

SecureFM Revision History

SecureFM 6.4.2

Mac OS Tiger Compatibility

SecureFM 6.4.2 is compatible with Mac OS Tiger 10.4.x (as well as Panther 10.3.x)

Previous versions of SecureFM 6 cause FileMaker to shut down on launch when running in Tiger.

Known Issue: Menu Title Graying on Tiger

Normally, when disabling an entire menu < 0 > or < -1 > the menu title (“File”, “Edit”, etc.) is grayed out, along with the individual menu items. When running on Tiger, the menu title is not grayed when the menu is disabled. The menu items are still correctly disabled and grayed, however.

SecureFM 6.4.1

SecureFM 6.4.1 was used for internal testing purposes and was not publicly released.

SecureFM 6.4.0

New Function:

Menu-FMPWindow Function on Windows

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.

Bug Fix:

Network Registration Bug Fix

Fixes a registration bug in SecureFM that could result in an incorrect count of other registered users. This enabled some users to illegally exceed their license limits.

SecureFM 6.3.3

Important: OS Compatibility and Security

SecureFM 6.3.3 with MenuMagic is designed to work with Windows XP and Mac OS 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.

Bug Fixes:

Disabling for Paused Find Fix

A bug in SecureFM 6.3.1/6.3.2 could cause disabled menu items to be incorrectly re-enabled during a scripted Find with a pause. SecureFM 6.3.3 fixes that loophole.

Scripted Mode Change Bug Fixed for Windows

SecureFM 6.3.1 had a bug that could prevent it from recognizing a change in modes (Browse, Find, Layout, Preview) when the mode change was initiated by a script step. This problem could prevent menus from correctly updating as FileMaker entered the new mode. SecureFM 6.3.2 fixed this bug for Mac OS X. SecureFM 6.3.3 fixes it for Windows.

Mac Statusbar Error Removed

SecureFM 6.3.2 added an error note when attempting to call Menu-Statusbar on Mac OS X stating that the function only applies to the Windows platform. Although the error message was accurate, the SecureFM Menu Editor file generates cross-platform strings that will trigger Menu-Statusbar on either platform under the assumption that it is simply ignored on the Mac platform. With the change in 6.3.2, that caused some confusing errors. SecureFM 6.3.3 returns to the original behavior of ignoring Menu-Statusbar on Mac without returning an error message.

Hot Key Change: Caps Lock Ignored as Modifier

With previous versions of SecureFM/MenuMagic, you often had to define each Hot Key combination twice: one with the primary key and associated modifier keys, and an identical combination that also included the caps lock key. The reason was that FileMaker Pro interpreted the caps lock key as an additional modifier key. If you wanted your Hot Key script to be triggered regardless of whether or not the caps lock key was down, you needed both variations in the Menu-HotKey string.

SecureFM 6.3.3 changes its behavior to now ignore the status of the caps lock key. If, for example, you define a script to be triggered by Ctrl – d with a Hot Key string like –

```
<*>:<B,<<4,d,MyScript>>>
```

-- it will work whether or not the caps lock key is down, without requiring an extra line in the code.

Demo File Fix:

Hot Key String Calculation Fix

When calculating Hot Key strings that are not identical on all platforms, the SecureFM Menu Editor file was mistakenly leaving out a closing parenthesis “)”. The string could still be applied and tested by the Editor file, but it could not be pasted into a script step or calculation. This has now been corrected.

SecureFM 6.3.2 (Mac OS X only)

Bug Fixes:

Replace Keyboard Command Disabled on Mac

When the Replace Contents command was disabled in the Records menu, SecureFM was not correctly disabling the corresponding keyboard command (Cmd =) in SecureFM 6.3.1 for Mac OS X. (This problem did not occur on Windows.) SecureFM 6.3.2 fixes this bug.

Scripted Mode Change Bug Fixed

SecureFM 6.3.1 had a bug that could prevent it from recognizing a change in modes (Browse, Find, Layout, Preview) when the mode change was initiated by a script step. This problem could prevent menus from correctly updating as FileMaker entered the new mode. SecureFM 6.3.2 fixes this bug.

