

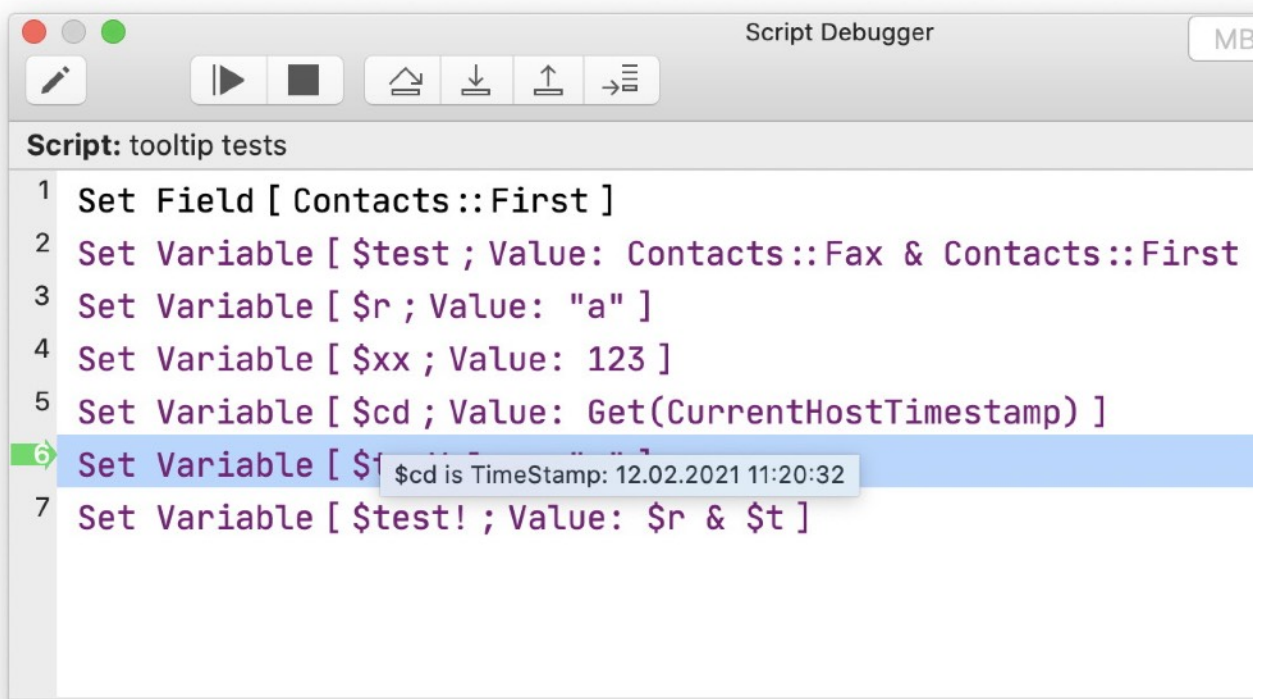
# Simplifying Debugging in FileMaker with MBS Plugin on macOS

Debugging can be one of the most challenging aspects of software development, especially in complex environments like FileMaker. Thankfully, the [MBS FileMaker Plugin](#) offers a range of features that make this process more manageable and efficient. Here are six ways the MBS Plugin can transform your debugging experience on macOS.

## 1. Tooltips for Easy Value Inspection

The MBS Plugin enhances the debugger by displaying tooltips that show the contents of variables and fields. When you hover over a variable, a tooltip appears, providing instant access to its value without the need to run additional scripts or create temporary fields for debugging purposes. This real-time inspection allows you to understand how data flows through your scripts, enabling quicker problem identification. Additionally, the context menu allows you to copy these values directly to your clipboard. This means you can easily reference or share them without interrupting your workflow.

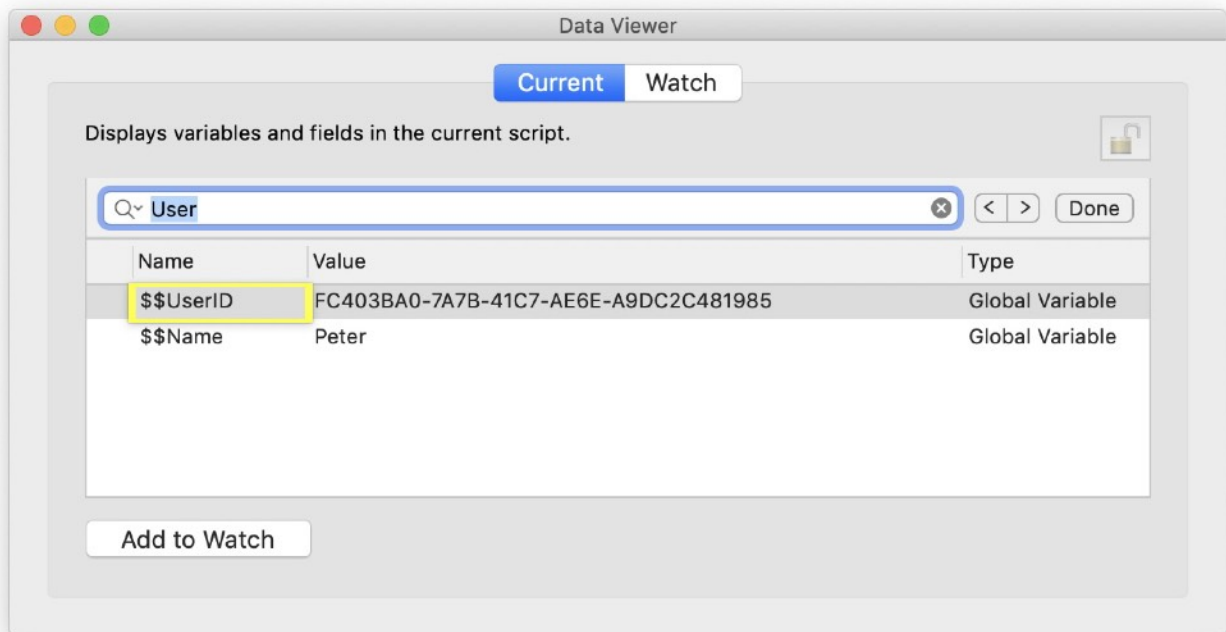
see also [Tooltips for Script Debugger](#)



