

# Naming Conventions For FileMaker Pro



Version 7.0 November 2013  
© 2004-2013, ClickWorks bvba  
Cogels-Osylei 36/3, 2600 Berchem  
[www.clickworks.be](http://www.clickworks.be) - [info@clickworks.be](mailto:info@clickworks.be)

## Version History

- 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
Relationship Graph.....	3
Fields.....	3
Layout folders.....	4
Layout names.....	4
Script names.....	4
Script parameters/results.....	4
Valuelists.....	4

## General

No spaces, use upper/lowercase		<i>CMPid, bIsActive, nameFirst</i>
Short names		<i>CustPhone, SupZip</i>
Mark hard coded object names	name + <b>_dnr</b> (do not rename)	<i>itemList_dnr</i>
Mark items to delete	name + <b>_DELETE</b>	

## Custom Functions

Regular functions	<b>cf_</b> + name	<i>cf_DateToYYMMDD</i>
App constants	<b>zk_</b> + name (uppercase)	<i>zk_RESULT_OK</i>

## TO names

Regular tables:	3 or 4 char code (uppercase)	<i>CMP, CON, PRO</i>
Application tables	<b>_</b> + name	<i>_dummy</i>
Interface table	<b>_INF</b>	
Virtual List table	<b>_VIR</b>	
Application Library	<b>_LIB</b>	

## 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>
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>zzc + name</b>	<i>zzcAge</i>
Summary field	<b>zzs + name</b>	<i>zzsRecordCount</i>
Constant	<b>zzk + name</b>	<i>zzk1</i>
Boolean field	<b>is + name</b>	<i>isActive</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	TO code <i>CMP, CON, PRO, ...</i>

## Script names

General convention	entity.scriptName(paramList)=resultlist	<i>CIT.picker(Country)=zipCode</i>
Regular module	use TO name	<i>CMP.new</i>
Generic, cross-app	<b>GEN.</b> + name	<i>GEN.tryToCommit</i>
Global, app-wide	<b>GLO.</b> + name	<i>GLO.viewAs(type)</i>
Startup, shutdown	<b>INI.</b> + name	<i>INI.startup, INI.shutdown</i>
Utility, cross-app	<b>UTI.</b> + name	<i>UTI.unlockStatusArea</i>
Security : account creation/delete	<b>SEC.</b> + name	<i>SEC.createResetAccount(name, pwd, privSet)</i>
Navigation	<b>NAV.</b> + name	<i>NAV.buildTabsThisLayout</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>