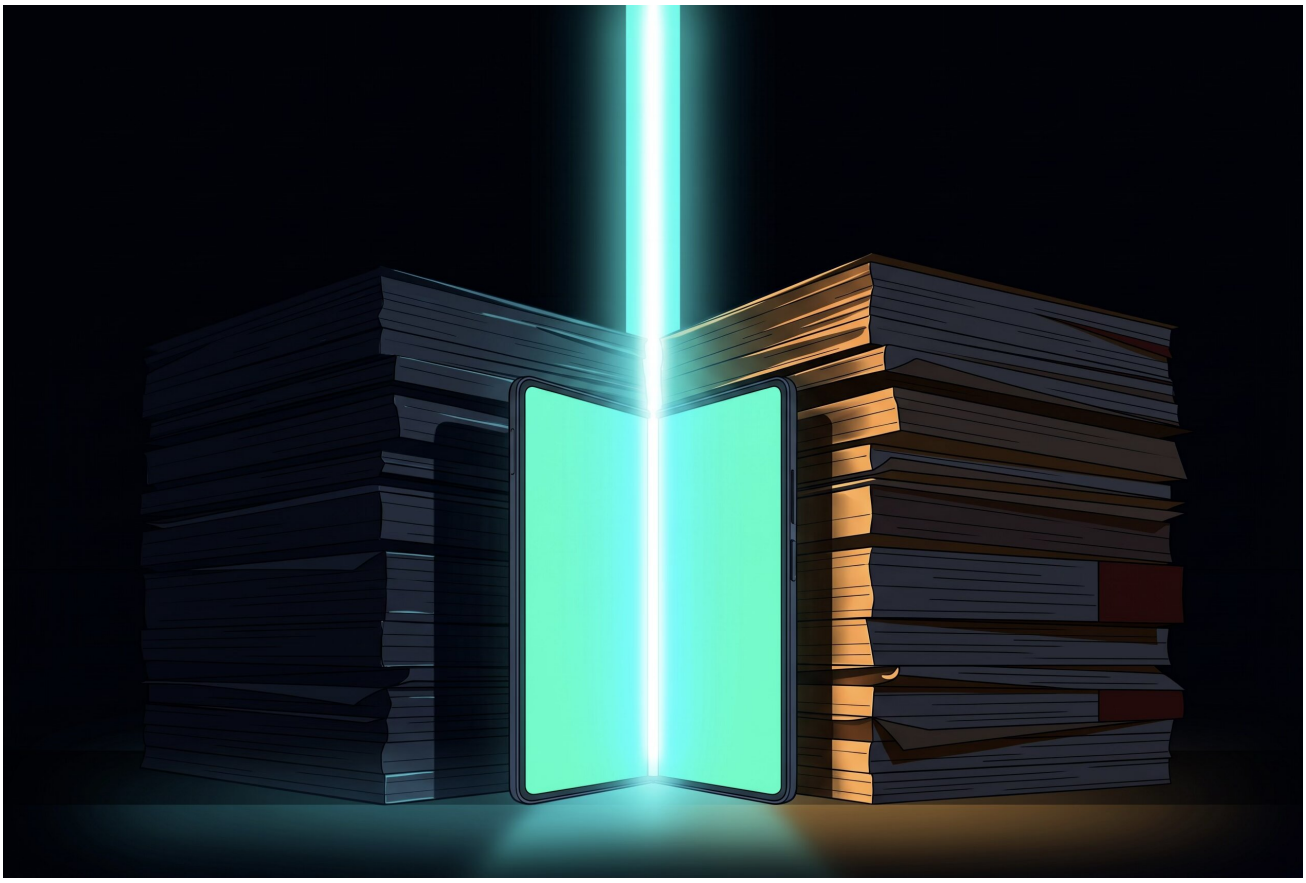


Cupertino 1.5.0: Five Months of Apple Documentation, Diffed

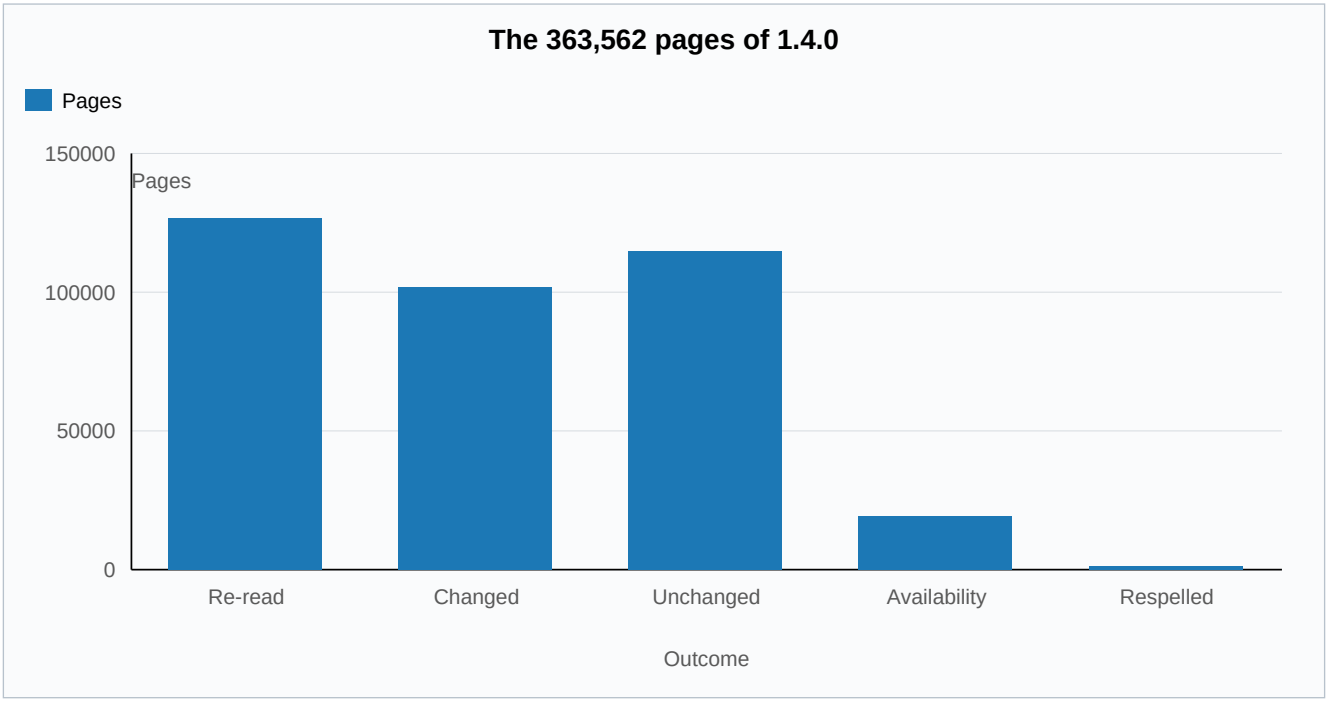
In June I diffed Apple's documentation against the version that had been on everyone's disk since May, and read the difference like a tea leaf. This time I diffed my own database against itself. The 1.4.0 database holds Apple's documentation as it was on 2026-06-20. The 1.5.0 database was built on 2026-11-09. Every page has an address, so I matched the two by address, compared the stored content, and counted.

The short version: **363,562 pages became 391,056, and no page was lost.** The longer version has a surprise in it, two defects, and a foldable iPhone.

What happened to the 363,562 pages

Of the pages in 1.4.0, 362,474 have the same address in 1.5.0, and 1,088 have an address that differs only in spelling: case, underscores, hyphens, a number prefix. Those are the same pages, so I counted them as neither new nor removed. Removed: zero.

New: 27,494 pages, 7.6 percent more than before. But the new pages are not the main story. Look at what happened to the pages that were already there.

