

SecureFM 6™ Plug-In for FileMaker Pro *with MenuMagic*

User Guide

SecureFM with MenuMagic is really two plug-ins in one. A basic **SecureFM** license gives you a set of core security functions, letting you entirely remove FileMaker menus or selectively disable menu commands, keyboard shortcuts, toolbars, and checkboxes.

With a **MenuMagic** license, you get all of the core security functions PLUS the ability to rename menus and menu items, and override normal FileMaker menu functions with your own scripts. Additional **MenuMagic** functions give you the ability to automatically run a script whenever the front file window changes or when a user enters data into a field.

With **SecureFM**, you can prevent inadvertent data loss or corruption by selectively disabling individual FileMaker menu commands such as Delete Record, Delete All Records, Replace, Close, or Quit, while keeping other essential commands available to the user. With the power of **MenuMagic**, you can create highly customized menus for your solutions.

The **SecureFM** functions are available for Mac OS 9, Mac OS X, and Windows. The **MenuMagic** functions work with Mac OS X and Windows 98–XP.

This is the best solution available to protect your data and customize FileMaker Pro's menus!

Overview of SecureFM with MenuMagic

New Millennium Communications first created SecureFM 1.0 for FileMaker Pro 3.0 in 1997, originally as a system extension on the Macintosh platform. We released a radically enhanced and more flexible version in 1999, SecureFM 3.0 for FileMaker Pro 4.x, as a cross-platform FileMaker plug-in, and added the "Configurator" (now called the Menu Editor). We have maintained it since then to be compatible with each new release of FileMaker Pro and we have added features to improve the degree of security offered.

The new "MenuMagic" features are by far the greatest enhancement that we have made to the plug-in since 1999. The features that are now included are part of the original vision that we had for SecureFM, but it has taken years to get this technology ready for market. We are very excited about this release.

SecureFM 5.0, 5.5, and 6.0 have been able to:

1. Disable individual menu items and their keyboard equivalents
2. Remove entire menus
3. Hide the toolbars
4. Hide the status bar (that useless little bar at the bottom on Windows)
5. Assign a password to your registration so that only the developer can change these menu definitions
6. Create a default menu definition to disable selected menu items as soon as FileMaker is launched, before a file is open

SecureFM with MenuMagic adds the following capabilities:

7. Rename menus and menu items
7. Call your own scripts from menu items instead of the FileMaker function
7. Automatic, file-specific, changing of menu item names and their actions (changed when a window comes to the front... no script!)
7. Call your own scripts from FileMaker's keyboard shortcuts, contextual menus, and toolbar icons!
7. Assign custom hot keys which will call your scripts, using any modifier keys (or no modifiers)
7. Trigger scripts from a calculation, for instance when tabbing from a field
7. Trigger scripts by the event of a window coming to the front (regardless of how it is brought to the front)

You can now create custom controlled user-interfaces in FileMaker Pro with dynamic file-specific menus, without having to control navigation by script. Think about that for a minute...

"New Record" can be "New Customer" or "New Widget" and run your custom script.

"Delete Record" can become "Delete Customer" or "Delete Widget", and can be properly controlled.

"Show All Records" can be "Show My Records"

Control-F can run your custom find script

The toolbar icons and contextual menus can call your custom scripts

AND (trumpet sounds...):

To change files, you don't need to have the user click a button to change your menu definitions. As soon as the file comes to the front, the menu definitions change automatically!

Ease of Deployment with the “Menu Table”

You can easily create your custom menus for each file with the “Menu Editor”, and you can easily deploy your custom menus with the “Menu Table”.

The SecureFM Menu Table file, which is provided with the SecureFM demo files, has no dependencies on other files. You can rename it and put it into your own solution. It is designed to be called whenever you need it and then closed. Most commonly, it is opened during a startup script; selected “menu definitions” are applied (stored in the plug-in's memory) by looping through a found set of records; and then the file is closed.

The Menu Table file is open so you can customize it however you wish.

Contents

OVERVIEW OF SECUREFM WITH MENUMAGIC	2
EASE OF DEPLOYMENT WITH THE “MENU TABLE”	3
CONTENTS	4
INSTALLING SECUREFM	7
REGISTERING SECUREFM	8
Registering SecureFM in the Demo File	
Registering SecureFM in Your Solutions	
MENU-REGISTER	10
Registering with the "Menu-Register" function	
Auto Registering Using SetStartup	
Detecting if SecureFM is Installed	
MENU-VERSION	16
MENU-DISABLE	17
File List	
Applying to All Files	
Disabling Keyboard Commands	
Very Important: Disabling FileMaker's Preferences Command	
Disabling Exit / Quit	
Disabling Window Closeboxes	
Disabling Contextual Menus	
Platform Menu Differences	
Creating Cross-Platform Strings with Status (CurrentPlatform)	
FMP Version Differences	

Branching for Different Versions of FMP

MENU-TOOLBAR	42
MENU-MENUBAR	48
MENU-STATUSBAR	50
MENU-SETSTARTUP	53
MENU-RENAME	56
Renaming Contextual Menu Items	
MENU-SCRIPT	60
Scripting Contextual Menu Items	
Scripting Closeboxes	
Scripting Toolbar Buttons	
MENU-HOTKEY	63
STRING PREFIXES AND DIFFERENT TYPES OF MENU DEFINITIONS	69
Default Menu Definition	
File-Specific Menu Definitions	
RESTORING MENUS AND SPECIAL SETTINGS	72
No Files Open Menu Definition	
Restoring Menus	
Clear All Menu Definitions	
Clear Current Setting	
OTHER SECUREFM FUNCTIONS	76
MENU-STORETEXT	76
MENU-GETTEXT	77
MENU-GETMENUSELECTION	79

MENU-DOFMPMENUITEM	80
MENU-EVENT	81
WindowActivate	
ScriptError	
SFMDisable	
MENU-PERFORMSCRIPT	86
Triggering a Script on Exiting a Field	
Triggering a Script on Navigating to a Layout	
MENU-GETSCRIPTCALLER	88
MENU-EXTRACTPARAMETER	89
MENU-ISFRONTWINDOW	91
MENU-FMPWINDOW	93
SECUREFM'S PASSWORD FEATURE	94
USING SECUREFM WITH RUNTIME SOLUTIONS	95
TROUBLESHOOTING	97

Installing SecureFM



Installation on Macintosh:

Place the SecureFM plug-in into the FileMaker Extensions folder inside your FileMaker Pro folder.



Installation on Windows:

Place the SecureFM.fmx plug-in into the FileMaker System folder inside your FileMaker Pro folder.

Remove old versions: Do not forget to remove any older versions of **SecureFM** that were previously installed. The old plug-in may not be automatically replaced. Having more than one version of **SecureFM** can cause problems.

Restart FileMaker after installing the plug-in.

Make sure SecureFM is selected in the Plug-Ins section of the Application Preferences (found in the Edit Menu).

Installation Note: The plug-ins for Windows, Macintosh OS 9, and Macintosh OS X are each different. Please make sure you are using the correct version for your platform.

Important: OS Compatibility and Security

Beginning with versions 6.3.3/7.0.3, SecureFM with MenuMagic is designed to work with the latest versions of Windows XP and Mac OS X (10.3.6).

Although SecureFM 6.3.3 generally will work correctly with earlier versions of the Mac OS X and Windows, it is only thoroughly tested for security and functionality on the latest OS versions. If you want to use SecureFM on systems running older versions of Windows and Mac OS X, you may need to use SecureFM 6.3.1/6.3.2. Always test essential functions first, especially when deploying on systems that are not up-to-date!

If you are not able to control which version of the system software your SecureFM-protected files will be run on, you may want to test for system version in your scripts.

Registering SecureFM

Registering SecureFM in the Demo File

To register **SecureFM** enter your user name and license key string into the Registration String field provided in the registration area of the SecureFM Menu Editor (included with demo). If you have not yet downloaded the current version of the plug-in and associated files from our website, please do so.

<http://www.newmillennium.com>

Once you have entered your user name and license key, click the Register button on SecureFM Menu Editor's Register layout. You will know that the plug-in has registered properly if the Registration Pass/Fail field contains the word: Registered.

Registering SecureFM in Your Solutions

You can register **SecureFM** in your own FileMaker Pro solutions by using the external "Menu-Register" function in a script. The registration script step can be incorporated into any script in your solution, as long as it is called before you intend the menus to be modified. Therefore it is commonly put into an "On Open" script that is automatically run every time your solution's main file is launched.

You can use a simple Set Field script step to register **SecureFM**. In the example below we are performing the registration Set Field step into a text field named SFM Response. The registration syntax is:

External ("Menu-Register", "My Company|MACCODE|WINDOWSCODE|Optional Password")

The code for the Macintosh platform must appear after the first pipe separator. The code for the Windows platform must appear after the second pipe separator. Even if you are only using the registration code for a single platform, you must still include both pipe separators in the parameter - but only the code for the current platform is checked.

Please note that the user name and license key are case-sensitive.

For a more complete discussion of how to incorporate the registration process into your solutions, see the Menu-Register function discussion of the SecureFM Documentation.

Important Registration Note:

Registering **SecureFM** once does not mean it is permanently registered! It is only registered until FileMaker is shut down. **Each time FileMaker is launched SecureFM needs to be re-registered.** This function enables you to test in your solution to make sure that SecureFM is installed.

SecureFM expires after 60 minutes if not registered. After that point, FileMaker menus are no longer disabled, and the user is warned that an unregistered version of **SecureFM** is in use.

Menu-Register

External ("Menu-Register", parameter)

SFM Available with a SecureFM or MenuMagic license.

The parameter must include the user name and the registration code for either the Mac or Windows platform (or both). You can also add an optional password for additional security (more about this function later). Each of the pieces of information are separated by a pipe "|" character:

"My Company|MACCODE|WINDOWSCODE|Optional Password"

The code for the Macintosh platform must appear after the first pipe separator. The code for the Windows platform must appear after the second pipe separator. Even if you are only using the registration code for a single platform, you must still include both pipe separators in the parameter - but only the code for the current platform is checked.

Please note that the user name and license key are case-sensitive.

SecureFM expires after 60 minutes if not registered. After that point, FileMaker menus are no longer disabled, and the user is warned that an unregistered version of SecureFM is in use.

Important Registration Note:

Registering SecureFM once does not mean it is permanently registered! It is only registered until FileMaker is shut down. **Each time FileMaker is launched SecureFM needs to be re-registered.** This function enables you to test in your solution to make sure that SecureFM is installed.

Two Levels of Registration: SecureFM and MenuMagic

The SecureFM license you use determines whether only the core "SecureFM" security functions or all "SecureFM with MenuMagic" functions will be available.

The basic 'SecureFM' registration key activates the essential security and registration functions:

- Menu-Register
- Menu-Version
- Menu-Disable
- Menu-Menubar
- Menu-Toolbar

Menu-Statusbar
Menu-SetStartup
Menu-StoreText
Menu-GetText
Menu-IsFrontWindow

The 'SecureFM with MenuMagic' registration key activates all of the above functions, along with the rest of the menu and file control functions:

Menu-Rename
Menu-Script
Menu-HotKey
Menu-Event
Menu-PerformScript
Menu-GetScriptCaller
Menu-DoFMPMenuItem
Menu-ExtractParameter

If you are running with a SecureFM registration key, the MenuMagic functions will be active in demo mode for a limited period of time – 1 hour from launch.

You can register SecureFM in two ways:

- 1)** You can use the "Menu-Register" function every time your solution is opened, such as in an On Open script, or
- 2)** You can use the "Menu-SetStartup" function to store the registration as part of FileMaker's default configuration. To do this, you must first register SecureFM (as well as make sure any other "Menu-Disable" settings you want are also active) and then call the "Menu-SetStartup" function. The registration code as well as the default Menu-Disable setting will be stored in the "SecureFM.ini" file, and SecureFM will automatically re-register itself every time FileMaker is started up.

Registering with the "Menu-Register" function

You can register SecureFM by calling the external "Menu-Register" function, usually from a script, though it can also be done from a calculation field or within a validation by calculation. You can include a registration script step in any script in your solution, as long as it is called before you intend the menus to be modified. Therefore it is commonly put into an "On Open" script.

The syntax in the script is:

External ("Menu-Register", "My Company|MACCODE|WINDOWSCODE|Optional Password")

To set a script as the file's On Open script, go into FileMaker's Edit: Preferences: Document... menu. Make sure "Perform Script" is selected under 'When Opening "FILENAME"' in the General Document Preferences, and select your On Open script in the pull-down menu to the right.

If you have a multiple file solution, you must make sure during the opening process that at least one of your files will open and properly register SecureFM in this manner.

Auto Registering Using SetStartup

This method of registering SecureFM is helpful if you want the menus to be modified as soon as FileMaker is launched, but you can't be sure which file will be opened first. It can also be essential if you don't want to allow access to modifying a script in a file. It is not as secure as putting the Register step into a script, however.

To use this method, register SecureFM once via script using the correct Licensee Name and Registration Codes. (This can be done from the SecureFM Menu Editor file.)

Create a default SecureFM preference file using SecureFM's "Menu-SetStartup" command. (Again, this can be done using the SecureFM Menu Editor file. See the section below regarding SecureFM's "Menu-SetStartup" for more details.)

In the new preference file ("SecureFM.ini"), SecureFM will store all registration information (as well as the default Menu-Disable string) and will re-register itself automatically when FileMaker is started up. The preference file has the suffix ".ini" and must be placed in the FileMaker Extensions file for Mac or FileMaker's System directory for Windows. Place this file into the appropriate folder, along with SecureFM itself, on each computer.

If your Registration String contains both the Mac and Windows Registration Codes, the preference file will be cross-platform and can be copied to either type of platform across a network, as long as your Menu-Disable string has no differences between platforms (see "Menu-SetStartup" below for more details).

Detecting if SecureFM is Installed

The "Menu-Register" call can be used to detect whether SecureFM is installed, as it always returns one of the errors or confirmations below:

- "Registered SecureFM" & the version number,
- "Invalid Registration SecureFM" & the version number,
- "Expired Registration SecureFM" & the version number,
- "Incorrect Password SecureFM" & version number, or
- blank ("") if the plug-in is not installed.

Re-Registering in the Same FileMaker Session

If you try to register SecureFM with an empty string, it returns the registration status, without altering the current registration. This can be helpful for checking whether SecureFM has already been registered.

Thus, if you have already registered SecureFM in the same FileMaker session, and then send the command:

External ("Menu-Register", "")

SecureFM will NOT alter the registration status. Instead, it will return the current registration status.

If, however, you include any data in the registration string, SecureFM will evaluate the new registration string.

Checking for Plug-In Version

See the documentation on the Menu-Version function for a discussion of how to easily check the plug-in's version number.

Important Registration Security Note

Any invalid registration string will void the current registration UNLESS the original registration uses a password. When registered with a password, SecureFM will not upgrade, downgrade, or unregister if an invalid registration string is sent or if a registration string without the correct password is sent. (See below for information on registering with a password.)

If security is an important part of your SecureFM implementation, then you should always register with a password!

Password Registration

SecureFM's Password is an advanced feature that gives you an added layer of security, allowing you to ensure that only the solution creator can control FileMaker's menus and toolbars.

Using a password is OPTIONAL (but strongly recommended for security). This makes SecureFM backward compatible with plug-in commands designed to run on earlier versions of SecureFM. It also means you can simply choose not to add a password to your solution if you feel that this advanced level of security isn't necessary for your solution.

Password for Added Security

Registering SecureFM with a password prevents users from re-configuring the menus unless they know the password. When SecureFM is registered with a password, that password must also be used when setting all menu configurations and toolbar access.

If you don't use a password, a FileMaker-skilled user who has script access to any FileMaker file and understands the syntax of SecureFM could enable all menu items. Even a novice user who has access to the SecureFM Menu Editor file (which can be downloaded by anyone from New Millennium's web site) can simply hit the "Restore All Menus" button. Registering SecureFM with a password prevents these possibilities.

Ex: Registering With a Password

Registering with a password uses the following format, with the optional password as the fourth part of the parameter:

```
External ("Menu-Register", "My Company|MACCODE|WINDOWSCODE|Optional Password")
```

The password parameter can be up to 100 characters long.

