



FMWebschool Inc. 10307 West Broad St. Suite 179 Glen Allen VA 23060 Phone: 800.353.7950 Fax: 407-386-3109



<http://www.fmwebschool.com/usbsentry.php>

FileMaker USB Solution Protector

Introduction.....	2
What is USB Sentry?	2
Why Use USB Sentry Instead of Other Solutions?	2
System Requirements.....	2
How to Use USB Sentry?	2
Installation.....	3
Windows	3
FileMaker Pro	3
Macintosh.....	3
FileMaker Pro	3
USB Sentry Quick Start Guide.....	4
Section 1 – Plug-In Installation.....	4
Locating the USB Sentry Plug-In	4
Installing the Plug-In into your FileMaker Installation	4
Verifying the Installation	5
Section 2 – Database Creation	7
Creating the database File	7
Creating the Database Tables.....	8
Creating the Database Fields	10
Creating the USB Sentry FileMaker Layout.....	13
Section 3 – Creating Users.....	15
Adding a New Privilege Set.....	15
Adding a New User Account	18
Section 4 – Creating USB Sentry Scripts	19
Importing Standard USB Sentry Scripts	19
Adding the Scripts to the Layouts.....	25
Creating the Startup Script.....	27
Adding the Administrative Script	28
Setting File Startup Options.....	29

Section 5 – Testing the Solution and Finishing Notes	31
Registering the Solution to a USB Device.....	31
Ordering Information.....	34
Support.....	34
Function Reference	35
USB_SENTRY_GetKeys function.....	35
USB_SENTRY_Verify function	35
USB_SENTRY_Version function	36

Introduction

What is USB Sentry?

USB Sentry is a FileMaker Pro plug-in which allows the application to retrieve unique identifiers from connected USB devices. Retrieving USB device information can serve as a powerful copy protection method, or even a method to authenticate a group of users based on their personal USB drives.

Why Use USB Sentry Instead of Other Solutions?

USB Sentry provides a unique advantage to the developer – a combination of hardware based copy protection and application portability. That’s right, USB Sentry is supported both on the Mac and Windows – and the unique USB identifying key is identical on both systems. This means that if you ever want to expand into a cross platform solution, all your current USB Sentry compatible device will just work!

System Requirements

FileMaker Pro Client – version 7 or higher
Installation of the USB Sentry Plug-In

How to Use USB Sentry?

USB Sentry is used through two external functions that are accessible through the FileMaker Calculation dialog. One of these functions returns a list of USB devices that includes their true manufacturer assigned serial numbers, while the other allows a quick comparison of a stored device key with the current list of devices.

Installation

USB Sentry plug-in is installed just like any other FileMaker plug-in. Simply place it into the appropriate extensions folder and it will load the appropriate external functions into your solution.

Windows

FileMaker Pro

Copy USB_Sentry.fmx into:

- C:\Program Files\FileMaker\FileMaker Pro\Extensions

Make sure to restart FileMaker Pro completely after installing the plug-in.

Macintosh

FileMaker Pro

Copy USB_Sentry.fmplugin into:

- /Library/FileMaker Pro/Extensions/

Make sure to restart FileMaker Pro completely after installing the plug-in.

USB Sentry Quick Start Guide

Section 1 – Plug-In Installation

Locating the USB Sentry Plug-In

The USB Sentry distribution package should resemble Figure 1-1. Browse the ‘Win Plug-In’ directory if you are a Windows user, or the ‘Mac Plug-In’ if you are a Mac user.

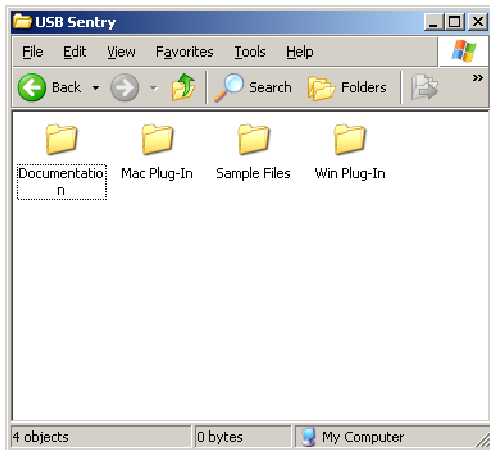


Figure 1-1: USB Sentry Distribution Files

Next select the actual plug-in file, just like in Figure 1-2 – then copy it into your clipboard.

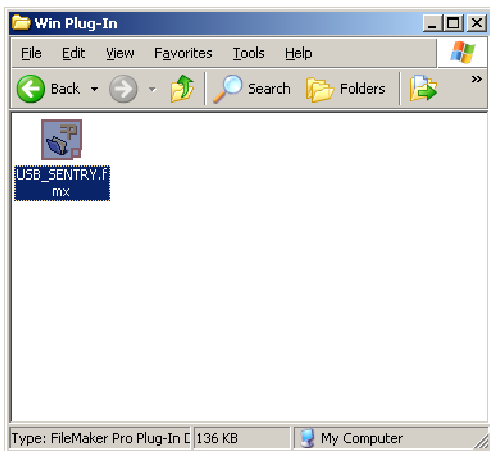


Figure 1-2: The Plug-In Itself

Installing the Plug-In into your FileMaker Installation

Next locate your FileMaker Pro or FileMaker Pro Advanced installation. For Windows users the installation is located by default within '**C:\Program Files\FileMaker\FileMaker Pro**' while for Mac users this should be

‘/Library/FileMaker Pro/’. Once you have located this directory, place the plug-in file into the **Extensions** directory – this directory should resemble Figure 1-3.

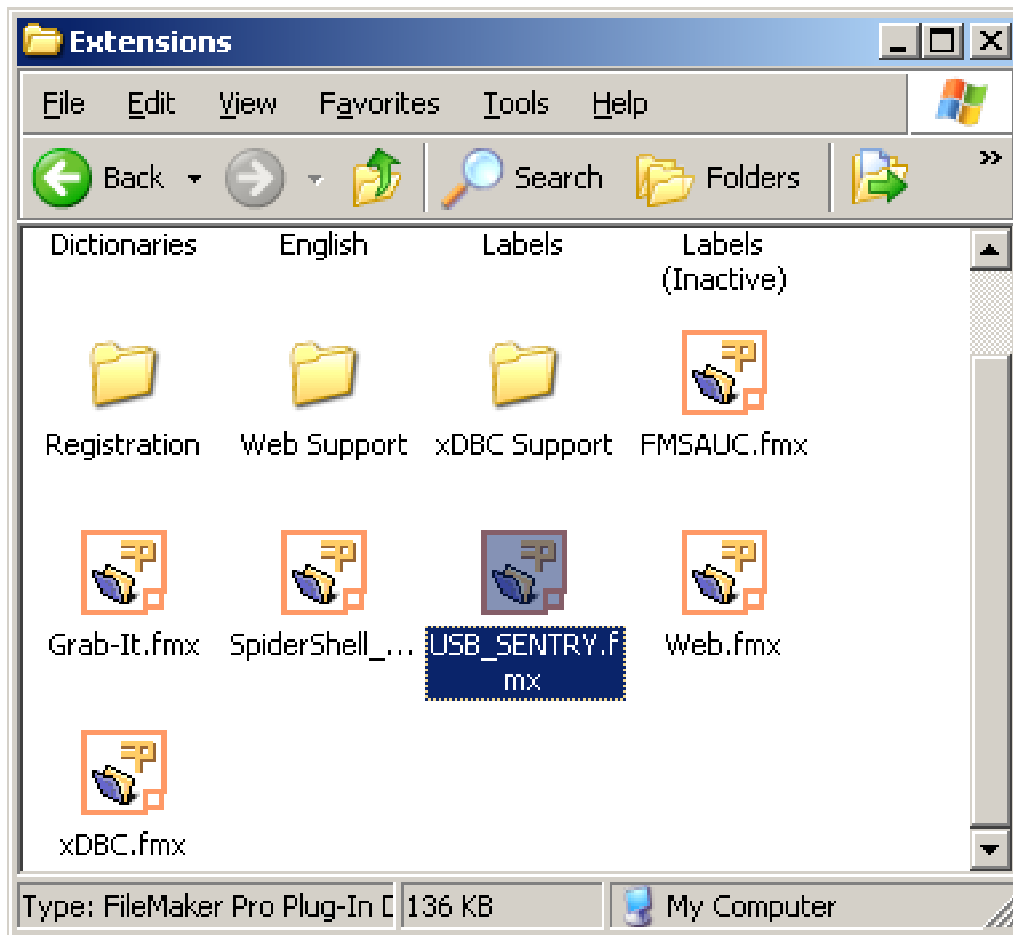


Figure 1-3: FileMaker Pro Extensions Folder

Verifying the Installation

Once the plug-in has been correctly placed into the Extensions directory it is time to make sure it is correctly detected. If you have a copy of FileMaker Pro open, please restart it now, then open FileMaker Pro. Next, use the Edit->Preferences menu on Windows or the FileMaker Pro->Preferences Menu on the Mac to open up the FileMaker Pro preferences dialog. This menu is shown in Figure 1-4.

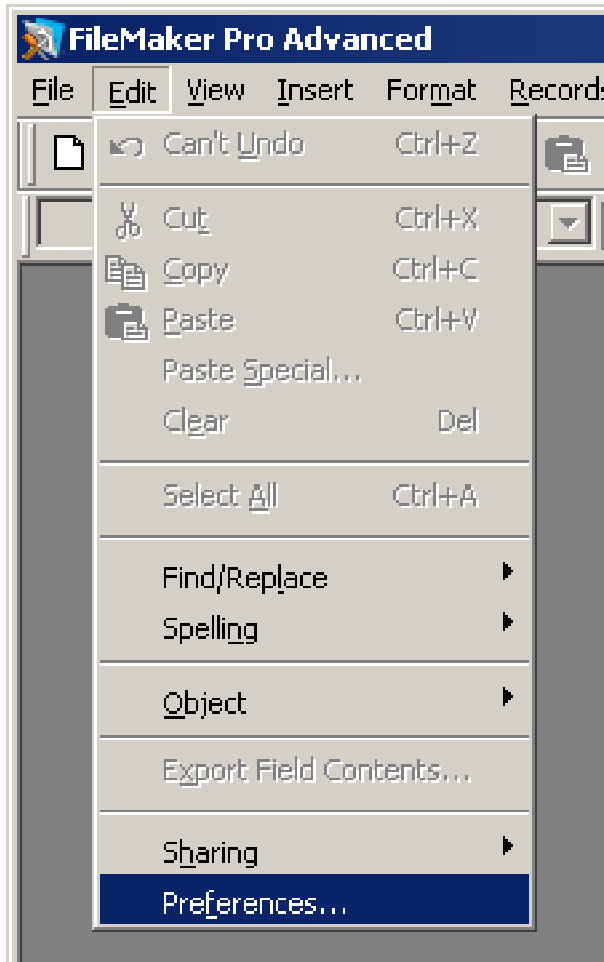


Figure 1-4: FileMaker Pro Preferences Menu Option

In the preferences menu find the 'Pug-Ins' tab and click it. Within the list of 'Enabled Plug-Ins' you should now see 'USB SENTRY' – if you do not see it at this point then please go back and redo the plug-in installation steps. This dialog should closely resemble Figure 1-5.

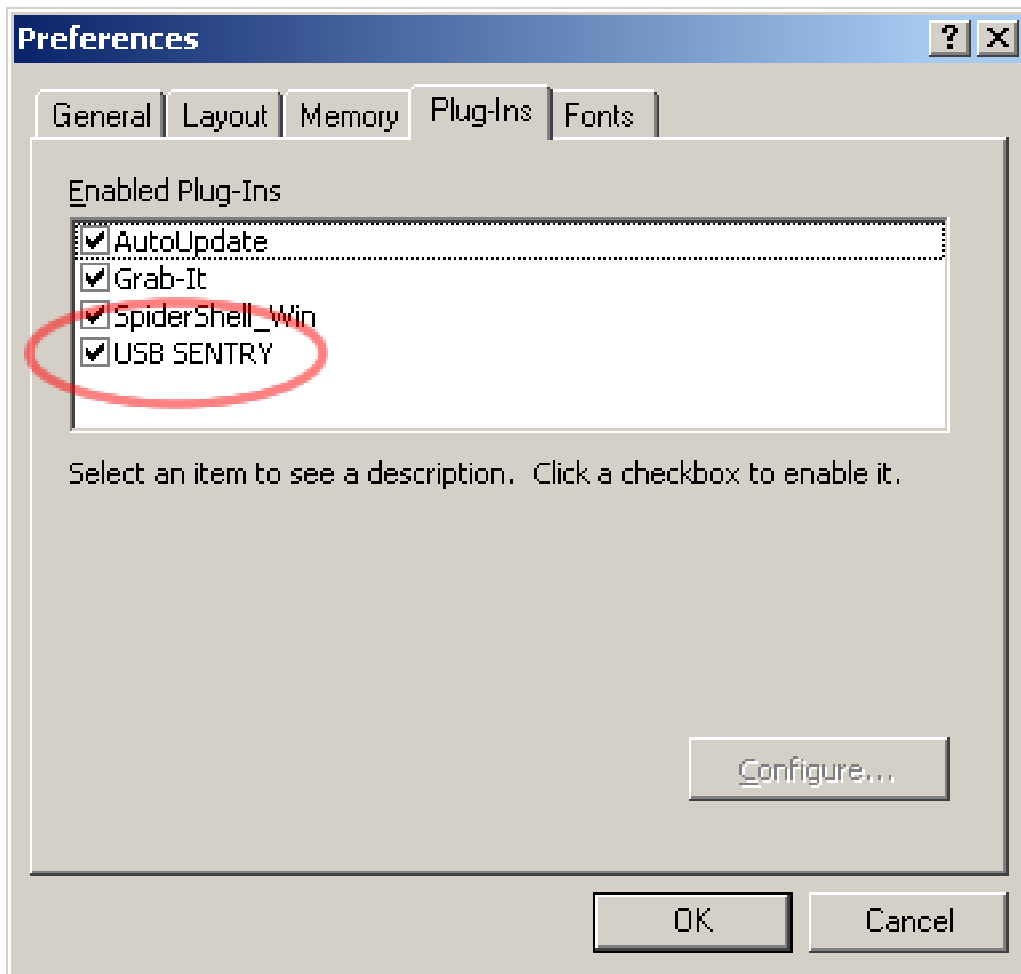


Figure 1-5: The Plug-In Enabled within FileMaker Pro

Section 2 – Database Creation

Creating the database File

In this quick start guide we will create a very small solution that will implement USB Sentry. Using the methods learned here you should be able to integrate USB Sentry into your solution with ease and confidence. The first step is to create a new database, in this case let's call this new file 'MySolution.fp7' – this step can be seen within Figure 2-1.

