shortcut™

FileMaker® 8
Calculation Functions

Steve Lane
Scott Love
Bob Bowers

The calculation dialog in FileMaker serves as a fundamental element in nearly all development activities. The calculation dialog allows developers easy access to the data fields in their solutions and to a complete function list. This shortcut presents a complete description of each calculation function, lists examples, and in many cases offers additional comments on usage. In this shortcut you will also find references to a companion book, **Special Edition Using FileMaker 8**, where you can find additional information on functions.

que®

www.quepublishing.com

Visit us on the Web: www.quepublishing.com

Pearson Education, Inc.
Rights and Contracts Department
One Lake Street
Upper Saddle River, NJ 07458
United States of America
Fax: (201)236-3290

ISBN 0-768-66834-4

First release, August 2006

Parameters: None

Data type returned: **Text**

Description:

Returns the registered name of the computer hosting the database file.

To change the registered name on a computer:

On Windows, the computer name is found on the Network Identification tab of the System Properties control panel. The Full Computer Name option displays the current registered name.

On Mac OS, the computer name is found within System Preferences, under the Sharing settings.

If a client connects to a file hosted by FileMaker Server, Get(HostName) returns the name of the server. The host name can be configured with the Server Administration tool. By default, FileMaker Server uses the system's name, but a custom name can be supplied instead.

We find it helpful in our practice to place a Get(HostName) display on a prominent layout within our solutions so that we can see during development whether or not we're working on a live version, a development version, or a scratch file on our local laptops.

Examples:

If the computer is named "Maturin"

```
Get(HostName)
```

results in Maturin.

Get(LastError)

Category: **Get**

Syntax: **Get (LastError)**

Parameters: None

Data type returned: **Number**

Get(LastError)

Description:

Returns the number of the error generated by the most recent script step. If there was no error, then Get (LastError) returns 0. Use this function in combination with Set Error Capture [On] to trap and handle errors raised in scripts.

A common source of bugs in scripts is not remembering that the Get (LastError) function returns the error code from only the most recently executed script step. For example, in this script

```
Perform Find
If (Get(ErrorCaptureState) = 1)
   Show Custom Dialog (Get (LastError))
End If
```

the Get (LastError) step returns the result of the execution of the If statement, not the error code generated by the Find step.

➔ *For a complete listing of error codes,* ***see*** *Chapter 11, "FileMaker Error Codes."*

➔ *For more discussion on debugging and troubleshooting,* ***see*** Special Edition Using FileMaker 8, *"Debugging and Troubleshooting."*

Note that if a script is running on the Mac OS and calls an AppleScript routine, any errors generated will also be passed through to and presented via this function.

Note too that if an error occurs in FileMaker while performing an SQL query, an SQLSTATE error will be returned by ODBC.

Examples:

Consider the following script:

```
Set Error Capture [On]
Print Setup[Restore]
SetVariable [$error; value: Get (LastError)]
```

If the user cancels out of the Print Setup dialog, Get (LastError) returns 1 (user canceled action). If the Print Setup step executes successfully, Get (LastError) returns 0.

Get(LastMessageChoice)

Category: **Get**

Syntax: **Get (LastMessageChoice)**

Parameters: None

Data type returned: **Number**

Description:

Returns a number corresponding to the button clicked as a result of the Show Custom Dialog script step.

Though it has a value of 1, the default button on a dialog is always on the far right side. For example, if there are three buttons, they will appear in 3-2-1 (Cancel, Maybe, OK) order.

Returns:

1 for the default button.

2 for the second button.

3 for the third button.

Examples:

For the following script step, where the default button is labeled OK, the second button is labeled Maybe, and the third button is labeled Cancel:

```
Show Custom Dialog ["test";"Proceed?"]
```

If the user chooses OK, Get (LastMessageChoice) returns 1.

If the user chooses Maybe, Get (LastMessageChoice) returns 2.

If the user chooses Cancel, Get (LastMessageChoice) returns 3.

You can then use an If() statement to handle each possibility appropriately.

Note that if a custom dialog has input fields, it is only when the user clicks the default right-most button will the data be inserted into the input field.

Get(LastODBCError)

Category: **Get**

Syntax: **Get (LastODBCError)**

Parameters: None

Data type returned: **Text**

Description:

Returns a string that shows the ODBC error state (SQLSTATE), as published by ODBC standards, based on ISO/IEF standards.

The ODBC error state is cleared at the time the next ODBC-related script step is performed. Anytime before that happens, you can check to see whether an ODBC error was generated.

By setting the Set Error Capture script step to On, you can suppress the error messages that a user sees during execution of a script that uses ODBC functions.

Examples:

When attempting to execute a SQL statement with an invalid field name, Get (LastODBCError) returns S0022.

If no error is encountered, Get (LastODBCError) returns 00000.

Get(LayoutAccess)

Category: **Get**

Syntax: **Get (LayoutAccess)**

Parameters: None

Data type returned: **Number**

Description:

Returns a number that represents the current user's record access privileges level for the current layout. Privileges are assigned in the Custom Layout Privileges dialog box.

The Get (LayoutAccess) function can be used to alert users of restricted privileges at the layout level. Note that Get (LayoutAccess) returns information about only the current layout. Record access privileges for any other layout are not represented.

Note also that Get (LayoutAccess) does not return information about whether or not the layout itself is accessible, but rather what access the user has to edit record data via the current layout.

The Get (RecordAccess) function evaluates record access privileges independent of the Get (LayoutAccess) function. To fully evaluate record access, evaluate the return values of both the Get (LayoutAccess) and Get (RecordAccess) functions.

Examples:

Function	Results
Get(LayoutAccess)	Returns 0 if the custom layout privileges of an account's privilege set allow "no access" to records via this layout. Returns 1 if the custom layout privileges of an account's privilege set allow "view only" access to records via this layout. Returns 2 if the custom layout privileges of an account's privilege set allow "modifiable" access to records via this layout.

Get(LayoutCount)

Category: **Get**

Syntax: **Get (LayoutCount)**

Parameters: None

Data type returned: **Number**

Description:

Returns the total number of layouts in the current file.

Get(LayoutCount) returns the total number of layouts within a file, including hidden layouts and layouts the user doesn't have privileges to see.

Examples:

Function	Results
Get(LayoutCount)	Returns 3 when there are three layouts in a database file.

Get(LayoutName)

Category: **Get**

Syntax: **Get (LayoutName)**

Parameters: None

Data type returned: **Text**

Description:

Returns the name of the layout currently displayed in the active window.

To change the name of a layout, in Layout mode, go to the Layouts menu, select the Layout Setup menu item, and then click the General tab. Layouts do not need to be uniquely named.

Examples:

Function	Results
Get(LayoutName)	Returns Data Entry when the Data Entry layout is displayed.
	Returns Invoice List when the Invoice List layout is displayed.