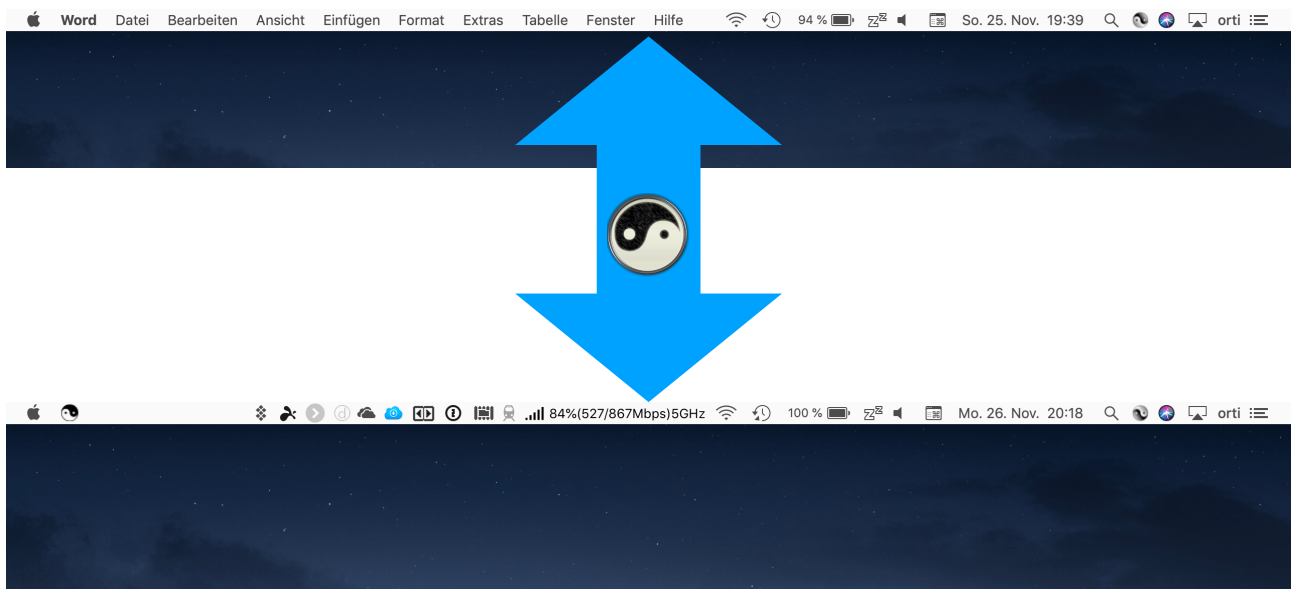




A_{ccessMenuBarApps}

tames your *Mac Menu Bar* on small screens

English	2-5
Deutsch	6-9



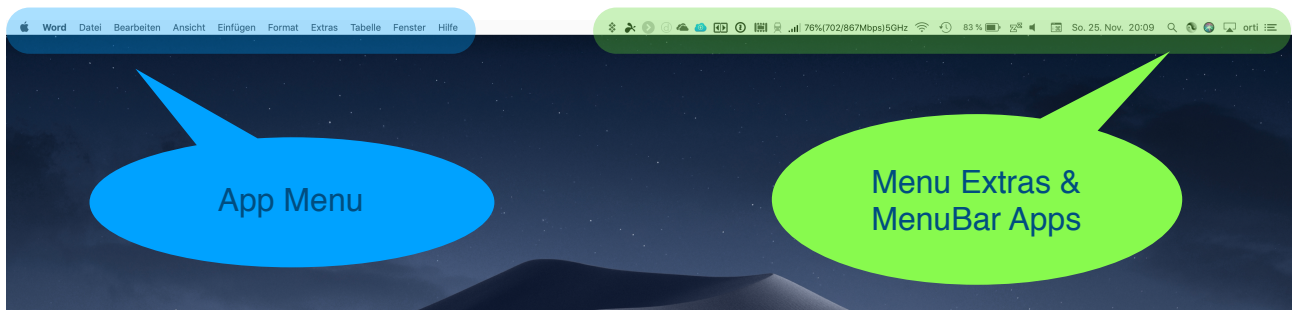


A_{ccessMenu}BarApps

AccessMenuBarApps does what the name suggests: It gives the user instant access to all menu bar apps located on the right side of the *Mac Menu Bar*.

