FileWave Apple Profile Editor Explained

Description

Configuration of Apple devices largely relies upon Profiles. Profiles contain Payloads: settings defined to control aspects of the Operating System and user experience.

FileWave has a built-in editor allowing the building of Profiles. This editor is designed to ensure Profiles only contain Payload keys that meets Apple's definitions, which in turn provides peace of mind that Profiles will instal on devices when associated.

Apple Definitions

Apple's developer pages list definitions of all payloads and as such, which keys are available within each payload; including, amongst other things, whether keys are required or optional.

Below is the link to Apple's definition for the Lock Screen Message Payload and example content:

▼ Example: Lock Screen Message

https://developer.apple.com/documentation/devicemanagement/lockscreenmessage

```
Dict {
    PayloadDisplayName = Lock Screen Message
    PayloadScope = System
    PayloadType = Configuration
    PayloadRemovalDisallowed = false
    PayloadContent = Array {
        Dict {
            PayloadVersion = 1
            PayloadDisplayName = Lock Screen Message
            PayloadType = com.apple.shareddeviceconfiguration
            IfLostReturnToMessage = FileWave IT
            LockScreenFootnote = %custom_field.asset_tag%
            AssetTagInformation = %custom_field.asset_tag%
            PayloadEnabled = true
            PayloadIdentifier = ML1063.local.93367c30-cfe5-4c58-a2a0-
83190666231b.com.apple.shareddeviceconfiguration.1b7de9ad-fc3d-4f97-9338-a26d7811f974
            PayloadUUID = 1b7de9ad-fc3d-4f97-9338-a26d7811f974
        }
    }
    ConsentText = Dict {
        default =
    }
    PayloadIdentifier = ML1063.local.93367c30-cfe5-4c58-a2a0-83190666231b.Configuration.93367c30-cfe5-4c58-
a2a0-83190666231b
    PayloadVersion = 1
    PayloadUUID = 93367c30-cfe5-4c58-a2a0-83190666231b
}
```

> ℹ️ For the keen eyed, the values provided for two of the possible keys contain parameters from inventory. This allows far more flexibility than fixed details.

Apple's guide shows there are 3 possible keys and their value types:

- AssetTagInformation -- String
- IfLostReturnToMessage -- String
- LockScreenFootnote -- String

FileWave Mechanics

When creating a Profile in FileWave, any Payload included will automatically have ALL default values set within the Payload. The editor is then used to customise chosen values.

> ℹ️ macOS will react to all missing Payload Keys by applying default values

Below is an example for managing Finder:

