← keytyapp/Keyty

How can I reset my settings to default? Uninstalling and reinstalling doesn't work.

Fast View as codemap

Why reinstalling doesn't reset settings

Keyty persists all settings through `UserDefaultsStore`, which wraps `UserDefaults.standard` KeyboardVisualizerSettings.swift:71-73. `UserDefaults` data is stored outside the app bundle (in `~/Library/Preferences`), so deleting/reinstalling the app does not clear it — this is standard macOS behavior, not a bug in Keyty's code.

What I found in the codebase

I did not find any explicit "Reset to Defaults" button/action in the provided context — only `registerDefaults()` methods that **register** default values (used for `UserDefaults.register`, which only supplies fallback values and does not overwrite existing stored values) KeyboardVisualizerSettings.swift:75-77. This is confirmed by tests that check registered defaults land in the store KeyboardVisualizerSettingsTests.swift:29-50.

Because there's no visible in-app reset feature, the practical workaround is to manually clear the app's `UserDefaults` domain, e.g. via Terminal:

```
defaults delete <bundle-identifier>
```

(replace `<bundle-identifier>` with Keyty's actual bundle ID, which I don't have visibility into from this index).

Notes

- The wiki page on Settings Persistence confirms the architecture: settings use `@Stored` property wrapper over `KeyValueStore` / `UserDefaultsStore`, and `InMemoryKeyValueStore` is only used in tests, not production.
- I could not find the app's bundle identifier or any dedicated "reset settings" UI/action in the indexed files. If such a feature exists, it may be in a file excluded by index size limits — starting a full Devin session would let me search the complete codebase (e.g. `AppSettings`, settings panes under `Features/Settings`) for a reset action.

Wiki pages you might want to explore:

- Settings Persistence (KeyValueStore) (keytyapp/Keyty)

keytyapp/Keyty Apps/Keyty/Sources/Keyty/Features/Visualizers/Keyboard/KeyboardVisualizerSettings.swift

```
68          self.placementChangesSubject.eraseToAnyPublisher()
69      }
70
71      init(store: KeyValueStore = UserDefaultsStore()) {
72          self.store = store
73      }
74
75      func registerDefaults() {
76          self.registerStoredDefaults()
77      }
78
79      func applyCustomPlacement(screenID: CGDirectDisplayID, normalizedX: CGFloat, normalizedY: CGFloat) {
80          self.screenID = screenID
```

keytyapp/Key… Apps/Keyty/Tests/KeytyTests/Features/Visualizers/Keyboard/Keycaps/KeyboardVisualizerSettingsTests....

```
26          super.tearDown()
27      }
28
29      func testRegistersDefaults() {
30          self.settings.registerDefaults()
31
32          XCTAssertEqual(self.store.bool(forKey: KeyboardVisualizerSettingsKeys.isEnabled), true)
33          XCTAssertEqual(self.store.integer(forKey: KeyboardVisualizerSettingsKeys.axis), KeyboardVisualizerStackA
34          XCTAssertEqual(self.store.integer(forKey: KeyboardVisualizerSettingsKeys.maxCount), KeyboardVisualizerSe
35          XCTAssertEqual(self.store.double(forKey: KeyboardVisualizerSettingsKeys.fadeDelay), 2.0, accuracy: 0.000
36          XCTAssertEqual(self.store.double(forKey: KeyboardVisualizerSettingsKeys.fadeDuration), 0.2, accuracy: 0.
37          XCTAssertEqual(self.store.integer(forKey: KeyboardVisualizerSettingsKeys.theme), KeyboardVisualizerTheme
38          XCTAssertEqual(self.store.bool(forKey: KeyboardVisualizerSettingsKeys.usesCustomThemePalette), false)
39          XCTAssertEqual(self.store.integer(forKey: KeyboardVisualizerSettingsKeys.style), KeycapStyle.apple.rawVa
40          XCTAssertEqual(self.store.double(forKey: KeyboardVisualizerSettingsKeys.scale), 1.0, accuracy: 0.0001)
41          XCTAssertEqual(self.store.integer(forKey: KeyboardVisualizerSettingsKeys.placementMode), KeyboardVisuali
42          XCTAssertEqual(self.store.double(forKey: KeyboardVisualizerSettingsKeys.customPositionNormalizedX), 0.5,
43          XCTAssertEqual(self.store.double(forKey: KeyboardVisualizerSettingsKeys.customPositionNormalizedY), 0.5,
44          XCTAssertEqual(self.store.integer(forKey: KeyboardVisualizerSettingsKeys.customHorizontalAlignment), Key
45          XCTAssertEqual(self.store.double(forKey: KeyboardVisualizerSettingsKeys.windowPadding), Double(Size.Keyb
46          XCTAssertEqual(self.store.bool(forKey: KeyboardVisualizerSettingsKeys.onlyShowModifiedKeystrokes), false
47          XCTAssertEqual(self.store.bool(forKey: KeyboardVisualizerSettingsKeys.showSpecialKeys), true)
48          XCTAssertEqual(self.store.bool(forKey: KeyboardVisualizerSettingsKeys.showMediaKeyButtons), true)
49          XCTAssertEqual(self.store.bool(forKey: KeyboardVisualizerSettingsKeys.showMouseEvents), true)
50      }
51
52      func testSharedKeyboardVisualizerSettingsFallbackAndClamping() {
```

Ask a follow-up question

Fast