What problem addresses AccessMenuBarApps?

On the right side of the menubar there are a lots of more or less useful apps, like Time machine, Airport, Bluetooth, ... (*Menu Extras* coming with macOS) and 3rd party *MenuBar Apps* like TripMode, DropBox or NoSleep. The available space for these *Menu Extras* and *MenuBar Apps* depends on the *App Menu* of the currently active application on the left side. The more menu items the application has, the less space is available for the *Menu Extras* or *MenuBar Apps*.



Depending on screen resolution the space between *App Menu* and *MenuBar Apps* is sufficient to see all *MenuBar Apps* or not. The *App Menu* has always priority. With a giant display of an 27" iMac or a large display of an 15" MacBookPro the menubar normally is no problem, because the screen is so large that there is always enough space between *App Menu* and *MenuBar Apps*. But if you have a 13" Macbook (Pro | Air) with a resolution of 1280x800, most time some of your *Menubar Apps* are hidden by the *App Menu*. And with Murphy's law: If you want to use a *Menubar App*, exactly this app is hidden by the *App Menu*. That sucks.



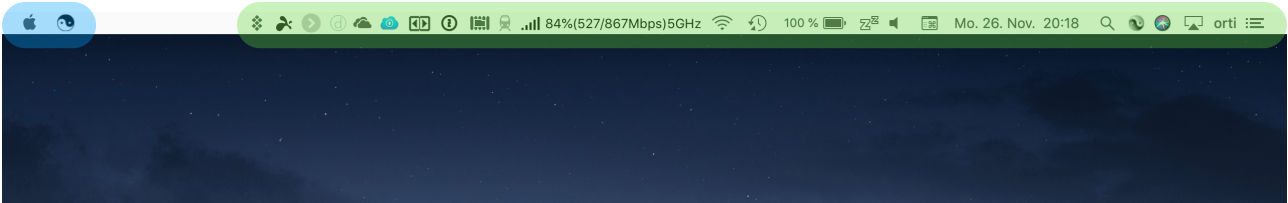
If you want to avoid this you have two options: to accept the inevitable or limit yourself to a very limited set of *MenuBar Apps* (if you are a friend of Occam's razor :-).

Or is there a 3rd option? *AccessMenuBarApps* tries to give you a 3rd option.

How does AccessMenuBarApps work?

The basic principle

The basic principle of *AccessMenuBarApps* is very simple: start an application that has a reduced *App Menu*, so that there is a maximum space between *App Menu* and *MenuBar Apps*. And therefore all *MenuBar Apps* are visible and accessible (unless you have definitely too many *MenuBar Apps*).



This principle is very simple. So there are a lot of apps out there which have a reduced *App Menu* (like toHide, Menuola, NoMenuBar, ...). But no of these apps take care about how to activate/deactivate the reduced *App Menu* efficiently.

AccessMenuBarApps addresses how to activate/deactivate the reduced *App Menu* efficiently. *AccessMenuBarApps* gives you the option to switch between full *App Menu* of the currently running application and the reduced *App Menu* (and vice versa) by keyboard shortcuts, mouse gestures or menubar icon.

AccessMenuBarApps is a background app that is always running in background and is waiting for keyboard and mouse events. You don't have to start and quit the app every time you need a reduced *App Menu*. The app is started once and is just waiting for being activated or deactivated, so the response time is very short.

Options for activating and deactivating

Using keyboard shortcuts

By using a keyboard shortcut (by default ⌘Space) you can switch at any time between full *App Menu* of the currently running application and the reduced *App Menu*.

