

## **Cobra User Contract**

### **Versioning**

Cobra will follow a steady release cadence. Non breaking changes will be released as minor versions quarterly. Patch bug releases are at the discretion of the maintainers. Users can expect security patch fixes to be released within relatively short order of a CVE becoming known. For more information on security patch fixes see the CVE section below. Releases will follow Semantic Versioning. Users tracking the Master branch should expect unpredictable breaking changes as the project continues to move forward. For stability, it is highly recommended to use a release.

### **Backward Compatibility**

We will maintain two major releases in a moving window. The N-1 release will only receive bug fixes and security updates and will be dropped once N+1 is released.

### **Deprecation**

Deprecation of Go versions or dependent packages will only occur in major releases. To reduce the change of this taking users by surprise, any large deprecation will be preceded by an announcement in the #cobra slack channel and an Issue on Github.

### **CVE**

Maintainers will make every effort to release security patches in the case of a medium to high severity CVE directly impacting the library. The speed in which these patches reach a release is up to the discretion of the maintainers. A low severity CVE may be a lower priority than a high severity one.

### **Communication**

Cobra maintainers will use GitHub issues and the #cobra slack channel as the primary means of communication with the community. This is to foster open communication with all users and contributors.

### **Breaking Changes**

Breaking changes are generally allowed in the master branch, as this is the branch used to develop the next release of Cobra.

There may be times, however, when master is closed for breaking changes. This is likely to happen as we near the release of a new version.

Breaking changes are not allowed in release branches, as these represent minor versions that have already been released. These version have consumers who

expect the APIs, behaviors, etc, to remain stable during the lifetime of the patch stream for the minor release.

Examples of breaking changes include: - Removing or renaming exported constant, variable, type, or function. - Updating the version of critical libraries such as `spf13/pflag`, `spf13/viper` etc... - Some version updates may be acceptable for picking up bug fixes, but maintainers must exercise caution when reviewing.

There may, at times, need to be exceptions where breaking changes are allowed in release branches. These are at the discretion of the project's maintainers, and must be carefully considered before merging.

### **CI Testing**

Maintainers will ensure the Cobra test suite utilizes the current supported versions of Golang.

### **Disclaimer**

Changes to this document and the contents therein are at the discretion of the maintainers. None of the contents of this document are legally binding in any way to the maintainers or the users.