OpenC

OpenC bv | Verlengde Poolseweg 16 | 4811 CL Breda
+31 76 524 46 09 | info@openc.nl | www.openc.nl

Guideline for naming conventions

bic: **INGBNL2A** | iban: **NL02 INGB 0007 3560 76**
kvk: 55001971 | btw: 851524217b01

Intro

People give all things a name. Naming things is a way to give order to the world. It also helps other people to know what you are referring to. In science, the classification of species is called a taxonomy.

In any software development project, giving things a proper name, is also of crucial importance, for much of the same reasons. Don't tax your taxonomy and make use or our experience.

Summary

At OpenC we apply a consistent and documented system for naming things in Studio (and in the rest of the Claris platform). Here is a summary of the rules of thumb we apply:

Studio name — make it short and sharp, for example: 'OpenC DEV'
Important when you have to manage and develop several Claris Studios and switch between them.

User names — include the department or organisation name, for example: 'Oscar Pensea of OpenC'
There is no description field for the user, at the time of this writing. So we include the name of the organisation in the User name. We also add the word 'of' because sometimes we user that account name on a View, like this: 'Hi Remco van Buren of OpenC!' That makes more sense then something like 'Hi OpenC | Remco van Buren!'

Group names — include the role, for example: 'OurClient, Users'
In Claris Studio (2025) there is no description field for a Group, at the time of this writing. That is why we put a description of the user role into the Group name field.

Hub names — keep the user in mind, for example: 'Your Clients'
If the user has Member rights, and logs into Studio, and then it must be immediately clear what the Hubs they have access to, mean to them. Examples of hub names: 'Sales funnel' of 'Project AI'.

View names — short and meaningful for the user, for example: 'Your Clients Form'
The tile of a View indicates what kind of View it is. For example, a Spreadsheet View is a green tile with a table-icon in it. A Form View is a mid-blue tile with a form-icon in it. But still, at OpenC we put the name of the View type into the description, at the end, without the word 'View'. Some examples:

- 'Clients Sheet' (Sheet as a shorter word for Spreadsheet View).
- 'Clients Form' (for Form View).
- 'Clients CV' (for Custom View).
- 'Clients Media' (for Gallery View).

Frame names
This functionality is yet to come, we'll add our guidelines at a later moment

Table names — Begin-capital and in plural, for example: 'Clients'
When a table is created in Studio, it gets a default name such as 'Spreadsheet16'. Change it, otherwise it will be hard later on the select the right table, in Studio Calculations or in Claris Connect flow steps.

Field names — apply the the underscore_method, but be aware! For example: 'client_name'
In general there are two conventions of naming tables and fields: CamelCase and underscore_method. At OpenC we have decided to use the lower case underscore_method. We have several reasons for this choice. With underscore_method:

- We do not have to use quotes In Studio Calculations.
- We have no field names in curly brackets, which improves readability in Claris Connect.
- The meaning of certain words in English is clear. Examples: after-school and after school, backyard and back yard, or undervalue and under value.
- CamelCase is a bit harder to read for some people than the underscore_method.

Object names on Views — Put the field name and the object type in the Object name, for example: 'client_full_name_shorttext'
Always give the Object a proper name. We include the field name, then one underscore and then the object type. Other example: 'client_list' or 'client_sheet'.

Display labels and placeholders – be clear to the user

- The Display Label instructs the user what to enter. It should be to the point, so that the user knows what to enter.
- The placeholder can further assist the user: it can contain an instruction such as 'Enter your name' or it can give an example.
- If it does not add value to the user we leave the placeholder empty. The user then has to read less text, and it gives a cleaner look to the screen.