You can configure the shortcut in the preferences window of *AccessMenubarApps*, which is available via *App Menu*. If you don't specify a shortcut in the preferences window the default ⌘Space will be used.

Using mouse gestures

By using a mouse gesture you can switch at any time between full *App Menu* of the currently running application and the reduced *App Menu*. The following mouse gestures are supported:

- **Single Knocking** on the top border of the screen
knocking one times will exchange the app menus
- **Double Knocking** on the top border of the screen
knocking two times will exchange the app menus
- **Pushing** (Squeezing) the top border of the screen
pushing/squeezing for a second (similar to Exposé's active corner; but here is it an active border) will exchange the app menus

- **Point and Click** at the top border of the screen
a simple mouse click will exchange the app menus

Mouse gestures will work when moving and/or dragging the mouse. So, if you have a menubar app, that is a target of drag and drop, but is currently not visible in the menubar, you can work like this: drag the file, text, ... as usually, than move the dragged mouse to the top border of the screen („push/„squeeze“ or „knock“); than the reduced menu will appear and the formerly hidden Menubar App is visible and you can use it as the target of your drag and drop action.

Support of mouse gestures can be enabled/disabled and configured in the preferences window. Here you can also specify that mouse gestures should only work for a subarea of the screen border to avoid conflicts with active corners of Exposé or conflicts with using the app menu.

Using menubar icon

If the menubar icon of *AccessMenuBarApps* is visible and accessible (this is the originally problem :-), you can click on that icon to switch between full *App Menu* of the currently running application and the reduced *App Menu*. If you don't want to use this option, you can hide the menubar icon in the preferences.

Using dock icon

You can click on the dock icon to switch between full app menu of the currently running application and the reduced *App Menu*. If you don't want to use this option, you can hide the dock icon in the preferences.

When hided dock icon is activated the dock icon is only shown while the reduced *App Menu* is shown. During normal operation the dock icon is hided.

Other Preferences

Sound effects

In the preferences window you can specify whether each switching, knocking and pushing/squeezing should be accompanied by sound effects. If sound effects are on, the exchange of the *App Menus* will last a little bit longer, because the switching is synchronized with the sound effects.

So, if you want to work efficiently you shouldn't use sound effects, although they are very nice in the beginning (but they are only a gimmick).

Visual effects

Switching and mouse gestures (knocking and pushing/squeezing) can be visualized on the screen. So you can see the sensitive area of the top border of the screen. In the preferences window you can switch the visual effects on or off. You can also choose the color used for visualization of the sensitive area.

Animated icon in the dock

If you want to use the dock icon for switching between full app menu and reduced app menu, you can choose to animate the dock icon either as an analog clock showing the current time or as some eyes tracking the mouse cursor.

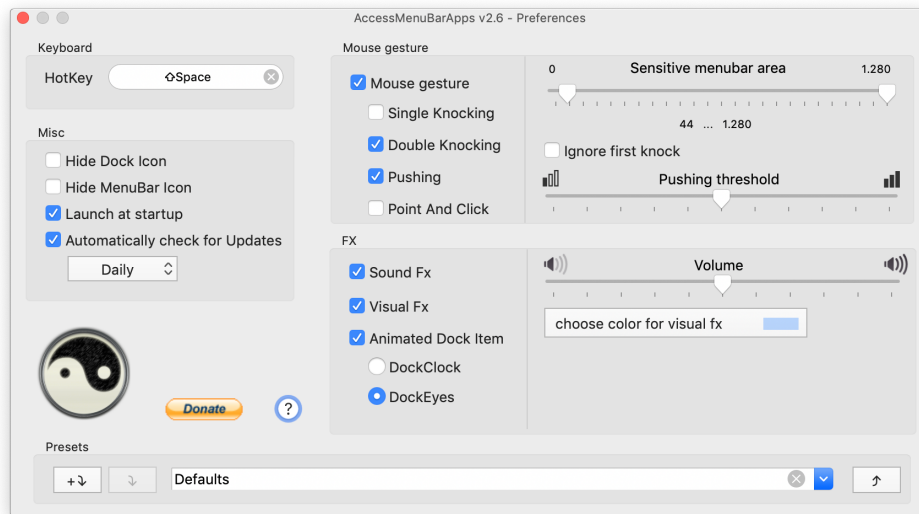
Launch at startup

If you activate the option „launch at startup“ in the preferences window, *AccessMenuBarApps* will be added to the front of the list of Login items, so that the *AccessMenuBarApps*

icon should be the rightmost 3rd party menubar app and therefore most time visible and accessible.

