

# Naming Conventions For FileMaker Pro



CLICKWORKS

© 2004-2023, ClickWorks BV  
Cogels-Osylei 36/3, 2600 Berchem  
www.clickworks.eu - info@clickworks.eu

## Version History

- Version 7.4, November 2023: further simplification and clarification.
- Version 7.3, July 2022: Simplifications, dropping a few conventions we don't use anymore.
- Version 7.2, December 2020: updated the logo.
- Version 7.1, November 2017: three char code prefixes in table names
- Version 7.0, November 2013: split checklist and guidelines, major review of script naming
- Version 6.1, February 2013: added value lists topic.
- Version 6.0, August 2012: guidelines Filemaker GO, indexing: define indexing for each field
- Version 5.5, May 2012: script comment line for context.
- Version 5.4, February 2012: custom functions for global constants + GEN, UTI, GLO, ... explained
- Version 5.3, September 2011: Layouts: always add developer layout
- Version 5.2, May 2011: altered TO color coding advice, minor changes in naming conventions.
- Version 5.1, June 2010: no more HUB table, minor changes in naming conventions.
- Version 5.0, March 2010: altered naming conventions, added FM11 features. Altered "Error Capture" advice.
- Version 4.0, September 2009: altered naming conventions, added FM10 features
- Version 3.0, September 2007: simplified naming conventions, added FM9 features
- Version 2.5, March 2007
- Version 2.0, October 2006
- Version 1.0, May 2004

## Index

Version History .....	2
Index .....	2
General .....	3
Custom Functions.....	3
TO names .....	3
Table names.....	3
Relationship Graph .....	3
Fields .....	4
Layout folders.....	4
Layout names.....	4
Script names .....	4
Script parameters/results .....	5
Valuelists .....	5

## General

Legacy projects		Stick with the existing naming convention. Don't start your own.
No spaces, use upper/lowercase		<i>_CMPid, isActive, nameFirst</i>
No special characters		The only save chars are tilde and underscore. Avoid dots in field and table names.
Mark items that are potentially obsolete	name + <b>_OBSOLETE</b>	
Externally referenced object names (hard coded)		Use (field) comments to document external usage. This will show up in tools like FMperception or Inspector.

## Custom Functions

Group by type	<b>Type.name</b>	<i>Date.YYYYMMDD</i>
App constants	<b>ZK. + name (uppercase)</b>	<i>ZK.RESULT.OK</i>

## TO names

Regular tables:	3 or 4 char code (uppercase)	<i>CMP, CON, PRO</i>
Globals table	<b>GLO</b>	

## Table names

Regular tables:	3 or 4 char code (uppercase) + full name	<i>CMP_Company</i>
Avoid special characters		

## Relationship Graph

Anchor Buoy in Squid formation		
Comment block as separator		
Underscore separates levels		<i>CMP_CON</i>
Sorting	<b>@ + {asc desc} + fieldName</b>	<i>CMP_CON@descNameLast</i>
Special attributes	<b>~ (tilde)</b>	<i>CMP_CON~manager</i>
External Data Source/ESS	<b>code+suffix</b>	<i>Examples : CMPe (external) or CMPws (webshop) or CMPs (SQL)</i>
Delete related records	Red color	
Allow creation...	Green color	
Creation + Delete	Orange color	

## Fields

Primary key	<b>__id</b>	
Foreign key	<b>_ + TO code + id</b>	<i>_CMPid</i>
Multikey	<b>Name + s</b>	<i>_CMPids</i>
Utility fields	<b>zzz + name</b>	<i>zzzModTS</i>
Calculated field	<b>Name + cu or cs</b>	<i>Age_cu (unstored), LastFirst_cs</i>
Summary field	<b>zzs + name or name_st or sc</b>	<i>zzsSales (if not part of business logic) or Sales_st (total), CountOf_sc (count)</i>
Constant	<b>zzk + name</b>	<i>zzk1</i>
Boolean field	<b>b + name</b>	<i>bIsActive, bHasParent, bAllowUpdate</i>

## Layout folders

Empty layouts for use in scripts	<b>developer</b>
UI layouts under construction	<b>under construction</b>
Layouts to check for removal	<b>cleanup</b>

## Layout names

UI layouts	no convention	
Developer layouts (data view)	Full Table Name	<i>CMP_Company, CON_Contact, ...</i>
Internal layouts (for scripts)	Three letter code of Anchor TO Name	<i>CMP, CON</i>

## Script names

General convention	<b>Entity.scriptName(paramList) = resultList</b>	<i>CIT.picker(Country)=zipCode</i>
Regular module, context independent	<b>TOname.scriptName</b>	<i>CMP.new if script can be called from any context.</i>
Regular module, context dependent	<b>TOname :scriptName</b>	<i>CMP:new if script only works in CMP context.</i>
Generic, cross-app	<b>GEN. + name</b>	<i>GEN.tryToCommit</i>
Global, app-wide	<b>GLO. + name</b>	<i>GLO.viewAs(type)</i>
Startup, shutdown	<b>INI. + name</b>	<i>INI.startup, INI.shutdown</i>
Utiity, cross-app	<b>UTI. + name</b>	<i>UTI.unlockStatusArea</i>
Security : account creation/delete	<b>SEC. + name</b>	<i>SEC.createResetAccount(name, pwd, privSet)</i>
Navigation	<b>NAV. + name</b>	<i>NAV.buildTabsThisLayout</i>
Scheduled Scripts on Server	<b>SRV.name or dnr suffix</b>	<i>SRV.RunNightly, SendMailsDNR</i>

## Script parameters/results

separator	; (semicolon)	<i>GEN.mailTo(Address ;body ;msg)</i>
optional	[name] (square brackets)	<i>INI.startup([bNoSplash])</i>
optional + 2 <sup>nd</sup> optional depending on first	[option1[ ;option2]]	<i>INV.createEdit([INvid[ ;INLid]])</i>
predefined set of values	{value1  value2}	<i>INV.print(set={one found})</i>
optional + predefined with default	[{value1  value2}=default]	<i>INV.print([set={one found}])</i>
predefined result example		<i>SEC.deleteAccount(name)={true false}</i>

## Valuelists

Generic hard-coded lists	<b>GEN.</b> + name	<i>GEN.1, GEN.OK, GEN.oneTwo</i>
Table-specific hard coded	TO name + . + name	<i>CMP.Status</i>
Field values	TO name + :: + FieldName	<i>CON::LastName</i>
Related field values	TO chain + :: + FieldName	<i>CMP_CON::LastName, ORD_CON~supplier ::LastName</i>
Multiple Fields	field1 + _ + field2	<i>CON::id_LastName</i>