

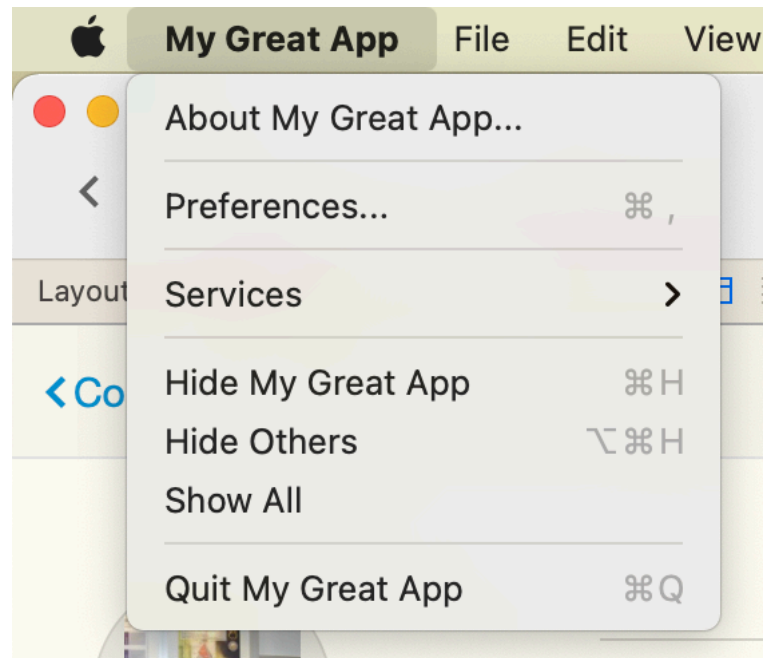
# **Customize your FileMaker solution**

When you deliver your FileMaker solution on macOS, you may customize the look of your application. This may include custom menus, toolbars and dock icons.

## **Customize Menu**

You probably use a custom menu in FileMaker when you do a serious solution. Because you may not want to expose all the normal menu commands to your users, but reduce the feature set to what they actual need.

Early in the start script you may want to call [App.SetProcessName](#) to set your solution name. This defines for example how the top left menu next to the Apple Menu is named on macOS.



After you set the custom menu, you may further customize the menu and rename the content of the some menu entries. For example here we change the hide command to have a different name:

```
MBS("MenuBar.SetMenuCommandTitle"; "FileMaker Pro Advanced¶Hide  
FileMaker Pro Advanced"; "Hide MyApp")
```

If you change the process name, you may need to change the menu path in the call.

## Customize Dock

You can put an alias on the desktop for your solution to double click and launch it. The alias can have a custom icon and point to the fmp12 file. Then when the application launches, you may want to have the same icon in the dock visible for your user. Just call [App.SetDockIcon](#) function and pass the image you like to use. This should be a PNG file in a container with a proper mask.

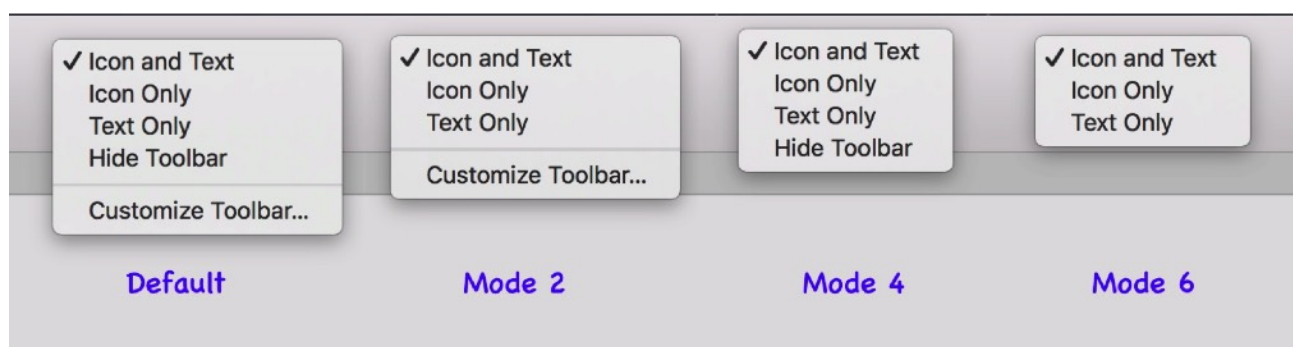
Does your application shows incoming messages, tasks to do, orders to process or other ways things to do? You may want to use [App.SetDockBadgeLabel](#) to show the number of items in the dock icon, just like mail.



## Customize Toolbar

The [Window.SetToolbarConfiguration](#) function allows you to set a custom toolbar configuration. You can define what toolbar buttons to show when you come to a window. You may not want to show buttons to export or delete records.

You may also use [Window.ConfigureToolbarMenu](#) to configure the toolbar contextual menu and disable it. Or you just remove the customize toolbar command, so the user can't add the items, which you disabled before.



## Customize Dialog

If you need to show a dialog with custom icon, you may want to use our [dialog](#) functions and use [Dialog.SetIcon](#) to put in a custom icon for the picture. And on that way you may do dialogs with four buttons or other special things. See our [Dialog](#) functions for details.

Since [dialog](#) functions work for macOS and Windows, you may use them in several places. We also have [ListDialog](#) functions to show a list to pick items. To show progress, we have [ProgressDialog](#) functions, where you can also customize the appearance with an icon.

## Customize Window

A window can change the title. It doesn't need to be the name of the file, but with [Window.SetTitle](#) you can rename it. Be aware that if you refer to windows by name in plugin functions, you may then need to use the new name since `Get(FileName)` will not match any more.

If you like the user to not be able to close a particular window, just use [Window.SetCloseButton](#) function to disable the close button. Similar you can use [Window.SetMiniaturizeButton](#) to disable minimization or [Window.SetZoomButton](#) to disable zooming.

If you don't like the elastic scrolling on macOS for a form layout, please use [Window.Scrollbar.SetScrollElasticity](#) function when entering or leaving a layout. You may want to turn it on when switching to list display and turn it off for form view. Internally form view is a list in FileMaker to show only one item.

Some people liked the feature to scroll through records with the mouse wheel. If you like to get that behavior back, you may use [FM.MouseWheelScriptTrigger](#) function to define a script to trigger via mouse wheel.

## And Windows?

For Windows, I'd like to point you to [Window.SetTitleIcon](#) and [Window.SetTitle](#) functions to change the title of the window and the little icon next to it. You may also use [Window.SetProgress](#) to show progress of work in the taskbar for your window.

We have [Files.CreateAlias](#) function implemented for Windows, so you can create a shortcut to your database file on the desktop. And if you have an icon file, you may give it a custom icon for your solution.

Let us know if you have more ideas to customize.