Ex: Registering With No Password

Since the password is entirely optional, the password can be left blank when registering SecureFM. If no parameter is used, the third pipe character "|" following the Windows registration code must be left out.

External ("Menu-Register", "My Company|MACCODE|WINDOWSCODE")

Using SecureFM with a Password

After SecureFM has been registered with a password, all subsequent commands which have a password parameter (Menu-Disable, Menu-Rename, Menu-Script, Menu-Toolbar, etc.) will require the password as well; otherwise, an "Incorrect Password" error will be returned.

This requirement also applies to the Menu-Register function. Once SecureFM is registered with a password, you can only re-register using the same password. This prevents a user with script access from removing or changing the password. Once set, the only way to remove or change the password is by restarting FileMaker.

Password Protection

In our SecureFM demo files, we store the password in a field because that is the simplest way to demonstrate how the function works. However, storing sensitive data like passwords and license codes in FileMaker fields is not very secure. For the greatest security, this sensitive data should be manually entered into the script steps themselves, and not held in fields.

Menu-Version

External ("Menu-Version", "")

SFM Available with a SecureFM or MenuMagic license.

The Menu-Version function returns just the version number of SecureFM without any other text. This makes it easy to determine the version of SecureFM currently in use.

External ("Menu-Version", "")

No parameter is used.

The response returns only SecureFM's version number, such as:

6.3.0

Menu-Disable

External ("Menu-Disable", parameter)

SFM Available with a SecureFM or MenuMagic license.

This is the primary SecureFM command. It allows you to specify which FileMaker menu items (or entire menus) you want to disable. Menu items can be disabled for all files by using a "default" parameter or different menu items can be disabled for different files.

Disabling Menus and Menu Items

SecureFM recognizes specific menus and specific items in those menus, not by name, but by number. Menus are counted left to right, from File (1) to Help (9). (The Arrange menu, which is available only in Layout mode, is considered to be menu number 10.) The individual menu items are counted top to bottom.

A simple disable string such as--

<1,B,<2>>

--is read as--

1 - FileMaker menu affected. Counting left to right, the File menu is the first FileMaker menu.

B - Mode affected. In this case, we are affecting the Browse mode. You can specify one of the following five options for this middle clause:

B = Browse mode
F = Find mode
L = Layout mode
P = Preview mode
A = All modes (see notes below)

2 - Menu item(s) to be disabled. In this case, the second item from the top in the File menu is the "Open..." command.

File List

The string must also include a prefix that tells SecureFM which files should have their menus affected.

Strings generally follow the format:

<File List>
<Menu String>

To disable a menu item in two files named “Customers.FP5” and “Customer Items.FP5” the string would look like:

<Customers, Customer Items>
<1,B,<2>>

Note that the “.FP5” file name extension is not used in the file list.

Applying to All Files

You can also apply a menu configuration globally to all files by using an asterisk “<*>” instead of a list of file names:

<*>
<1,B,<2>>

By using a default menu definition, you can enforce security globally on new files, without needing to know the name of every file that might be opened.

Menu Item Numbers and Separators

Be aware that the SEPARATOR LINES appearing in the menu pull-downs ARE COUNTED as menu items. Thus, if you want to disable the "Cut" command in the Edit menu, you must disable item THREE, not item two, because the separator is item two:

<*>
<2,B,<3>>



A Simple Example - Disabling a Single Item

The parameter (or "configuration string") is grouped into three clauses, each enclosed in <> ("greater than" and "less than" brackets). Here is a very simple example:

```
<*>:
<1,B,<2>>
```

This configuration string tells SecureFM to disable the "Open..." command in FileMaker's File menu when in Browse mode.

Disabling Keyboard Commands

When you disable a menu item that can be called by keyboard command -- for example "Close" and Command-W on Mac or Ctrl-W on Windows -- SecureFM will also automatically disable the associated keyboard command(s). Go ahead and try:

```
<*>:
<1,B,<3>>
```

Another Example - Disabling Multiple Items

If you want to disable more than one item within a menu, simply separate the item numbers by commas. For example:

```
<*>:
<1,B,<1,2,3,4>>
```

SecureFM interprets this as: In the File menu (the first menu from the left), when in Browse mode, disable "New Database...", "Open...", "Open Remote...", and "Close" (the first, second, and third items).



Meaningless Items in Strings

There are several menu items which FileMaker does not enable in certain modes (they are always gray). You can configure your string to disable them with SecureFM, but it will have no additional effect since they aren't selectable in the first place. You can even configure your string to disable separators, but it will also have no effect.

A More Complex Example - Disabling Multiple Menus

If you want to disable items in more than one menu, simply create a different configuration string for each menu. (You may want to separate the string for each menu with a paragraph return to make the string easier to read and edit, though this is not required). For example:

```
<*>:  
<1,B,<1,2,3>>  
<7,B,<1>>
```

With the above string, SecureFM will disable the first three items of the File menu and also the first item of the Script menu ("ScriptMaker...").

You can add as many items to the string as necessary:

```
<*>:  
<1,B,<1,2,3>>  
<3,B,<3>>  
<6,B,<1,2,3,5>>  
<7,B,<1>>
```

Free Form String Formatting

You can freely add spaces, tabs, and returns to your strings for the purpose of readability.

Ex: Disabling for More than One Mode

Use the same format above to disable menu items in more than one mode. For example, suppose you want to disable the first three items in the File menu for both Browse and Layout modes, your string will look like this:

```
<*>:  
<1,B,<1,2,3>>
```

```
<1,L,<1,2,3>>
```

Also note, the items disabled do NOT have to be identical in different modes. You can, for example, disable all three of the above items in Browse mode, but only disable the "New Database..." command in Layout mode. Your string would then look like this:

```
<*>:  
<1,B,<1,2,3>>  
<1,L,<1>>
```

Ex: Disabling the Same Item in ALL Modes

If you want to disable the same item or items in ALL four modes, instead of four lines in the string (one for each mode), you can use the <A> clause instead. For example:

```
<*>:  
<7,A,<1>>
```

Will disable the "ScriptMaker..." command in the Script menu in all four modes. It has exactly the same effect as:

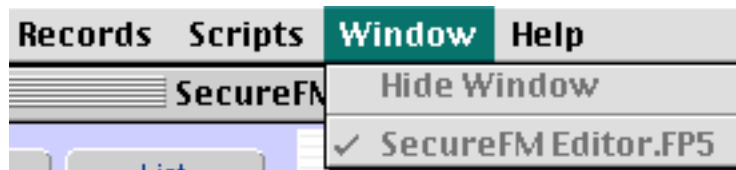
```
<*>:  
<7,B,<1>>  
<7,F,<1>>  
<7,L,<1>>  
<7,P,<1>>
```

Important Note About the "<A>" Clause: Be cautious when using the disable in all modes "<A>" command. SecureFM will disable that item number in all modes, EVEN IF THE ITEM IS DIFFERENT IN ANOTHER MODE. For example, in the Records/Requests/Layouts menu, item number 8 is "Show All Records" in Browse and Preview, but it is "Perform Find" in Find mode and "Set Layout Order..." in Layout mode. Using the string—

```
<*>:  
<6,A,<8>>
```

—will disable ALL of those menu items!

Ex: Disabling an Entire Menu



If you want to disable all items in a menu, rather than individually listing each item number, you can use a zero ("`<0>`") in the items brackets. For example, suppose you want to completely disable the Window menu (in all modes) to prevent the user from bypassing your solution's navigation buttons. Your string would look like this:

```
<*>:  
<8,A,<0>>
```

The Window menu will still pull down but all items will be grayed out.

Ex: Removing an Entire Menu



If you want to completely remove the menu from FileMaker's menu bar, you can use a negative one ("`<-1>`") in the items brackets. (Note: this function is not available on OS X.) For example--

```
<*>:  
<8,A,<-1>>
```

--will cause the Windows menu to disappear from FileMaker's menu bar, until you choose to re-enable it.

To remove the Window menu in all modes except Layout mode, use:

```
<*>:  
<8,B,<-1>>  
<8,F,<-1>>  
<8,P,<-1>>
```

This can be an especially useful configuration, particularly when combined with an On Open or Startup string which disables the Layout Mode menu item, and a hidden button which re-enables Layout Mode. This enables a developer to work on a solution at a user's desk without having to quit and relaunch with a different password.



Removing Menus on Windows

When you remove a FileMaker menu on the Windows NT, ME, 2000, or XP environments, a small space of 15 pixels is created in FileMaker's menu bar in place of the removed menu.



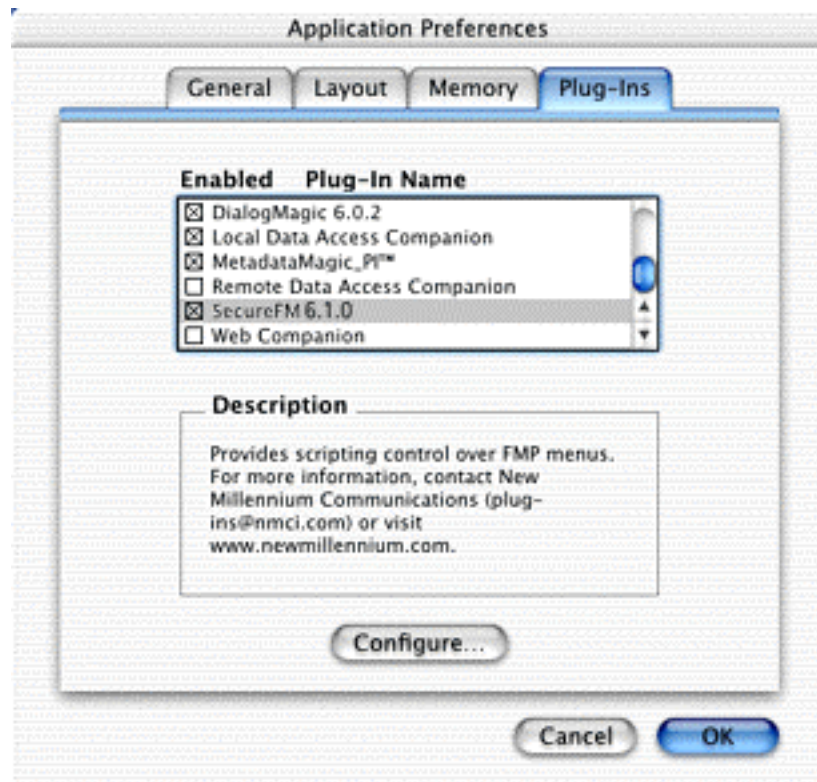
Menus Can Only be Disabled, Not Removed on Mac OS X

Attempting to remove a menu on OS X will result in the menu being disabled instead. You can still use the remove command, for example "`<8,B,<-1>>`", but the menu will be grayed out rather than removed, as if you were using the string "`<8,B,<0>>`".

You can, however, use the Menu-Menubar command to completely remove the FileMaker Pro menu bar, if you wish. (See the documentation on Menu-Menubar for more information.)

Controlling Specific Functions

Very Important: Disabling FileMaker's Preferences Command



FileMaker gives users a very simple way to disable a plug-in: In the Edit menu (or the “FileMaker Pro” Application menu in OS X), select Preferences: Application: Plug-Ins; then uncheck whichever plug-ins you want. If a user does this, the plug-in immediately ceases to function. **If this is done with SecureFM, all disabled FileMaker functions are IMMEDIATELY RE-ENABLED!**

You can prevent a user from doing this by disabling Preferences in the Edit menu.



The cross-platform string to do this in FileMaker Pro 6 is:

For FileMaker Pro 5.5 and 5.0:

```
Case (
Status (CurrentPlatform) = 1,
"<*>:
<2,B,<13>>
<2,F,<13>>
<2,L,<14>>
<2,P,<13>>",
```

```
Status (CurrentPlatform) = -1,
"<*>:
<-1,A,<3>>",
```

```
Abs (Status (CurrentPlatform)) = 2,
"<*>:
<2,B,<16>>
<2,F,<16>>
<2,L,<17>>
<2,P,<16>>")
```

```
Case (Status (CurrentPlatform) = 1,
"<*>:
<2,B,<12>>
<2,F,<12>>
<2,L,<13>>
<2,P,<12>>",
```

```
Status (CurrentPlatform) = -1,
"<*>:
<-1,A,<3>>",
```

```
Abs (Status (CurrentPlatform)) = 2,
"<*>:
<2,B,<14>>
<2,F,<14>>
<2,L,<15>>
<2,P.<14>>")
```

There may be times, however, when you, as a developer, or other system managers will need access to the Preferences menu. You can still access Application Preferences or Document Preferences via script. So, for example, you

can have an Open Application Preferences script that is conditional based on User Name, or create a button that is available on a management-only layout.

You may want to disable the Preferences menu item in all but Layout mode (and, of course, make certain that Layout mode isn't available to most users).

An additional protection is to use the "SFMDisable" Menu-Event function available with a MenuMagic license, which can be used to automatically quit FileMaker Pro if SecureFM is disabled. See the Menu-Event documentation for more information.

Disabling Exit / Quit


The Exit / Quit command can be disabled by using menu item number "40" in the File menu. If you want to disable Exit / Quit in Browse mode, you can use:

<*>:
<1,B,<40>>

This format for disabling Exit / Quit minimizes confusion over the numerical location of Exit / Quit within the File menu. Differences in the File menu of several versions of FileMaker, as well as the potential to display recently opened files, mean the numerical location of the Exit / Quit command can vary significantly.



X Disabling Quit on Mac OS X



In Mac OS X, the Quit command no longer appears at the bottom of the File menu. Instead, the Quit command is the final menu item in the “FileMaker Pro” Application menu, menu # <-1>. The correct string for disabling Quit in Mac OS X is:

<*>:
<-1,B,<40>>

Disabling Recently Opened Files in the File Menu



If you want to tightly control user navigation, but you don't want to entirely disable or remove the File menu, you may need to use the following technique. In addition to controlling FileMaker's Window menu, you must also disable the list of recently opened files that may appear in the File menu. To complicate the situation, varying numbers of recently opened files can be listed in the File menu based on how the Application Preferences are set.

As mentioned below, it is important to disable Preferences in the Edit menu (or Application menu for OS X), but it is possible that "Show recently opened files" is already marked. This means that users can navigate to other FileMaker files, even if you have attempted to prevent this by disabling the Window menu. (Note: Our DialogMagic plug-in allows you to unmark the "Show recently opened files" preference from within a script. For more information, visit our website at <http://www.newmillennium.com>.)

Since the "Show recently opened files" preference allows users to vary the number of displayed files, we can't always know exactly how many menu items will appear in the File menu. The maximum number of displayed files, however, is 9. When recently opened files are

displayed, FileMaker also adds another divider to the File menu, making the maximum number of File menu items 31 (21 normal menu items, 9 recently opened files, one additional divider).

The solution is to simply disable several menu items past number 21 (or 20 in FMP 5.0) in the File menu:

```
<*>:  
<1,A,<21,22,23,24,25,26,27,28,29,30,31>>
```

Even if "Show recently opened files" is not marked, and the last menu item is number 21, no errors will occur. When a menu item number does not exist, SecureFM simply ignores it.

Disabling Window Closeboxes

SecureFM will automatically disable the window closebox when the associated Close menu command is disabled.



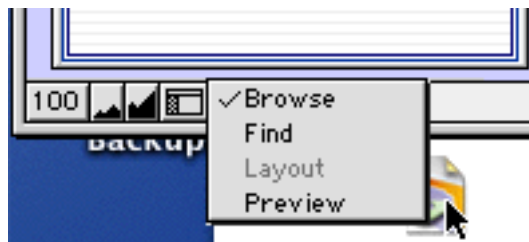
On the Mac platform, the closebox is disabled when you disable the File: Close command.



In Windows, the file window's closebox is disabled when you disable the Close command in the File menu. Since the application window's closebox is, in effect, a Quit button, SecureFM also disables the application closebox when the Exit / Quit command is disabled in the File menu.

Disabling the Mode Footer Pop-Up Menu

When you disable access to any of the four modes (Browse, Find, Layout, Preview) in the View menu, access is also disabled in the mode pop-up menu that appears in FileMaker's footer bar.



