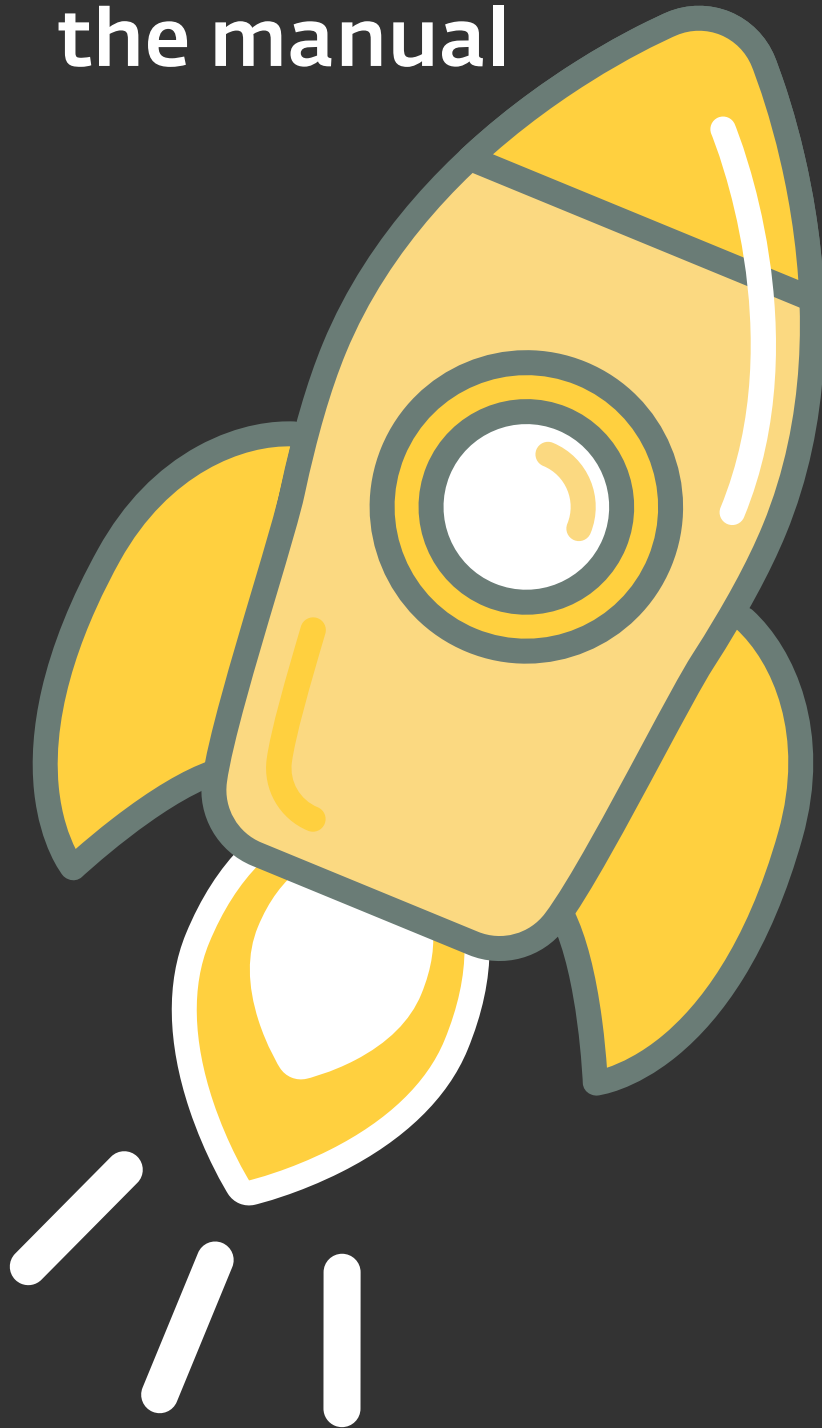


FM STARTER 3

the manual



Starter file for FileMaker projects

FM Starter 2 Handbuch

Manual for FM Starter 3, English

Manual version 3.2, November 12, 2024.

FM Starter 3 is a starter file for new FileMaker projects and a product of Kursiv Consulting, Netherlands.

Developer and copyright: Karsten Risseuw.

Learn more

fmstarter.com/

Index

Introduction.....	6
Welcome.....	7
Passwords.....	9
What is new?.....	11
Requirements and disclaimer.....	12
Learning videos.....	13
First steps.....	14
Basic concepts.....	15
6 Basic concepts.....	16
FileMaker's Sex-Appeal.....	18
Structures before rules.....	19
Start small and think big.....	21
You need a dashboard.....	23
No conventions.....	25
6 Tips for an easy start.....	28
Documentation.....	29
This manual.....	30
Help function.....	31
Comments in the scripts.....	33
ToDo module.....	34
How does that work?.....	36
Working with a structure.....	37
Special windows.....	38
Everything is modular.....	39
Interface design with Themes.....	41
No limitations.....	43
Your first project.....	45
First things first.....	46

Step by step.....	48
Create your own solution.....	49
Login.....	50
Dashboard settings.....	53
Application and ABOUT the software.....	54
Basic settings.....	56
Startup Script.....	58
Navigation.....	61
Navigation quick guide.....	62
Rapid application development.....	66
Using FileMaker security.....	67
Security basics.....	68
Security settings.....	69
Access rights.....	70
Additional security settings.....	74
Building your app.....	76
Adding tables.....	77
Adding scripts.....	78
Adding modules.....	79
Structure your file.....	82
Windows.....	84
Why windows?.....	85
User windows.....	87
Main window.....	88
Help window.....	89
Developer windows.....	91
Developer dashboard.....	92
Multiple developers.....	95
Toolbox.....	98
ToDo window.....	105

Modules.....	115
What are modules?.....	116
FM Unlock.....	118
FM Unlock: Introduction.....	119
FM Unlock: Test mode.....	122
FM Unlock: License types.....	130
FM Unlock: How it works.....	133
GNV Global Navigation.....	135
Navigation table.....	136
Navigation settings.....	140
Enlarging the buttonbar.....	144
Multiple platforms.....	145
QFI QuickFind.....	146
QSO QuickSort.....	147
MTL Multilingual Text Labels.....	148
How does it work?.....	150
Languages.....	152
Editing languages.....	153
Working with the ToolBox.....	156
Translating value lists.....	158
DeepL translations.....	161
SBU Simple Backup.....	167
HELP Help Modul.....	170
UAC User Accounts.....	173
User accounts.....	175
Expiring passwords.....	177
Default administration account.....	180
Rights.....	181
GDPR compliance.....	183
Data separation model.....	186

ToDo developers.....	187
Plug-ins.....	192
Plug-ins: Installation.....	194
Plug-ins: Registration.....	196
Design.....	199
Introduction about design.....	200
Creating the user interface.....	201
What are themes?.....	202
Theme overview.....	205
FMS01 Standard Theme.....	208
Support.....	213
Links.....	214

Introduction

An introduction to FM Starter.

Welcome to FM Starter

Hello FileMaker adventurer. Nice to have you on board!

This introduction will help you get the most out of FM Starter. We'll talk briefly about FM Starter, what it is and what it isn't, and how to work with it. The rest of the manual can be referred to as needed while working with the software.

What is FM Starter?

FM Starter is a starter file for new FileMaker projects. It is a set of powerful and well-integrated modules that provide basic functionality for almost any FileMaker project. This allows you to immediately focus on your actual project instead of wasting time on the basics. Start your next FileMaker project with FM Starter!

FM Starter comes with the following basics:

File structure

- Modules for basic functionalities
- Standard theme for good design
- Good integration of all components.

FM Starter is also the perfect start to learn to work better with FileMaker. It's an open file, well documented, that shows you at least one way to get started. If you are just getting started with FileMaker or thinking about how to create your own starter file, this can give your ideas a boost.

What is FM Starter not?

FM Starter is not a ready-to-use solution. It is neither an address database nor an ERP solution. It does, however, provide the basis for developing all of these things. It's like an empty truck - it drives and carries big loads, but it's up to you to decide what you want to transport. As a starting point, it's not perfect for every conceivable use. You may need to customize the design or features, add modules or change settings. FM Starter is designed to do just that. This is how it should work. FM Starter saves you tons of time, but doesn't free you from the hassle of developing it yourself.

Those who use it, benefit from it!

Next steps

Watch the [video tutorials](#) to get a first impression or go to Getting Started to learn more about the [concepts behind FM Starter](#).

Passwords

You have just downloaded FM Starter. Congratulations!

Where are the passwords?

The passwords are part of the delivery. A text file with the names and passwords can be found in the “FM Starter” folder.

Please note that there are 2 types of names and passwords:

1. Developer password for full access
2. Administrator password to simulate regular user access.

Please note that these settings are only an aid for getting started. Familiarize yourself with how to use them and then change them as you wish to create your own level of security.

Developer access

To be able to access the file as a developer, you must force the pop-up window of the standard FileMaker password. This is done on Mac by pressing the ALT key when opening the file. On Windows, press the LEFT SHIFT key when opening the file. The standard FileMaker password window opens, you enter the name and password and then have full access to the file.

Alternative: As soon as the file is opened, even with administrator access, you have a new login in the navigation. This will also open the standard FileMaker login window.

If you do not want this, you can deactivate the UAC user account module and change the software as you wish.

Administrator and user access

If you just double-click on the file, the user administration module starts and prompts you to enter a valid user password. The file has a default administrator account configuration that you can (and should) change when using the solution. This user management is enabled by default for the FM startup file, but you can disable the user management module in the developer dashboard if you have a simple solution without needing a comprehensive user management tool.

Find more help on this directly on fmstarter.com.

New features

The upgrade is the result of many small development steps. All new features are described in the module "ToDo". There you will find a complete list in a separate window.

Updates

When you buy FM Starter or newer, you purchase a user license with a full year of free upgrades. You will receive all new versions for one year.

If you would like to receive support and new functions after this year, you will need to update the free upgrades. To do this, you need to renew your license. To do this, log in to your account at fmstarter.com. Once you are logged in, purchased products automatically list your personal discounts.

Requirements and disclaimer

Requirements

FM Starter consists of a single FileMaker file. It does not work in and of itself, but requires a full version of FileMaker Pro to use the file.

Minimum requirement: FileMaker Pro 19.1 for Mac or Windows. The reason for this is the add-on for automatic DeepL translations, which is already integrated here. If newer functions are used in the future, you may need to use the latest version. Version 19.1 is the minimum version that you can use to open this file. This does not mean that the scope of performance of FM Starter will remain at this version 19.1 forever. We recommend that you always use the latest version of FileMaker.

Get your FileMaker license here: filemaker.com

Disclaimer

FM Starter is not intended to fulfill any particular purpose, task or requirement beyond what the file contains. You receive a solution as is, with no further guarantees. Note that it is intended that you extend the functionality yourself and customize it to your needs.

Many variants can be developed for each specific task. To serve as many FileMaker developers as possible, FM Starter does not integrate all possible options. Simplification is a friend of the developer.

Nevertheless, we welcome suggestions and feedback via support@kursiv.com.

Learning videos

Looking for an easy way to get started? Take a look at the documentation for this product on fmstarter.com. You will find videos and other supporting materials there.

First steps

Let's go!

Basic concepts

It is easier to work with FM Starter if you understand some of the underlying concepts. That's what this chapter is all about.

Understanding what is going on

FM Starter is the result of many years of FileMaker development. Over time, you learn which things work well and which don't. You see that certain development approaches are faster, easier to understand and easier to use. Even though FM Starter is just one file, it's like a condensed version of that experience.

One of the hardest things in development is to simplify things. **The simpler a solution, the easier it is to maintain and develop later.** The time it takes to simplify is time well spent for the future.

The following concepts form the basis for FM Starter:

1. **Structures are more important than rules.** Simple folder structures, for example for layouts or scripts, support clarity.
2. **Separation of parameterization and functionality.** As far as possible, settings are stored in normal layouts. This simplifies the setup of the start file. Developers have their own window for this, the “Developer Dashboard”. All settings that are important for the developer are stored in this developer dashboard. The settings for your own further developments can also be stored here.
3. **Modular approaches.** FM Starter works a lot with modular concepts, namely FileMaker modules and add-ons. These already work in themselves, with their own tables, layouts and scripts. The same approach has been used everywhere because it has proven itself over the years.
4. **Multilingualism.** FM Starter is multilingual from the ground up. This has many advantages, as all labels are managed centrally in a table. It is advisable to use this solution even if you only want to set up one language. Help texts are also stored here and called up by the HELP module.
5. **Dependencies.** As far as possible, mutual dependencies between modules are avoided. In FM Starter, however, some modules deliberately use the possibilities of other modules. For example, the navigation is multilingual and also uses information from the user file to automatically create a navigation for different user groups.
6. **Use relations correctly.** FM Starter does not use relations because this is not necessary for the scope of services. However, this only applies to the scope of FM Starter. Normally you will use relations between tables as soon as you integrate your own tables with data. Every developer is free to use a relation model of their choice. We strongly recommend to consider the Anker-Boje model, which is used by most developers

worldwide. This model helps with clarity and facilitates both development and subsequent maintenance.

More on these points in a moment.

FileMaker's Sex-Appeal

FileMaker has great sex appeal as a development platform. FileMaker is a so-called low-code development software. It is easy to use for beginners and therefore attracts many people from a wide variety of backgrounds. It does a fantastic job in this way and allows people to change their business in a very pragmatic way.

The longer you work with FileMaker and the more you learn, the more you realize that more skills are needed. If you want to increase the potential and professionalize your projects, you need to learn new things: build new techniques, develop design skills and much more.

This is where FM Starter comes in. FM Starter was developed for those who, like myself, have a very different and non-technical background and want to advance their own work.

Other typical users of FM Starter are developers who would like to create their own starter solution but simply don't have the time. FM Starter can also simplify your life.

FileMaker is a really great piece of software designed to make things happen. FM Starter allows you to get started with FileMaker development much faster.

More structure-based, less rule-based

FM Starter is not perfect, and it's certainly not the only approach to doing things. There are many developers out there, and almost everyone has their own preferences. Some developers work intuitively and their applications have an organic growth. Others are more like architects and carefully plan every detail before they start work. Still others prefer to work with some sort of set of rules, to the point where most of the work is related to following those rules.

I'm not going to judge any of these approaches. We will probably all be at each of these stages at least once in our working lives. Some approaches prove helpful, while others have limitations. Having no structure at all is a limitation, but over-regulation can also complicate things. An alternative to these extremes could be to focus on transparency.

For example, to create transparency, one can consider the following:

- Keep each step or function simple and focused.
- Document every step
- Maintain a simple structure.

Transparency should help you to do the right things. It's not an explanation of details, but a way of approaching things. Instead of getting lost in the details, it's a big help to get a simple structure. A structure answers the question of where to find something rather than how to do something. While you'll probably change and improve your programming skills over time, you don't have to reinvent every single step you've ever taken. Using a structure allows you to track and maintain your functions independently of programming.

[Modules](#) are a good way to show how this works.

The purpose of a structure is to organize things. To see how this is done, take a look at the following parts of FM Starter:

- Script window
- Layout window
- Database window.

Another way to create a simple structure is to use [separate windows](#), such as for the main application and for the developer settings. This keeps a clean and tidy environment for users and developers, which is easier to maintain than if you hide developer functions on the user pages.

Start small. Think big.

For example, a house can be built with building blocks. The same idea can also be applied to the creation of software.

Focus the development on solving a single detail. Build larger parts from it. Start small. Think big. In this way, complex processes are simplified. Instead of building everything at once, create parts or modules and then build them together - according to your needs.

This idea has been used in FM Starter in a number of ways. The key is the use of modules. Each module solves a specific task. It relies (mostly) only on itself, and it can be used almost anywhere. This applies to [global navigation](#), [multi-lingual text labels](#) and many other modules.

Modules function within themselves. Each comes with own tables, layouts, scripts, etc. This makes it very easy to distinguish one module from another and one function from another. It makes sense to bundle scripts for a specific module. You start to think in terms of group functionality. If your task can be clearly defined, it is logical to add a table for that specific task, even if it only contains a few fields. Custom tables, custom layouts, custom scripts - to keep things neat and tidy.

I used to have tables for specific purposes that stored settings for all sorts of other parts of the application. One central table for everything, so to speak. These solutions turned out to be very difficult to maintain. If you ever want to take out a certain part of the solution to use it elsewhere, you're stuck in these weird alien relationships with other tables. So it's much more elegant to create modular groups of tables, scripts and functions.

Starting small and thinking big is an approach that can grow with your own development. If requirements change, nothing is lost. This approach will guide you and you will be able to find your way out of new challenges.

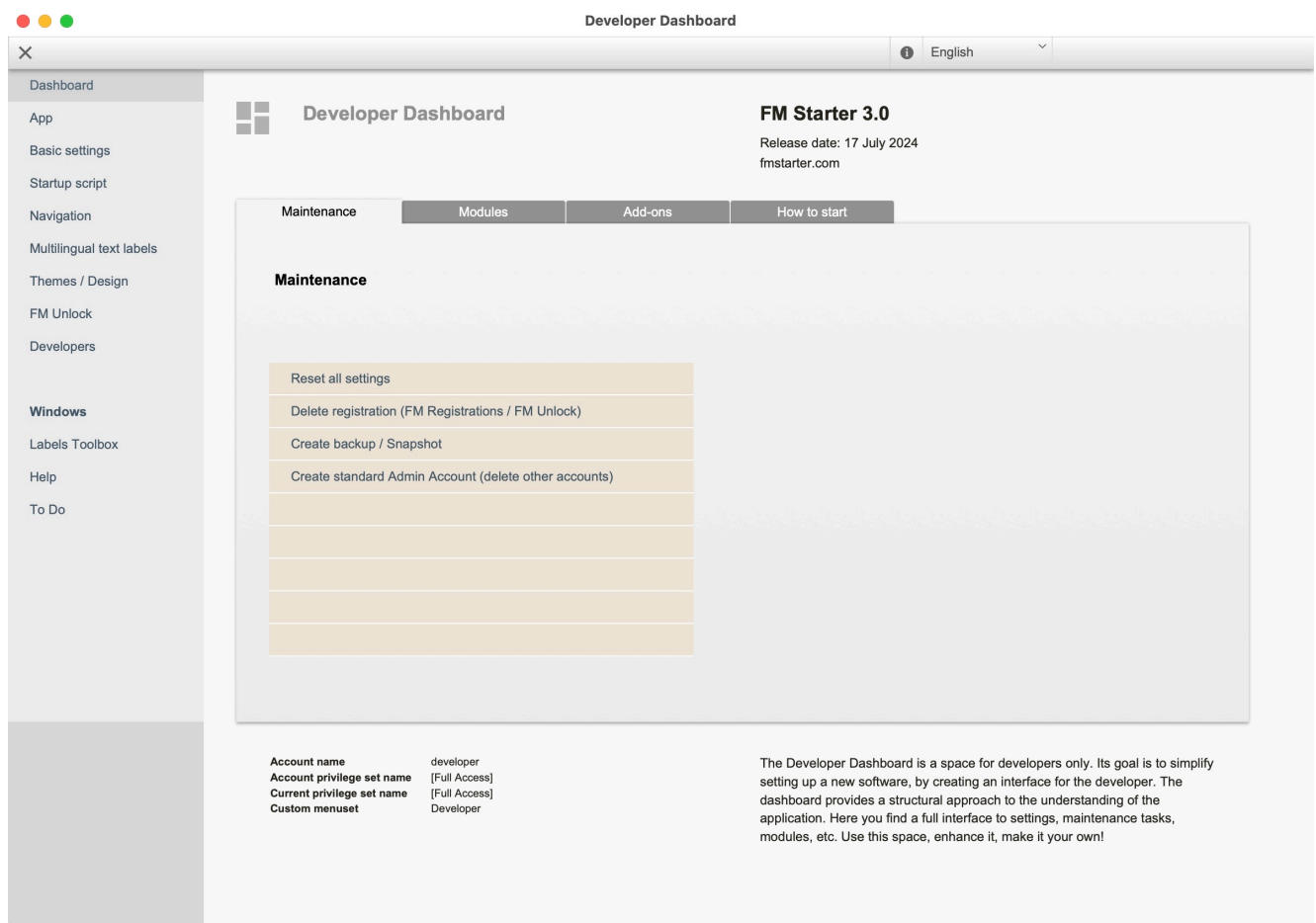
FM Starter is not a system, but a kit and a structure.

Cockpit

As a developer, you need a dashboard. The dashboard is a user interface that is intended just for you. In FM Starter there is a separate window where you can find all your settings, maintenance scripts and other things you need. The dashboard is like a cockpit for the developer.

Make everything editable

The dashboard is a place where you can configure your application. How is that possible? The answer is simple: it was designed that way. In other words: Wherever it makes sense, the settings in your script are not hardwired, but are outsourced to a settings field or variable that can be configured via a suitable interface or another script.



Configuring instead of editing

The idea of making everything editable is not to create extra work, but to make life easier once it has been set up. A good example is the use of multilingual text labels. Instead of writing the information directly into the layout, you can build it up dynamically. For example, which default language should apply? Well, you can configure this in the developer dashboard. Change the language from a popup, press Save, and you're done. You can also add a language field to each user account, which is evaluated at the time of login. This way, each user receives the desired language when logging in.

The control center

Making everything configurable is a good thing. It will save you a lot of time in the future. It will make your customers happy. All other settings of the application are also listed in the dashboard:

- Maintenance scripts
- All modules and their settings
- All add-ons and their settings
- Themes and their descriptions

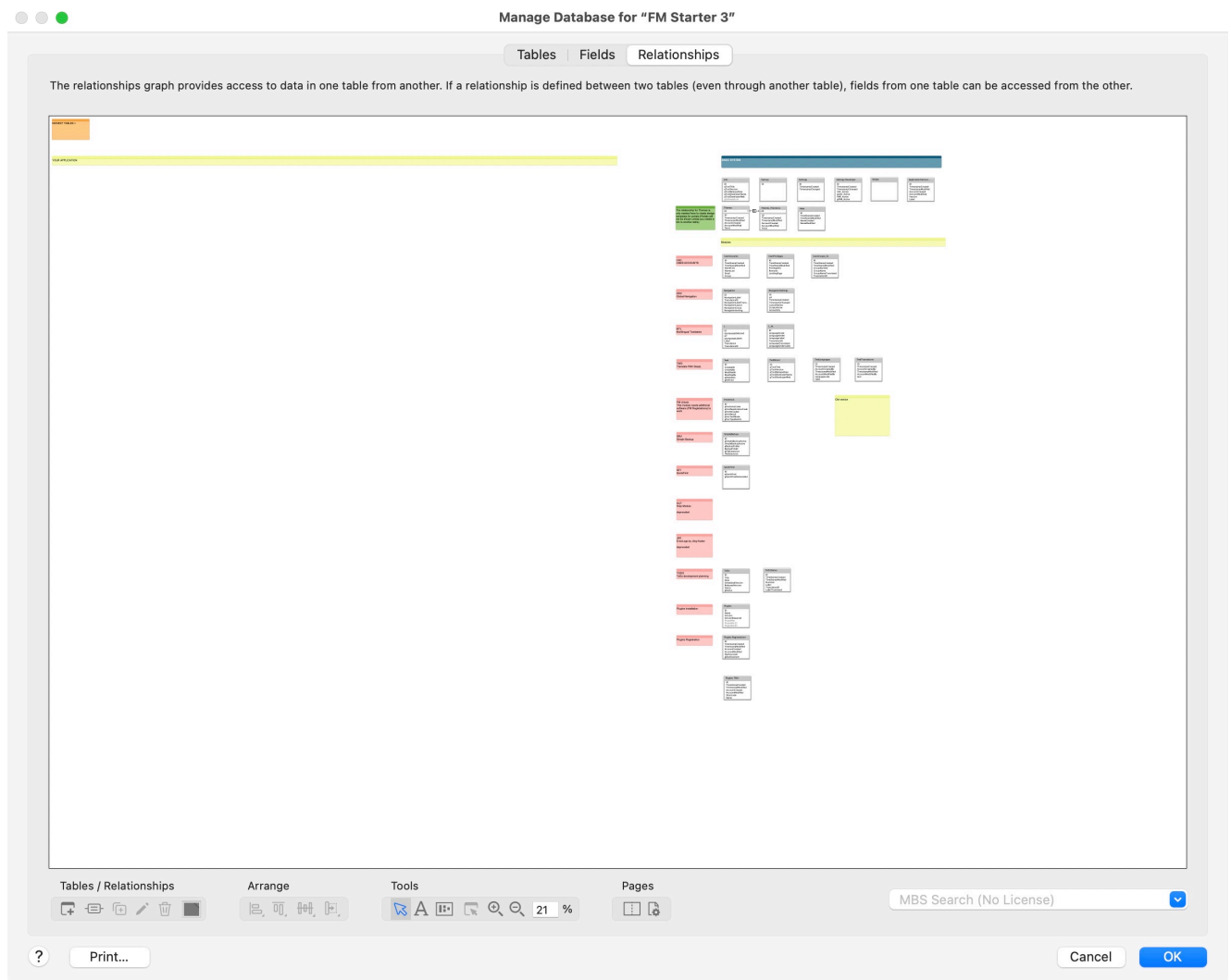
In the developer dashboard, you can reset your application, create a backup, set the version, check your to-do list and much more. When you start adding your own tools to the solution, consider listing them in the developer dashboard as well.

No conventions

FM Starter is independent of any conventions. You will only find a structure, like a Christmas tree, to which you can attach your own glitter and magic.

- There is no convention for file names, field names, etc.
- There is no requirement to describe things in a certain way.
- There is no relation model like selector-connector or anything else that you have to use to make it work.

To make it easy for you, you can add all these things as you like. You can work exactly the way you want. FM Starter itself is completely independent of any conventions. If you look at the relationship diagram, for example, you won't see any relationships at all. Only one relationship is used for the design of portal series, because these cannot be displayed without a relationship.



FM Starter relies on structures rather than conventions. No relationships are used in FM Starter because they are not required for the current scope of services. However, this is not a default, but merely the way FM Starter is structured free of conventions. You choose the relationship model that suits you best for your application. Most developers use the Anker-Boje model, but you can also use other approaches. The purpose of FM Starter is to make it easier for you to get started. That's why you don't need to learn and follow specific guidelines to make your development work.

The relationship diagram above shows FM Starter 3. Only 1 relationship is used, which is only there to enable the creation of cut-out rows. These cannot

be displayed without a relationship and therefore cannot be designed. Because FM Starter has an integrated design template (design, theme), this one relationship was unavoidable, but it has no functional task.

6 tips for a better project start:

1. **Work with a copy of FM Starter.** If something does not seem to work, you can always make a comparison with the original file.
2. **Get to know FM Starter.** Invest some time to watch the videos, get to know the manual and understand the concepts.
3. **Think modular.** FM Starter was largely built using modules or modular approaches. Not everything is realized at once, but step by step and component by component. This approach can also be extended to your project. What are your project components? Develop one at a time.
4. **Slow build-up.** Start with a single project part. Start with the most important table of this project part. First create the basic layouts for data entry and list view. Then create the basic functions with a script that is specifically intended for this table and/or for this layout (create data record, delete, sort data records, etc.). Always keep these tables, layouts and scripts cleanly separated from other parts of the project. This slow approach preserves clarity.
5. **Slow expansion.** Only when the basic layouts with basic functions are in place do you add the remaining tables in this part of the project and proceed in the same way. When all tables with all basic layouts and basic functions are in place, the basic structure is complete.
6. **Implement workflows.** Go from the generic to the specific. Once the basic functionality has been created, it is time for the actual workflows. These should be the workflows for which the application was created. Where necessary, these workflows should also use their own tables and layouts, but in any case their own scripts.

Now also look at the chapter “[Your first project](#)”.

Help functions

FM Starter has several support functions, both for you as a developer and for the end user of your product.

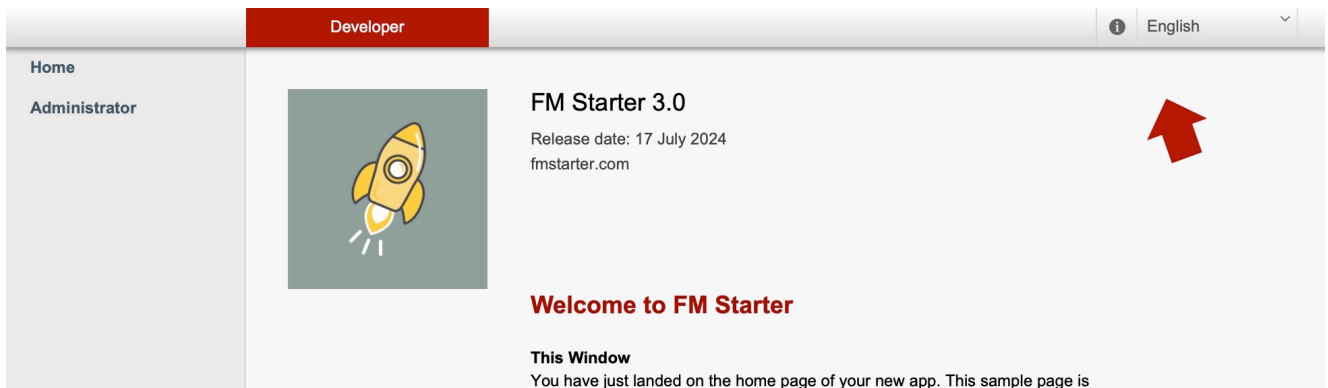
As a first port of call to understand things, you have this manual. It is intended as a reference, especially to find a starting point if something is completely new to you. You can also use the videos on the fmstarter.com website or the help functions directly in FM Starter.

If you are wondering how to create your own manuals, you have many options. This manual was written using Scrivener, which offers the benefits of an offline versatile writing tool with configurable output options.

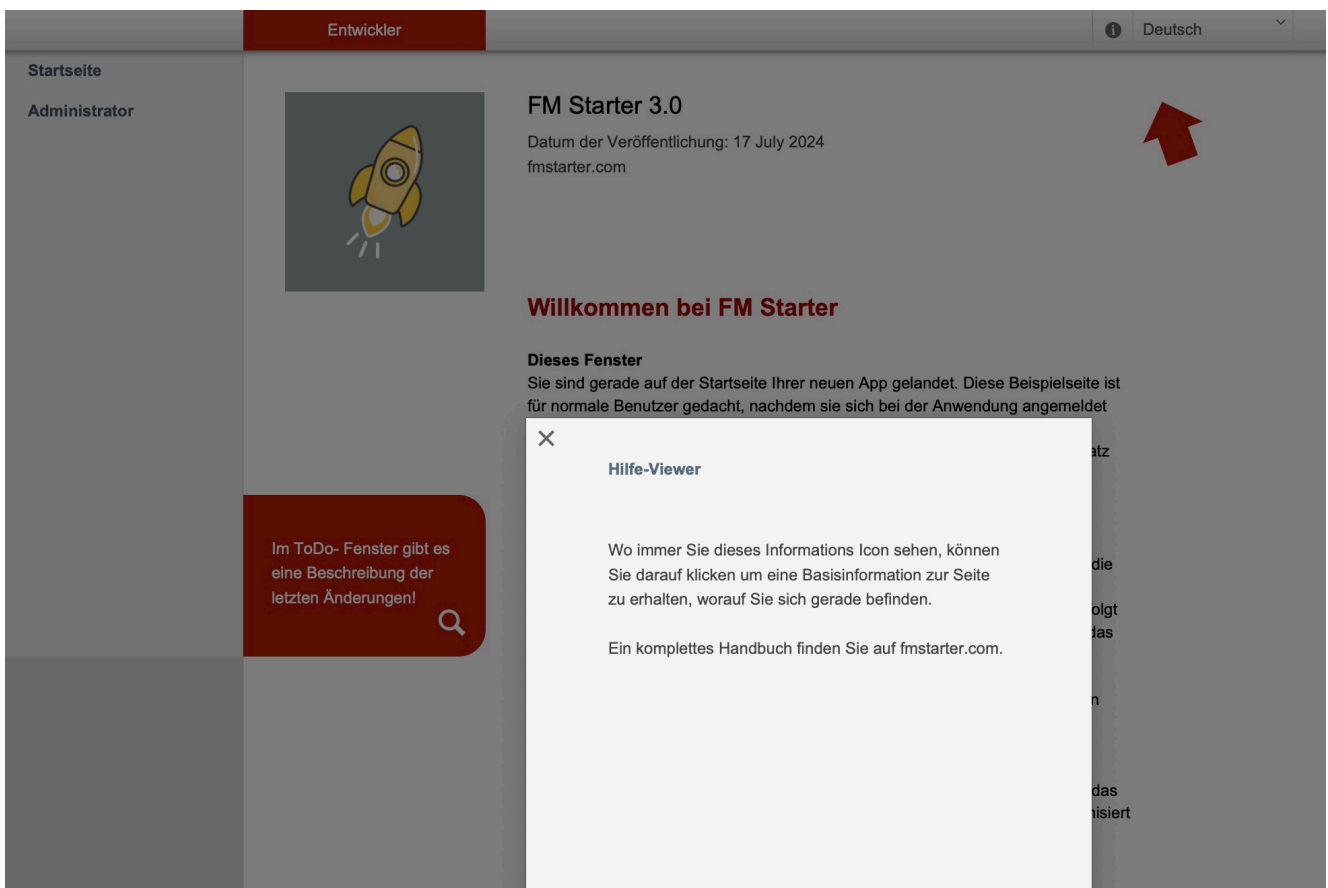
- Literatureandlatte.com

Help function

A help module makes it possible to create any number of information buttons. They are linked to the MTL Multilingual Text Label module and only require the text number of the desired label. This automatically supports all languages of the application.



The help module has been newly created in version 3 and help windows are displayed as card windows:



The help module is used to help the developer, but can also be used for your own application.

- [Help module](#)

Comments in the scripts

Always check the scripts for more information. Scripts have extensive comments, and modules even have complete read-me scripts. Here you will find helpful support for your development.

All comments are in English.

The screenshot displays the 'Scripts' application interface. On the left, a sidebar lists various scripts and modules, including 'FM Starter Startup', 'FM Starter CloseDown', 'MODULES', 'FILEMAKER MODULES HERE', 'FM Unlock', 'UAC User Accounts', 'README - UAC User Accounts', 'LICENSE - UAC User Accounts', 'UAC Settings', 'UAC Public', 'UAC UserLogin' (highlighted), 'UAC Private', 'GNV Global Navigation', 'QFI QuickFind Module', 'QSO QuickSort Module', 'MTL Multilingual Text Labels', 'SBU Simple Backup', 'ToDo Development', 'Plugins Installation', 'Plugins Registration', 'HELP', 'Add-ons', 'DEVELOPER Settings and Checks', 'Scripts for buttons', and 'APPLICATION SCRIPTS'. The main area on the right shows the content of the 'UAC UserLogin' script, which includes version history, login steps, and script parameters.

```
1
2
3 # User Login
4 # -----
5 # Link this script within the startup script with the parameter "startup" to activate it upon launching the file.
6
7
8
9 # History
10 # -----
11 # version 2.7 - script parameters added (landingpage, usersettings) which are called though only from within the part "login". This is new, but not different.
    Simplified a few things.
12
13 # version 2.6 - Corresponding to FM Starter, Version 2.08 > See ToDo-window for details.
    This version has implemented an enhancement to reflect and handle expiring passwords.
    Added script parameter: newpassword
    The login-screen has been enhanced and slides are used to handle the following steps.
    These are the new login-steps:
    1. search the user name
    2. check if the found user has an expiring password
    3. either present the regular password field or create a new password
    4. login.
14
15 # version 2.5 - Bugfixes. If you use an older version, replace the CONTENT of your old script with the CONTENT of this script.
16
17 # version 2.4 - Bugfix. Thanks to Bermward Jelonek for pointing this out.
    In the part "FindUser" the variables set have now been put in Quotes with Quote (...) to avoid characters like "#" and "&" be used and give results as wildcards. An additional IF ()
    has been added to find empty values first.
18
19 THIS PART:
    If [ IsEmpty ( UserAccounts::gAccessName ) or IsEmpty ( UserAccounts::gAccessPassword ) ]
        Set Variable [ $$Feedback ; Value: "Access name or password is missing." ]
        Refresh Object [ Object Name: "Feedback" ; Repetition: 1 ]
        Exit Script [ Text Result: "Access Name or Password is missing" ]
    End If
    Set Variable [ $AccessName ; Value: Quote ( UserAccounts::gAccessName ) ]
    Set Variable [ $AccessPassword ; Value: Quote ( UserAccounts::gAccessPassword ) ]
15
16
17
18 # Declaration of script parameters
    Call this script with one of the following parameters to execute that part of the script.
19 # -----
20 # startup
21 # finduser
22 # newpassword
23 # login
24 # landingpage
25 # usersettings
26 # logout
27 # relogin
28 # showpassword
29
30
31 # Declaration of variables used
32 # -----
33 Set Variable [ $Parameter ; Value: Get ( ScriptParameter ) ]
```

ToDo module

The to-do list is an aid for developers. Links to the tool become visible as soon as you log in with a developer account. You can find the to-do list in the Developer drop-down menu or in the dashboard navigation (developer window).



The to-do list helps you to keep track of issues and functions that you need to work on. You can assign each issue to a release version and assign a status. This little tool has its own window where you can check it while you are working in another window.

To Do

To Do's Deutsch

FM Starter 3.0, 17 July 2024 4 Erledigt

	Nummer	Version	Titel	Status	
				4 Erledigt	
↕ ↗	10096	3.0	Window flashing reduced	4 Erledigt	🗑
↕ ↗	10095	3.0	Old help window removed	4 Erledigt	🗑
↕ ↗	10094	3.0	Layout Details Enhanced	4 Erledigt	🗑
↕ ↗	10093	3.0	GoToLayout enhanced	4 Erledigt	🗑
↕ ↗	10092	3.0	Dutch added as language	4 Erledigt	🗑
↕ ↗	10091	3.0	Smaller fixes and improvements	4 Erledigt	🗑
↕ ↗	10090	3.0	ToDo-window: DefaultTopics	4 Erledigt	🗑
↕ ↗	10089	3.0	Main window size dynamic	4 Erledigt	🗑
↕ ↗	10088	3.0	Fix Unlock Module	4 Erledigt	🗑
↕ ↗	10087	3.0	GN Global Navigation	4 Erledigt	🗑
↕ ↗	10086	3.0	New Help module	4 Erledigt	🗑
↕ ↗	10084	3.0	ToDo-window: Info-Icon	4 Erledigt	🗑
↕ ↗	10083	3.0	Start screen	4 Erledigt	🗑
↕ ↗	10082	3.0	DeepL fix	4 Erledigt	🗑
↕ ↗	10081	2.9	Navigation Settings	4 Erledigt	🗑
↕ ↗	10080	2.9	Plug-ins Registration backup	4 Erledigt	🗑

Although the module is very simple, you can build on it to create something like a customer view on the list, or extend it to a help system so that users can report errors.

You can think of it like this:

“ToDo” is just before “TaDa!”

How does that work?

Delve into some other aspects of FM Starter.

Working with a structure

FM Starter is easy to use, but it is important to understand the concepts. Since most of the functions are divided into smaller parts, all logically grouped, it is advisable to first take a closer look at the structure.

Understanding the structure of FM Starter means this:

- Know and understand the functions of the individual **windows**.
- Everything in FM Starter has a **modular** structure. Each module solves a task with its own tables, layouts and scripts.
- **Add-ons** are a newer variant of modules. They are supported as of FileMaker 19.1.
- Take a look at the **script window** to see how everything is organized.
- Take a look at the **layout window** to see how everything is organized.
- Take a look at the **relationship graph** to see how everything is organized.
- Master your FileMaker basics and turn to filemaker.com when you need something.

Don't panic!

If you're new to FileMaker or if FM Starter looks a little overwhelming to you, don't worry. You'll soon find your way around.

As a suggestion: Make a copy of your FM Starter file and play with it.

Learning by doing is a good way to get a better understanding. Start a small project just to learn how to use FM Starter. Go to the “[Your first project](#)” section if you want instructions.

Special windows

Things have been split up in FM Starter. The use of dedicated windows for specific tasks is one such simplification. For example, the developer has a dedicated window called Dashboard, where most of the settings for each module or for the application can be easily set. The user has another window, that of the main application. Keeping these two windows separate is an aid to understanding where you should keep or display things.

List of windows

- **Main window** (your application window, the start page)
- **Developer Dashboard** (all developer settings)
- **Toolbox** (a small window for creating and searching for multilingual text labels)
- **Help window** (display of help texts)
- **ToDo window** (a list of tasks for the developer).

FM Starter is nevertheless a single-window application, as the user generally only has a single window in front of them. Exceptions are perhaps a preview of a document and occasionally a map window. See also the chapter “[Windows](#)”. Only the developer benefits from extra windows, for example to search for and edit text labels while working in another window in layout mode.

This is how it works

When a user starts the application, only the main window is displayed. All functions are intended for execution in this window.

Everything is modular

When Todd Geist introduced the idea of FileMaker modules in 2013, the aim was to make development more flexible and portable. Unlike other programming languages, FileMaker does not allow the creation of code snippets. Everything in FileMaker is packaged in a file. There are no code snippets. The idea of FileMaker modules was a way out and allowed the creation of individual small building blocks that could be integrated into other projects with a few steps.

Modularity has proven to be an excellent way of working. That's why everything in FM Starter is clearly organized in “modules”. The successors to the idea of FileMaker modules are the FileMaker add-ons, which can now be created directly from FileMaker and have largely the same structure as the former FileMaker modules.

In FM Starter, several modules are combined in the start file. The modules have a simple structure. This is best understood in the script folder of a module. Each module has its own folder. This contains several subfolders:

- Module folder
 - ReadMe script
 - License information
 - Settings subfolder
 - Subfolder Test (optional)
 - Public subfolder
 - Subfolder Private (optional)

Modules and also the later add-ons separate function from parameterization. This allows the same module or add-on to be easily adapted to different solutions. Settings can be found in the “Settings” folder. Default settings can

be adjusted there. They are loaded from the main script, which is stored in the “Public” folder. You should only ever address scripts in the “Public” folder. The “Test” subfolder contains scripts for test sequences, if there are any. The “Private” subfolder contains scripts that never need to be adapted, but simply do their job.

Working with modules

These concepts are an important basis for FM Starter. Because they work so well, many functions have been set up as modules in FM Starter. All modules have the same structure. Understanding one module structure automatically means you will understand all the other modules, no matter who developed it. It should not be important to understand how a module works if you know where to set the parameterization and how to address the public script.

FM Starter uses these concepts and makes the settings available via the developer window (the dashboard) to further improve usability and speed up deployment.

Add-ons

Since version 19.1, FileMaker offers something new called add-ons. Add-ons are the next step up from modules. There are many similarities between add-ons and modules. Add-ons can be created directly in FileMaker and integrated very easily. FM Starter has modules built in, much like add-ons would today. Search the web for more resources, or visit fmstarter.com for more ideas.

Creating a great user experience

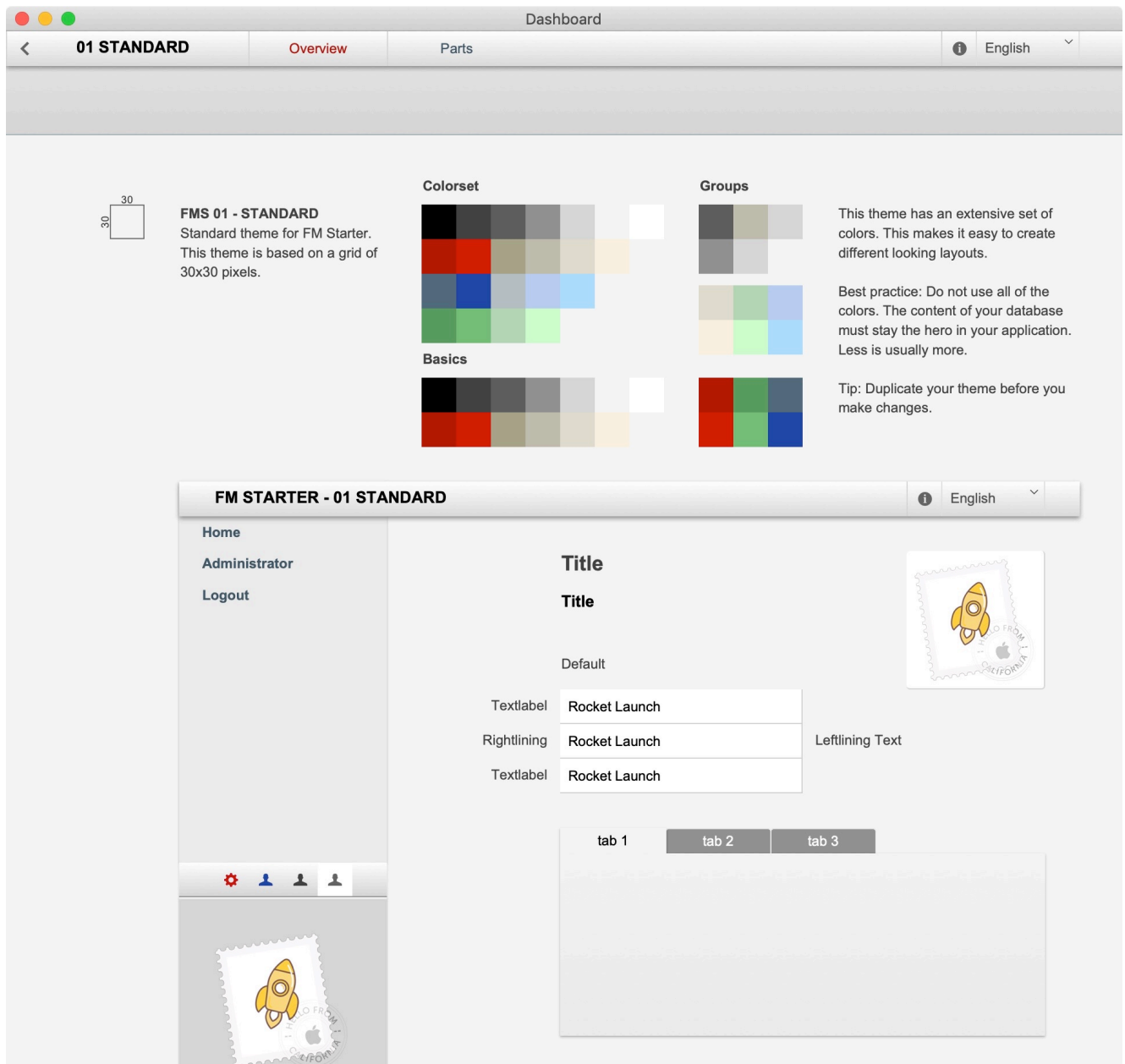
The user interface is probably the most important part of creating a solution. A consistent and well-designed user interface is one of the most underestimated challenges when developing a solution. For the user, however, the interface is at the center of the experience. How to navigate, how to find the most important parts of each screen, how to get feedback on actions performed - all these things create an impression. You want that experience and impression to be excellent? Then you need to think about the design.

FileMaker has good and flexible layout options that are now linked to styles and themes. The setting of any object can be configured and used as a “style sheet” There are styles for text objects, for buttons, for layout areas and anything else you can think of. A set of styles can be saved as a theme. Themes can be assigned to layouts, after which styles of this theme are applied to individual objects or layout parts. Using this on all pages supports the consistency of the look of the solution.

Starter theme included

FM Starter comes with a default theme (FMS01 Standard), with a set of options for each object type and with a set of basic colors. While the detailed settings of this theme can only be checked by clicking on an object and viewing its parameters in the inspector, there are two pages dedicated to this theme in the developer window (dashboard).

Further information can be found [here](#).



FM Designer

In addition to FM Starter, there is also FM Designer. FM Designer offers 10 themes as well as a development environment for your own themes. All themes are 100% compatible with FM Starter. For example, you can design a page with the standard theme in FM Starter and switch the entire design with just a few clicks after loading an additional theme from FM Designer.

Read more about FM Designer > [here](#).

No limitations

FM Starter has no limitations. It gives you all the basics and leaves out the rest. As it is a pure FileMaker document with no distractions, you can do with FM Starter whatever you can do with FileMaker. The advantages are obvious: shorter learning curve, immediate results, faster adaptation. Reduce to the maximum!

Basics

- Software registration (optional, with our FM Registrations product)
- Login and user administration
- Navigation
- Multilingual text labels
- Additional modules for frequently used tasks
- Standard theme for design and layout.

There is neither a flood of options to choose from nor a rigid framework to adhere to. The strength of FM Starter lies in its ease of use. This is a result of using only one structure and working with building blocks (such as modules).