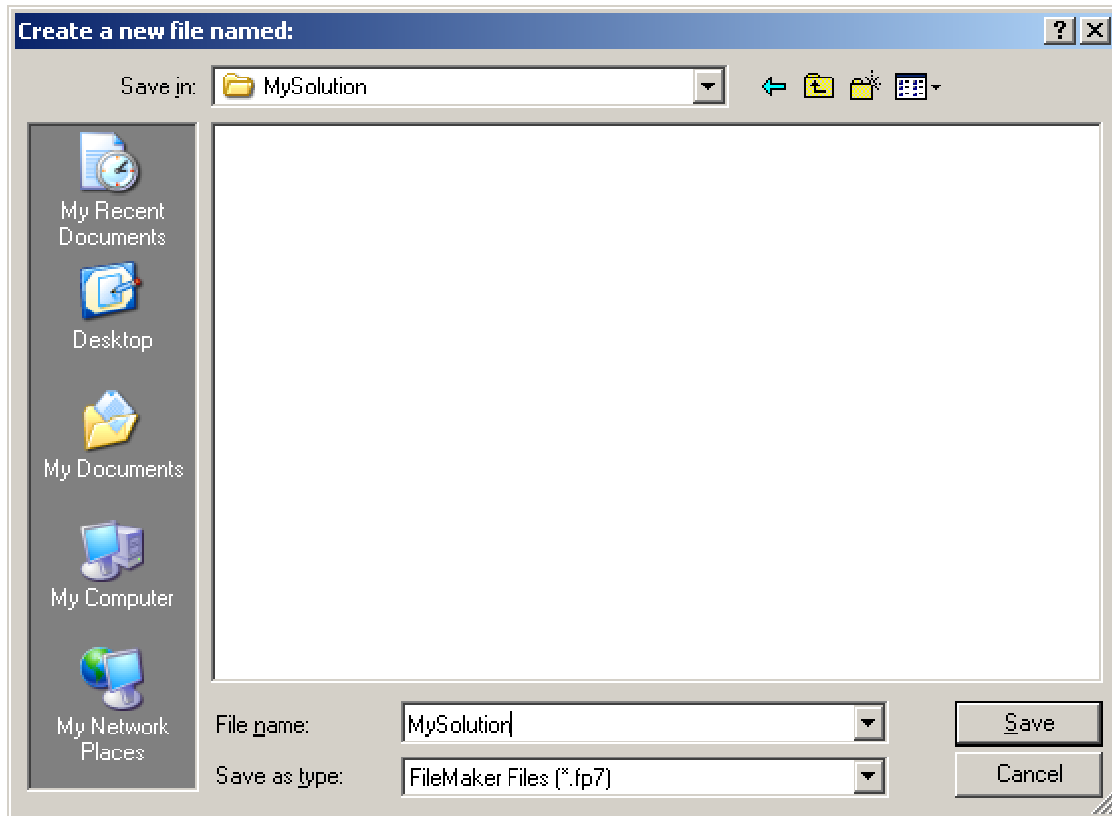


Figure 2-1: Creating a new FileMaker File

Creating the Database Tables

Next we create the database tables, for the main 'MySolution' table we will just create a simple 'data' text field. Create this text field now – Figure 2-2 shows the finished 'MySolution' table.

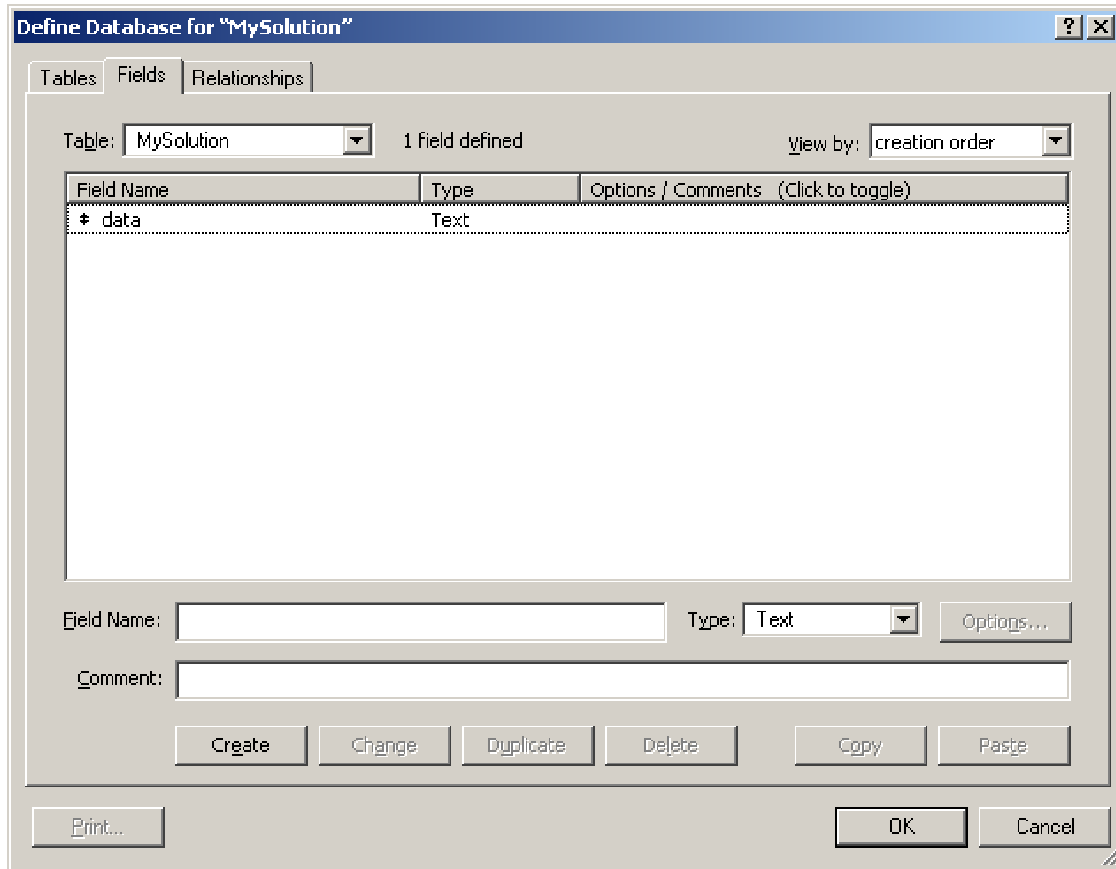


Figure 2-2: Finished 'MySolution' Table

Next it is time to create the actual USB Sentry table. It is very important to follow the exact capitalization and naming used within the screenshots below, otherwise importing USB Sentry scripts will be a much more complicated process. First create a new database table naming it '**USB_SENTRY**' just like Figure 2-3.

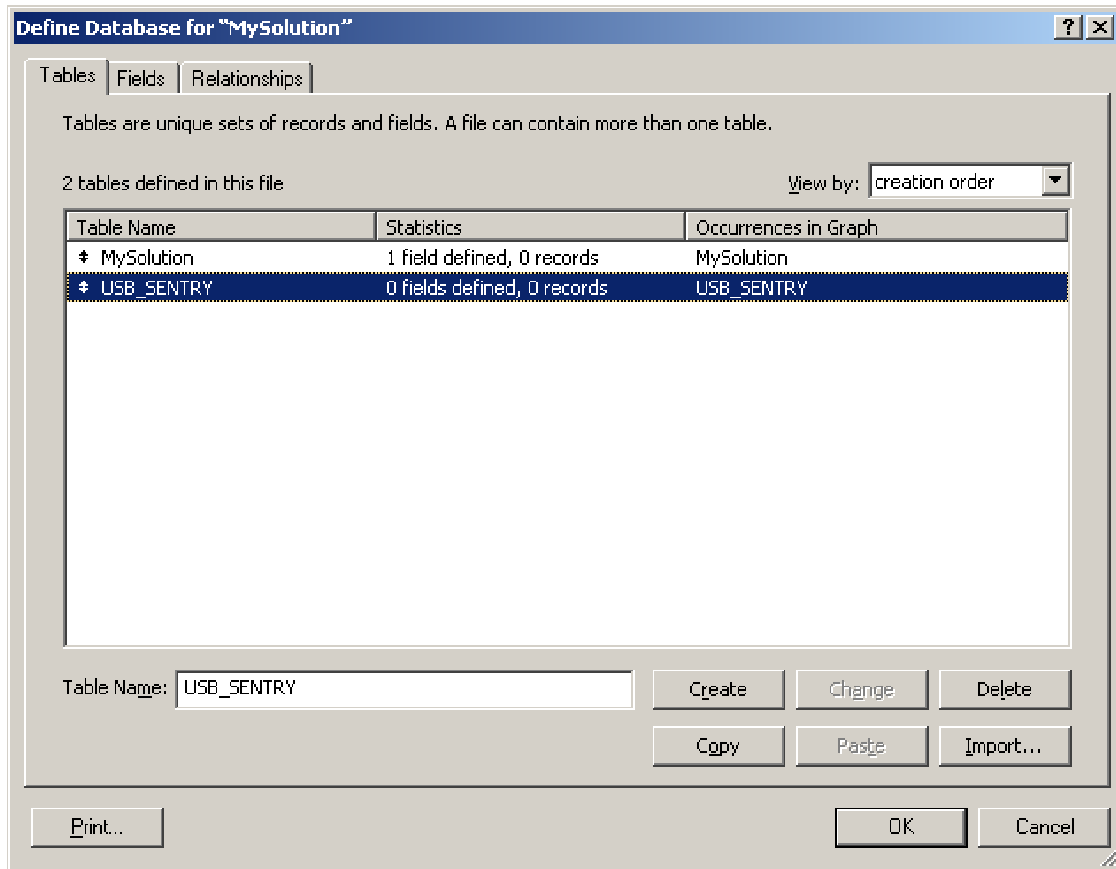


Figure 2-3: Creating the USB_SENTRY table

Creating the Database Fields

Now that the USB_SENTRY table has been created, it is time to add three fields to it. The first field is the 'storedKey', it must be a text field – this will hold the device serial number. Next it is the 'registrationKey' text field, this will hold your USB Sentry registration key if you have purchased the product. That last field is the deviceSelect numeric field, a temporary field needed for selecting devices from a list within one of the scripts. The finished set of fields should be identical to Figure 2-4.