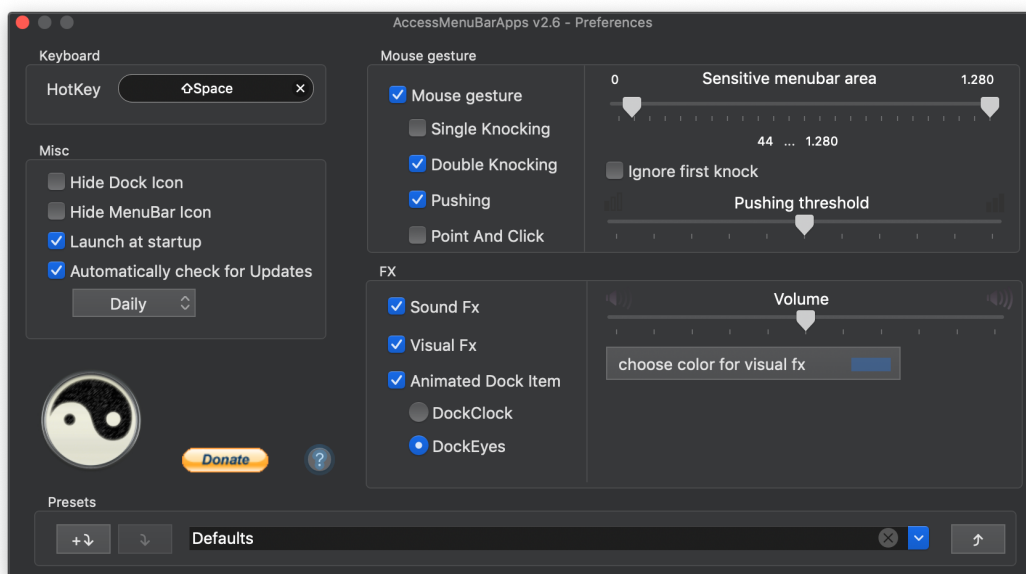
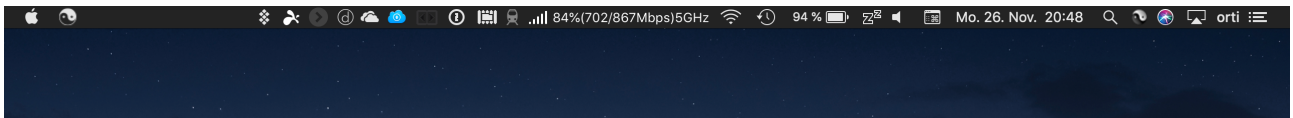
Save preferences

If you want to use different preferences for different contexts, you can easily save your preferences as presets and restore your preferences from stored presets.



Dark Mode

AccessMenuBarApps supports dark-mode of macOS Mojave.