Beyond the basics

Just like FM Starter consists of modules, you may want to add more modules to extend the functionality. You can download them from people who provide them, create them yourself or have someone else create them for you.

The same goes for the design. You can start with the basic theme provided in this solution, which works well. You can also duplicate the theme and start customizing the settings for each object to your own requirements.

FM Starter is a tool for developers. It is designed for you to go beyond all limits.

No limitations

FM Starter can do everything that FileMaker can do, because FM Starter is nothing more than a FileMaker file. There are no limitations. While FM Starter simplifies many things, it does not replace development. The use of FM Starter therefore requires that you develop yourself and familiarize yourself with the possibilities of FileMaker.

Your first project

Step by step: How do I set up a new project?

First things first

A few tips before you get started.

Create a backup copy

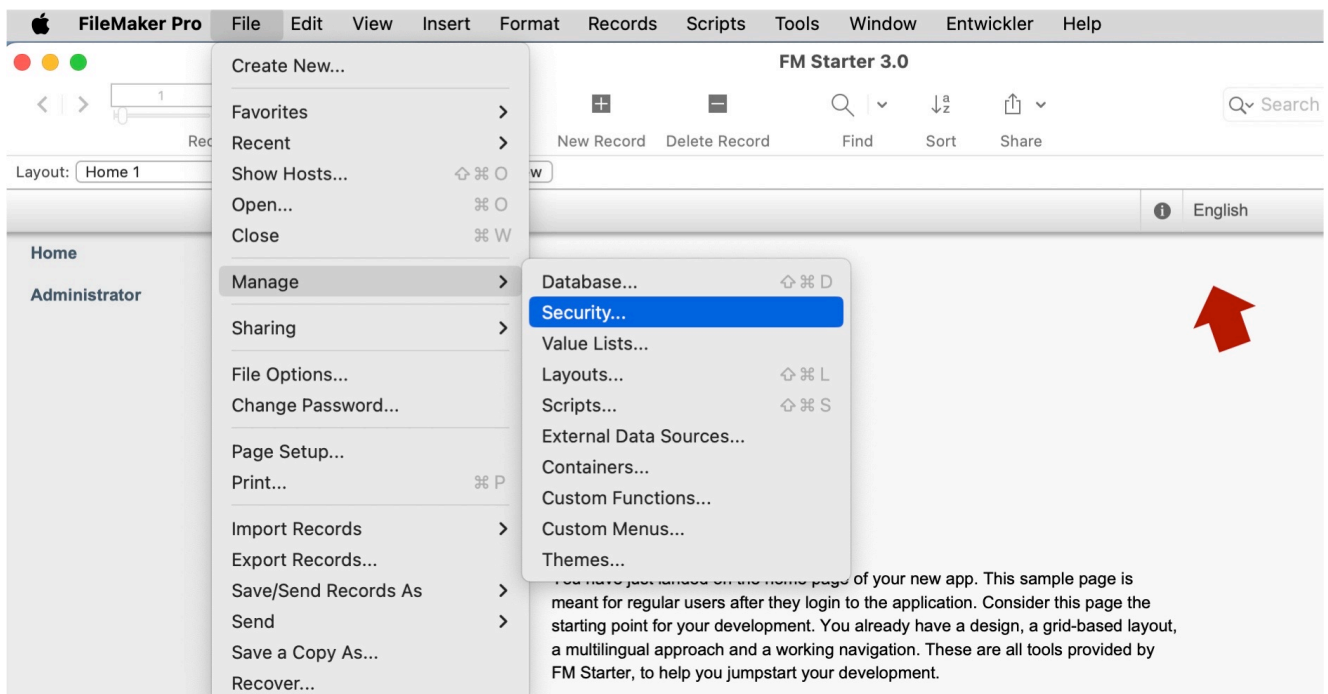
You have downloaded this product from your order. Now create a backup first. This will allow you a clean restart if you need it.

Creating a unique developer password

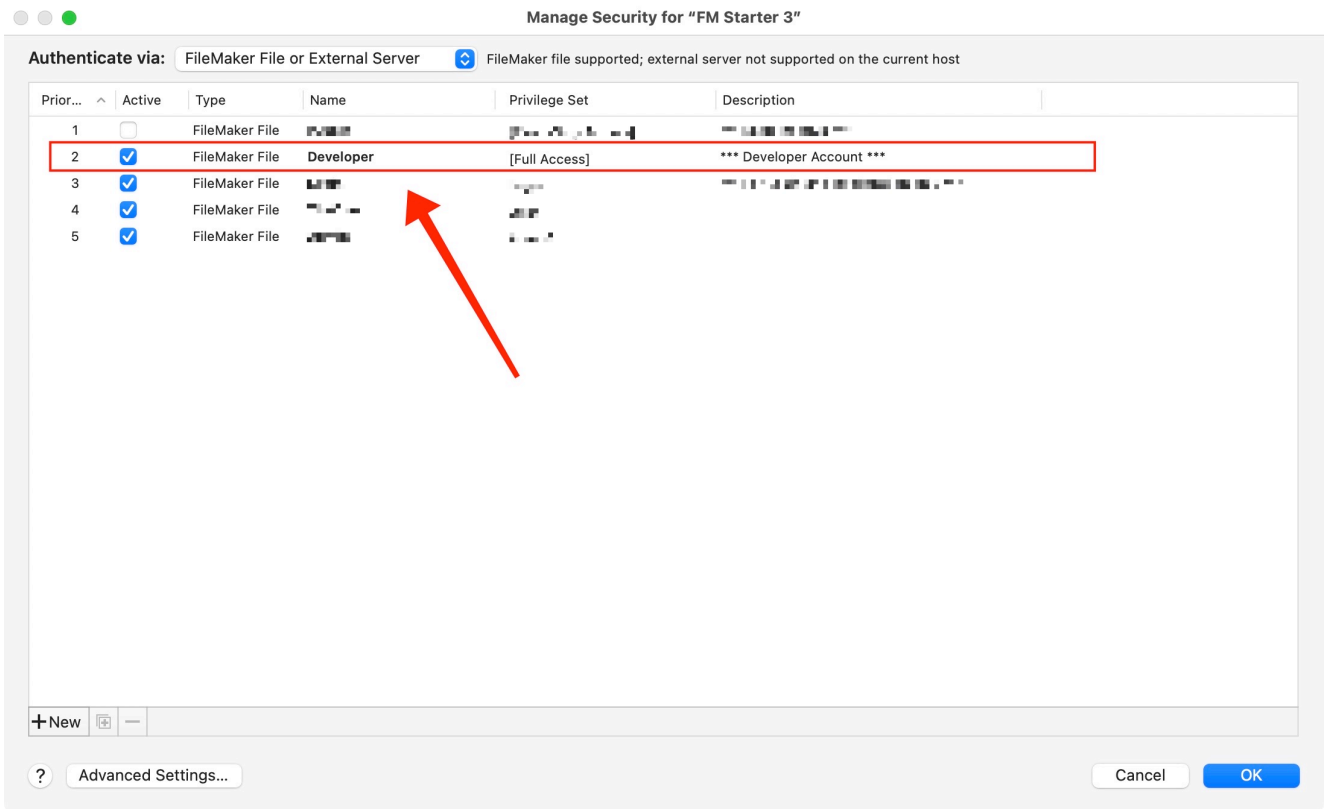
Your access name and password are part of the delivery. To make FM Starter your own, start by changing the developer password.

To change the developer password, go to:

File > Manage > Security



Search for the developer account:



Only change the password for this account. Leave the other accounts untouched at this point. You can change them later. Make sure that you make a note of the new developer name and password.

Here we go!

Now that you have made this solution your own, continue with the next steps.

Step by step

Simplify your life

Want to create your first project with FM Starter? Great! You'll find it's a simple process. FM Starter is a straightforward approach to getting your software up and running in no time!

Best of all, once you're familiar with FM Starter, you'll understand that you'll be able to easily take care of the development yourself. The foundation is already there.

Creating a custom solution

The first thing you want to do is to duplicate the FM start file for a new project.

- Duplicate the FM Starter file
- Rename the duplicated file.
- Place the duplicated file in a new project folder.
- Create a backup folder as a subfolder of the project folder

FM Starter has an integrated [backup module](#) that helps to take snapshots of the file during development (offline development only).

Login

Once you have set up a new project and a new file, you can start the file.

Login as a developer

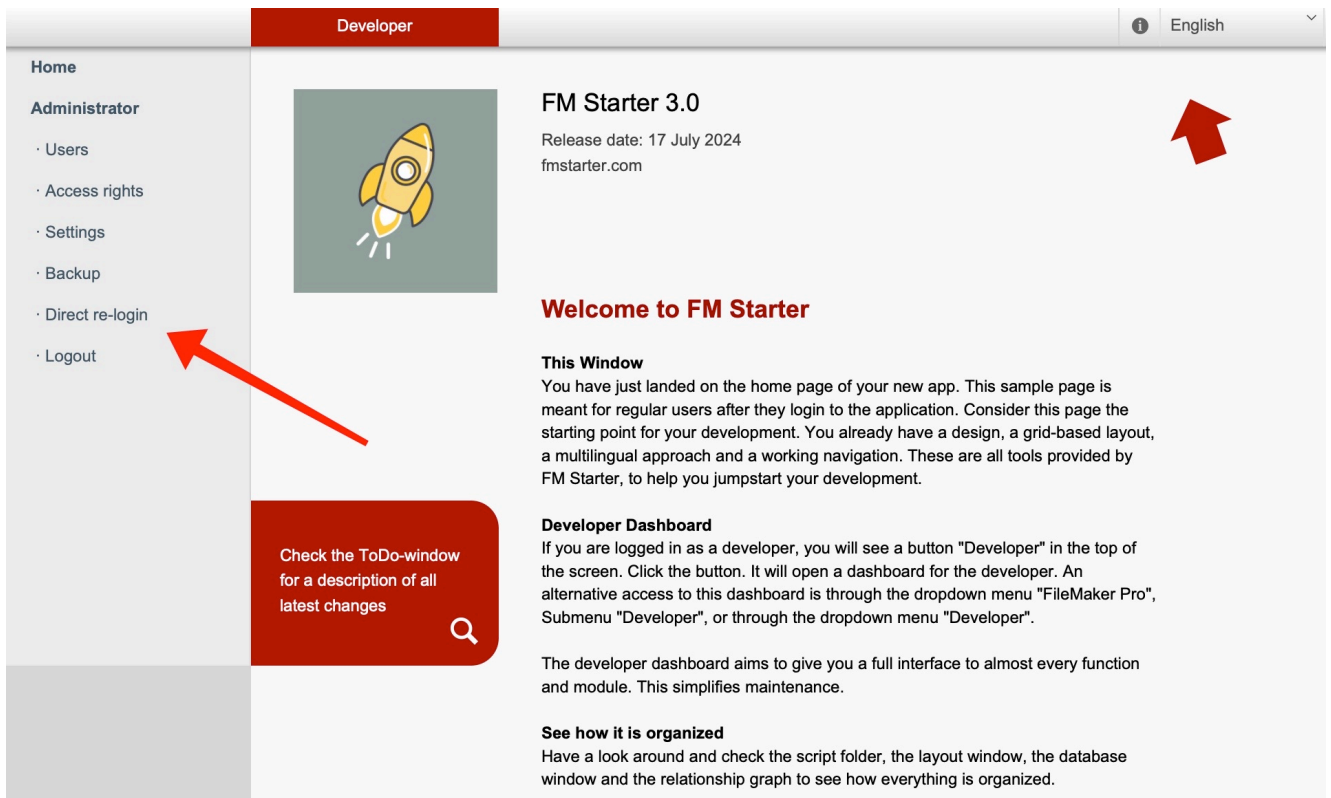
For security reasons, the default start of the file does not have access to a developer account with full access. To open the file directly as a developer, use one of these methods:

- Mac: Press ALT during startup.
- Windows: Press CTRL during the start process.

This will display the standard FileMaker login screen. Use the specified name and password. Check the security settings as soon as your project is in development.


Log in again

If you are logged in - even as a different user - you can log in again at any time with a different account, including a developer account. In the default setting of the FM Starter, this option is included in the navigation under Logout. Re-login simply opens the FileMaker login screen.



Logout

The logout option takes you back to the normal login screen while the current user is logged out.

English

Login

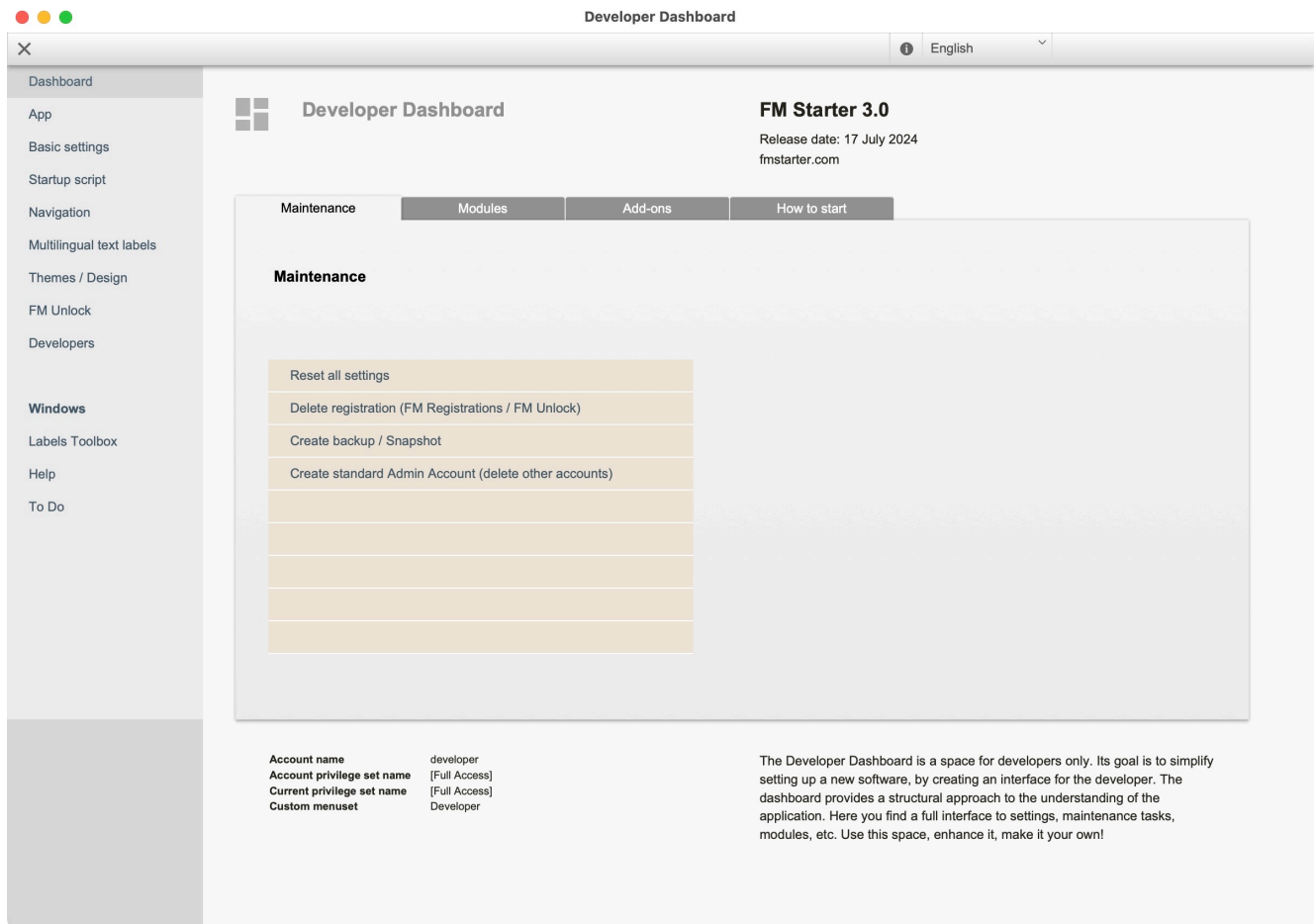
Find user | Password

Email

Find user

Developer dashboard

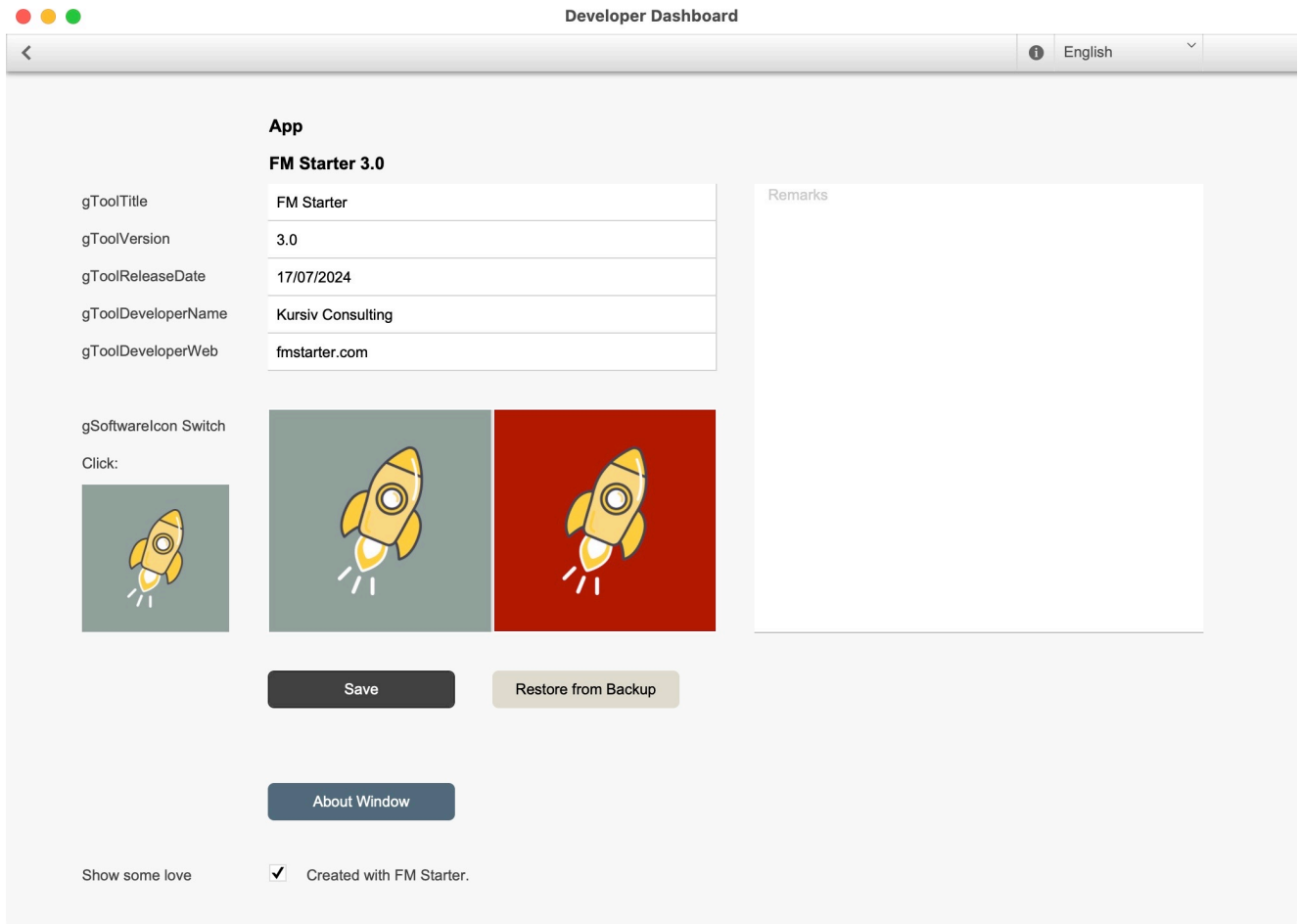
You are now logged in as a developer and have access to the developer dashboard. Click on the Developer button on the homepage or select Dashboard from the Developer drop-down menu.



FM Starter is ready to use when you buy it. Everything will work as it is already configured. You can add tables and layouts to quickly add your own workflows to FM Starter. However, to make it your own, take a quick look at the settings in the dashboard. Here you can configure your solution.

Application settings

The application settings contain the name of the software, the release date and some other information.



The screenshot shows a web application titled "Developer Dashboard" with a language dropdown set to "English". The main content area is titled "App" and "FM Starter 3.0". It contains a form with the following fields:

Field	Value
gToolTitle	FM Starter
gToolVersion	3.0
gToolReleaseDate	17/07/2024
gToolDeveloperName	Kursiv Consulting
gToolDeveloperWeb	fmstarter.com

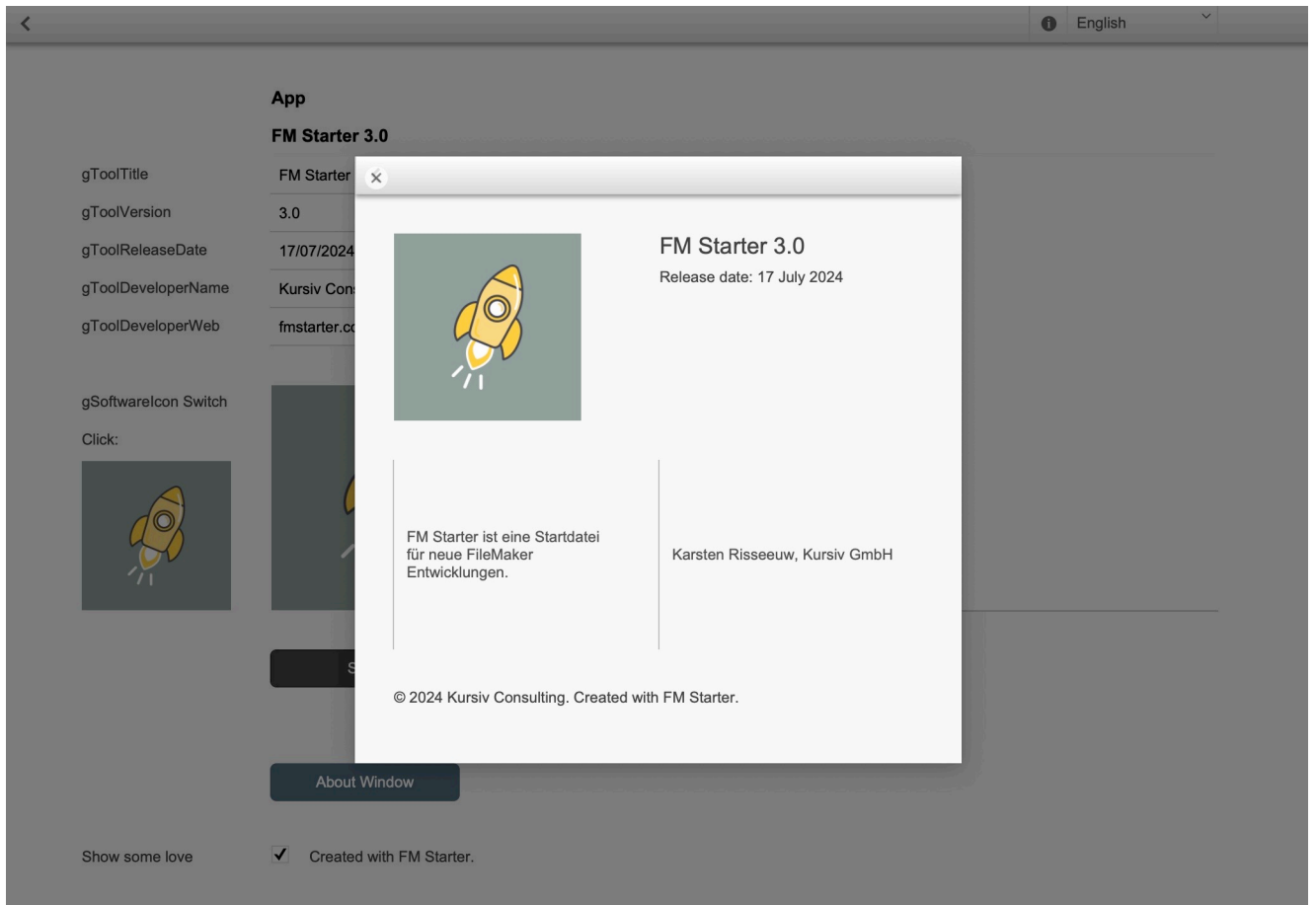
Below the form is a "gSoftwareIcon Switch" section with a "Click:" label and three icons: a small rocket on a grey background, a medium rocket on a grey background, and a medium rocket on a red background. To the right of the icons is a large "Remarks" text area.

At the bottom of the form are three buttons: "Save", "Restore from Backup", and "About Window".

At the bottom of the page are two checkboxes: "Show some love" and "Created with FM Starter." (which is checked).

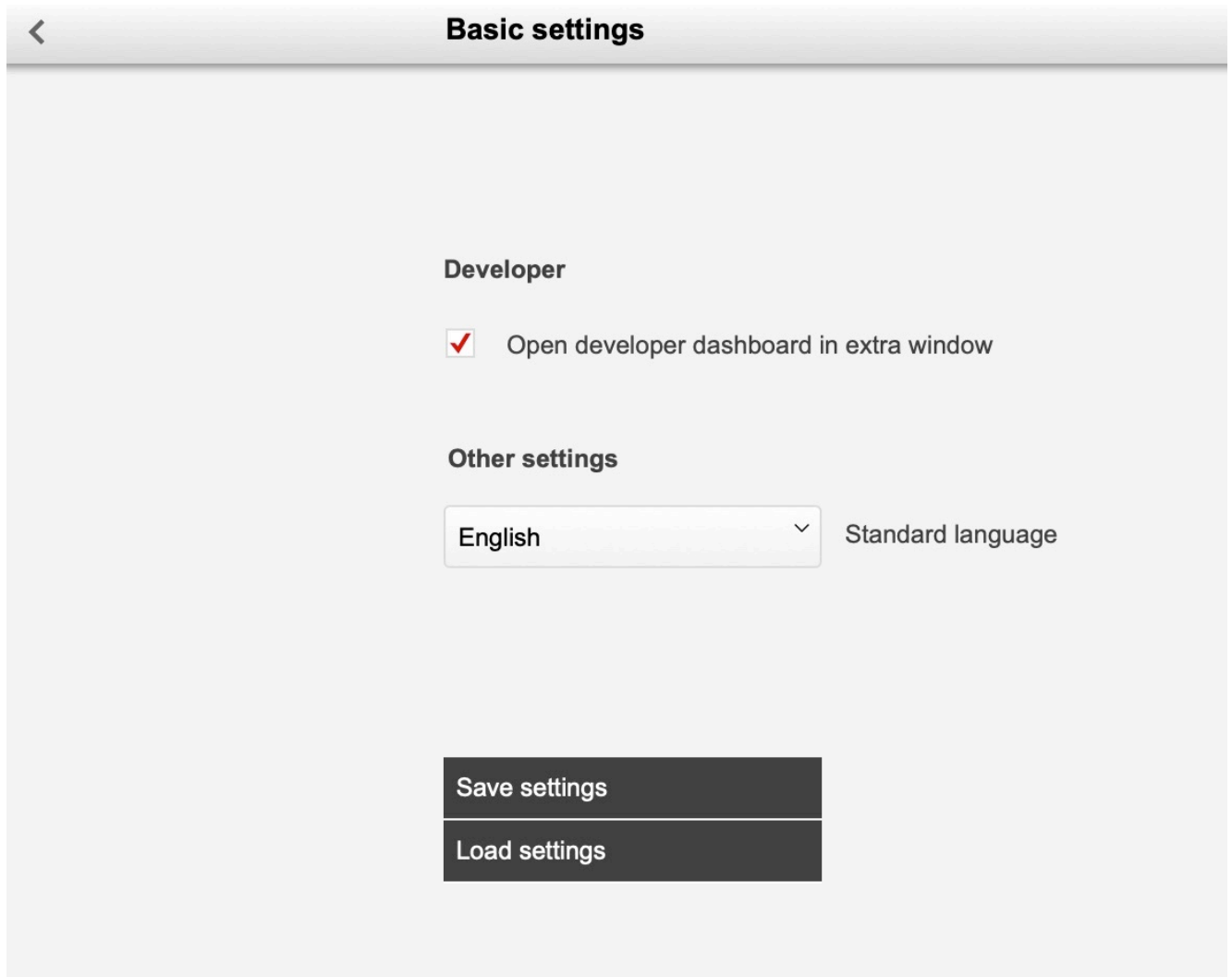
“About the software” window

To edit the About window, click the About button. This info window is part of the menu structure and can be accessed by end users via the FileMaker menu or the application name. Add your company and support details.



Start by exploring the settings

The basic settings are the place where some basics can be configured, such as the setting of the primary language to be loaded when the file is opened.



The screenshot shows a window titled "Basic settings" with a back arrow icon on the left. The window contains two sections: "Developer" and "Other settings". Under "Developer", there is a checked checkbox labeled "Open developer dashboard in extra window". Under "Other settings", there is a dropdown menu currently showing "English" with a downward arrow, and the text "Standard language" to its right. At the bottom of the window, there are two buttons: "Save settings" and "Load settings".

If you change a setting, save the settings. They are saved in a table with a single record and loaded when you start the file.

When you start to expand FM Starter with your own workflows and logic, you may want to make additional settings accessible via this layout. Add to this window as you wish.

Startup Script

The settings for the startup script store information such as the size of the main window (for the end user) and the layouts used throughout the startup script. The Developer Dashboard simplifies the settings and management for the developer. Keep this simplicity as you continue to develop the solution.

Dashboard

English

Startup script

Main window

x	0	Width	1200
y	0	Height	Get (WindowDesktopHeight)

Layouts used

Start screen	Start Screen	The first layout you see when starting the application
Registration	FmrStartup	Registration page (FMR License Registration must be active)
Login	UserLogin	Login page (UAC UserAccounts must be active)
Landing page	Home 1	Standard landing page (only if the module UAC User Accounts is not active)
Developer	Home 1	Destination layout, when logging in as a developer

Optional modules

☐ Activate FM Unlock
FM Unlock is the counterpart of FM Registrations. Where FM Registrations encrypts license information, FM Unlock decrypts this information. FM Registration allows you to create unlock codes for your own FileMaker solutions. If you own a license for FM Registrations, you can activate this option, add your secret keys in the script "FM Unlock Settings" and you are ready to go!

☒ UAC User Accounts. Manage user accounts and access to the application.

[Learn more about FM Registrations](#)

Save settings

Load settings

Layouts

Various standard layouts can be reassigned here. The values are adopted by various scripts. As a rule, the default settings do not need to be adjusted!

Everything works as it is set.

- **Start screen:** This is similar to the flash screen you see when starting many software products. FM Starter shows the name and logo of the application as well as an indication of the start process.

- **Registration page:** This page is only used if you activate the FM Unlock module. With the optional product FM Registrations you can create your own unlock codes for your standard software solutions created with FileMaker. FM Unlock is the counterpart to FM Registrations and is used to unlock codes created with FM Registrations. If you do not activate this FM Unlock module, the layout will not be used here (you do not need to deactivate it).
- **Login page:** The login screen is required if you activate the UAC UserAccounts module. This module manages multiple users and gives each one their own login data and settings. If you do not activate this module, the layout is not used here (you do not have to deactivate it).
- **Target page:** The target page is only used if the UAC UserAccounts module has been deactivated. As a rule, the UAC UserAccounts module manages the landing pages per user group. If you deactivate this module, you need a default landing page - which you specify here. Voilà!
- **Developer page:** If you log in with a developer account, you can define your own login page. It might be a good idea to assign as empty a page as possible from a table without important data records - to avoid conflicts with large amounts of data.

Optional modules

The optional modules include

- **FM Registrations**, another product from Kursiv that allows you to create customized unlock codes for your own FileMaker solutions. FM Registrations encodes information about a license, and FM Starter 2 has the decoding already integrated. This makes it very easy to use FM Starter with FM Registrations. If you have purchased a license for FM Registrations, simply check the box to enable the check in the FM Starter start script. Learn more about FM Registrations at fmstarter.com.

- **UAC User Accounts.** This is a standard module and part of FM Starter. This user administration allows you to register several users of your software, each with their own account. We recommend that you activate this function by default. If you wish, you can simply deactivate the checkbox and there will be a standard login procedure in the startup script instead.

Save the settings after making changes.

The settings are made in global fields and saved in a table with a single data record. At startup, the settings are loaded into global fields or global variables for the respective user.

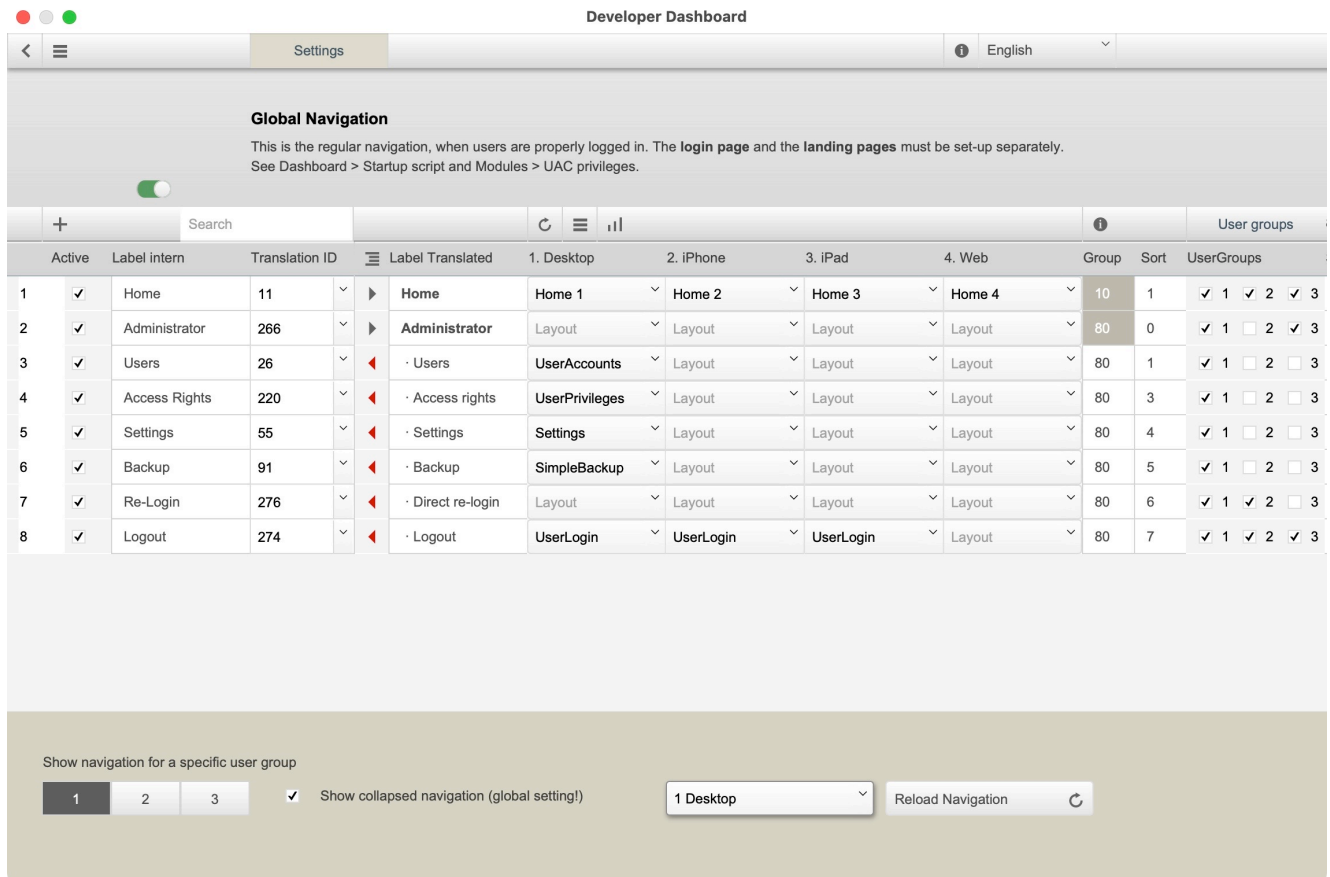
At the heart of the user experience

Once you start adding your own tables or additional layouts, you will want to reflect these in your navigation. Extending the navigation with new layouts is the easiest way to create a sense of coherence and the quickest way to create an integrated interface.

Important: Do not use any of the existing tables, layouts or scripts. **Always work with your own tables, layouts and scripts and do not touch any scripts or solutions from FM Starter at the beginning**, except via the setting layouts from the developer dashboard.

How it works

Let us give you an example where you add a table with the name **Addresses** to your database. You then create two different layouts, one for **editing** the address and another layout as a **list view**. Then you go to the developer dashboard and global navigation:



The screenshot shows the 'Developer Dashboard' with a 'Settings' tab selected. Below the 'Global Navigation' header, there is a table with 13 columns: Active, Label intern, Translation ID, Label Translated, 1. Desktop, 2. iPhone, 3. iPad, 4. Web, Group, Sort, and UserGroups. The table contains 8 rows of data. At the bottom, there is a section for 'Show navigation for a specific user group' with a dropdown menu set to '1 Desktop' and a 'Reload Navigation' button.

Active	Label intern	Translation ID	Label Translated	1. Desktop	2. iPhone	3. iPad	4. Web	Group	Sort	UserGroups	
1	✓	Home	11	▶ Home	Home 1	Home 2	Home 3	Home 4	10	1	✓ 1 ✓ 2 ✓ 3
2	✓	Administrator	266	▶ Administrator	Layout	Layout	Layout	Layout	80	0	✓ 1 2 ✓ 3
3	✓	Users	26	◀ · Users	UserAccounts	Layout	Layout	Layout	80	1	✓ 1 2 3
4	✓	Access Rights	220	◀ · Access rights	UserPrivileges	Layout	Layout	Layout	80	3	✓ 1 2 3
5	✓	Settings	55	◀ · Settings	Settings	Layout	Layout	Layout	80	4	✓ 1 2 3
6	✓	Backup	91	◀ · Backup	SimpleBackup	Layout	Layout	Layout	80	5	✓ 1 2 3
7	✓	Re-Login	276	◀ · Direct re-login	Layout	Layout	Layout	Layout	80	6	✓ 1 ✓ 2 3
8	✓	Logout	274	◀ · Logout	UserLogin	UserLogin	UserLogin	Layout	80	7	✓ 1 ✓ 2 ✓ 3

Navigation is set up using a navigation table. The navigation is loaded from this table. Each row is a data record. Each data record is an entry in the navigation.

Add a new data record (+). Add a label field, a translation ID, define the indentation and select the desired layout from the popup. If your layout does

not appear because you have just created it, you can refresh the list of layouts at the touch of a button.

Global Navigation
This is the regular navigation, when users are properly logged in. The **login page** and the **landing pages** must be set-up separately. See Dashboard > Startup script and Modules > UAC privileges.

LABEL LAYOUTS ⓘ

+ ☒ Search

	Active	Label intern	Translation ID		Label Translated	1. desktop	2. iPhone	3. iPad	4. Web		
1	✓	Home	11	▼	Home	Home 1	Home 2	Home 3	Home 4	10	1
2	✓	Administrator	266	▼	Administrator	Layout	Layout	Layout	Layout	80	0
3	✓	Users	26	▼	· Users	UserAccounts	Layout	Layout	Layout	80	1
4	✓	Access Rights	220	▼	· Access rights	UserPrivileges	Layout	Layout	Layout	80	3
5	✓	Settings	55	▼	· Settings	Settings	Layout	Layout	Layout	80	4
6	✓	Backup	91	▼	· Backup	SimpleBackup	Layout	Layout	Layout	80	5
7	✓	Re-Login	276	▼	· Direct re-login	Layout	Layout	Layout	Layout	80	6
8	✓	Logout	274	▼	· Logout	UserLogin	UserLogin	UserLogin	Layout	80	7

Refresh value list for all layouts

All this is explained in detail under [GNV Global Navigation](#). Add a group number and give it a sort number. This defines where the entry is displayed in the navigation. Repeat this for each layout you want to add.

After you have added the layouts, you must reload the navigation:

Developer Dashboard

Settings

English

Global Navigation

This is the regular navigation, when users are properly logged in. The **login page** and the **landing pages** must be set-up separately. See Dashboard > Startup script and Modules > UAC privileges.

Search												User groups	
Active	Label intern	Translation ID	Label Translated	1. Desktop	2. iPhone	3. iPad	4. Web	Group	Sort	UserGroups			
1	✓	Home	11	▶ Home	Home 1	Home 2	Home 3	Home 4	10	1	✓ 1 ✓ 2 ✓ 3		
2	✓	Administrator	266	▶ Administrator	Layout	Layout	Layout	Layout	80	0	✓ 1 2 ✓ 3		
3	✓	Users	26	◀ · Users	UserAccounts	Layout	Layout	Layout	80	1	✓ 1 2 3		
4	✓	Access Rights	220	◀ · Access rights	UserPrivileges	Layout	Layout	Layout	80	3	✓ 1 2 3		
5	✓	Settings	55	◀ · Settings	Settings	Layout	Layout	Layout	80	4	✓ 1 2 3		
6	✓	Backup	91	◀ · Backup	SimpleBackup	Layout	Layout	Layout	80	5	✓ 1 2 3		
7	✓	Re-Login	276	◀ · Direct re-login	Layout	Layout	Layout	Layout	80	6	✓ 1 ✓ 2 3		
8	✓	Logout	274	◀ · Logout	UserLogin	UserLogin	UserLogin	Layout	80	7	✓ 1 ✓ 2 ✓ 3		

Show navigation for a specific user group

1

2

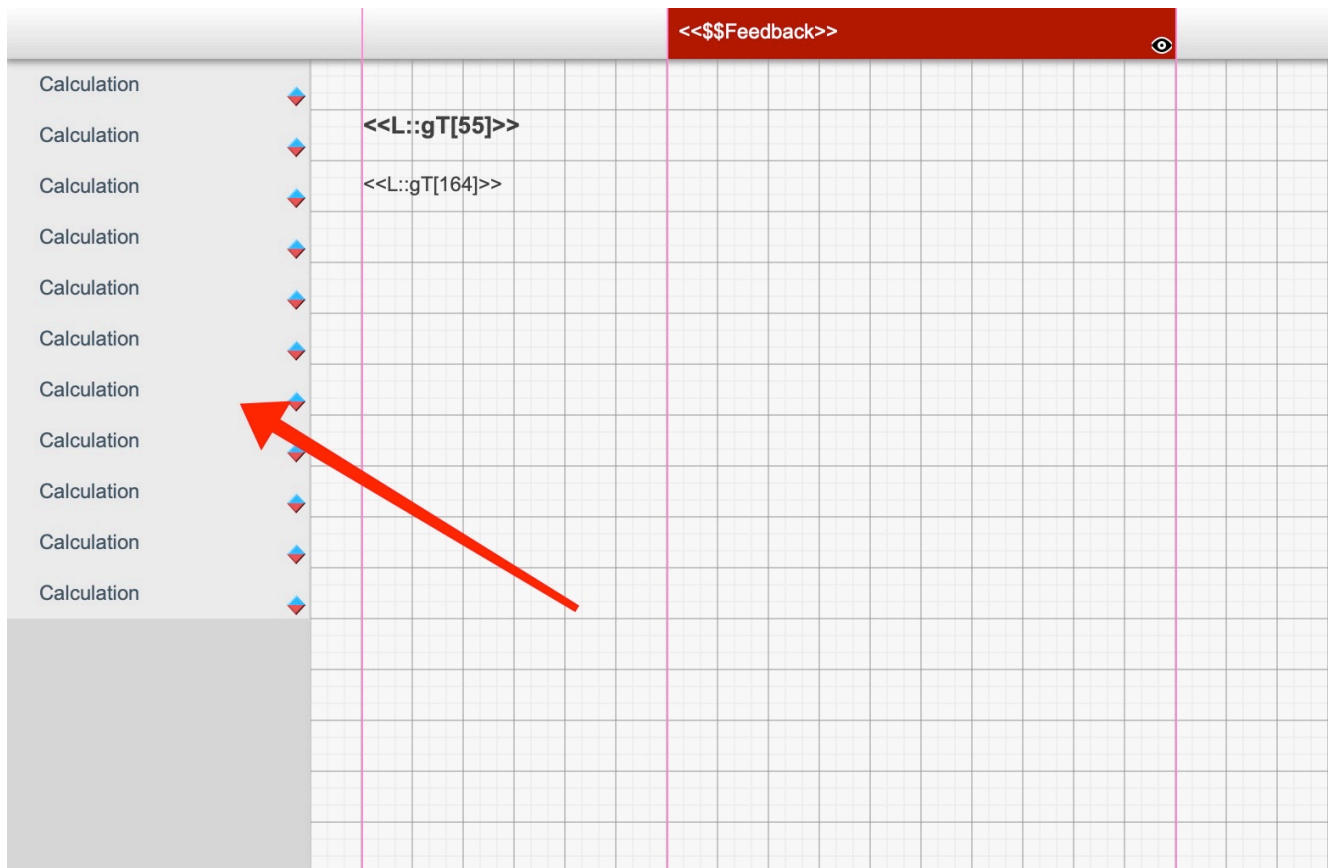
3

☒ Show collapsed navigation (global setting!)

1 Desktop

Reload Navigation

Adding the navigation to new layouts



That's it already! Everything is ready, the navigation works!

If you need a longer navigation, you can increase the number of segments in the button bar. New elements must then be set manually in the same way as the original elements. Particularly important: The script parameter for each button corresponds to the number of the sequence in this button list (1...2...3...4... etc.).

Rapid Application Development (RAD)

If you have followed the steps so far, you will understand that it is very easy to quickly create a working software with some sample pages. FileMaker is fast. With FM Starter it's even faster. This is called Rapid Application Development (RAD).

FM Starter is the perfect environment for rapid application development, where you present a client with a real and clickable workflow of a FileMaker application - not just a series of mockups, but a real FileMaker application that shows the look and feel of a future development in raw form. Just use what's in this starter file and you'll have your presentation and next project set up in no time.

Safety is an issue

FileMaker has some excellent security features that we support with FM Starter. We have implemented several good practices, but it is important that you are aware of the security issues yourself. Here are some suggestions for dealing with security.

Developer password

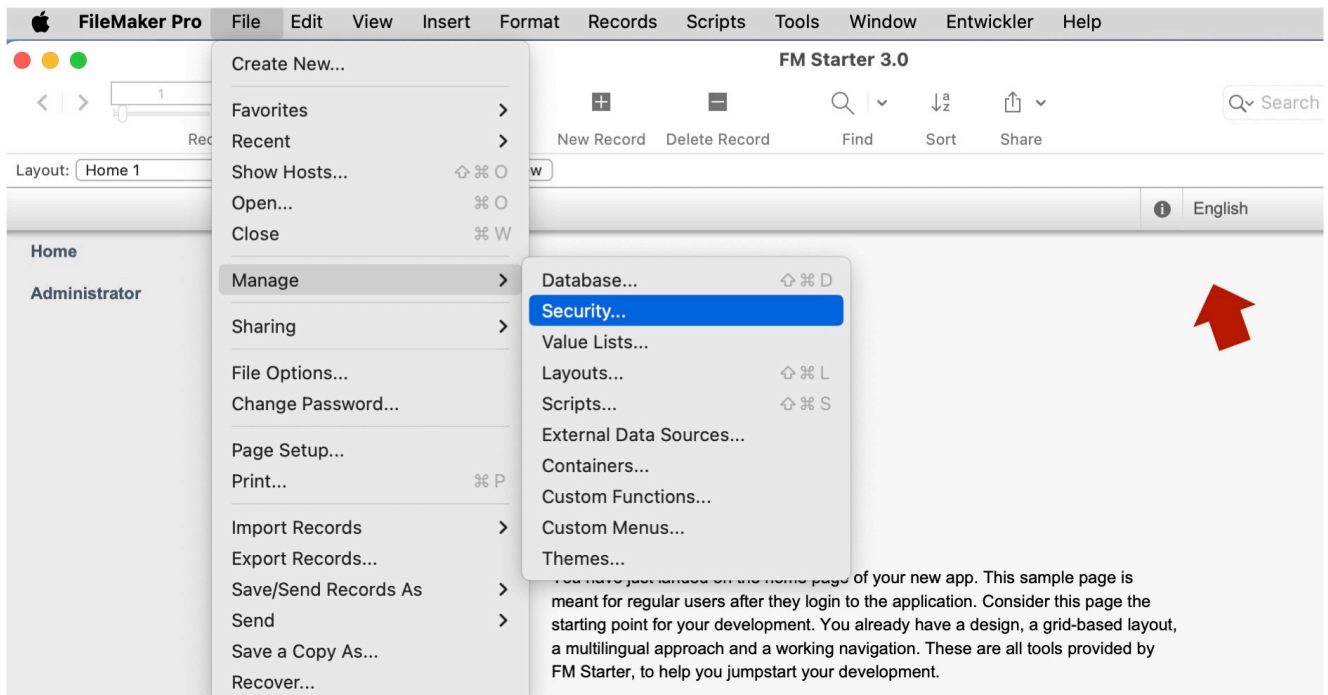
FM Starter is supplied with a default password. Change this when you start your next project. I recommend using a unique developer password for each project you create.