Controlling the Minimize Window Command



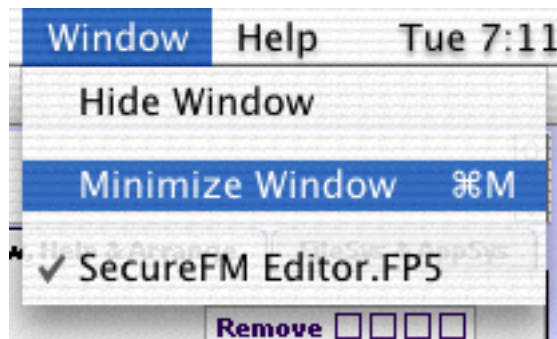
Minimize Window on Windows Platform

On the Windows platform, the Minimize command appears in both the File System and the Application System menus, relating to the File and Application windows, respectively. When disabling those two Minimize commands, the Minimize buttons in the title bars are also disabled. **In addition, the "Toggle Window [Minimize]" command is disabled in scripts. This is the only function for which disabling the menu item also disables the script function.** If you want to still be able to minimize windows in your scripts, the workaround is to change the SecureFM configuration immediately before the "Toggle Window [Minimize]" step and then restore the original setting immediately afterward - that is, add a Set Field step, setting External("Menu-Disable") without the Minimize item being disabled, just before the Toggle Window step, and then set the original configuration again afterward. This problem only occurs with the Minimize menu item; Maximize works correctly.



Minimize Window on Mac OS X

On Mac OS X, a new Minimize Window command appears in the Window menu. Disabling that menu item ("`<8,A,<3>`") disables the Minimize Window menu item, its associated keyboard command (Cmd-M), and the yellow Minimize button in the OS X FileMaker title bar, as well as disabling the ability to minimize the window by double-clicking on the title bar.



Disabling the Restore Button

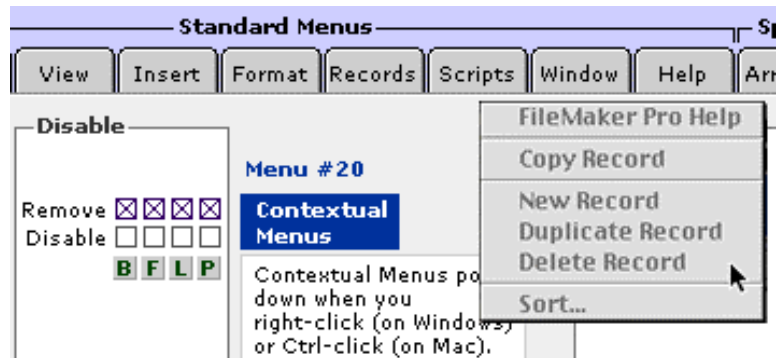
You can disable the Restore button that appears in the file window title bar. SecureFM treats the Restore button in the file window title bar as if it is the same as the first divider in the Window menu (menu #8, item #2).

To disable the Restore Window widget, use:

<8,B,<2>>

Disabling Contextual Menus

Contextual Menus are the pop-up menus that appear in FileMaker when you do Ctrl-Mouse Click (on the Mac) or Right Mouse Click (on Windows).



Even though these menus may not be widely used by the average user, through them FileMaker Pro offers access to powerful FileMaker commands, such as Delete Record and Duplicate Record. EVEN IF YOU HAVE DISABLED THESE COMMANDS IN FILEMAKER'S MENU BAR, THEY ARE STILL AVAILABLE TO THE USER IN THE CONTEXTUAL MENUS! (SecureFM Menu Editor will, however, warn you when you have checked a menu item to be disabled but have not disabled the appropriate Contextual Menu.) For complete security, you must also disable the Contextual Menus.

Contextual Menus have been assigned a menu number of twenty ("<20>"), and SecureFM only allows you to disable ("<0>") or remove ("<-1>") them; you cannot disable individual items in the Contextual Menus.

To be certain you haven't left any backdoors in your SecureFM configuration, it is usually a good idea to include the following in your configuration string:

<20,B,<-1>>

<20,F,<-1>>

<20,P,<-1>>

Disabling New Database in FileMaker's Initial Open Dialog

If you want to prevent users from being able to create new FileMaker databases under all circumstances, in addition to disabling the "New Database..." command in the File menu, you must also disable the ability to create new databases in FileMaker's initial open dialog.

This can be done using SecureFM's Menu-SetStartup command. See the documents on Menu-SetStartup for a discussion on how to do this.

Disabling Copy All Records

The Copy All Records keyboard command on Macintosh, “Cmd-Opt-C” is disabled when either Copy (“<2,A,<4>>”) or Export Records (“<1,A,<17>>”) is disabled.

X On OS X, “Copy All Records” appears in place of “Copy” in the Edit menu when the Option key is held down. This too is grayed out when either Copy or Export Records is disabled. If Export Records is disabled and not Copy, then the Copy command remains available in edit. When the option key is hit, Copy becomes Copy All Records but is grayed out.

Disabling Services in OS X Application Menu Returns an Error

Attempting to disable “Services” in the OS X application menu returns an error. If you disable the entire OS X application menu, the Services menu item will be grayed out and unselectable, but its associated command keys will still work. The only way to completely disable Services is to use the Menu-Menubar function.

Platform Menu Differences



Platform Differences Between FileMaker Menus

If you plan on using SecureFM for a cross-platform solution, there are a few key differences you must be aware of between FileMaker's menus for the various platforms.

This is essential to keep in mind when creating SecureFM configuration strings since SecureFM disables menu items by position number, not by name. The same menu item may have a different menu position on different platforms.

The SecureFM Menu Editor file makes these differences easy to recognize.

File Menu <1>

The File menu is identical on both Mac and Windows except that on Mac OS X, the Quit command has been completely removed and placed in the new “FileMaker Pro” Application menu.

Edit Menu <2>

The Edit menu on FileMaker for Windows contains two items not in the Mac Edit menus, Paste Special... and Object, affecting subsequent menu item numbers. As a result, the very important Preferences menu item is item #14 (#15 in Layout mode) on Windows, but it is item #12 (#13 in Layout mode) on Mac Classic. On Mac OS X, the Preferences command has been completely removed and placed in the new “FileMaker Pro” Application menu.

There is an additional variation with FileMaker Pro 6 for Windows: the Edit menu has an additional separator between Spelling and Object. This, along with the addition of FileMaker Pro 6’s new Find/Replace menu item, makes the Preferences menu item #16 on FileMaker Pro 6 for Windows and item #15 on Mac Classic.

View Menu <3>

On Windows, the View menu has an additional Status Bar menu item (item #11, in Layout mode item #17) allowing the user to show or hide the Status Bar area that appears at the bottom of the Application window on the Windows platform. (For more information, see the Menu-SetStatusbar command.)

Insert Menu <4>

The Insert menu on Windows has one item not found on Macintosh: Object...

Window Menu <8>

While Windows, Mac Classic, and Mac OS X all show a list of displayed files in the Window menu, they differ in the other menu items shown. On Windows, there are four commands: Tile Horizontally, Tile Vertically, Cascade, and Arrange Icons. On Mac Classic, the only command is Hide Window. On Mac OS X, the new Minimize Window command has been added, along with a divider.

Help Menu <9>

Several platform differences exist in the Help menu. The first four items in the Mac Classic Help menu relate to Balloon Help and do not appear in either the Mac OS X or Windows Help menus. In addition to the core Help menu items of FileMaker Pro Help, Contents and Index, and FileMaker on the Web, the Windows Help menu also has two additional items: What’s This?, and About FileMaker Pro...



File System and Application System Menus on Windows

SecureFM allows you to disable two additional menus that appear only on the Windows platform: the File System menu and the Application System menu.



The File System menu appears when you click on the FileMaker file icon to the far left of the file window title bar.

File System menu <0>



The Application System menu appears when you click on the FileMaker application logo to the far left of the application window title bar.

Application System menu <-1>

SecureFM allows control of these two windows, allowing you the ability to fully disable a user's capability to close windows, go to open files using the "Next" command, or maximize or minimize windows.

X “FileMaker Pro” Application Menu on Mac OS X



In FileMaker for Mac OS X, there is a new “FileMaker Pro” Application menu that appears immediately to the left of the standard File menu. The two most important items in the Application menu to be aware of for security purposes are Quit and Preferences (which appear in the File and Edit menus, respectively, on other platforms). The menu number corresponds with the Windows Application System menu:

Application menu <-1>

X Dynamic Menus on OS X

Holding down the Option key when a menu is pulled down alters certain menu items. The dynamic menu item is generally related to the standard menu item. For example, Paste becomes Paste Text Only. One dynamic menu item change to be aware of is that Copy becomes Copy All Records (see Controlling Copy All Records elsewhere in the Menu-Disable documentation).

When SecureFM disables a particular menu item, it also disables its associated dynamic menu item and keyboard command.

Creating Cross-Platform Strings with Status (CurrentPlatform)

Using the FileMaker function Status (CurrentPlatform) allows you to check which platform is currently running FileMaker Pro, allowing you to specify which variation of the configuration string you want to activate.

Status (CurrentPlatform) returns the following results for each specific platform:



Windows 95, 98, and ME = 2



Windows NT, 2000, and XP = -2



Mac Classic = 1



Mac OS X = -1

You can use Status (CurrentPlatform) to create CONDITIONAL configuration strings.

If you need to be aware of each platform in your configuration string, the following conditional statement will work:

Case (Status (CurrentPlatform) = 1,
MAC CLASSIC STRING,

Status (CurrentPlatform) = -1,
MAC OS X STRING,

Abs(Status (CurrentPlatform)) = 2,
WINDOWS STRING)

Note that the use of the absolute value function ("Abs") will return 2 for either Status (CurrentPlatform) = 2 or -2.

This technique can be used for any discrepancy in the menus between Mac and Windows versions of FileMaker Pro. You can also use this technique if you want different configurations based on platform.

Let's take a look at some examples of how this works.

Ex: Two Platform Conditional Strings

As a simple example, the Windows Edit menu has two more items ("Paste Special..." and "Object") that are not found in the Macintosh Edit menu. This causes, the same item to have a different number depending on the platform. For example, the "Clear" command is Edit item number 6 on the Mac (the same for Mac Classic and OS X), but number 7 on Windows.

If you use the string,

```
"<*>:  
<2,B,<6>>"
```

SecureFM will disable "Clear" on Mac solutions, but will disable "Paste Special..." if you try to use that solution on a Windows machine!

The solution to this problem is to create a conditional configuration string. To tell SecureFM to disable the "Clear" command correctly for both Macintosh and Windows, your string will need to look like this:

```
Case (Abs(Status (CurrentPlatform)) = 1,  
"<*>:<2,B,<6>>",  
"<*>:<2,B,<7>>")
```

SecureFM will then disable the Edit menu's item number 6 if on a Mac (Abs (Status (CurrentPlatform)) = 1); otherwise, on Windows, it will disable item number 7.

Ex: Multi-Platform Conditional Strings

Suppose your configuration string is different for all three platforms – Windows, Mac Classic, and Mac OS X – and your solution may be run on any of them.

A common example of this is disabling the Preferences menu item in Browse mode. On Windows, it is menu item 14 in the Edit menu (“<2,B,<14>”), on Mac Classic it is menu item 12 in the Edit menu (“<2,B,<12>”), and on Mac OS X it is in an entirely different menu, menu item 3 in the Application menu (“<-1,B,<3>”).

It is easy, however, to create a conditional string to cover all three possibilities:

```
Case (Status (CurrentPlatform) = 1,  
"<*>:<2,B,<12>"),
```

```
Status (CurrentPlatform) = -1,  
"<*>:<-1,B,<3>"),
```

```
Abs (Status (CurrentPlatform)) = 2,  
"<*>:<2,B,<14>")
```

SecureFM Menu Editor Automatically Generates Cross-Platform Configurations

The ‘Menu Editor’ within the SecureFM Menu Editor file automatically generates cross-platform configuration strings. For most uses, the strings created in the Menu Editor can be used without further editing or even a clear understanding of what the string means.

For further exploration of how to construct cross-platform strings, see the fully-commented ‘EXAMPLES’ scripts within the SecureFM Menu Editor file.

FMP Version Differences

Menu Differences Between FileMaker Pro 5.0, FileMaker Pro 5.5, FileMaker Developer 5.5, FileMaker Pro 6, and FileMaker Developer 6

File Menu <1>

There is a difference between the menus in FileMaker Pro 5.0 and later versions in that there is an additional menu item in the File menu. "Open Remote..." is the third File menu item, bumping "Close" and all subsequent File menu items up one item number.

Thus, a string designed to disable the "Close" command on FMP 5.0--

```
"<*>:
```

<1,B,<3>>"

--must be different for FMP 5.5 and 6:

"<*>:
<1,B,<4>>"

FileMaker Developer 5.5 and FileMaker Developer 6 differ from FileMaker Pro 5.5 and FileMaker Pro 6 by adding one further File menu item: 'Database Design Report...' is item number 10. An extra divider is also added. Thus, all items in the File menu that were items 10 or above in FileMaker Pro 5.5 and 6 are bumped up two positions in FileMaker Developer 5.5 and 6.

Runtime files created with FileMaker Developer 6's Developer Tool have a significantly altered File menu that essentially allows only the Print, Import, Export, and Quit/Exit commands. Other functions like New Database, Open, and Close are completely removed.

Edit Menu <2>

In FileMaker Pro 6, a new Find/Replace command has been added as menu item #10 on Mac Classic and OS X, and item #11 on Windows. This is a powerful command that potentially can allow users to modify data in any record within the database and should, therefore, be disabled if your configuration is designed to protect data, lock records, etc. This new item also affects the position of the very important Preferences menu item which should always be disabled.

View Menu <3>

In Runtime files, the Layout Mode menu item is completely removed since there is no access to Layout mode in Runtime files. This means that all subsequent items in the View menu are one position number less than in standard FileMaker Pro menus.

Insert Menu <4> -- Layout Mode only

There are three items in the Insert menu in Layout mode only which do not exist in FileMaker Pro 5.0: "Graphic Object" (in addition to the "Object" item that appears in Windows in the Insert menu), "Button", and "Portal". The menu numbering for the Insert menu in Layout mode is therefore significantly different for FileMaker Pro 5.0 than the other versions. The Insert menu for all other modes, however, is identical.

Scripts Menu <7>

"Debug Scripts" becomes item number three in FileMaker Developer 5.5 and FileMaker Developer 6, and another divider is added. For this reason, the

displayed scripts in the Scripts menu start with item number 5 in FileMaker Developer 5.5 and 6, rather than item number 3.

Runtime files removes the ScriptMaker menu item. This means that the only items that appear in the Script menu are displayed scripts. The FileMaker Developer also allows you to rename the Scripts menu.

Window Menu <8>

In Mac versions of FileMaker Pro and FileMaker Developer prior to version 6, there is no Bring All to Front command. This means that the displayed files in the Window menus of earlier versions of FileMaker begin with items 3 (on Mac Classic) or 5 (on OS X).

Help Menu <9>

The Help menu in Runtime files has no items unless you create custom items in the FileMaker Developer Tool.

Branching for Different Versions of FMP

If your solution modifies individual items in the File, Scripts or the Insert menus, your SecureFM configurations may be different depending on which version of FileMaker Pro is running. (This is not a concern if you are not modifying those three menus or if you are completely disabling or removing them.)

You can use the Status (CurrentAppVersion) command to verify which version of FileMaker Pro is running, and use the TextToNum command to make sure the version is evaluated in numerical format.

Ex: Branching Your Script for Non-Dev Versions of FMP

As a simple example, if you wish to disable the "Close" command in the File menu, but you don't know if FMP 5.0 or a more recent version will be running, you can branch your script with an If statement:

```
If [TextToNum (Status (CurrentAppVersion)) >= 5.5]
    Set Field ["Response Field", External("Menu-Disable", "<*>:<1,B,<4>>")]
Else
    Set Field ["Response Field", External("Menu-Disable", "<*>:<1,B,<3>>")]
End If
```

Checking the TextToNum value of Status (CurrentAppVersion) allows for a simple numerical evaluation, but it will not reveal whether FileMaker Developer edition is running. The TextToNum function drops all non-numerical values and converts the remaining numerals into a decimal number.

