

SAVEit

Version 1.5

A Text File Saving Plug-in for Filemaker 4.x / 5.x

Developed by Comm-Unity Networking Systems

**Saving Text with
FileMaker is fast
and easy!**



SAVEit Support Email:
saveit@comm-unity.net

Support Website:
<http://filemaker.comm-unity.com/>

Comm-Unity Networking Systems

Table of Contents

Introduction	3
Installation and Configuration	4
Basics	4
After Save	5
About	6
Registering	7
How to Use FileMaker Plug-ins	8
How to Use the SAVEit Plug-in	10
SAVEit External Function Reference	12
Understanding Error Responses	14
Credits	15
Contact Information	15

SAVEit

Introduction

SAVEit is a very simple utility plug-in that will allow you to save information from FileMaker to a text document. Because you can use the set field script step to activate plug-in functions, SAVEit becomes extremely flexible.

Features

SAVEit's Features include:

- Dynamically set the file name
- Dynamically set the path
- Dynamically create directories/folders
- Dynamically collect and save data from current and related databases
- Dynamically tell SAVEit what to do if the file already exists
- Save text via a dialog box
- Register SAVEit via a function. (Mostly for developers)

Possible uses include, but is not limited to:

- Dynamic website "Include" files for use with FileMaker's Web Companion
- Archive records as text files
- Create a static web site from a set of database records
- Save delimited files for import to other programs
- Save a daily, weekly, or monthly activity log

Installation and Configuration

To install the plug-in, first make sure FileMaker Pro is closed. Next, unzip the SAVEit zip file on Windows, or unstuff the SAVEit stuffit file on Macintosh, and then double-click the SAVEit_Installer application. This will automatically place the SAVEit plug-in file into the "System" folder on Windows, or the "FileMaker Extensions" folder on Macintosh, inside your FileMaker Pro 4.x or 5.x folder. If you have a previous version of SAVEit, the installer will overwrite it. (Note to Windows Users: The Installer will not run from inside an Unzipping program like WinZip or ZipMagic. You must unzip it to a folder before running the installer.)

After you install the plug-in as described above, open FileMaker and set the default preferences. To do this, go to Edit->Preferences->Application, click on the Plug-ins tab, and double-click the SAVEit plug-in.

Basics

Once the Configuration Dialog is open, click the "Basics" tab (See Figure 1). On this tab you can set the Default File Path either by typing it in or by clicking on the "..." button to use a dialog. You can also set the Default File Name here, which SAVEit uses when you have not