Change the password in the security settings of the file (see next section).

FileMaker security settings

FileMaker manages the security settings per user and creates a so-called user account, which gives access to the file with name and password. To manage the security settings, go to

> File > Manage > Security.



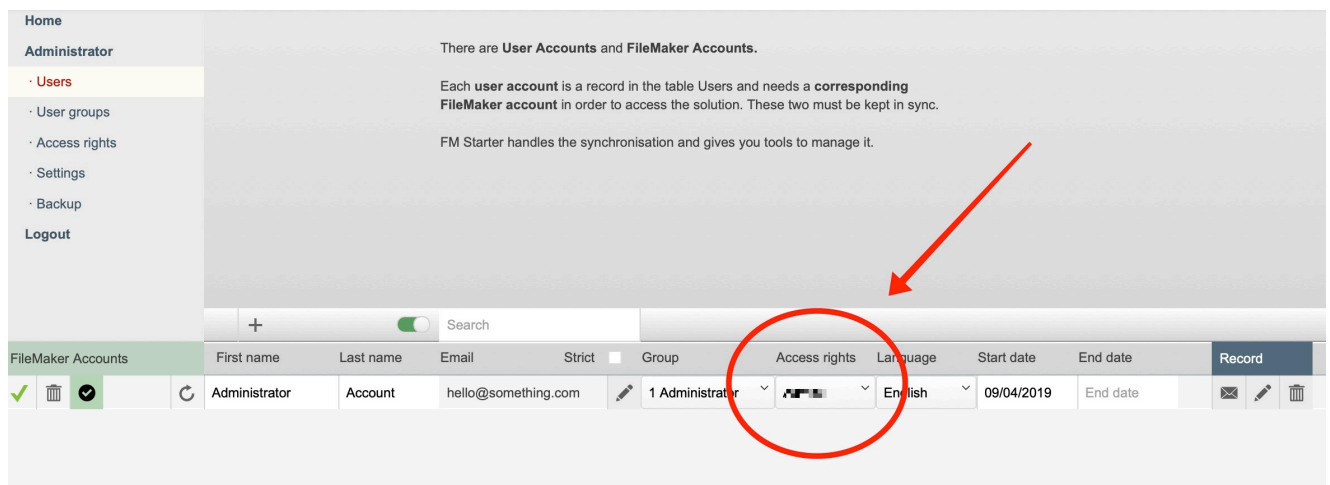
Two things belong together: Accounts and access rights. The latter are saved as sets of different settings. Access rights must be assigned to each account. There are several prepared settings and sets to choose from.

If you want to adjust the details, you must do this under the FileMaker security settings. You can also create new access rights there. New access rights must then be added with their name in the developer dashboard so that they can be used. See the next section.

Access permissions

Access rights are generated in the FileMaker security settings and listed in the FM Starter in a separate table as a list of names. This table is used to populate a value list. The value list is a simple way to assign any privilege set to any account in the user table.

User table

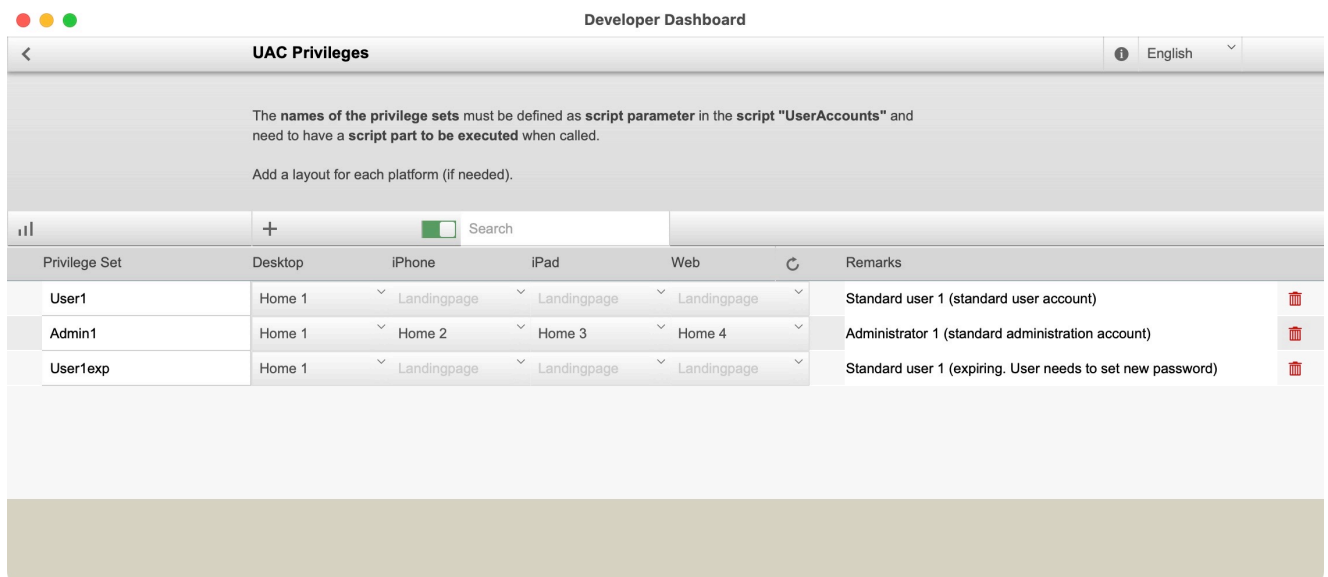
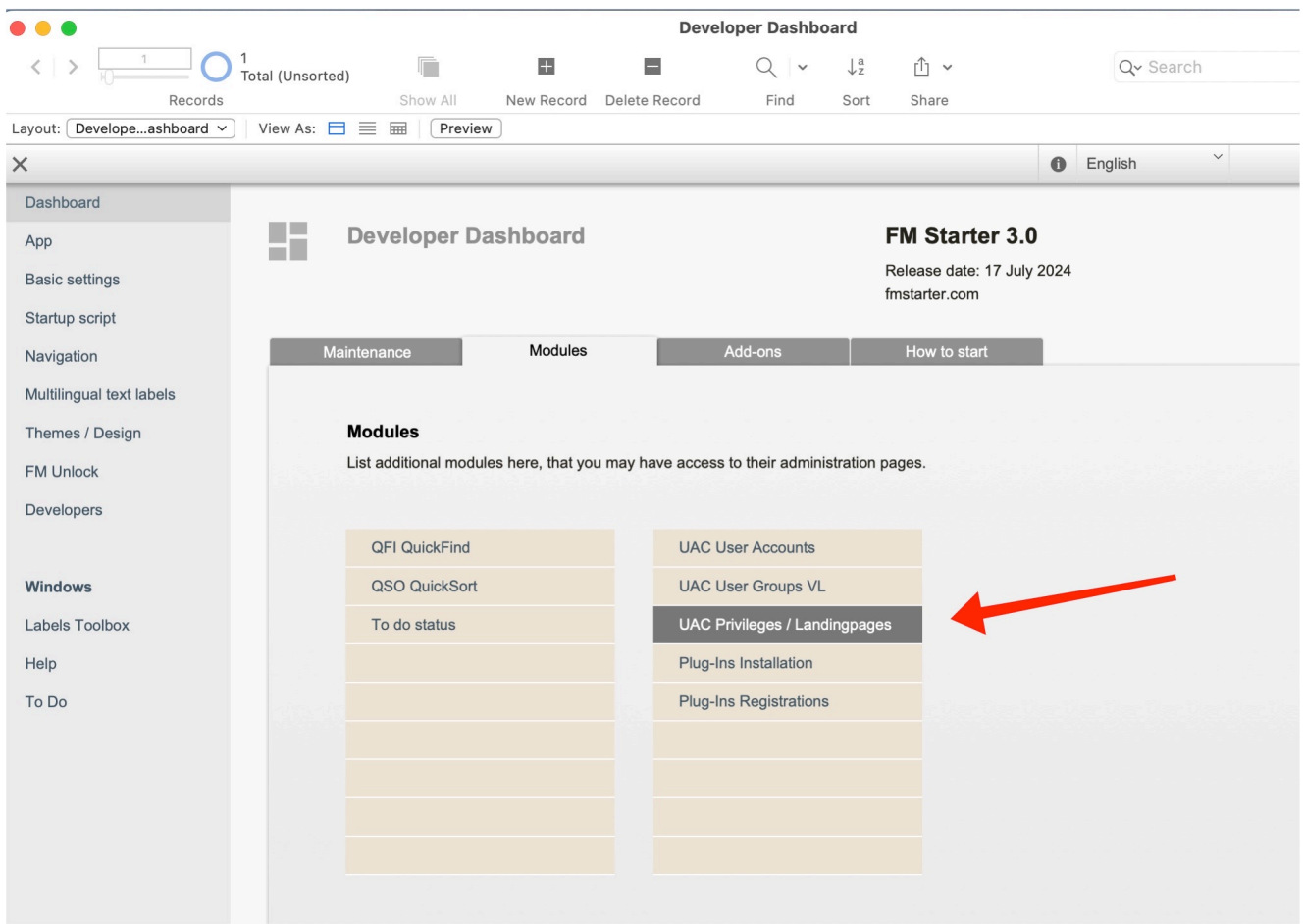


In the default setting of FM Starter, you have two views of this table. One view is for the administrator and only serves as a reference for the administrator, while the developer has an editable screen.

Administrator view

The administrator view provides the administrator with some basic information. The information cannot be edited.

Developer view (Developer Dashboard)



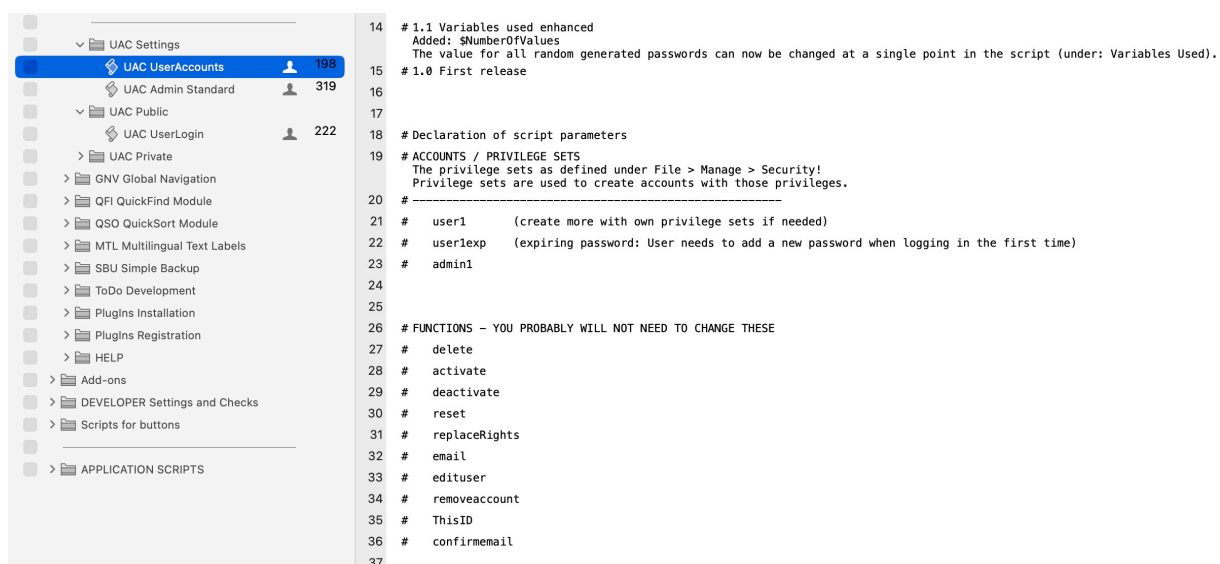
Access rights are therefore defined in the FileMaker security settings, but are listed in FM Starter. The sets can be edited or new ones created in the Developer Dashboard. It is only important that the exact name of the permission matches the settings in FileMaker Security for it to work.

Together with the correct names of the access permissions, target pages can also be listed for each set. For example, if you create a permission set for the sales department, they can be given their own landing page where they will land after logging in. The combination of login account and access rights can be used to set up a separate entry point for each defined group.

Security is a continuous learning and implementation process

There are many more safety issues that need to be addressed. This manual is not the place to discuss them. An absolutely secure solution is not available anywhere, but there are proven approaches that you can rely on. Without naming specific issues, here are some topics you need to address inside and outside of FileMaker.

- Backup, Backup, Backup!
- Create secure passwords and store them in a safe place.
- GDPR compliance does not allow you to store unencrypted passwords in your solution: Force users to recreate their passwords at first login and regularly from then on (FileMaker security settings), and let FileMaker manage the passwords. In the standard scope of delivery of FM Starter there is the access rights set with the name “User1exp”. This is a set which requires the user to set a new password when logging in for the first time. This password is not stored in the database, but is managed exclusively within FileMaker. You can find out how this works in the UAC UserAccounts script.



- Set access to layouts, scripts and functions carefully (FileMaker security settings).

- Do not provide layouts and/or scripts via drop-down menus.
- Handle custom menus with care (Tools > Custom menus).
- Do not try to overwrite any settings in the FM Starter or change anything in the startup scripts if you do not know what you are doing. The basics are fine!
- Control direct access to your files.
- FileMaker Security Settings > File Access can control the level of access for other FileMaker accounts and files.

Keep in mind that security issues will change and evolve, and your application will need to adapt to new trends and better security options implemented by FileMaker itself.

Now it's your turn

One of the advantages of FM Starter is the structure of the start file. Everything is grouped logically. Do the same when you get started.

- First think about the tables you need to create.
- Create a few but meaningful layouts for each table (for example: edit and list view).
- Then create the layouts by creating a grid for each page
- create a grid for each page,
- assigning the FM Starter theme and
- copying existing elements from FM Starter to the new page if necessary.
- Now create scripts for each layout so that it remains clear
- Group scripts and layouts into the tables for which they apply
- If more complex workflows are to be created, wait until the functions per layout and table are working properly. Only then create new layouts and scripts for more complex tasks.

This will help you to keep track of your own developments:

- Create small things that work.
- Build bigger things from smaller things.
- Comment on your script steps!
- If possible, make your script settings editable via the developer dashboard.
- Stay focused and keep things clean. Avoid spaghetti programming where everything is linked to everything else.

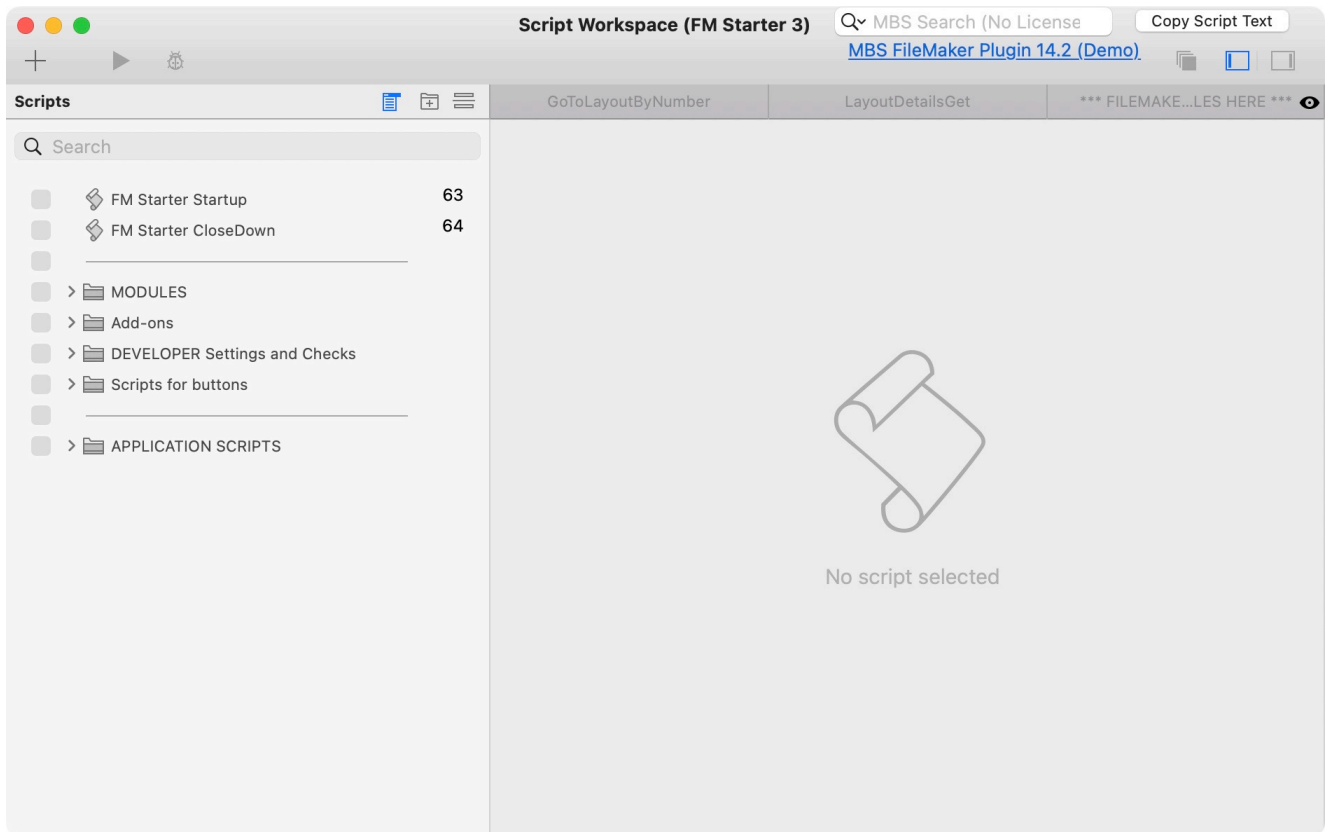
Adding tables

Think of tables as organizational aids. If you can group a lot of information (read: fields) for a specific purpose, they probably deserve their own table.

It is possible to keep custom functions lean and organized by assigning (a) separate table(s), scripts and layouts to each clearly definable task or subset of the database. Keep everything as simple as possible. Don't build complex workflows until the underlying concepts and tables are properly implemented.

Adding scripts

Add your own scripts to the start file and organize them according to the predefined structure.

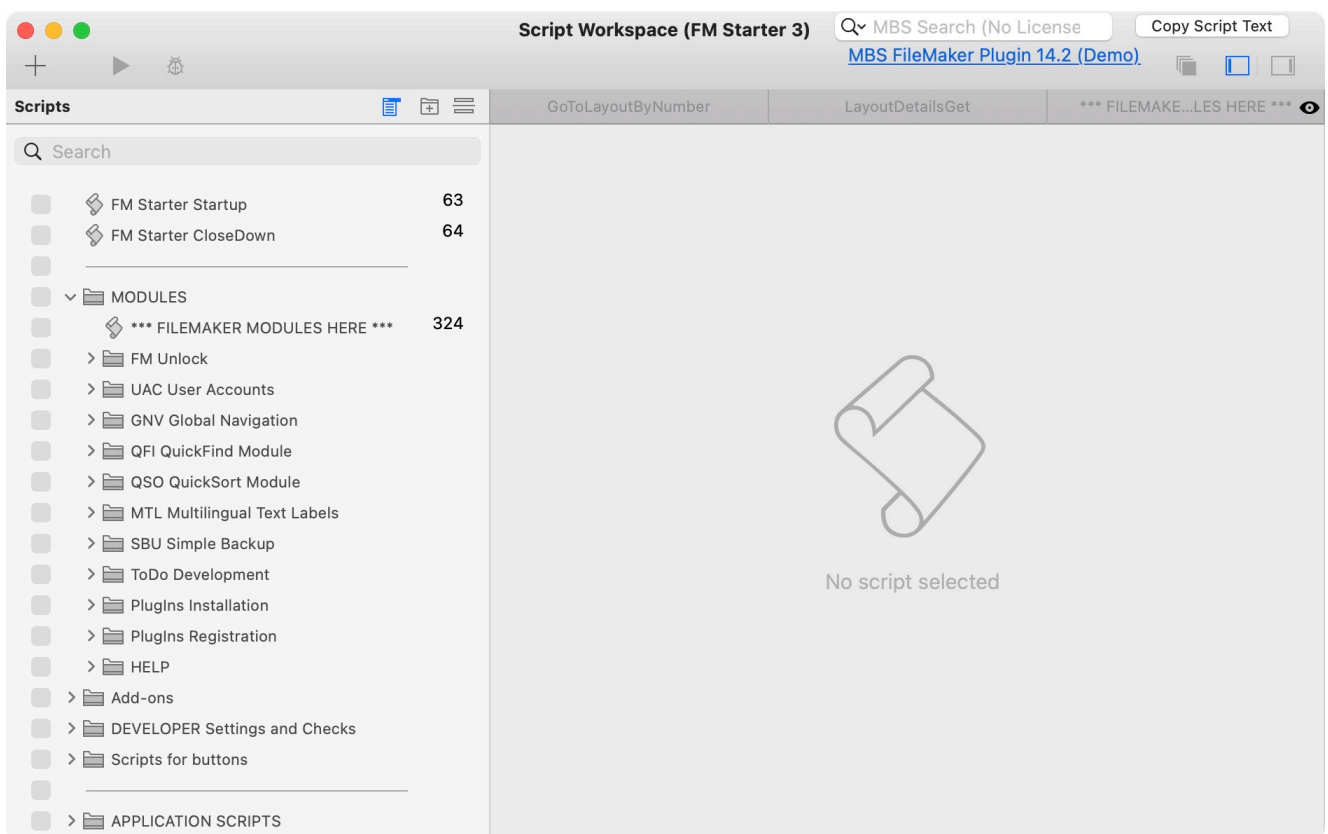


- Place your own new modules in the MODULE folder.
- Place your own add-ons in the Add-ons folder.
- Place all developer-specific functions in the DEVELOPER Settings and Checks folder.
- “Scripts for buttons” is the folder name under which FileMaker saves individual action steps when you turn them into scripts. It is recommended that you keep this name, as FileMaker automatically places the scripts created in this way in this folder.
- APPLICATION SCRIPTS is the folder that is intended for your own scripts.

Adding modules

FM Starter has a modular structure. Different modules solve different tasks. FileMaker modules are the brainchild of Todd Geist. Even though modules have now been replaced by add-ons, the basic structure is the same. The compact presentation, clear separation of settings and functionality and the outsourcing of settings in a regular layout have contributed significantly to the development of FM Starter.

Modules are building blocks for FileMaker programming. Each module solves a specific problem and this works extremely well. FM Starter comes with a number of modules that are already fully integrated:



The modular approach makes it very clear. Want to see how the navigation works? Take a look at the Global Navigation module. Want to expand the ToDo development module? It's easy - you'll find everything in one place.

The idea of FileMaker modules is still valid, but has technically been replaced by add-ons, which are much easier to integrate.

One of the big advantages of modules and add-ons is that you don't have to understand how another developer approaches the problem. Each module separates the settings and functionality. This applies to all modules and add-ons, regardless of the type of programming. It is the structure that counts. If you want to adjust the functionality of a module, you usually only have to adjust the settings scripts.



Since each module is equipped with its own tables, scripts and layouts, many modules can easily integrate their own layouts for the settings of this module. In FM Starter, the integration of modules is taken to the next level by making these settings layouts of modules accessible in the developer dashboard.

Modules or add-ons?

As of FileMaker Pro 19.1, FileMaker offers a modular approach with so-called “add-ons”. An add-on is very similar to a module and can also be understood as such. However, it is much easier to extend an existing solution with an add-on than it is with modules. Fewer steps are required and there are many more things you can do.

Which is better, modules or add-ons? Well, there's really no difference, except for the ease of use of add-ons. FM Starter's modular approach is modern. If you want to extend your solution with new “features”, you might prefer add-ons to modules, as add-ons are much easier to integrate. Ultimately, however, add-ons are not magic, and neither are modules. Both provide solid FileMaker functionality in nice and usable “pieces”. They are simply building blocks.

Structuring your file

Every great design starts with an even better story.

- Lorinda Mamo

When designing your application, think of a story you want to create. In this story, there are different chapters and several characters that play. These are parts of the structure. Something is happening on stage and you want to present that to your audience in the best possible way. Others may talk about workflow and stakeholders, but those are just different words for the same thing. Something should be happening, and what is it?

During programming, you may want to reflect the story: Group your functions, layouts and tables around specific purposes and tasks. Each of these topics is like a chapter, a subset of your workflow, a category of what you want to do. To keep these subsets organized, it will help you to structure your file.

If you already start with a workflow, you have a mindset and a process in mind from start to finish. If you've already done that, you may need to consider which parts of the workflow stand on their own. It can be something as simple as:

- Edit layout
- List layout for the view
- Print layout for printing

In this example, you need the same table and the same information, but in your story you can do three different things: Edit the information, display the information, and create an output for the information. In your scripts, you can

reflect this simple structure, which will help you manage your solution over time.

Maintaining a simple structure is extremely beneficial. However, overdoing a structure makes things too complicated. It's best to find out what works for you. If you're not sure what's best, you can rely on the structures FM Starter suggests as a starting point.

Create your own story - happy storytelling!

Application window

This is where the action takes place.

Why windows?

One of the basic ideas behind FM Starter is the division of the file into different parts:

1. A part for the end user of the software
2. A part for the developer of the software

FM Starter mirrors these parts by giving each of these groups its own window. This concept is easy to understand and will most likely be used intuitively by you.

Multiple windows

The **main window** is the window that you see when you start the file. This is the normal user view. When you open the file regularly, you “land” in the application, so to speak. However, the developer has several windows: the **developer dashboard** as the central developer window, but additional windows provide support for certain tasks where you may want to work with two or more windows side by side.

No multi-window approach

We do not propose a multi-window approach. This concept is outdated and comes from a time when each application was divided into countless files. Today it is different - unless you have special needs. The user in FM Starter has only one window, the main window. Additional windows are only for the developer and only for certain tasks where it is advantageous to have several windows side by side.

One approach would be, for example: The user works in a single window. If another window appears, it is not normally used for editing, but only for displaying information. The help window is such an additional window and is

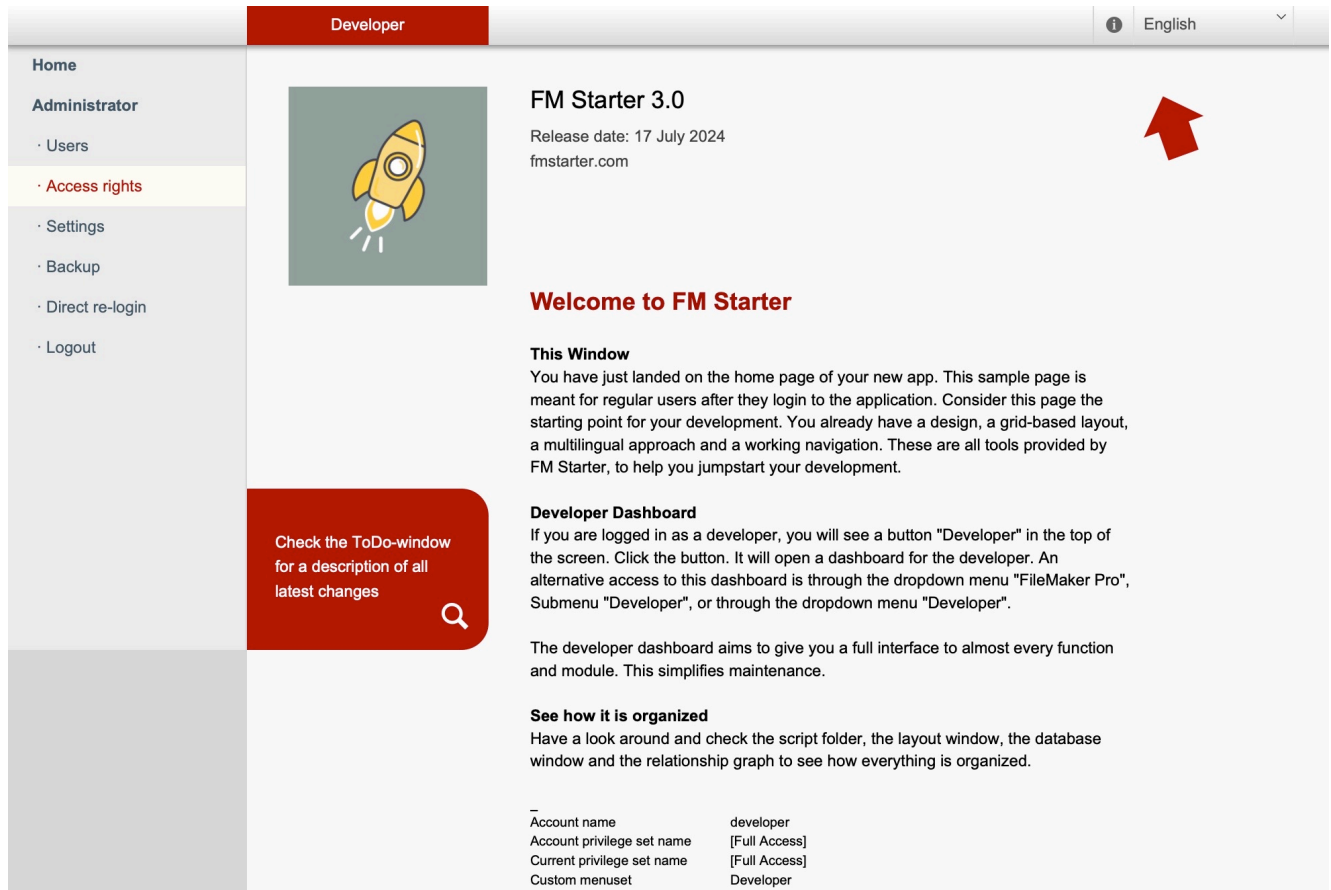
displayed as a card window. You can also create a print preview of a document in another window. A single-window approach does not mean that you cannot have different windows, but you will only edit data records in the main window (at most in a card window placed above it).

User windows

You design the user experience.

Main window

The main window is the window that you see when you are logged in as a regular user.



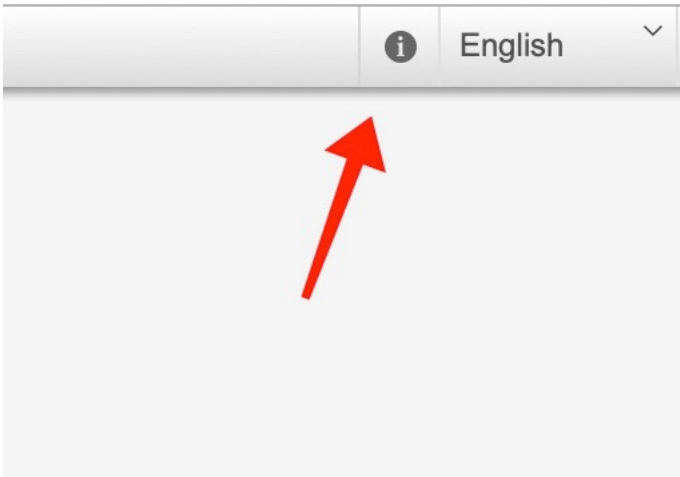
This window is your application window. If you create your own application, this is the window in which the global navigation works and in which you have embedded your user layouts.

Here you can see your own landing page. However, you can give each user group its own landing page. You do this in the developer dashboard. This allows you to give the administration a different starting position and menu structure than the sales department, for example. User groups can not only be given their own start page, but the navigation can also be specially tailored to user groups. See also the navigation table with settings.

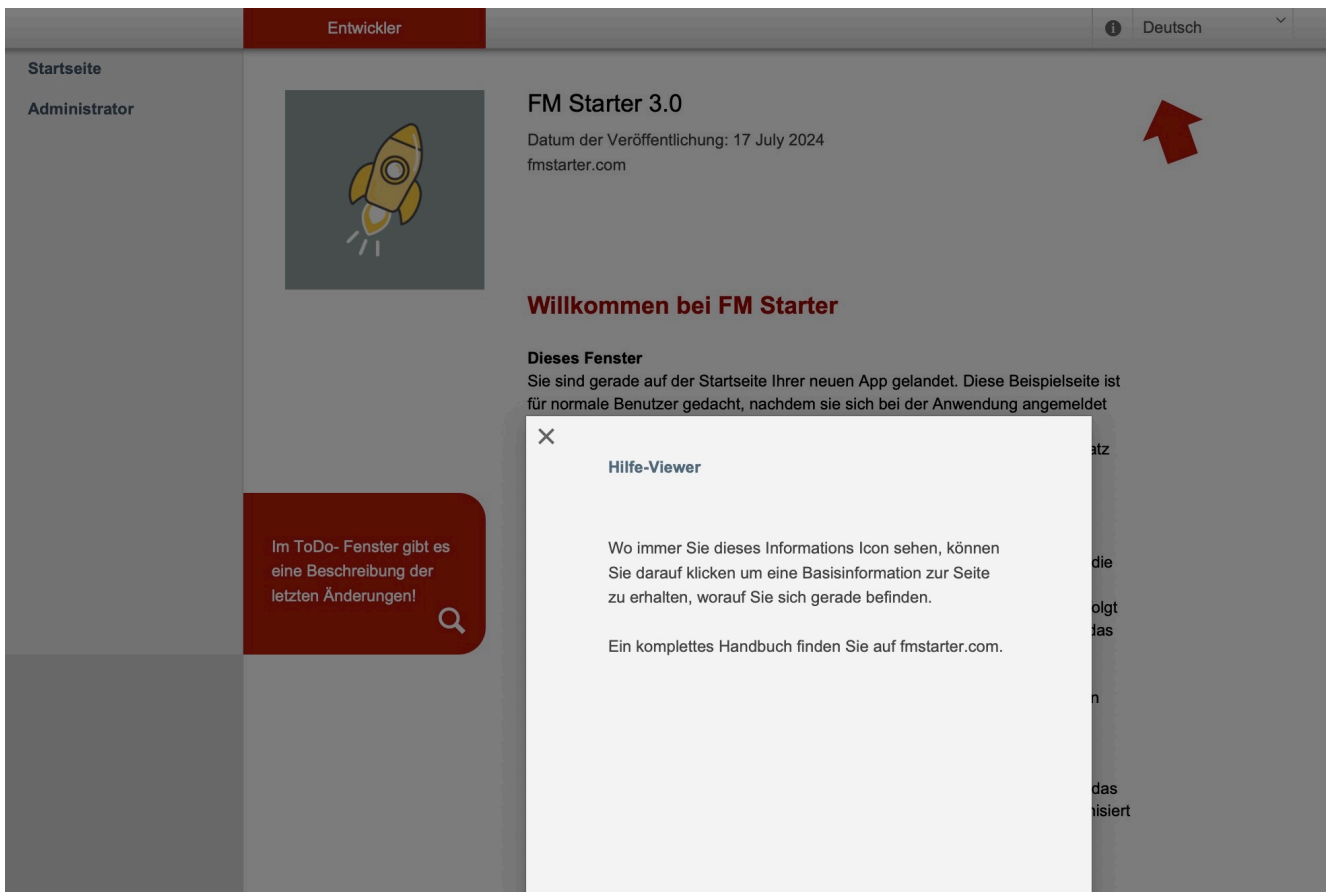
Help display

The help window is an exception to the rule that there is only one window. Only help texts are displayed in the help window.

Display the help window by clicking on one of these buttons:



This opens a help window with information:



This help is a new module in version 3 and has been greatly simplified compared to the previous module. Everything is now based on the already integrated multilingualism. The display is now not in a separate window, but in a card window above the page.

Help texts are now created in the toolbox and can therefore be automatically created in several languages. The HELP module is hidden behind a button. The button refers to the main script of the HELP module and is called with the number of the toolbox entry as a script parameter.

Further information can be found > [here](#).

Developer window

The developer has their own window, the **developer dashboard**. Everything that applies to settings specifically for the developer can be found here.

It makes sense for the developer to manage the parameterization of all program parts centrally in one place, for example. It can also be advantageous to manage program parts such as navigation or text modules in separate tables. Only the developer is responsible for managing this information. The developer therefore benefits from a separate working environment.

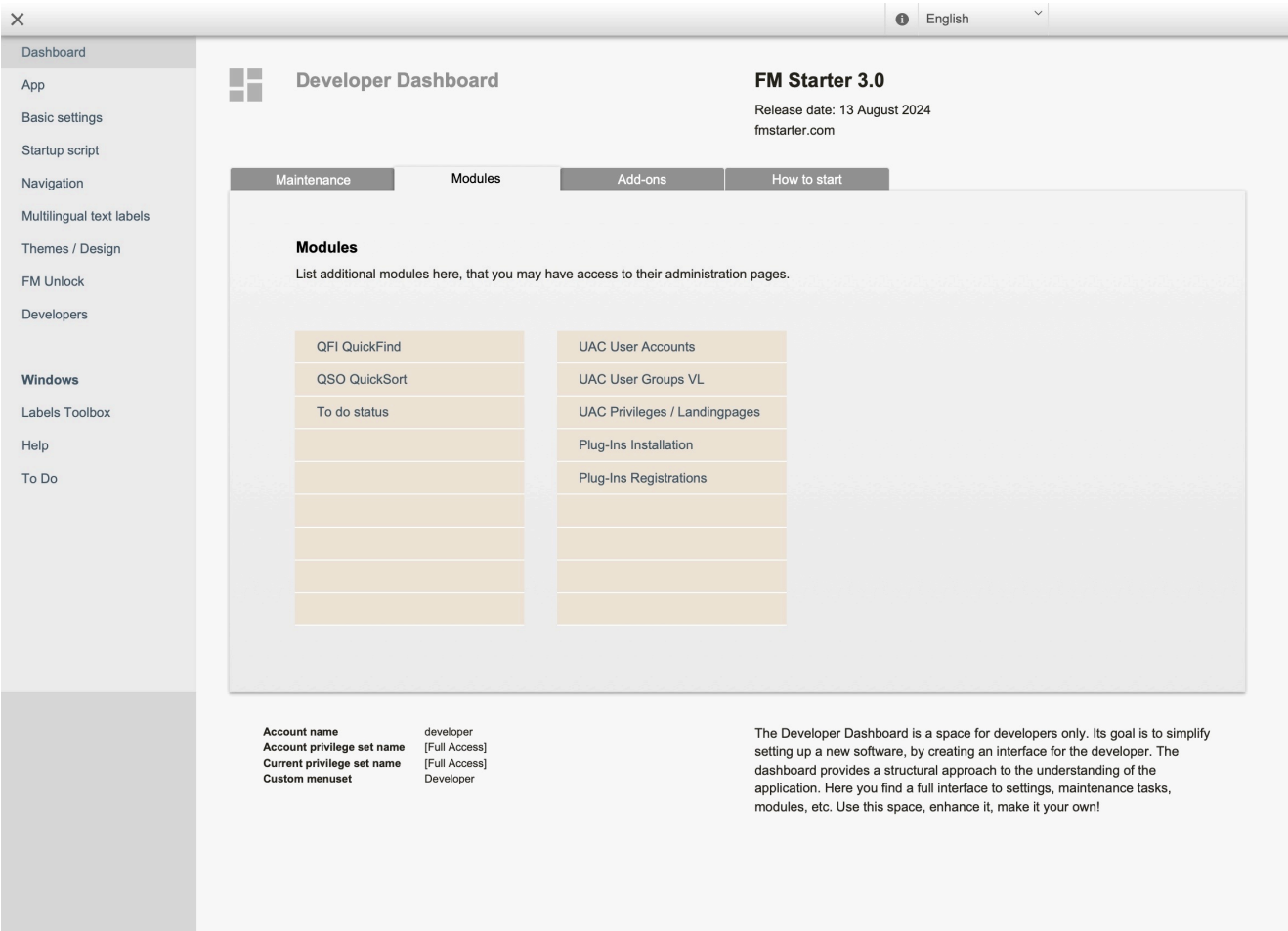
FM Starter offers such a working environment for developers in the form of a developer dashboard.

Developer dashboard

The developer dashboard is a central concept of FM Starter. In this window, the developer has access to most of the settings and configurations for modules and the application. We think you deserve some luxury at work.

Creating a developer dashboard sounds like a lot of work, but it saves a lot of time, especially if you regularly update or configure a solution.

The developer dashboard:



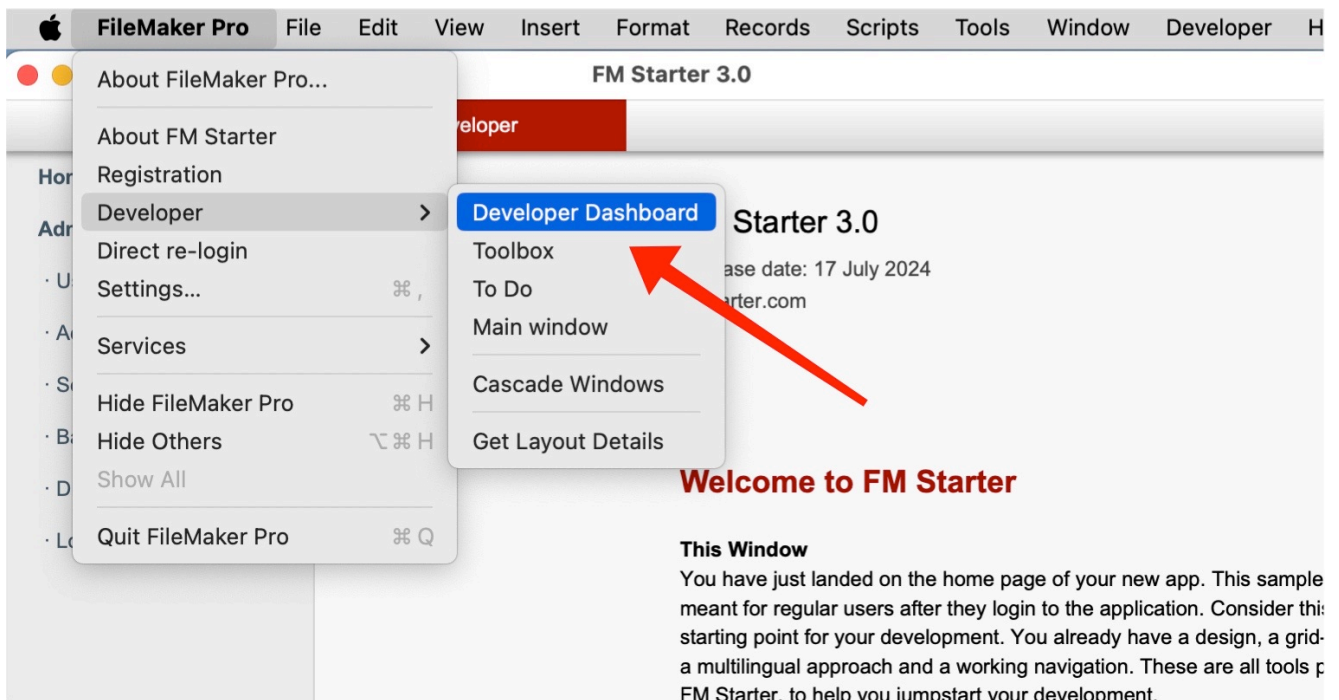
How to access the developer dashboard

There are 3 ways to access the developer dashboard. All these options are only visible if you are logged in with the developer account.

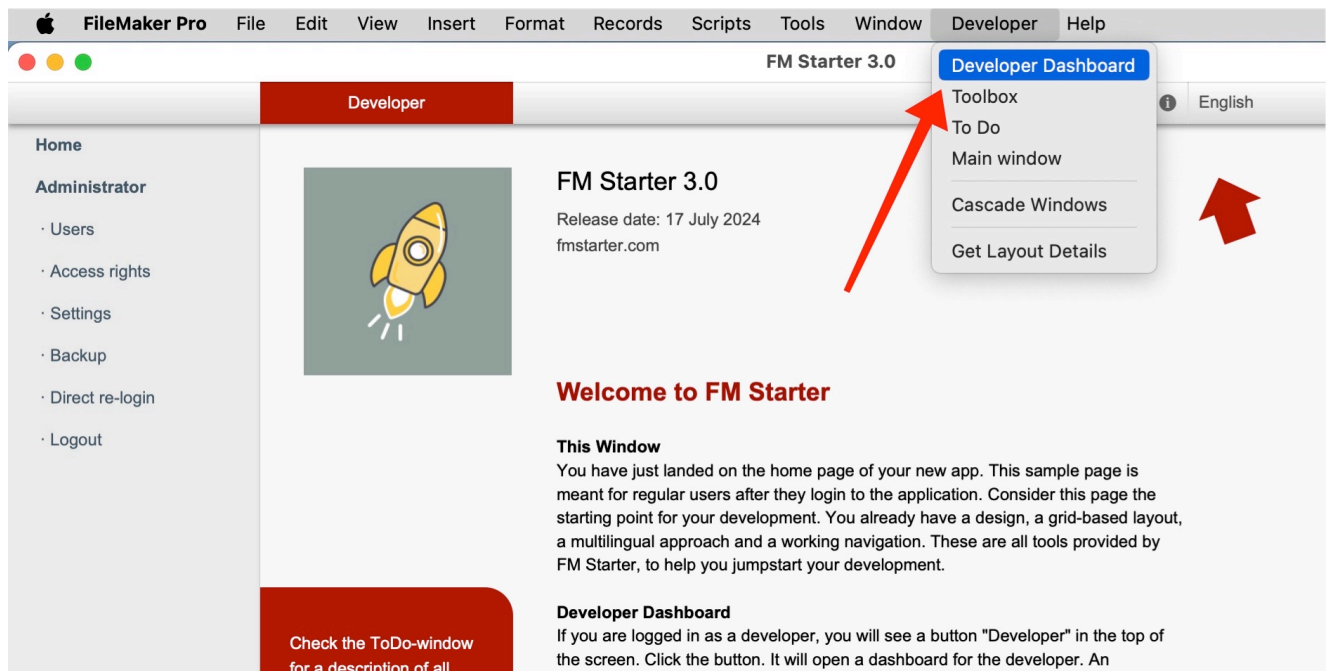
1. Button Developer in the home layout:



2. Drop-down menu FileMaker Pro Advanced > Developer:



3. Drop-down menu Developer:



If you need additional options, you can add them yourself.

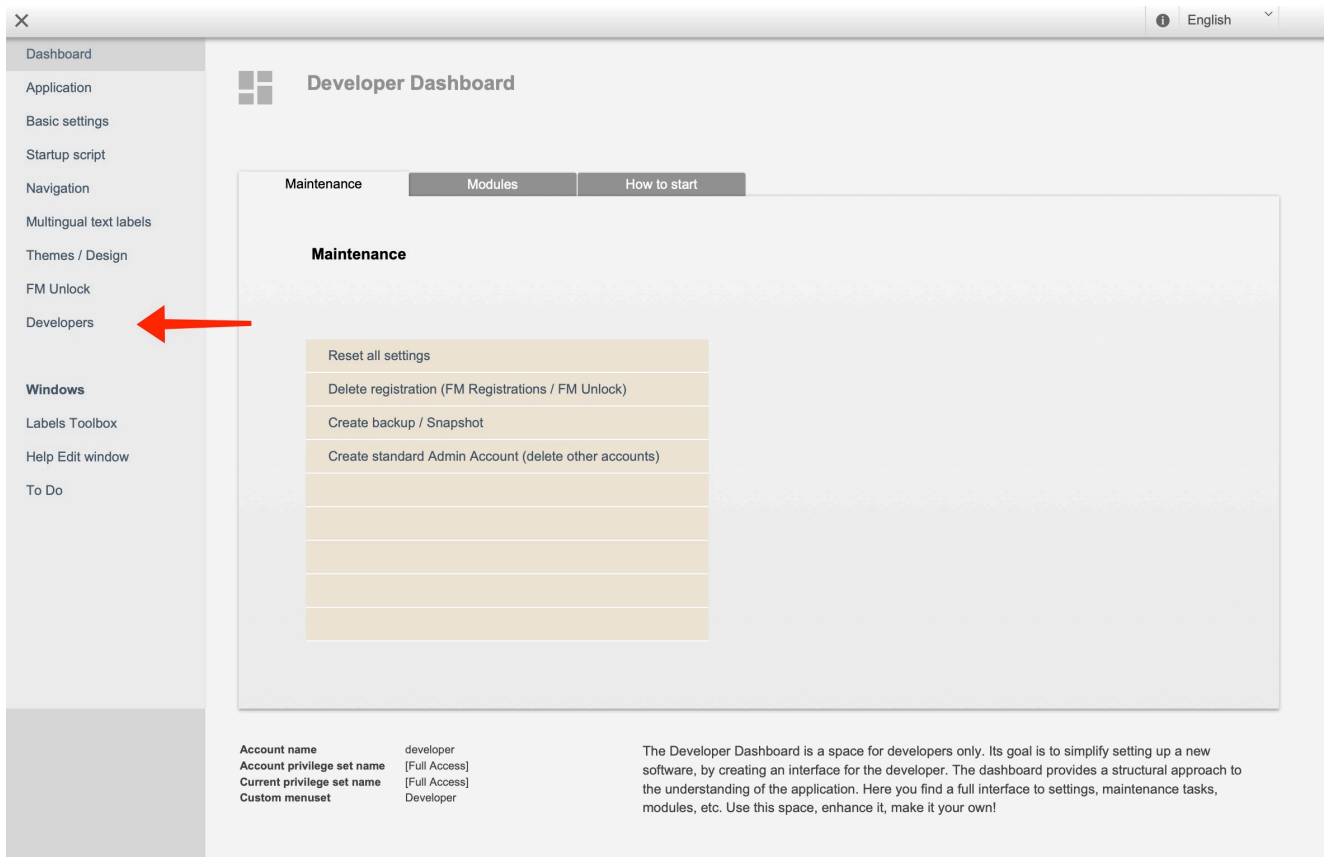
Many FileMaker projects are handled by a single developer. However, some projects require collaboration. In this case, the project is often located on a server to which all developers have access.