▼ Finder Payload for Desktop

Looking at the FileWave Profile Editor > Finder > Preferences Payload details, for Desktop, FileWave lists 3 items enabled by default:
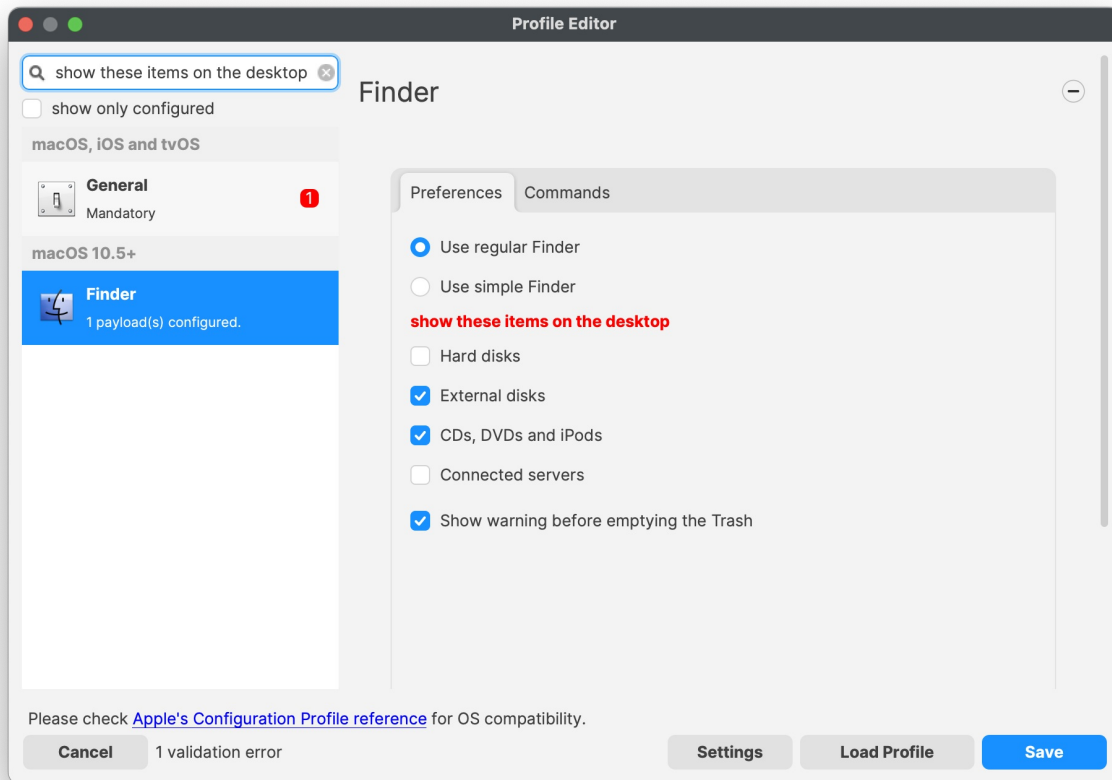
It can be seen that this a direct representation of the default values defined in Apple's documentation:

https://developer.apple.com/documentation/devicemanagement/finder

ShowExternalHardDrivesOnDesktop -- boolean
If false, the system doesn't show external hard drives on the Desktop.