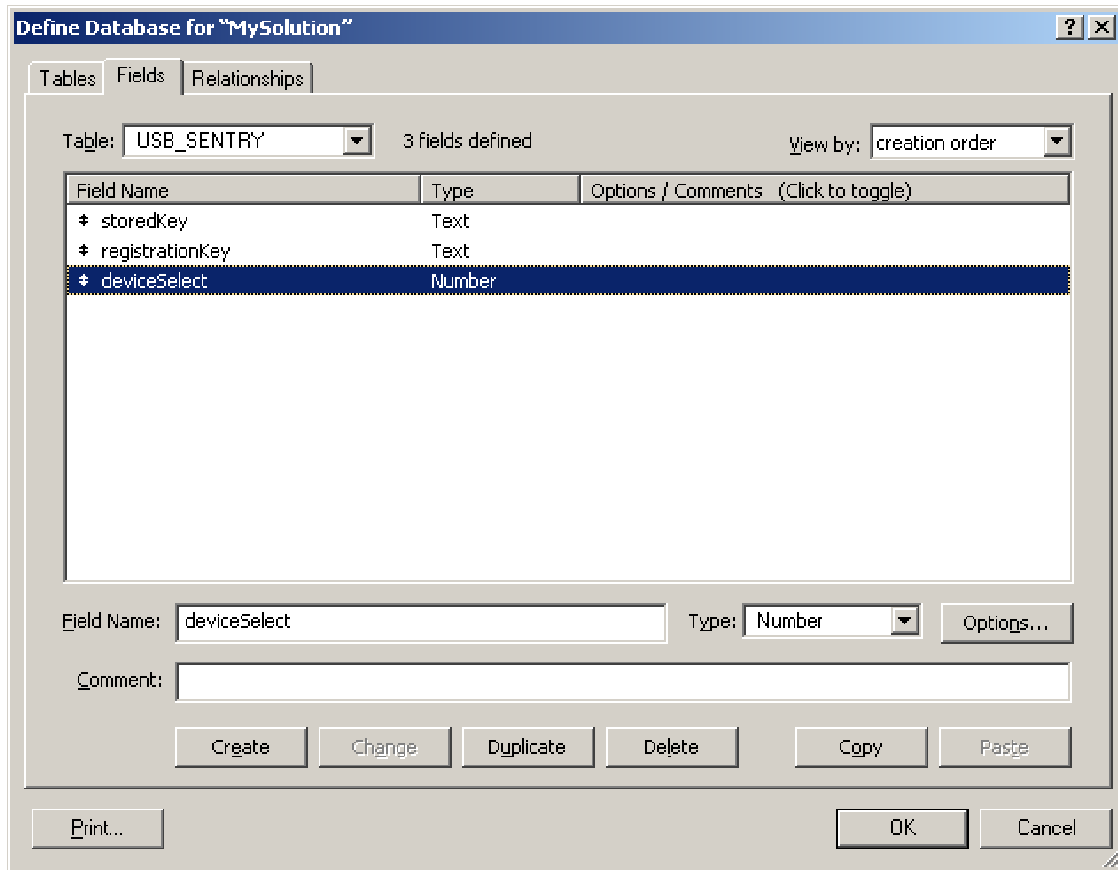


Figure 2-4: The Fields within the USB_SENTRY Table

Next all three fields have to be set as global fields. This is accomplished by selecting each field, and pressing the 'Options...' key. A dialog will come up with four tabs at the top, select the 'Storage' tab. Within this tab there is a checkbox for 'Use global storage', check this for all fields – Figure 2-5 demonstrates the location of this checkbox.

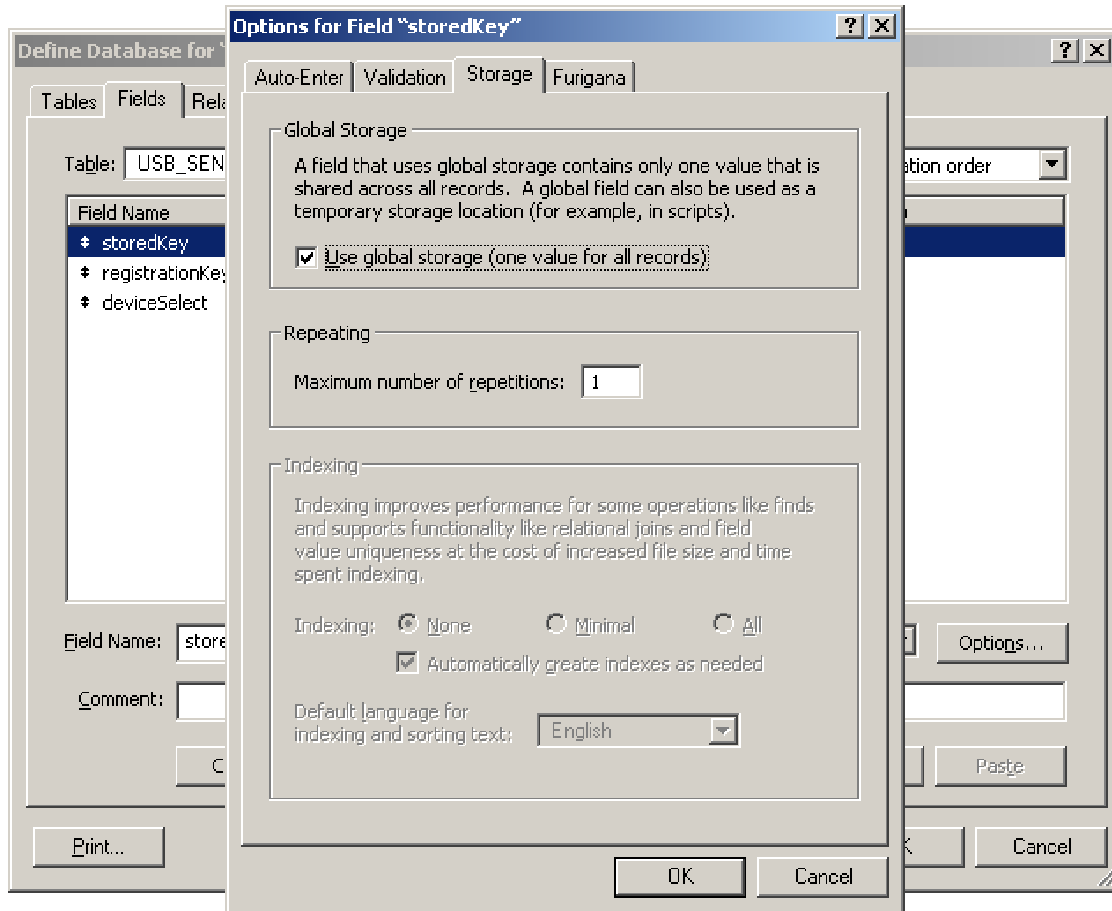


Figure 2-5: Dialog to Set a Field as a Global

Once all three fields have been set as globals, the fields list should be identical to Figure 2-6, if you do not see the third column in that list try clicking the headings to toggle it between 'Options' and 'Comments'.

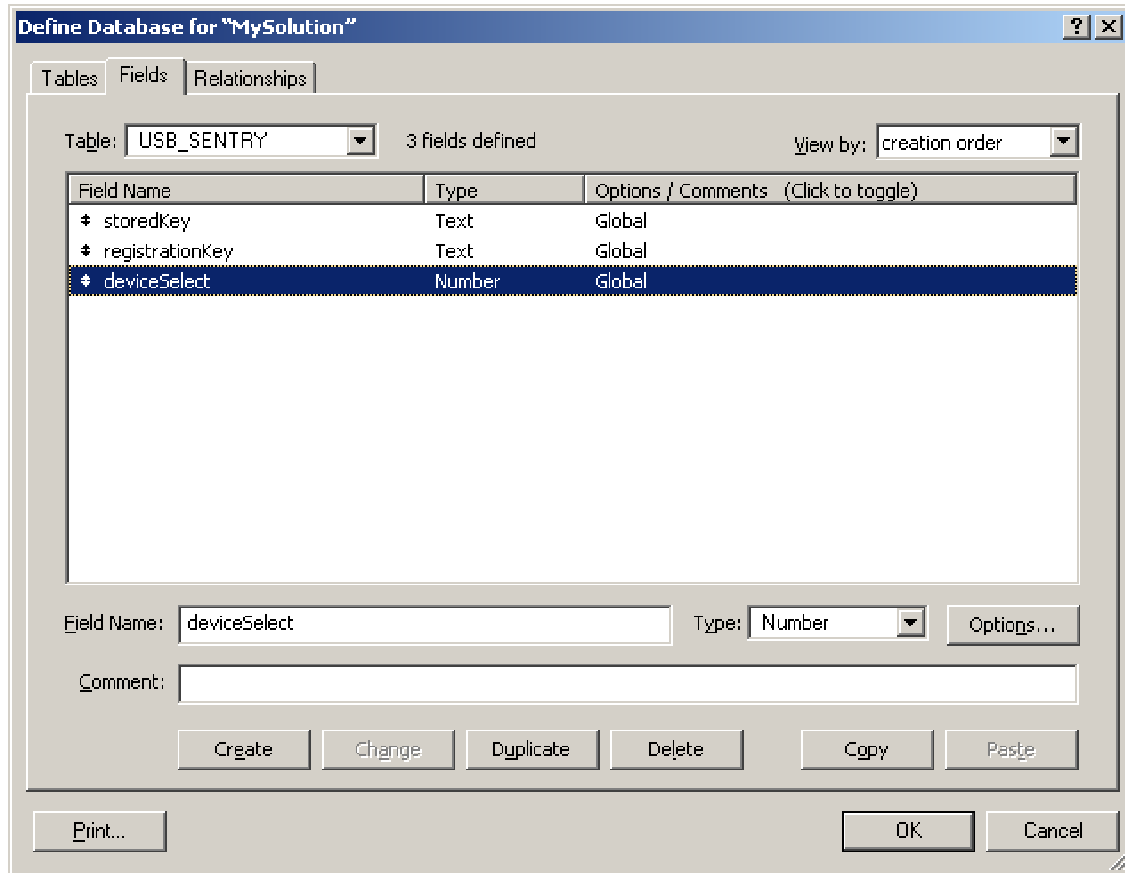


Figure 2-6: Completed Table Definition

Creating the USB Sentry FileMaker Layout

A layout named 'USB_SENTRY' was most likely created automatically when you created the table. This layout should be deleted at this time since we will be creating a new one specifically designed for USB Sentry. To delete the layout go into layout mode and then use the 'Layout' menu to find the delete option.

Next create a new layout named 'USB_SENTRY' with options identical to Figure 2-7.

New Layout/Report

Create a Layout/Report

Show records from: USB_SENTRY

Layout Name: USB_SENTRY

☒ Include in layout menus

Select a layout type

- Standard form
- Columnar list/report
- Table view
- Labels
- Vertical labels
- Envelope
- Blank layout

Shows one record at a time. Fields appear on separate lines. The field label is on the left and the field data is on the right. Good for data entry.

< Back Next > Cancel Help

Figure 2-7: Creating a New Layout

Select all three fields to be on the layout and then select a template for this layout. Once the template has been selected add a new button to this layout just like in Figure 2-8, currently this button can be set to execute a 'Perform Script' without any script selected.

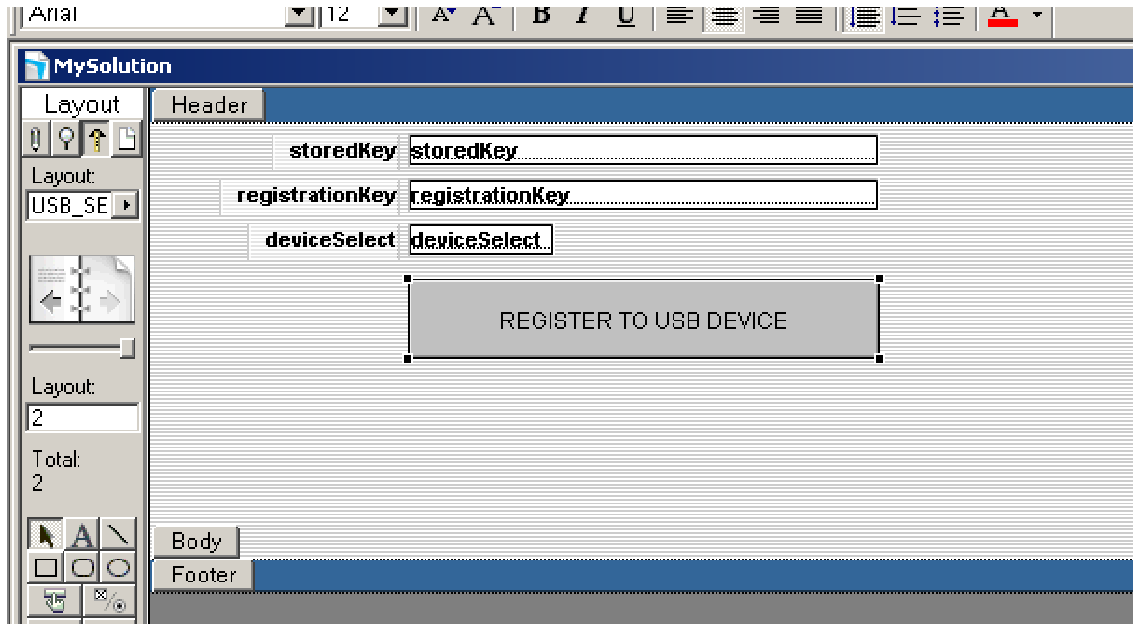


Figure 2-8: The USB_SENTRY Layout with a Button

Section 3 – Creating Users

Since USB Sentry is designed to protect your solution, it is only logical to allow the user access only to specific areas of the solution. USB Sentry should be used with its layout and records disabled from access by a non-administrative user, therefore this section will do just that by adding a new privilege set and then a user that uses it.

Adding a New Privilege Set

Access the 'Accounts & Privileges' dialog by using the 'File' menu and then selecting the 'Define' sub menu. Figure 3-1 shows how to access this menu item.