Additional tips:

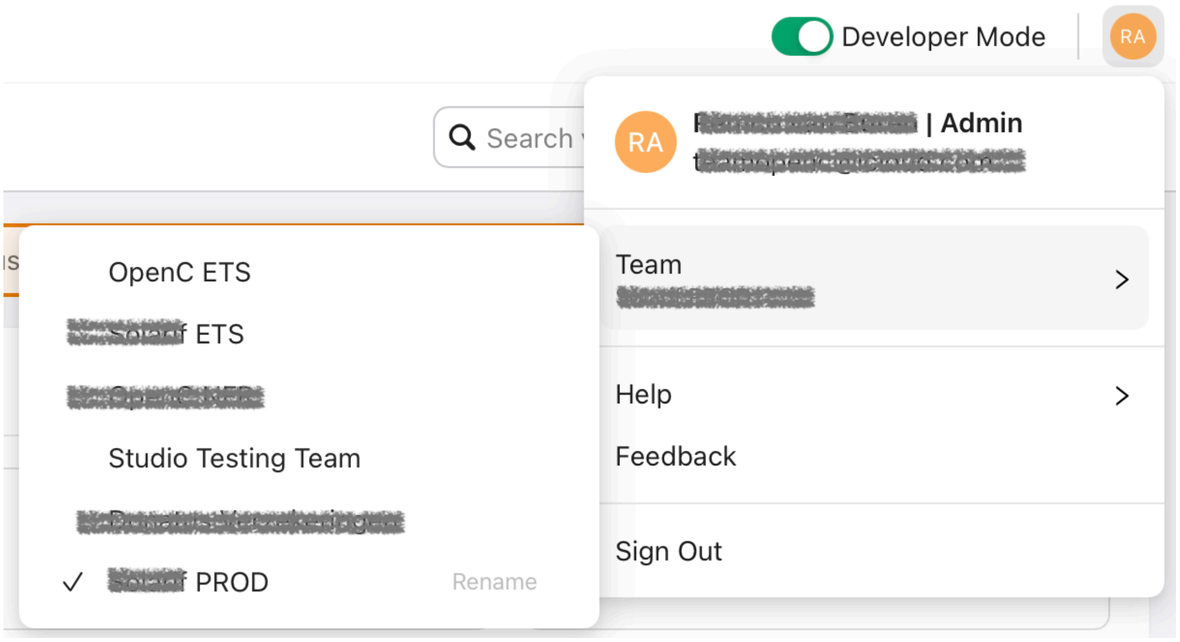
Numbers are numbered
In Studio we only use Number-fields, if we need a field for numeric calculations, such as add or divide. If we need a field to enter a statuscode, such as '0', '1' or '2', we never use a number-field. Because these values are not for calculating number. So for those, we only use a short text field.

Spreadsheet Objects in a View
Be aware that field names and also field descriptions cannot be labeled on screen, in certain objects, such as a Spreadsheet Object. These objects show the field name and description, for example 'client_name'. If that's a problem for your users, you have to rename the field to 'Client name' for example.

Guidelines for naming conventions

Naming the Studio Org: short and sharp

As a developer, you might log in with one Claris ID, and then have access to multiple Studio organisations (abbreviated to Studio Org). Typically you have a Claris Studio Org, for every client organisation or for every separate business unit of your company. Every Studio Org has its own Studio Org ID. You cannot change the ID. In developer mode, you can rename the Studio Org name. It is essential that you assure yourself that you are developing in the right Studio Org. So give every Studio Org a short distinctive name. If you switch to another Studio Org, you can immediately see to which one you have to go to.



User names, for example: ‘Oscar Pensea of OpenC’

A user name should of course contain the full name of the user.
At OpenC, we add the department, business unit or organisation name to it. Example: ’Remco van Buren of OpenC’. In this way it is clear where this user belongs. In Studio there is no separate field to describe where a particular user belongs.

  - We add the word ‘of’ because sometimes we user that account name on a View, like this: ‘Hi Remco van Buren of OpenC!’ That makes more sense than ‘Hi OpenC | Remco van Buren!’.
  - If you have users in a Studio that speak different languages, we tend to use a comma in stead of the word ‘of’. On a view this looks like: ‘Hi Remco van Buren, OpenC!’

- Username of a member or developer of the OpenC client: ‘Oscar Penrose’. Without the organisation name, because colleagues most of the time know each other. Or ‘Oscar Penrose | Marketing’, so with the department added.