Default: true

ShowRemovableMediaOnDesktop -- boolean
If false, the system doesn't show removable media items on the Desktop.
Default: true

WarnOnEmptyTrash -- boolean
If false, the system doesn't warn the user before emptying the trash.
Default: true

All other desktop values have a default of 'false'

> ✅ Building Profiles in FileWave should ensure you always have correctly configured Payloads.
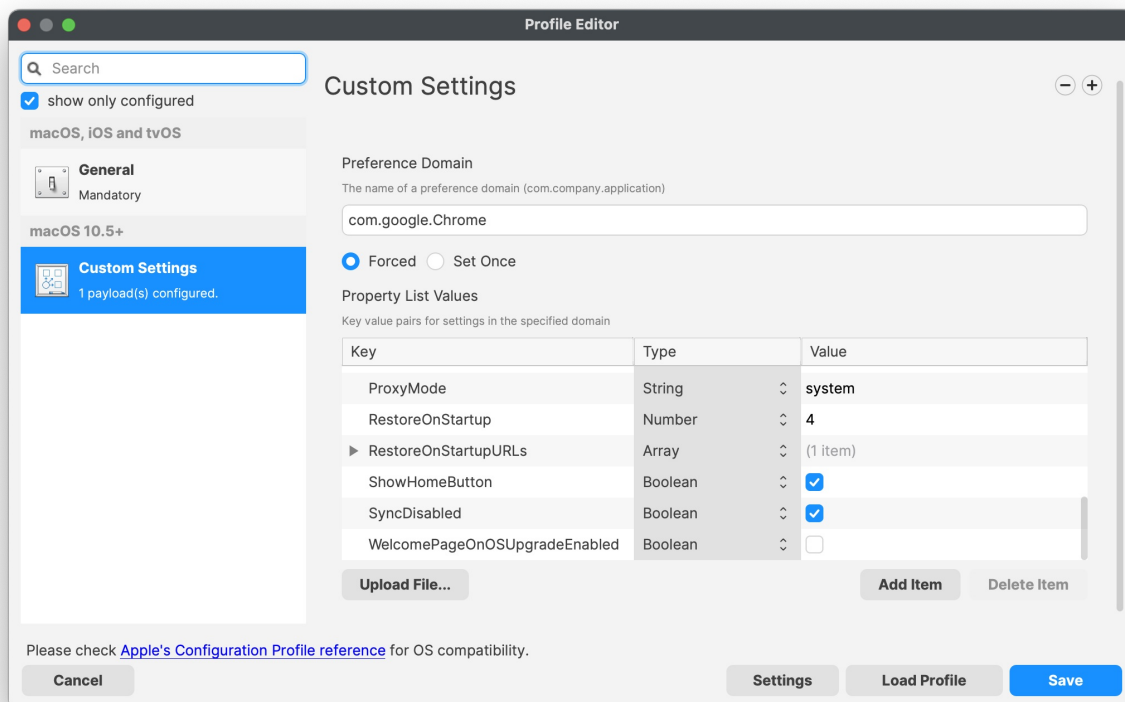
Custom Settings

