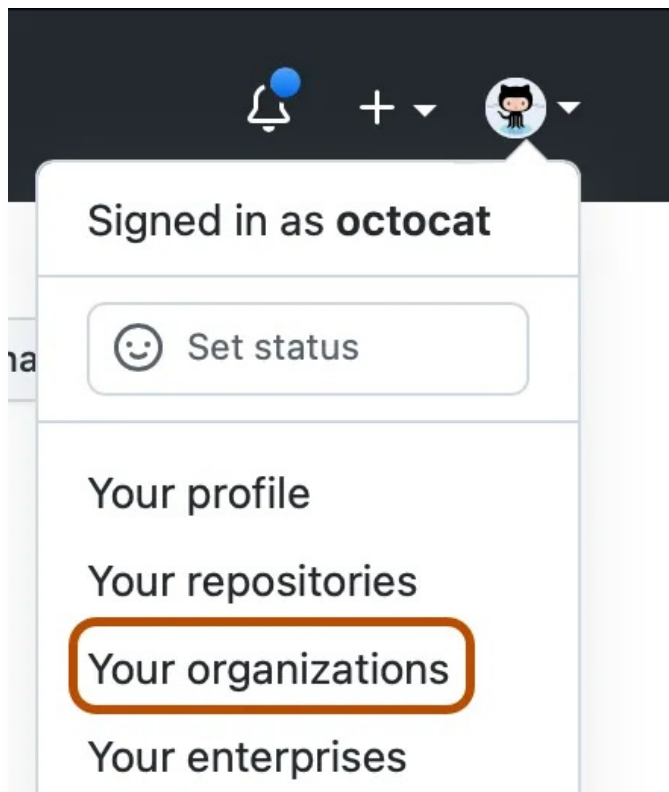
```Shell
gh ado2gh reclaim-mannequin --github-org DESTINATION --mannequin-user
MANNEQUIN --target-user USERNAME
```

By default, the organization member will receive an invitation via email, and the mannequin will not be reclaimed until the member accepts the invitation.

Reclaiming mannequins in your browser 🔗

1. In the top right corner of GitHub Enterprise Server, click your profile photo, then click **Your organizations**.

2. Next to the organization, click **Settings**.

3. In the "Access" section of the sidebar, click ⇄ **Import/Export**.

4. To the right of the mannequin you want to reclaim, click **Reattribute**.

5. In the search field, type the username of the organization member you want to attribute the mannequin's contributions to, then click the member.

> **Note:** You can only send attribution invitations to user accounts that are already members of the organization.
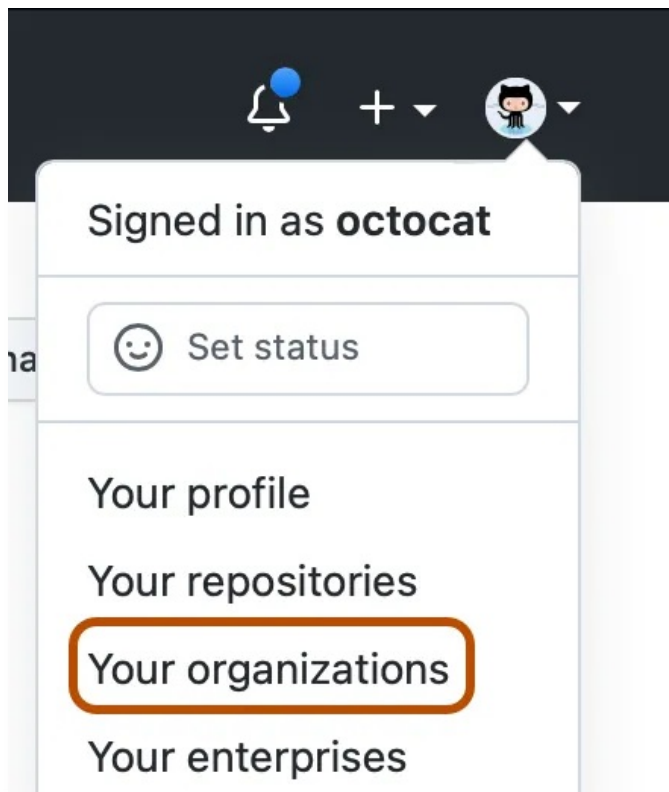
6. Click **Invite**. By default, the organization member will receive an invitation via email, and the mannequin will not be reclaimed until the member accepts the invitation.

Viewing the status of your attribution invitations 🔗

You can view the status of all attribution invitations for your organization.

- Invited: The user has been sent an invitation, but has not replied to the invitation yet.
- Completed: The user has accepted, or the invitation process was skipped. The user's contributions have been reattributed.
- Rejected: The user chose not to be credited for the mannequin's contributions.

1. In the top right corner of GitHub Enterprise Server, click your profile photo, then click **Your organizations**.

2. Next to the organization, click **Settings**.

3. In the "Access" section of the sidebar, click ⇄ **Import/Export**.

4. Under "⇄ Import/Export", click **Attribution Invitations**.

Managing authorship for Git commits 🔗

Authorship for Git commits is not associated with mannequins and cannot be attributed to GitHub users by reclaiming mannequins. Instead, commit authorship is attributed to user accounts on GitHub based on the email address that was used to author the commit in Git.

In many cases, users can reattribute commits to themselves by adding the email address used to author the commit to their user account on GitHub. For more information, see "Adding an email address to your GitHub account."

However, if you use Enterprise Managed Users, users cannot add email addresses to their user account on GitHub and will therefore not be able to reattribute Git commits. Only commits authored by a user's primary email address in your identity provider (IdP) will be attributed to managed user accounts.

Additionally, commits authored by a GitHub-provided `noreply` email address cannot be reattributed, because you can't manually add a `noreply` email address to a user account. For more information, see "Setting your commit email address."