

## Define a hotkey to copy field name for layout objects in FileMaker

Let's help a few FileMaker developers using FileMaker a lot to edit layouts. Sometimes you may want to get the field name for a layout object. Whether you like to paste it to ChatGPT to use the names or you just need to know what field to use for a calculation, a hotkey to get the field name into the clipboard can be very useful.

To do this job, we need a little calculation to evaluate at runtime in the layout editor. This would be a Let statement to do four things:

First we need to run the copy command in FileMaker's Edit menu. To learn the command codes, please use [MenuBar.ListMenuCommands](#) to scan the menu bar. In the [MenuBar.RunMenuCommand](#) function our plugin will locate the menu item and ask it to perform the action. Please note that you can't run this while a script is performing as the menu command is then blocked. But you can of course use our [Schedule](#) functions to run it later and then also schedule another script to run a second later.

Once completed, we check the clipboard for something copied with [Clipboard.GetFileMakerData](#) function. Passing Auto to the function as mode makes life easier as the plugin will just check what is available in the clipboard. But we are only interested in layout objects, so we could also pass Layout there.

Next we search the XML with [XML.Query](#) function. The query goes down the path from the snippet to the layout, the object on the layout, then into the Field Object to get the name entry. Since you can select multiple entries, the result is a list and we ask the plugin with flag 2 to return as text list without the XML nodes.

Last we use [Clipboard.SetText](#) to put the list on the clipboard, so you can paste it somewhere else. Here is the full calculation:

```
Let([
copy = MBS( "MenuBar.RunMenuCommand"; 57634 );
xml = MBS( "Clipboard.GetFileMakerData"; "Auto" );
name = MBS("XML.Query"; xml; "/fmxmlsnippet/Layout/Object/FieldObj/Name"; "";
2);
r = MBS( "Clipboard.SetText"; name )
]; name)
```

Next we need to escape the quotes with backslashes. Then we can use for the [HotKey.SetEvaluate](#) call. To register the hotkey we call [Hotkey.Register](#) and pass whatever code you like to press. The example below defines option-control-command-c as key combination. Here is the script:

```
# registers a hotkey option-control-command-c to copy the names of fields
referenced by selected layout objects in the layout editor
Set Variable [ $Hotkey ; Value: MBS("Hotkey.Register"; "C"; "option control
command") ]
# set calculation to run
Set Variable [ $r ; Value: MBS("HotKey.SetEvaluate"; $Hotkey; "Let([
copy = MBS( \"MenuBar.RunMenuCommand\"; 57634 );
xml = MBS( \"Clipboard.GetFileMakerData\"; \"Auto\" );
name = MBS(\"XML.Query\"; xml; \"/fmxmlsnippet/Layout/Object/FieldObj/Name\";
\"\"; 2);
r = MBS( \"Clipboard.SetText\"; name )
]; name)\" ) ]
# make it permanent and go to the preferences file
Set Variable [ $r ; Value: MBS( \"HotKey.SetPermanent\"; $HotKey; 1 ) ]
```

Please try. You may just add this to a database as a script and run it to install the command. The example file Hotkeys.fmp12 is included with our plugin download and includes the script above to copy. Download here: [Hotkeys.zip](#) Have fun!