Prior to Profile Payloads, Apple had a mechanism known as Managed Preferences (MCX). These were either controlled using Apple Server or as with Profiles, could be installed locally on devices.

In many ways, they are essentially the same thing; an xml structured file containing configuration. Despite Profiles taking over from MCX many years ago, MCX management still exists today in the latest macOS. The FileWave Profile Editor is able to leverage these and they are managed with the Custom Settings Payload.

This means that not only can the OS be managed, but also provides a mechanism for controlling 3rd party Applications whose configuration is defined by this same implementation.

▼ Example Google Chrome Custom Settings

Custom Settings define the Preference Domain (this is the name of the plist file which controls those settings) and then the Property List Values to be controlled:

⚠ Custom Settings have a slightly different format than Profiles. It is not possibly to directly import a Profile into a Custom Settings Payload

Importing Profiles

It is possible to find keys that are not defined by Apple, yet may still seem to function. As such, it would seem on face value, reasonable to add such keys to the Payload. Yet, FileWave only includes those that are defined by Apple, so how could this be possible?

One method would be the use of Custom Settings, however it is possible to use FileWave to Import any Profile, either from another FileWave Server, built using an Editor or from another tool. There are though some important considerations when importing from other tools.

FileWave Definitions

As noted above, FileWave definitions are built from Apple's definitions. Additionally, FileWave will always configure ALL key/value pairs to default if not already defined. However, the flip side of this is that FileWave will also REMOVE any keys that are not part of the definition.
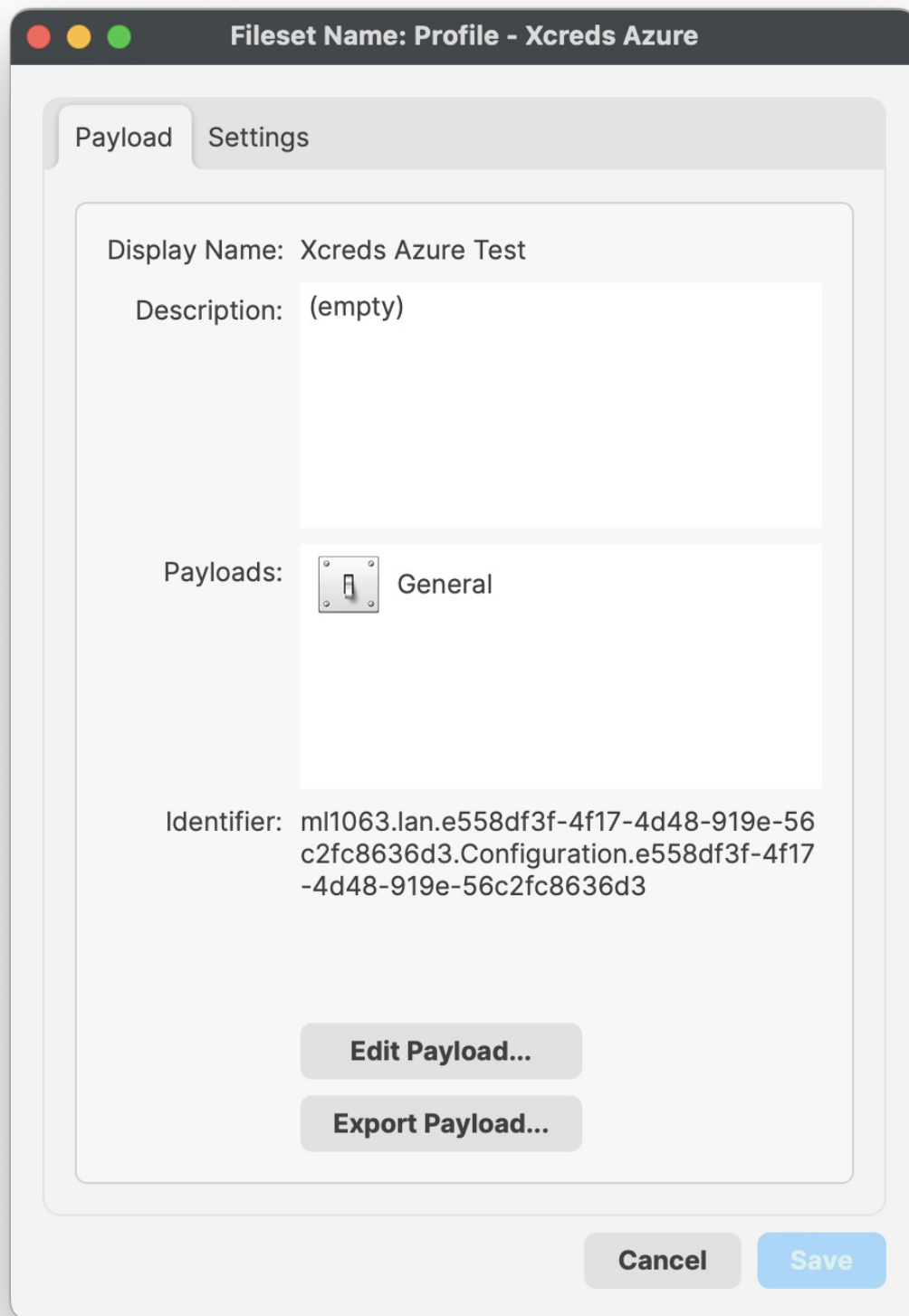
Importing Undefined Keys

If the Profile being imported contains keys that are not part of the definition, the Profile should never be altered or saved within FileWave.

✓ To avoid the need to save the Profile to be imported, drag and drop the Profile into FileWave Central Filesets view or use FileWave Anywhere to upload the Profile. Avoid duplication, as noted below.

The Profile, if opened, will not display unknown keys and if the payload only contains unknown definitions, it will appear as if it doesn't even exist within the Profile.

The only item that appears to exist in the Payload is General:

Fileset Name: Profile - Xcreds Azure

Payload | Settings

Display Name: Xcreds Azure Test

Description: (empty)

Payloads: 🔲 General

Identifier: ml1063.lan.e558df3f-4f17-4d48-919e-56
c2fc8636d3.Configuration.e558df3f-4f17
-4d48-919e-56c2fc8636d3

Edit Payload...

Export Payload...