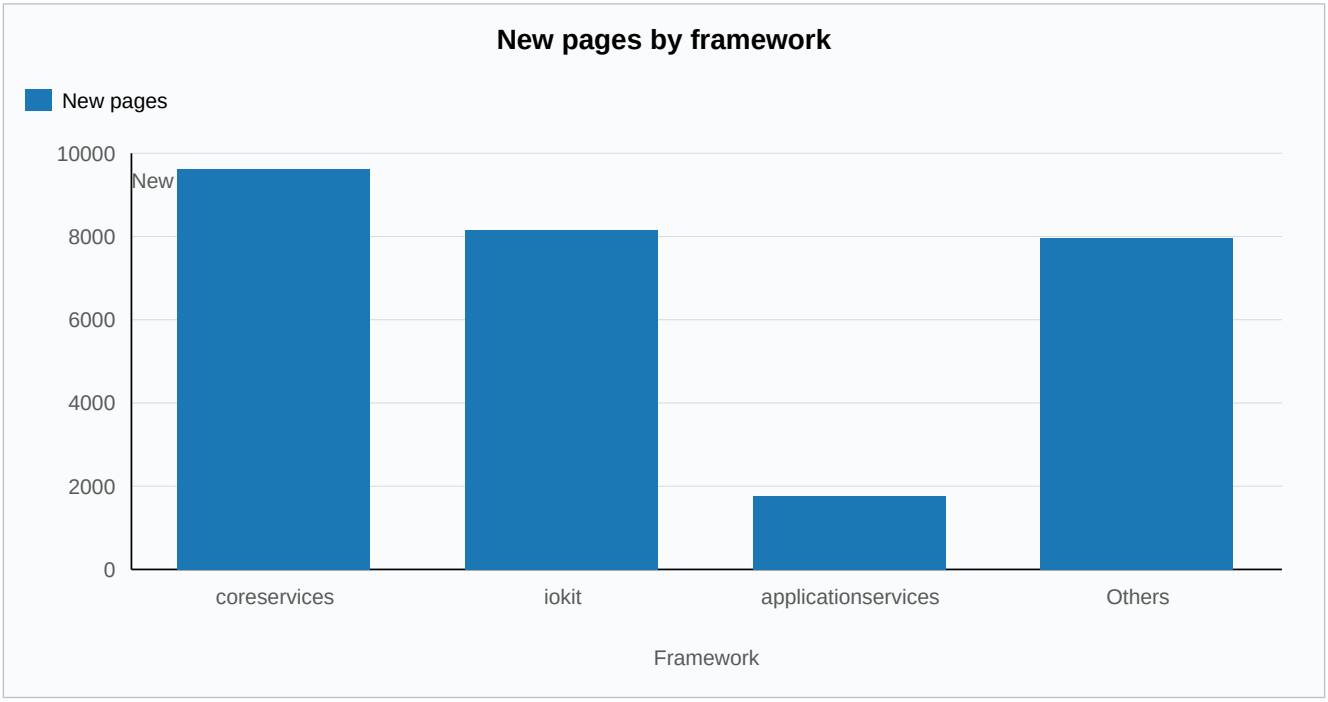


The labels, in full. Re-read: stored from Apple's structured data. Changed: the content differs. Unchanged: the same. Availability: only the minimum versions differ. Respelled: the same page under a new spelling of its address.

Only 114,869 pages, about a third, are unchanged. For 126,695 of them the content differs because they were stored in a different way. More on that below.

Where the 27,494 new pages went

Three frameworks hold 71 percent of them: `coreservices` (9,613 pages), `iokit` (8,147) and `applicationservices` (1,767). These are older, C-based system frameworks.



