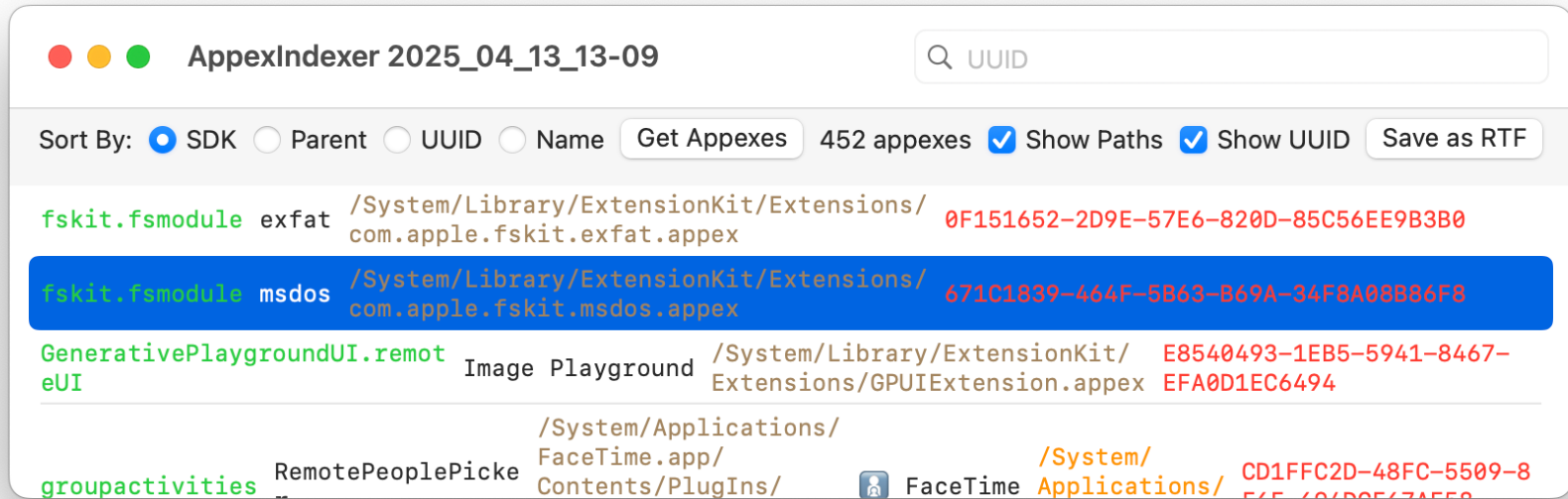


Start

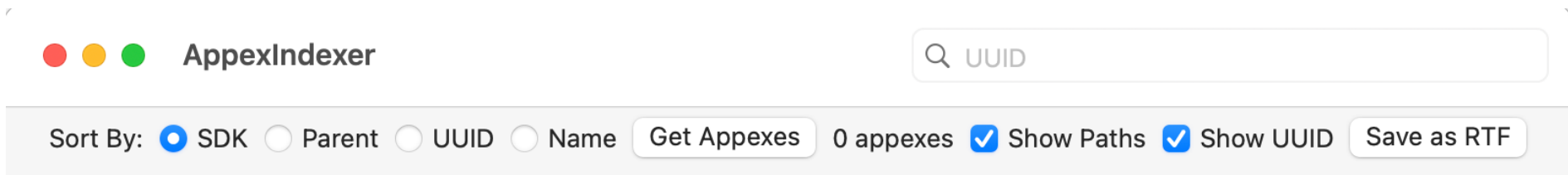


In System Settings, select **General**, then **Login Items & Extensions**, and scroll down to the end. You'll see a section headed **Extensions**. Those are app extensions, or *appexes*, and are featured in the list shown in AppexIndexer.

They cover a wide range, from code that modifies Dock tiles to Safari extensions and modules used to generate Quick Look thumbnails and previews for custom file types. AppexIndexer lists all those registered by macOS 14.6 Sonoma and later on your Mac, ordering them in functional groups.

[→ Toolbar](#)[→ Appex display](#)[→ Search](#)[→ Technical Information](#)

Toolbar



To fetch the current list of appexes, click the **Get Appexes** button.

When they are displayed in AppexIndexer's window, their total number is shown in the middle of the toolbar, and the window name changed to include the date and time they were obtained.

Sort by gives you a choice of order in which they're sorted alphabetically:

- **SDK** is the type or `NSExtensionPointIdentifier` of the appex, omitting the first two parts, `com.apple`.
- **Parent** is the name of the parent app containing that appex.
- **UUID** is the UUID identifier, widely used to identify the appex in the log and elsewhere.
- **Name** is the appex's display name.

In addition to basic info about each appex, you can opt to:

- **Show Paths**, which shows the path to the appex and that to its parent.
- **Show UUID**, which gives the appex's UUID in full.