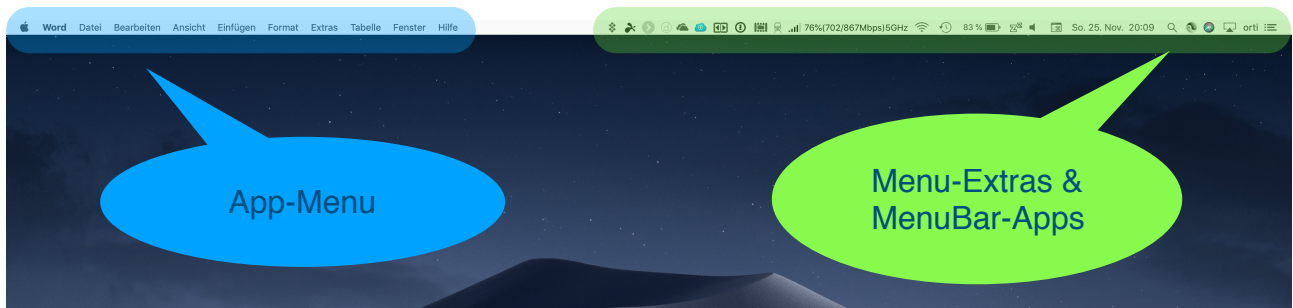
A_{ccessMenuBarApps}

Was macht AccessMenuBarApps?

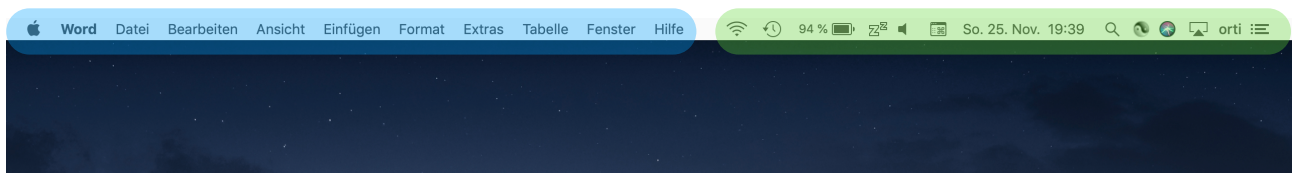
AccessMenuBarApps macht, was der Name sagt: Die App ermöglicht den schnellen, unmittelbaren und direkten Zugriff auf die *MenuBar-Apps*, die im rechten Teil der MenuBar residieren.

Welches Problem löst AccessMenuBarApps?

Im rechten Teil der MenuBar nisten sich üblicherweise eine Reihe von nützlichen und weniger nützlichen Apps ein. Der zur Verfügung stehende Platz richtet sich nach dem App-Menu der gerade aktiven Anwendung auf der linken Seite. Je mehr Menüpunkte die Anwendung hat, desto weniger Platz steht für die Menu-Extras oder MenuBar-Apps zur Verfügung.



Je nach Bildschirmgröße ist dies ein kleineres oder größeres Problem. Auf einem großen Bildschirm, wie bei einem 27" iMac oder einem 17" MacBookPro, ist dir zur Verfügung stehende Platz in der Regel groß genug, um alle sinnvollen MenuBar-Apps immer im Zugriff zu haben. Je kleiner der Bildschirm, desto größer ist allerdings die Wahrscheinlichkeit, dass gerade die Menubar-App, die gerade benötigt wird, vom App-Menu verdeckt wird.



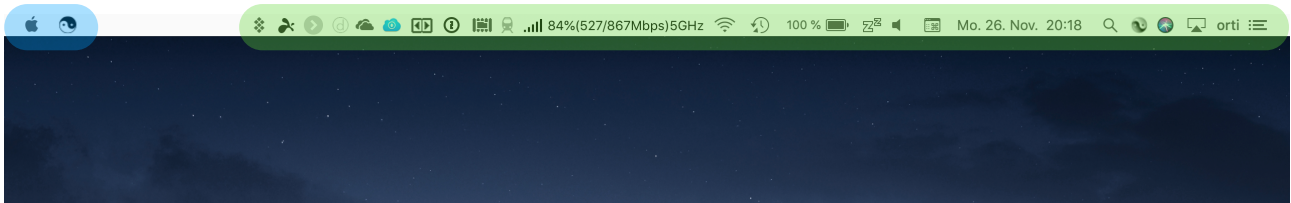
Besonders betroffen sind Besitzer eines MacBookAir oder eines 13" MacBook oder 13" MacBookPro. Betroffene haben bisher eigentlich nur zwei Möglichkeiten: entweder sie verzichten ganz auf Menubar-Apps oder aber sie beschränken sich auf die unbedingt notwendigen Menubar-Apps und akzeptieren, dass bei manchen Apps die Menubar-Apps nicht zugreifbar sind.