There is also the Account name in Studio. In the case of a Claris ID, the account name is always the e-mail address with which the user logs in.

Group names, for example: OurClient, Users

You can add Studio Users to Groups. Naming groups depends on the user group. At OpenC, we name groups as follows: An abbreviation of the Company name, and then we add the type of user, in plural, for example:

- OpenC, Users
- OpenC, Devs
- OurClient, Clients
- TheirSupplierX, Users
- TheirSupplierY, Devs.

In Claris Studio (2025) there is no description field, so that is why we put this data into the Group name field.

Hub names: Keep them short and meaningful for the user

A hub contains a set of views that a user can select to enter a view on the underlying data. The only two rules of thumb that we use at OpenC, are:

1. We want give a Hub a name, with the user in mind.
   - If the user has Member rights, and logs into Studio, and then it must be immediately clear what Hubs mean to them. Examples of hub names: ‘Sales funnel’.
   - If the user has Developer rights, then we name the Hub for that developer differently. Example: ‘Sales funnel DEV’. DEV stands for ‘developer’. Why? Because a hub for a developer might contain the same Views as the hub for the Member, but added to that are the Spreadsheet Views.
2. We want to give it a short name. Why? Suppose you have two hubs with partly the same name then on-screen it can look like this:
   - Leads, Prospects and Active Clients, for dept X, will show on screen as: Leads, Prospects and Act…
   - Leads, Prospects and Active Clients, for dept Y, will also show on screen as: Leads, Prospects and Act…

In this case, the user does not know which Hub to choose.

View names: Same here, short and meaningful for the user

A View is a way of looking at the underlying data. You can give a view a name. A view name has to be unique, in a given Studio Org.

The tile of a View indicates what kind of view it is. For example a Spreadsheet View is a green tile with a table-icon in it, and a Form View is a dark blue tile with a form-icon in it. Having said that, at OpenC we put the name of the View type into the description, at the end, but without the word ‘View’.

Examples:
- ‘Clients Sheet’ (Sheet as a shorter word for Spreadsheet)
- ‘Clients Form’ (
- ‘Clients CV’ (for Custom View)
- ‘Clients Media’ (for Gallery View)

We do not use the word ‘View’ because that is what it is.