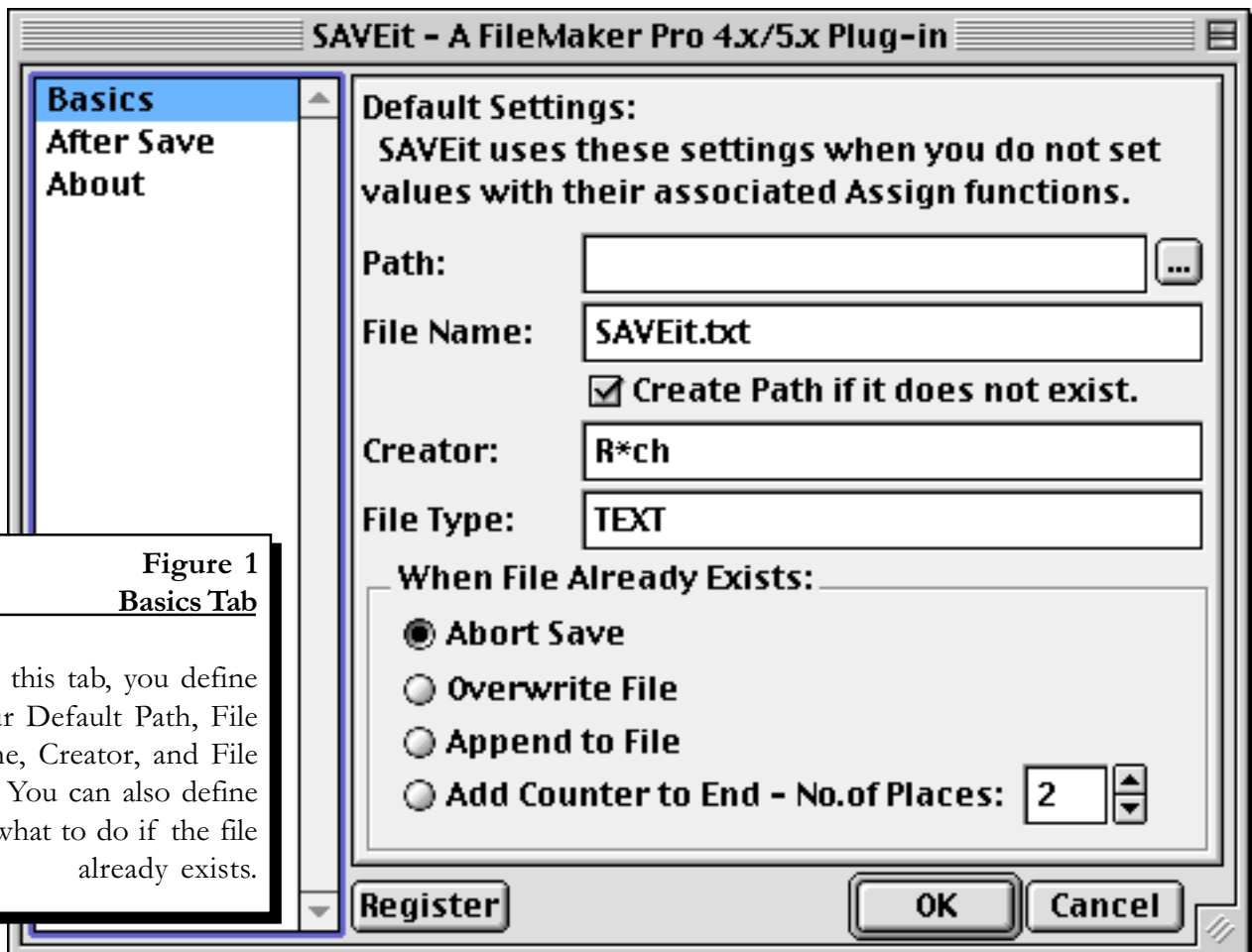


Figure 1
Basics Tab

On this tab, you define your Default Path, File Name, Creator, and File Type. You can also define what to do if the file already exists.

used "SAVE-FileNameAssign" to assign a file name. If the path that you are trying to save to does not exist, SAVEit can create it for you if you check the "Create path if it does not exist" check box. (Note: If the path does not exist, and this is not checked, SAVEit will return an error to FileMaker Pro.)

There are four actions that SAVEit can take when the file you are saving to already exists. You can specify which one of the actions to take by selecting one of the four options on the Basics tab. The first option simply aborts the save and returns an error to FileMaker Pro. The second option will erase the existing file and save the new file in its place. The third option will append the new text to the old file. Finally, the fourth option is to save the file with a unique name based on the settings. For example, if there is already a file named "SAVEit.txt", SAVEit will save the new text with the name "SAVEit01.txt" if the place setting is 2. If you save again, SAVEit will save the new text with the name "SAVEit02.txt". You can change the number of places that SAVEit will use to rename the file that you are saving when you use the fourth option. If the setting is 1 you can have 10 possible unique file names. If the setting is 2, you can have 100 possible unique file names. The setting can go up to 4 allowing 10,000 possible unique names.

On the Macintosh, you have two extra settings for SAVEit. These are the Creator and File Type of the file that SAVEit saves. By default, these are "R*ch" and "TEXT", which usually opens in BBEdit. You can change these settings so that you can open the file in the application that you want.

After Save

The "After Save" settings (See Figure 2) will determine what action SAVEit takes after each Save or Append. If you would like SAVEit to pop up a dialog window letting you know that your file was saved, check off the "Popup 'File Saved' Dialog" option. This option is often helpful when debugging your scripts.

You also have the option to clear the following fields: File Path, File Name, If Exists, Place, Body, Creator, and File Type. (Note: the last two are only available on the Macintosh version.) Fields that are checked will clear after each Save or Append. Fields that are not checked retain their set values after each Save or Append.