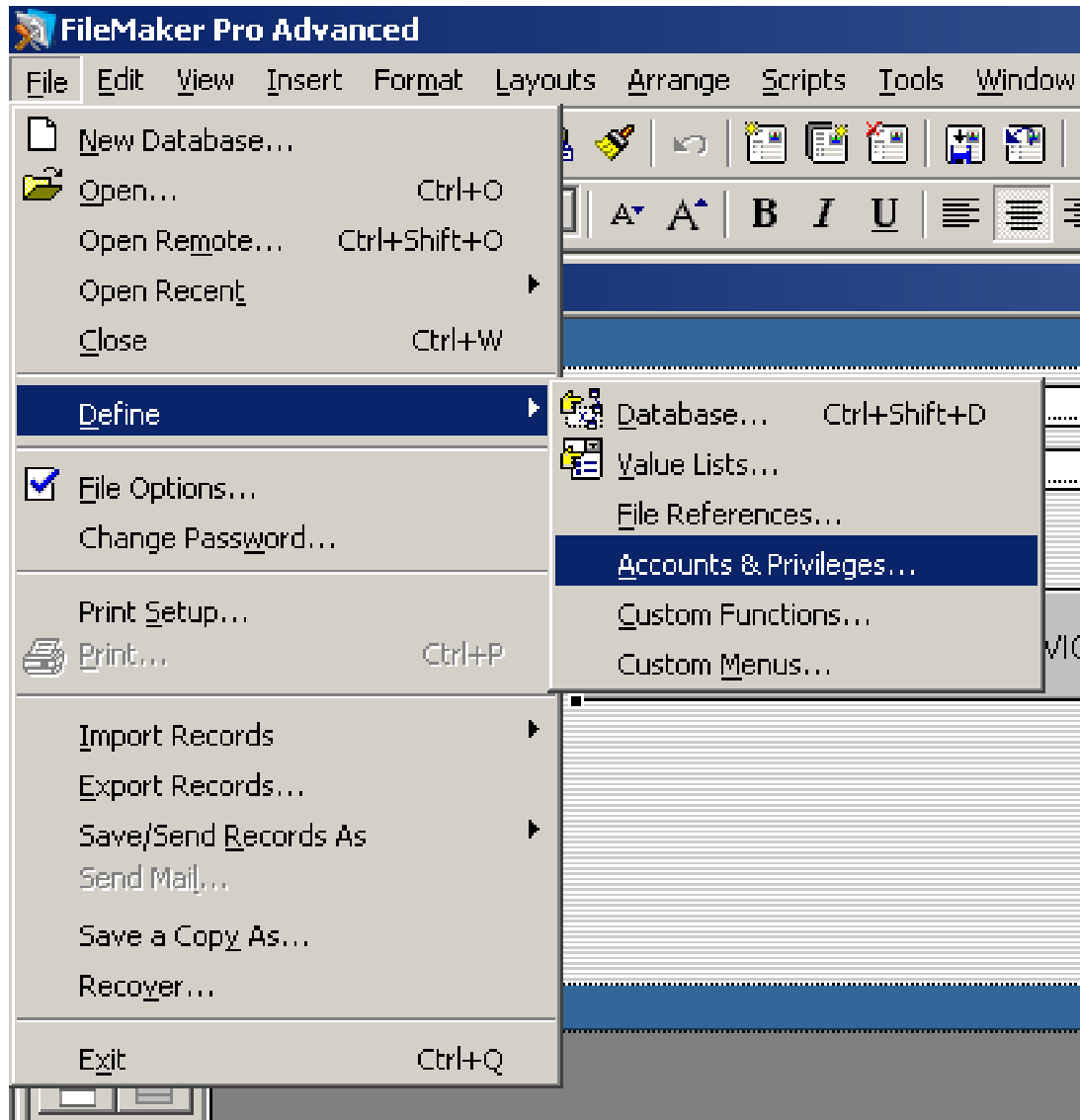


Figure 3-1: Accessing the 'Accounts & Privileges' Menu

Next it is time to create the new user privilege set, click the 'New...' button under the 'Privilege Sets' tab – Figure 3-2 outlines the location of this button.

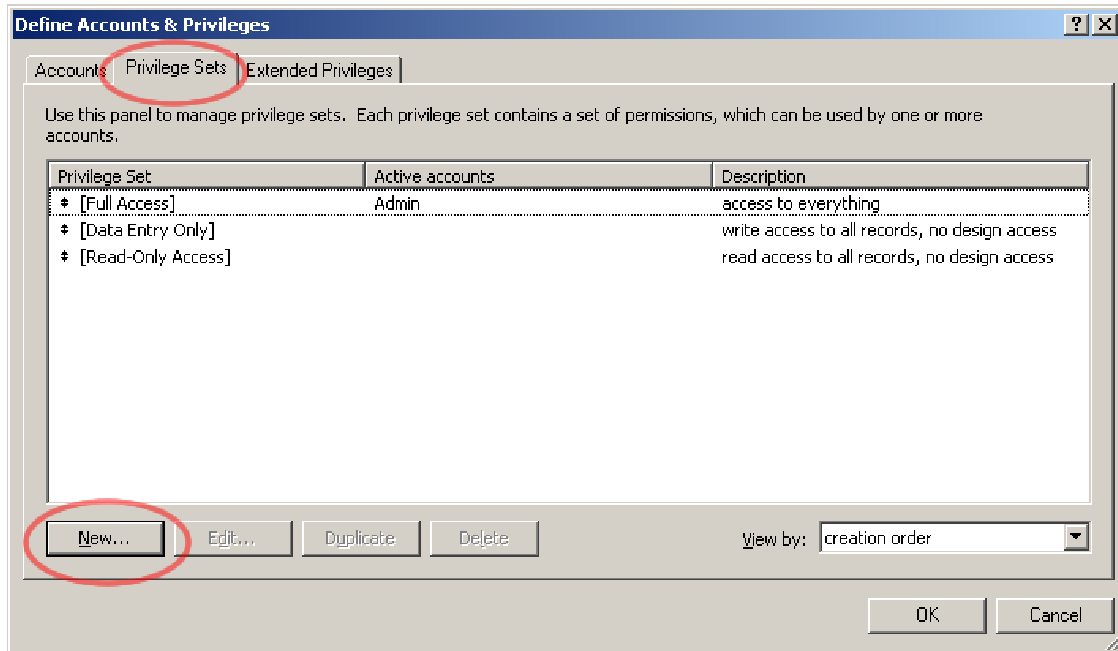


Figure 3-2: Creating a New Privilege Set

Once 'New...' has been clicked a dialog will come up for the privilege set settings. Match these settings to Figure 3-3.

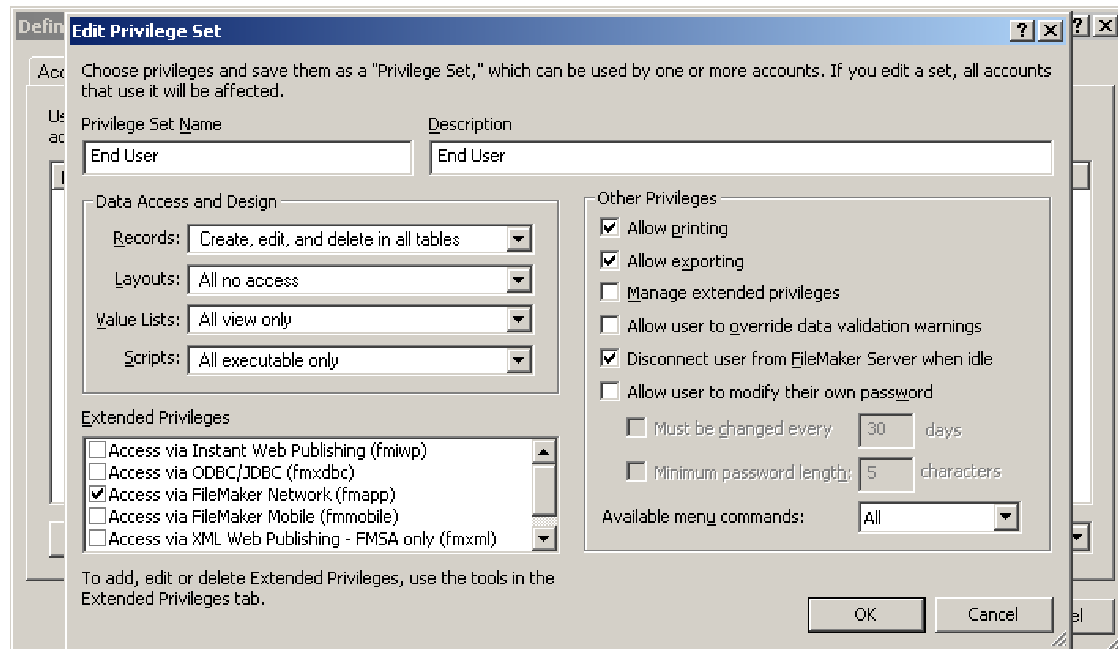


Figure 3-3: New Privilege Set Settings

Once all the settings match Figure 3-3, it is time to customize the layout access preferences. Under the layouts drop down in this same dialog find the 'custom' option and match your settings to Figure 3-4.

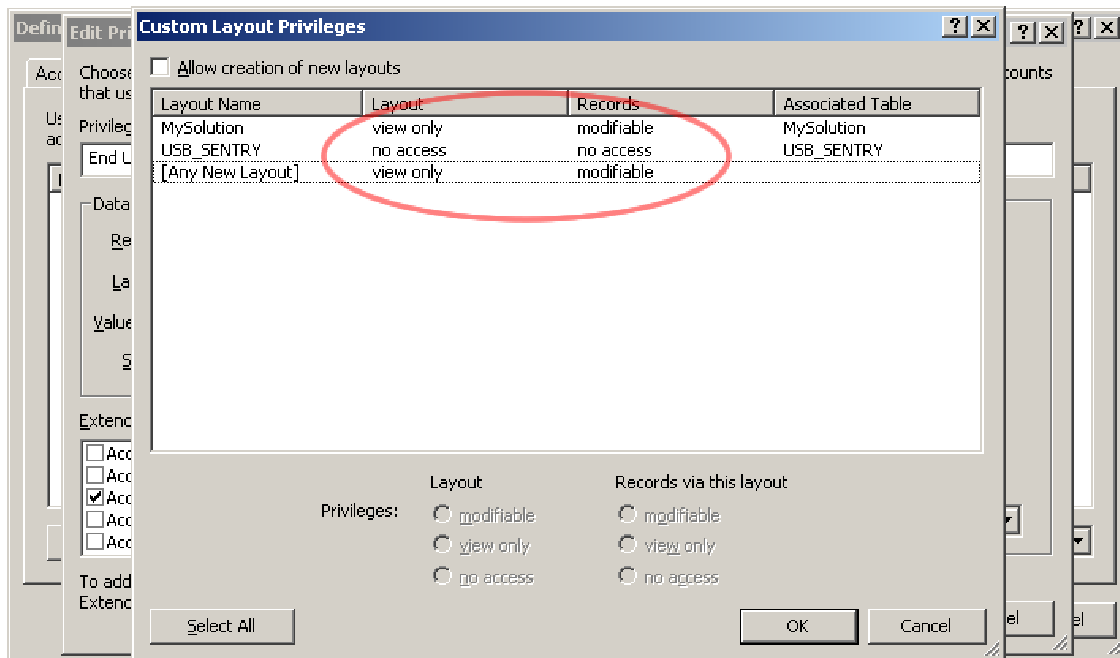


Figure 3-4: Custom Layout Access and Modification Settings

Adding a New User Account

With the privilege set completed, it is time to add a new account. Find the 'New...' button using Figure 3-5 as a reference.

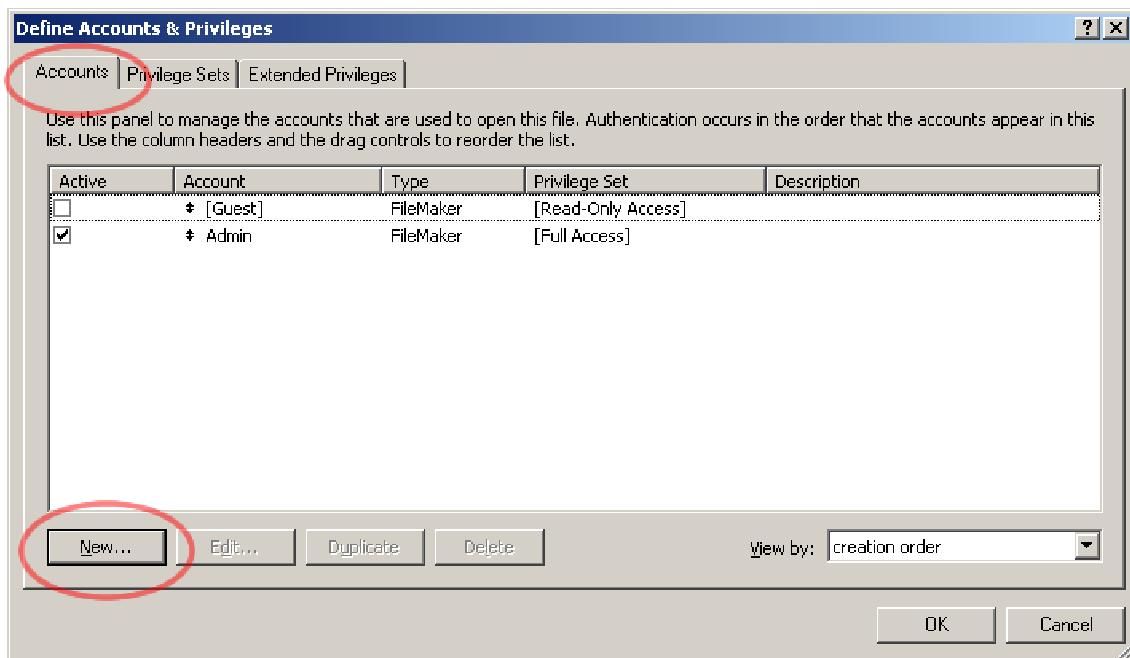


Figure 3-5: Adding a New Account

Once ‘New...’ has been clicked a new account dialog will come up. The user name in this case should be ‘User’, and the password can also be set to ‘User’ at this time – for your own solution you will likely have multiple users at this stage. Select the newly created privilege set for this user just like it is shown in Figure 3-6.

