

Contributing to govmomi

Getting started

First, fork the repository on GitHub to your personal account.

Note that *GOPATH* can be any directory, the example below uses *\$HOME/govmomi*. Change *\$USER* below to your github username if they are not the same.

```
export GOPATH=$HOME/govmomi
go get github.com/vmware/govmomi
cd $GOPATH/src/github.com/vmware/govmomi
git config push.default nothing # anything to avoid pushing to vmware/govmomi by default
git remote rename origin vmware
git remote add $USER git@github.com:$USER/govmomi.git
git fetch $USER
```

Installing from source

Compile the govmomi libraries and install govc using:

```
go install -v github.com/vmware/govmomi/govc
```

Note that **govc/build.sh** is only used for building release binaries.

Contribution flow

This is a rough outline of what a contributor's workflow looks like:

- Create a topic branch from where you want to base your work.
- Make commits of logical units.
- Make sure your commit messages are in the proper format (see below).
- Update CHANGELOG.md and/or govc/CHANGELOG.md when appropriate.
- Push your changes to a topic branch in your fork of the repository.
- Submit a pull request to vmware/govmomi.

Example:

```
git checkout -b my-new-feature vmware/master
git commit -a
git push $USER my-new-feature
```

Stay in sync with upstream

When your branch gets out of sync with the vmware/master branch, use the following to update:

```
git checkout my-new-feature
git fetch -a
```

```
git rebase vmware/master
git push --force-with-lease $USER my-new-feature
```

Updating pull requests

If your PR fails to pass CI or needs changes based on code review, you'll most likely want to squash these changes into existing commits.

If your pull request contains a single commit or your changes are related to the most recent commit, you can simply amend the commit.

```
git add .
git commit --amend
git push --force-with-lease $USER my-new-feature
```

If you need to squash changes into an earlier commit, you can use:

```
git add .
git commit --fixup <commit>
git rebase -i --autosquash vmware/master
git push --force-with-lease $USER my-new-feature
```

Be sure to add a comment to the PR indicating your new changes are ready to review, as github does not generate a notification when you git push.

Code style

The coding style suggested by the Golang community is used in govmomi. See the [style doc](#) for details.

Try to limit column width to 120 characters for both code and markdown documents such as this one.

Format of the Commit Message

We follow the conventions on [How to Write a Git Commit Message](#).

Be sure to include any related GitHub issue references in the commit message.

Reporting Bugs and Creating Issues

When opening a new issue, try to roughly follow the commit message format conventions above.