

# Importing from other version control systems with the administrative shell

## In this article

Importing projects from Mercurial

Importing projects from Subversion

Importing projects from Team Foundation Version Control

Further reading

Using the administrative shell, you can import from Subversion, Mercurial and Team Foundation Version Control to Git repositories on GitHub Enterprise Server.

### Who can use this feature

Site administrators can use the administrative shell to import data from other version control systems.

## Importing projects from Mercurial

- 1 SSH into your GitHub Enterprise Server instance. If your instance comprises multiple nodes, for example if high availability or geo-replication are configured, SSH into the primary node. If you use a cluster, you can SSH into any node. For more information about SSH access, see "[Accessing the administrative shell \(SSH\)](#)."

```
ssh -p 122 admin@HOSTNAME
```

- 2 Make a raw clone of the project using the command below, specifying the URL of the source project, and a path to a temporary repository:

```
$ git-import-hg-raw HG-CLONE-URL/PATH/REPO-NAME.git
# Creates a new repository with one or more Git refs in "refs/import/" in
the specified path.
```

- 3 Review the comma-separated (CSV) file in `/PATH/REPO-NAME.git/git-import/raw-authors.csv`. It should contain these columns:

- `ID`: The author as stored in the original repository, followed by a unique identifier
- `NAME`: The author as stored in the original repository

To map authors from the original repository to an email address and name, create a new CSV file with the columns `ID`, `(ignored)`, `GIT_EMAIL`, `GIT_NAME`. You can use this CSV file to replace the author information for anything by "ID" with "GIT\_EMAIL" and "GIT\_NAME".

For example, if you want to map the original author ID of `octocat@111111-2222-3333-`

4444-55555555555 to a new user with the email address of `octocat@github.com` and name of `The Octocat`, the CSV file should include the line:

```
octocat@111111-2222-3333-4444-55555555555, ,octocat@github.com,The Octocat
```

- 4 Rewrite the authors and branches using the CSV file:

```
git-import-rewrite --flavor hg --authors /PATH/AUTHORS-MAP-FILE.csv  
/PATH/REPO-NAME.git
```

- 5 If you haven't yet, [create a new empty repository on GitHub Enterprise Server](#).
- 6 Change the current working directory to your local repository.

- 7 Push the imported repository to GitHub Enterprise Server:

```
git push --mirror PUSH-URL-ON-GITHUB-ENTERPRISE
```

## Importing projects from Subversion

- 1 SSH into your GitHub Enterprise Server instance. If your instance comprises multiple nodes, for example if high availability or geo-replication are configured, SSH into the primary node. If you use a cluster, you can SSH into any node. For more information about SSH access, see "[Accessing the administrative shell \(SSH\)](#)."

```
ssh -p 122 admin@HOSTNAME
```

- 2 Make a raw clone of the project using the command below, specifying the URL of the source project, and a path to a temporary repository:

```
$ git-import-svn-raw SVN-CLONE-URL /PATH/REPO-NAME.git  
# Creates a new repository with one or more Git refs in "refs/import/" in  
the specified path.
```

- 3 Review the comma-separated (CSV) file in `/PATH/REPO-NAME.git/git-import/raw-authors.csv`. It should contain these columns:

- `ID`: The author as stored in the original repository, followed by a unique identifier
- `NAME`: The author as stored in the original repository

To map authors from the original repository to an email address and name, create a new CSV file with the columns `ID`, (ignored), `GIT_EMAIL`, `GIT_NAME`. You can use this CSV file to replace the author information for anything by "ID" with "GIT\_EMAIL" and "GIT\_NAME".

For example, if you want to map the original author ID of `octocat@111111-2222-3333-4444-55555555555` to a new user with the email address of `octocat@github.com` and name of `The Octocat`, the CSV file should include the line:

```
octocat@111111-2222-3333-4444-55555555555, ,octocat@github.com,The Octocat
```

- 4 Rewrite the authors and branches using the CSV file:

```
git-import-rewrite --flavor svn --authors /PATH/AUTHORS-MAP-FILE.csv
```

```
/PATH/REPO-NAME.git
```

- 5 If you haven't yet, [create a new empty repository on GitHub Enterprise Server](#).
- 6 Change the current working directory to your local repository.
- 7 Push the imported repository to GitHub Enterprise Server:

```
git push --mirror PUSH-URL-ON-GITHUB-ENTERPRISE
```

## Importing projects from Team Foundation Version Control

- 1 SSH into your GitHub Enterprise Server instance. If your instance comprises multiple nodes, for example if high availability or geo-replication are configured, SSH into the primary node. If you use a cluster, you can SSH into any node. For more information about SSH access, see "[Accessing the administrative shell \(SSH\)](#)."

```
ssh -p 122 admin@HOSTNAME
```

- 2 Make a raw clone of the project using the command below, specifying the URL of the source project, and a path to a temporary repository:

```
$ git-import-tfs-raw TEAM-FOUNDATION-CLONE-URL /PATH/REPO-NAME.git
# Creates a new repository with one or more Git refs in "refs/import/" in
the specified path.
```

- 3 Review the comma-separated (CSV) file in `/PATH/REPO-NAME.git/git-import/raw-authors.csv` . It should contain these columns:

- `ID` : The author as stored in the original repository, followed by a unique identifier
- `NAME` : The author as stored in the original repository

To map authors from the original repository to an email address and name, create a new CSV file with the columns `ID`, (ignored), `GIT_EMAIL`, `GIT_NAME` . You can use this CSV file to replace the author information for anything by "ID" with "GIT\_EMAIL" and "GIT\_NAME".

For example, if you want to map the original author ID of `octocat@111111-2222-3333-4444-555555555555` to a new user with the email address of `octocat@github.com` and name of `The Octocat` , the CSV file should include the line:

```
octocat@111111-2222-3333-4444-555555555555, ,octocat@github.com,The Octocat
```

- 4 Rewrite the authors and branches using the CSV file:

```
git-import-rewrite --flavor tfs --authors /PATH/AUTHORS-MAP-FILE.csv
/PATH/REPO_NAME.git
```

- 5 If you haven't yet, [create a new empty repository on GitHub Enterprise Server](#).
- 6 Change the current working directory to your local repository.

- 7 Push the imported repository to GitHub Enterprise Server:

```
git push --mirror PUSH-URL-ON-GITHUB-ENTERPRISE
```

## Further reading

---

- "[Command-line utilities](#)"

### Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)