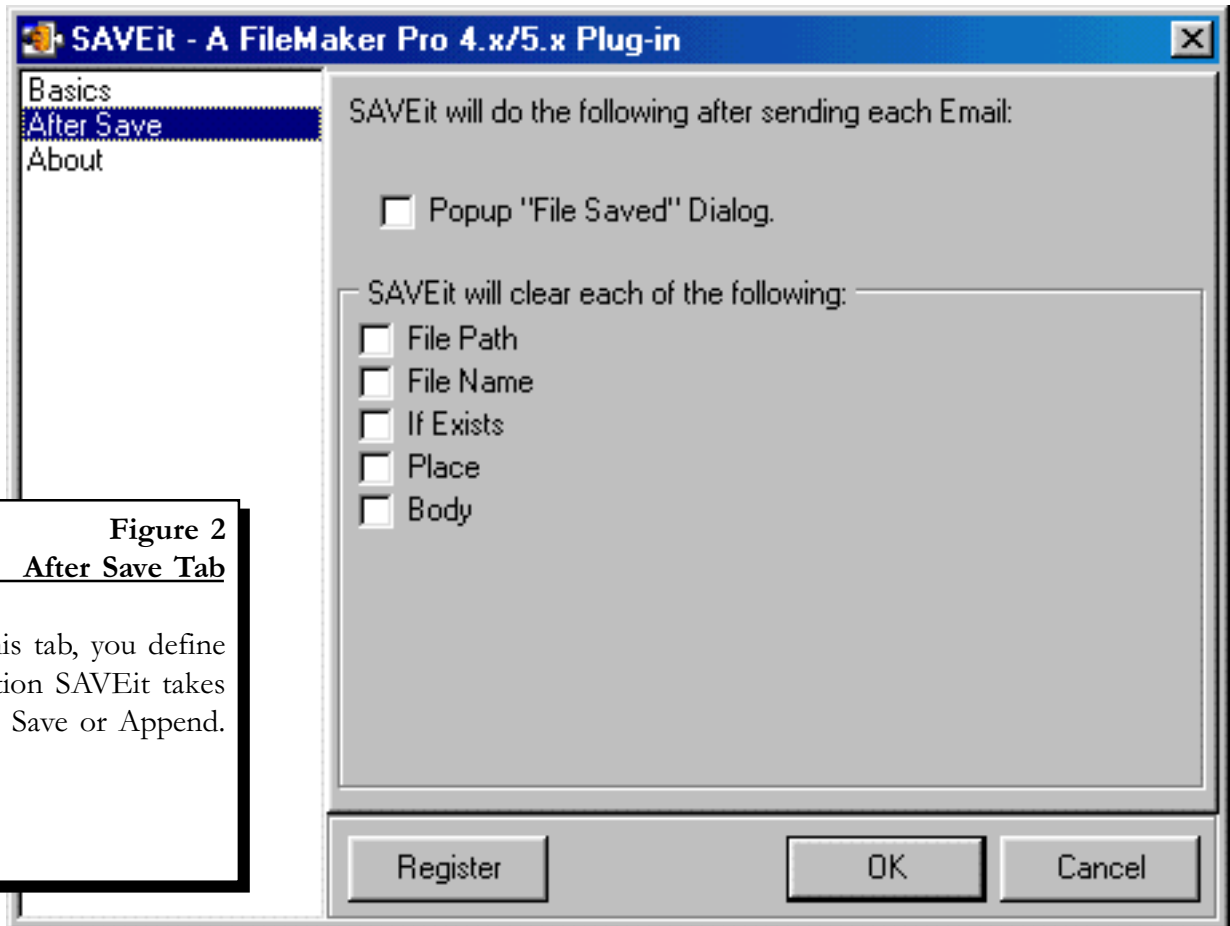


Figure 2
After Save Tab

On this tab, you define what action SAVEit takes after each Save or Append.

About

The About Tab simply reports which version of SAVEit that you are using. You can also use the "SAVE-Version" function to bring up the Configuration Dialog by passing the function the string "CONFIGURE" or the string "ABOUT". To make sure that you have the most recent version of SAVEit, please visit our website (<http://www.filemaker.comm-unity.net>).

How to Register

With version 1.2 of SAVEit and in future versions, you can register SAVEit from the Configuration Dialog (See Figure 3) once you have purchased it from our secure website or by ordering by phone. After you purchase SAVEit, we will send you a serial number to register your copy by using the "Register" button in the Configuration Dialog. We have also decided to include a register function so that developers can easily register SAVEit with their bound solutions once they have purchased a Developers license. For more information on purchasing SAVEit and other exciting plug-ins, visit our website (<http://www.filemaker.comm-unity.net>) and choose the "Purchase" link from the tool bar.

