Supplemental arguments and settings

In this article

GitHub Actions Importer has several supplemental arguments and settings to tailor the migration process to your needs.

Legal notice

This article provides general information for configuring GitHub Actions Importer's supplemental arguments and settings, such as optional parameters, path arguments, and network settings.

Optional parameters

GitHub Actions Importer has several optional parameters that you can use to customize the migration process.

Limiting allowed actions

The following options can be used to limit which actions are allowed in converted workflows. When used in combination, these options expand the list of allowed actions. If none of these options are supplied, then all actions are allowed.

- `--allowed-actions` specifies a list of actions to allow in converted workflows. Wildcards are supported. Any other actions other than those provided will be disallowed.

  For example:

  ```
  --allowed-actions actions/checkout@v4 actions/upload-artifact@* my-org/*
  ```

  You can provide an empty list to disallow all actions. For example, `--allowed-actions=`.

- `--allow-verified-actions` specifies that all actions from verified creators are allowed.

- `--allow-github-created-actions` specifies that actions published from the `github` or `actions` organizations are allowed.

  For example, such actions include `github/super-linter` and `actions/checkout`.

  This option is equivalent to `--allowed-actions actions/* github/*`.

Using a credentials file for authentication

The `--credentials-file` parameter specifies the path to a file containing credentials for different servers that GitHub Actions Importer can authenticate to. This is useful when build scripts (such as `.travis.yml` or `jenkinsfile`) are stored in multiple GitHub Enterprise Server instances.

A credentials file must be a YAML file containing a list of server and access token combinations. GitHub Actions Importer uses the credentials for the URL that most closely matches the network request being made.

For example:

```
- url: https://github.com
  access_token: ghp_mygeneraltoken
- url: https://github.com/specific_org/
  access_token: ghp_myorgspecifictoken
- url: https://jenkins.org
  access_token: abc123
  username: marty_mcfly
```

For the above credentials file, GitHub Actions Importer uses the access token `ghp_mygeneraltoken` to authenticate all network requests to `https://github.com`, *unless* the network request is for a repository in the `specific_org` organization. In that case, the `ghp_myorgspecifictoken` token is used to authenticate instead.

Alternative source code providers

GitHub Actions Importer can automatically fetch source code from non-GitHub repositories. A credentials file can specify the `provider`, the provider URL, and the credentials needed to retrieve the source code.

For example:

```
- url: https://gitlab.com
  access_token: super_secret_token
  provider: gitlab
```

For the above example, GitHub Actions Importer uses the token `super_secret_token` to retrieve any source code that is hosted on `https://gitlab.com`.

Supported values for `provider` are:

- `github` (default)
- `gitlab`
- `bitbucket_server`
- `azure_devops`

Controlling optional features

You can use the `--features` option to limit the features used in workflows that GitHub Actions Importer creates. This is useful for excluding newer GitHub Actions syntax from workflows when migrating to an older GitHub Enterprise Server instance. When using the `--features` option, you must specify the version of GitHub Enterprise Server that you are migrating to.

For example:

```
gh actions-importer dry-run ... --features ghes-3.3
```

The supported values for `--features` are:

- `all` (default value)
- `ghes-latest`
- `ghes-<number>`, where `<number>` is the version of GitHub Enterprise Server, `3.0` or later. For example, `ghes-3.3`.

You can view the list of available feature flags by GitHub Actions Importer by running the `list-features` command. For example:

Shell

```
gh actions-importer list-features
```

You should see an output similar to the following.

```
Available feature flags:

actions/cache (disabled):
        Control usage of actions/cache inside of workflows. Outputs a comment if
not enabled.
        GitHub Enterprise Server >= ghes-3.5 required.

composite-actions (enabled):
        Minimizes resulting workflow complexity through the use of composite
actions. See https://docs.github.com/en/actions/creating-actions/creating-a-
composite-action for more information.
        GitHub Enterprise Server >= ghes-3.4 required.

reusable-workflows (disabled):
        Avoid duplication by re-using existing workflows. See
https://docs.github.com/en/actions/using-workflows/reusing-workflows for more
information.
        GitHub Enterprise Server >= ghes-3.4 required.

workflow-concurrency-option-allowed (enabled):
        Allows the use of the `concurrency` option in workflows. See
https://docs.github.com/en/actions/reference/workflow-syntax-for-github-
actions#concurrency for more information.
        GitHub Enterprise Server >= ghes-3.2 required.

Enable features by passing --enable-features feature-1 feature-2
Disable features by passing --disable-features feature-1 feature-2
```

To toggle feature flags, you can use either of the following methods:

- Use the `--enable-features` and `--disable-features` options when running a `gh actions-importer` command.
- Use an environment variable for each feature flag.

You can use the `--enable-features` and `--disable-features` options to select specific features to enable or disable for the duration of the command. For example, the following command disables use of `actions/cache` and `composite-actions`:

```
gh actions-importer dry-run ... --disable-features=composite-actions
actions/cache
```

You can use the `configure --features` command to interactively configure feature flags and automatically write them to your environment:

```
$ gh actions-importer configure --features
```

```
✔ Which features would you like to configure?: actions/cache, reusable-workflows
✔ actions/cache (disabled): Enable
? reusable-workflows (disabled):
› Enable
  Disable
```

Disabling network response caching

By default, GitHub Actions Importer caches responses from network requests to reduce network load and reduce run time. You can use the `--no-http-cache` option to disable the network cache. For example:

```
gh actions-importer forecast ... --no-http-cache
```

Path arguments

When running GitHub Actions Importer, path arguments are relative to the container's disk, so absolute paths relative to the container's host machine are not supported. When GitHub Actions Importer is run, the container's `/data` directory is mounted to the directory where GitHub Actions Importer is run.

For example, the following command, when used in the `/Users/mona` directory, outputs the GitHub Actions Importer audit summary to the `/Users/mona/out` directory:

```
gh actions-importer audit --output-dir /data/out
```

Using a proxy

To access servers that are configured with a HTTP proxy, you must set the following environment variables with the proxy's URL:

- `OCTOKIT_PROXY`: for any GitHub server.
- `HTTP_PROXY` (or `HTTPS_PROXY`): for any other servers.

For example:

```
export OCTOKIT_PROXY=https://proxy.example.com:8443
export HTTPS_PROXY=$OCTOKIT_PROXY
```

If the proxy requires authentication, a username and password must be included in the proxy URL. For example, `https://username:password@proxy.url:port`.

Disabling SSL certificate verification

By default, GitHub Actions Importer verifies SSL certificates when making network requests. You can disable SSL certificate verification with the `--no-ssl-verify` option. For example:

```
gh actions-importer audit --output-dir ./output --no-ssl-verify
```

Legal notice

Portions have been adapted from https://github.com/github/gh-actions-importer/ under

the MIT license:

```
MIT License

Copyright (c) 2022 GitHub

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```