FileMaker Version	Status (CurrentAppVersion) Response	Converted with TextToNum
FileMaker Pro 6.0v4	Pro 6.0v4	6.04
FileMaker Developer 6.0v4	Pro 6.0v4 Dev	6.04

To check for whether the Developer edition is running, you must also use either PatternCount or the RightWords function to check for “Dev”.

Ex: Branching Your Script for FMPro and FMDev

Suppose you want to disable Export Records and Save a Copy As in the File menu in Browse mode. On FileMaker Pro 5.0, the configuration string is "<1,B,<16,17>>". With FileMaker Pro 5.5 and 6, the Open Remote menu item bumps subsequent File menu items up one, making the string "<1,B,<17,18>>". In FileMaker Developer 5.5 and 6, with its Database Design Report menu item and the extra divider, subsequent menu items get bumped up an additional two positions, making the string "<1,B,<19,20>>".

We use the greater than “>” operator in order to refer to all versions of FileMaker Pro greater than “5.5”, which will include all 5.5 and 6 versions (5.5v1, 5.5v3, 6.0v2, 6.0v4, etc.).

```
If [TextToNum(Status(CurrentAppVersion)) > 5.5] and
not PatternCount(Status(CurrentAppVersion), "Dev")
Comment [#FileMaker Pro 5.5 or FileMaker Pro 6]
Set Field ["Response Field", External("Menu-Disable",
"<*>:<1,B,<17,18>>")]
Else
If[TextToNum(Status(CurrentAppVersion)) > 5.5 and
PatternCount(Status(CurrentAppVersion), "Dev")]
Comment [#FileMaker Developer 5.5 or Developer 6]
Set Field ["Response Field", External("Menu-Disable",
"<*>:<1,B,<19,20>>")]
Else
If[Left(TextToNum(Status(CurrentAppVersion)),3) = 5.0]
Comment [#FileMaker Pro 5.0]
Set Field ["Response Field", External("Menu-Disable",
```



```
"<*>:<1,B,<16,17>>"]End If  
End If  
End If  
End If
```

Creating Strings for Different Platforms with the Menu Editor

When creating a new menu configuration in the SecureFM Menu Editor file's 'Menu Editor,' the 'Calculate String for' preference defaults to 'Selected Version of FMP.' When you click the 'Create String' button, the Menu Editor will calculate the string for only the selected version of FileMaker Pro. To change this preference, click on the 'View Strings' button to go to the Strings Palette. You can then change the radio button to 'All FMP Versions'. The Menu Editor will then calculate the string for FileMaker Pro 5.0, 5.5, 6, and Developer 5.5 and 6.

To see the configuration string for a different version of FileMaker Pro, select the version you want to see in the FMP Version # popup menu in the upper left of the Menu Editor layout, or Strings Palette layout.

For further exploration of how to branch your scripts based on the version of FileMaker in use, see the fully-commented 'EXAMPLES' scripts within the SecureFM Menu Editor file.

Errors

FileMaker's Set Field String Length Limitation

For some menu configurations, the length of the string can become quite long. This can create a problem because FileMaker can only process strings with 253 characters or fewer contained between two quotes.

The solution is to break the string up into two (or more) strings separated by an ampersand (&).

For example, suppose you have a string like:

```
"...  
<2,B,<2,8>>  
<2,F,<1,3,4,5,9,11>>  
<2,L,<1,5,6,9,10>>  
<2,P,<4,6,7,10>>
```



```
<3,A,<1>>  
..."
```

You can break it into two strings separated by an ampersand ("&"):

```
"...  
<2,B,<2,8>>  
<2,F,<1,3,4,5,9,11>>  
<2,L,<1,5,6,9,10>>"  
&  
"<2,P,<4,6,7,10>>  
<3,A,<1>>  
..."
```

As long as you have fewer than 253 characters contained between two quotation marks, FileMaker will process the string without a problem.

Errors in the String

When you call the Menu-Disable command in a Set Field script step, SecureFM returns an empty string if successful, or an error message if the parameter is incorrect in some way.

Configuring Menus with a Password

If you have registered SecureFM with a password (see Menu-Register), the same password is required to change the menu configuration or remove toolbars. To set a menu configuration with a password, you must add an additional pipe character "|" and the password to the end of the Menu-Disable string.

Ex: Password in a Simple String

For example, if your original Menu-Disable configuration was:

```
<*>:  
<7,A,<-1>>
```

The string would become:

```
<*>:  
<7,A,<-1>>|Password"
```

Ex: Password in a Multi-Line String

This format works the same when the string is multiple lines:

```
"<*>:  
<7,B,<-1>>  
<7,F,<-1>>  
<7,P,<-1>>|Password"
```

Ex: Password in a Cross-Platform String

If you are using a string that branches based on platform, it is important to include the password at the end of each platform's string:

```
Case (Status (CurrentPlatform) = 1,
```

```
"<*>:  
<2,B,<12>>  
<2,F,<12>>  
<2,P,<12>>  
<2,L,<13>>|Password",
```

```
Status (CurrentPlatform) = -1,
```

```
"<*>:  
<-1,A,<3>>|Password",
```

```
Abs (Status (CurrentPlatform)) = 2,
```

```
"<*>:  
<2,B,<14>>  
<2,F,<14>>  
<2,P,<14>>  
<2,L,<15>>|Password")
```

Password Error

If the password is missing or wrong in the Menu-Disable string, SecureFM returns "Incorrect password" and does not change the currently active configuration.

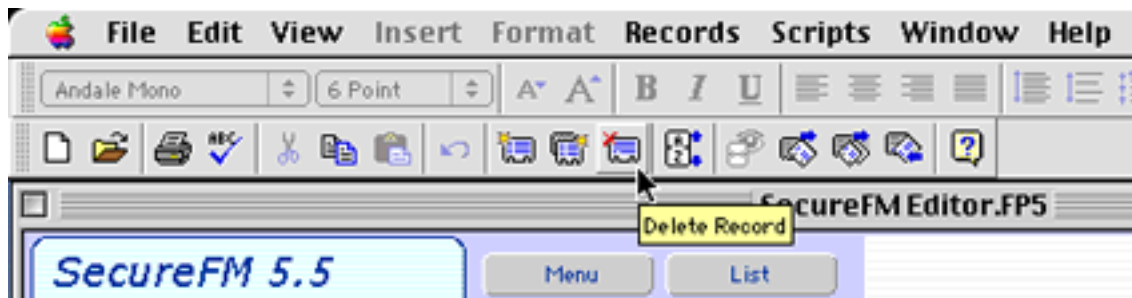
Menu-Toolbar

External ("Menu-Toolbar", parameter)

SFM Available with a SecureFM or MenuMagic license.

SecureFM is able to easily remove FileMaker Pro's toolbars: the Standard Toolbar (which includes such commands as New Record and Delete Record) and the Text Formatting Toolbar (which includes font sizes, text alignment, etc.).

Disabling the toolbars, especially the Standard Toolbar, can be very important to create a secure environment for your FileMaker solution. Even if you disable a function like Delete Record in FileMaker's Records menu, IT IS STILL AVAILABLE IN THE TOOLBAR! You must ALSO disable the Standard Toolbar.



To have a truly secure environment, you may need to disable FileMaker's toolbars. SecureFM gives you that ability.

Ex: Removing Both Toolbars

The following command:

External ("Menu-Toolbar", "0|0")

disables both the Standard Toolbar and the Text Formatting Toolbar in ALL modes.

Toolbar Command's Syntax

The zero on the left tells SecureFM to remove the Standard Toolbar. The zero on the right removes the Text Formatting Toolbar. Restated:

Left of pipe divider - Standard Toolbar

Right of pipe divider - Text Formatting Toolbar

There are three possible commands you can send to SecureFM for each of the toolbars:

- 0 - Remove toolbar
- 1 - Display toolbar
- (blank) - No change to toolbar status

Note that the pipe character divider ("|") must always be used in the parameter, even if only one of the toolbars is being modified.

Ex: Removing Just One Toolbar

If you just want to remove the Standard Toolbar and leave the Text Formatting Toolbar alone, use:

`External ("Menu-Toolbar", "0|")`

Remember that the pipe character divider ("|") must be used, even though only one of the toolbars is being modified.

Why Have a 'No Change' Option?

Suppose you want to remove only the Standard Toolbar, but it doesn't matter to you whether the Text Formatting Toolbar is visible or not. If you use the command:

`External ("Menu-Toolbar", "0|0")`

the Text Formatting Toolbar will also be removed. And if you use the command:

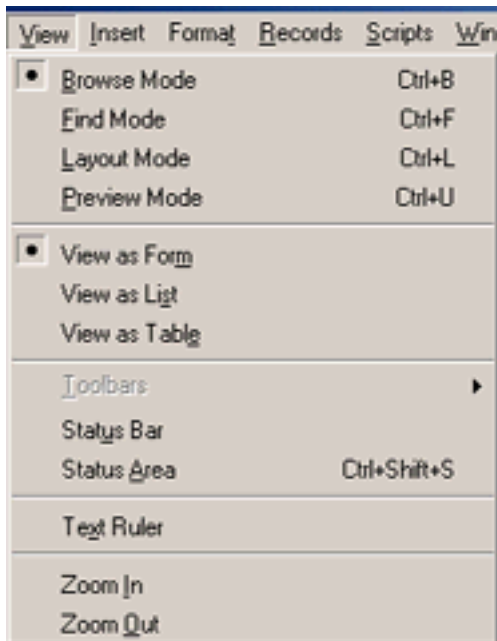
`External ("Menu-Toolbar", "0|1")`

the Text Formatting Toolbar will automatically appear. By leaving the Text Formatting Toolbar's command blank:

`External ("Menu-Toolbar", "0|")`

the Text Formatting Toolbar is left however it was previously set by the user or by a previous Menu-Toolbar command.

Important: Disabling the Toolbars Menu Item



It is very important that you also disable the Toolbars menu item in the View menu!

"<3,B,<10>>"

If you don't, users will be able to manually re-enable the toolbars at any time. (SecureFM Menu Editor will warn you if you have checked one or both of the toolbars to be disabled but have not disabled the View: Toolbars menu item as part of your configuration string.)

Toolbar Command and FileMaker's Modes

Removing (or displaying) toolbars with the SecureFM Menu-Toolbar command affects ALL modes (Browse, Find, Layout and Preview). You cannot control the toolbars in individual modes.

FileMaker's Other Toolbars

SecureFM only gives you direct control of the Standard Toolbar and Text Formatting Toolbar. FileMaker's other toolbars (Arrange and Tools) are available only in Layout mode and are not controlled by SecureFM. You can, however, restrict access to those other toolbars by disabling the Toolbars item in the View menu for Layout mode.

Re-Enabling Toolbars

The "Restore All Menus" button does NOT cause disabled toolbars to reappear. It will, however, re-enable the Toolbars item in the View menu, which allows you to manually activate the toolbars. In order to make both toolbars automatically reappear, you must use the following command:

External ("Menu-Toolbar", "1|1")

Important: 3 Second Pause Sometimes Necessary in Scripts

Because of the way that SecureFM disables the toolbars, one or both toolbars may not be properly removed when you are using the Menu-Toolbar command in a script **AND:**

- a) The Menu-Toolbar command is the first script step, **OR**
- b) The Menu-Toolbar command is followed by multiple script steps or by a step which causes the script to not be completed without user interaction, such as Show Message.

It is extremely important that you test any scripts using the Menu-Toolbar command on all platforms. If you discover any problems, there is a simple solution: using FileMaker's Pause/Resume Script step.

Use a single Pause/Resume Script step set to three or more seconds:

[Pause/Resume Script \[0:00:03\]](#)

Setting the duration to one or two seconds doesn't always work reliably.

For more examples of how this technique should be used, see the Example Scripts included among SecureFM Menu Editor's scripts.

Important: Zero Second Pause Technique in Previous Versions

In past versions of SecureFM, we recommended an alternate technique of using two Pause/Resume Script steps set to zero seconds each. We have found that the zero second technique does not work reliably with FileMaker Pro 6 on Mac Classic. The three second pause, however, works without problem (so long as you follow the guidelines mentioned above).

X Toolbars and FileMaker Pro for Mac OS X

In FileMaker Pro 5.5 and 6 for Macintosh OS X, the Toolbars are not available. Attempting to use the Menu-Toolbar command to disable Toolbars on OS X will cause an error to be returned.

If the Menu-Toolbar command is included as part of a Set Field step that also includes the Menu-Disable command, FileMaker's menus will still be properly modified. In other words, the error message only notifies you that the Toolbars

cannot be modified on OS X, but all other SecureFM functions in the step will still be implemented.

Your script can either ignore the response or make the Menu-Toolbar conditional based on Status (CurrentPlatform) <> -1.

Disabled Toolbars in SetStartup Default

You may need to use the above Pause/Resume technique if you have disabled toolbars as part of FileMaker's default configuration using the SetStartup command (see below) AND your solution opens with an On Open script.

If your On Open script has multiple steps or it requires user interaction (with a Show Message step, for example), you may need to use the Pause/Resume technique in your On Open script to be certain that FileMaker's toolbars will properly disable on startup. Because of this, it is also a good idea to make sure that all files in your solution will call the main On Open script when started up. Evaluate and, if necessary, test your solution to determine if this is required.

Scripting Rather than Disabling Toolbars with MenuMagic

The individual buttons on the Toolbars can be set to call your own custom scripts rather than perform their normal functions using the Menu-Script function. You can, for example, set the Delete Record button on the Standard Toolbar to trigger your own script that will only delete records for users with a management level password.

For more information on how to script the toolbar buttons, see the Menu-Script function documentation.

Configuring Toolbars with a Password

If you have registered SecureFM with a password (see Menu-Register), the same password is required to change the menu configuration or toolbars. To configure toolbars with a password, you must add an additional pipe character "|" and the password to the end of the Menu-Toolbar string.

Ex: Password in a Simple String

For example, if your original Menu-Toolbar string was:

[External \("Menu-Toolbar", "0|0"\)](#)

It would become:

`External ("Menu-Toolbar", "0|0|Password")`

Ex: Configuring Without a Password

If SecureFM was registered without a password, the password part of the parameter and the last pipe character must be omitted:

`External ("Menu-Toolbar", "0|0")`

Important!: In SecureFM versions 5.06 to 6.0x, the pipe character could be included even when no password was present – hence this was valid:

`External ("Menu-Toolbar", "0|0|")`

In SecureFM 6.1 and later, this is no longer valid! You cannot include the last pipe character unless a password is included.

Password Error

If the password is missing or wrong in the Menu-Toolbar string, SecureFM returns "Incorrect password" and does not change the currently active toolbar configuration.

Menu-Menubar

External ("Menu-Menubar", parameter)

SFM Available with a SecureFM or MenuMagic license.

The Menu-Menubar function gives you the ability to instantly hide FileMaker Pro's entire menu bar, disable all command keys, and disable all widget buttons in the window title bars (such as close window, maximize window, etc). This function also lets you prevent the file window from being moved or resized on Windows.

This is a simple function that only uses two parameters, marked 1 for on or 0 for off:

1|1 = Normal state

0|1 = Remove FMP menu bar. The entire menu bar is hidden. No contextual or application popup menus will show (but user-defined field popup menus will still work). All FMP and OS keyboard commands are disabled; only Menu-HotKeys will work.

1|0 = Disable all window widget buttons: Close Window, Maximize Window, Minimize Window, Restore Window. (On the Windows platform, the Application System and File System menus are disabled with this parameter.) This parameter also prevents the file window from being moved or resized.

0|0 = Remove FMP menu bar and disable widget buttons.

Ex: Basic Menubar Example

To completely remove the FileMaker Pro menu bar, disable keyboard commands, and disable all window title bar "widget" buttons:

External ("Menu-Menubar", "0|0")

Warning: Be Careful Not to Get Trapped!