AccessMenuBarApps versucht, eine Lösung des Problems bereitzustellen.

Wie funktioniert AccessMenuBarApps?

Das Grundprinzip

Das Grundprinzip von AccessMenuBarApps ist ganz einfach: Starte eine Anwendung, die ein möglichst kleines App-Menu hat, so dass zwischen App-Menu und Menu-Extras so viel Platz wie möglich ist und schon sind alle Menubar-Apps zugreifbar.



Dieses Prinzip ist sehr einfach und man kommt ebenso schnell auf die Idee, einfach ein simples Programm mit minimalem App-Menu zu erstellen, wie es beispielsweise auch toHide, Menuola oder NoMenubar machen. Es bleibt dann nur noch das Problem des Startens dieses Programms.

AccessMenuBarApps geht einen kleinen Schritt weiter und stellt auch gleich eine Lösung für das Aktivieren und Deaktivieren der App mit dem minimalen Menu per Tastaturkürzel, Mausgeste oder Menubar-Icon. Die App bleibt im Hintergrund aktiv und muss nicht jedes mal neu gestartet und beendet werden, sondern die App muss nur in den Vordergrund bzw. Hintergrund gebracht werden.

Möglichkeiten der Aktivierung und Deaktivierung

per Tastaturkürzel

Mit einem Tastenkürzel (per default ⇧Space) kann jederzeit zwischen der aktuell aktiven App und AccessMenuBarApp umgeschaltet werden. Das Tastenkürzel kann individuell angepasst werden, falls das Tastenkürzel schon anders belegt sein sollte. Falls kein Tastenkürzel festgelegt wird, wird automatisch das Tastenkürzel ⇧Space aktiv.

per Mausgeste

Mit einer Mausgeste kann jederzeit zwischen der aktuell aktiven App und AccessMenuBarApp umgeschaltet werden. Es werden vier Arten von Mausgesten unterstützt:

- **Single Knocking** — einfaches Anklopfen am oberen Bildschirmrand
Wenn man mit dem Mauszeiger einmal an den oberen Bildschirmrand klopft wird umgeschaltet.
- **Double Knocking** — zweifaches Anklopfen am oberen Bildschirmrand
Wenn man mit dem Mauszeiger zweimal an den oberen Bildschirmrand klopft wird umgeschaltet.
- **Pushing** — Drücken gegen den oberen Bildschirmrand
Wenn man mit dem Mauszeiger länger gegen den oberen Bildschirmrand drückt wird umgeschaltet.

- **Point And Click** —-Zeigen und Klicken am obersten Bildschirmrand
Wenn man mit dem Mauszeiger auf den obersten Bildschirmrand zeigt und klickt, wird umgeschaltet

Die Unterstützung von Mausgesten kann im Einstellungsdialog komplett ausgeschaltet werden und ggf. auf eine Art von Mausgeste beschränkt werden. Im Einstellungsdialog kann ebenfalls ein Bereich des oberen Bildschirmrandes festgelegt werden, der auf Mausgesten reagieren soll. Hiermit kann man beispielsweise Konflikte mit „Aktive Ecken“ von Exposé vermeiden.

per Menubar-Icon

Sofern das Icon von AccessMenuBarApps in der Menubar zugreifbar ist (das ist ja eigentlich das zu lösende Problem:-), kann durch Klick auf dieses Icon zwischen der aktuell aktiven App und AccessMenuBarApps umgeschaltet werden. Wenn diese Option nicht gewollt ist, besteht die Möglichkeit, das Menubar-Icon auszublenden.

per Dock-Icon

Mit einem Klick auf das Dock-Icon kann zwischen der aktuell aktiven App und AccessMenuBarApps umgeschaltet werden. Wenn diese Option nicht gewollt ist, besteht die Möglichkeit das Dock-Icon auszublenden.

