



Creating an add-on in Claris FileMaker Pro

Purpose

This document covers the steps needed to develop, use, and distribute add-ons using Claris FileMaker Pro.

About add-ons.

FileMaker Pro 19 introduced support for add-ons. Add-ons are predesigned sets of components that work together to provide rich functionality to your app — like calendars, kanban boards, and photo galleries. Besides the add-ons included with FileMaker Pro 19.1.2, you can download additional add-ons from Claris Marketplace or third-party sources.

FileMaker Pro 19 provides the ability to create add-ons to use in your projects or provide to others to enhance the functionality of their custom apps.

The ability to create add-ons in FileMaker Pro is provided as a tech preview at this time. The method of creating add-ons may change in future versions of FileMaker Pro.

Creating an add-on.

Creating an add-on involves adding and grouping objects on one or more “special” layouts and “packaging up” those objects using the **Save a Copy as Add-on Package** script step. When executed, this script step stores its results in the following location:

Windows	AppData\Local\FileMaker\Extensions\AddonModules
macOS	Library/Application Support/FileMaker/Extensions/AddonModules

Create a special layout.

The process of creating an add-on in FileMaker Pro begins by creating a special layout. Name this layout:

`__FMAddOnTemplateDirectives_en`

- The special layout name begins with two underscore characters ("__").
- Replace "en" with any ISO 639.1 language code. If you want your add-on to include more than one localized layout, create different layouts with the appropriate ISO 639.1 language code. For more information on ISO 639.1 language codes, see https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes.
- This layout is only for creating your add-on. The resulting add-on package will not include this special layout.
- The file where you create your add-on needs to have a unique file ID. A file ID is a numeric identifier for a data file. Creating a new FileMaker file creates a new, unique file ID. Alternately, you can navigate to **File menu > Manage > Security > Advanced Settings > File Access** and click **Reset All...** to generate a new file ID in an existing FileMaker file

The sole purpose of this layout is to create, arrange, and group the objects that will be part of your add-on.

Add objects to special layout.

Arrange objects on your layout for your add-on to function correctly. As part of this process, you can include the following in an add-on:

- Buttons
- Custom functions
- Custom menus
- Layouts
- Object anchor settings
- Object styles and theming
- Sample data
- Scripts and script folders

- Table schema
- Table occurrences

- Add-ons do not support security features such as accounts, passwords, and privilege sets at this time.

Group objects on special layout.

Group all layout items/objects that you want to be part of the final add-on package together. The layout should only contain one and only one group of objects. This add-on package will not include any objects that are not part of the group of objects. The group's name can be anything you choose.

- If you DO NOT include a button in the group of objects, it will not be part of the add-on package. However, any script that is attached to that button will be part of the add-on package.
- Whatever group name you use will be visible to those who use your add-on when viewed in Layout mode. It is NOT necessary to name this group.
- If your add-on contains additional layouts, make sure the layouts are associated with table occurrences that apply to your add-on. If you are not using the layout(s) to store data — consider deleting the table occurrence tied to the layout.
- If an add-on might appear multiple times on the same layout, consider naming your group [**<^FMXML_AddonInstanceUUID>**]. Doing so will provide a unique object name to an add-on if added to a layout multiple times.

Execute Save a Copy as Add-on Package script step.

When you have the necessary objects and other elements defined for your add-on, execute a script that contains the **Save a Copy as Add-on Package** script step to create your add-on.

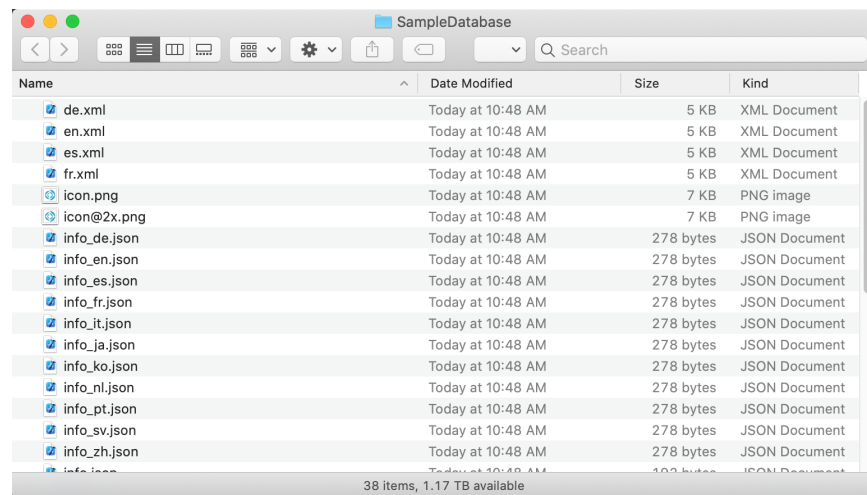
The **Save a Copy as Add-on Package** script step contains two parameters.

