

# Naming Prefixes

Many lists of objects in FileMaker give us the option to view those objects in alphabetical order. In calculation dialogs, the list of available functions is in alphabetical order (even when you choose to group functions by type), for example. Related objects will cluster together in alphabetically sorted lists if they begin with the same characters. By prefixing related objects with the same word(s), developers can suggest that they represent similar information or that they are inter-compatible in some way.

city fax firstName lastName mobile postCode phone state	<b>Bad</b>
addressCity addressPostCode addressState nameFirst nameLast phone phoneFax phoneMobile	<b>Good</b>

The general rule of thumb is to name in the direction of **general** or **wider scope** to more **specific scope**. This forces grouping when a list is sorted in alphabetical order.

Group prefix naming is similar in spirit to the "apps" variant of [Hungarian notation](#), but more verbose. Practically always, some objects in databases are more closely related to each other than others. For example, a field for storing a contact's first name has more in common with the field for last name than it has in common with a field for a phone number. For another, a \$windowName variable is related to a window, just like a \$windowWidth variable; but we'd like to distinguish those from \$objectName and \$objectWidth variables referring to properties of a different entity. The database doesn't know that, but developers and users need to.

Some conventions and functions on FileMakerStandards.org already follow this idea:

- The convention for naming [summary fields](#) is a prefix of "summary".
- The example UUID functions in the FileMakerStandards.org [GitHub repository](#) all begin with "UUID".
- The convention for privately-scoped entities, including [custom functions](#) and [Let\(\) variables](#), is to prefix them with a tilde "~".

The last example is a "sigil", a single-character type identifier. Sigils can be a more concise notation for entities that get used very frequently, but they can be confusing to developers who are unfamiliar with your conventions. Reserve sigils for naming entities critical to techniques you find yourself using very frequently in many unrelated contexts, techniques you can't live without. It may even be a good idea to avoid using sigil's then. For one example, the name-value [script parameter functions](#) all use a hash character "#" prefix. For another, this website suggests using UUIDs for [primary keys](#) in every table, but the example functions all still use the "UUID" prefix rather than a special-purpose single character.

## Known Prefixes

### Custom Functions

- Device\*: for functions dealing with the device and/or platform FileMaker is running in, e.g., DeviceIsServer, DeviceIsMac, DeviceIsPad, etc.
- Error\*: for error handling functions, e.g., ErrorDescription
- Key\*: for detecting and handling keystrokes, e.g., KeyCode, KeyShift, etc.
- UUID\*: for creating and manipulating universally unique identifiers, e.g., UUIDRandom, UUIDGetAsRFC4122
- Value\*: for functions dealing with return-delimited lists, e.g., ValueSort, ValueExists