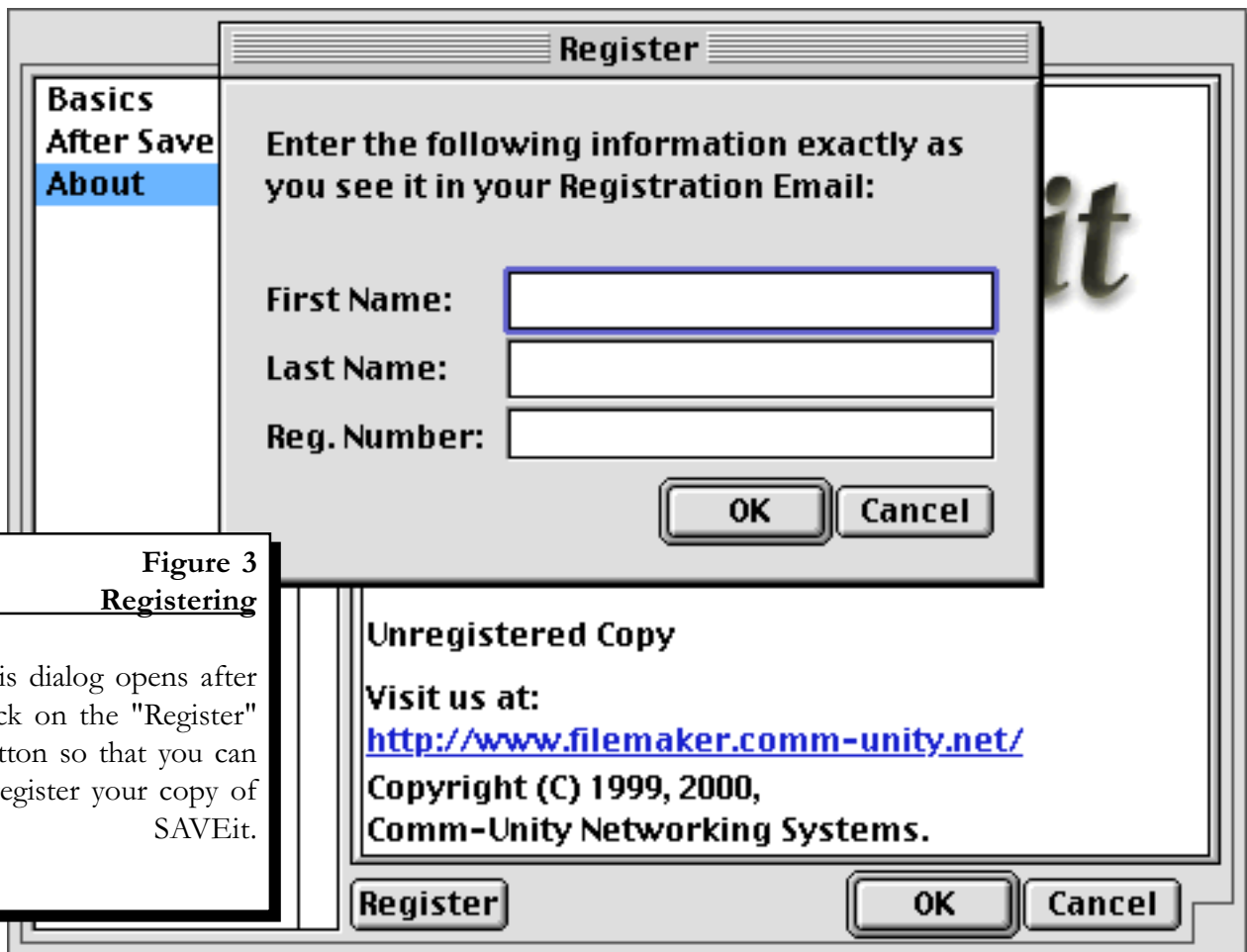


Figure 3
Registering

This dialog opens after you click on the "Register" button so that you can register your copy of SAVEit.

How to Use FileMaker Plug-ins

FileMaker Inc. introduced a very simple plug-in architecture when they released FileMaker Pro 4.0. Originally intended to aid only in complex calculations, the plug-in architecture took off in ways that FileMaker Inc. had not expected. Though there are now many different types of plug-ins, they all work the same basic way. By understanding the basics of plug-in interaction you will be able to understand how many developers approach plug-in development.