Because this function globally removes all FileMaker Pro menus while disabling all keyboard commands, you must be very careful not to get trapped! Once the menu bar is removed, you cannot open ScriptMaker or call any scripts, except by button – even to restore the menu bar. If you also disable window widget buttons, you cannot quit FileMaker Pro or even close file windows unless you have scripted buttons on an accessible layout to perform those functions or restore access to the menus and widgets.

You must also be careful not to close all files or minimize all file windows with the menu bar removed.

Configuring the Menubar Function with a Password

If you have registered SecureFM with a password (see Menu-Register), the same password is required to change the Menu-Menubar setting. To configure the menu bar with a password, you must add an additional pipe character "|" and the password to the end of the Menu-Menubar string.

Ex: Password in a Simple String

For example, if your original Menu-Menubar string was:

`External ("Menu-Menubar", "0|0")`

It would become:

`External ("Menu-Menubar", "0|0|Password")`

Password Error

If the password is missing or wrong in the Menu-Menubar string, SecureFM returns "Incorrect password" and does not change the menu bar setting.

Menu-Statusbar

External ("Menu-Statusbar", parameter)

SFM Available with a SecureFM or MenuMagic license.

The Menu-Statusbar function allows you to remove the Status Bar at the bottom of the FileMaker window on the Windows platform. This gives you better control of the application window's size and appearance from within your scripts.

The Status Bar appears only on the Windows platform. There is no parallel to the Status Bar on the Macintosh.

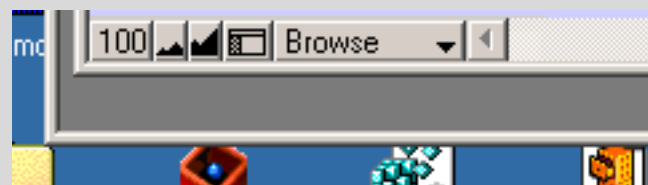
Ex: Removing the Status Bar



The following command--

External ("Menu-Statusbar", "0")

--removes the Status Bar in ALL modes.



Statusbar Command's Syntax

There are two possible commands you can send to SecureFM for the Status Bar:

0 - Remove Status Bar

1 - Display Status Bar

Disabling the Status Bar Menu Item

It is important that you also disable the Status Bar menu item in the View menu in your Windows platform configurations.

```
If[Abs(Status(CurrentPlatform)) = 2,  
External("Menu-Disable",  
    "<3><B><11>"  
    "<3><F><11>"  
    "<3><P><11>"  
    "<3><L><17>"  
End If
```

If you don't, users will be able to manually re-enable the Status Bar at any time. (SecureFM Menu Editor will warn you if you have marked the Status Bar to be disabled but have not disabled the View: Status Bar menu item as part of your configuration string.)

Statusbar Command and FileMaker's Modes

Removing (or displaying) the Status Bar with the SecureFM Menu-Statusbar command affects ALL modes (Browse, Find, Layout and Preview). You cannot control the Status Bar in individual modes.

Re-Enabling the Status Bar

The "Restore All Menus" button does NOT cause the disabled Status Bar to reappear. It will, however, re-enable the Status Bar item in the View menu, which allows you to manually activate the Status Bar. In order to make the Status Bar automatically reappear, you must use the following command:

```
External ("Menu-Statusbar", "1")
```

Statusbar and Set Startup

The Menu-SetStartup function will not save the Status Bar's position. If you want the Status Bar to be either displayed or hidden when your solution is launched, you must call the Menu-Statusbar command in a startup script.

Configuring the Status Bar with a Password

If you have registered SecureFM with a password (see Menu-Register), the same password is required to use the Statusbar command. To configure the Status Bar with a password, you must add an additional pipe character "|" and the password to the end of the Menu-Statusbar string.

Ex: Password in a Simple String

For example, if your original Menu-Statusbar string was:

```
External ("Menu-Statusbar", "0")
```

It would become:

```
External ("Menu-Statusbar", "0|Password")
```

Ex: Configuring Without a Password

If SecureFM was registered without a password, the password part of the parameter and the last pipe character must be omitted:

```
External ("Menu-Statusbar", "0|")
```

is not valid. You must use:

```
External ("Menu-Statusbar", "0")
```

Password Error

If the password is missing or wrong in the Menu-Statusbar string, SecureFM returns "Incorrect password" and does not change the Status Bar.

Menu-SetStartup

External ("Menu-SetStartup", "")

SFM Available with a SecureFM or MenuMagic license.

Understanding this command is necessary if you want a SecureFM menu configuration to apply to FileMaker on startup. This method of initializing SecureFM is useful if you want to have a default configuration of SecureFM which modifies the menus as soon as FileMaker is launched, regardless of which file is opened.

Because the file created by the SetStartup command is modifiable by a knowledgeable user, this method of configuring FileMaker's menus should only be used to control the initial launch process. It is always a good rule of thumb to configure FileMaker's menu's in your solution's On Open script, as well.

SetStartup Parameter is Not Used

Note that, although FileMaker's format for External commands assumes that there will be a parameter string, the parameter is not used by SecureFM for this function. The startup configuration is set based on the SecureFM Menu-Disable and Menu-Toolbar settings that are active when the SetStartup command is called.

Rather than look to a parameter in the External "Menu-SetStartup" command to set the default configuration, SecureFM sets the default based on:

- 1) The currently ACTIVE menu configuration (set using the "Menu-Disable" command),
- 2) The currently ACTIVE toolbar configuration (set using the "Menu-Toolbar" command), and
- 3) The currently ACTIVE registration (set using the "Menu-Register" command).

(Menu-Rename and Menu-Script configurations are not stored by Menu-SetStartup.)

Any of the above configurations you want set into the default must first be made active BEFORE using the "Menu-SetStartup" command.

The syntax for this function is always:

External ("Menu-SetStartup", "")

The parameter should always be empty.

Passwords and SetStartup

While encoded registration information, as well as Menu-Disable and Menu-Toolbar settings, are stored in the default preferences file ("SecureFM.ini"), no passwords are saved. This means that if you are relying on the default preferences to register SecureFM, it will do so with a blank password.

If you want SecureFM to be encoded with a password, you must re-register with the password (in a script, field calculation, etc.) after startup. For more details, see the section on Password Registration.

Using SetStartup in the SecureFM Menu Editor

In order to create a default configuration from the SecureFM Menu Editor file, you must create the SecureFM configuration string you want, activate it by clicking the 'Apply Now' button, and then click the Set Startup button.

SecureFM then creates a text file called "SecureFM.ini" that stores the default configuration data. This file is saved in the FileMaker Extensions folder (Mac) or FileMaker's System directory (Windows). When FileMaker is started, the default Menu-Disable settings will immediately take effect without the need to run any script or even have a file be open.

The SecureFM.ini file can be copied directly to workstations, making it very easy to configure a default configuration throughout a network.

SecureFM.ini for Different Platforms

The configuration string stored by SecureFM in the default preference file ("SecureFM.ini") is platform-specific. Even if the string you activate is conditional based on platform (see above), the configuration stored in the SecureFM.ini file will only contain information for the platform you are using.

If you are setting the default preference for a cross-platform solution -- and if the string is different for the two platforms -- you must create two separate SecureFM.ini files (one for Mac, one for Windows).

If the string is different for the two platforms, be aware that you must call the SetStartup function to create the SecureFM.ini file for each platform on the CORRECT platform. That is, you must create a Windows SecureFM.ini file on a Windows machine, and a Mac SecureFM.ini file on a Mac.

Disabling New Database Creation in FMP's Initial Open Dialog

If you want to prevent users from being able to create new FileMaker databases under all circumstances, in addition to disabling the New Database... command in the File menu, you must also disable the ability to create new databases in FileMaker's initial open dialog.

This 'initial open dialog' occurs when a user opens the FileMaker application directly, rather than opening a database file created by FileMaker.



On the Mac, the initial dialog contains a New button.



On Windows, the dialog's Open button can be used to create a new database.

To disable new database creation in the initial open dialog, disable the New Database... command ("`<1><A><1>`") along with any other commands you want disabled as a default setting, and then use the SetStartup command to set the default.

The New button will be completely removed on the Mac. On Windows, the Open button will be disabled, however users can still open FileMaker files by double-clicking them in the selection window.

X **Disabling New Database Bypasses FileMaker's Initial Open Dialog on OS X:** Using the Set Startup command to disable New Database creation in the .ini file results in OS X completely bypassing the initial Open dialog when FileMaker Pro is launched.

Important Note -- Initial 'Templates' Dialog: When opening FileMaker directly, you can actually get one of two different initial dialogs. One is the open dialog mentioned above. The other is a 'templates dialog' that also allows for the creation of new databases using radio buttons (as well as opening existing FileMaker files and templates). SecureFM DOES NOT DISABLE THE NEW DATABASE COMMAND IN THE 'TEMPLATES' DIALOG. To make sure only the open dialog appears, you must uncheck the 'Show Templates in New Database dialog' checkbox in the Application Preferences (in the Edit menu) for all copies of FileMaker on the network. This can be done automatically with New Millennium's DialogMagic plug-in: <http://www.newmillennium.com>.

Menu-Rename

External ("Menu-Rename", parameter)

MM Available with a MenuMagic license only.

The Menu-Rename function gives you the ability to rename menus and menu items.

Rename Parameter Format:

```
<File List>:  
<Menu#,Mode,  
    <  
        <Item#,ItemName >,  
        <Item#,ItemName >  
    >  
>
```

For example, in a file named "Widgets.fp5", in order to rename the Records menu (menu 6) in Browse mode (B) to "Widgets" and to rename the first few items as well, you can use a Menu-Rename string, such as:

```
<Widgets>:  
<6,B,  
    <  
        <0,Widgets>,  
        <1,New Widget>,  
        <2,Duplicate Widget>,  
        <3>Delete Widget>  
    >  
>
```

Tabs and Returns in Strings

SecureFM ignores tabs and returns in the menu definition strings. Tabbed indents are used for clarity, but they are not required. Spaces are also ignored, except when part of a menu or script name.

FMP Disabled Items

If an item is normally disabled by FileMaker Pro (based on mode, field selected, etc.) that item will still be disabled, even if you have modified it with SecureFM functions. SecureFM does not override FileMaker Pro's normal disabling process. If you have renamed an item which FileMaker Pro has disabled, it will appear renamed to the user, but the item will be grayed out.

Similarly, if you have used FileMaker Pro's access privileges to limit access to menu items by defining a password's available menu commands as "Editing Only" or "None", the limitations of that password privilege will apply and the appropriate items will be grayed out.

Shortcut Keyboard Commands

You can set shortcut keys (Alt-keys) when renaming a menu or menu item by using the ampersand "&" character. The character immediately following the ampersand "&" in the new name will be used as the shortcut key.

To rename the Records menu "Widgets" and define the keyboard shortcut Alt-D to pop down the menu:

```
<Widgets>:  
<6,B,  
    <  
    <0,Wi&dgets>  
    >  
>
```

To rename the "New Record" menu "New Widget" and define the keyboard shortcut "N" to select it once the menu has been opened:

```
<Widgets>:  
<6,B,  
    <  
    <1,&New Widget>  
    >  
>
```

Duplicated Shortcut Keys

Because SecureFM gives you the freedom to set any keyboard shortcut you want, it is possible to give the same shortcut key to more than one menu or to more than one item within the same menu. Be aware that FileMaker Pro will ignore a keyboard shortcut if it is duplicated among the menu names or among the items within a particular menu.

AppSys and FileSys Menu Items Cannot be Renamed

Menu-Rename cannot be used with the Application System (AppSys) and File System (FileSys) menus (menus 0 and -1, respectively, on the Windows platform). Only menus 1 through 10 (File, Edit, View, etc.) can be renamed by SecureFM.

Items in the AppSys and FileSys menus can still be disabled, however, using the Menu-Disable command.

Dividers Cannot be Renamed

You can send a Menu-Rename string that attempts to rename a divider (based on its numerical position in the menu), but that line of the string will have no effect. If you are using the Menu Editor, it will not allow you to attempt to rename dividers.

Sub-Menu Items Cannot be Renamed

Menu items with sub-menus can be renamed, but the sub-menu items themselves cannot be renamed or scripted. If you want to prevent user access to a particular sub-menu item you must disable the menu item that contains it.

Renaming Contextual Menu Items

Contextual Menus are the popup menus that appear when a user does a right-click (on Windows) or Ctrl-click (on Mac).

In the Contextual Menus (menu 20), renaming of items automatically occurs when the corresponding menu items have been renamed in FileMaker Pro's main menus.

For example, if we use the Widget rename example already mentioned, and then pop down a Contextual Menu in Browse mode when clicking on a layout background, instead of New Record, Duplicate Record, and Delete Record, we get New Widget, Duplicate Widget, and Delete Widget among the Contextual Menu items.

Note: The toolbar icon tool tips, activated by hesitating when mousing over an icon, are not altered by SecureFM.

Passwords

When the plug-in is registered with an optional password, the Menu-Rename string must include the same password for it to work. At the very end of the string, add a pipe character "|" and then the password. This is true for all SecureFM commands that affect menus, including Menu-Disable and Menu-Script.

Menu-Script

External ("Menu-Script", parameter)

MM Available with a MenuMagic license only.

Menu-Script gives you the ability to override FileMaker Pro's normal action when a particular menu item is selected and, instead, run one of your own scripts. With this function, you can make normal FileMaker Pro functions conditional or, used in conjunction with Menu-Rename, you can repurpose FileMaker Pro's menus.

Script Parameter Format:

```
<File List>:
<Menu#,Mode,
    <
    <Item#,ScriptName >,
    <Item#,ScriptName >
    >
>
```

Building on the previous Menu-Rename example, in the file named Widgets.fp5, if you want to redirect the first three menu items in the Records (renamed "Widgets") menu to run your own scripts , the string might look like:

```
<Widgets>:
<6,B,
    <
    <1,My Create New Widget Script>,
    <2,My Duplicate Widget Script>,
    <3,My Delete Widget Script>
    >
>
```

Tabs and Returns in Strings

SecureFM ignores tabs and returns in the menu definition strings. Tabbed indents are used for clarity, but they are not required. Spaces are also ignored, except when part of a menu or script name.

FileSys and AppSys Menus Cannot be Scripted on Windows

Menu-Script cannot be used with the FileSys and AppSys menus on the Windows platform.

Sub-Menu Items Cannot be Scripted

Sub-menu items cannot be renamed or scripted. If you want to prevent user access to a particular sub-menu item you must disable the menu item that contains it.

Scripting Contextual Menu Items

Contextual Menus are the popup menus that appear when a user does a right-click (on Windows) or Ctrl-click (on Mac). Items in Contextual Menus (menu 20) are automatically set to call custom scripts when the corresponding items in FileMaker Pro's main menus have been scripted.

For example, if we set the Widget script string example above, and then pop down a Contextual Menu in Browse mode, selecting Delete Record (or Delete Widget, if you have renamed the item) will cause the script named "My Delete Widget Script" to run instead of FileMaker's normal Delete Record process.

Scripting Closeboxes

You can attach scripts to the menu items in the File menu for Exit/Quit and Close, and the corresponding close boxes will then also trigger these scripts.

If you attach a script to the Exit/Quit menu item in the File menu `<1,B,<<40,ScriptName>>>`, that script will also be called if you click on the application window closebox. Mapping a script to the Close menu item in the File menu `<1,B,<<4,ScriptName>>>` will call that script when the file window close box is clicked.

Scripting Toolbar Buttons

As with Contextual Menus, buttons on the toolbars will automatically call your custom scripts if their corresponding items in FileMaker Pro's main menus have

been set to call scripts. (However, the mouse-over "tool tips" are not renamed when the corresponding menu items are renamed.)

Scripted Menu Items in Layout Mode

Be aware that, when FileMaker Pro is in Layout mode, running a script will switch FileMaker Pro into Browse mode.

With the Menu-DoFMPMenuItem function (see below), you can, however, switch back to Layout mode as the final step of your script.

Scripting the Same Menu Item in All Modes

While it may seem convenient to attach a single script to a menu item for all modes, this is often not a good idea. The same menu item may have a different function in different modes, or certain script steps may behave differently in different modes. There is also the issue of running scripts in Layout mode mentioned above. If you assign a script to the same menu item for all modes, make sure you have tested it and thought out all of the consequences. In most circumstances, it makes more sense to define scripts separately for each mode in which you want to override FileMaker Pro's normal menu functions.

