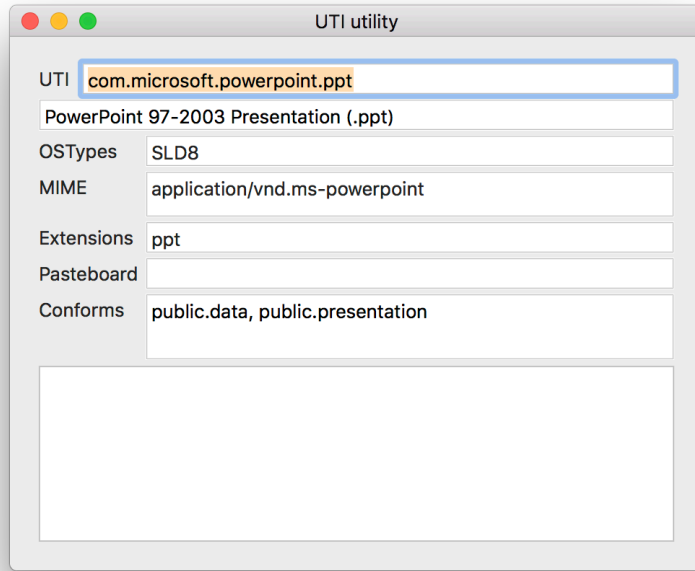


Start

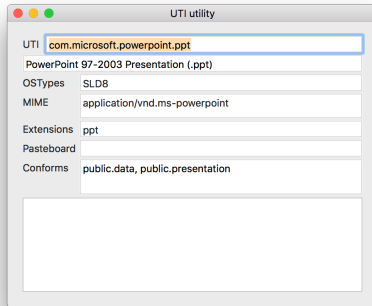


UTlutility looks up different type specifiers for files, folders, and other items, based on Apple's **Uniform Type Identifiers** (UTI). It can also discover UTIs which are not documented elsewhere.

Apple's documentation of UTIs hasn't been updated for over nine years, and omits many UTIs which are now in common use, both public UTIs with names starting in `public.` and many private ones. Converting between UTIs and other type specifiers such as filename extensions is also not easy. UTlutility opens up access to all of these without having to use Terminal.

→ [Lookup](#) → [Discovering UTIs](#) → [UTI structure](#) → [Major UTIs](#) → [Change list](#)

Lookup



UTlutility's main window contains seven text boxes and a large text scroller at the bottom. These show from the top:

- **UTI**: type or paste a UTI in here, press return, and if it is valid and recognised by your Mac, other details will be displayed.
- below the UTI box is the **descriptor** for that UTI, which explains that type.
- **OSTypes** gives the macOS OSType for the supplied UTI. You can also type or paste an OSType into this box, press return, and details of that OSType will be used to find its UTI.
- **MIME** gives the MIME type for the supplied UTI. You can also type or paste a MIME type here, press return, and see its UTI and other details.
- **Extensions** gives the filename extension for the supplied UTI. You can also type or paste an extension (without the period), press return, and see its UTI and other details.
- **Pasteboard** gives the macOS Pasteboard type for the supplied UTI. You can also type or paste a Pasteboard type, press return, and see its UTI and other details.
- **Conforms** lists those UTIs to which the supplied UTI conforms. In other words, these are more general types which are higher up in the tree of UTIs.
- The scrolling text box at the bottom shows **additional information** where it is available. This includes the icon file, reference URL, version, and any exported and imported type declarations. For the great majority of UTIs, these are not given.

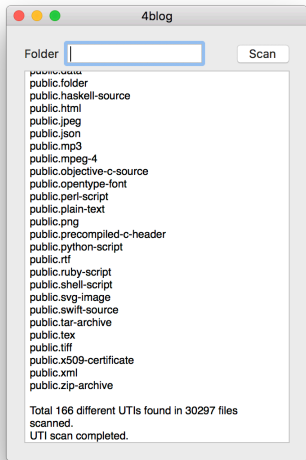
To look up any of these, type any of the following into their respective box: UTI, OSType, MIME type, filename extension (without the period or stop), or Pasteboard type. Press Return and the app shows all known identifiers and information for the one which you supplied.