- Pass in the window name of the add-on window. You can use a calculation to customize the name of the window if needed.
- Replace UUIDs: The UUID is how FileMaker Pro differentiates between add-ons. If two add-ons have the same UUID, then only one of them will display. This is important if you use the same FileMaker app to generate different add-ons. In this case, you will want to make sure that Replace UUIDs is set for "On." If you use a separate file for each add-on you create, replacing the UUID is

less critical.

- To avoid having the script that creates the add-on be part of the actual add-on, consider running the script that contains **Save a Copy as Add-On Package** from a separate file.
- By default, when you create an add-on, the add-on's name will be the name of the file it was created from. You can rename the add-on after it has been created if you need to.

When the Save a Copy as Add-on Package is successfully run, FileMaker Pro will automatically open a window for you to view the generated files.



- The info_xx.json files can be edited to customize your add-on further. See the **Customizing Your Add-On** section for more details.
- These files can be compressed ("zipped") and distributed to others. See the **Distribute your Add-On** section for more details.

Customizing your add-on.

Edit the info_xx.json information file.

As part of the add-on creation process, a JSON formatted "info" file is created as part of the package. This info file is named:

info_xx.json

...where "xx" is replaced with the ISO 639.1 language code used as part of the special layout or layouts of your add-on. This info file contains the following information that can be customized:

Title	This is the name of your add-on. This name will also appear in the install dialog on the Add-ons tab in the left pane.
Description	A summary of the usage and functionality of your add-on. Consider also adding versioning information here too.
Category	Helps in grouping your add-on in the install dialog box
Features	Entered in as a JSON array, this section provides a list of features and functionality of your add-on.
Optimized	Displays information about how an add-on will work on different devices and platforms.

Here is an example of a properly formatted info_en.json info file:

```
{
  "Title" : "Sample Pricing",
  "Description" : "This is a simple pricing add-on that can be set up quickly.  Version 1.00",
  "Category" : "Pricing and Invoicing",
  "Features" :
  [
    "Pricing" , "Invoicing"
  ],
  "Optimized" :
  [
    "Desktop", "Tablet", "Mobile"
  ]
}
```

Customize the icon for your add-on.

When you create your add-on, FileMaker Pro provides the ability to customize the displayed icon when installing your add-on.

By default, the icon is a Claris logo. But, you can replace the following files with your own icons if you wish:

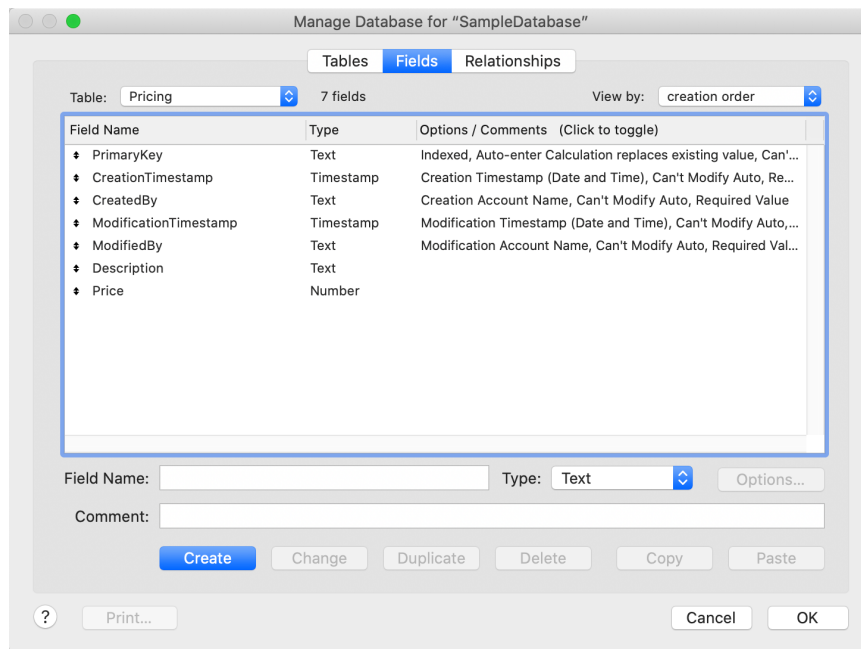
- icon.png
- icon@2x.png

Using relationships with an add-on.