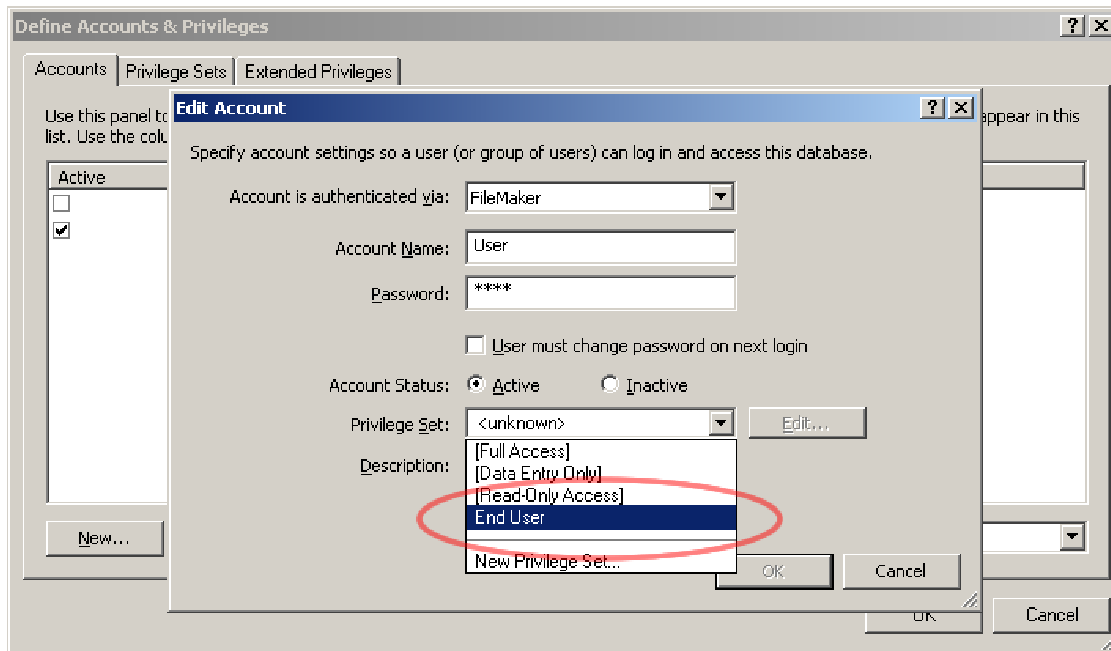


Figure 3-6: Selecting a Privilege Set for the New User

Section 4 – Creating USB Sentry Scripts

Importing Standard USB Sentry Scripts

USB Sentry itself is just a plug-in that returns a list of USB devices, however to use it effectively within FileMaker it should be combined with a bit of FileMaker logic for a smooth user experience. We have supplied a number of scripts that can be easily imported into the FileMaker ScriptMaker, these scripts provide a convenient selector for multiple USB devices as well as a script that checks for the USB device.

The first step to importing these scripts is to open up ScriptMaker, this can be found under the ‘Scripts’ menu and displayed within Figure 4-1.



Figure 4-1: Finding the ScriptMaker Dialog

Once the ScriptMaker is open, select the import button as shown in Figure 4-2.

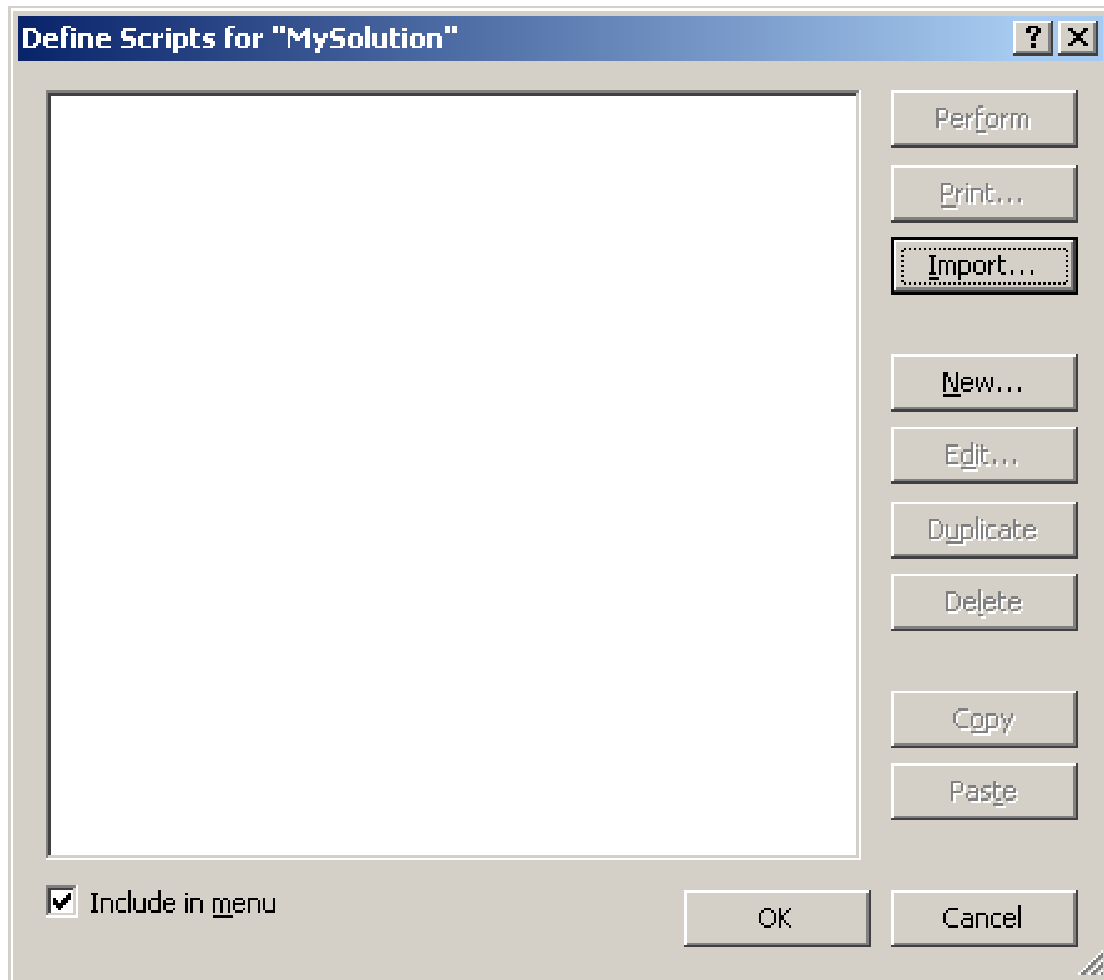


Figure 4-2: Finding the ScriptMaker Import Button

Once the 'Import' button has been pressed, find your USB Sentry distribution files and then under the 'Sample Files' directory find the 'Software Protection' directory. Within this directory you should have 'USB Sentry.fp7', select and open this file just like in Figure 4-3.

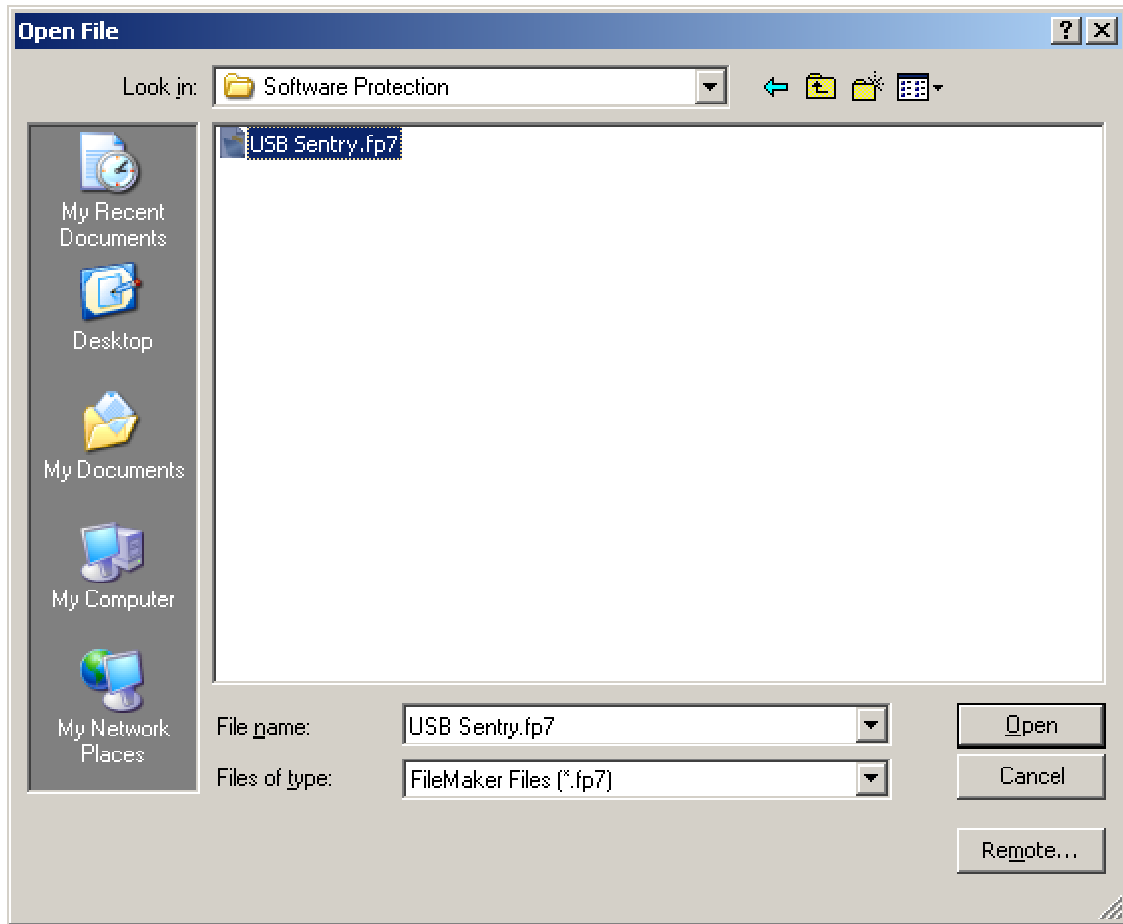


Figure 4-3: Selecting the Sample Files to Import the Scripts From

Once the file has been selected, you should see a list of scripts just like in Figure 4-4. From this list, select the first five scripts and click 'Ok'.

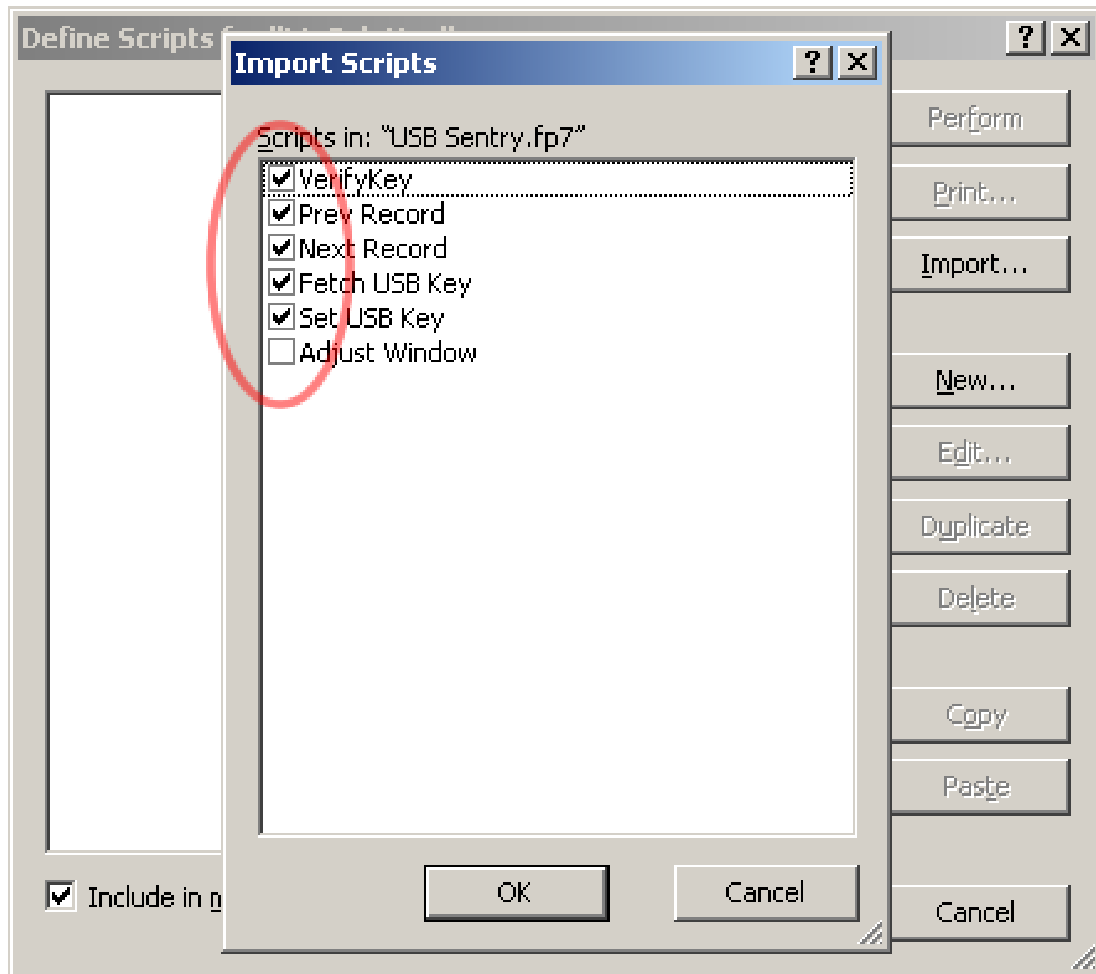


Figure 4-4: Selecting Scripts to Import

Once the scripts have been imported, it is best to set them as unlisted within the scripts menu. Figure 4-5 shows the checkboxes that should be **unchecked** to remove the script names from the menu.