Weitere Einstellungen

Sound Effekte

Im Einstellungsdialog kann (als Spielerei) festgelegt werden, ob beim Umschalten zwischen den Apps Soundeffekte genutzt werden sollen. Sind die Sound Effekte aktiv, so ist beim Anklopfen an der oberen Bildschirmkante ein Klopfgeräusch, beim Drücken gegen die obere Bildschirmkante ein Türknarren und beim Umschalten ein Zuschlagen einer Tür vernehmbar.

Sind die Sound Effekte aktiv, dauert das Umschalten zwischen den Apps etwas länger, da das Umschalten mit den Soundeffekten synchronisiert ist.

Visuelle Effekte

Das Umschalten zwischen den Menüleisten sowie die Mausgesten-Steuerung kann durch visuelle Effekte unterstützt werden. Hierbei sieht man unter anderem den aktiven Bereich der oberen Bildschirmkante, der auf Mausgesten reagiert. Im Einstellungsdialog können diese visuellen Effekte an- bzw. ausgeschaltet werden.

animiertes Icon im Dock

Wenn Sie das Dock Icon zum Umschalten zwischen den Menüleisten nutzen, dann können sie das Dock Icon animieren. Es gibt die Möglichkeit, das Dock-Icon von AccessMenuBarApps in eine analoge Uhr, die die aktuelle Uhrzeit anzeigt, oder in ein Augenpaar, das den Maus-Cursor verfolgt, zu verwandeln.

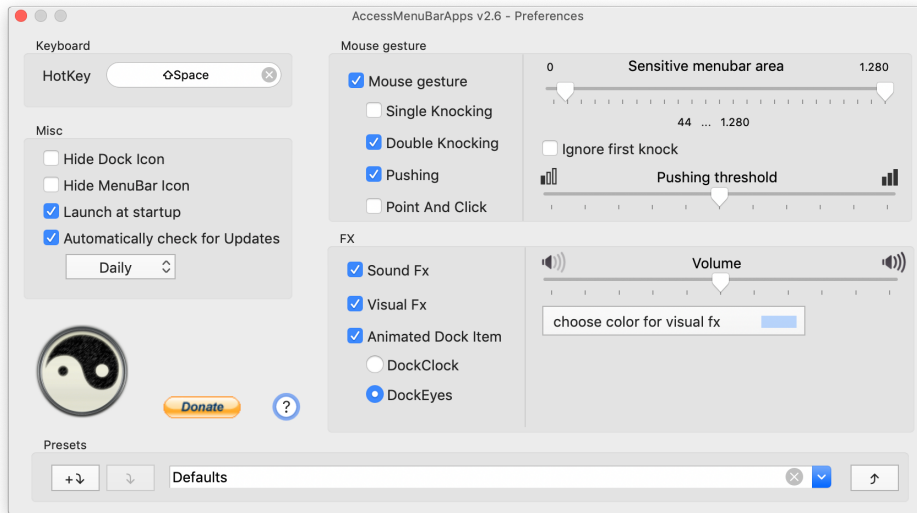
Die analoge Uhr im Dock (DockClock) oder das Augenpaar (DockEyes) kann über den Einstellungsdialog oder das Dock-Menü ein- und ausgeschaltet werden.

Beim Systemstart aktivieren

Im Einstellungsdialog kann festgelegt werden, ob AccessMenuBarApps automatisch beim Systemstart gestartet werden soll. Ist das der Fall, wird AccessMenuBarApps als erste App der Anmeldeobjekte eingetragen, so dass die App möglichst weit rechts in der Menubar sitzt.

Speichern von Einstellungen

Wenn man AccessMenuBarApps mit verschiedenen Konfigurationen für verschiedene Situationen nutzen möchte, so kann man Einstellungen als Preset speichern und bei Bedarf schnell abrufen.



Dark Mode

AccessMenuBarApps unterstützt den Dark-Mode von macOS Mojave durch geeignete Farben und Icons.