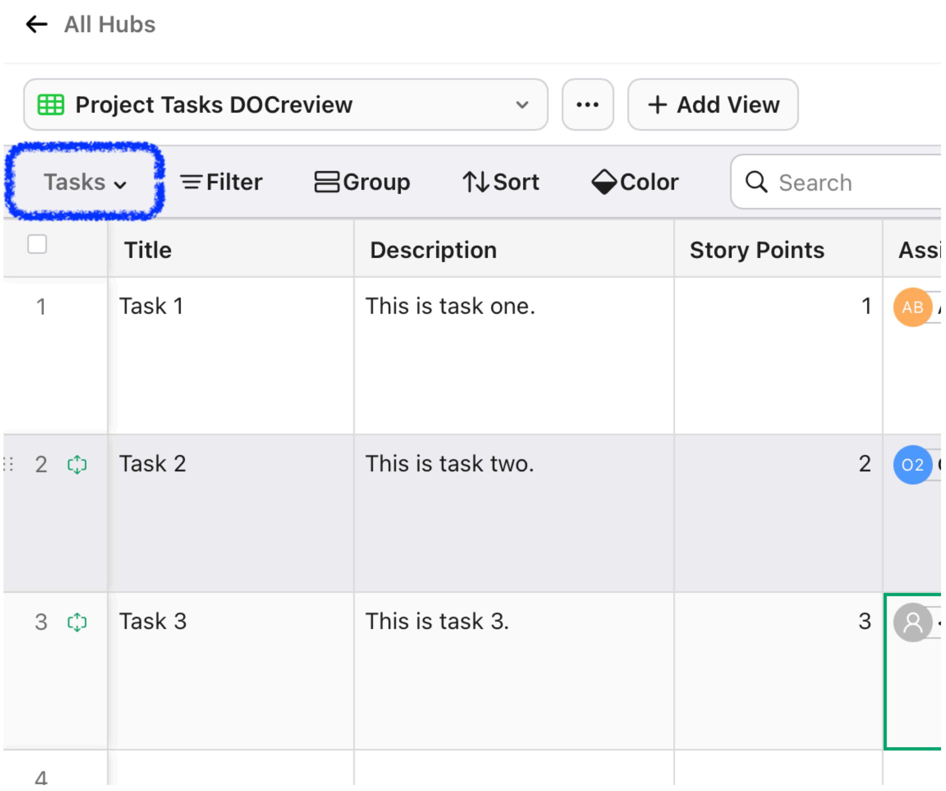
Frame or Pane names

To be added later.

Table names, for example: 'Clients'

When a table is created in Studio, it is given a default name such as 'Spreadsheet16'.

Just below the name of the Spreadsheet View, the table name is shown in the upper-left corner.



	Title	Description	Story Points	Assi
1	Task 1	This is task one.	1	AB
2	Task 2	This is task two.	2	O2
3	Task 3	This is task 3.	3	
4				

We immediately replace that default name with a unique and meaningful name, for instance 'Tasks' or 'Clients'.

For table names we use a capital beginletter and we always do it in plural. For example: Clients or Suppliers. Why plural? Because in a table almost always the data of more than one client is recorded. Why a capital letter? To separate a table name from a field name.

Field names: use underscore_method, preferably in English but be aware

In general there are two types of naming tables and fields: CamelCase and Underscore. At OpenC we have decided to use the lower case underscore method. We have several reasons for this choice:

1. In Studio calculations, you have to make use of quotes for a field name with spaces in it.
2. In Connect, spaces are allowed in field names, but it puts the field name in curly brackets, which makes readability somewhat harder.
3. If the Studio database is used by external systems, it is often better to avoid spaces between separate words.
4. Some words in English can have a (slightly or very) different meaning. Examples: after-school and after school, backyard and back yard, or undervalue and under value.
5. CamelCase is harder to read for some people than words in the the underscore_method.

At OpenC, we preferably name all fields a word in the English language. Why? Because English is the *de facto* standard in software development. Should you ever share your application with other developers, it makes it easier for them to understand the application, compared to a totally different language.

Also, we do not tend to name fields with non-descriptive field names such as '_a' or other non-descriptive terms. We always try to name a field with proper English words. Some of them, we do abbreviate, such as 'Nr' for Number or 'id' for 'identifier'.

Keep the following in mind: in the current version of Studio (2025), in some objects on a View, there is no label, so the naked field name is shown to the user. For example: 'client_name'. You would rather present to the user a different label such as 'Client Name'. You have to take that into account when you give a field their name.

The same applies to the short description of a field. At OpenC we only use that if the field name itself cannot clearly indicates what it stands for.

Then there is the following guideline we follow: We never give the unique identifier a general name such as 'primary_key', 'id', 'record_ID or 'UUID' (Universal Unique Identifier), for example. Better use: client_id, for the identifier in the Clients-table.

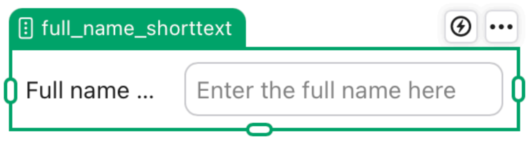
You can use numbers in Table names and Field names in Claris Studio.
It is possible to use specials characters in Table names and Field names, but we avoid them. They or not needed, and we want to avoid potential problems when other systems cannot make use of %, #, ', " etc. in table names and field names.

Do not use double underscores in field names. In certain external systems they can cause some trouble, such as not returning find results. Some AI-systems also appear to have trouble with double underscores.