SecureFM 6.3.1

Bug Fixes:

Close Box / Scripted Mode Items Conflict Fixed

SecureFM 6.3.1 fixes a bug that disabled the window close box when one of the Mode items (Browse, Find, Layout, Preview) was scripted

Registration after Demo Period Has Expired Bug Fixed

If you waited to register SecureFM until after the 1 hour demo period had expired, SecureFM 6.3.0 would not register. This bug has now been fixed.

Network Registration Bug Fixed

A bug that could cause problems with registration when on a networked computer has also been fixed.

Menu-PerformScript Bugs Fixed

A bug existed in SecureFM 6.3.0 for Windows that could cause Menu-PerformScript to fail to find its intended script when the function was called in one file, the file was closed, and then the function was called in a second file. SecureFM 6.3.1 fixes this bug.

A second PerformScript bug in SFM 6.3.0 for Windows was also fixed that could sometimes cause Menu-PerformScript to fail when

- the FMP application was launched directly
- a file was opened in the initial open file dialog box
- and the Menu-PerformScript function was triggered in that initial file without first navigating to other subsequently opened files.

SecureFM 6.3.1 fixes this bug, as well.

SecureFM 6.3.0

Registration and Licensing Changes:

Important: Registration Change

A significant registration change begins with SecureFM 6.3.0. Any invalid registration string will void the current registration UNLESS the original registration included 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.

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

As before, sending a blank "" registration string will return the current registration status information but will not upgrade, downgrade, or invalidate the registration status.

Bug Fixes:

Windows Scripts Menu Renaming Bug Fixed (MenuMagic)

Renaming the Scripts menu – not the items within the menu, but the menu title itself – was unreliable in the Windows version of SecureFM 6.2.0. That bug is now fixed.

Windows XP/2000 Scripts Menu Removal Bug Fixed

SecureFM 6.2.0 had a bug that made removing the Scripts menu on Windows XP and 2000 unreliable. The problem did not occur on other versions of Windows and simply disabling the Scripts menu worked properly. SecureFM 6.3.0 fixes this bug.

SetStartup Old-Style Syntax Bug Fixed

Disable strings stored by the SecureFM.ini file created by the Menu-SetStartup command did not work when launching the FileMaker Pro application if the original disable string was an old-style string. Menu-SetStartup only worked with the new-style string syntax. That bug is fixed in SecureFM 6.3.0.

New Functions:

Menu-Version

The new Menu-Version function returns just the version number of SecureFM, such as "6.3.0" without any other text. This makes it much easier to determine the version of SecureFM currently in use. In previous versions, the only way to determine version number was to weed out the extra text returned by the Menu-Register function.

Menu-Menubar

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.

Warning: Be Careful Not to Get Trapped!

Because this function globally removes all FMP 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.

Menu-ExtractParameter (MenuMagic)

This function 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.

The format used by Menu-ExtractParameter is:

External ("Menu-ExtractParameter"; "Parameter#|TestString")

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

Function Changes and Enhancements:

Menu-HotKey Enhancements (MenuMagic)

Keycodes and Non-Text Keys

You can now define the primary key used by the Menu-HotKey function by a keycode number (as well as the old method of typing the character itself). 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 is a three digit numeral that represents each key on the platform keyboard. You can, for example, use "097" instead "a" for the primary key in the Menu-HotKey command. More importantly, you can now refer to non-text keys, such as the arrow keys, and reconfigure them with this function.

To find out the correct keycode value for a specific key, see the table included with the SecureFM User Guide or use the Menu-GetMenuSelection function.

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 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.

Blank Modifier Key Parameter Returns an Error

In SecureFM 6.2.0, the modifier key parameter could be left blank when you did not want a modifier key (Command, Control, Alt, etc.) to be required along with the primary key to trigger a Hot Key script. (For example, "<,a,ScriptName>") With SecureFM 6.3.0, a blank modifier key parameter will cause SecureFM to return an error. If you want the Hot Key script to be triggered when the primary key alone is typed, then you must enter a zero into the modifier key parameter ("<0,a,ScriptName>").