To start with, plug-ins are used by creating calculations. You can use calculations in many places within FileMaker including a calculation field, the "Set Field" script step, the "Paste Result" script step, as well as a text field that has a calculated value. There are a couple of other ways to create calculations, but these are the major avenues that are currently being used for plug-ins. The most commonly used avenue for plug-ins is the "Set Field" script step. It is easy to use, plus, the plug-in can report its actions in the field that you are setting.

FileMaker's External Plug-in architecture is as follows:

External ("FunctionNamePrefix-FunctionName", parameter)

For example if you take our Version function, which returns the version of SAVEit that you are using, it would look like this:

External ("SAVE-Version", parameter)

To actually use the function though, you will have to give it a parameter. Since the Version function does not need any special information, you can just set it to the empty string ("") in order to use the function:

External ("SAVE-Version", "")

The parameter of the plug-in architecture is where you put the information for the related function. For example, if you are setting your path using the SAVE-PathAssign function, it would look like this:

External ("SAVE-PathAssign", "C:\My SAVEit Files")

Or if you are using a text field or a global field named "Path" to hold your path value, it would look like this:

External ("SAVE-PathAssign", Path)

The difference between using a field to hold the value and putting the real value into the calculation is the quotation marks. This is because FileMaker will interpret anything not in quotes as a field in your database.

Though the calculation dialog box is limited in space, FileMaker has made it somewhat easier to deal with functions by using their groupings of functions in the top right corner of the calculation dialog. To view all of the plug-in functions currently installed, choose view by "External Functions". Once chosen, you should see the Web companion external functions, and if you have SAVEit installed you will see the SAVEit functions listed as well. Choosing to view by "External Functions" can greatly increase the ease of your script writing because the functions that you need are right there at your finger tips.

How to Use the SAVEit Plug-in

There are a few settings that you will need to set before saving each file from FileMaker Pro. To save one file from FileMaker you will need to set the file path, the file name, and the text you want to save. There are several ways of doing this. The easiest way to save a file with SAVEit is to use the SAVE-BodyAppend and SAVE-DlgSave functions which look like the following:

```
External("SAVE-BodyAssign", "Text to save") &
External("SAVE-DlgSave", "")
```

With this example, SAVEit will present you with a save dialog box that will allow you to navigate to the directory that you want to save to and specify the name of the file. In the above example, "Text to save" is some quoted text that will be saved to the file. You could specify a field in your database that has the text that you want to save.

Keep in mind that you can set several default values in the Configuration Dialog, which was explained earlier in this documentation.

You can create files dynamically by mixing functions such as SAVE-PathAssign, SAVE-FileNameAssign, SAVE-Save, and SAVE-Append with powerful FileMaker script steps such as the loop and navigational steps.

For example, you could export a found set of records using the following script as a guide:

Script Step	Script Parameters
Go to Record/Request/Page Specify:	First
Set Field	Field Button: Result Specify Button: External("SAVE-PathAssign", Path)
Loop	
Set Field	Field Button: Result Specify Button: External("SAVE-FileNameAssign", File Name) & External("SAVE-BodyAssign", Text) & External("SAVE-DlgSave", "")
Go to Record/Request/Page Specify:	Next Exit after last: Checked
End Loop	

This script first goes to the first record, then it sets the folder/directory that all of the files will be saved in. Then it starts a loop in which a file name is assigned and then the text in the "Text" field is saved. The script then goes to the next record and starts over at the beginning of the loop until all of the records have been saved.

For more examples of how to use SAVEit, please see the "SAVEit_Example" database included with this archive.

SAVEit External Function Reference

The following is a list of all available functions and a complete description of each. Note that there are several functions that can have default values set in the Configuration Dialog. See the Installation and Configuration section for a complete understanding of how the Configuration Dialog works. Functions with an asterisk ("*") are new with version 1.5.

SAVE-Version

This function returns the current version of SAVEit when the parameter string is empty (""). You can also use this function to display the configuration dialog. If you pass it the parameter "CONFIGURE", the configuration dialog will open. If you pass it the parameter "ABOUT", it will display the configuration dialog with the "About" tab as the first tab shown.

SAVE-Register

This function allows you to register your copy of SAVEit via a function rather than using the Configuration Dialog. It is mostly meant for developers so that they can register plug-ins for bound solutions. This function looks like the following:

```
External("SAVE-Register", "First Name|Last Name|Serial Number")
```

The parameter consists of your first name, last name, and serial number all separated by the pipe character ("|"). ("|" is created by typing shift-backslash.)