You can design and use add-ons to create relationships automatically within a custom app. Providing this functionality requires a few more steps to accomplish.

Add objects to special layout.

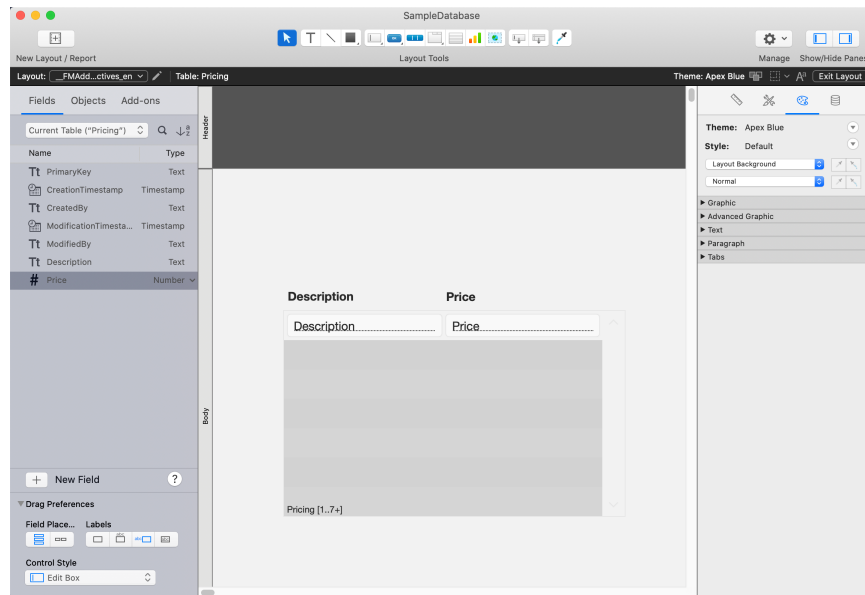
For this example, our add-on will consist of a portal for capturing pricing information. The context of the layout will be the “pricing” table. The pricing table will have a description and a price field to go along with the default fields that FileMaker Pro generates when creating a new file.



On the special layout (see CREATE A “SPECIAL” LAYOUT section above), add the necessary portal and portal fields to your layout. Your layout might look something like this:

Group objects on special layout.

Group your portal and portal fields.



Add a foreign key to the special layout.

Next, you need to set up a foreign key. A foreign key is a field in one table that identifies a record in another table. Values in foreign keys do not have to be unique in the table, and they can be empty (null). Define the foreign key in the same table as your special layout. Add this foreign key to your special layout and give the foreign key field (not the field label) an object name of:

```
<RelationshipName> | RightTable
```

Replace <RelationshipName> with the name of the relationship used in your add-on. In our example, the foreign key field object name is:

```
Pricing | RightTable
```

Group and name the foreign key field and field label.

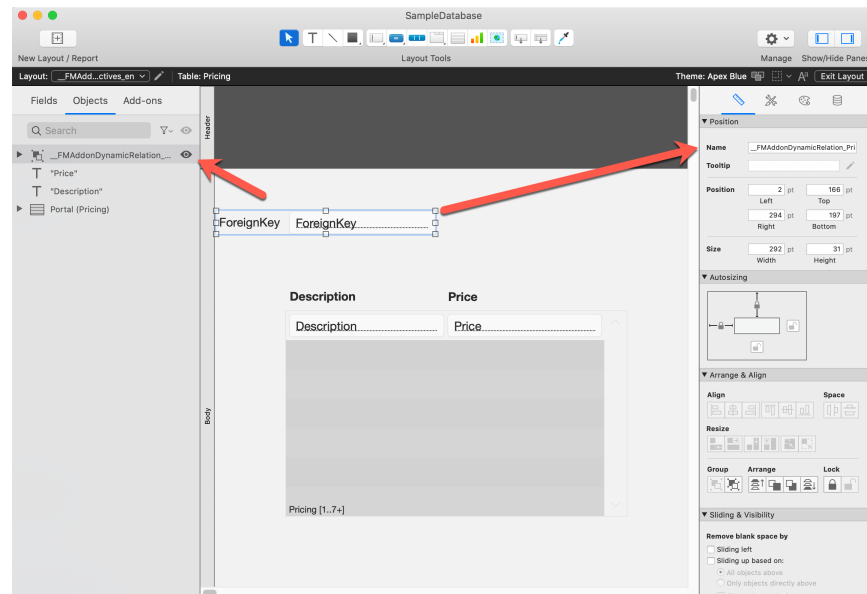
Next, highlight the foreign key field and its field label then group them. Name this group as follows:

```
__FMAddonDynamicRelation_<RelationshipName>
```