'Script not Found' Error is Not Captured in SecureFM 6.3

With previous versions, when a MenuMagic function attempted to call a script that could not be found, the plug-in would show a message that the script could not be found. That error dialog no longer appears. This change has allowed us to fix the record exiting issue (see below).

Menu-GetMenuSelection Changes

Menu-GetMenuSelection on OS X

Menu-GetMenuSelection now works on OS X. Previously, the function only worked on the Windows platform.

New Keycode Parameter in Response

A new parameter has been added to the Menu-GetMenuSelection response to show the keycode of the last key typed. The keycode is a numeric value, up to three digits long, that represents the last key typed. This keycode value can then 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.

The format now returned is:

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

SecureFM Script Calls No Longer Cause the Record to be Exited (MenuMagic)

Previously, when SecureFM called a script with one of its functions (Menu-Script, Menu-HotKey, Menu-PerformScript, etc.), all fields were exited as if an Exit Record/Request script step had first been executed. This could cause problems with scripts that were supposed to affect the currently selected field, such as a scripted paste, etc. This has now been fixed.

Restore Window Widget Button Can Now be Disabled and Scripted

Beginning with SecureFM 6.3.0, you can disable or script the Restore Window widget button that appears in the file window title bar. SecureFM treats it 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>>>

To script it, use:

<8,B,<<2,ScriptName>>>>

The other window widget buttons (close, minimize, maximize) can already be controlled by disabling or scripting their corresponding menu items within FileMaker Pro's menus.

Scripts Menu Can be Renamed, but Not Menu Items (MenuMagic)

You can safely rename the Scripts menu (menu #7), but attempting to rename menu items in the Scripts menu will cause SecureFM to respond with an error.

Changes in Ability to Control OS X App Menu

Renaming the OS X "FileMaker Pro" Menu (MenuMagic)

In previous version of SecureFM, we warned that you should not attempt to rename the "FileMaker Pro" application menu on OS X because a duplication of the menu could occur in certain circumstances. That bug has been fixed with SecureFM 6.3.0. You can now rename the OS X application menu and its items.

Disabling "Services" 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 new Menu-Menubar function.

Arrange Menu only Modifiable for Layout Mode

Although the Arrange menu (menu #10) only appears in Layout mode, past versions of SecureFM allowed you to set strings for any mode, while treating "L" (Layout mode) and "A" (all modes) as being identical.. Beginning with SecureFM 6.3.0, you can only modify the Arrange menu for Layout mode. Specifying any other mode for the Arrange menu, including "All", will result in an error.

Cmd-? Can Be Disabled and Scripted

Cmd-? is now disabled when FileMaker Pro Help is disabled in the Help menu on OS X. It will also correctly call a script if the menu item has been scripted. This did not work in previous versions of SecureFM.

Additional Find Mode Contextual Menu Items Scriptable on OS X (MenuMagic)

The following Find contextual menu items are now correctly scripted in OS X when the corresponding menu items are scripted in the Requests menu (menu #6) in Find mode: Add New Request, Duplicate Request, Delete Request.

Menu Editor File Changes and Fixes:

Statusbar Function Removed from Disable String

Previous versions of the SecureFM Menu Editor/"Configurator" file were incorporating the Menu-Statusbar string into the cross-platform Menu-Disable string with an evaluation of the current platform. But, because FileMaker Pro 6 evaluates all sides of a calculation, even the segments that are not found to be "true," the Menu-Statusbar command was being incorrectly called on all platforms. Since the Menu-Statusbar function only applies to Windows, an error would be returned on the Mac platform when the Statusbar function was included. (The rest of the string would still be correctly activated, however.

The updated SecureFM Menu Editor file does not include the Statusbar string in the cross-platform Menu-Disable string. You can, instead, copy the automatically generated Menu-Statusbar string and paste it into a separate Set Field script step.

SecureFM 6.2.0

Re-Enable All Syntax Backward Compatibility Restored

Blank Menu-Disable command now restores the default (but not the file-specific) menu definition, making it backward compatible with the older syntax.

