Naming Conventions

Element names

This is a quick visual overview of the various FileMaker elements such as tables, fields, layouts, scripts, functions and other objects. Each element of FileMaker has its own subsection below.

```
Tablename
dataFieldName
summaryCountField
GLOBALFIELD
CLIPBOARD (is a reserved temporary global field)
unstored_Calculation
id (is the primary key in all tables)
id_Tablename (is a foreign key)
idList_Tablename (is a pseudo-schema field for multi-key values related to solution schema)*
keys_Tablename (is a utility field for multi-key values used for UI specific functionality)
$variableName
$$GLOBAL.VARIABLE
FunctionalArea » Tablename
@Tablename (is the name for developer table occurrences)
Tablename: Layout Name (form|list)
portal.layoutObjectName
Script Name
Script Name ( with ; parameters )
CustomFunction
~PrivateCustomFunction
~locallyScopedLetVariable
$~privateVariableName
$$~PRIVATE.GLOBAL.VARIABLE
```

ⓘ *** Pseudo-Schema**

FileMaker Pro provides a unique feature called a multi-key - not to be confused with a compound, or more commonly, a concatenated key. Wherein a normalized data structure will typically use a singular key value for each and every record, FileMaker provides the ability to store multiple keys, hence a multi-key, list of values within a text field. Because of this feature, it is possible to create what filemakerstandards.org calls pseudo-schema. In the rest of the database world, this approach may not be considered valid, where isolated within the FileMaker world results in a feature which is heavily used by developers. While not suggested for normal use as part of your schema within a FileMaker solution, multi-key values for the purpose of maintaining relational structure amongst data would use the prefix of **idList_** as opposed to the more utilitarian, or more specifically for the UI, **keys_** prefix. See more info within Field naming.

ⓘ **Developer note**

The primary goal of these naming conventions is to create **highly readable code**. Concerns related to brevity are superseded by clarity and maintainability. These naming standards are also specific to FileMaker Client development. They do not address Web, Mobile or SQL based deployment. For more information related to these aspects of FileMaker then read about Solution development models

Table naming

Table occurrence naming

Field naming

Variable naming

Layout naming

Script naming

Custom function naming