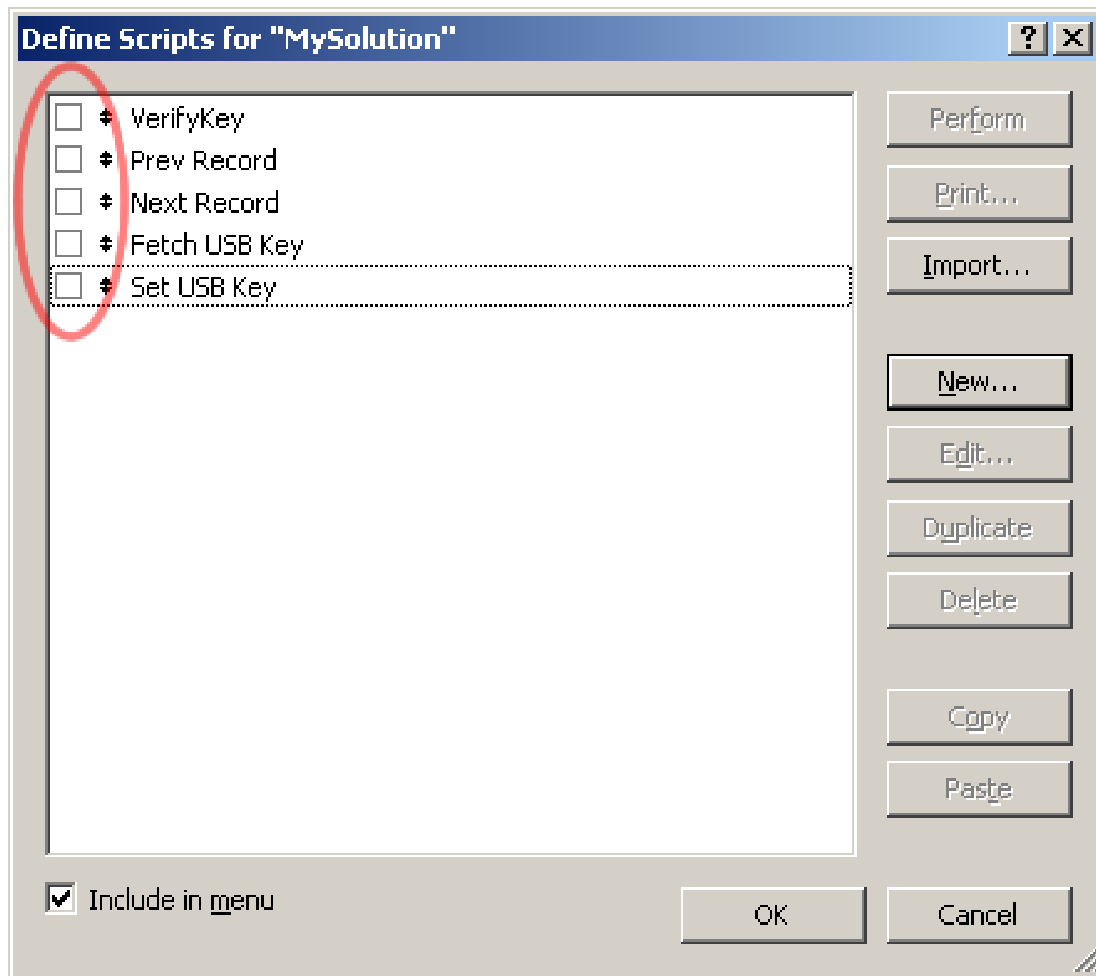


Figure 4-5: Unchecking 'Include in menu' Options

Next we have to give administrative right to two of the scripts. Select the VerifyKey script and check the 'Run script with full access privileges' option. Figure 4-6 displays this process.

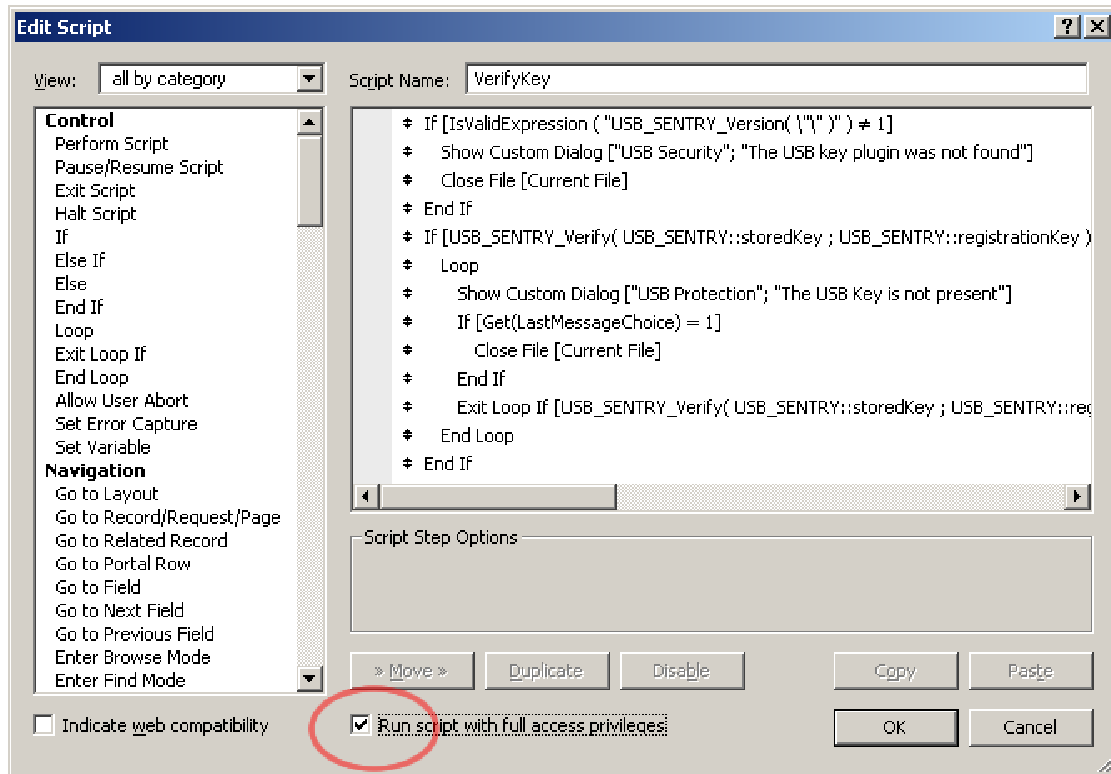


Figure 4-6: Setting the 'VerifyKey' Script to be Ran with Full Access Privileges

Next we repeat the same process for the Fetch USB Key script just like in Figure 4-7.

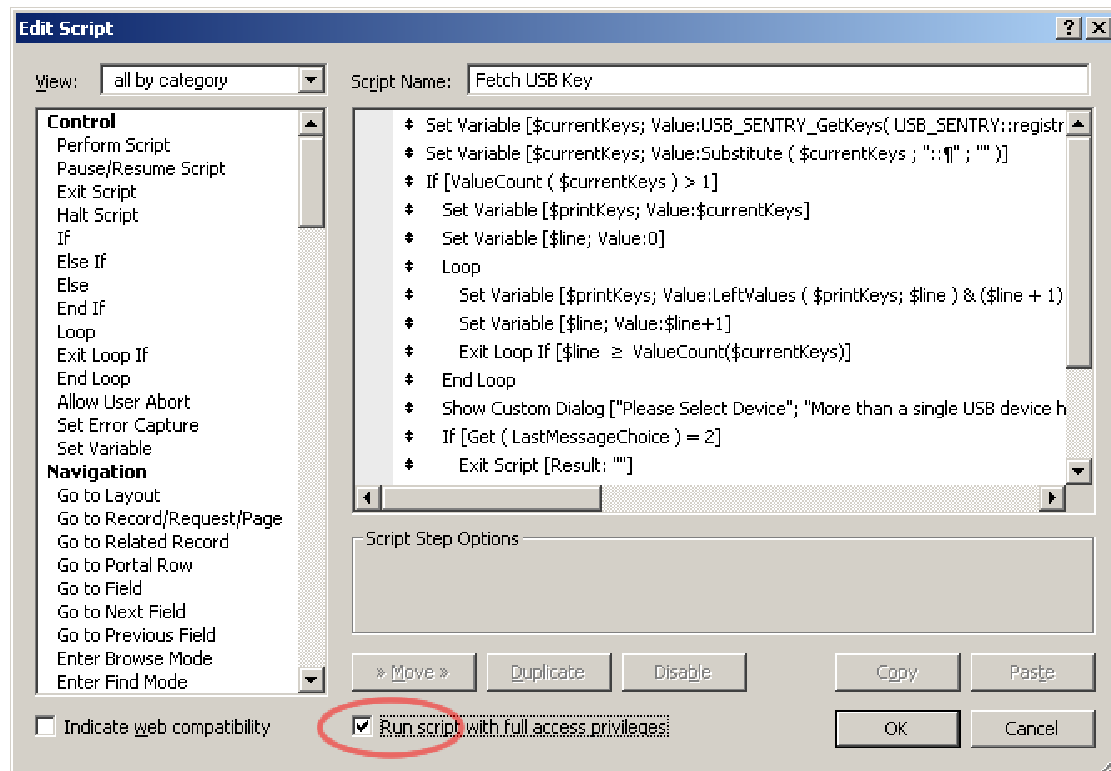


Figure 4-7: Setting the 'Fetch USB Key' Script to be Ran with Full Access Privileges

Once all the steps above are complete, exit the ScriptMaker by pressing Ok.

Adding the Scripts to the Layouts

The last step in this process is to add the scripts to the user layouts in order to verify the USB device at different stages of the application. First enter layout view and go to the USB_SENTRY layout. Within this layout edit the 'REGISTER TO USB DEVICE' button and configure the 'Perform Script' script step to call the 'Set USB Key' script – Figure 4-8 clarifies this process.

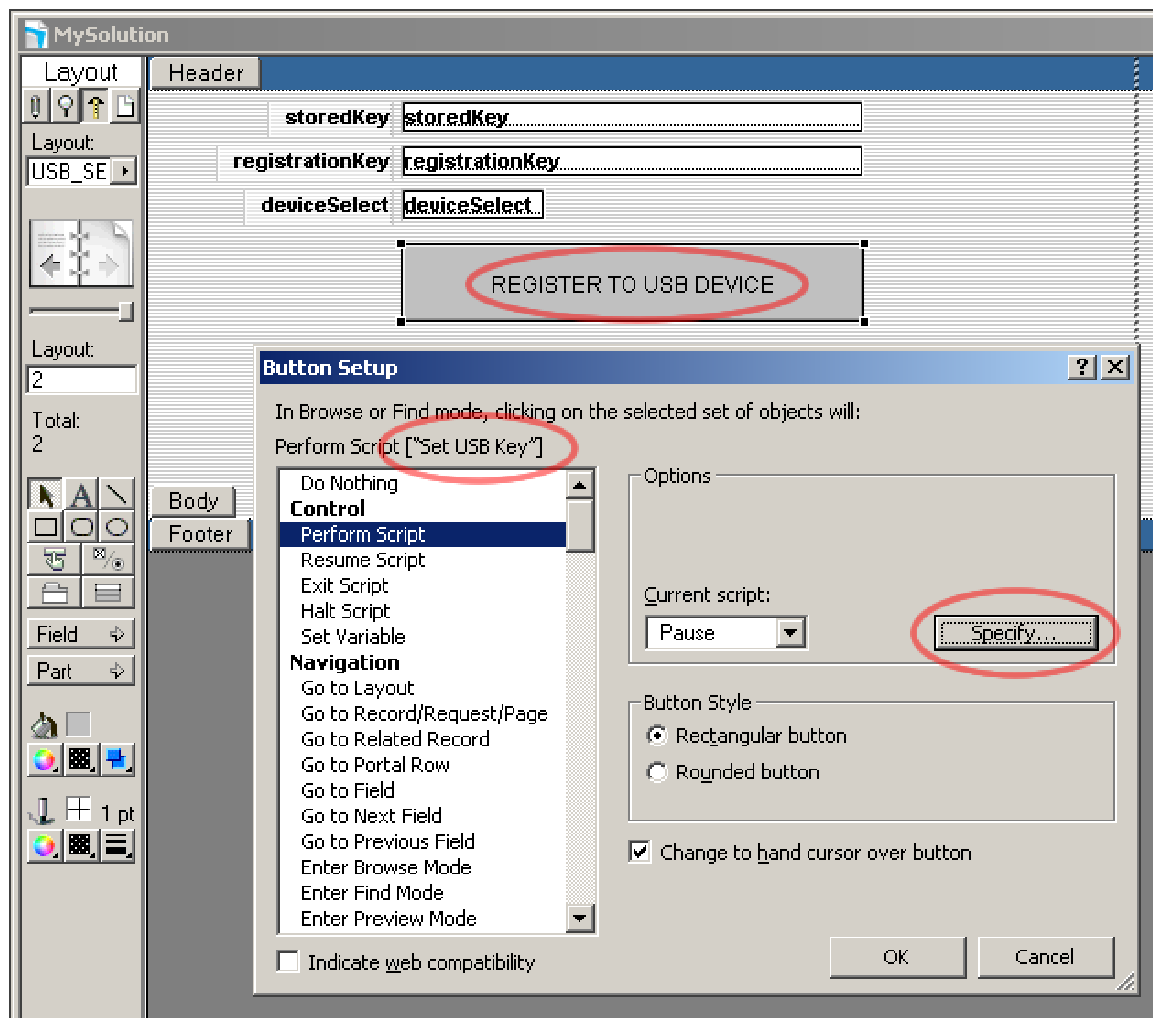


Figure 4-8: Adding the 'Set USB Key' Script to the Button

With this script in place we proceed to the 'MySolution' layout which will have its navigation implemented with checks for the USB device. Add a new button to the layout and add the 'Perform Script' script step to it with the 'Prev Record' set as the script name. Figure 4-9 displays this step. Once you are done, name the button 'Previous'.

