Generating Zsh Completion For Your cobra.Command

Please refer to Shell Completions for details.

Zsh completions standardization

Cobra 1.1 standardized its zsh completion support to align it with its other shell completions. Although the API was kept backwards-compatible, some small changes in behavior were introduced.

Deprecation summary

See further below for more details on these deprecations.

- `cmd.MarkZshCompPositionalArgumentFile(pos, []string{})` is no longer needed. It is therefore **deprecated** and silently ignored.
- `cmd.MarkZshCompPositionalArgumentFile(pos, glob[])` is **deprecated** and silently ignored.
  - Instead use `ValidArgsFunction` with `ShellCompDirectiveFilterFileExt`.
- `cmd.MarkZshCompPositionalArgumentWords()` is **deprecated** and silently ignored.
  - Instead use `ValidArgsFunction`.

Behavioral changes

Noun completion |Old behavior|New behavior| |—|—| |No file completion by default (opposite of bash)|File completion by default; use `ValidArgsFunction` with `ShellCompDirectiveNoFileComp` to turn off file completion on a per-argument basis| |Completion of flag names without the `-` prefix having been typed|Flag names are only completed if the user has typed the first `-`| `cmd.MarkZshCompPositionalArgumentFile(pos, []string{})` used to turn on file completion on a per-argument position basis|File completion for all arguments by default; `cmd.MarkZshCompPositionalArgumentFile()` is **deprecated** and silently ignored| |`cmd.MarkZshCompPositionalArgumentFile(pos, glob[])` used to turn on file completion **with glob filtering** on a per-argument position basis (zsh-specific)|`cmd.MarkZshCompPositionalArgumentFile()` is **deprecated** and silently ignored; use `ValidArgsFunction` with `ShellCompDirectiveFilterFileExt` for file **extension** filtering (not full glob filtering)| |`cmd.MarkZshCompPositionalArgumentWords(pos, words[])` used to provide completion choices on a per-argument position basis (zsh-specific)|`cmd.MarkZshCompPositionalArgumentWords()` is **deprecated** and silently ignored; use `ValidArgsFunction` to achieve the same behavior|

Flag-value completion

Old behavior	New behavior
No file completion by default (opposite of bash)	File completion by default; use `RegisterFlagCompletionFunc()` with `ShellCompDirectiveNoFileComp` to turn off file completion
`cmd.MarkFlagFilename(flag, []string{})` and similar used to turn on file completion	File completion by default; `cmd.MarkFlagFilename(flag, []string{})` no longer needed in this context and silently ignored
`cmd.MarkFlagFilename(flag, glob[])` used to turn on file completion **with glob filtering** (syntax of `[]string{"*.yaml", "*.yml"}` incompatible with bash)	Will continue to work, however, support for bash syntax is added and should be used instead so as to work for all shells (`[]string{"yaml", "yml"}`)
`cmd.MarkFlagDirname(flag)` only completes directories (zsh-specific)	Has been added for all shells
Completion of a flag name does not repeat, unless flag is of type `*Array` or `*Slice` (not supported by bash)	Retained for `zsh` and added to `fish`
Completion of a flag name does not provide the `=` form (unlike bash)	Retained for `zsh` and added to `fish`

Improvements

- Custom completion support (`ValidArgsFunction` and `RegisterFlagCompletionFunc()`)
- File completion by default if no other completions found
- Handling of required flags
- File extension filtering no longer mutually exclusive with bash usage
- Completion of directory names *within* another directory
- Support for `=` form of flags