“Developers” entry in the developer dashboard

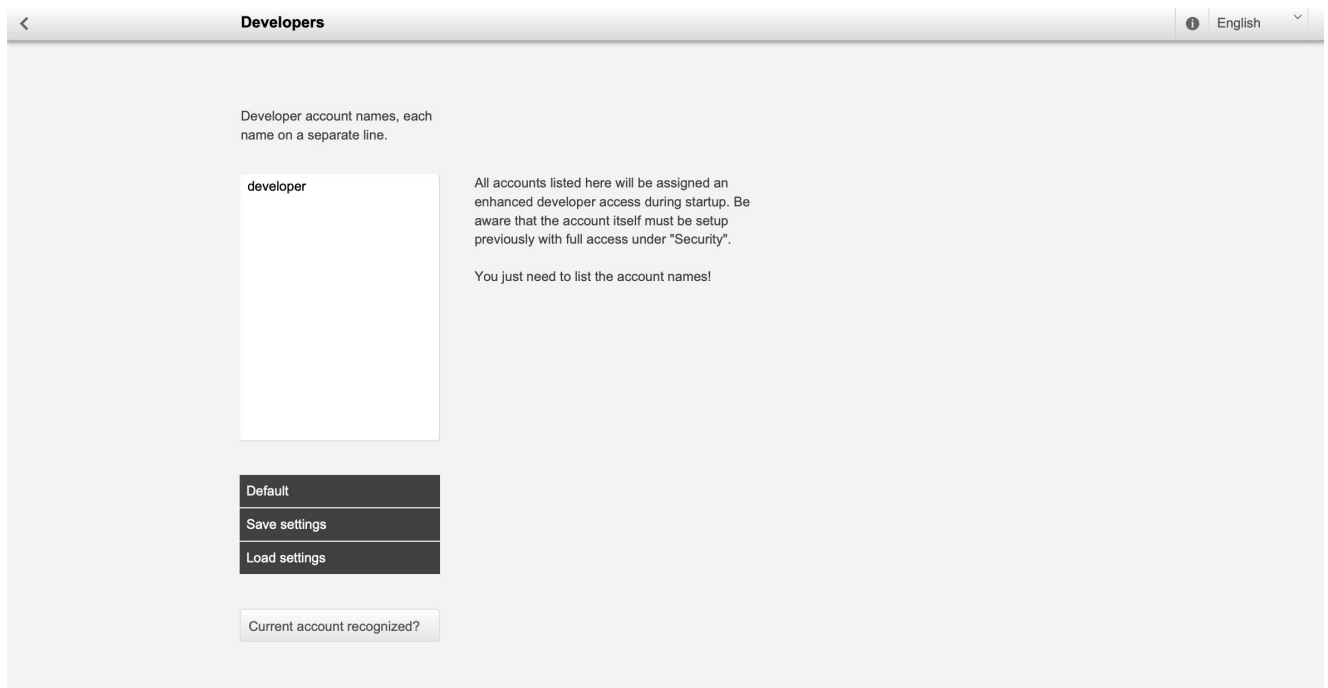
Up to and including version 2.5, there was only one developer account for FM Starter. Of course, you could add additional developer accounts at any time, but there were some special settings for the standard developer account that were set in the startup script. These included, for example, a menu set specifically for developers.

This has been extended in version 2.6. It is now possible to list several accounts as “developers”, which are all taken into account in the startup script and therefore receive additional functionalities.

Developer accounts do not run via the regular user administration. They must be created with full access rights directly in the FileMaker security settings. However, once they have been created, they can now be defined as a “developer account” in the developer dashboard. There is a separate entry for this in the left-hand navigation.



On the “Developers” page, there is a single field in which you should only list the names of the developer accounts.



A separate line should be used for each account name. As the startup script only takes effect once the login has taken place, there are no security concerns here. Anyone whose name is recognized as a developer is already logged in. All names in this list are recognized when logging in to the startup script and receive all the settings of a developer account accordingly, such as

- Own developer menu with developer dropdown for all windows
- Navigation setting with which all navigation entries are displayed.

Toolbox window for translations

The toolbox is an additional window for managing translations and text labels. To open the toolbox, use one of the drop-down menus used for the developer windows or use one of the buttons found here and there.

The toolbox is a specific view of the MTL Multilingual Text Labels module. Since the generation of text labels is always needed by developers, it is outsourced to a narrow window that can be used in layout mode directly next to each page.

Toolbox (FM Starter 3.0)

×

English

Find text labels / Translations

+

🗑

☒

Search

Click to copy:		ID	Label	Edit	
<<L::gT[481]>>	↶	481	Theme Object Info	✎	🗑
<<L::gT[480]>>	↶	480	Theme Parts Info	✎	🗑
<<L::gT[340]>>	↶	340	Standard Theme Fm Starter	✎	🗑
<<L::gT[341]>>	↶	341	Themes Introduction	✎	🗑
<<L::gT[479]>>	↶	479	Mtl Start Translation	✎	🗑
<<L::gT[28]>>	↶	28	Street	✎	🗑
<<L::gT[478]>>	↶	478	Mtl Intro	✎	🗑
<<L::gT[477]>>	↶	477	Navigation Sorting	✎	🗑
<<L::gT[476]>>	↶	476	Navigation Intro	✎	🗑
<<L::gT[475]>>	↶	475	Basic Settings Intro	✎	🗑
<<L::gT[474]>>	↶	474	Application Introduction	✎	🗑
<<L::gT[421]>>	↶	421	Regular Login Information	✎	🗑

461 / 461

Copy

☐ 1 Label

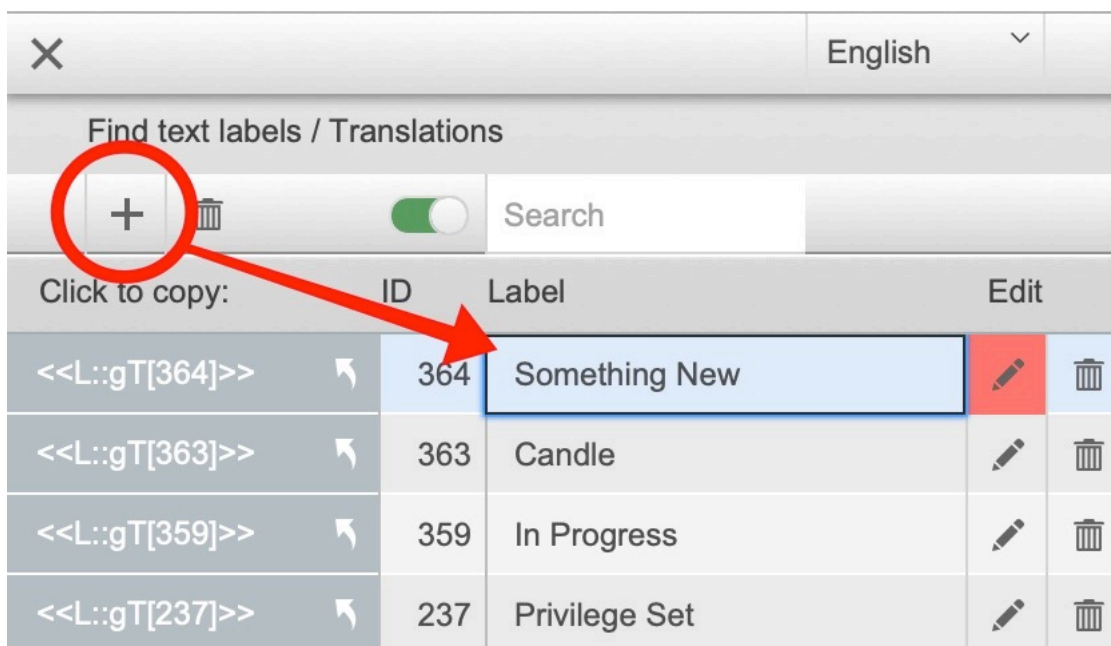
☐ 2 Translation ID

☒ 3 Merge Field

☐ 4 Field

How to create a new text label

Click Plus (+) to add a new text field and enter a name to define the text field. This is not the label itself, but an internal name for the label. The length of a text label can consist of words only, but also of phrases or whole paragraphs. For longer texts, you need something by which you can easily distinguish one label from another.



To continue, press [TAB] and a pop-up will open with all the language fields in which you can create the correct text labels. Alternatively, you can also click on the Edit icon.

The automatic DeepL translation can also be activated directly in this popover window. All you have to do is enter your own API key for the DeepL API and you can have it translated automatically.

Toolbox (FM Starter 3.0)

×

English

Find text labels / Translations

+

🗑️

🔴

Search

Click to copy:

ID

Label

Edit

×

Delete Record

translate

↺

EN	Delete record	✎	🗑️
DE	Datensatz löschen	✎	🗑️
IT	Supprimer l'enregistrement	✎	🗑️
ES	Borrar registro	✎	🗑️
JA	レコードの削除	✎	🗑️
NL	Verwijder record	✎	🗑️

<<L::gT[38]>>	↺	38	Edit	✎	🗑️
<<L::gT[39]>>	↺	39	Name	✎	🗑️
<<L::oTI40]>>	↺	40	Version	✎	🗑️

461 / 461

Copy

● 1 Label

● 2 Translation ID

● 3 Merge Field

● 4 Field

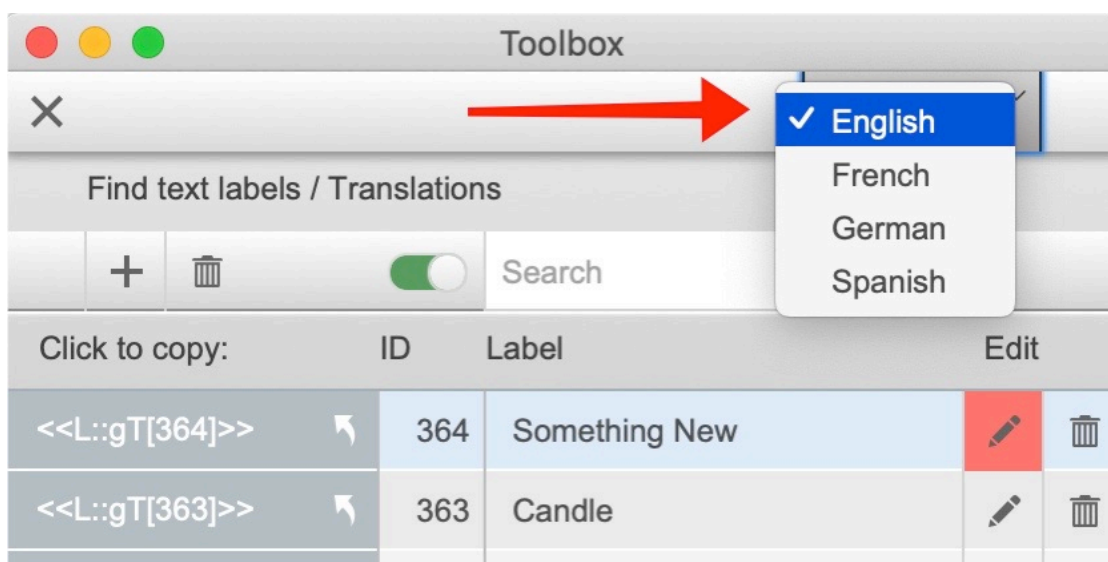
When you are finished, click outside the pop-up menu.

DeepL-API

How and where to activate the DeepL translation can be found > [here](#).

How to make new text labels visible

New text labels are not automatically visible. The translations must be reloaded. You can reload the translations by selecting any (also: the same) language from the language selection pop-up menu.



How to copy text labels to your layout

The left-hand column called Click To Copy shows a preview of how you can copy a placeholder for text labels. Click to copy and then paste into your layout.

X				English	
Find text labels / Translations					
+		<input checked="" type="checkbox"/>	Search		
Click to copy:	ID	Label	Edit		
<<L::gT[364]>>	364	Something New			
<<L::gT[363]>>	363	Candle			
<<L::gT[359]>>	359	In Progress			
<<L::gT[237]>>	237	Privilege Set			

As you can see in this example, you are copying a merge field as a placeholder that you can easily place on any layout as a text element. However, there are other options you can choose from. You will find these options at the bottom of the toolbox window to meet the requirements for text labels as fields etc.

<<L::gT[348]>>	348	Product Information		
<<L::gT[345]>>	345	How To Introduction		

339 / 339

Copy

☐ 1 Label
☐ 2 Translation ID
☒ 3 Merge Field
☐ 4 Field

If you select a different option here, a different preview will be displayed in the left-hand column. You will copy exactly what you see in the left-hand column.

Further information on text labels can be found in the documentation for the [MTL Multilingual Text Labels](#) module.

ToDo window

The “To Do” window is created as a list of tasks, bug fixes and functions. As a simple aid, it can be used to collect all the necessary work in one place, link it to a specific version and check the progress of each part.

To Do

×

+

To Do's

English

▼

Select

▼

×

Number

Version Scheduled

Title

Status

1 To do

10009

1.0

German manual

1 To do

▼

3 In progress

10008

1.0

English manual

3 In progress

▼

4 Done

10007

2b096

Developer Window optimized

4 Done

▼

10006

2b096

Plug-in registration implemented

4 Done

▼

10004

2b095

Basic theme simplified

4 Done

▼

10005

2b095

UAC Edit Email bugfix

4 Done

▼

10003

2b094

Startup script simplified

4 Done

▼

10002

2b093

UAC Bugfixes

4 Done

▼

10000

2b092

GNV removed relations

4 Done

▼

10001

2b092

ToDo module basics

4 Done

▼

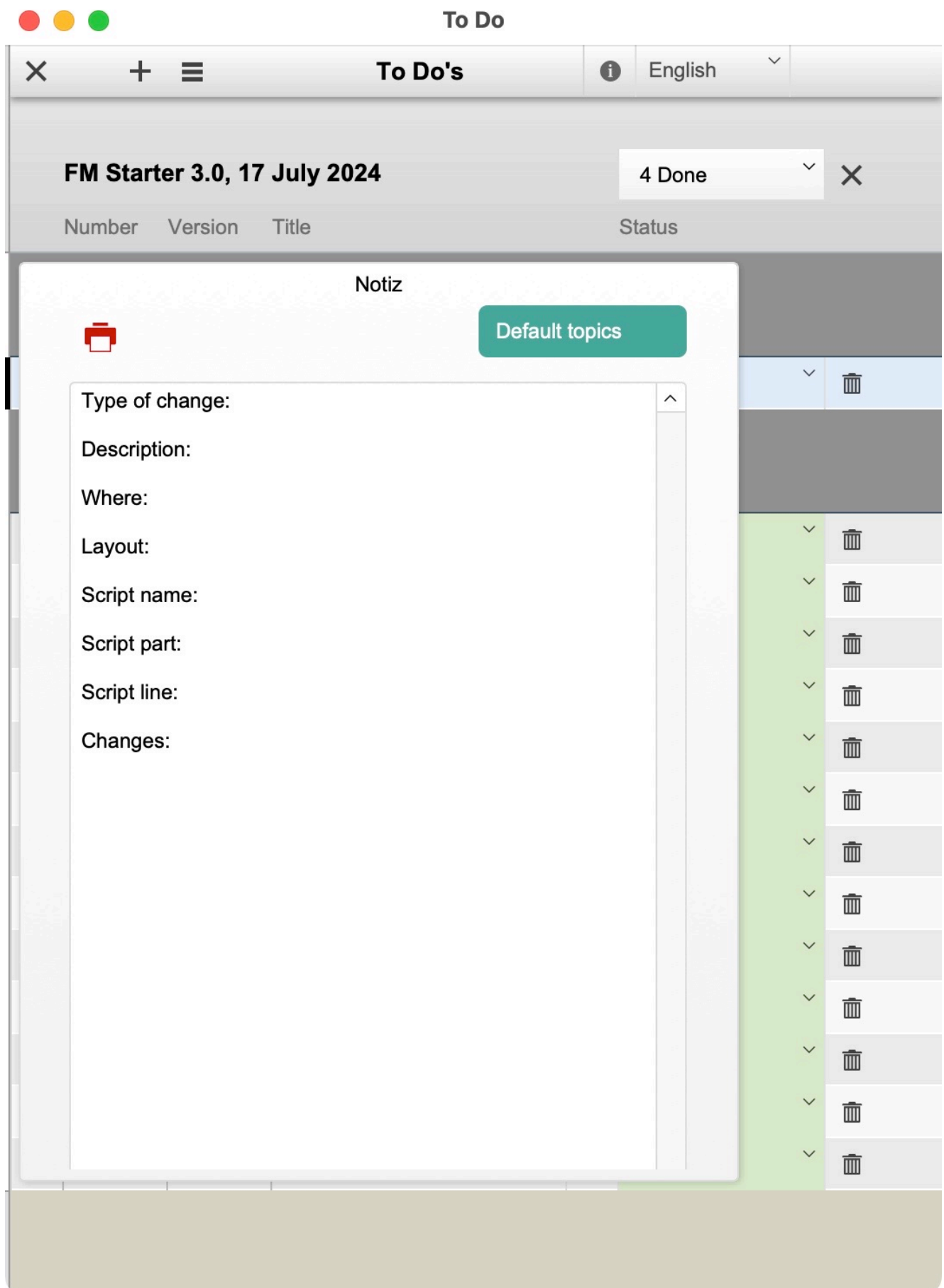
The list is sorted according to the specified status. Each entry has a unique number and further information can be added:

1. Version for the intended implementation
2. Label for orientation in the list
3. Link to the detail popover
4. Status button
5. Button "Delete".

There is no functionality beyond this point. However, it should be easy to create a list of fixes and improvements and display them somewhere on an end user page.

Add new to-do item


To add a new element, click on the plus (+) symbol. Add the information. A standard outline can be written directly into the field using the green button.



The ToDo window provides an overview of all adjustments and corrections and can also be used as a planning tool for in-house development.

The print icon allows you to generate a print preview, for example as a basis for a meeting.

To Do

<	10097		1 To do	▼	
---	-------	--	---------	---	---

Neues ToDo

Type of change:

Description:

Where:

Layout:

Script name:

Script part:

Script line:

Changes:

There is also a list view that you can use:

To Do

×

+

≡

To Do's

i

English

▼

FM Starter 3.0, 17 July 2024

4 Done

▼

×

Number

Version

Title

Status

1 To do

▼

	10097		Neues ToDo	<div>i</div>	1 To do	▼	
<div>4 Done</div> <div>▼</div>							
	10096	3.0	Window flashing reduced	<div>i</div>	4 Done	▼	
	10095	3.0	Old help window removed	<div>i</div>	4 Done	▼	
	10094	3.0	Layout Details Enhanced	<div>i</div>	4 Done	▼	
	10093	3.0	GoToLayout enhanced	<div>i</div>	4 Done	▼	
	10092	3.0	Dutch added as language	<div>i</div>	4 Done	▼	
	10091	3.0	Smaller fixes and improvements	<div>i</div>	4 Done	▼	
	10090	3.0	ToDo-window: DefaultTopics	<div>i</div>	4 Done	▼	
	10089	3.0	Main window size dynamic	<div>i</div>	4 Done	▼	
	10088	3.0	Fix Unlock Module	<div>i</div>	4 Done	▼	
	10087	3.0	GN Global Navigation	<div>i</div>	4 Done	▼	
	10086	3.0	New Help module	<div>i</div>	4 Done	▼	
	10084	3.0	ToDo-window: Info-Icon	<div>i</div>	4 Done	▼	
	10083	3.0	Start screen	<div>i</div>	4 Done	▼	

The list view can be printed or exported:

To Do

<

FM Starter

List of changes (History)
Printed 17 July 2024.

	Neues ToDo	1 To do
3.0	Window flashing reduced	4 Done
3.0	Old help window removed	4 Done
3.0	Layout Details Enhanced	4 Done
3.0	GoToLayout enhanced	4 Done
3.0	Dutch added as language	4 Done
3.0	Smaller fixes and improvements	4 Done
3.0	ToDo-window: DefaultTopics	4 Done
3.0	Main window size dynamic	4 Done
3.0	Fix Unlock Module	4 Done
3.0	GN Global Navigation	4 Done
3.0	New Help module	4 Done
3.0	ToDo-window: Info-Icon	4 Done
3.0	Start screen	4 Done
3.0	DeepL fix	4 Done
2.9	Navigation Settings	4 Done
2.9	Plug-ins Registration backup	4 Done
2.9	User Login enhanced	4 Done
2.9	FMR Unlock removed	4 Done

Modules

The modules are at the core of FM Starter. Find out more about the FileMaker modules and the modules we use in FM Starter.

What are modules?

The FileMaker modules are a proposal by Todd Geist, which he has been promoting on a website since 2013. This website no longer exists. However, the idea of FileMaker modules was to make it easier to use FileMaker code for multiple projects. Since FileMaker is not a text-based programming language, the exchange remains difficult. But these were the ideas behind FileMaker modules:

- Each module is stored as a separate file
- Each module can be implemented in other files via a few predefined steps
- Each module has its own folder
- Each folder has a fixed structure
- Each module has a description in the ReadMe script
- A license is attached to each module, which is also available in a script within the module folder.
- All modules supplied with the FM Starter can be used free of charge if you have purchased the FM Starter. You can reuse the modules individually in other projects you make, but you cannot pass these modules on to other people. Exceptions can be defined per module: Check the licenses in the module files!
- You can find free modules at fmstarter.com.
- FileMaker modules have now been replaced by FileMaker add-ons.

Modules and their administration pages can be accessed via the developer dashboard.

Freedom to improve

Think of modules as starting points. Depending on the module and its functionality, you may want to customize it. Because each module separates functionality and parameterization, it is common for modules to have a settings page through which they can be easily adapted to the respective needs.

Modules and add-ons

Since FileMaker Pro 19.1 there are so-called “add-ons”. These are very similar to modules, but are much easier to integrate. Both modules and add-ons promote a modular approach to development and make it particularly easy to incorporate existing developments from other developers.

However, modules and add-ons also show how modular development can be achieved. In a nutshell: Develop small building blocks as a module or add-on. Use several modules for more complex processes.

Modules and add-ons are useful for functions that can be used multiple times. Each module or add-on should solve a clear task.

FM Unlock

FM Unlock module

The FM Unlock module is a decoding module for our FM Registrations product, which you can purchase at fmstarter.com.

What is FM Registrations?

FM Registrations is a license code generator for your FileMaker solutions. If you have a solution that you want to sell multiple times, you can create unlock codes for your customers and sell them a unique license, update or upgrade.

What is FM Unlock used for?

FM Unlock is the counterpart to FM Registrations. While FM Registrations encodes the registration data, FM Unlock decodes this registration data.

Do I need FM Unlock?

You only need FM Unlock if you use FM Registrations. This module simplifies the setup of all the functions you need to work with FM Registrations. FM Starter is the perfect companion for FM Registrations.

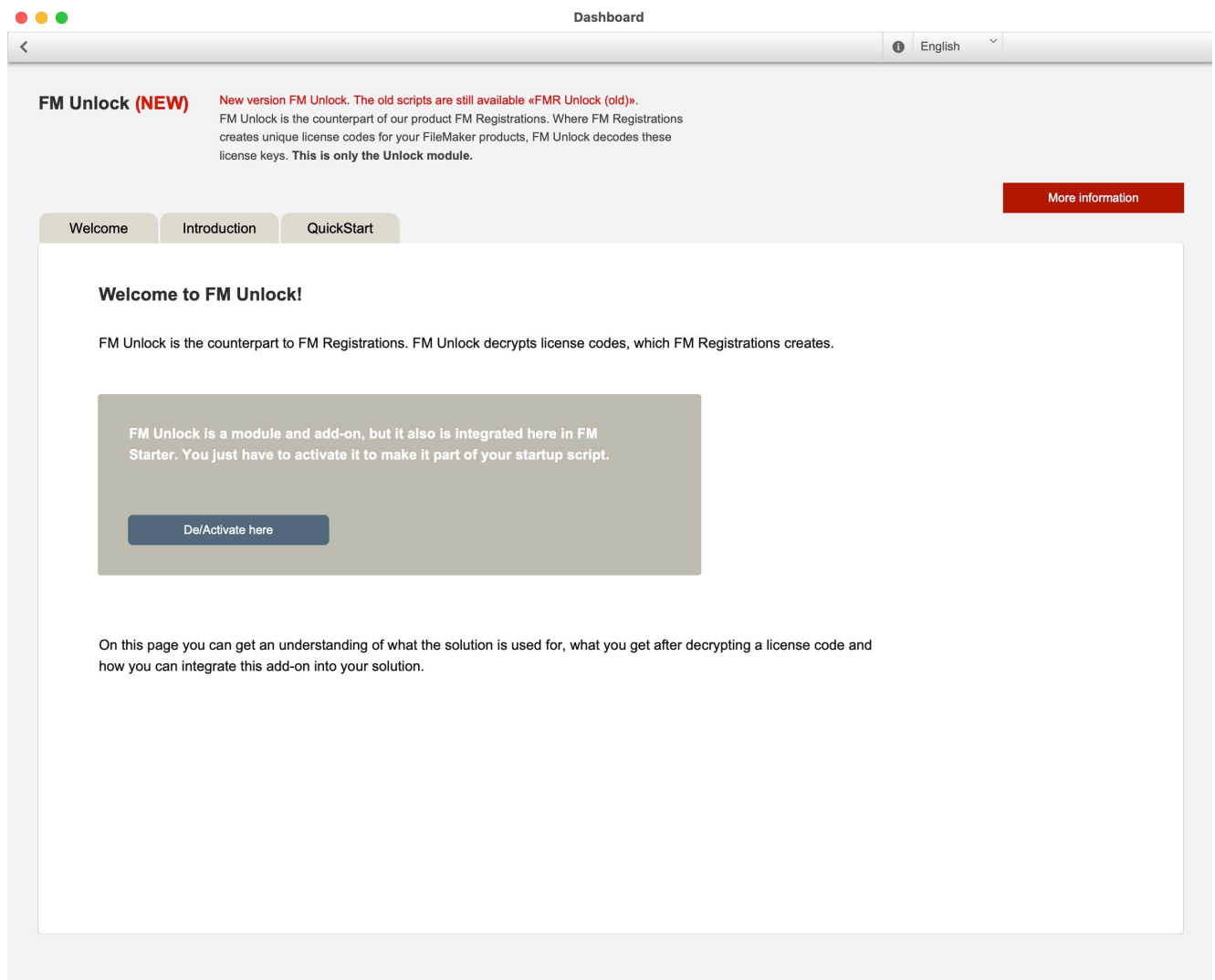
FM Unlock introduction

Before you start implementing your own secret keys, make sure that you test the module in test mode.

To see all settings of the module, go to developer Dashboard > FM Unlock (left navigation).

As of FM Starter 2.30, there is a new display. Nothing has changed in terms of functionality, but the interface has been simplified to make it easier to use.

Here is the “Welcome” start screen:



Here you will also find the information that FM Unlock is already part of FM Starter. So you don't need to integrate anything else. The sample file from FM Registrations is already part of the solution here. However, if you want to activate the solution for end users, it must be activated. There is also a note about this on the welcome screen. The activation itself is carried out in the settings for the start script.

After the initial instructions, you will now find a brief introduction to FM Unlock under the "Introduction" tab.

The screenshot shows a web application titled "Dashboard" in a browser window. The interface has a header with a language selector set to "English". Below the header, there's a section for "FM Unlock (NEW)" with a red banner stating: "New version FM Unlock. The old scripts are still available «FMR Unlock (old)». FM Unlock is the counterpart of our product FM Registrations. Where FM Registrations creates unique license codes for your FileMaker products, FM Unlock decodes these license keys. This is only the Unlock module." To the right of this section is a red button labeled "More information".

Below the banner is a tabbed interface with three tabs: "Welcome", "Introduction", and "QuickStart". The "Introduction" tab is currently selected. The content of the "Introduction" tab is as follows:

Learn

Learn the basics first. Install later.

Learn

You learn the basics under the tab "QuickStart". Under QuickStart, first use the Test Mode. This shows you an example code with the exact same workflow as for a real code. You will see the entire registration process and the information which you, as a developer, get out of it.

Once you understand the basic functionality, think about a workflow. You have to create the proper workflow in your solution. This is an example:

Opening file > Startup Script with Checks

- load any available registration (see: FM Unlock Startup Script)
- check the registration
- if there is no valid registration, > forward to a registration page
- if there is a valid registration, > forward to a login page
- if the login is valid, > forward to the final landing page.

You need: Startup Script, Initial secure FileMaker Account listed in the File Settings, Registration page, Checks. A basic workflow is part of this example file "FM Unlock". The information is also available in the add-on.

Install

It is easy to install an add-on. You can use the free tool "Add-On Lab FREE" to find the proper directory to simply install the ZIP file.

At the bottom right of the content area, there is a yellow button labeled "Add-On Lab FREE".

FM Unlock should be integrated into a workflow when the program is started. This is particularly easy in FM Starter - it only takes two steps:

1. Activate (see Welcome screen)
2. Use the secret keys of the FM registration definition (see section “How it works”)

The procedure provides for any existing registration data to be read in when the file is started, on the basis of which you can then check whether they are valid. If no registration data is available, a link is provided to a registration layout where the license data can be entered. If the registration is completed successfully, an account and menu set are activated and redirected to a new page. This process is already part of FM Unlock and only needs to be activated. It can then be customized as required.

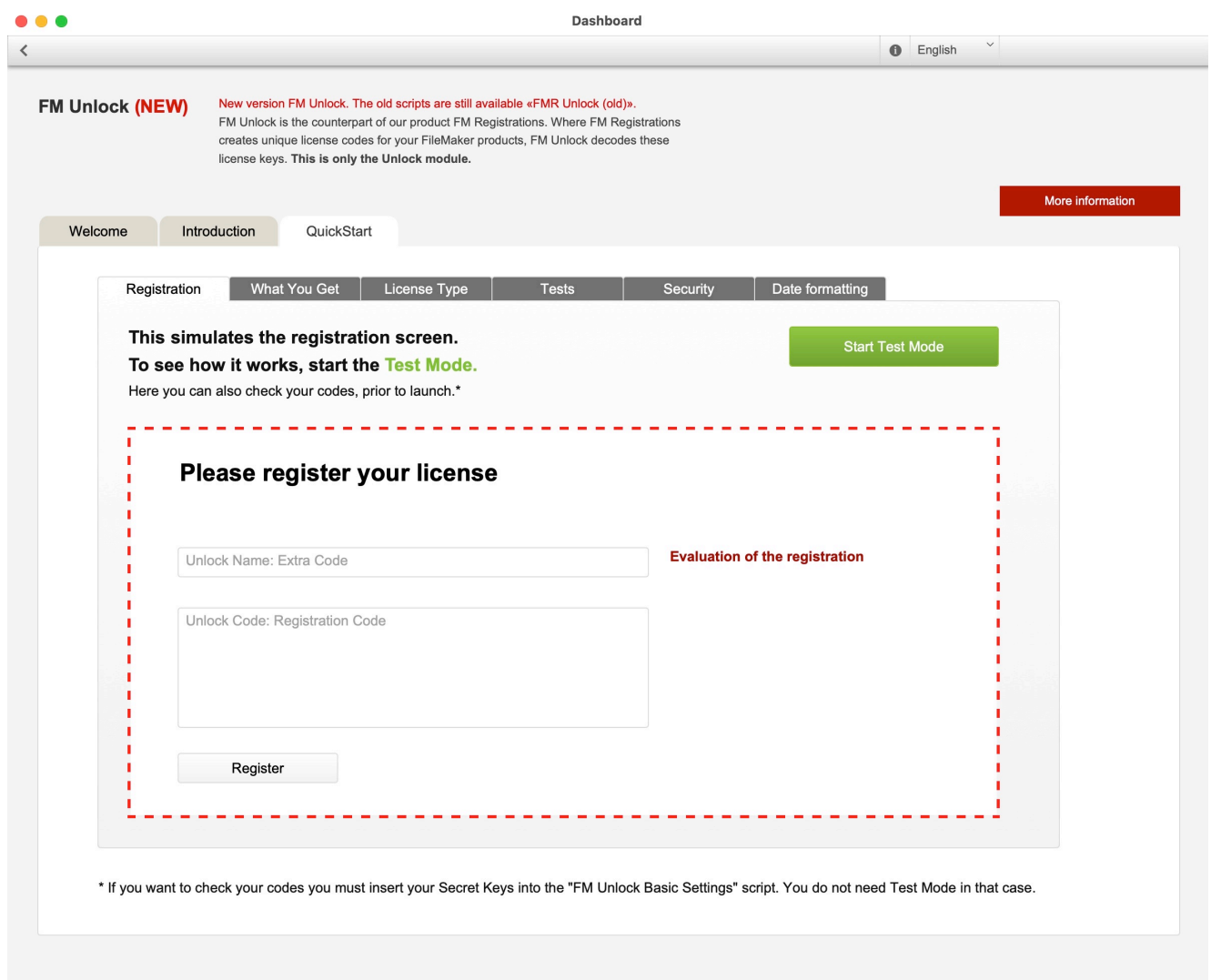
Now that you understand the basics of the planned workflow, it's time to see how it works.

Test mode

The test mode of FM Unlock shows how it works. It is a reference to how it works with successful implementation and what information you receive during registration. Get to know the module using the test mode.

QuickStart

The test mode can be found under the “QuickStart” tab. Everything that the FM Unlock module does is clearly displayed here. It is the place where you can understand everything about the decryption of the license codes.



The first thing you will see is the “Please register your license” area. This part, outlined with a dashed red line, is what you need as a “registration window” for the user. Here, however, it is for test purposes only.

Click on the green “Start Test Mode” button to start the test mode. The test mode is activated and license data is written into the fields.

The screenshot shows a web application titled "FM Unlock (NEW)" with a "Dashboard" header. The main content area has a navigation bar with "Welcome", "Introduction", and "QuickStart" tabs. Below this is a sub-navigation bar with "Registration", "What You Get", "License Type", "Tests", "Security", and "Date formatting" tabs. The "Registration" tab is active, displaying a simulation of the registration screen. It includes a green "Start Test Mode" button and a checkbox labeled "gFmrTestMode" which is checked. The main form area is titled "Please register your license" and contains two input fields: "gFmrExtraCode" with the value "MarkBright" and "gFmrRegistrationCode" with a long alphanumeric string. A blue "Register" button is at the bottom of the form. To the right of the form, the text "Evaluation of the registration" is visible. A footer note states: "* If you want to check your codes you must insert your Secret Keys into the 'FM Unlock Basic Settings' script. You do not need Test Mode in that case."

Dashboard

English

FM Unlock (NEW) New version FM Unlock. The old scripts are still available «FMR Unlock (old)». FM Unlock is the counterpart of our product FM Registrations. Where FM Registrations creates unique license codes for your FileMaker products, FM Unlock decodes these license keys. **This is only the Unlock module.**

More information

Welcome Introduction QuickStart

Registration What You Get License Type Tests Security Date formatting

This simulates the registration screen.
To see how it works, start the **Test Mode.**
Here you can also check your codes, prior to launch.*

Start Test Mode

☒ gFmrTestMode

Please register your license

gFmrExtraCode
MarkBright

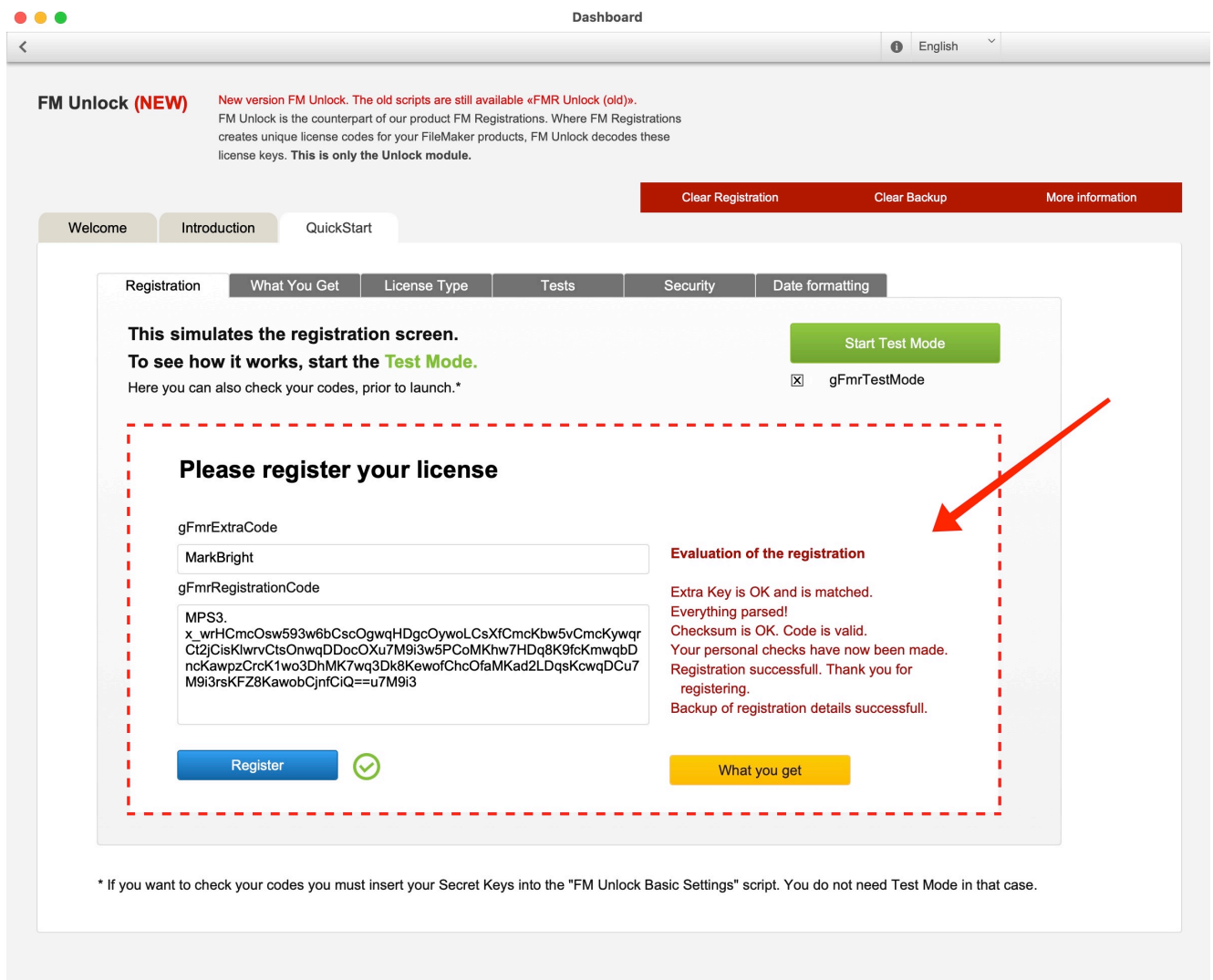
gFmrRegistrationCode
MPS3.
x_wrHCmcOsw593w6bCscOgwqHDgcOywoLCsXfCmcKbw5vCmcKywqr
Ct2jCisKlwrVctsOnwqDDocOXu7M9i3w5PCoMKhw7HDq8K9fcKmwqbD
ncKawpzCrcK1wo3DhMK7wq3Dk8KewofChcOfaMKad2LDqsKcwqDCu7
M9i3rsKFZ8KawobCjnfCiQ==u7M9i3

Evaluation of the registration

Register

* If you want to check your codes you must insert your Secret Keys into the "FM Unlock Basic Settings" script. You do not need Test Mode in that case.

Now click on the blue “Register” button to check the registration. The main script of FM Unlock runs in the background and checks all the details. Feedback on all the individual steps now appears on the right:

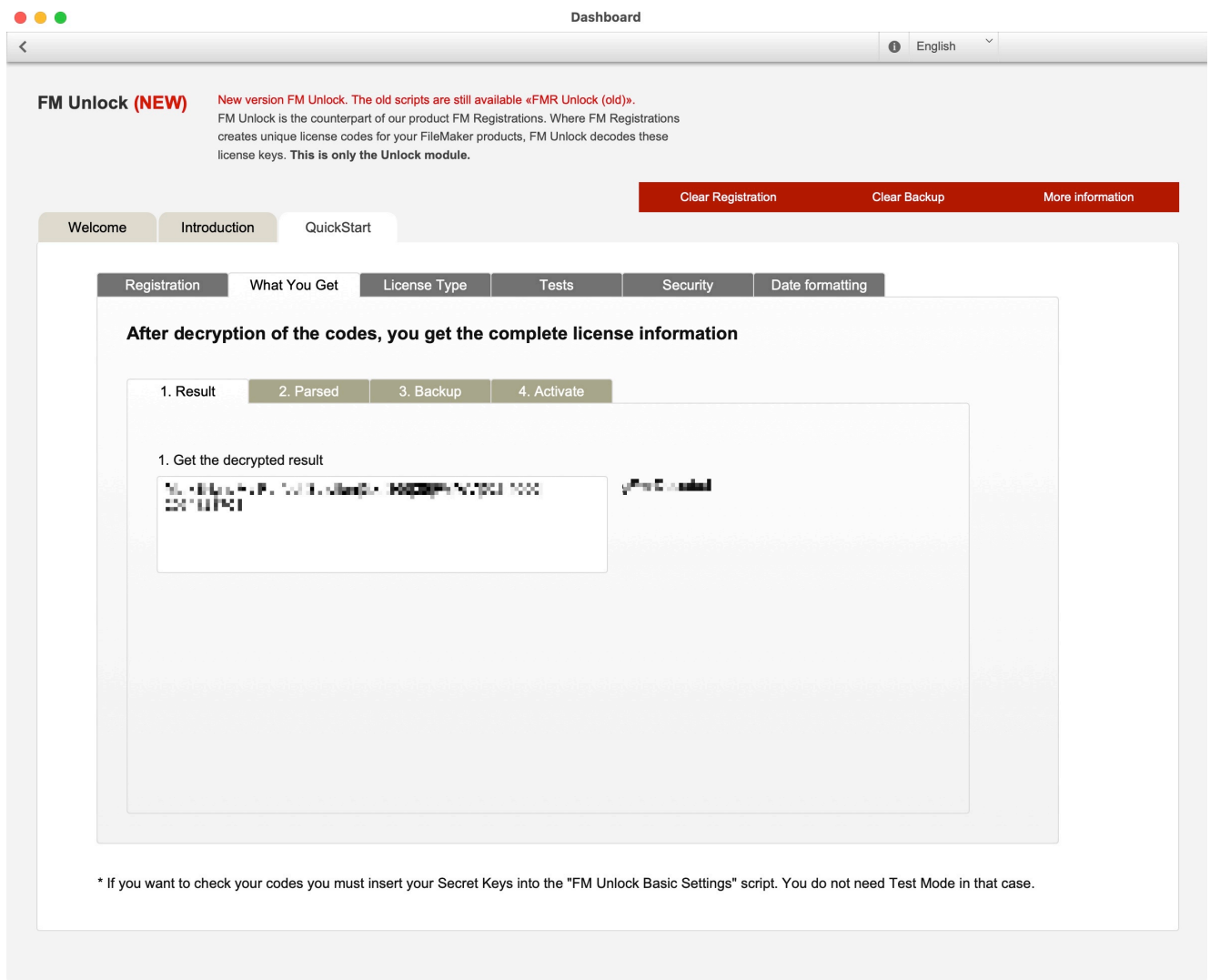


This evaluation shows what has been done and that the individual steps are OK. A yellow “What you get” button now appears at the end. Click on this button and you will be forwarded to the second tab “What you get”. The decrypted information from the license code is shown here and broken down into individual parts so that it can be used.