## 2. Efficient Searching in the Data Viewer

Navigating through data can often feel cumbersome, particularly when dealing with large datasets. However, the MBS Plugin simplifies this with its powerful search feature in the Data Viewer. To initiate a search, first click on the title of the Data Viewer window to activate it. Then, ensure the list below is focused (indicated by a blue border). Once you've done this, using Command-F opens the search function, allowing you to type in specific values you're looking for. This targeted searching helps you quickly locate critical data points, significantly reducing the time you would spend scrolling through extensive lists.

see also [Search tables](#)



### 3. Detecting Unassigned Variables in Script Workspace

Typos and uninitialized variables can lead to frustrating bugs and unexpected behavior in your scripts. The MBS Plugin proactively addresses this issue by scanning your scripts for variables that are used but not assigned. When you attempt to use a variable without first giving it a value, it can result in errors that are often hard to trace back. By highlighting these discrepancies early on, the MBS Plugin helps you identify potential issues before they escalate into larger problems. This feature fosters good coding practices, encouraging developers to assign values to their variables before use. Consequently, it not only saves you debugging time but also helps maintain a cleaner codebase.

```
9      Variable setzen [ $index ; Wert: $index +1 ]
10     Script ausführen [ <Unbekannt> ]
11     Schleife (Ende)
12 Ende (wenn)
13 Variable setzen [ $test1 ; Wert: $test7 ]           $test7 not found!
14
```

Please note that you can declare variables to the plugin in various ways using comments. e.g. the script may start with a line like

```
# @variable $Parameter
```

to inform the plugin about a variable defines in a custom function.

see also [Check variables](#)

### 4. Code Folding for Improved Readability

Long scripts can become overwhelming, particularly when they contain many functions or repetitive logic. The MBS Plugin offers a code folding feature that allows you to collapse sections of your code, making it easier to manage and navigate through lengthy scripts. By hiding less relevant parts, you can focus on the sections that require your immediate attention, which is crucial when diagnosing issues or optimizing performance. This capability enables you to maintain clarity and structure in your code, facilitating better organization. Additionally, as your project grows, this feature allows for a more modular approach to debugging, allowing you to isolate sections without losing context.

see also [Code Folding](#)

## 5. Quick Script Searching in the Debugger

Finding specific code snippets in a large script can be time-consuming. Fortunately, the MBS Plugin allows you to search directly within the debugger, streamlining this process. By pressing Command-F, you activate the search field, where you can input your query. Once you hit return, the debugger highlights instances of your search term, allowing for swift identification of relevant code. Use Command-G to jump to the next search result, facilitating seamless navigation through your script. This feature is particularly useful when you're trying to locate the origin of an error or to verify the implementation of a specific logic. With this functionality, you can quickly move between multiple instances, enhancing your overall efficiency during the debugging process.

see also [Search script](#)

## 6. Go to Line Control for Easy Navigation

In collaborative development environments, pinpointing a specific line in your script can be crucial for effective communication. The MBS Plugin features a "go to line" control, enabling you to jump directly to a specified line number with ease. For instance, if you suspect an issue is on line 256, your colleague can navigate there quickly without having to scroll through potentially hundreds of lines. This feature not only saves time but also enhances collaboration, as team members can easily reference and discuss specific lines of code. This level of precision can make a significant difference during code reviews or troubleshooting sessions, where clear communication is key.

see also [Line numbers and GoTo](#)

## Bonus: Enhanced Readability with Font and Zoom Options

In addition to all these features, the MBS Plugin allows you to adjust the font size in both the debugger and Script Workspace for better readability. A larger font can reduce eye strain, especially during long debugging sessions. Moreover, the ability to zoom in or out using the plus and minus buttons in the top right corner enhances comfort, allowing you to adapt your workspace to your visual preferences. This flexibility ensures you can work comfortably, regardless of your screen size or environment. By customizing your workspace, you can create a more user-friendly debugging experience, making it easier to stay focused on the task at hand.

see also [Font setting for FileMaker Script Workspace and Calculations](#)

```
1 Set Variable [ $F ; Value: MBS( "SyntaxColoring.SetFontName"; "Noteworthy" ) ]
2 Set Variable [ $S ; Value: MBS( "SyntaxColoring.SetFontSize"; 15 ) ]
3 Set Variable [ $r ; Value: MBS( "ScriptWorkspace.StyledScriptText"; "StyledText" ) ]
4 Show Custom Dialog [ "Font and Size " ; $r ]
```

## Conclusion

Debugging in FileMaker can be a daunting task, but the MBS Plugin equips you with essential tools to simplify the process. With features like tooltips, efficient searching, unassigned variable detection, code folding, quick script searching, and easy navigation to specific lines, you can tackle debugging with confidence and ease. Embrace these enhancements to make your development process not just easier, but also more productive! By leveraging the capabilities of the MBS Plugin, you can streamline your debugging efforts and ultimately deliver more robust and reliable FileMaker applications.