Passwords

When the plug-in is registered with an optional password, the Menu-Script string must include the same password to work. At the very end of the string, add a pipe character "|" and then the password.

Menu-HotKey

External ("Menu-HotKey", parameter)

MM Available with a MenuMagic license only.

The Menu-HotKey function gives you the ability to trigger a custom script through any keyboard command, such as Alt-E, Ctrl-I, Cmd-Shift-Z, function keys, etc. These hot key combinations can be completely independent of the keyboard commands normally associated with FileMaker Pro's menus. If you define a hot key combination that is also associated with a FileMaker menu function (such as Ctrl-E, which normally calls Delete Record), the hot key script will run instead.

External ("Menu-HotKey", Hot Key Parameter)

Hot Key Parameter Format:

```
<File List>:  
<Mode,  
    <  
        <Modifier Key #, Primary Key, Script Name >,  
        <Modifier Key #, Primary Key, Script Name >  
    >  
>
```

Modifier Keys and Primary Key

A modifier key doesn't result in text appearing when it is held down; instead, it modifies the primary key being typed. For example, in the key combination Alt-z, "z" is the primary key, and Alt is the modifier key.

Important: Use Lower Case or 'Base' Key

Use lower case or the 'base' key when defining the primary character in the string. In other words, use "b", not "B"; "2", not "@"; "/", not "?"; etc.

Modifier Key Number

FileMaker Pro assigns a number to each of the modifier keys:

1	=	Shift
2	=	Caps Lock
4	=	Control
8	=	Alt / Option
16	=	Command (Mac)

Menu-HotKey uses the same modifier key numbers.

Command Key Note: The Command key (16) is a Mac-only modifier key.

Caps Lock Key Note: The Caps Lock key (2) is ignored by the Menu-HotKey command in SecureFM with MenuMagic version 6.3.3/7.0.3 .

(For further discussion of modifier key numbers, see the FileMaker Pro documentation on the Status (CurrentModifierKeys) function.)

A simple example

If you want to trigger a script named "Go to Utility" whenever Alt-z is typed in Browse mode in a file named Widgets.fp5, the string would look like this:

```
<Widgets>:  
<B,  
    <  
    <8,z,Go to Utility>  
    >  
>
```

Multiple Modifier Keys

If you want to trigger a script with a primary character typed in combination with more than one modifier key, the Menu-HotKey function follows the convention used by the Status (CurrentModifierKeys) function and lets you add the two modifier key numbers together.

5 = Control (4) + Shift (1)

9	=	Alt / Option (8) + Shift (1)
12	=	Alt / Option (8) + Control (4)
17	=	Command (16) + Shift (1)
20	=	Command (16) + Control (4)
24	=	Command (16) + Option (8)

Another example

To modify the simple example above to trigger the hot key script with Alt-Shift-z, all we need to do is change the modifier key number to 9:

```
<Widgets>:
<B,
    <
    <9,z,Go to Utility>
    >
>
```

Hot Keys with No Modifier Keys

The Hot Key function can also trigger a script based only on a primary key being typed without an accompanying modifier key.

You can, for example, repurpose the i, j, k, and m keys to call up, left, right, and down scripts in a FileMaker-based game. Just use a zero “0” for the modifier key section of the string:

```
<Game File>:
<B,
    <
    <0,i,Move Up>,
    <0,j,Move Left>,
    <0,k,Move Right>,
    <0,m,Move Down>
    >
>
```

Of course, a Hot Key string without modifier keys should be used carefully since it will trigger scripts during normal typing activities.

If you leave the modifier key blank, SecureFM will return an error.

Hot Keys with Modifier Keys Only

If, on the other hand, you want to trigger a Hot Key script when a modifier key is typed on its own – such as the Option key – just set the primary key to zero “0”:

```
<*>:  
<B,  
    <  
    <8,0,My Option Script>,  
    >  
>
```

Special Primary Keys

You can also define the primary key with a keycode number. This makes it possible to attach Hot Key scripts to keys that do not normally produce text characters, such as arrow keys, function keys, etc.

The keycode must always be a three-digit numeral. For example, 97 must be written as 097.

The Menu-HotKey format can now be rendered in either format below:

```
External ("Menu-HotKey", "<*>:<B,<<4,a,ScriptName>>>")  
External ("Menu-HotKey", "<*>:<B,<<4,097,ScriptName>>>")
```

To find out the correct keycode value for a specific key, use the Menu-GetMenuSelection function or consult the table below.

Keycode Numbering Not Cross-Platform

The keycode numbers are unique to each operating system. Many keys, such as function keys, have different keycode numbers on different platforms. For example, F1 = 222 on OS X, but F1 = 112 on Windows. When using the three-digit keycode, you may have to create different Menu-HotKey strings for each platform.

Function Keys on OS X

SecureFM uses the standard keycode numbering for the function keys on OS X. Be aware that these numbers do not follow a linear pattern. For example, on OS X F1 = 222, F2 = 220, F3 = 199, etc. Use Menu-GetMenuSelection or consult the table below to make sure you are using the correct number for each function key.

On Panther versions of OS X, the F10 and F11 function keys will still call Expose functions, even if you have assigned Hot Key scripts to those keys.

On Windows, even if set to call a Hot Key script, the F1 key will still call up the Help Topics window, but the Hot Key script will run in the background.

Special Key Codes

Key	Windows Key Code	OS X Key Code
Clear	144	208
Delete	008	008
Del x>	046	127
Enter	013	003
Return	013	013
Esc	027	027
End	035	004
Help	--	005
Home	036	001
Page Down	034	012
Page Up	033	011
Space	032	032
Tab	009	009
Up Arrow	038	030
Down Arrow	040	031
Left Arrow	037	028
Right Arrow	039	029
F1	112	222
F2	113	220
F3	114	199
F4	115	218
F5	116	196
F6	117	197
F7	118	198
F8	119	200
F9	120	201
F10	--	209

F11	--	203
F12	--	211
F13	--	205
F14	--	207
F15	--	213
F16	--	206
Keypad 1	097	049
Keypad 2	098	050
Keypad 3	099	051
Keypad 4	100	052
Keypad 5	101	053
Keypad 6	102	054
Keypad 7	103	055
Keypad 8	104	056
Keypad 9	105	057
Keypad 0	096	048
Keypad .	110	046
Keypad =	--	061
Keypad /	111	047
Keypad *	106	031
Keypad -	109	045
Keypad +	107	043

Clearing Hot Key Settings

As with Menu-Disable, Menu-Rename, and Menu-Script, you can clear all Hot Key settings by using SecureFM's 'clear all' parameter ":-"

[Set Field \[Response Field, External \("Menu-HotKey", ":-"\)\]](#)

Passwords

When the plug-in is registered with an optional password, the Menu-HotKey string must include the same password to work. At the very end of the string, add a pipe character "|" and then the password.

String Prefixes and Different Types of Menu Definitions

Beginning with SecureFM 6.1, menu definition strings require a prefix in addition to the actual code that defines which menu items are altered. This new prefix gives you the ability to create default menus that are applied globally to all files, while other menu definitions are active only for specific files in your solution. This also gives you the advantage of being able to load several different menu definitions all at once when your solution is launched to be applied only under certain circumstances.

Menu definition strings generally follow the format:

<File List or String Prefix>
<Menu Definition String>

Default Menu Definition

The default menu definition is globally applied to each file when no file-specific menu definition applies to the particular file. You may commonly want to use this definition to limit availability of functions for all files in your solution without needing to set specific rules per file or if you are a system administrator and you want to universally disable certain features for your users (New Database, Find/Replace, Replace Contents, Relookup Contents).

By using a default menu definition, you can enforce security globally on new files, without needing to know the name of every file that might be opened. The “default” prefix is “<*>”.

Default String Format:

<*>:
<Menu Definition String>

Disable Example:

<*>:
<1,B,<1,2,3>>
<2,B,<0>>

For security reasons, it often makes sense to create a default disable menu definition.

On the other hand, you may not want to use a default at all. This would be the case if your users could be opening files which are not part of your solution and you want them to have full FileMaker Pro functionality in any files other than ones which you specifically want to control. In this situation, to have a “universal setting” for all of your files only, you would use a “file-specific” menu definition, listing ALL of your files in the prefix (see below for more info about file-specific menu definitions).

Rename Example (the following definition would affect all files (and therefore may or may not be desirable):

```
<*>:
<6,B,
    <
    <0,Widgets>,
    <1,New Widget>,
    <2,Duplicate Widget>,
    <3>Delete Widget>
    >
>
```

In many solutions, it makes sense to rename menus and menu items uniquely for each file. For that reason, you may not want to set a default rename menu definition and, instead, use only file-specific rename menu definitions (see below).

Script Example:

```
<*>:
<6,B,
    <
    <1,My New Record Script>,
    <2,My Duplicate Record Script>,
    <3,My Delete Record Script>
    >
>
```

For a script menu definition to work as a default, you must have scripts with the same names in each file – even if the scripts themselves work differently from to file. Otherwise, you may prefer to only use file-specific script menu definitions (see below).

File-Specific Menu Definitions

A file-specific menu definition becomes active every time one of the files associated with it is brought to the front.

File-Specific String Format:

```
<FileName1,FileName2,etc>:  
<Menu Definition String>
```

File Extensions Generate an Error: SecureFM will return an error for file names that include the file type extension. (This requirement insures cross-platform compatibility in your strings.) For example, if you want to create a file-specific menu definition for a file named “Widgets.fp5”, the string prefix must use only “Widgets” without the “.fp5” extension.

The following examples apply to three files: Widgets.fp5, Widget History.fp5, and Widget Assembly.fp5.

Disable Example:

```
<Widgets,Widget History,Widget Assembly>:  
<1,B,<1,2,3>>  
<2,B,<0>>
```

Rename Example:

```
<Widgets,Widget History,Widget Assembly>:  
<6,B,  
    <  
    <0,Widgets>,  
    <1,New Widget>,  
    <2,Duplicate Widget>,  
    <3>Delete Widget>  
    >  
>
```

Script Example:

```
<Widgets,Widget History,Widget Assembly>:  
<6,B,  
    <  
    <1,My Create New Widget Script>,  
    >  
>
```



```
<2,My Duplicate Widget Script>,  
<3,My Delete Widget Script>  
>  
>
```

If you define a file-specific script menu definition for more than one file, remember that you must have the scripts with the same names in each file – even if the scripts themselves work differently from to file.

Restoring Menus and Special Settings

No Files Open Menu Definition

Finally, you can set up a “no files open” menu definition that applies when the FileMaker Pro application is running but no individual files are open.

You can use a “no files open” menu definition for Menu-Disable and Menu-Rename, but there is no reason to have a “no files open” Menu-Script definition since at least one file must be open to hold the necessary scripts.

If you have a default menu definition active but not a “no files open” menu definition, the default will apply. The prefix for “no files open” is “<~>”.

No Files Open String Format:

```
<~>:  
<Menu Definition String>
```

Disable Example:

```
<~>:  
<1,B,<1,2,3>>  
<2,B,<0>>
```

Rename Example:

```
<~>:  
<6,B,  
    <  
    <0,Widgets>,  
    <1,New Widget>,  
    <2,Duplicate Widget>,  
    <3>Delete Widget>  
    >  
>
```

Since scripts are always associated with an open file, a Menu-Script definition for the situation when no files are open becomes meaningless.

Menu Definition Hierarchy

It is possible to have any or all of the different types of menu definitions set at the same time. Which one will be shown is determined by the following order of precedence:

8. File-specific, if a particular file has been assigned a file-specific menu definition and that file is frontmost. (The No-File configuration <~> is considered a file-specific definition that applies when no files are open.)
9. The default menu definition <*>, if one is defined.
10. If neither a file-specific nor the default applies, then FileMaker Pro's normal menus are shown unchanged.

For example, if a file-specific menu definition has been defined for the file Widgets, and Widgets is the front window, then its file-specific menus are shown. If no menu definition has been set for the frontmost file, then the Default menu definition is shown. If no Default has been defined, then the normal FMP menus are unaltered.

Restoring Menus

The old technique of restoring FileMaker Pro's normal menus – used in previous versions of SecureFM – was to send a blank "" menu definition string with the Menu-Disable command.

To accommodate the several types of menu definitions that can now be active at the same time, SecureFM offers several levels of restoring menus which can be used with Menu-Disable as well as the new Menu-Rename and Menu-Script functions.

Clear All Menu Definitions

If you set Menu-Disable, Menu-Rename, or Menu-Script with a colon followed by a dash “:-” as the parameter, SecureFM will revert back to FileMaker Pro menus, clearing all default, file-specific, and “no file open” menu definitions.

Important Note: Be aware that the Clear All Menu Definitions command affects all SecureFM strings in all files.

Clear All Format:

External ("Menu-Disable", ":-")

External ("Menu-Rename", ":-")

External ("Menu-Script", ":-")

Use FileMaker Pro Menus

You may want to specify FileMaker Pro's normal menus for a specific file. You may do this temporarily (for instance as a developer doing some administrative or development function at a user's computer) or in the instance when you have a default setting applied to all files but want to specifically have normal FileMaker Pro menus for a certain file or files.

To display FileMaker Pro menus, use a blank string "".

It is important to understand that this is not the same as clearing menu definitions. Clear All Menu Definitions ":-" and Clear Current Setting "-" actually remove menu definitions from SecureFM's memory. Using a blank string "", on the other hand, sets a menu definition for the given context telling SecureFM not to override FileMaker Pro's normal menus; it does not clear any other menu definitions that may be set, such as the default setting.

Use FileMaker Pro Menus Format:

<File List>:

(Nothing follows the colon.)

To set a file-specific menu definition to use FileMaker Pro's menus:

External ("Menu-Disable", "<FileName1,FileName2,etc.>:")

External ("Menu-Rename", "<FileName1,FileName2,etc.>:")

External ("Menu-Script", "<FileName1,FileName2,etc.>:")

Clear Current Setting

SecureFM stores an internal table of menu definitions. Sending a dash (think of it as a minus sign) "-" removes the currently stored menu definition for that setting (default, file-specific, or no files open).

For example, "<*>:-" will remove the definition (if any) that had been set as the default.

The menu definition hierarchy (see above) determines the result when clearing existing settings.

Clear Current Setting Format:

<File List or String Prefix>:-

Backward Compatibility

For backward compatibility, the updated version of SecureFM still handles the old menu definition string format (used by pre-6.1 versions of SecureFM).

When no prefix is included with the disable string, it is assumed to be a default menu definition that applies to all files. This means that all old-style SecureFM strings are treated as default menu definitions.

Important: Re-Enable All Syntax Change

The one significant change to the Menu-Disable command that is not backward compatible is the “re-enable all” command.

The old format used to re-enable menus was simply to send a blank parameter with the Menu-Disable function:

External (“Menu-Disable”, “”)

The new format to re-enable all menus for all files requires the special character combination “:-” with Menu-Disable:

External (“Menu-Disable”, “:-”)

In other words, if you are upgrading from a pre-6.1 version of SecureFM, you do not need to change any disable menu definitions, but you must update any “re-enable all” commands.

When all files are closed

If a user closes all file windows but keeps the FileMaker Pro application open, the “no file open” menu definition (see above) will be used. If you have not set a “no files open” definition, then the default definition will be used. If neither a default nor “no file open” menu definition has been set, then FileMaker Pro’s normal menus will become active.

Other SecureFM Functions

Menu-StoreText

External ("Menu-StoreText", parameter)

SFM Available with a SecureFM or MenuMagic license.

The Menu-StoreText function stores a block of text in an internal SecureFM dictionary. It can be used to store a string for use in SecureFM, or for storing any other value for any purpose. It can be useful for defining a variable for use later in a script or at any time during the user's session.

Store Text Parameter Format:

External ("Menu-StoreText", Key|StoredText)

The stored text can be anything and any length, though it may be limited by FileMaker's plug-in API or the total application memory available for FileMaker Pro. The key is case sensitive and can be up to 50 characters. The SecureFM dictionary is limited to 200 entries.