1. Result

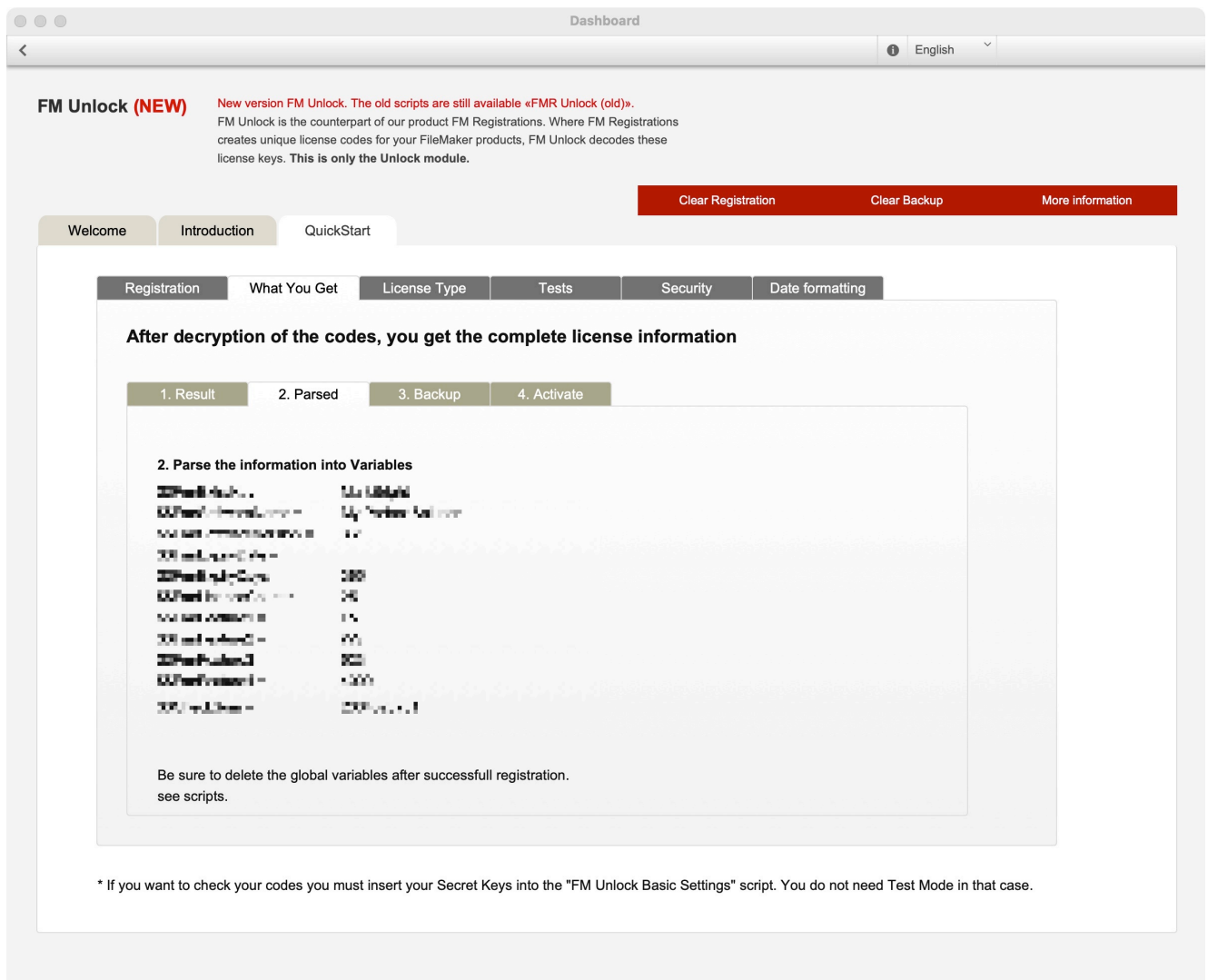
This first part contains the string that you generated in FM Registrations when you created the license. You will not find the operation of FM Registrations in this manual, but in simple terms FM Registrations is the code generator in which you describe a license. The license is a line of text (string) which is then

encrypted into a license code. FM Unlock decrypts this information and makes the same line of text available again:



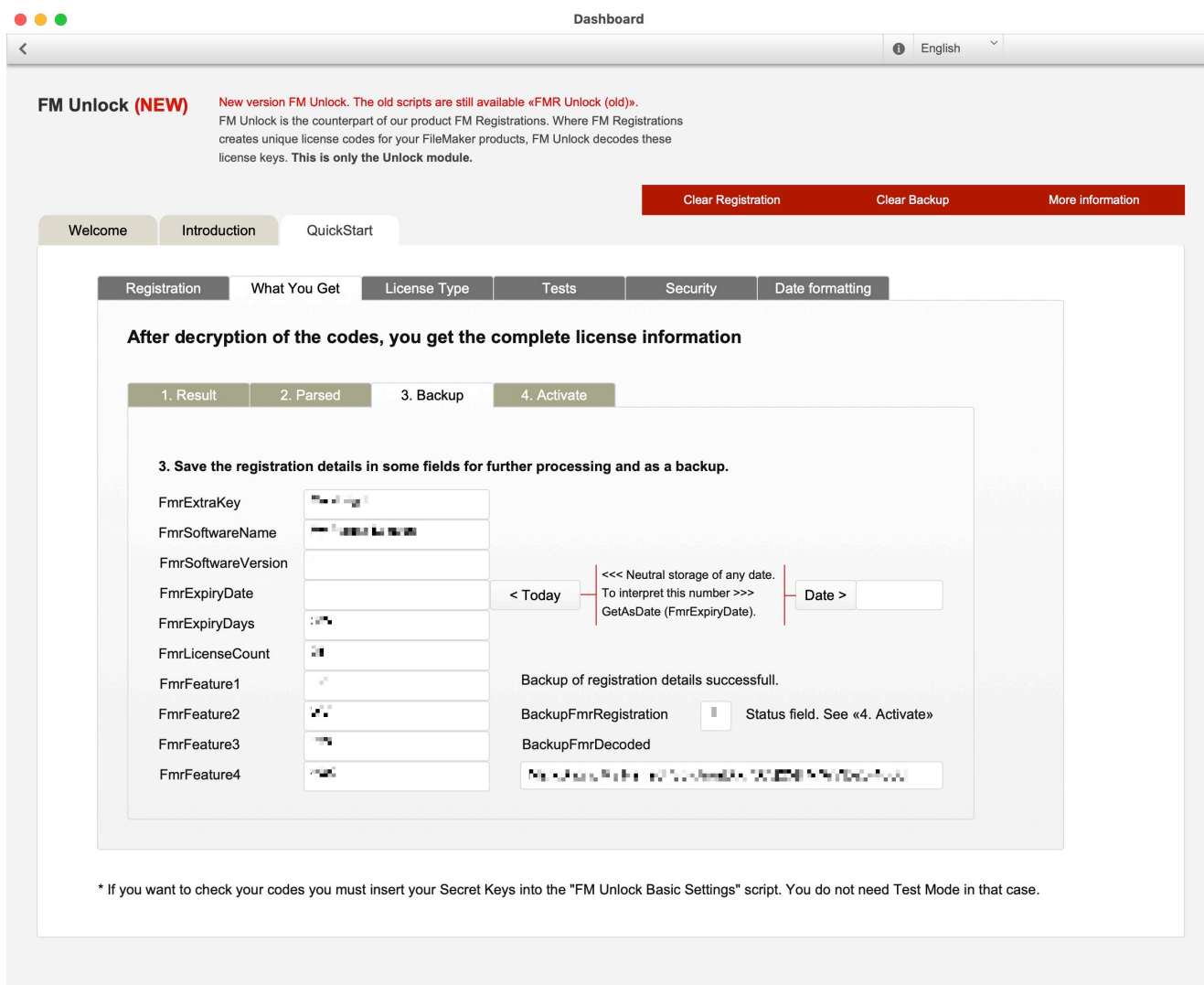
2. Parsed

Parsing describes the process by which a text string is broken down into individual parts. This is what makes the information useful. In this sub-tab, the information is displayed with global variables. This is only during the registration process. The information can therefore be used very easily in your own scripts so that you can make settings in your own scripts based on this information.



3. Backup

FM Unlock now also creates a data record to save the registration. The registration can therefore be read each time the program is started. When the program is started, you can therefore see whether the software has already been registered and whether the registration is valid. Otherwise you would have to repeat the registration every time you start the program. However, this is precisely what you want to avoid.



4. Activate

Once all data has been checked and saved, a final evaluation of the entire process is required. If the registration is valid, then an “OK” is required. This is the summary. This summary is written in a field. The check of this field at program start is the fastest check of the registration. You can write something in this field, for example “1” or “A” or “OK”. Make sure you test on what you enter here.

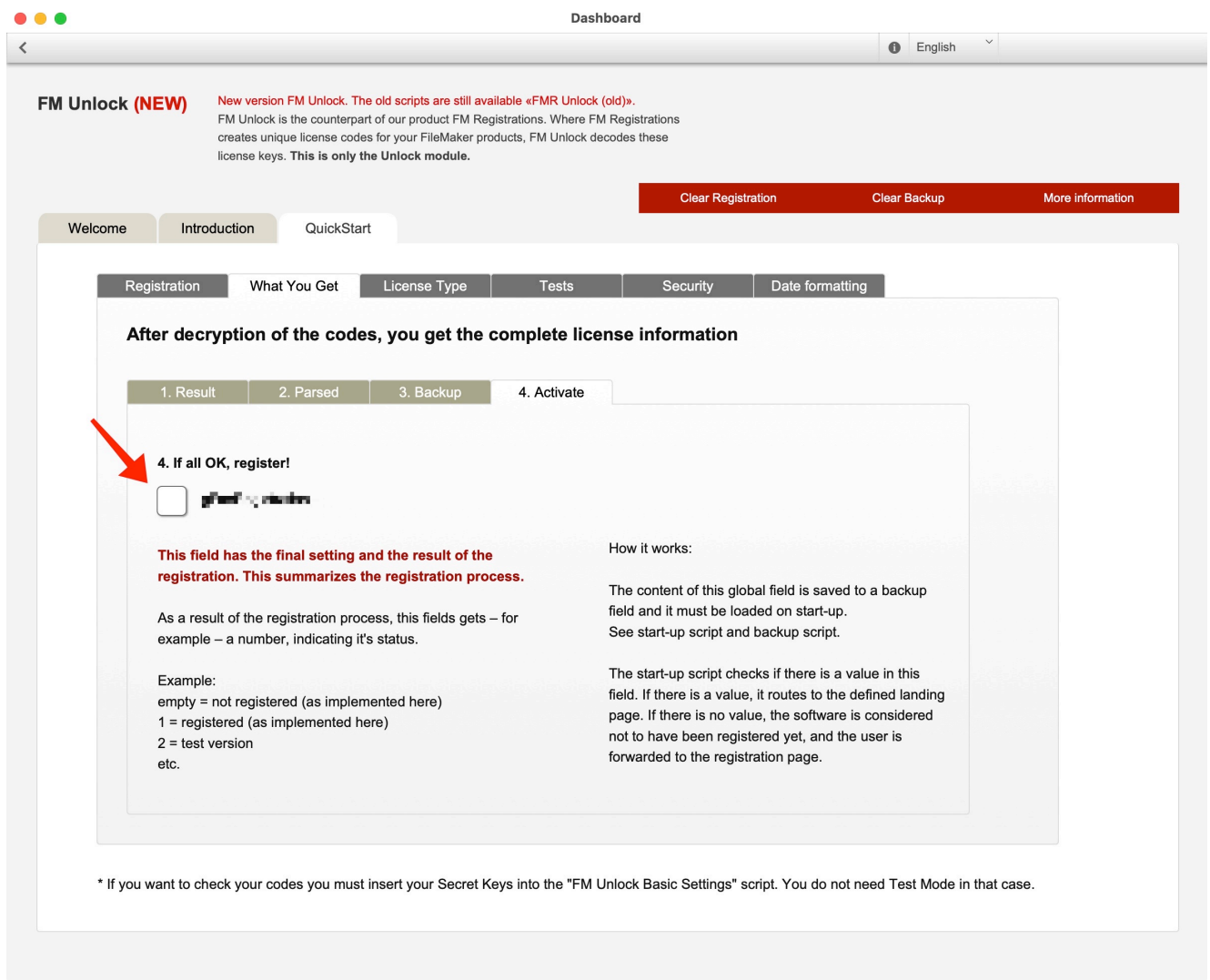
Let us imagine that we define the result as follows:

- A “blank field” means: Not registered
- “1” means: Registered

- “2” means: test version

You can of course define your own values here. It is not the value that is important, but the function. When the program is started, the registration is loaded and a quick check of the value in this field can already determine which version the customer has. The value can function as a “switch” in a workflow.

The activation field:



FM Unlock (NEW) New version FM Unlock. The old scripts are still available «FMR Unlock (old)».

FM Unlock is the counterpart of our product FM Registrations. Where FM Registrations creates unique license codes for your FileMaker products, FM Unlock decodes these license keys. **This is only the Unlock module.**

[Clear Registration](#) [Clear Backup](#) [More information](#)

Welcome Introduction **QuickStart**

Registration What You Get License Type Tests Security Date formatting

After decryption of the codes, you get the complete license information

1. Result 2. Parsed 3. Backup 4. Activate

4. If all OK, register!

☐

This field has the final setting and the result of the registration. This summarizes the registration process.

As a result of the registration process, this field gets – for example – a number, indicating it's status.

Example:
empty = not registered (as implemented here)
1 = registered (as implemented here)
2 = test version
etc.

How it works:

The content of this global field is saved to a backup field and it must be loaded on start-up. See start-up script and backup script.

The start-up script checks if there is a value in this field. If there is a value, it routes to the defined landing page. If there is no value, the software is considered not to have been registered yet, and the user is forwarded to the registration page.

* If you want to check your codes you must insert your Secret Keys into the "FM Unlock Basic Settings" script. You do not need Test Mode in that case.

From the beginning to the end of the registration, the test mode shows the exact process and what you should get as a result. The test mode uses the

same scripts as a normal registration. If the test mode works, your activations will also work. The only condition is of course that you have used the secret keys from FM Registrations in FM Unlock and have created the license codes with the same secret keys.

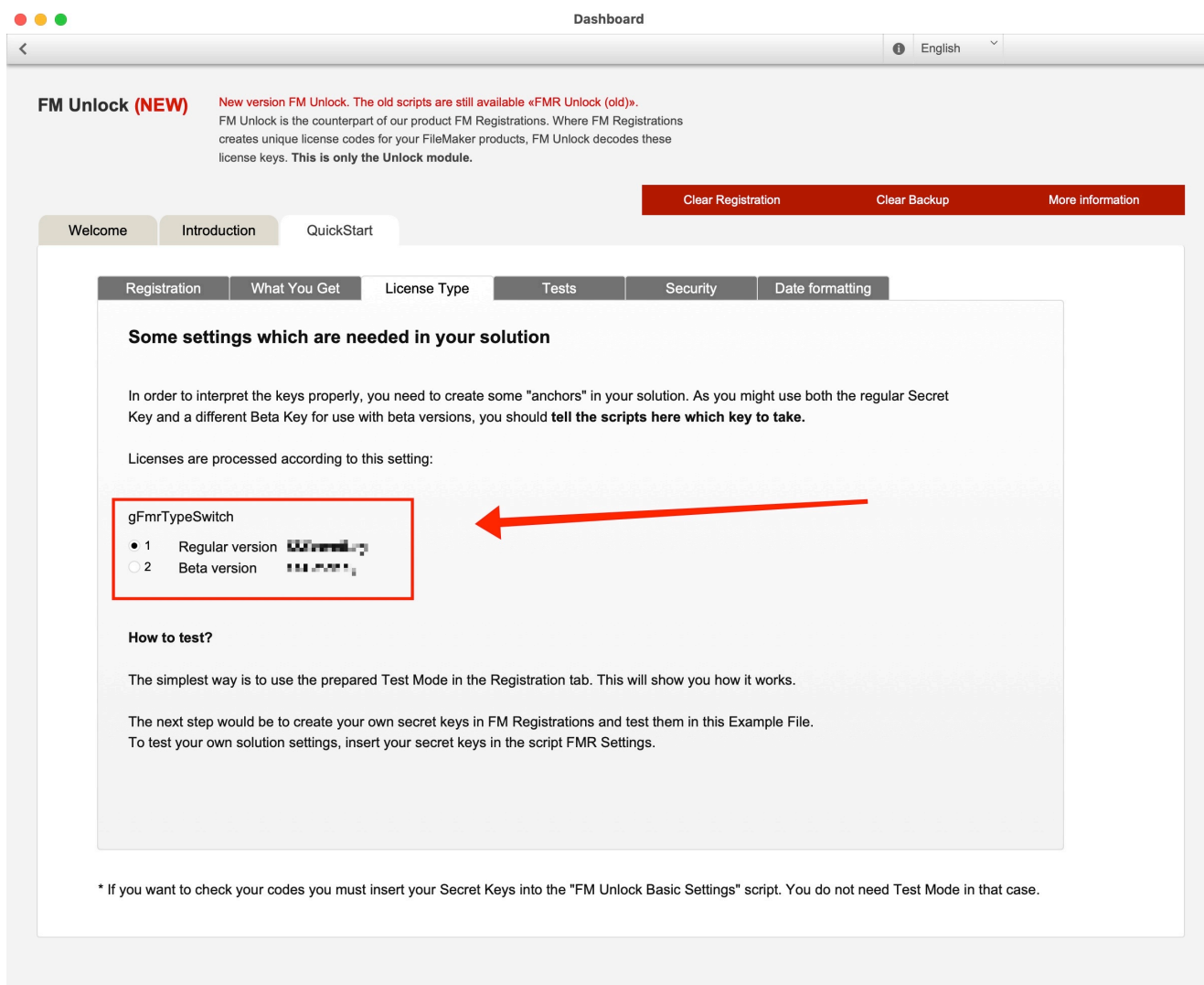
License type

The third tab “License Type” is only about one setting. You can choose whether you want to verify a regular license or a beta version. The reason for this distinction is that you often work with beta versions before the official release of a new software solution. You may want to run these with certain limitations so that they do not remain valid indefinitely (for example).

The license type switch now defines which secret key is used for decryption. This already starts in FM Registrations. Which keys are involved?

In FM Registrations, you create a separate entry for each software, each version. This allows you to restrict license codes to certain versions. Each entry has 2 secret keys, one for the regular version and another for the beta version.

Both keys must also be transferred to FM Unlock (see: “[How it works](#)”). This switch therefore decides which secret key is used for decryption.



For this to work correctly, the license code must already be defined as “Beta” or “Regular” in FM Registrations. Each software update of the software can then be activated with the same key as long as this switch is set to the same. If you want to switch from a beta version to a regular version, for example, this requires a switch and new unlock codes are needed.

If the release of the software becomes definitive, a new version appears for which “regular license codes” are created. It must therefore already be taken into account during code generation and then in parallel here in this default setting for the software version. In FM Unlock, this switch must also be correctly set to “Regular version” so that the “Regular” decryption works.

- If you create a license code for a regular version, it will only work if the license type of your application is set to “Regular version”.
- If you create a license code for a beta version, it will only work if the license type of your application is set to “Beta version”.

This difference allows you to clearly separate beta versions from final versions.

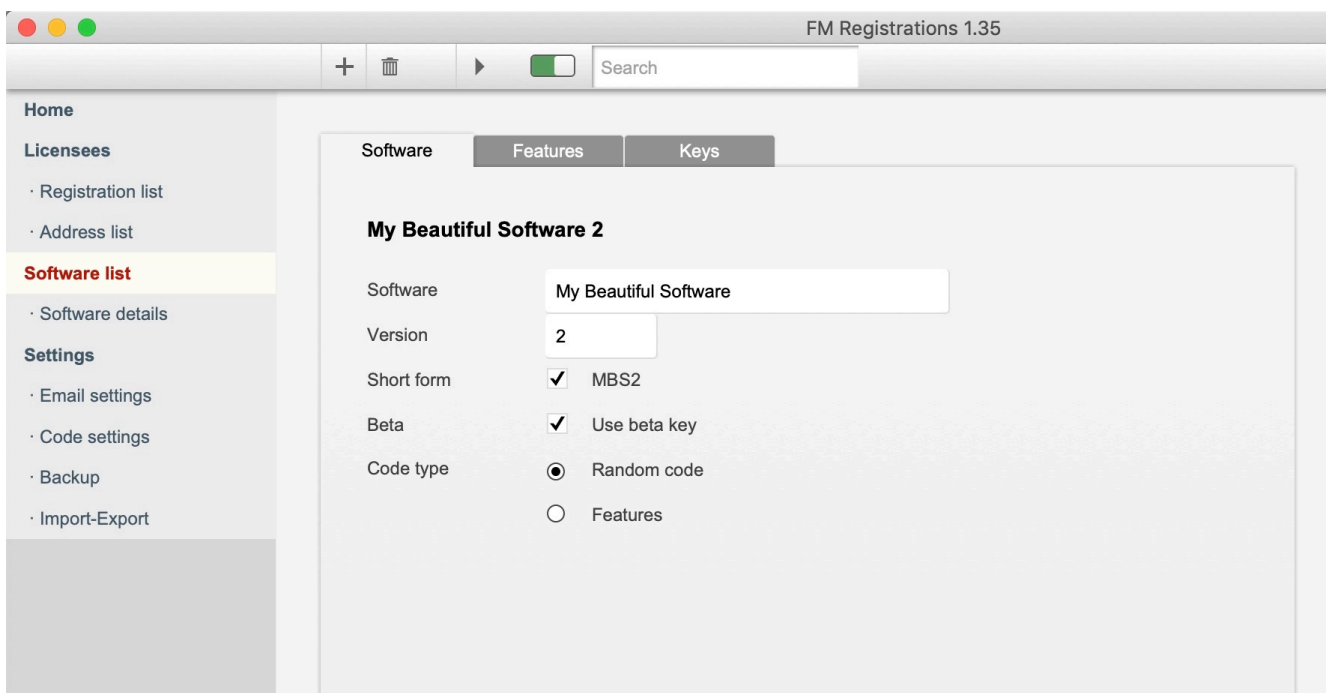
FM Unlock, how it works

FM Unlock decrypts the license codes that you generate with FM Registrations. To achieve this, you need to copy some secret keys from FM Registrations and paste them into a script in FM Unlock. These keys are used to encrypt and decrypt the license code information.

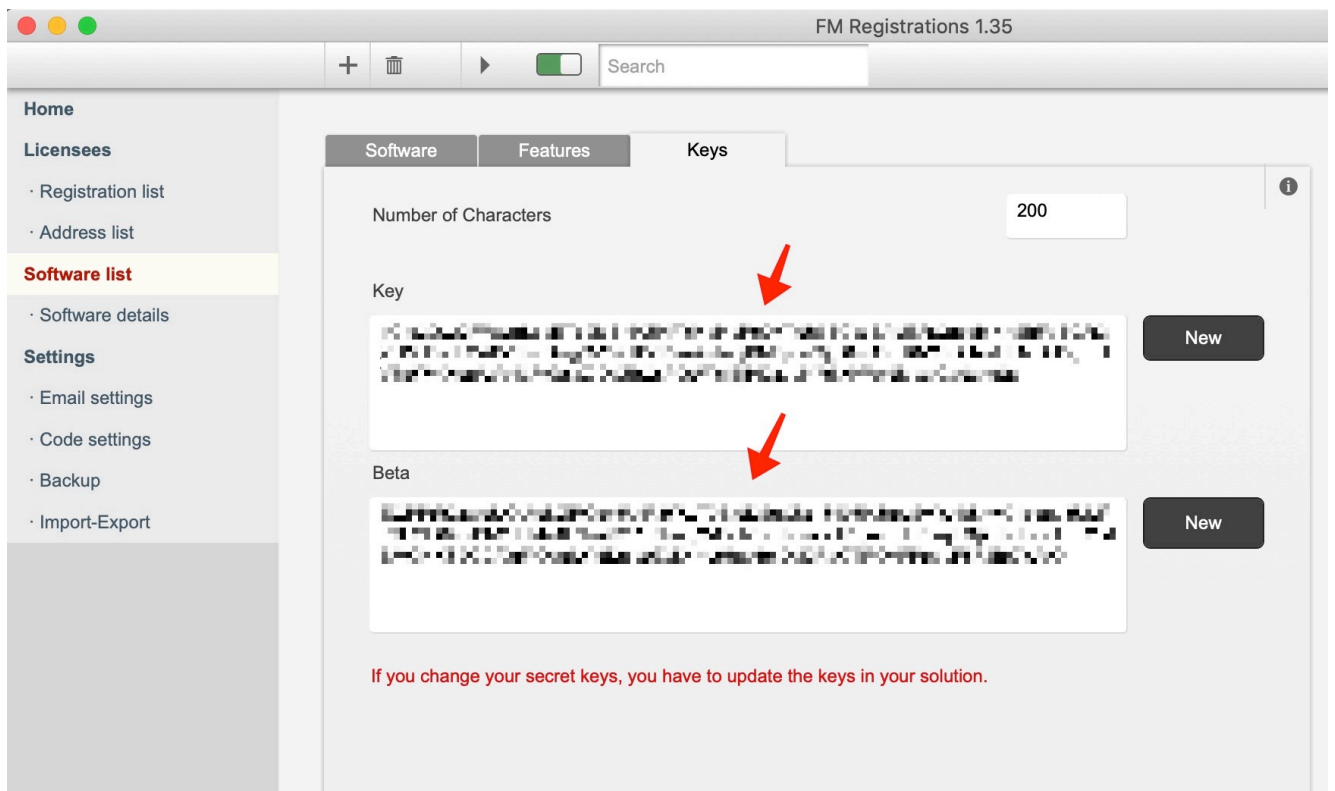
Where can you get these secret keys?

You need FM Registrations, an optional software that you can purchase at fmstarter.com. FM Registrations is also part of the FM Developer Bundle.

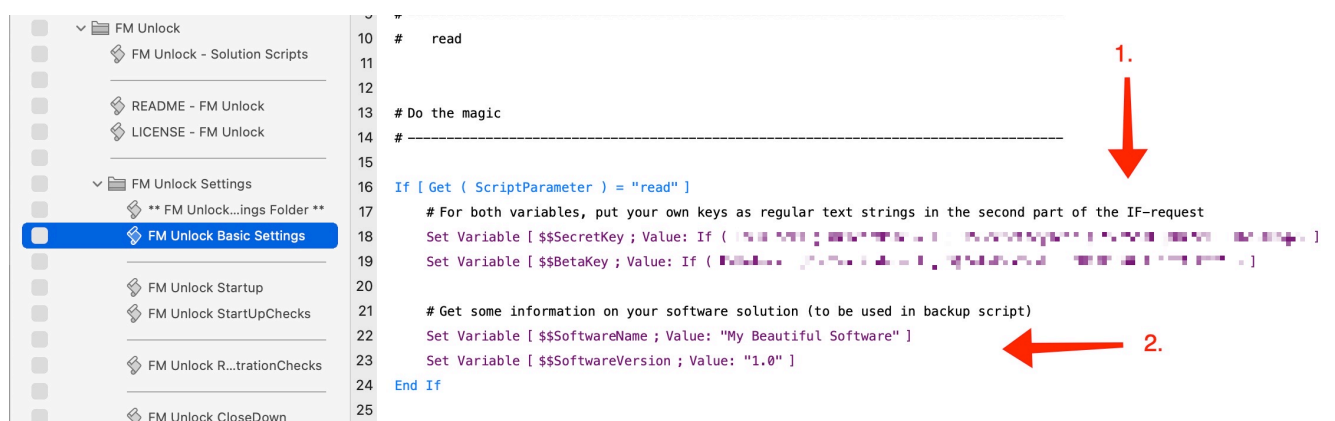
Create a new software product in FM Registrations, enter a name and a version:



Now go to the Keys tab, where you create both a **standard** and a **beta** key for a beta version. These are the secret keys.



Copy these keys and paste them into FM Starter, the FM Unlock module, in the FM Unlock Basic Settings script, as shown in the screenshot below.



GNV Global Navigation

This module dynamically creates a navigation that is truly global, as in “global variables” and “global fields”. We believe that this navigation module is one of the most powerful and useful parts of FM Starter.

Brilliantly global

This solution is brilliantly global as you can just copy and paste the navigation block into any layout of any table and it works. No further settings are required. This navigation is not dependent on relationships, but on 2 global repeating fields that are displayed as a navigation block via a button bar.

Table-based

This navigation module is table-based, i.e. you can manage each clickable element in a regular list. A navigation table also allows you to add further information to each element, such as the layout you want to navigate to or the access rights. Table-based navigation is extremely flexible. When the navigation is loaded, the values of the table are read out and made available via the aforementioned global repeating fields.

Button bar

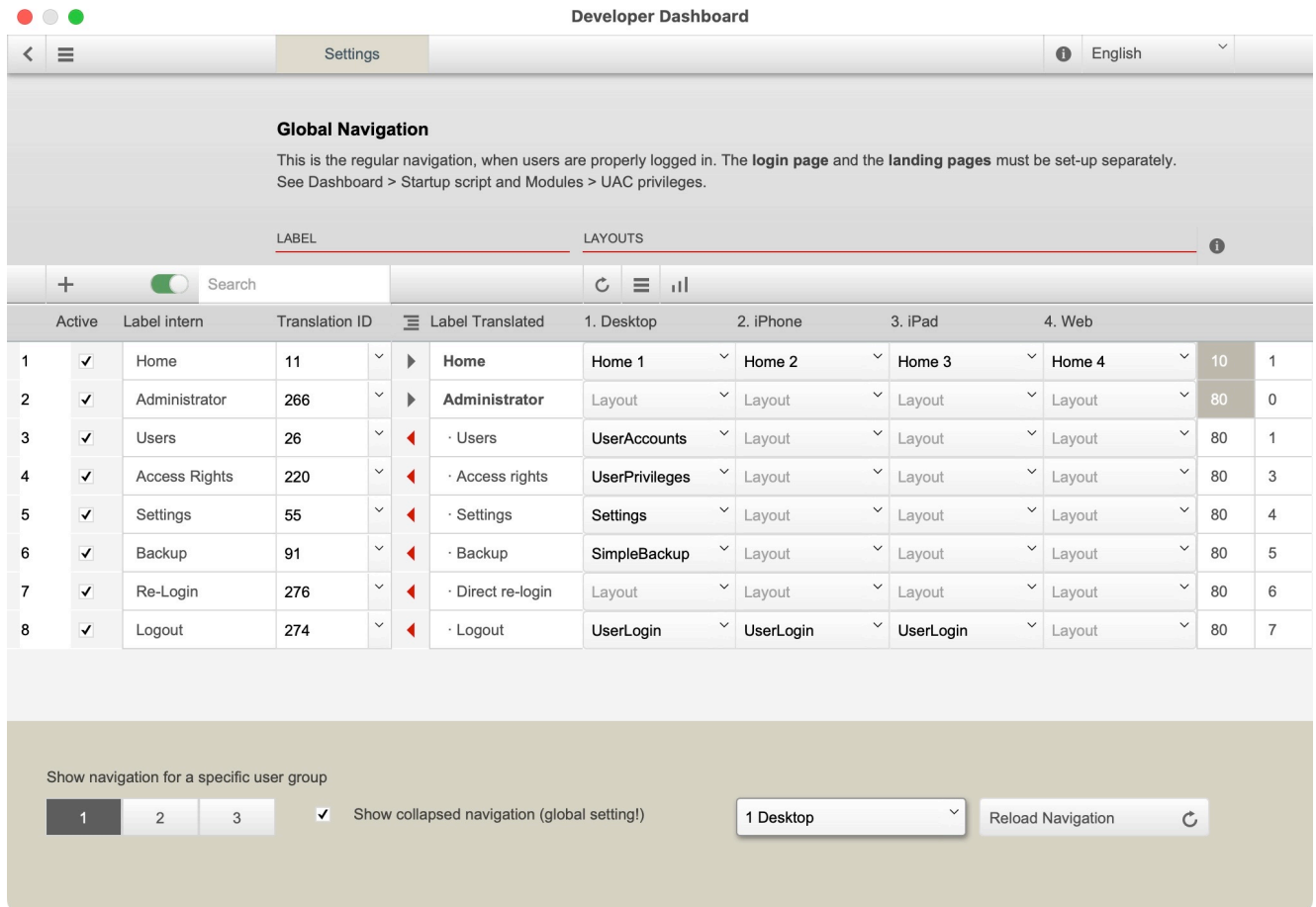
The values from the global repeating fields can now be used via a button bar. The navigation is displayed in a button bar. This button bar must be prepared for each clickable button (button bar element), but once this has been done, no further changes are required. You can simply use the prepared navigation block as it is available in the FM Starter.

Just a single navigation

This is a module to create only a single navigation block. There is no option to have a split navigation (as in: Top Navigation and Left Navigation), similar to the one in FM Starter 1. The reason for this is obvious: simpler is better. Of course, you are free to extend the navigation.

Navigation table

The navigation table is located on the developer dashboard. Open the dashboard and find the navigation on the left-hand side of the screen.



The screenshot shows the 'Developer Dashboard' with the 'Settings' tab selected. The 'Global Navigation' section is active, displaying a table of navigation elements. The table has columns for 'Active', 'Label intern', 'Translation ID', 'Label Translated', and four device layouts: '1. Desktop', '2. iPhone', '3. iPad', and '4. Web'. Each row represents a navigation element with its respective settings and counts.

	Active	Label intern	Translation ID		Label Translated	1. Desktop	2. iPhone	3. iPad	4. Web		
1	✓	Home	11	▼	▶ Home	Home 1	Home 2	Home 3	Home 4	10	1
2	✓	Administrator	266	▼	▶ Administrator	Layout	Layout	Layout	Layout	80	0
3	✓	Users	26	▼	▶ · Users	UserAccounts	Layout	Layout	Layout	80	1
4	✓	Access Rights	220	▼	▶ · Access rights	UserPrivileges	Layout	Layout	Layout	80	3
5	✓	Settings	55	▼	▶ · Settings	Settings	Layout	Layout	Layout	80	4
6	✓	Backup	91	▼	▶ · Backup	SimpleBackup	Layout	Layout	Layout	80	5
7	✓	Re-Login	276	▼	▶ · Direct re-login	Layout	Layout	Layout	Layout	80	6
8	✓	Logout	274	▼	▶ · Logout	UserLogin	UserLogin	UserLogin	Layout	80	7

Below the table, there is a section for 'Show navigation for a specific user group' with tabs 1, 2, and 3. Tab 1 is selected. There is a checkbox for 'Show collapsed navigation (global setting!)' which is checked. A dropdown menu shows '1 Desktop' and a 'Reload Navigation' button with a refresh icon.











After opening the navigation module, you will see the navigation table. Each row represents a navigation element. The table can be searched, sorted, restricted, etc. This creates a selection of elements to be displayed in the navigation block.

For example, if a main category of the navigation is shown or hidden, the list is regenerated based on a specific search and the navigation is completely recreated. This happens immediately after each click.

Several options are available per element in the navigation table. Check each option carefully:

1. Active: Only use active entries
2. Label internal: The label for this table (only for the developer)
3. Translation ID: Integration with the MTL Multilingual Text Labels translation module.
4. Label translated: The label in the current translation, based on the translation ID.
5. Layouts 1...4 are for different platforms: Desktop, iPhone, iPad and Web
6. The layouts can be filled from a popup. The popup is filled when the file is opened. There are several buttons under the title "Layouts". The first button updates the list of layouts.
7. Group: Entry of the first level in the list (without indentations). These groups are sorted first.
8. Sort: additional sorting information within groups
9. User groups: The admin (1) can see all pages. Other user groups can be assigned to a page.

If you scroll to the right, further information is hidden in this table:

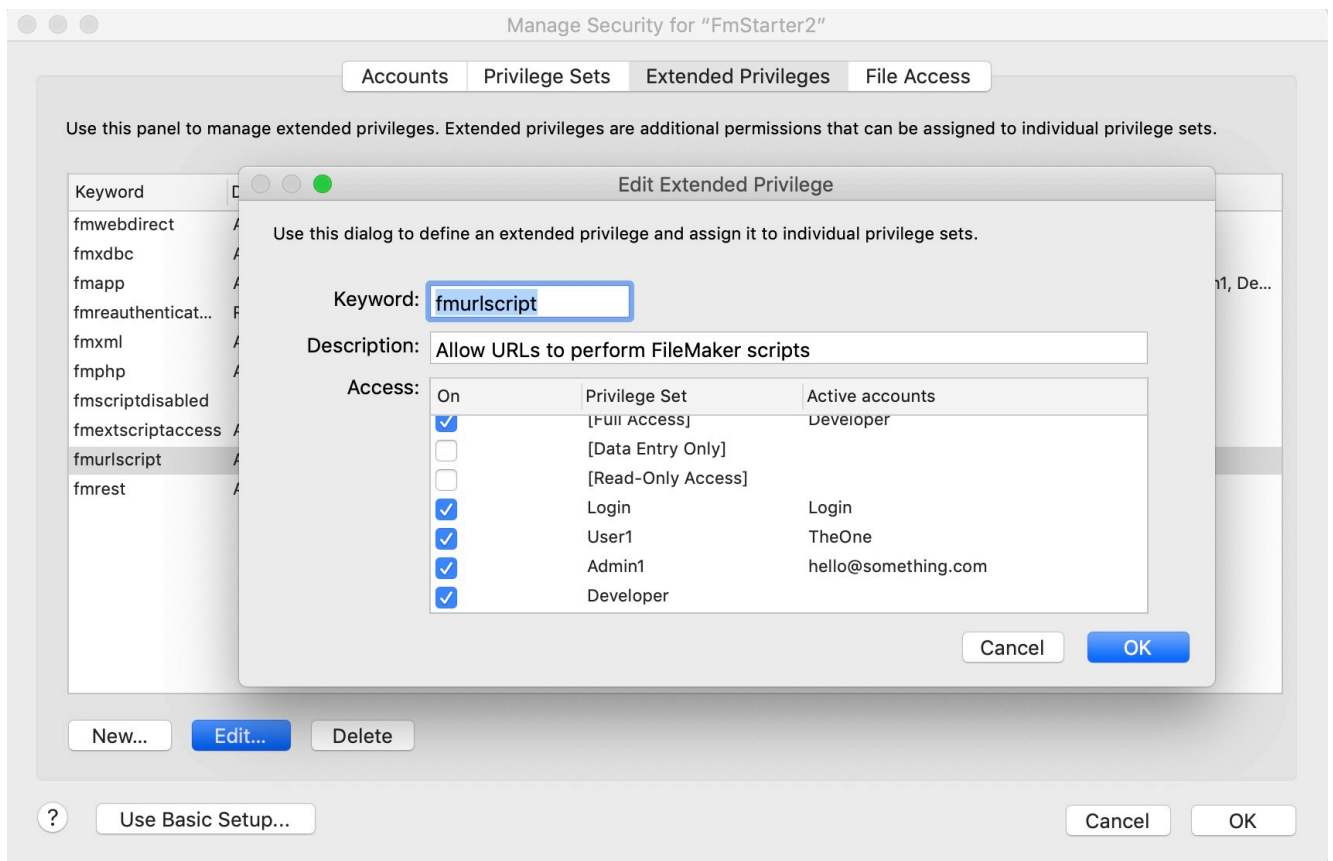
10 Records in selection / 10 records total.			
Script Name		Parameter	
Script name	Script Parameter		Remarks
Script name	Script Parameter		Remarks
UAC UserFunctions	clearfeedback		Remarks
Script name	Script Parameter		Remarks
UAC UserFunctions	sort		Remarks
Script name	Script Parameter		Remarks
Script name	Script Parameter		Remarks
Script name	Script Parameter		Layout defined in UserLogin script
UAC UserLogin	logout		Layout defined in UserLogin script
UAC UserLogin	relogin		No Layout

- Script Name: Trigger a script
- Parameters: Script parameter
- [Delete] Delete record
- Remarks: For your eyes only.

How to activate scripts

For scripts to work, you need to enable two things:

1. Enable Use Scripts in the settings section of this screen (bottom left) and
2. Enable URL usage in File > Settings > Security > Extended access rights.

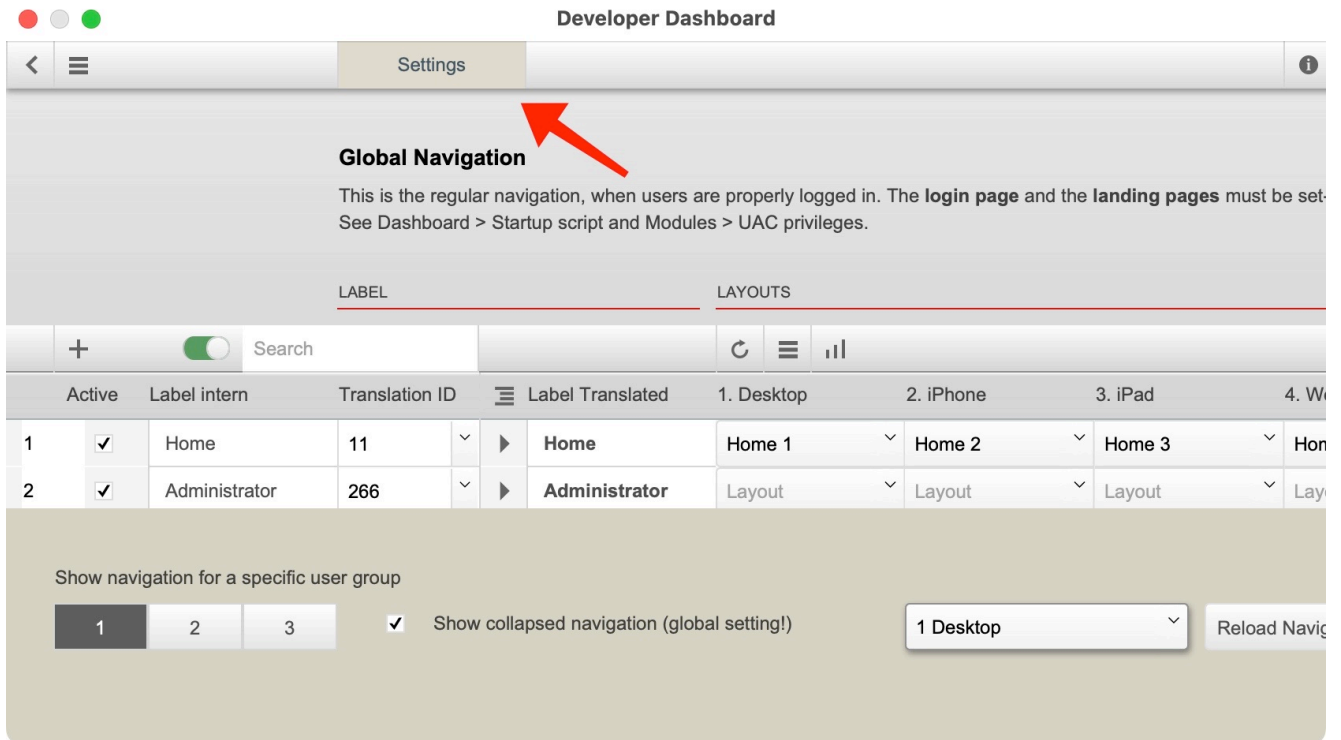


Problem case “Dot in file name”

If you use a dot in the file name (in addition to “.fmp12”), this will confuse FileMaker. We generally advise against using dots in the file name, other than to separate the file extension. If you have to do this anyway, scripts may no longer be able to be accessed. Since version 2.1, you can still use additional dots in the FileMaker database name, but you must then explicitly specify the file extension. To do this, activate the “Use Extension” option and enter a current FileMaker file extension in the text field (e.g. fmp12, fmpur).

Navigation settings

The navigation settings have been moved to a separate map window. Click on “Settings” to open this window.



The screenshot shows the 'Developer Dashboard' with the 'Settings' tab selected. A red arrow points to the 'Global Navigation' section. Below the title, there is a descriptive text: 'This is the regular navigation, when users are properly logged in. The **login page** and the **landing pages** must be set. See Dashboard > Startup script and Modules > UAC privileges.'

The main content area is divided into two tabs: 'LABEL' and 'LAYOUTS'. The 'LABEL' tab is active, showing a table with the following data:

	Active	Label intern	Translation ID		Label Translated	1. Desktop	2. iPhone	3. iPad	4. W
1	<input checked="" type="checkbox"/>	Home	11	▼	▶ Home	Home 1 ▼	Home 2 ▼	Home 3 ▼	Hon
2	<input checked="" type="checkbox"/>	Administrator	266	▼	▶ Administrator	Layout ▼	Layout ▼	Layout ▼	Lay

Below the table, there is a section titled 'Show navigation for a specific user group' with three buttons: '1', '2', and '3'. The '1' button is selected. To the right of these buttons is a checkbox labeled 'Show collapsed navigation (global setting!)' which is checked. Further right is a dropdown menu currently showing '1 Desktop'. At the bottom right of this section is a button labeled 'Reload Navig'.

The navigation settings:

Navigation Settings

Navigation settings:

☒ Active entries only

☒ Filter Groups (collapse/expand sub groups)

☒ Use Scripts

☒ Use file extension

fmp12

☐ Exclude Scripts in WebDirect

☐ Navigation Warnings

☒ AutoResize Windows

AutoResize Minimum Width

1200

Load Settings

Save Settings

Show navigation for a specific user group

1

2

3

Test and Reload Navigation

1 Desktop

Reload Navigation

Clear Navigation

Navigation shows:

1 Label

2 Layout

3 Instructions

4 Script

5 Script Parameter

6 Label Translated

Parameters (List):

Home|Home 1|10|||1|1|1

Administrator|80||1|2|1

Users|UserAccounts|80|UAC UserFunctions|clearfeedback|1|3|2

Access Rights|UserPrivileges|80|UAC UserFunctions|sort|1|4|2

Settings|Settings|80||1|5|2

Backup|SimpleBackup|80|||1|6|2

Re-Login||80|UAC UserLogin|relogin|1|7|2

Logout|UserLogin|80|UAC UserLogin|logout|1|8|2

Pro 21.0.1

1

You can use the navigation settings to configure the display, appearance and behavior of the navigation. Make sure that you save your settings when you make changes.

As the label is generated dynamically, you can also display something other than the (translated) text label, such as the layout, the stored script or the default label.

You will find the settings for the individual navigation points in the dynamically generated parameter list. These are saved separately and (re)generated together with the labels. When the navigation is loaded, the information is stored in two global repeating fields, one with the labels and one with the stored parameters.

Only active entries

Switch this option ON (default) to display only active entries. This allows you to edit the navigation entries before you make them available to all users.

Filtering groups

The filtering makes navigation collapsible. FM Starter comes with a 2-level navigation. If you click on any level 1 entry, that section will expand and collapse when you click on it again. Use the levels to create logical groups of features of pages. Filtering is ON by default, i.e. active. If a layout is stored for a level 1 entry, the system navigates to it. If a layout is missing, this entry is only expanded.

Each navigation entry can be assigned to any user group. By default, each entry is assigned to the administrator (group 1). You can activate each individual entry to display it in the navigation of any group. For second-level entries to work, you must also activate the corresponding first-level entry.

Using scripts

Navigation can be used to navigate, but also to trigger scripts. To do this, you need to activate two things:

1. activate the “Use scripts” setting.
2. activate the security settings for the respective authorization to include URLs.

(File > Manage > Security > Advanced >
Extended access rights >)

fmurlscript must be active for the authorizations used.

The use of scripts is enabled by default.

Navigation alerts

Enable some warnings in the navigation, useful when working on the navigation scripts (for developers only). Navigation warnings are OFF by default.

Enhancement of the navigation element

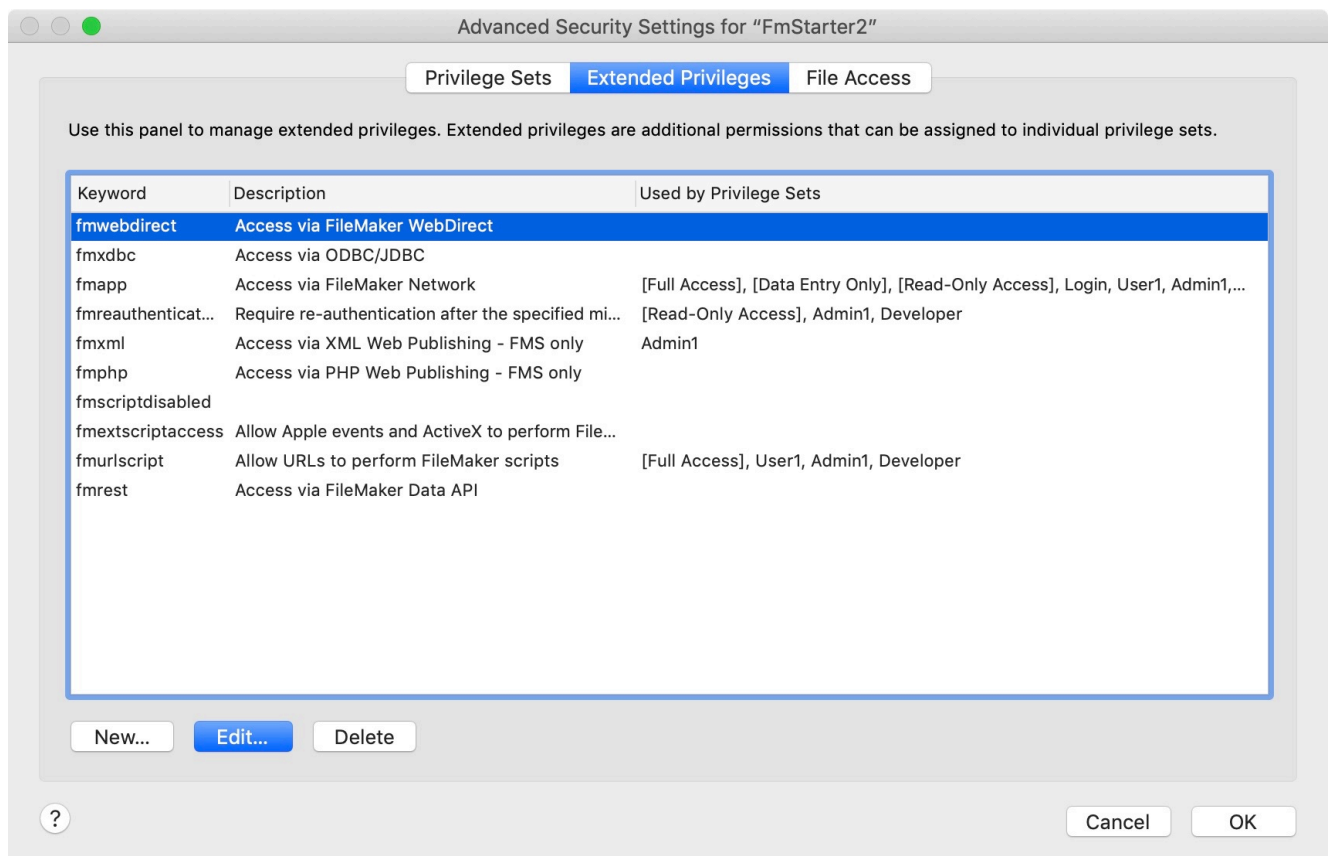
The navigation element that comes with the FM Starter has a fixed number of buttons in the button bar. If you need more, you need to create more elements in the bar (which is easy). Make sure you check the settings of each element. Each element must have a script parameter that corresponds to the number of the element. The first element has “1” as script parameter, the next “2” and so on.

This numbering is given as a script parameter because there is currently no way of querying the repeat number of the clicked element. In addition, the elements of a button bar can be moved within the bar. It is therefore up to you to maintain the correct order of the numbering, as it refers to the respective entries in the generated navigation list.

Multiple platforms

Disclaimer: Although the navigation table supports multiple platforms, further customization may be required. FM Starter is not intended to enable every detail and every possible variant, nor to circumvent or eliminate all possible limitations, but to provide a starter file for your own further development.

WebDirect, for example, is not activated in the file. You must make these settings yourself and activate WebDirect support if you need it.

