

# Background

In 2005, FileMaker Inc. released a white paper which proposed possible methodologies and conventions under which a FileMaker developer could guide their development of a FileMaker solution. It was both a collaboration and contained contributions from a number of well-respected developers.

Here is the PDF: [FileMaker Development Conventions](#)

Interestingly, the document opens with a statement about FileMaker's flexibility and laments the "**constraints or concerns other developers experience**".

*FileMaker developers have the freedom to rapidly create and modify applications without having to deal with many of the constraints or concerns other developers experience. When employing an interactive approach to design, a tool like FileMaker lets you modify and extend functionality with little regard to dependencies elsewhere in the solutions. However, the more complex the solutions, the more difficult it is to maintain.*

While the trailing statement about "**extend functionality with little regard to dependencies**" is the dead on boon for developing in FileMaker, these two positions stand in contrast to each other - at least for me.

Having worked with other programming languages such as PHP and Java/Groovy and within those, a tightly typed Content Management System named Drupal, I've found that following a strict set of well-thought-out standards is far from a "constraint" or "concern".

Rather, the biggest benefit of a strict system is one that is highly freeing - in that I'm able to think about the solution at hand instead of the conventions being used. Strong documentation about the system makes the solution much more maintainable - which is one of the primary goals of all code.

Further down in the first page of the document, we find the following.

*No naming convention can address every conceivable development methodology and solution design. Furthermore, it would be counter-productive to try to establish such a rigid rule set.*

While the first sentence is true, I highly disagree with the second. When working with a team of developers, whether in the same location or spread across the world, it's the very standards upon which you establish, which promotes the high degree of collaboration which is possible.

For this reason, and based on my long-time experience with FileMaker, I am providing my own standards within this public forum. This semi-closed wiki is open to your comments and feedback. If you would like to become publicly involved in evolving the standards documented here, then please contact me by including information about why you would like to become involved and what you would like to contribute.

My goal is to refine these documented standards and provide as much documentation about them as possible. As was pointed out in the 2005 document, there are many developers using FileMaker, and no one standard has been adopted. It's **not my goal** to make this the single standard. The goal is to provide one which is well documented and prevent the beneficiary from having to do so himself or herself. This is leverage and it's a good thing!

In closing I'd like to comment on the following.

*The FileMaker developer base is extremely diverse. The approachability of the product makes it a clear choice for novice application developers, and the rapid application development nature of the tool is attractive to more seasoned developers, too. The amount of effort a developer needs to apply to naming conventions is different at each end of the spectrum.*

At the same time, the lack of any form of strict conventions is the very cause of frustration with "novice created solutions" and leads to development frustration for both novices and seasoned developers. This is apparent when either trying to understand their own developmental evolution covered over the course of a "learn-as-you-go development project" or starting out with zero idea about how one could have made their project more standardized from the beginning.

For this very reason, I present the genesis of the "ISO Standards for FileMaker Development".

Sincerely,

Matt Petrowsky