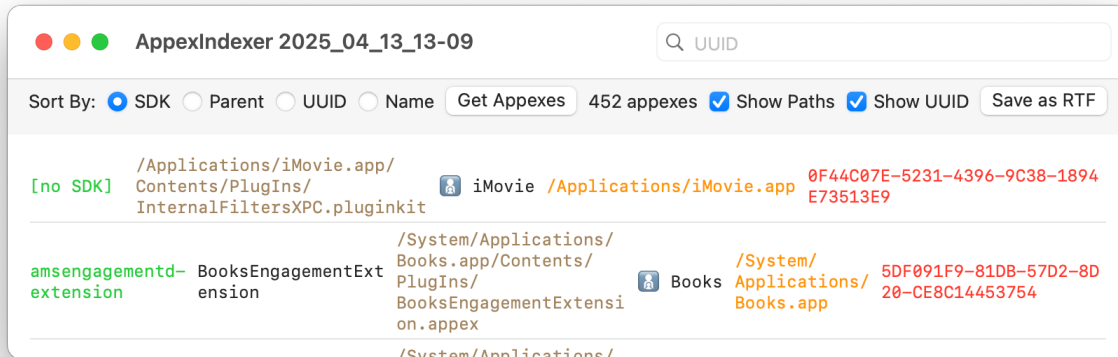
The **Save as RTF** button saves the current list of appexes as a Rich Text Format file.

→ [Appex display](#)

→ [Search](#)

→ [Technical information](#)

Appex display



Fields shown for appexes are:

SDK/NSExtensionPointIdentifier, **appex display name**, **appex bundle path**,  **parent name**, **parent bundle path**, **UUID**.

Regardless of sort order selected, those are always shown in that order. Not all fields are given for every appex, and many lack a parent. Rows in the appex list can be selected continuously with the **Shift** key held, and discontinuously with the **Command** key. Selected rows can then be copied as text using the **Copy** command, or **Command-C**.

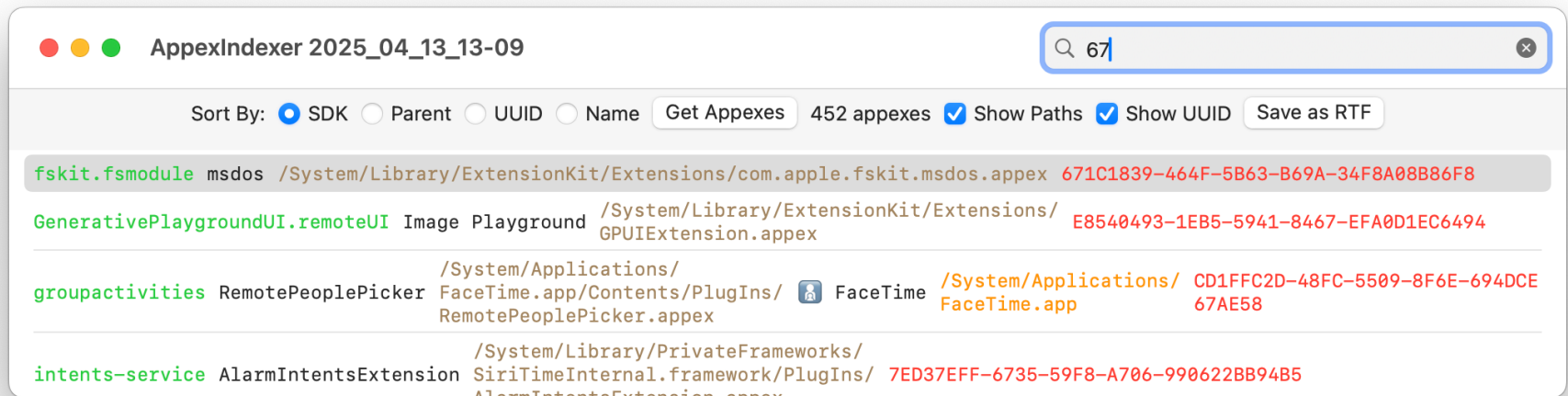
Saving the current list in Rich Text uses similar colours.

[→ Toolbar](#)

[→ Search](#)

[→ Technical information](#)

Search



Locating any appex on the basis of its name, function or parent is best performed using sort order options. To locate one by its UUID, type any sequence of characters from the UUID into the  Search box at the top right of the window. As you type more characters, the number of entries will reduce until you see just the one. The characters can come from any part of the UUID, not just its start.

If **Show UUID** isn't ticked, the search results will still display UUIDs to assist. If AppexIndexer can't find an appex UUID containing the characters in the Search box, you will be informed in the window. To return to the full display of appexes, click on the  at the right end of the search box. Although you can select and copy appex details from the Search list, the **Save as RTF** button still saves them all.

→ [Toolbar](#)

→ [Appex display](#)

→ [Technical Information](#)

Technical Information

The appex list is obtained using the command `pluginkit -m -vv`

The array of parsed entries is then displayed in a SwiftUI List of coloured Text fields. The requirement for Sonoma or later is to support the use of the parsing engine to be `Observable`; older equivalents are too clumsy and pose memory management issues, particularly with long lists.

Apple's official list of `NSExtensionPointIdentifiers` [is here](#), and much shorter than those you'll encounter in `AppexIndexer`. Apple's front page for appexes [is here](#).

`AppexIndexer`'s search feature was inspired and aided by Tiago Gomes Pereira at [Create with Swift](#).

Change list

1.0 (8):

- added sort options
- added UUID support & display options
- added search support.

1.0 (5):

- first release.

13 April 2025.