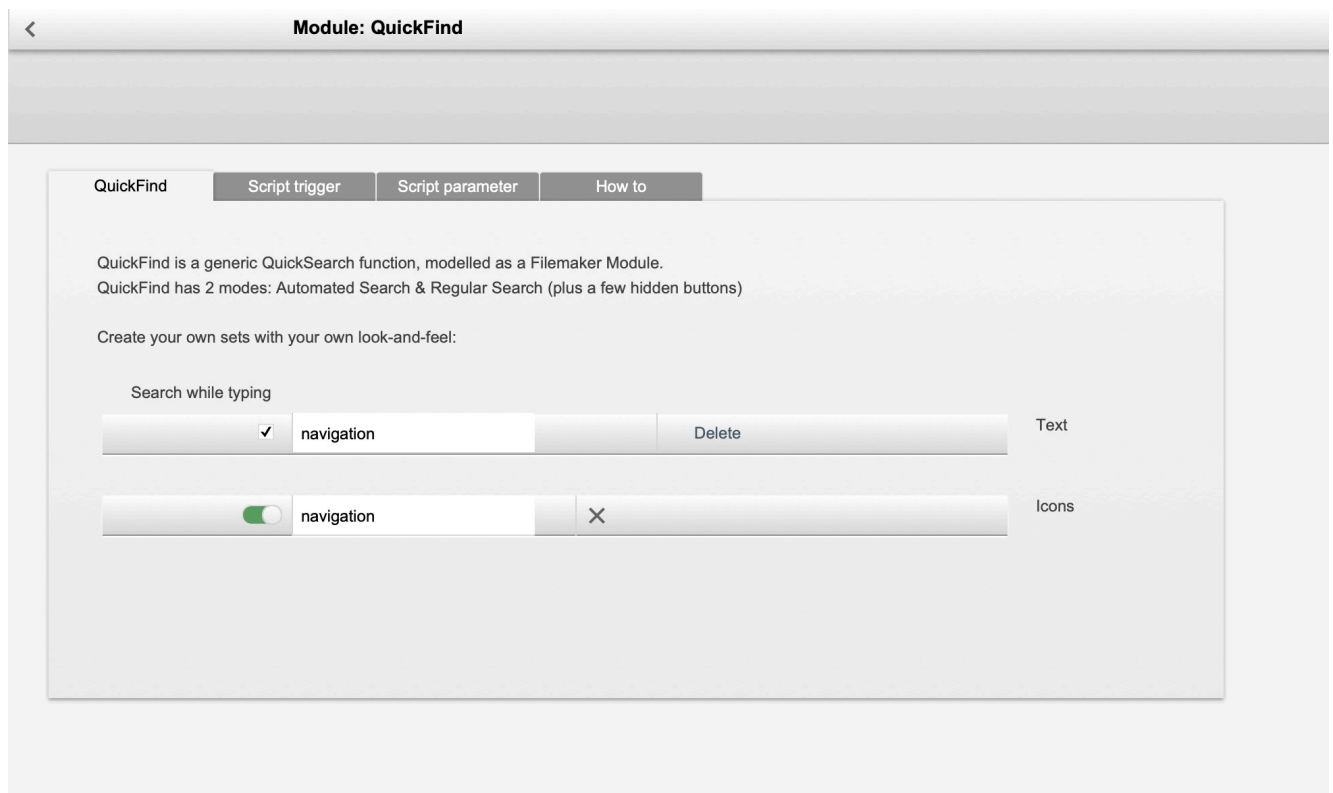


Quick Find module

The Quick Find module is a generic quick search function that is modeled as a FileMaker module. It was already integrated in FM Starter 1 and reappears with the same functionality.

This module is generic and works on any table or layout. Simply copy a complete set of buttons and fields from any existing page or from the module page to paste them anywhere. That's it!

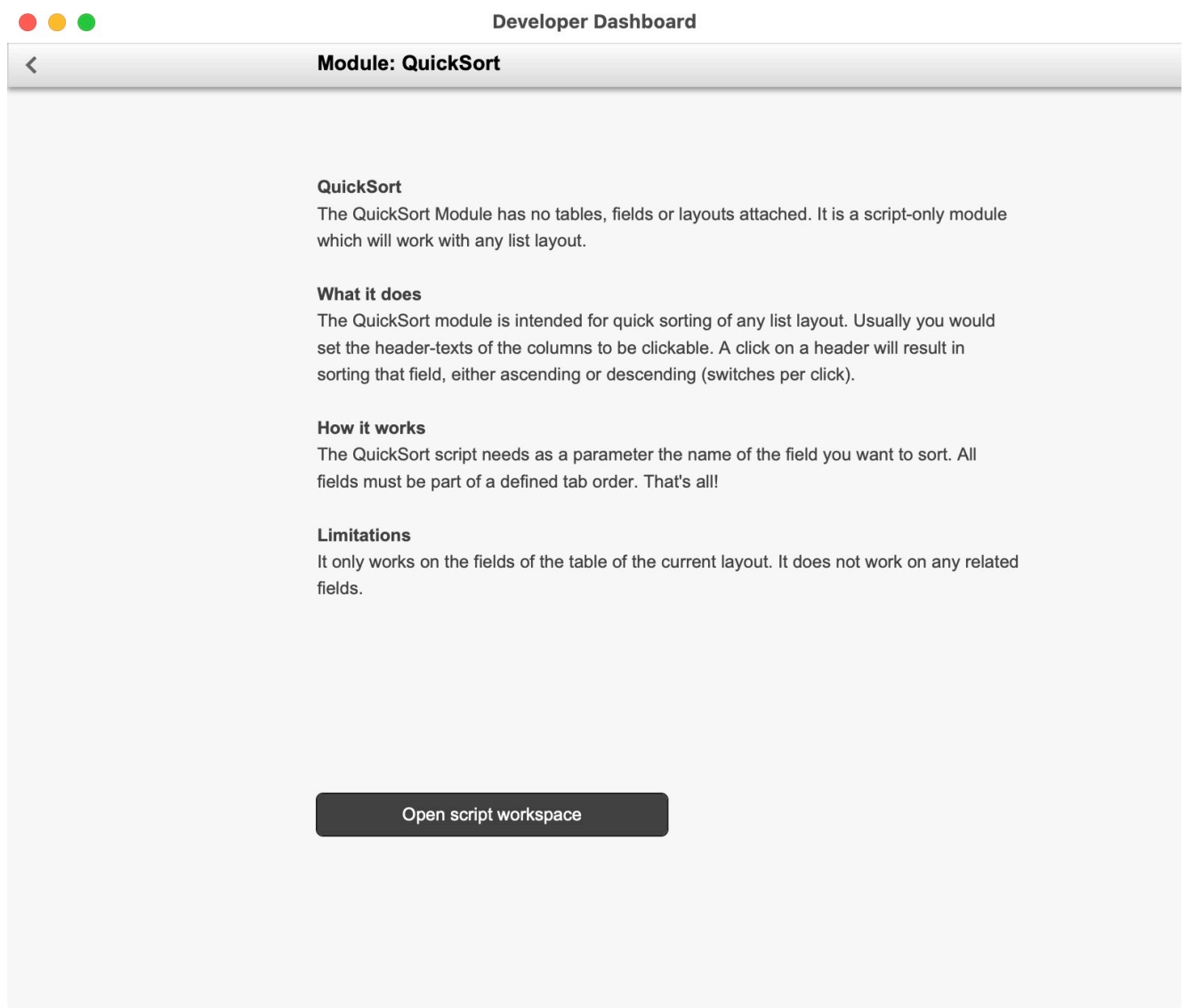
There are two variants available: One with text, another with icons. They work identically.



QuickSort module

The QuickSort module is a generic sorting function that consists of just one script. A description can be found on the module page.

The QuickSort module is particularly useful for quickly sorting the fields in a list.



The screenshot shows a web interface titled "Developer Dashboard" with a sub-header "Module: QuickSort". The page contains the following sections:

- QuickSort**
The QuickSort Module has no tables, fields or layouts attached. It is a script-only module which will work with any list layout.
- What it does**
The QuickSort module is intended for quick sorting of any list layout. Usually you would set the header-texts of the columns to be clickable. A click on a header will result in sorting that field, either ascending or descending (switches per click).
- How it works**
The QuickSort script needs as a parameter the name of the field you want to sort. All fields must be part of a defined tab order. That's all!
- Limitations**
It only works on the fields of the table of the current layout. It does not work on any related fields.

At the bottom of the page, there is a button labeled "Open script workspace".

MTL Multilingual Text Labels

Multilingual text labels for the user interface.

Developer Dashboard

<	DeepL	Translate	Languages	Popup	About	ⓘ						
+	<input checked="" type="checkbox"/> Search			Toolbox	Start translation in column			2				
Multilingual text labels			SOURCE LANGUAGE		TRANSLATIONS							
			EN		DE		IT		ES		JA	
			English		German		Italian		Spanish		Japanese	
Selection	↻	Label-ID	Internal Label	English	German	Italian	Spanish	Japanese				
translate	▶	11	Home	Home	Startseite	Accueil	Inicio	ホーム				
translate	▶	12	English	English	Englisch	Anglais	Inglés	イングリッシュ				
translate	▶	13	German	German	Deutsch	Allemand	Alemán	ジャーマン				
translate	▶	14	French	French	Französisch	Français	Francés	フレンチ				
translate	▶	15	Spanish	Spanish	Spanisch	Espagnol	Español	スパニッシュ				
translate	▶	16	Delete	Delete	Löschen	Supprimer	Borrar	削除				
translate	▶	17	New Record	New record	Neuer Rekord	Nouveau record	Nuevo récord	新記録				
translate	▶	18	Delete Record	Delete record	Datensatz löschen	Supprimer l'enregistrement	Borrar registro	レコードの削除				
translate	▶	19	Search	Search	Suche	Recherche	Busque en	検索				
translate	▶	20	Userlist	User list	Benutzerliste	Liste des utilisateurs	Lista de usuarios	ユーザーリスト				
translate	▶	21	Registrations	Registrations	Anmeldungen	Enregistrements	Registros	登録件数				
translate	▶	22	Software	Software	Software	Logiciel	Software	ソフトウェア				
translate	▶	23	Features	Features	Eigenschaften	Caractéristiques	Características	特徴				
translate	▶	24	Developer	Developer	Entwickler	Développeur	Desarrollador	デベロッパー				
translate	▶	26	Users	Users	Benutzer	Utilisateurs	Usuarios	ユーザー				
translate	▶	27	Company	Company	Unternehmen	Entreprise	Empresa	会社概要				
translate	▶	28	Street	Street	Straße	Rue	Calle	ストリート				
translate	▶	29	Zip	ZIP	ZIP	ZIP	ZIP	ジップ				
translate	▶	30	City	City	Stadt	Ville	Ciudad	都市				
			1	2	3	4	5					
461 Total												

FileMaker is not automatically multilingual. In any case, it requires its own development. This has advantages, but first things first:

Default method in FileMaker

FileMaker allows you to write text directly into any layout. In my experience, this is how most developers label. It's easy to start with. However, as the solution grows, managing the text becomes more challenging. (How can you quickly replace the word “Save” with “Save” throughout the application?) And as soon as you want to add a language, you have to start all over again and re-list all the texts in a multilingual system if you already have one.

FM Starter makes your FileMaker solution multilingual from the ground up. All text elements are managed centrally in a language table, which is already an advantage for a single language, but also allows multilingualism. In addition, the [HELP module](#) is also based on the MTL Multilingual Textlabel module. Help texts can therefore be conveniently managed with this multilingual module, whether for just one or several languages.

Dynamic approach

Using dynamically generated text labels is the right way to go. All your text labels are managed in a single table for this purpose. This makes management extremely easy. By referencing the same entry, you can use the same texts in multiple places while changing everything in the same table.

Access the module via the [developer dashboard](#) or the [Toolbox](#). The Toolbox window can be conveniently used alongside a layout in progress.

Tip

We strongly recommend that you always work with this module, even if you are only thinking of using a single language.

How does it work?

The text labeling module offers an easy way to keep all your texts in a single table. Learn how it works here:

Table-based

All text labels are saved in a single table. The labels themselves are saved in a single field. Each language has its own repetition of this field. It is easy to search for, add or edit an entry.

Global repetition field

When the text labels are initialized, the selected language (= selected repetition) is loaded into a single global field with 32,000 repetitions (currently maximum). Each text label in the table has its own number in a separate field. This number is used to store this translation in the corresponding repetition number with this value. If gaps occur or the numbering is changed, previous definitions remain valid.

Displaying the translation

New translations become visible after reloading the language or by selecting a different language. To reload, select the language from one of the language pop-up menus in the top bar.

How to integrate the text labels

Text labels can be integrated as fields or placeholders by calling this specific global repeating field, with the number of this repetition in square brackets. Something like this:

<<L::gT[35]>>

The table is called L, the field is gT and the repetition number is 35. This keeps the codes short.

Toolbox

The easiest way to find, create and copy text labels is to use the [Toolbox](#). This allows you to simply copy and paste merge fields like the ones shown above.

Numbering redefined

The MTL Multilingual Text Labels module is used both for the basic development of FM Starter by Kursiv Software and for your own development. To avoid having to update your solution one day with new functions and text labels from the FM Starter development, you can separate your own label numbering from the FM Starter numbering. By separating them, you always avoid overlaps.

How this is done: The numbering is a simple serial number defined in the TranslationID field. Open the field description in the database window (File > Manage > Database) and set the value to a higher number (e.g. 5000) to avoid future overlaps. The labels you create will then start at 5000.

Languages

FM Starter is currently supplied with 6 standard languages that are used for the user interface:

1. English
2. German
3. French
4. Spanish
5. Japanese
6. Dutch

This selection can be used, changed or extended. This is entirely up to you.
Any language can be used.

Does FM Starter translate automatically?

FM Starter has a built-in module for automatic translations. This allows you to use the DeepL translation. See [here](#).

Editing languages

You can define the number of languages, as well as the language selection itself. The initial situation of FM Starter is just one example.

First access

Open the Multilingual Text Labels module from the left-hand navigation in the developer dashboard.

<

Multilingual text labels

+

Search

Toolbox




Global Repetition Field

Label-ID		Label	English	German	French
	11	 Home	Home	Home	Accueil
	12	 English	English	Englisch	Anglais
	13	 German	German	Deutsch	Allemand


- The label ID is assigned automatically when you select a text in the pop-up menu.
- The label is a label for the developer (you can create longer text snippets for which you choose a short “name”).
- There is a single field for the language names, with as many repetitions as there are languages.
- There is a single field for the text labels, with as many repetitions as there are languages.
- Search for any value using the search field.
- Open the toolbox with the “Toolbox” button.

Defining languages

The pop-up for language selection is created separately. To customize the naming of the individual languages or to add, remove or extend this information, click on “Define Languages”:

	i English	Define Languages >
Clear active translation 		
	Spanish	Active Translation
	Inicio	Home
	Inglese	English
	Alemán	German
	Francese	French
	Español	Spanish
	Borrar	Delete

A new window opens in which you can edit the language list. A value list for the pop-up menu with the languages is created from this list.

<
+

☒
Search

English

English

Languages

Multilingual valuelists use labels which must be translated with the regular module.
At a language change, the value lists must be updated.

Code	Order	Label	Translation-ID	Translated
EN	1	English	12	English
DE	2	German	13	German
FR	3	French	14	French
ES	4	Spanish	15	Spanish

If you add new languages, make sure that you also create a new data record in the translations and link this to the correct translation ID. This ensures that the name of the language is displayed correctly.

Updating the languages

Each change to the labeling is only activated after a language has been reloaded. To do this, simply select one of the languages from the language pop-up again. This reloads the languages.

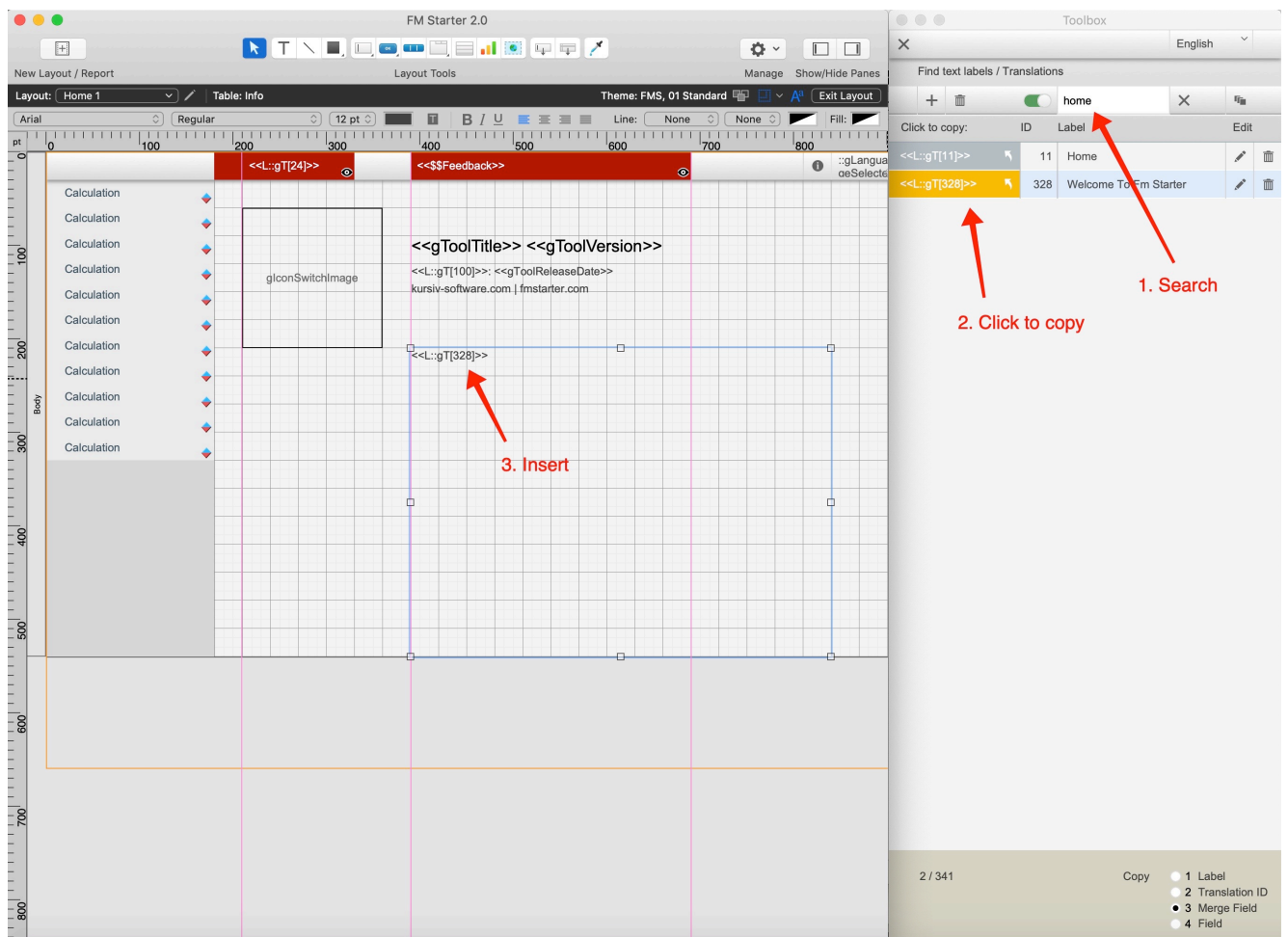
This value list is updated each time a language is reloaded. How this works is explained in the section Editing value lists.

Working with the Toolbox

While you can work with the module's large table, you will most likely use the toolbox most often. See the section on the Toolbox window.

Find and copy any text labels

Open the Toolbox window. The main window can now switch to layout mode and you can easily search for, create or copy any text label from the Toolbox window.



1. Search for the text name you are looking for.
2. Click on the button to copy the placeholder.
3. Paste the code into any text object.

Creating a new text label

To find out how to create new text labels, read the [Toolbox](#) section.

Translating value lists

Value lists cannot normally be translated. FM Starter still supports translated value lists, but it requires some scripts for this.

Tip

There is an article and video on this topic on fmstarter.com.

What you should know

- Each value list must be created with its own table.
- Additional fields form the translation logic and save the translated label.
- When selecting or reloading a language, the value lists must be updated. All value lists must therefore be listed in a script that updates the information when the language is changed.

How does it work?

Value lists that need to be translated are defined in a separate table. Several fields are listed in this table. For example, there is a translation ID that refers to the language table. The correct reference number can be conveniently selected from a pop-up. There is also a calculated text field based on this translation ID, which is updated when the language is initialized or changed and calculates the label in the current language. The value list refers to the translated field.

Examples

Several value lists are already used when you try FM Starter for the first time. The list of languages is one such list:

<

+

🗑️

🔴

Search

Languages

Multilingual valuelists use labels which must be translated with the regular module. At a language change, the value lists must be updated.

English

1 English

EN

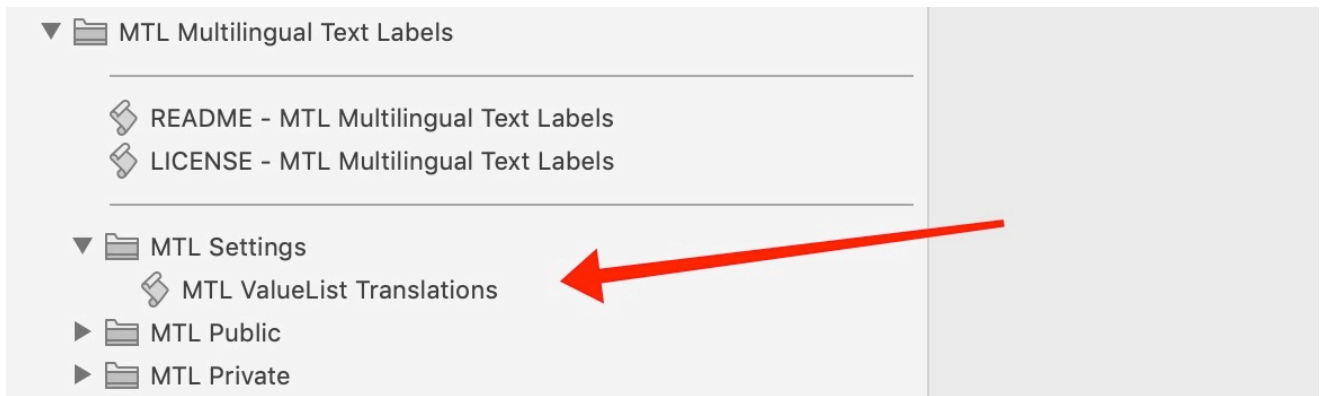
▼

	Code	Order	Label	Translation-ID	Translated
	EN	1	English	12	English
	DE	2	German	13	German
	FR	3	French	14	French
	ES	4	Spanish	15	Spanish

- The code is specific to this list, you can create a simple short code for a drop-down menu.
- The order refers to the sort order in the drop-down or pop-up menu.
- The label is the internal reference.
- The translation ID comes from a popup of all translations and enables the different translations of this label.
- The Translated field displays the text labels with the selected language.

Including value lists for translation

Value lists are not updated automatically. Each value list must be updated when a language is changed or reloaded. This is done by a script in the MTL Settings folder:



The MTL ValueList Translations script must contain and update all tables in the value list. Check the script to see how it is done. It is simple! When you create new value lists, add them to this script just as other value lists have been added.

Creating multilingual software solutions is no longer a hassle because FM Starter provides automatic translation using [DeepL.com](https://www.deepl.com). All you need is an API account at DeepL.

Translate texts automatically with DeepL

You can use DeepL to automate translations for multilingual text labels. This function has been integrated into version 2.5 of FM Starter. It is fully integrated into the Multilingual text labels module.

What is DeepL?

DeepL is a German product and company that creates translations with the help of artificial intelligence. Its automated translations are probably the best available today and significantly outperform Google Translate, Yandex, Bing and other services.

How good is it?

DeepL is really good. But it is not perfect, as language is quite complex. Always check the translations. The automatic translations are a great help and speed up the translation process enormously, but it is always advisable to check before releasing new languages for the user interface.

Real life: I speak several languages fluently, but I don't speak Spanish or Japanese, which are both interface languages in FM Starter. While I can decipher some Spanish, unfortunately I have no idea about Japanese. Japanese is 100% DeepL and not verified. I apologize to native speakers for any errors they find.

Create a DeepL account

For this solution to work for you, you need to set up an account at DeepL.com. For software developers, in most cases it is sufficient to open a free account with which you will receive a certain number of translations free of charge each month.

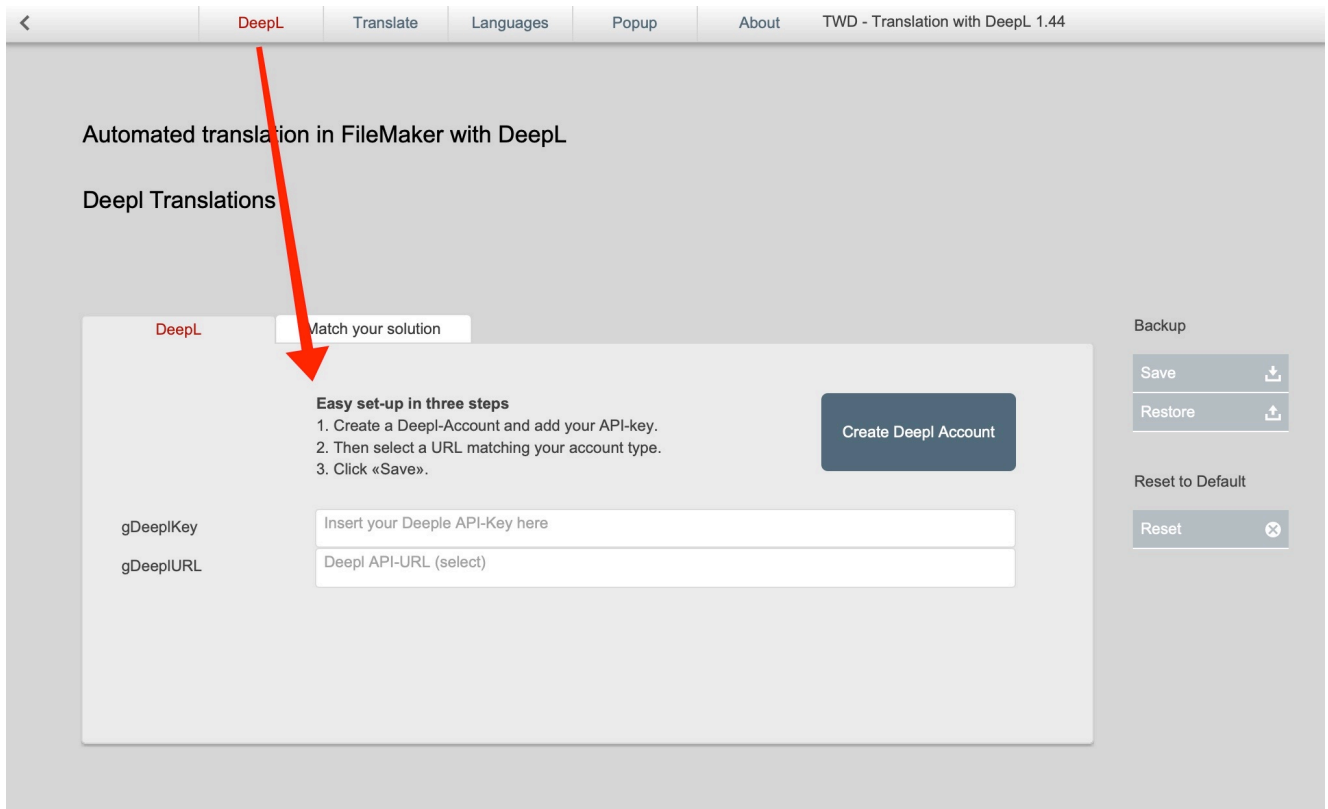
API-Key

As soon as you have created your DeepL account, you will receive your API key in your account.

Three simple steps

Only three simple steps are required to start translating:

1. Copy your API key from your DeepL account and paste it into the corresponding field in FM Starter. (Developer Dashboard > Multilingual text labels > DeepL)
2. Select the URL for the type of account you have created.
3. Save.



Ready for translation

There is nothing else to do. You are ready to go with the basic settings. Switch to the “Translate” section in the top menu.

DeepL

Translate

Languages

Popup

About

Make sure that the titles are set correctly for each language. Any change immediately triggers a translation for this column, but you can deselect this option first. Take a look at how easy it is to set the titles of the available languages in DeepL.

Three translation options

There are three options available for translations:

1. **Line:** Click on “Translate” to the left of each row. This will translate all columns (field repetitions) for this data set, starting with the column specified in the top menu (“Translations in column start...”)
2. **Column:** If you select or reselect a language title, a translation is triggered for this column. If you do not want this, simply cancel the dialog that opens. If you confirm, only this column will be retranslated, using the first column or repetition as the source text.
3. **Selection:** The green button on the left above the data records offers the translation of a selection of data records. You can choose between all data records (starting with the first data record in the selection) or start with the currently active data record. With this option, all languages for a data record

are translated, starting with the column specified above and then running through the remaining data records.

Language settings

The language settings for DeepL translations are shown in a separate list. If DeepL provides additional languages, simply expand this list to include them in the translations.

< DeepL Translate Languages Popup About				
All languages supported by DeepL				
You can add more languages if they become available. Check DeepL.com!				
code label description				
	BG	Bulgarian		
	CS	Czech		
	DA	Danish		
	DE	German		
	EL	Greek		
	EN	English		
	ET	Estonian		
	FI	Finnish		
	FR	French		
	HU	Hungarian		
	IT	Italian		

Popup

The pop-up menu takes you to the page that defines the pop-up. You list the languages you want to include in your menu and select the translation for this language that you have found or made available in the translation. This popup is converted into a multilingual value list as implemented in FM Starter since version 2.

DeepL
Translate
Languages
Popup
About

Language Pop-up

This popup is for your frontend. It must be set-up for switching languages. Copy-and-paste one of the popups at the right, as there already are script triggers included. Here you link to translated labels. If you work on new languages, you can do so without updating this popup. Only after finishing the language, you can include the new language here, to make it available.

English

1 English

EN

1

The **DeepL** translation works with language codes, while the **language switch** is based on numbers.

Repetition number of translation field

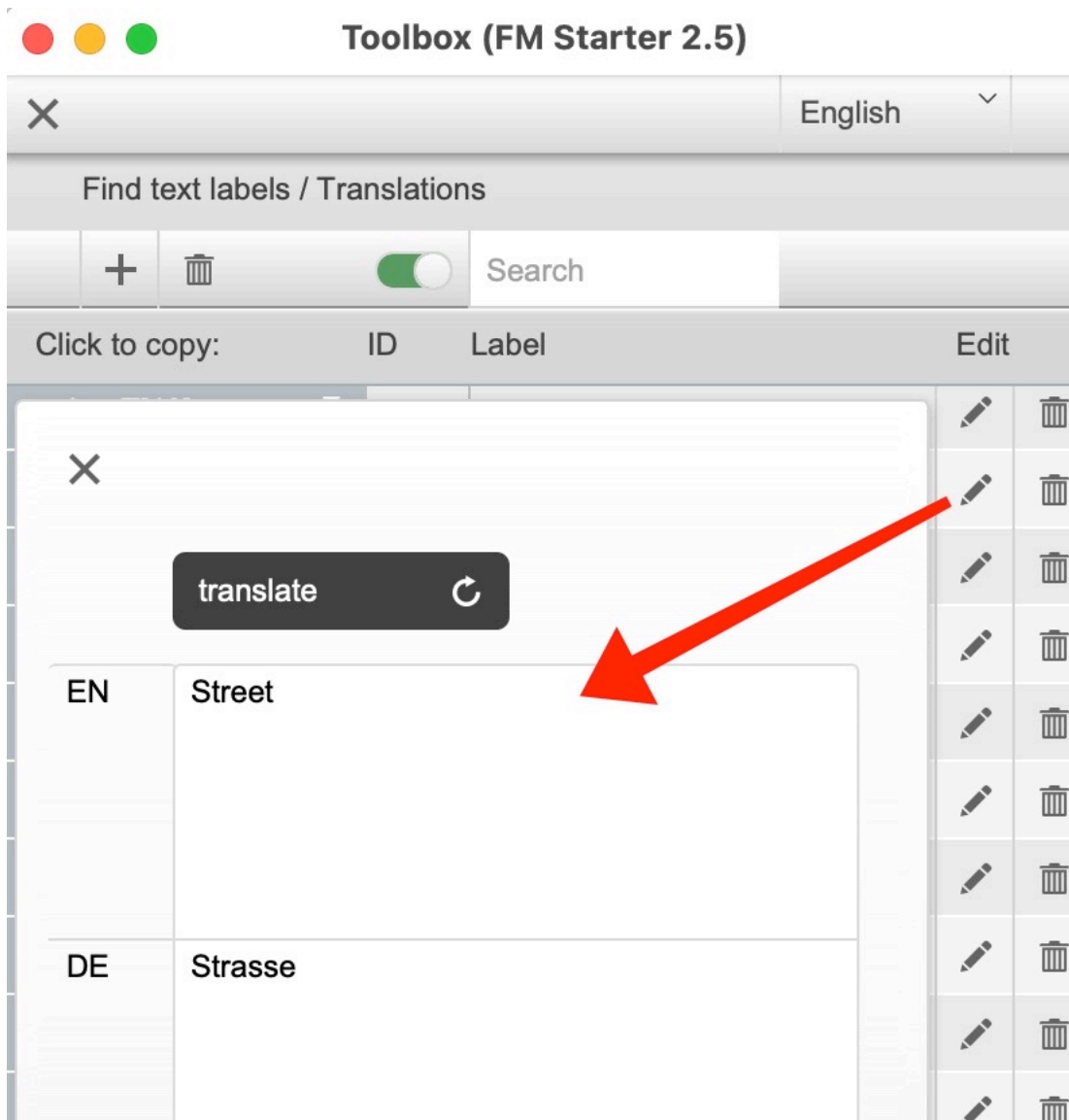
+	Code	Order	Label	Translation-ID	Translated
	EN	1	English	12	English
	DE	2	German	13	German
	FR	3	French	14	French
	ES	4	Spanish	15	Spanish
	JA	5	Japanese	432	Japanese

The DeepL language list is a list of all languages available on DeepL.com. It works with a language code as provided by DeepL. The popup is different. This is a list of interface languages for FM Starter only. It works with numbers (“order”) that must match the repeating fields of the title in the Translate section.

The popup is a language selector in the user interface. Script triggers are activated when the value of the popup is changed. It switches between the available languages and the selected language is reloaded.

Toolbox

The automatic translation with DeepL is also integrated into the toolbox window. Just as in the main window, you must fill in the first repetition field as the source for the translation. Then click on the button and the remaining languages will be translated.

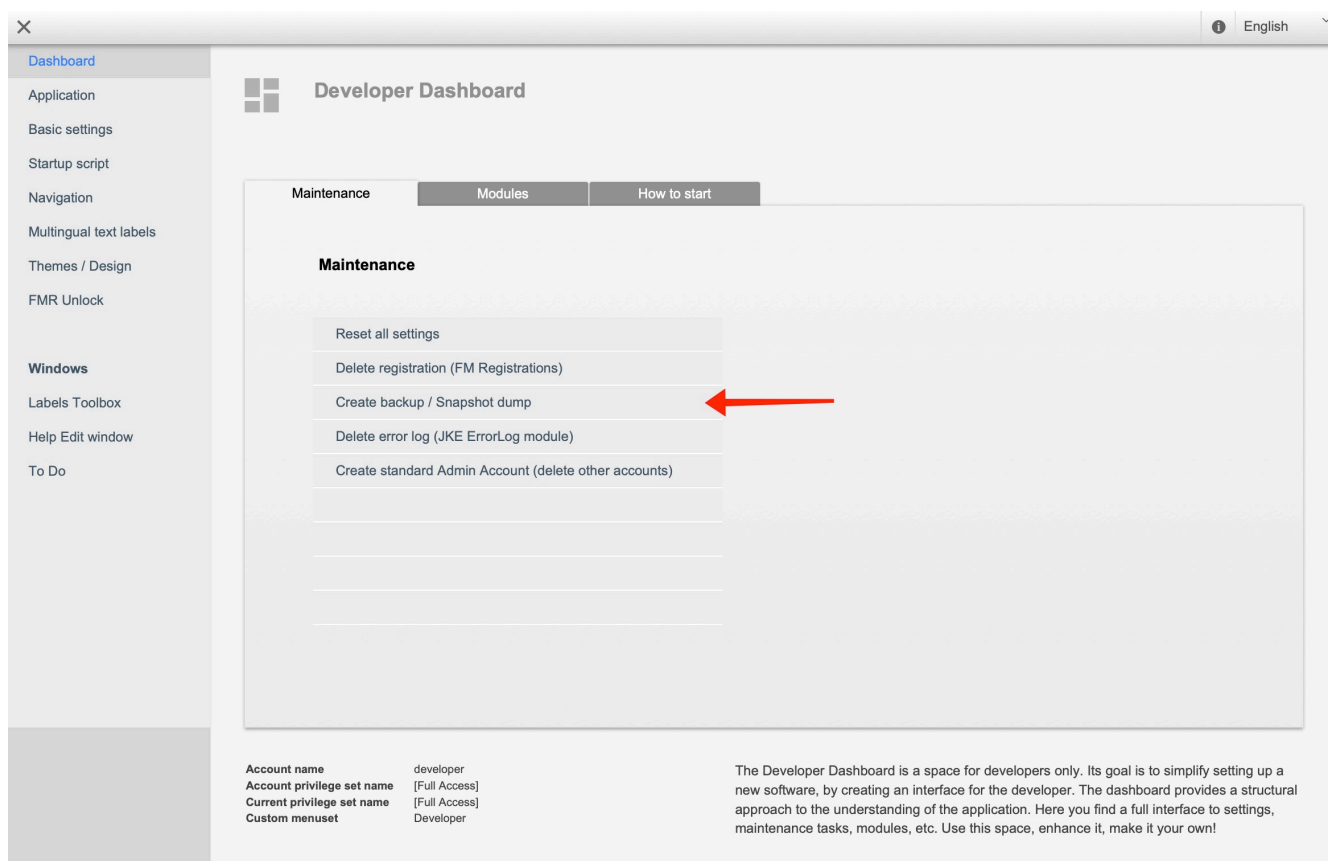


SBU Simple Backup

The Simple Backup module allows you to quickly create a snapshot of your file(s) if you are working locally. If you are working with a hosted version of your solution, you cannot use this module. In this case, please use the backup functions of FileMaker Server.

Snapshots for the developer

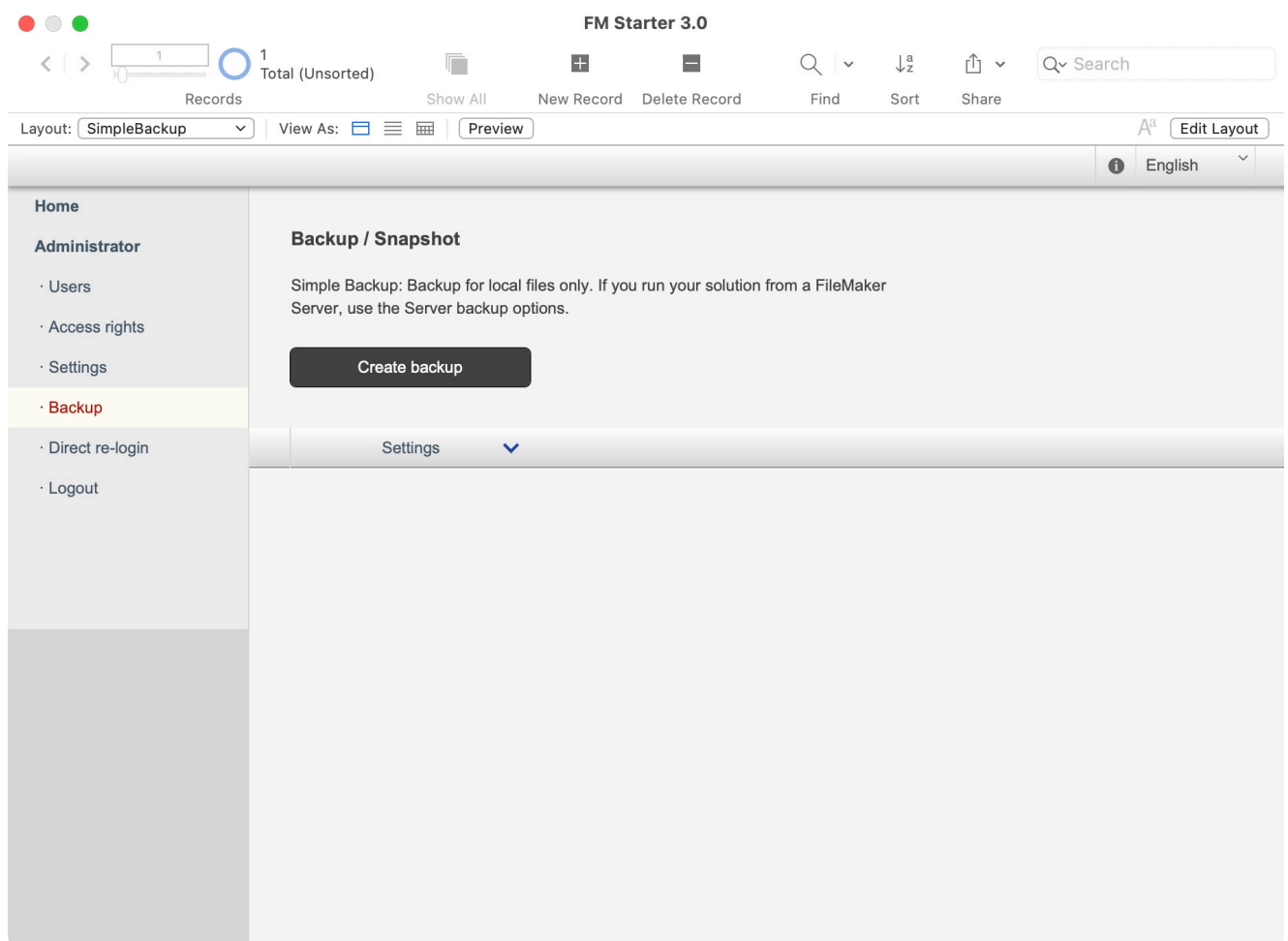
In FM Starter, the Simple Backup module is part of the maintenance functions on the dashboard. From here, the developer can create a new snapshot at any time at the touch of a button.



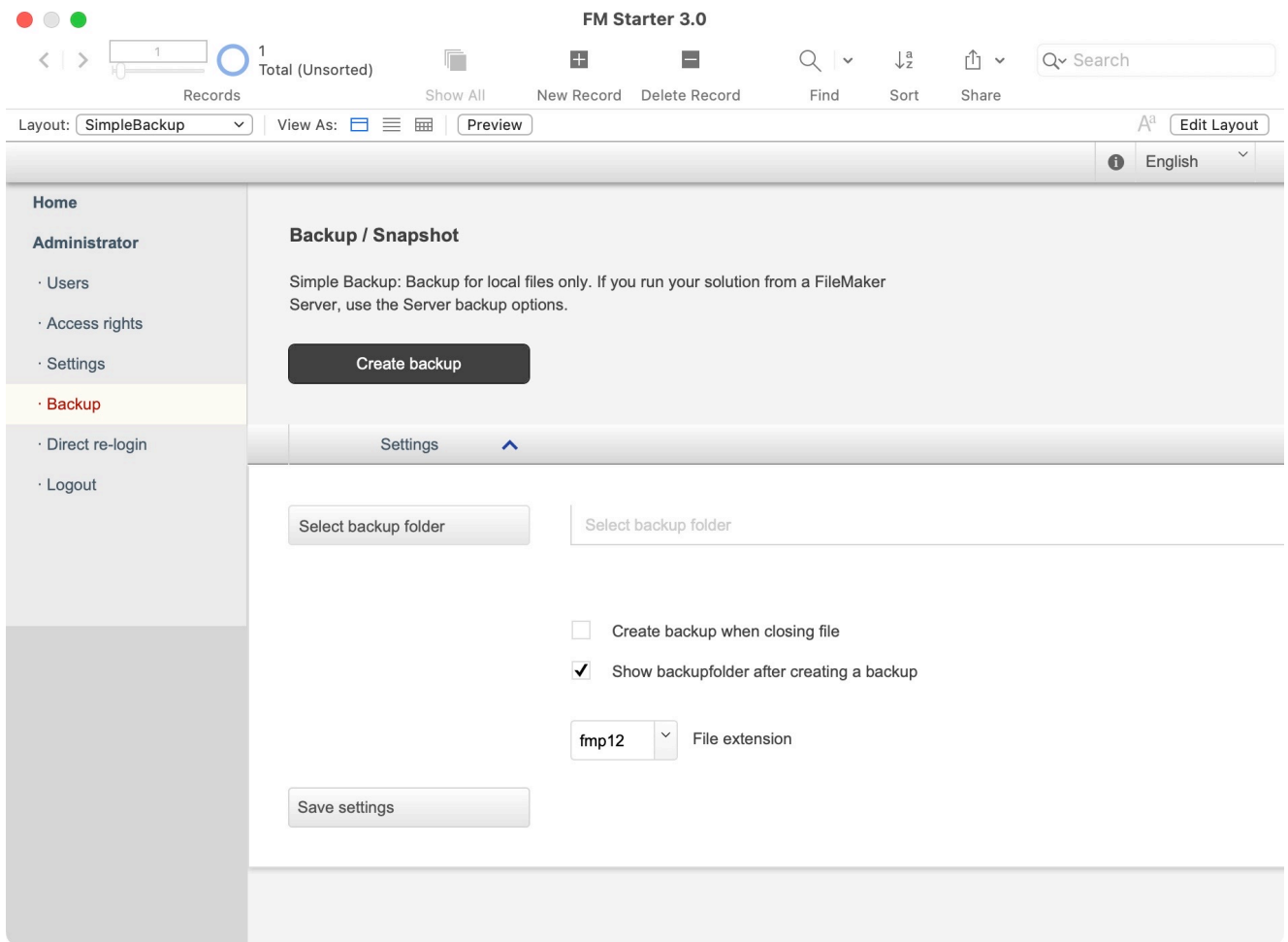
Backups for the administrator

Another implementation is for the administrator, who can now easily create a backup as long as the file is hosted locally.

Making this module available to the administrator is only a suggestion. You decide what suits your solution best.



Open the settings for more information:



To remove this page from the administrator settings, simply remove the link to this page in the navigation module.

Backups for the developer

The module is also linked via the developer dashboard > Modules.

Further information can be found directly in the module's scripts.

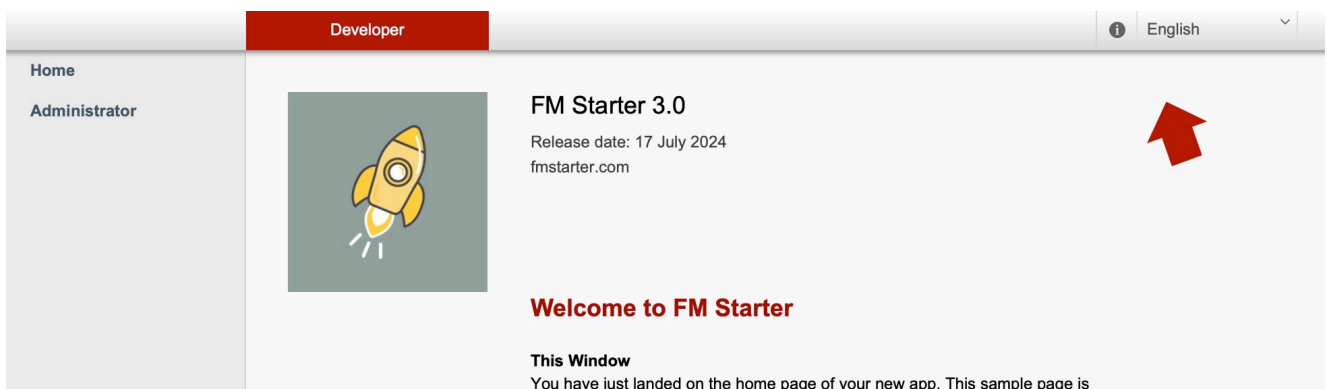
HELP module

The HELP module has already been discussed. It is a small module that can be used to display a help text anywhere. This is perfect for small help texts in any layout.