Object names: give it the field name and the object type

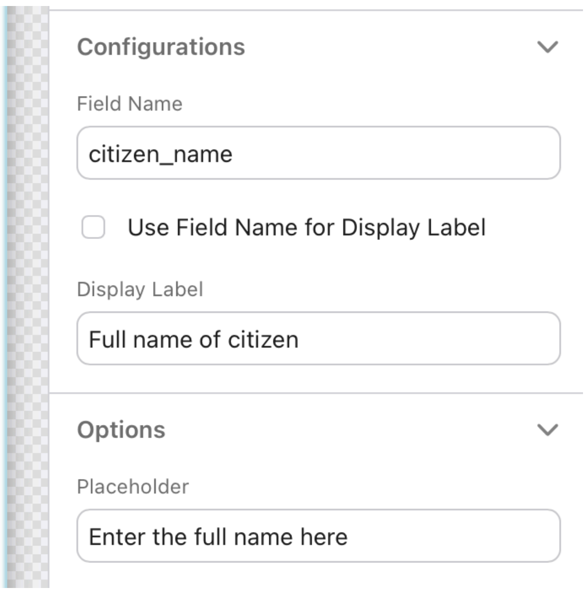
In the image you see a part of a Custom View with a Field Object. In developer mode. The Field Object is marked in green, with a green tab with the Object name in it. It is also a handle to move the object to another place. And a green square around it, with handles to change the size of the object.

Always give the Object a name. At OpenC, our naming convention is as the image shows: the field name and then underscore and then the type of object. In this case it is a ‘shorttext’-object. Our naming convention helps to identify what field we are talking about (full_name), and what kind of object type this is (shorttext).



full_name_shorttext

Full name ... Enter the full name here



Configurations

Field Name

citizen_name

Use Field Name for Display Label

Display Label

Full name of citizen

Options

Placeholder

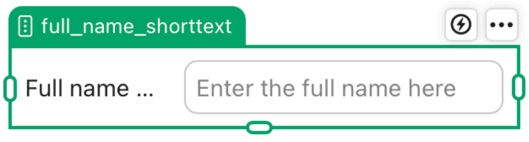
Enter the full name here

Display labels and Placeholders

The Display Label instructs the user what to enter. (Notice that in the image, the display label is only shown partly: ‘Full name …’. You can change the width or put the label right above the box, in stead of on the left, where it is currently located).

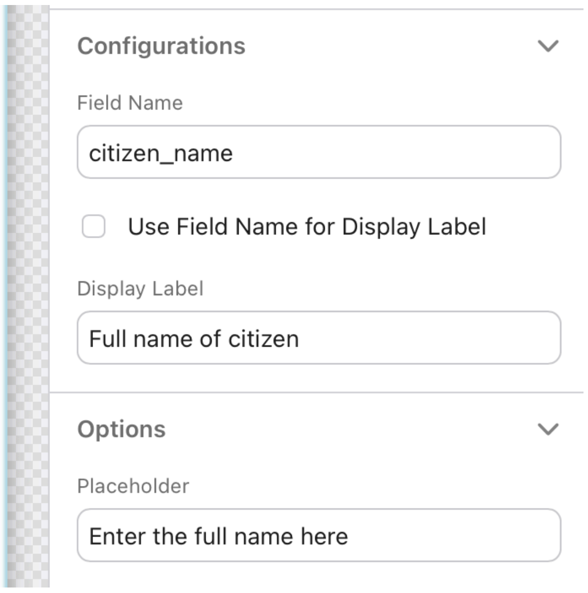
This display label should be very to the point, so that the user knows what to enter.

The placeholder can further assist the user: it can contain an instruction such as ‘Enter your name’ or it can give an example. Is it does not add value to the user we leave the placeholder empty. The user then has to read less text, and it gives a cleaner look to the screen.


full_name_shorttext

Full name ...

Enter the full name here


Configurations

Field Name: citizen_name

☐ Use Field Name for Display Label

Display Label: Full name of citizen

Options

Placeholder: Enter the full name here

Version history of this document

- 1.01 / 26 May 2025 / Replaced id-field paragraph by number-field paragraph