Seven frameworks are new in the database: `apple-ads-platform-api` (755 pages),

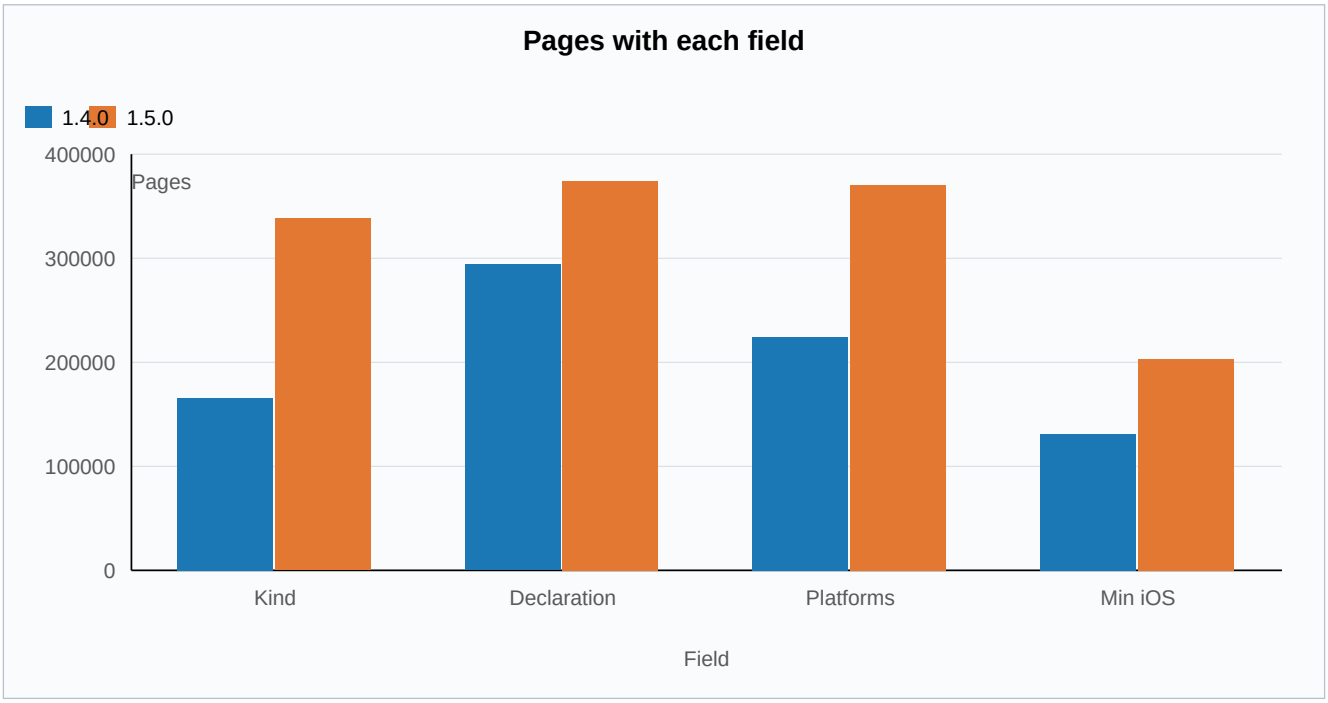
`videodriverkit` (563), `sirievaluations` (144), `homeextensions` (134), `app-store-review-guidelines` (129), `appletvfeed` (77) and `visionentitlementservices` (36). The database now holds 424 frameworks.

The real change is how pages are stored

In 1.4.0 most pages were stored from a scrape of the web page. In 1.5.0, 126,695 of them, 35 percent of the old database, are stored from Apple's structured data instead. A scrape gives you text with the site's navigation glued on. The structured data gives you the declaration, the kind of page, the platforms and the real title.

The numbers show it:

- Titles that still end in the web page's suffix fell from 129,074 to 2,854.
- Pages with a declaration went from 293,877 to 374,369.



A filter can only use what a page carries, so a page with a known kind and platforms is a page a filter can use.

The documentation describes a foldable iPhone

Search the new database for the word "hinge" and Apple's own pages answer with **iPhone Duo**.

The overview, "Preparing your app for iPhone Duo", describes it as a foldable phone with a compact outer display and a large inner display, each with a front camera. The inner camera is hidden when the phone is closed. As it opens, closes, folds partway and rotates, content moves between the displays, and when it is partly folded, views adapt to the folding region. In some poses the navigation bar, toolbar and tab bar sit together, vertically, on the side of the display. The page also says an app built with Xcode 26 or earlier does not extend under the status bar and camera.

All of these pages are in the "new" list of the diff, so none was in the June database. In SwiftUI, UIKit and XCUIAutomation together there are about 180 new pages whose address contains `arrangement`, `hinge` or `reservedregion`. I counted by address, so treat that number as a guide. Of the 206 new SwiftUI pages, 174 start at iOS 27 or later, and of the 266 new UIKit pages, 249 do.