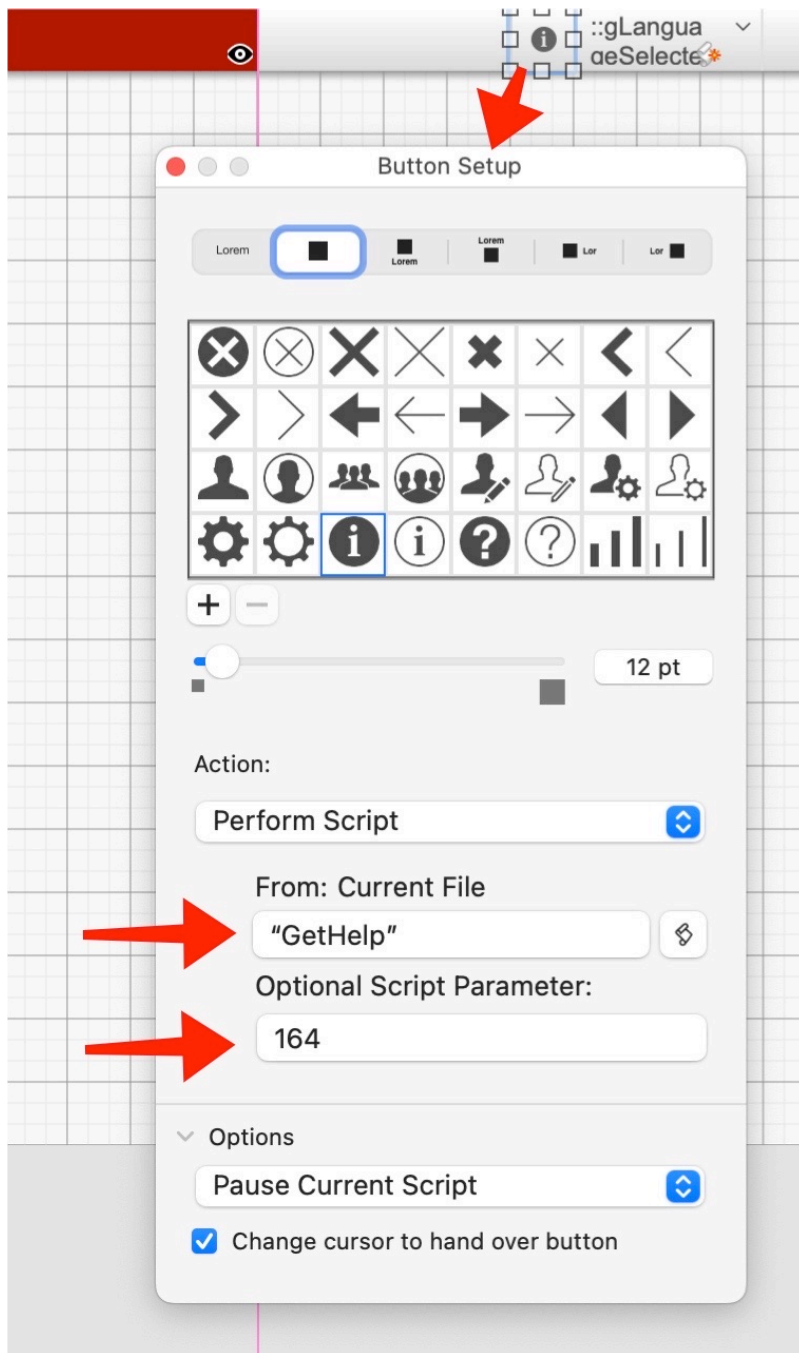
The HELP module is new in version 3.0. It is the same module that was already used in FrankCRM. The module is simpler and more flexible than its predecessor and automatically adopts the language of the user interface. Texts are created directly in the MTL module for multilingual text labels.

HELP button

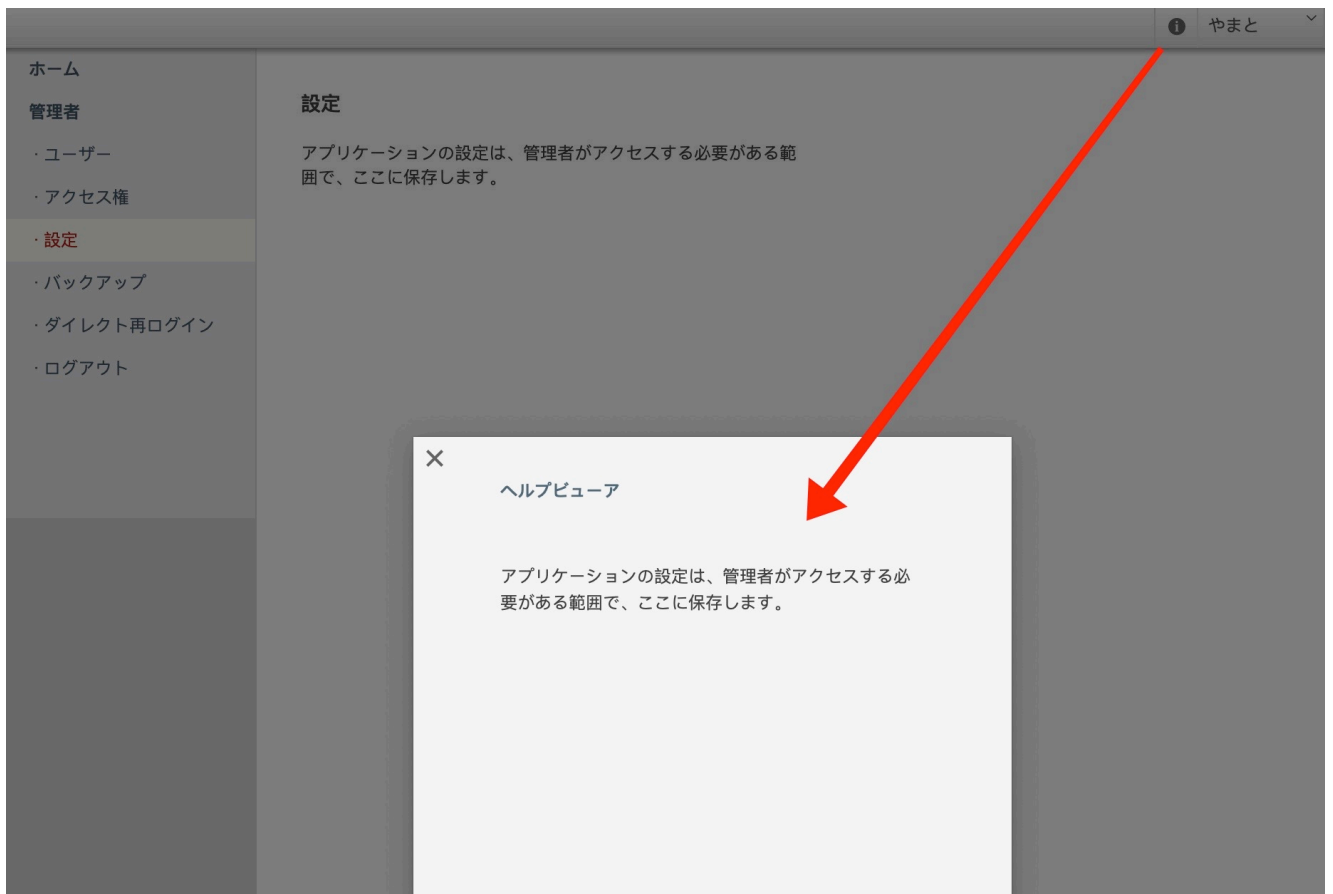
The information icon is used to call up the HELP module. It is very simple: The button refers to the HELP script and the desired number of the text label is retrieved from the toolbox window as a parameter.



Double-click on the button in the layout. Then select the script “GetHelp” and copy the ID of the desired text block from the toolbox as a parameter.



That is all! The help text now appears in a small card window and is automatically displayed in the language of the user interface.



UAC User Accounts

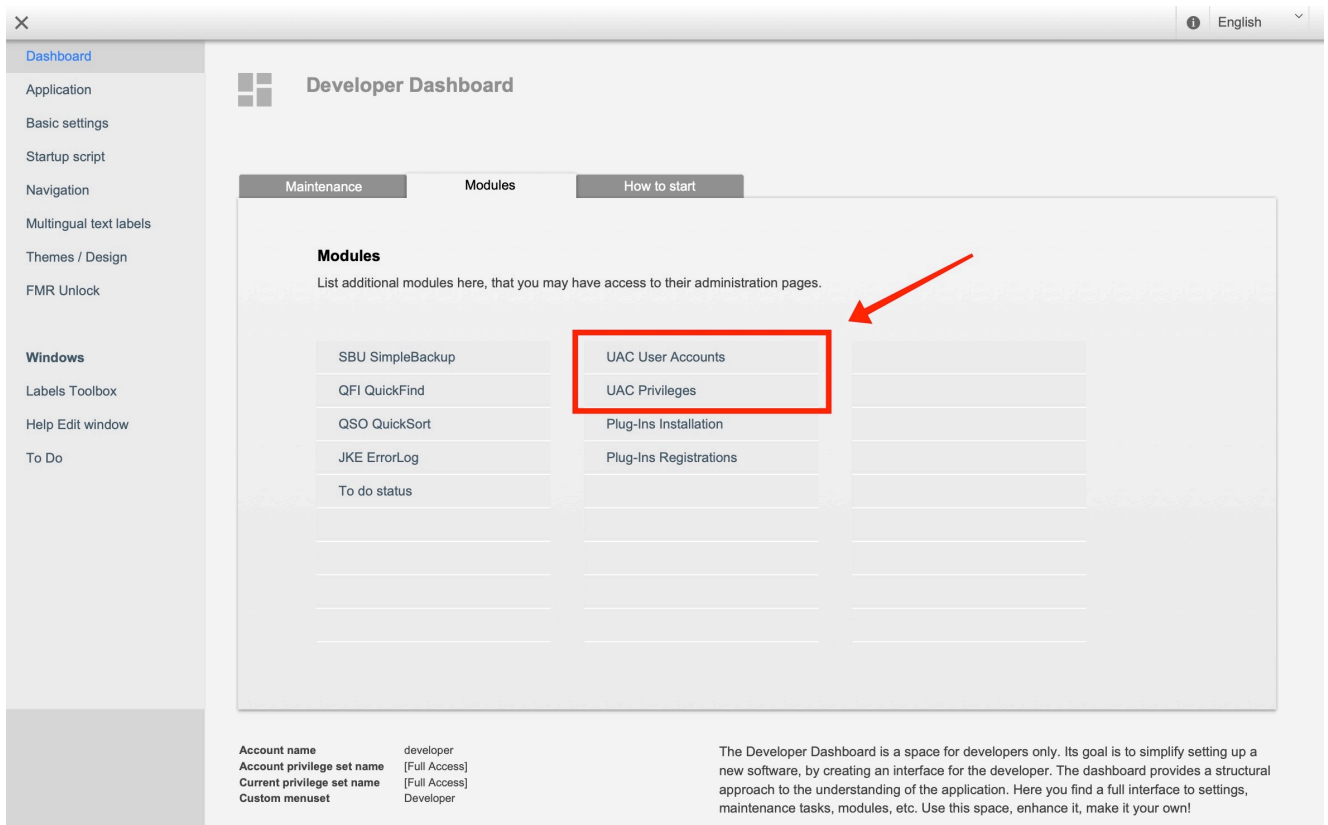
User account management is at the heart of any multi-user application. Give each user their own access to your solution and maintain a simple, configurable and clear list of users.

Caveats and advice

- Please note that this module does not regulate security, but merely provides an interface between FileMaker Security and the user (usually the administrator).
- Security must primarily be configured by the developer in the FileMaker security settings.
- The configurations must then be listed in the settings scripts of the UAC User Accounts module.
- The module then helps to make the security settings transparent and editable for the administrator.

Access to the module

The module has pages for both the developer and the administrator. Access for the developer is via the developer dashboard > Modules.



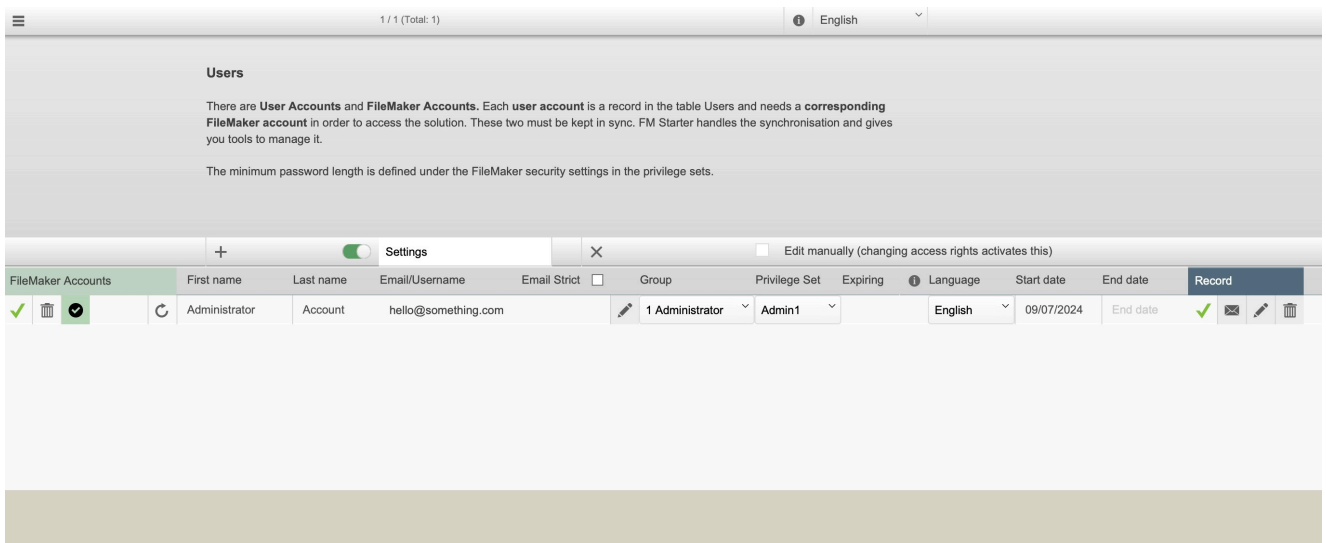
The module consists of three tables: The user list itself, a list of privilege sets and a list of user groups. This list helps you to create a FileMaker account for each user.

- UAC user accounts - corresponds to the FileMaker account
- UAC permissions - corresponds to the FileMaker permissions
- UAC user groups - additional groups for organizing your user database

Access for the administrator is explained in the next section.

User accounts

User accounts are created in a user table. From here, a FileMaker account is generated with the settings from the user entry. The user accounts are listed on a separate layout that is visible to the administrator of the solution. The administrator can create and define accounts create and define accounts, then send access to the users.



The screenshot shows the 'Users' layout in FileMaker. At the top, there's a header bar with a menu icon, '1 / 1 (Total: 1)', and a language dropdown set to 'English'. Below the header, a section titled 'Users' contains explanatory text: 'There are User Accounts and FileMaker Accounts. Each user account is a record in the table Users and needs a corresponding FileMaker account in order to access the solution. These two must be kept in sync. FM Starter handles the synchronisation and gives you tools to manage it.' and 'The minimum password length is defined under the FileMaker security settings in the privilege sets.' Below this text is a toolbar with a '+' icon, a 'Settings' button, and an 'Edit manually (changing access rights activates this)' checkbox. The main area displays a table with the following columns: FileMaker Accounts, First name, Last name, Email/Username, Email Strict, Group, Privilege Set, Expiring, Language, Start date, End date, and Record. A single record is visible with the following data: FileMaker Accounts (checked), First name (Administrator), Last name (Account), Email/Username (hello@something.com), Email Strict (unchecked), Group (1 Administrator), Privilege Set (Admin1), Expiring (empty), Language (English), Start date (09/07/2024), End date (End date), and Record (checked).

FileMaker Accounts	First name	Last name	Email/Username	Email Strict	Group	Privilege Set	Expiring	Language	Start date	End date	Record
✓	Administrator	Account	hello@something.com	<input type="checkbox"/>	1 Administrator	Admin1		English	09/07/2024	End date	✓

The user table contains a list of all user accounts that have been created in this module. Other accounts that were created directly in FileMaker are not displayed here. This also includes user accounts that are deliberately not created via this module so that they do not accidentally end up in the hands of users.

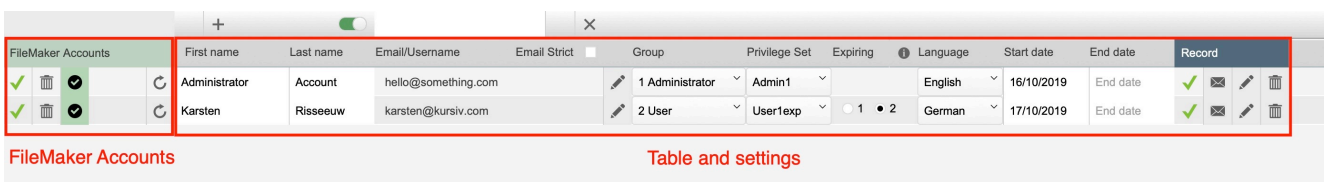
Important: For this reason there are two registrations:

1. Login for users in this table (login module)
2. Login for developers (direct FileMaker login)

On the far left you will see some functions that relate to the underlying FileMaker account. Here you can create, delete and activate a user's FileMaker account.

User accounts are managed in a separate table. This makes it easy to integrate customizations such as the language for this user, a start and/or end date for accessing the software, a registration email and more.

Each row is a data record and corresponds to a user. The largest part of each row is taken up by the user table information. Here you list information about the user, send an e-mail, create a new login, assign an authorization to a user group, assign an authorization (access rights), etc.



	First name	Last name	Email/Username	Email Strict	Group	Privilege Set	Expiring	Language	Start date	End date	Record
✓ ✕ ↺	Administrator	Account	hello@something.com		1 Administrator	Admin1		English	16/10/2019	End date	✓ ✕ ↺
✓ ✕ ↺	Karsten	Risseeuw	karsten@kursiv.com		2 User	User1exp	1 • 2	German	17/10/2019	End date	✓ ✕ ↺

FileMaker Accounts Table and settings

Tip 1: For more information about this screen, see the help button in the layout.

Tip 2: Check the settings script of this module for more information.

This module is merely a **bridge** to the FileMaker accounts. Always bear that in mind. While the FileMaker accounts are very cumbersome to handle (as they are hidden on different screens), this user table makes it easier to handle. Everything is based on a central table and is controlled by scripts. Therefore, the module can easily be extended with additional functionalities.

Expiring passwords for better security

FileMaker Pro offers several options for creating secure accounts. The UAC UserAccounts module is a script-based solution to make these FileMaker accounts more user-friendly and simpler. However, the module does not include all the options that FileMaker offers. While the module is a first step for a variety of scenarios, you may want to expand the functionality to meet your specific needs.

Expiring passwords

Whenever you create a FileMaker account with an expiring password, the user must renew the password at the next login. This is now possible with the UAC UserAccount module. If you select this option, the new password is no longer stored in the user table and therefore cannot be used to identify the account. The user's entry is still correct in the table and the user name is also valid, but if the password expires and is changed, the password in the user table is not updated. The identification must be transferred to FileMaker itself. This type of password makes the solution GDPR-compliant.

Any password can be set to expire so that the user must create a new password the next time they log in. This is not a functionality of FM Starter, but a functionality of FileMaker itself. How this is handled can be defined in the script:

Scripts > UAC UserAccounts > UAC Settings > UAC UserAccounts

FM Starter has set up a user account for expiring passwords. You can extend this if required. See under > [Authorizations](#).

A new login procedure

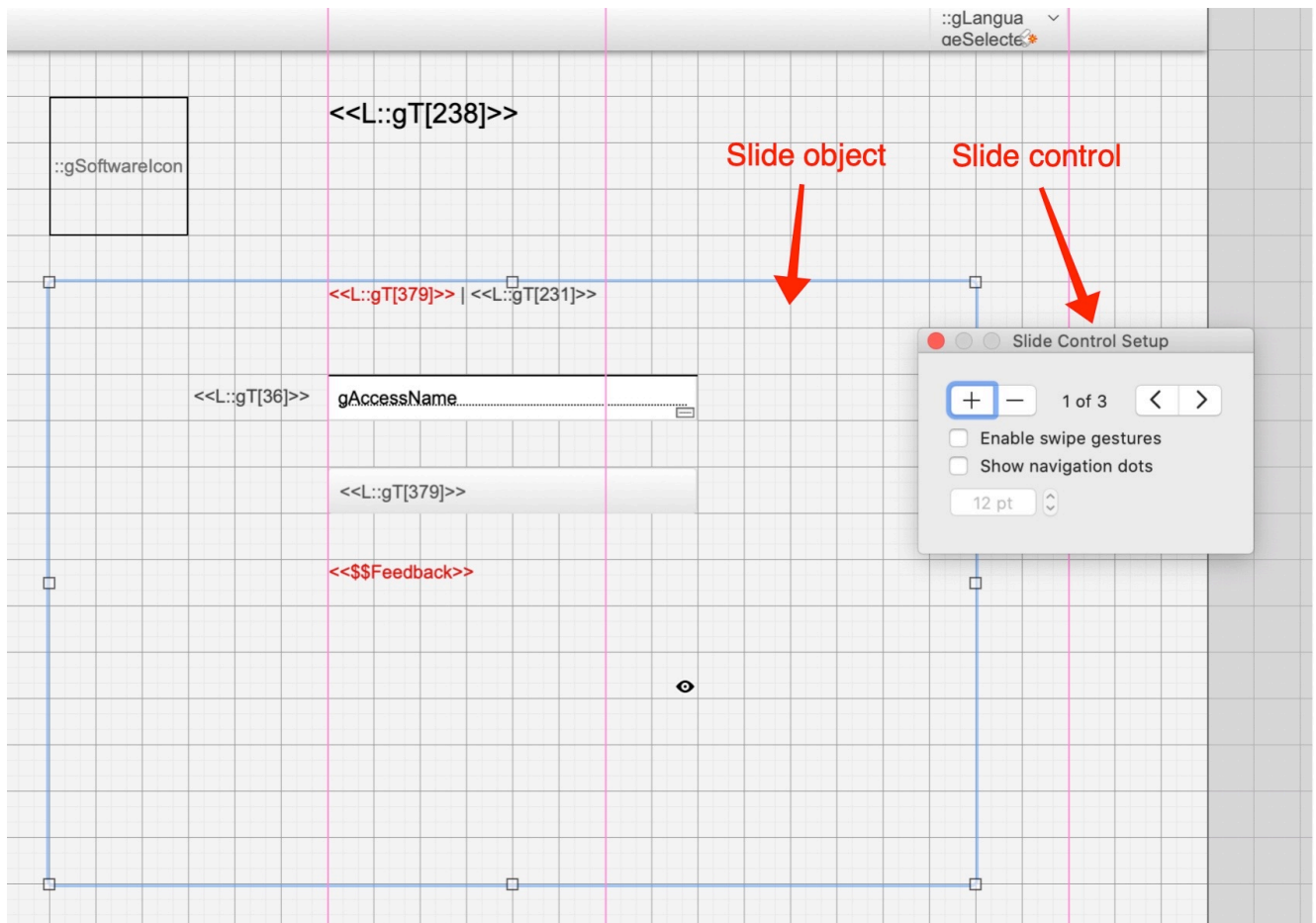
We have introduced multi-stage verification, as is already used on many websites.

In a first step, the user name/email is requested and the module searches whether an account with this user name is active. If the user is found and the type of account is recognized, the password field is displayed. After entering the password, the login is checked and access is granted as usual.

The new login procedure required a redesign of the login page. The page now has a transparent **slide object with 3 slides**:

1. Slide 1: User name/email
2. Slide 2: Password
3. Slide 3: Fields for a new password if the account password is to expire and the user has to choose a new password.

The slide object is transparent. To see it, open the login page, select a menu set with full access (> Tools > Custom menus), switch to layout mode and double-click next to a field. The slide menu will appear to help you navigate. During login, the navigation is taken over by the scripts.



The new procedures are as follows at the start:

1. Slide 1 > User name / email
 - Non-expiring password found > go to slide 2
 - Expiring password found > go to slide 3
2. Slide 2 > Enter password
 - If the password is correct, access is granted.
3. Slide 3 > Enter new password
 - A new account password is created.
 - The expiry option is deactivated.
 - The user is now redirected to > Slide 2
 - The new password is required to enable access.

If no user name/email is found, an error message is displayed.

Standard administration account

This module requires at least one active account, otherwise no one will be able to log in. By default, a single admin account is created, which you can access via the normal login page:

Name: Admin

Password: Admin

It is strongly recommended that you start creating your own logins instead and do not use this facility beyond the initial steps with FM Starter.

When all records are deleted, a new default admin account is automatically created.

The default admin account is created by a small script that can be extended with any details you deem important:

Scripts > UAC UserAccounts > UAC Settings > UAC Admin Default

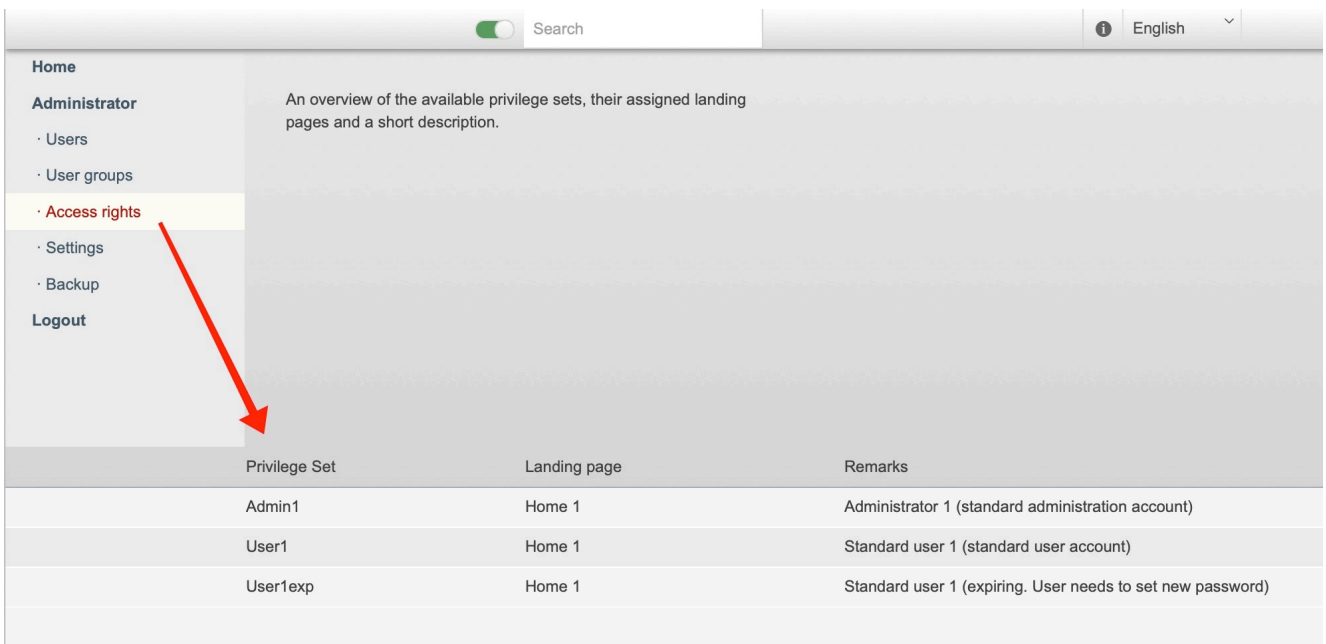
Permissions / Rights

When creating a new account (data record), the administrator can assign access rights for this specific account. These access rights are FileMaker authorizations.

- Privileges are a FileMaker security setting. Create or edit privilege sets in the FileMaker security settings.
- FM Starter requires the exact names of the authorizations that are listed in FM Starter, in the table of authorizations (User Rights)
- The administrator assigns an authorization to each new data record.
- A new FileMaker account can only be created with a selected privilege set.

Two views

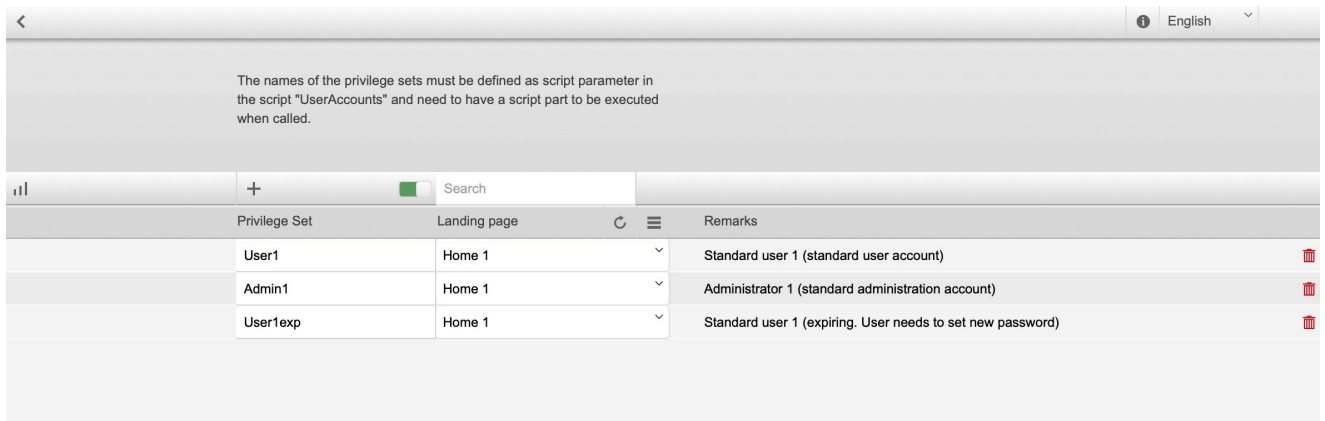
1st view for the administrator (not adjustable, for information only):



The screenshot shows the FileMaker Starter interface. On the left is a sidebar with a menu. The 'Access rights' item is highlighted in yellow, and a red arrow points from it to a table. The table has three columns: 'Privilege Set', 'Landing page', and 'Remarks'. It lists three privilege sets: 'Admin1', 'User1', and 'User1exp', all assigned to 'Home 1'. The top of the interface includes a search bar and a language dropdown set to 'English'.

Privilege Set	Landing page	Remarks
Admin1	Home 1	Administrator 1 (standard administration account)
User1	Home 1	Standard user 1 (standard user account)
User1exp	Home 1	Standard user 1 (expiring. User needs to set new password)

Second view for the developer via the dashboard (adjustable):



The Permission field must have exactly the same name as defined in the FileMaker security settings. The target page can be defined for each of these account types. This is evaluated during login and the user is directed to the group's own target page.

When an account is created, the privilege set assigned to the script is used. Another option in this script step is to specify whether the account password should be renewed at the next login. An authorization has been defined in this way and has its own script part to define it. You can find out how to do this in the “UAC UserAccounts” script.

> Scripts > Modules > UAC UserAccounts > UAC Settings > UAC UserAccounts.

GDPR compliance

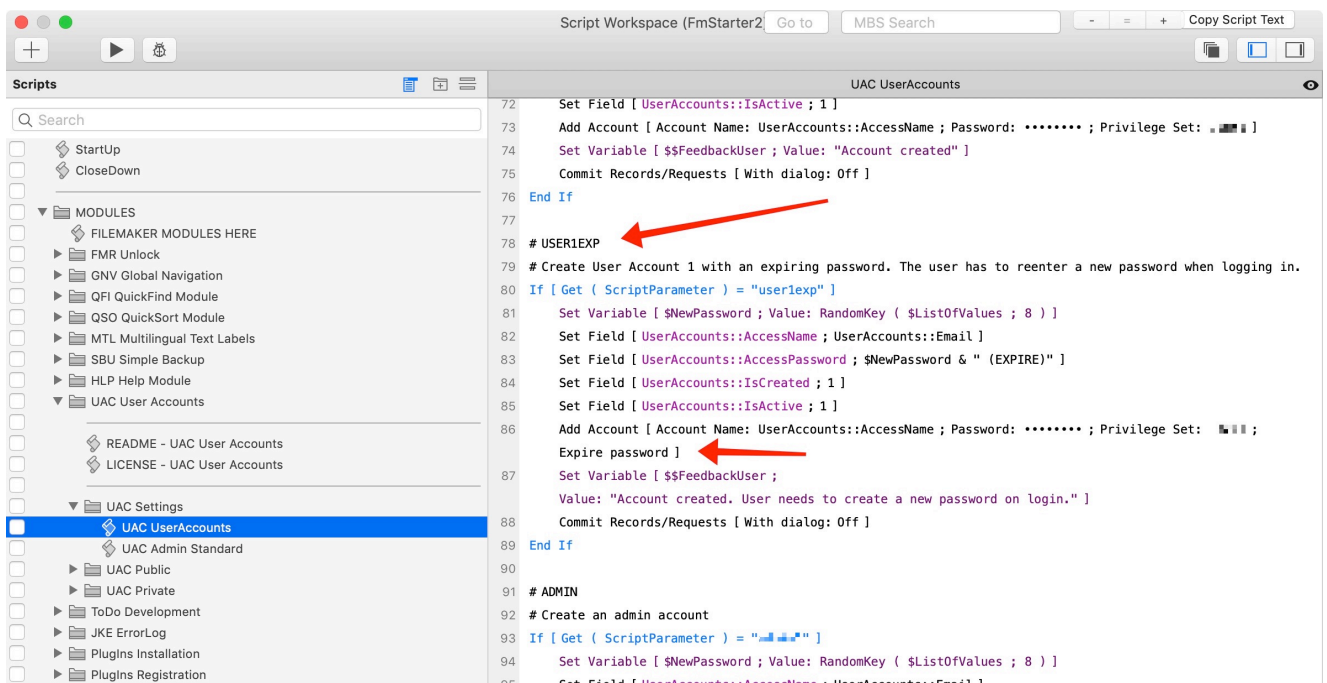
Since May 2018, when the GDPR rules on data protection were activated in the European Union, there has been a lot of talk about the requirements for databases. The basic rule is that no passwords should be stored in databases.

This situation is easy to achieve with FileMaker, as long as we make a simple decision to create user accounts where the user has to renew the password at the first login. Once the user changes the password, it is completely managed by FileMaker and is no longer accessible to any administrator.

To set up such an account, select the correct privilege set. FM Starter contains several privilege sets, but one of them is named

- User1exp

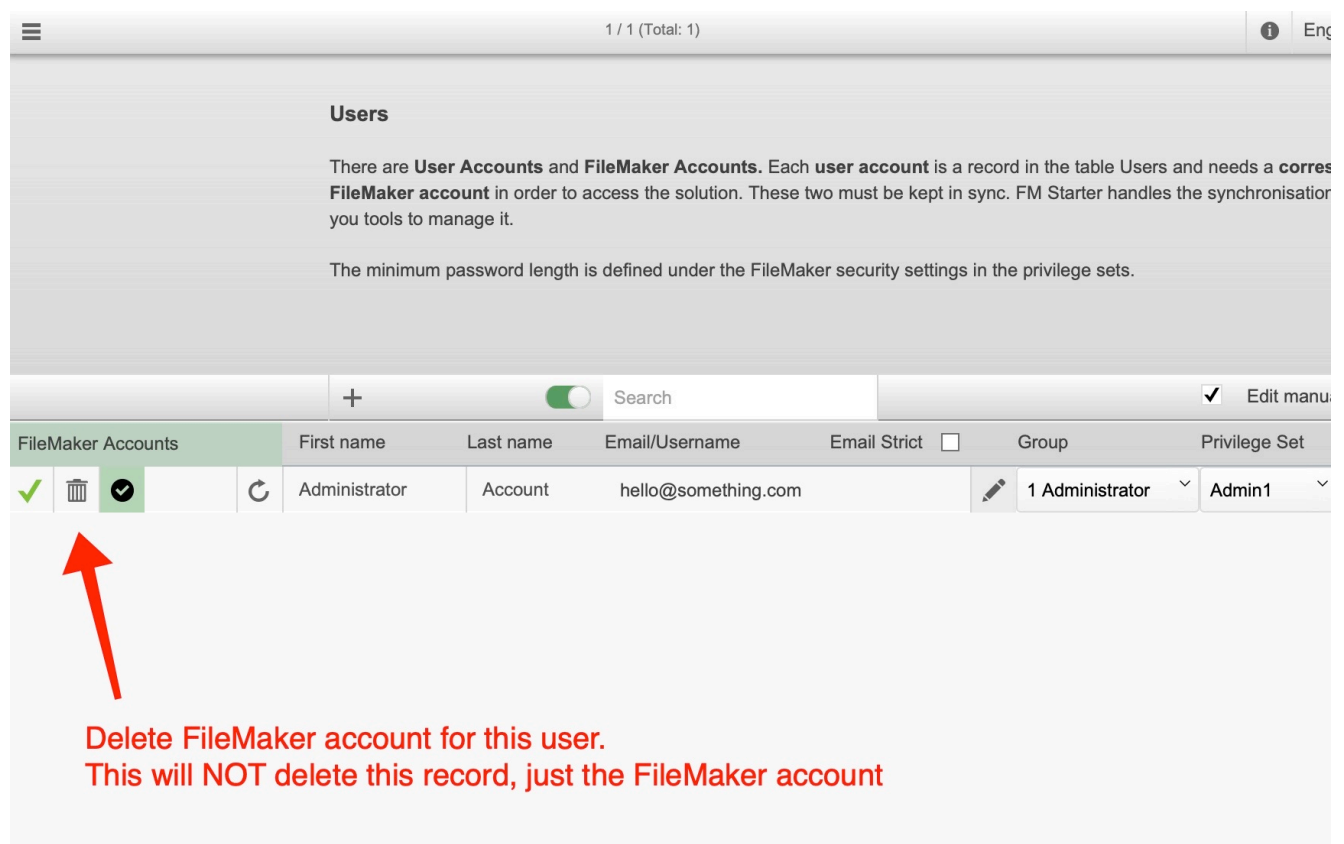
This permission set requires the user to create a new password when logging in for the first time. The setup for this account is created in the scripts:



The “Add account” script step has the option to expire the password. Simply activate it if required.

Forgot your expired password?

If the user loses their "expired" password, you must delete the FileMaker account (not the record) from the user table and recreate the account to generate a new password. This is done here:







The screenshot shows the FileMaker Users table interface. At the top, there's a header bar with a menu icon, a status bar showing "1 / 1 (Total: 1)", and a language dropdown set to "Eng". Below the header, there's a section titled "Users" with explanatory text: "There are **User Accounts** and **FileMaker Accounts**. Each **user account** is a record in the table Users and needs a **correct FileMaker account** in order to access the solution. These two must be kept in sync. FM Starter handles the synchronisation you tools to manage it." and "The minimum password length is defined under the FileMaker security settings in the privilege sets." Below this is a table with columns: "FileMaker Accounts", "First name", "Last name", "Email/Username", "Email Strict", "Group", and "Privilege Set". The table contains one record: "Administrator", "Account", "hello@something.com", "1 Administrator", and "Admin1". A red arrow points to the delete icon (a trash can) in the "FileMaker Accounts" column of the first record. Below the table, there's a red text box that says: "Delete FileMaker account for this user. This will NOT delete this record, just the FileMaker account".

1 / 1 (Total: 1)

Users

There are **User Accounts** and **FileMaker Accounts**. Each **user account** is a record in the table Users and needs a **correct FileMaker account** in order to access the solution. These two must be kept in sync. FM Starter handles the synchronisation you tools to manage it.

The minimum password length is defined under the FileMaker security settings in the privilege sets.

FileMaker Accounts	First name	Last name	Email/Username	Email Strict	Group	Privilege Set
   	Administrator	Account	hello@something.com	<input type="checkbox"/>	1 Administrator	Admin1

Delete FileMaker account for this user.
This will NOT delete this record, just the FileMaker account

Once the FileMaker account is deleted, you can create a new account and it will regenerate a new password. Next step:

1 / 1 (Total: 1)

Eng

Users

There are **User Accounts** and **FileMaker Accounts**. Each **user account** is a record in the table Users and needs a **corresponding FileMaker account** in order to access the solution. These two must be kept in sync. FM Starter handles the synchronisation of these two accounts for you tools to manage it.

The minimum password length is defined under the FileMaker security settings in the privilege sets.


Account deleted: Admin

+

Search

✓ Edit manually

FileMaker Accounts	First name	Last name	Email/Username	Email Strict <input type="checkbox"/>	Group	Privilege Set
Create account	Administrator	Account	hello@something.com		1 Administrator	Admin1



Create a new account with a new password for the user.
This is the correct way, when a user has expiring and self-set passwords and lost their access.

Data separation model

Occasionally a developer wants to know whether FM Starter supports a data separation model. What is this? With such a model, the developer separates the data into several files. In most cases, the user interface is separated from the data. This is done by introducing two files, one for the user interface and one for the data. There are situations in which this model has its advantages. But does it also work with FM Starter and - more specifically - with the UAC User Accounts module? The short answer is: No.

UAC User Accounts does not support a separation model

The separation model is not supported by the UAC User Account module. This is not due to FM Starter, but to the way FileMaker works. In order for 2 files to talk to each other, the same accounts must exist in all files. There is no synchronization option in the UAC User Account module and no additional module to accomplish this task.

Can it still be achieved? Yes, it is possible, but you have to program it yourself and adapt it to your own needs. We recommend creating a separate module for this task, which can then simply be integrated into any new file. This is the only way to ensure that the solution meets your requirements.

ToDo Developer

This is a simple module to keep track of your to-dos. It can include functions, planning, errors and many other things related to the software. It is a simple planning aid as well as a basis for further development.

For example, you can use the module to create a list of bug fixes and features for a specific release. The module is very simple and opens in an additional window:

To Do

X

+

≡

To Do's

i

Deutsch

▼

FM Starter 3.0, 17 July 2024

4 Erledigt

X

Nummer

Version

Titel

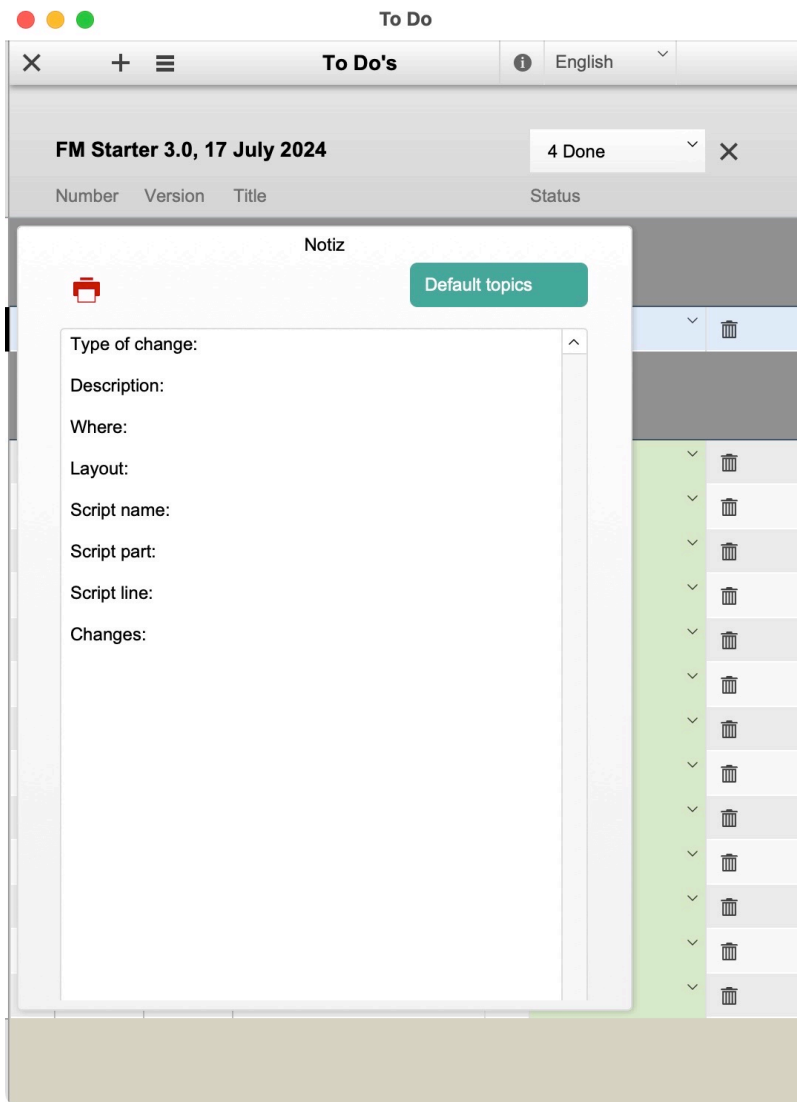
Status

4 Erledigt

▼

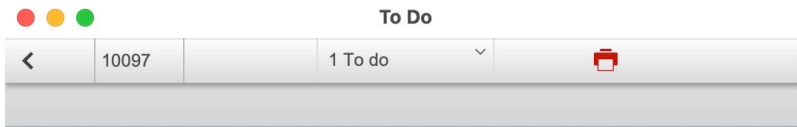
	10096	3.0	Window flashing reduced		4 Erledigt	▼	
	10095	3.0	Old help window removed		4 Erledigt	▼	
	10094	3.0	Layout Details Enhanced		4 Erledigt	▼	
	10093	3.0	GoToLayout enhanced		4 Erledigt	▼	
	10092	3.0	Dutch added as language		4 Erledigt	▼	
	10091	3.0	Smaller fixes and improvements		4 Erledigt	▼	
	10090	3.0	ToDo-window: DefaultTopics		4 Erledigt	▼	
	10089	3.0	Main window size dynamic		4 Erledigt	▼	
	10088	3.0	Fix Unlock Module		4 Erledigt	▼	
	10087	3.0	GN Global Navigation		4 Erledigt	▼	
	10086	3.0	New Help module		4 Erledigt	▼	
	10084	3.0	ToDo-window: Info-Icon		4 Erledigt	▼	
	10083	3.0	Start screen		4 Erledigt	▼	
	10082	3.0	DeepL fix		4 Erledigt	▼	
	10081	2.9	Navigation Settings		4 Erledigt	▼	
	10080	2.9	Plug-ins Registration backup		4 Erledigt	▼	

You can add detailed information to each entry:



A standard structure for each entry is possible at the touch of a button. This allows you to describe in detail where and what has been done.

The print icon can be used to display a printable view:



Neues ToDo

Type of change:

Description:

Where:

Layout:

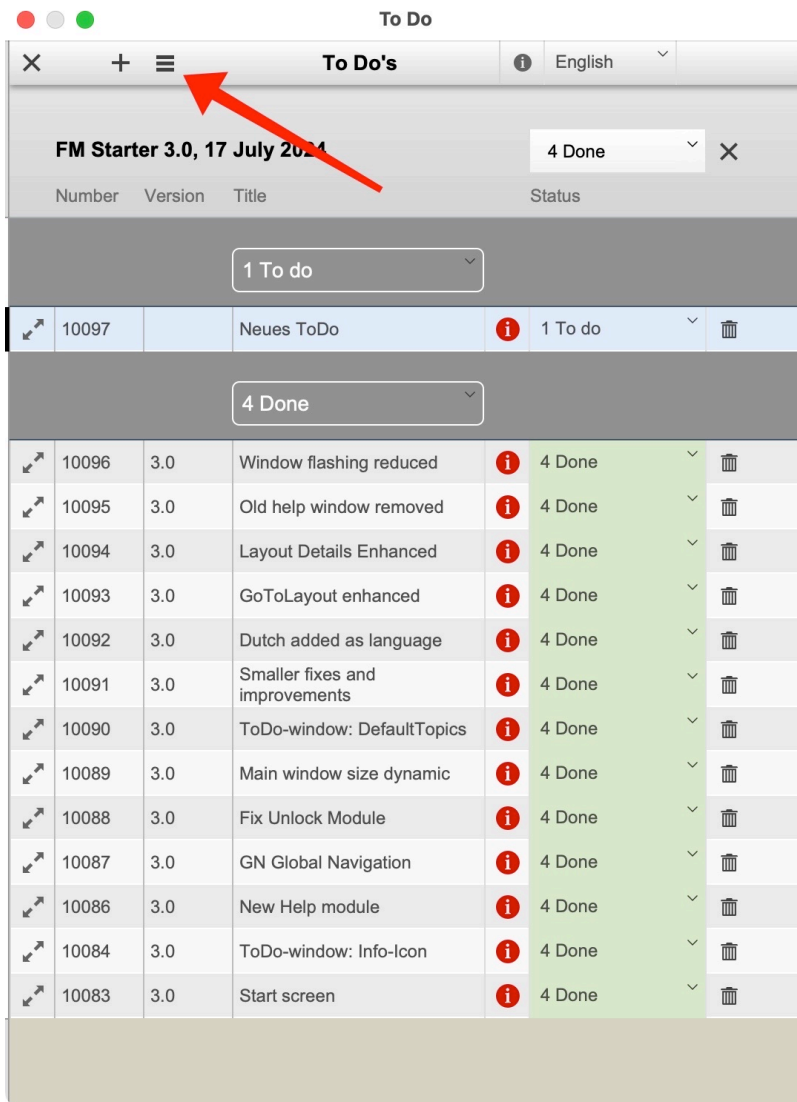
Script name:

Script part:

Script line:

Changes:

There is also a print function in the list view:



It first leads to an overview from which either a print or an export can be triggered.