Cancel | Save

Exporting the Profile though and looking at its contents, it can be seen that there is more to this Profile than can be shown using the FileWave Editor:

```
Dict {
    PayloadVersion = 1
    PayloadDisplayName = Xcreds Azure
    PayloadScope = System
```

```
            PayloadType = Configuration
            PayloadRemovalDisallowed = false
            PayloadContent = Array {
                Dict {
                    PayloadVersion = 1
                    PayloadDisplayName = XCreds
                    PayloadUUID = 216961FC-A8FE-4E1B-8253-747D3A4A184B
                    PayloadType = com.twocanoes.xcreds
                    discoveryURL = https://login.microsoftonline.com/xxx/.well-known/openid-configuration
                    loginWindowBackgroundImageURL = file:///Users/Shared/random.heic
                    scopes = profile openid offline_access
                    PayloadIdentifier = ml1063.lan.4301329C-0440-4BB7-B8E8-B498DDE2448C.com.twocanoes.xcreds.216961FC-
        A8FE-4E1B-8253-747D3A4A184B
                    clientID = xxx
                    PayloadOrganization =
                }
            }
            ConsentText = Dict {
                default =
            }
            PayloadIdentifier = ml1063.lan.e558df3f-4f17-4d48-919e-56c2fc8636d3.Configuration.e558df3f-4f17-4d48-919e-
        56c2fc8636d3
            PayloadOrganization = FileWave
            PayloadUUID = e558df3f-4f17-4d48-919e-56c2fc8636d3
        }
```
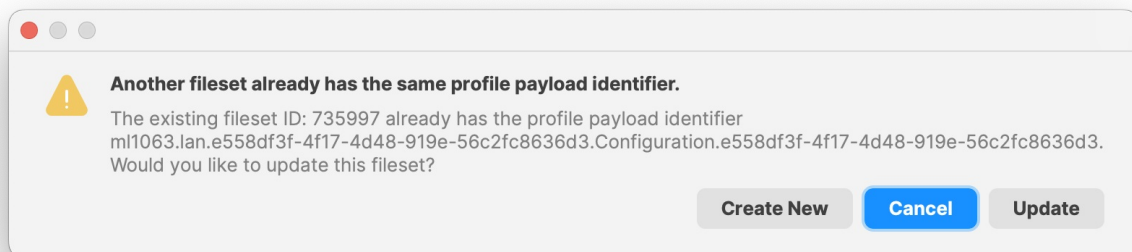
Importing Existing Profiles

Each Profile has a Unique Identifier. If at the time of importing, a Profile with the same Identifier already exists within FileWave, a prompt will be shown, asking if this should be newly created or if it should overwrite the current Profile.

> ⚠️ This action will, as with all save actions with Profiles, causes all missing, but defined keys, to be added with default values, whilst all undefined keys will be removed.

macOS GUI

The macOS GUI does not always necessarily display the setting that is defined. In some instances, it may even look like it allows the user to alter the value, however, to the user it will appear not to work, since in reality it is managed.

Undefined Keys

Looking into plist files or binaries of applications, additional, undefined keys can sometimes be found. What's more, it may be possible to use these keys, either within an imported Payload or Custom Settings, with the desired effect. However, since they are undefined, it cannot be guaranteed that these settings will work on all versions of macOS.