SAVE-PathAssign

The PathAssign function is where you set the full path of the file you will save. On Windows this will look like "C:\My Documents\". On Macintosh this will look like "Macintosh HD:My Documents:".

SAVE-FileNameAssign

Use SAVE-FileNameAssign to assign the name of the file that you will save. The parameter can be a string such as "SAVEit.txt" or a field in your database.

SAVE-BodyAssign*

Use the SAVE-BodyAssign function to assign the text "Body" of the file you will save. Note: If you use the SAVE-BodyAssign function twice, the second one will overwrite the first.

SAVE-BodyAppend*

The SAVE-BodyAppend function will add to the text "Body" of the file you will save assigned by SAVE-BodyAssign. If you have not set a body with SAVE-BodyAssign, SAVE-BodyAppend will assign the current value to the "Body" of the file. You can use SAVE-BodyAppend multiple times without overwriting previous settings, unlike the SAVE-BodyAppend function. Use this function if you need to save more than 64k of text. (The current versions of FileMaker Pro can only send 64k of text to a plug-in at one time.)

SAVE-Save

This function will save the text that you have assigned with SAVE-BodyAssign and SAVE-BodyAppend. The parameter should be "". Note that this function is affected by the settings of the SAVE-IfExistsAssign function as well as the related default settings in the Configuration Dialog.

SAVE-DlgSave

The SAVE-DlgSave function works exactly like the SAVE-Save function, except that instead of saving with the File Path and File Name you assigned, it allows you select a file with a dialog box. In other words, with this function you can navigate to the directory that you want to save your text to and name the file like a normal application.

SAVE-Append

The append function works exactly like the save function, except that it appends the "Body" text to the text file you specify rather than overwrite it. If the file that you are trying to append to does not exist, SAVEit will create the file with the current save parameters.

SAVE-DlgAppend

The SAVE-DlgAppend function works exactly like the SAVE-DlgSave function except that SAVEit will append the "Body" text to your text file that you select rather than overwrite the file.

SAVE-PathAcquire

This function allows you to use a Standard Open Dialog to navigate to a desired folder, and then returns the full path of the selected folder. Note: SAVE-PathAcquire only returns a valid path. It does not use the selected folder as the save File Path.

SAVE-IfExistsAssign

With this function, you can dynamically set whether SAVEit aborts the save, overwrites a file, appends, or gives the current save a unique name. The parameter of this function is either 1 to abort, 2 to overwrite, 3 to append, or 4 to create a unique name.

SAVE-PlaceAssign

This function allows you to dynamically set the number of places that SAVEit will use when creating unique file names. Below is a table showing what each setting will do.

Setting	Unique Numbers	Example
1	10	SAVEit1.txt
2	100	SAVEit01.txt
3	1000	SAVEit001.txt
4	10,000	SAVEit0001.txt

SAVE-MakeDir

The SAVE-MakeDir function creates a directory with the path that you specify. The path must be a full path. On Windows, this should look like "C:\My Documents\New Folder". On Macintosh, this should look like "Macintosh HD:New Folder:". Note that you can also set SAVEit to automatically create directories on the "Basics" tab of the Configuration Dialog.

Understanding Error Responses

Every function in SAVEit returns a response indicating the success or failure of that function. If the function is successful, it will return a response indicating that it set the value you were trying to set, or completed the task that needed to be completed. If however, the function is not successful, it will return an Error Response. This Error Response is in the form of:

ERROR: <Function Name>: <Error Description>

For example, if you forgot to set the "Body" of your text file using SAVE-BodyAssign, and you attempt to save the text using SAVE-Save, the Save function will return the following Error Response:

ERROR: Save: Cannot Save: Nothing assigned to body

Error responses always start with the word "ERROR" in all caps, followed by a colon, followed by the function that had the error, followed by a colon, followed by the actual error that occurred. You can use the various FileMaker Text Functions to extract the different parts of the Error Response for your own use.

Credits

Programming and documentation by Jake Traynham

Concept, web design, and documentation by Jesse Traynham

Contact Information

Email: SAVEit@comm-unity.net

Web: <http://www.filemaker.comm-unity.net>

Phone: 817-560-4226

You can write us at

Comm-Unity Networking Systems

8652 Hwy 80 West

Fort Worth, Texas 76116