To Do

FM Starter

List of changes (History)
Printed 17 July 2024.

	Neues ToDo	1 To do
3.0	Window flashing reduced	4 Done
3.0	Old help window removed	4 Done
3.0	Layout Details Enhanced	4 Done
3.0	GoToLayout enhanced	4 Done
3.0	Dutch added as language	4 Done
3.0	Smaller fixes and improvements	4 Done
3.0	ToDo-window: DefaultTopics	4 Done
3.0	Main window size dynamic	4 Done
3.0	Fix Unlock Module	4 Done
3.0	GN Global Navigation	4 Done
3.0	New Help module	4 Done
3.0	ToDo-window: Info-Icon	4 Done
3.0	Start screen	4 Done
3.0	DeepL fix	4 Done
2.9	Navigation Settings	4 Done
2.9	Plug-ins Registration backup	4 Done
2.9	User Login enhanced	4 Done
2.9	FMR Unlock removed	4 Done

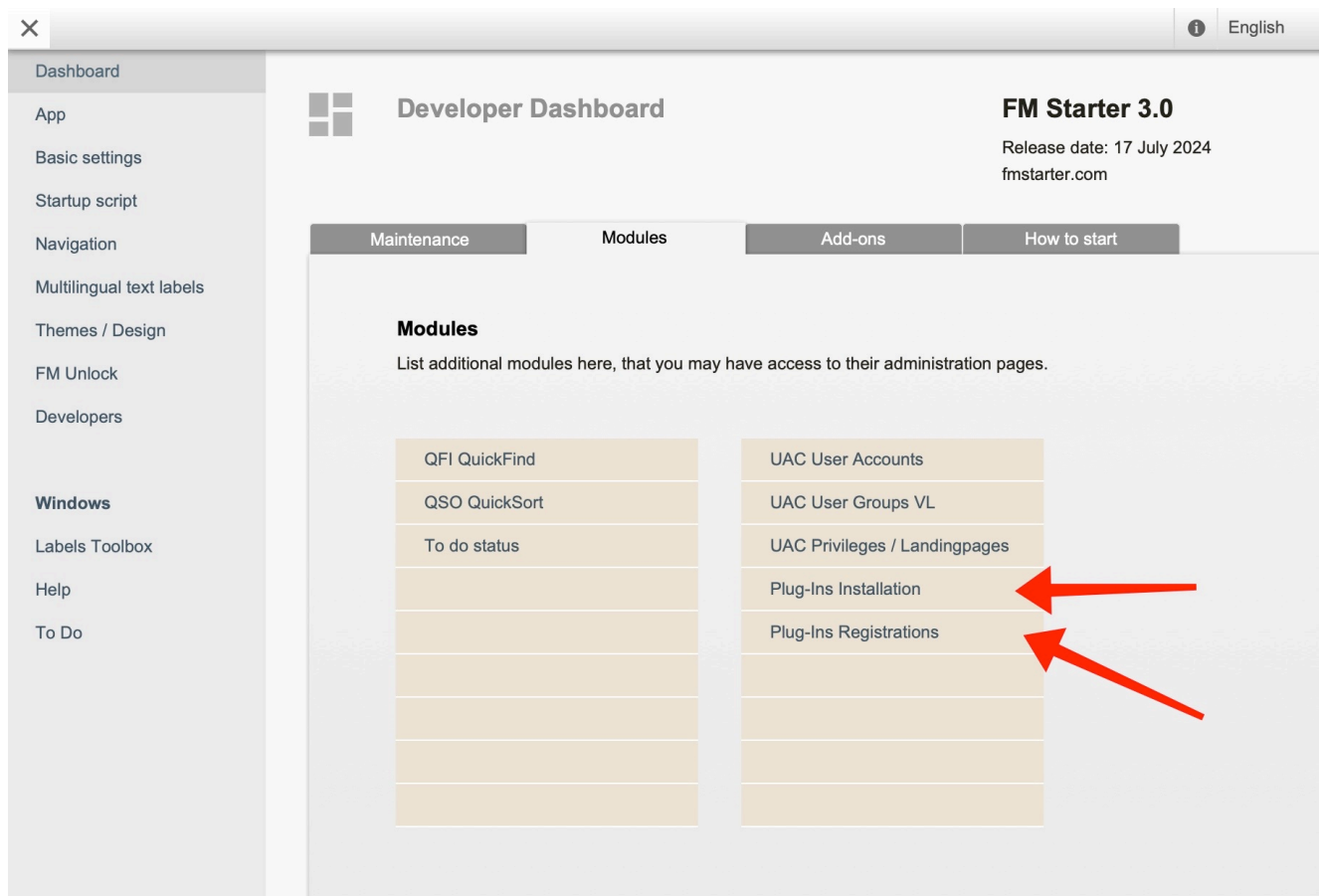
Plugins

The FileMaker functionality can be extended with the help of plug-ins. Plug-ins usually come from external suppliers. In order for plug-ins to be used in FileMaker, they must be installed and usually also registered.

FM Starter deliberately does not require any plug-ins, but supports the listing and registration of known variants.

Two modules are dedicated to the installation and registration of plug-ins.

1. Plug-ins installation
2. Plug-ins: Registrations.



Consider this implementation as a starting point. If you have special requirements or plug-ins from other manufacturers, adapt the modules to your needs.

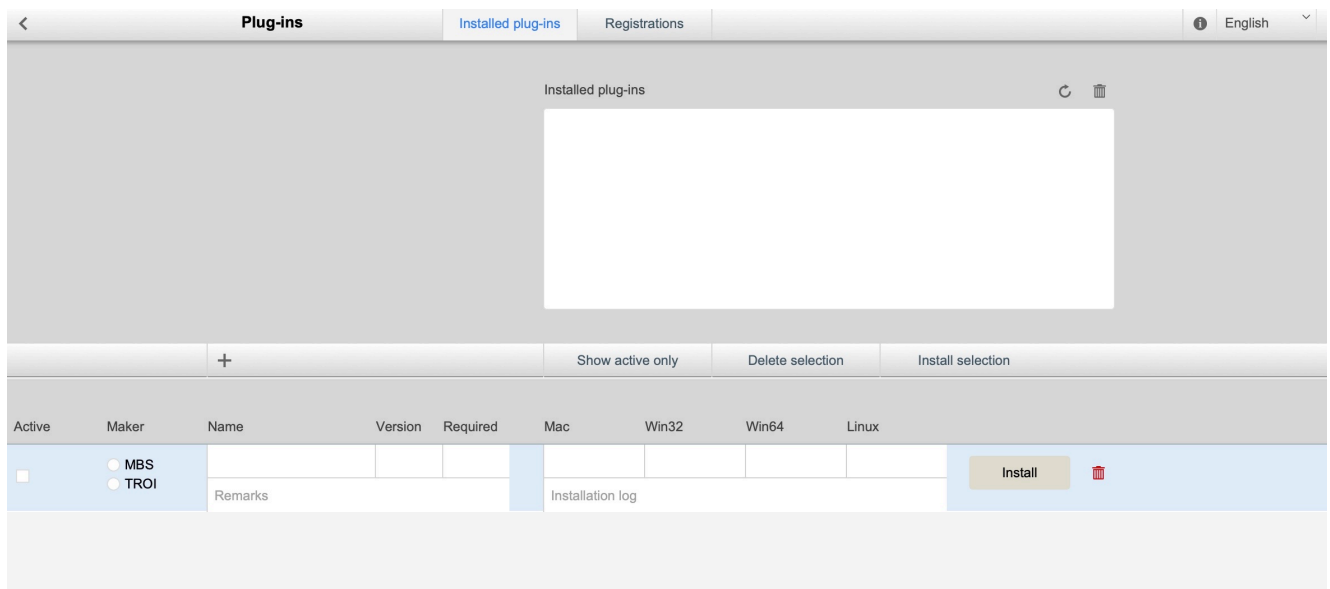
Installation of plugins

Plug-ins are external software solutions that can extend the standard functionality of FileMaker. FileMaker plug-ins can solve problems for which FileMaker itself has no options.

The plug-ins installation module does just that: it installs plug-ins and provides feedback if something goes wrong.

Where can I find the plugin installation module?

Find the module in the Developer Dashboard > Modules.



Adding a new plugin

Add a new data record, then ...

- fill in the information on the left and
- add the files on the right-hand side (per platform).

Add the developer

There is a switch for some popular plug-in developers (MBS/TROI). The switch does not do very much. The switch is intended for your further development, to simplify the selection of all plug-ins of a single developer or to get the verification of the files depending on the developer. This can be important

if you want to create installations in different locations or with different settings.

Install the plugin

Click on **Install** to install exactly these files (the correct file for the correct platform will be installed).

Further actions

- Show only active plugins
- Delete selection
- Install selection

Registering the plugins

As a rule, plug-ins must be registered. As this is a different task, we have created a separate module for registering plug-ins.

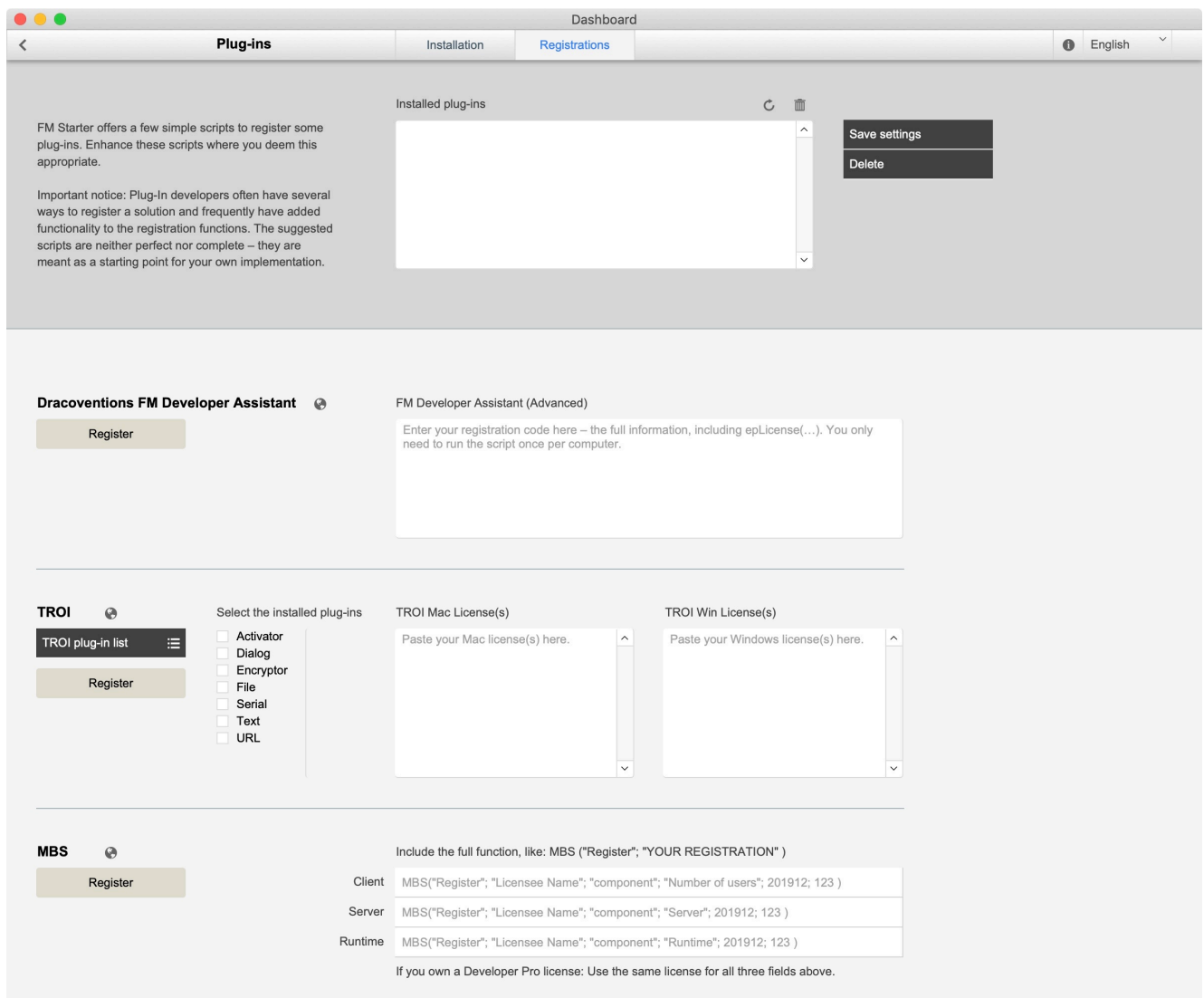
Registering plugins

After [installation](#), most commercial plug-ins require some kind of registration. For some, this may be a registration when starting a file. For others, it may be necessary to set up a registration each time a plug-in is called. Sometimes several options are available at once for you to choose from.

The plugin registration module contains some basic ideas and implementations for registering a plug-in. It is by no means complete, comprehensive or exhaustive. It is intended to give you a quick introduction to the topic and a basic implementation. You go on from there yourself.

Where can I find the plug-ins registration module?

The module can be found in the Developer Dashboard > Modules.



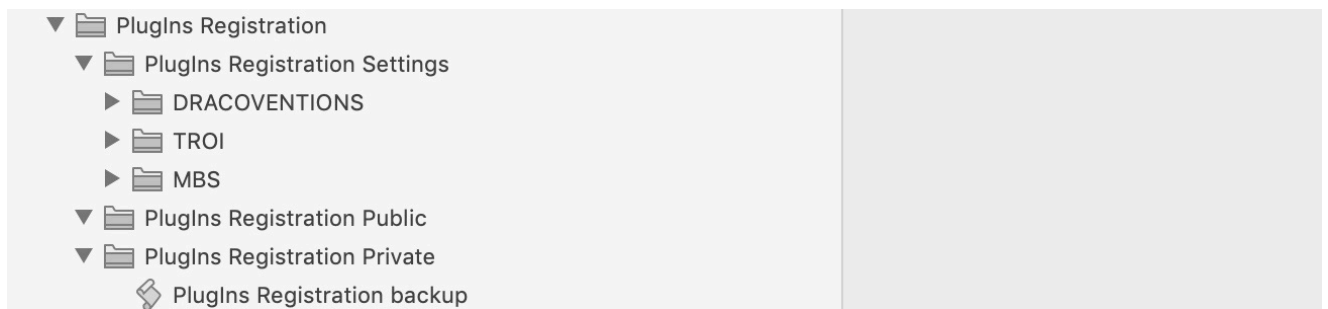
Supported products

The module contains 3 preconfigured registrations for the following products:

1. Dracoventions: FM Developer Assistant
2. TROI Plugins (see also plug-in list)
3. MBS Plugin

Setting up the scripts

As the mechanisms for each product are very different, there are only (simple!) setting scripts for each manufacturer:



To add additional products, you can create your own subfolders with your own scripts and then link them to the layout.

Design

Creating a user interface.

“Design is not just how it looks and how it feels. Design is how it works.”

Steve Jobs

About design

This short chapter is about the design, the look and feel of your application and how you can explain wordlessly how it works.

Design is at the heart of your development. Design is what your customers see, and that's all they want to know. Design represents the functionality in an interface that supports the customer's workflow. Design is essential, and: You can't non-design. So you better make sure that the design serves the purpose and the user.

FM Starter is designed to simplify your design process. You will still have to make your own decisions, but there is something of a design framework that you can use. It consists of color sets, predefined styles for almost every object and a variety of application examples (the FM Starter application itself).

Creating the user interface

The goal is not to make your user interface as realistic as possible. The goal is to add details that help users recognize what an element is for and how they can interact with it, and not to add more than these details.

- Lukas Mathis

The user interface doesn't need any fancy buttons. A great user interface disappears into the user's subconscious and simply helps them to accomplish the task at hand. User interface design is a purpose-driven art that has nothing to do with embellishment or fancy ideas.

Best practices

Pay attention to the following factors for a better layout:

- Consistency of layout elements, colors, functions
- Always keep the same elements in the same place on the page
- Reduced color set
- Highlighted colors for temporary feedback only
- Simplified masks

In general, "less is more".

What are themes?

A “theme” is a “design” or layout. Although it is not the finished product, it offers a selection of pre-defined elements that should simplify and standardize the design.

It is best to compare a theme with a CSS (Cascading Style Sheet), as used in web design. It is a central description of the appearance of all elements. A theme or design describes design elements and thus helps with a uniform design. FM Starter is supplied with a standard theme. It is used in all layouts. You can use this theme or duplicate it to create your own set.

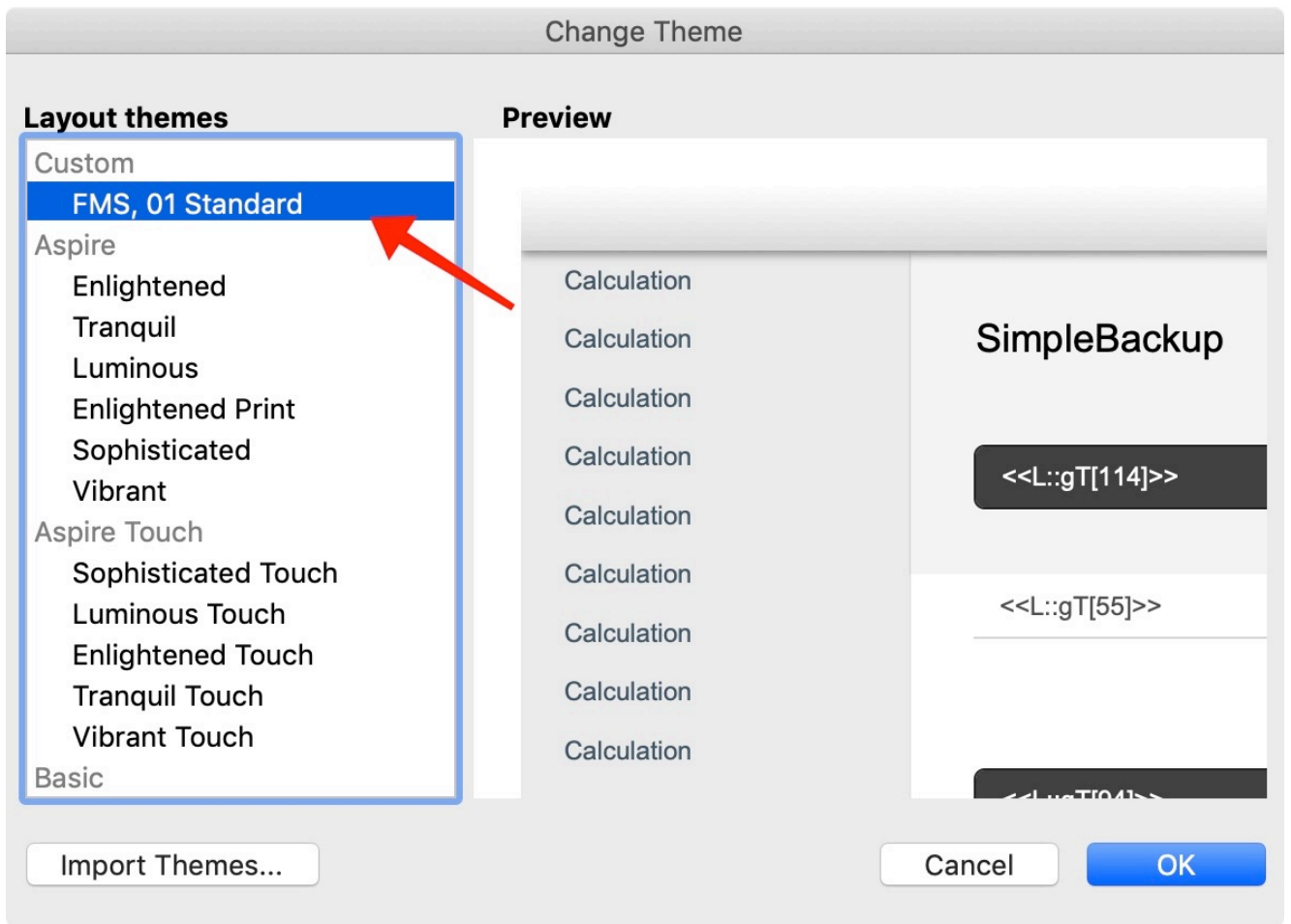
How is a theme structured?

A theme is a set of styles that give your layouts, objects and fields a consistent look. It helps your end user to perceive the software as uniform, making it easier to navigate and understand.

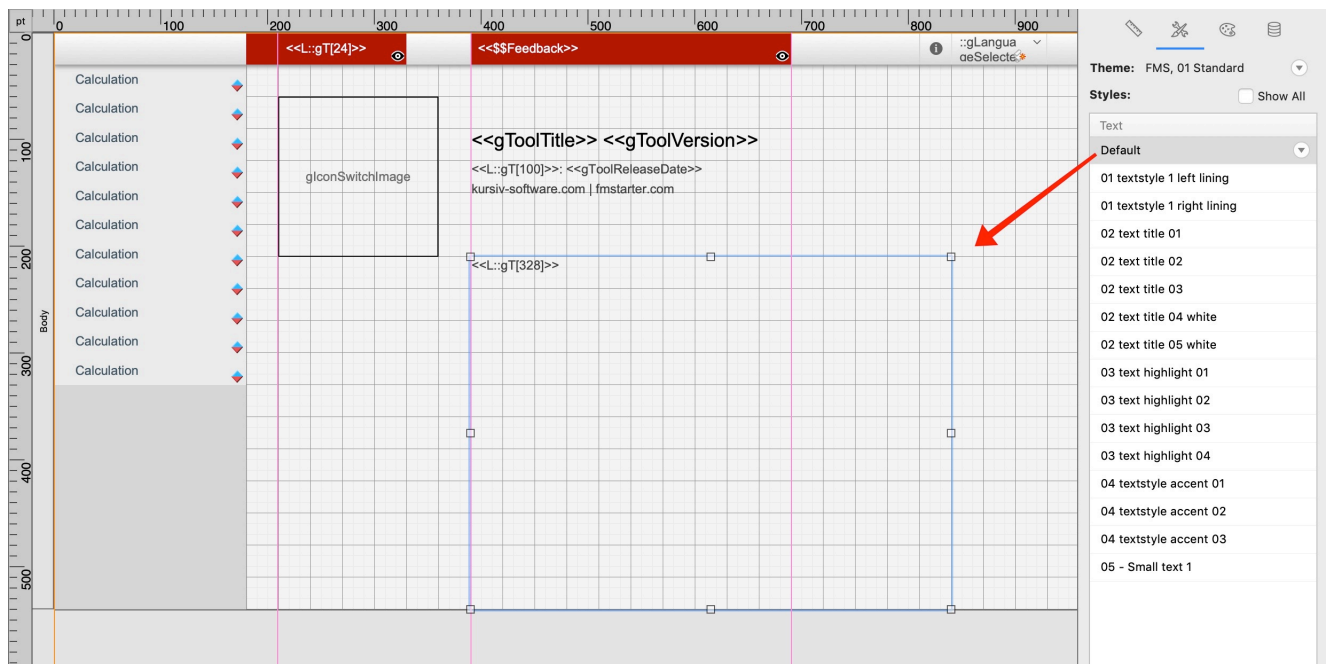
Typical features are, for example, definitions for text sizes, colors of objects, the behavior and appearance of buttons and much more. It is advisable to align a theme to the size of a specific design grid. FM Starter, for example, was built on a 30x30 pt grid. All objects and layout areas are based on a size of 30 pt or a multiple of this. This is reflected in the standard theme.

How do you use a theme?

A theme (or topic) is selected for a specific layout. You can change the design for each layout at any time when you are in layout mode.



Once activated, all style sheets for all objects from this design are available for the respective layout. To start using them, click on any object and then select a style template in the inspector.



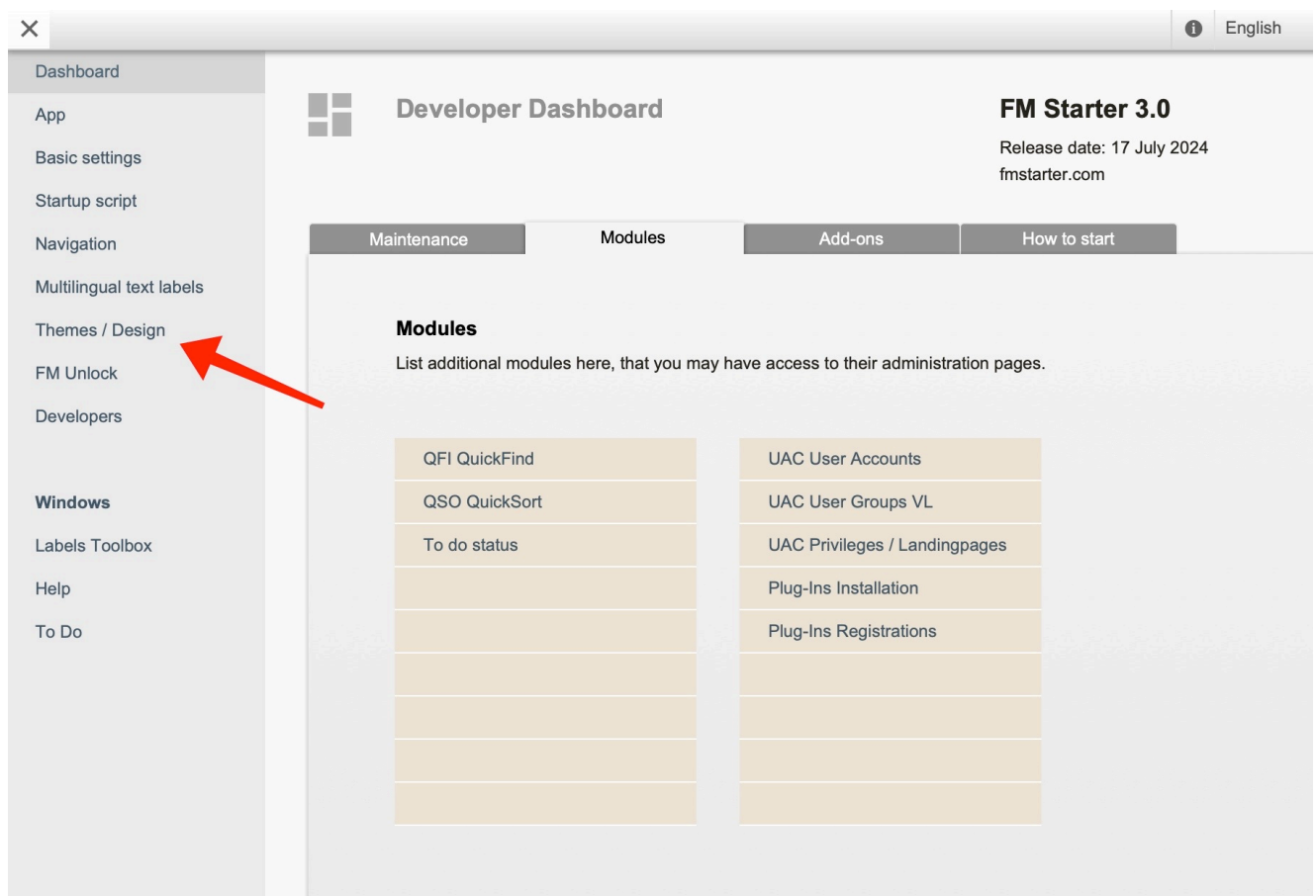
Further information on themes and how they work can be found in the FileMaker Help.

Theme section

Themes have their own section in the Developer Dashboard.

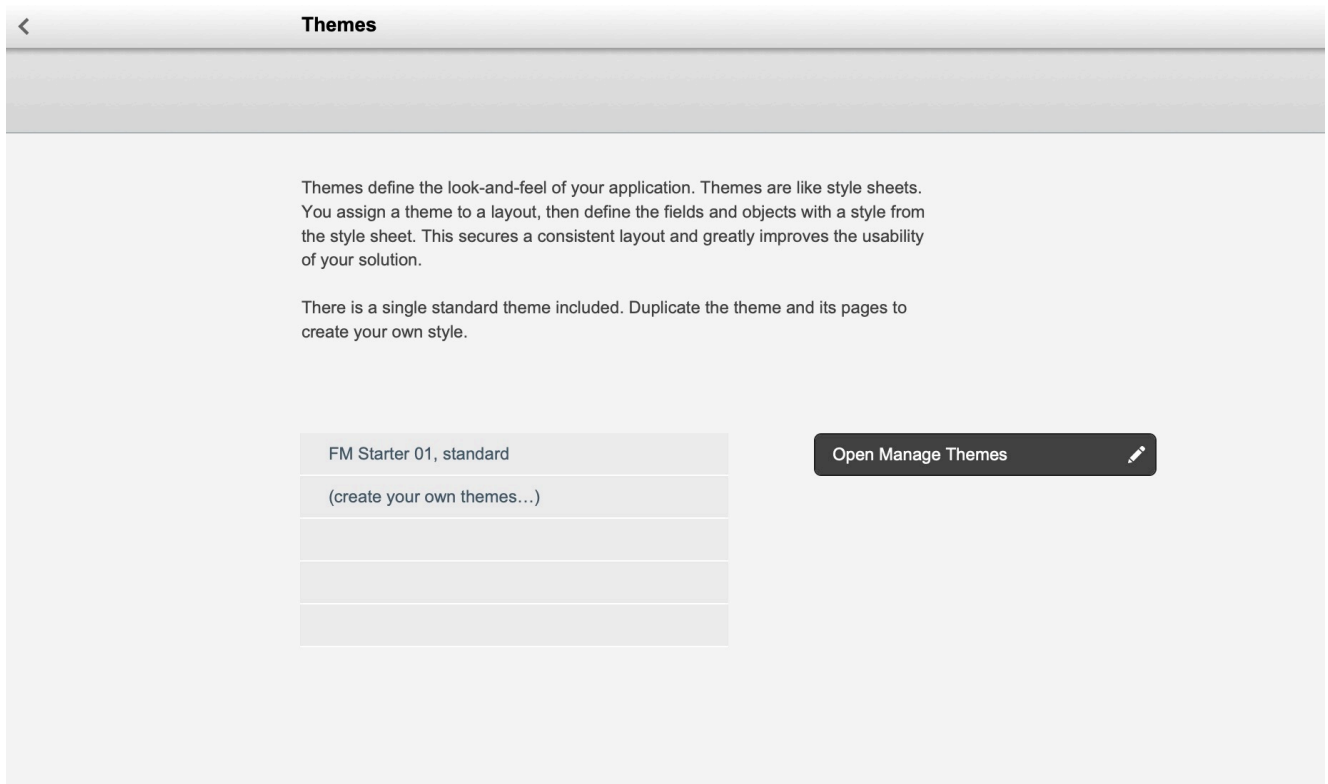
Topics are an integral part of FM Starter

The default theme is part of FM Starter. In the developer dashboard in the left navigation there is a link to the Themes / Design section. Here you will find a display of the theme / design used. You can also manage your own themes there.



Your own playground

The idea behind this section is to give you the structure to experiment, duplicate and test new designs. The following screen shows a possible list of designs:

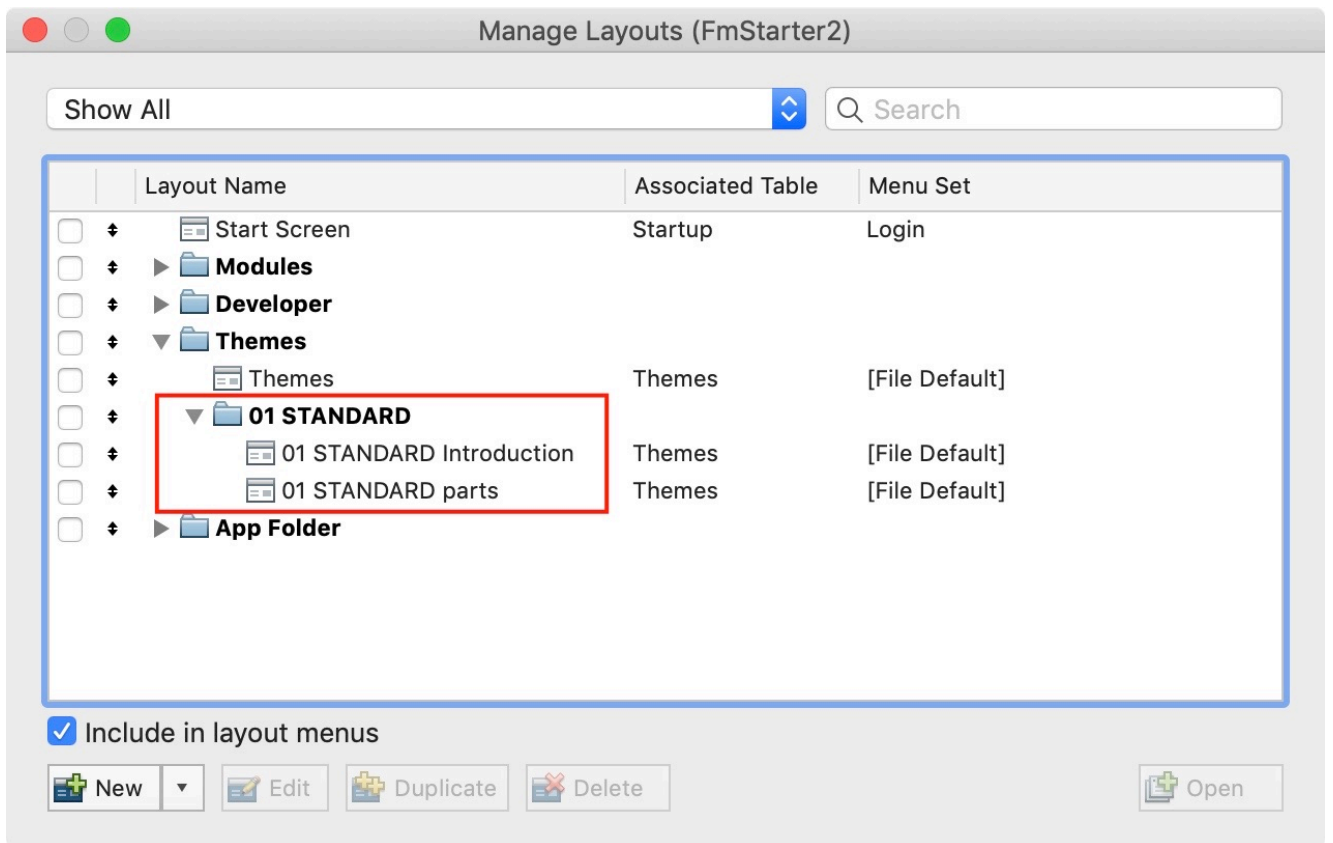


If you click on the link to FM Starter 01, Standard, the detailed pages for this topic will open.

The Theme Table

The theme area uses the theme table. This is a table with only a few fields that are needed to display certain things. This is the table you want to use to display other themes on master pages.

In the “Manage layouts” window, you will see that there is also a theme area, and the default theme has its own subfolder with 2 layouts.



Create your own designs

If you want to create new designs, it is recommended that you duplicate these layouts and this folder. Then duplicate the design and assign the new design to the new layouts. This way you have a clean start, can use the same layout and edit the differences step by step.

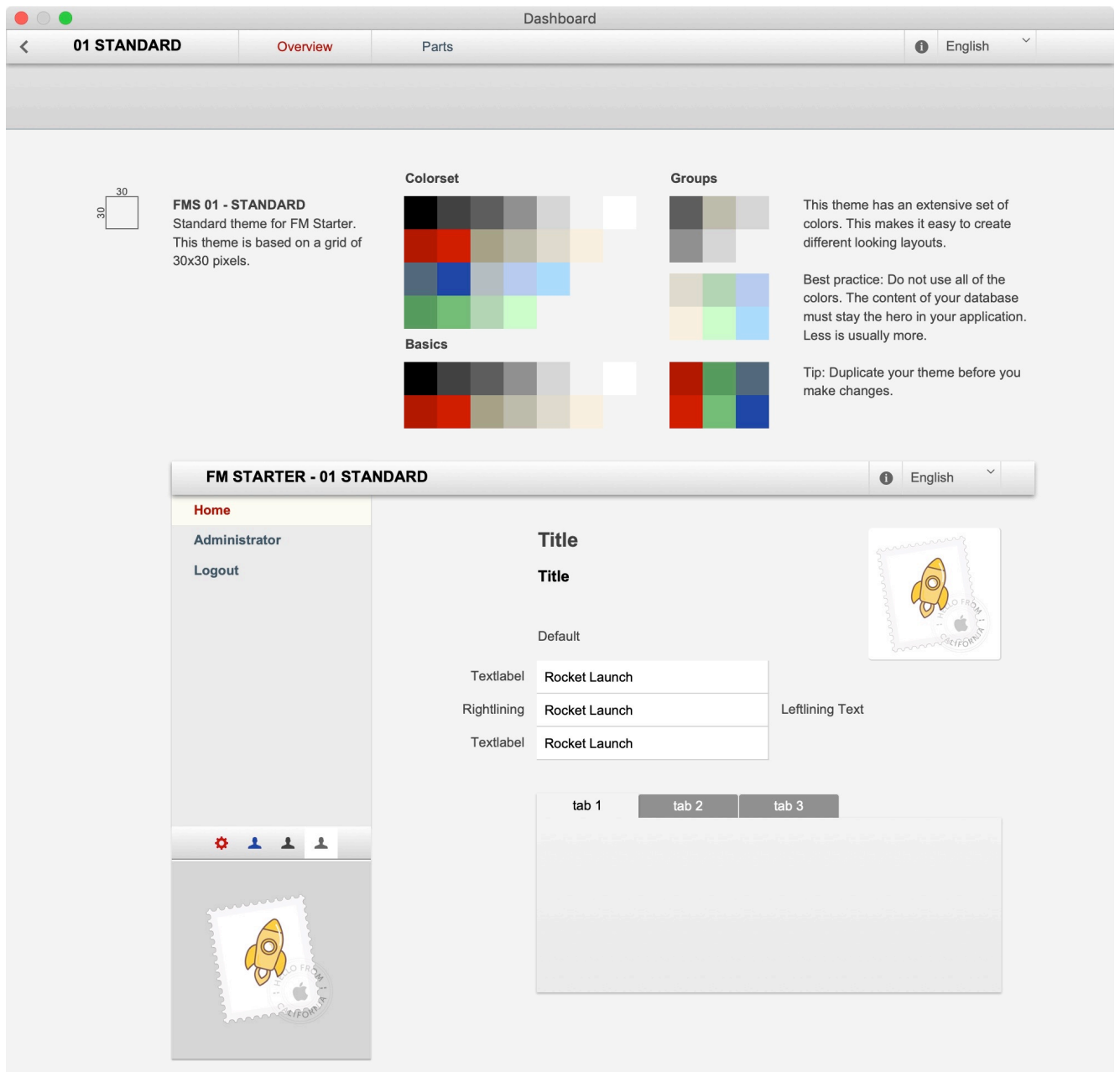
When your set is ready, create a link from/to the Themes page.

FMS01 Standard Theme

FM Starter comes with a single design developed specifically for FM Starter. It is a reduced set of style templates for each type of object, carefully tested and optimized. The set is large enough to offer variations from which you can choose your preferences.

The default theme can be accessed via the developer dashboard > Themes/Design > and select the right theme from the list (if you restart with FM Starter, you will only have the default theme).

1. Overview



The idea of this page is to make a few standard elements visible to give you a simplified view of this topic. You will also find a color set, with basic colors and several color groups. Although FileMaker does not (yet) support color sets today, it may be helpful to copy these colors into a system palette or something similar.

The color palette for this theme is quite large. This gives you several possibilities to create a darker or lighter version from the same set, a more or less

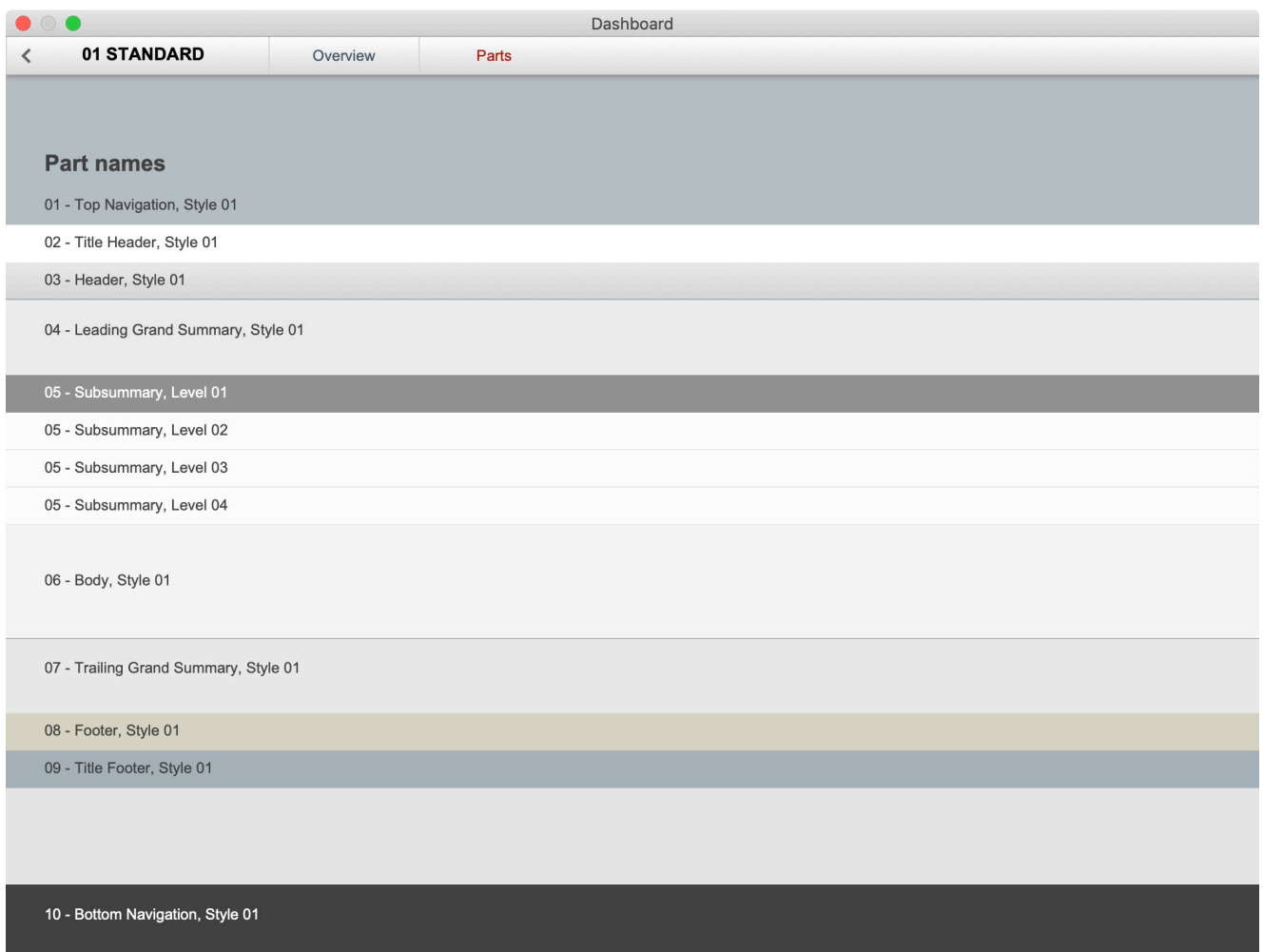
colorful version or simply a variety of choices for certain highlights. This is illustrated by the color groups.

Tip:

Keep a reduced set of colors in your design. The focus should be on the content, not the design.

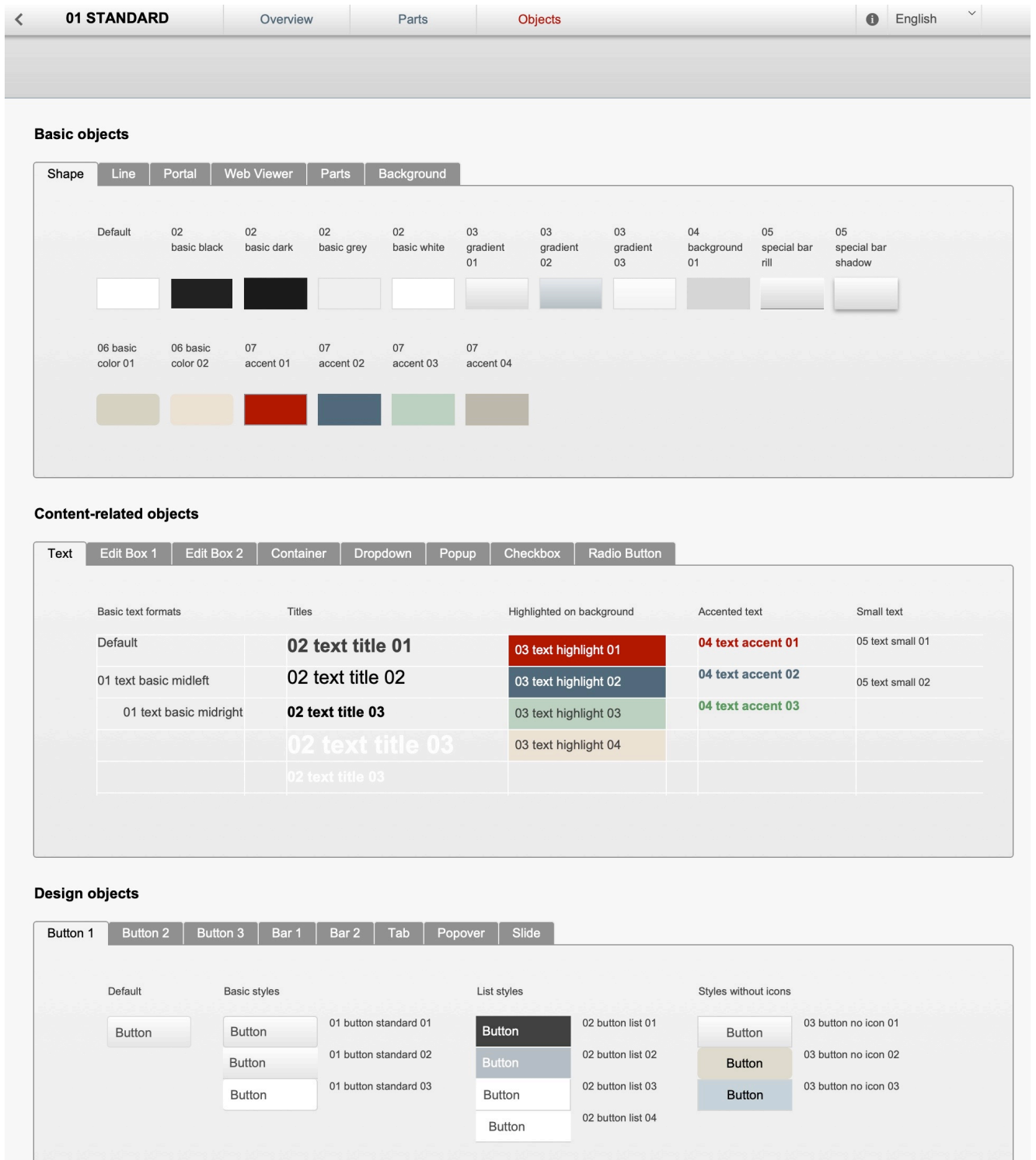
2. Parts

The second page is the Parts page, which shows a hierarchy of layout components:



3. Objects

The third part deals with individual objects. There is an example object for each style variant and each object type.



All three parts together offer a complete index of all styles, each with a preview. These pages describe, with their own styles, all variants of the theme.

If you want to create your own themes, you can duplicate these pages and list them under a new name. You can then duplicate the standard theme and apply it to the new pages. This is the prerequisite for being able to customize each object according to your own ideas and save it in the duplicated theme as an updated style sheet.

Support

FM Starter is an excellent starting point for your FileMaker development. This manual does not cover everything there is to say, as we assume you will also review the basic FileMaker manuals.

If you have a question or feedback about FM Starter, we'd love to hear from you. Just send us an e-mail to support@kursiv.com.

Links

Detailed information, videos and more about FM Starter can be found on the website

- fmstarter.com

Links to this website can also be found in the developer dashboard.

--