▼ Example: Apple Undefined Key

Looking at the macOS Restrictions Payload, one of the services available for control is AirDrop:
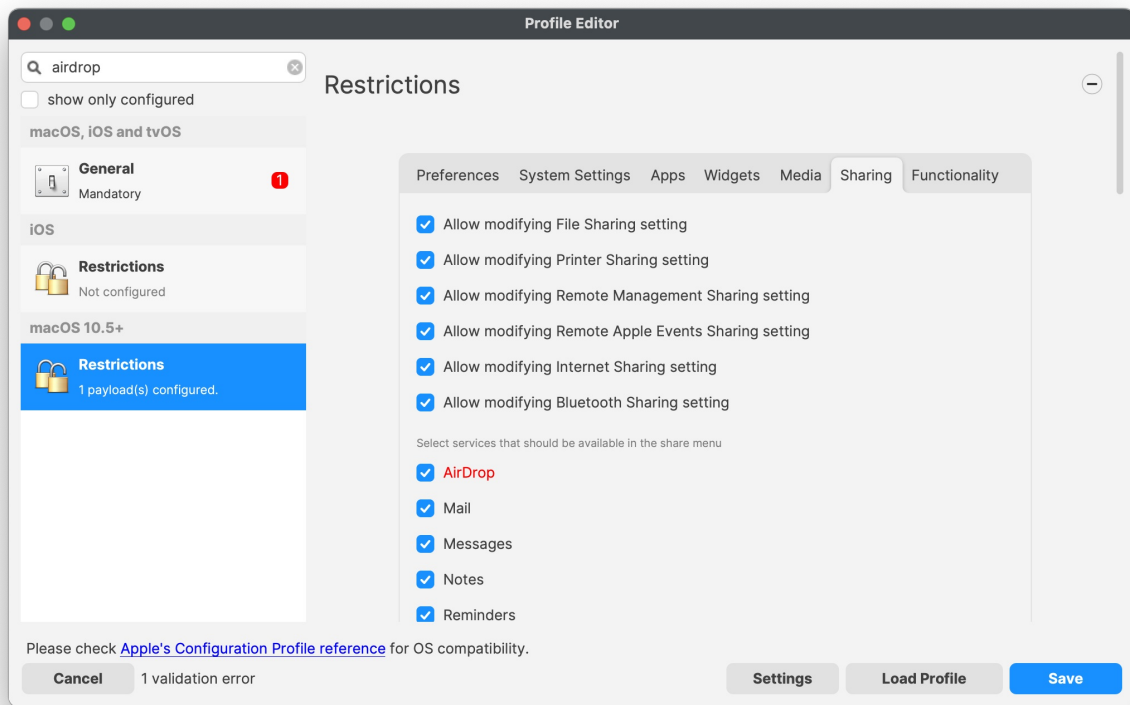
```
allowAirDrop -- boolean
If false, the system disables AirDrop. Requires a supervised device. Available in iOS 7 and later, and macOS 10.13 and later.
Default: true
```
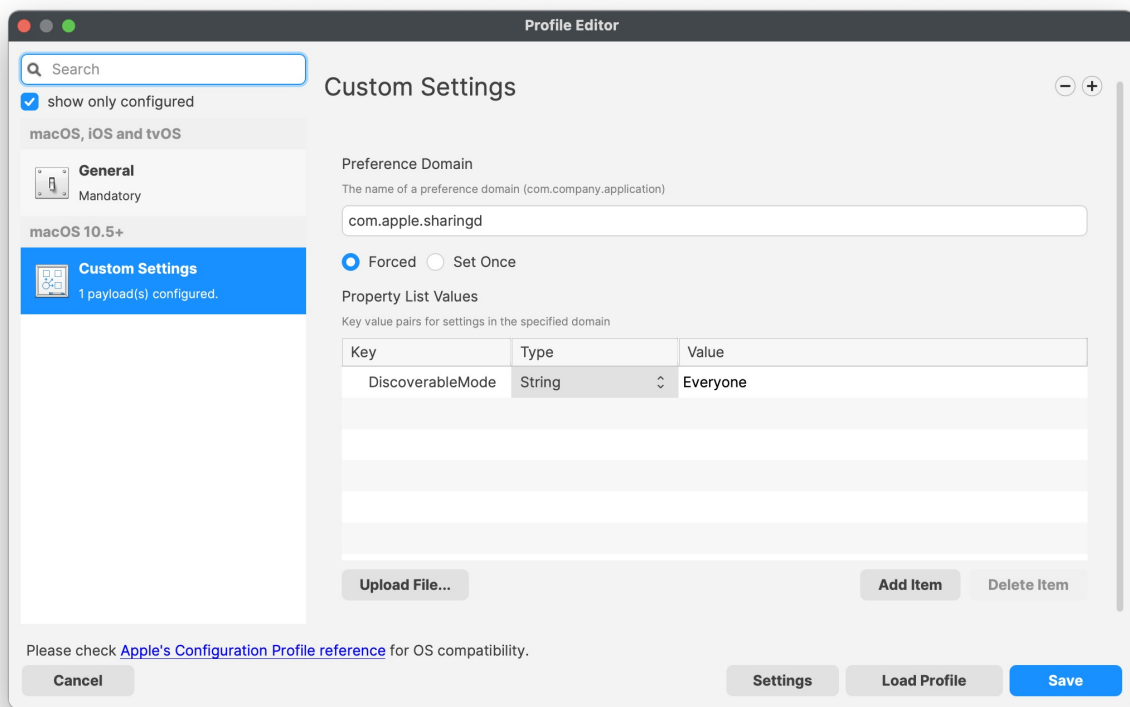
When using AirDrop there are options for discovery:

