

# Cabal User Guide

- [Reporting bugs and deficiencies](#)
- [Stability of Cabal interfaces](#)
  - [Cabal file format](#)
  - [Command-line interface](#)
    - [Very Stable Command-line interfaces](#)
    - [Stable Command-line interfaces](#)
    - [Unstable command-line](#)
  - [Functions and Types](#)
    - [Very Stable API](#)
    - [Semi-stable API](#)
    - [Unstable API](#)
  - [Hackage](#)

## Reporting bugs and deficiencies

Please report any flaws or feature requests in the [bug tracker](#).

For general discussion or queries email the libraries mailing list [libraries@haskell.org](mailto:libraries@haskell.org). There is also a development mailing list [cabal-devel@haskell.org](mailto:cabal-devel@haskell.org).

## Stability of Cabal interfaces

The Cabal library and related infrastructure is still under active development. New features are being added and limitations and bugs are being fixed. This requires internal changes and often user visible changes as well. We therefore cannot promise complete future-proof stability, at least not without halting all development work.

This section documents the aspects of the Cabal interface that we can promise to keep stable and which bits are subject to change.

### Cabal file format

This is backwards compatible and mostly forwards compatible. New fields can be added without breaking older versions of Cabal. Fields can be deprecated without breaking older packages.

### Command-line interface

#### Very Stable Command-line interfaces

- `./setup configure`
- `--prefix`
- `--user`
- `--ghc, --uhc`
- `--verbose`
- `--prefix`
- `./setup build`
- `./setup install`
- `./setup register`
- `./setup copy`

## Stable Command-line interfaces

### Unstable command-line

## Functions and Types

The Cabal library follows the [Package Versioning Policy](#). This means that within a stable major release, for example 1.2.x, there will be no incompatible API changes. But minor versions increments, for example 1.2.3, indicate compatible API additions.

The Package Versioning Policy does not require any API guarantees between major releases, for example between 1.2.x and 1.4.x. In practise of course not everything changes between major releases. Some parts of the API are more prone to change than others. The rest of this section gives some informal advice on what level of API stability you can expect between major releases.

### Very Stable API

- `defaultMain`
- `defaultMainWithHooks defaultUserHooks`

But regular `defaultMainWithHooks` isn't stable since `UserHooks` changes.

### Semi-stable API

- `UserHooks` The hooks API will change in the future
- `Distribution.*` is mostly declarative information about packages and is somewhat stable.

### Unstable API

Everything under `Distribution.Simple.*` has no stability guarantee.

## Hackage

The index format is a partly stable interface. It consists of a `tar.gz` file that contains directories with `.cabal` files in. In future it may contain more kinds of files so do not assume every file is a `.cabal` file. Incompatible revisions to the format would involve bumping the name of the index file, i.e., `00-index.tar.gz`, `01-index.tar.gz` etc.