→ [Discovering UTIs](#)

→ [UTI structure](#)

→ [Major UTIs](#)

Discovering UTIs



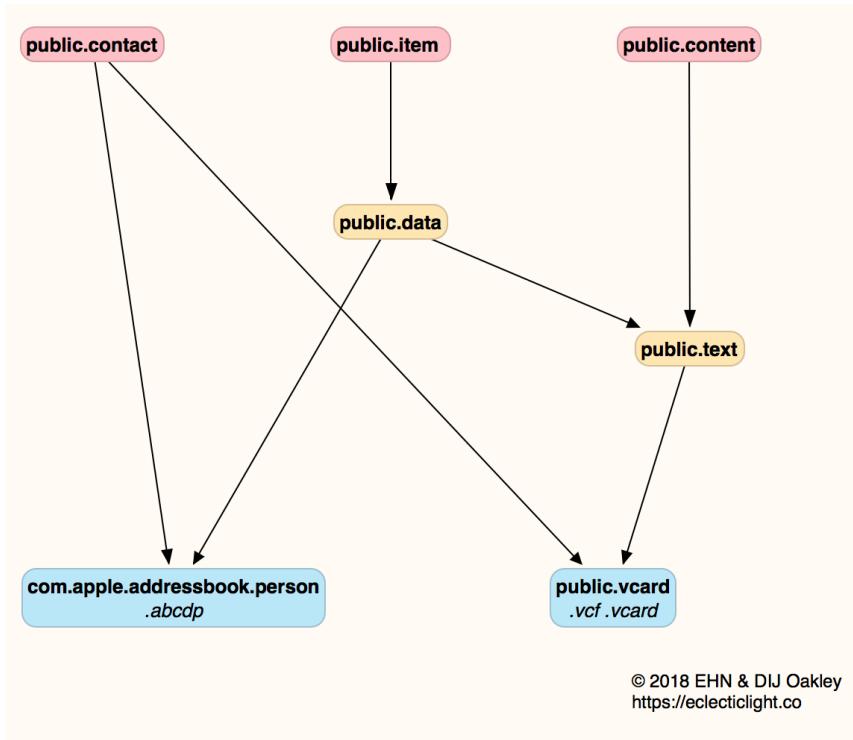
Discovering which UTIs are in use is normally difficult. Although Precize now shows the UTI for individual files, the best way to discover which are in current use is to open UTILITY's crawler view, using the **Open Crawler** command in its **Window** menu. Either click on the **Scan** button to select a folder to examine, or type a path (~ is allowed as shorthand for your Home folder) into the **Folder** textbox, then press the **Scan** button.

UTILITY then checks the UTI of every file and folder within the selected folder, and supplies a list of those found in the scrolling box below, which you can copy and paste it into another document, or copy and paste from into the main window to look up a UTI there.

⚠ Privacy protection in Mojave prevents UTILITY from looking inside folders containing protected private data, unless you add the app to your **Full Disk Access** list, in the **Privacy** tab of the **Security & Privacy** pane.

→ [UTI structure](#) → [Major UTIs](#)