All text stored is globally available. You can store text by calling the Menu-StoreText command and then later retrieve the text using the Menu-GetText function (see below). You can store text in one file and retrieve it in another file. In other words, the SecureFM dictionary functions allows you to store truly global data without being dependent on a particular FileMaker file being open.

Menu-StoreText uses application memory to hold the text. You may need to be aware of memory allocated to FileMaker Pro if you are storing many large blocks of text.

When using this function, it is a good idea to also trap for errors returned by the plug-in.

The stored text dictionary is cleared each time FileMaker Pro is shut down.

Menu-GetText

External ("Menu-GetText", parameter)

SFM Available with a SecureFM or MenuMagic license.

Menu-GetText retrieves text that has been stored by SecureFM using the Menu-StoreText function (see above). The key must match the corresponding key used in the original Menu-StoreText command.

Get Text Parameter Format:

External ("Menu-GetText", Key)

The Menu-GetText function does not return any errors. If no match for the key is found, SecureFM returns a blank response ("")

Creating a Menu Definition Dictionary

The Menu-StoreText and Menu-GetText functions have many possible uses, but one handy application is to create a menu definition dictionary, allowing you to store many menu definitions and activate them by sending a simple key word or phrase, rather than setting elaborate strings.

For example, suppose you want to work with two different disable menu definitions based on a logged in user's privilege level.

For basic access, you might have a string like (oversimplified):

<*>:
<1,A,<0>>

For management level access, your string might be (also oversimplified):

<*>:
<1,B,<5,6,7,8,9,10>>

You can store them in the SecureFM dictionary:

External ("Menu-StoreText", "Basic|<*>:<1,A,<0>>")

External ("Menu-StoreText", "Management|<*>:<1,B,<5,6,7,8,9,10>>")

Then, if a user is logged in with “Basic” privileges, you can easily disable menus correctly without worrying further about menu definition string codes:

External (“Menu-Disable”, External (“Menu-GetText”, “Basic”))

On the other hand, if a user logs in with “Management” privileges, you can easily call up the correct menu definition with:

External (“Menu-Disable”, External (“Menu-GetText”, “Management”))

Note that there may be other, and in some cases preferable, ways to store multiple menu definitions, including in the SecureFM Menu Table file which comes with SecureFM.

Menu-GetMenuSelection

External ("Menu-GetMenuSelection", parameter)

SFM Available with a SecureFM or MenuMagic license.

The Menu-GetMenuSelection function allows you to find out which FileMaker Pro menu item was last selected by a user and in which file. It also gives you the numeric key code of the last keystroke typed.

External ("Menu-GetMenuSelection", "")

No parameter is used.

Menu-GetMenuSelection returns five values in its response, each separated by a pipe character "|":

Menu #|Menu Item #|(unused)|File Name|Key Code

Menu Selection Example

If the last menu item selected by the user was the Open command (menu item #2) in the File menu (menu #1), and this was done in a file named "Customers.fp5", the Menu-GetMenuSelection response might be:

1|2|0|Customers|222

Key Code

The last value in the response is the keycode. The keycode is a numeric value, up to three digits long, that represents the last key typed.

This keycode value can be used by Menu-HotKey to attach hot key functions to keys that do not produce text characters, such as arrow keys, function keys, etc.

When checking for keycode values, be aware that not all keycodes are identical on all platforms. Some keys do indeed have the same value on different platforms, while others, especially the non-text keys, quite often have different values on different platforms.

Unused Third Value

The third value in the response is always zero and is currently unused.

Menu-DoFMPMenuItem

External ("Menu-DoFMPMenuItem", parameter)

MM Available with a MenuMagic license only.

This function gives you the ability to call any FileMaker Pro menu item directly from within a script.

Parameter Format:

External ("Menu-DoFMPMenuItem", Menu#|Item#)

Many FileMaker Pro menu items already have an equivalent script step available, but not all. For example, there are script steps already available to enter Browse, Find, and Preview modes, but there is no script step to enter Layout mode. The Menu-DoFMPMenuItem function lets you directly call the Layout Mode item in the View menu.

Using our Layout mode example above, the View menu is menu # 3 (counting from the left) and Layout Mode is item # 3 (counting from the top). The command would then be:

External ("Menu-DoFMPMenuItem", "3|3")

Preceding Script Step

To work reliably, this function must be preceded by another script step. You can use a Pause/Resume Script step set to zero seconds, but that may cause problems if you have other scripts queued up to run. The solution we recommend is to precede the Menu-DoFMPMenuItem command with a Go to Layout script step:

[Go to Layout \[original layout\]](#)

[Set Field \[Response Field, External \("Menu-DoFMPMenuItem", "3|3"\)\]](#)

Note that the Go to Layout script step will exit the record, causing the cursor to exit the currently selected field. If you want the triggered menu item to act on the selected field, such as with the Paste command, then the preceding script step should not be Go to Layout. A blank Set Field step works well:

[Set Field \[""\]](#)

[Set Field \[Response Field, External \("Menu-DoFMPMenuItem", "2|5"\)\]](#)

Menu-Event

External ("Menu-Event", parameter)

MM Available with a MenuMagic license only.

The Menu-Event function allows you to trap for certain events. There are three specific events which are available:

1. A new file window becoming active
2. When SecureFM is unable to call a script
3. When a user attempts to disable SecureFM in the Application Preferences.

Whenever one of those events occurs, SecureFM can be set to run a specified script.

The Menu-Event function is global and stays active until cleared by sending a new or blank "" event parameter.

External ["Menu-Event", Event Parameter(s)]

The event parameter format varies slightly depending on which types of events are being trapped:

WindowActivate
ScriptError
SFMDisable

Important: Using the Menu-Event command clears all previously set Menu-Event functions. If you intend to use more than one of these functions, you will most commonly want to set all of them at the same time, usually when your solution is launched (see below for more details).

WindowActivate

The specified script is immediately called whenever the active file window changes. This means that whenever a user brings a new file window to the front, the WindowActivate script will run.

The WindowActivate script can be a script with the same name in every file within your solution, or you can call a single script in a central file.

In other words, this powerful function gives you the ability to detect whenever a new file window has become the front-most file – even when the user has manually navigated between files – and will then run any script you specify.

Event Parameter Format:

`<WindowActivate,ScriptName>`

So the syntax for the full function would be:

`External("Menu-Event",<WindowActivate,ScriptName>)`

ScriptError

When a SecureFM function triggers a script – such as when a user selects a menu item that has been modified to call a custom script with the Menu-Script function – but the expected script is not found, a generic dialog will appear explaining the problem. If, however, a "ScriptError" script has been set, and the initial script is not found, the ScriptError script will run instead.

Event Parameter Format:

`<ScriptError,ScriptName>`

For troubleshooting purposes, you may want your ScriptError script to make use of the Menu-GetScriptCaller function (see below) to find out which script failed and why.

The syntax for specifying a file name when referring to a script in SecureFM is to precede the script name with the file name in square brackets, hence:

`<ScriptError,[FileName]ScriptName>`

Failure of the ScriptError Script

If the ScriptError script itself fails, the generic error dialog will appear, showing the original script and which function called it.

Suppressing the ScriptError

There are times when you may not want a script error to be reported to the user, either as SecureFM's generic dialog or as part of a ScriptError script.

In order to avoid these situations, you can set the ScriptError to be suppressed by using a tilde character "~" in place of a script name:

`<ScriptError,~>`

This will prevent any error dialog from appearing.

SFMDisable

This function provides added security to ensure that a user can't disable SecureFM and still have access to your solution.

Event Parameter Format:

`<SFMDisable,ScriptName>`

The specified script will run immediately if any user attempts to disable SecureFM by unchecking it in the Application Preferences plug-ins list. This is important because unchecking SecureFM will immediately re-enable all menu items. If a user tries to turn off SecureFM, you may want a script to run to immediately quit FileMaker Pro. The specified script should probably be in a file which is guaranteed to be open if your solution is running. When that file closes, it should most likely close all files in your solution.

The syntax for specifying a file name when referring to a script in SecureFM is to precede the script name with the file name in square brackets, hence:

`<SFMDisable,[FileName]ScriptName>`

Another option you have to prevent a user from deactivating SecureFM in Application Preferences is to disable the Preferences menu item with the Menu-Disable command. However, if a savvy user creates a FileMaker file with a script which opens Application Preferences, they could still get around your disabling menu definition – so the SFMDisable option of the Menu-Event function will enable you to close that loophole. Airtight security will require careful thought regarding how your “On Open” and “On Close” scripts work in a primary file in your solution. At a minimum, the On Open script should most likely verify that SecureFM is registered, and otherwise not allow your solution to launch.

Menu-Event Resets All Previously Defined Events

Using the Menu-Event command clears all previously set Menu-Event functions. For example, if you use Menu-Event to specify a WindowActivate script and then, in a separate script step, use Menu-Event to set a ScriptError script, the WindowActivate script will be forgotten – unless you also respecify the WindowActivate script, as well.

The Menu-Event command can include any or all of the specifiers (WindowActivate, ScriptError, SFMDisable) in the same step. These specifiers are not separate commands that operate independently of each other. They can be thought of as sub-functions of the Menu-Event command.

For this reason, it usually makes sense to set all three Menu-Event values in the same step:

```
External("Menu-Event",  
"<WindowActivate,ScriptName>  
<ScriptError,ScriptName>  
<SFMDisable,ScriptName>")
```

Under most circumstances, the Menu-Event specifiers are not functions you would set dynamically at different times. It makes sense for most solutions to set all applicable Menu-Event specifiers once in a single Menu-Event command when your solution is launched.

Configuring the Event Commands with a Password

If you have registered SecureFM with a password (see Menu-Register), the same password is required to use the Menu-Event command. To use a password, you must add an additional pipe character "|" and the password to the end of the Menu-Event string.

Ex: Password in a Simple String

For example, if your original Menu-Event string was:

```
External ("Menu-Event", "<WindowActivate,Disable Menus>")
```

It would become:

```
External ("Menu-Event", "<WindowActivate,Disable Menus>|Password")
```

Password Error

If the password is missing or wrong in the Menu-Event string, SecureFM returns "Incorrect password" and does activate the Event command.

Menu-PerformScript

External ("Menu-PerformScript", parameter)

MM Available with a MenuMagic license only.

Menu-PerformScript immediately runs the specified script. With Menu-PerformScript you can trigger a script from any FileMaker Pro calculation - this could be a calculation field, a field validation, an auto-enter calculation, etc...

Parameter Format:

External ("Menu-PerformScript", ScriptName)

Triggering a Script on Exiting a Field

You can trigger a script when a user exits a field. To do this, define a validation by calculation for the field, using the Menu-PerformScript function in the field's validation and set the validation to only evaluate when the field has been modified. Whenever the field is modified (and then exited), the validation will be evaluated and your script will be triggered. It may be a good idea to also include the boolean "1" to make the validation "true." The field validation calculation will look like:

1 or External ("Menu-PerformScript", "Field Exit Script")

To make certain that the specified script runs in the correct record, see the note below on "Optional Parameter for Menu-PerformScript and Menu-GetScriptCaller".

Triggering a Script on Navigating to a Layout

Another use of Menu-PerformScript is to run a script whenever a user navigates to a new layout or changes the current record. This provides complete control over navigation without having to lock the Status Area.

To do this, create a calculation field with the Menu-PerformScript function in it:

External ("Menu-PerformScript", "Layout Navigation Script")

Make sure the calculation is set to be unstored. Make the calculation field large and place it in the background on all applicable layouts. Every time a user switches to that layout or changes records, FileMaker Pro will automatically recalculate the field – triggering the script.

To be sure that the script is not triggered when another file is in front, see the documentation on the Menu-IsFrontWindow function below.

Be careful using this technique if you are also using a WindowActivate script trigger, since you could have two redundant (or conflicting) scripts being triggered by the window coming to the front and by the layout evaluating.

Optional Parameter for Menu-PerformScript and Menu-GetScriptCaller

Additional information can be submitted with Menu-PerformScript, and then retrieved again by Menu-GetScriptCaller. As part of this process, an optional parameter can be submitted with the Menu-PerformScript function:

[External \("Menu-PerformScript", ScriptName|OptionalParameter\)](#)

This optional parameter can contain any data up to 300 characters in length.

You can later retrieve that data using Menu-GetScriptCaller (see below). If the last SecureFM function to call a script was Menu-PerformScript and an optional parameter was included, Menu-GetScriptCaller will return the optional parameter as the third parameter in its response:

[\[File Name\]Script Name|SecureFM Function|Optional Parameter](#)

(For more information on Menu-GetScriptCaller, see the documentation below.)

Why this is needed:

One common way to use the Menu-PerformScript function is to put it in a field validation calculation. In that situation, Menu-PerformScript gets triggered every time the field is modified, and allows the developer to modify other data in the same record based on the field value that was just modified. This works well most of the time.

However, there are times when a field can be exited by means other than simply tabbing to another field or clicking in the background of the layout. A field can be exited by an event which leaves the record – clicking on another record (if in list view), using the record rolodex in FileMaker Pro's Status Area to switch records, running a script which leaves the record, etc. – and all of these events will trigger the script, but the script will operate on the data in the wrong record.

By adding an optional parameter, you can submit information about the status of the database when the script was triggered, such as the current record ID, which can then be used in the triggered script to evaluate if the record is still the same or if you must first return to the correct record.

Menu-GetScriptCaller

External ("Menu-GetScriptCaller", "")

MM Available with a MenuMagic license only.

Menu-GetScriptCaller can be used as part of a ScriptError script (see Menu-Event) to find out which script failed and which SecureFM function attempted to call the script.

External ("Menu-GetScriptCaller", "")

No parameter is used.

Returns the following information:

[File Name]Failed Script Name|SecureFM Function

For example—

"[Widgets]Go to Menu|WindowActivate"

-- means that the attempt to run a script named "Go to Menu" in the Widgets.fp5 file failed. The SecureFM function that attempted to call the script was the WindowActivate specifier for the Menu-Event function. In your ScriptError script, you can parse this result to provide an informative error message, either for a user, or, perhaps more importantly, for yourself when you are developing.

Menu-ExtractParameter

External ("Menu-ExtractParameter", parameter)

SFM Available with a SecureFM or MenuMagic license.

Allows you to easily extract a value from a list separated by pipe characters "|". This is helpful when used in conjunction with Menu-GetScriptCaller or any other plug-in result (not limited to New Millennium plug-ins), that returns multiple values separated by pipe characters.

Parameter Format:

External("Menu-ExtractParameter", Parameter#|ParameterList)

Menu-ExtractParameter views the first parameter ("Parameter#") on its own; all subsequent parameters divided by pipe characters "|" are seen as a separate "Parameter List" string.

The first parameter tells SecureFM which of the subsequent parameters to extract.

For example, if the Menu-ExtractParameter string is "3|Joe|Fred|Nancy|Susan", SecureFM will return "Nancy", the third parameter (after the parameter #).

If the Parameter # specifies a number higher than the number of parameters in the Test String, Export-ExtractParameter returns the last one. If the Parameter Number is zero "0" or a negative number, the number of parameters is returned instead.

Menu-ExtractParameter is even helpful for general use in parsing lists. If you have a list of values separated by return characters, you can first substitute pipe characters for the paragraph returns, and then easily extract the values in the list with this function. This is much easier than wrestling with FileMaker's "Position" and "Middle" functions for multiple values.

Example: Extracting Parameters from Menu-GetScriptCaller

To return the name of the last script triggered by SecureFM, you can use:

External("Menu-ExtractParameter", "1|" & External("Menu-GetScriptCaller", ""))

To get the function that called it:

```
External("Menu-ExtractParameter","2"& External("Menu-GetScriptCaller", ""))
```

Counting the Number of Parameters

Setting the initial parameter to zero “0” tells Menu-ExtractParameter to count the number of parameters separated by pipe characters that follow. For example, sending—

```
External("Menu-ExtractParameter","0|apples|oranges|papayas")
```

—returns “3” to the response field.

When Parameters are Divided by a Different Character

If you have a string of values, but they are separated by a character other than the pipe character, you can still use the Menu-ExtractParameter function. All you need to do is use FileMaker Pro’s Substitute command.