Replace <RelationshipName> with the name for the foreign key above. For our example, rename the group as follows:

```
__FMAddonDynamicRelation_Pricing
```

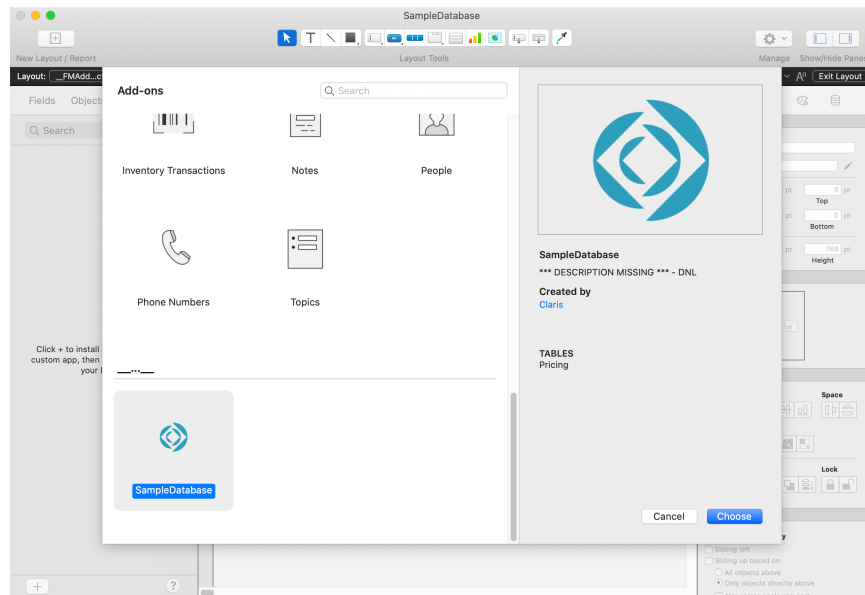
- The group name begins with two underscore characters.
- There is one underscore character between the name of the group and the relationship.
- The foreign key field and foreign key field label WILL NOT be part of the add-on when users drag the add-on onto their layout.



Using your add-on.

After you have created the add-on, relaunch FileMaker Pro.

When you relaunch and switch to Layout mode, the newly created add-on will display in the Add-ons tab in the left pane. You can install the add-on by selecting it and clicking Choose. Once installed, to use the add-on, drag and drop it onto the desired layout.



- Customizing the info_xx.json file will populate the information on the screen's right side when selecting/choosing an add-on to install. See the **Customizing Your Add-On** section for more details.
- Installing an add-on installs all parts of an add-on, including supporting schema, scripts, themes, layouts, etc.

Distribute your add-on.

To distribute your add-on so that others can use it, compress (zip) the folder that contains your add-on. Those that want to use your add-on can unzip this file in the following location:

Windows	Users\username\AppData\Local\FileMaker\Extensions\AddonModules
macOS	/Users/username/Library/Application Support/FileMaker/Extensions/ AddonModules

- Relaunch FileMaker Pro after copying files to the above location(s).

Once you install an add-on, it is available to anyone who designs an app using that copy of FileMaker Pro.

Summary

As you can see, creating an add-on in FileMaker Pro is easy and provides the ability to add additional functionality to your custom apps in a fraction of the time it would take to build something from scratch. Use them or make them on Claris Marketplace so that others can benefit from your development work. We look forward to seeing what you can do with add-ons in the future.