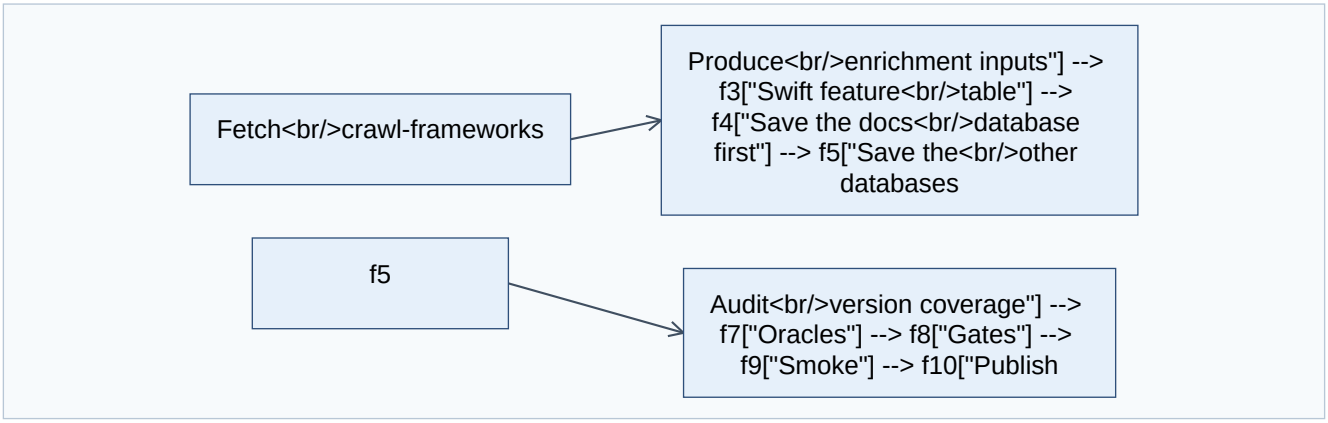
```
Unsupported Mermaid diagram: line 2: only rectangle labels like `A[Label]` are
supported
flowchart LR
  subgraph L["Layout"]
    direction TB
    r1["ReservedRegion<br/>SwiftUI and UIKit"] -->
a1["ArrangementView<br/>SwiftUI"]
    r1 --> a2["UIArrangementViewController<br/>UIKit"]
    a1 --> s1["OverlayArrangementViewStyle<br/>SplitArrangementViewStyle"]
  end
  subgraph H["The hinge"]
    direction TB
    h1["DeviceHinge<br/>DeviceHingeContext"] -->|"tested with"|
h2["XCUIHinge<br/>in your UI tests"]
  end
  subgraph C["The cameras"]
    direction TB
    c1["Registering a camera capture<br/>accessory on iPhone Duo"]
    c2["Choosing a camera by the<br/>direction it faces"]
  end
```

Apple describes `ArrangementView` as a view that lays out a primary and a secondary view by size class, orientation and reserved regions. The overlay style layers the primary view over the secondary one, or places them side by side on a folded display. `XCUIHinge` is "a reference to the hinge on iPhone Duo". I have not seen the device. I read the documentation, which is what Cupertino is for.

What else changed

- **WWDC session transcripts** are now a ninth database: 1,915 entries across 197 frameworks, with each session's frameworks and Swift version.
- **The App Review Guidelines** are pages, one for each numbered section.
- **Versions on every result.** Search shows the Swift version of each result and the OS versions it needs. Version filters now work on packages and samples, and a filtered search says how many pages it left out for want of a version.
- **Checks.** There are oracles that re-derive the stored versions from the raw files and compare them row by row, and the release gate runs the search oracle for every source.
- **The release tool** now cuts a release against Codeberg. The old GitHub workflow is gone.

The whole corpus refresh is one command, `cupertino-rel refresh`. It ends with audits, oracles, gates and a smoke test, and only then is anything published.