If, for example, you have a field named “Shopping List” that contains a list of values separated by returns—

```
apples  
oranges  
papayas
```

—and you want to extract the first item in the list, you can substitute the pipe character “|” for the return character “¶”:

```
External("Menu-ExtractParameter","1|" & Substitute (Shopping List, "¶", "|"))
```

Menu-IsFrontWindow

External ("Menu-IsFrontWindow", parameter)

SFM Available with a SecureFM or MenuMagic license.

This function quickly tells you if a specific FileMaker file is the front or “active” file window.

Parameter Format:

External("Menu-IsFrontWindow", FileName)

Returns 1 if the FileName parameter is identical to the front window title; 0 otherwise.

Prevent Menu Definitions and PerformScript in Background Files

An advanced (and fairly specialized) technique for using some of the SecureFM functions is to incorporate them into an unstored field calculation and then hiding that field on a layout. You can incorporate Menu-Disable, Menu-Rename, and Menu-Script into the field calculation and every time a user navigates to the layout, FileMaker Pro will automatically recalculate the field -- triggering the SecureFM functions. In other words, this technique can dynamically reconfigure FileMaker Pro's menus whenever a user switches layouts.

You can also use this technique in conjunction with Menu-PerformScript to trigger a script each time a user goes to a different layout (see above in the description of Menu-PerformScript).

This field calculation technique can help you create highly dynamic environments, but it has one potential problem, as well:

FileMaker Pro sometimes recalculates fields that appear on layouts, even if the file is not the front file window. If one of these SecureFM functions is in an unstored field calculation on a layout, this can cause menus to be modified or scripts to run unexpectedly. You can also get errors with Menu-PerformScript because the script being called is in the background file but a different file is the front window.

The Menu-IsFrontWindow function allows you to prevent this problem. You can incorporate Menu-IsFrontWindow into your field calculation to make sure Menu-Disable, Menu-PerformScript, etc. are triggered only when the file is the front window.

Let's take an example:

Suppose you want the Window menu (menu 8) to be disabled every time a user navigates to a particular layout. You can create an unstored calculation field with the following disable string:

`External ("Menu-Disable", "<*>:<8,A,<0>>")`

You can then place that field on the layout, hiding it if you like. Then, every time a user navigates to that layout through any means, FileMaker Pro will automatically recalculate the field, triggering the Menu-Disable command to disable the Window menu.

The potential problem is that, under certain circumstances, FileMaker might also attempt to recalculate the field, and thus disable the Window menu, even if another file window is in front.

We need to rework the string so that FileMaker Pro's attempt to recalculate it will not trigger Menu-Disable if the file is not the front window.

To do this, the Menu-Disable function must be completely removed from the field's calculation and from any dependent fields. The way to solve this is to store the Menu-Disable string (shown above) as text in a field which is referenced by a GetField function. If the name of the file we expect to be the front window is Widgets.fp5, then our unstored calculation field becomes:

`GetField(
Case(External("Menu-IsFrontWindow", "Widgets"), Field Name of Field Which
Contains the Disable Function))`

Setting the calculation up that way means that, when Widgets.fp5 is the front window, Menu-IsFrontWindow is positive and the Menu-Disable string in the field which is referred to in the GetField function will be triggered. If, however, another file is the front window, Menu-IsFrontWindow is negative and the Menu-Disable function is never even evaluated (nor triggered).

The same approach can work with Menu-PerformScript or any other plug-in function you incorporate into an unstored calculation field on a layout.

Menu-FMPWindow

External ("Menu-FMPWindow", "")

SFM Available with a SecureFM or MenuMagic license.

This function gives you the ability to move and size the FileMaker application window.

This is a Windows-only function since there is no separate application window on the Macintosh platform.

Script Parameter Format:

`left|top|width|height`

Setting Position and Dimensions

To position the top left corner of the application window at 20, 20 with dimensions of 800 x 600, the string would be:

`20|20|800|600`

You can use FileMaker Pro's built-in functions to get the current screen height and screen width and position the application window specific to each display. You can, for example, calculate how to precisely center the application window.

Maximizing the Application Window

To maximize the application window, use the following string:

`-1|-1||`

SecureFM's Password Feature

Password Registration

SecureFM's Password is an advanced feature that gives you an added layer of security, allowing you to ensure that only the solution creator can control FileMaker's menus and toolbars.

Using a password is entirely OPTIONAL. This makes SecureFM backward compatible with plug-in commands designed to run on earlier versions of SecureFM. It also means you can simply choose not to add a password to your solution if you feel that this advanced level of security isn't necessary for your solution.

Password for Added Security

Registering SecureFM with a password prevents users from re-configuring the menus unless they know the password. When SecureFM is registered with a password, that password must also be used to change menu configurations and toolbar access.

If you don't use a password, a FileMaker-skilled user who has script access to any FileMaker file and understands the syntax of SecureFM could enable all menu items. Even a novice user who has access to the SecureFM Menu Editor file (which can be downloaded by anyone from New Millennium's web site) can simply hit the "Re-enable All" button. Registering SecureFM with a password prevents these possibilities.

For details on how to use passwords with SecureFM, see the documentation on Menu-Register and any other SecureFM functions you wish to use in your solution.

Using SecureFM with Runtime Solutions

SecureFM can be used to add an increased layer of security for your stand-alone FileMaker Runtime solutions.

Creating a Plug-Ins Folder for Runtime Solutions

In a normal (non-runtime) FileMaker solution, plug-ins must be placed in the FileMaker Extensions folder (on Mac) or the System folder contained in the FileMaker folder (on Windows). The same is true for runtime solutions, but you must manually create these folders.

To create a plug-ins folder for your runtime solution:

- Within the folder containing your runtime solution, create a new folder.



If your runtime solution is designed to run on Macs, name the folder "FileMaker Extensions".



If your solution will run on Windows, name the folder "System". Then place your copy of SecureFM into the new folder.

- When you relaunch your runtime solution, check the plug-in preferences in the Edit menu. Make sure SecureFM is listed among the plug-ins, and that it is marked as enabled.

Menu Differences in Runtime Solutions

Remember that SecureFM disables FileMaker's menu items based on their numeric location, not by name. Since some menus in a stand-alone runtime solution are different from standard FileMaker menus, a configuration designed to run on a standard FileMaker database often will not modify the same items in a runtime solution.

Just as with different versions of FileMaker Pro and FileMaker Developer, runtime files can require a menu definition string unique to its menus. You can calculate the correct string using the SecureFM Menu Editor file or by manually adjusting strings created for other versions of FileMaker menus.

Menus Affected by the Runtime Binding Process

Four menus get modified when turning your FileMaker solution into a stand-alone runtime solution: File, View, Scripts, and Help. If your SecureFM string does not modify any of these menus, then it should not require any modification to work in your runtime solution.

- File Menu

Many items get removed from the File menu in runtime solutions, leaving users with many fewer functions.

- View Menu

The View menu remains identical, except that the Layout Mode menu item is completely removed. This changes the item numbers for all subsequent items in the menu, reducing their item number by one.

- Scripts Menu

In runtime solutions, the ScriptMaker menu item (and the subsequent divider) get removed, so you only see the displayed scripts.

- Help Menu

The FileMaker-specific Help menu items get removed. If, during the binding process, you specified a script to act as a Help script, the script's name will be added to the bottom of the Help menu items.

If your SecureFM configuration affects one of these four menus, the menu items may need to be adjusted to work properly on the runtime solution's menus. Remember that the menu items are counted from top to bottom. (And don't forget that the horizontal menu dividers count as menu items.)

Troubleshooting

If you are encountering unexpected results when using SecureFM with your FileMaker solution, try the following troubleshooting steps to help pinpoint the source of the problem.

- Make sure there is only one version of SecureFM installed

Having two different versions of SecureFM installed at the same time may cause problems. Delete any earlier versions of SecureFM. Simply dragging the new plug-in into the correct folder will not assure that the old version will be replaced because their names may be different.

- Make sure SecureFM is installed correctly

If SecureFM appears to not be doing anything, make sure the plug-in is installed correctly.



Installation on Macintosh:

Place the SecureFM plug-in into the FileMaker Extensions folder inside your FileMaker Pro folder.



Installation on Windows:

Place the SecureFM.fmx plug-in into the FileMaker System folder inside your FileMaker Pro folder.

Restart FileMaker after installing the plug-in.

Make sure SecureFM is selected in the Plug-Ins section of the Application Preferences (found in the Edit Menu on Windows and Mac OS 9 and in the FileMaker menu on Mac OS X).

- Restart FileMaker

If you are using an unregistered version of SecureFM, it ceases to function 60 minutes after FileMaker Pro is launched. To restore SecureFM functions before you are ready to purchase a user license, you must restart FileMaker Pro.

- Check the Registration Status

Make sure that SecureFM is actually installed and registered. You can do this by attempting to re-register SecureFM:

```
Set Field ["Registration Response Field", "External ("Menu-Register",  
"LicenseeName|MacRegistrationCode|WinRegistrationCode")"]
```

If the response in the Registration Response Field is blank, SecureFM is not installed; in which case, none of the SecureFM functions will work properly. You should first make sure that the SecureFM plug-in is placed into the System folder found within the FileMaker folder (on Windows), or in the FileMaker Extensions folder (on Mac). After you have done that, go to the Application Preferences. Select the Plug-Ins tab and make sure that SecureFM is in the list and checked.

If the Registration Response Field is showing an "Invalid Registration", SecureFM is running in Demo mode. Double check your registration codes if you have already purchased a user license.

MenuMagic Registration Note: When the plug-in is registered with a basic SecureFM license key, the core security functions (Menu-Disable, Menu-Toolbar, etc.) will work fine, but the MenuMagic functions (Menu-Rename, Menu-Script, Menu-Event, etc.) will run in demo mode for only 1 hour from launch. If some of the plug-in functions appear to be working properly but the new MenuMagic functions are not, make sure that you are registering the plug-in with the correct license key.

- Check the SecureFM response.

When an error occurs, SecureFM will normally return a response in your Set Field step. You can pause your script and visually check SecureFM's response before proceeding.

For example:

```
Set Field ["Response Field", "External ("Menu-Disable", "<*>:<1,B,<1,2,3>>  
<7,B,<1>>")"]  
Pause/Resume Script
```

Or, build an error check directly into your script:

```
Set Field ["Response Field", "External ("Menu-Disable", "<*>:<1,B,<1,2,3>>  
<7,B,<1>>")"]
```

```
If [not IsEmpty (Response Field)]  
  Beep  
  Show Message ["An error has occurred."  
  Halt Script  
End If
```

- When attempting to configure FileMaker's menus, SecureFM gives the response "Incorrect Password".

This indicates that there is an inconsistency between the password used when registering SecureFM and the password used in your current SecureFM command. This can happen in the following situations:

You attempted to re-register SecureFM with a new password after already registering SecureFM with a different password.

You must first clear the password field, quit FileMaker and then relaunch the SecureFM Menu Editor file before you can enter a new password.

(The only time you can register with a new password without having to first restart FileMaker is if you initially registered without a password and later want to register with a password.)

You attempted to configure FileMaker's menus or toolbars, but the password in the Menu-Disable, Menu-Rename, Menu-Script, or Menu-Toolbar step does not match the password used in the Menu-Register step.

Make sure the password is at the end of the configuration string and that it matches the password used in the Menu-Register step.

(If 'Incorrect Password' appears twice in SecureFM's response, it is probably because you are attempting use multiple functions in the same step and both have the incorrect password.)

For more information about SecureFM's password capabilities, see the Password section of the SecureFM documentation.

- When attempting to configure FileMaker's menus, SecureFM gives the response "\$999:SetToolbar is not implemented for OS X".

This response occurs only when you are attempting to use the Menu-Toolbar command on Mac OS X. In FileMaker Pro 5.5 and 6 for Macintosh OS X, the toolbars

are not available. Attempting to use the Menu-Toolbar command to disable toolbars on OS X will cause an error to be returned.

If the Menu-Toolbar command is included as part of a Set Field step that also includes the Menu-Disable command, FileMaker's menus will still be properly modified. In other words, the error message only notifies you that the toolbars cannot be modified on OS X, but all other SecureFM functions in the step will still be implemented.

Your script can either ignore the response or make the Menu-Toolbar conditional based on Status (CurrentPlatform) <> -1.

- FileMaker gives the message "This text constant does not end with a quotation mark (") or is longer than 253 characters" when you attempt to enter a string into a Set Field script step.

For some menu configurations, the length of the string can become quite long. This can create a problem because FileMaker can only process strings with 253 characters or fewer contained between two quotes.

The solution is to break the string up into two (or more) strings separated by an ampersand (&).

For example, suppose you have a long string like:

```
" ...  
<2,B,<2,8>>  
<2,F,<1,3,4,5,9,11>>  
<2,L,<1,5,6,9,10>>  
<2,P,<4,6,7,10>>  
<3,A,<1>>  
..."
```

You can break it into two strings separated by an ampersand:

```
" ...  
<2,B,<2,8>>  
<2,F,<1,3,4,5,9,11>>  
<2,L,<1,5,6,9,10>>"  
&  
"<2,P,<4,6,7,10>>  
<3,A,<1>>  
..."
```


As long as you have fewer than 253 characters contained between two quotation marks, FileMaker will process the string without a problem.

- The toolbars don't always disappear when you use the Menu-Toolbar command.

You probably need to add a Pause/Resume step to your script. Because of the way that SecureFM disables the toolbars, one or both toolbars may not be properly removed when you are using the Menu-Toolbar command in a script **AND**:

- a) The Menu-Toolbar command is the first script step, **OR**
- b) The Menu-Toolbar command is followed by multiple script steps or by a step which causes the script to not be completed without user interaction, such as Show Message.

It is extremely important that you test any scripts using the Menu-Toolbar command on all platforms. If you discover any problems, there is a simple solution: using FileMaker's Pause/Resume Script step.

We have found that it works to use a Pause/Resume Script step, set to a duration of 0:00:03 (three seconds). Place this script step immediately **AFTER** the Menu-Toolbar command:

```
Set Field [SFM Response, "External ("Menu-Toolbar", "0|0")]  
Pause/Resume Script [0:00:03]  
Beep  
Show Message ["OK", "We just removed the toolbars!"]  
If [Status (CurrentMessageChoice) = 1]...
```

Setting the duration to one or two seconds doesn't always work reliably.

- The closebox is disabled when your files run on a Mac, but the closebox isn't disabled when you move your files to a Windows machine.

SecureFM automatically deactivates the closebox that appears in the window title bar of Macintosh files when the Close command is disabled under the File menu (<1,A,<3>>). However, this will NOT disable the closebox on FileMaker files running on Windows.

The reason for this is that there are actually two closeboxes in Windows: a closebox for the file window, and a closebox for the application window. The Windows closeboxes are not disabled when the Close command is disabled under the File menu. To deactivate the file window's closebox, you must disable the Close

command in the FileSys menu (<0,A,<7>>). And to deactivate the application window's closebox, you must disable the Close command in the AppSys menu (<-1,A,<7>>). Or, you can completely disable the FileSys and AppSys menus:

<Filename>:

<0,A,<0>>

<-1,A,<0>>

If you want to make sure that all closeboxes and Close commands are disabled on all platforms, try the following configuration:

External ("Menu-Disable",

Case (

Status (CurrentPlatform) = 1,

"<Filename>:

<1,A,<4>>

<2,B,<12>>

<2,F,<12>>

<2,L,<13>>

<2,P,<12>>",

Status (CurrentPlatform) = -1,

"<Filename>:

<1,A,<4>>

<-1,A,<3>>",

Abs (Status (CurrentPlatform)) = 2,

"<Filename>:

<1,A,<4>

<2,B,<14>

<2,F,<14>

<2,L,<15>

<2,P,<14>

<0,A,<7>

<-1,A,<7>")

)

If you have further questions about SecureFM, please visit our SecureFM FAQs web page at <<http://www.newmillennium.com>>



New Millennium Communications is a software development company located in Boulder, Colorado. We specialize in making tools and advanced templates for FileMaker Pro developers.

New Millennium Communications
1332 Pearl Street
Boulder, CO 80302 USA
303-444-1476

www.newmillennium.com

plug-ins@nmci.com