- No One
- Contacts Only (Requires iCloud login)
- Everyone

Looking at a user's plist file, the key defining their setting may be seen:

```
% defaults read ~/Library/Preferences/com.apple.sharingd.plist DiscoverableMode
Off
```

Taking that info a Custom Settings Payload:

Applying to the same device whose AirDrop discovery is set as 'Off':

AirDrop lets you share instantly with people nearby.

Allow me to be discovered by: Everyone ∨
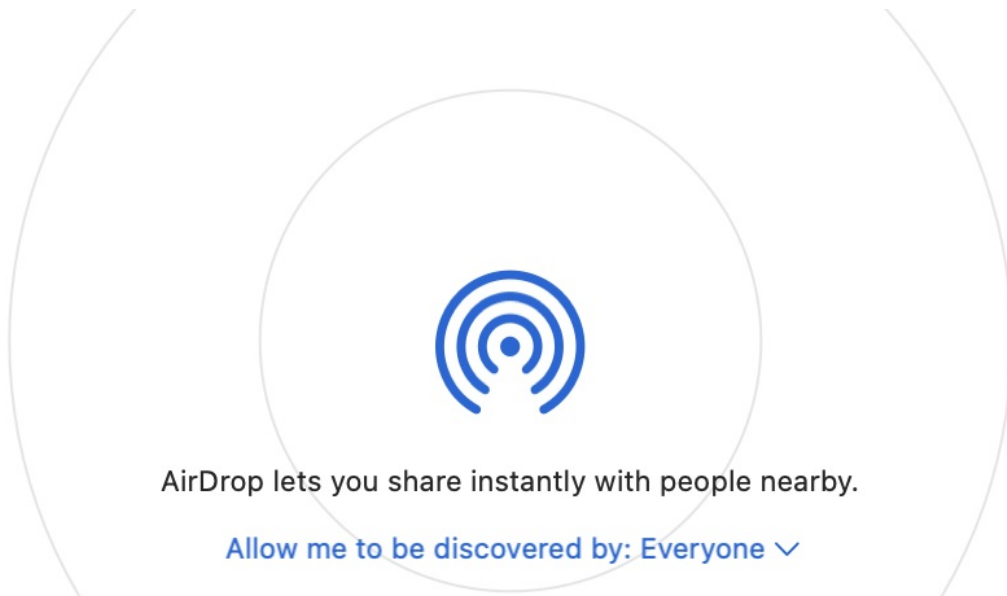
For the user, it is configured as 'Off', yet the Payload is defining this to be 'Everyone' and this is reflected in the user experience.

> ⚠ Remember, this is an undefined key and may not function as desired with all macOS versions.

Missing Keys

At times, Apple add/remove keys from Payload definitions. If a key is considered to be missing from FileWave, but is clearly defined in Apple's developer documentation, then consider creating a ticket through FileWave Support.

Conflicting Keys

It would be worth noting in this topic, conflicting Payloads. If two payloads were overlapping in content, but with different settings, what should be the outcome.

As per Apple's documentation: https://support.apple.com/en-gb/guide/deployment/dep9a318a393/web

- If you have multiple configuration profiles containing the same payloads with different settings, the resulting behaviour is undefined. This doesn't apply to the restrictions payload. In this case, the more restrictive value for the setting is applied.

Conclusion

FileWave Profile Editor protects against malformed Payloads, ensuring they meet Apple's requirements.

If a key is required that is not defined by Apple, consider using a Custom Settings Payload to deliver that key/value pair.

If importing an undefined key within a non-Custom Settings Payload, do not edit the Profile to ensure it remains untouched.