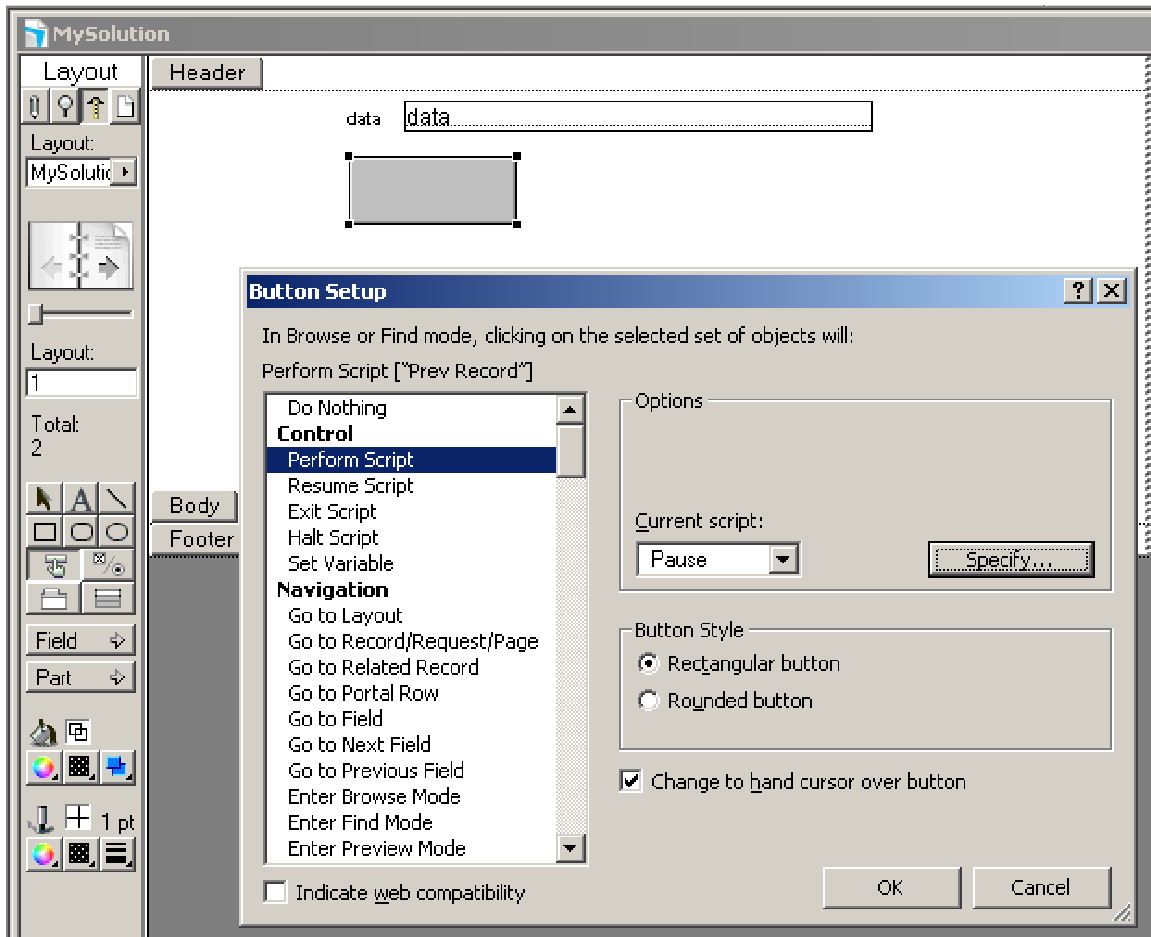


Figure 4-9: Adding the 'Prev Record' Script to the Button

Next create another button, adding the 'Next Record' script to it and calling it 'Next' just like in Figure 4-10.

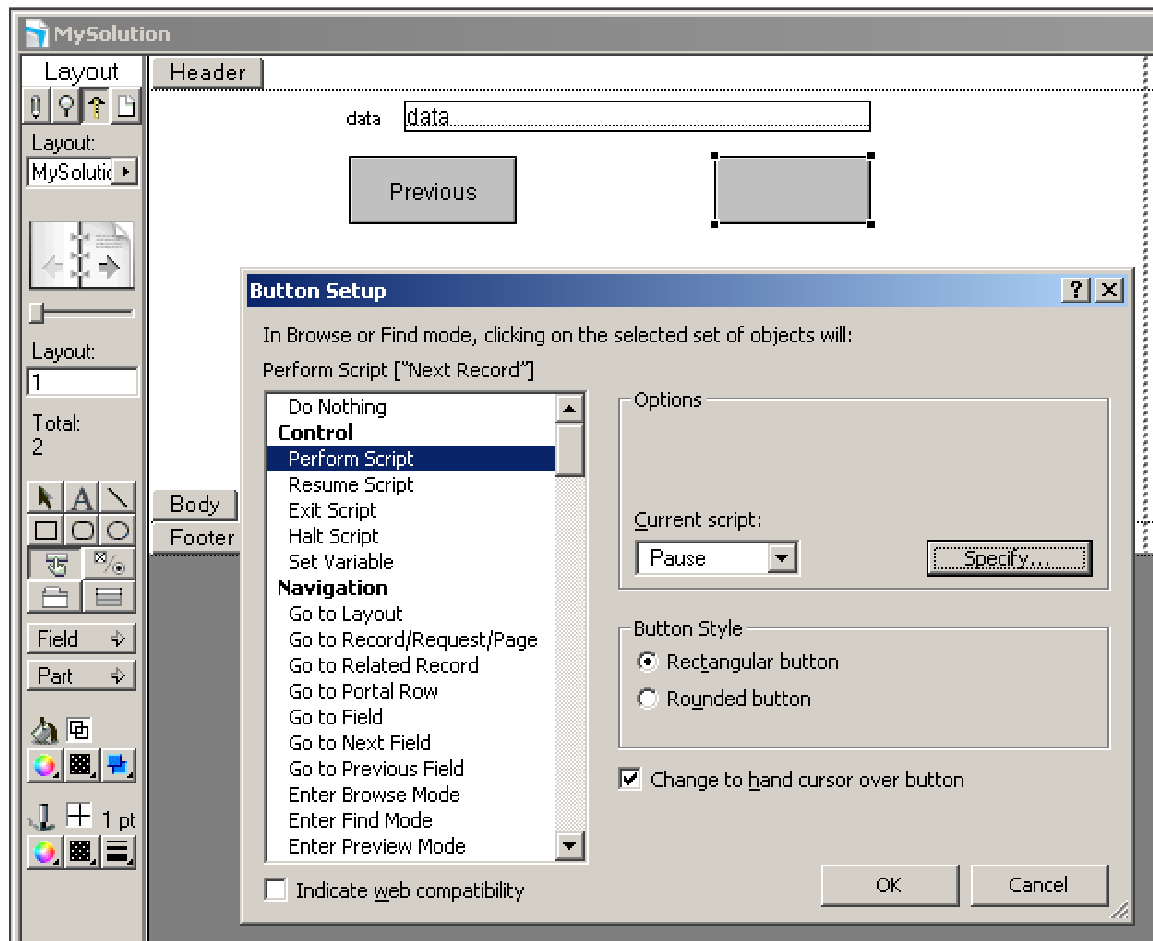


Figure 4-10: Adding the 'Next Record' Script to the Button

Creating the Startup Script

A very important step when protecting your solution is to create a good startup script that locks the interface and the layouts for a regular user and allows them to only use the navigation that you supply. Create this script now by accessing the ScriptMaker once again and then clicking the 'New' button to create a new script. This script should be named 'Startup' with the script steps in Figure 4-11.

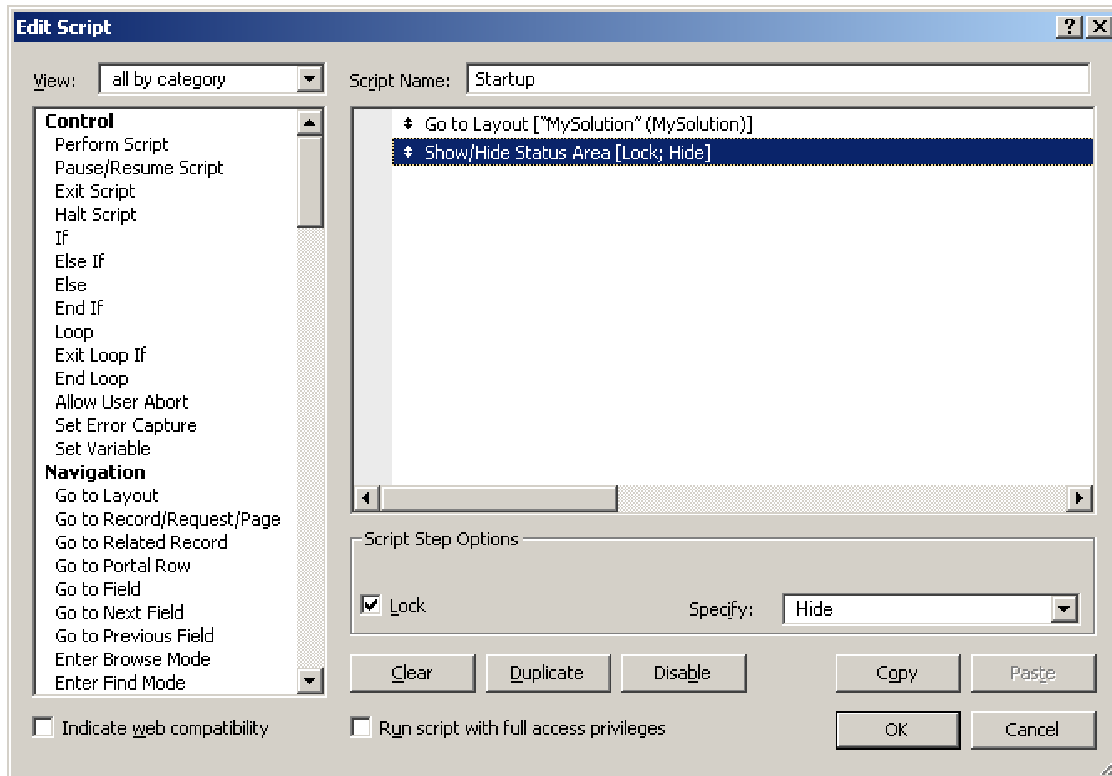


Figure 4-11: Contents of the 'Startup' Script

Adding the Administrative Script

Since the startup script locks the solution, it is important to have a convenient way to unlock it for the developer. We have supplied one such possible script which you can customize to your liking. This script is listed in Figure 4-12. This script is best left within the scripts menu by clicking the checkbox next to it in the ScriptMaker.