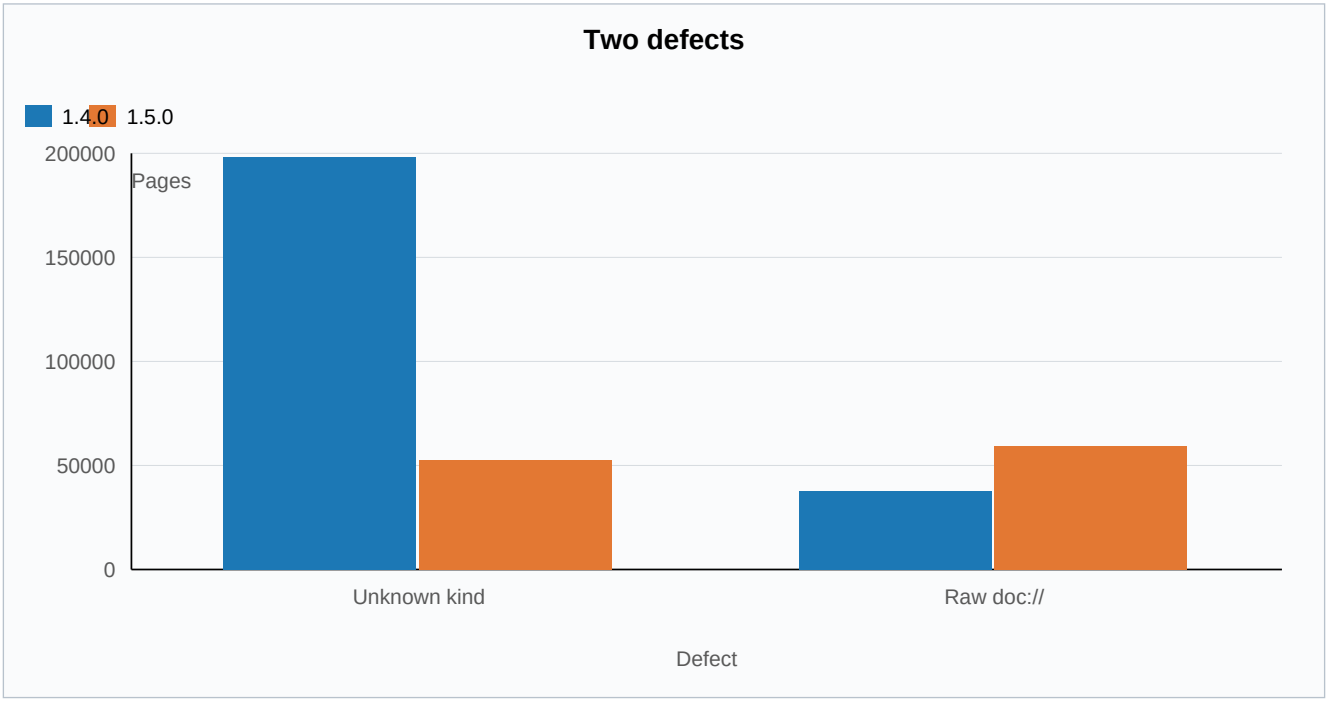
What the diff does not prove

Here is the part I would rather you hear from me.

It proves that no page was lost. It does not prove that no content was lost. 228,406 pages changed. In 75,445 of them, less than half of the old body lines survive exactly. I read seven of those pages. They showed reformatting: links became bold list items, and the availability text that used to be glued to the abstract moved into its own field. They also showed one real defect. I did not read all 75,445, so "reformatted" is an impression, not a measurement.

The comparison found two defects:

- **Raw references in the text.** Pages whose text contains an inline reference printed as `doc://com.apple.appkit/...` where the symbol's name belongs grew from 37,636 to 59,360 pages. That is a regression in the conversion from the structured data. The largest are `appkit` (5,478 pages), `uikit` (4,757) and `foundation` (4,307).
- **Pages without a kind.** 52,777 pages, 13.5 percent, still have no kind. That is down from 198,110, but a kind filter treats them as untyped.



Abstract coverage also fell, from 61.4 to 56.6 percent, although 27,494 pages were added. In six samples I read, the old abstract was the page header and navigation run together, so those were

not real abstracts, but I did not check that for every page. `matter` and `kernel` were almost entirely rewritten, 24,224 of 24,390 and 23,182 of 39,651 pages, so they are the first place to look by hand.

Both have fixes on the development branch, and neither is released yet. The second fix gives a kind to about 33,000 of those 52,777 pages. The diff itself was a one-off script outside the repository, so the numbers above cannot yet be reproduced with a repository tool. If the diff is to be repeated, it should become a `cupertino-rel` command.

Update

Homebrew has 1.5.0:

```
brew update && brew upgrade cupertinohq/tap/cupertino
cupertino setup
```

`cupertino setup` downloads the matching database bundle, 714 MB. Then ask your agent something that was not in June's database, for example how an app adapts to a hinge, and see what it finds.

The full diff, framework by framework, with the analysis, is in the repository under `docs/diff-2026-06-20-to-2026-11-09`.