SecureFM 6.1.0

Registration

Beginning with version 6.1, SecureFM can now take two different registration strings. A 'SecureFM' registration key only permanently activates the 'Disable' function. The 'SecureFM with MenuMagic' registration key activates all of the other functions – 'Rename', 'Script', 'Event', 'PerformScript' and the others. 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.

Disable String Syntax Changes

SecureFM versions 6.1 and later use a slightly different syntax for the Menu-Disable command.

The old Menu-Disable strings used the following format:

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

The new format for the Menu-Disable string is:

```
<FILE LIST>:  
<1,B,<1,2,3>>  
<2,B,<0>>
```

The New File List Prefix

Beginning with SecureFM 6.1, menu definition strings require a prefix in addition to the actual code that defines which menu items are disabled. This new prefix gives you the ability to create menu definitions which are active only for specific files in your solution. There is also a specific prefix (<*>) which allows you to define a default menu definition which is applied globally to all files.

This 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. The default menu definition is globally applied to each file when no file-specific menu definition applies to the particular file.

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

You can even set up a ‘no files open’ menu definition that applies when the FileMaker Pro application is running but no individual files are open.

Each of these options are defined by using a unique “string prefix” at the beginning of the SecureFM string, either a comma-separated list of the files to which the menu definition should be applied, or a specific symbol for the “default” <*> or for “no files open” <~>.

For details on how to work with these different options, see the User Guide documentation.

Backward Compatibility

For backward compatibility, the updated version of SecureFM still handles the old menu definition string format.

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

6.1 only! Does not apply to 6.2.0 and subsequent versions.

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

The old format 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 SecureFM 6.0.x or earlier, you do not need to change any disable menu definitions, but you must update any “re-enable all” commands, if you put them into your solution, or if you want to continue to use an older SecureFM Editor file with the newer SecureFM plug-in.

Restored in 6.2.0

Blank Menu-Disable command now restores the default (but not the file-specific) menu definition, making it backward compatible with the older syntax.

Final Pipe Character When No Password is Used

In SecureFM 5.0.6 through 6.0.x, the final pipe character “|” was considered optional when no password was used with Menu-Register, Menu-Disable, etc. Beginning with SecureFM 6.1, however, the plug-in will return an error if a final pipe character is NOT followed by a password:

```
External ("Menu-Register", "My Company|MACCODE|WINDOWSCODE|")
External ("Menu-Disable", "<*>:<1,A,<1,2,3>|")
External ("Menu-Toolbar", "0|0|")
```

When no password is used, no final pipe character should be used to separate the rest of the string from the non-existent password:

```
External ("Menu-Register", "My Company|MACCODE|WINDOWSCODE")
External ("Menu-Disable", "<*>:<1,A,<1,2,3>")
External ("Menu-Toolbar", "0|0")
```

The ‘Configurator’ is now the ‘Menu Editor’

Because the new MenuMagic functions allow you to radically redesign FileMaker Pro’s menus for your solutions, the ‘Configurator’ in the Editor file has been expanded to allow you to build your own custom menus and is now referred to as the ‘Menu Editor.’

Because the terms “configurations” and “configuration strings” were not always self-explanatory, we now use the phrase “menu definition.”

Interface Changes in the Menu Editor

The SecureFM Menu Editor file now has separate tabs for each FileMaker Pro menu.

Also, for each menu, there are now radio buttons to toggle between views for each mode, as well as a primary ‘Overview’ that displays what will apply to all modes. We needed to do this because your custom menu item names and scripts will often be specific to a particular mode. For example, you may call a custom script for Delete Record in Browse mode, but you probably won’t want that same script to be triggered by selecting Delete Layout in Layout mode.

For a full explanation of new features in the Menu Editor, see the ‘Getting Started’ documentation.

“Configurations” are now “Menu Definitions”

In previous SecureFM documentation we referred to configurations and configuration strings. Because this language was not always self-explanatory to users, we have adopted the phrase “menu definition” for greater clarity. The meaning, however, is the same.