UTI structure



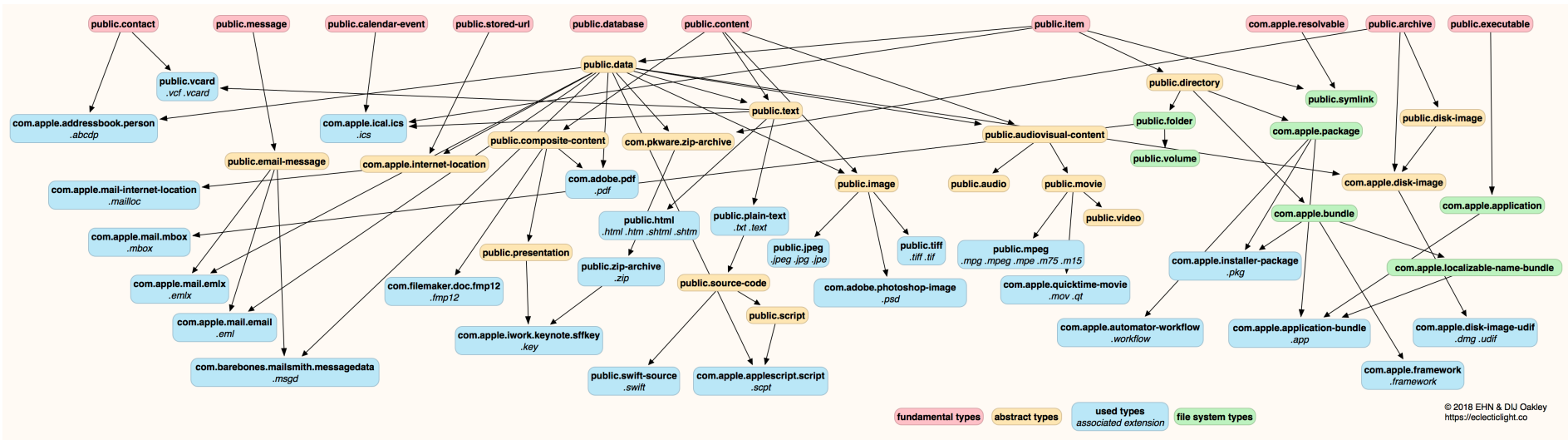
UTIs are a hierarchical taxonomic system. Apart from the top-level UTIs, every UTI has at least one parent, and may have one or more children.

For example, an exchangeable address book entry which might ordinarily have the filename extension `.vcf` or `.vcard` has the UTI `public.vcard`, which is a child of the fundamental UTI `public.contact`, and the derived UTI `public.text`, which in turn is a child of `public.content` and `public.data`, a child of `public.item`. In Apple's terminology, `public.vcard` **conforms to** `public.contact` and `public.text`.

→ [Major UTIs](#)

Major UTIs

Regular UTIs, both public and private, are extremely widely used. You don't have to look hard on a Mac now to come across 600 or more different types, and visualising even the most important is very difficult. The map below shows little more than fifty of the most common and important UTIs used in macOS. This is made the more difficult because of the shallowness of their conformity tree: a large number of individual image types are direct children of `public.image`, for example.



→ Major UTIs (concluded).

Major UTIs (concluded)

At the top of the UTI hierarchy are at least ten fundamental types:

- `public.item` – the base type for items of any kind
- `public.content` – the base type for document content
- `public.archive` – the base type for archives
- `public.executable` – the base type for executable data (code)
- `public.contact` – the base type for contact information
- `public.message` – the base type for all forms of message
- `public.calendar-event` – the base type for all scheduled events
- `public.stored-url` – the base type for URLs
- `public.database` – the base type for databases
- `com.apple.resolvable` – the base type for items which can be resolved by the Alias Manager.

Dynamic UTIs are created on the fly when needed, and have become so common as to be a problem. When surveying UTIs associated with the contents of most folders, meaningless dynamic UTIs such as `dyn.ah62d4rv4ge8085pt` generally outnumber regular UTIs, and most of them map to unknown filename extensions or pasteboard types.

Change list

1.0:

- ported to Swift 5 and Xcode 10.2.1
- added code integrity check when opening app
- added support link to Help menu
- added new Help book.

1.0b3:

- ported to Swift 4.2.1 and Xcode 10.1
- notarized for Mojave.

1.0b2:

- added output of iconFile, referenceURL, version, exported and imported type declarations, to scrolling text box
- tidied conforms listing
- added full support for Dark Mode
- added custom app icon
- ported to Swift 4.2
- built using Xcode 10B3.

1.0b1:

- initial release.

28 May 2019.

→ [Acknowledgement](#)

Acknowledgement

UTlutility uses Hiroki Kato's UTI extensions to Swift, which are supplied under the MIT License:

```
// The MIT License (MIT)
//
// Copyright (c) 2015 Hiroki Kato
//
// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to deal
// in the Software without restriction, including without limitation the rights
// to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
// copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
//
// The above copyright notice and this permission notice shall be included in all
// copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
// OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
// SOFTWARE.
```

I am very grateful to Hiroki for making my task so straightforward.