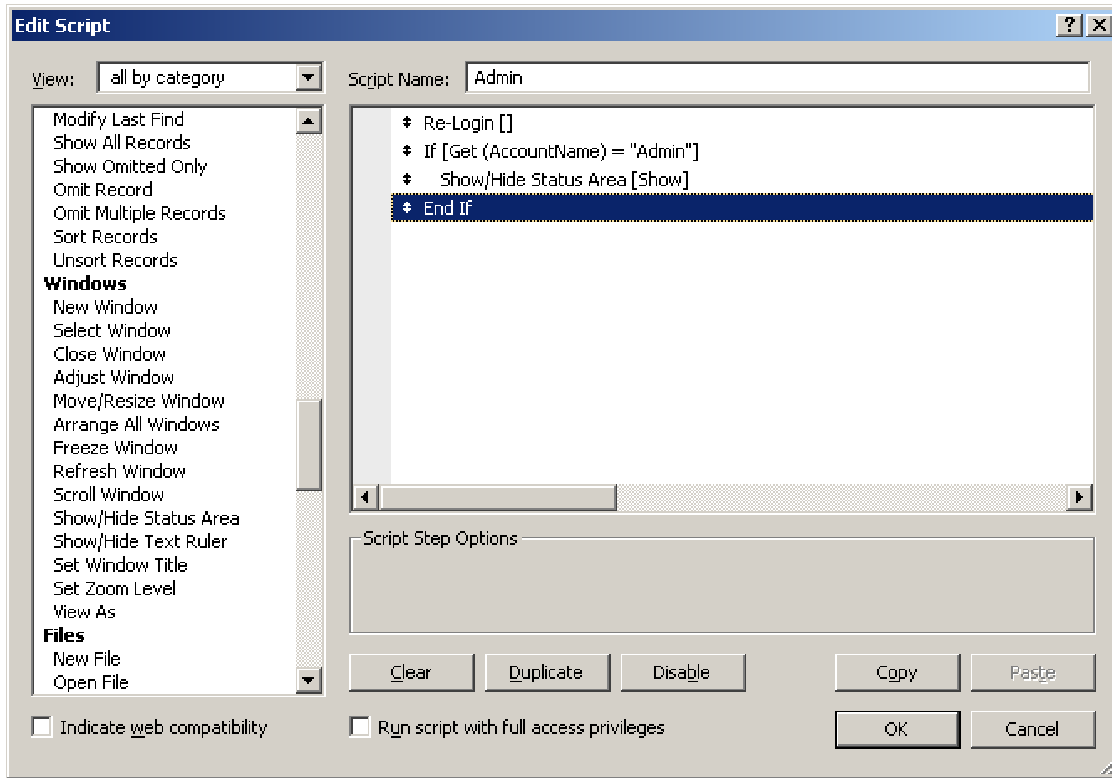


Figure 4-12: The 'Admin' Administrative Script

Setting File Startup Options

The last step is to set the 'Startup' script to be run on startup and a correct layout to be switched into. To do so access the 'File Options...' option from the 'File' menu, use Figure 4-13 as your guide.

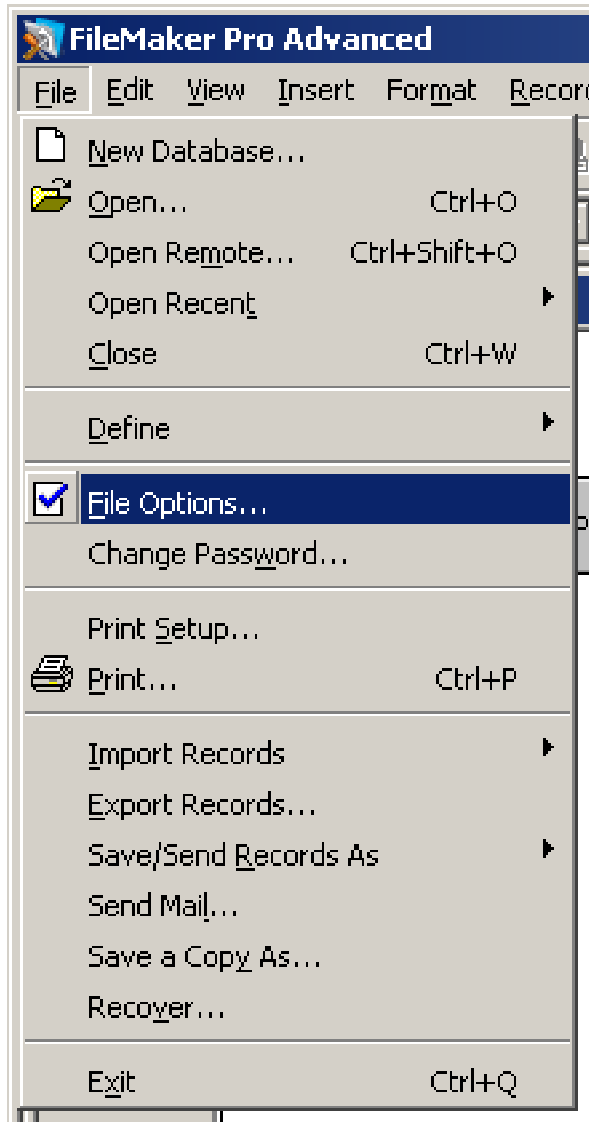


Figure 4-13: Accessing the File Options Section

Next remove the 'Log in using' option to force a user login, or set the user login to the user account if you want to skip authentication. Select a default layout, which in this case is our 'MySolution' layout. Also select the 'Perform script' option, choosing 'Startup' as the script. All of these settings are outlined in Figure 4-14.

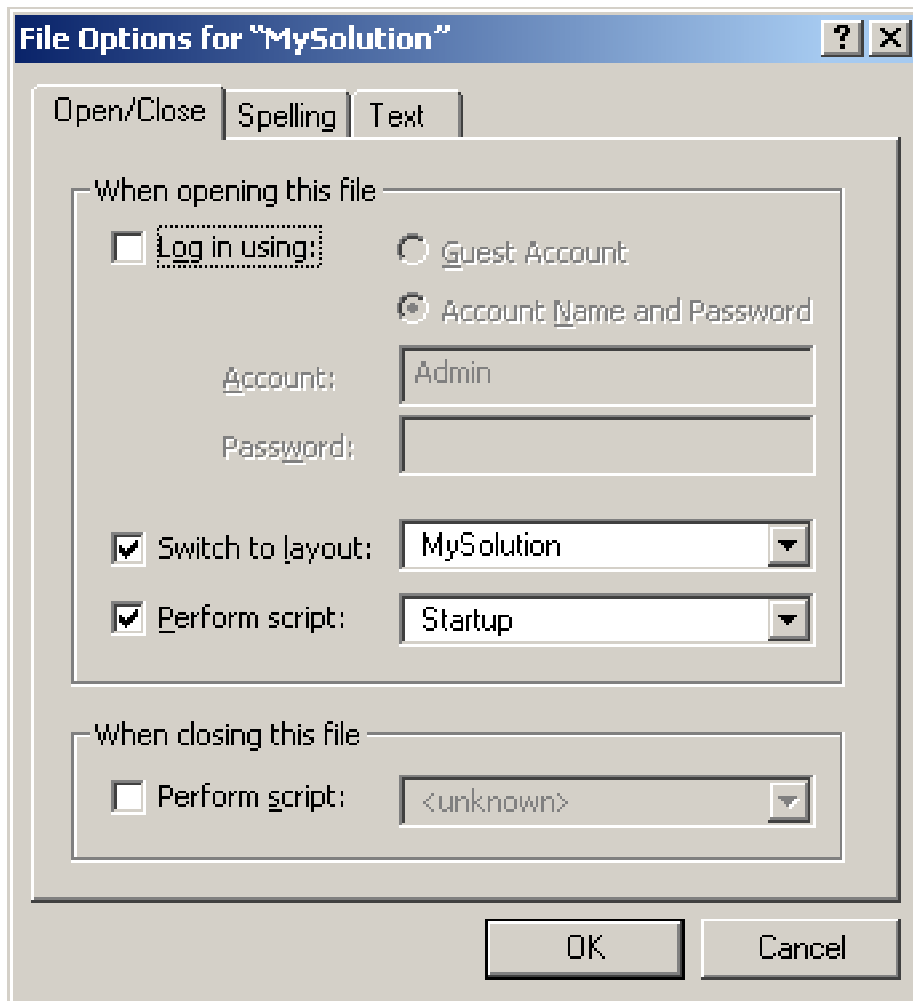


Figure 4-14: Setting the File Options

Section 5 – Testing the Solution and Finishing Notes

The last step is to test how this solution works. We suggest at this stage to add a few records with data to the MySolution table in order to use the next and previous buttons properly.

Registering the Solution to a USB Device

To register a USB device to your solution you must access the 'USB_SENTRY' layout. This is accomplished by **first** going into the 'Layout' mode by clicking the third button in the side bar, and then selecting 'USB_SENTRY' from the list – Figure 5-1 clarifies this process.

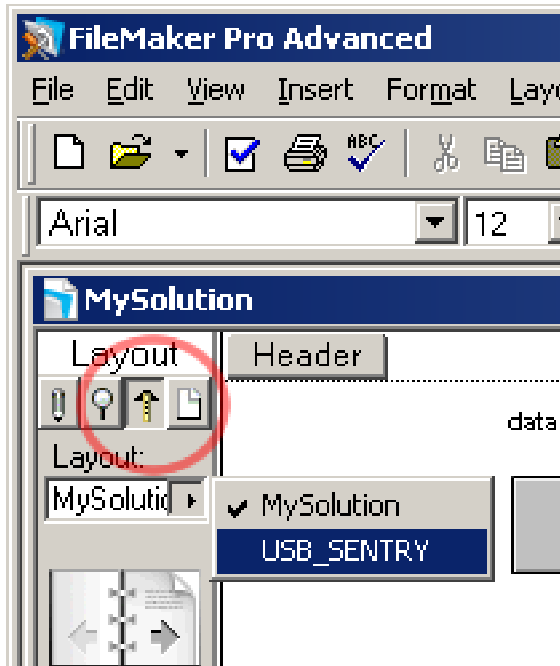


Figure 5-1: Accessing the 'USB_SENTRY' Layout

Next if you have purchased USB Sentry and have a registration key, enter it into the 'registrationKey' area – otherwise leave it blank. Then insert your USB device into the USB port and wait for it to be recognized and scanned by the operating system. Once that is done, click the 'REGISTER TO USB DEVICE' button. If you only have a single USB device then its manufacturer, product, and serial number will appear in the 'storedKey' field – otherwise a dialog will appear asking to select a device. This dialog will list a number of items that can go beyond the end of the dialog – therefore if you do not see all the devices try expanding the dialog window. Figure 5-2 shows the layout after the button was clicked and a device selected.

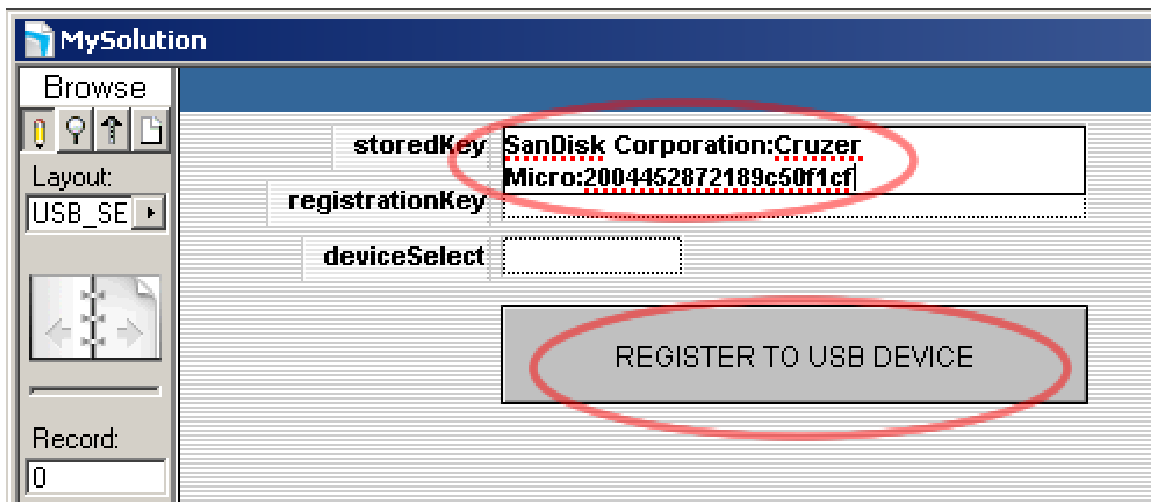


Figure 5-2: USB Device Registered to the Solution

Once the USB device has been registered, go back to the MySolution layout and try using the previous and next buttons – they should be working properly. However try removing your USB device at this time and then using those buttons, a dialog should now come up that will either exit the application or try to access the USB device until it does so successfully. You now have a USB Sentry solution ready!

Ordering Information

To order this plug-in please visit the plug-in home page:

<http://www.fmwebschool.com/usbsentry.php>

Alternatively you may place your order by calling 1-800-353-7950.

Support

Support questions should be directed to our web help forum, located at

<http://www.fmwebschool.com/frm/>. Questions concerning your registration information or order details should be sent to orders@fmwebschool.com

Function Reference

USB_SENTRY_GetKeys function

Prototype

USB_SENTRY_GetKeys(registration key)

Parameters

registration key	Your registration key if the plug-in has been bought, otherwise use empty double quotes for a blank string or 'DEMO'
-------------------------	--

Result

This function returns a list of USB devices or a 'No Compatible USB Devices Found' string. The list of USB devices can be manipulated using the FileMaker Text functions that working with values such as 'GetValue'.

Notes

The registration key differs between Windows and Mac operating systems, please use the 'Get(SystemPlatform)' function to distinguish between the two keys in a cross platform solution.

USB_SENTRY_Verify function

Prototype

USB_SENTRY_Verify(stored key ; registration key)

Parameters

stored key	The stored USB device unique identifier string, this is used as a search string within the list of currently connected devices
registration key	Your registration key if the plug-in has been bought, otherwise use empty double quotes for a blank string or 'DEMO'

Result

This function returns a text value of "TRUE" when the device with the passed-in key is connected or "FALSE" when the device was not found.

Notes

The registration key differs between Windows and Mac operating systems, please use the 'Get(SystemPlatform)' function to distinguish between the two keys in a cross platform solution.

USB_SENTRY_Version function

Prototype

USB_SENTRY_Version(version format)

Parameters

version format	Can be left blank or “short” for a short numeric value of the version. Use “long” for a full plug-in name and version. Use “platform” to get the operating system string.
-----------------------	---

Result

This function returns a text string about the plug-in or the operating system based on the version format parameter.